



ugr

Universidad
de **Granada**

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

Control de estrés en mayores

Aplicación para SmartWatch

Autor

Víctor José Rubia López (alumno)

Directores

María José Rodríguez Fórtiz (tutor1)

María Luisa Rodríguez Almendros (tutor2)



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, julio de 2022



Control de estrés en mayores

Aplicación para SmartWatch.

Autor

Víctor José Rubia López (alumno)

Directores

María José Rodríguez Fórtiz (tutor1)

María Luisa Rodríguez Almendros (tutor2)

Control de estrés en mayores: Aplicación para SmartWatch

Víctor José Rubia López (alumno)

Palabras clave: estrés, reloj inteligente, fotopletismografía, variación de frecuencia cardíaca, sistemas distribuidos, Machine Learning, limpieza de señal, metodologías ágiles, web, móvil, Android, WearOS, Jetpack Compose, Material Design 3.0.

Resumen

Cuando una persona realiza actividades instrumentales de la vida diaria, como el uso del transporte público, puede exponerse a ciertos niveles de estrés. Este estrés puede estar causado por factores internos a la persona, como la edad o la experiencia previa en el uso del medio de transporte, o por factores externos como accidentes, atascos o las características de la infraestructura que proporciona el consorcio de transporte a sus usuarios. Para mejorar la salud de estas personas, resulta necesario conocer qué factores hacen que aumente su nivel de estrés, de modo que se pueda tratar con un especialista. En ámbito clínico, se realizan tests para conocer el nivel de estrés de una persona, mientras que en los consorcios de transporte se realizan cuestionarios a los usuarios sin contemplar el estrés producido en el uso de sus servicios como un apartado a valorar y dejando a un lado que esto es un problema que afecta a la sociedad en la actualidad. Por esto, resulta necesario automatizar el proceso de recopilación de niveles de estrés en el transcurso de este tipo de actividad instrumental de la vida diaria. El objetivo de este trabajo fin de grado es el de investigar, analizar y desarrollar un entorno multisistema intercomunicado, para monitorizar el nivel de estrés de una persona en el transcurso de una actividad diaria como el uso del transporte público para sus desplazamientos. En el sistema intervienen un reloj inteligente, un dispositivo móvil y un sistema web multiusuario, con diferentes aplicaciones y servicios desarrollados. La evaluación del estrés se consigue a través de la obtención, filtrado y limpieza de una señal analógica de fotopletismografía (PPG) a través de un sensor ubicado en el reloj inteligente, mediante la que se puede obtener las características para la variación de frecuencia cardíaca y con la que podemos medir el nivel de estrés de un periodo de tiempo. Este sistema será capaz de predecir el nivel de estrés mediante técnicas de Machine Learning, teniendo un modelo entrenado con un conjunto de datos relacionado con un experimento de niveles de estrés. Los niveles de estrés en conjunción con el etiquetado que puede registrar el usuario en el reloj durante la realización de la actividad, ayudan a comprender los motivos por los cuales los usuarios sufren estrés en el desempeño de sus actividades diarias. La realización de este proyecto supone dar explicación a los aspectos que producen estrés al usuario en la realización de actividades de su vida diaria, dándole una explicación y pudiendo llevar un seguimiento más fidedigno para el tratamiento de su estrés. En el desarrollo de este proyecto se han utilizado diferentes tecnologías muy actuales para los dispositivos y plataformas que se tienen.

Monitoring elderly stress: SmartWatch App

Víctor José, Rubia López (student)

Keywords: stress, smartwatch, photoplethysmography, heart rate variability, distributed systems, Machine Learning, signal cleaning, agile methodologies, web, smartphone, Android, WearOS, Jetpack Compose, Material Design 3.0.

Abstract

When a person carries out instrumental activities of daily life, such as using public transport, he or she may be exposed to certain levels of stress. This stress can be caused by internal factors of the individual, such as age or previous experience in using means of transport, or by external factors such as accidents, traffic jams or due to features of the infrastructure provided by the transport consortium to its users. In order to ensure people's health, it is essential to be aware of the causes that increase their stress level, so that it can be treated by a specialist. In the clinical field, tests are carried out to find out a person's level of stress, while in the transport consortium, surveys are made to users without considering the stress produced by the use of their services as a section to be evaluated and leaving aside the fact that this is a problem that affects society today. For this reason, it becomes necessary to automate the process of collecting stress levels in the course of this type of instrumental activity of daily life. The aim of this final degree project is to research, analyse and develop an intercommunicated multisystem environment to monitor the stress level of a person in the course of a daily activity such as the use of public transport for commuting. The system involves a smartwatch, a smartphone and a multi-user web system, developing different applications and services. The stress assessment is achieved by obtaining, filtering and cleaning an analogue photoplethysmography (PPG) signal through a sensor located in the smartwatch, through which the characteristics for heart rate variation can be obtained and with which we can measure the stress level of a period of time. This system will be able to predict the stress level using Machine Learning techniques by having a trained model with a dataset related to a stress level experiment. The stress levels in conjunction with the labelling that the user can register on the watch during the activity help to understand the reasons why users suffer stress in the course of their daily activities. The implementation of this project involves explaining the aspects that cause stress to the user in the course of their daily life activities, giving them an explanation and being able to carry out a more reliable follow-up for the treatment of their stress. In the development of this project, several current technologies have been used for the devices and platforms that have been developed.

D. **María José Rodríguez Fórtiz (tutor1)**, Profesor del Departamento Lenguajes y Sistemas Informáticos de la Universidad de Granada.

D. **María Luisa Rodríguez Almendros (tutor2)**, Profesor del Departamento Lenguajes y Sistemas Informáticos de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado *Control de estrés en mayores, Aplicación para SmartWatch*, ha sido realizado bajo su supervisión por **Víctor José Rubia López (alumno)**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 8 de julio de 2022.

Los directores:

María José Rodríguez Fórtiz (tutor1)
(tutor2)

María Luisa Rodríguez Almendros

Agradecimientos

En primer lugar, a mi familia, mis amigos y mi pareja por inspirarme y apoyarme a llegar hasta aquí. En segundo lugar, a mis tutoras María José Rodríguez Fórtiz y a María Luisa Rodríguez Almendros, por brindarme la oportunidad de realizar un proyecto de esta envergadura y darme su apoyo y consejos cuando lo necesitaba.

Índice general

1. Introducción	1
1.1. Planteamiento	1
1.2. Objetivos	4
1.3. Estructura de la memoria	5
1.4. Planificación temporal	6
1.5. Presupuesto	9
1.5.1. Servidor Web	9
1.5.2. Reloj inteligente	11
1.5.3. Teléfono móvil	11
1.5.4. Dominio web	12
1.5.5. Conexión a internet	12
1.5.6. Coste humano	12
1.5.7. Presupuesto final	13
2. Estado del Arte y Tecnologías	15
2.1. Contexto del trabajo	15
2.2. Dominio del problema	17
2.3. Personas mayores en transportes públicos	17
2.3.1. Accesibilidad	18
2.3.2. Asequibilidad	19
2.3.3. Disponibilidad	19
2.4. Definición de estrés	19
2.5. Métodos para la medición del estrés	20
2.6. Sensores para medir el estrés	21
2.7. Detección de estrés mediante sensor PPG	22
2.8. Trabajos previos sobre tratamiento de señales y datos	24
2.9. Metodologías que se pueden utilizar	26
2.9.1. Metodologías de desarrollo software	26
2.9.2. Patrones arquitectónicos de diseño software	26
2.9.3. Diseño en Android	32
2.10. Tecnologías que se pueden utilizar	32
2.10.1. Tecnologías para desarrollar aplicaciones basadas en Web	32
2.10.2. Tecnologías para desarrollar aplicaciones para <i>smartphone</i> Android	35

2.10.3. Tecnologías para desarrollar aplicaciones para <i>smartwatch</i> WearOS	37
2.11. Trabajos relacionados	37
2.12. Aplicaciones similares	39
3. Propuesta	41
3.1. Metodología de trabajo	41
3.1.1. Marco de trabajo	41
3.1.2. Métodos de investigación	41
3.1.3. Descripción del alcance del sistema	42
3.1.4. Recursos	43
3.1.5. Diagrama de arquitectura	44
3.2. Plan de iteraciones del proyecto	46
3.2.1. Lista inicial del producto (Product Backlog)	46
3.2.2. Cálculo de velocidad	50
3.2.3. Descripción de las entregas	50
3.3. Primera iteración	51
3.3.1. Tarjetas de las Historias de Usuario	51
3.3.2. Descomposición de las H.U. en tareas de desarrollo	55
3.3.3. Planificación temporal de la primera iteración	59
3.3.4. Realización de las tareas de desarrollo	61
3.3.5. Retrospectiva de la iteración	88
3.4. Segunda iteración	90
3.4.1. Tarjetas de las Historias de Usuario	90
3.4.2. Descomposición de las H.U. en tareas de desarrollo	93
3.4.3. Planificación temporal de la segunda iteración	97
3.4.4. Realización de las tareas de desarrollo	99
3.4.5. Retrospectiva de la iteración	128
3.5. Tercera iteración	129
3.5.1. Tarjetas de las Historias de Usuario	129
3.5.2. Descomposición de las H.U. en tareas de desarrollo	132
3.5.3. Planificación temporal de la tercera iteración	136
3.5.4. Realización de las tareas de desarrollo	138
3.5.5. Retrospectiva de la iteración	181
3.6. Cuarta iteración	182
3.6.1. Tarjetas de las Historias de Usuario	182
3.6.2. Descomposición de las H.U. en tareas de desarrollo	187
3.6.3. Planificación temporal de la cuarta iteración	192
3.6.4. Realización de las tareas de desarrollo	194
3.6.5. Retrospectiva de la iteración	197
4. Evaluación y validación	205
4.1. Sujeto 1	205
4.2. Sujeto 2	208

5. Conclusiones y Trabajos Futuros	213
5.1. Conclusiones	213
5.2. Trabajos futuros	215
5.2.1. Aplicación Web	215
5.2.2. Aplicación Móvil	216
5.2.3. Aplicación Reloj Inteligente	216
5.2.4. Modelo de predicción y características obtenidas	216
5.3. Valoración personal	217
A. Manual de instalación	219
A.1. Sistema web	219
A.2. Aplicación móvil	220
A.3. Aplicación reloj inteligente	220
B. Manual de usuario	223
B.1. Aplicación móvil	223
B.1.1. Identificación del usuario	224
B.1.2. Pantalla de usuario identificado	225
B.1.3. Recuperar contraseña	226
B.1.4. Cerrar sesión	227
B.2. Plataforma Web	227
B.2.1. Registro de terapeuta	227
B.2.2. Identificación de terapeuta y recuperación de contraseña	230
B.2.3. Gestión de usuarios	231
B.2.4. Gestión de actividades	239
B.3. Reloj inteligente	245
B.3.1. Iniciar actividad	247
B.3.2. Registrar etiquetas	248
B.3.3. Detener una actividad en transcurso	250
C. Documentacion	251

Índice de figuras

1.1. Diagrama de Gantt que describe la planificación del proyecto.	8
2.1. Predicción de ingresos del mercado de dispositivos <i>wearables</i> médicos en todo el mundo en 2021 y 2026. (Statista, 2021a)	23
2.2. Señal PPG característica. (Mohan y col., 2016)	25
2.3. Primeros 4 minutos de una medición real realizada con el <i>smartwatch</i> cedido para este proyecto.	25
2.4. Principales patrones arquitectónicos de diseño software en Android. (Dang, 2020)	30
2.5. Diagrama de la arquitectura de una aplicación Android típica. (“Guide to app architecture”, s.f.)	31
2.6. El proceso general del método que proponen en el proyecto desarrollado por el IITP (Heo y col., 2021)	38
2.7. Dispositivos y Aplicaciones de medición de estrés	40
3.1. Mobvoi TicWatch Pro	44
3.2. Diagrama de Arquitectura del Sistema	44
3.3. Diagrama de Gantt para la planificación de las tareas de la iteración 1	60
3.4. Página inicial de la aplicación web	63
3.5. Página de registro de la aplicación web	64
3.6. Página de identificación de la aplicación web	65
3.7. Página del panel de control para gestión de usuarios	66
3.8. Modal para añadir un usuario dentro de la sección de gestión de usuarios	67
3.9. Pantalla inicio de sesión en Android	68
3.10. Pantalla de usuarios mayores identificados en la aplicación Android	69
3.11. Logotipo del proyecto	69
3.12. Esquema de tablas generadas en la Base de Datos por RoR en la iteración 1	73
3.13. Página principal del sistema web	74
3.14. Página de registro en el sistema web	74
3.15. Página de identificación en el sitio web	75
3.16. Página de gestión de usuarios en el panel de control del sitio web	75
3.17. Diálogo modal para registrar un usuario mayor en la plataforma	76
3.18. Correo que se envía al usuario mayor para que conozca sus datos de acceso	76

3.19. Tests realizados sobre la Aplicación Web RoR en la iteración 1	77
3.20. Diagrama de paquetes para la aplicación Android en la iteración 1.	81
3.21. Resultados de implementación en Android.	84
3.22. Diagrama de clases correspondiente al paquete <i>data</i> de la aplicación Android en la iteración 1	85
3.23. Diagrama de clases correspondiente al paquete <i>domain</i> de la aplicación Android en la iteración 1	86
3.24. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación Android en la iteración 1	87
3.25. Tests Unitarios y de Componentes de la Aplicación Android en la iteración 1	88
3.26. Diagrama de Gantt para la planificación de las tareas de la iteración w . .	98
3.27. Bocetado y navegabilidad de la sección de gestión de actividades de usuarios mayores en la plataforma web	100
3.28. Bocetado y navegabilidad en una primera aproximación al desarrollo de la aplicación WearOS en la iteración 2	102
3.29. Esquema de tablas generadas en la Base de Datos por RoR en la iteración	2105
3.30. Implementación de los bocetos del sistema web de la iteración 2	106
3.31. Tests realizados sobre la Aplicación Web RoR en la iteración 2	107
3.32. Prototipado de alta resolución con Figma de la aplicación WearOS en la iteración 2	110
3.33. Diagrama de comunicación entre sistemas en la iteración 2	115
3.34. Diagrama de paquetes de la aplicación wearOS en la iteración 2	117
3.35. Diagrama de clases correspondiente al paquete <i>data</i> de la aplicación wearOS en la iteración 2 (Parte 1)	118
3.36. Diagrama de clases correspondiente al paquete <i>data</i> de la aplicación wearOS en la iteración 2 (Parte 2)	119
3.37. Diagrama de clases correspondiente al paquete <i>data</i> de la aplicación wearOS en la iteración 2 (Parte 3)	120
3.38. Diagrama de clases correspondiente al paquete <i>domain</i> de la aplicación wearOS en la iteración 2	121
3.39. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación wearOS en la iteración 2 (Parte 1)	122
3.40. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación wearOS en la iteración 2 (Parte 2)	123
3.41. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación wearOS en la iteración 2 (Parte 3)	124
3.42. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación wearOS en la iteración 2 (Parte 4)	125
3.43. Tests Unitarios y de Componentes de la Aplicación wearOS en la iteración	2126
3.44. Diagrama de Gantt para la planificación de las tareas de la iteración 3 . .	137
3.45. Nuevo bocetado de la sección de visualización de actividades para un usuario en el sistema web	138

3.46. Resultado de diseño del correo de recordatorio de contraseña por parte del usuario mayor.	139
3.47. Diseño de alta resolución con Figma de la aplicación WearOS en la iteración 3	140
3.48. Pantalla de recuperación de contraseña en Android	141
3.49. Diseño de la navegabilidad de la aplicación Android en la iteración 3	142
3.50. Resultados de implementación en Android de los bocetos de la iteración 3.	143
3.51. Diagrama de paquetes para la aplicación Android	144
3.52. Diagrama de clases correspondiente al paquete <i>data</i> de la aplicación Android en la iteración 3	145
3.53. Diagrama de clases correspondiente al paquete <i>domain</i> de la aplicación Android en la iteración 3	146
3.54. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación Android en la iteración 3	147
3.55. Tests Unitarios y de Componentes de la Aplicación Android en la iteración 3	148
3.56. Representación gráfica de los datos obtenidos en crudo	150
3.57. Representación gráfica de la señal de PPG obtenida en un experimento	151
3.58. Filtrado de paso de banda mediante Butterworth de la señal PPG	153
3.59. Filtrado de paso de banda mediante Butterworth de la señal PPG	154
3.60. Cantidad de datos pertenecientes a cada clase del conjunto de datos.	156
3.61. Matriz de correlación de Pearson para comprobar la dependencia de las variables.	157
3.62. Matriz de correlación de Pearson tras aportar las variables que podemos obtener.	158
3.63. Gráficos de dispersión de algunas variables	159
3.64. Matrices de confusión de los modelos generados por distintos algoritmos de clasificación.	161
3.65. Esquema de tablas generadas en la Base de Datos por RoR en la iteración 3	164
3.66. Detalle de una actividad en la plataforma web con su gráfico de nivel de estrés.	166
3.67. Tests realizados sobre la Aplicación Web RoR en la iteración 3	167
3.68. Diagrama de comunicación entre sistemas en la iteración 3	168
3.69. Diagrama de paquetes de la aplicación wearOS en la iteración 3	169
3.70. Diagrama de clases correspondiente al paquete <i>data</i> de la aplicación wearOS en la iteración 3 (Parte 1)	170
3.71. Diagrama de clases correspondiente al paquete <i>data</i> de la aplicación wearOS en la iteración 3 (Parte 2)	171
3.72. Diagrama de clases correspondiente al paquete <i>data</i> de la aplicación wearOS en la iteración 3 (Parte 3)	172
3.73. Diagrama de clases correspondiente al paquete <i>domain</i> de la aplicación wearOS en la iteración 3	173
3.74. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación wearOS en la iteración 3 (Parte 1)	174

3.75. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación wearOS en la iteración 3 (Parte 2)	175
3.76. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación wearOS en la iteración 3 (Parte 3)	176
3.77. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación wearOS en la iteración 3 (Parte 4)	177
3.78. Diagrama de clases correspondiente al paquete <i>presentation</i> de la aplicación wearOS en la iteración 3 (Parte 5)	178
3.79. Tests Unitarios y de Componentes de la Aplicación wearOS en la iteración 3	179
3.80. Diagrama de Gantt para la planificación de las tareas de la iteración 4 . . .	193
3.81. Nuevos bocetos del sistema web en la iteración 4	199
3.82. Navegación del sistema web en la iteración 4	200
3.83. Bocetado de alta resolución del indicador de pérdida de conexión.	201
3.84. Diálogo modal para editar y visualizar un usuario mayor en la plataforma	201
3.85. Indicador para mostrar que una actividad no ha sido visualizada aún por el terapeuta	202
3.86. Correos electrónicos implementados en esta iteración	203
4.1. Gráfico de los niveles de estrés acumulados en el transcurso de la actividad por el Sujeto 1.	206
4.2. Tabla con las etiquetas indicadas por el sujeto 1 en el transcurso de la actividad.	207
4.3. Gráfico de los niveles de estrés acumulados en el transcurso de la actividad por el Sujeto 2.	209
4.4. Tabla con las etiquetas indicadas por el sujeto 2 en el transcurso de la actividad.	210
B.1. Pantalla inicial de un teléfono Android.	223
B.2. Correo electrónico recibido tras el registro.	224
B.3. Pantalla para la identificación de un usuario.	224
B.4. Pantalla tras el inicio de sesión del usuario con conexión correcta al reloj inteligente.	225
B.5. Pantalla tras el inicio de sesión del usuario sin conexión al reloj inteligente.	225
B.6. Pantalla de recuperación de contraseña.	226
B.7. Correo recibido tras la solicitud de contraseña.	226
B.8. Pantalla de usuario identificado para cerrar sesión.	227
B.9. Pantalla inicial de la plataforma web.	228
B.10. Pantalla de registro de la plataforma web.	229
B.11. Correo de verificación de cuenta para el terapeuta. Para poder empezar a usar la plataforma es necesario pulsar sobre el botón de "Verificar Cuenta"	229
B.12. Pantalla inicial de la plataforma web.	230
B.13. Pantalla para identificarse en la plataforma web.	231
B.14. Acceso a panel de control desde la pantalla de inicio.	232
B.15. Panel de control acceso a gestión de usuarios.	233

B.16. Panel de control sección de gestión de usuarios.	234
B.17. Gestión de usuarios registrar un usuario.	235
B.18. Formulario para introducir información para registrar un usuario.	236
B.19. Gestión de usuarios editar un usuario.	237
B.20. Formulario para editar la información de un usuario registrado.	238
B.21. Eliminar un usuario registrado.	239
B.22. Sección de gestión de actividades del panel de control.	240
B.23. Sección de gestión de actividades del panel de control.	241
B.24. Página de visualización de las actividades de un usuario.	242
B.25. Borrado de actividades registradas en el sistema para un usuario.	243
B.26. Listado de actividades forma de ver los detalles.	244
B.27. Página de visualización de los detalles de una actividad realizada por un usuario.	245
B.28. Pantalla tras el inicio de sesión del usuario sin conexión al reloj inteligente.	246
B.29. Pantalla de aviso de usuario no identificado en el teléfono.	246
B.30. Pantalla de aviso cuando el reloj no se encuentra conectado al móvil.	246
B.31. Pantalla de aviso cuando el modelo de reloj no es compatible con la apli- cación.	246
B.32. Pantalla inicial para el inicio de la medición de una actividad.	247
B.33. Pantalla de selección del tipo de actividad de una lista.	247
B.34. Pantalla de aviso del comienzo de la medición para la actividad iniciada.	247
B.35. Pantalla con un menú para parar la actividad y para registrar etiquetas.	247
B.36. Pantalla de menú de medición de actividad.	249
B.37. Pantalla con el listado de etiquetas pertenecientes a la categoría Estados.	249
B.38. Pantalla con una etiqueta seleccionada del listado.	249
B.39. Botón al final del listado de etiquetas para confirmar la selección.	249
B.40. Pantalla de medición donde se muestra el botón para detener la medición de la actividad.	250

Índice de cuadros

2.1. Características de la VFC más relevantes para predecir el nivel de estrés	24
2.2. Comparativa de las características de metodologías de desarrollo software más comunes. (Saeed y col., 2019)	27
2.3. Comparativa de las fortalezas de metodologías de desarrollo software más comunes. (Saeed y col., 2019)	27
2.4. Comparativa de las debilidades de metodologías de desarrollo software más comunes. (Saeed y col., 2019)	28
2.5. Características de las tres de arquitecturas de desarrollo software más extendidas en desarrollo en Android. (Dang, 2020)	29
2.6. Comparativa de los frameworks de desarrollo de aplicaciones Android más relevantes	36
3.1. Product Backlog del proyecto	50
3.2. Velocidad del desarrollador del proyecto	50
3.3. Descripción de las entregas del proyecto	51
3.4. Comparación entre los repositorios de estas librerías de gestión de usuarios para RoR.	70
3.5. Puntos de entrada de la API en la iteración 1	72
3.6. Resultados de realizar las pruebas de aceptación de RoR para la iteración 1	79
3.7. Resultados de realizar las pruebas de aceptación de Android para la iteración 1	89
3.8. Puntos de entrada de la API en la iteración 2	104
3.9. Resultados de realizar las pruebas de aceptación de la iteración 2	128
3.10. Características obtenidas del procesamiento de los segmentos de señal filtrados	155
3.11. Variables con una correlación de Pearson positiva respecto a la condición de estrés	156
3.12. Puntos de entrada de la API en la iteración 2	165
3.13. Resultados de realizar las pruebas de aceptación de la iteración 3	181
3.14. Resultados de realizar las pruebas de aceptación de la iteración 4	197
4.1. Perfil de María del Carmen, sujeto 1 de nuestra prueba.	206
4.2. Perfil de Juan Carlos, sujeto 2 de nuestra prueba.	209

Capítulo 1

Introducción

1.1. Planteamiento

Garantizar la movilidad a cualquier persona es uno de los retos y objetivos de la Accesibilidad Universal; se trata, en suma, de un factor clave para el objetivo de la plena participación y del logro de un medio físico normalizado, accesible a todos. A lo largo de nuestro día, nos enfrentamos a distancias y recorridos en los cuales resulta imprescindible utilizar alguna modalidad de transporte como apoyo al desplazamiento.

En la actualidad se reconoce la movilidad como un derecho humano fundamental (*Declaración Universal de los Derechos Humanos*, 1948), por lo que se deben proyectar, construir y operar sistemas de transporte público a los que todos puedan acceder, utilizar y abandonar en condiciones de seguridad, comodidad y eficacia. Por tanto, debemos atender a las necesidades de todos los tipos de usuarios que puedan usar el transporte público.

Atendiendo a las normas de entornos accesibles las personas se pueden agrupar de tres formas (“Manual para un Entorno Accesible”, 2005). Las personas que pueden usar el transporte público convencional sin problemas de movilidad y de comunicación ocupan el primer grupo. A estos, las mejoras en accesibilidad les suponen un apoyo diario y una mayor comodidad. El segundo grupo esta comprendido por aquellas personas que sólo pueden usar el transporte público si está completamente adaptado, tanto en su infraestructura, como en sus vehículos y los elementos que los vinculan. Por último, el tercer grupo son aquellas personas que tienen graves problemas de movilidad y que no podrían usar el transporte público ni estando adaptado, y para las que la mejor solución pasaría a ser un servicio de puerta a puerta y personalizada, como taxis y VTCs. Por tanto, el grupo más vulnerable sería el segundo, ya que los medios existentes no están del todo adaptados.

Las personas mayores están en algunos de estos grupos y muchos de estos utilizan el transporte público incluso para moverse entre sitios cercanos por sus dificultades para

caminar. En España, la población de la tercera edad se situó en 2021 en 9,38 millones de personas (Statista, 2021b), de las cuales un 23 % afirman algunas dificultades en cuanto a su movilidad. La degradación física, cognitiva y sensorial que se produce al envejecer pueden limitar la interacción de éstos con algunos medios de transporte. Esto conlleva, en la mayoría de los casos, a una disminución en la seguridad de uso y a que el usuario desconfíe de estos servicios, pudiendo llegar incluso a prescindir de ellos. La desconfianza en estos servicios viene motivada por un diseño que no tiene en cuenta las características propias de estas personas y que olvidan sus necesidades y su papel como usuarios. Se ha de tener en cuenta que el transporte público fomenta además la disminución de la soledad y de los síntomas de depresión (Matsuda y col., 2019) (Yang y col., 2019).

Para solventar los problemas del transporte público que puedan causar estrés a las personas mayores, se debe atender ciertos parámetros como la calidad de la prestación del servicio, la calidad de la infraestructura, la calidad del material móvil y cómo se plantea la interconexión entre estos. (“Manual para un Entorno Accesible”, 2005) Para conseguirlo, debemos tener en cuenta ciertos requerimientos:

- **Necesidades en cuanto a las infraestructuras.** Se deben tener en cuenta aspectos como avisar de forma visual y acústica tanto de la llegada del vehículo como la próxima parada. Entre otros, se destaca la necesidad de sistemas de climatización para lograr un buen grado de confortabilidad. Esto supone que situaciones donde no existan pantallas digitales donde se informe en tiempo real de la próxima llegada o no se provea de un cartel informativo con el horario de paso, provoquen situaciones de estrés a los usuarios del sistema.
- **Requerimientos del material móvil.** Se debe atender a proporcionar suficiente número de plazas sentadas y, con esto, plazas para personas de movilidad reducida. En general, todo tipo de ruidos y vibraciones deben de evitarse para hacer el transcurso del viaje ameno. Si esto no se asegura, los usuarios no se sentirán tan cómodos durante su trayecto y los consorcios de transporte no estarían teniendo en cuenta estas necesidades.
- **Adecuada prestación del servicio.** Para prestar un buen servicio a los usuarios de transportes públicos se debe dar un trato adecuado a las necesidades y circunstancias de cada cliente. Además, se debe ofrecer información útil como incidencias, demoras o averías del servicio con medios accesibles para todos, así como garantizar la puntualidad y correcta frecuencia del servicio. Por último, se debe abogar por el correcto uso del servicio, evitando así vandalismos y situaciones que puedan generar sensación de peligro entre los usuarios.

Hoy en día, los consorcios de transporte obtienen y procesan multitud de datos que obtienen de sus pasajeros, que les permiten llegar a conocer cómo expandir su red de forma adecuada y a cómo obtener un mayor beneficio. Sin embargo, en cuanto a las encuestas de movilidad para medir la satisfacción por parte del usuario, se limitan a un

cuestionario de relleno manual que se hace a un grupo reducido de usuarios, y solo por parte de ciertos consorcios de transporte. Por ejemplo, en encuestas realizadas por los consorcios de transporte de Granada y Madrid se observa que se limitan a preguntar aspectos técnicos como tiempos de acceso y salida de las estaciones, tiempos de espera y tiempos en trayecto. (Consortio Regional Transportes de Madrid, 2004) (Consortio de Transporte Metropolitano del Área de Granada, s.f.)

La mayoría de las personas mayores perciben que tienen una buena calidad de vida y manifiestan un incremento en su felicidad subjetiva cuando mantienen relaciones familiares y sociales, con afecto positivo, en presencia también de salud y materiales de apoyo. Mismamente, las emociones positivas tienen efectos positivos sobre la salud, concretamente el sistema cardiovascular e inmunológico. Estos están relacionados con una mayor longevidad y con un menor riesgo de padecer accidentes cerebro-vasculares y con un menor tiempo de recuperación en caso de enfermedad o situación médica aguda (García Martínez, 2017). En cambio, las emociones negativas se asocian a problemas de salud mental, como depresión y aumento de riesgo de mortalidad en personas mayores. Por esta razón, suele incrementarse la frecuencia de sentir emociones negativas cuando la persona tiene problemas de salud o de apoyo social. En un trabajo del departamento de psicología y sociología de la Universidad de Las Palmas de Gran Canaria (Luján Henríquez, 2015) se realiza un experimento en el que se demuestra que la salud mental percibida por mayores está directamente relacionada con la emotividad y la capacidad de sobreponerse o resiliencia. Los mayores con problemas de depresión, ansiedad o estrés suelen manifestar conductas de negatividad, justificación de fallos, evaluación selectiva negativa, defensividad, discordancia e hipercontrol anticipatorio, todo ello relacionado con problemas en las habilidades cognitivas y emocionales. A la luz de este experimento, se sugiere trabajar en estrategias cognitivo-emocionales para modificar emociones, comportamientos y aspectos de inteligencia emocional, interpersonal e intrapersonal.

Se presenta, por tanto, un desconocimiento de las causas que generan estrés en el transcurso de tiempo en el que un usuario realiza un trayecto en transporte público como parte de su actividad diaria. Entre estas causas, se encuentran los posibles retrasos de la red de transporte, la incertidumbre derivada por falta de información del estado de la red en formato accesible, que el usuario llegue tarde debido a una larga espera en la parada, o si ha ocurrido algún incidente y esto provoca atascos en la vía (Cantwell y col., 2009). Conforme a esto, conocer la experiencia de los usuarios en tiempo real, llegando a registrar el nivel de estrés según estados o contextos de viaje sería de alto interés para el consorcio de transportes, pues así se podrían tomar acciones para minimizar, por ejemplo, las barreras que supone el uso del transporte público a personas de tercera edad. De igual modo, para las terapias psicológicas tendría un alto interés, ya que el terapeuta podría monitorizar el grado de estrés y planificar una intervención, dada la información sobre las circunstancias durante la realización de cierta actividad, de forma que pueda prevenir el aumento de estrés en su paciente.

En los últimos años se han realizado numerosos estudios (Schwartz, s.f.) (Stults-Kolehmainen & Sinha, 2014) acerca de cómo influye el estrés en el abandono de la

realización de actividades de la vida cotidiana, especialmente atendiendo a mayores, que sufren el riesgo de quedarse aislados y caer en depresión por tener una concepción de no adaptabilidad a la vida actual. Los consorcios tratan de mejorar la experiencia del usuario mediante encuestas, mientras que los psicólogos pueden tratar estos síntomas de estrés mediante sesiones de terapia, sin embargo, estos no resultan ser tan fidedignos como podrían ser otros mecanismos que obtengan información más objetiva y puntual sobre circunstancias ocurridas en el transcurso de la actividad. Ciertamente, existen sistemas que utilizan electrocardiogramas para obtener el nivel de estrés en un momento concreto, e incluso indican cuando tomar un respiro. No obstante, podría ser más fiable si estas mediciones se hicieran de forma continua y si se pudiesen supervisar de forma autónoma por parte de un terapeuta con el previo consentimiento del paciente.

De ello resulta necesario admitir mi inspiración a desarrollar un sistema que use tecnología para analizar el grado de estrés en el transcurso de una actividad diaria como el uso de transportes públicos, mediante el conocimiento de factores fisiológicos, ambientales y sociales que rodean al usuario, como si se encuentra solo o acompañado. Por esto, me mueve querer facilitar y extender el uso de transportes públicos al máximo número de personas posible con ayuda de las TICs.

1.2. Objetivos

El objetivo de este trabajo fin de grado es el análisis, desarrollo e implementación de un entorno multisistema para la monitorización de datos fisiológicos para todo tipo de usuarios durante el transcurso de una actividad diaria como el usar el transporte público. Además, será de especial utilidad para personas mayores o con alguna discapacidad. Este sistema permitirá, por un lado, modelar una actividad en cuanto a grado de estrés producido y conocer los factores que la producen, por otro lado, otorgará a los terapeutas la posibilidad de hacer un seguimiento más preciso a sus pacientes mediante el registro de sus usuarios y la visualización de sus actividades y, por otro, mejorar la atención y servicios prestados por consorcios de transporte mediante el análisis de los datos fisiológicos y etiquetados que se recopilan. Todo esto se llevará a cabo de una forma poco invasiva y con una solución de bajo coste, mediante el uso de un reloj inteligente con sensores y una plataforma web para el terapeuta. Este objetivo general se conseguirá mediante la consecución de los objetivos específicos que se enumeran a continuación:

- Estudiar el dominio del problema a resolver, el estrés, sus causas, síntomas y mecanismos de medición, incluidos aquellos que recogen datos fisiológicos.
- Analizar cómo el uso los medios de transporte puede ser estresante y averiguar si los proveedores de este servicio se ocupan de evaluar el estrés de los pasajeros y de tomar medidas.
- Desarrollar un método para obtener mediciones fisiológicas de un usuario a través de un reloj inteligente.

- Realizar un estudio acerca de las metodologías actuales para la limpieza/preprocesado de señales fisiológicas.
- Desarrollar un método para calcular el nivel de estrés de un usuario a partir de sus mediciones fisiológicas.
- Proporcionar una herramienta para la supervisión de usuarios mayores por parte de terapeutas, un sistema que, a modo de panel de control, permita registrar usuarios y visualizar los resultados de estrés de éstos cuando usan los medios de transporte.
- Optimizar los recursos necesarios para que un mismo reloj pueda recoger datos de varios usuarios, mediante el uso de una aplicación móvil.
- Desarrollar una aplicación para relojes inteligentes que registre actividades de tipo viaje en transporte público, pudiendo indicar el usuario el medio de transporte, junto con una serie de etiquetas correspondientes a estados de viaje, contextos relacionados con situaciones que se pueden dar durante dicho viaje, emociones y sentimientos básicos.
- Analizar la validez del método desarrollado y probarlo en situaciones reales con sujetos.

1.3. Estructura de la memoria

Para cumplir con estos objetivos planteados, esta memoria de trabajo fin de grado se estructura en cinco capítulos, cuyos contenidos se describen a continuación:

- En el capítulo 1, se encuentra la Introducción, donde se ha descrito la motivación y justificación para llevar a cabo este proyecto, explicando la necesidad existente junto con los problemas de la actualidad. Además, se proponen mejoras sobre lo ya existente y se explica porqué se ha escogido esta temática. De igual modo, se expone un objetivo general junto con objetivos más específicos justificados. Adicionalmente, se incluye la planificación temporal mediante un Diagrama de Gantt y el presupuesto que conllevaría realizar el trabajo.
- En el capítulo 2, se describe el Estado del Arte y Tecnologías, donde se describen los antecedentes al proyecto mediante la exposición de tecnologías base potenciales, la explicación del dominio del problema a resolver, la descripción de trabajos relacionados y analizando las aplicaciones similares que existen en la actualidad.
- En el capítulo 3, se realiza la Propuesta, donde se elige una metodología de trabajo de forma justificada y se detallan su desarrollo y los resultados junto a la exposición de herramientas de ingeniería, como los diagramas y plantillas que se usan.

- En el capítulo 4, se muestra la Evaluación y Validación del proyecto, donde se exponen casos reales de prueba del proyecto desarrollado y se demuestra su funcionalidad.
- En el capítulo 5, se indican las conclusiones obtenidas a partir del desarrollo que se ha realizado en este trabajo fin de grado. Asimismo, se proponen trabajos futuros sobre este proyecto.
- Como anexo A, se proporciona un manual de instalación de los sistemas desarrollados.
- Como anexo B, se proporciona un manual de usuario para explicar las funcionalidades del sistema.
- Como anexo C, se proporciona la documentación del código desarrollado para la aplicación del reloj y móvil.

1.4. Planificación temporal

Para la realización del presente trabajo fin de grado, se ha realizado una planificación basada en iteraciones para llevar a cabo el desarrollo. La planificación del proyecto abarca desde la primera reunión con las tutoras, donde se desarrolla una primera aproximación del proyecto, hasta las últimas revisiones de esta memoria. La duración total del proyecto es de aproximadamente 6 meses.

Para la planificación se ha tenido un periodo previo al comienzo de las iteraciones, donde se aporta la explicación por parte de las tutoras al desarrollador, de modo que se puedan elaborar las historias de usuario y puedan corregirse. De este modo, se puede elaborar un plan de iteraciones y obtener el número de iteraciones que se van a necesitar. Se ha intentado distribuir la carga de trabajo de la forma más equitativa posible entre iteraciones. Se puede visualizar el diagrama de Gantt en la figura 1.1.

A continuación, se explica el diagrama de Gantt de la figura 1.1, con más detalle:

- Al comienzo del proyecto, se establece un periodo de 10 días en el que se tiene la reunión inicial con las tutoras, donde se abarca la redacción de las historias de usuario que cumplen las necesidades para la realización del proyecto. Con esto, se elabora un plan de iteraciones y se comienza con la investigación, instalación y configuración del entorno de desarrollo necesario para comenzar las iteraciones.
- En el periodo de iteraciones, se realiza la extracción de las pruebas de aceptación de las historias de usuario correspondientes a la iteración, junto con la redacción de las tareas asociadas. Tras esto, se realiza una planificación mediante un diagrama de Gantt. A continuación, se procede a realizar las tareas asociadas a las historias de usuario, para finalmente, obtener documentos entregables que poder mostrar en la reunión de retrospectiva con las tutoras.

- Tras el periodo de iteraciones, se comienza con el desarrollo de la memoria, escribiendo los capítulos que son necesarios para documentar un proyecto con calidad.

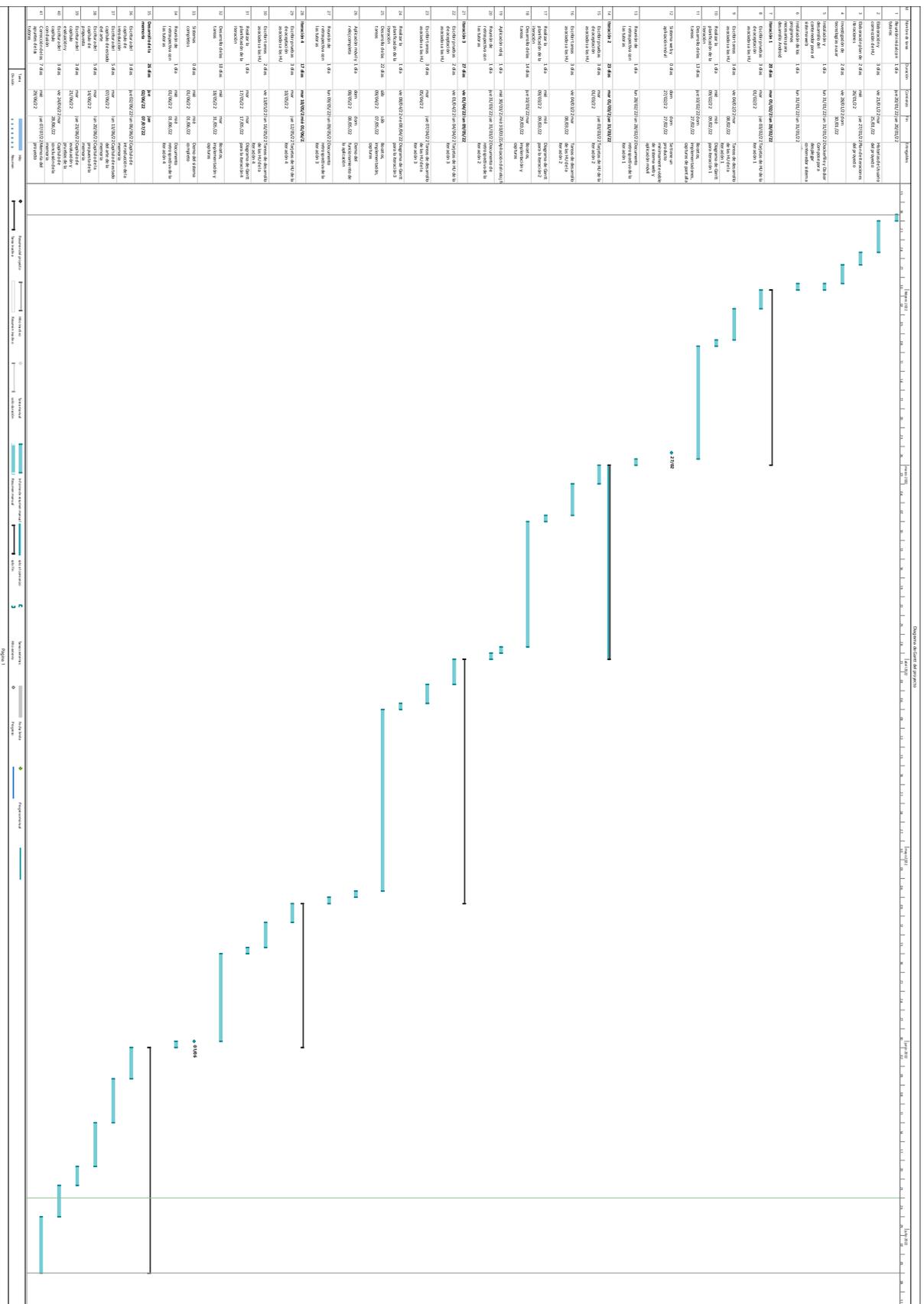


Figura 1.1: Diagrama de Gantt que describe la planificación del proyecto.

1.5. Presupuesto

El presupuesto para llevar a cabo el presente trabajo fin de grado puede desglosarse en los siguientes puntos:

- **Servidor Web.** Un equipo que albergue el servidor web para el terapeuta y que provea las funcionalidades necesarias para el buen funcionamiento de las aplicaciones para el móvil y reloj inteligente. Además, almacenará los datos de los usuarios en una base de datos.
- **Reloj inteligente.** Un dispositivo WearOS versión 2.0 en adelante para poder ejecutar la aplicación desarrollada de medición de actividades.
- **Teléfono móvil.** Un dispositivo móvil Android para poder ejecutar la aplicación desarrollada para cambiar el usuario del reloj inteligente.
- **Dominio Web.** Una dirección web para no depender de una IP dinámica, de modo que los distintos dispositivos puedan conectarse al servidor desde fuera de la red local.
- **Conexión a internet.** Se necesita de una red de banda ancha para la interconexión de los sistemas desde fuera de una red local.
- **Coste humano.** El coste por las horas de trabajo realizado por el desarrollador del proyecto.

A continuación, se han estudiado las necesidades de los sistemas y se hace una propuesta para cada sección del desglose anterior.

1.5.1. Servidor Web

Se recomienda un computador capaz de manejar concurrentemente varios usuarios en el sistema, al igual que la ejecución del algoritmo de predicción.

Además, el servidor debe contar con amplio almacenamiento para guardar toda la información que se recibe en crudo de los sensores de los usuarios. Para una disponibilidad de servicio más estable se recomienda conectar el computador a través de cable Ethernet a internet.

Tras estudiar las necesidades del sistema web, se recomienda la adquisición del equipo Dell PowerEdge T40, cuyo precio es 354,88 € sin IVA o un modelo similar. Este equipo posee las siguientes características técnicas:

- **Procesador.**
 - Modelo del procesador: E-2224G

- Frecuencia del procesador: 3,5 GHz
- Frecuencia del procesador turbo: 4,7 GHz
- Familia de procesador: Intel Xeon E
- Número de núcleos de procesador: 4
- Caché del procesador: 8 MB
- Número de procesadores instalados: 1
- Potencia de diseño térmico (TDP): 71 W
- **Memoria.**
 - Memoria interna: 8 GB
 - Tipo de memoria interna: DDR4-SDRAM
 - Ranuras de memoria: 4 x UDIMM
 - ECC: Si
 - Velocidad de memoria del reloj: 2666 MHz
 - Disposición de la memoria: 1 x 8 GB
 - Memoria interna máxima: 64 GB
- **Medios de almacenamiento.**
 - Almacenamiento: Parte frontal, un HDD SATA de 1 TB de 3,5" de forma predeterminada, hasta 3 HDD SATA de 3,5", máximo 12 TB
 - Interfaz del HDD: SATA
 - Capacidad del HDD: 1000 GB
 - Número de HDDs instalados: 1
 - Velocidad de rotación del HDD: 7200 RPM
 - Número de HDDs soportados: 3
 - Tamaños de disco duro soportados: 3.5"
 - Compatibilidad con RAID: Si
 - Tipo de unidad óptica: DVD-RW
 - Interfaces de disco de almacenamiento soportados: SAS, SATA
- **Unidad gráfica.**
 - Modelo de adaptador gráfico: Intel UHD Graphics P630
- **Conexiones.**
 - Tipo de interfaz ethernet: Gigabit Ethernet
- **Puertos e Interfaces.**

- Ethernet LAN (RJ-45) cantidad de puertos: 1
- Cantidad de puertos USB 2.0: 4
- Cantidad de puertos tipo A USB 3.2 Gen 1 (3.1 Gen 1): 5
- Cantidad de puertos tipo C USB 3.2 Gen 1 (3.1 Gen 1): 1
- Puerto de ratón PS/2: 2
- Puerto serial: 1

1.5.2. Reloj inteligente

Para asegurar la compatibilidad con la aplicación desarrollada, el reloj inteligente debe poseer entre sus características un sensor de fotopletismografía (PPG) que permita recoger información sobre el pulso cardiaco y el volumen de sangre. El que se detalla a continuación es un modelo sucesor al que se ha usado para el desarrollo del proyecto. Se propone un reloj que tenga conectividad 4G para poder usarlo independientemente de tener el móvil cerca al reloj o no.

El dispositivo propuesto es el TicWatch Pro 3, que tiene un coste de 284,39 € sin IVA, o un modelo similar. Este dispositivo posee las siguientes características técnicas:

- Dimensiones y peso: 47 x 48 x 12,2 mm. 42 gramos.
- Tamaño y tecnología de pantalla: AMOLED de 1,4 pulgadas
- Procesador: Snapdragon Wear 4100
- Memoria RAM: 1 GB
- Almacenamiento interno: 8 GB
- Capacidad de batería: 577 mAH
- Conectividad: Bluetooth Low Energy 4.2.
- Sensores: Sensor de ritmo cardíaco PPG, Giroscopio, Barómetro, GPS y NFC.
- Versión WearOS: 2.0 con futura actualización a 3.0

1.5.3. Teléfono móvil

Los requerimientos del dispositivo móvil son escasos, ya que tan solo se necesita un teléfono Android con una versión superior a la 11.0. Este tendrá la aplicación de WearOS con el reloj inteligente enlazado y deberá disponer de una SIM con un plan de datos suficiente para las mediciones. Se propone una tarifa de teléfono con 20 GB de datos para consumir a lo largo de un mes, con un coste de 7,9 € sin IVA en el operador O2.

El dispositivo móvil que se propone posee una gran pantalla, ideal para personas mayores con dificultad de visión y con un precio contenido de 150,02 € sin IVA o un modelo similar.

Este dispositivo cuenta con las siguientes características:

- Tamaño y tecnología de pantalla: TFT LCD de 6,6 pulgadas
- Procesador: Exynos 850
- Memoria RAM: 4 GB
- Almacenamiento interno: 64 GB
- Capacidad de batería: 5.000 mAh
- Conectividad: 4G, WiFi, Bluetooth Low Energy 5.0
- Sensores: GPS, Acelerómetro, Giroscopio.

1.5.4. Dominio web

Para poder hacer uso de nuestro sistema fuera de una red local es necesario conectarse a él desde fuera. Para solventar el coste adicional que supone que las operadoras establezcan una IP fija, se optará por el uso de un dominio web. Se propone el servicio usado para la realización del presente proyecto, que tiene un coste anual de 37,92 € sin IVA y está ofrecido por Google Domains.

1.5.5. Conexión a internet

Para el buen funcionamiento del servidor web, es necesario poseer una conexión a internet de banda ancha, a ser posible, mediante fibra óptica, ya que reduce bastante los tiempos de latencia en las peticiones que se le realicen a nuestro servidor.

Se propone, pues, una tarifa de la compañía O2 que ofrece 300 megabits de conexión simétrica por fibra óptica por un precio de 21,33 € al mes sin IVA.

1.5.6. Coste humano

Atendiendo a que el perfil del desarrollador del presente trabajo fin de grado sería de Junior Software Developer, tanto por sus estudios, como por las necesidades del proyecto, se ha procedido a consultar en diversas plataformas como Glassdoor (Glassdoor, 2022) y Epitech (Epitech, 2022) el sueldo medio actual que se ingresa netamente por este tipo de perfiles en España. Lo que se ha encontrado, de media, es un sueldo mensual de 1650 €. Esto supone que la hora de trabajo se fije en 10,31 €.

A esto hace falta sumarle el coste de la Seguridad Social. Como se trata de un contrato eventual y no es indefinido supone un coste del 32,1 % del sueldo bruto. Esto haría que el coste del empleado al empresario ascendiese a la cantidad de 2.179,65 € mensuales.

1.5.7. Presupuesto final

Como la duración del proyecto es de 6 meses, el coste total del proyecto, sumando los dispositivos, servicios necesario y coste humano sería de 14.108,93 €, con el posterior pago de 32,39 € al mes sin IVA, para continuar teniendo conexión a internet, un plan de datos móviles y el dominio web, a modo de poder seguir usando el sistema desarrollado.

Capítulo 2

Estado del Arte y Tecnologías

2.1. Contexto del trabajo

Este trabajo se enmarca dentro de la post-pandemia, donde muchos usuarios, especialmente personas mayores o con alguna discapacidad, sienten mucho estrés a la hora de verse obligados a tener que usar un medio de transporte público, ya que son, en su mayoría, personas de riesgo. En el caso de los mayores, este estrés se une al que ya tienen por presentar muchos de ellos problemas de movilidad y algún deterioro cognitivo.

Dado que las personas viven cada vez más vidas longevas y con una buena salud, no se puede negar que éstas tienen potencial para realizar contribuciones importantes a la sociedad. Sin embargo, las personas mayores son vulnerables, muy frecuentemente, a situaciones de exclusión, marginalización y discriminación.

La relación de población mayor respecto a la población mundial ocupa una proporción cada vez mayor, siendo la población de la cuarta edad (80 años y superior) la que crece más rápido. Bank, 2021 Mismamente, la población mayor que alcanza la jubilación está cada vez más sana y en forma. El incremento de la esperanza de vida y la mejora en el estado de salud de personas mayores es un logro importante y tiene un gran potencial en términos de capacidad laboral, capacitación y experiencia que la sociedad debe aprovechar de forma productiva. La experiencia que se tiene con personas mayores activas nos muestra que las personas mayores que están integradas en la sociedad tienen una mayor calidad de vida y viven de una forma más longeva y sana.

La sociedad necesita considerar, de una forma más activa, cómo integrar personas mayores y asegurar su participación en una sociedad cohesiva de todo tipo de edades. Las personas mayores se integran en la sociedad de muchos modos. Forman parte de los núcleos sociales de sus amigos y familiares, participan de forma activa en clubes y asociaciones, realizan voluntariados y son económicamente activos. Sin embargo, las personas mayores pueden ser vulnerables a la exclusión. Las principales barreras para lograr una participación igualitaria de las personas mayores en la sociedad incluyen la

pobreza, la mala salud, el bajo nivel educativo, la falta de medios de transporte, la accesibilidad a los servicios y la discriminación por edad. Por ello, lograr la inclusión y la participación social de este grupo involucra varios aspectos. Involucra a todos los grupos sociales e individuos en las estructuras políticas, sociales, culturales y económicas de la sociedad para que puedan participar en la toma de decisiones sobre los asuntos que les afectan. Esto requiere que exista un consenso para reducir y eliminar la exclusión, asegurando que todos los afectados reciban ayuda por parte de la sociedad. (United Nations Economic Commission for Europe, 2009)

Para lograrlo, se pueden llevar a cabo acciones concretas. Como en este trabajo fin de grado se abordará el tema de transportes, se debe mencionar que este es un área crucial para la inclusión de personas mayores en la sociedad. Para lograrlo, el transporte público debe ser asequible y accesible, al igual que todas las estrategias de implantación deben abordar las disparidades entre zonas urbanas y rurales. Para que las personas mayores que vivan en zonas rurales puedan quedarse a vivir sin tener problemas en cuanto a sus desplazamientos y no tener que mudarse a zonas urbanas, pueden estudiarse estrategias que faciliten sus viajes mediante, por ejemplo, el estudio de rutas a demanda del usuario.

Para aumentar la asequibilidad, varios países han optado por ofrecer el transporte a cero coste o ofrecen un precio reducido para las personas mayores. El tema con mayor complejidad en la actualidad es la accesibilidad. Los factores que afectan negativamente a la movilidad de personas mayores y con alguna discapacidad son la mala accesibilidad a las estaciones, plataformas bajas que requieren que se suban escalones, y la falta de asistencia en estaciones y vehículos, junto con la falta de información accesible sobre estaciones y trenes o buses accesibles.

España legisló sobre estos problemas de forma específica mediante la Ley de Igualdad de Oportunidades, No Discriminación y Accesibilidad Universal de 2003 (Boletín Oficial del Estado, 2003b). Esta ley fue derogada en 2013 y su reemplazo es en la actualidad la Ley General de derechos de las personas con discapacidad y de su inclusión social (Boletín Oficial del Estado, 2022).

En cuanto al marco legal Europeo existen también directivas acerca de accesibilidad en buses (Boletín Oficial del Estado, 2003a) y derechos específicos para personas con discapacidad en aviones (Boletín Oficial del Estado, 2002).

A nivel estatal, se debe instar a las compañías de transporte público para que mejoren la accesibilidad en trenes, buses, paradas y estaciones. Algunos consorcios ya han implantado autobuses de piso bajo y han mejorado el acceso a trenes. Para mejorar la accesibilidad, se debe tener también en cuenta el procedimiento para adquirir un billete. Esto es, que las máquinas expendedoras pueden no ser suficientemente claras a la hora de explicar cómo adquirir el billete. Sabemos que hay aplicaciones que ofrecen información sobre autobuses o metro, por ejemplo, permitiendo realizar acciones como la compra de tickets o la consulta de horarios o de llegada del próximo vehículo, sin embargo muchas personas mayores no están familiarizadas con tecnologías como Internet y móviles inteligentes, y las aplicaciones pueden no ser usables para ellos.

Las personas mayores o con alguna discapacidad necesitan tener una gran confianza para realizar viajes o sus trayectos en medios de transporte público. La disponibilidad de información relacionada con el sistema de reserva de billetes, los servicios de las estaciones y los servicios a bordo es crucial. De igual forma, se ha de mostrar claramente la información sobre personal de asistencia a mayores y personas con discapacidad y que este personal muestre una actitud positiva para el usuario mayor o con discapacidad, lo que le hará ganar confianza. (United Nations Economic Commission for Europe, 2009)

Invertir en este sector puede ser un buen negocio para empresas de transporte público, especialmente cuando en la mayoría de países las personas mayores son un grupo que cuenta con los recursos y tiempo necesarios para viajar, pero no pueden o no quieren conducir largas distancias en coche.

2.2. Dominio del problema

2.3. Personas mayores en transportes públicos

Los cambios en los patrones de movilidad de las personas mayores están causadas por diferentes factores, dependiendo de las circunstancias individuales.

Una encuesta realizada en Alemania (Bundesministeriums für Verkehr, 2017) demuestra que los motivos de desplazamiento de las personas mayores son principalmente para compras y ocio. Por este motivo, se consideró que el acceso a sanidad, supermercados, oficinas de correos y otros establecimientos sociales, culturales y de ocio se debe tener en cuenta a la hora de implantar accesibilidad en transportes públicos locales. Entre todos estos, el acceso a sanidad se reconoció como servicio esencial.

Al margen de los motivos de desplazamiento, se detectaron otros factores importantes para las personas mayores como la frecuencia del servicio, las distancias y la duración de los desplazamientos. En comparación con grupos de edad más joven, se aprecia que la media de longitud de desplazamiento decrece de forma constante a partir de los 50 años de edad. Esto puede deberse, en parte, a la reducción o abandono de los desplazamientos a puestos de trabajo derivados de la jubilación.

Un estudio Australiano (Sammer y col., 2012), que investigó acerca de los patrones de movilidad y las barreras que experimentaban grupos de personas con discapacidades, tanto físicas como sensoriales, muestra que las personas mayores se someten a barreras relacionadas con la falta de accesibilidad en el entorno, con la insuficiente información que se da y con servicios inadecuados para ellos.

Para mantener activa a la población mayor, que cada vez crece más rápido, en sus actividades diarias es vital que los servicios de transporte público sean adecuados para las necesidades específicas de las personas mayores. Para abordar esta cuestión, resulta necesario identificar los requisitos del transporte público que tienen las personas mayores dependiendo de sus circunstancias. (Shrestha y col., 2017)

Las discapacidades de las personas mayores se pueden clasificar en deficiencias de movilidad, visuales, auditivas y cognitivas. Por otra parte, podemos analizar las necesidades de las personas mayores desde una perspectiva del sistema de transporte, como por ejemplo, cuales son las características de un sistema de transporte adecuado para las personas mayores.

Entre los problemas que se identifican, se encuentran la falta de accesibilidad a autobuses y a las instalaciones de las paradas de bus, la falta de disponibilidad de información o de accesibilidad de ésta, la dificultad para orientarse y saber dónde se está, y la falta de disponibilidad de aseos, entre otros. Los usuarios mayores no solo tienen en cuenta esto, sino que también tienen en cuenta las condiciones de todo el trayecto, desde su hogar hasta que llegan al lugar deseado. En particular, los usuarios mayores se preocupan por la calidad del servicio, lo que incluye el personal de seguridad, la fiabilidad del medio de transporte, la frecuencia del servicio y su continuidad, el confort, la limpieza, la atención al cliente, la disponibilidad de información en tiempo real y la asequibilidad del título de transporte. (Shrestha y col., 2017). Todos estos son factores estresantes que hay que controlar.

2.3.1. Accesibilidad

En cuanto a problemas con la accesibilidad encontramos algunos puntos clave como la localización de paradas relevantes respecto al inicio del desplazamiento y el destino, la calidad de la infraestructura en las partes que hay que hacer el desplazamiento a pie y la accesibilidad de los autobuses en sí (si son de piso bajo o no). El diseño del bus en sí mismo puede ser también un factor importante en la accesibilidad a personas mayores, ya que estas podrían necesitar un asiento libre para ellas y que fuera accesible.

Algunas características que harían ideales los buses para las personas mayores son la entrada sin escalones al vehículo, barandillas de sujeción, asientos prioritarios en la parte delantera, información sonora en tiempo real, espacio para sillas de ruedas, entre otros. Las características ideales de una parada de bus para una persona mayor serían la proximidad a su domicilio, ubicación en un lugar visible y bien iluminado, ofrecer información clara en tiempo real, limpieza y protección ante fenómenos atmosféricos, proveer asientos y puntos de ayuda. (Shrestha y col., 2017)

Un estudio demuestra que cruzar calles y las paradas de autobús están entre los principales factores que hacen que las personas mayores dejen de usar el transporte público. (Koffman y col., 2010). Por tanto, para una persona mayor, la parada del autobús debe estar bien mantenida, debe tener un buen acceso y el tráfico de la zona no debe ser muy rápido.

2.3.2. Asequibilidad

La asequibilidad es un problema importante para muchas personas mayores, ya que, debido a la jubilación, tendrán unos ingresos más bajos. Debido a la edad, también es menos probable que tengan vehículo propio y que lo conduzcan y que, por tanto, tengan que depender más del transporte público. (Smith y col., 2006).

Entre las necesidades de las personas mayores se encuentran las tarifas reducidas, facilidad de uso, billetes flexibles y un esquema de tarifas sencillo.

2.3.3. Disponibilidad

Es esencial que exista disponibilidad de transporte público al alcance de los domicilios y destinos de las personas mayores, con horarios y frecuencias que cubran sus necesidades. Para dar una solución a las situaciones donde las personas mayores viven en zonas de baja demanda y, por esto, no es viable económicamente dar un servicio con alta frecuencia, se puede ofrecer un servicio bajo demanda, atendiendo a las necesidades del usuario mayor.

2.4. Definición de estrés

Con el mundo cambiando muy rápido y el ajetreo diario, a menudo se afirma el estar estresado. De hecho, el estrés es la causa principal por la que las personas acuden a un médico, psicólogo o especialista. La Organización Mundial de la Salud (OMS) define el estrés como «el conjunto de reacciones fisiológicas que prepara al organismo frente a la acción». (Torrades, 2007) Por tanto, se afirma que el estrés es una respuesta que nuestro organismo realiza durante determinadas situaciones, a priori amenazantes, y que hace que estemos listos para lo que venga. (Ramírez Loeffler & Rodríguez Vílchez, 2012)

El problema surge cuando el estrés se vuelve crónico, es decir, el cuerpo se mantiene en alerta incluso cuando no hay peligro, causando problemas de salud con síntomas físicos, motores, cognitivos y emocionales (Ramírez Loeffler & Rodríguez Vílchez, 2012). A continuación se exponen algunos de estos síntomas pertenecientes a cada grupo:

- **Síntomas físicos.** Entre otros, se dan taquicardias, temblores, sudoración, náuseas, dolor de cabeza y fatiga.
- **Síntomas motores.** Encontramos que se tiende a evitar situaciones temidas, tipos de respuesta como hiperactividad, intranquilidad motora, entre otros.
- **Síntomas cognitivos.** Cuando una persona está en situación de estrés puede sufrir desorientación, olvidos frecuentes, miedo, temor a que los demás se den cuenta de nuestras dificultades, agotamiento físico y psíquico, entre otros.

- **Síntomas emocionales.** Una persona con ansiedad se sentirá incapaz de afrontar situaciones debido a sus auto-valoraciones negativas, que hacen que se imagine todos los males que pueden derivarse de la situación. Estas respuestas emocionales se caracterizan por el miedo, la tristeza, la preocupación, la sensación de inseguridad, enfado o ira e irritabilidad.

El ejemplo más claro que se da en los transportes públicos y que derivan en estrés es el de tener que realizar un desplazamiento largo. De acuerdo con un estudio realizado en Suecia (Hansson y col., 2011), se explica que un desplazamiento largo conlleva, en la mayoría de ocasiones, el realizar un trasbordo entre buses o trenes, hacer conexiones y no perder de vista las paradas. En consecuencia, el usuario tiene un mayor riesgo de equivocarse o sufrir un retraso o pérdida de conexión, debido a que es algo que le es imprevisible e incontrolable.

Así pues, los desplazamientos pueden suponer tener que ajustar la vida cotidiana a los horarios de los autobuses o trenes (Kluger, 1998), lo que provoca inflexibilidad y pérdida de control. De acuerdo con un estudio realizado en Nueva Jersey (Wener y col., 2003), se compararon las respuestas psicológicas y psicofisiológicas de usuarios que realizaban rutas de Nueva Jersey a Nueva York por dos vías distintas. Una de ellas era la antigua línea, con un recorrido que debía hacer un trasbordo antes de llegar al destino y la otra de ellas era una línea nueva con un trayecto directo. Los resultados que se obtuvieron fueron que los que usaron la nueva línea tuvieron niveles de estrés mucho menores que los que usaron la línea antigua, ya que, siendo directa, tardaba menos en llegar al destino y era más predecible.

2.5. Métodos para la medición del estrés

Una forma de evaluar los niveles de estrés es utilizar una escala de auto-informe en un marco clínico (Chen y col., 2021). A continuación, se describen un par de ejemplos donde se utiliza esta forma de evaluación del nivel de estrés:

- La creación de escenarios que inducen estrés ayudan a los investigadores a recopilar y probar señales fisiológicas o de comportamiento asociadas con el estrés. Estas pruebas se basan, generalmente, en que se les pide a los sujetos que realicen una tarea específica que puede causar estrés, rellenen unos cuestionarios y cumplan con ciertas condiciones. Tras esto, se puede inferir la relación con el estrés, al monitorizar los cambios de las señales durante el experimento y compararlas con las respuestas a los cuestionarios.
- Un método desarrollado por la Universidad de Trier (Kirschbaum y col., 1993), consiste en una prueba en la que se le solicita al sujeto que haga el papel de candidato a un puesto de trabajo y que se prepare para un discurso de 5 minutos. Tres personas hacen de público asumiendo roles de entrevistadores y directivos, a los

que el sujeto debe convencer de su idoneidad para el puesto de trabajo imaginario. Si el sujeto termina su presentación antes de los 5 minutos se le pide que continúe. Cuando termine su presentación, se le pide que realice cálculos mentales diciendo en voz alta los números a partir del 1022 de 13 en 13. Si el sujeto comete un error, debe volver a empezar desde el 1022. Tras esta prueba de 10 minutos de duración, se le permite al sujeto descansar, monitorizando también sus señales fisiológicas.

- En cuanto al uso de dispositivos *wearable*¹ en la detección de estrés se muestra cómo los sensores de estos podrían beneficiarnos en cuanto a la detección del estrés. Las señales fisiológicas útiles que son de interés en la investigación de la detección del estrés son la actividad cardíaca (ECG), la actividad cerebral (EEG), la actividad muscular (EMG), la actividad electrodérmica de la piel (EDA) y la temperatura de la piel y cuerpo. (Chen y col., 2021)

2.6. Sensores para medir el estrés

Como el estrés deriva en múltiples consecuencias fisiológicas, se pueden usar sensores para lograr su detección. A continuación, se describen los sensores que existen en la actualidad y que se usan para medir el estrés.

- **Sensor ECG.** Es capaz de obtener la actividad eléctrica del corazón, pudiendo representarse de forma gráfica a modo de electrocardiograma. Normalmente, de esta actividad eléctrica se pueden distinguir tres tipos de ondas, la onda P, que representa la despolarización auricular (Cadogan & Buttner, 2022), el complejo QRS (Larkin, 2021) y la onda T, que representa la repolarización de los ventrículos (Burns & Buttner, 2021). La mayoría de los estudios sobre la actividad cardíaca están relacionados con tres componentes del corazón: el dominio del tiempo, el dominio de la frecuencia y las características no lineales del corazón. La investigación en el dominio del tiempo se centra en parámetros como la frecuencia cardíaca (FC), los intervalos entre latidos (RR) y la variabilidad de frecuencia cardíaca (VFC). (Hernando y col., 2018) Para los intervalos RR se pueden estudiar tanto su valor medio, como la desviación estándar o la raíz cuadrada de las medias. (Chen y col., 2021) (Ahn y col., 2019)
- **Sensor EEG.** El sensor de electroencefalografía es capaz de obtener la actividad eléctrica del cerebro. Cuando una persona se somete a estrés, el cerebro activa muchos sistemas secretores de neuropéptidos en respuesta. Como resultado de esta activación, se liberan hormonas corticosteroides suprarrenales, que se conocen como “hormonas del estrés”. Esta actividad es la que se puede medir para conocer el estrés de dicha persona. (Subhani y col., s.f.)

¹Un wearable es un dispositivo electrónico que se usa en el cuerpo humano y que interactúa con otros dispositivos para transmitir o recopilar datos. Los ejemplos más claros y más conocidos de wearables son los relojes inteligentes y las pulseras de actividad.

- **Sensor EMG.** El sensor de electromiografía permite medir la actividad eléctrica de los músculos. Existe un estudio relacionado con la tensión que se acumula en el trapecio, en el que se demuestra cómo la señal EMG posee amplitudes significativamente mayores durante el estrés en comparación con el descanso y menos lagunas (períodos de relajación). (Wijsman y col., 2010)
- **Sensor EDA.** El sensor de actividad electrodérmica mide los cambios de conductividad producidos en la piel debido al aumento de la actividad de las glándulas sudoríparas. En la actualidad, existen dispositivos que lo incorporan para medir el nivel de sudoración en las manos, pues está directamente relacionado con el estrés. (Gradl y col., 2019).
- **Sensor PPG.** El sensor de fotopleletismografía realiza una medición óptica de forma sencilla y barata y se utiliza a menudo para controlar la frecuencia cardíaca. Es un sensor no invasivo que utiliza una fuente de luz y un fotodetector en la superficie de la piel para medir las variaciones volumétricas de la circulación sanguínea. Mediante la aplicación de diversas técnicas de limpieza de señal, podemos obtener la Variación de Frecuencia Cardíaca (VFC). Existen varios estudios que consiguen medir el nivel de estrés basándose en la VFC obtenida de esta señal. (Heo y col., 2021) (Li y col., 2018)

2.7. Detección de estrés mediante sensor PPG

En los últimos años la demanda de dispositivos *wearables* que monitoricen nuestra salud en tiempo real ha crecido bastante, como se puede observar en la figura 2.1 y, como se ha visto en la sección anterior, la detección de estrés a través de señales fisiológicas mediante dispositivos *wearables* ya es posible.

La señal eléctrica ECG que atraviesa los músculos más finos alrededor del corazón y se activa durante un ciclo cardíaco, y los datos se recogen colocando sensores en el pecho cerca del corazón. (Dupre y col., 2005)

Por otro lado, la señal PPG representa una onda arterial variable durante cada ciclo cardíaco, y los datos se recogen mediante sensores en la muñeca. (Greene y col., 2016) Por consiguiente la incorporación de este sensor en dispositivos *wearables* es barata y beneficiosa. La mayoría de estudios sobre medición en estrés que se hicieron anteriormente utilizaron señales ECG ya que la señal PPG contiene más ruido que la ECG. No obstante, se realizó un estudio (Mohan y col., 2016) en el que se estudió el nivel de estrés a partir de una sola señal PPG, ya que es más simple y más apropiada para la vida cotidiana.

La figura 2.2 muestra la forma que tiene una señal PPG típica. Sin embargo, cuando el usuario lleva este sensor en su muñeca y está realizando actividades cotidianas, la señal se muestra con bastante ruido e incluso con zonas valle en las que no se ha medido nada, como se puede apreciar en la figura 2.3 .

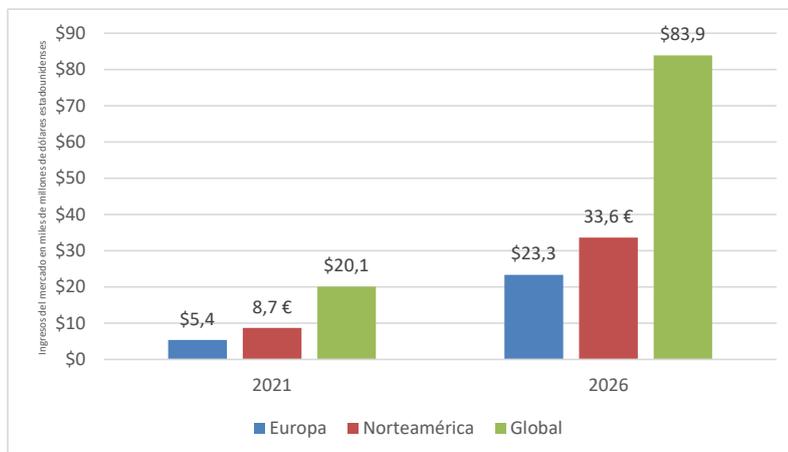


Figura 2.1: Predicción de ingresos del mercado de dispositivos *wearables* médicos en todo el mundo en 2021 y 2026. (Statista, 2021a)

Esto hace que la detección de picos, de sístole y diástole, y de los valles sea muy complicada. Para esto, existe una librería llamada HeartPy (van Gent, 2019) para la limpieza de ruido de señales PPG obtenidas desde un *smartwatch*.

Para obtener el estrés de una señal PPG se debe atender a la variación de frecuencia cardíaca (VFC) (Shaffer & Ginsberg, 2017). La VFC se calcula a partir de los cambios en los intervalos de picos sucesivos (McCraty & Shaffer, 2015). Las señales deben dividirse en tamaños de ventana adecuados para calcular la VFC. Las características más usadas se listan en la siguiente tabla:

ABREVIATURA	DESCRIPCIÓN	UNIDAD	ECUACIÓN
MEAN_RR	Media de todos los intervalos RR	$l/-$ min	
MEDIAN_RR	Mediana de todos los intervalos RR	ms	
SDRR	Desviación estándar de los intervalos RR	ms	$\sqrt{\frac{\sum_{i=1}^{N-1} (RR_{i+1} - RR_i)^2}{N-1}}$

RMSSD	Raíz cuadrada de la media de la suma de los cuadrados de la diferencia entre intervalos RR adyacentes.	<i>ms</i>	
SDSD	Desviación estándar de todo el intervalo de diferencias entre intervalos RR adyacentes	<i>ms</i>	
SDRR_RMSSD	Ratio (división) entre SDRR y RMSSD	-	
HR	Frecuencia cardíaca (latidos por minuto)	<i>l/- min</i>	
pNN50	% de intervalos RR adyacentes que difieren en más de 50 ms	%	$\frac{\sum_{i=1}^N (R_i - R_{i+1} > 50ms)}{N-1}$
SD1	Descriptor del gráfico de Poincaré de la VFC (Var. Frec. Card.) a corto plazo	<i>Ms</i>	$\sqrt{\text{variance} \left(\frac{RR_i - RR_{i+1}}{\sqrt{2}} \right)}$
SD2	Descriptor del diagrama de Poincaré del H a largo plazo	<i>Ms</i>	$\sqrt{\text{variance} \left(\frac{RR_i + RR_{i+1}}{\sqrt{2}} \right)}$
MEAN_REL_RR*	Media de todos los intervalos RR relativos	<i>ms</i>	
MEDIAN_REL_RR*	Mediana de todos los intervalos RR relativos	<i>ms</i>	
SDRR_REL_RR*	Desviación estándar de todo el intervalo RR relativo	<i>ms</i>	
RMSSD_REL_RR*	Raíz cuadrada de la media de la suma de los cuadrados de la diferencia entre el intervalo RR relativo adyacente	<i>ms</i>	
SDRR_RMSSD_REL_RR*	Relación entre SDRR_REL_RR y RMSSD_REL_RR	-	

Cuadro 2.1: Características de la VFC más relevantes para predecir el nivel de estrés

Para la obtención de las características marcadas en la tabla 2.1 mediante un asterisco, es necesario implementar sus cálculos modificando la librería HeartPy (van Gent, 2019), que es posible gracias a que posee una licencia MIT, el resto de las características se obtienen directamente de la implementación base de esta librería.

2.8. Trabajos previos sobre tratamiento de señales y datos

En esta sección, se analizarán los trabajos que se han realizando anteriormente y que tienen relación con el presente trabajo fin de grado.

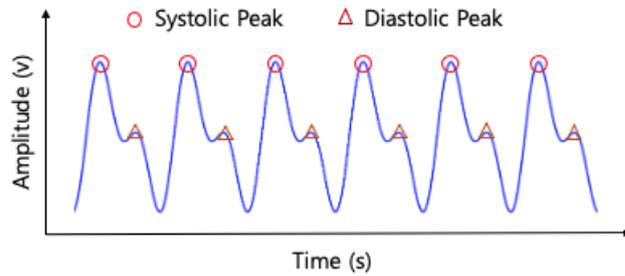


Figura 2.2: Señal PPG característica. (Mohan y col., 2016)

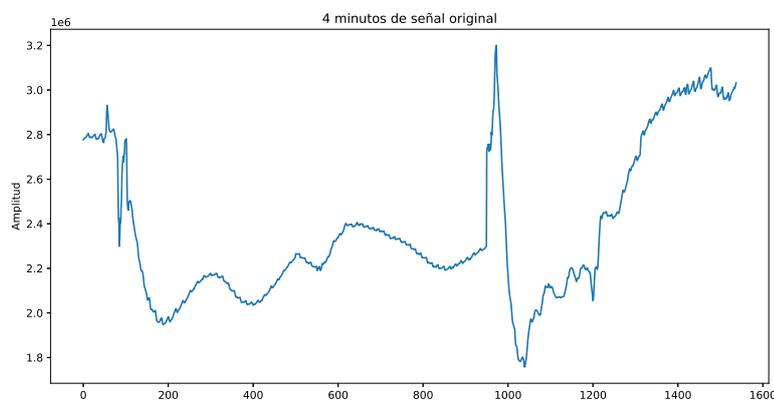


Figura 2.3: Primeros 4 minutos de una medición real realizada con el *smartwatch* cedido para este proyecto.

Para comenzar, la librería HeartPy (van Gent, 2019) nos permite cargar un conjunto de señales PPG en crudo, para su posterior procesamiento. Es necesario indicarle cuál es el ratio de muestreo, por lo que, en nuestro caso, al ser un *smartwatch* puede presentar variaciones. Para resolver este inconveniente todos los datos que se recojan se han de etiquetar con una estampa de tiempo, de modo que la librería sea capaz de detectar cuál es el ratio de muestreo. Es importante recalcar que, según el *smartwatch* con el que se trabaje, se obtiene un ratio distinto. Por esto, la librería indica que para determinar la variación de frecuencia cardíaca (VFC) se necesitaría al menos una señal de entre 9 Hz a 12 Hz.

Habiendo hecho esto, la librería es capaz de detectar los picos sistólicos y de calcular valores relacionados con la variación de frecuencia cardíaca (VFC) al igual que las pulsaciones por minuto (PPM) de las distintas ventanas temporales que componen una medición completa.

Para la comprobación de su funcionamiento, se referencian tres ejemplos de interés en cuanto a la aplicación de esta librería sobre un conjunto de mediciones PPG. (Paul-

vangentcom, 2019) (Paulvangentcom, 2021) (Paulvangentcom, s.f.)

En segundo lugar, profundizamos en el estado actual de la predicción de nivel de estrés en base a las características expuestas en la tabla 2.1. Unos investigadores de la *Facultad de Computación y Ciencias de la Información de la Universidad de Radboud*, en los Países Bajos, realizaron en 2014 un proyecto, denominado SWELL (Koldijk y col., 2018), para detectar situaciones de estrés mientras se trabajaba en oficinas mediante sensores que no son intrusivos. Mediante este proyecto, se obtuvo un conjunto de datos, mediante la experimentación, que modela el nivel de estrés de las personas en función de los datos que obtenían de distintos sensores. Entre los sensores que se utilizaron para el experimento, se encontraba un *Microsoft Kinect* (Cong & Winters, 2017), para la detección de posturas, un sensor para medir la variación de frecuencias cardíacas y otro sensor para medir la conductividad eléctrica de la piel.

En este experimento participaron 25 personas realizando trabajos de oficina típicos, como escribir informes, hacer presentaciones, leer el correo electrónico, buscar información, entre otros. Los factores que usaron para manipular sus condiciones laborales, y así, inducir situaciones de estrés, fueron tanto interrupciones mediante correos electrónicos y la presión por el límite de tiempo. La experiencia subjetiva de los participantes sobre la carga de trabajo que tenían, el esfuerzo mental que requería, sus emociones y el estrés que percibían se evaluó mediante cuestionarios que se tomaron como válidos de base para el experimento.

2.9. Metodologías que se pueden utilizar

Para comenzar esta sección, veremos que se pueden encontrar metodologías diversas en cuanto al desarrollo de software, en cuanto a patrones arquitectónicos de diseño software y en cuanto al diseño del software.

2.9.1. Metodologías de desarrollo software

En las tablas 2.2, 2.3, 2.4 se hace una comparativa sobre las metodologías de desarrollo software que exponen las características de cada una, junto con sus fortalezas y sus debilidades. Esto se realiza atendiendo a varios criterios como tipologías de proyectos, sus tamaños, el entorno de desarrollo y la disponibilidad de recursos.

2.9.2. Patrones arquitectónicos de diseño software

Como en nuestro proyecto desarrollaremos varias aplicaciones (móvil, reloj inteligente y web) que deben poder comunicarse entre sí, se deben tener en cuenta los patrones arquitectónicos de diseño software que existen en la actualidad y cuáles son las recomendaciones para cada lenguaje de programación. El propósito de seguir un patrón de diseño

METODOLOGÍA	CARACTERÍSTICAS
Cascada	La documentación debe ser bastante detallada y se debe realizar una planificación muy cuidada. Al ser un proceso lineal, cada fase tiene sus propios entregables.
Iterativo e incremental	El Product Owner del proyecto se involucra totalmente. Se tiene un número de iteraciones construidas en él como modelo inicial. Buen diseño por encima de buena documentación.
Programación extrema	El Product Owner decide qué tarea debe iniciarse primero. Acelerar la publicación de resultados. Se hacen pruebas unitarias. Contacto con el Product Owner de forma continua, debido a que se trabaja con él.
Ágil: Scrum	Desarrollo iterativo. Reuniones diarias. El equipo de desarrollo es auto-organizado

Cuadro 2.2: Comparativa de las características de metodologías de desarrollo software más comunes. (Saeed y col., 2019)

METODOLOGÍA	FORTALEZAS
Cascada	Sus fases son simples y están muy bien definidas, es simple de manejar y fácil de entender. Encaja mejor con proyectos pequeños.
Iterativo e incremental	El Product Owner del proyecto da su opinión constantemente. Se realizan numerosas revisiones durante el proyecto. El código se entrega en fases tempranas del proyecto. Encaja mejor con proyectos de medio y gran tamaño.
Programación extrema	Todo se hace rápido. El código se entrega en fases tempranas del proyecto. Se obtiene retroalimentación continua por parte del Product Owner. Encaja bien con proyectos tanto pequeños, como medianos y grandes.
Ágil: Scrum	Los productos se entregan en poco tiempo. Se obtiene retroalimentación de forma continua por parte del Product Owner. La inclusión de nuevos requisitos es rápida. Encaja bien con proyectos tanto pequeños, como medianos y grandes.

Cuadro 2.3: Comparativa de las fortalezas de metodologías de desarrollo software más comunes. (Saeed y col., 2019)

METODOLOGÍA	DEBILIDADES
Cascada	El código del proyecto suele entregarse tarde, no gestionándose bien la inclusión de nuevos requisitos, y no permite mucho cambio para resolver errores de diseño o planificación.
Iterativo e incremental	Cada iteración es inflexible pues sus fases son como un proyecto en cascada a pequeña escala.
Programación extrema	Falta documentación. Los desarrolladores no están dispuestos a hacer programación en pareja. Los programadores no están dispuestos a escribir pruebas antes de desarrollar código. Es necesario reunirse con frecuencia.
Scrum	La necesidad de desarrolladores con experiencia previa. La escasez de documentación. Es difícil estimar el coste y el tiempo al principio de proyectos grandes.

Cuadro 2.4: Comparativa de las debilidades de metodologías de desarrollo software más comunes. (Saeed y col., 2019)

es el de incrementar la modularidad, la flexibilidad, la testeabilidad y la mantenibilidad del presente proyecto.

En cuanto a patrones de diseño software para aplicaciones de Android, pues en nuestro caso será este el sistema del *smartwatch* y del *smartphone* vinculado, encontramos que los más extendidos son, el patrón Modelo-Vista-Controlador (MVC), el patrón Modelo-Vista-Presentador (MVP) y el patrón Modelo-Vista-ViewModel (MVVM). En la tabla 2.5 se exponen las distintas características que poseen y en la figura 2.4 se ilustran las relaciones entre vista, controlador, modelo, presentador y ViewModel.

Las principales diferencias que existen en cuanto a estos patrones arquitectónicos se pueden clasificar de acuerdo con los siguientes puntos (Lalani, 2022):

- **Arquitectura.** Los tres modelos se construyen de forma diferente. En MVC, la capa superior de la arquitectura es la vista, que se integra con el controlador y la capa del modelo se encuentra por debajo de estos. En MVP ocurre lo mismo, salvo que la vista da acceso directo al modelo. Esta exposición a toda la vista del modelo puede dar algunos problemas de seguridad, por ello, MVVM los solventa mediante una construcción diferente y la limitación de exposición entre cada capa.
- **Rendimiento.** Durante pruebas de rendimiento de Interfaz de Usuario, MVP es el mejor por tener una mayor fiabilidad. Lo que la diferencia del resto es que no pone obstáculos mientras se renderizan los cuadros de la aplicación. Por el contrario, MVVM está un poco por detrás en rendimiento debido a que tiene que hacer un mayor procesamiento. Por ejemplo, el Data Binding² crea una sobrecarga adicional que puede afectar drásticamente al rendimiento cuando se trata de realizar tareas complejas. Por otro lado, MVC presenta un rendimiento en medio de estos dos, ocupando una posición intermedia en cuanto a la evaluación del rendimiento.

²Vincula los componentes de la IU en los diseños con las fuentes de datos de la aplicación mediante un formato declarativo.

PATRÓN	CARACTERÍSTICAS
MVC	El punto de entrada es con el Controlador. Relación muchos a muchos entre la Vista y el Controlador. La Vista no conoce al Controlador. La Vista conoce qué debe recibir del Modelo.
MVP	El punto de entrada es con la Vista. Relación uno a uno entre la Vista y el Presentador. La Vista tiene una referencia a su Presentador y el Presentador conoce sus Vistas. La vista no conoce al modelo. El Presentador actualiza el modelo.
MVVM	El punto de entrada es con el Controlador. Relación uno a uno con el ViewModel y la Vista. El ViewModel no conoce a sus Vistas. La Vista no conoce al modelo. El ViewModel actualiza la vista

Cuadro 2.5: Características de las tres de arquitecturas de desarrollo software más extendidas en desarrollo en Android. (Dang, 2020)

- **Reusabilidad.** En cuanto a reusabilidad, MVVM se pone a la cabeza. Ofrece la mejor reusabilidad, especialmente cuando estamos procesando datos. Aunque el Data Binding no sea bueno para el rendimiento, tiene un impacto positivo en la compatibilidad. Tras este se situaría MVP, ya que MVC ocuparía el último lugar debido a su reusabilidad limitada y a sus problemas de mantener el estado de la aplicación.
- **Modularidad.** Tanto MVP como MVVM se sitúan a la cabeza, y es este último el que intenta separar el desarrollo de la Interfaz de Usuario de la lógica de negocio. El MVC tiene limitaciones en la modularización.
- **Tamaño de proyecto.** El patrón MVC es adecuado para proyectos pequeños, a diferencia del patrón MVVM que es conveniente para proyectos grandes. El patrón MVP sería adecuado tanto para proyectos pequeños como grandes.

Dada esta información sobre las tres arquitecturas, resulta necesario exponer las recomendaciones que Google detalla en su Guía para Desarrolladores Android (“Guide to app architecture”, s.f.).

Entre los principios recomendados por Android respecto a la arquitectura que se ha de seguir, encontramos la modularidad. Es un error común escribir todo el código en una

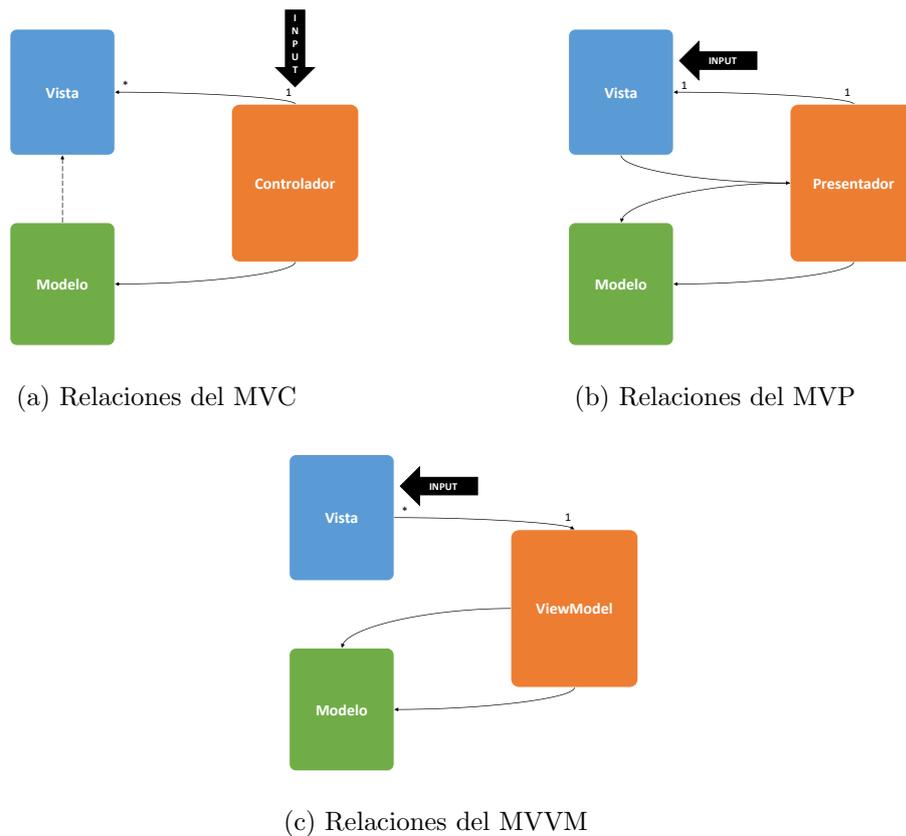


Figura 2.4: Principales patrones arquitectónicos de diseño software en Android. (Dang, 2020)

*Activity*³. Las clases de Interfaz de Usuario, como la *Activity*, deben contener únicamente la lógica con la que interactúa el usuario. Debemos mantener estas clases lo más limpias posibles, para así evitar problemas de ciclos de vida y mejorar la testeabilidad de las clases.

Adicionalmente, en la misma guía, se indica que se deben usar modelos de datos persistentes, que son independientes de la UI y de otros componentes de una aplicación. Estos modelos son ideales debido a que la aplicación puede seguir funcionando en caso de que una red de conexión no esté disponible. Si nos basamos en este principio, la aplicación será más robusta y testeable.

En cuanto a las capas arquitectónicas que debe tener una aplicación encontramos la capa de Interfaz de Usuario (UI Layer), encargada de mostrar los datos de la aplicación

³Una *Activity* en Android es una pantalla de la interfaz de usuario de la aplicación Android. En ese sentido, una actividad de Android es muy similar a las ventanas de una aplicación de escritorio. Una aplicación Android puede contener una o más actividades, es decir, una o más pantallas. La aplicación Android comienza mostrando la actividad principal, y a partir de ahí la aplicación puede permitir abrir actividades adicionales.

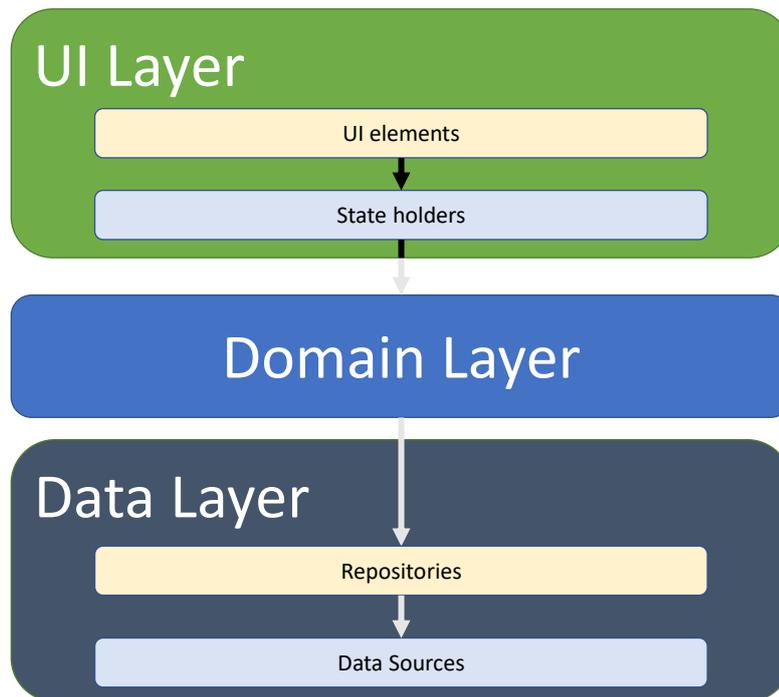


Figura 2.5: Diagrama de la arquitectura de una aplicación Android típica. (“Guide to app architecture”, s.f.)

en la pantalla, la capa de Dominio (Domain layer), para simplificar y reusar las interacciones entre la capa de UI y la de Datos, y por último la capa de Datos (Data Layer), encargada de contener la lógica de negocio de la aplicación y los datos. Estas se pueden ver representadas en la figura 2.5.

La capa de Interfaz de Usuario se compone tanto de los elementos que se renderizan en la pantalla, como los State Holders⁴, que manejan los datos y los exponen a la interfaz de usuario, gestionando también la lógica.

En cuanto a la capa de Datos, se expone que es la que contiene la lógica de negocio y determina cómo la aplicación crea, almacena y cambia los datos. Esta capa está compuesta de repositorios, que a su vez pueden contener fuentes de datos. Se debe crear un repositorio por cada tipo de dato que se gestione en la aplicación. Los repositorios son responsables de tareas como exponer datos al resto de la aplicación, centralizar los cambios de los datos, resolver conflictos entre múltiples fuentes de datos y abstraerlas del resto de la aplicación y contener la lógica de negocio.

⁴Un ViewModel puede mantener el estado de los componentes en una Vista. El ViewModel proporciona a la UI acceso a las otras capas, como la de dominio y la de datos. Entonces puedes definir las variables de estado usando LiveData, y definir los métodos que cambiarán el estado de estas variables. Por esto se les llama State Holders.

Por último se explica la capa de Dominio, que es la encargada de encapsular lógicas de negocio que son reusadas por varios ViewModel. Esta capa favorece la reusabilidad y aminora la complejidad.

Además se debe usar inyección de dependencias para mejorar la reusabilidad, la refactorización y la facilidad de testeo (“Dependency injection in Android”, s.f.) pues en las aplicaciones hay clases que dependen de otras para funcionar correctamente.

2.9.3. Diseño en Android

En la actualidad, Android posee unas directrices de diseño que se acuñan como Material Design. (Google, 2014) Recientemente, se han visto renovadas, alcanzando la versión 3. Seguir estas normas asegura que el diseño de nuestras aplicaciones sea accesible, universal y consistente, pudiendo diferenciar y entender los distintos componentes de la interfaz.

Además, al ser estas directrices las más generalizadas en torno al mercado de aplicaciones Android, se aprovecha un lenguaje visual bien conocido. Esto hace que, subconscientemente, los usuarios tengan un mayor nivel de confianza y seguridad en la aplicación, ya que se asocia con Google. (Woodhead, 2018)

Además con el uso de la versión 3 de Material Design, no debemos preocuparnos por la gama de colores a usar en nuestra aplicación, ya que es el sistema el encargado de escogerlo en función de la configuración escogida por el usuario.

2.10. Tecnologías que se pueden utilizar

En cuanto a tecnologías que podemos usar para el desarrollo del presente trabajo fin de grado encontramos tres categorías. Las tecnologías disponibles para el desarrollo de aplicaciones de la web, para *smartphone* Android y para *smartwatch* WearOS.

2.10.1. Tecnologías para desarrollar aplicaciones basadas en Web

Para desarrollar una aplicación web debemos tener en cuenta la parte del cliente (frontend) y el del lado del servidor (backend). Además de estas dos, debemos atender también a las tecnologías de base de datos que existen actualmente.

Tecnologías para la parte del cliente (frontend)

Las tecnologías que se usan para el desarrollo frontend son principalmente Angular, Express.js, Vue.js, HTML, CSS o SCSS, JavaScript, jQuery y React entre otros. (Kohan, 2022)

Las tecnologías mencionadas se pueden dividir en frameworks, lenguajes y librerías. En cuanto a frameworks se describen los mencionados:

- **Angular.** Permite el desarrollo de aplicaciones para todas las plataformas. Es un framework basado en componentes, de código libre, que usa *TypeScript* como lenguaje para su programación basada en componentes creando aplicaciones web escalables. Posee una colección de bibliotecas bien integradas y un conjunto de herramientas que ayudan al desarrollo y pruebas.
- **Express.js.** Es un framework de JavaScript que crea un entorno para desarrollar aplicaciones de tipo servidor especialmente aquellas que sirven una API.
- **Vue.js.** Es un framework de Javascript para crear interfaces de usuario y aplicaciones de una sola página. Su programación es de forma declarativa y también por componentes. Utiliza el patrón MVVM y se distingue de los demás por sus funciones de apoyo para la vista.

En cuanto a los lenguajes se describen los mencionados:

- **HTML.** Es un lenguaje de marcado que se usa para estructurar una página web y su contenido. Los frameworks mencionados anteriormente utilizan este lenguaje para su funcionamiento.
- **CSS / SCSS.** Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado como HTML.
- **JavaScript.** Es un lenguaje de programación que permite implementar funciones complejas en páginas web estáticas, de modo que actualicen su contenido dinámicamente, se puedan mostrar mapas interactivos, gráficos 2D y 3D, entre otros.

Por último, las librerías mencionadas se describen a continuación:

- **jQuery.** Es una librería de JavaScript que actúa como una capa superior para realizar, de una forma más sencilla, funciones como modificar textos, procesamiento de datos de formularios, mover elementos en pantalla y reproducir animaciones.
- **React.** Es una librería de JavaScript, que de forma análoga a Vue.js, se centra en desarrollar interfaces de usuario basadas en componentes.

Tecnologías para la parte del servidor (backend)

Para el backend, encontramos tecnologías como ASP.NET, Ruby on Rails, Flask, Django, PHP y Perl, entre otros. (Kohan, 2022) En cuanto a los frameworks mencionados se describirán a continuación:

- **ASP.NET.** Es un framework basado en .NET y C#, que se diferencia del resto, por ser de código abierto y multiplataforma para crear aplicaciones web dinámicas. Puede usarse tanto en proyectos grandes como pequeños y tiene una gran comunidad de soporte detrás.
- **Ruby on Rails.** Es un framework basado en Ruby, factor diferencial, que sigue el paradigma del patrón MVC. Aboga por la simplicidad, brindando la posibilidad de desarrollar aplicaciones del mundo real con menos código que otros frameworks y con una configuración mínima.
- **Flask.** Es un framework basado en Python minimalista que permite crear aplicaciones web basadas en componentes, por lo que es altamente escalable y su configuración es sencilla.
- **Django.** Es otro framework que se basa en Python muy configurable y aún más escalable que Flask. Tiene múltiples librerías que hacen que tareas básicas de implementación de funciones en sitios web como Autenticación o creación de APIs REST sean realmente simples.

A continuación se describen los lenguajes de programación mencionados:

- **PHP.** Es un lenguaje de programación de código abierto que se usa en desarrollo web y puede ser incrustado en HTML. Lo que le distingue de los frameworks que se han desarrollado anteriormente, es que el código es ejecutado en el servidor, generando HTML que se envía directamente al cliente, de modo que no se sepa el código subyacente.
- **Perl.** Es un lenguaje de programación interpretado de propósito general que solía ser el más popular en cuanto a desarrollo web debido a su capacidad de manipulación de texto y su rápido ciclo de desarrollo. En la actualidad existen frameworks que permiten la creación de aplicaciones web modernas.

Tecnologías para bases de datos

De acuerdo con un estudio realizado recientemente, entre las bases de datos más extendidas en la actualidad, encontramos que MySQL, PostgreSQL, SQLite y MongoDB ocupan los primeros puestos. (StackOverflow, 2022)

Estos se pueden clasificar en cuanto a si son relacionales o no relacionales (NoSQL). Para su clasificación, se abordará primeramente la definición y las diferencias que existen entre ambas.

- **Bases de datos relacionales.** Una base de datos relacional almacena y proporciona acceso a datos que están relacionados entre sí. Se basa en el modelo relacional, que es una forma intuitiva y sencilla de representar datos en tablas. En este tipo

de base de datos, cada fila de una tabla es un registro con una identificación única (clave). Las columnas de la tabla contienen los atributos de dichos datos y cada registro suele tener un valor para cada atributo, lo que permite establecer relaciones entre los datos. (Oracle, s.f.)

- **Bases de datos no relacionales.** Las bases de datos no relacionales almacenan datos de forma no tabular, basándose en estructuras de datos como los documentos. Un documento puede ser muy detallado y contener distintos tipos de información en diferentes formatos, por lo que es capaz de digerir y organizar varios tipos de datos de forma conjunta haciéndolo muy flexible. Se suelen usar cuando es necesario organizar grandes cantidades de datos complejos y diversos. (MongoDB, s.f.)

Las diferencias que existen entre ambas son que las bases de datos no relacionales son más flexibles que las relacionales ya que los datos no se limitan a un grupo estructurado y el acceso a datos en no relacionales es más rápido que en relacionales. Por otro lado, en bases de datos relacionales los datos se estructuran fácilmente en categorías, estando siempre en estado consistente y se pueden definir relaciones entre dichos datos de forma sencilla. (Pawlan, 2022)

Tecnologías de despliegue de aplicaciones web

Para lograr que una aplicación web que sea desarrollada se pueda visualizar desde fuera de la máquina en la que se desarrolla, su código, configuraciones y archivos necesitan ser desplegados en otra máquina con recursos suficientes que esté asociada a una url para poder acceder a ella desde cualquier lugar (Safwany, 2018). Existen varias formas de abordar esto. En cuanto a despliegue de aplicaciones web para desarrollo local encontramos que la opción más recurrida son los contenedores. Los contenedores son una forma de sistema operativo virtualizado. Un contenedor puede usarse para ejecutar desde pequeños microservicios hasta aplicaciones más grandes. Dentro de un contenedor se encuentran todos los ejecutables, código binario, librerías y archivos de configuración necesarios. Los contenedores tienen la ventaja de que son bastante ligeros y portables. (“What are containers?”, 2022) En cuanto a despliegue de aplicaciones también se puede optar por el alquiler de soluciones en la nube como Google Cloud, Amazon Web Services, Heroku, Azure, entre otros, que hacen mucho más simple las tareas de configuración.

2.10.2. Tecnologías para desarrollar aplicaciones para *smartphone* Android

Para desarrollar aplicaciones para Android, debemos cumplir ciertos requisitos. El primero de ellos es tener en un ordenador el SDK de Android descargado. Este se puede descargar automáticamente al instalar Android Studio. Existen programas alternativos para el desarrollo de aplicaciones Android, como IntelliJ IDEA, Visual Studio Code y

Eclipse, aunque Google recomienda usar su solución Android Studio desarrollada en conjunto con JetBrains.

Una vez hecho esto, es necesario valorar el estado del arte actual en cuanto a desarrollo Android. Existen hoy en día numerosos frameworks que hacen que el desarrollo sea más rápido y sencillo. Entre estos, encontramos que los más extendidos son Flutter, Ionic, React Native y Angular.

Debemos tener en cuenta también que estos Frameworks trabajan sobre distintos lenguajes de programación. Flutter lo hace sobre Dart, Ionic sobre HTML5, y React Native junto con Angular sobre JavaScript/TypeScript.

En la tabla 2.6 se expone una comparativa entre los principales frameworks de desarrollo de aplicaciones móviles.

	INTERFAZ DE USUARIO	MERCADO Y COMUNIDAD	RENDIMIENTO	PLATAFORMAS	LENGUAJES	PRECIO
REACT NATIVE	Nativa	La más grande	Muy cercano a nativo	Android 4.0.3+	JavaScript y React.JS	Gratuito
FLUTTER	Nativa	El segundo más grande	Muy bueno	Android 4.1.x+	Dart (basado en Java y C++)	Gratuito
IONIC	CSS y HTML no nativo	El tercero más grande	No muy bueno	Android 4.4+	CSS, HTML5 y JavaScript	Gratuito con versión de pago
KOTLIN	Nativa	Últimas posiciones, aunque creciendo rápidamente	El mejor	Android 1.0+	Es un lenguaje de propósito general.	Gratuito

Cuadro 2.6: Comparativa de los frameworks de desarrollo de aplicaciones Android más relevantes

Respecto a los lenguajes de programación usados en Android, tal y como indica Google en uno de sus artículos publicados en el blog de desarrolladores Android (Winer, 2019)), desde que lanzaron soporte para que Android soportase aplicaciones escritas en Kotlin, se ha incrementado en gran medida el número de desarrolladores que usan este lenguaje a día de hoy. Ellos establecen que en torno al 60% del top 1.000 aplicaciones Android en 2019 contenían parte de su código escrito en Kotlin.

Adicionalmente, lo que valoran los desarrolladores de usar Kotlin para desarrollar sus aplicaciones en Android es que es un lenguaje expresivo y conciso, brinda más seguridad a las aplicaciones evitando errores de programación comunes, es interoperable, ya que se puede llamar código Kotlin desde Java o viceversa y, por último, tiene una simultaneidad estructurada, lo que hace que se simplifiquen las tareas de codificar funciones bloqueantes y asíncronas mediante corrutinas. Además, a diferencia de los frameworks que existen en el mercado, Kotlin permite acceso a bajo nivel a funciones de la API de Android. (Google, s.f.-b)

En cuanto a diseño de Interfaces de Usuario, Google brinda a los desarrolladores varias opciones. Entre ellas nos encontramos una apuesta por el diseño declarativo de interfaces de usuario, dejando de lado al tradicional método imperativo (XML). Esta metodología se refleja tanto en la forma con la que Flutter diseña sus interfaces, como la nueva tecnología lanzada por Google llamada Jetpack Compose. (Yigit, 2022) Jetpack Compose es un kit de herramientas para compilar Interfaces de Usuario nativas, simplificando y acelerando el desarrollo de las mismas en Android. Esto se consigue ya que

se necesita menos código para mostrar los mismos componentes que con XML y es más intuitivo y legible, ya que es declarativo y por componentes⁵.

2.10.3. Tecnologías para desarrollar aplicaciones para *smartwatch* WearOS

Al ser WearOS un sistema operativo basado en Android, la metodología para el desarrollo de aplicaciones para esta plataforma es parecida a la subsección anterior.

Sin embargo, al ser el reloj inteligente un dispositivo que tiene características distintas a las de un *smartphone*, los frameworks que se pueden usar para agilizar el desarrollo de aplicaciones se reducen considerablemente.

Por un lado, encontramos que Flutter, de forma nativa, no tiene soporte para WearOS. Sin embargo, al tener una comunidad muy activa, han lanzado una librería para poder diseñar las interfaces adaptándose a las características de pantalla que poseen estos dispositivos. (Community, 2022)

En esencia, Flutter tiene un desarrollo bastante poco avanzado en cuanto a dispositivos *wearables* y no posee los componentes de interfaz de usuario recomendados por las directrices de diseño de Material Design para WearOS.

Como hemos comentado, en la actualidad, Material Design 3 se ha establecido como estándar de diseño de interfaces de Usuario y, según Google, la mejor opción es usar Jetpack Compose para incluir este diseño en las aplicaciones para WearOS. (Google, s.f.-a)

Por esto, también se recomienda en las guías para desarrolladores Android, elaborada por Google, que se use Kotlin como lenguaje de programación junto con Jetpack Compose para desarrollar aplicaciones para WearOS con versiones 2.0 en adelante. (Google, s.f.-c)

2.11. Trabajos relacionados

En la presente sección se analizan algunas evidencias científicas y revisiones recientes de uso de tecnologías o metodologías para resolver un problema similar al del presente trabajo fin de grado.

Comenzaremos por un proyecto desarrollado por el Instituto de Planificación y Evaluación de las Tecnologías de la Información y la Comunicación (IITP) financiado por el Ministerio de Ciencia del Gobierno Coreano, en el que tres investigadores consiguen detectar el estrés con un sensor PPG mediante la orquestación de múltiples métodos de eliminación de ruido y de detección de picos. En la figura 2.6 se puede ver el proceso general del método que han desarrollado. (Heo y col., 2021)

⁵Jetpack Compose permite desarrollar componentes pequeños, que serían partes de la Interfaz de Usuario, sin que estén vinculados a un *Activity* concreto, de modo que sea fácil su reutilización, su mantenimiento y su testeo.

El objetivo de este proyecto fue mejorar el rendimiento de la detección de estrés mediante el procesamiento de señales basadas en fotoplestimografía. Debido a que esta señal se puede recoger de dispositivos *wearable* pero están bastante afectadas por ruidos tanto internos como externos, es necesario desarrollar un método de eliminación de ruidos. Se propone un método en dos pasos, filtrando primeramente el ruido en términos de frecuencia y posteriormente eliminarlo en términos de tiempo. También desarrollaron un método de detección de picos múltiples dado un conjunto para extraer características de dicha señal.

Probaron dicha funcionalidad usando un conjunto de datos público desarrollado por la Universidad de Siegen llamado WESAD (Van Laerhoven, 2021) y obtuvieron una tasa de acierto del 96.5 %.

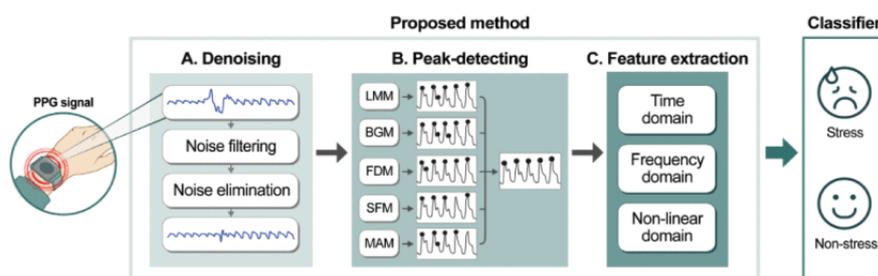


Figura 2.6: El proceso general del método que proponen en el proyecto desarrollado por el IITP (Heo y col., 2021)

Por otro lado, un proyecto publicado en la Revista de Ingeniería Eléctrica y Ciencias de la Computación de Indonesia estudia cómo desarrollar un método nuevo para reconocer estrés mediante una señal de fotoplestimografía (PPG). (Halim y col., 2017) Justifican el uso de este sensor ya que es de bajo consumo, barato y fácil de manejar. Para abordar el desarrollo, se comienza con el estudio de la señal PPG, indicando que se obtiene mediante la pulsioximetría. La absorción de la luz por parte de la pulsioximetría permite medir la saturación de oxígeno en la sangre (SpO2) y la frecuencia cardíaca (FC). Un pulsioxímetro identifica una señal pulsada a partir de la señal PPG, que suele ser en un pequeño porcentaje. Por esto, cualquier movimiento durante una prueba puede causar ruidos en la medición de la señal. La señal PPG cuenta con cuatro puntos diferenciados, siendo estos los puntos diastólicos, los puntos sistólicos, la muesca dicrótica y la onda dicrótica. A partir del análisis de esta señal se experimenta con 5 sujetos para poder hacer una extracción de características y una clasificación para modelar el nivel de estrés mediante el análisis de la amplitud de la señal obtenida. Para obtener resultados en este estudio, se les pidió a los sujetos que se relajasen durante 1 minuto. Tras esto, se les comienza a medir y se les pide que realicen el Test de Stroop (Halim y col., 2017), que muestra el tiempo de respuesta de una tarea y se utiliza a menudo para ilustrar la naturaleza del procesamiento espontáneo frente al control visual consciente. También se puede utilizar para inducir estrés al encuestado. Tras representar las señales obtenidas durante el periodo de inactividad y el periodo de estrés se puede apreciar que la amplitud

de la señal no es la misma. Se concluye que el pico sistólico de la señal de estrés es mucho mayor comparado con el estado normal

Por último se describe un procesamiento realizado por un investigador de la Universidad de New Brunswick, el cual usa el conjunto de datos SWELL (Koldijk y col., 2018) para hacer una predicción del nivel de estrés (PJ, 2021). Este conjunto contiene datos recopilados, a partir de 25 sujetos que participaron en un experimento de 3 horas de duración, de sensores que miden la Variación de Frecuencia Cardíaca, la actividad electrodérmica de la piel junto con otros sensores como un Kinect que medía las posturas corporales, una cámara web que medía las expresiones faciales y programas de análisis de interacción con el ordenador. Para hacerlo, obtiene del conjunto de datos las condiciones, que resultan ser tres *sin estrés*, *interrupción* y *momento de presión*. Se observa que el conjunto de datos está desbalanceado, habiendo más muestras de *sin estrés* que el resto. Para finalizar, se define un modelo mediante *Random Forest* y lo entrena con el conjunto de entrenamiento de SWELL. Para comprobar el rendimiento de este modelo, se toma el conjunto de test, dado también por el conjunto SWELL, y se obtiene un subconjunto formado por 20 mediciones, para las cuales el modelo entrenado acierta la totalidad.

2.12. Aplicaciones similares

A día de hoy existen múltiples dispositivos *wearables* en el mercado que cuentan con aplicaciones preinstaladas para medir el nivel de estrés. En la figura 2.7 se muestran algunos de ellos. Cabe destacar que esta funcionalidad es algo que se está implantando en últimos modelos, por lo que es algo que se lleva haciendo desde hace poco tiempo. Muchos dispositivos suelen llevar una Suite de Salud⁶ integrada con un apartado para medición de estrés, por lo que se pueden consultar los resultados procesados directamente desde dicho apartado.

Comenzaremos por el Samsung Galaxy Watch 4, el cual posee una aplicación para medir el estrés en un instante determinado dentro de la Suite de Samsung llamada Samsung Health (Samsung, s.f.). Esta medición se basa en la Variación de Frecuencia Cardíaca (VFC), comprobando cómo un intervalo de pulsaciones es diferente de otro, y de un sensor biométrico para realizar un análisis de impedancia bioeléctrica, que se sitúa sobre los dos botones del reloj. Por esto es necesario para tomar la medición que el usuario posicione sus dedos sobre ambos botones. La medición debe realizarse en un entorno donde el usuario se encuentre inmóvil y, a poder ser, sentado. Este *smartwatch* es un producto que no resulta tener un precio muy asequible, siendo en el momento en el que se redacta esta memoria, el reloj tope de gama de la marca. (Summers, 2020)

Por otro lado, seguiremos con la pulsera Xiaomi Mi Band, que al contrario del reloj anterior, es un producto bastante asequible. Posee una aplicación para la medición de estrés en un instante determinado. Parecido al reloj, la Mi Band usa los datos obtenidos por sus sensores para obtener la Variación de Frecuencia Cardíaca (VFC) y los compara

⁶Una suite es una colección de dos o más productos software agrupados y vendidos juntos.



(a) Samsung Galaxy Watch 4



(b) Xiaomi Mi Band



(c) FitBit

Figura 2.7: Dispositivos y Aplicaciones de medición de estrés

con modelos de estrés entrenados. Mismamente, se pide al usuario que se esté quieto para realizar una medición fidedigna. (Kumar, 2021)

Por último, trataremos cómo Fitbit posee la tecnología mas avanzada en *smartwatch* para la medición de estrés fiable en un instante de tiempo mediante una aplicación propia. Sus últimos modelos, poseen un sensor de actividad electrodérmica, la cual es capaz de detectar pequeños cambios en el nivel de sudor de la piel, lo cual cambia también la conductancia de la piel. Como el nivel de sudor está controlado por el sistema nervioso simpático, estos cambios pueden indicar tendencias significativas sobre la respuesta del cuerpo ante el estrés. El dispositivo es capaz de detectar las respuestas del sensor electrodérmico en la palma de la mano o en la punta de los dedos porque es aquí donde hay más glándulas sudoríparas que en otras partes del cuerpo. (Fitbit, 2022)

Capítulo 3

Propuesta

3.1. Metodología de trabajo

Para comenzar, comentaremos cómo vamos a llevar a cabo el desarrollo del presente trabajo fin de grado mediante el desarrollo primeramente, del marco de trabajo a seguir, acompañado de la descripción de recursos que serán necesarios, seguido de qué métodos de investigación se usan.

3.1.1. Marco de trabajo

Para comenzar con nuestra propuesta, estableceremos antes la forma en la que se trabajará sobre este problema. Primeramente, se va a seguir una metodología de desarrollo ágil, dividiendo el trabajo en iteraciones de un mes de duración. Cada mes, se agendará una reunión con las tutoras de este trabajo fin de grado a fin de mostrar los avances hasta el momento y obtener una retroalimentación de la que poder pulir aspectos en el desarrollo.

3.1.2. Métodos de investigación

Para este proyecto, se escogerá una combinación de métodos de investigación, tales como la revisión bibliográfica, entrevista a las tutoras y sus compañeros de grupo de investigación y la observación de resultados recogidos mediante la experimentación. Llevaremos a cabo una investigación a partir de una combinación cuantitativa, a través de estudios empíricos recogiendo datos y procesándolos.

3.1.3. Descripción del alcance del sistema

El presente proyecto fin de grado consiste en la implementación en un entorno multi-sistema distribuido, que posibilite la comunicación entre los sistemas que lo componen, que son:

- Una aplicación web de gestión de usuarios mayores para terapeutas.
- Una aplicación de reloj inteligente para la medición de estrés y sus causas en la realización de actividades de la vida diaria (como por ejemplo usar medios de transporte).
- Una aplicación para móvil que permita cambiar el usuario que usa el reloj a modo de optimizar los recursos.

La aplicación web para el terapeuta debe tener las siguientes características:

- Identificación del terapeuta para que los datos recabados de los usuarios mayores sean de acceso restringido.
- Un panel de control de acceso restringido donde se encuentre la información asociada a los usuarios mayores.
- En el panel de control debe existir una sección para la gestión de los usuarios mayores, darlos de alta en el sistema, poder visualizarlos, editarlos y eliminarlos.
- En el panel de control debe existir además, una sección donde se puedan visualizar, para cada usuario, los niveles de estrés procesados y las etiquetas de una actividad.
- La visualización de actividades para un usuario debe estar categorizada por tipo de actividad.
- Existirán tres tipos de actividades, viajar en bus, viajar en metro y viajar en tren.
- Para la vista del procesamiento de una actividad se debe mostrar el nivel de estrés mediante un gráfico y una tabla con las etiquetas asociadas a contextos de viaje, estados de viaje, emociones y sentimientos.
- Los datos en crudo de medición deben poder exportarse por si se quiere hacer un estudio posterior.

En cuanto a la aplicación móvil, se debe cumplir con las siguientes funcionalidades:

- Debe contener una pantalla de identificación mediante la cual se cambiará el usuario del reloj inteligente.

- Se quiere notificar al usuario cual es el estado de la conexión con su reloj inteligente.
- La aplicación no necesita estar activa en pantalla para que se establezca una comunicación con el reloj (puede estar en background).
- La aplicación debe guardar la sesión hasta que no se cierre.
- Se debe poder cerrar sesión para cambiar de usuario.
- Debe contar con un diseño accesible.

Para terminar, la aplicación del reloj inteligente debe contar con las siguientes características:

- Debe comunicarse con el móvil para identificarse en el sistema con el usuario que esté identificado en el móvil.
- Cuando se quiere comenzar una actividad, se muestra un listado con los tipos de actividades disponibles.
- Se debe indicar mediante una pantalla adicional qué se va a mostrar y qué se debe hacer a continuación.
- En el menú principal durante el transcurso de la actividad se debe poder detener la medición de la actividad en transcurso y registrar una etiqueta.
- El registro de etiquetas debe realizarse por categorías, siendo las categorías, estados del viaje compuesto de etiquetas como esperando, en viaje, trasbordo o en destino; contextos, compuesto de etiquetas como larga espera, abarrotado, sentado, solo o acompañado, ruidoso y en peligro; emociones, como la felicidad, la tristeza, el miedo, la ira, la sorpresa y el asco; y por último sentimientos, como nervioso o tranquilo.

3.1.4. Recursos

Para el desarrollo del proyecto, se nos ha facilitado un dispositivo *smartwatch* modelo TicWatch Pro de la marca Mobvoi (Figura 3.1), el cual posee WearOS 2.0 y un sensor PPG.

Además de esto, se necesitará un *smartphone* que cuente con sistema operativo Android para enlazarlo al *smartwatch*, junto con un ordenador personal a modo de servidor para la aplicación web del terapeuta, que provea una API y almacene la base de datos, y un dominio para poder conectarnos al servidor desde fuera de la red local.



Figura 3.1: Mobvoi TicWatch Pro

3.1.5. Diagrama de arquitectura

La solución que se propone se organiza de acuerdo al siguiente diagrama de arquitectura mostrado en la figura 3.2. A continuación, se detalla y se justifica el por qué de la decisión de llevar a acabo una arquitectura como la que se muestra.

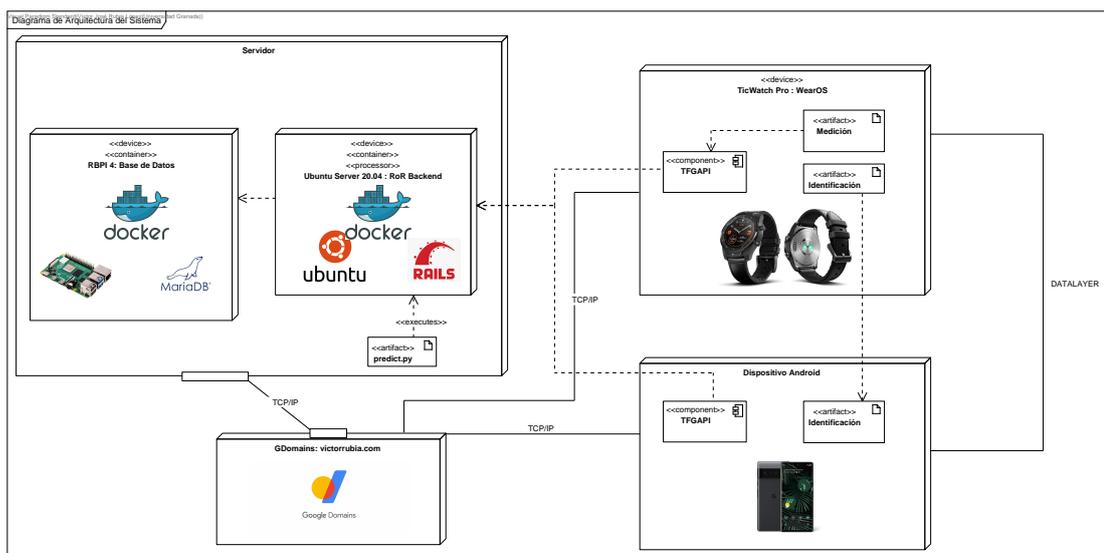


Figura 3.2: Diagrama de Arquitectura del Sistema

Se ha tratado de hacer una arquitectura dividida en módulos para poder desplegar cada parte en sistemas independientes, de modo que se optimice el uso de cada sistema y podamos cumplir las necesidades del proyecto. Se tienen tres módulos: servidor web, móvil y reloj inteligente.

En primer lugar, se desarrollará el módulo del servidor, que está dividido en dos sistemas:

- **Base de Datos.** La base de datos la proporciona una Raspberry Pi 4 con un sistema operativo Raspbian OS. Sobre este sistema se encuentra Docker instalado para el manejo de contenedores. El sistema ejecuta un contenedor con mariaDB para almacenar la base de datos de la aplicación web y lo expone a través del dominio contratado a través del puerto 6033.
- **Servidor de Aplicación Web.** Se trata de un ordenador personal con Ubuntu Server 20.04 instalado sobre el que se encuentra instalado Docker para ejecutar un contenedor con Ruby On Rails, framework de Ruby usado para la creación del sistema web. Este se conecta a la base de datos que se encuentra alojada en el sistema mencionado en el punto anterior. Además este contenedor es el encargado de ejecutar el procedimiento de predicción de obtención del nivel de estrés. Como características el ordenador posee 16 GB de RAM y 500 de disco duro SSD, junto con un procesador Intel i7.

En cuanto al módulo del dispositivo móvil se describen a continuación las características:

- **Dispositivo móvil.** Se trata de un teléfono inteligente modelo Pixel 6 de la marca Google el cual posee 8 GB de RAM y 128 GB de almacenamiento. Como versión de Android tiene la 12, que es la última disponible en el mercado. Este dispositivo debe estar enlazado al reloj inteligente a utilizar y contendrá una aplicación desarrollada para poder identificar a distintos usuarios en el reloj inteligente a modo de que pueda ser utilizado por varios usuarios optimizándose los recursos mediante la conexión con el servidor de la aplicación web.

Para el módulo del reloj inteligente describimos las características que posee:

- **Dispositivo reloj inteligente.** Se proporciona un reloj de la marca Mobvoi modelo TicWatch Pro el cual posee 512 MB de RAM y 4 GB de almacenamiento. Este contendrá la aplicación de medición de actividades al igual que un módulo de identificación de usuario para el cual se necesitará estar conectado al dispositivo móvil. Por otro lado, el reloj se conectará al servidor de la aplicación web para poder realizar las mediciones de actividades.

En cuanto al módulo que interconecta los demás módulos, describimos las características a continuación:

- **Dominio adquirido.** Para la evaluación de lo desarrollado en este trabajo fin de grado es necesario poder acceder desde fuera de la red local a los servidores para que las aplicaciones puedan funcionar. Esto no solo permite el uso fuera de la red local, si no que también hace que nos podamos conectar al servidor mediante una dirección única, al contrario de lo que pasa con la mayoría de proveedores de internet que ofrecen IPs dinámicas. Para ello, se establece un túnel de conexión entre el servidor y el dominio web.

3.2. Plan de iteraciones del proyecto

3.2.1. Lista inicial del producto (Product Backlog)

La tabla 3.1 contiene las historias de usuario obtenidas durante las reuniones con las tutoras del proyecto y han sido ordenadas por prioridad. Adicionalmente, se ha añadido la estimación del esfuerzo para cada historia, obtenido utilizando el método del *Planning Poker* (Gandomani y col., 2019).

ID	Historia de Usuario	Estimación	Prioridad	Entrega
HU-1	Como terapeuta quiero poder registrarme en el sistema para tener acceso a los datos.	5	1	1
HU-3	Como terapeuta quiero poder iniciar sesión en el sistema para poder entrar al sistema.	2	1	1
HU-16	Como terapeuta quiero que la sección de usuarios y de actividades sea únicamente accesible cuando el terapeuta se haya identificado	1	1	1
HU-5	Como terapeuta quiero tener una sección de usuarios para gestionarlos desde aquí.	5	1	1
HU-6	Como terapeuta quiero registrar usuarios mayores en mi plataforma para llevar un registro de ellos.	5	1	1
HU-8	Como terapeuta quiero poder visualizar a los usuarios registrados en el sistema para poder ver sus nombres.	1	1	1
HU-31	Como terapeuta quiero que el sistema proporcione las credenciales mediante correo a los usuarios para que puedan identificarse en la plataforma.	0.5	1	1

ID	Historia de Usuario	Estimación	Prioridad	Entrega
HU-17	Como usuario quiero poder iniciar sesión con los datos proporcionados por el terapeuta en la aplicación móvil enlazada con el reloj inteligente.	5	1	1
HU-20	Como usuario quiero poder iniciar una actividad desde el reloj inteligente para medir el nivel de estrés.	5	2	2
HU-11	Como terapeuta quiero visualizar una actividad mediante un gráfico que indique el nivel de estrés para que se lea de forma rápida.	5	2	2
HU-10	Como terapeuta quiero hacer el seguimiento de los usuarios mediante una sección de actividades categorizadas y ordenadas por fecha descendente.	5	2	2
HU-21	Como usuario quiero clasificar las actividades en “Viajar en bus”, “Viajar en metro” y “Viajar en tren” para poder registrar el tipo de actividad antes de iniciar una actividad.	2	2	2
HU-32	Como usuario quiero seleccionar en un listado los tipos de actividades para iniciar una actividad.	1	2	2
HU-24	Como usuario quiero poder añadir etiquetas en el transcurso de una actividad para etiquetar instantes de tiempo y conocer las razones dadas un nivel de estrés.	5	2	2
HU-12	Como terapeuta quiero visualizar una actividad mediante las etiquetas mostrándolas con la hora y de acuerdo con su categoría para que se lea de forma rápida.	3	2	2

ID	Historia de Usuario	Estimación	Prioridad	Entrega
HU-27	Como usuario quiero poder parar una actividad para terminar el registro de la medición de estrés.	3	3	3
HU-25	Como usuario quiero que las etiquetas estén clasificadas según su tipología para tener una mejor organización.	3	3	3
HU-26	Como usuario quiero poder seleccionar una o más de una etiqueta dependiendo de la categoría para poder incluir mayor detalle de información.	2	3	3
HU-23	Como usuario quiero poder realizar la medición incluso si en el transcurso se pierde la conexión de red para no perder los datos recogidos hasta dicho momento.	3	3	3
HU-22	Como usuario quiero que se me indique qué pantalla viene a continuación mediante una pantalla informativa para obtener una retroalimentación por parte de lo que esté ocurriendo en el reloj.	2	3	3
HU-19	Como usuario quiero poder recuperar mis credenciales en la aplicación móvil para recibir un correo electrónico con ellas y no molestar al terapeuta.	5	3	3
HU-18	Como usuario quiero poder cerrar sesión en la aplicación móvil enlazada con el reloj inteligente para que pueda ser usado por otro usuario.	3	3	3

ID	Historia de Usuario	Estimación	Prioridad	Entrega
HU-13	Como terapeuta quiero visualizar una actividad independientemente de que esté finalizada o se esté en transcurso para tener un seguimiento completo del usuario.	3	3	3
HU-15	Como terapeuta quiero poder eliminar una actividad de la lista de actividades realizadas por un usuario para dejar de tenerla registrada en el sistema.	3	4	4
HU-14	Como terapeuta quiero que mediante un indicador se notifique de qué actividades no han sido aún visualizadas por mí para poder localizarlas rápidamente.	3	4	4
HU-2	Como terapeuta quiero recibir un correo para confirmar el registro.	3	4	4
HU-4	Como terapeuta quiero poder cerrar sesión en el sistema para poder dejar de usarlo.	2	4	4
HU-7	Como terapeuta quiero poder editar a los usuarios registrados en el sistema para cambiar sus datos en caso de que exista algún cambio o problema.	3	4	4
HU-9	Como terapeuta quiero poder eliminar a los usuarios registrados en el sistema para que no puedan usar su cuenta más.	3	4	4
HU-30	Como usuario quiero que la aplicación del reloj me indique cuándo se ha quedado sin conexión para poder solventarlo lo antes posible sin tener que perder información.	3	4	4

ID	Historia de Usuario	Estimación	Prioridad	Entrega
HU-28	Como usuario quiero que la aplicación móvil sea accesible para personas mayores o con alguna discapacidad.	3	4	4
HU-29	Como usuario quiero que la aplicación del reloj sea accesible para personas mayores o con alguna discapacidad.	3	4	4

Cuadro 3.1: Product Backlog del proyecto

3.2.2. Cálculo de velocidad

En metodologías ágiles, la velocidad, se usa para medir la cantidad media de trabajo que un equipo puede completar en una iteración, en nuestro caso, de duración de un mes. Se usarán puntos de historia para realizar la estimación de las historias de usuario.

Para comenzar el proyecto, se seguirán recomendaciones de cómo estimar la velocidad inicial (digitalAI, s.f.) teniendo en cuenta que podré dedicarle finalmente dos tercios de las horas propuestas.

	PERSONA	HORAS
	Víctor José Rubia Lopez	36
TOTAL		36

Cuadro 3.2: Velocidad del desarrollador del proyecto

Por tanto se puede calcular la velocidad de desarrollo de la siguiente forma, atendiendo a las horas que puede hacer el desarrollador por iteración.

$$velocidad = 36 \times 0,7 = 25 PH$$

3.2.3. Descripción de las entregas

Como el esfuerzo necesario suma 100 puntos de historia y la velocidad del desarrollador es de 25 puntos de historia, para el desarrollo del proyecto se van a realizar cuatro entregas de una iteración cada una de ellas. El desarrollo del proyecto comienza el día 1 de febrero del 2022. En la tabla 3.3 se detalla de qué trata cada entrega y cuál es su fecha.

Entregas	Objetivos	Fecha de entrega
1	Tener una aplicación web que permita gestionar usuarios	1 de marzo
2	Tener una aplicación móvil que permita identificarse como usuario y transmitir la cuenta a la aplicación del reloj.	1 de abril
3	Tener una aplicación web que muestre las actividades de los usuarios y una aplicación del reloj que permita iniciar actividades	1 de mayo
4	Tener una aplicación web que muestre el detalle para las actividades de los usuarios y una aplicación de reloj que permita introducir etiquetado a la actividad.	1 de junio

Cuadro 3.3: Descripción de las entregas del proyecto

3.3. Primera iteración

3.3.1. Tarjetas de las Historias de Usuario

Se da comienzo con la extracción de las historias de usuario correspondientes a la primera iteración y el desarrollo de las tarjetas de historias de usuario.

Identificador: HU-1	Registro en el sistema web	
Descripción: Como terapeuta quiero poder registrarme en el sistema para tener acceso a los datos.		
Estimación: 5	Prioridad: 1	Entrega: 1
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Poder registrarse como terapeuta introduciendo datos de correo y contraseña. ▪ Registrarse con un correo que está previamente registrado debe dar error. ▪ Se manda un correo de confirmación a la dirección proporcionada sin error. ▪ Se indica mediante un pop-up que se ha realizado el registro correctamente. 		
Observaciones:		

Identificador: HU-3	Identificación en el sistema web	
Descripción: Como terapeuta quiero poder iniciar sesión en el sistema para poder entrar al sistema.		
Estimación: 2	Prioridad: 2	Entrega: 1
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Iniciar sesión con las credenciales correctas y que no exista ningún error. ▪ Iniciar sesión con las credenciales correctas y que se muestre un pop-up indicando el correcto inicio de sesión. ▪ Iniciar sesión con credenciales incorrectas debe mostrar un pop-up indicando error. ▪ Iniciar sesión con credenciales correctas debe permitir el acceso al Panel de Control. 		
Observaciones:		

Identificador: HU-16	Acceso restringido a Panel de Control	
Descripción: Como terapeuta quiero que la sección de usuarios y de actividades sea únicamente accesible cuando el terapeuta se haya identificado.		
Estimación: 1	Prioridad: 3	Entrega: 1
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Cuando se inicie sesión con las credenciales correctas debe poder accederse al Panel de Control. ▪ Si no hay ningún terapeuta identificado y se intenta acceder al Panel de Control, se debe denegar el acceso mediante mostrar el formulario de identificación. 		
Observaciones:		

Identificador: HU-5	Gestion de usuarios en aplicación web	
Descripción: Como terapeuta quiero tener una sección de usuarios para gestionarlos desde aquí.		
Estimación: 5	Prioridad: 4	Entrega: 1
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El terapeuta debe estar identificado para acceder al apartado de gestión de usuarios. ▪ Se listan los usuarios registrados actualmente en el sistema. 		
Observaciones:		

Identificador: HU-6	Añadir usuario a la plataforma	
Descripción: Como terapeuta quiero registrar usuarios mayores en mi plataforma para llevar un registro de ellos.		
Estimación: 5	Prioridad: 5	Entrega: 1
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El terapeuta debe estar identificado para añadir un usuario al sistema. ▪ Se muestra un botón en la sección de gestión de usuarios para añadir un usuario. ▪ Introducir un correo electrónico ya existente hará que no se guarde el usuario. ▪ Introducir un correo electrónico no válido hará que no se guarde el usuario. ▪ No introducir un nombre/pseudónimo hará que no se guarde el usuario. ▪ No introducir una contraseña hará que no se guarde el usuario. 		
Observaciones:		

Identificador: HU-8	Visualizar datos de usuarios del sistema	
Descripción: Como terapeuta quiero poder visualizar a los usuarios registrados en el sistema para poder ver sus nombres.		
Estimación: 1	Prioridad: 6	Entrega: 1
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El terapeuta debe estar identificado para visualizar los datos. ▪ Se debe mostrar el nombre de los usuarios registrados en el sistema. 		
Observaciones:		

Identificador: HU-31	Visualizar credenciales de los usuarios	
Descripción: Como terapeuta quiero que el sistema proporcione las credenciales mediante correo a los usuarios para que puedan identificarse en la plataforma.		
Estimación: 0.5	Prioridad: 7	Entrega: 1
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El terapeuta debe estar identificado para visualizar los datos. ▪ Se debe mostrar la contraseña del usuario recién registrado en el correo que se le envíe. 		
Observaciones:		

Identificador: HU-17	Aplicación móvil companion para identificación de usuario	
Descripción: Como usuario quiero poder iniciar sesión con los datos proporcionados por el terapeuta en la aplicación móvil enlazada con el reloj inteligente.		
Estimación: 5	Prioridad: 8	Entrega: 1
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Introduciéndose un correo no válido la aplicación no procede a iniciar sesión y se lo indica al usuario. ▪ No introduciendo nada tanto en el campo de correo como de contraseña hará que no se inicie sesión y se le indicará al usuario. ▪ Introduciendo las credenciales correctas la aplicación debe mostrar un check verde a modo de feedback y pasar a la siguiente pantalla. 		
Observaciones:		

3.3.2. Descomposición de las H.U. en tareas de desarrollo

Tras escribir las pruebas de aceptación de las historias de usuario que se abarcan en la primera iteración, se descompone cada una de ellas en tareas de desarrollo.

HU-1	Como terapeuta quiero poder registrarme en el sistema para tener acceso a los datos.	5 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-1	Realizar un boceto de la pantalla que permitirá el registro en el sistema.	0.5
T-2	Investigar librerías existentes para registro de usuarios en el sistema web.	1
T-3	Implementar la librería necesaria para el registro de usuarios.	2
T-4	Implementar el boceto de la pantalla de registro.	1
T-5	Realizar las pruebas de aceptación asociadas al registro de terapeutas.	0.5
Observaciones:		

HU-3	Como terapeuta quiero poder iniciar sesión en el sistema para poder entrar al sistema.	2 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-6	Realizar un boceto de la pantalla que permitirá el inicio de sesión en el sistema web.	0.5
T-7	Implementar el boceto de la pantalla de inicio de sesión.	1
T-8	Realizar las pruebas de aceptación asociadas al inicio de sesión de terapeutas.	0.5
Observaciones:		

HU-16	Como terapeuta quiero que la sección de usuarios y de actividades sea únicamente accesible cuando el terapeuta se haya identificado.	1 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-9	Investigar qué forma hay para restringir el acceso a páginas concretas sin estar identificado en el sistema web.	0.25
T-10	Implementar la restricción de acceso al panel de control si no se está identificado	0.5
T-11	Realizar las pruebas de aceptación asociadas a la restricción de acceso a la zona de identificado.	0.25
Observaciones:		

HU-5	Como terapeuta quiero tener una sección de usuarios para gestionarlos desde aquí.	5 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-12	Realizar un boceto de la sección de usuarios dentro del panel de control.	0.5
T-13	Implementar el boceto de la sección de usuarios del panel de control.	2
T-14	Implementar el acceso a API mediante clave privada	2
T-15	Realizar las pruebas de aceptación asociadas a la sección de gestión de usuarios.	0.5
Observaciones:		

HU-6	Como terapeuta quiero registrar usuarios mayores en mi plataforma para llevar un registro de ellos.	5 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-16	Realizar un boceto para el registro de usuarios mayores en la sección de gestión del sitio web.	1
T-17	Implementar el boceto del registro de usuarios mayores.	3
T-18	Realizar las pruebas de aceptación asociadas al registro de usuarios mayores en la plataforma.	1
Observaciones:		

HU-8	Como terapeuta quiero poder visualizar a los usuarios registrados en el sistema para poder ver sus nombres.	1 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-19	Realizar un boceto para la visualización de los nombres de usuarios mayores registrados en la plataforma	0
T-20	Implementar el boceto asociado a la visualización de los nombres de usuarios mayores.	0.5
T-21	Realizar las pruebas de aceptación asociadas a la visualización de usuarios mayores registrados.	0.5
Observaciones:		

HU-31	Como terapeuta quiero que el sistema proporcione las credenciales mediante correo a los usuarios para que puedan identificarse en la plataforma.	0.5 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-22	Implementar la funcionalidad asociada a mandar por correo las credenciales a los usuarios.	0.5
Observaciones:		

HU-17	Como usuario quiero poder iniciar sesión con los datos proporcionados por el terapeuta en la aplicación móvil enlazada con el reloj inteligente.	5 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-23	Realizar los bocetos de la aplicación móvil.	0.5
T-24	Implementar el boceto de inicio de sesión en la aplicación móvil.	1
T-25	Establecer la conexión con el inicio de sesión a la API.	1
T-26	Implementar punto de entrada en la API para la identificación de los usuarios mayores.	2
T-27	Realizar las pruebas de aceptación asociadas al inicio de sesión en la aplicación móvil.	0.5
Observaciones:		

3.3.3. Planificación temporal de la primera iteración

En la figura 3.3 se puede visualizar un diagrama de Gantt para la planificación detallada de la primera iteración.

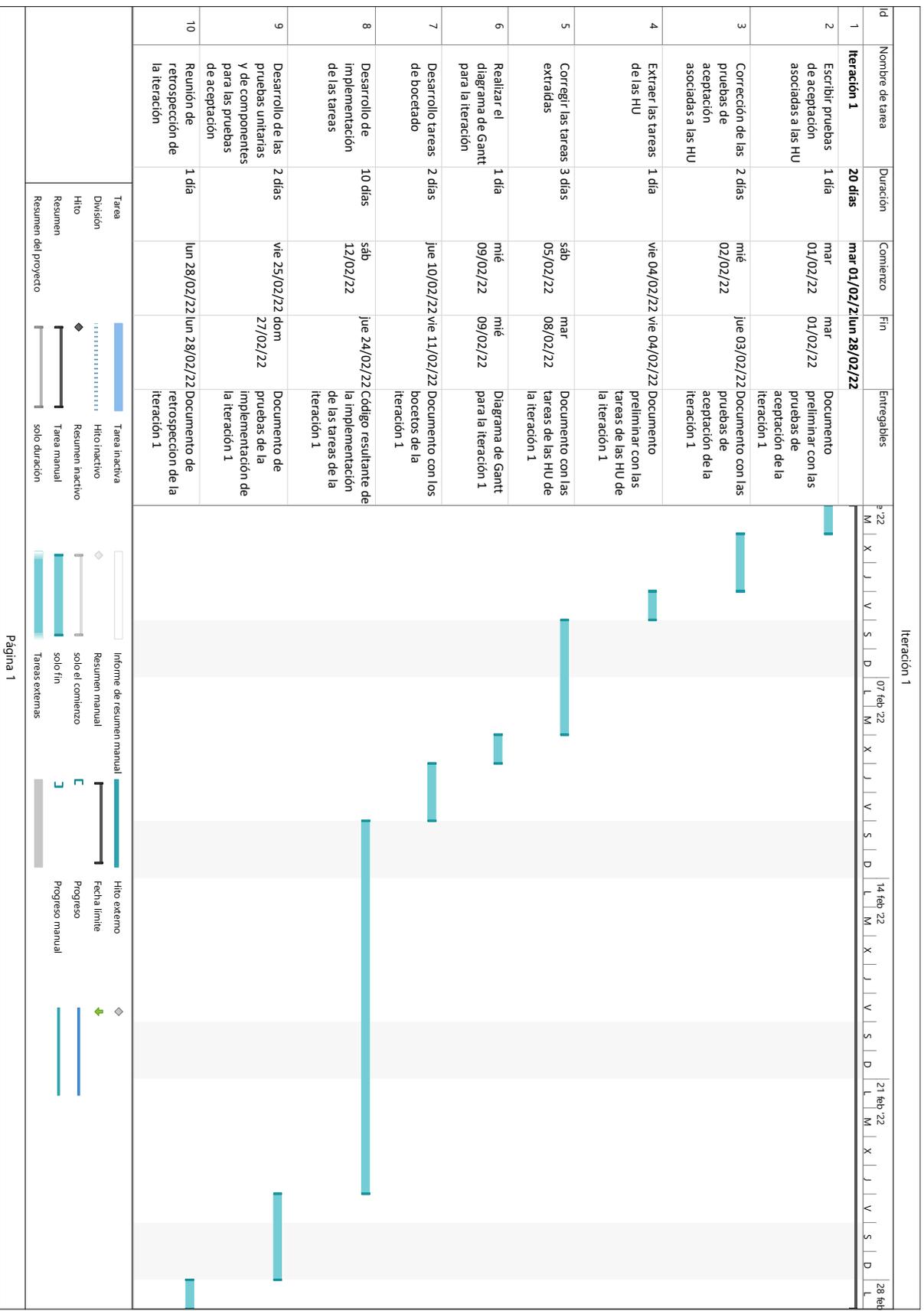


Figura 3.3: Diagrama de Gantt para la planificación de las tareas de la iteración 1

3.3.4. Realización de las tareas de desarrollo

Antes de comenzar con las tareas extraídas de las historias de usuario, resulta necesario configurar el entorno de desarrollo sobre el que se va a trabajar. Como las aplicaciones necesitan que el sistema web le proporcione información, se comenzará a desarrollar dicho sistema web.

Para establecer un sistema web que provea una API y que sea sencillo de mantener y probar nos valdremos del framework Ruby On Rails (RoR), ya que es una alternativa muy elegida hoy en día por su gran cantidad de librerías y soporte que tiene. Además, se elige realizar un contenedor Docker para el desarrollo del sistema web, de modo que nos abstraigamos del sistema anfitrión sobre el que se ejecute y consigamos otorgarle versatilidad.

Para lograr el desarrollo de esto, se ha desarrollado un *Dockerfile* que contiene la imagen oficial de Ruby On Rails disponible en *DockerHub*, además de las dependencias necesarias para la posterior predicción de niveles de estrés, como Pandas para el manejo de ficheros csv, Matplotlib para la realización de gráficos y SciKit Learn para la realización de modelos y predicciones. El código resultante se puede visualizar en la figura 3.1

```
1 FROM ruby:3.1.0
2 RUN apt-get update -qq && apt-get install -y build-essential npm yarn nodejs
   ↪ default-mysql-client default-libmysqlclient-dev pip
3 RUN apt remove cmdtest -y
4 RUN python3 -m pip install pandas matplotlib
5 RUN python3 -m pip install -U scikit-learn
6 RUN npm install -g npm@latest
7 RUN npm install -g yarn
8 WORKDIR /tfg
9 COPY Gemfile /tfg/Gemfile
10 COPY Gemfile.lock /tfg/Gemfile.lock
11 RUN bundle install
12
13 # Add a script to be executed every time the container starts.
14 COPY entrypoint.sh /usr/bin/
15 RUN chmod +x /usr/bin/entrypoint.sh
16 ENTRYPOINT ["entrypoint.sh"]
17 EXPOSE 3000
18
19 # Configure the main process to run when running the image
20 CMD ["rails", "server", "-b", "0.0.0.0"]
```

Listing 3.1: Dockerfile capaz de virtualizar un sistema que ejecute Ruby On Rails con las dependencias necesarias para la plataforma a desarrollar

Además se ha tenido que realizar un *Docker Compose* para poder ejecutar a la vez un contenedor con MariaDB que sirva al sistema web de Ruby On Rails. MariaDB es un sistema de gestión de bases de datos cuyo creador fue desarrollador de MySQL, por lo que está estrechamente relacionado. Además, MySQL no es totalmente de código abierto y MariaDB sí lo es. Esto hace que tenga una gran comunidad detrás que le da soporte e incluye funcionalidades más completas y novedosas que las que se incluyen con

MySQL. Este *Docker Compose* puede consultarse en la figura 3.2. Esto permite que se pueda desplegar la base de datos y se puedan establecer túneles de red entre distintos contenedores. Usaremos PHPMyAdmin para apoyarnos, de forma adicional a Ruby On Rails, en cuanto a la visualización, edición y eliminación de datos en de la base de datos. Para aumentar la seguridad, no se expondrá el servicio de base de datos sobre el puerto por convención 3306, si no que se invertirá y se expondrá sobre el 6033. De igual forma, la plataforma PHPMyAdmin se expondrá sobre el puerto 8081 y el servidor web sobre el puerto 3000. Como el sistema requiere de envío de correos, se proporcionará la clave privada necesaria para dicho envío mediante una variable de entorno con la finalidad de que no se filtre en la red.

```

1 version: "3.9"
2 services:
3   db:
4     image: mariadb
5     command: --max_connections=10000
6     container_name: db
7     environment:
8       MYSQL_ROOT_PASSWORD: admin
9       MYSQL_USER: admin
10      MYSQL_PASSWORD: admin
11     volumes:
12       - ./data/db:/var/lib/mysql
13     ports:
14       - "6033:3306"
15   phpmyadmin:
16     image: phpmyadmin/phpmyadmin
17     container_name: pma_tfg
18     links:
19       - db
20     environment:
21       PMA_HOST: db
22       PMA_PORT: 3306
23       PMA_ARBITRARY: 1
24     restart: always
25     ports:
26       - 8081:80
27   web:
28     build: .
29     command: bash -c "rm -f tmp/pids/server.pid && rake assets:precompile &&
30       ↪ bundle exec bin/rails s -p 3000 -b '0.0.0.0'"
31     container_name: rails
32     restart: always
33     environment:
34       - GMAIL_PASS=${GMAIL_PASS}
35     volumes:
36       - ./tfg
37     ports:
38       - "3000:3000"
39     depends_on:
40       - db

```

Listing 3.2: Docker Compose para orquestar los contenedores necesarios para que la plataforma funcione correctamente

Para continuar, se ha tenido que usar un dominio web para poder realizar el desarrollo, ya que desde donde se realiza, la dirección IP pública es dinámica y para el desarrollo

a lo largo de los meses no es conveniente estar cambiando la IP a usar en el código fuente todo el tiempo. Por esto, se ha decidido hacer accesible el sistema web a través de dicho dominio. Sin embargo, como el dispositivo que contiene la base de datos y el dispositivo que ejecuta el sistema web se encuentran conectados de forma local, no expondremos la base de datos al dominio público, de modo se preserve así su seguridad. Una vez se ha podido desplegar el sistema web, comenzamos con las tareas de desarrollo.

Para la realización de las tareas asociadas con esta primera iteración comenzamos con el bocetado del sistema web. Se ha optado por usar un diseño basado en *Bootstrap 5* (Twbs, 2022), ya que está bastante estandarizado y posee muchos elementos que nos interesa usar, como las barras de navegación, los paneles de control y las ventanas emergentes (modales). En cuanto a los iconos a utilizar en la plataforma web, optaremos por el uso de los que proporciona Feather, ya que son simples, universales y de código abierto (Feathericons, 2022). En cuanto al bocetado de Android, se han tratado de representar los elementos característicos de la interfaz de usuario proporcionada por el sistema operativo. Para los iconos, se opta por los que proporciona Material Icons (Google, 2022b), por ser también universales y sencillos en el universo Android.

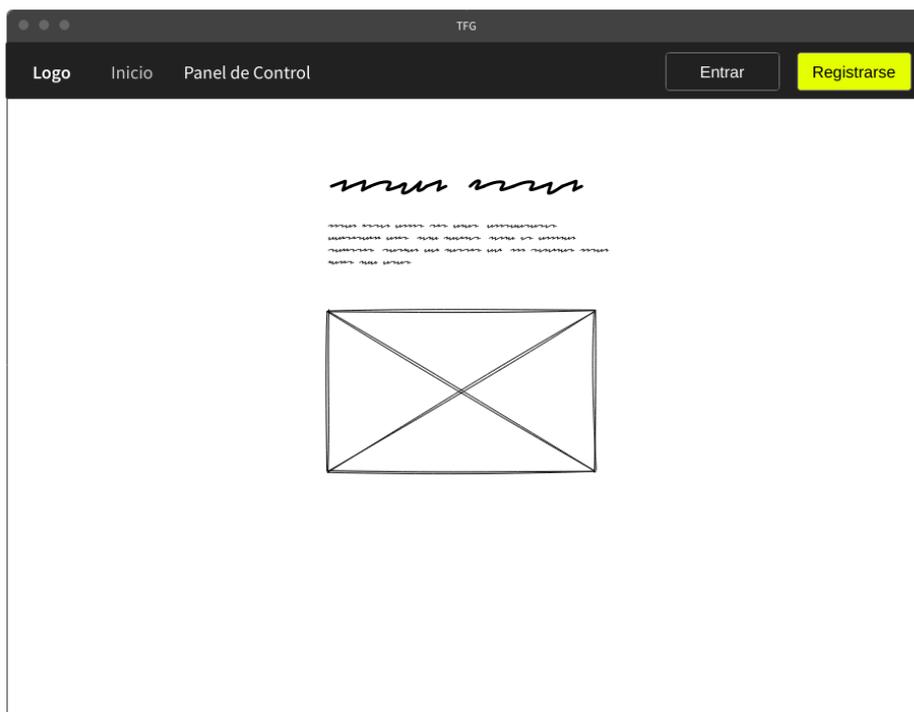
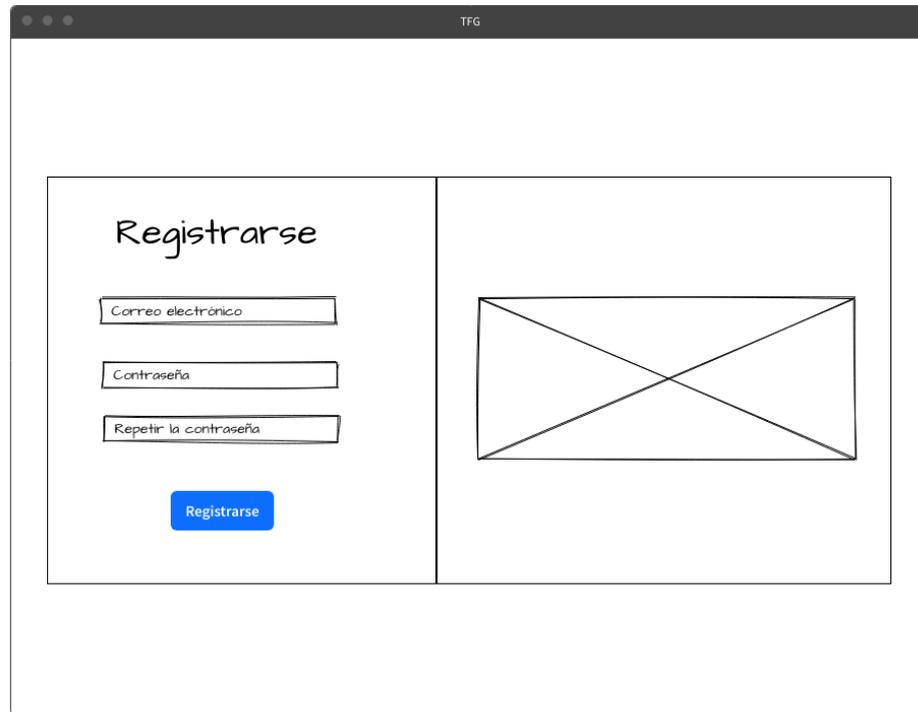


Figura 3.4: Página inicial de la aplicación web

Como vemos en la página inicial del sistema web expuesto en la figura 3.4, se ha optado por un diseño compuesto por una barra de navegación en color negro que contiene, de izquierda a derecha, el logotipo del proyecto, los enlaces al inicio y al panel de control

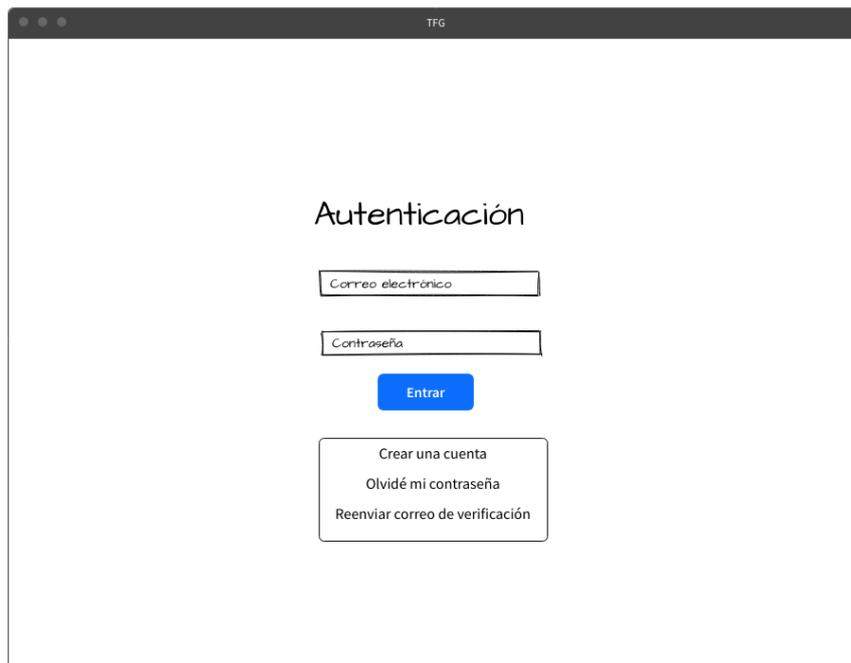
y botones tanto para identificarse como para registrarse.



The image shows a wireframe of a web registration page. The page is titled "Registrarse" and is divided into two main sections. The left section contains a registration form with three input fields: "Correo electrónico", "Contraseña", and "Repetir la contraseña". Below these fields is a blue button labeled "Registrarse". The right section contains a large rectangular box with a diagonal cross, representing a placeholder for a logo or a seal of identity.

Figura 3.5: Página de registro de la aplicación web

En la pantalla de registro de la figura 3.5 se encuentra el formulario y el logotipo a modo de sello de identidad. Se ha optado por el registro mediante correo electrónico y no por nombre de usuario para poder, con un solo dato, identificar al usuario de forma unívoca y poder mandarle la contraseña en caso de que la olvide, a dicho correo.



The image shows a web browser window with a dark title bar containing the text 'TFG'. The main content area is white and centered. At the top, the word 'Autenticación' is displayed in a large, black, sans-serif font. Below this, there are two input fields: the first is labeled 'Correo electrónico' and the second is labeled 'Contraseña'. A blue button with the text 'Entrar' is positioned below the password field. At the bottom, a white box with a thin border contains three links: 'Crear una cuenta', 'Olvidé mi contraseña', and 'Reenviar correo de verificación'.

Figura 3.6: Página de identificación de la aplicación web

En cuanto a la pantalla de autenticación de la figura 3.6, vemos nuevamente un formulario para introducir el correo y la contraseña. Además se incluyen enlaces por si se quiere crear una cuenta, recuperar la contraseña o reenviar el correo de verificación de la cuenta.

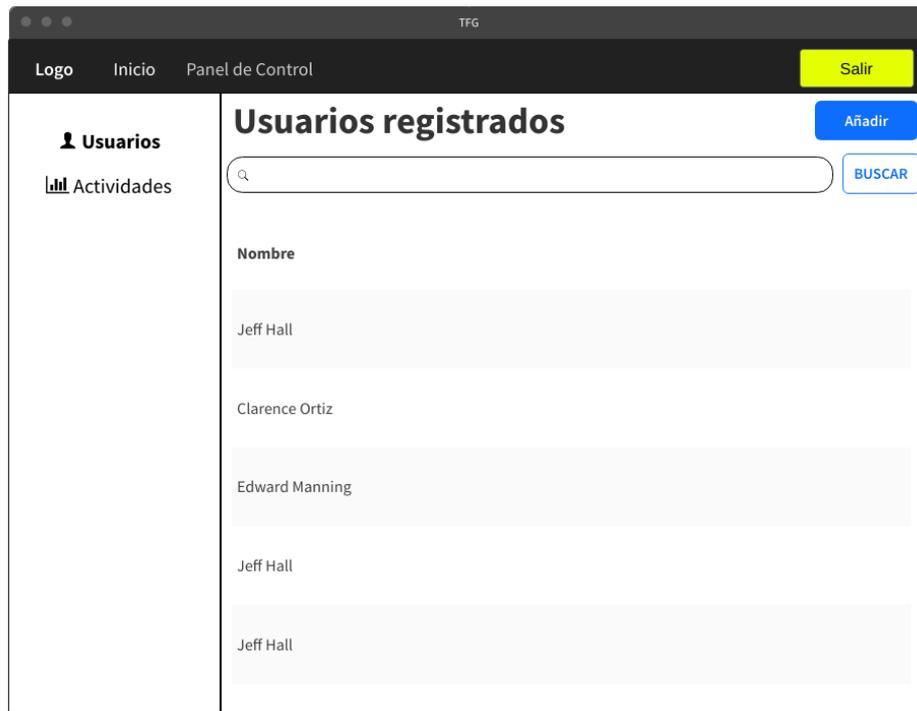


Figura 3.7: Página del panel de control para gestión de usuarios

En la pantalla representada por la figura 3.7, se muestra la barra de navegación de igual forma que en la página principal, para conseguir una estética consistente. A esto se añade, dentro del panel de control, una sección lateral a modo de navegación entre las distintas secciones que se tienen en el sistema para terapeutas identificados. En cuanto a la página de usuarios, tendremos un buscador y una lista de usuarios ordenados por fecha de registro ascendente, mostrando su nombre. Además se proporciona un botón para añadir usuarios al sistema.

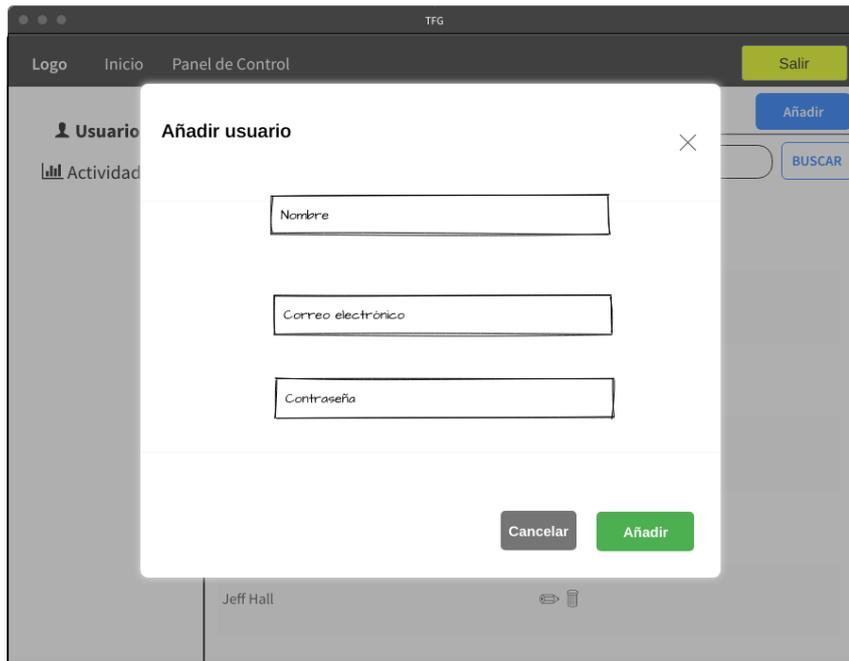


Figura 3.8: Modal para añadir un usuario dentro de la sección de gestión de usuarios

En el boceto desarrollado en la figura 3.8 se muestra una ventana emergente de tipo modal, la cual da sensación de que no estamos abandonando la pantalla de la que se venía. Se muestra un formulario en el que se solicita al terapeuta información esencial sobre el usuario como un nombre o pseudónimo, un correo para notificarle del registro y en caso de que se olvide de su contraseña, que pueda solicitarla, y una contraseña para su identificación en la plataforma.

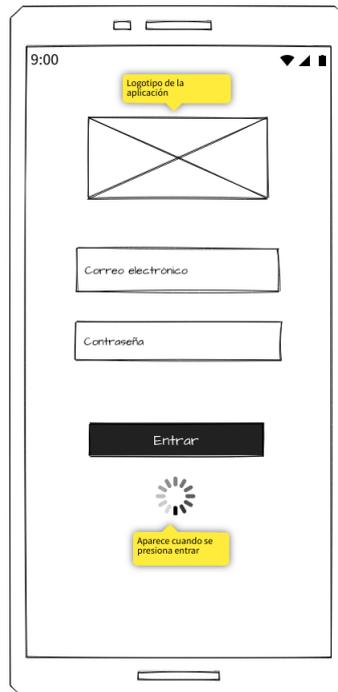


Figura 3.9: Pantalla inicio de sesión en Android

Comentando ahora los bocetos de la aplicación Android, el primero de ellos es la pantalla de inicio de sesión de la figura 3.9. En esta pantalla se encuentra el logotipo del proyecto junto a un formulario de identificación. Cuando el usuario introduce datos incorrectos, se le avisa mediante indicadores en el propio campo de texto. Una vez se pulse en el botón para entrar, se muestra un indicador de progreso y de corrección en las credenciales.



Figura 3.10: Pantalla de usuarios mayores identificados en la aplicación Android

Cuando se ha comprobado que las credenciales aportadas son correctas, se continúa con la pantalla de la figura 3.10. Está compuesta por una barra superior y una parte principal en la que se ven, englobados en una tarjeta, los datos para el usuario identificado. Además se proporciona un indicador de estado de conexión con el reloj.

En este punto, se vió necesario añadir a las tareas una que fuese la elaboración del logotipo del proyecto para poder introducirlo posteriormente en la implementación de los bocetos.

Para ello, se ha optado por usar un icono compuesto de una cabeza y unos rayos que simbolizan el estrés en personas. Como la temática del proyecto es medir el estrés en transportes, el logotipo acuñará estas palabras. Se muestra en la figura 3.11



Figura 3.11: Logotipo del proyecto

En cuanto a las librerías que aportan la funcionalidad de identificación y registro al framework de Ruby On Rails, se encuentran Devise y RodAuth. A continuación, en la tabla 3.4, se muestra una tabla comparativa de ambas librerías de acuerdo con

los resultados de sus repositorios de GitHub, en los que hay datos como el número de personas que participan activamente en el desarrollo del proyecto, cada cuanto se hace una nueva release, etc. (AwesomeRuby, 2022)

Rodauth		Devise
	Repositorio	
1,251	Estrellas	22,502
27	Observadores	481
70	Forks	5,422
31 días	Ciclo de lanzamientos	100 días
Hace 29 días	Última versión	Hace 6 meses
Hace 23 días	Último commit	Hace 4 días
Ruby	Language	Ruby
MIT License	License	MIT License

Cuadro 3.4: Comparación entre los repositorios de estas librerías de gestión de usuarios para RoR.

Como viendo que, a pesar que la popularidad de Rodauth no es tan elevada como la de Devise, su ciclo de lanzamientos y mantenimiento es mucho más activo, se ha optado por elegir RodAuth como encargado de gestionar tanto los accesos a la API como a la plataforma web. Además RodAuth permite, de forma sencilla, establecer en el archivo de configuración de rutas de RoR los niveles de acceso según la identificación de usuarios.

A continuación, se procede a la implementación de los bocetos desarrollados, una vez corregidos y aprobados por las tutoras del proyecto. Comenzaremos con el sistema web, indicando que se deben crear modelos tanto para almacenar a los terapeutas como a los usuarios mayores. De esto se genera el siguiente esquema de tablas en la base de datos.

En la figura 3.12 se aprecia al lado derecho las tablas autogeneradas por la librería RodAuth, que serán las encargadas de almacenar a los terapeutas que se registren en el sistema. Por otro lado, en la parte izquierda se visualiza la tabla necesaria para representar en nuestro sistema a los usuarios mayores. Se ha de enfatizar el hecho de que los datos que se recopilan se pueden anonimizar mediante el uso de un apodo o pseudónimo y que solo se recopilan los datos personales del usuario necesarios para el estudio. De esta forma aseguramos que se cumple la Ley de Protección de Datos (BOE, 2018). Tras la generación del modelo perteneciente a los usuarios mayores, necesitamos implementar los controladores y las vistas para implementar las funcionalidades que se piden.

En esta fase del proyecto se realiza la implementación de una API con acceso restringido mediante una autenticación Bearer y un token bearer. Para ello, se sigue una guía en la que se indica cómo hacerlo. (Polito, 2021) En esta iteración se implementa la identificación a través de la API mediante el envío de usuario y contraseña a un punto de entrada en la API, y que se reciba su token bearer con el que poder empezar a realizar solicitudes a la plataforma. Aunque aún no se implementen estas solicitudes, resulta necesario tener puntos de entradas para iniciar una actividad, añadir información sobre sensores a la actividad, añadir etiquetas a una actividad y terminar una actividad. Para esto se debe realizar las peticiones con una clave privada. Por ello, en esta iteración, la API contará con un punto de entrada para identificarse, solicitando usuario y contraseña. Si estas credenciales son correctas, se devolverá la información completa del usuario, conteniendo incluso su token bearer. Otro punto de entrada con el que contará la API en esta iteración será el de recuperar la contraseña, solicitando un correo electrónico, comprobando que dicho correo es válido y está registrado en el sistema. Si esto es así, se genera un correo que se envía a esta misma dirección con el recordatorio de contraseña y se responde en la API con un código 200 (Successful).

La autenticación Bearer es un esquema de autenticación HTTP que utiliza tokens de seguridad llamados token Bearer. Usaremos el esquema de autenticación Digest (Shekh-Yusef y col., 2015). Este token dará acceso al portador del mismo a la API de nuestro sistema web. Generalmente, un token Bearer es una cadena encriptada generada por el servidor en respuesta a un inicio de sesión. El dispositivo cliente de la API, debe enviar este token en las cabeceras de autorización cuando realice las peticiones de envío y recepción de datos. Cabe destacar que, para tener una seguridad efectiva, el sistema web debe estar bajo HTTPS con un certificado digital SSL, autenticando la identidad del sitio web y habilitando una conexión cifrada. Además, el token Bearer que se usa en nuestro sistema estará firmado mediante una clave privada generada por el servidor y que está almacenada mediante una variable de entorno.

Las razones por las que se ha elegido usar un token bearer se exponen en el siguiente listado:

- **Para verificar la identidad del usuario.** El usuario debe proporcionar una información a modo de verificación en la que el servidor pueda confiar. A esto se le llama token y está compuesto por datos en formato JSON encapsulados y encriptados por un algoritmo hash.
- **Para cumplir con el estándar OAuth 2.0.** Este es el estándar abierto para autorización en web. Está diseñado para permitir que una aplicación o un sitio web puedan acceder a recursos, en nombre de un usuario, que se encuentren en otras aplicaciones web.

Por tanto, la autorización y el acceso a datos de la API se hará bajo este token bearer, de modo que un usuario mayor no pueda visualizar los datos de otro usuario

mayor, tampoco se pueda suplantar la identidad y tampoco se puedan hacer registros en el sistema sin que se deje constancia de qué usuario los ha realizado.

Los puntos de entrada en la API para esta iteración serían los que se muestran en la tabla 3.5

TIPO	DIRECCIÓN	DESCRIPCIÓN
GET	/users	Para obtener los datos del usuario identificado. Se necesita aportar un token Bearer.
GET	/users/get_api_key	Para obtener el token Bearer dados un correo y una contraseña registradas en el sistema.
GET	/users/password_recovery	Para solicitar un correo con el recordatorio de contraseña dado un correo electrónico.

Cuadro 3.5: Puntos de entrada de la API en la iteración 1

En las figuras 3.13, 3.14, 3.15, 3.16 y 3.18 se muestra la visualización tras la implementación en la aplicación web.

En cuanto a las pruebas de aceptación realizadas para el sistema web, se han desarrollado tests de Ruby que comprueben que la funcionalidad desarrollada cumple con los requerimientos. Los resultados pueden observarse en la tabla de la figura 3.19 junto con la tabla de la figura 3.6.

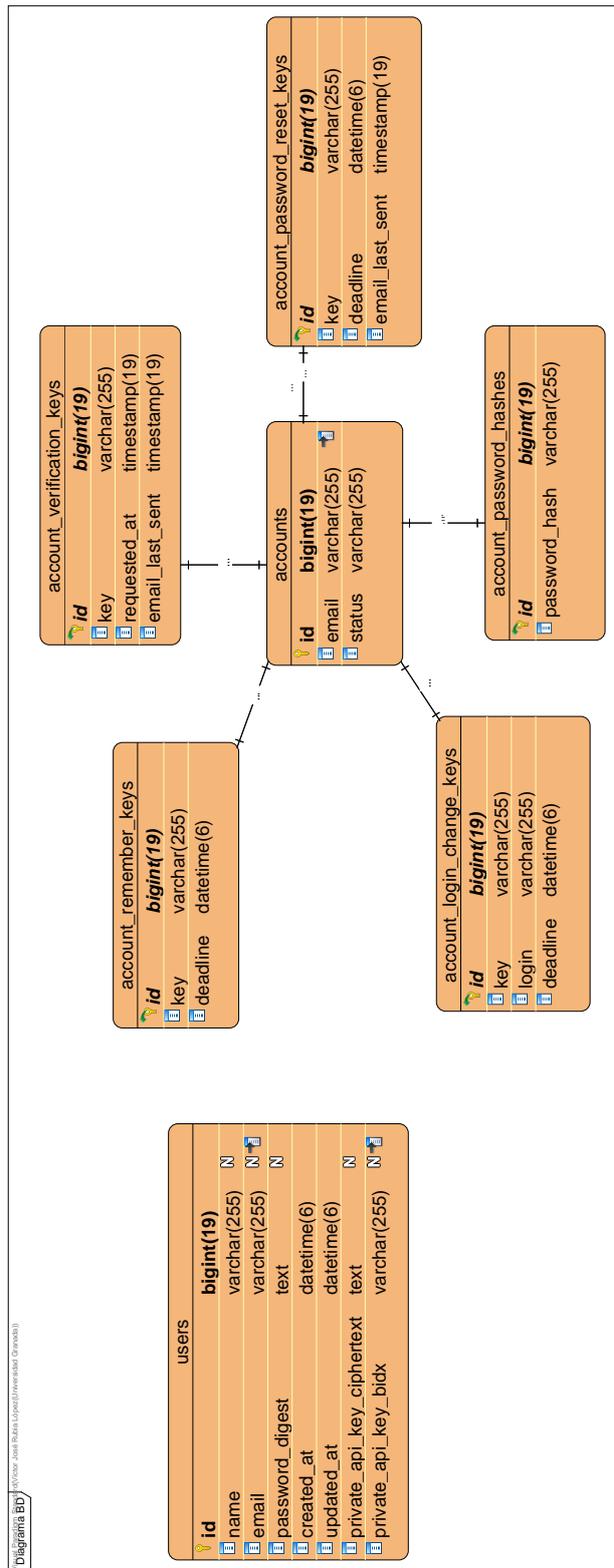


Figura 3.12: Esquema de tablas generadas en la Base de Datos por RoR en la iteración 1

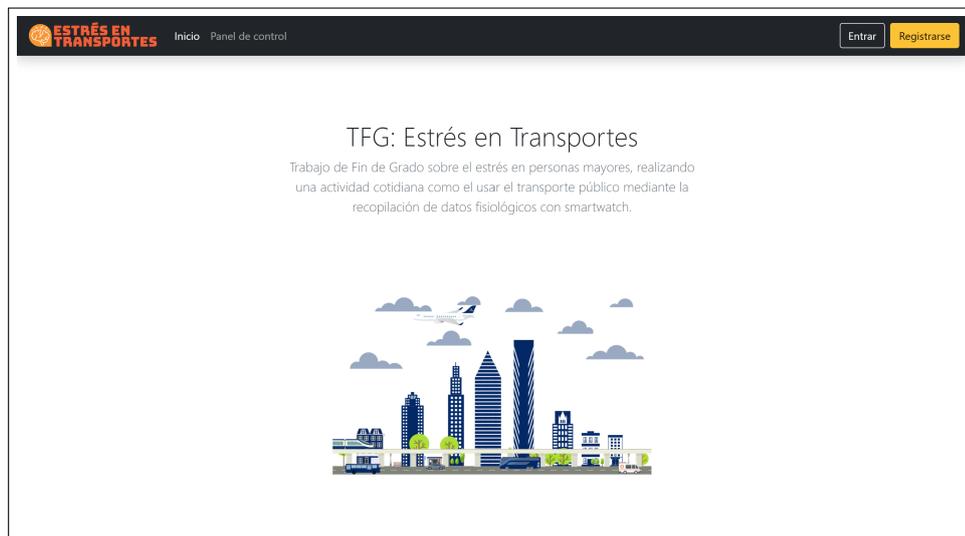


Figura 3.13: Página principal del sistema web

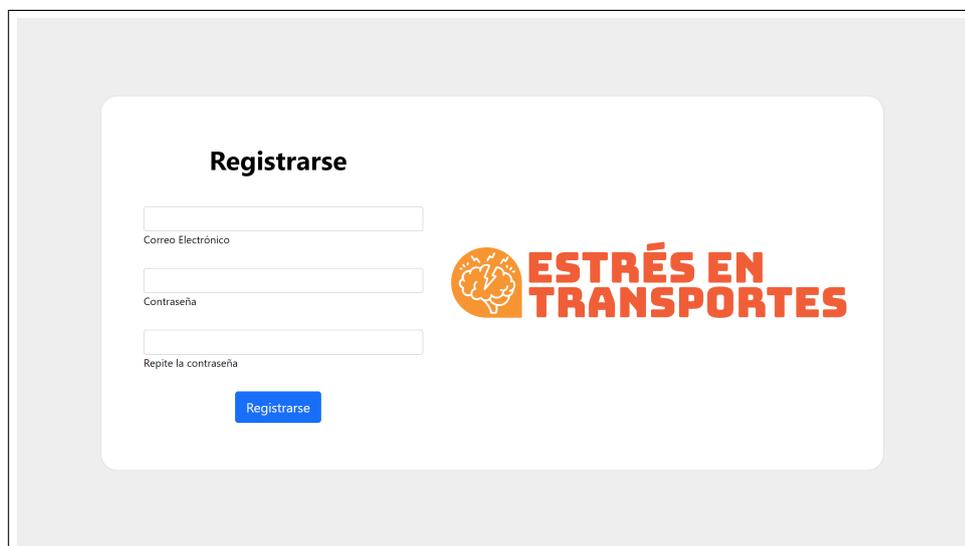
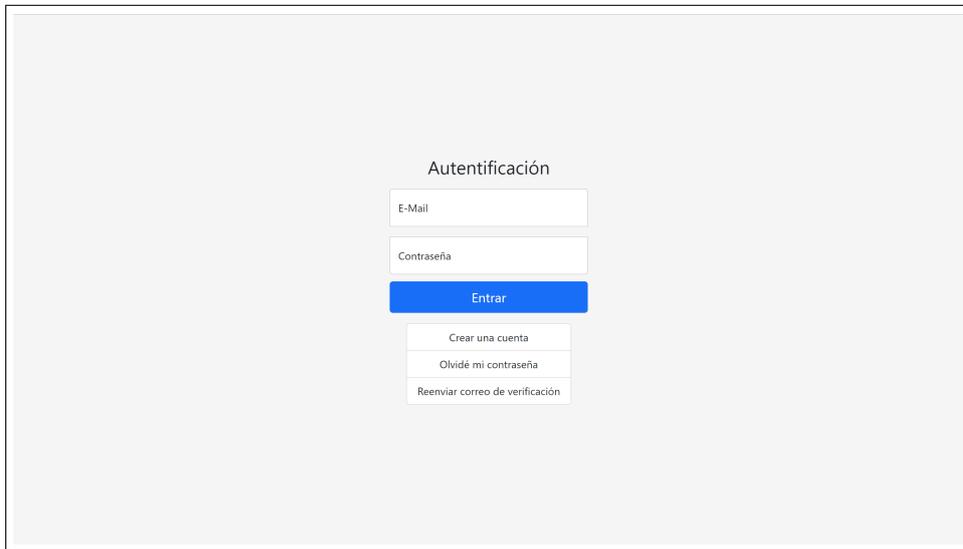
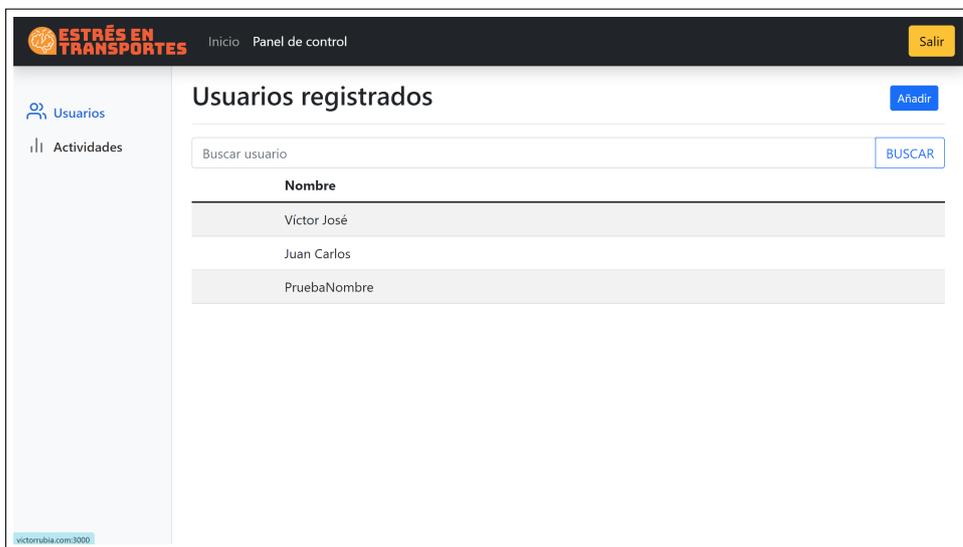


Figura 3.14: Página de registro en el sistema web



The screenshot shows a login form titled "Autenticación". It contains two input fields: "E-Mail" and "Contraseña". Below these fields is a blue "Entrar" button. Underneath the "Entrar" button are three links: "Crear una cuenta", "Olvidé mi contraseña", and "Reenviar correo de verificación".

Figura 3.15: Página de identificación en el sitio web



The screenshot shows the "Usuarios registrados" page in the control panel. The page has a dark header with the logo "ESTRÉS EN TRANSPORTES" and navigation links "Inicio" and "Panel de control". A "Salir" button is in the top right. The main content area has a search bar "Buscar usuario" with a "BUSCAR" button and an "Añadir" button. Below the search bar is a table with the following data:

Nombre
Victor José
Juan Carlos
PruebaNombre

At the bottom left, there is a small text "victornbia.com3000".

Figura 3.16: Página de gestión de usuarios en el panel de control del sitio web



The image shows a modal dialog titled "Añadir usuario" (Add user) overlaid on a control panel. The dialog contains three input fields: "Nombre" (Name), "Correo electrónico" (Email), and "Contraseña" (Password). At the bottom right of the dialog are two buttons: "Cancelar" (Cancel) and "Añadir" (Add).

Panel de control

Usuarios registrados

Añadir usuario

Nombre

Víctor José

Juan Carlos

PruebaNombre

Añadir usuario

Nombre

Correo electrónico

Contraseña

Cancelar Añadir

Figura 3.17: Diálogo modal para registrar un usuario mayor en la plataforma



Figura 3.18: Correo que se envía al usuario mayor para que conozca sus datos de acceso

TestRun	Total	Passed
All Tests - Generated by RubyMine on 28/2/22 9:38	12	12
DashboardControllerTest		
test_required_authentication	296ms	passed
test_should_get_index	279ms	passed
test_should_get_users	238ms	passed
Api::V1::UsersControllerTest::Unauthorized		
test_should_not_get_user	157ms	passed
UsersControllerTest		
test_should_create_user	216ms	passed
DashboardTest		
test_searching_an_user	289ms	passed
test_creating_an_user	350ms	passed
WelcomeControllerTest		
test_should_get_index	30ms	passed
AuthenticationTest		
test_creating_an_account	49ms	passed
test_logging_in_and_logging_out	298ms	passed
Api::V1::UsersControllerTest::Authenticated		
test_should_get_user	178ms	passed
test_should_get_user's_apiKey	150ms	passed

Figura 3.19: Tests realizados sobre la Aplicación Web RoR en la iteración 1

ID	PRUEBA DE ACEPTACIÓN	ESTADO	OBSERVACIONES
1	Poder registrarse como terapeuta introduciendo datos de correo y contraseña	Correcto	Comprobado en test unitario <i>test_creating_an_account</i> .
2	Registrarse con un correo que está previamente registrado debe dar error	Correcto	Comprobado de forma manual, se muestra un error.
3	Se manda un correo de confirmación a la dirección proporcionada sin error.	Correcto	Comprobado de forma manual
4	Se indica mediante un pop-up que se ha realizado el registro correctamente.	Correcto	Comprobado de forma manual, se muestra un pop-up.
5	Iniciar sesión con las credenciales correctas y que no exista ningún error.	Correcto	Comprobado en test unitario <i>test_logging_in</i> .
6	Iniciar sesión con las credenciales correctas y que se muestre un pop-up indicando el correcto inicio de sesión.	Correcto	Comprobado de forma manual, se muestra un pop-up.
7	Iniciar sesión con credenciales incorrectas debe mostrar un pop-up indicando error.	Correcto	Comprobado de forma manual, se muestra un pop-up
8	Iniciar sesión con credenciales correctas debe permitir el acceso al Panel de Control.	Correcto	Comprobado mediante test unitario <i>test_required_authentication</i> y <i>test_should_get_index</i> .
9	Cuando se inicie sesión con las credenciales correctas debe poder accederse al Panel de Control.	Correcto	Comprobado mediante test unitario <i>test_should_get_index</i> .
10	Si no hay ningún terapeuta identificado y se intenta acceder al Panel de Control, se debe denegar el acceso mediante mostrar el formulario de identificación.	Correcto	Comprobado mediante test unitario <i>test_required_authentication</i> .
11	El terapeuta debe estar identificado para acceder al apartado de gestión de usuarios.	Correcto	Comprobado mediante test unitario <i>test_should_get_users</i> y <i>test_required_authentication</i> .

12	Se listan los usuarios registrados actualmente en el sistema.	Correcto	Comprobado mediante test unitario <i>test_should_get_users</i>
14	El terapeuta debe estar identificado para añadir un usuario al sistema.	Correcto	Comprobado mediante test unitario <i>test_required_authentication</i> .
15	Se muestra un botón en la sección de gestión de usuarios para añadir un usuario.	Correcto	Comprobado mediante test unitario <i>test_creating_an_user</i> y <i>test_should_create_user</i> .
16	Introducir un correo electrónico ya existente hará que no se guarde el usuario.	Correcto	Comprobado de forma manual, se muestra un pop-up de error.
17	Introducir un correo electrónico no válido hará que no se guarde el usuario.	Correcto	Comprobado de forma manual, se muestra un pop-up de error.
18	No introducir un nombre/pseudónimo hará que no se guarde el usuario.	Correcto	Comprobado de forma manual, se muestra un pop-up de error.
19	No introducir una contraseña hará que no se guarde el usuario.	Correcto	Comprobado de forma manual, se muestra un pop-up de error.
20	El terapeuta debe estar identificado para visualizar los datos de los usuarios.	Correcto	Comprobado mediante test unitario <i>test_required_authentication</i> .
21	Se debe mostrar el nombre de cierto usuario.	Correcto	Comprobado mediante test unitario <i>test_should_get_users</i> .
23	Se debe mostrar la contraseña de cierto usuario.	Correcto	Comprobado de forma manual con la correcta recepción del correo con las credenciales.

Cuadro 3.6: Resultados de realizar las pruebas de aceptación de RoR para la iteración 1

A continuación, se pasa al desarrollo de la aplicación móvil. Atendiendo a las diferentes opciones que tenemos disponibles para el desarrollo, se opta por comenzar a desarrollar en Flutter, por tener conocimientos de haberlo usado anteriormente y por ser un framework que permite desarrollar aplicaciones de calidad de una forma bastante rápida. A medio desarrollar la aplicación surgió un problema y era el cómo este framework implementaba la conexión con relojes inteligentes. El hecho es que no lo soporta

de forma nativa, aunque sí que existen algunas librerías que ofrecen cierto soporte y las guías oficiales de desarrolladores de Android no indican cómo hacerlo de esta forma, si no que únicamente indican como hacerlo mediante el desarrollo de una aplicación Android nativa mediante Java o Kotlin.

Por esta razón se decide abandonar el desarrollo de la aplicación móvil en Flutter y se comienza a estudiar cómo realizarla de forma nativa mediante Kotlin. Se establece que seguiremos el patrón arquitectónico Model-View-ViewModel (MVVM) ya que es el que recomienda Google y el que más extendido está dentro de sus ejemplos. La estructura a seguir en el proyecto se expone en el diagrama de paquetes de la figura 3.20. En ella se aprecia cómo se separa la capa de datos de la capa de dominio y de la capa de presentación.

El paquete correspondiente a la capa de datos almacena los repositorios que contienen las funciones y los modelos para la gestión de los datos con la API proporcionada por el sistema web y el almacenamiento local, ya que se quiere que la aplicación mantenga la sesión aunque se cierre la aplicación. Por otro lado, la capa de dominio es la encargada de contener los casos de uso y funciona a modo de interfaz con los repositorios de la capa de datos. El paquete que se corresponde con la capa de presentación contiene las dependencias necesarias para el buen funcionamiento de la aplicación, las pantallas, que se componen de vistas, y la ejecución de los casos de uso. Para la conexión con la base de datos se usa la librería *Retrofit 2.0*.

```

1 /**
2  * Get the user information from the backend.
3  *
4  * @param email The user email.
5  * @param password The user password.
6  * @return The user information.
7  */
8 @GET("users/get_api_key/")
9 suspend fun getUserInfo(@Query("email") email : String ,
10                        @Query("password_digest") password : String ) : Response<
11                            ↳ User>
12 /**
13  * Requests a password remember to the backend.
14  *
15  * @param email The user email to request the password remember.
16  * @return The response of the request.
17  */
18 @GET("users/password_recovery/")
19 suspend fun requestPasswordRecovery(@Query("email") email : String ) : Response<*>

```

Listing 3.3: Funciones pertenecientes a la app móvil de la interfaz de conexión con la API en la iteración 1.

En la figura 3.3 se aprecian las funciones necesarias para que la aplicación móvil se comunique con la API. La primera de ellas se usa para obtener la información del usuario. Esto es, la clave API para que el usuario pueda hacer operaciones de envío de datos y de inicio de actividad en la plataforma. Para ello, basta con identificarse mediante correo y contraseña para poder recibir dicha información. Esta es la función que se ejecuta cuando el usuario se identifica en la figura 3.9. Por otro lado, la función de solicitar recuperación de contraseña será la que se ejecute en la figura 3.48, para la cual es necesario enviar el email que solicita la recuperación a la API. La API se encargará de comprobar si este correo es válido y está registrado en el sistema, por lo que el dispositivo móvil no haría ninguna comprobación más allá de la corrección gramática de la dirección de correo. Además, como la API no devuelve información que se pueda representar internamente con un modelo, se toma cualquier tipo, para comprobar si la solicitud se ha completado o no.

Como casos de uso en la aplicación móvil, se han desarrollado las funcionalidades de obtener un usuario, el de recuperar contraseña y el de eliminar usuario de la memoria local. Este último es el encargado de eliminar la información del almacenamiento local del

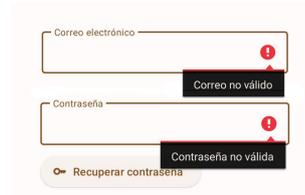
usuario identificado cuando pulsa sobre el botón de cerrar la sesión, ya que, cuando un usuario inicia sesión la aplicación guarda su sesión en la base de datos local proporcionada por la librería Room, de Google (Google, 2022c). De este modo, hasta que el usuario mayor no cierre sesión en la aplicación móvil, no se le volverá a pedir sus credenciales aunque cierre la aplicación en el administrador de tareas del teléfono. Así facilitamos al usuario la utilización de la aplicación, al estilo de otras que existen en el mercado como Twitter, Instagram o Facebook.

El resultado de la implementación se muestra en las figuras 3.21a 3.21b y 3.21c. En esta implementación se puede apreciar que se ha optado por seguir las directrices de diseño de Material Design 3.0, la versión más actualizada de Google. Las razones para seguirlas son, a parte de que es su versión más reciente, la facilidad del usuario a la hora de reconocer elementos de la interfaz, ya que se unifica el diseño de la aplicación con el resto de las aplicaciones del terminal y porque no existe la necesidad de establecer una paleta de colores para la aplicación, ya que el sistema tiene unos predefinidos y este adapta la tonalidad de la interfaz dependiendo de los ajustes del usuario.

En las figuras 3.22, 3.23 y 3.24 se muestran los diagramas de clases obtenidos de la aplicación Android en esta iteración.



(a) Pantalla Principal de App Android



(b) Comprobación de corrección gramatical de campos de identificación



(c) Pantalla de Identificados App Android

Figura 3.21: Resultados de implementación en Android.

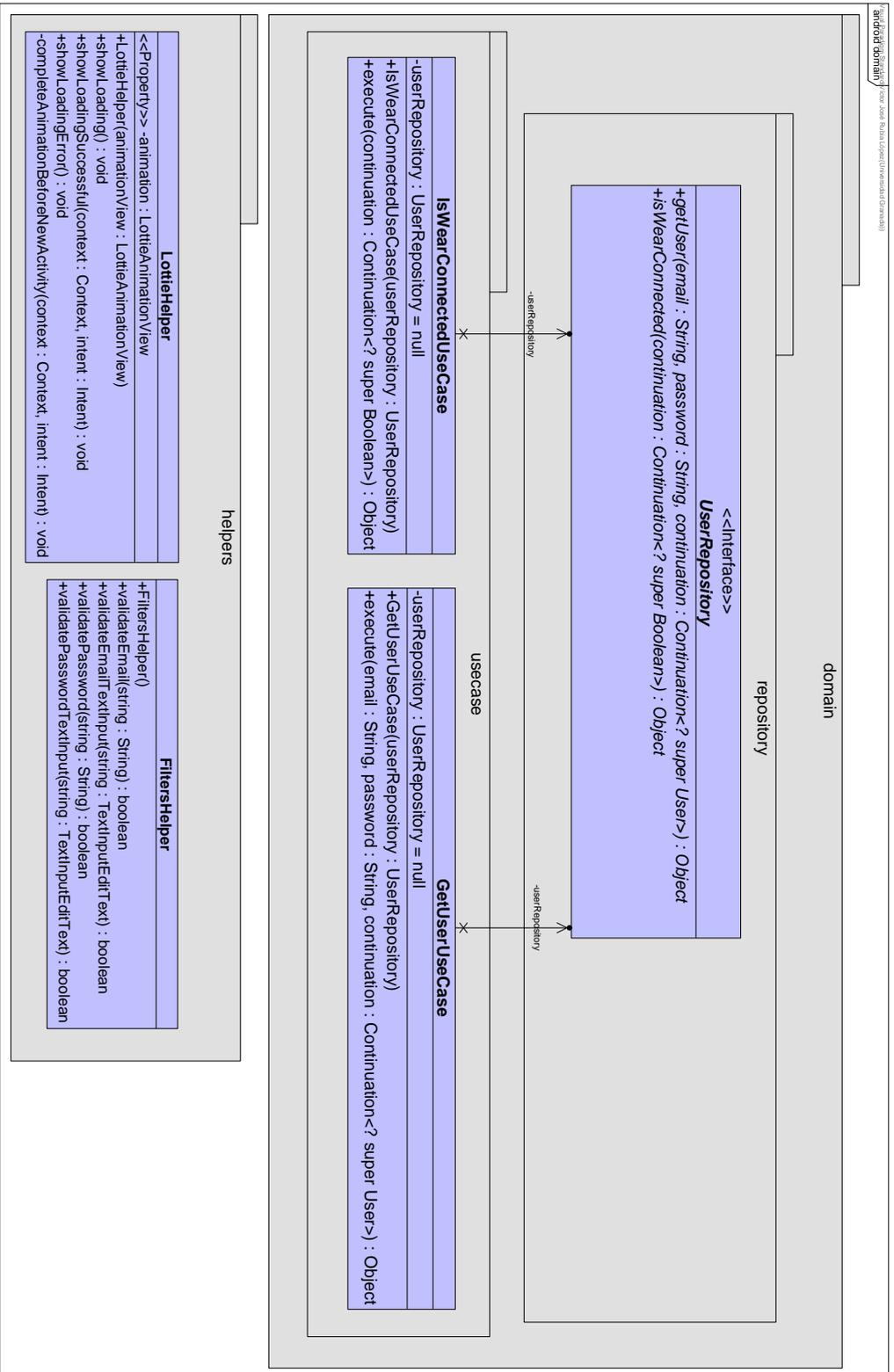


Figura 3.23: Diagrama de clases correspondiente al paquete *domain* de la aplicación Android en la iteración 1

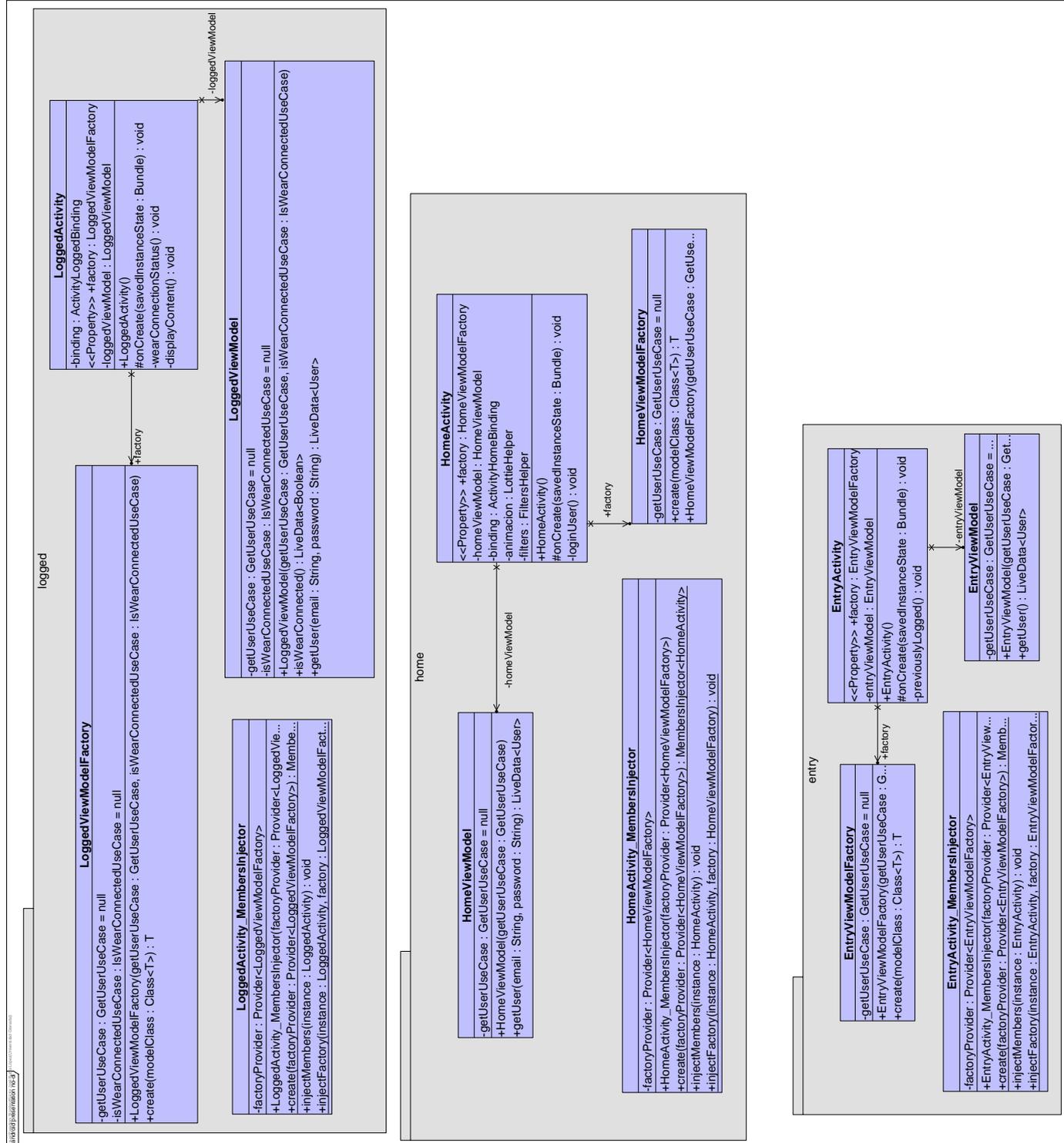


Figura 3.24: Diagrama de clases correspondiente al paquete *presentation* de la aplicación Android en la iteración 1

En cuanto a las pruebas de aceptación, se han llevado a cabo tanto mediante la experimentación, comprobando que la funcionalidad pedida se cumple, mediante la implementación de tests unitarios y de componentes. Se muestran los tests unitarios y de componentes de la aplicación Android relacionados con las pruebas de aceptación asociadas a recibir una respuesta correcta del servidor cuando se le solicita la identificación junto con el envío correcto de las credenciales al servidor, se comprueba también las funciones para validar el correo junto con la contraseña y por último se realiza una prueba de componente en la que se muestra en la pantalla de identificados la información asociada al usuario que se ha identificado. Los resultados se pueden consultar en la tabla de la figura 3.25.

TestRun	Total	Passed
Unit Tests Mobile - Generated by IntelliJ IDEA on 1/3/22 11:30	5	5
TFGServiceTest		
TFGServiceTest.getUser_receivedResponse_correctContent	36ms	passed
TFGServiceTest.getUser_sentRequest_receivedExpected	15ms	passed
FiltersHelperTest		
FiltersHelperTest.validatePassword_returnsCorrectVerdict	7943ms	passed
FiltersHelperTest.validateEmail_returnsCorrectVerdict	88ms	passed
HomeViewModelTest		
HomeViewModelTest.getUser_returnCurrentRecord	312ms	passed

Figura 3.25: Tests Unitarios y de Componentes de la Aplicación Android en la iteración 1

Las pruebas de aceptación se han llevado a cabo en base a estos test unitarios y de componentes realizados y, en los casos que no se han podido llevar a cabo, se han realizado comprobaciones manuales. Se pueden observar los detalles en la tabla 3.7

3.3.5. Retrospectiva de la iteración

En esta iteración, el desarrollador se ha visto con más carga de trabajo de lo previamente establecido, ya que no se había tenido en cuenta que el despliegue inicial del sistema iba a ser tan costoso y he tenido que instruirme más acerca de cómo realizar un *Dockerfile* que cumpla las necesidades concretas del producto en lugar de usar imágenes oficiales que se pueden obtener de *DockerHub*. Por otro lado, causó retraso el hecho de haber comenzado el desarrollo de la aplicación Android mediante el Framework Flutter para realizar un desarrollo rápido y de calidad, ya que tras llevar algo avanzado el desarrollo, descubrí que, mediante esta capa de alto nivel, no se tiene tanto control en cuanto a la comunicación a bajo nivel entre reloj inteligente *WearOS* y este dispositivo Android, por lo que he tenido que rescindir del uso del mismo. He tenido que optar, en base a las lecturas de las guías de desarrollo de *Android Developers*, por la realización de una aplicación Android Nativa en lenguaje Kotlin y siguiendo el patrón arquitectóni-

ID	PRUEBA DE ACEPTACIÓN	ESTADO	OBSERVACIONES
24	Introduciéndose un correo no válido la aplicación no procede a iniciar sesión y se lo indica al usuario.	Correcto	Comprobado mediante test unitario <code>validateEmail_returnsCorrectVeredict</code> .
25	No introduciendo nada tanto en el campo de correo como de contraseña hará que no se inicie sesión y se le indicará al usuario.	Correcto	Comprobado mediante test unitario <code>validatePassword_returnsCorrectVeredict</code> .
26	Introduciendo las credenciales correctas la aplicación debe mostrar un check verde a modo de feedback y pasar a la siguiente pantalla.	Correcto	Comprobado mediante tests unitarios <code>getUser_returnCurrentRecord</code> , <code>getUser_receivedResponse_correctContent</code> y <code>getUser_sentRequest_receivedExpected</code> .

Cuadro 3.7: Resultados de realizar las pruebas de aceptación de Android para la iteración 1

co MVVM, siendo ambos nuevos para el desarrollador, lo que requirió una formación previa.

Dejando a un lado lo que ha llevado más trabajo de lo que se preveía en esta iteración, hablaremos sobre los objetivos que sí que hemos cumplido. Se detallan, a continuación, los objetivos que se han cumplido para esta iteración.

- La realización de unos bocetos con calidad, supervisados por las tutoras, con su correspondiente navegabilidad.
- El despliegue del sistema web con la realización de las pantallas asociadas a las tareas de esta iteración.
- El registro e identificación de terapeutas y una API con acceso bajo clave privada Bearer.
- El diseño de la estructura de tablas para la base de datos, el diagrama de paquetes para la aplicación Android y su correspondiente diagrama de clases.
- Se ha conseguido implementar las tareas de desarrollo relacionadas con la aplicación móvil.
- Se ha investigado sobre la seguridad en sistemas web, especialmente en cuanto al acceso a APIs y seguridad en cuanto a la realización de *Dockerfile* y *Docker Compose*.

3.4. Segunda iteración

3.4.1. Tarjetas de las Historias de Usuario

Se da comienzo con la extracción de las historias de usuario correspondientes la segunda iteración y el desarrollo de las tarjetas de historias de usuario.

Identificador: HU-20	Iniciar una actividad	
Descripción: Como usuario quiero poder iniciar una actividad desde el reloj inteligente para medir el nivel de estrés.		
Estimación: 5	Prioridad: 9	Entrega: 2
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Se debe obtener el ID de la actividad iniciada. ▪ Únicamente se iniciará una actividad cuando el usuario mayor ha iniciado sesión en el teléfono móvil asociado al reloj. 		
Observaciones:		

Identificador: HU-11	Visualizar niveles de estrés para una actividad en plataforma web	
Descripción: Como terapeuta quiero visualizar una actividad mediante un gráfico que indique el nivel de estrés para que se lea de forma rápida.		
Estimación: 5	Prioridad: 10	Entrega: 2
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El terapeuta debe estar identificado para visualizar los niveles de estrés de una actividad concreta. ▪ El gráfico muestra si existe alguna medición para dicha actividad. ▪ El gráfico debe enlazar los puntos mediante una línea suave, incluso si no existe medición en tiempos correlativos. 		
Observaciones:		

Identificador: HU-10	Categorizar las actividades en la pantalla de visualización.	
Descripción: Como terapeuta quiero hacer el seguimiento de los usuarios mediante una sección de actividades categorizadas y ordenadas por fecha descendente.		
Estimación: 5	Prioridad: 11	Entrega: 2
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El terapeuta debe estar identificado para visualizar las actividades categorizadas. ▪ Cada tipo de actividad debe mostrarse en la columna asociada a su categoría. ▪ El nombre de la actividad debe coincidir con la categoría que se muestra. 		
Observaciones:		

Identificador: HU-21	Categorías de actividad en el sistema	
Descripción: Como usuario quiero clasificar las actividades en “Viajar en bus”, “Viajar en metro” y “Viajar en tren” para poder registrar el tipo de actividad antes de iniciar una actividad.		
Estimación: 2	Prioridad: 12	Entrega: 2
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Todas las actividades que se registren deben tener una categoría igual a uno de estos tres tipos de actividad. ▪ No puede existir una actividad que se registre en el sistema con un nombre distinto a las tres categorías 		
Observaciones:		

Identificador: HU-32	Añadir usuario a la plataforma	
Descripción: Como usuario quiero seleccionar en un listado los tipos de actividades para iniciar una actividad.		
Estimación: 1	Prioridad: 13	Entrega: 2
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El nombre de la actividad iniciada debe coincidir con el tipo de actividad seleccionado en la lista. ▪ Solo se debe poder seleccionar una única categoría. 		
Observaciones:		

Identificador: HU-24	Visualizar datos de usuarios del sistema	
Descripción: Como usuario quiero poder añadir etiquetas en el transcurso de una actividad para etiquetar instantes de tiempo y conocer las razones dadas un nivel de estrés.		
Estimación: 5	Prioridad: 14	Entrega: 2
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Se deben poder seleccionar una o varias etiquetas del listado. ▪ Una vez terminado el registro, se deben almacenar en el servidor. ▪ La etiqueta pulsada se debe corresponder a la etiqueta que se quería seleccionar. ▪ Se debe tener una actividad en transcurso. ▪ Un usuario mayor debe estar identificado en la aplicación. 		
Observaciones:		

Identificador: HU-12	Visualizar credenciales de los usuarios	
Descripción: Como terapeuta quiero visualizar una actividad mediante las etiquetas mostrándolas con la hora y de acuerdo con su categoría para que se lea de forma rápida.		
Estimación: 3	Prioridad: 15	Entrega: 2
Pruebas de aceptación: <ul style="list-style-type: none">▪ El terapeuta debe estar identificado para visualizar la actividad y sus etiquetas asociadas.▪ Las etiquetas deben mostrarse bajo su categoría correspondiente.▪ La hora que se muestra debe corresponder a la hora del registro de dichas etiquetas.		
Observaciones:		

3.4.2. Descomposición de las H.U. en tareas de desarrollo

Tras escribir las pruebas de aceptación de las historias de usuario que se abarcan en la segunda iteración, se descompone cada una de ellas en tareas de desarrollo.

HU-20	Como usuario quiero poder iniciar una actividad desde el reloj inteligente para medir el nivel de estrés. .	5 PH
-------	---	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-28	Realizar un boceto de las pantallas que llevan hasta el inicio de una actividad en el reloj.	1
T-29	Implementar los bocetos de las pantallas que inician una actividad en el reloj.	1.5
T-30	Conectar la funcionalidad del reloj que inicia una actividad con un punto de entrada en la API.	1.5
T-31	Realizar pruebas de aceptación asociadas a iniciar una actividad desde el reloj.	1

Observaciones:

HU-11	Como terapeuta quiero visualizar una actividad mediante un gráfico que indique el nivel de estrés para que se lea de forma rápida.	5 PH
-------	--	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-32	Realizar un boceto de la forma en la que se visualiza el gráfico de los niveles de estrés para una actividad.	0.5
T-33	Investigar gráficos que poder implementar en el sistema web	0.5
T-34	Implementar el boceto realizado para la visualización del gráfico de los niveles de estrés de una actividad.	1
T-35	Obtención de los datos del nivel de estrés para una actividad.	2.5
T-36	Realizar las pruebas de aceptación asociadas a los gráficos de visualización de niveles de estrés de una actividad.	0.5

Observaciones:

HU-10	Como terapeuta quiero hacer el seguimiento de los usuarios mediante una sección de actividades categorizadas y ordenadas por fecha descendente.	5 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-37	Realizar bocetado de la sección de actividades dentro del panel de control.	1
T-38	Implementar bocetado de la sección de listado de actividades	3
T-39	Realizar pruebas de aceptación asociadas a la sección de lista de actividades.	1
Observaciones:		

HU-21	Como usuario quiero clasificar las actividades en “Viajar en bus”, “Viajar en metro” y “Viajar en tren” para poder registrar el tipo de actividad antes de iniciar una actividad..	2 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-40	Implementar categorización en la aplicación web	1
T-41	Realizar las pruebas de aceptación asociadas a la categorización de actividades en el sistema.	0.5
Observaciones:		

HU-32	Como usuario quiero seleccionar en un listado los tipos de actividades para iniciar una actividad..	1 PH
-------	---	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-42	Realizar el bocetado del listado de actividades del reloj,	0.25
T-43	Implementar el listado de actividades en la aplicación del reloj	0.5
T-44	Realizar las pruebas de aceptación asociadas al listado de actividades al iniciar una actividad en el reloj.	0.25

Observaciones:

HU-24	Como usuario quiero poder añadir etiquetas en el transcurso de una actividad para etiquetar instantes de tiempo y conocer las razones dadas un nivel de estrés.	5 PH
-------	---	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-45	Realizar bocetado de las pantallas del reloj en cuanto a añadir etiquetas a una actividad en transcurso	1
T-46	Implementar bocetado asociado a las pantallas del reloj que permiten registrar etiquetas a una actividad en transcurso.	2
T-47	Conectar el registro de etiquetado a un punto de entrada en la API para que quede registrado en el sistema.	1
T-48	Realizar las pruebas de aceptación asociadas al añadir etiquetas a una actividad en el reloj.	1

Observaciones:

HU-12	Como terapeuta quiero visualizar una actividad mediante las etiquetas mostrándolas con la hora y de acuerdo con su categoría para que se lea de forma rápida..	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-49	Realizar bocetado de la pantalla de detalle de una actividad mostrando las etiquetas registradas para dicha actividad.	0.5
T-50	Implementar bocetado de la pantalla de detalle de una actividad mostrando las etiquetas registradas de dicha actividad.	1.5
T-51	Añadir las etiquetas al servidor.	0.5
T-52	Realizar las pruebas de aceptación asociadas al mostrar las etiquetas de una actividad en la pantalla de detalle.	0.5
Observaciones:		

3.4.3. Planificación temporal de la segunda iteración

En la figura 3.26 se puede visualizar un diagrama de Gantt para la planificación detallada de la segunda iteración.

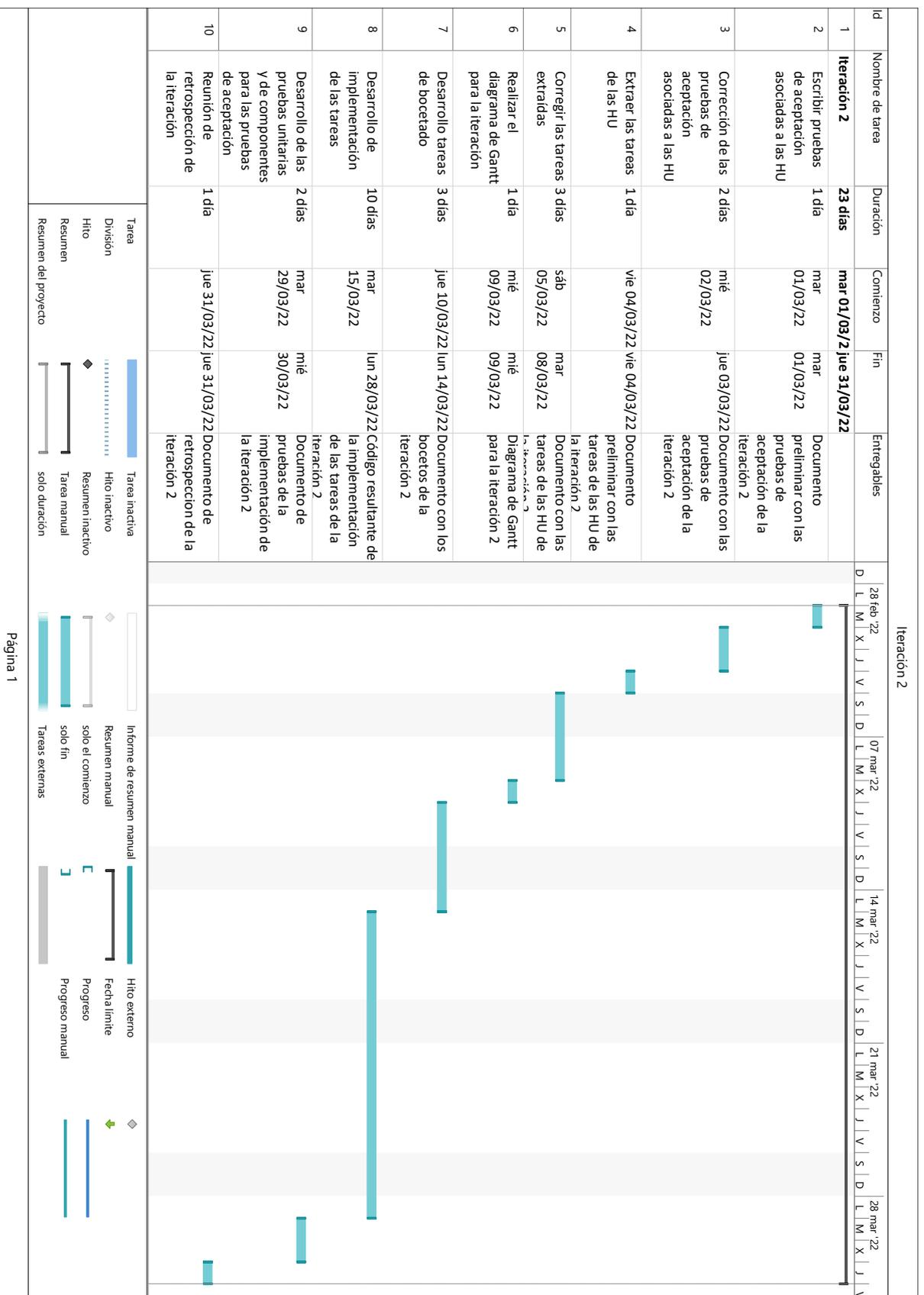


Figura 3.26: Diagrama de Gantt para la planificación de las tareas de la iteración w

3.4.4. Realización de las tareas de desarrollo

Para la realización de las tareas asociadas a la segunda iteración, comenzamos con el bocetado de la aplicación web. En la figura 3.27 se muestran los bocetos correspondientes a la sección de las actividades registradas en el sistema que los usuarios mayores han llevado a cabo.

Se muestra una página muy similar a la desarrollada para la gestión de los usuarios, en la que se visualizan los usuarios mayores registrados en el sistema y un botón para visualizar sus actividades. Pulsándolo, navegaremos a una página donde veremos las tres columnas que clasifican las actividades registradas en el sistema por el usuario seleccionado.

Se muestra un botón para poder acceder a visualizar los detalles de la actividad. Si se pulsa sobre el botón de visualizar los detalles de una actividad, el sistema nos mostrará una página donde se encuentra el gráfico que muestra los niveles de estrés lo largo de la escala de tiempo por la cual ha transcurrido la actividad. Asociada también a la actividad, encontramos una tabla de etiquetas de estados de la actividad, que se muestran por orden de instante de tiempo creciente. Se categorizan las distintas etiquetas según su tipología para mejorar su lectura.

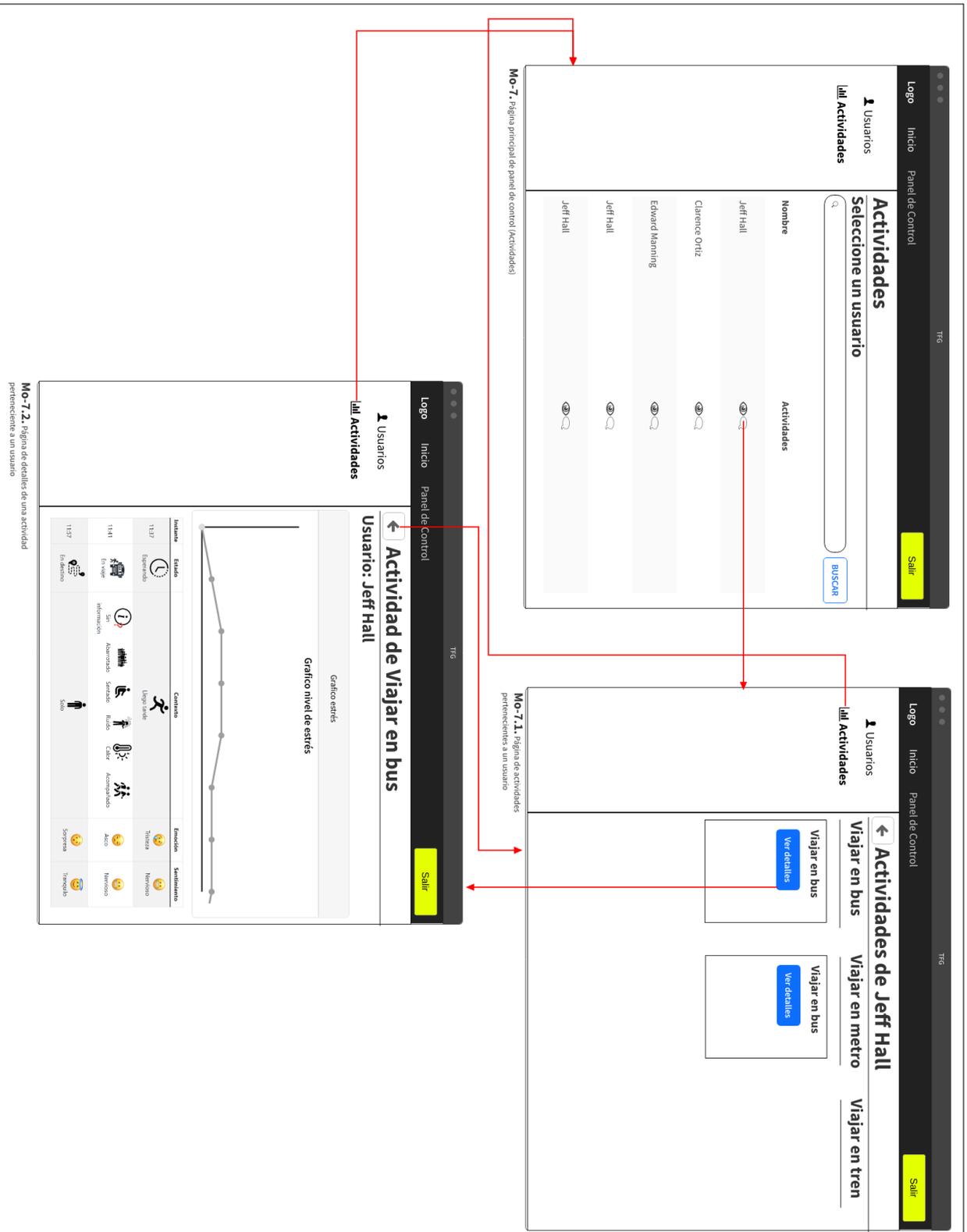


Figura 3.27: Bocetado y navegabilidad de la sección de gestión de actividades de usuarios mayores en la plataforma web

A continuación se desarrollan los bocetos relacionados con las tareas del reloj inteligente. Antes de comenzar a realizarlos se estudió la forma y tamaño en la que se disponían los elementos en la pantalla en dispositivos WearOS y qué elementos de interfaz existían. Además, junto a las tutoras, se estableció una serie de etiquetas predefinidas que nos sirven para el desarrollo de la plataforma. En la figura 3.28 se muestra el bocetado y navegación completos de una primera aproximación de la aplicación del reloj que fueron corregidos y comentados con las tutoras.

Comenzaremos comentando el bocetado realizado para la aplicación del reloj con las pantallas que aparecen cuando el usuario abre la aplicación. A modo de indicación se le mostrará al usuario que se está intercambiando datos entre este dispositivo y el teléfono vinculado, para obtener el token bearer de identificación, explicado en la página 71, con el que realizar las operaciones con la aplicación web. Si no hubiese red disponible o si el reloj no estuviese conectado a ningún teléfono se indicaría de igual modo que no hay conexión a internet con un pantalla con icono y texto descriptivo.

Tras mostrar en la pantalla de sincronización los segundos que dure el intercambio de información, la aplicación dirigirá al usuario a una pantalla en la que encuentra un botón para iniciar una actividad. Cuando el usuario pulse sobre este botón se le mostrará un listado con distintos tipos de actividad, entre las que debe elegir aquella que concuerde más con el tipo de actividad que va a realizar. En nuestro caso, tenemos tres preestablecidas que son viajar en bus, viajar en metro y viajar en tren.

Cuando el usuario haya seleccionado una actividad del listado, se le mostrará una pantalla a modo de aviso en relación a que su actividad ha dado comienzo de forma satisfactoria y tras unos segundos se le dirigirá a la pantalla de menú de medición. En esta pantalla se encuentran la opción asociada al registro de etiquetas en el transcurso de la actividad.

Cuando el usuario desea registrar etiquetas en la actividad, se le hace una categorización básica entre estados y emociones. En esta primera aproximación se ve cómo todos los estados están agrupados en una misma categoría y de igual forma ocurre con las emociones. Como se ha mencionado anteriormente, los estados que aquí se muestran se han extraído de la literatura consultada y expuestos en el capítulo 2 comprobándose previamente también con las tutoras a modo de conocer qué parámetros son los que afectan a los usuarios mayores en cuanto al uso del transporte público. En cuanto a las emociones, estas han sido extraídas de la convención de emociones básicas establecidas recientemente por psicólogos (Cherry, 2021). Estas son felicidad, tristeza, miedo, ira, sorpresa y asco. Por último se le pide al usuario que registre si se siente nervioso o tranquilo.

Tras este registro de etiquetas, la aplicación dirigirá al usuario a la pantalla principal de medición, pudiendo nuevamente registrar etiquetas o terminar la actividad.

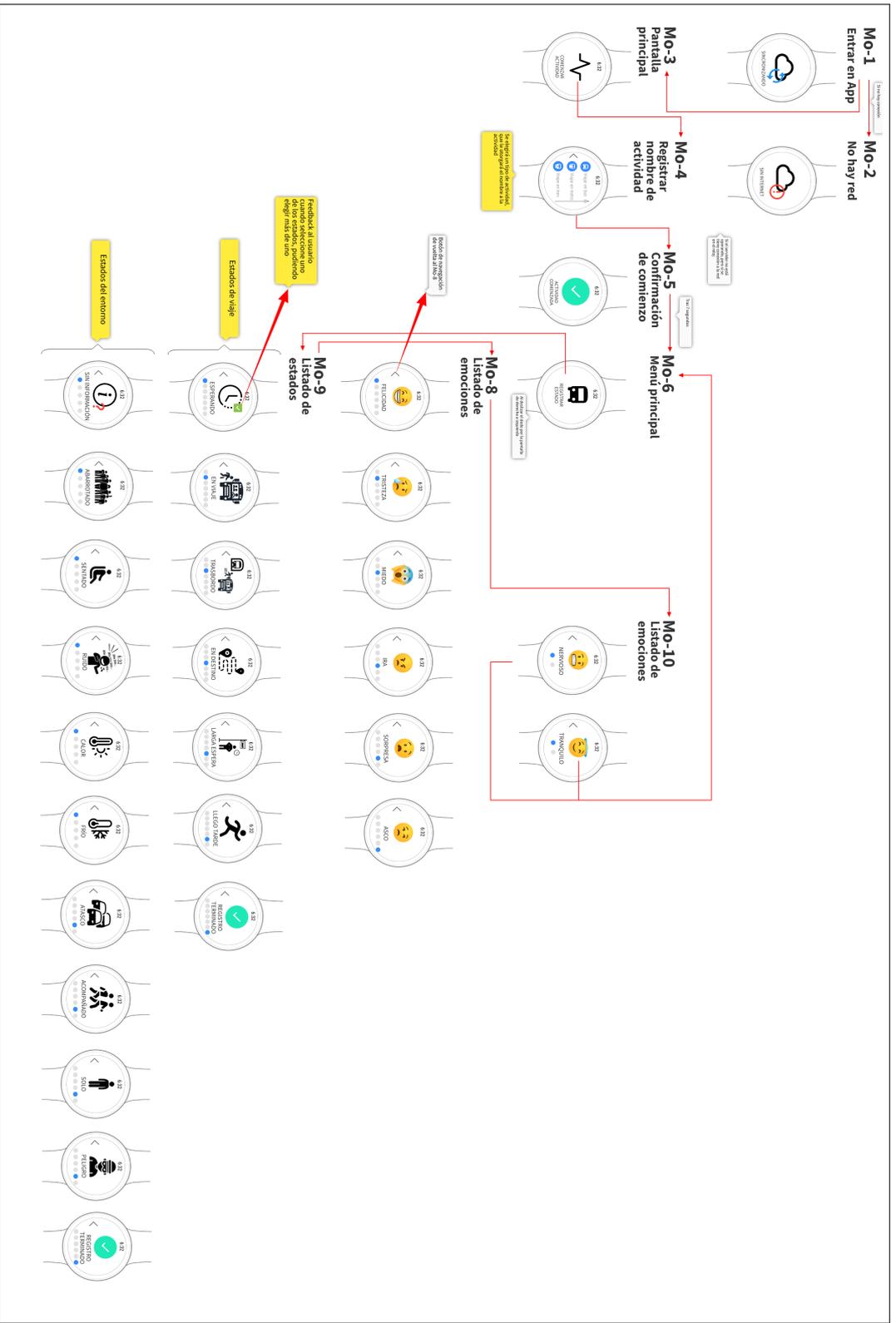


Figura 3.28: Bocetado y navegabilidad en una primera aproximación al desarrollo de la aplicación WearOS en la iteración 2

Tras realizar las tareas asociadas al bocetado, se continúa con la implementación. Comenzaremos comentando las tareas de desarrollo relacionadas con el sistema web. Antes de comenzar, debemos tener en cuenta los modelos que van a existir en nuestro sistema. Un modelo es una clase dentro del paquete de datos que representan los datos internos de los objetos que queramos tener en el sistema, en nuestro caso serían actividades, usuario, etiquetas y medición ppg. Por esto, comenzaremos creando el modelo correspondiente a las actividades. Estas llevan asociadas un nombre, una fecha de inicio y fin, y una asociación al usuario al que pertenece. Se pueden visualizar las tablas resultantes en la figura 3.29.

Como a su vez las actividades deben llevar asociadas a ellas información tanto de sensores fisiológicos recogidos por el reloj inteligente, como de etiquetas marcadas por el usuario, debemos crear también estos dos modelos en nuestra plataforma. Para ello, establecemos que el modelo de las etiquetas estará compuesto por un atributo para almacenar en formato *JSON* las etiquetas seleccionadas por el usuario y otro atributo para almacenar la estampa o marca de tiempo asociada. Como aún no se ha optado por una forma de medir y predecir estrés, los modelos restantes se implementarán en próximas iteraciones siguiendo la misma filosofía que con las etiquetas, ya que son información asociada a la actividad.

De forma análoga a la primera iteración, se muestran en la tabla 3.8 los puntos de entrada añadidos a la API de modo que el usuario pueda iniciar una actividad y registrar las etiquetas asociadas.

Tras esto, se procede a la implementación de los bocetos junto con el desarrollo de su funcionalidad. Se comienza con la sección de actividades que muestra, en primer lugar, a los usuarios mayores que tenemos registrados en el sistema, tal y como se hace en la sección de gestión de usuarios.

En esta pantalla, aportamos también un buscador y un botón al lado de cada usuario para poder visualizar sus actividades. Se puede ver el resultado de la implementación en la figura 3.64a. Cuando se presione sobre el botón, el sistema web nos debe dirigir a la página que se muestra en la figura 3.64b. Aquí se muestran tres columnas, una para cada categoría de actividad. Bajo cada columna, aparecen las diferentes actividades que el usuario ha ido realizando a lo largo del tiempo. Por otro lado, se muestra el botón para visualizar los detalles de una actividad concreta mediante la navegación a la página que se muestra en la figura 3.64c.

TIPO	DIRECCIÓN	DESCRIPCIÓN
GET	/users	Para obtener los datos del usuario identificado. Se necesita aportar un token Bearer.
GET	/users/get_api_key	Para obtener el token Bearer dados un correo y una contraseña registradas en el sistema.
GET	/users/password_recovery	Para solicitar un correo con el recordatorio de contraseña dado un correo electrónico.
GET	/activities	Obtiene la información de todas las actividades para un usuario dado por un token Bearer.
GET	/activities/id	Obtiene la información de una actividad dada por un ID si es perteneciente al usuario dado por su token Bearer.
POST	/activities	Se crea una nueva actividad en el sistema para el usuario dado por el token Bearer. Se debe aportar el nombre de la actividad, que lo hará pertenecer a una categoría, junto con la fecha de inicio. La fecha de fin es opcional, para poder incluirla más tarde.
PUT	/activities/id	Actualiza el registro de una actividad del sistema. Se usa para finalizar una actividad e incluirle la fecha de fin.
POST	/tags	Se crea un nuevo registro de etiquetado para una actividad en transcurso del sistema. Se debe aportar el conjunto de etiquetas seleccionadas en formato JSON y la estampa de tiempo.

Cuadro 3.8: Puntos de entrada de la API en la iteración 2

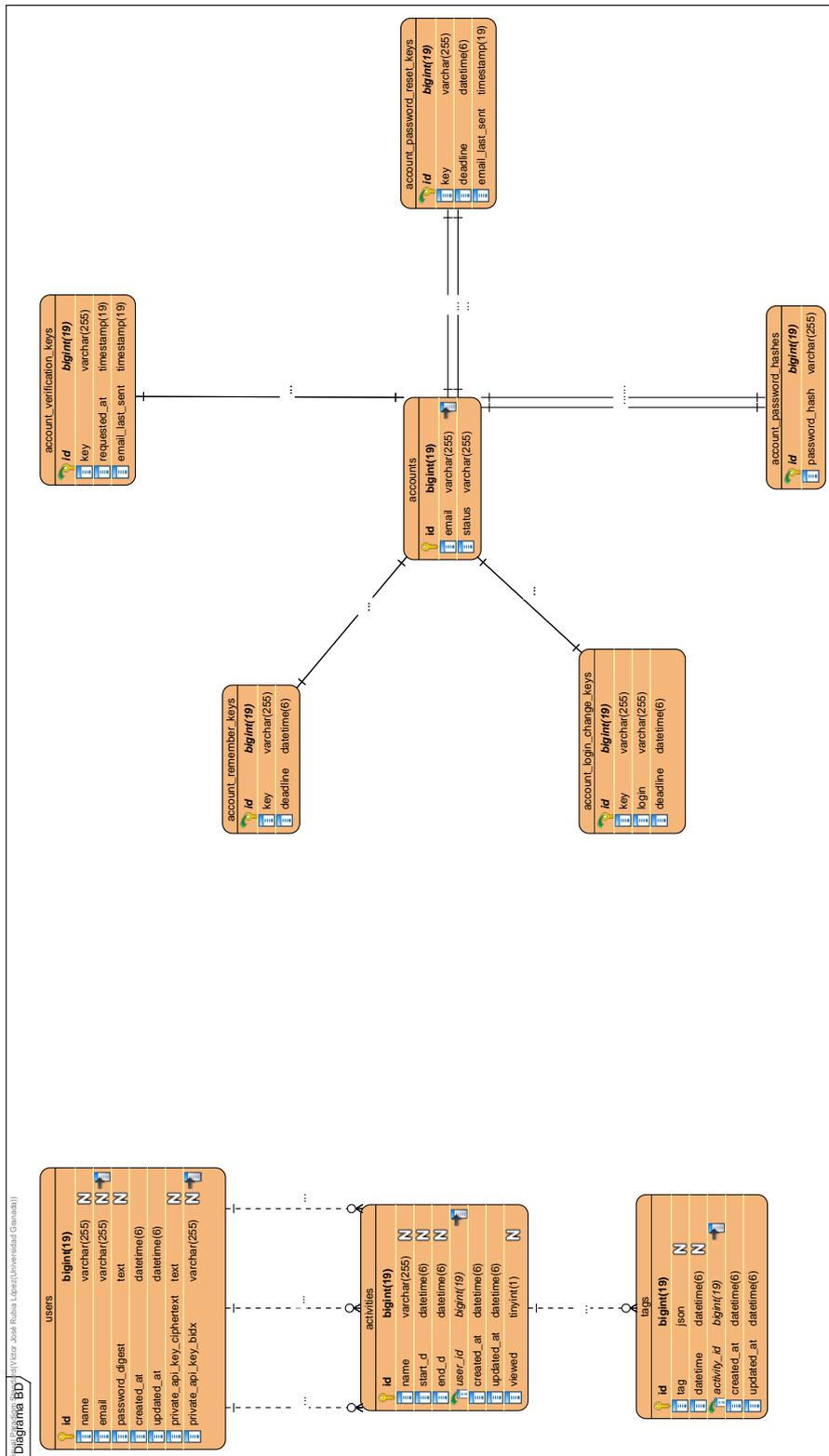
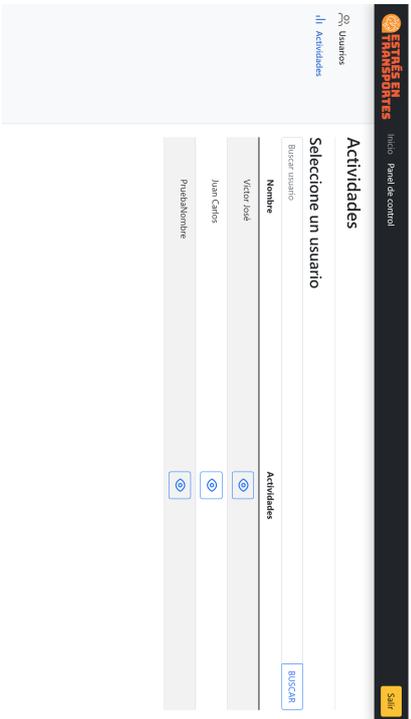
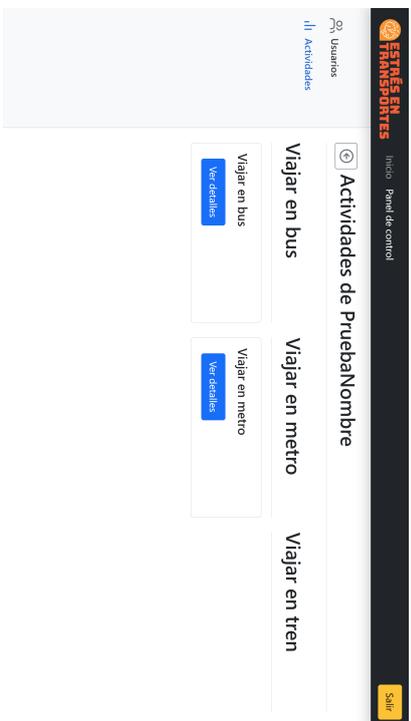


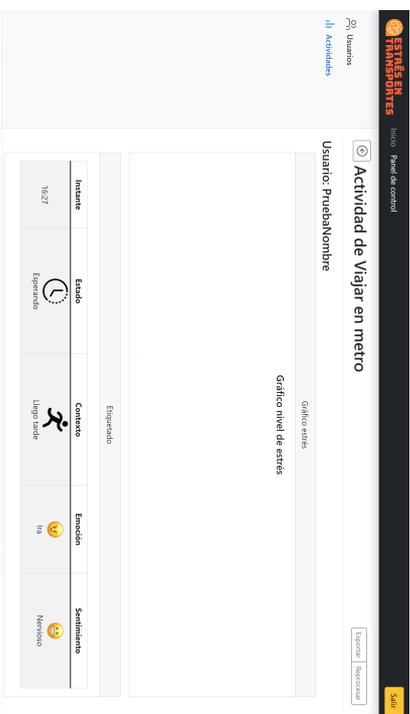
Figura 3.29: Esquema de tablas generadas en la Base de Datos por RoR en la iteración 2



(a) Sección de actividades



(b) Sección de actividades de un usuario



(c) Detalles de una actividad

Figura 3.30: Implementación de los bocetos del sistema web de la iteración 2

En este punto se realizan las pruebas de aceptación asociadas a las tareas de desarrollo de esta segunda iteración. Los resultados se muestran, como en la iteración anterior, en la tabla de la figura 3.31.

TestRun	Total	Passed
All Tests - Generated by RubyMine on 31/3/22 16:12	22	22
Api::V1::ActivityControllerTest::Unauthorized		
test_should_not_get_activities	1542ms	passed
test_should_not_update_another's_activity	758ms	passed
Api::V1::TagControllerTest::Authenticated		
test_should_create_tag	761ms	passed
DashboardControllerTest		
test_should_get_view_activities	707ms	passed
test_required_authentication	296ms	passed
test_should_get_activity_details	1521ms	passed
test_should_get_activities	230ms	passed
test_should_get_index	279ms	passed
test_should_get_users	238ms	passed
Api::V1::UsersControllerTest::Unauthorized		
test_should_not_get_user	157ms	passed
UsersControllerTest		
test_should_create_user	216ms	passed
DashboardTest		
test_searching_an_user	289ms	passed
test_creating_an_user	350ms	passed
WelcomeControllerTest		
test_should_get_index	30ms	passed
AuthenticationTest		
test_creating_an_account	49ms	passed
test_logging_in_and_logging_out	298ms	passed
TagsControllerTest		
test_should_destroy_tag	215ms	passed
test_should_update_tag	189ms	passed
Api::V1::UsersControllerTest::Authenticated		
test_should_get_user	178ms	passed
test_should_get_user's_apiKey	150ms	passed
Api::V1::ActivityControllerTest::Authenticated		
test_should_create_activity	297ms	passed
test_should_handle_404	229ms	passed

Figura 3.31: Tests realizados sobre la Aplicación Web RoR en la iteración 2

Una vez hemos terminado con las tareas de desarrollo relacionadas con el sistema web, pasamos al desarrollo de la aplicación para el reloj inteligente.

Para el desarrollo en el reloj inteligente se pensó en primera instancia usar el framework Flutter, basado en el lenguaje Dart, que de forma análoga al desarrollo en móvil mediante este framework escrito en la iteración anterior permite realizar código fiable en menor tiempo. Para esto se comenzó investigando sobre la forma en la que se puede llevar a cabo.

Existe una librería llamada *wear* (fluttercommunity.dev, 2022b) mantenida por la comunidad de Flutter para llevar el desarrollo mediante este framework a dispositivos con WearOS. Lo único que aporta esta librería es la posibilidad de manejar de forma sencilla el diseño de la interfaz para pantallas tanto redondas como cuadradas y sus tamaños reducidos en comparación a teléfonos.

Tras el comienzo del desarrollo basándome en esta librería, investigué las posibilidades que brinda este framework para acceder a los datos en crudo de los sensores del reloj. Existe una librería desarrollada por los autores del framework llamada *Sensors Plus* (fluttercommunity.dev, 2022a), sin embargo, esta se centra más en sensores físicos como el acelerómetro, el giroscopio y el magnetómetro en lugar de sensores fisiológicos como podrían ser los de PPG o ECG. Investigando acerca de librerías que fuesen capaz de registrar estos sensores y obtener datos de ellos encontré una llamada *ppg* (kuprel, 2019). Sin embargo, esta no es viable para nuestro proyecto ya que se basa sobre otra librería que se desarrolló anterior a *Sensors Plus* que ya está discontinuada y el autor no lo mantiene desde hace 2 años, siendo únicamente compatible con los dispositivos *Fossil Sport* y *Fossil Gen 5*. Realicé una modificación de esta librería para intentar adaptarla al reloj que nos dan en el proyecto, sin embargo no logré hacerla funcionar.

Por esta razón tuve que abandonar el desarrollo en Flutter para la aplicación del reloj. En su lugar, se optó por seguir las recomendaciones de la guía de Android Developers y se empezó de nuevo el desarrollo en Android Nativo usando Kotlin y Jetpack Compose para el diseño de la interfaz. En cuanto a diseño, nos guiaremos por las directrices de Material Design 3.0, que es la versión más reciente publicada por Google, ya que, como en el desarrollo móvil, aporta universalidad y cohesión a nuestro producto, además de que los componentes se encuentran listos para su uso y no es necesaria su implementación.

De igual forma, los bocetos desarrollados se tienen que ver modificados ya que fueron desarrollados siguiendo las directrices de Android Watch, pues la herramienta usada para hacer el bocetado no tenía implementado Material Design 3.0 para WearOS aún. Esto se debe a que en esta última revisión de Material Design se ha cambiado la forma en la que se disponen los elementos en las pantallas y la forma en la que se interactúa con ellas. Por ejemplo, ya no se recomienda implementar listados de desplazamiento horizontal, los cuales bocetamos en cuanto a los menús donde teníamos más de una opción como la selección de etiquetas o el menú principal de medición. Esto se debe a que ahora la interacción para cerrar la aplicación y volver atrás en el reloj se realiza mediante un gesto de desliz de izquierda a derecha por lo que se puede confundir con un gesto para avanzar en un listado horizontal. Es por esto que se opta por cambiarlo a un listado vertical y por un rediseño del menú principal de medición. El resto se ha tratado de mantener como se tenía en el anterior bocetado, salvo que se ha hecho una diferenciación entre emociones y sentimientos, de acuerdo con las aproximaciones más recientes de la psicología, ya que no es lo mismo sentirse tranquilo o nervioso que tener emociones como las listadas en el UI-7 de la figura 3.32. Este diseño se ha hecho mediante un prototipado de alta resolución, ya que las tutoras ya habían corregido y dado su visto bueno a los bocetos anteriores, lo que permite obtener una visión más realista del producto final. En la figura 3.32 se

puede visualizar el resultado teniendo en cuenta que la navegabilidad es la misma que la mostrada en la figura 3.28.

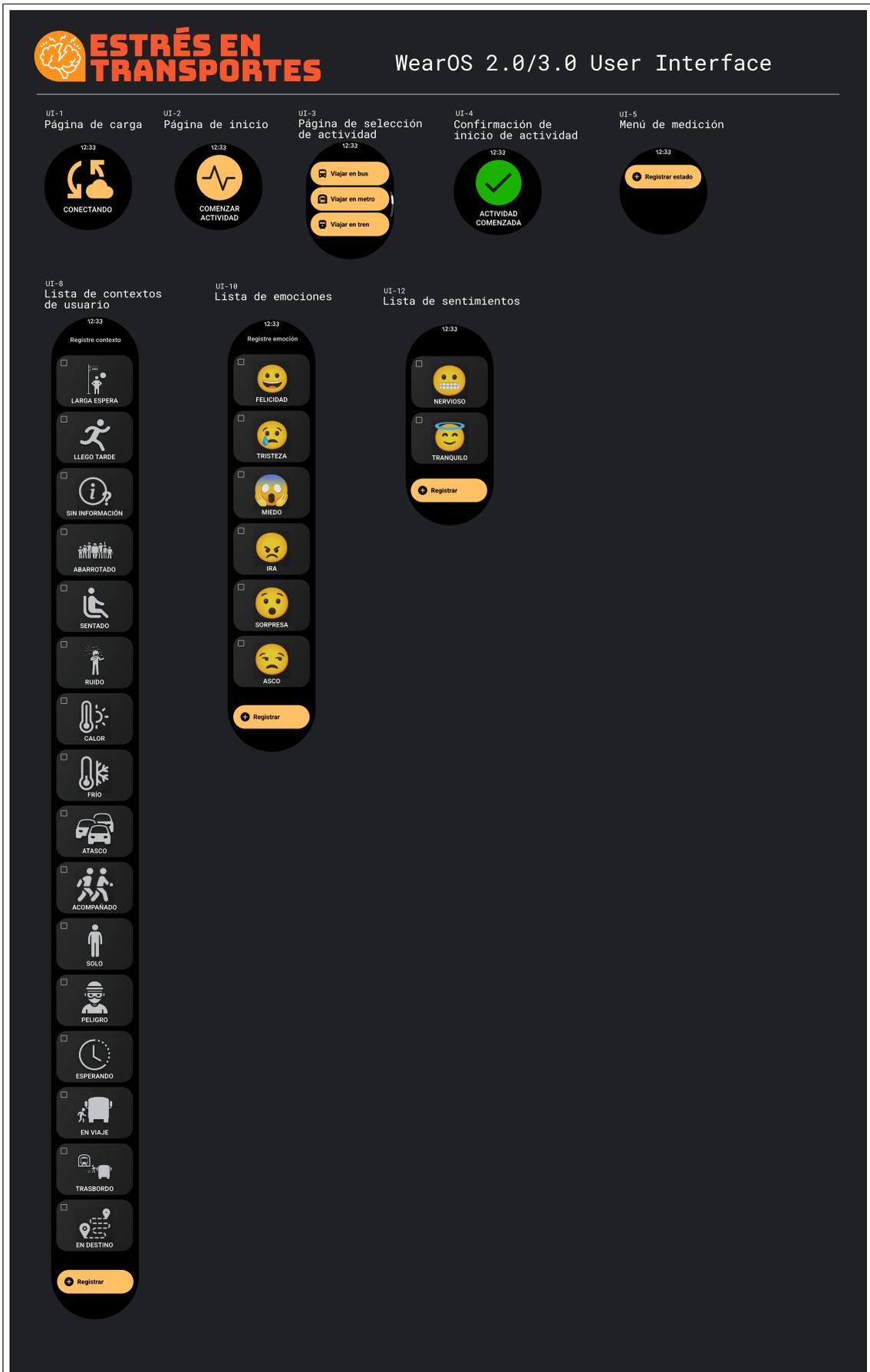


Figura 3.32: Prototipado de alta resolución con Figma de la aplicación WearOS en la iteración 2

El resultado de la implementación es idéntico a lo mostrado en la figura 3.32, por lo que no se incluyen capturas de pantallas asociadas a esto. Para el desarrollo de la aplicación para el reloj inteligente, seguiremos de nuevo un patrón arquitectónico Model-View-ViewModel (MVVM) ya que es el que se recomienda según las guías de desarrollo de Android Developers. Además la mayoría de ejemplos que se encuentran sobre desarrollo para wearOS siguen este patrón, ya que hace más fácil su mantenimiento y pruebas.

De forma análoga a la aplicación móvil, se usa la librería *Retrofit 2.0* para realizar las peticiones a la API del sistema web. En la figura 3.4 se muestran las funciones de la interfaz de conexión a la API de la aplicación del reloj inteligente.

```
1 /**
2  * Creates an activity.
3  *
4  * @return The list of activities.
5  */
6 @POST("activities/")
7 suspend fun newActivity(@Header("Authorization") apiKey : String ,
8                        @Query("activity[name]") name : String ,
9                        @Query("activity[start_d]") startTimestamp : String ) :
10                        ↳ Response<Activity>
11 /**
12  * Adds tag to an activity
13  **
14  * @param apiKey
15  * @param tag
16  * @param dateTime
17  * @param activityID
18  * @return The added tag
19  */
20 @FormUrlEncoded
21 @POST("tags/")
22 suspend fun addTag(@Header("Authorization") apiKey : String ,
23                  @Field("tag[tag]") tag: String ,
24                  @Field("tag[datetime]") dateTime: Date ,
25                  @Field("tag[activity_id]") activityID : Int ,
26 ) : Response<Tag>
```

Listing 3.4: Funciones pertenecientes a la app del reloj inteligente de la interfaz de conexión con la API

La primera función es la que nos sirve para crear un registro de actividad nuevo en la plataforma, de modo que le tenemos que proporcionar el token Berarer del usuario identificado en el dispositivo móvil junto con el nombre de la actividad iniciada, que lo tomará del tipo de actividad que el usuario haya escogido y la fecha de inicio, que corresponderá al momento en el que el usuario decide iniciarla.

La segunda función corresponde a añadir etiquetas a una actividad. Para ello, se debe proporcionar el token junto a un JSON con el conjunto de nombres de las etiquetas, el instante de tiempo en el que se ha llevado a cabo el registro y el id de actividad al que pertenecen las etiquetas. Se ha optado por almacenar las etiquetas en formato JSON ya que es bastante usado hoy en día en cuanto a almacenamiento de datos grandes, es fácil de manejar en las aplicaciones y está soportado por Ruby On Rails.

De esto resulta necesaria la implementación de la funcionalidad para que exista una comunicación entre el reloj y el dispositivo móvil de modo que cuando el usuario esté identificado en la aplicación del móvil el reloj sea capaz de consultar cuál es el token bearer, explicado en la página 71, asociado al usuario identificado en el dispositivo móvil.

Comenzaremos por añadir una funcionalidad a la aplicación del móvil que se ocupe de enviar el token al reloj inteligente. Este caso de uso se ejecutará cuando el usuario se identifique en la aplicación, de modo que si la aplicación del reloj está ejecutándose y ya había identificado previamente a otro usuario, pueda recibir el nuevo token y limpiar su sesión. Estos pasos se pueden consultar de forma visual en la figura 3.33.

Tras esto, se configura un servicio de escucha basado en MessageClient de la API de Google Play (Developers, 2022). Este se estará ejecutando en segundo plano de forma permanente, incluso si la aplicación se encuentra cerrada. Esto soluciona la problemática de que el usuario tenga que tener la aplicación abierta cuando hiciese uso de la aplicación en su reloj para que pudiesen intercambiar datos.

El funcionamiento de comunicación mediante esta API se basa en el uso de “capacidades” que se registran en dispositivos que quieren comunicarse. A los dispositivos que están conectados se les llama nodos. Aquellos nodos que deseen transmitir un mensaje a otro nodo, deben primero preguntar a todos los nodos que estén conectados a él si poseen la capacidad que están buscando. Aquellos que sí la tengan serán nodos potenciales y se obtendrá su ID.

Una vez obtenido el ID del mejor nodo, se le mandará un mensaje bajo esa “capacidad” en el que se indique, mediante un vector de bytes, el contenido del mensaje. Será de esta lógica de la que nos valdremos para que el reloj pueda conocer el token del usuario que esté identificado en la aplicación móvil.

En la figura 3.5 se muestra la implementación de la solicitud que hace el reloj inteligente al dispositivo móvil, ejecutándose las funciones en el orden en el que se muestran. Estas tres funciones son las mínimas necesarias para establecer un paso de mensajes entre dispositivo Android y WearOS. La función *updateBestNode*, es la encargada de obtener aquellos dispositivos WearOS que tengan la “capacidad” requerida por nuestra aplicación, obteniendo el mejor nodo como el más cercano en señal al dispositivo Android. Por otro lado, tenemos la función *initMessageBroadcaster*, encargada de iniciar la función anterior en el contexto de la aplicación y añadiendo un servicio de escucha para saber cuándo se ha terminado de obtener las “capacidades” de todos los dispositivos vinculados. Por último tenemos la función *broadcastMessage*, encargada de transmitir una secuencia de bytes al nodo correspondiente al dispositivo WearOS obtenido por las funciones anteriores.

```
1 /**
2  * Updates the best node ID to send the message to.
3  *
4  * @param context The application context.
5  */
6 private fun updateBestNode(context: Context) {
7     try {
```

```

8      val info = Tasks.await(
9          Wearable.getCapabilityClient(context)
10             .getCapability("api_key_sender", CapabilityClient.FILTER_REACHABLE
11                 ↪ )
12         )
13         for (node in info.nodes) {
14             if (node.isNearby) {
15                 bestNodeID = node.id
16             }
17         } catch (e: Exception) {
18             Log.e("Exception!", e.toString())
19         }
20     }
21
22 /**
23  * Initializes the message broadcaster.
24  *
25  * @param context The application context.
26  */
27 private fun initMessageBroadcaster(context: Context?) {
28
29     if (context != null) {
30         Wearable.getCapabilityClient(context).addListener({ updateBestNode(context
31             ↪ ) }, "apde_receive_logs")
32     }
33
34     // Can't do this on the main thread
35     CoroutineScope(Dispatchers.IO).launch {
36         updateBestNode(context!!)
37     }
38 }
39
40 /**
41  * Broadcasts the api key request to the best node.
42  *
43  * @param context The application context.
44  */
45 private fun broadcastMessage(context: Context?) {
46     Handler(Looper.getMainLooper()).post {
47         try {
48             if (context != null) {
49                 bestNodeID?.let { Wearable.getMessageClient(context).sendMessage(
50                     ↪ it, "api_key_sender", ByteArray(0)) }
51             } catch (e: java.lang.Exception) {
52                 Log.e("Exception!", e.toString())
53             }
54         }
55     }

```

Listing 3.5: Funciones para solicitar el token del usuario al dispositivo móvil

Por otro lado, en la figura 3.6 se encuentra la implementación de un servicio de escucha background para la recepción de mensajes en el dispositivo móvil que sigue la lógica de comunicación que se ha descrito en la explicación de comunicación entre dispositivos. Se aprecia una función *onMessageReceived*, la cual se ejecuta cada vez que se recibe un mensaje por parte de un dispositivo WearOS vinculado a un teléfono Android.

Esta función filtra los mensajes para obtener aquellos que nos interesen, mediante la “capacidad” que hemos preestablecido. Tras esto, se obtiene el ID del nodo que ha enviado dicho mensaje y se envía a este nodo un mensaje con la clave API que obtenemos de la base de datos local Room de la aplicación, en la cual se encuentra el usuario que está identificado en la plataforma. Ha sido bastante complejo realizar este servicio ya que no existe documentación detallada acerca de cómo hacer este procedimiento, por ello se adjunta en la figura 3.6.

```

1  override fun onMessageReceived(messageEvent: MessageEvent) {
2      if (messageEvent.path == API_KEY_CAPABILITY_NAME) {
3          Log.d("Servicio", "Received API Request");
4          val receiverID = messageEvent.sourceNodeId
5
6          var responseLiveData = liveData {
7              val user = factory.execute("", "")
8              emit(user)
9          }
10
11         val observer = Observer<User?> {
12             if (it != null) {
13                 Log.i("Servicio", "Envio $it")
14                 Handler(Looper.getMainLooper()).post {
15                     try {
16                         Wearable.getMessageClient(applicationContext)
17                             .sendMessage(receiverID, "api_key", it.apiKey.
18                                 ↪ toByteArray())
19                     } catch (e: java.lang.Exception) {
20                         Log.e("Servicio", "Mensaje no se ha podido mandar. Razon:
21                             ↪ $e")
22                     }
23                 }
24             } else {
25                 Handler(Looper.getMainLooper()).post {
26                     try {
27                         Wearable.getMessageClient(applicationContext)
28                             .sendMessage(receiverID, "error", "errorNoUserLoggedIn
29                                 ↪ ".toByteArray())
30                     } catch (e: java.lang.Exception) {
31                         Log.e("Servicio", "Mensaje no se ha podido mandar. Razon:
32                             ↪ $e")
33                     }
34                 }
35                 Log.e("Servicio", "ERROR Servicio")
36             }
37         }
38     }
39     Handler(Looper.getMainLooper()).post { responseLiveData.observeForever(
40         ↪ observer) }
41 }

```

Listing 3.6: Función que se ejecuta en la recepción de mensajes de WearOS en el dispositivo móvil

Teniendo esto implementado, conseguimos aportar la funcionalidad de interconexión entre el dispositivo móvil y reloj inteligente. Para ilustrar la lógica de interconexión entre el dispositivo móvil, el reloj inteligente y el sistema web, se ha elaborado un diagrama

de comunicación que representa la funcionalidad realizada hasta esta iteración.

En la figura 3.33 se ilustra el flujo de actividad normal que realizaría un usuario mayor a la hora de usar la plataforma para medir sus actividades. Se comienza abriendo por primera vez la aplicación Android e iniciando sesión con el correo y la contraseña. Una vez el usuario confirme su inicio, la aplicación realizará una petición a la API para obtener la información del usuario mediante las credenciales de identificación proporcionadas. En el caso en el que las credenciales sean erróneas, se le notifica al usuario mediante un icono de error en pantalla. En otro caso, el teléfono recibe el token Bearer del usuario a identificar.

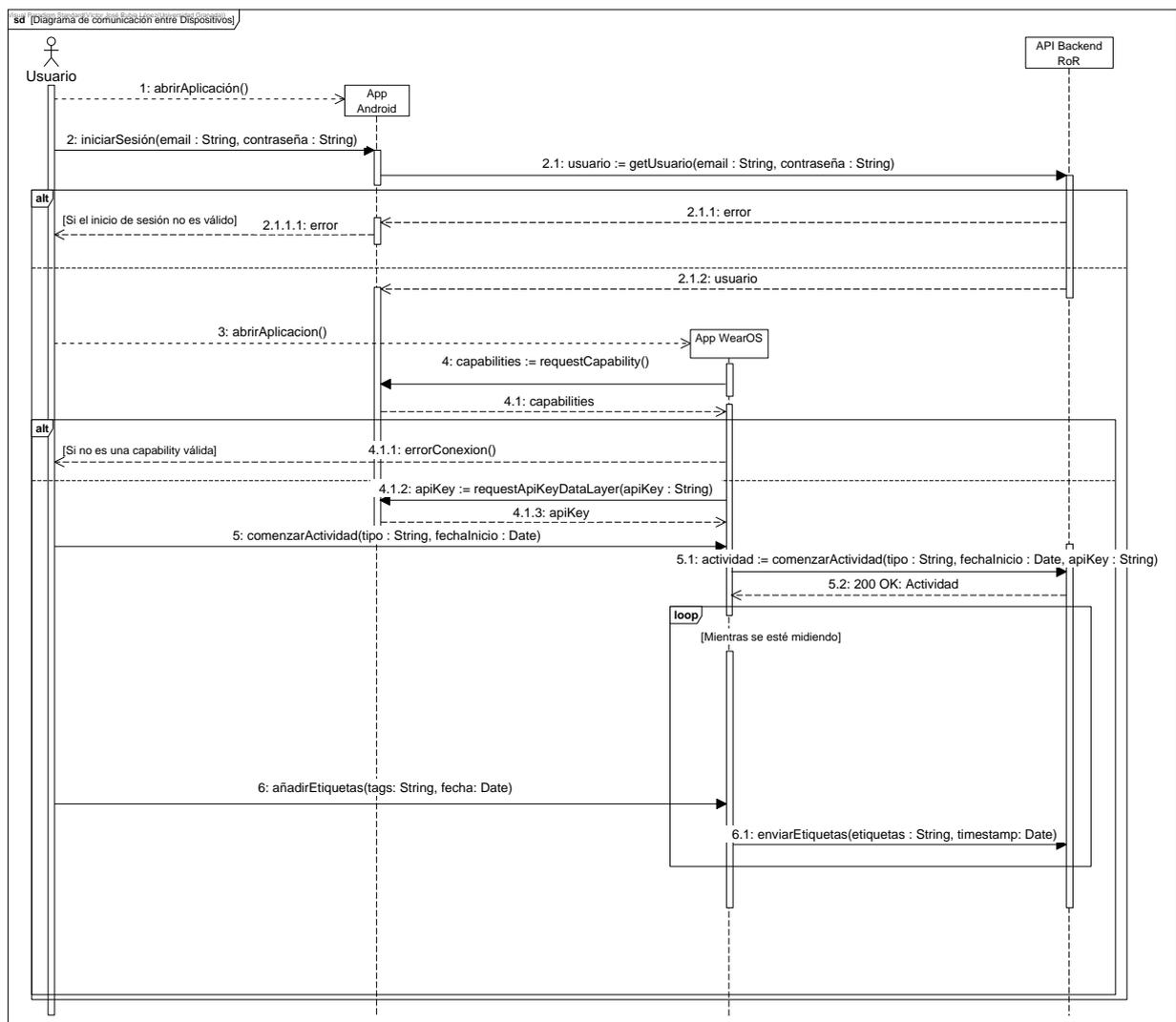


Figura 3.33: Diagrama de comunicación entre sistemas en la iteración 2

Quando el usuario mayor abre la aplicación del reloj, esta solicita a los nodos conec-

tados, en nuestro caso al teléfono vinculado, sus “capacidades”. Si el reloj inteligente no recibe ninguna “capacidad” válida para su propósito (identificar al usuario mayor), se muestra un error de conexión. En caso contrario, bajo la capacidad de identificación, solicita el reloj al móvil el token Bearer para poder hacer solicitudes a la API. El teléfono, por su parte, contesta con el token del usuario que se encuentra identificado en la aplicación.

En este momento, el usuario mayor puede comenzar una actividad, lo que hace que el reloj solicite a la API el registro de una nueva actividad. Mientras la actividad esté en transcurso y el usuario no decida terminarla, el usuario puede introducir etiquetas que serán enviadas al servidor.

Tras la exposición del diagrama de comunicación en cuanto a la lógica de interconexión entre los sistemas, se expone el diagrama de paquetes obtenido de la implementación de las tareas de desarrollo de la iteración 2 en la figura 3.34.

Se aprecia el uso del patrón arquitectónico MVVM, ya que todos los paquetes se encuentran englobados dentro de tres paquetes principales. El paquete *data*, correspondiente a la capa de datos, el paquete *presentation*, correspondiente a la capa de presentación y el paquete *domain*, correspondiente al dominio. A continuación, se detalla el contenido de cada capa.

- **Paquete *data*.** Este paquete alberga la interfaz de conexión con la API del sistema web, junto con la interfaz de almacenamiento de datos local, tanto en caché como en almacenamiento persistente. Por otro lado, contiene también los distintos modelos que conforman la aplicación del reloj, como por ejemplo el de actividad, etiquetado o usuario. Por último, contiene los repositorios donde se encuentran las interfaces e implementaciones para la obtención de datos para los modelos del sistema.
- **Paquete *presentation*.** Este paquete contiene a su vez un gran paquete para la inyección de dependencias junto con los casos de uso, y un paquete por cada vista que se tenga en la aplicación. Cada paquete de vista puede almacenar un ViewModel que contenga la ejecución de los casos de uso y una *Activity* que forme la correcta visualización y funcionamiento de la pantalla.
- **Paquete *domain*.** Este paquete contiene la definición de los casos de uso. La lógica de estos se lleva a cabo en los repositorios del paquete *data*.

De forma análoga, se muestran los diagramas de clases correspondientes a los paquetes de *data* y *domain* mostrados en el diagrama de paquetes en las figuras 3.35, 3.36, 3.37 y 3.38. Debido al inmenso tamaño del diagrama de clases obtenido para el paquete de *presentation*, se ha tenido que dividir en paquetes más pequeños para su correcta visualización.

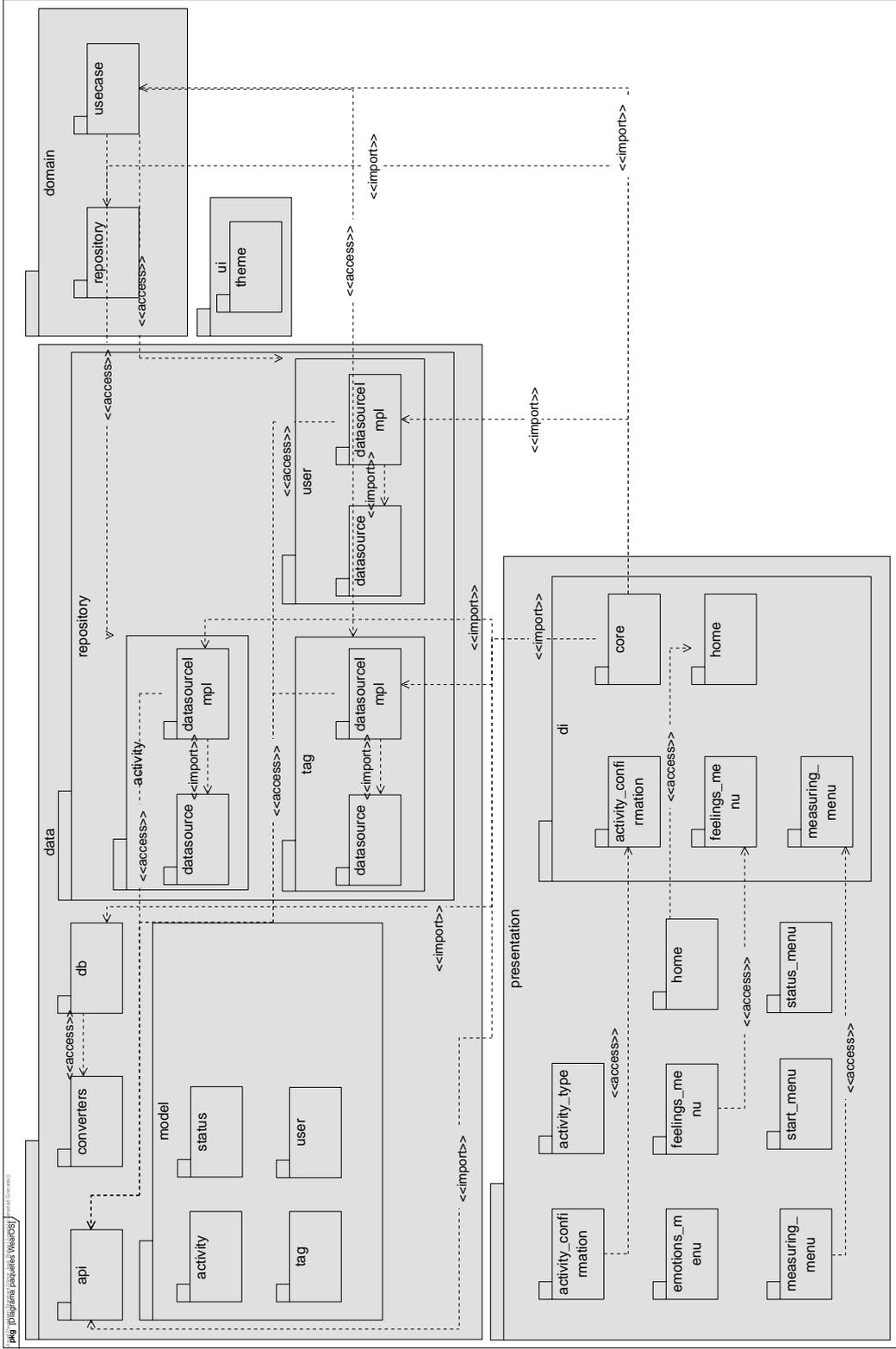


Figura 3.34: Diagrama de paquetes de la aplicación wearOS en la iteración 2

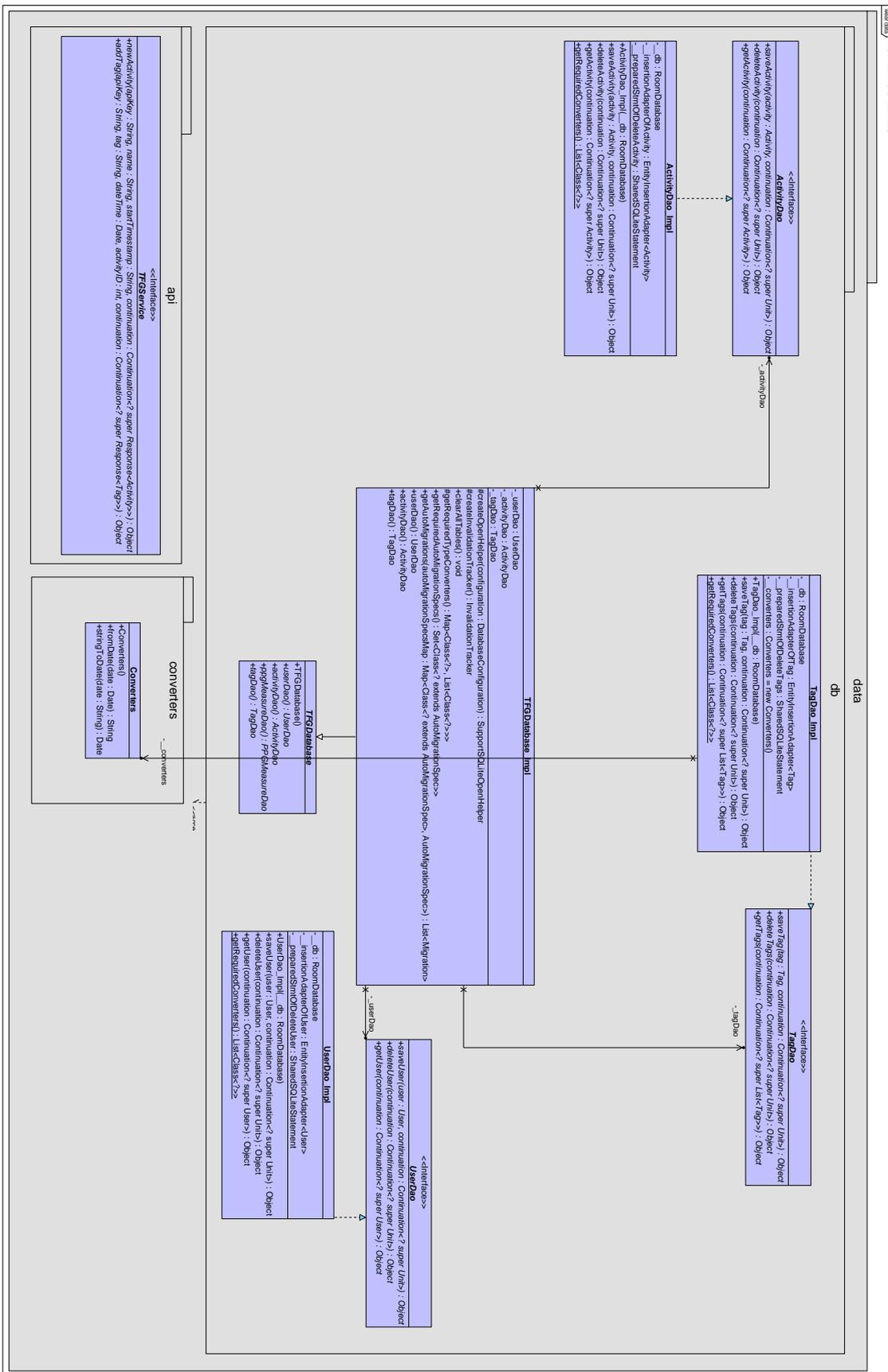


Figura 3.35: Diagrama de clases correspondiente al paquete data de la aplicación wearOS en la iteración 2 (Parte 1)

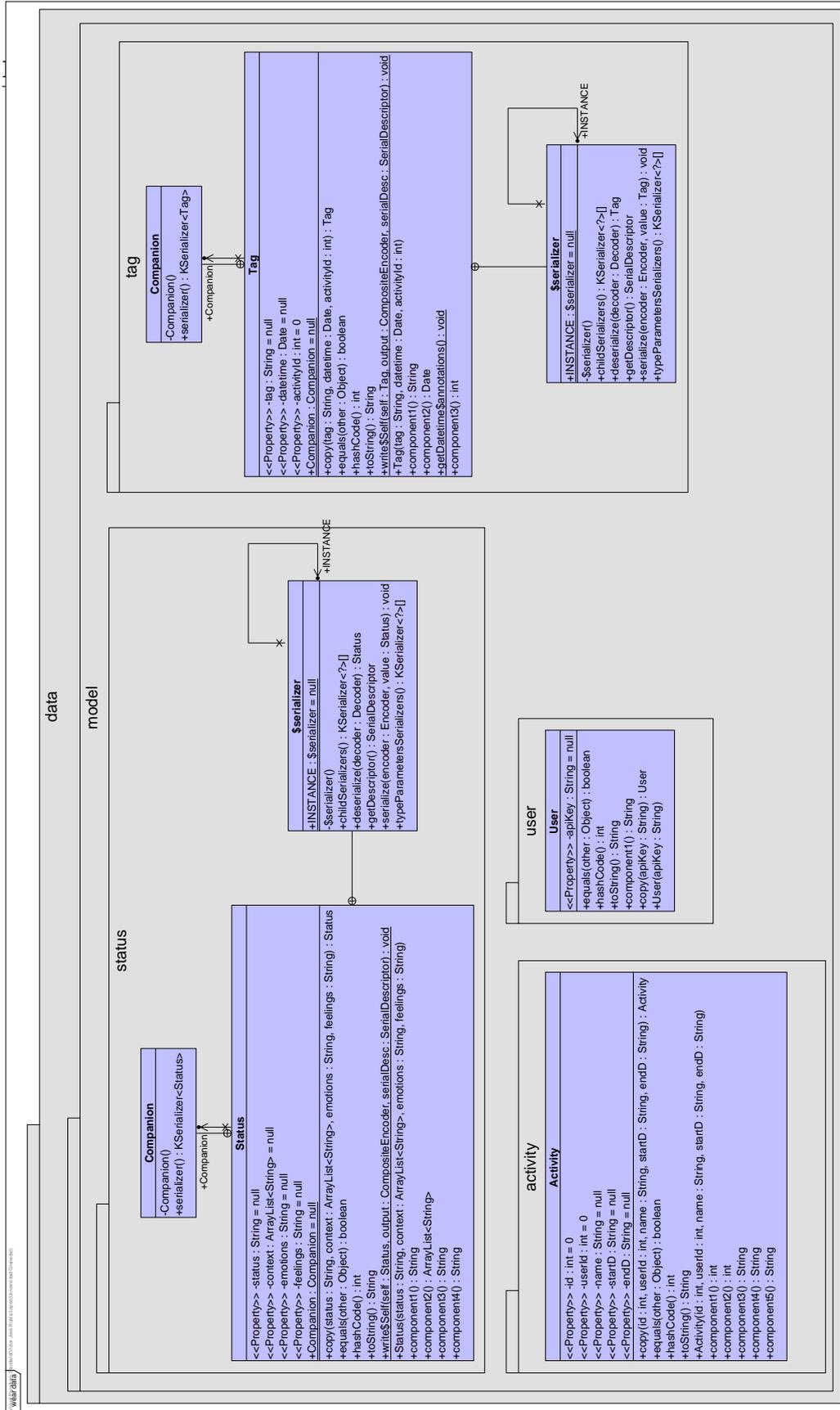


Figura 3.36: Diagrama de clases correspondiente al paquete *data* de la aplicación wearOS en la iteración 2 (Parte 2)

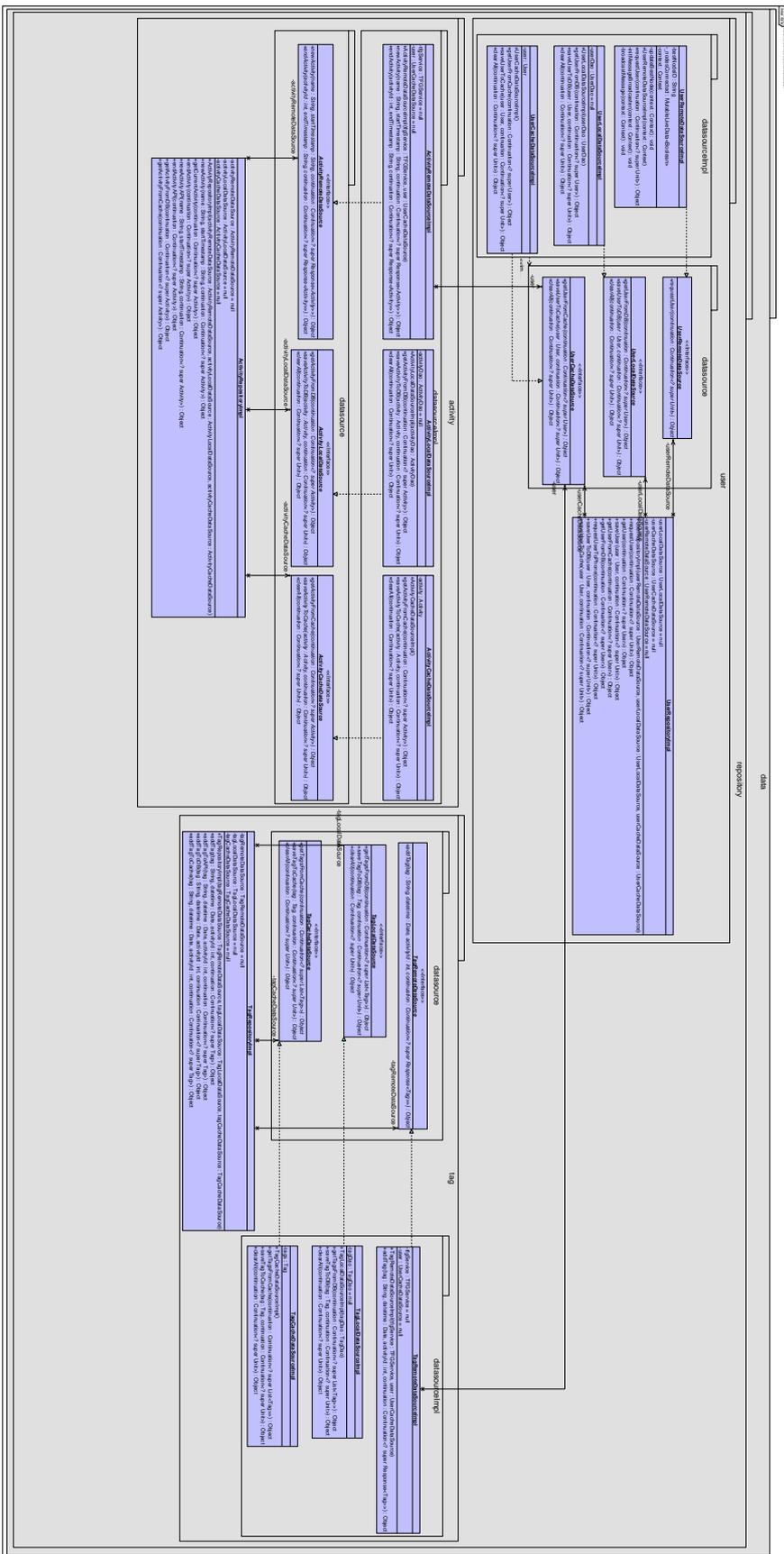


Figura 3.37: Diagrama de clases correspondiente al paquete *data* de la aplicación wearOS en la iteración 2 (Parte 3)

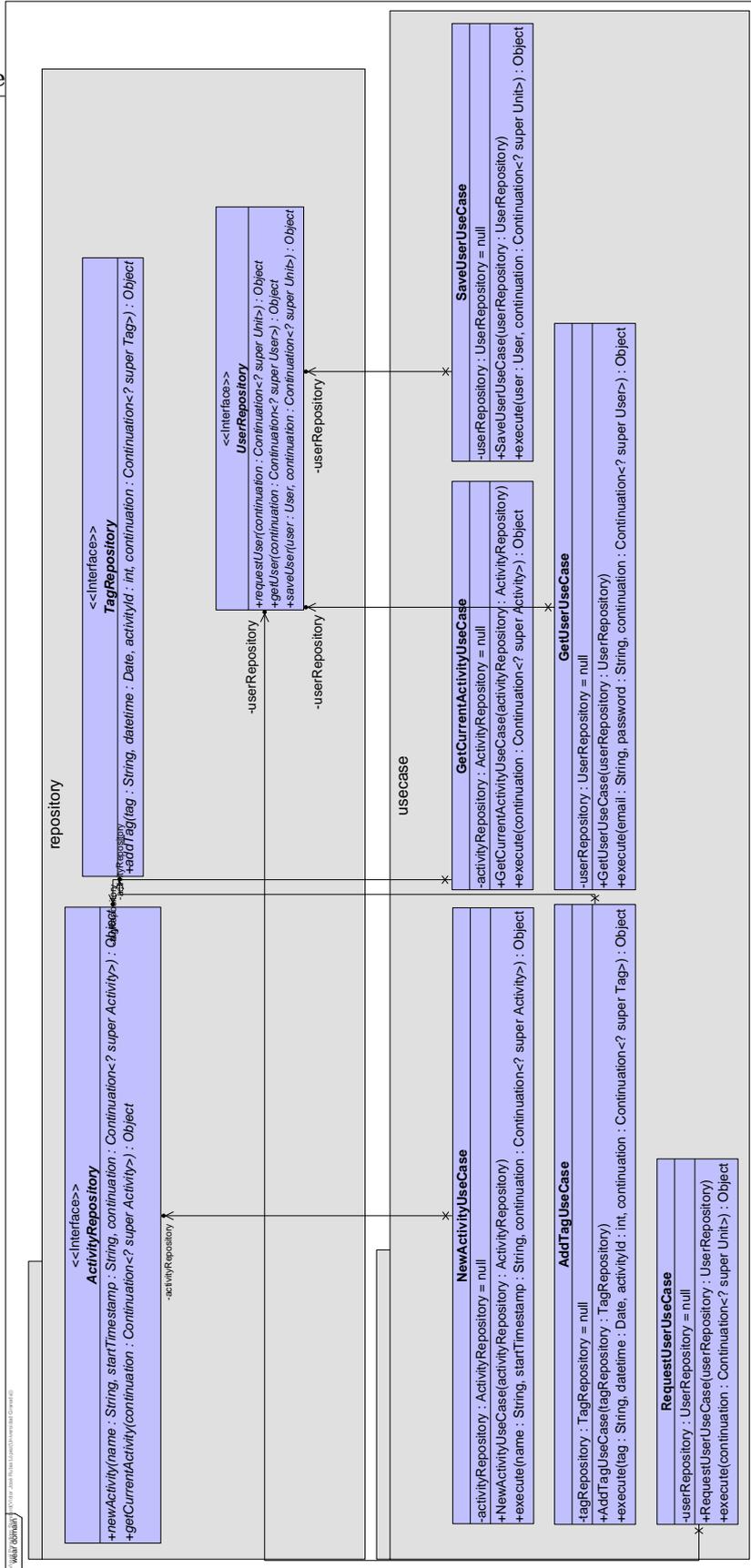


Figura 3.38: Diagrama de clases correspondiente al paquete *domain* de la aplicación wearOS en la iteración 2

3.4. Segunda iteración

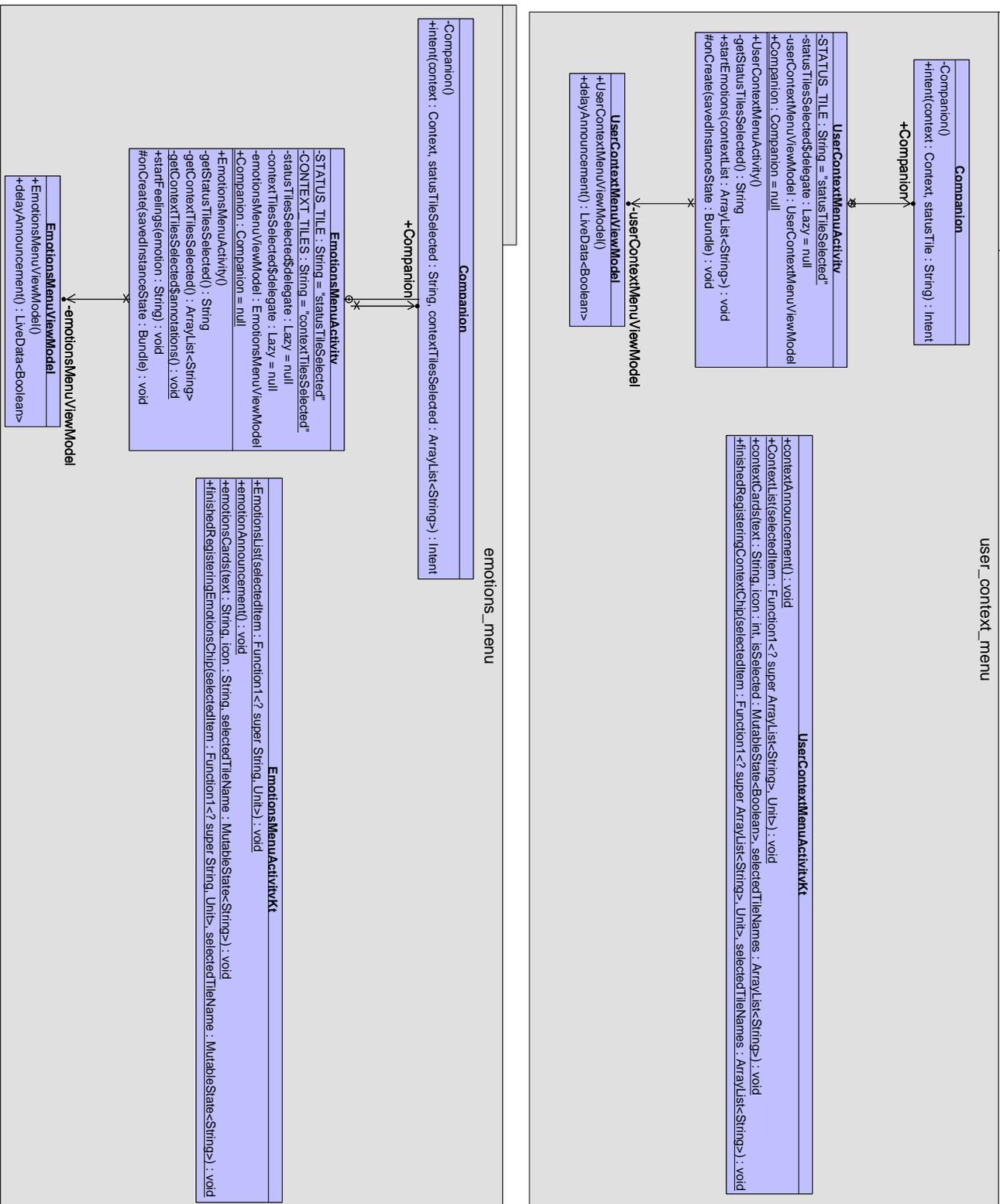


Figura 3.39: Diagrama de clases correspondiente al paquete *presentation* de la aplicación wearOS en la iteración 2 (Parte 1)

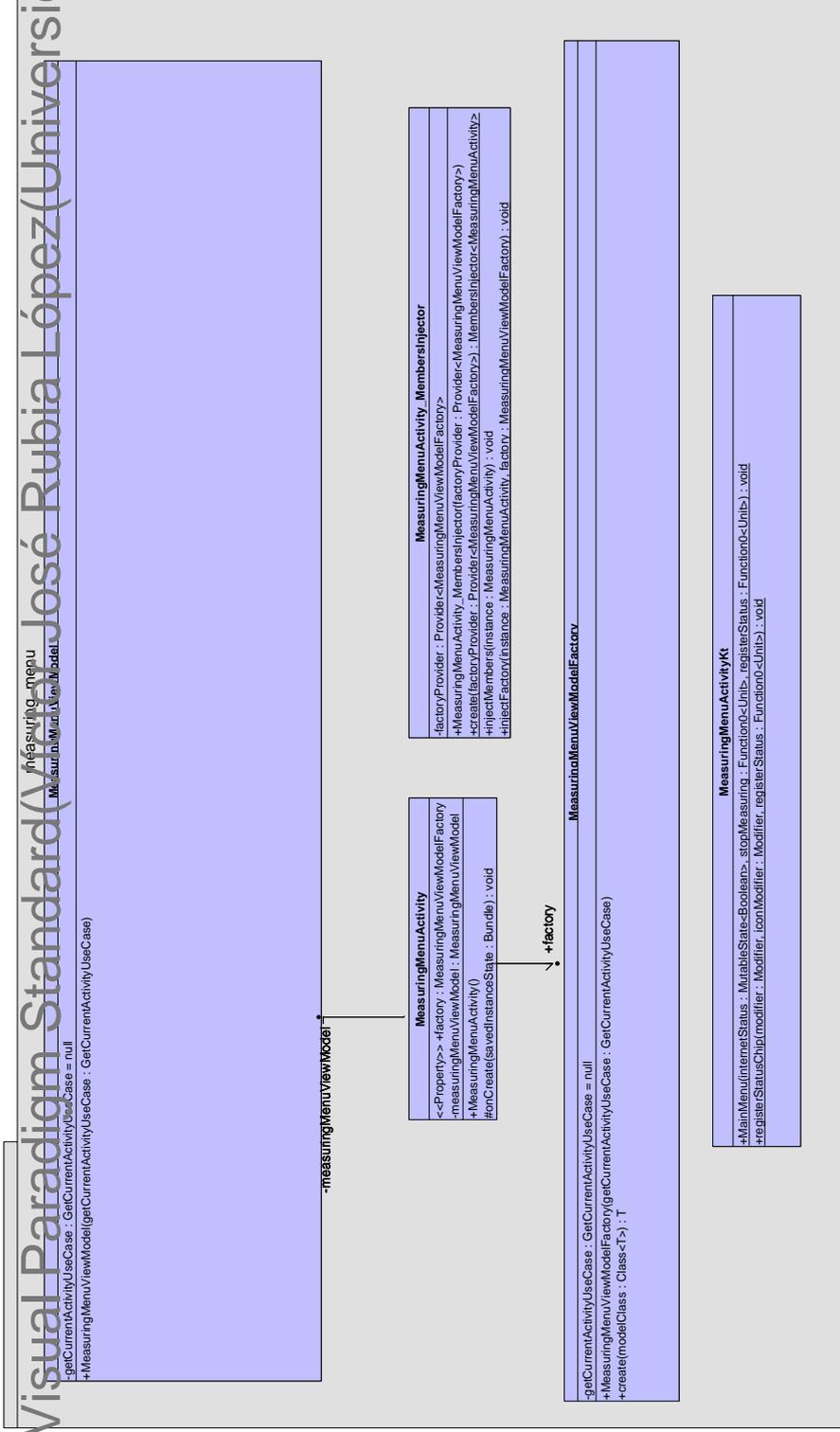


Figura 3.40: Diagrama de clases correspondiente al paquete *presentation* de la aplicación wearOS en la iteración 2 (Parte 2)

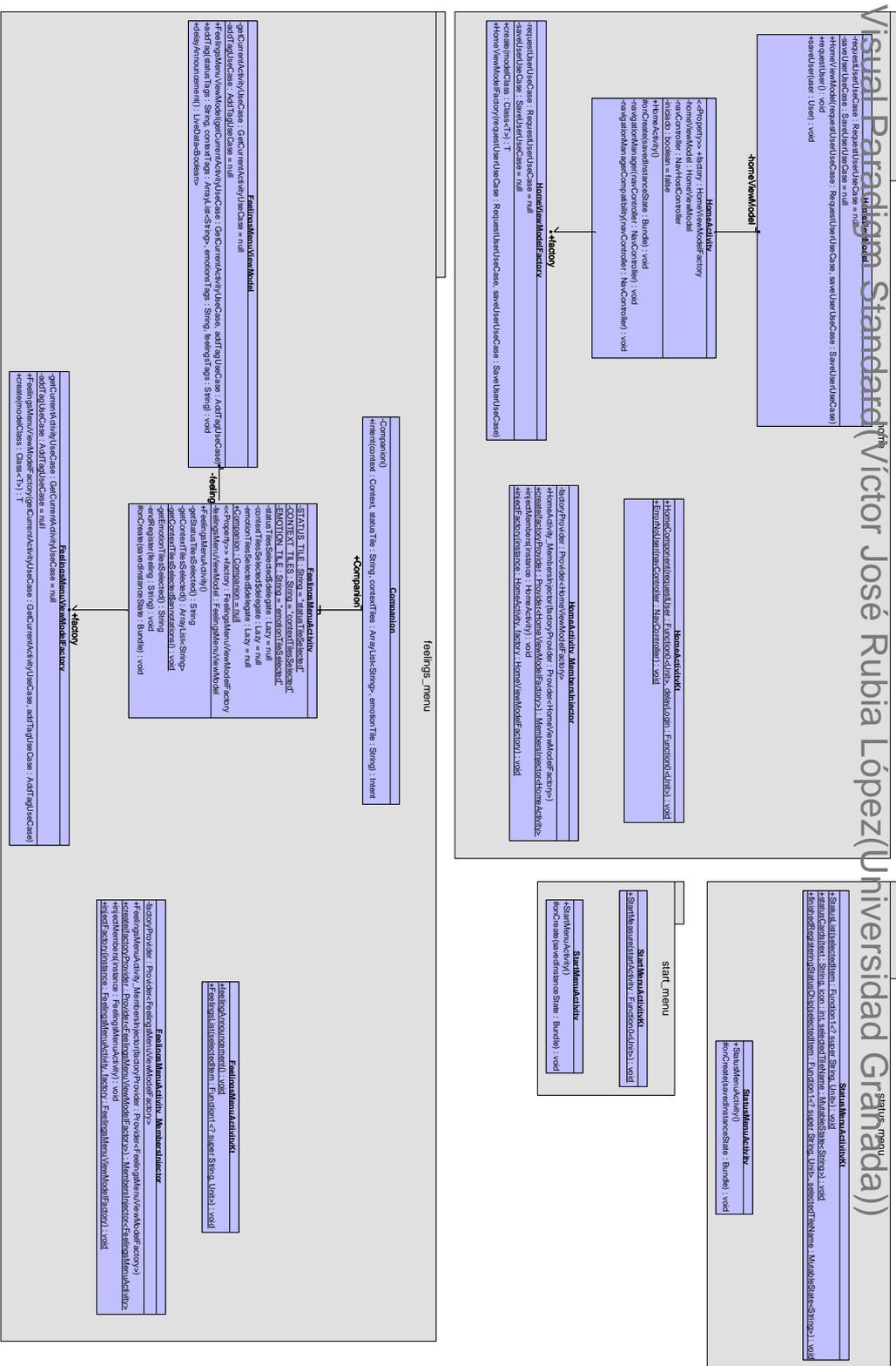


Figura 3.41: Diagrama de clases correspondiente al paquete *presentation* de la aplicación wearOS en la iteración 2 (Parte 3)

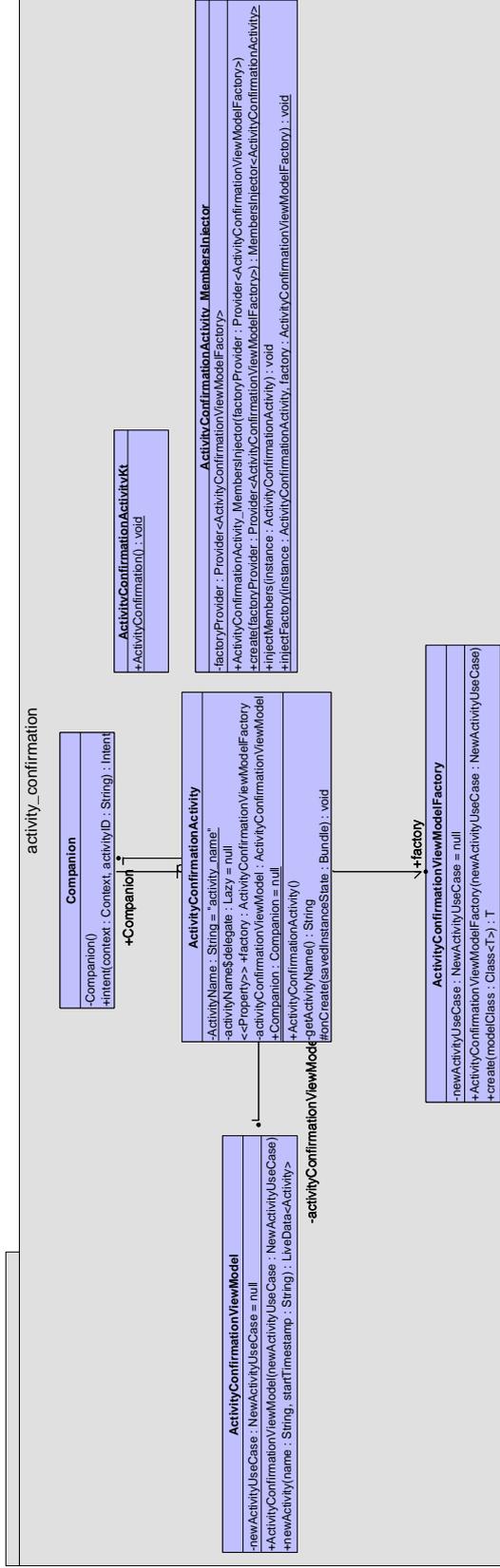
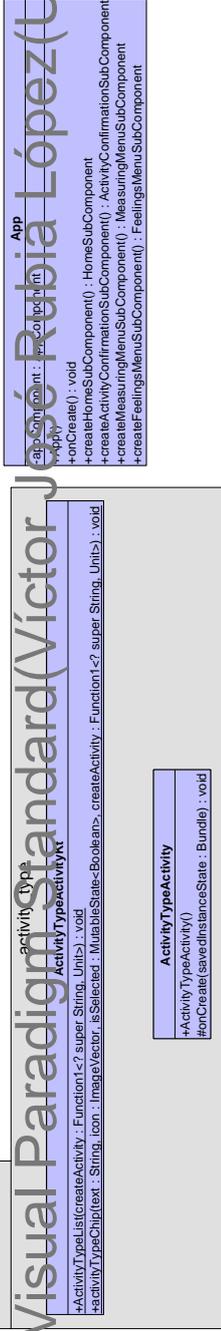


Figura 3.42: Diagrama de clases correspondiente al paquete *presentation* de la aplicación wearOS en la iteración 2 (Parte 4)

En cuanto a las pruebas de aceptación, se muestran en la tabla de la figura 3.43 las realizadas en esta iteración. Aparecen únicamente tests unitarios, ya que para las pruebas de integración no existe aún un framework estable para probar las funcionalidades de forma automatizada, por lo que este tipo de pruebas se ha llevado a cabo de forma manual y probando su corrección hasta en tres dispositivos distintos, el proporcionado por el proyecto, una máquina virtual y un Samsung Galaxy Watch 4. Además en la tabla 3.9, se incluye, de una forma más detallada, la evaluación de las pruebas de aceptación.

TestRun	Total	Passed
Unit Tests Wear - Generated by IntelliJ IDEA on 1/4/22 11:31	3	3
TFGServiceTest		
TFGServiceTest.addTag_sentRequest_receivedExpected	1157ms	passed
TFGServiceTest.newActivity_sentRequest_receivedExpected	13ms	passed
ActivityConfirmationViewModelTest		
ActivityConfirmationViewModelTest.newActivity_returnCurrentRecord	6124ms	passed

Figura 3.43: Tests Unitarios y de Componentes de la Aplicación wearOS en la iteración 2

ID	PRUEBA DE ACEPTACIÓN	ESTADO	OBSERVACIONES
27	Se debe obtener el ID de la actividad iniciada.	Correcto	Comprobado mediante test unitario <i>test_should_create_activity</i> .
28	Únicamente se iniciará una actividad cuando el usuario mayor ha iniciado sesión en el teléfono móvil asociado al reloj.	Correcto	Comprobado de forma manual, si no está identificado se muestra un mensaje para que se identifique.
29	El terapeuta debe estar identificado para visualizar los niveles de estrés de una actividad concreta.	Correcto	Comprobado mediante test unitario <i>test_required_authentication</i> .
30	El gráfico muestra si existe alguna medición para dicha actividad.	Correcto	Comprobado mediante test unitario <i>test_should_get_activity_details</i> .
31	El gráfico debe enlazar los puntos mediante una línea suave, incluso si no existe medición en tiempos correlativos.	Correcto	Comprobado de forma manual, se unen los puntos aunque no sean correlativos, sin dejar un espacio en blanco.
32	El terapeuta debe estar identificado para visualizar las actividades categorizadas.	Correcto	Comprobado mediante test unitario <i>test_required_authentication</i> .

33	Cada tipo de actividad debe mostrarse en la columna asociada a su categoría.	Correcto	Comprobado de forma manual.
34	El nombre de la actividad debe coincidir con la categoría que se muestra.	Correcto	Comprobado de forma manual.
35	Todas las actividades que se registren deben tener un nombre igual a uno de estos tres tipos de actividad.	Correcto	Comprobado de forma manual en el código.
36	No puede existir una actividad que se registre en el sistema con un nombre distinto a las tres categorías	Correcto	Comprobado de forma manual en el código.
37	El nombre de la actividad iniciada debe coincidir con el tipo de actividad seleccionado en la lista.	Correcto	Comprobado de forma manual en el código.
38	Solo se debe poder seleccionar una única categoría.	Correcto	Comprobado de forma manual en el código y en ejecución.
39	Se deben poder seleccionar una o varias etiquetas del listado.	Correcto	Comprobado de forma manual en el código y en la ejecución.
40	Una vez terminado el registro, se deben registrar en el servidor.	Correcto	Comprobado mediante test unitario <i>add-Tag_sentRequest_receivedExpected</i> .
41	La etiqueta pulsada se debe corresponder a la etiqueta que se quería seleccionar.	Correcto	Comprobado de forma manual en el código y en la ejecución.
42	Se debe tener una actividad en transcurso.	Correcto	Se ha comprobado de forma manual en la ejecución.
43	Un usuario mayor debe estar identificado en la aplicación.	Correcto	Se ha comprobado de forma manual en la ejecución y con el código.
44	El terapeuta debe estar identificado para visualizar la actividad y sus etiquetas asociadas.	Correcto	Se ha comprobado mediante test unitario <i>test_required_authentication</i> .
45	Las etiquetas deben mostrarse bajo su categoría correspondiente.	Correcto	Se ha comprobado de forma manual en el código y en la ejecución.

46	La hora que se muestra debe corresponder a la hora del registro de dichas etiquetas.	Correcto	Se ha comprobado de forma manual con el código y en la ejecución.
----	--	----------	---

Cuadro 3.9: Resultados de realizar las pruebas de aceptación de la iteración 2

3.4.5. Retrospectiva de la iteración

Esta iteración ha estado marcada por la necesidad de adaptarme rápidamente a otras formas de desarrollo. Para comenzar, hablaremos de los retrasos que se han producido debido al cambio necesario en la tecnología de desarrollo que se iba a usar. Esto ha hecho que el desarrollador tenga que formarse en un lenguaje y en una forma de realizar aplicaciones que no conocía mediante la lectura de guías de desarrollo oficiales y cursos en plataformas de aprendizaje online.

Esto ha servido para poder llevar a cabo una implementación de calidad, asegurando la mantenibilidad y legibilidad del código gracias a llevarlo a cabo mediante Clean Architecture (Picón, 2022) y usar el patrón MVVM.

Además, el desarrollador ha aprendido a usar Figma para el desarrollo de prototipos en alta resolución, lo que ha aportado a las tutoras una visión del producto muy fidedigna y que ha gustado bastante.

Otro de los aspectos que más trabajo ha necesitado es la comunicación entre reloj inteligente y el dispositivo móvil. La documentación que se proporciona es muy poco descriptiva y muy técnica, por lo que se han necesitado numerosas horas para tratar de comprender el funcionamiento de la API de comunicación.

En cuanto a la implementación del sistema web, no han ocurrido problemas relevantes, ya que se ha seguido con la filosofía de implementación que realizábamos en la iteración anterior, por lo que hemos podido servirnos bastante de la reusabilidad del código, especialmente de la API.

Aún habiendo sucedido los inconvenientes mencionados, los objetivos de la iteración se han cumplido correctamente y se ha demostrado su correcto funcionamiento. En las próximas iteraciones debemos ya comenzar la investigación de cómo obtener los sensores del reloj y cómo interpretarlos para conseguir predecir el nivel de estrés, junto con la finalización de tareas de retoques del sistema.

3.5. Tercera iteración

3.5.1. Tarjetas de las Historias de Usuario

Se da comienzo con la extracción de las historias de usuario correspondientes la tercera iteración y el desarrollo de las tarjetas de historias de usuario.

Identificador: HU-27	Terminar una actividad	
Descripción: Como usuario quiero poder parar una actividad para terminar el registro de la medición de estrés.		
Estimación: 3	Prioridad: 16	Entrega: 2
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Se debe haber iniciado una actividad antes de poder pararla. ▪ Se debe tener conexión a internet para poder parar una actividad. ▪ Una vez que se pare la actividad se envían los últimos datos al servidor y se actualiza la fecha de finalización de la actividad. 		
Observaciones:		

Identificador: HU-25	Clasificar etiquetas por tipología.	
Descripción: Como usuario quiero que las etiquetas estén clasificadas según su tipología para tener una mejor organización.		
Estimación: 3	Prioridad: 17	Entrega: 3
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Deben existir pantallas que avisen de la tipología de las etiquetas siguientes a escoger. ▪ Los listados de etiquetas deben de estar encabezados por la tipología de etiquetas que se muestran. ▪ Debe estar una actividad en transcurso y que el usuario pulse sobre registrar etiquetas para que se muestren las etiquetas clasificadas. 		
Observaciones:		

Identificador: HU-26	Selección única o múltiple de etiquetas.	
Descripción: Como usuario quiero poder seleccionar una o más de una etiqueta dependiendo de la categoría para poder incluir mayor detalle de información.		
Estimación: 2	Prioridad: 18	Entrega: 3
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Debe estar una actividad en transcurso y que el usuario haya pulsado sobre el botón de registro de etiquetas para poder seleccionar etiquetas. ▪ Si la categoría es de múltiple selección y se pulsa sobre una etiqueta, esta debe seleccionarse, junto a las demás que se pulsen. ▪ Si la categoría es de selección única, si se pulsa sobre una etiqueta y previamente había una seleccionada, esta última pasará a de seleccionarse en favor a la nueva pulsación. 		
Observaciones:		

Identificador: HU-23	Medir sin Internet.	
Descripción: Como usuario quiero poder realizar la medición incluso si en el transcurso se pierde la conexión de red para no perder los datos recogidos hasta dicho momento.		
Estimación: 3	Prioridad: 19	Entrega: 3
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ La aplicación no se cierra si se produce una desconexión a internet. ▪ La aplicación es capaz de enviar todos los datos recabados cuando recupere la conexión a internet. ▪ La aplicación sigue midiendo en el transcurso de la actividad. ▪ No se puede terminar la actividad en transcurso si no se tiene conexión a internet. 		
Observaciones:		

Identificador: HU-22	Aviso de próxima acción a realizar	
Descripción: Como usuario quiero que se me indique qué pantalla viene a continuación mediante una pantalla informativa para obtener una retroalimentación por parte de lo que esté ocurriendo en el reloj.		
Estimación: 2	Prioridad: 20	Entrega: 3
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Cuando se pulsa un botón, se muestra una pantalla durante unos segundos explicativa indicando qué ocurre a continuación. ▪ Se debe mostrar la pantalla explicativa al menos durante 3 segundos. 		
Observaciones:		

Identificador: HU-19	Recuperar credenciales del usuario mayor	
Descripción: Como usuario quiero poder recuperar mis credenciales en la aplicación móvil para recibir un correo electrónico con ellas y no molestar al terapeuta.		
Estimación: 5	Prioridad: 21	Entrega: 3
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El correo para el cual se recupera la contraseña debe estar registrado en el sistema. ▪ Si el correo existe en el sistema, se debe mandar correctamente un correo con la contraseña del usuario. 		
Observaciones:		

Identificador: HU-18	Cambiar de usuario mayor en el reloj	
Descripción: Como usuario quiero poder cerrar sesión en la aplicación móvil enlazada con el reloj inteligente para que pueda ser usado por otro usuario.		
Estimación: 3	Prioridad: 22	Entrega: 3
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Cuando se inicie sesión en el teléfono móvil se debe notificar a la aplicación del reloj del cambio de usuario que ha sucedido. 		
Observaciones:		

Identificador: HU-13	Visualizar una actividad	
Descripción: Como terapeuta quiero visualizar una actividad mediante las etiquetas mostrándolas con la hora y de acuerdo con su categoría para que se lea de forma rápida.		
Estimación: 3	Prioridad: 23	Entrega: 3
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ La actividad debe estar iniciada en el sistema. ▪ Se deben mostrar las etiquetas correctas asociadas a la actividad. ▪ Se debe mostrar un indicativo cuando la actividad esté sucediendo en directo y no esté terminada. 		
Observaciones:		

3.5.2. Descomposición de las H.U. en tareas de desarrollo

Tras escribir las pruebas de aceptación de las historias de usuario que se abarcan en la tercera iteración, se descompone cada una de ellas en tareas de desarrollo.

HU-27	Como usuario quiero poder parar una actividad para terminar el registro de la medición de estrés. .	3 PH
-------	---	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-53	Añadir al bocetado de la pantalla de medición en la aplicación del reloj un botón para parar la actividad.	0.5
T-54	Implementar el bocetado realizado para para una actividad	1
T-55	Conectar la terminación de la actividad con la API	1
T-56	Realizar las pruebas de aceptación asociadas a terminar una actividad.	0.5

Observaciones:

HU-25	Como usuario quiero que las etiquetas estén clasificadas según su tipología para tener una mejor organización.	3 PH
-------	--	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-57	Realizar bocetado de las pantallas de etiquetado de instante en actividad de la aplicación del reloj.	1
T-58	Implementar bocetado de pantallas de etiquetado de la aplicación del reloj	1.5
T-59	Relizar pruebas de aceptación asociadas a la clasificación de etiquetas en la aplicación del reloj	0.5

Observaciones:

HU-26	Como usuario quiero poder seleccionar una o más de una etiqueta dependiendo de la categoría para poder incluir mayor detalle de información..	2 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-60	Añadir al boceto de la pantalla de etiquetas una casilla de selección.	0.5
T-61	Implementar el nuevo boceto de la pantalla de etiquetas con la casilla de selección.	1
T-62	Realizar pruebas de aceptación asociadas a la selección única o múltiple de etiquetas.	0.5
Observaciones:		

HU-23	Como usuario quiero poder realizar la medición incluso si en el transcurso se pierde la conexión de red para no perder los datos recogidos hasta dicho momento.	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-63	Implementar la funcionalidad necesaria para no perder datos si se desconecta el reloj de internet	2.5
T-64	Realizar las pruebas de aceptación asociadas a la continuación de la medición sin conexión a la red	0.5
Observaciones:		

HU-22	Como usuario quiero que se me indique qué pantalla viene a continuación mediante una pantalla informativa para obtener una retroalimentación por parte de lo que esté ocurriendo en el reloj.	2 PH
-------	---	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-65	Realizar bocetado de pantallas de aviso de acción a continuación para el reloj	0.75
T-66	Implementar bocetado de las pantallas de aviso del reloj.	1
T-67	Realizar las pruebas de aceptación asociadas a las pantallas de aviso de acción a continuación del reloj	0.25

Observaciones:

HU-19	Como usuario quiero poder recuperar mis credenciales en la aplicación móvil para recibir un correo electrónico con ellas y no molestar al terapeuta.	5 PH
-------	--	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-68	Diseñar e implementar el correo electrónico a mandar.	1
T-69	Implementar el mandar correos electrónicos a través de una petición en la API.	1.5
T-70	Implementar pantalla en la aplicación móvil para poder rellenar con el correo el formulario de recuperación.	1.5
T-71	Realizar las pruebas de aceptación asociadas a recuperar credenciales por parte del usuario mayor.	1

Observaciones:

HU-18	Como usuario quiero poder cerrar sesión en la aplicación móvil enlazada con el reloj inteligente para que pueda ser usado por otro usuario.	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-72	Realizar boceto donde se pueda cerrar sesión en la aplicación móvil por parte del usuario mayor.	0.5
T-73	Implementar el boceto asociado al cierre de sesión por parte del usuario mayor.	1.5
T-74	Realizar las pruebas de aceptación asociadas a cerrar sesión en la aplicación móvil	1
Observaciones:		

HU-13	Como terapeuta quiero visualizar una actividad independientemente de que esté finalizada o esté en transcurso para tener un seguimiento completo del usuario.	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-75	Añadir al bocetado de la sección de actividades el indicativo de si una actividad está en transcurso o está finalizada.	1
T-76	Implementar funcionalidad necesaria para mostrar actividades no finalizadas en el listado de actividades.	1.5
T-77	Realizar pruebas de aceptación asociadas a visualizar actividades en transcurso o finalizadas.	0.5
Observaciones:		

3.5.3. Planificación temporal de la tercera iteración

En la figura 3.44 se puede visualizar un diagrama de Gantt para la planificación detallada de la tercera iteración.

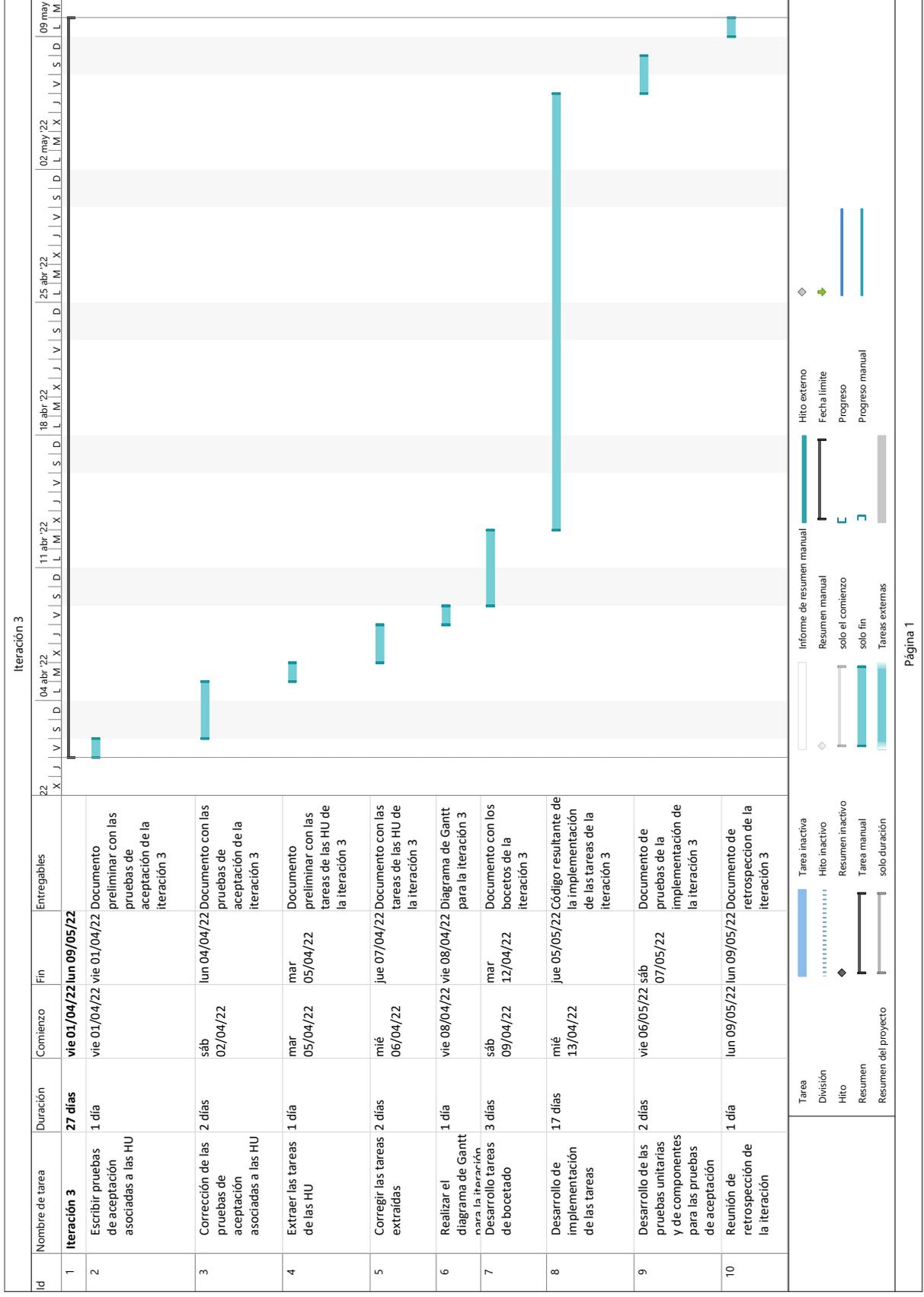


Figura 3.44: Diagrama de Gantt para la planificación de las tareas de la iteración 3

3.5.4. Realización de las tareas de desarrollo

Respetando la estructura de iteraciones anteriores, comenzaremos con las tareas de bocetado. Cabe mencionar que algunas de las tareas de bocetado de esta iteración están relacionadas con añadir ciertos elementos al bocetado de iteraciones anteriores.

Comenzaremos exponiendo los bocetos correspondientes al sistema web RoR. El boceto de la figura 3.45 añade, al boceto ya realizado para la sección de actividades, una forma de indicar si una actividad está en transcurso o si ya está finalizada. Para ello, nos serviremos de la parte baja del elemento de tarjeta que lista las actividades y colocaremos aquí, en el caso de que la actividad esté terminada, la fecha de finalización, o, en caso contrario, un icono a modo de indicación para conocer que la actividad está en transcurso.

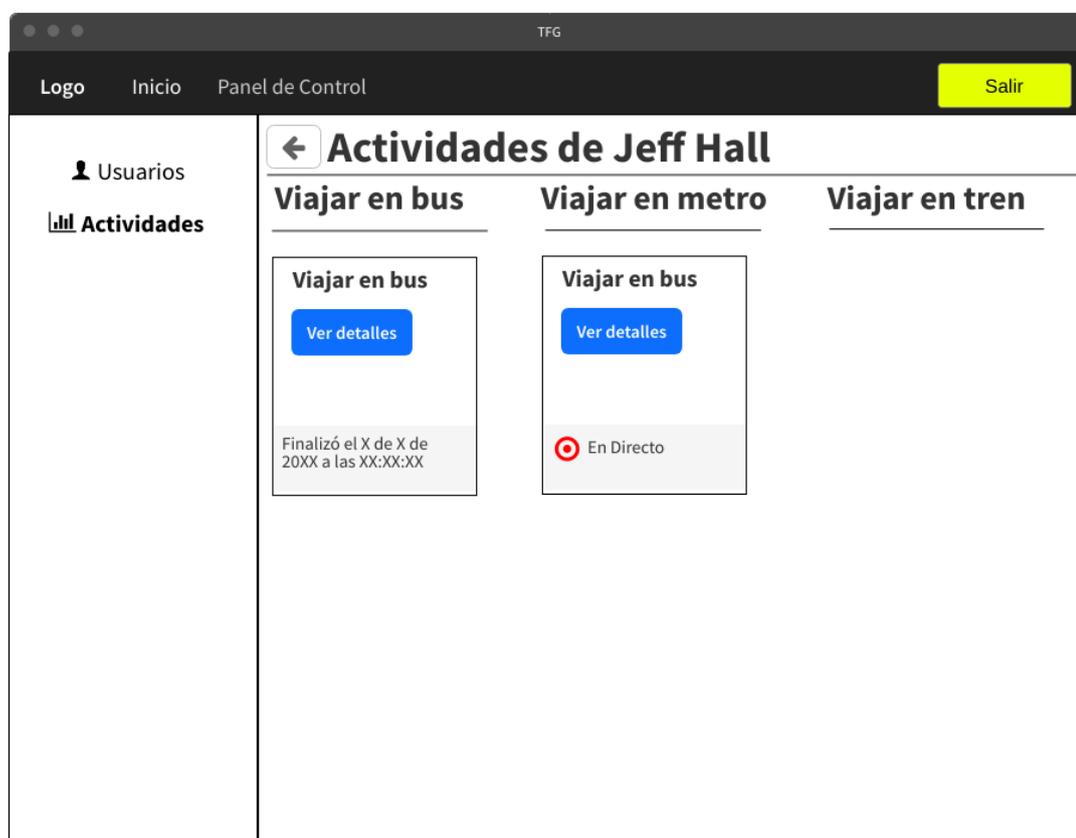


Figura 3.45: Nuevo bocetado de la sección de visualización de actividades para un usuario en el sistema web

Además, se ha realizado el diseño del correo que envía el sistema cuando el usuario mayor solicita recuperar su contraseña. Hasta ahora no estaba esta funcionalidad totalmente implementada. Tan sólo se hacía la petición al sistema web, pero éste no

respondía a la petición mandando el mensaje al usuario. La forma en la que el usuario mayor recibirá su recordatorio de contraseña puede visualizarse en la figura 3.46. En cuanto a tareas de bocetado relacionadas con el sistema web, estarían todas finalizadas.



Figura 3.46: Resultado de diseño del correo de recordatorio de contraseña por parte del usuario mayor.

A continuación se exponen las tareas de bocetado en relación a la aplicación del reloj inteligente.

La tarea relacionada con añadir al bocetado de la pantalla del menú de medición un botón de detención ha sido desarrollada. Para esto, ha sido necesario aportar el token junto con el id de la actividad a finalizar y el instante de terminación, que corresponderá al momento en el que el usuario pulse el botón de detenimiento. Además, las tutoras realizaron la sugerencia de que las etiquetas estuviesen clasificadas según su tipología, poder seleccionar más de una etiqueta en algunas secciones, añadir un icono si se pierde la conexión, y las pantallas informativas para la secuencia de registro de etiquetas. También se ha añadido una pantalla adicional donde se indica si el reloj usado es incompatible con la aplicación. El resultado de estas modificaciones se puede ver en la figura 3.47, apreciándose que está diseñado, de nuevo, en alta resolución, ya que no es tanto el esfuerzo necesario para añadir nuevas pantallas a partir de las ya realizadas y modificar las que se necesitasen, y da una visión más real del producto final.

La modificación más relevante es la clasificación según la tipología de las etiquetas. Previamente se encontraban las etiquetas en una misma pantalla y era más complicado para el usuario diferenciar entre seleccionar estados de viaje y contextos en los que se encontraba. Por esto se ha realizado una diferenciación entre estos dos tipos de etiquetas y se ha separado en dos pantallas el registro de estas.

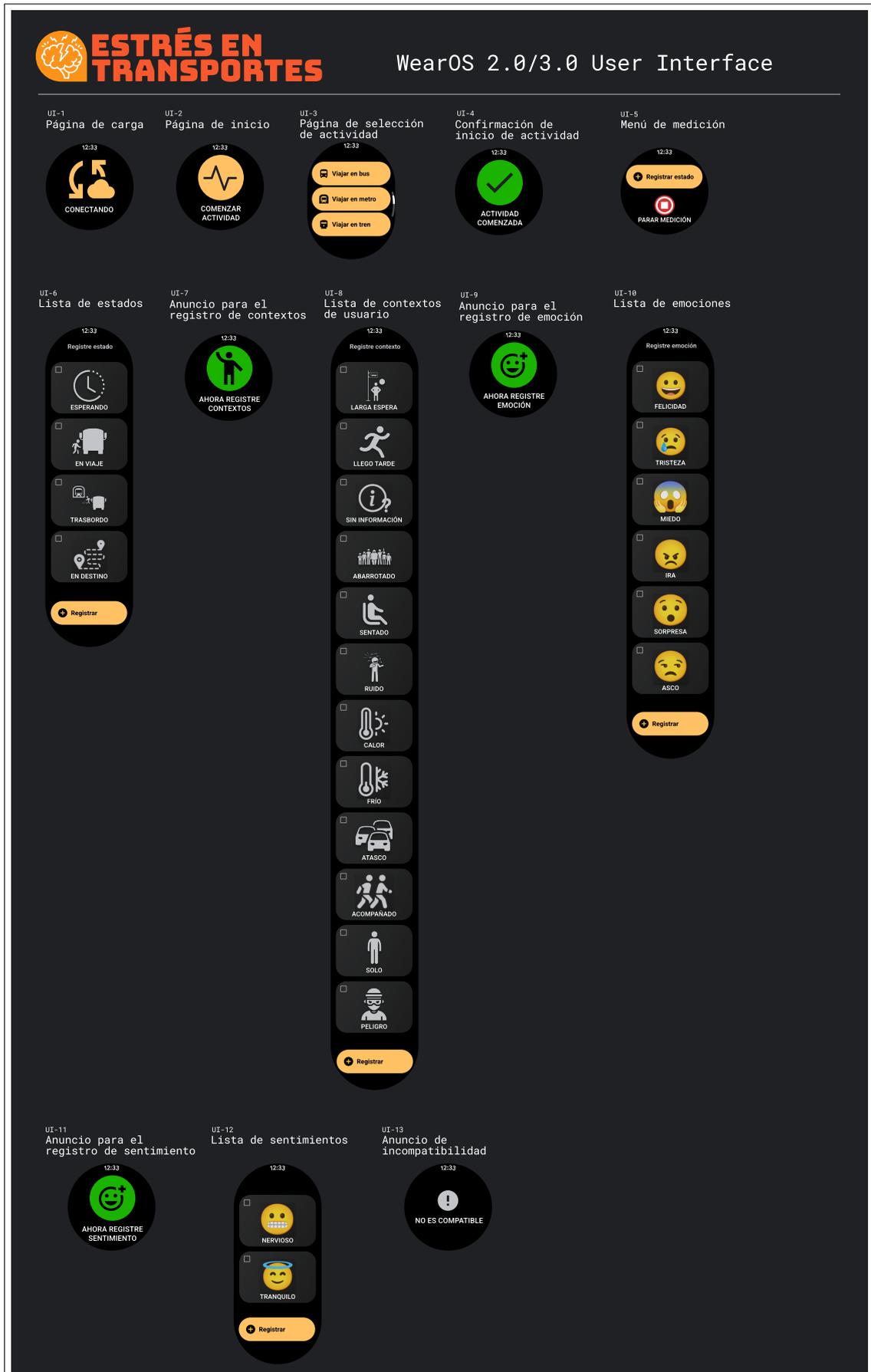


Figura 3.47: Diseño de alta resolución con Figma de la aplicación WearOS en la iteración 3

En cuanto a tareas de bocetado de la aplicación móvil, se han añadido las funcionalidades de recuperar contraseña y de cierre de sesión. Los bocetos que implementan estas funcionalidades, incluyendo su navegación, se pueden ver en la figura 3.48.



Figura 3.48: Pantalla de recuperación de contraseña en Android

Por último se muestra en la figura 3.48 la pantalla de recuperación de contraseña por parte del usuario mayor en la aplicación Android. Esta se compone por un botón de regreso a la pantalla principal y por un formulario en el que se debe introducir el correo electrónico para poder recuperar la contraseña.

Para mostrar la navegabilidad de la aplicación, se muestran en la figura 3.49 estos bocetos con flechas que indican hacia qué boceto se dirige cuando se pulsa sobre un elemento.

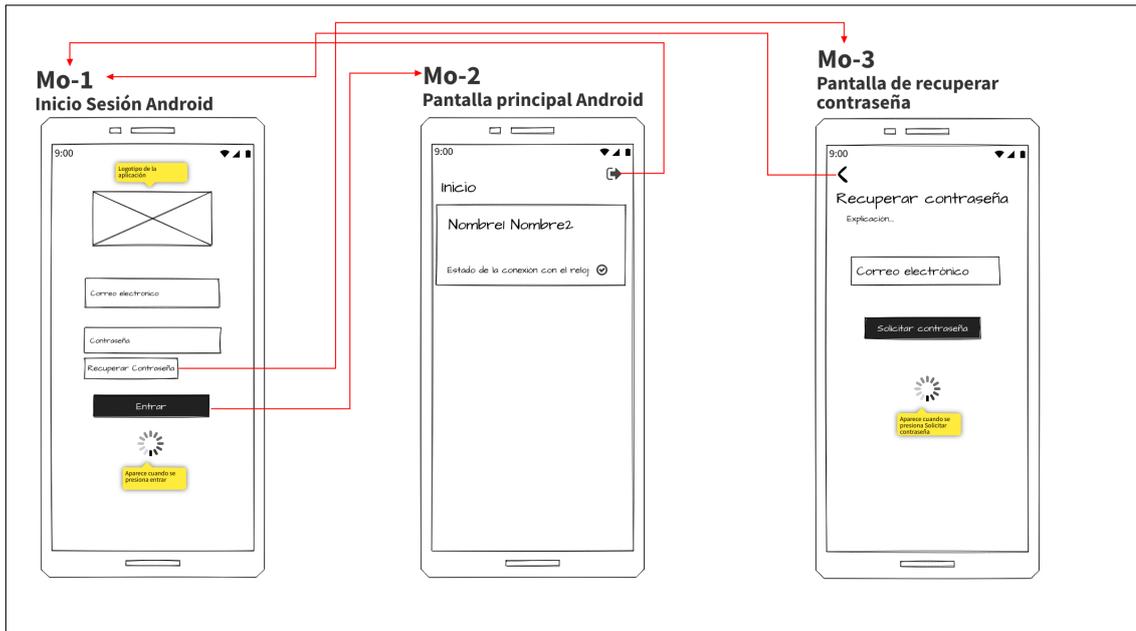


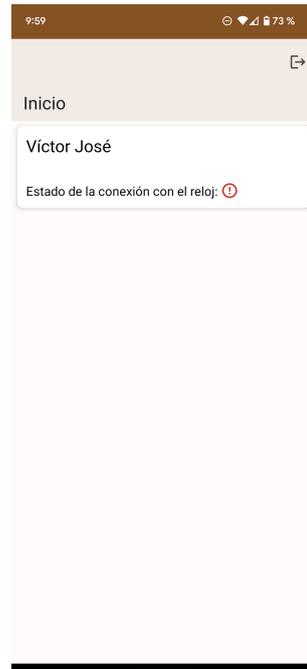
Figura 3.49: Diseño de la navegabilidad de la aplicación Android en la iteración 3

Tras implementar los bocetos, realizaremos la implementación de las pantallas. Se puede visualizar el resultado en las figuras 3.50a, 3.50b y 3.50c. Ahora tenemos la funcionalidad de poder cerrar sesión una vez esté el usuario identificado. Para la implementación de la recuperación de contraseña se ha optado por una animación *Lottie* para describir gráficamente la finalidad, junto con una descripción en texto.

Hecha la implementación, obtenemos el diagrama de paquetes mostrado en la figura 3.51 para la aplicación Android. Junto a este, se proporcionan los diagramas de clases correspondientes a cada paquete en las figuras 3.52, 3.53 y 3.54.



(a) Pantalla Principal de App Android



(b) Pantalla de Identificados App Android



(c) Pantalla de Recuperación de Contraseña

Figura 3.50: Resultados de implementación en Android de los bocetos de la iteración 3.

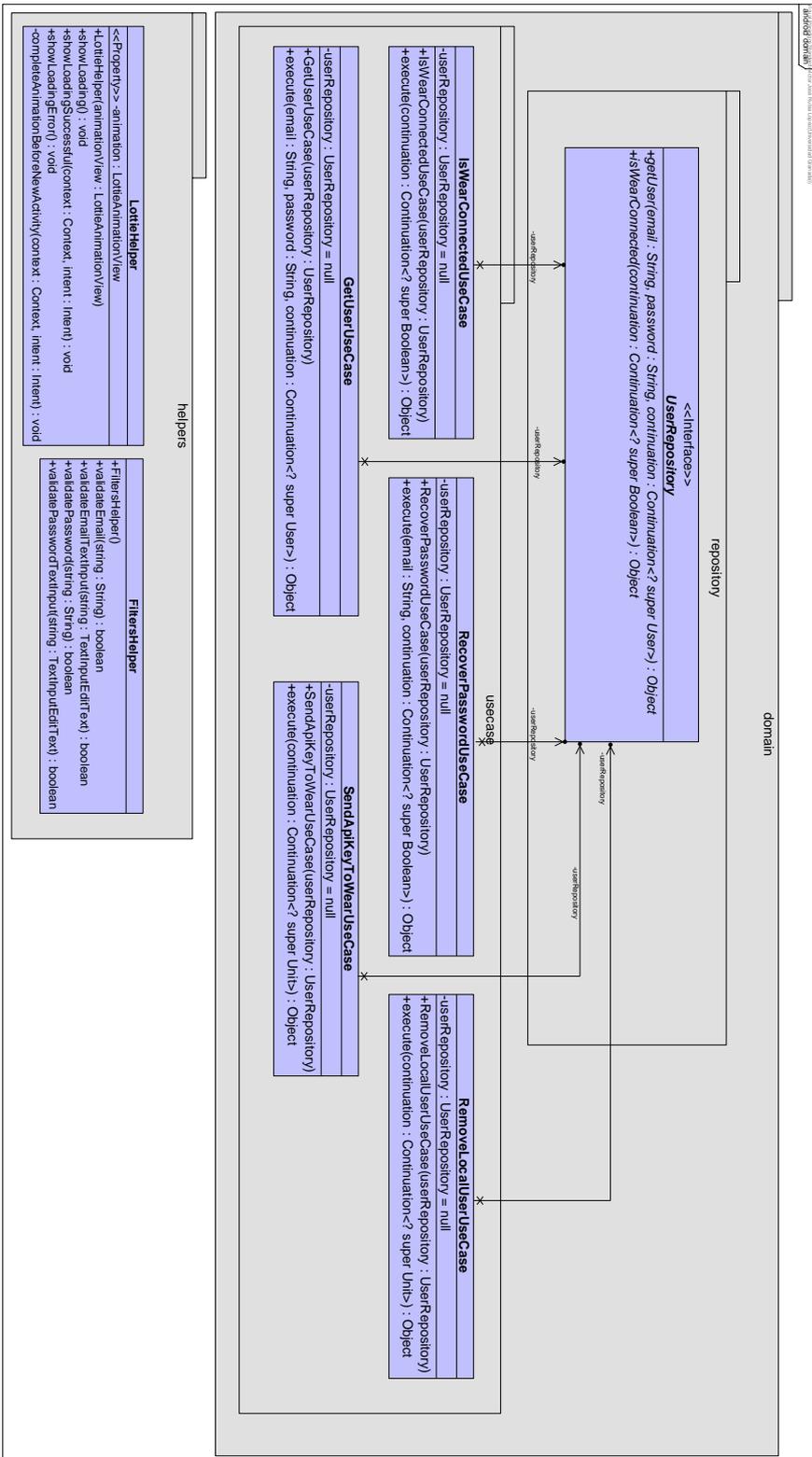


Figura 3.53: Diagrama de clases correspondiente al paquete *domain* de la aplicación Android en la iteración 3

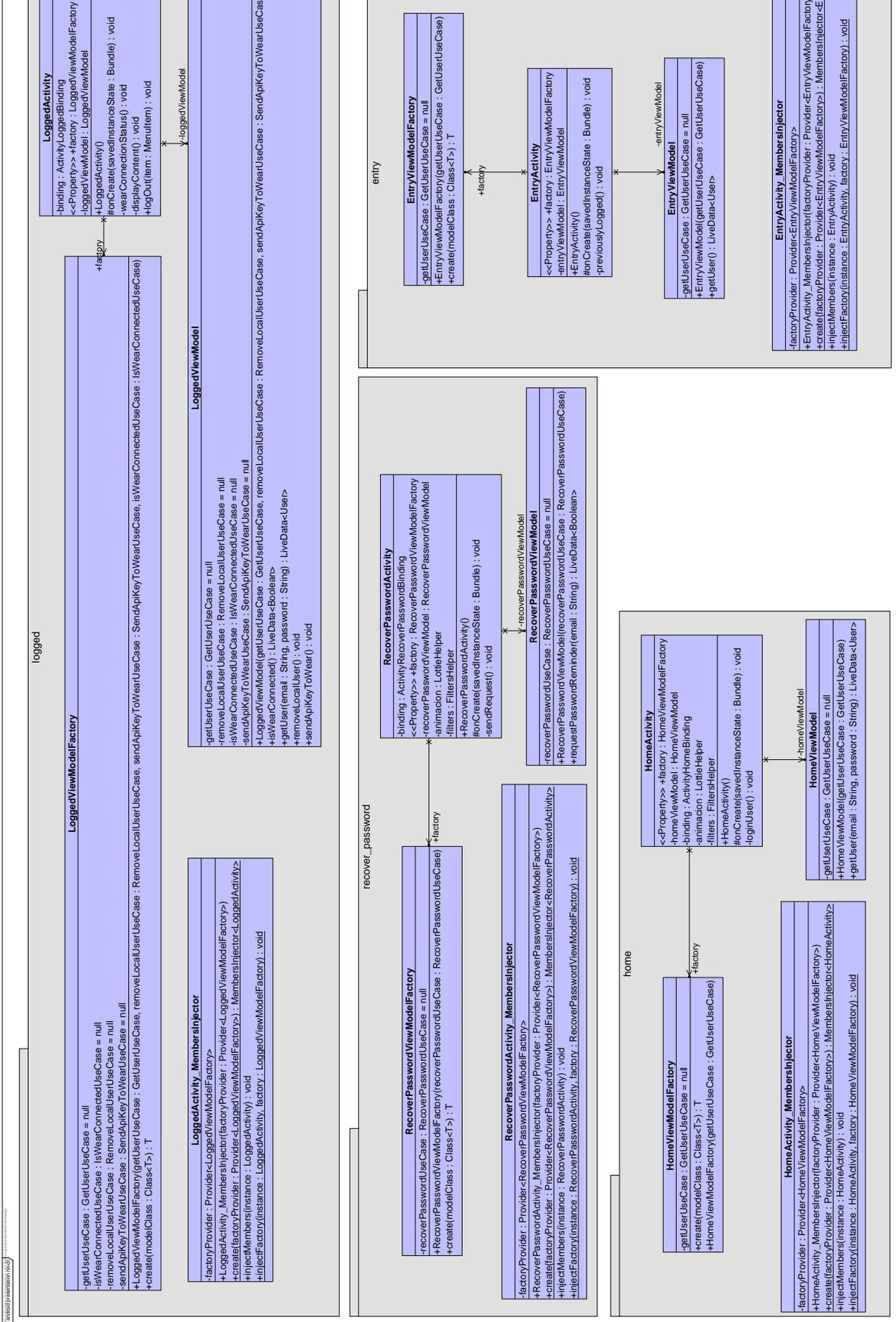


Figura 3.54: Diagrama de clases correspondiente al paquete *presentation* de la aplicación Android en la iteración 3

Para terminar con las tareas de desarrollo relacionadas con la aplicación Android, se deben realizar las pruebas de aceptación de dichas tareas. En la tabla de la figura 3.55 se pueden visualizar dos nuevos tests unitarios y de componentes que prueban la funcionalidad de solicitar un recordatorio de contraseña y el cierre de sesión del usuario mayor.

TestRun	Total	Passed
Unit Tests Mobile - Generated by IntelliJ IDEA on 29/4/22 9:30	8	8
TFGServiceTest		
TFGServiceTest.recoverPassword_receivedResponse_correctContent	1445ms	passed
TFGServiceTest.getUser_receivedResponse_correctContent	36ms	passed
TFGServiceTest.getUser_sentRequest_receivedExpected	15ms	passed
FiltersHelperTest		
FiltersHelperTest.validatePassword_returnsCorrectVerdict	7943ms	passed
FiltersHelperTest.validateEmail_returnsCorrectVerdict	88ms	passed
HomeViewModelTest		
HomeViewModelTest.getUser_returnCurrentRecord	312ms	passed
LoggedViewModelTest		
LoggedViewModelTest.removeUser_returnsNull	30ms	passed
RecoverPasswordViewModelTest		
RecoverPasswordViewModelTest.recoverPassword_returnTrue	23ms	passed

Figura 3.55: Tests Unitarios y de Componentes de la Aplicación Android en la iteración 3

A continuación, se describirá la forma en la que se ha llegado a desarrollar la funcionalidad de medición de estrés en función de los datos recabados por los sensores del reloj inteligente. Se puede consultar este proceso de medición de forma visual en la figura 3.68.

Para comenzar el proceso vamos a optar por la medición de estrés mediante la Variación de Frecuencia Cardíaca (VFC), que como se ha expuesto en el capítulo 2, es un método bastante extendido. Para obtener esta variación de frecuencia cardíaca necesitamos tener una representación de las ondas generadas por la actividad cardiaca.

Para conocer si el reloj inteligente proporcionado para el proyecto posee sensores de representación de actividad cardíaca codificamos un script que nos liste los sensores disponibles, el cual se puede ver en la figura 3.7

```

1 val sensorList: List<Sensor> = mySensorManager!!. getSensorList(Sensor.TYPE_ALL)
2 for (currentSensor in sensorList) {
3     Log.d(
4         "List sensors",
5         "Name: " + currentSensor.name + " /Type_String: " + currentSensor.
6             ↳ stringType + " /Type_number: " + currentSensor.type
7     )
8 }

```

Listing 3.7: Script para listar los sensores del reloj

Hecho esto, se ha tomado el reloj inteligente del proyecto y se ha ejecutado el script 3.7 para conocer la lista de sensores que posee. Dichos sensores se listan en la figura 3.8

```

1 Accelerometer /Type-String: android.sensor.accelerometer /Type-number: 1
2 Gyroscope /Type-String: android.sensor.gyroscope /Type-number: 4
3   Name: Magnetometer /Type-String: android.sensor.magnetic_field /Type-number: 2
4   Name: Light /Type-String: android.sensor.light /Type-number: 5
5 Orientation /Type-String: android.sensor.orientation /Type-number: 3
6 Significant Motion /Type-String: android.sensor.significant_motion /Type-number:
   ↪ 17
7   Name: Gravity /Type-String: android.sensor.gravity /Type-number: 9
8 Linear Acceleration /Type-String: android.sensor.linear_acceleration /Type-number:
   ↪ 10
9   Name: Rotation Vector /Type-String: android.sensor.rotation_vector /
   ↪ Type-number: 11
10 Geomagnetic Rotation Vector /Type-String: android.sensor.
   ↪ geomagnetic_rotation_vector /Type-number: 20
11   Name: Game Rotation Vector /Type-String: android.sensor.game_rotation_vector /
   ↪ Type-number: 15
12   Name: Magnetometer (uncalibrated) /Type-String: android.sensor.
   ↪ magnetic_field_uncalibrated /Type-number: 14
13 Gyroscope (uncalibrated) /Type-String: android.sensor.gyroscope_uncalibrated /
   ↪ Type-number: 16
14 Heart Rate PPG /Type-String: android.sensor.heart_rate /Type-number: 21
15 Tilt Detector /Type-String: android.sensor.wrist_tilt_gesture /Type-number: 26
16 Activity Recognizer /Type-String: android.sensor.mobvoi_activity_recognizer /
   ↪ Type-number: 69632
17   Name: Step Detector /Type-String: android.sensor.step_detector /Type-number:
   ↪ 18
18 Step Counter /Type-String: android.sensor.step_counter /Type-number: 19
19   Name: Activity Data /Type-String: android.sensor.mobvoi_activity_data /
   ↪ Type-number: 69633
20   Name: Off Body Detector /Type-String: android.sensor.
   ↪ low_latency_offbody_detect /Type-number: 34
21 Heart Rate PPG Raw Data /Type-String: com.google.wear.sensor.ppg /Type-number:
   ↪ 65572

```

Listing 3.8: Listado de sensores con sus identificadores del TicWatch Pro

En este listado encontramos un único sensor que tiene el dispositivo para conocer la frecuencia cardíaca siendo el sensor PPG. Por este motivo, en nuestro proyecto se opta a usar el sensor PPG para medir el nivel de estrés. En nuestro caso, escogeremos el sensor por el que podemos obtener los datos en crudo, siendo este el identificado por *com.google.wear.sensor.ppg*.

A continuación, se procede a investigar cómo se realiza la obtención de los datos de un sensor. Para lograrlo, he creado una aplicación para el reloj, que solicita permisos para acceder a los sensores. Además, se debe tener un gestor de sensores declarado en la clase de la aplicación, el cual se encargará de registrar y suprimir las vinculaciones a sensores en el transcurso de la aplicación.

Para continuar, se le solicita al gestor de sensores la lista de sensores del dispositivo, sobre la cual iteraremos buscando aquel que sea el PPG de datos en crudo. Cuando se haya localizado, nos guardaremos su identificador para, posteriormente, indicar al gestor de sensores que queremos su registro. Para registrarlo, se debe indicar la frecuencia de muestreo del sensor que se desea, que en nuestro caso será el predeterminado. Además,

para completar su registro, la clase debe extender de *SensorListener*, que implementa funciones que se ejecutan dependiendo de la información que se reciba del sensor. La más relevante es una llamada *onSensorChanged*, pues es la que se ejecuta cada vez que el sensor toma una muestra. En esta función debemos implementar la lógica para almacenar la información local o remotamente del sensor.

Tras numerosos intentos de obtener datos del sensor de PPG, finalmente he podido estudiar que se obtiene en dos canales diferentes, aunque en nuestro proyecto solo vamos a usar los datos que se obtienen por uno de ellos. Tras hacer varias mediciones y obtener datos numéricos en formato .CSV a través de la memoria del reloj, los he analizado en Excel. He podido representarlos en un gráfico, sin embargo, estos carecían de la forma de una señal PPG típica. Por esto, he procedido a normalizar los datos con valores entre 0 y 1 y tras representarlos en un gráfico, he podido ver que toman una forma más similar a una señal PPG, lo cual se muestra en la figura 3.56.

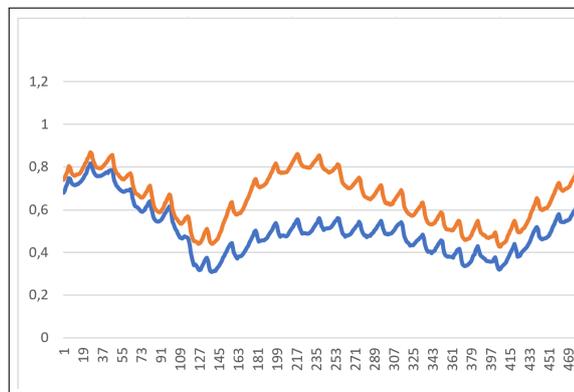


Figura 3.56: Representación gráfica de los datos obtenidos en crudo

Sin embargo, tras la lectura de artículos relacionados con la extracción de características de datos PPG crudos, (Heo y col., 2021) y (Gradl y col., 2019), y tras sospechar que estas ondas que he representado son bastante irregulares, se constata que la lectura del sensor PPG es susceptible a ruidos y será necesario implementar un proceso de limpieza.

Esto me lleva a realizar otro periodo de investigación de métodos de limpieza ya existentes de señales PPG obtenidas por dispositivos *wearable*. Tras la lectura de los artículos expuestos en el capítulo 2, se ha llegado a descubrir un módulo de Python llamado Heartpy (van Gent, 2019), el cual es capaz de procesar señales PPG obtenidas mediante smartwatch y obtener información bastante relevante, como los datos que se listan a continuación:

- Pulsaciones por minuto (bpm).
- Intervalo de tiempo entre pulsaciones (ibi).
- La desviación típica de los intervalos R-R normales (sdnn).

- La desviación típica de las diferencias sucesivas entre NNs (intervalos R-R normalizados) adyacentes (sdsd).
- La raíz cuadrada de la media de los cuadrados de las diferencias sucesivas entre NNs adyacentes (rmssd).
- La proporción de NN20 dividida por el número total de NN (pnn20).
- La proporción de NN50 dividida por el número total de NN (pnn50).
- Desviación absoluta mediana de los intervalos RR (hr_mad).
- Ratio de respiraciones en Hz (breathingrate).

Estos datos se pueden obtener para una serie de mediciones acotadas o para el conjunto completo. Por esto, concluimos que este módulo será el elegido para continuar la investigación e interpretar la señal PPG.

Se comienza el estudio investigando cómo usar este módulo para la limpieza de nuestra señal PPG con la creación de un script de Python que importe dicho módulo. Primeramente, se debe tener un .CSV que contenga los datos en crudo del sensor PPG y su estampa de tiempo asociada. Este archivo será importado en el script mediante la librería *Pandas*. Podemos ver la representación gráfica de la muestra que hemos tomado a modo de ejemplo representada en la gráfica de la figura 3.57.

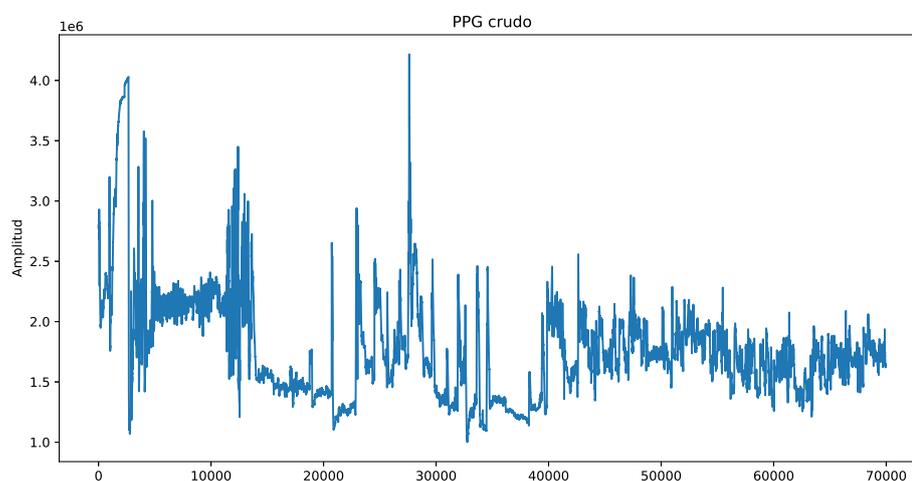


Figura 3.57: Representación gráfica de la señal de PPG obtenida en un experimento

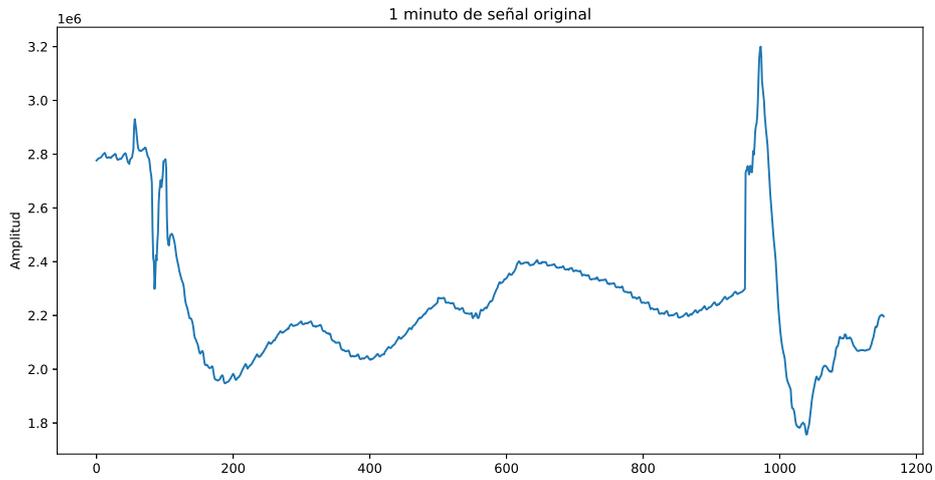
Se observa que existe un gran ruido en fases iniciales de la medición, lo que puede deberse a que el usuario estuviese ajustándose el reloj mejor a su muñeca en el transcurso

del experimento y al despegar el sensor de la muñeca la señal se ve afectada. Posteriormente se observa un periodo más regular, en el que el sujeto estaría seguramente en reposo. Tras este periodo, se vuelve a apreciar una amplitud bastante grande y una posterior disminución en la amplitud de la onda. Poco antes de la mitad de la medición se observa un pico que puede estar provocado por factores como la separación del sensor del reloj de la muñeca o algún movimiento brusco. Incluso, puede darse el caso en el que existan secciones donde intermitentemente el sensor del reloj no haya medido nada.

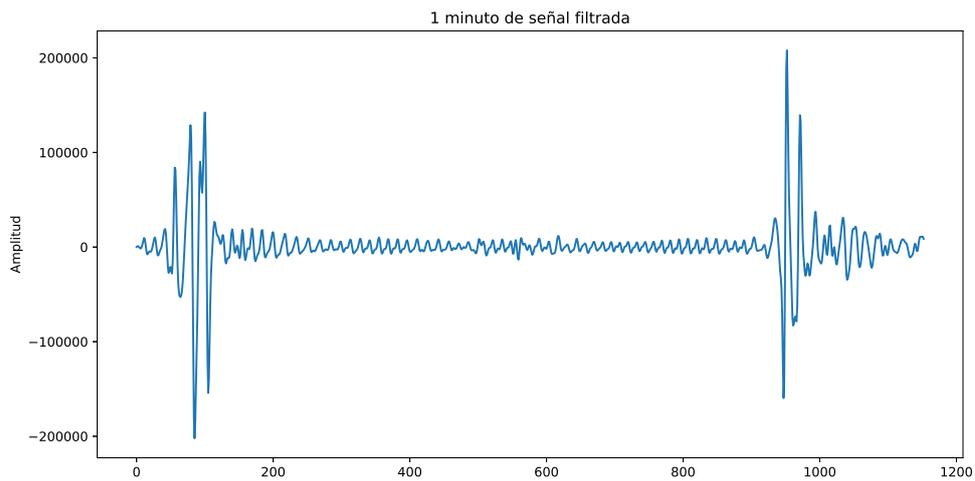
Para poder realizar el proceso de limpieza, es necesario que el módulo conozca primero la frecuencia de muestreo del dispositivo. Como esta frecuencia es dinámica y depende de cada dispositivo, la calcularemos en base al conjunto de mediciones gracias a que tenemos los instantes de tiempo de cada medición.

Para este caso de prueba se obtiene que la frecuencia de muestreo es de 19.231 Hz. Con esta información podemos filtrar la señal gracias a la función *filter_signal* que proporciona el módulo basada en un filtro Butterworth de paso de banda que elimina aquellas señales que están por debajo de 0.7 Hz, correspondientes a 42 pulsaciones por minuto y por encima de 3.5 Hz, correspondientes a 210 pulsaciones por minuto, ya que se considera que esto no son frecuencias válidas para el ritmo cardíaco de una persona.

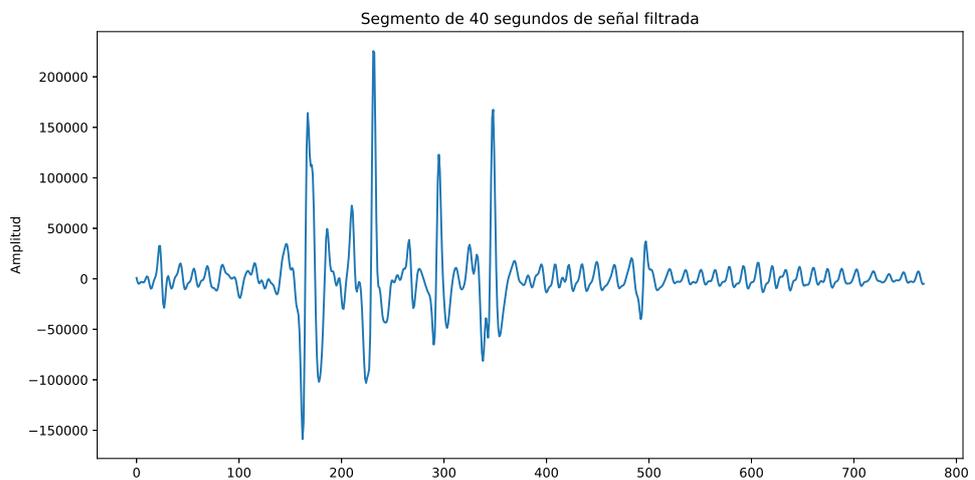
En las figuras 3.58a, 3.58b y 3.58c se observa el resultado de la aplicación del filtro sobre la medición obtenida en este experimento.



(a) Un minuto de medición en crudo.



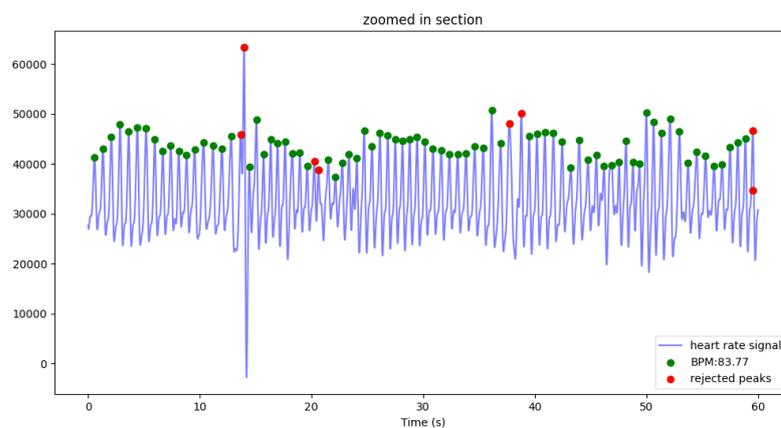
(b) Un minuto de medición con el filtro aplicado.



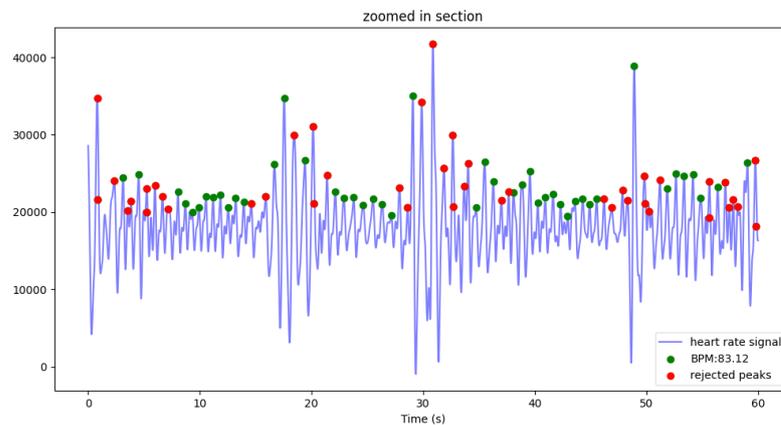
(c) 40 segundos de medición con el filtro aplicado.

Figura 3.58: Filtrado de paso de banda mediante Butterworth de la señal PPG

Tras el filtrado, como nuestro sensor PPG mide con una frecuencia de muestreo de 19 Hz, debemos volver a hacer el muestreo a 100 Hz, ya que la frecuencia de muestreo que obtenemos con nuestro sensor es algo baja para el análisis de umbral adaptativo que usa el módulo HeartPy. Asimismo, debemos recalcular la nueva frecuencia de muestreo. Una vez tenemos la limpieza de la señal hecha, podemos valernos de la funcionalidad de este módulo que es capaz de extraer características relevantes relacionadas con la VFC de segmentos de la medición. Para ello, dividiremos la señal en segmentos lo suficientemente amplios como para que el módulo pueda extraer características más fiables. Obtendremos el procesamiento gráficamente para conocer el funcionamiento en profundidad. Se pueden visualizar en las figuras 3.59a y 3.59b.



(a) Sección de 1 minuto de señal filtrada procesada.



(b) Otra sección de 1 minuto de señal filtrada procesada

Figura 3.59: Filtrado de paso de banda mediante Butterworth de la señal PPG

Como se observa en la leyenda, el módulo es capaz de distinguir cuales son picos

anómalos de otros que considera que son correctos. En base a esto es capaz de medir características de la señal como los que se listan en la tabla 3.10

Figura 3.59a	Figura 3.59b
bpm: 83.772170	bpm: 83.123254
ibi: 716.228316	ibi: 721.819670
sdsn: 63.748882	sdsn: 92.076990
sdsd: 21.613480	sdsd: 29.186428
rmssd: 34.147942	rmssd: 50.178252
pnn20: 0.520548	pnn20: 0.757576
pnn50: 0.136986	pnn50: 0.272727
hr_mad: 58.397941	hr_mad: 57.198113
sd1: 24.056773	sd1: 35.389027
sd2: 85.599175	sd2: 109.618849
s: 6469.293007	s: 12187.194424
sd1/sd2: 0.281040	sd1/sd2: 0.322837
breathingrate: 0.166667	breathingrate: 0.266667

Cuadro 3.10: Características obtenidas del procesamiento de los segmentos de señal filtrados

A continuación se trata de investigar cómo usar estas características para la predicción del nivel de estrés en personas. Tras leer artículos acerca de modelos predictivos de estrés basados en biometría (Ahn y col., 2019), (Subhani y col., s.f.) y (Gradl y col., 2019), llegué a encontrar el conjunto de datos SWELL (Koldijk y col., 2014). En este proyecto finalmente optamos por usar Machine Learning para entrenar un modelo usando este conjunto, ya que posee grandes cantidades de datos clasificados según tres tipos de estrés: *No estresado*, *Periodos de Interrupción* (los cuales podríamos interpretar como estrés medio) y *Periodos de Estrés*. Otra razón que apoya nuestra elección es que la clasificación de este modelo se hace en base a características muy similares a las que poseemos.

No usaré mis propios datos obtenidos ya que no se ha podido trabajar con sujetos durante la experimentación debido a la pandemia. Además, el reducido tiempo para la realización del proyecto ha derivado en la toma de decisión de usar algo que ya estaba hecho. Obtener datos y etiquetarlos es una tarea que requiere bastante tiempo, por lo que para obtener un modelo de calidad, serían necesarios obtener bastantes mediciones.

Comenzaremos estudiando el conjunto de datos mediante la realización del gráfico de clases. Se puede visualizar en la figura 3.60 que las clases están desbalanceadas. Se ha probado a hacer *oversampling* y *undersampling* pero el resultado, que posteriormente se analiza, no cambia, por lo que continuaremos dejando el conjunto así.

Si obtenemos la matriz de correlación de Pearson para este conjunto, visible en la figura 3.62, se puede hacer una selección de aquellas características (*features*) que poseen una correlación de Pearson positiva con las condiciones de estrés. Estas se muestran en

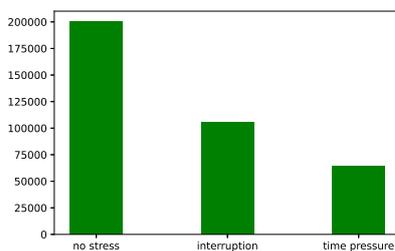


Figura 3.60: Cantidad de datos pertenecientes a cada clase del conjunto de datos.

la tabla 3.11.

MEAN_RR	MEDIAN_RR
SDRR_RMSSD	MEDIAN_REL_RR
SDRR_RMSSD_REL_RR	

Cuadro 3.11: Variables con una correlación de Pearson positiva respecto a la condición de estrés

Esto supone que debemos de implementar en el módulo HeartPy los cálculos necesarios para estas variables, ya que no nos lo proporciona de forma directa. Tras la implementación de las fórmulas que se exponen en el capítulo 2 para el cálculo de estas variables en el módulo HeartPy, procedemos a acotar las variables del conjunto de datos quedándonos con las que se pueden extraer del módulo HeartPy.

En la figura 3.62 se muestra la matriz de correlación de Pearson con el conjunto de variables reducidas. En ella se ve, como el aportar las variables que no son las de la tabla 3.11 es algo que no tiene correlación con la condición de estrés. Sin embargo, como el módulo nos proporciona igualmente esta información, seguiremos adelante con esta forma, ya que no deja de ser información adicional.

La necesidad de reducir las variables no sólo es por el hecho de quedarnos con las variables con una correlación positiva de Pearson, si no porque el conjunto de datos posee variables obtenidas a partir de dispositivos que aportan información que no podemos obtener mediante nuestro *smartwatch* como, por ejemplo, la conductancia de la piel. Vemos en las figuras 3.63a y 3.63b una gráfica de dispersión que corrobora el resultado de la matriz de correlación de Pearson.

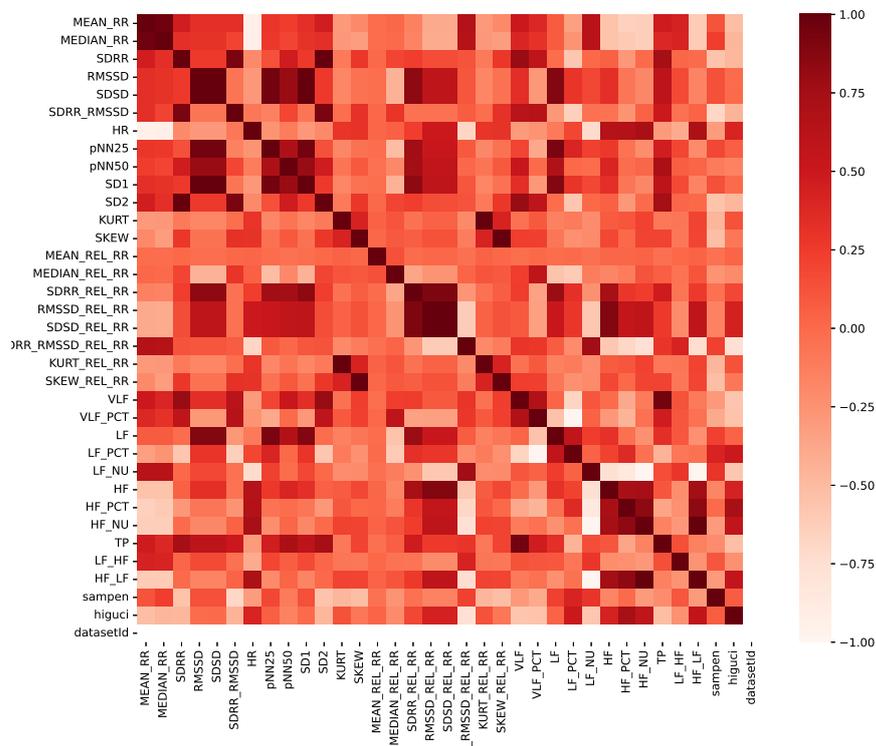


Figura 3.61: Matriz de correlación de Pearson para comprobar la dependencia de las variables.

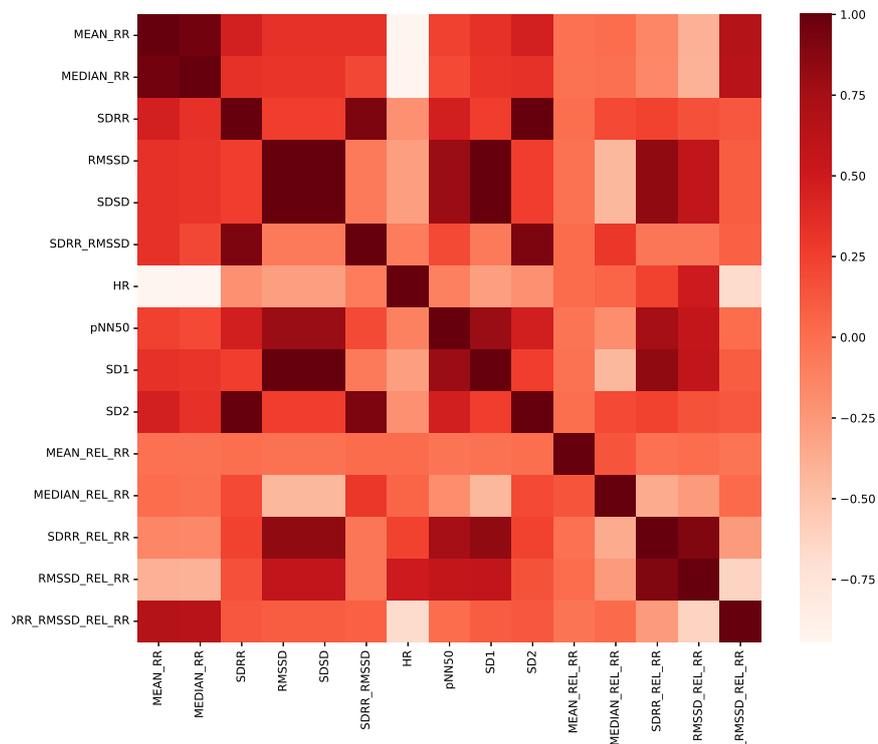
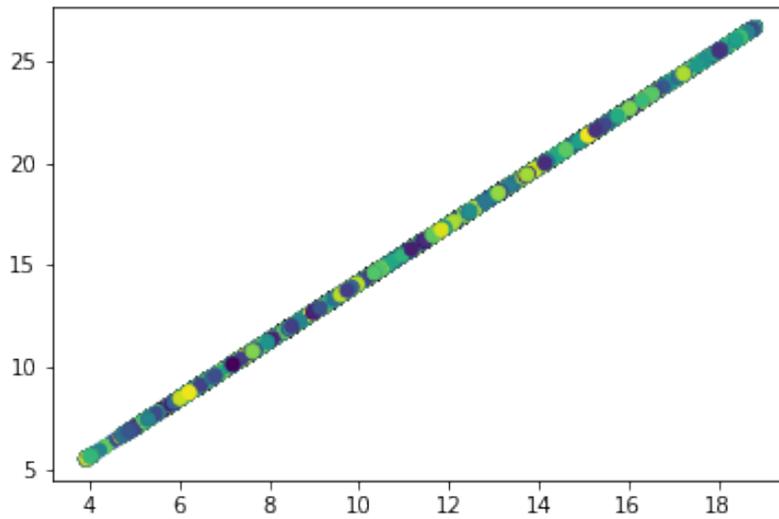
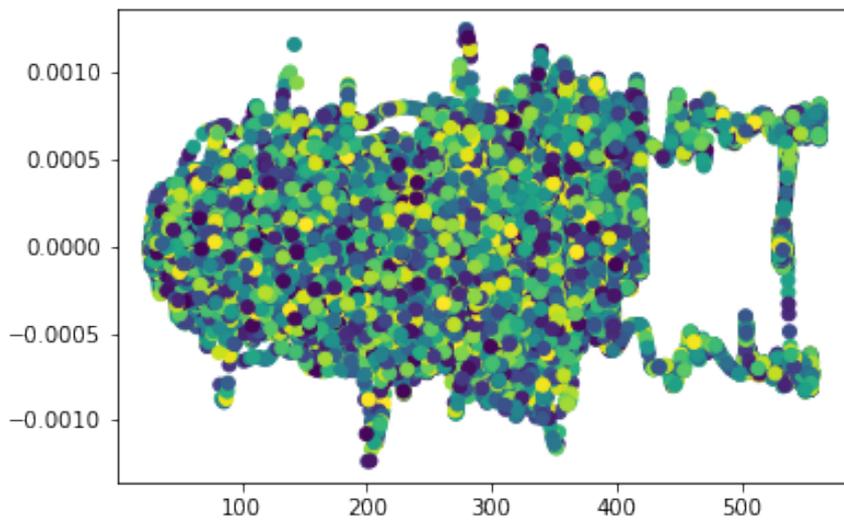


Figura 3.62: Matriz de correlación de Pearson tras aportar las variables que podemos obtener.



(a) Gráfico de dispersión entre SD1 y RMSSD. Alta correlación.



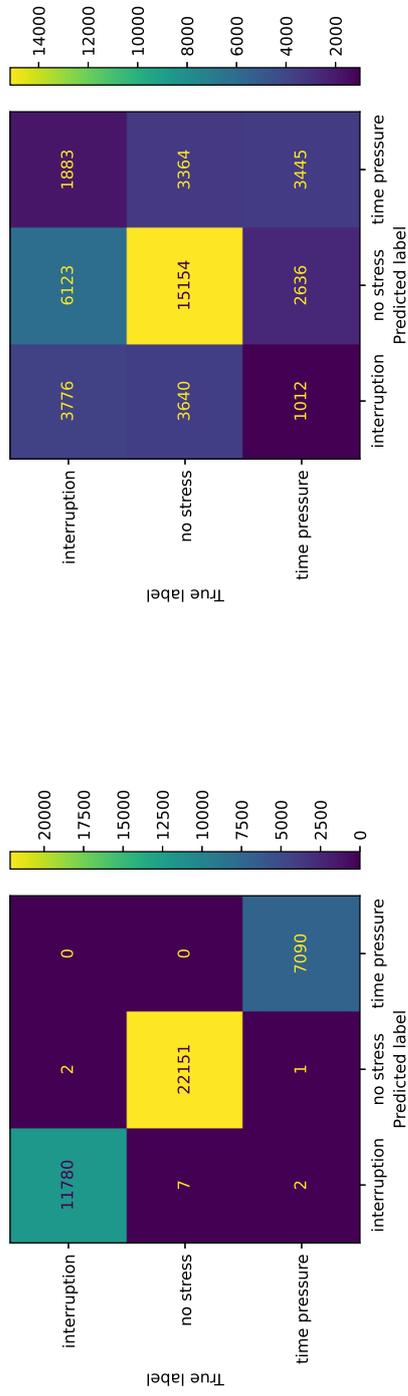
(b) Gráfico de dispersión entre SDRR y MEAN_REL_RR. Baja correlación.

Figura 3.63: Gráficos de dispersión de algunas variables

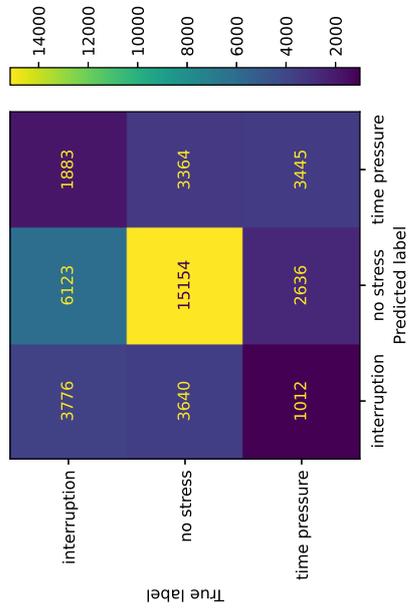
A continuación procedemos a realizar un modelo en base a este conjunto de datos para poder predecir niveles de estrés dados conjuntos de datos a partir de mediciones PPG tomadas durante periodos de tiempo. Optaremos por probar tres algoritmos de clasificación mediante el uso del módulo de Python *Scikit Learn*, ya que son los más usuales y los que más se ajustan al problema a resolver según sus características. Se desarrolla en un listado los resultados de precisión de acierto obtenidos para cada algoritmo:

- **KNeighborsClassifier**. El clasificador basado en el voto de los k-vecinos más cercanos obtiene para este modelo una precisión de acierto de 0.9785
- **GaussianNB**. El clasificador gaussiano Naive Bayes obtiene para este modelo una precisión de acierto de 0.54122
- **RandomForestClassifier**. El clasificador de Random Forest obtiene para este modelo una precisión de acierto de 0.99951

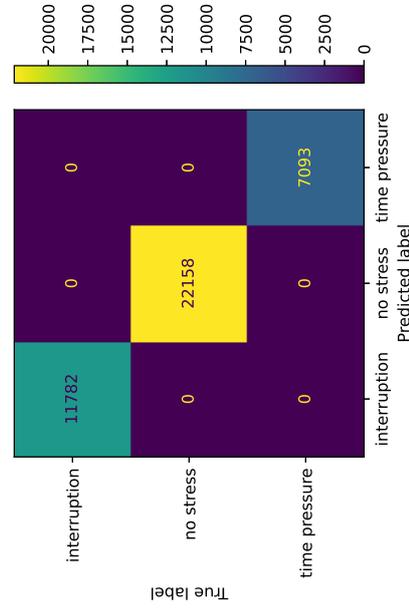
En la figura 3.64 se pueden observar las matrices de confusión, las cuales indican las muestras que se han predicho en función de la condición de estrés. Aquellas muestras que no estén dentro de la diagonal de la matriz, serán falsas predicciones, por lo que se busca reducir estos campos al máximo.



(a) KNeighborsClassifier



(b) GaussianNB



(c) RandomForestClassifier

Figura 3.64: Matrices de confusión de los modelos generados por distintos algoritmos de clasificación.

Atendiendo a estos resultados, optaremos por el modelo mediante el clasificador de KNeighbors para realizar nuestras predicciones de nivel de estrés sobre las variables obtenidas de la señal PPG que recogemos midiendo la actividad de un usuario ya que, aunque aparentemente el clasificador Random Forest obtiene una precisión mayor, cuando se ha experimentado a predecir niveles de estrés, se ha visto que los resultados en cuanto a la precisión de predicción obtenida por el de KNeighbors eran mejores. Estos resultados se constatan con las matrices de confusión de los modelos entrenados mediante los tres algoritmos, ya que vemos que el clasificador de RandomForest acierta todo, lo que se sospecha que es sobreajuste tras los experimentos realizados con este modelo. Por otro lado, vemos en la matriz de confusión del clasificador gaussiano Naive Bayes que se aciertan menor número de muestras nuevas que en el clasificador de k-vecinos más cercanos.

Continuaremos con la implementación en el sistema web del modelo que almacenará los datos en crudo obtenidos por el reloj inteligente. Para ello, se ha optado por que los datos se reciban en formato JSON. Esto se hace así, ya que el reloj enviaría alrededor de 20 peticiones por segundo al servidor para que registrase cada una de las mediciones y esto haría que colapsase. Mediante un JSON, se quieren aglutinar un conjunto de mediciones y cada cierto número de mediciones mandarlas al servidor para que tampoco corramos el riesgo de llenar la memoria del reloj, ya que es algo limitada. En la figura 3.65 se puede visualizar el esquema de la base de datos correspondiente a haber añadido este modelo, junto con otro modelo que se ha creado para almacenar los niveles de estrés tras el procesamiento de los datos en crudo.

- **Tabla *ppg_measures***. Será la encargada de almacenar los datos recibidos por el reloj inteligente del sensor PPG. Se compone de una variable JSON que representa un conjunto de mediciones donde cada una posee su dato obtenido por el sensor PPG en crudo y su estampa de tiempo. Es necesario añadir al modelo de actividad una asociación con este modelo, ya que una actividad tendrá múltiples mediciones del sensor PPG.
- **Tabla *stresses***. Será la encargada de almacenar los niveles de estrés en distintos instantes de tiempo de una actividad tras el procesamiento de las señales en crudo PPG. Una vez se hayan procesado, se borrarán los datos en crudo de la tabla *ppg_measures* a fin de optimizar espacio.

Tras haber realizado las modificaciones al modelo de nuestro sistema web, debemos crear los puntos de entrada en la API para que la aplicación del reloj inteligente pueda mandar al servidor los datos que recabe del sensor PPG. En la tabla 3.12 se muestran todos los puntos de entrada de la API que se tienen en esta iteración.

En cuanto a implementación en el sistema web, solo quedaría terminar el gráfico que se muestra en la página de visualización de detalles de una actividad. Como ya tenemos el modelo *stresses*, el cual almacena los niveles de estrés en cada instante del transcurso de una actividad, podemos obtener todos los registros de este modelo para la actividad

que se esté visualizando. En este proyecto optaremos por usar la librería de JavaScript *Chart.js* (Chartjs, 2022) para la representación del gráfico. Se hará un gráfico de línea en el que se vea la evolución de nivel de estrés a modo de línea del tiempo. En la figura 3.66 se puede ver el resultado de esta implementación.

Por último, se ha realizado un script de Python el cual predice, dado un conjunto de mediciones, el nivel de estrés, para que el sistema web lo ejecute cuando se termine una actividad o cuando el terapeuta desee visualizar el procesamiento de una actividad en transcurso.

En cuanto a la verificación de la funcionalidad, se han llevado a cabo las pruebas de aceptación. La funcionalidad ha sido probada mediante tests unitarios, por lo que se acumulan junto a los de iteraciones anteriores en la tabla de la figura 3.67.

Para lograr lo mostrado en la figura 3.66 es necesario implementar la funcionalidad asociada al envío de los datos recabados por el sensor PPG a la API del sistema web, junto a las demás tareas que se detallan en esta iteración. En la interfaz de conexión con la API de la aplicación del reloj se han implementado las funciones necesarias para parar una actividad y para poder enviar mediciones del sensor PPG.

La implementación de la obtención de datos de los sensores se ha llevado a cabo como un caso de uso más, de modo que se inyecta como una dependencia en el apartado de medición de actividad. De este modo, obtenemos un nuevo diagrama de comunicación entre los tres sistemas donde se aprecia la inclusión de la medición del sensor PPG. Se puede visualizar en la figura 3.68.

En cuanto a diagramas de paquetes, se aprecia que tenemos paquetes nuevos en relación a la medición del sensor PPG. El diagrama de paquetes mostrado en la figura 3.69, correspondiente a la aplicación wearOS, será igual para el resto del desarrollo, por lo que este será la versión final. De igual forma se muestran los diagramas de clases en las figuras 3.73, 3.73, 3.74 y 3.78 correspondientes a cada paquete, de forma que se considera finalizado el desarrollo de la aplicación del reloj.

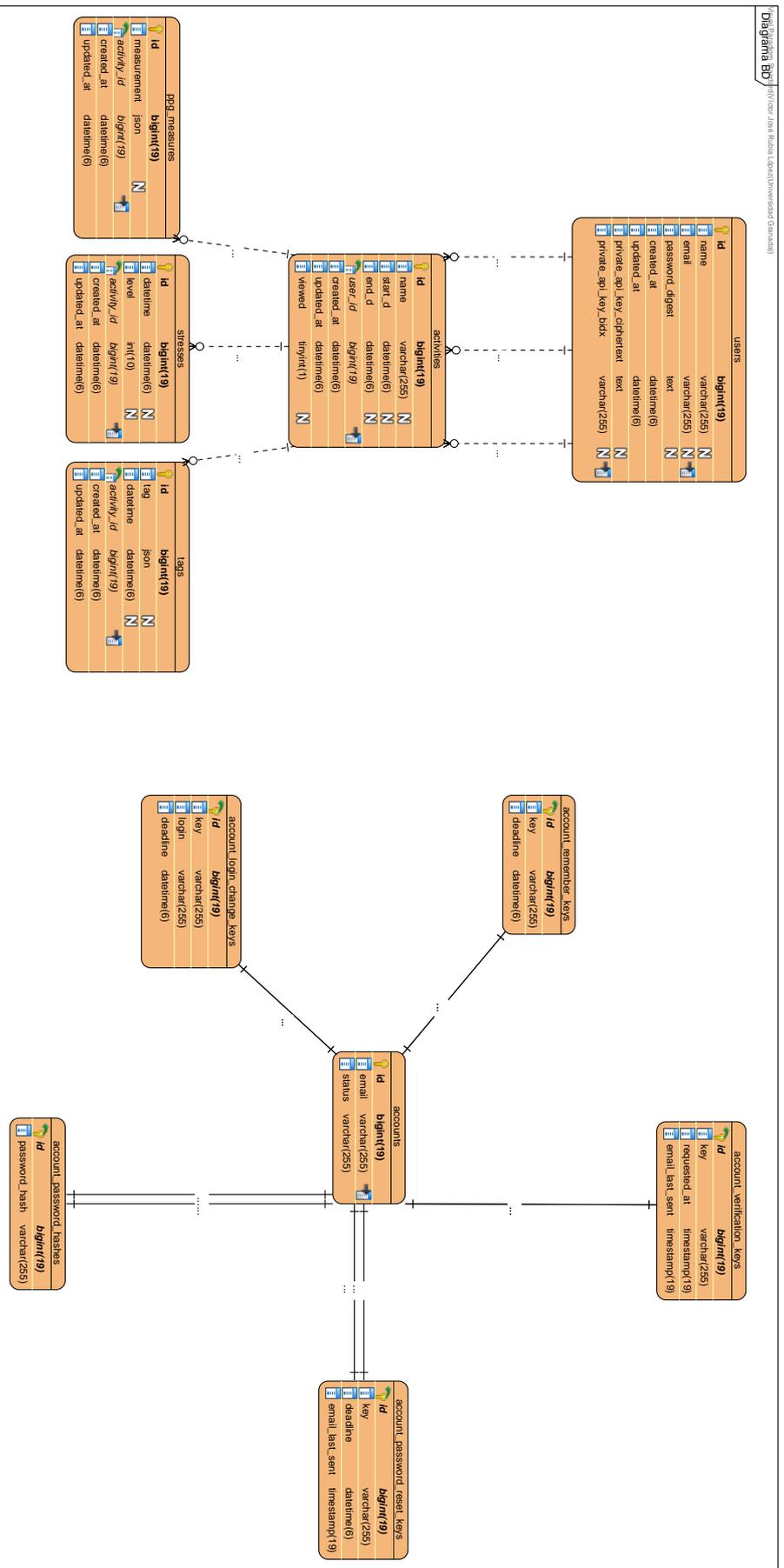


Figura 3.65: Esquema de tablas generadas en la Base de Datos por RoR en la iteración 3

TIPO	DIRECCIÓN	DESCRIPCIÓN
GET	/users	Para obtener los datos del usuario identificado. Se necesita aportar un token Bearer.
GET	/users/get_api_key	Para obtener el token Bearer dados un correo y una contraseña registradas en el sistema.
GET	/users/password_recovery	Para solicitar un correo con el recordatorio de contraseña dado un correo electrónico.
GET	/activities	Obtiene la información de todas las actividades para un usuario dado por un token Bearer.
GET	/activities/id	Obtiene la información de una actividad dada por un ID si es perteneciente al usuario dado por su token Bearer.
POST	/activities	Se crea una nueva actividad en el sistema para el usuario dado por el token Bearer. Se debe aportar el nombre de la actividad, que lo hará pertenecer a una categoría, junto con la fecha de inicio. La fecha de fin es opcional, para poder incluirla más tarde.
PUT	/activities/id	Actualiza el registro de una actividad del sistema. Se usa para finalizar una actividad e incluirle la fecha de fin.
POST	/tags	Se crea un nuevo registro de etiquetado para una actividad en transcurso del sistema. Se debe aportar el conjunto de etiquetas seleccionadas en formato JSON y la estampa de tiempo.
POST	/ppg_measures	Se crea un nuevo registro de medida para una actividad en transcurso del sistema. Se debe aportar un JSON con los datos de medición (dato del sensor en crudo y su estampa de tiempo) y el id de la actividad a la que pertenece.

Cuadro 3.12: Puntos de entrada de la API en la iteración 2



Figura 3.66: Detalle de una actividad en la plataforma web con su gráfico de nivel de estrés.

TestRun	Total	Passed
All Tests - Generated by RubyMine on 26/5/22 9:38	32	32
Api::V1::ActivityControllerTest::Unauthorized		
test_should_not_get_activities	1542ms	passed
test_should_not_update_another's_activity	758ms	passed
Api::V1::TagControllerTest::Authenticated		
test_should_create_tag	761ms	passed
ActivitiesControllerTest		
test_should_destroy_activity	761ms	passed
test_should_update_activity	182ms	passed
DashboardControllerTest		
test_should_get_view_activities	707ms	passed
test_required_authentication	296ms	passed
test_should_get_activity_details	1521ms	passed
test_should_get_activities	230ms	passed
test_should_get_index	279ms	passed
test_should_get_users	238ms	passed
Api::V1::UsersControllerTest::Unauthorized		
test_should_not_get_user	157ms	passed
UsersControllerTest		
test_should_update_user	232ms	passed
test_should_create_user	216ms	passed
test_should_destroy_user	372ms	passed
DashboardTest		
test_searching_an_user	289ms	passed
test_creating_an_user	350ms	passed
test_edit_an_user	307ms	passed
test_remove_an_user	465ms	passed
PpgMeasuresControllerTest		
test_should_destroy_ppg_measure	312ms	passed
test_should_update_ppg_measure	231ms	passed

Figura 3.67: Tests realizados sobre la Aplicación Web RoR en la iteración 3

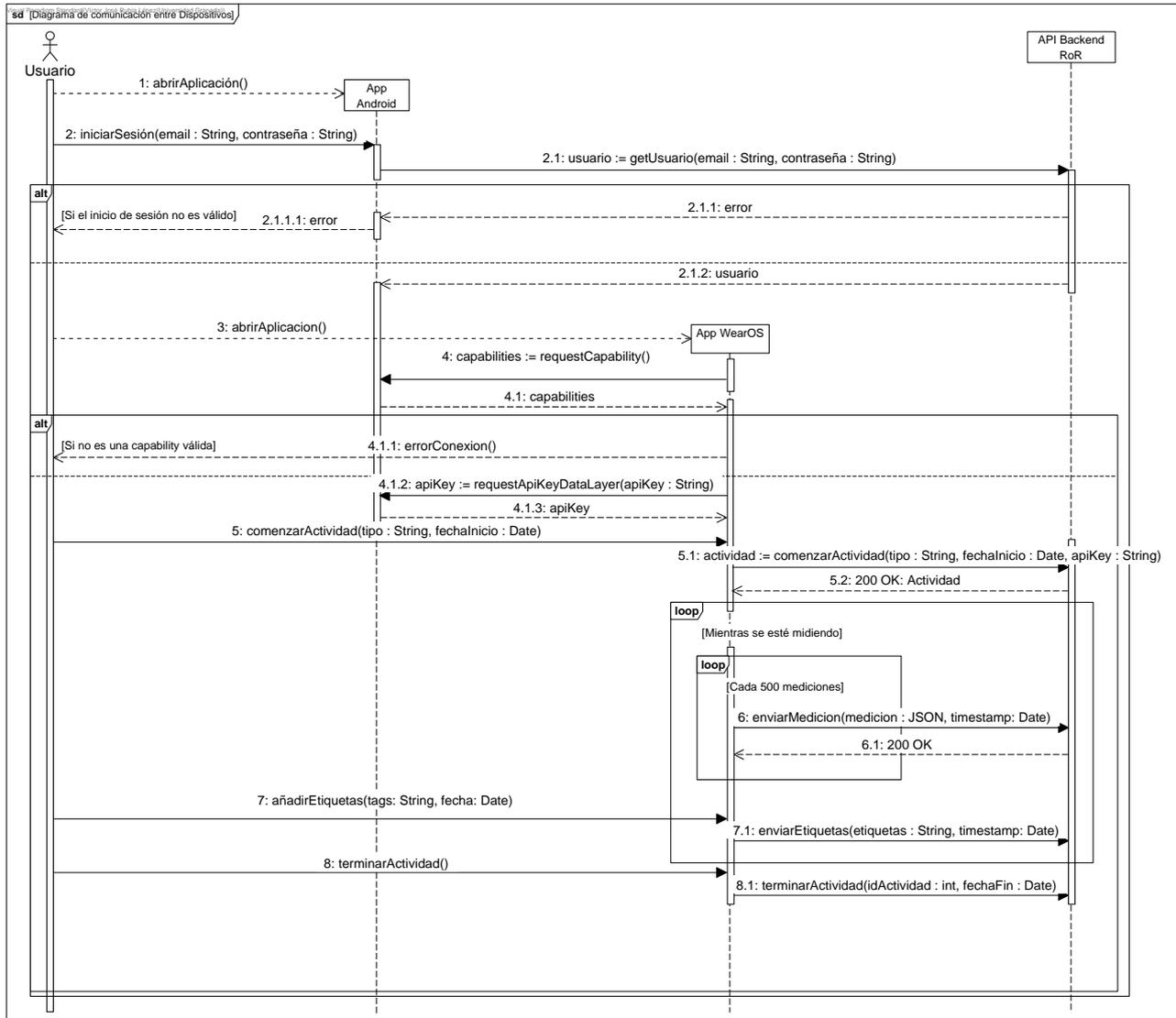


Figura 3.68: Diagrama de comunicación entre sistemas en la iteración 3

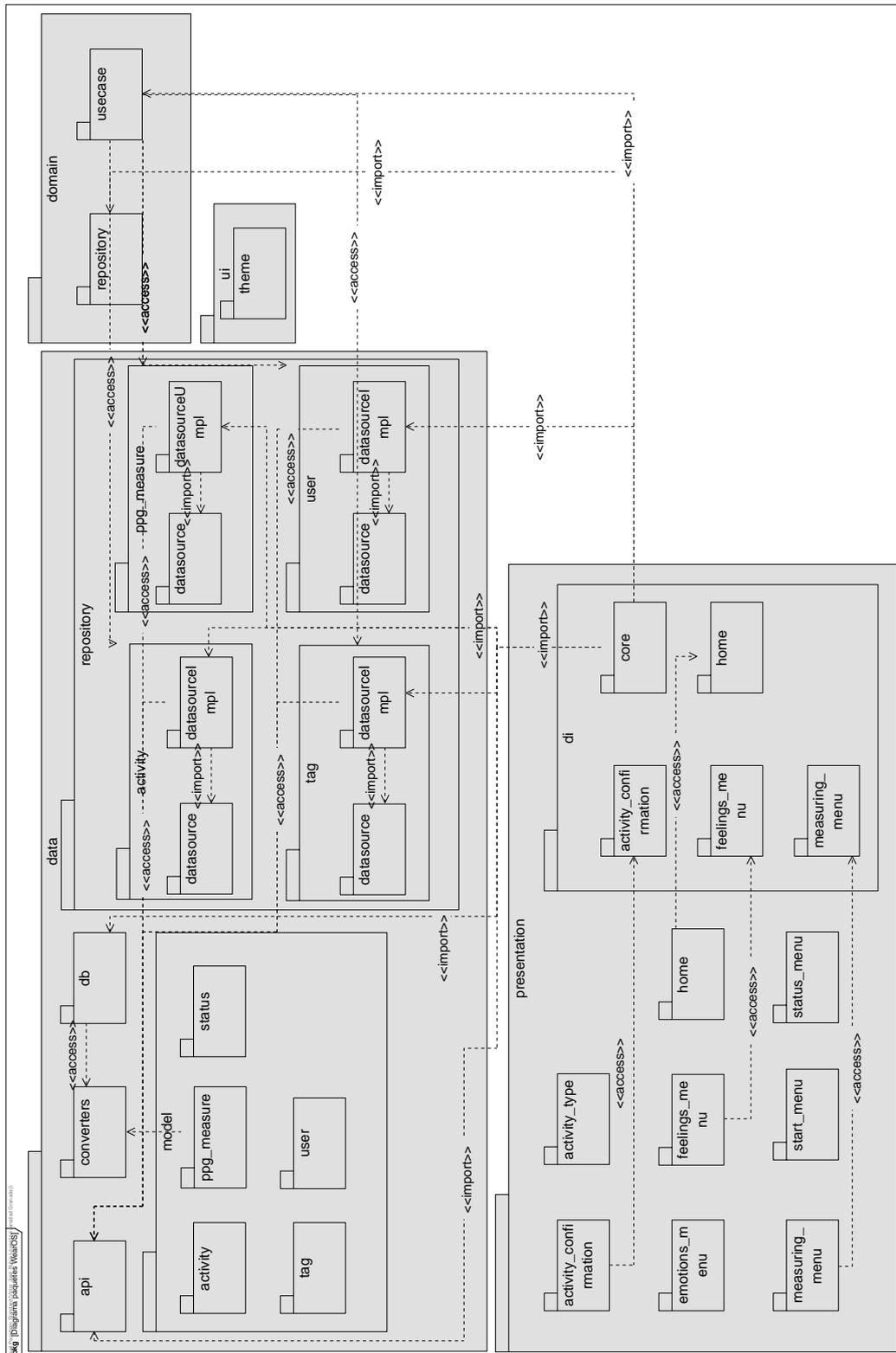


Figura 3.69: Diagrama de paquetes de la aplicación wearOS en la iteración 3

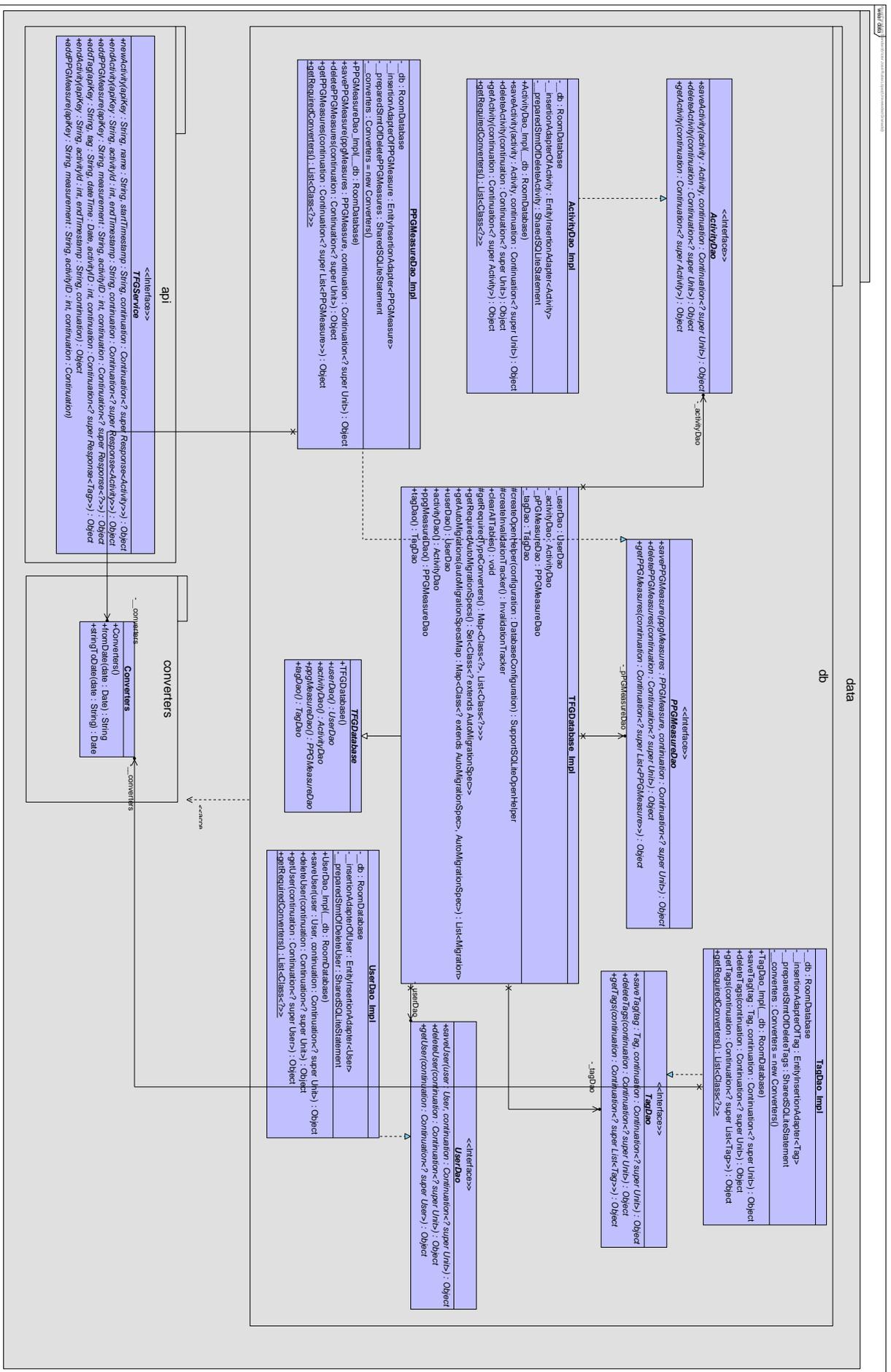


Figura 3.70: Diagrama de clases correspondiente al paquete *data* de la aplicación wearOS en la iteración 3 (Parte 1)

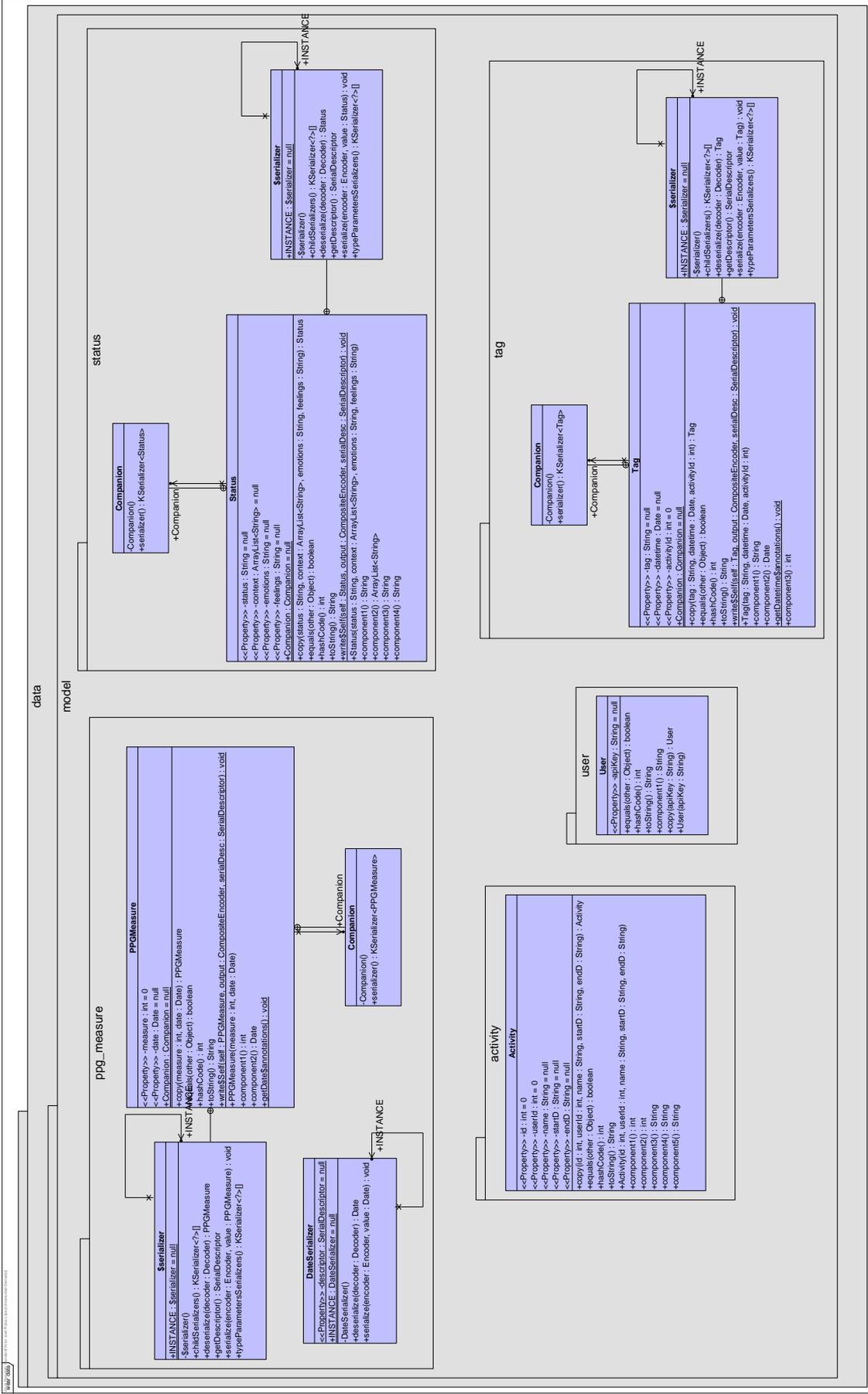


Figura 3.71: Diagrama de clases correspondiente al paquete *data* de la aplicación wearOS en la iteración 3 (Parte 2)

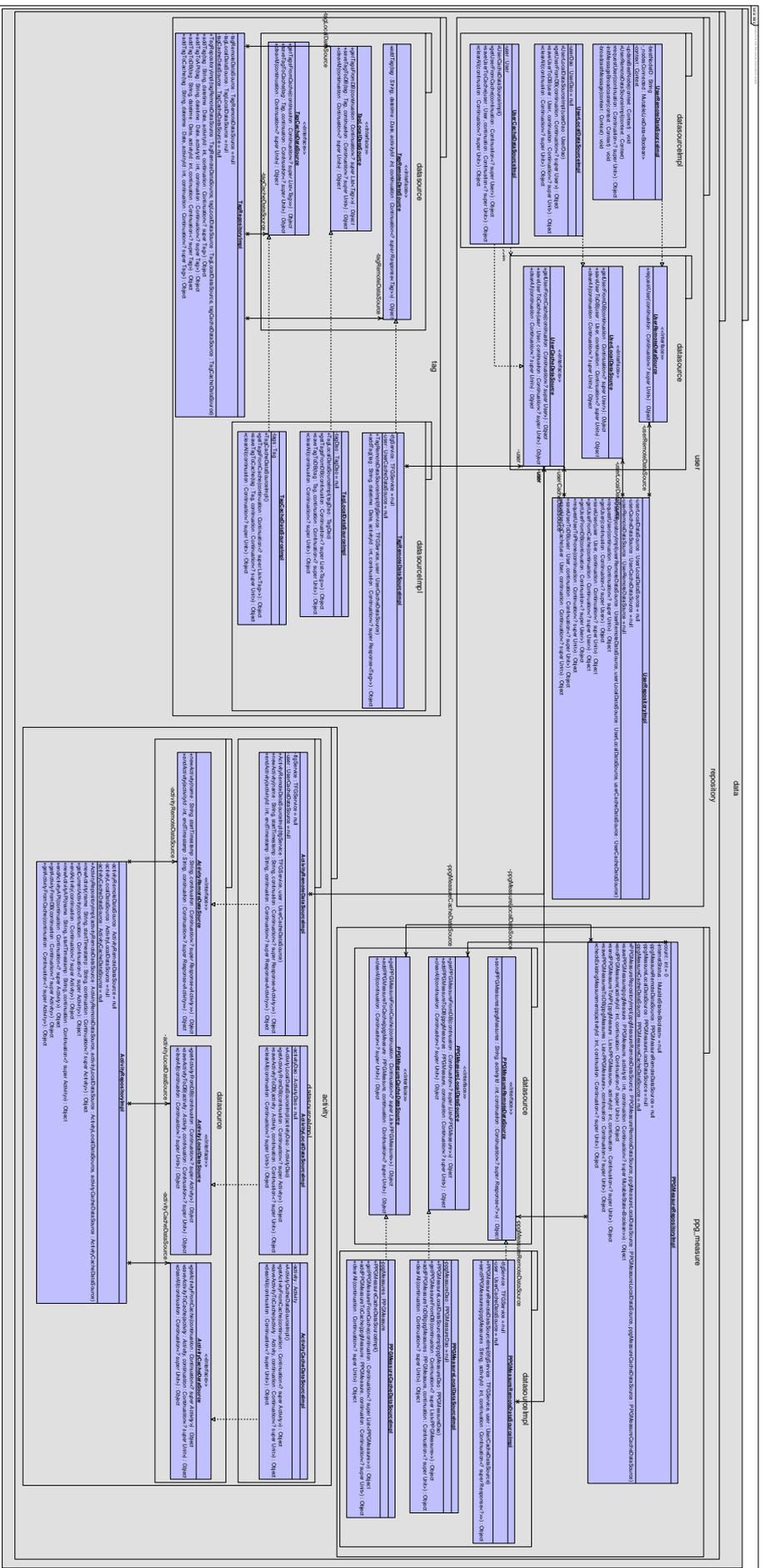


Figura 3.72: Diagrama de clases correspondiente al paquete *data* de la aplicación wearOS en la iteración 3 (Parte 3)

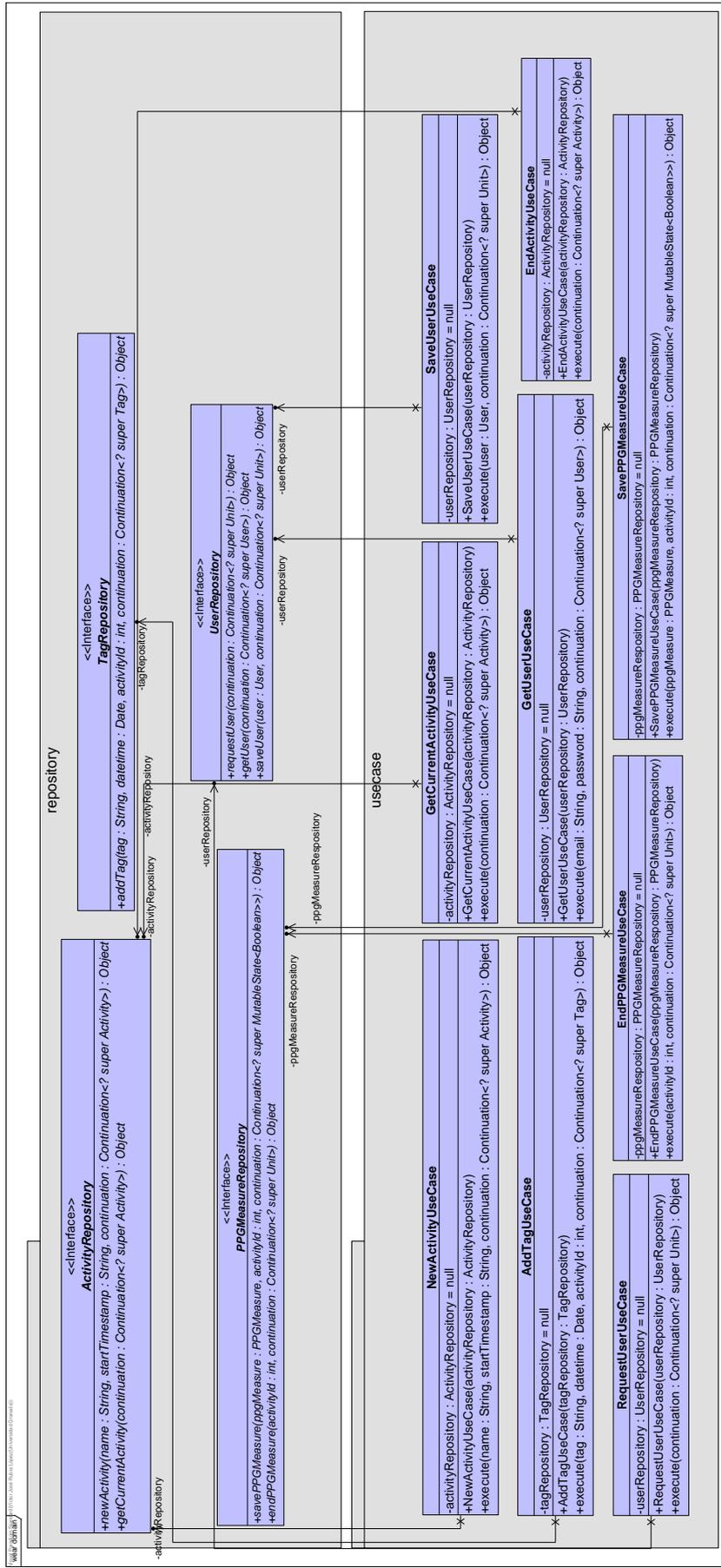


Figura 3.73: Diagrama de clases correspondiente al paquete *domain* de la aplicación wearOS en la iteración 3

Visual Paradigm Standard (Vctor José Rubia López (Universidad Granada))

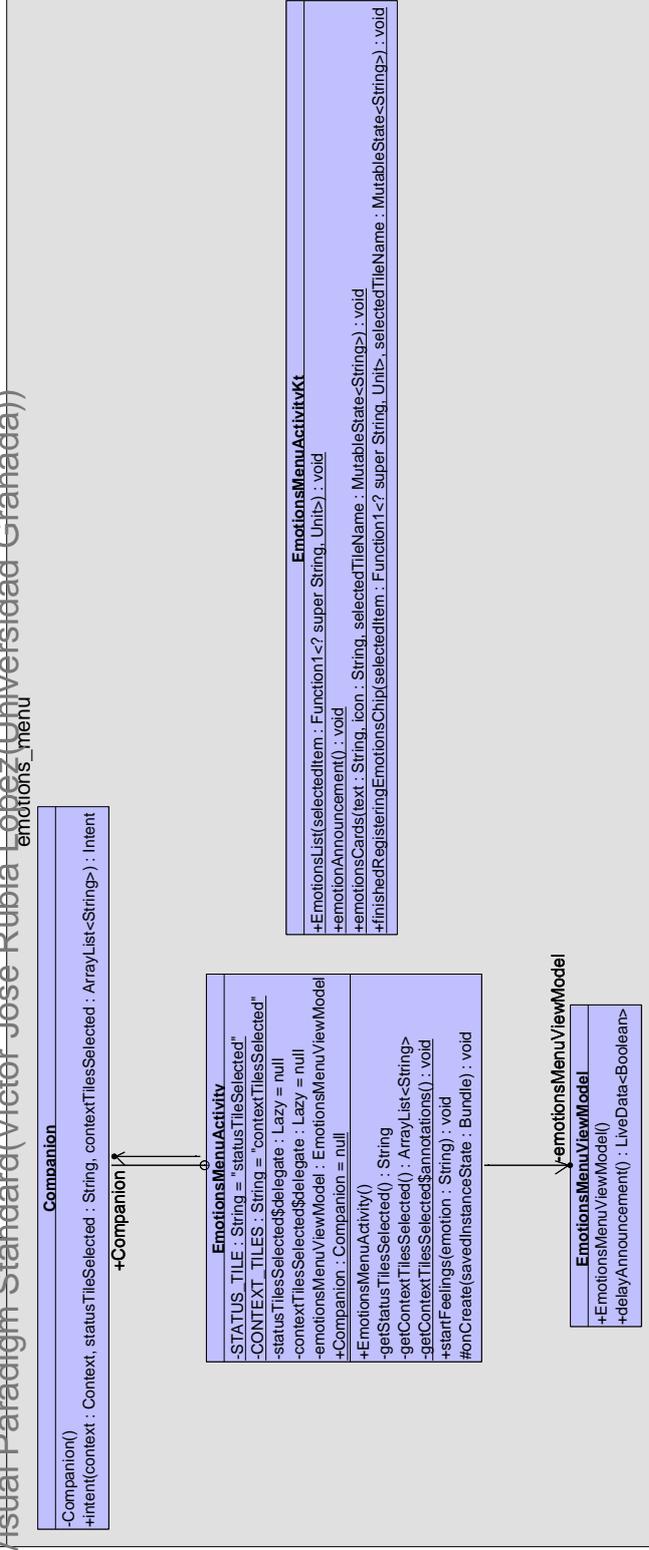


Figura 3.75: Diagrama de clases correspondiente al paquete *presentation* de la aplicación wearOS en la iteración 3 (Parte 2)

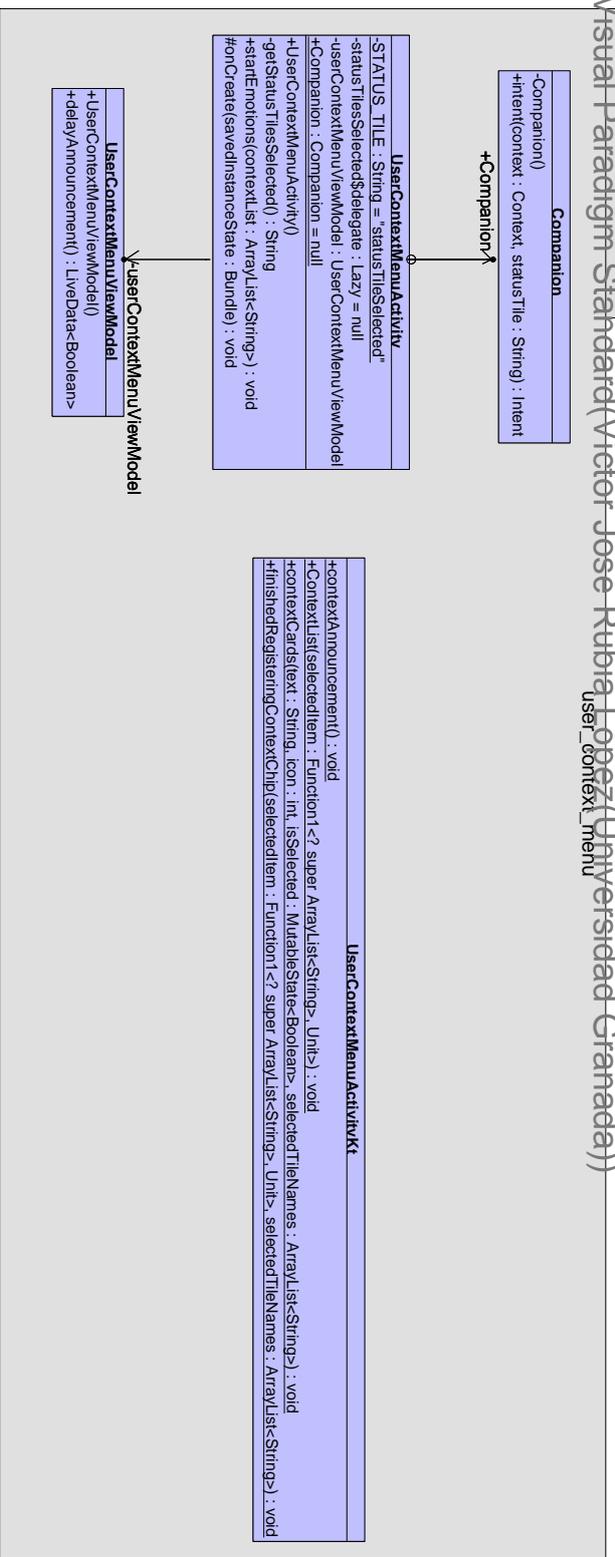


Figura 3.76: Diagrama de clases correspondiente al paquete *presentation* de la aplicación wearOS en la iteración 3 (Parte 3)

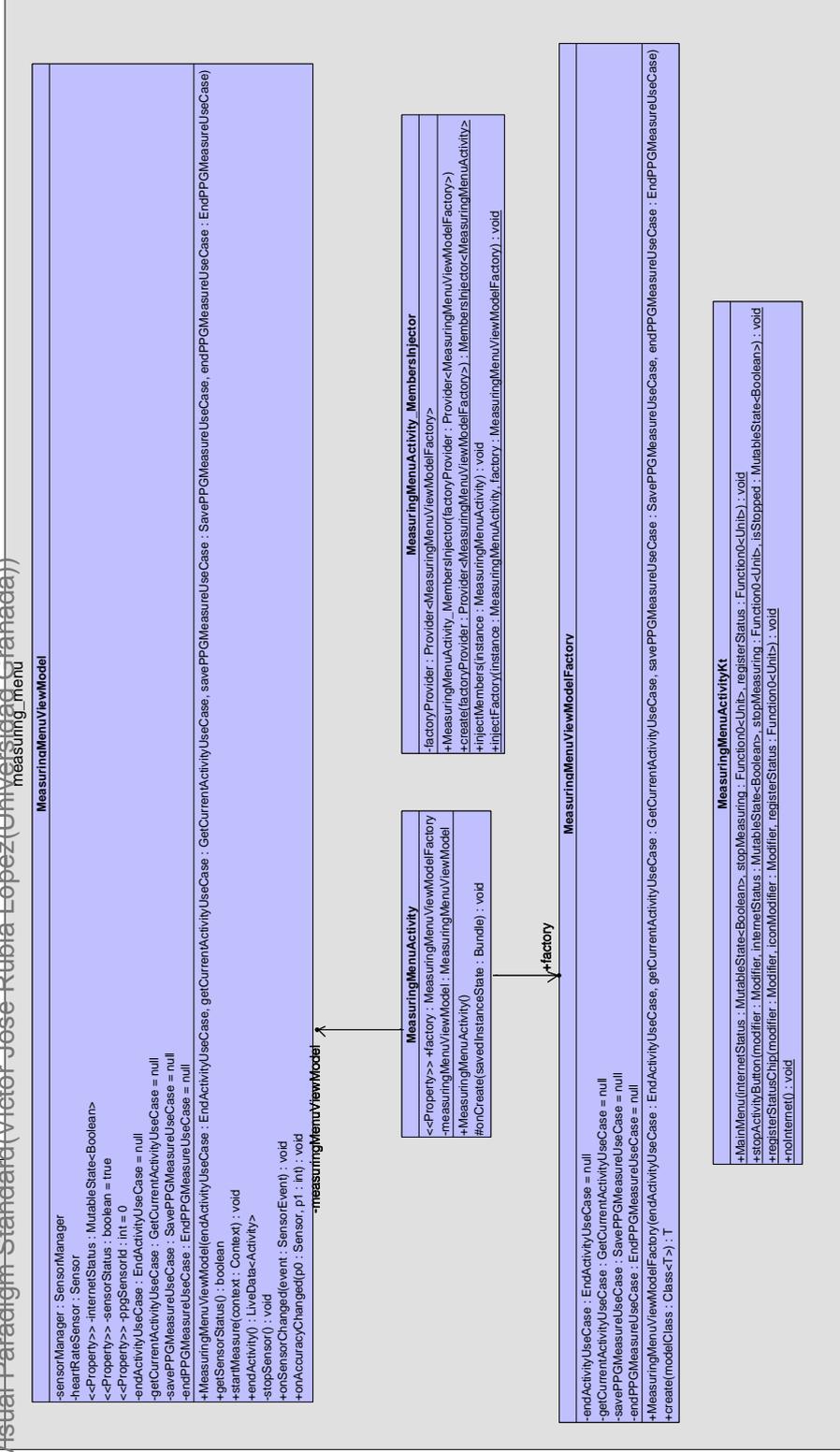


Figura 3.77: Diagrama de clases correspondiente al paquete *presentation* de la aplicación wearOS en la iteración 3 (Parte 4)

En cuanto a la realización de pruebas de aceptación, se muestran en la tabla de la figura 3.79 las realizadas para el reloj inteligente. Además se detalla la evaluación para las pruebas de aceptación de esta iteración en la tabla 3.13.

TestRun	Total	Passed
Unit Tests Wear - Generated by IntelliJ IDEA on 30/4/22 16:22	5	5
TFGServiceTest		
TFGServiceTest.addTag_sentRequest_receivedExpected	1157ms	passed
TFGServiceTest.endActivity_sentRequest_receivedExpected	19ms	passed
TFGServiceTest.newActivity_sentRequest_receivedExpected	13ms	passed
TFGServiceTest.addPPGMeasure_sentRequest_receivedExpected	18ms	passed
ActivityConfirmationViewModelTest		
ActivityConfirmationViewModelTest.newActivity_returnCurrentRecord	6124ms	passed

Figura 3.79: Tests Unitarios y de Componentes de la Aplicación wearOS en la iteración 3

ID	PRUEBA DE ACEPTACIÓN	ESTADO	OBSERVACIONES
47	Se debe haber iniciado una actividad antes de poder pararla.	Correcto	Se ha comprobado de forma manual con la ejecución y el código.
48	Se debe tener conexión a internet para poder parar una actividad.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
49	Una vez que se pare la actividad se envían los últimos datos al servidor y se actualiza la fecha de finalización de la actividad.	Correcto	Se ha comprobado mediante test unitario <i>endActivity_sentRequest_receivedExpected</i> y <i>addPPGMeasure_sentRequest_receivedExpected</i> .
50	Deben existir pantallas que avisen de la tipología de las etiquetas siguientes a escoger.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
51	Los listados de etiquetas deben de estar encabezados por la tipología de etiquetas que se muestran.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.

52	Debe estar una actividad en transcurso y que el usuario pulse sobre registrar etiquetas para que se muestren las etiquetas clasificadas.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
53	Debe estar una actividad en transcurso y que el usuario haya pulsado sobre el botón de registro de etiquetas para poder seleccionar etiquetas.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
54	Si la categoría es de múltiple selección si se pulsa sobre una etiqueta, esta debe seleccionarse, junto a las demás que se pulsen.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
55	Si la categoría es de selección única, si se pulsa sobre una etiqueta y previamente había una seleccionada, esta última pasará a deseleccionarse en favor a la nueva pulsación.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
56	La aplicación no se cierra si se produce una desconexión a internet.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
57	La aplicación es capaz de enviar todos los datos recabados cuando recupere la conexión a internet.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
58	La aplicación sigue midiendo en el transcurso de la actividad.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
59	No se puede terminar la actividad en transcurso.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
60	Cuando se pulsa un botón se muestra una pantalla durante unos segundos explicativa indicando qué ocurre a continuación.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
61	Se debe mostrar al menos durante 3 segundos.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.

62	El correo para el cual se recupera la contraseña, debe estar registrado en el sistema.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
63	Si el correo existe en el sistema, se debe mandar correctamente un correo con la contraseña del usuario.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
64	Cuando se inicie sesión en el teléfono móvil se debe notificar a la aplicación del reloj del cambio de usuario que ha sucedido.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
65	La actividad debe estar iniciada en el sistema.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
66	Se debe mostrar un indicativo cuando la actividad esté sucediendo en directo y no esté terminada.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.

Cuadro 3.13: Resultados de realizar las pruebas de aceptación de la iteración 3

3.5.5. Retrospectiva de la iteración

Para concluir esta iteración, se ha realizado una retrospectiva. En este periodo ha existido una gran carga, ya que no se tenían en cuenta las tareas de investigación para poder limpiar señales del sensor y nos hemos excedido del tiempo que habíamos previsto inicialmente. A pesar de esto, se ha aportado la funcionalidad completa a la aplicación móvil, pudiendo visualizarse en listado siguiente.

- Se envía el token Bearer del usuario mayor obtenido del servidor tras la identificación al reloj inteligente vinculado.
- Se ha desplegado un servicio de escucha en el background para que cuando la aplicación esté cerrada y se quiera usar la aplicación del reloj inteligente, el móvil pueda enviarle el token Bearer del usuario mayor identificado.
- Se ha añadido la funcionalidad de recuperar contraseña mediante el correo electrónico asociado al usuario mayor.

Se ha aportado también la funcionalidad completa al sistema web, la cual se puede conocer en el siguiente listado.

- Se puede registrar e identificar a terapeutas en el sistema.
- Se pueden almacenar en el sistema los usuarios mayores, pudiendo visualizarlos posteriormente.
- Se puede visualizar las actividades registradas por los usuarios mayores junto a sus fechas de finalización y categorizadas por su tipología.
- Si la actividad está en transcurso se indica mediante un icono que está en directo.
- Se puede representar los niveles de estrés de una actividad mediante un gráfico.
- Se muestran en una tabla las etiquetas registradas para una actividad.

Por último, se ha desarrollado la funcionalidad completa de la aplicación del reloj, de modo que el usuario puede comenzar una actividad y el reloj envía la información del sensor PPG al servidor, para que este pueda procesarla, obteniendo las características dividiendo la señal en ventanas de un minuto y, mediante el módulo HeartPy, obtener las características para cada minuto. Con estas características, se formará un conjunto de datos etiquetados con los que podemos predecir los niveles de estrés dado nuestro modelo entrenado con el conjunto de datos SWELL. Hemos logrado medir hasta 6 horas de forma continua antes de que el reloj se quedase sin batería tras hacer un gran esfuerzo en optimizar el rendimiento de la batería. En este estadio, podemos decir que tenemos bastante avanzada la implementación de los sistemas.

3.6. Cuarta iteración

3.6.1. Tarjetas de las Historias de Usuario

Se da comienzo con la extracción de las historias de usuario correspondientes la cuarta iteración y el desarrollo de las tarjetas de historias de usuario.

Identificador: HU-15	Eliminar una actividad	
Descripción: Como terapeuta quiero poder eliminar una actividad de la lista de actividades realizadas por un usuario para dejar de tenerla registrada en el sistema.		
Estimación: 3	Prioridad: 24	Entrega: 4
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ La actividad debe existir en el sistema para eliminarse ▪ Se deben borrar todos los datos asociados con la actividad. ▪ La actividad debe estar terminada para eliminarse 		
Observaciones:		

Identificador: HU-14	Indicar actividades no visualizadas aún.	
Descripción: Como terapeuta quiero que mediante un indicador se notifique de qué actividades no han sido aún visualizadas por mí para poder localizarlas rápidamente.		
Estimación: 3	Prioridad: 25	Entrega: 4
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ La actividad debe no haber sido visualizada antes para mostrar el indicador. ▪ Una vez se haya visualizado, al menos una vez, el indicador debe no aparecer. 		
Observaciones:		

Identificador: HU-2	Correo de confirmación de registro del terapeuta.	
Descripción: Como terapeuta quiero recibir un correo para confirmar el registro.		
Estimación: 3	Prioridad: 26	Entrega: 4
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El correo debe enviarse a la misma dirección con la que se registró el terapeuta. ▪ El correo debe contener un enlace para confirmar el estado de la cuenta. 		
Observaciones:		

Identificador: HU-4	Terapeuta cierra sesión	
Descripción: Como terapeuta quiero poder cerrar sesión en el sistema para poder dejar de usarlo.		
Estimación: 2	Prioridad: 27	Entrega: 4
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Al cerrar sesión no se puede acceder a contenido restringido para usuarios identificados. ▪ Al cerrar sesión se puede identificar nuevamente. ▪ Al cerrar sesión se puede registrar un terapeuta nuevo. 		
Observaciones:		

Identificador: HU-7	Editar usuarios mayores	
Descripción: Como terapeuta quiero poder editar a los usuarios registrados en el sistema para cambiar sus datos en caso de que exista algún cambio o problema.		
Estimación: 3	Prioridad: 28	Entrega: 4
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El terapeuta debe estar identificado. ▪ Para el campo de correo electrónico se debe introducir un correo válido. ▪ El campo nombre/pseudónimo no puede estar vacío. ▪ El campo de contraseña no puede estar vacío. ▪ La clave de la API ha de poder cambiarse. ▪ Cuando se introduzcan nuevos valores para cualquier campo, este cambio se debe hacer efectivo en el sistema. 		
Observaciones:		

Identificador: HU-9	Eliminar usuarios mayores	
Descripción: Como terapeuta quiero poder eliminar a los usuarios registrados en el sistema para que no puedan usar su cuenta más.		
Estimación: 3	Prioridad: 29	Entrega: 4
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ El usuario a eliminar debe existir en el sistema. ▪ Se deben borrar todas las actividades asociadas al usuario cuando éste se elimina. ▪ Ya no se debe permitir iniciar sesión con este usuario eliminado a través de la API en la aplicación móvil. 		
Observaciones:		

Identificador: HU-30	Notificar desconexión	
Descripción: Como usuario quiero que la aplicación del reloj me indique cuándo se ha quedado sin conexión para poder solventarlo lo antes posible sin tener que perder información..		
Estimación: 3	Prioridad: 30	Entrega: 4
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ Se debe mostrar un icono en la parte superior cuando se quede sin conexión a internet. ▪ No debe cerrarse la aplicación cuando se pierda la conexión. 		
Observaciones:		

Identificador: HU-28	Aplicación móvil accesible	
Descripción: Como usuario quiero que la aplicación móvil sea accesible para personas mayores o con alguna discapacidad.		
Estimación: 3	Prioridad: 31	Entrega: 4
Pruebas de aceptación:		
<ul style="list-style-type: none"> ▪ La aplicación móvil debe poder ser descrita por el asistente TalkBack. ▪ La aplicación móvil cumple las guías de accesibilidad usando TalkBack. 		
Observaciones:		

Identificador: HU-29	Aplicación reloj accesible	
Descripción: Como usuario quiero que la aplicación del reloj sea accesible para personas mayores o con alguna discapacidad.		
Estimación: 3	Prioridad: 32	Entrega: 4
Pruebas de aceptación: <ul style="list-style-type: none"> ▪ La aplicación del reloj debe ser sencilla de usar. ▪ Los iconos deben ser entendibles y universales. ▪ Los espacios y márgenes que se dejen deben ser suficientes. ▪ No deben existir pantallas que aparezcan y desaparezcan muy rápido. ▪ Se deben cumplir las guías de accesibilidad básicas 		
Observaciones:		

3.6.2. Descomposición de las H.U. en tareas de desarrollo

Tras escribir las pruebas de aceptación de las historias de usuario que se abarcan en la cuarta iteración, se descompone cada una de ellas en tareas de desarrollo.

HU-15	Como terapeuta quiero poder eliminar una actividad de la lista de actividades realizadas por un usuario para dejar de tenerla registrada en el sistema. .	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-78	Añadir al boceto de listado de actividades un botón para poder eliminar una actividad terminada.	0.5
T-79	Implementar el boceto asociado a añadir un botón para eliminar una actividad.	2
T-80	Realizar pruebas de aceptación asociadas a la eliminación de actividades terminadas.	0.5
Observaciones:		

HU-14	Como terapeuta quiero que mediante un indicador se notifique de qué actividades no han sido aún visualizadas por mí para poder localizarlas rápidamente.	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-81	Añadir al boceto de listado de actividades un indicador que notifique qué actividades no han sido aún vistas por el terapeuta.	0.5
T-82	Implementar boceto de indicador de actividades no visualizadas por el terapeuta.	2
T-83	Realizar pruebas de aceptación asociadas al indicador de actividades no visualizadas.	0.5
Observaciones:		

HU-2	Como terapeuta quiero recibir un correo para confirmar el registro.	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-84	Diseñar e implementar el correo para confirmación de registro de terapeuta	1
T-85	Implementar el uso del diseño personalizado de correo para confirmación de registro de terapeuta.	1
T-86	Realizar pruebas de aceptación asociadas al envío de correo cuando se registra un terapeuta.	1
Observaciones:		

HU-4	Como terapeuta quiero poder cerrar sesión en el sistema para poder dejar de usarlo.	2 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-87	Realizar bocetado de la pantalla de cierre de sesión de un terapeuta.	0.5
T-88	Implementar el boceto de la pantalla de cierre de sesión de un terapeuta.	1
T-89	Realizar pruebas de aceptación asociadas al cierre de sesión de un terapeuta.	0.5
Observaciones:		

HU-7	Como terapeuta quiero poder editar a los usuarios registrados en el sistema para cambiar sus datos en caso de que exista algún cambio o problema.	3 PH
------	---	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-90	Realizar boceto donde se muestra un botón para editar un usuario mayor en el sistema.	0.5
T-91	Implementar el boceto asociado a editar un usuario mayor del sistema	2
T-92	Realizar las pruebas de aceptación asociadas con editar un usuario mayor del sistema	0.5

Observaciones:

HU-9	Como terapeuta quiero poder eliminar a los usuarios registrados en el sistema para que no puedan usar su cuenta más.	3 PH
------	--	------

Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-93	Añadir al boceto de la sección de gestión de usuarios donde se muestre, con el rol de terapeuta, un botón para eliminar a un usuario mayor registrado en el sistema.	0.5
T-94	Implementar el boceto asociado a la eliminación de un usuario mayor registrado.	1.5
T-95	Realizar las pruebas de aceptación asociadas a la eliminación de un usuario mayor registrado.	1

Observaciones:

HU-30	Como usuario quiero que la aplicación del reloj me indique cuándo se ha quedado sin conexión para poder solventarlo lo antes posible sin tener que perder información.	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-96	Realizar boceto donde se indique mediante un icono al usuario que ha perdido la conexión, pero que sigue realizando mediciones.	0.5
T-97	Implementar el boceto asociado a la pérdida de conexión.	2
T-98	Realizar las pruebas de aceptación asociadas a pérdida de conexión y medición continuada.	0.5
Observaciones:		

HU-28	Como usuario quiero que la aplicación móvil sea accesible para personas mayores o con alguna discapacidad.	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-99	Revisar bocetos para asegurar la accesibilidad de la aplicación para las personas mayores	0.5
T-100	Implementar la compatibilidad de la aplicación móvil con el asistente TalkBack y revisar implementaciones de aquellos bocetos que hayan cambiado.	2
T-101	Realizar las pruebas de aceptación asociadas a la accesibilidad con personas mayores	0.5
Observaciones:		

HU-29	Como usuario quiero que la aplicación del reloj sea accesible para personas mayores o con alguna discapacidad.	3 PH
Identificador	Título de la tarea de desarrollo	Estimación (días ideales)
T-102	Revisar bocetos para asegurar la accesibilidad de la aplicación del reloj para las personas mayores	0.5
T-103	Implementar la diferencias de la aplicación del reloj con aquellos bocetos que hayan cambiado.	2
T-104	Realizar las pruebas de aceptación asociadas a la accesibilidad de la aplicación del reloj	0.5
Observaciones:		

3.6.3. Planificación temporal de la cuarta iteración

En la figura 3.80 se puede visualizar un diagrama de Gantt para la planificación detallada de la cuarta iteración.

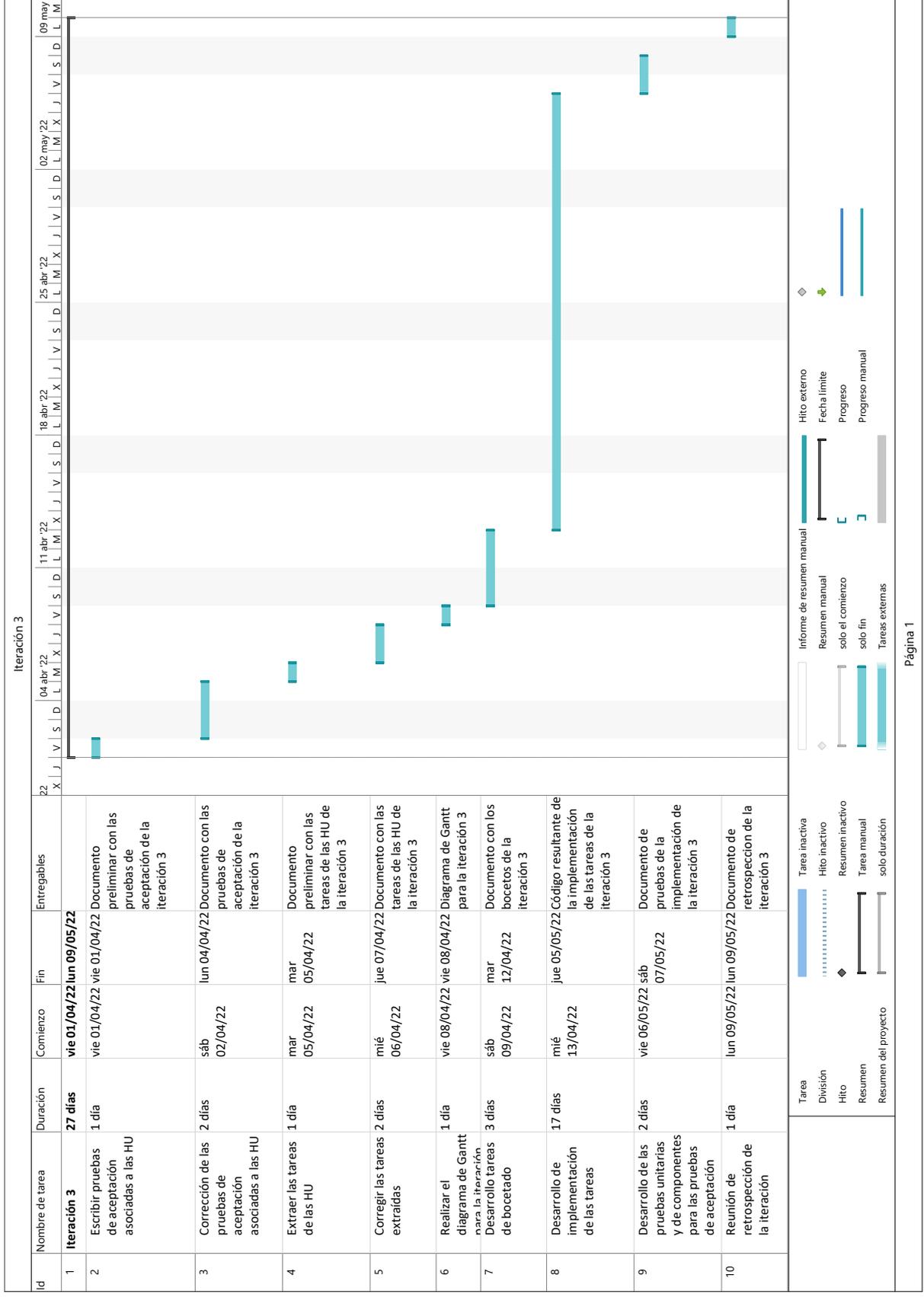


Figura 3.80: Diagrama de Gantt para la planificación de las tareas de la iteración 4

3.6.4. Realización de las tareas de desarrollo

Comenzaremos la realización de las tareas de la cuarta iteración añadiendo las modificaciones que se nos piden a los bocetos que tenemos ya realizados.

Primeramente, editaremos los bocetos del sistema web a los cuales se han añadido los de cerrar sesión como terapeuta, la página de recordar contraseña y la página para enviar un nuevo correo de verificación de registro de terapeuta. Se han modificado los bocetos ya existentes para la página de gestión de usuarios y de actividades, con el fin de añadir botones para editar y eliminar usuarios registrados. Estos pueden visualizarse en la figura 3.81, junto con la navegabilidad final del sistema web en la figura 3.82.

En este último boceto realizado sobre la edición de un usuario, puesto que para editar es necesario previamente visualizar los datos de un usuario, se ha optado por el diseño que se muestra en la figura 3.81. Se provee un formulario con los datos requeridos en el registro, estando la contraseña oculta y su visualización se mostraría pulsando sobre un botón. En cuanto a aspectos técnicos, se puede visualizar la clave privada con la que, en los dispositivos del usuario, se realizan las peticiones al servidor. Se puede solicitar una nueva en caso de que existiese algún problema o brecha de seguridad en el sistema.

Tras el diseño de los bocetos del sistema web, pasamos a comprobar los cambios necesarios que hay que realizar para el bocetado de la aplicación móvil. Estos cambios tienen que ver con la comprobación del cumplimiento con las características de una interfaz accesible.

Para el cuidado de la accesibilidad en la aplicación móvil se ha seguido la guía de accesibilidad de Material Design (Google, 2022a), junto con la habilitación del uso de TalkBack, siguiendo una estructura de elementos de izquierda a derecha y de arriba hacia abajo, describiendo cada elemento de forma adecuada.

En cuanto a la aplicación del reloj, debemos añadir un boceto para la funcionalidad que indique cómo se debe mostrar una pantalla cuando el usuario pierde la conexión a internet. El diseño se ha llevado a cabo siguiendo las recomendaciones de la guía de estilo de Google. En la figura 3.83 se muestra el resultado del boceto diseñado.

De igual forma, se corrobora que la aplicación para el reloj cumple estándares de accesibilidad mediante la comprobación de que las pantallas son simples, no desaparecen elementos repentinamente, se indica cada acción a realizar y los botones y elementos son suficientemente grandes para que puedan ser pulsados sin mayor problema, atendiendo a que es una aplicación de reloj inteligente y por esto mismo ya se tienen muchas limitaciones de accesibilidad.

Tras realizar las tareas de bocetado para esta iteración procedemos a llevar a cabo su implementación comenzando por el sistema web.

Se han podido llevar a cabo las implementaciones de forma satisfactoria. Para lograrlo, se ha añadido la funcionalidad de poder eliminar actividades ya terminadas para un usuario, de modo que podamos optimizar el espacio o borrar aquellas que no nos

interesen ya. La funcionalidad de borrado de un registro en Ruby On Rails viene ya implementada de serie, por lo que es bastante sencillo posicionar un botón de borrado de actividad con esa funcionalidad en cada una de las tarjetas que aparecen en la página. Por otro lado, se ha añadido la posibilidad de editar y borrar a los usuarios mayores que tenemos registrados en el sistema. Para el formulario de edición se ha seguido la misma aproximación que para registrarlos, mediante un modal que aparece sobre la página. En él, se encuentra un formulario con los campos rellenos con los datos del usuario a editar, estando la contraseña oculta mediante un botón. Se puede visualizar en la figura 3.84.

Por otro lado, se ha implementado la funcionalidad para que se muestre un indicador cuando la actividad no haya sido previamente visualizada por el terapeuta. Para ello, ha sido necesario incluir una nueva variable en el modelo actividad que sea capaz de almacenar si una actividad ha sido previamente visualizada o no. En el momento en el que el terapeuta entra a ver dicha actividad, el valor de esta columna cambiará, de modo que no salga el indicador más. Esta se puede ver en la figura 3.85.

En cuanto a los bocetos y funcionalidad para el cierre de sesión, recuperación de contraseña y reenvío de correo de activación, como son muy simples y nos hemos valido de la funcionalidad que aporta la librería RodAuth no ha existido ningún problema en implementarlos.

Se ha llevado a cabo la implementación de los envíos de correo electrónico para satisfacer las tareas de envío de correo. En las figuras 3.86a, 3.86b y 3.86c se puede visualizar su resultado, donde se aprecia una imagen corporativa sólida.

En cuanto a las pruebas de aceptación para evaluar la corrección de las tareas desarrolladas en esta iteración, se muestran en la tabla 3.14.

ID	PRUEBA DE ACEPTACIÓN	ESTADO	OBSERVACIONES
67	La actividad debe existir en el sistema para eliminarse	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
68	Se deben borrar todos los datos asociados con la actividad.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
69	La actividad debe estar terminada para eliminarse	Correcto	Se ha comprobado de forma manual con el código y la ejecución, solo se muestra el botón de eliminar para aquellas actividades finalizadas.
70	La actividad debe no haber sido visualizada antes para mostrar el indicador.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.

71	Una vez se haya visualizado, al menos una vez, el indicador debe no aparecer.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
72	El correo debe enviarse a la misma dirección con la que se registró el terapeuta.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
73	El correo debe contener un enlace para confirmar el estado de la cuenta.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
74	Al cerrar sesión no se puede acceder a contenido restringido para usuarios identificados.	Correcto	Se ha comprobado mediante test unitario <i>test_required_authentication</i> .
75	Al cerrar sesión se puede identificar nuevamente.	Correcto	Se ha comprobado mediante test unitario <i>test_login_in_and_login_out</i> .
76	Al cerrar sesión se puede registrar un terapeuta nuevo.	Correcto	Se ha comprobado mediante test unitario <i>test_creating_an_account</i> .
77	El terapeuta debe estar identificado para editar un usuario mayor.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
78	Para el campo de correo electrónico se debe introducir un correo válido.	Correcto	Se han aplicado unos filtros en el código para comprobar la gramática y la estructura de este campo.
79	El campo nombre/pseudónimo no puede estar vacío.	Correcto	Se han aplicado unos filtros en el código para comprobar la gramática y la estructura de este campo.
80	El campo de contraseña no puede estar vacío.	Correcto	Se han aplicado unos filtros en el código para comprobar la gramática y la estructura de este campo.
81	La clave de la API ha de poder cambiarse.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
82	El usuario debe existir en el sistema.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.

83	Se deben borrar todas las actividades asociadas al usuario.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
84	Ya no se debe permitir iniciar sesión con este usuario a través de la API en la aplicación móvil.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
85	Se debe mostrar un icono en la parte superior cuando se quede sin conexión a internet.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
86	No debe cerrarse la aplicación.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
87	La aplicación móvil debe poder ser descrita por el asistente TalkBack.	Correcto	Se ha comprobado de forma manual con el código y la ejecución.
88	La aplicación del reloj debe ser sencilla de usar.	Correcto	Se ha comprobado de forma manual. Tiene una lógica de navegación sencilla y repetitiva.
89	Los iconos deben ser entendibles y universales.	Correcto	Se ha comprobado de forma manual. Se ha usado un único repositorio de iconos para la aplicación.
90	Los espacios y márgenes que se dejen deben ser suficientes.	Correcto	Se ha comprobado de forma manual.
91	No deben existir pantallas que aparezcan y desaparezcan muy rápido.	Correcto	Se ha comprobado de forma manual. Desaparecen tras varios segundos.

Cuadro 3.14: Resultados de realizar las pruebas de aceptación de la iteración 4

3.6.5. Retrospectiva de la iteración

Tras el retraso de la iteración anterior, se dejó a esta con un menor tiempo del que le correspondía al comienzo del proyecto. Esto hizo al desarrollador pensar que iba a tener poco tiempo para la implementación de las tareas asociadas a esta iteración. Sin embargo, siendo tareas menos prioritarias que el resto, tenían una complejidad de implementación mucho menor, lo que ha facilitado mucho el acabar a tiempo esta iteración.

En cuanto al sistema web, se han añadido las siguientes funcionalidades:

- Se pueden eliminar usuarios mayores almacenados en el sistema.
- Se pueden editar usuarios mayores almacenados en el sistema.
- Se pueden eliminar actividades realizadas por usuarios mayores almacenadas en el sistema.
- Se notifica al usuario mayor por correo electrónico de la edición y creación de su perfil en el sistema.
- El terapeuta puede cerrar sesión en el sistema.
- Se envían correos electrónicos de confirmación cuando se registra un terapeuta en el sistema.
- Se añade un indicador a aquellas actividades que aún no han sido visualizadas por el terapeuta para facilitar su distinción.

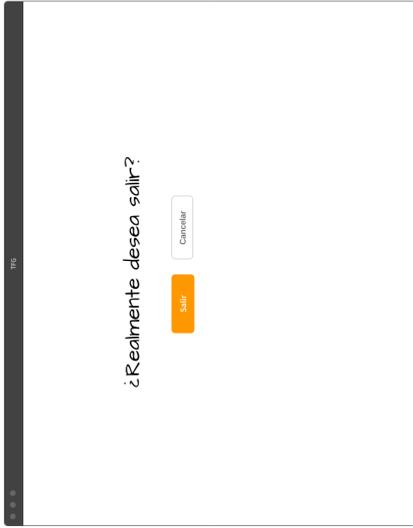
En cuanto a la aplicación del reloj, se aporta las siguientes características:

- Se notifica al usuario mayor mediante un icono en la parte superior de la pantalla de la pérdida de conexión a internet.
- Se ha cuidado la accesibilidad en las distintas pantallas de la aplicación.

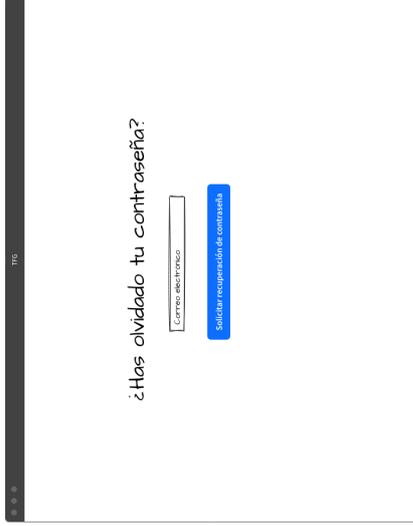
En cuanto a la aplicación del móvil, se tienen las siguientes novedades:

- Se ha cuidado la accesibilidad en los elementos de la aplicación.
- Se puede usar la aplicación mediante TalkBack.

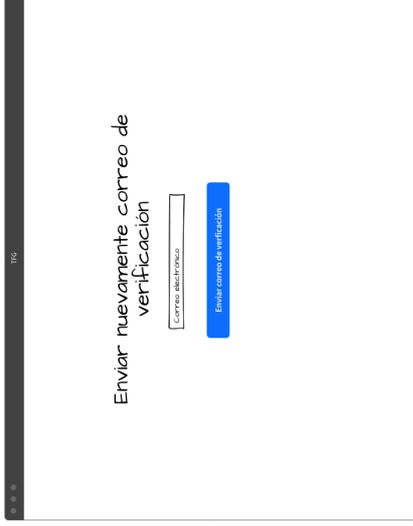
Se concluye el desarrollo pues, con la aceptación por parte de las tutoras de no seguir incluyendo funcionalidades y pasar ahora la experimentación con lo desarrollado, que se expone en el siguiente capítulo.



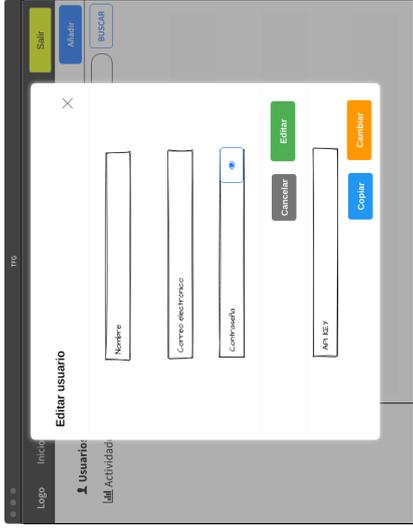
Mo-4. Página salir



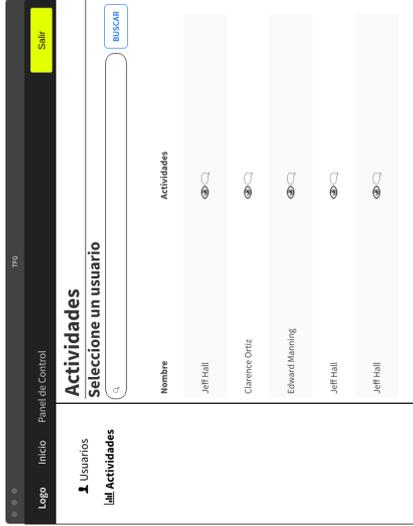
Mo-5. Página recuperar contraseña



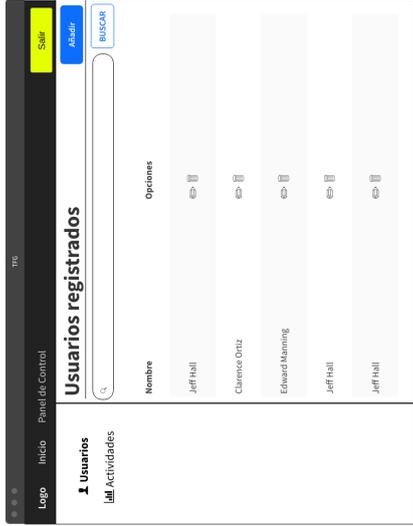
Mo-6. Página reenviar verificación



Mo-8.2. Modal para editar un usuario dentro de la sección de gestión de usuarios.



Mo-7. Página principal del panel de control (actividades)



Mo-8. Página del panel de control para gestión de usuarios

Figura 3.81: Nuevos bocetos del sistema web en la iteración 4

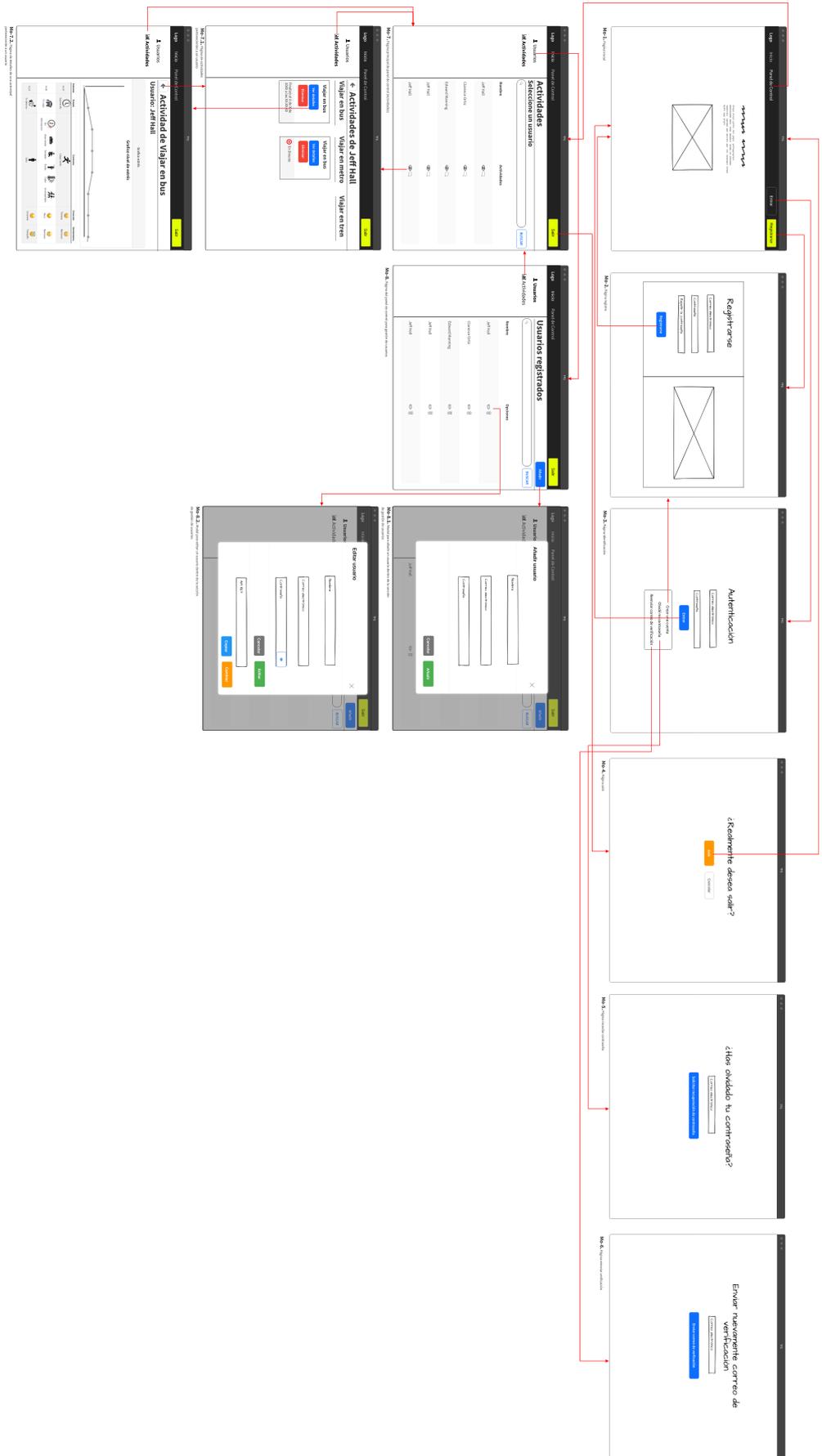


Figura 3.82: Navegación del sistema web en la iteración 4

UI-5
Menú de medición

(a) Con conexión a internet

UI-5-BIS
Menú de medición
No internet

(b) Sin conexión a internet

Figura 3.83: Bocetado de alta resolución del indicador de pérdida de conexión.

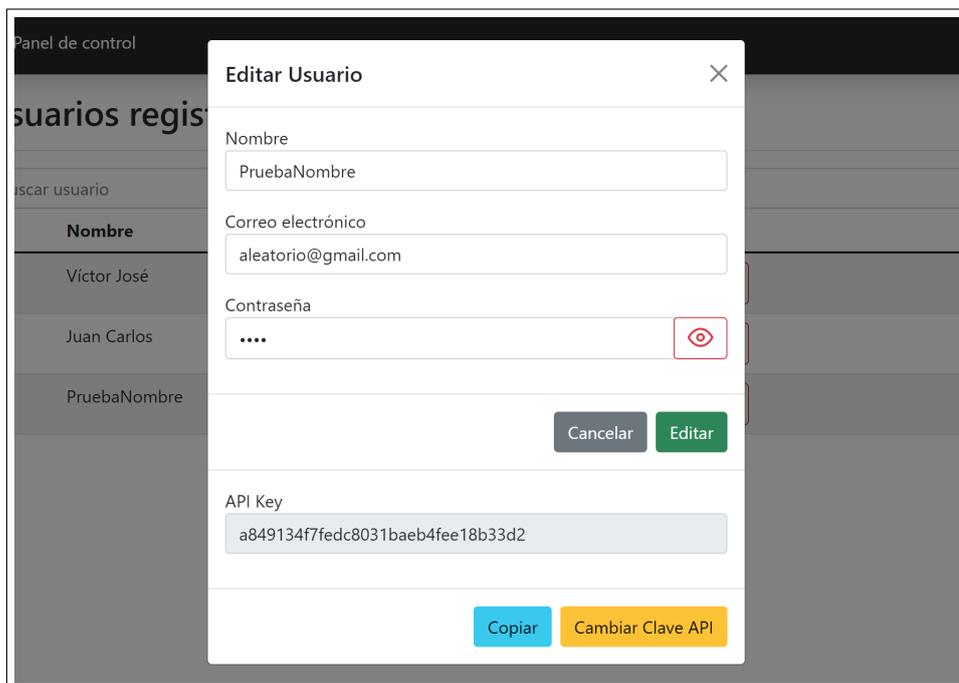


Figura 3.84: Diálogo modal para editar y visualizar un usuario mayor en la plataforma

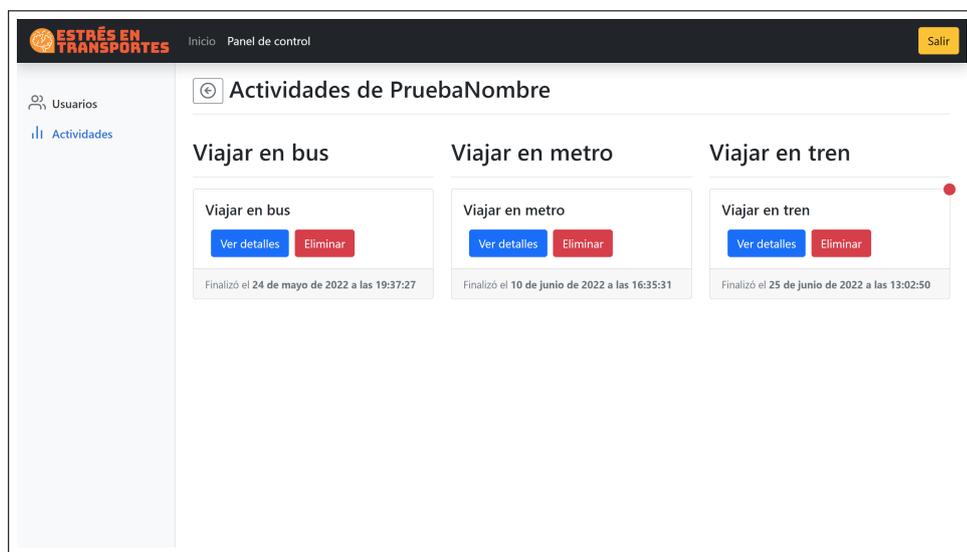


Figura 3.85: Indicador para mostrar que una actividad no ha sido visualizada aún por el terapeuta



(a) Cuando se registra un terapeuta en el sistema debe verificar su correo.



(b) Cuando el terapeuta solicita recuperar su contraseña.



(c) Cuando se edita un usuario, se le notifican sus cambios mediante correo.

Figura 3.86: Correos electrónicos implementados en esta iteración

Capítulo 4

Evaluación y validación

En este capítulo se expondrá la evaluación técnica de la propuesta desarrollada, en la que se comprueba si el método desarrollado para la detección de estrés en el transcurso de una actividad de la vida diaria (como el uso de los transportes públicos) es efectiva.

4.1. Sujeto 1

Para llevar a cabo la evaluación, hemos pedido a una persona mayor que use nuestro reloj para efectuar la medición de un trayecto en autobús cualquiera, así que nosotros actuaremos como un terapeuta. Veremos en la evaluación si distintos factores con los que esta persona se encuentre durante su trayecto, afectan a su nivel de estrés, de modo que nuestra aplicación nos ayude a explicar esos momentos de estrés.

El perfil del sujeto con el que vamos a realizar el primer experimento puede leerse en la tabla 4.1.

PERSONAJE	
Nombre	María del Carmen
Edad	70
Sexo	Mujer
Educación	Módulo superior de peluquería
Contexto de uso	
Cuándo	Lleva su reloj puesto durante todo el día, ya que le sirve para no estar dependiendo de su móvil constantemente. Además le ayuda a mantenerse activa ya que le motiva mediante logros el hacer actividad física.



Dónde	Excepto mientras se ducha, se lleva su reloj puesto allá donde vaya.
Misión	
Objetivo	Ha empezado a ir al psicólogo tras la pandemia y quiere poder volver a realizar actividades que antes de la pandemia no le causaban estrés y ansiedad, como montarse en transporte público.
Expectativas	Espera que sea fácil de usar y que sea sencilla de entender. Busca que la aplicación ayude a su terapeuta a llevar su caso.
Motivación	
Deseo	Quiere que la aplicación la ayude a mejorar su tratamiento de nivel de estrés en el uso de transporte público.
Actitud hacia la tecnología	
Tímida, precavida.	

Cuadro 4.1: Perfil de María del Carmen, sujeto 1 de nuestra prueba.

El experimento ha consistido en pedirle al sujeto que salga de su casa y se dirija al centro de Granada partiendo desde la Escuela Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada.

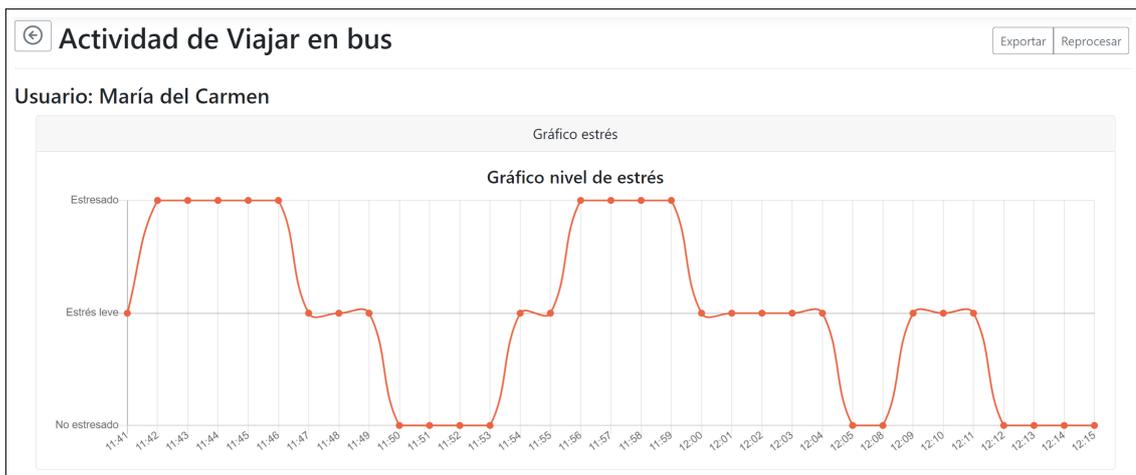


Figura 4.1: Gráfico de los niveles de estrés acumulados en el transcurso de la actividad por el Sujeto 1.

En la figura 4.1 se muestra la gráfica de niveles de estrés acumulados durante el transcurso de la actividad. Se aprecia que al comienzo de la medición se predice que el nivel de estrés es bastante elevado. Posteriormente, se observa un periodo de relajación, con una posterior vuelta a niveles de estrés alto. Tras esto, se observa un periodo final de relajación. Estos periodos pueden ser explicados con la ayuda del etiquetado que el sujeto ha realizado.

Con ayuda de la figura 4.2 se pueden justificar los niveles de estrés del gráfico. Tras

Etiquetado							
Instante	Estado	Contexto				Emoción	Sentimiento
11:44	 Esperando	 Llego tarde	 Solo	 Calor		 Asco	 Nervioso
11:46	 En viaje	 Larga espera	 Sentado	 Ruido	 Solo	 Ira	 Nervioso
11:52	 En viaje	 Sentado			 Solo	 Asco	 Tranquilo
11:57	 En viaje	 Sentado	 Ruido	 Solo	 Peligro	 Miedo	 Nervioso
12:09	 En viaje	 Sentado	 Ruido	 Solo	 Atasco	 Sorpresa	 Nervioso
12:11	 En destino		 Acompañado			 Felicidad	 Tranquilo

Figura 4.2: Tabla con las etiquetas indicadas por el sujeto 1 en el transcurso de la actividad.

hablar con el sujeto acerca de su experiencia he podido dar una mejor explicación a la gráfica obtenida y he comprobado cómo la aplicación ha sido útil para clasificar el nivel de estrés y para asociar el etiquetado. La medición de niveles de estrés ha sido bastante precisa, a excepción de algunos puntos que pueden verse al final de la medición, donde se ha tenido un periodo (12:05-12:08) que se ha clasificado como no estrés, sin embargo, se puede deducir que la persona seguía un poco estresada, porque tanto antes como después lo estaba. La rapidez con la que el sistema procesa la señal permite y facilita el seguimiento de la medición de la actividad en transcurso en directo, lo cual le añade bastante valor, pudiendo contactar con el sujeto de prueba para preguntarle si todo va bien o poder ayudarle con lo que necesitase.

En este caso de prueba, el sujeto salió de la escuela de ingeniería informática y llegó a la parada de autobús, donde tuvo que esperar bajo un asfixiante calor al autobús para el cual no se sabía el tiempo restante hasta su llegada. Cuando ya se pudo montar, encontró asientos disponibles, lo que hizo bastante ameno el trayecto. Sin embargo, entró al autobús un grupo de personas que no estaban respetando las leyes actuales de uso de mascarilla y no se les prohibió la entrada, por lo que el sujeto empezó a tener miedo y nerviosismo. Estas personas se bajaron un par de paradas después lo que hizo que el sujeto volviese a un estado más calmado. Justo llegando a la parada se formó un atasco, por el cual el sujeto se estresó de nuevo, debido a que creía que se iba a bajar ya. Tras unos minutos logró bajarse del autobús y encontrarse con su hijo, que estaba esperando

a que llegase.

El sujeto recomendaría el uso de este sistema a aquellas personas que, como ella, están tratando de conocer las causas que le producen estrés en la realización de actividades cotidianas para poder reducirlo. Este sistema le ha parecido muy útil ya que ha podido conocer qué aspectos le afectan más, como por ejemplo, el estar sola o que el ambiente tenga mucho ruido, con el objetivo de poder trabajar en qué mejorar. En un futuro, estaría dispuesta a usarlo de nuevo, ya que no sólo le ha ayudado a conocer las características que le causan estrés, si no también le ha ayudado a sentirse menos sola gracias al estar ocupada teniendo que elegir entre las distintas etiquetas.

En conclusión, la prueba técnica ha sido satisfactoria.

4.2. Sujeto 2

Para llevar a cabo la evaluación, hemos pedido a una persona de mediana edad que use nuestro reloj para llevar a cabo la medición de un trayecto en un autobús de Granada, con la finalidad de evaluar el sistema de transporte público de la ciudad. Al igual que para el sujeto anterior, veremos en el procesamiento de la actividad si distintos factores con los que esta persona se encuentre durante su trayecto, afectan a su nivel de estrés, de modo que nuestra aplicación nos ayude a explicar esos momentos de estrés.

El perfil del sujeto con el que vamos a realizar el segundo experimento puede leerse en la tabla 4.2.

PERSONAJE	
Nombre	Juan Carlos
Edad	45
Sexo	Hombre
Educación	Técnico auxiliar de enfermería
Contexto de uso	
Cuándo	Se suele poner el reloj cuando va a trabajar, de lunes a viernes, ya que le permite seguir en contacto con sus allegados sin tener que coger el móvil mientras trabaja.
Dónde	Únicamente en el trayecto para el trabajo, en el trabajo y de vuelta a casa desde el trabajo.
Misión	



Objetivo	Todos los días laborales hace uso del transporte público para ir a trabajar, lo que le causa mucho estrés y va a participar en una encuesta digital para el consorcio de transportes del área de Granada para mejorar la calidad del servicio.
Expectativas	Espera que el sistema sea fácil de usar y le entretenga en sus monótonos desplazamientos al trabajo. Busca que la aplicación ayude a la empresa prestadora del servicio de transporte a mejorar sus servicios.
Motivación	
Deseo	Quiere que la aplicación ayude a que el transporte público de su ciudad mejore y así no le cause estrés y cierto rechazo a la hora de tener que usarlo.
Actitud hacia la tecnología	
Paciente, positivo, colaborador, confiado.	

Cuadro 4.2: Perfil de Juan Carlos, sujeto 2 de nuestra prueba.

El experimento ha consistido en pedirle al sujeto que cuando se sienta estresado en alguno de sus trayectos de ida o vuelta del trabajo, que abra la aplicación y se disponga a medirse de modo que se pueda conocer qué causas son las que le hacen tener estrés y cuales son las que le hacen tranquilizarse.

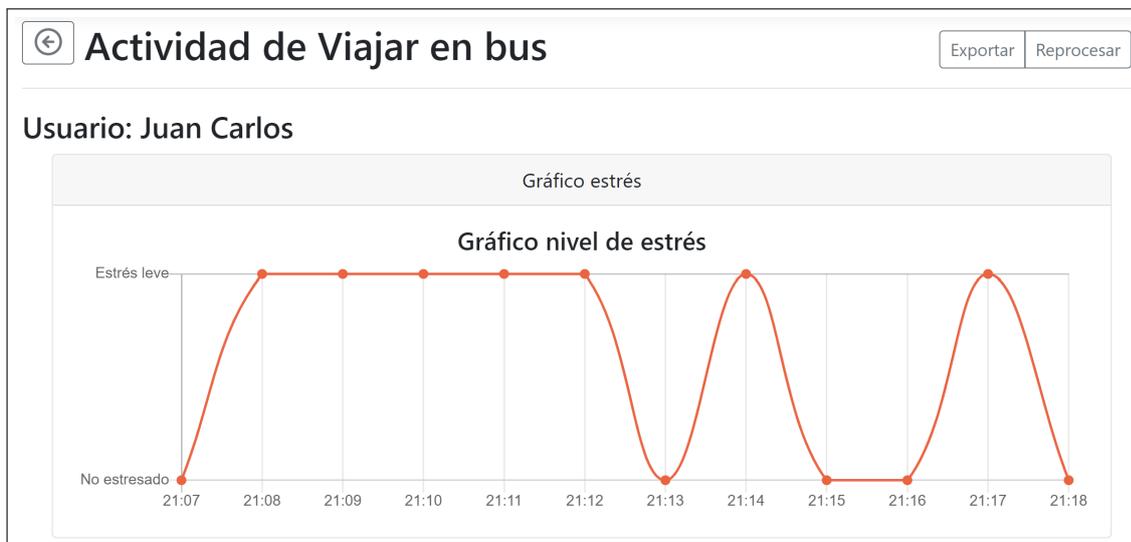


Figura 4.3: Gráfico de los niveles de estrés acumulados en el transcurso de la actividad por el Sujeto 2.

En la figura 4.3 se muestra la gráfica de niveles de estrés acumulados durante el transcurso de su experimento. Se aprecia que al comienzo de la medición se predice que el nivel de estrés crece hasta un punto medio. Tras esto, se observa un periodo de estrés

medio, con una posterior bajada a un estado más tranquilo aunque con sus subidas y bajadas. Estos periodos pueden ser explicados con la ayuda del etiquetado que el sujeto ha realizado.

Etiquetado								
Instante	Estado	Contexto					Emoción	Sentimiento
21:08	 En viaje	 Llego tarde	 Sentado	 Ruido	 Calor	 Acompañado	 Asco	 Nervioso
21:10	 En viaje	 Abarrotado	 Sentado	 Ruido	 Atasco	 Acompañado	 Asco	 Nervioso
21:13	 En viaje	 Sentado		 Ruido		 Acompañado	 Tristeza	 Tranquilo
21:18	 En destino			 Acompañado			 Felicidad	 Tranquilo

Figura 4.4: Tabla con las etiquetas indicadas por el sujeto 2 en el transcurso de la actividad.

Con ayuda de la figura 4.4 se pueden justificar los niveles de estrés del gráfico. Después de hablar con el sujeto sobre su experiencia, pudimos interpretar mejor el gráfico resultante y verificar qué tan útil es la aplicación para la clasificación de estrés y para asociar sus etiquetas. La medición de niveles de estrés ha sido al comienzo bastante precisa, a excepción de algunos puntos que pueden verse al final de la medición, donde se ha tenido un periodo (21:13-21:18) en el que se aprecian subidas y bajadas en el nivel de estrés, sin embargo, se puede deducir que la persona estaba mayormente no estresada, ya que la mayoría de puntos se encuentran en este nivel. La rapidez con la que el sistema de procesamiento de señal proporciona, permite y facilita el seguimiento de la actividad en transcurso del usuario, lo que aporta mucho valor a la aplicación, al poder contactar con el sujeto de la prueba para preguntar si todo está bien o para poder asistirle cuando lo necesite.

Para este caso de prueba, el sujeto salió de su puesto de trabajo y llegó a la parada de autobús, donde tuvo que esperar al autobús para el cual no se sabía el tiempo restante hasta su llegada, sin embargo, estaba hasta este momento tranquilo. Cuando ya se pudo montar, observó que el autobús estaba bastante viejo, por lo que era ruidoso, hacía calor, ya que no tenía aire acondicionado y tenía ganas de llegar a casa, por lo que marcó que llegaba tarde. Entre todas las paradas que hacía el autobús, comenzó a llenarse bastante

lo cual hizo que el sujeto se sintiese nervioso, junto al atasco que hay a la hora de la salida de los trabajos. Tiempo después, se bajaron la mayoría de los viajeros y salió de la zona donde había atascos, el sujeto sintió tristeza pero estaba a la vez tranquilo, ya que había tenido un día estresante en la oficina. Para terminar, marcó cuándo llegó a su destino, al mismo tiempo que se sentía feliz y tranquilo.

Estos datos nos dan información para mejorar el servicio que se presta como por ejemplo, valorar que se ha de ofrecer una frecuencia de servicio mayor, para bajar la afluencia de gente que se monta en un vehículo, y se han de cambiar aquellos autobuses que sean viejos, para ofrecer mayor comodidad a los usuario y que les provoque menor rechazo.

Este sujeto ha encontrado esta forma de encuestar a los usuarios muy entretenida y amena y espera que más personas, como él, quieran participar en mejorar el sistema de transporte público de su ciudad. Este sistema le ha parecido muy útil ya que gracias a él el consorcio ha podido conocer qué aspectos afectan más, como por ejemplo, que esté abarrotado un vehículo o que exista mucho ruido. En un futuro, estaría dispuesto a usarlo de nuevo, ya que no sólo espera que ayude a mejorar el sistema de transporte de su ciudad, si no también le ha ayudado a entretenerse en el trayecto gracias al registro de etiquetas que ha tenido que realizar.

En conclusión, la prueba técnica ha sido satisfactoria.

Capítulo 5

Conclusiones y Trabajos Futuros

En este último capítulo, se presentan los objetivos que se han cumplido en este trabajo fin de grado y se indican algunas líneas futuras de desarrollo.

5.1. Conclusiones

Para empezar, cabe señalar que la realización de este trabajo de fin de carrera se ha podido realizar gracias a los conocimientos adquiridos en las asignaturas cursadas a lo largo del grado junto con la capacidad de formación autodidacta igualmente obtenida en el grado. Gracias a esto se ha podido seguir una metodología de trabajo eficaz y se ha podido cumplir con las necesidades planteadas para el proyecto por las tutoras.

En esencia, se han podido cumplir todos los objetivos marcados al comienzo del proyecto mediante el desarrollo de un entorno multisistema para la monitorización de datos fisiológicos, para todo tipo de usuarios y durante el transcurso de una actividad diaria, como el usar el transporte público. Este entorno permite, por un lado, modelar una actividad en cuanto a grado de estrés producido y, por otro lado, conocer los factores que la producen. Por otro lado, otorga la posibilidad a los terapeutas de hacer un seguimiento preciso a sus pacientes mediante el registro de ellos en la plataforma y el poder consultar las actividades que realizan. Por último, también podría permitir que los consorcios de transporte presten una mejor atención y servicio, a través del análisis de los datos fisiológicos y los etiquetados que se recopilan. A continuación, se exponen todos los objetivos específicos que se habían planteado y que se han logrado en el desarrollo del presente trabajo fin de grado:

- Se ha podido estudiar en profundidad el estrés en la sección 2.3, conociendo los factores que lo pueden producir, en especial en situaciones de uso de transporte público, y entendiendo qué factores afectan en especial al grupo de personas mayores.

- En la sección 2.5 se han investigado los mecanismos de medición de estrés que existen en la actualidad, entendiendo las necesidades en la actualidad de llevar a cabo un proyecto como este, ya que la mayoría son aún manuales.
- En la sección 1.1 se ha estudiado si los consorcios de transporte se ocupan de valorar el nivel de estrés que se produce al usuario, mediante la extracción de varios cuestionarios realizados en años anteriores, viéndose que no es algo que tengan en cuenta.
- En la sección 3.5.4 se ha investigado y desarrollado una metodología de medición de estrés mediante la obtención de datos en crudo del reloj inteligente y procesándolos a través de un algoritmo de clasificación basado en un modelo ya existente que clasifica niveles de estrés en función de parámetros cardíacos.
- En la sección 3.5.4 se ha investigado y usado un módulo para la limpieza de la señal obtenida por el sensor del reloj, ya que originalmente se obtiene con bastante ruido. La limpieza está basada en un filtro Butterworth de paso banda, para poder obtener las características necesarias para que el modelo desarrollado pueda predecir el nivel de estrés dado un conjunto de medidas.
- Poniendo en conjunción al modelo entrenado mediante el algoritmo de clasificación k vecinos más cercanos con el conjunto de datos existente y al conjunto de características obtenidas de un conjunto de mediciones por el módulo HeartPy, se ha podido calcular el nivel de estrés de un usuario a partir de sus mediciones fisiológicas en el capítulo 4 de evaluación técnica.
- A lo largo del capítulo 3 se ha desarrollado una plataforma para que terapeutas puedan registrar y visualizar los resultados de estrés que conlleva la realización de actividades por parte de sus pacientes o usuarios, como el uso de medios de transporte.
- En la sección 3.5.4 se ha optimizado el uso de los recursos existentes de modo que un mismo reloj pueda recoger datos de varios usuarios, desarrollando una aplicación móvil para la identificación de distintos usuarios para el reloj inteligente vinculado, de modo que se pueda facilitar la investigación sin necesidad de usar múltiples relojes.
- A lo largo del capítulo 3 se ha desarrollado una aplicación para reloj inteligente que es capaz de registrar actividades de tipo viaje en transporte público, y permitiendo al usuario indicar el medio de transporte usado junto con una serie de etiquetas asociadas a estados de viaje, contextos relacionados con situaciones que se pueden dar durante el trayecto, emociones y sentimientos, todo ello mientras toma medición del sensor fisiológico y la envía al servidor.
- Para la realización del proyecto se han usado metodologías ágiles para llevar un desarrollo continuo, equilibrado y con retroalimentación constante por parte de las

tutoras, de modo que se haya obtenido un software al gusto de las mismas y con calidad.

5.2. Trabajos futuros

El desarrollo que se ha llevado a cabo en este trabajo fin de grado puede extenderse en el futuro de varias formas, una relacionada con la extensión de las tres aplicaciones desarrolladas y otra referente al modelo de predicción y características que se usan para medir el nivel de estrés en el usuario.

En relación a la extensión de las aplicaciones desarrolladas se desglosan a continuación las propuestas para el futuro para cada aplicación:

5.2.1. Aplicación Web

Para la aplicación web se proponen las siguientes funcionalidades:

- Soporte para que varios terapeutas coexistan en el sistema cada uno con sus usuarios registrados, de modo que no tengan que compartir los usuarios registrados en común.
- Mejora en la visualización del gráfico de estrés, aparte de ver las etiquetas a modo de tabla, se puede añadir una funcionalidad donde se colocan las etiquetas sobre el instante de tiempo del gráfico de estrés.
- Desarrollar una funcionalidad para añadir etiquetas y clasificaciones, de modo que sea el terapeuta el que tenga el control de qué información se quiera conocer y de los iconos que se le muestran al usuario.
- Añadir una funcionalidad para crear dinámicamente tipos de actividad, de medios de transporte público o de cualquier otra actividad instrumental de la vida diaria, de modo que al usuario le aparezcan en la lista de selección de tipo de actividad.
- Filtros y una clasificación más precisa a la hora de mostrar las actividades en la plataforma web.
- Visualización de las posiciones GPS del usuario en un mapa a medida que se ha realizado una actividad.
- Almacenar los datos de forma encriptada y realizar la instalación del sistema web en un servidor seguro cifrado mediante el certificado SSL, para que la transmisión de datos se realice de forma encriptada y con un protocolo seguro como HTTPS.

5.2.2. Aplicación Móvil

Para la aplicación móvil se sugieren las siguientes mejoras relacionadas con dar feedback al usuario:

- Mostrar al usuario identificado un emoticono o imagen basada en los últimos registros de estrés para que le ayude a relajarse.
- Mostrar al usuario un gráfico donde pueda consultar los niveles de estrés acumulados en las últimas actividades, clasificándolas por “hoy”, “últimos días”, “desde hace una semana”, “desde hace un mes”, o seleccionando por rango de fechas.

5.2.3. Aplicación Reloj Inteligente

Para la aplicación del reloj inteligente se plantean las siguientes mejoras:

- Obtención dinámica del listado de tipos de actividades disponibles en el sistema, de modo que el terapeuta pueda configurar los tipos de actividad.
- Obtención dinámica de las etiquetas con sus iconos registradas en el sistema por el terapeuta.
- Uso de otros sensores como el de GPS para enviar la localización del usuario.

En relación a la extensión del modelo de predicción desarrollado, se pueden sugerir mejoras en cuanto al uso de otros conjuntos de datos, o en cuanto al uso en combinación con otros sensores del reloj inteligente, a modo de mejorar la predicción de nivel de estrés.

5.2.4. Modelo de predicción y características obtenidas

Respecto a las mejoras del modelo de predicción se sugieren las siguientes mejoras:

- Investigar de la tasa de acierto con un modelo distinto, entrenándolo con un conjunto de datos como el WESAD y ver si la tasa de acierto se mejora. Incluso se podría hacer un conjunto de datos a partir del WESAD y el SWELL y estudiar si se mejora el resultado original.
- Investigar mediante el uso de otros sensores, como el acelerómetro y el giroscopio, si se puede mejorar el gráfico de estrés, mediante la detección de estar haciendo actividad física, ya que fisiológicamente se estaría dando una respuesta similar al estrés, pero obviamente no lo consideramos un factor de estrés producido por el uso del transporte público.

5.3. Valoración personal

La realización del presente trabajo fin de grado ha supuesto un gran reto personal, ya que se ha estudiado en profundidad un problema, como es el estrés, que afecta a gran parte de la sociedad pero que se trata de invisibilizar o normalizar. Además, se ha tenido que tomar conciencia de los problemas de usuarios mayores o con alguna discapacidad en la realización de actividades diarias como el uso del transporte público, lo cual me ha hecho empatizar aún más con las necesidades de estos colectivos. En cuanto a la realización de un proyecto de esta envergadura, debido a la complejidad de la puesta en conjunto de tres sistemas que se comuniquen de forma simultánea, ha sido necesario realizar algún curso para conocer el desarrollo en Android nativo mediante Kotlin. La toma de decisiones en cuanto al uso de ciertas tecnologías, han hecho que mis habilidades como ingeniero se apliquen y se extiendan bastante. Valoro de forma positiva que este trabajo se pueda ampliar en bastantes direcciones en un futuro. En cuanto a lo personal, considero que he sido muy constante en la realización de este proyecto, esforzándome al máximo en todos los ámbitos que abarca este trabajo, comunicándome regularmente con las tutoras y formándome en todo aquello que era necesario para realizar el desarrollo de una forma correcta y siguiendo convenciones, a fin de que este proyecto pueda ser mantenido y ampliado en un futuro.

Ha sido muy gratificante tanto ver cómo en el proceso de desarrollo el sistema iba tomando su forma final como la evaluación técnica que se ha realizado con un sujeto, la cual se llevó a cabo de forma satisfactoria y diluye las numerosas dificultades encontradas en el camino. Esta experiencia me ha ayudado a ver cómo, mediante la constancia, se puede lograr realizar un buen estudio, propuesta y desarrollo de un proyecto que resuelva un problema presente en nuestra sociedad.

Apéndice A

Manual de instalación

En este capítulo se expondrá cómo instalar las plataformas desarrolladas en este trabajo fin de grado.

A.1. Sistema web

Para realizar la instalación del sistema web, debemos tener un computador con unas características similares a las que se exponen en la sección 1.5. Se mostrará los pasos a seguir para la instalación del sistema en un sistema operativo basado en Linux como Ubuntu 20.04.

Comenzaremos con la instalación de Docker en nuestro computador actualizando los índices de paquetes de apt para poder usar repositorios sobre HTTPS.

```
1 $ sudo apt-get update
2
3 $ sudo apt-get install \
4     ca-certificates \
5     curl \
6     gnupg \
7     lsb-release
```

Añadiremos, tras esto, la clave oficial GPG de Docker:

```
1 $ sudo mkdir -p /etc/apt/keyrings
2 $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
   ↪ /etc/apt/keyrings/docker.gpgcurl -fsSL https://download.docker.com/linux/
   ↪ ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

Después de ejecutar lo anterior, configuraremos el repositorio:

```
1 $ echo \
2     "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
   ↪ https://download.docker.com/linux/ubuntu \
3     $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev
   ↪ /null
```

Habiendo realizado los pasos anteriores, instalaremos el motor Docker ejecutando los siguientes comandos:

```
1 $ sudo apt-get update
2 $ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Teniendo ya instalado el motor de Docker en el servidor, debemos localizar el árbol de directorios del proyecto. Esto se puede hacer, bien consultando los archivos adjuntos a esta memoria, o bien clonando el repositorio de GitHub asociado a este proyecto, mediante el siguiente comando.

```
1 $ git clone https://github.com/VictorRubia/TFG.git
```

Una vez tenemos localizados los archivos y carpetas del proyecto, nos situaremos con un terminal dentro de la carpeta “backend”. Habiendo hecho esto, ejecutaremos el comando que despliega el servidor.

```
1 $ cd backend
2 $ docker-compose up
```

Finalmente, una vez que termine la instalación, podremos acceder al servidor en la dirección <http://localhost:3000/>.

A.2. Aplicación móvil

Para instalar la aplicación móvil podemos acceder al siguiente directorio del proyecto, donde encontraremos un archivo denominado *mobile-debug.apk*, que podemos transferir al teléfono e instalar habilitando instalación por orígenes desconocidos.

```
1 $ cd app/mobile/build/outputs/apk/debug/
```

De forma alternativa se puede obtener el *.apk*, tras realizar la instalación y el despliegue del servidor web, accediendo desde el navegador del ordenador donde se haya desplegado el servidor a la dirección http://localhost:3000/download_android.apk.

A.3. Aplicación reloj inteligente

Para poder instalar la aplicación en un reloj inteligente debemos tener activado el modo desarrollador, que se puede realizar accediendo a la configuración del reloj, desplazándonos a la categoría “Sistema”, tras esto pulsaremos sobre “Información” y pulsaremos 7 veces sobre “Número de compilación”, hasta que se muestre un mensaje por pantalla donde se indique que se ha activado correctamente el modo desarrollador.

Una vez hemos hecho esto, podemos activar dentro de las opciones de desarrollador la “Depuración ADB sobre Wi-Fi”. Nos aparecerá una dirección IP que debemos de

guardar para introducirla posteriormente en el terminal del ordenador que usemos para transferir la aplicación.

En el computador debemos tener instalado ADB, y si no se tiene ya instalado se debe realizar la instalación mediante la ejecución de del siguiente comando y comprobar que se ha instalado correctamente

```
1 $ sudo apt install android-tools-adb
2 $ adb version
```

A continuación debemos establecer una conexión entre el ordenador, sobre el que hemos ejecutado los comandos del paso anterior, y el reloj inteligente introduciendo el siguiente comando en el terminal e introduciendo la IP que nos guardamos previamente correspondiente al reloj inteligente.

```
1 $ adb connect (IP_RELOJ_INTELIGENTE)
```

Tras esto, se debe enviar el apk a la memoria interna del reloj inteligente e instalarlo mediante el siguiente comando. El archivo *.apk* para el reloj inteligente se puede encontrar en la ruta que se muestra a continuación. De igual forma, se puede descargar el archivo *.apk* habiendo instalado y desplegado el servidor web en el siguiente enlace http://localhost:3000/download_wearos_apk

```
1 $ cd app/wear/build/outputs/apk/debug/
2 $ adb push wear-debug.apk /sdcard/
3 $ adb -e install wear-debug.apk
```

695792369

Apéndice B

Manual de usuario

En este capítulo se explica cómo usar los distintos sistemas y las funcionalidades que ofrece.

B.1. Aplicación móvil

Tras la instalación de la aplicación en el dispositivo móvil, la iniciaremos pulsando sobre el icono de la aplicación, como se muestra en la figura B.1



Figura B.1: Pantalla inicial de un teléfono Android.

B.1.1. Identificación del usuario

A continuación, en la figura B.3, se puede ver una pantalla donde existen dos campos para introducir un correo electrónico y una contraseña. Estas deben haber llegado a su correo electrónico cuando el terapeuta lo dio de alta en la plataforma y debe tener la forma que se muestra en la figura B.2. Procedemos a introducirlas en los campos y a pulsar sobre entrar para identificarnos y poder empezar a realizar mediciones en el reloj inteligente.



Figura B.2: Correo electrónico recibido tras el registro.

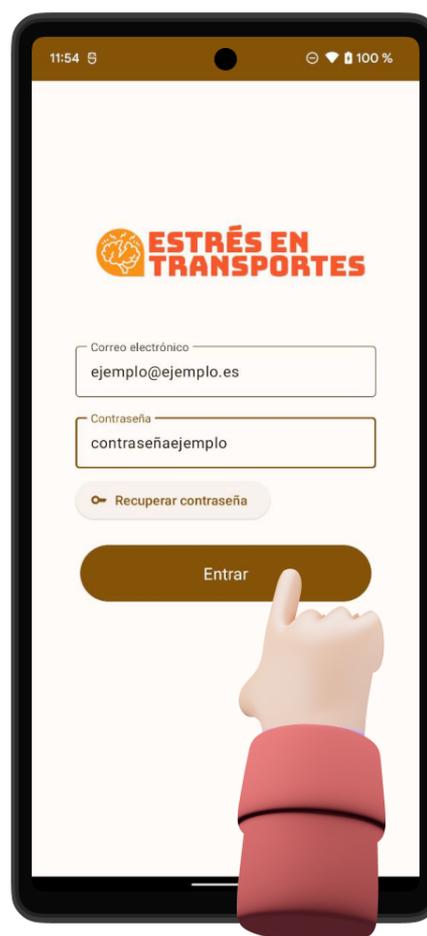


Figura B.3: Pantalla para la identificación de un usuario.

B.1.2. Pantalla de usuario identificado

En la figura B.4, se muestra el aspecto de la pantalla tras haberse identificado el usuario en la aplicación. Esta pantalla muestra el nombre o seudónimo del usuario junto al estado de la conexión con el reloj inteligente. Será un icono de verificación verde cuando el reloj esté correctamente conectado al teléfono, y en caso contrario, se mostrará una cruz de color rojo como se ve en la figura B.5. En este punto el usuario puede dejar de usar la aplicación del teléfono, incluso cerrándola desde el administrador de tareas, ya que será capaz de identificar en el reloj al usuario que se encuentre identificado en el teléfono cuando quiera comenzar a medir sus actividades.



Figura B.4: Pantalla tras el inicio de sesión del usuario con conexión correcta al reloj inteligente. Figura B.5: Pantalla tras el inicio de sesión del usuario sin conexión al reloj inteligente.

B.1.3. Recuperar contraseña

Si se quiere recuperar la contraseña, porque se haya olvidado, se puede pulsar sobre el botón “Recuperar contraseña” de la pantalla para la identificación (Figura B.3). En esta pantalla (Figura B.6) debemos introducir el correo electrónico con el que el terapeuta nos registró en el sistema. La contraseña será enviada al correo que se ha indicado (Figura B.7).



Figura B.6: Pantalla de recuperación de contraseña.



Figura B.7: Correo recibido tras la solicitud de contraseña.

B.1.4. Cerrar sesión

Para cambiar el usuario que utiliza el reloj, se puede usar la función de cerrar sesión. En la figura B.8 se muestra arriba a la derecha un botón para el cierre de sesión de un usuario ya identificado. De esta forma, se volverá a la pantalla de identificación que se muestra en la figura B.3.

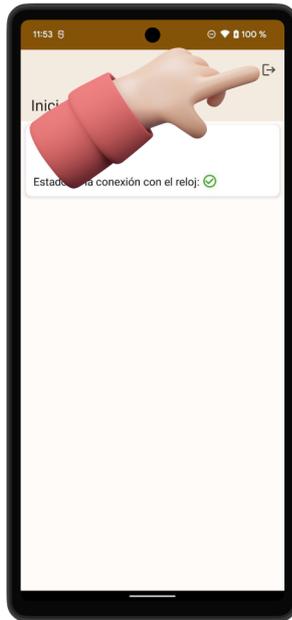


Figura B.8: Pantalla de usuario identificado para cerrar sesión.

B.2. Plataforma Web

En la plataforma web los terapeutas pueden darse de alta para monitorizar las actividades que realicen los usuarios de la plataforma. En esta plataforma el terapeuta puede gestionar a los usuarios, registrándolos, editándolos y eliminándolos junto con el poder visualizar el procesamiento de niveles de estrés acumulados en el transcurso de las actividades realizadas por dichos usuarios. En los siguientes apartados se expone cómo usar la plataforma web.

B.2.1. Registro de terapeuta

Para comenzar a usar la plataforma web es indispensable que un terapeuta se registre en el sistema. Para ello, nos situaremos en la página de inicio y situaremos el cursor sobre

el botón situado en la barra negra en la esquina superior derecha llamado “Registrarse” tal y como se muestra en la figura B.12. A continuación, veremos una pantalla como la de la figura B.10, en la que debemos rellenar los datos que se piden, correo electrónico y contraseña. Tras esto, pulsando el botón registrarse, llegará al buzón de correo de la dirección indicada un mensaje para verificar la cuenta como el de la figura B.11.



Figura B.9: Pantalla inicial de la plataforma web.



Figura B.10: Pantalla de registro de la plataforma web.



Figura B.11: Correo de verificación de cuenta para el terapeuta. Para poder empezar a usar la plataforma es necesario pulsar sobre el botón de "Verificar Cuenta"

B.2.2. Identificación de terapeuta y recuperación de contraseña

En el caso en el que se tenga ya una cuenta registrada como terapeuta en la plataforma y queramos identificarnos en el sitio web, debemos situarnos en la pantalla inicial y pulsar sobre el botón de “Entrar” situado en la barra de navegación negra en la esquina superior derecha de la pantalla, como aparece en la figura B.12. Tras esto, debemos rellenar el formulario con el correo electrónico y contraseña del terapeuta que va a identificarse y pulsar sobre el botón “Entrar” como se muestra en la figura B.13. En esta página podemos encontrar también la opción para crear una cuenta, para recuperar la contraseña si se ha olvidado y la opción de volver a enviar el correo de verificación de cuenta en el caso en el que no haya llegado al buzón de correo del terapeuta.



Figura B.12: Pantalla inicial de la plataforma web.

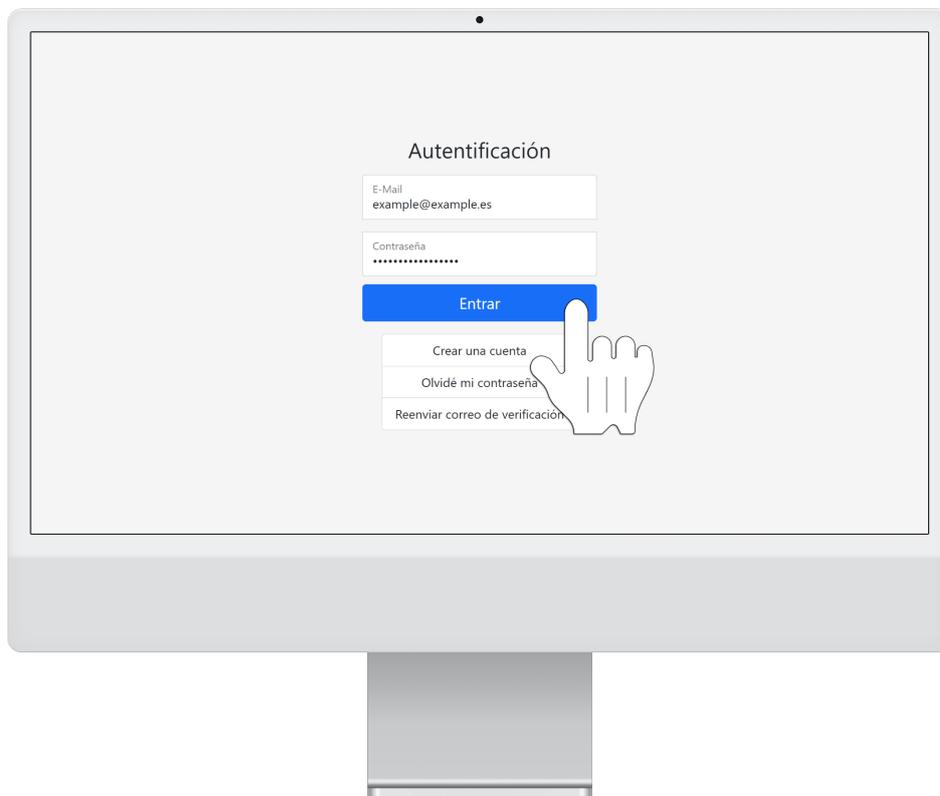


Figura B.13: Pantalla para identificarse en la plataforma web.

B.2.3. Gestión de usuarios

Una vez está el terapeuta identificado en la plataforma, se puede proceder a visitar el panel de control donde poder gestionar los usuarios que tenemos en la plataforma, como se muestra en la figura B.14. Para ello, debemos, en primer lugar, pulsar sobre “Usuarios” en la barra lateral izquierda tal y como se muestra en la figura B.15. Tras esto, vemos en un listado a los usuarios que tenemos registrados en la plataforma como se muestra en la figura B.16. Se proporciona además, un buscador para facilitar encontrar al usuario deseado en las ocasiones en las que tenemos numerosos usuarios.



Figura B.14: Acceso a panel de control desde la pantalla de inicio.

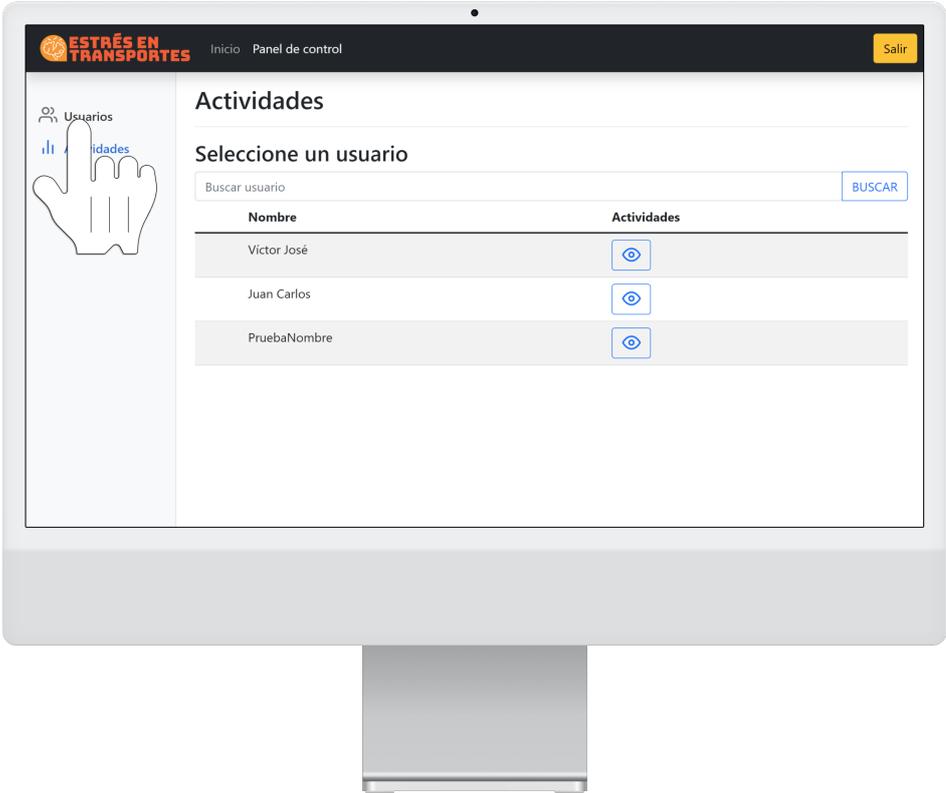


Figura B.15: Panel de control acceso a gestión de usuarios.

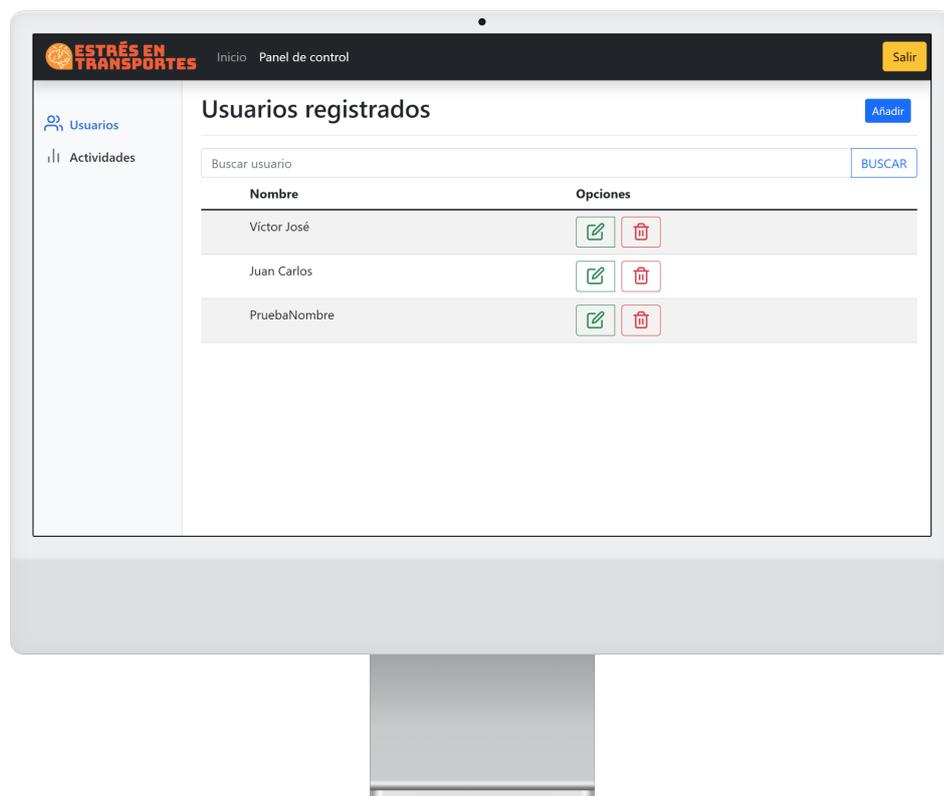


Figura B.16: Panel de control sección de gestión de usuarios.

Registrar un usuario

Para registrar a un usuario en la plataforma web, pulsaremos sobre el botón azul de la esquina superior derecha llamado “Añadir”, como aparece en la figura B.17. Aparecerá entonces una ventana en medio de la pantalla en la que se encuentra un formulario a rellenar con el nombre o apodo del usuario, el correo del usuario y una contraseña para que el usuario pueda identificarse en la plataforma, como se muestra en la figura B.18. La contraseña la puede establecer el terapeuta o puede consultarlo las preferencias del usuario. Tras rellenarlo todo, se debe pulsar sobre el botón verde “Añadir” para completar el registro del nuevo usuario. Esto hará que se le envíe un correo a la dirección proporcionada con los datos de identificación introducidos, como se muestra en la figura B.2.

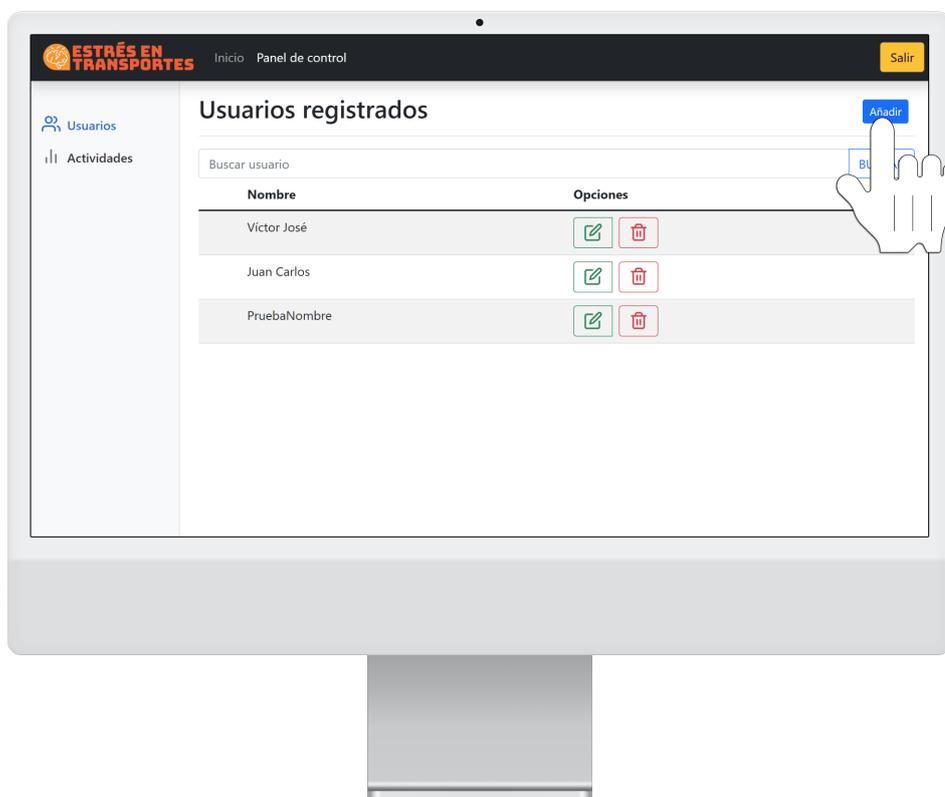


Figura B.17: Gestión de usuarios registrar un usuario.

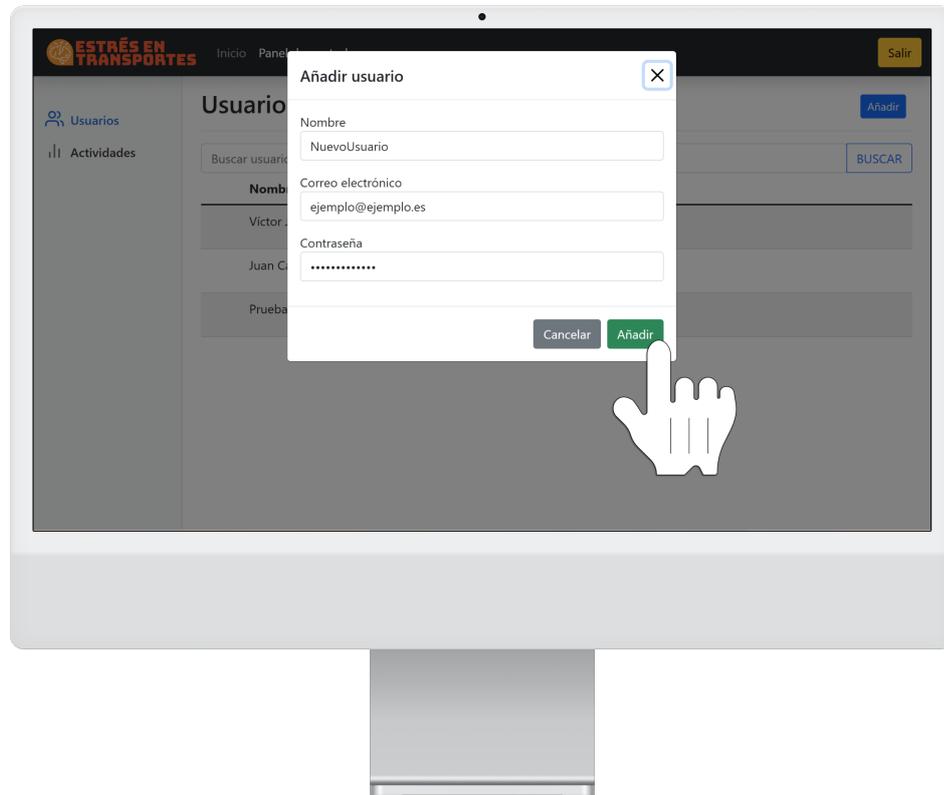


Figura B.18: Formulario para introducir información para registrar un usuario.

Edición de usuario

Para poder cambiar la información de un usuario registrado nos situaremos sobre el apartado de gestión de usuarios dentro del panel de control. Tras esto, pulsaremos sobre el botón verde indicado mediante un lápiz bajo la columna “Opciones” de la tabla en la fila donde se encuentre el usuario para el que queremos editar la información, como se muestra en la figura B.19. Tras esto, se muestra una ventana sobre la pantalla con un formulario que se encuentra relleno con la información perteneciente al usuario seleccionado. Sobre esto, podemos realizar las modificaciones y poder guardarlas pulsando sobre “Editar”, como se muestra en la figura B.20.

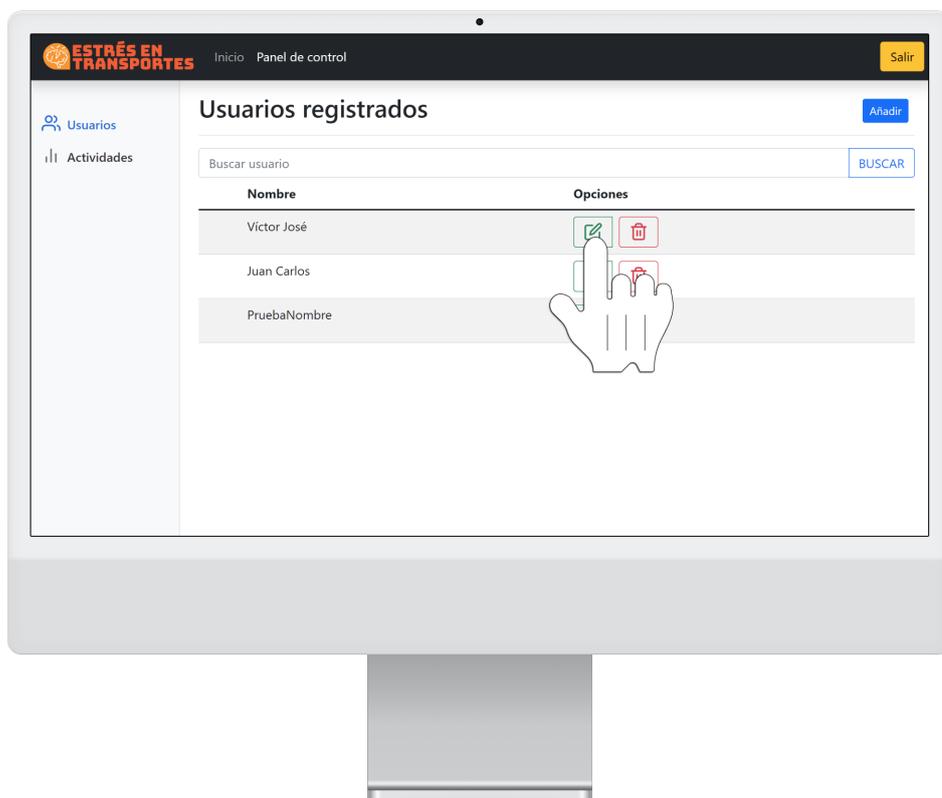


Figura B.19: Gestión de usuarios editar un usuario.

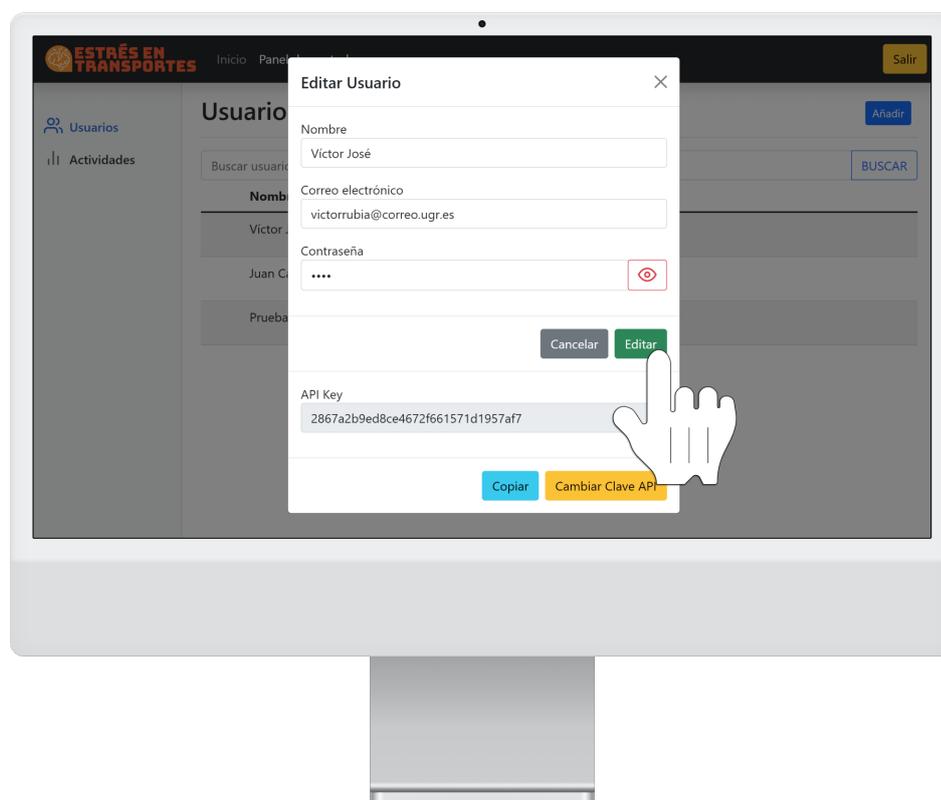


Figura B.20: Formulario para editar la información de un usuario registrado.

Eliminar usuario

Si se quiere eliminar un usuario que está registrado en el sistema, nos volveremos a situar sobre el apartado de gestión de usuarios dentro del panel de control. En esta página, pulsaremos sobre el botón rojo indicado mediante una papelera bajo la columna “Opciones” de la tabla en la fila donde se encuentre el usuario que queramos eliminar, como se muestra en la figura B.21. Tras esto se mostrará una ventana emergente para confirmar la acción de borrado.

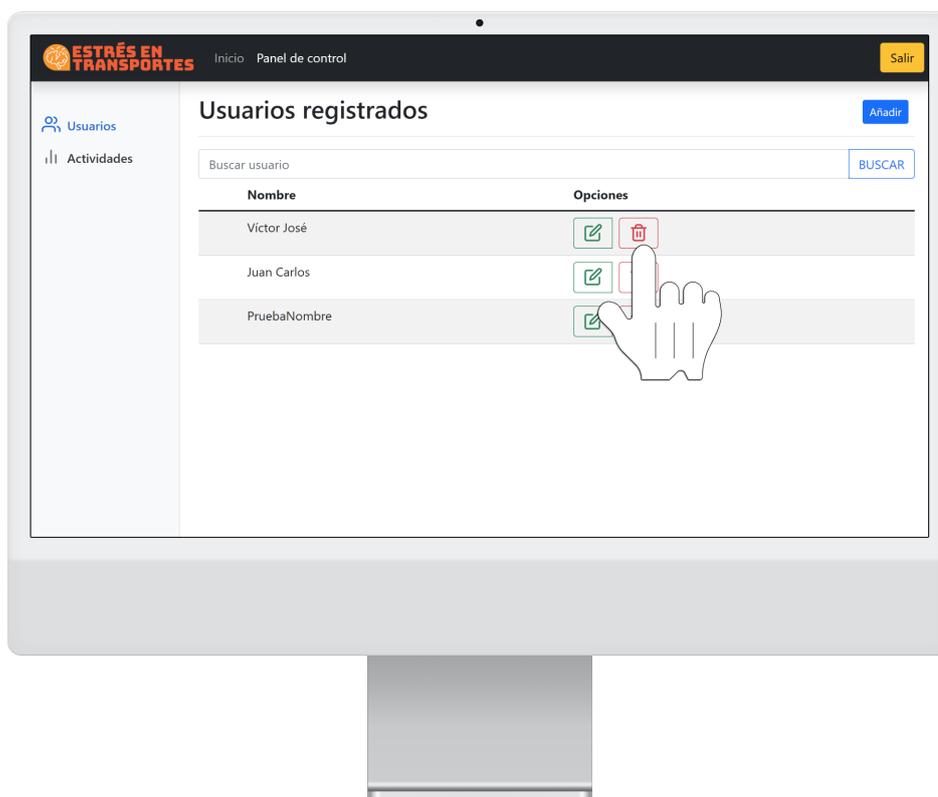


Figura B.21: Eliminar un usuario registrado.

B.2.4. Gestión de actividades

Para la visualización de las actividades realizadas por los usuarios registrados en el sistema debemos dirigirnos a la sección “Actividades” dentro del panel de control. Lo podemos hacer pulsando en la barra lateral izquierda sobre el enlace “Actividades” o bien pulsando nuevamente en el enlace “Panel de control” situado en la barra de navegación superior. Tras esto, se muestra una página donde se pueden ver en una tabla los usuarios registrados en el sistema para los que podemos ver actividades realizadas.

Se proporciona además un buscador para facilitar la búsqueda de usuarios en situaciones en las que existan numerosos usuarios registrados. Se puede ver esta página en la figura B.22

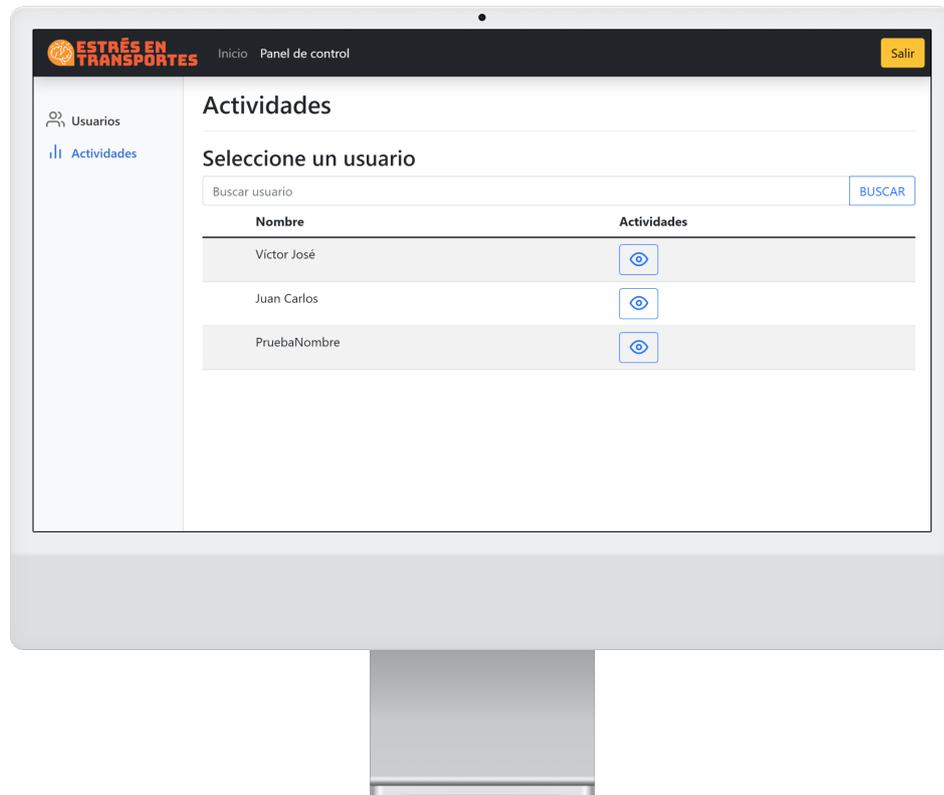


Figura B.22: Sección de gestión de actividades del panel de control.

Visualización de actividades para un usuario

Para visualizar las actividades que ha realizado un usuario pulsaremos sobre el botón de color azul indicado mediante un ojo bajo la columna “Actividades” de la tabla para la fila donde se encuentre el usuario para el que queremos visualizar las actividades, como se puede ver en la figura B.24. Tras esto, se muestra una página dividida en tres columnas donde se sitúan las actividades realizadas por el usuario ordenadas por fecha de finalización descendente e indicando dicha fecha de finalización en la parte baja de la tarjeta de actividad.

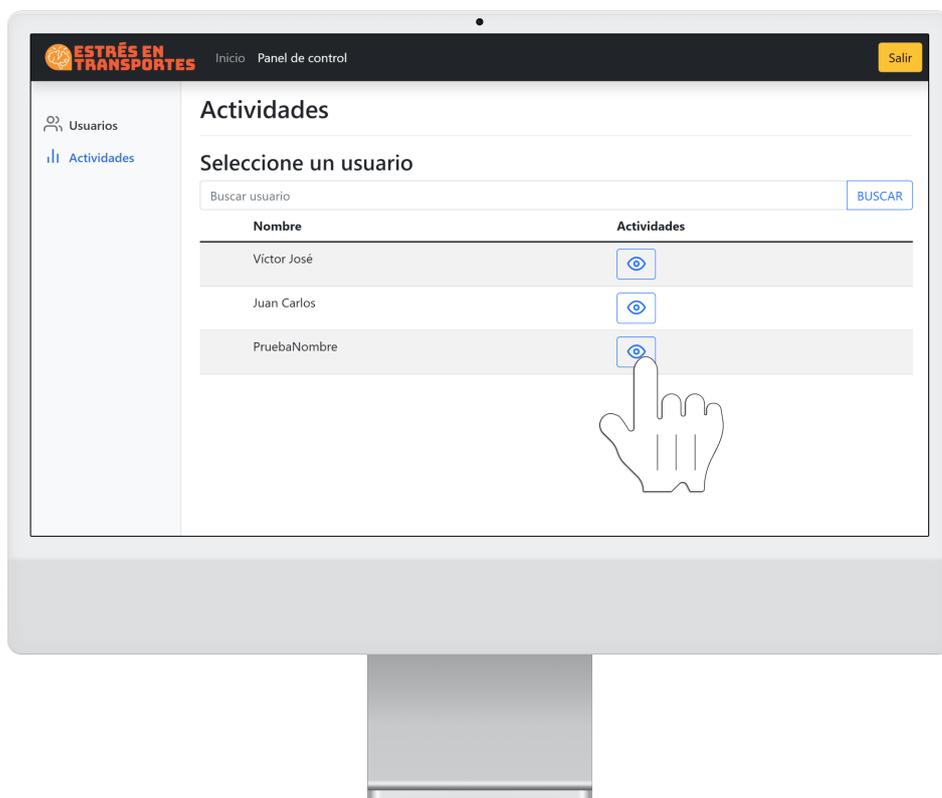


Figura B.23: Sección de gestión de actividades del panel de control.

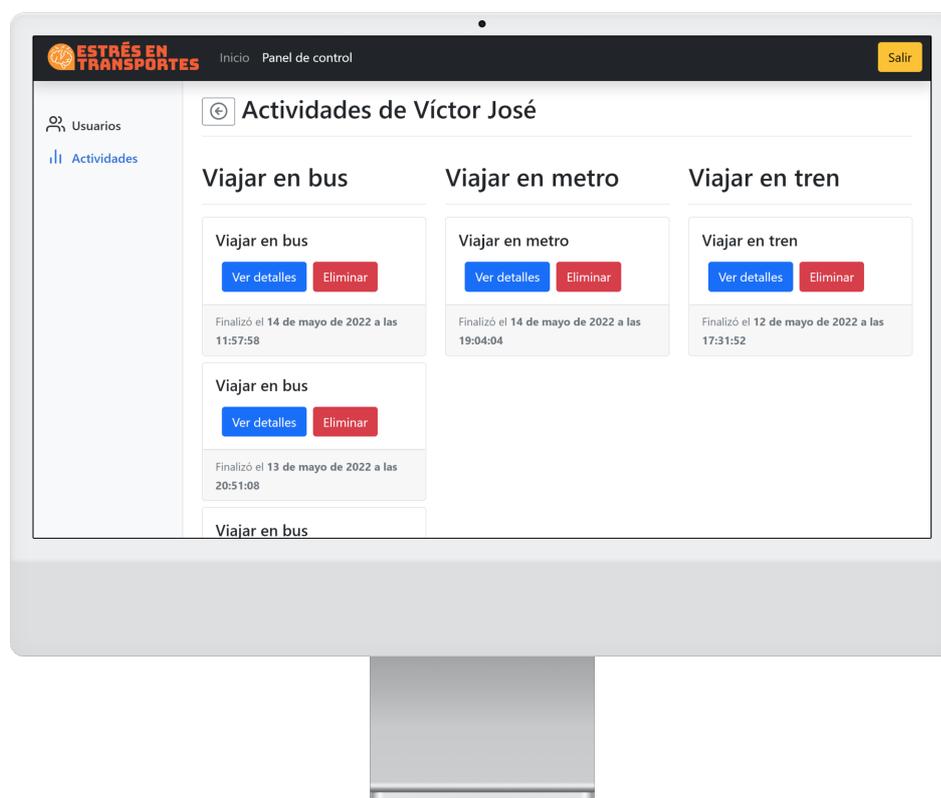


Figura B.24: Página de visualización de las actividades de un usuario.

Eliminar actividades de un usuario

Para eliminar una actividad registrada en el sistema basta con pulsar sobre el botón rojo etiquetado por “Eliminar” que se sitúa en la tarjeta de la actividad deseada, como se muestra en la figura B.25. Aparecerá una ventana emergente para confirmar la eliminación de la actividad seleccionada.

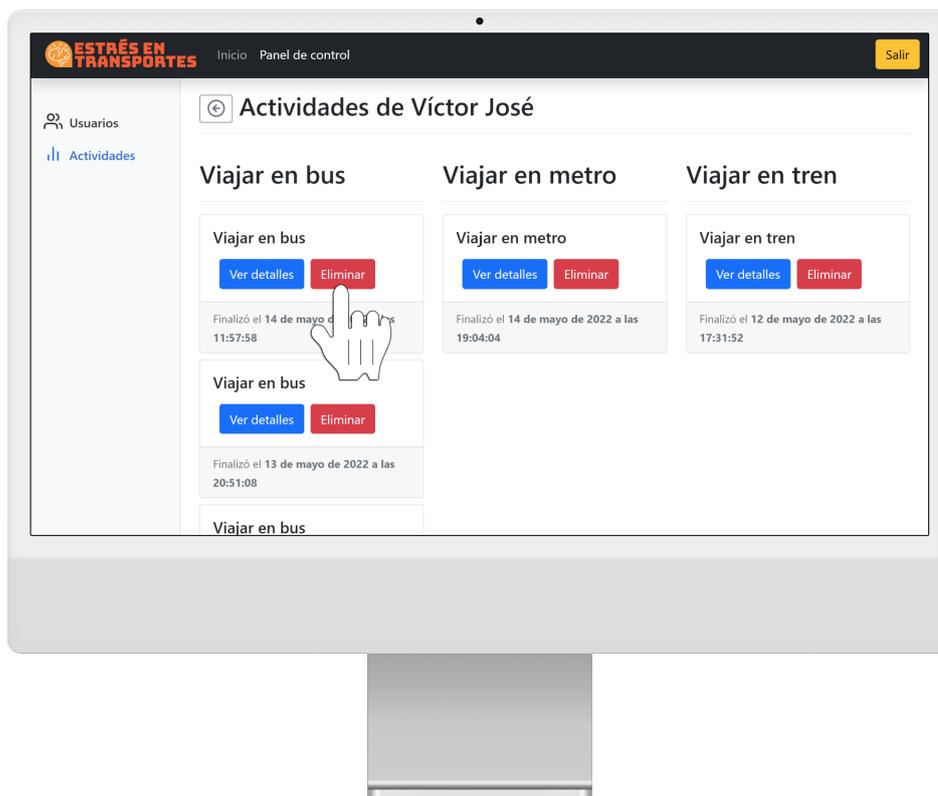


Figura B.25: Borrado de actividades registradas en el sistema para un usuario.

Visualizar detalles de una actividad

Si se quiere visualizar el procesamiento de la actividad realizada por el usuario se debe pulsar sobre el botón azul etiquetado por “Ver detalles” situado en la tarjeta de la actividad deseada, como en la figura B.26. Tras esto, se muestra una página como en la figura B.27, en la que se muestra en la parte superior un botón para volver atrás, junto con el tipo de actividad y un botón para extraer los datos recabados del sensor fisiológico en crudo en formato .CSV y un botón para volver hacer el procesamiento en caso de que el terapeuta detecte anomalías. En el caso en el que la actividad esté transcurriendo en directo, se muestra un indicador junto al tipo de actividad que se detalla. Tras esto,

se muestra el nombre del usuario al que pertenece esta actividad junto con el gráfico con los niveles de estrés procesados a lo largo del tiempo por el que ha transcurrido la actividad. Por último, se muestra una tabla con las etiquetas registradas por el usuario en el transcurso de la actividad junto al instante de tiempo. De este modo podemos unir los registros de etiquetas con la línea de tiempo del gráfico de niveles de estrés de la actividad.

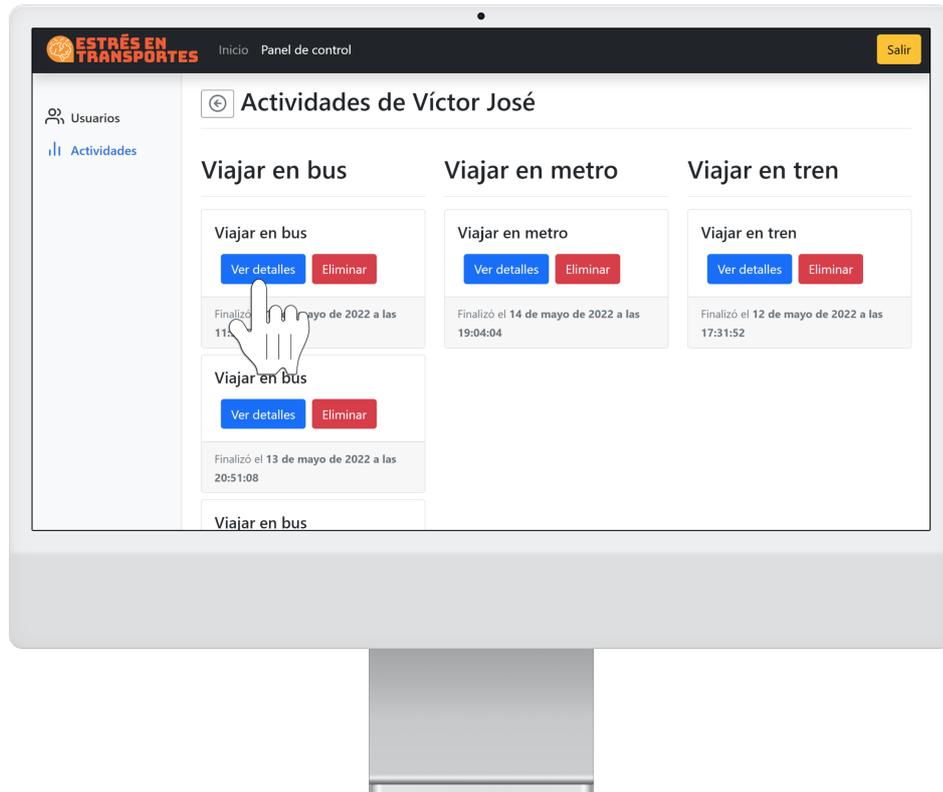


Figura B.26: Listado de actividades forma de ver los detalles.



Figura B.27: Página de visualización de los detalles de una actividad realizada por un usuario.

B.3. Reloj inteligente

Al iniciar la aplicación del reloj, se muestra una pantalla como la visible en la figura B.28, que obtiene el usuario identificado tras unos segundos necesarios de conexión con el teléfono móvil. Cuando lo recibe, pasa a la pantalla inicial de “Iniciar actividad”. Si se da la situación de que en el teléfono no existe un usuario identificado, el reloj avisará con una pantalla como la que se muestra en la figura B.29, pudiendo indicar cuando el usuario haya iniciado sesión en el móvil que vuelva a iniciarse la aplicación pulsando sobre el botón “Listo”. En el caso en el que el reloj no esté conectado al teléfono, se avisará con una pantalla como la que se muestra en la figura B.30, para que cuando se haya conectado el usuario el reloj al teléfono pueda pulsar sobre el botón “Reintentarlo”. Por último se podría mostrar una pantalla a modo de aviso indicando que la aplicación no es compatible con el modelo de reloj en el que se ha instalado, como aparece en la figura B.31 y no pudiendo utilizar la aplicación.



Figura B.28: Pantalla tras el inicio de sesión del usuario sin conexión al reloj inteligente. Figura B.29: Pantalla de aviso de usuario no identificado en el teléfono.



Figura B.30: Pantalla de aviso cuando el reloj no se encuentra conectado al móvil. Figura B.31: Pantalla de aviso cuando el modelo de reloj no es compatible con la aplicación.

B.3.1. Iniciar actividad

Para que un usuario inicie la medición de una actividad se debe pulsar sobre el botón “Comenzar actividad” situado en primer plano de la pantalla, como se muestra en la figura B.32. Antes de comenzar la actividad, el usuario debe elegir a qué categoría pertenece el tipo de actividad que va realizar de una lista que se muestra por pantalla mediante el toque sobre el ítem de la lista que más se ajuste, como se muestra en la figura B.33. Tras esto se debe visualizar una pantalla de aviso que confirme el comienzo de la medición como se muestra en la figura B.34. Una vez hecho esto, se debe visualizar el menú de medición, representado en la figura B.35, en la cual tenemos la opción de detener la actividad y de registrar etiquetas.



Figura B.32: Pantalla inicial para el inicio de la medición de una actividad.

Figura B.33: Pantalla de selección del tipo de actividad de una lista.



Figura B.34: Pantalla de aviso del comienzo de la medición para la actividad iniciada.

Figura B.35: Pantalla con un menú para parar la actividad y para registrar etiquetas.

B.3.2. Registrar etiquetas

Para poder registrar etiquetas en el transcurso de una actividad debemos pulsar sobre el botón “Registrar estado”, como se muestra en la figura B.36 para visualizar un listado con las siguientes etiquetas clasificadas en las siguientes categorías.

Estados

- Esperando.
- En viaje.
- Traslado.
- En destino.

Contextos

- Larga espera.
- Llego tarde.
- Sin información.
- Abarrotado.
- Sentado.
- Ruido.
- Calor.
- Frío.
- Atasco.
- Acompañado.
- Solo.
- Peligro.

Emociones

- Felicidad.
- Tristeza.
- Miedo.
- Ira.
- Sorpresa.
- Asco.

Sentimientos

- Nervioso.
- Tranquilo.

Cada categoría está separada en una pantalla dedicada con un listado que contiene las etiquetas anteriormente expuestas con iconos ilustrativos y un texto que los describe, se muestra un ejemplo de cómo se ve la pantalla cuando se pulsa sobre “Registrar estado” en la figura B.37. Para seleccionar una etiqueta se debe pulsar sobre el recuadro que lo engloba, de modo que se vea, cuando está seleccionada, un fondo más claro y un icono de verificación verde en el borde superior izquierdo, como se muestra en la figura B.38. En cada pantalla, se muestra al final un botón para confirmar el registro de etiquetas asociadas a la categoría en la que se sitúe el usuario, como se muestra en la figura B.39. Cuando se presiona este botón, se muestra durante unos segundos una pantalla explicativa para indicar al usuario qué categoría se va a mostrar a continuación. En el

momento en el que el usuario termina de pasar por todas las categorías de etiquetas, se vuelve a la pantalla mostrada en la figura B.35.



Figura B.36: Pantalla de menú de medición de actividad.



Figura B.37: Pantalla con el listado de etiquetas pertenecientes a la categoría Estados.



Figura B.38: Pantalla con una etiqueta seleccionada del listado.



Figura B.39: Botón al final del listado de etiquetas para confirmar la selección.

B.3.3. Detener una actividad en transcurso

Para detener una actividad que se encuentra iniciada en el reloj, se debe pulsar sobre el botón circular rojo indicado mediante un cuadrado en la parte inferior del menú de medición y etiquetado por “Parar medición”. Se muestra en la figura B.40 la forma en la que detener una actividad. Tras esto, se vuelve a la pantalla B.32.



Figura B.40: Pantalla de medición donde se muestra el botón para detener la medición de la actividad.

Apéndice C

Documentacion

Tras comentar todo el código realizado para las plataformas Android y WearOS, se ha usado un motor de generación de documentación de APIs en formato HTML llamado Dokka. La generación de estos HTML se pueden consultar tanto para Android (Rubia, 2022a) como para WearOS (Rubia, 2022b).

Bibliografía

- Ahn, J. W., Ku, Y. & Kim, H. C. (2019). A Novel Wearable EEG and ECG Recording System for Stress Assessment [s19091991[PII]]. *Sensors (Basel, Switzerland)*, 19(9), 1991. <https://doi.org/10.3390/s19091991>
- AwesomeRuby. (2022). Rodauth vs devise. <https://ruby.libhunt.com/compare-rodauth-vs-devise>
- Bank, W. (2021). Distribución porcentual de la población de España de 2004 a 2020, por rango de edad. <https://es.statista.com/estadisticas/501064/poblacion-de-espana-por-grupo-de-edad/>
- BOE. (2018). *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*. BOE. https://www.boe.es/diario_boe/txt.php?id=BOE-A-2018-16673
- Boletín Oficial del Estado. (2002). Reglamento (CE) nº 1107/2006 del Parlamento Europeo y del Consejo, de 5 de julio de 2006, sobre los derechos de las personas con discapacidad o movilidad reducida en el transporte aéreo. <https://www.boe.es/doue/2006/204/L00001-00009.pdf>
- Boletín Oficial del Estado. (2003a). Corrección de errores de la Directiva 2001/85/CE del Parlamento Europeo y del Consejo, de 20 de noviembre de 2001, relativa a las disposiciones especiales aplicables a los vehículos utilizados para el transporte de viajeros con más de ocho plazas además del asiento del conductor, y por la que se modifican las Directivas 70/156/CEE y 97/27/CE. <https://www.boe.es/doue/2003/125/L00014-00014.pdf>
- Boletín Oficial del Estado. (2003b). Ley 51/2003, de 2 de diciembre, de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad. <https://www.boe.es/eli/es/l/2003/12/02/51/con>
- Boletín Oficial del Estado. (2022). Ley 6/2022, de 31 de marzo, de modificación del Texto Refundido de la Ley General de derechos de las personas con discapacidad y de su inclusión social, aprobado por el Real Decreto Legislativo 1/2013, de 29 de noviembre, para establecer y regular la accesibilidad cognitiva y sus condiciones de exigencia y aplicación. <https://www.boe.es/eli/es/l/2022/03/31/6>
- Bundesministerium für Verkehr, B. u. S. (2017). *Mobilität in Deutschland 2017*. <http://www.mobilitaet-in-deutschland.de/>
- Burns, E. & Buttner, R. (2021). T wave. <https://litfl.com/t-wave-ecg-library/>
- Cadogan, M. & Buttner, R. (2022). P wave. <https://litfl.com/p-wave-ecg-library/>

- Cantwell, M., Caulfield, B. & O'Mahony, M. (2009). Examining the Factors that Impact Public Transport Commuting Satisfaction. *Journal of Public Transportation*, 12. <https://doi.org/10.5038/2375-0901.12.2.1>
- Chartjs. (2022). Chartjs. <https://github.com/chartjs/Chart.js>
- Chen, J., Abbod, M. & Shieh, J.-S. (2021). Pain and stress detection using wearable sensors and devices—A review. *Sensors*, 21(4), 1030.
- Cherry, K. (2021). The 6 types of basic emotions and their effect on human behavior. <https://www.verywellmind.com/an-overview-of-the-types-of-emotions-4163976>
- Community, F. (2022). Wear: Flutter Package. <https://pub.dev/packages/wear>
- Cong, R. & Winters, R. (2017). How does the xbox kinect work. <https://www.jameco.com/jameco/workshop/howitworks/xboxkinect.html>
- Consortio de Transporte Metropolitano del Área de Granada. (s.f.). Encuesta de Calidad. <https://ctagr.es/consorcio-de-transporte/calidad/encuesta-de-calidad>
- Consortio Regional Transportes de Madrid. (2004). Encuesta Domiciliaria de Movilidad en la Comunidad de Madrid. https://www.crtm.es/media/495601/edm2004_cuestionario.pdf
- Dang, A. T. (2020). MVC VS MVP VS MVVM. <https://levelup.gitconnected.com/mvc-vs-mvp-vs-mvvm-35e0d4b933b4>
- Declaración Universal de los Derechos Humanos* [Art. 13. Consultado el 15 de febrero de 2022]. (1948). Oficina del Alto Comisionado de las Naciones Unidas para los Derechos Humanos. https://www.ohchr.org/sites/default/files/UDHR/Documents/UDHR_Translations/spn.pdf
- Dependency injection in Android. (s.f.). <https://developer.android.com/training/dependency-injection>
- Developers, G. (2022). Messageclient. <https://developers.google.com/android/reference/com/google/android/gms/wearable/MessageClient>
- digitalAI. (s.f.). Agile velocity. <https://digital.ai/glossary/agile-velocity>
- Dupre, A., Vincent, S. & Iaizzo, P. A. (2005). Basic ECG Theory, Recordings, and Interpretation. En P. A. Iaizzo (Ed.), *Handbook of Cardiac Anatomy, Physiology, and Devices* (pp. 191-201). Humana Press. https://doi.org/10.1007/978-1-59259-835-9_15
- Epitech. (2022). Sueldos de un Desarrollador Junior: Salario de un developer junior según su experiencia. <https://www.epitech-it.es/sueldo-desarrollador-junior/>
- Feathericons. (2022). FeatherIcons. <https://github.com/feathericons/feather>
- Fitbit. (2022). How does my Fitbit device measure my EDA response? https://help.fitbit.com/articles/en_US/Help_article/2077.htm
- fluttercommunity.dev. (2022a). *Sensors_{plus} : FlutterPackage*. https://pub.dev/packages/sensors_plus
- fluttercommunity.dev. (2022b). Wear Flutter Package. <https://pub.dev/packages/wear>
- Gandomani, T. J., Faraji, H. & Radnejad, M. (2019). Planning Poker in cost estimation in Agile methods: Averaging Vs. Consensus. *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 066-071. <https://doi.org/10.1109/KBEI.2019.8734960>

- García Martínez, M. (2017). *Las emociones y el bienestar en las personas mayores* (Tesis doctoral).
- Glassdoor. (2022). Sueldo: Junior software developer. https://www.glassdoor.es/Sueldos/junior-software-developer-sueldo-SRCH_KO0,25.htm
- Google. (s.f.-a). Design principles WearOS. <https://developer.android.com/training/wearables/design/design-principles>
- Google. (s.f.-b). Enfoque de Prioridad de Kotlin en Android. <https://developer.android.com/kotlin/first?hl=es-419>
- Google. (s.f.-c). Why compose: Jetpack Compose. <https://developer.android.com/jetpack/compose/why-adopt>
- Google. (2014). Material Design Guidelines. <https://material.io/design/introduction>
- Google. (2022a). Accessibility Material Design. <https://material.io/archive/guidelines/usability/accessibility.html#accessibility-hierarchy-focus>
- Google. (2022b). Material icons. <https://fonts.google.com/icons>
- Google. (2022c). Save data in a local database using room nbsp; nbsp; android developers. <https://developer.android.com/training/data-storage/room>
- Gradl, S., Wirth, M., Richer, R., Rohleder, N. & Eskofier, B. M. (2019). An Overview of the Feasibility of Permanent, Real-Time, Unobtrusive Stress Measurement with Current Wearables. *Proceedings of the 13th EAI International Conference on Pervasive Computing Technologies for Healthcare*, 360-365. <https://doi.org/10.1145/3329189.3329233>
- Greene, S., Thapliyal, H. & Caban-Holt, A. (2016). A Survey of Affective Computing for Stress Detection: Evaluating technologies in stress detection for better health. *IEEE Consumer Electronics Magazine*, 5(4), 44-56. <https://doi.org/10.1109/MCE.2016.2590178>
- Guide to app architecture. (s.f.). <https://developer.android.com/topic/architecture>
- Halim, N., Sidek, K. & Mansor, H. (2017). Stress Recognition Using Photoplethysmogram Signal. *Indonesian Journal of Electrical Engineering and Computer Science*, 8, 495-501. <https://doi.org/10.11591/ijeecs.v8.i2.pp495-501>
- Hansson, E., Mattisson, K., Björk, J., Östergren, P.-O. & Jakobsson, K. (2011). Relationship between commuting and health outcomes in a cross-sectional population survey in southern Sweden. *BMC Public Health*, 11(1), 834. <https://doi.org/10.1186/1471-2458-11-834>
- Heo, S., Kwon, S. & Lee, J. (2021). Stress Detection With Single PPG Sensor by Orchestrating Multiple Denoising and Peak-Detecting Methods. *IEEE Access*, 9, 47777-47785. <https://doi.org/10.1109/ACCESS.2021.3060441>
- Hernando, D., Roca, S., Sancho, J., Alesanco, Á. & Bailón, R. (2018). Validation of the Apple Watch for Heart Rate Variability Measurements during Relax and Mental Stress in Healthy Subjects. *Sensors*, 18(8). <https://doi.org/10.3390/s18082619>
- Kirschbaum, C., Pirke, K.-M. & Hellhammer, D. H. (1993). The 'Trier Social Stress Test' – A Tool for Investigating Psychobiological Stress Responses in a Laboratory Setting. *Neuropsychobiology*, 28(1-2), 76-81. <https://doi.org/10.1159/000119004>

- Kluger, A. N. (1998). Commute variability and strain. *Journal of Organizational Behavior*, 19(2), 147-165. [https://doi.org/https://doi.org/10.1002/\(SICI\)1099-1379\(199803\)19:2<147::AID-JOB830>3.0.CO;2-Y](https://doi.org/https://doi.org/10.1002/(SICI)1099-1379(199803)19:2<147::AID-JOB830>3.0.CO;2-Y)
- Koffman, D., Weiner, R., Pfeiffer, A. & Chapman, S. (2010). Funding the public transportation needs of an aging population. *American Public Transportation Association*.
- Kohan, B. (2022). Guide to web application development. <https://www.comentum.com/guide-to-web-application-development.html>
- Koldijk, S., Sappelli, M., Verberne, S., Neerinx, M. & Kraaij, W. (2014). The SWELL Knowledge Work Dataset for Stress and User Modeling Research. <http://cs.ru.nl/~skoldijk/SWELL-KW/Dataset.html>
- Koldijk, S., Neerinx, M. A. & Kraaij, W. (2018). Detecting Work Stress in Offices by Combining Unobtrusive Sensors. *IEEE Transactions on Affective Computing*, 9(2), 227-239. <https://doi.org/10.1109/TAFFC.2016.2610975>
- Kumar, R. (2021). How does the mi fit bit measure stress? <https://www.quora.com/How-does-the-Mi-Fit-Bit-measure-stress>
- kuprel. (2019). PPG: Flutter Package. <https://pub.dev/packages/ppg>
- Lalani, S. (2022). MVC VS MVP VS MVVM : 10 differences you should know. <https://xperti.io/blogs/mvc-vs-mvp-vs-mvvm/>
- Larkin, J. (2021). QRS interval. <https://litfl.com/qrs-interval-ecg-library/>
- Li, F., Xu, P., Zheng, S., Chen, W., Yan, Y., Lu, S. & Liu, Z. (2018). Photoplethysmography based psychological stress detection with pulse rate variability feature differences and elastic net. *International Journal of Distributed Sensor Networks*, 14(9), 1550147718803298. <https://doi.org/10.1177/1550147718803298>
- Luján Henríquez, I. (2015). Aspectos socio-emocionales y salud mental y física percibidas en personas mayores. *Revista INFAD de Psicología. International Journal of Developmental and Educational Psychology.*, 2(1), 81-90. <https://doi.org/10.17060/ijodaep.2015.n1.v2.115>
- Manual para un Entorno Accesible [NIPO: 214-05-006-9]. (2005). Real Patronato sobre Discapacidad.
- Manual para un Entorno Accesible [NIPO: 214-05-006-9]. (2005). Real Patronato sobre Discapacidad.
- Matsuda, N., Murata, S., Torizawa, K., Isa, T., Ebina, A., Kondo, Y., Tsuboi, Y., Fukuta, A., Okumura, M., Shigemoto, C. & Ono, R. (2019). Association Between Public Transportation Use and Loneliness Among Urban Elderly People Who Stop Driving [PMID: 31321253]. *Gerontology and Geriatric Medicine*, 5, 2333721419851293. <https://doi.org/10.1177/2333721419851293>
- McCraty, R. & Shaffer, F. (2015). Heart Rate Variability: New Perspectives on Physiological Mechanisms, Assessment of Self-regulatory Capacity, and Health risk. *Glob Adv Health Med*, 4(1), 46-61.
- Mohan, P. M., Nagarajan, V. & Das, S. R. (2016). Stress measurement from wearable photoplethysmographic sensor using heart rate variability data. *2016 Internatio-*

- nal Conference on Communication and Signal Processing (ICCSP)*, 1141-1144. <https://doi.org/10.1109/ICCSP.2016.7754331>
- MongoDB. (s.f.). What is a non-relational database? <https://www.mongodb.com/databases/non-relational>
- Oracle. (s.f.). What is a relational database? <https://www.oracle.com/database/what-is-a-relational-database/>
- Paulvangentcom. (s.f.). Analysing smart ring data. https://github.com/paulvangentcom/heartrate_analysis_python/blob/master/examples/4_smartring_data/Analysing_Smart_Ring_Data.ipynb
- Paulvangentcom. (2019). Analysing a ppg signal. https://github.com/paulvangentcom/heartrate_analysis_python/blob/master/examples/1_regular_PPG/Analysing_a_PPG_signal.ipynb
- Paulvangentcom. (2021). Analysing smartwatch data. https://github.com/paulvangentcom/heartrate_analysis_python/blob/master/examples/3_smartwatch_data/Analysing_Smartwatch_Data.ipynb
- Pawlan, D. (2022). Relational vs. Non-Relational Database: Pros amp; Cons. <https://aloo.co/blog/relational-vs-non-relational-database-pros-cons>
- Picón, A. (2022). Android-desmitificando clean architecture. <https://medium.com/kotlin-dev-reactor/android-desmitificando-clean-architecture-2d0ed6aa2de6>
- PJ, S. (2021). Swell dataset analysis for stress prediction. <https://www.kaggle.com/code/shreyaspj/swell-dataset-analysis-for-stress-prediction>
- Polito, S. (2021). Build an API in rails with authentication. <https://stevepolito.design/blog/build-an-api-in-rails-with-authentication/>
- Ramírez Loeffler, P. & Rodríguez Vílchez, Z. (2012). *GESTIÓN Y CONTROL DEL ESTRÉS* (1.ª ed.). Conzepto Comunicación Creativa.
- Rubia, V. (2022a). Documentación de App Android. <https://victorrubia.github.io/TFG/doc/app/mobile/>
- Rubia, V. (2022b). Documentación de App WearOS. <https://victorrubia.github.io/TFG/doc/app/wear/>
- Saeed, S., Jhanjhi, N. Z., Naqvi, M. R. & Humayun, M. (2019). Analysis of Software Development Methodologies. *International Journal of Computing and Digital Systems*. <https://journal.uob.edu.bh/handle/123456789/3583>
- Safwany, O. (2018). Web Application Deployment: Meaning. <https://www.quora.com/What-does-the-deployment-of-a-web-application-mean-in-software-development-and-how-can-it-be-done>
- Sammer, G., Uhlmann, T., Unbehaun, W., Millionig, A., Mandl, B., Dangschat, J. & Mayr, R. (2012). Identification of Mobility-Impaired Persons and Analysis of Their Travel Behavior and Needs. *Transportation Research Record*, 2320(1), 46-54. <https://doi.org/10.3141/2320-06>
- Samsung. (s.f.). Measure your stress level with Samsung Health. <https://www.samsung.com/us/support/answer/ANS00080574/>
- Schwartz, A. (s.f.). The impact of small stresses in daily life. <https://www.mentalhelp.net/blogs/the-impact-of-small-stresses-in-daily-life/>

- Shaffer, F. & Ginsberg, J. P. (2017). An Overview of Heart Rate Variability Metrics and Norms. *Front Public Health*, 5, 258.
- Shekh-Yusef, R., Ahrens, D. & Bremer, S. (2015). HTTP Digest Access Authentication. <https://doi.org/10.17487/RFC7616>
- Shrestha, B. P., Millonig, A., Hounsell, N. B. & McDonald, M. (2017). Review of Public Transport Needs of Older People in European Context. *Journal of Population Ageing*, 10(4), 343-361. <https://doi.org/10.1007/s12062-016-9168-9>
- Smith, N., Beckhelling, J., Ivaldi, A., Kellard, K., Sandu, A. & Tarrant, C. (2006). Evidence base review on mobility: choices and barriers for different social groups. *Department for Transport*.
- StackOverflow. (2022). StackOverflow Developer Survey 2022. <https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-databases>
- Statista. (2021a). Medical wearable devices market revenue worldwide in 2021 and 2026, by region (in billion U.S. dollars). <https://www.statista.com/statistics/1289674/medical-wearables-market-size-by-region/>
- Statista. (2021b). Número de habitantes de 65 años o más en España de 2002 a 2021 (en millones). <https://es.statista.com/estadisticas/630678/poblacion-de-espana-mayor-de-65-anos/>
- Stults-Kolehmainen, M. A. & Sinha, R. (2014). The Effects of Stress on Physical Activity and Exercise. *Sports Medicine*, 44(1), 81-121. <https://doi.org/10.1007/s40279-013-0090-5>
- Subhani, A. R., Xia, L. & Malik, A. S. (s.f.). EEG Signals to Measure Mental Stress.
- Summers, C. (2020). How does the Galaxy Watch Measure Stress? <https://www.quora.com/How-does-the-Galaxy-Watch-measure-stress>
- Torrades, S. (2007). Estrés y burn out. Definición y prevención. *Offarm*, 26(10), 104-107. <https://www.elsevier.es/es-revista-offarm-4-articulo-estres-burn-out-definicion-prevencion-13112896>
- Twbs. (2022). Bootstrap. <https://github.com/twbs/bootstrap>
- United Nations Economic Commission for Europe. (2009). Integration and participation of older persons in society. https://unece.org/DAM/pau/_docs/age/2009/Policy_briefs/4-Policybrief_Participation_Eng.pdf
- Van Laerhoven, K. (2021). WESAD: Multimodal dataset for Wearable stress and affect detection. <https://www.eti.uni-siegen.de/ubicomp/home/datasets/icmi18/index.html.en?lang=en>
- van Gent, P. (2019). HeartPy - Python Heart Rate Analysis Toolkit.
- Wener, R. E., Evans, G. W., Phillips, D. & Nadler, N. (2003). Running for the 7:45: The effects of public transit improvements on commuter stress. *Transportation*, 30(2), 203-220. <https://doi.org/10.1023/A:1022516221808>
- What are containers? (2022). <https://www.netapp.com/devops-solutions/what-are-containers>
- Wijsman, J., Grundlehner, B., Penders, J. & Hermens, H. (2010). Trapezius Muscle EMG as Predictor of Mental Stress. *ACM Transactions on Embedded Computing Systems (TECS)*, 12, 155-163. <https://doi.org/10.1145/2485984.2485987>

- Winer, D. (2019). Android's commitment to Kotlin. <https://android-developers.googleblog.com/2019/12/androids-commitment-to-kotlin.html>
- Woodhead, W. (2018). Should you use material design? <https://medium.com/pilcro/should-you-use-material-design-bfb596a04bae>
- Yang, Y., Langellier, B. A., Stankov, I., Purtle, J., Nelson, K. L. & Roux, A. V. D. (2019). Examining the possible impact of daily transport on depression among older adults using an agent-based model [PMID: 29543502]. *Aging & Mental Health*, 23(6), 743-751. <https://doi.org/10.1080/13607863.2018.1450832>
- Yiğit, M. (2022). Say hello to jetpack compose and compare with XML. <https://blog.kotlin-academy.com/say-hello-to-jetpack-compose-and-compare-with-xml-6bc6053aec13>

