



UNIVERSIDAD DE GRANADA

TESIS DOCTORAL

Desarrollo de algoritmos para análisis de imágenes de sensores bioquímicos de respuesta óptica y su aplicación en dispositivos móviles

por

Antonio Marín Sánchez

Directores

Encarnación Castillo Morales

Alfonso Salinas Castillo

Universidad de Granada

Departamento de Electrónica y Tecnología de los Computadores

Programa de Doctorado de Tecnologías de la Información y
la Comunicación

Febrero 2024

TESIS DOCTORAL

Desarrollo de algoritmos para análisis de imágenes de sensores bioquímicos de respuesta óptica y su aplicación en dispositivos móviles

Directores

Encarnación Castillo Morales

Alfonso Salinas Castillo

Departamento de Electrónica y Tecnología de los Computadores

Departamento de Química Analítica

Editor: Universidad de Granada. Tesis Doctorales
Autor: Antonio Marín Sánchez
ISBN: 978-84-1195-290-3
URI: <https://hdl.handle.net/10481/91124>

Abstract

Faculty of Science

Department of Electronics and Computer Technology

Development of algorithms for image analysis of biochemical sensors with optical response and its application to mobile devices

Biochemical sensors with optical response are widely used in fields such as medicine, food safety and environment. The possibility of implementing portable systems that are able to perform in-situ analysis is a very relevant line of research in recent years. For this reason, the use of smartphones as reliable and low-cost detection devices has recently attracted attention. However, the influence of lighting conditions and the low sensitivity of low-cost devices remain fundamental aspects to improve in this field.

This thesis will focus on the development of image processing algorithms of biochemical sensors with optical response for implementation on mobile devices. One of the lines of this research is the use of microfluidic paper-based analytical devices (μ PAD) with Quick Response (QR) codes. This synergy will allow the implementation of detectors that use QR code detection algorithms to measure device signal. This thesis studies the use of error coding of a QR code for the detection of colorimetric signals in it. Additionally, a system is implemented for detecting glucose and lactate concentrations in a QR code-based biosensor that is capable of correcting changes in luminosity in the measurement. Another relevant objective is the study of chemical sensors with a fluorimetric response. For this reason, an image processing algorithm is developed for detection fluorescence intensity in a μ PAD that is implemented on a smartphone. Finally, a mobile application capable of detecting fluorescent QR codes developed for determining the cyanide concentration is shown.

In summary, this work highlights the promising possibilities of mobile devices as optical biosensor detection tools.

Resumen

Facultad de Ciencias

Departamento de Electrónica y Tecnología de los Computadores

Desarrollo de algoritmos para análisis de imagen de sensores bioquímicos de respuesta óptica y su aplicación en dispositivos móviles

Los sensores bioquímicos de respuesta óptica son ampliamente usados en campos como la medicina, la seguridad alimenticia y el medioambiente. La posibilidad de implementar sistemas portables que sean capaces de realizar análisis in-situ es una línea de investigación muy relevante en estos últimos años. Por esto, el uso de teléfonos inteligentes como dispositivos de detección fiables y de bajo coste ha tenido recientemente un creciente interés. Sin embargo, la influencia de las condiciones de luminosidad y la baja sensibilidad de los dispositivos de bajo coste siguen siendo aspectos fundamentales a mejorar en este campo.

En esta tesis se va a profundizar en el desarrollo de algoritmos de procesamiento de imágenes que contengan sensores bioquímicos de respuesta óptica para su implementación en dispositivos móviles. Una de las líneas de esta investigación es el uso de dispositivos microfluídicos analíticos en papel (μ PAD) basados en códigos *Quick Response* (QR). Esta sinergia va a hacer posible implementar detectores que usen los algoritmos de detección de códigos QR para medir la señal del dispositivo. En esta tesis se estudia el uso de la codificación de errores de un código QR para la detección de señales colorimétricas incluidas en este código QR. Además, se implementa un sistema para detectar la concentración de glucosa y lactato en un biosensor basado en códigos QR que es capaz de corregir los cambios de luminosidad en la medición. Otra línea de interés en esta tesis es el estudio de sensores con respuesta fluorimétrica. De esta forma, se ha desarrollado un algoritmo de procesamiento de imágenes para la detección de la intensidad μ PAD con señal de fluorescencia que se implementa en un teléfono inteligente. Finalmente se muestra la implementación de una aplicación móvil capaz de detectar códigos fluorescentes para la determinación de la concentración de cianuro.

En resumen, este trabajo pone en valor las prometedoras posibilidades de los dispositivos móviles como herramientas de detección de biosensores ópticos.

Agradecimientos

Aquí esta...Este es el resultado de 5 años de trabajo. Este es el resultado de: nuevas amistades, de picos de estrés y de felicidad, de una pandemia, de un cambio de aires, de dar clases, del paro, de dos parejas, de dos sobrinas y por supuesto... de mi investigación. Después de estos años esto es lo que más ganas tenía de escribir, tenía ganas de agradecer a toda la gente que me ha ayudado y a todas las personas que han sufrido los códigos QR conmigo alguna vez.

Lo primero de todo, Alfonso y Encarni, gracias por vuestra ayuda y vuestros consejos, gracias por permitirme trabajar en algo tan interesante y desafiante. Gracias por todo lo que me habéis enseñado. Gracias.

Inma, creo que esta tesis es tan tuya como mía. Todas las horas en el laboratorio y todos los buenos resultados tienen una parte de ti.

Fran y Alex, el trabajo de fin de master que hice con vosotros me abrió las puertas a esta oportunidad. No solo me llevo un gran aprendizaje de esa época sino amistades que espero que perduren muchos años. Quiero agradecer también a Diego su acogimiento en el departamento y que me diera siempre grandes consejos y oportunidades.

Mi familia ha sido mis raíces. La educación que he recibido de vosotros me han hecho afrontar esta etapa con ilusión, tranquilidad y con una actitud inquebrantable. Gracias Mama, Papa, Ana, Isa, Marta, Sofía y Aila. Gracias por todo.

Víctor, Piolín, Hálvaro, Nacho, Jorge, Luis si alguna vez os he hablado de códigos QR o de medir el PH de la piscina demasiado... Lo siento. Gracias por las risas y por no perdernos nunca. Guille te agradezco muchísimo el apoyo que me has dado siempre, espero poder devolvértelo alguna vez.

Tati, gracias por enseñarme tanto sobre mí mismo y por aguantar mis aprendizajes, siempre nos apoyaremos. Esther, gracias por tus sonrisas y tus abrazos son luz y quiero tenerlos siempre cerca.

Índice

Abstract	v
Resumen.....	vii
Agradecimientos	ixi
Lista de figuras.....	xv
Lista de tablas	xxi
Lista de acrónimos	xxiii
1. Introducción.....	1
1.1 Motivación.....	1
1.2 Objetivos.....	3
1.3 Estructura de la tesis.....	4
2. Estado del arte.....	7
2.1 Colorimetría y fluorescencia	7
2.1.1 El color.....	7
2.1.2 Medida del color.....	9
2.1.3 Espacios de color	11
2.1.4 Fluorescencia	15
2.2 Detectores de color y de imagen.....	17
2.2.1 Fotodetectores y colorímetros.....	17
2.2.2 Sensores de imagen. CCD y CMOS.....	17
2.2.3 Dispositivos de captación de imágenes.....	19
2.3 Procesamiento de imagen digital	22
2.3.1 Algoritmos para la adquisición	23
2.3.2 Algoritmos para el preprocesamiento.....	25
2.3.3 Algoritmos para la segmentación.....	28
2.3.4 Algoritmos para la descripción y reconocimiento.....	33
2.4 Códigos bidimensionales	34
2.4.1 Tipos de códigos 2D	34
2.4.2 Características del código QR.....	35
2.4.3 Diseño del símbolo QR	37
2.5 Códigos de corrección de errores Reed-Solomon	39
2.5.1 Definición del código Reed-Solomon	39
2.5.2 Construcción de un código Reed-Solomon	40
2.5.3 Codificación Reed-Solomon.....	41
2.5.4 Decodificación Reed-Solomon	42

2.6 Librería Zebra Crossing (Zxing) Project.....	44
2.6.1 Extracción de imagen	45
2.6.2 Detección de código QR.....	46
2.6.3 Decodificación de código QR	48
3. Sensor colorimétrico integrado en un código QR con corrección de errores	51
3.1 Introducción.....	51
3.2 Conceptualización del sistema.....	53
3.2.1 Esquema general de un sistema de diagnóstico basado en códigos QR..	53
3.2.2 QR de color como control y detección en luminancia	55
3.2.3 Análisis RGB de la imagen y detección de errores.....	58
3.2.4 Ajuste de la umbralización	60
3.3 Implementación software	61
3.3.1 Diseño de la aplicación	61
3.3.2 Arquitectura de la aplicación.....	63
3.3.3 Lectura, extracción de errores y comparación de patrones de error ...	65
3.4 Metodología experimental.....	66
3.4.1 Hardware y software.....	66
3.4.2 Procedimiento de medida experimental	67
3.5 Resultados y discusión	67
3.5.1 Evaluación de la capacidad de corrección y la disposición de errores..	68
3.5.2 Evaluación de la detección de errores con la distancia.....	70
3.5.3 Condiciones de luminosidad y dispositivos.....	71
3.5.4 Influencia de los errores en color. Análisis en luminancia	72
3.5.5 Detección en luminancia del código QR impreso del color del control.	75
3.5.6 Detección de errores del código QR en canales separados RGB.....	76
3.5.7 Comparativa de las soluciones	79
3.5.8 Calibración del umbral para el ajuste de la detección de analitos.....	80
3.5.9 Detección de patrones de diagnóstico con uno y varios analitos.	81
3.5.10 Influencia de las condiciones de luminosidad en la detección de patrones de diagnóstico.....	84
3.5.11 Análisis con otros dispositivos.	86
3.6 Conclusiones	86
4. Biosensor multianalito basado en un código QR colorimétrico	89
4.1 Introducción.....	89
4.2 Conceptualización del sistema.....	91
4.2.1 Esquema general del sistema	91

4.2.2	Diseño del código QR	93
4.2.3	Corrección W/B	94
4.2.4	Corrección R/C	96
4.3	Materiales y métodos.....	98
4.3.1	Reactivos y materiales	98
4.3.2	Hardware y software.....	99
4.3.3	Fabricación y preparación del código QR biosensor	99
4.3.4	Procedimiento de medida experimental.....	100
4.3.5	Calibración de las curvas de caracterización de glucosa y lactato....	101
4.3.6	Análisis de muestras biológicas.....	102
4.4	Implementación software.....	102
4.4.1	Pruebas de concepto en Matlab, algoritmos y métodos.	103
4.4.2	Implementación de la aplicación QR Analyser en Android.....	104
4.4.3	Aplicación final. Experiencia de usuario (UX).....	105
4.4.4	Arquitectura de un sistema en la nube.....	106
4.5	Resultados y discusión.....	107
4.5.1	Caracterización analítica del QR colorimétrico.....	107
4.5.2	Estudio de la curva de referencia para diferentes smartphones y condiciones de iluminación	109
4.5.3	Estudio del ángulo de medida y el tamaño del QR.....	112
4.5.4	Aplicación con muestras biológicas.....	113
4.5.5	Estudio de estabilidad y selectividad.....	115
4.6	Conclusiones y contribuciones	117
5.	Determinación de señales de fluorescencia ratiométricas en μPAD.....	121
5.1	Introducción	121
5.2	Conceptualización e implementación del sistema	121
5.2.1	Esquema general del sistema	122
5.2.2	Implementación de la aplicación final.....	123
5.3	Metodología experimental	125
5.3.1	Reactivos y materiales	125
5.3.2	Hardware y software.....	126
5.3.3	Fabricación y preparación del μ PAD.....	126
5.4	Resultados	126
5.4.1	Determinación de intensidad de fluorescencia variando la ISO.....	127
5.5	Conclusiones y contribuciones	129
6.	Código QR fluorescente para la detección de cianuro.....	131

6.1	Introducción.....	131
6.2	Conceptualización del sistema e implementación software	132
6.2.1	Diseño del QR fluorescente.....	133
6.2.2	Esquema general del sistema de detección de códigos QR fluorescentes 134	
6.2.3	Arquitectura de la aplicación y experiencia de usuario.....	136
6.2.4	Algoritmo de apilamiento	137
6.3	Materiales y métodos.....	138
6.3.1	Reactivos y materiales	139
6.3.2	Hardware y software.....	139
6.3.3	Preparación de las tintas	140
6.3.4	Fabricación del código QR.....	141
6.3.5	Funcionalización del código QR en papel y μ PAD para análisis de cianuro.....	142
6.3.6	Procedimiento de medida y extracción de la señal colorimétrica	142
6.4	Resultados.....	143
6.4.1	Caracterización de la recta de cianuro.....	143
6.4.2	Empleo del ratiométrico R a distintas ISOs.....	145
6.4.3	Detección del QR fluorescente	147
6.4.4	Evaluación del algoritmo de apilamiento.....	149
6.5	Conclusiones y contribuciones	152
7.	Conclusiones	155
	Referencias.....	159

Lista de figuras

Figura 2.1. a) Esquema de percepción del color por un observador [30]. b) Espectro visible [30].	8
Figura 2.2. a) Sección cruzada del ojo humano [30], b) sensibilidad espectral de los conos del sistema visual humano [30].	8
Figura 2.3. a) Circulo tonal con 8 colores, b) incremento del brillo, c) incremento de la intensidad [30].	10
Figura 2.4. Funciones de <i>color-matching</i> del estándar CIE 1931 [30].	11
Figura 2.5. Representación gráfica de los distintos espacios de color, a) modelo RGB [42], b) modelo CIE 1931 [43], c) modelo HSV [44], d) sistema de Munsell [45].	13
Figura 2.6. Arquitectura de un fluorómetro [51].	16
Figura 2.7. Diagrama de funcionamiento de un sensor de imagen monocromático [55].	18
Figura 2.8. Patrón de un filtro de Bayer.	18
Figura 2.9. Esquema de funcionamiento de un escáner plano [57].	20
Figura 2.10. Diagrama de bloques funcional de una cámara digital [61].	21
Figura 2.11. Esquema de las etapas del procesamiento de imagen digital y visión por computador.	22
Figura 2.12. Representación de la función <i>gamma</i> , para distintos valores de γ .	24
Figura 2.13. Funciones básicas de transformación de intensidad en una imagen [69].	25
Figura 2.14. Ejemplos de ajustes de brillo y contraste, a) imagen original, b) aumento del brillo, c) y d) aumento del contraste, d), e) y f) funciones de transformación.	26
Figura 2.15. Transformaciones de intensidad aplicadas sobre cada canal RGB por separado, a) imagen original, b) imagen de una transformación de histograma por canal.	26
Figura 2.16. a) Imagen de test, b) histograma entre 0 y 255.	27
Figura 2.17. Técnica de emparejamiento de histograma, a) imagen original y su histograma, b) imagen oscurecida y su histograma, c) imagen restaurada histograma.	28
Figura 2.18. Ejemplos de transformaciones geométricas aplicadas a una imagen [61].	28
Figura 2.19. Ejemplo de la identificación de umbrales, T , para la umbralización por histograma [69].	29

Figura 2.20. Ejemplo de umbralización por histograma aplicada al caso de una huella dactilar [69].	30
Figura 2.21. Tipos de formas de un elemento estructural, a) cruz, b) cuadrado, c) lineal, d) disco [69].	30
Figura 2.22. Ejemplo de aplicación de las operaciones de erosión y dilatación, a) conjunto A y elemento B, b) erosión, c) dilatación [69].	31
Figura 2.23. Aplicación de una operación de dilatación con un elemento estructural de cruz sobre el resultado de una detección de texto [69].	32
Figura 2.24. Ilustración del procedimiento de identificación de las componentes conectadas [69].	32
Figura 2.25. Evolución de las versiones de un símbolo QR [78].	35
Figura 2.26. Casos de símbolos QR dañados que pueden ser detectados gracias a su capacidad de corrección de errores [79].	36
Figura 2.27. Ejemplo de acoplamiento estructurado de 4 símbolos QR en uno [79].	37
Figura 2.28. Diseño del símbolo del código QR.	37
Figura 2.29. Trama de bit de la información de formato y tipos de patrones de máscara [81].	38
Figura 2.30. Trama de bit del símbolo QR y dirección de lectura de bits [81].	39
Figura 2.31. Definición de un código Reed-Solomon [82].	40
Figura 2.32. Arquitectura simplificada de la librería Zxing.	45
Figura 2.33. Diagrama de funcionamiento del detector, a) patrón de detección, b) tamaño del módulo, c) dimensión, d) patrón de alineamiento, e) transformación proyectiva.	47
Figura 2.34. Resultado de la detección de un símbolo QR (izquierda) en un objeto BitMatrix (derecha) la 'X' y ' ' representan el 1 y 0 respectivamente [89].	48
Figura 2.35. Esquema del proceso de corrección de errores en la librería Zxing.	49
Figura 3.1. Ejemplo de una hipotética paleta de color de un analito. La línea punteada roja marca el umbral de detección. El color que representa el control se marca separado.	54
Figura 3.2. Esquema general de un sistema de diagnóstico de sensores químico-ópticos basado en códigos QR.	55
Figura 3.3. Diseño de un QR impreso con el color del control de la paleta de color. Los módulos de borde rojo indican la posición de los módulos biosensores.	56

Figura 3.4. Esquema conceptual del funcionamiento de la solución basada en el análisis del QR en luminancia.....	57
Figura 3.5. a) Diseño de un QR sensor, b) detección del QR en el canal R, c) detección del QR en el canal G, d) detección del QR en el canal B.....	58
Figura 3.6. Esquema conceptual del funcionamiento de la solución 2 para el análisis del QR en canales separados RGB.....	59
Figura 3.7. Comparación entre el umbral de detección del QR y el umbral de diagnóstico del analito. Definición del porcentaje de cambio del umbral $U(\%)$	61
Figura 3.8. Capturas de pantalla de la aplicación implementada, a) lista de registros inicial, b) introducir un nuevo registro, c) tipos de códigos QR, d) registro preparado para análisis, e) resultados del análisis.....	62
Figura 3.9. Imagen de la disposición de los errores en el QR y su nivel de corrección de errores (FEC).....	68
Figura 3.10. a) Diseño del QR con disposición de errores distribuida, b) tabla de identificación de los errores.....	70
Figura 3.11. Diseño de un QR para el testeo de la detección de patrones de diagnóstico. Módulos con borde rojo = posiciones de control, módulos con borde verde = posiciones positivas y módulos con borde azul = posiciones negativas.....	82
Figura 3.12. Diseño de un QR para el análisis de dos analitos simultáneamente. Módulos de borde rojo = posiciones control 1, borde amarillo = posiciones control 2, borde verde = posiciones analito 1 y borde azul = posiciones analito 2.....	83
Figura 4.1. Esquema general del sistema [108].....	92
Figura 4.2. Diseño del código QR colorimétrico junto con su curva de referencia.....	94
Figura 4.3. Diagrama de bloques del analizador de códigos QR colorimétrico.	94
Figura 4.4. Evaluación de la corrección W/B para la curva de referencia bajo una temperatura de color de 7000 K.	96
Figura 4.5. Modelado matemático de la desviación entre la curva de referencia no estándar, rm , (curva roja) y la estándar, rs , (curva azul).	97
Figura 4.6. a) Accesorio de soporte del código QR, b) proceso de preparación de los módulos biosensores del código QR y elementos del accesorio de soporte.	100
Figura 4.7. Curvas de caracterización de glucosa y lactato.....	102
Figura 4.8. Flujo de trabajo de las pruebas de concepto realizadas con Matlab.	103

Figura 4.9. Capturas del diseño visual de la aplicación QR Analyser. a) lista de registros, b) captura del código QR, c) captura del código desde una foto, d) visualización de resultados.....	106
Figura 4.10. Diseño de una arquitectura de computación en la nube basado en nuestra aplicación QR Analyser.	107
Figura 4.11. Calibración de las curvas de detección de glucosa y lactato dentro del código QR colorimétrico medidas en condiciones ambientales estándar. Datos promediados para tres ensayos.	108
Figura 4.12. Curva de referencia impresa en el QR, a) promedio de las curvas de referencia con distintos dispositivos, b) curva de referencia medida bajo diferentes ISOs para el dispositivo Pocophone F1, c) curva de referencia medida con diferentes balances de blancos para el dispositivo Pocophone F1.	110
Figura 4.13. a) Efecto de la corrección R/C y W/B para cambios de balance de blancos, b) efecto de la corrección R/C y W/B para cambios de ISO..	111
Figura 4.14. Curvas de referencia, a) area de los modulos frente a la distancia para diferentes tamaños de QR, b) curva de referencia variando la distancia de medición, c) curva de referencia en función del area del módulo, d) curva de referencia bajo diferentes angulos de medición.....	113
Figura 4.15. Respuesta del código QR para: a) glucosa a 800 μM , b) lactato a 800 μM , c) 800 μM glucosa en presencia de 2000 μM de interferentes, d) 800 μM lactato en presencia de 2000 μM de interferentes.	116
Figura 4.16. Test de estabilidad para el código QR desarrollado almacenado a 4°C en la oscuridad durante 30 días.	117
Figura 5.1. Esquema general del sistema para la determinación del nivel de fluorescencia de μPAD fluorimétrico diseñado.	123
Figura 5.2. Capturas de pantalla de la aplicación QuenchingApp, a) lista de registro de análisis realizados, b) elección de la imagen, nombre y nivel de erosión, c) resultados obtenidos tras la detección de los sensores de la imagen.	124
Figura 5.3. Imágenes del μPAD captada a ISO 2500, a) antes de la detección, b) después de la detección y de aplicar la erosión.....	127
Figura 5.4. Intensidad de fluorescencia de las componentes detectadas en función del radio de erosión para diferentes ISOs.	127
Figura 5.5. Intensidad de fluorescencia, equivalente al nivel de gris, en función de la ISO empleada en la captura de imágenes.....	128
Figura 5.6. Intensidad de fluorescencia normalizada en función de la ISO.	129
Figura 6.1. Ejemplo de diseños de un símbolo QR fluorescente para detección de cianuro bajo, a) luz visible, b) luz ultravioleta.....	134

Figura 6.2. Esquema general del sistema de detección de códigos QR fluorescentes.....	135
Figura 6.3. Esquema de funcionamiento del algoritmo de apilamiento.	138
Figura 6.4. Procedimiento de fabricación del código QR fluorescente, a) esquema de las tintas en los cartuchos de impresión, b) impresión de los códigos QR, b) código QR iluminado con la lámpara UV.....	141
Figura 6.5. Imagen de la que se extrae la recta de caracterización de cianuro a ISO 400.....	143
Figura 6.6. Rectas de caracterización de cianuro en los canales RGB y nivel de gris.	144
Figura 6.7. Rectas de caracterización del cianuro del canal B en distintas ISOs.	145
Figura 6.8. Evolución de la intensidad del ratiométrico en función de la ISO para cada canal RGB y nivel de gris.....	145
Figura 6.9. Recta de caracterización ratiométrica del cianuro a distintas ISOs.	146
Figura 6.10. Imágenes del QR fluorescente con cianuro depositado en distintas etapas de la detección a ISO 400. Concentraciones de cianuro a)-c) 0.1 μM , d)-f) 2 μM	148
Figura 6.11. El triple de la desviación estándar y la pendiente de la recta para ISO, a) 400 y b) 800, LOD para ISO c) 400 y d) 800, SNR para ISO e) 400 y f) 800.....	150
Figura 6.12. Imagen de la recta de caracterización a ISO 200.	151
Figura 6.13. Evaluación del apilamiento para ISO 200, a) desviación estándar del papel y pendiente de la recta, b) LOD, c) SNR.....	151

Lista de tablas

Tabla 2.1. Comparativa de la capacidad de códigos bidimensionales [77]...	35
Tabla 2.2. Niveles de corrección de errores del código QR y porcentaje máximo de errores corregidos.....	36
Tabla 3.1. Número de errores máximos detectados según la posición del error en el QR y su nivel de corrección de errores (FEC).....	69
Tabla 3.2. Análisis de la capacidad de detección de errores cambiando la distancia de medición.....	71
Tabla 3.3. Evaluación de la capacidad de corrección de errores variando el balance de blancos (BW).....	72
Tabla 3.4. Evaluación de la capacidad de corrección de errores variando la ISO. NR = No reconocido.	72
Tabla 3.5. Comparación del número de errores y patrones de error específicos detectados para distintos colores.	73
Tabla 3.6. Comparación del número de errores y los patrones de error detectados aplicando el color solo en los módulos de error o al QR completo (NR: No Reconocido).....	76
Tabla 3.7. Análisis del número de errores (E) y las posiciones de error (P) detectados para cada color analizando el código QR en los canales RGB de forma independiente.....	78
Tabla 3.8. Análisis del número de errores (E) y las posiciones de error (P) detectados para cada color analizando el código QR en los canales RGB de forma independiente. NR = No reconocido.	79
Tabla 3.9. Comparativa de las soluciones en el rendimiento con color.	80
Tabla 3.10. Ajuste del incremento del umbral para mejorar la detección de los errores de color del QR.....	81
Tabla 3.11. Comparación de la detección de patrones de diagnóstico para distintos colores.	82
Tabla 3.12. Resultados de los análisis para un código QR multianalito.....	84
Tabla 3.13. Comparación de la detección de patrones de diagnóstico cambiando el balance de blancos de la cámara. OK = Detección correcta. X = Detección incorrecta.....	85
Tabla 3.14. Comparación de la detección de patrones de diagnóstico ISO de la cámara. OK = Detección correcta. X = Detección incorrecta.	85
Tabla 3.15. Comparación del rendimiento del sistema entre distintos dispositivos.....	86
Tabla 4.1. Parámetros de la caracterización analítica del QR colorimétrico para glucosa y lactato en condiciones estándar.	109

Tabla 4.2. Determinación de glucosa y lactato en muestras biológicas usando el dispositivo Huawei PSmart.	115
Tabla 4.3. Determinación de glucosa y lactato en muestras biológicas usando el dispositivo Pocophone F1.	115
Tabla 6.1. Resultados de la recta de caracterización en cada canal.	144
Tabla 6.2. Datos de la recta de caracterización ratiométrica a distintas ISOs.	147
Tabla 6.3. Parámetros de la recta de caracterización inversa a ISO 400.	147
Tabla 6.4. Datos de la determinación de la concentración de cianuro en el QR fluorescente.	149

Lista de acrónimos

CD	<i>Carbon Dots</i>
CCD	<i>Charge Couple Device</i>
CFA	<i>Color Filter Array</i>
CIE	<i>Commission Internationale de l'éclairage</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
DCT	<i>Discrete Cosine Transform</i>
μPAD	<i>Microfluidic Paper based Analytic Device</i>
PID	Procesamiento de Imagen Digital
POC	<i>Point Of Care</i>
QR	<i>Quick Response</i>
SVM	<i>Support Vector Machine</i>
Zxing	<i>Zebra Crossing</i>

Capítulo 1

Introducción

1.1 Motivación

Estos últimos años la sociedad atravesó momentos de gran dificultad derivados de una pandemia global. Este suceso tuvo consecuencias socio-económicas a nivel mundial e hizo que la sociedad se replanteara mejoras para afrontar futuras crisis inesperadas. Una de las consecuencias de esta pandemia fue la saturación de los centros hospitalarios que favoreció los contagios múltiples, la mortalidad de los pacientes y un estrés insostenible de los profesionales sanitarios [1]. Una de las causas de la saturación de los centros sanitarios es la falta de medios de diagnóstico, rápidos, portables, fiables y de bajo coste que eviten a los pacientes ir a los centros médicos. [2]. La necesidad de dispositivos sensores portables no solo afecta al diagnóstico médico, sino que también se extiende a otras áreas como la monitorización medioambiental [3] o la seguridad alimenticia [4,5]. La falta de infraestructuras y recursos en zonas rurales y en países más empobrecidos son un aliciente más para promover este tipo de dispositivos que permita “democratizar” el acceso a instrumental de calidad a bajo coste [2,4].

El desarrollo y la implementación de sistemas analíticos portables va de la mano de la evolución de las plataformas de “*Point of Care*” (POC). Estos sistemas de diagnóstico se emplean en el punto de cuidado del paciente sin necesidad de ir a un hospital o requerir de maquinaria costosa especializada. El desarrollo de sistemas POC requiere de la tecnología analítica adecuada para atender al paciente *in-situ*. Para este propósito existen dispositivos biosensores y microfluídicos que permiten el desarrollo de métodos de análisis rápidos, portables, precisos y económicos [2]. Los biosensores son dispositivos analíticos que traducen una respuesta biológica en una señal de salida detectable (óptica, eléctrica o magnética). Comprenden tres elementos: (1) el elemento biológico objeto del análisis, (2) un transductor que de cómo salida una señal medible y (3) un conversor que traduzca esta señal a una comprensible para el usuario [6]. Los biosensores pueden combinarse con los sistemas microfluídicos para dar lugar a dispositivos de dimensiones reducidas. La microfluídica se basa en el uso de las fuerzas de capilaridad de los materiales para guiar muestras líquidas por el dispositivo y realizar la reacción deseada [3].

Gracias a esta tecnología es posible desarrollar dispositivos de diferentes materiales como: papel, silicio, cristal y polímeros [7]. Para aplicaciones POC los dispositivos microfluídicos basados en papel (μ PADs) son realmente interesantes debido a su bajo coste, reproducibilidad y facilidad de uso [8].

Por otro lado, los teléfonos inteligentes o *smartphones* han alcanzado una alta capacidad de procesamiento con cámaras integradas de gran calidad y resolución [9]. Además, presentan distintas posibilidades de comunicación mediante Wifi, 4G/5G, comunicación de campo cercano (NFC), Bluetooth o sistema de posicionamiento global (GPS). Se pueden encontrar una gran variedad de marcas y modelos de *smartphones* en el mercado a diferentes precios facilitando la accesibilidad de estos dispositivos a personas de distintos perfiles socio-económicos. Combinando estos dispositivos con el desarrollo de aplicaciones móviles “*Apps*” es posible implementar sistemas analíticos especializados y portables de forma más rápida y económica que con el desarrollo de *hardware* especializado. Es por esto que los *smartphones* son serios candidatos para cimentar las bases del desarrollo de aplicaciones analíticas para biosensores portables.

El uso de los teléfonos inteligentes como transductores de biosensores ha sido un área de creciente investigación en esta última década [2–4,9–13]. Estos sistemas usan como elemento captador de señal la cámara integrada en el dispositivo. El diseño y la implementación de algoritmos de procesamiento de imagen son fundamentales para identificar el biosensor en la imagen y medir su señal [12]. En el caso de la medición de señales colorimétricas las condiciones de luminosidad ambiental son un factor determinante ya que la fuente de luz usada puede modificar drásticamente los resultados. En la literatura se presentan distintas soluciones, algunas de ellas pasan por la fabricación de accesorios para el teléfono en los que se generen una luminosidad controlada para medir la muestra [14,15]. Son especialmente útiles para la medición de señales fluorescentes, ya que permiten controlar la fuente de excitación y crear un ambiente oscuro que favorece la detección [16–19]. Otro enfoque para solucionar esta problemática es el diseño e implementación de algoritmos de calibración de luminosidad [20–23] que permiten evitar el uso de estos accesorios, que en algunos casos dificultan el manejo del dispositivo. Sin embargo, la implementación de los algoritmos de procesamiento de imagen puede resultar un reto considerable cuando se quiera analizar múltiples analitos en un mismo biosensor [20,21,24,25].

En esta tesis doctoral se abordará el uso de teléfonos inteligentes como detectores de μ PADs de respuesta colorimétrica y fluorimétrica. Se aportarán soluciones a través del desarrollo de algoritmos de procesamiento de imagen que sean capaces de detectar y medir estos dispositivos. Durante el proceso se profundizará en una línea de investigación más reciente en la que se integran los μ PADs en códigos *Quick Response* (QR) [26–29]. Esta integración permite aprovechar los algoritmos de detección de códigos bidimensionales para facilitar el proceso de identificación del dispositivo analítico. Una de las aportaciones más innovadoras que se presentará en este trabajo es el aprovechamiento de la capacidad de corrección de errores de los códigos QR como elemento de detección de analitos en el μ PAD. Por otro lado, se presentan algoritmos de calibración de luminosidad para poder determinar la concentración de un reactivo bajo distintas condiciones ambientales. También, se implementa un sistema para la medición de μ PADs de señal fluorescente y un sistema de detección de códigos QR fluorescentes. Finalmente, se testea un algoritmo de apilamiento de imágenes para mejorar la sensibilidad del dispositivo para la detección de señales fluorimétricas. Como se puede observar, todos los resultados derivados de este trabajo profundizan en el uso de teléfonos inteligentes para la identificación de dispositivos analíticos colorimétricos o fluorescentes. Con esto, se pretende aumentar el estado del conocimiento general en esta área presentando soluciones novedosas y prometedoras en este campo.

1.2 Objetivos

El objetivo principal de esta tesis doctoral es desarrollar e implementar algoritmos para la detección de señales colorimétricas y fluorimétricas para su uso en teléfonos inteligentes. Para ello se han definido cuatro objetivos de desarrollo de soluciones específicas:

1. Desarrollar un sistema de detección de señales colorimétricas para teléfonos inteligentes basado en el uso de códigos QR y corrección de errores.
2. Diseñar e implementar un sistema de determinación multianalito de glucosa y lactato para teléfonos inteligentes basado en códigos QR.
3. Desarrollar una solución para la detección de sensores fluorimétricos ratiométricos para teléfonos inteligentes.

4. Implementar una aplicación en teléfonos inteligentes para la determinación del cianuro a partir de señales fluorimétricas en códigos QR fluorescentes.

1.3 Estructura de la tesis

Este documento está organizado en 7 capítulos en los que se expondrán los resultados de la tesis para que el lector pueda entender y evaluar de forma apropiada los logros conseguidos. En el capítulo 1, en el que nos encontramos, se presenta el contexto en el que se enmarca esta tesis, se ponen en valor sus aportes y se enuncian sus objetivos principales.

En el capítulo 2 se hace una revisión de los conceptos teóricos necesarios para el correcto entendimiento del documento. Se hará una revisión de los fundamentos de la colorimetría y el fenómeno de la fluorescencia. También se describirán las principales características de los detectores de color y de imagen en los que se basa el trabajo de esta tesis, así como en los principales algoritmos de procesamiento de imagen utilizados en esta investigación. Finalmente, se hace una revisión de los conceptos básicos sobre códigos bidimensionales, los códigos de corrección de errores y la implementación de detectores de códigos QR basados en librerías externas.

En los capítulos 3 y 4 se presentan soluciones desarrolladas durante la investigación. Estos capítulos presentan una estructura similar, en la que se comienza describiendo el contexto en el que se enmarca la solución presentada. Posteriormente, se aborda la conceptualización del sistema, consideraciones de su implementación *software*, la metodología experimental sobre la que se apoya y la discusión de los resultados obtenidos. Finalmente, se expone una breve conclusión sobre la solución implementada.

En el capítulo 3 se presenta una solución para la detección de señales colorimétricas basada en códigos QR en teléfonos inteligentes. En este capítulo se enfoca el trabajo desde el punto de vista del empleo de la capacidad de corrección de errores del código para la detección de señales colorimétricas.

En el capítulo 4 se muestra el desarrollo de una aplicación móvil para teléfonos inteligentes capaz de determinar la concentración de glucosa y lactato en códigos QR colorimétricos. Se implementan algoritmos de calibración de

la luminosidad para detectar varios analitos simultáneamente bajo las mismas referencias de calibración.

En el capítulo 5 se aborda la implementación de un sistema de detección de sensores fluorimétricos ratiométricos en teléfonos inteligentes. En este capítulo se desarrolla un algoritmo de procesamiento de imagen para la detección y medición de sensores fluorescentes.

En el capítulo 6 se explora el desarrollo de una aplicación móvil para la determinación de señales fluorimétricas basadas en códigos QR. Además, se testea un algoritmo de apilamiento de imágenes para intentar mejorar la sensibilidad del dispositivo en este tipo de mediciones.

Finalmente, en el capítulo 7 se resumen las conclusiones obtenidas en la tesis planteando posibles trabajos futuros de cara a continuar investigando en esta línea.

Capítulo 2

Estado del arte

Para poder abordar el contenido expuesto en esta investigación es necesario aclarar algunos conceptos teóricos que faciliten su entendimiento. En este capítulo se estudiará la ciencia del color, la colorimetría, que nos dará una base para entender cómo se mide el color, y se presentará también el fenómeno de la fluorescencia. Se abordará además el proceso de adquisición del color mediante la explicación del funcionamiento de los detectores de color y de imagen. Posteriormente, se profundizará en las bases de los algoritmos de procesamiento de imagen para poder entender los algoritmos desarrollados en esta tesis. Por otro lado, se presentarán los códigos bidimensionales, haciendo hincapié en el funcionamiento de los códigos QR y se profundizará en las bases teóricas de la corrección de errores Reed Solomon, empleada en los códigos QR. Finalmente, se describirá el funcionamiento de la librería *Zebra Crossing Project*, que se empleó como base para la implementación de detectores de códigos QR.

2.1 Colorimetría y fluorescencia

El eje central de esta tesis doctoral es implementar sistemas que sean capaces de medir el color de sensores químicos colorimétricos de forma precisa, lo que supone un reto que necesita una solución. Para ello, se acudirá a la colorimetría, la ciencia que estudia los procesos y técnicas mediante las cuales se mide el color [30]. La medición de sensores fluorescentes es otro punto importante, por lo que se debe hacer un estudio del fenómeno de la fluorescencia [30] y cómo se cuantifica.

2.1.1 El color

El color es un fenómeno complejo que está directamente relacionado con la luz. Esto implica que la elección del método de medida, el observador y las condiciones de luminosidad ambientales son determinantes en el resultado. Se podría hacer un símil con la física cuántica, donde la propia observación del experimento afecta al resultado de este. De esta problemática surge la ciencia de la colorimetría, que propone teorías y métodos para medir el color de forma precisa y estandarizada.

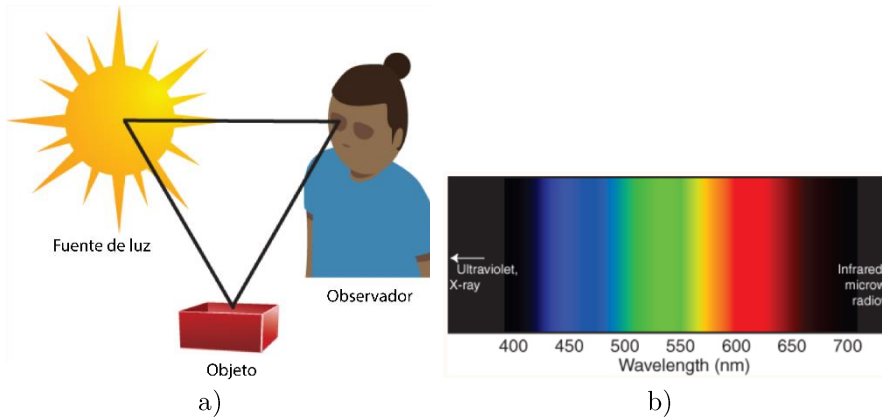


Figura 2.1. a) Esquema de percepción del color por un observador [30]. b) Espectro visible [30].

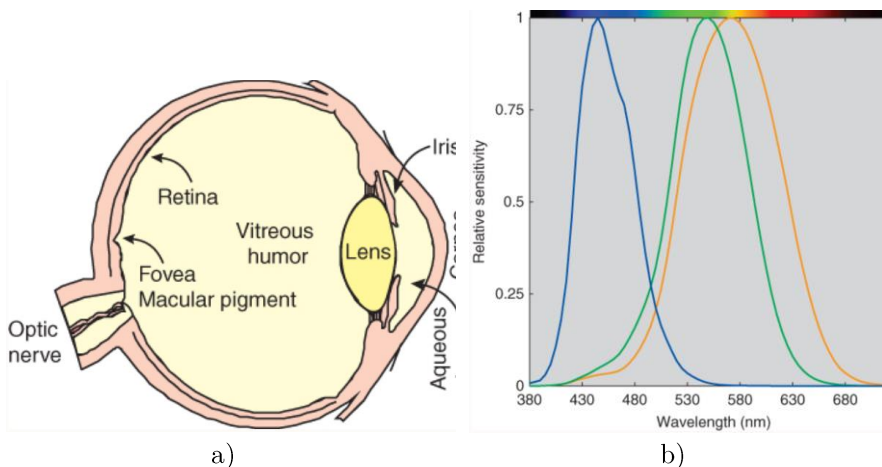


Figura 2.2. a) Sección cruzada del ojo humano [30], b) sensibilidad espectral de los conos del sistema visual humano [30].

Para poder medir el color hay que entender qué es, cómo se origina y cómo lo percibimos los seres vivos, en concreto los seres humanos. Como se mencionó anteriormente el color es un fenómeno físico íntimamente relacionado con la luz. El origen de este fenómeno depende de tres elementos, como se observa en la Figura 2.1a: la fuente de luz, el objeto observable y el observador [30]. La fuente de luz produce una onda del espectro electromagnético dentro del rango considerado como el espectro visible, el conjunto de longitudes de onda que nuestro sistema visual puede percibir. En la Figura 2.1b se puede ver que el rango de longitudes de onda que conforma el espectro visible va

desde 400 nm a 700 nm. Por encima de este rango se encuentra el rango ultravioleta y por debajo el infrarrojo, a los cuales no somos sensibles mediante la vista, pero sí mediante otros órganos como la piel [30]. La luz emitida incide sobre el objeto observable, el cual absorbe parte de las longitudes de onda y refleja otras. La luz reflejada por el objeto es percibida por el observador que según su sistema visual interpreta las longitudes de onda reflejadas por el objeto dándole una forma y un color.

Este proceso está estrechamente relacionado con el funcionamiento de nuestro sistema visual. En la Figura 2.2a se puede ver una sección cruzada del ojo. La luz entra en nuestro globo ocular e incide sobre la retina. La córnea y la lente son las encargadas de enfocar la imagen cambiando su forma. El iris modula la cantidad de luz que entra en el ojo y la retina contiene los fotorreceptores que absorben una porción de la luz generando una señal en el nervio óptico que es interpretada por nuestro cerebro [30]. En la retina se encuentran dos tipos de fotorreceptores: los bastones, con una alta sensibilidad que detectan objetos como sombras de grises, y los conos, con una menor sensibilidad son los encargados de detectar el color. Los seres humanos tenemos tres tipos de conos que responden a rangos diferentes de longitud de onda. En la Figura 2.2b se puede ver la sensibilidad espectral de los tres conos y cómo corresponden a diferentes regiones de longitud de onda larga, L (rojo), media, M (verde) y corta, S (azul) [30]. En la Figura 2.2b las curvas azul, verde y naranja, corresponden con los conos S, M y L, respectivamente.

2.1.2 Medida del color

Una vez entendido cómo percibimos los colores es importante plantearse de qué forma podemos describirlos. Tradicionalmente se han usado nombres y adjetivos para nombrar a los colores de forma universal: blanco, negro, rojo, amarillo, verde, azul, etc. [31]. Sin embargo, esta nomenclatura no es suficiente para describir todos los colores de forma precisa. Por ello surge la necesidad de establecer escalas de medida para organizar y especificar el mayor número de colores posibles. Siguiendo esta idea se han definido distintos sistemas para medir el color. En [32] se propuso un sistema basado en tres atributos fisiológicos independientes para describir el color: el tono, el brillo y la intensidad cromática. El tono se puede definir como la longitud de onda dominante del espectro de radiación de un color. En muchas ocasiones los colores se pueden organizar en un círculo tonal, como se puede ver en la Figura 2.3a. El brillo se define como la variación de luminosidad del color. En la Figura 2.3b se puede ver un ejemplo de tres incrementos de brillo (de izquierda a

derecha) tomando tres tonos distintos (blanco, verde y amarillo). Finalmente, la intensidad cromática se puede describir como la carga cromática que tiene un color. La Figura 2.3c muestra un ejemplo de incremento de la intensidad cromática de un rosa (aumento de izquierda a derecha) a partir de tres colores base (blanco, gris, negro).

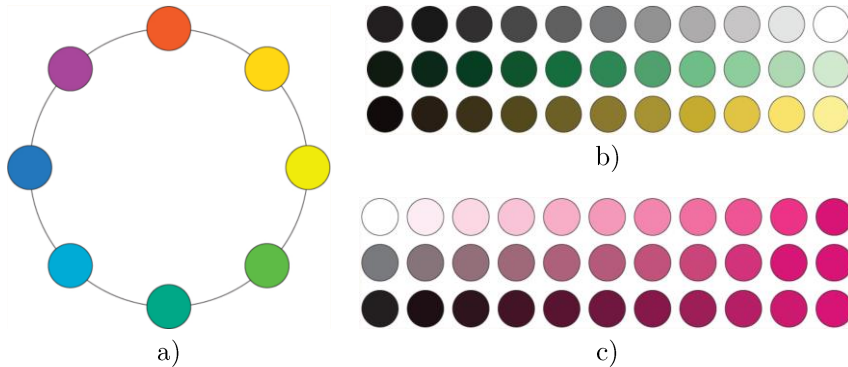


Figura 2.3. a) Circulo tonal con 8 colores, b) incremento del brillo, c) incremento de la intensidad [30].

En este punto aún queda por resolver la pregunta de: ¿cómo se mide el color? Si bien somos capaces de describirlo de forma más precisa, aún no se ha establecido un método de medida. Una de los procedimientos más sencillos para medir el color es mediante un método de *Visual Color Matching* o emparejamiento de color [30,33]. Éste consiste en comparar un color bajo una determinada iluminación con otro que se pueda reproducir con un sistema de tres fuentes de luz en tres tonos: rojo, azul y verde, (*Red, Blue y Green, RGB*). Variando cada una de estas fuentes de luz se comparan ambos colores y se determina, visualmente, si son similares. De esta forma se construye un sistema de tres estímulos con los que se define numéricamente un color [30,33]. Este sistema tiene el problema de que es dependiente del observador que compara ambos colores. Para paliar esto la Comisión Internacional de Iluminación (*Commission Internationale de l'éclairage, CIE*) estableció un sistema de medida estandarizado, el CIE 1931, que define unas curvas de sensibilidad espectral estándar, a partir de un grupo observadores [34,35]. En la Figura 2.4 se pueden ver las funciones de *color-matching* conocidas como las curvas de sensibilidad espectral de un observador estándar en el CIE 1931. Se mencionó anteriormente que la fuente de luz usada es determinante en la medición del color, por lo que la CIE definió varios estándares de iluminación con una letra

o un número. Podemos encontrar, el iluminante A, luz incandescente, el iluminante B, luz del sol directa, el iluminante C, promedio de la luz del día o el iluminante D, que representa las fases de la luz del día. Uno de los iluminantes más comunes en aplicaciones colorimétricas es el D65, ya que representa una fase de la luz natural del día con una temperatura de color de 6504K [36].

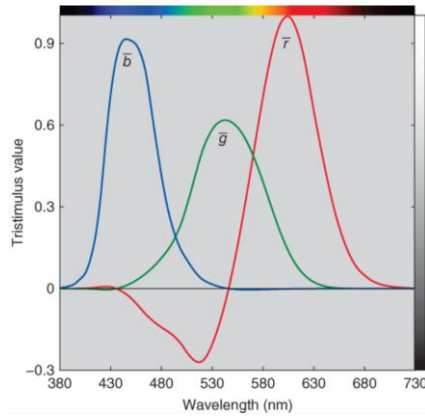


Figura 2.4. Funciones de *color-matching* del estándar CIE 1931 [30].

2.1.3 Espacios de color

Como resultado del estudio de la colorimetría se han definido distintos modelos o espacios de color con los que se pueden describir los colores. Un espacio de color es un modelo matemático abstracto que describe los colores como una secuencia de números que depende de cada modelo. Cada uno de estos espacios surge de una necesidad específica y se aplican dependiendo del contexto. Dado que los algoritmos desarrollados en este trabajo serán aplicados en sensores químicos colorimétricos, se van a analizar estos espacios de color atendiendo a sus ventajas y desventajas desde este punto de vista.

RGB

El espacio de color RGB es el más común en dispositivos de adquisición, ya que trabaja de forma similar al sistema visual humano. Define cada color como un sistema aditivo de tres componentes rojo, verde y azul. Cada componente varía de 0 a 2^{n-1} , donde n es el número de bits que se utilizan para codificar dichos valores [36]. Por ejemplo, para 8 bits los valores varían de 0 a 255. Es un sistema dependiente del dispositivo y de cada fabricante lo que provoca que un color de apariencia similar se codifique con valores de RGB diferentes en cada dispositivo. Se puede representar gráficamente como un

cubo, como se puede ver en la Figura 2.5a. Un inconveniente de este modelo es que tiene una alta dependencia con el dispositivo [12]. Este espacio de color ha sido usado en la literatura para la cuantificación del color en sensores químicos [37,38]. También se hace uso de este sistema en la literatura derivando la intensidad de color, Y , descrita mediante la ecuación (1), [39].

$$Y = \sqrt{R^2 + G^2 + B^2} \quad (1)$$

CIE 1931

Ya se mencionó el CIE 1931 como un modelo estándar que no dependía del dispositivo. Tiene la ventaja de que cubre todo el rango de colores percibidos por el ser humano pero su representación no es perceptualmente uniforme, lo que provoca dificultades a la hora de comparar colores [40]. Este modelo también es conocido como CIE XYZ debido a que cuenta con tres componentes X , Y , Z para representar los colores, donde la coordenada Y es equivalente a la luminancia. Estas componentes se relacionan con el modelo RGB como se muestra en la ecuación (2). La matriz de coeficientes es dependiente del iluminante empleado en cada aplicación. Por lo tanto, es un modelo difícil de escalar a distintas iluminaciones ambiente. Para representar gráficamente este modelo suele usarse un diagrama de cromaticidad, que puede verse en la Figura 2.5b. Pero para ello han de normalizarse las coordenadas usando las ecuaciones (3), (4) y (5). Se puede ver que la coordenada z es dependiente de x e y , lo que implica que un color puede representarse perfectamente en este sistema con dos coordenadas x e y junto a una Y que equivale a la luminancia.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4125 & 0.3576 & 0.1804 \\ 0.2127 & 0.7156 & 0.0722 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

$$x = \frac{X}{X + Y + Z} \quad (3)$$

$$y = \frac{Y}{X + Y + Z} \quad (4)$$

$$z = \frac{Z}{X + Y + Z} = 1 - x - y \quad (5)$$

CIELAB

El espacio CIELAB nace para dar solución al problema de falta de percepción uniforme de los colores que se da en el estándar CIE 1931. Con este sistema se obtiene un diagrama de cromaticidad perceptualmente uniforme, lo que lo hace muy interesante para comparar colores [41]. En este sistema la

coordenada L representa la luminancia con un rango de 0 a 100. Las coordenadas a^* y b^* representan el cambio de color desde el magenta al verde y del amarillo al cian, respectivamente, y varían de -60 a 60. Al igual que el sistema XYZ , este es independiente del dispositivo, pero requiere de un paso intermedio para obtener sus coordenadas, lo que equivale a un mayor consumo de recursos. Las ecuaciones (6), (7) y (8) muestran el cálculo de las coordenadas L , a^* y b^* a partir de XYZ .

$$L = \begin{cases} 116 \left(\frac{Y}{Y_n}\right)^{1/3} - 16, & \text{si } \frac{Y}{Y_n} > 0.008856 \\ 903.3 \frac{Y}{Y_n}, & \text{si } \frac{Y}{Y_n} \leq 0.008856 \end{cases} \quad (6)$$

$$a^* = 500 \left[\left(\frac{X}{X_n}\right)^{1/3} - \left(\frac{Y}{Y_n}\right)^{1/3} \right] \quad (7)$$

$$b^* = 200 \left[\left(\frac{Y}{Y_n}\right)^{1/3} - \left(\frac{Z}{Z_n}\right)^{1/3} \right] \quad (8)$$

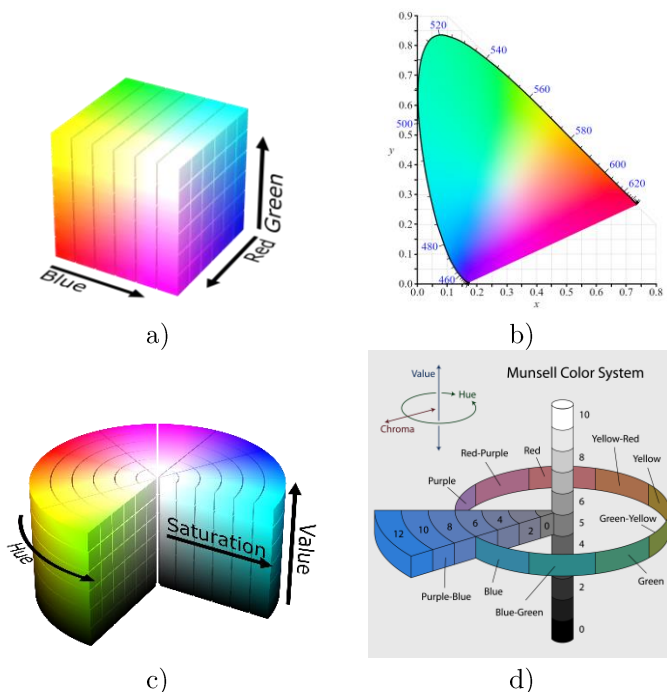


Figura 2.5. Representación gráfica de los distintos espacios de color, a) modelo RGB [42], b) modelo CIE 1931 [43], c) modelo HSV [44], d) sistema de Munsell [45].

HSV

El espacio HSV (*Hue, Saturation, Value*) es un modelo de color basado en el tono, esto quiere decir que existe un parámetro único para definir el tono de un color. Se ha demostrado que el tono es un parámetro muy robusto ante cambios de luminosidad ambiente, lo que resulta en una gran ventaja para identificar un color [46]. El tono suele variar entre 0 y 360°. El parámetro de saturación nos define la intensidad cromática del color y el valor hace referencia al brillo, ambos se miden entre 0 y 1 (Figura 2.5c). Otra ventaja de este espacio es que es perceptualmente uniforme, por ello es muy usado por diseñadores y artistas en programas de diseño. Una de sus principales desventajas es que deriva directamente del modelo RGB, como reflejan las ecuaciones (9), (10), (11) y (12), [47], lo que lo convierte en un modelo dependiente del dispositivo con el que se mida.

$$C = \max R, G, B - \min(R, G, B) \quad (9)$$

$$H = 60^\circ \times \begin{cases} \left(\frac{G-B}{C}\right)/6, \text{ si } \max R, G, B = R \\ \left(\frac{B-R}{C} + 2\right)/6, \text{ si } \max R, G, B = G \\ \left(\frac{R-G}{C} + 4\right)/6, \text{ si } \max R, G, B = B \end{cases} \quad (10)$$

$$S = \begin{cases} 0, \text{ si } C = 0 \\ \frac{C}{V}, \text{ otro caso} \end{cases} \quad (11)$$

$$V = \max(R, G, B) \quad (12)$$

Sistema de color de Munsell

La escala de Munsell fue publicada en 1905 y consiste en 1500 muestras de color ordenadas de forma sistemática [48]. Es un espacio de color basado en tono, H , que dividía el círculo tonal en cinco tonos principales (rojo, amarillo, verde, azul y morado) aunque más tarde fueron subdividiéndose hasta identificar 100 tonos distintos. Este espacio de color cuenta con dos componentes más, saturación, C , y brillo, V . Una de las principales características de este sistema es que tanto la saturación como el brillo presentan intervalos perceptuales similares a los del ser humano [49]. Este sistema es idóneo si se quiere tener una representación perceptual similar a la del ser humano, pero no es útil para detectores de color ni cámaras digitales.

CMYK

El espacio CMYK es un modelo sustractivo que es principalmente usado en impresión [50]. Esto es debido a que al imprimir ya contamos con un fondo de color (por ejemplo, folio en blanco) a partir del cual se van mezclando tintas que eliminan las longitudes de onda correspondientes para que se refleje un color determinado. El modelo CMYK usa como colores básicos el cian, el magenta y el amarillo. La coordenada K hace referencia al color negro que se añade para obtener un negro más puro en impresión. Este sistema puede obtenerse a partir del modelo RGB con la ecuación (13).

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (13)$$

YUV

El modelo YUV es un sistema que empezó a usarse en televisión para facilitar la compresión de información sin perder excesiva calidad en el color, y que actualmente sigue usándose en algunas aplicaciones móviles para tomar imágenes. El modelo cuantifica de forma separada la luminancia, Y , y las crominancias UV . Las coordenadas YUV se pueden obtener a partir de RGB, mediante la ecuación (14), y viceversa, usando la ecuación (15).

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (14)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.140 \\ 1 & -0.396 & -0.581 \\ 1 & 2.029 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \quad (15)$$

2.1.4 Fluorescencia

Hasta ahora se ha expuesto qué es el color, así como los procedimientos y herramientas matemáticas para medirlo. A continuación, se va a presentar otro fenómeno muy utilizado para diseñar sensores químicos de respuesta óptica, la fluorescencia. La fluorescencia es un tipo de fenómeno luminiscente, que se produce cuando un material emite luz sin ningún tipo de excitación térmica [30]. Dentro de estos fenómenos se encuentra la fotoluminiscencia, la cual se manifiesta cuando un material es excitado por una fuente de luz y el material emite luz con una longitud de onda diferente a la excitación original. La fluorescencia es un caso particular en el que la emisión de luz se da al mismo tiempo que la excitación inicial [33].

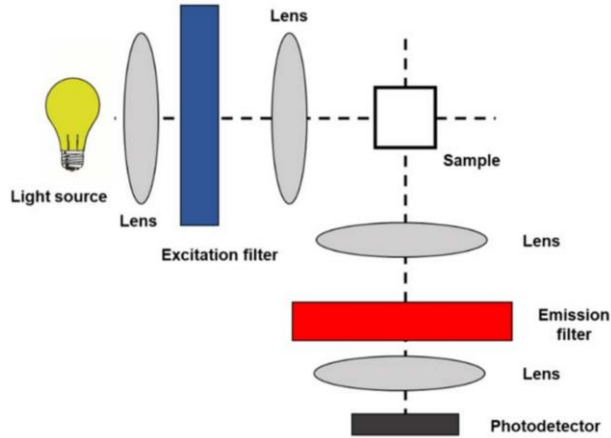


Figura 2.6. Arquitectura de un fluorómetro [51].

Para medir la fluorescencia de los materiales y sustancias se han usado tradicionalmente dos tipos de equipos de medición: los espectrómetros y los fluorómetros basados en filtros [51]. Los espectrómetros son equipos capaces de medir la radiación lumínica reflejada o absorbida por un objeto o sustancia en diferentes longitudes de onda. Estos equipos constan de distintos elementos ópticos y electrónicos como: fuentes de luz, monocromadores y fotodetectores donde algunos de ellos como los monocromadores constan de componentes opto-mecánicos como colimadores, espejos focales, rejillas difractoras, etc. Estos equipos son difíciles de convertir en una instrumentación portable debido principalmente a que su miniaturización es complicada sin perder resolución y sensibilidad [51]. Como contraparte están los fluorómetros, que permiten medir la fluorescencia a una longitud de onda fija. En la Figura 2.6 se puede ver un esquema de la arquitectura de un fluorómetro. Se puede apreciar cómo éste está basado en una fuente de luz dirigida hacia una muestra que es focalizada y filtrada con un conjunto de lentes y filtros. Al ser excitada la muestra con la fuente de luz, ésta emite luz en otra longitud de onda distinta que es dirigida a un fotodetector mediante un conjunto de lentes y filtros. Los sistemas portables están basados principalmente en este tipo de arquitectura debido a su sencillez y su facilidad de miniaturización. Este tipo de instrumentación portable busca tener un tamaño compacto y un bajo consumo. Actualmente los *smartphones* son unos dispositivos prometedores para este tipo de aplicaciones debido a su reducido tamaño, capacidad de procesamiento y fácil acceso a la población. Sin embargo, para la medición de sensores fluorescentes se necesitan unas condiciones de baja luminosidad que en ocasiones requieren de accesorios para el *smartphone* que aumentan su tamaño y dificultan su manipulación.

2.2 Detectores de color y de imagen

Una vez presentadas las bases de la colorimetría es hora de preguntarse cómo se puede captar la luz reflejada por un objeto para transformarla en una señal eléctrica interpretable por alguno de los modelos de color presentados previamente. Si se quisiera medir exclusivamente el color de un objeto se podría acudir a fotodetectores, espectrómetros o colorímetros que captan únicamente la radiación visible reflejada por los objetos en un punto concreto del espacio. Sin embargo, cuando se quiere captar información de la dimensión espacial del objeto se debería optar por usar dispositivos de imagen como cámaras digitales, escáneres y *smartphones*. Estos dispositivos cuentan con sensores de imagen que son capaces de captar la radiación en distintos puntos del espacio creando una imagen.

2.2.1 Fotodetectores y colorímetros

La base de la medición de la radiación visible fue el descubrimiento del efecto fotoeléctrico por Albert Einstein en 1905, [52] a partir de ese momento se dio la posibilidad de crear dispositivos capaces de traducir una señal luminosa en una eléctrica, como los fotodiodos. Un fotodiodo es un fotodetector basado en una unión PN de semiconductores que es capaz de transformar la luz recibida en voltaje o corriente dependiendo de su modo de operación. Se pueden encontrar ejemplos en la literatura donde este tipo de detectores son usados para medir el color de sensores químicos [13,53]. Otra forma de medir el color es a través de un espectrómetro, este es capaz obtener el espectro en frecuencia de una onda electromagnética como puede ser la luz. Esto puede ser muy útil para determinar el espectro de reflexión de un objeto, lo que determinaría cuál es la radiación que refleja, dando origen al color que percibimos. Por otro lado, están los colorímetros que son capaces de darnos las medidas de un color en distintos espacios de color de forma muy precisa. En investigación suelen usarse este tipo de colorímetros para validar modelos o nuevos dispositivos de medición [54].

2.2.2 Sensores de imagen. CCD y CMOS

Se pueden definir los sensores de imagen como un dispositivo compuesto por una matriz de fotodetectores dos dimensiones (2D) que es capaz de captar la luz procedente de distintos puntos del espacio para realizar una representación en 2D del mundo real. Estos sensores son la base del funcionamiento de las cámaras digitales, los escáneres y los *smartphone*.

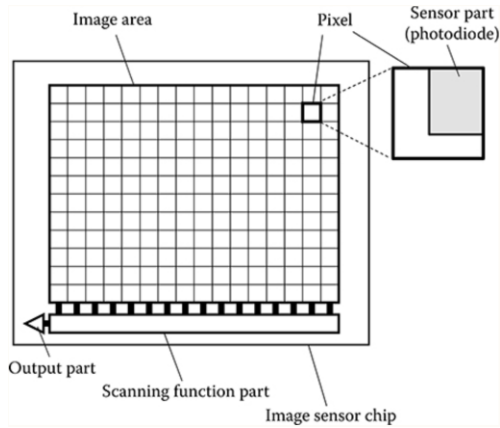


Figura 2.7. Diagrama de funcionamiento de un sensor de imagen monocromático [55].

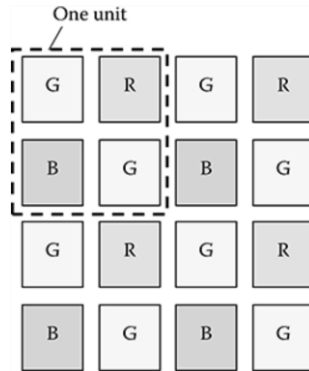


Figura 2.8. Patrón de un filtro de Bayer.

Las principales tecnologías de sensores de imagen que se pueden encontrar son CCD (*Charge-Coupled Devices*) y CMOS (*Complementary Metal-Oxide-Semiconductor*). Ambas tecnologías usan un mosaico en dos dimensiones en el que cada cuadrado del mosaico es un fotodiodo que capta la luz y la transforma a una señal eléctrica, este es conocido como píxel. Los sensores de imagen también cuentan con algún componente que realiza el escaneo de los píxeles e identifica sus coordenadas dentro de la imagen [55]. En la Figura 2.7 se muestra un ejemplo del diagrama de funcionamiento de un sensor de imagen monocromático. La resolución del sensor va determinada por el número de píxeles que tenga la matriz, cuanto mayor sea este, mayor calidad de la imagen se podrá conseguir. Para tomar imágenes en color se usa una agrupación de filtros de color (*Color Filter Array, CFA*) que se coloca encima del sensor para filtrar la luz incidente. Uno de los patrones de CFA más usados

es el de Bayer, que puede verse en la Figura 2.8. Este patrón consiste en colocar filtros de rojo (R), verde (G) y azul (B) en cuatro pixeles adyacentes que forman un pixel de color. El patrón da más peso al G en la conformación del pixel ya que el ser humano detecta mejor la luz en el rango del verde.

Los sensores CCD y CMOS basan su funcionamiento en la acumulación de carga en cada pixel de forma proporcional a la intensidad lumínica, sin embargo, presentan ciertas diferencias que se van a explicar a continuación. En los CCD, tras la acumulación de carga en el pixel, se realiza la transferencia de carga de cada pixel secuencialmente a un registro de desplazamiento que lo lleva fuera del chip a un convertor de carga a voltaje y posteriormente a un convertor analógico-digital (A/D) [56]. En cambio, en los sensores CMOS la conversión de carga a voltaje tiene lugar en cada pixel, lo que permite que las cargas pueden ser procesadas al mismo tiempo y de este modo se reducen los tiempos de respuesta, el consumo y el tamaño [12]. Por lo general los CCD presentan mayor calidad y menos ruido, sin embargo, los CMOS han tenido mayor penetración en el mercado ya que su fabricación se puede estandarizar como la de cualquier semiconductor.

2.2.3 Dispositivos de captación de imágenes

Dentro de los equipos de captación de imagen se pueden encontrar escáneres, cámaras digitales o los *smartphones*. Cada uno de ellos presenta unas características diferentes que se analizarán a continuación.

Escáneres

Un escáner es un dispositivo que permite digitalizar imágenes o documentos impresos que pueden ser posteriormente manipulados e interpretados por un ordenador. Un escáner consiste en un cabezal que recorre la pantalla del escáner emitiendo una luz blanca que se refleja en el objeto escaneado y es guiada mediante un conjunto de espejos hasta un sensor de imagen CCD, en la Figura 2.9 se puede ver un ejemplo. Los parámetros que determinan la calidad de un escáner son la resolución, la profundidad de color y el rango de densidad. La resolución, es el número de pixeles que están presentes en una imagen, y se mide en puntos por pulgada. La profundidad de color determina el rango de colores que puede representar un escáner; normalmente se encuentra entre 24 y 48 bits. El rango de densidad indica si el escáner es capaz de reproducir sombras y brillos de una pasada. Los escáneres presentan como principal ventaja que no son afectados por las condiciones de luminosidad

ambiente, lo que los hace muy precisos para medir el color de forma estándar. Entre sus principales desventajas están la limitación de que solo permiten escanear objetos planos, debido a la propia estructura del aparato. También se tendría la desventaja de que la mayoría de estos aparatos no son fácilmente portables y requieren de un procesamiento posterior de la imagen para obtener información relevante de forma automática.

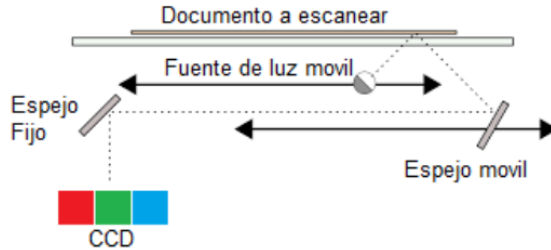


Figura 2.9. Esquema de funcionamiento de un escáner plano [57].

Cámaras digitales y *smartphones*

A diferencia de los escáneres, las cámaras digitales permiten capturar cualquier objeto dentro de una escena en tres dimensiones para digitalizarla en una imagen en dos dimensiones. En la Figura 2.10 se presenta un diagrama de bloques de una cámara digital típica. Las cámaras digitales basan su arquitectura en un sensor de imagen CCD o CMOS que toma la luz proveniente de un objetivo que focaliza la luz en este sensor mediante unas lentes. Este sensor cuenta con una ganancia llamada comúnmente ISO que permite controlar la sensibilidad del sensor. Tras este proceso las imágenes se obtienen en formato RAW, en bruto tal cual la captó el sensor. Posteriormente las cámaras cuentan con un sistema de procesamiento digital que preparan la imagen para llevarla a un formato estándar de reproducción de imágenes, como puede ser JPEG. En esta etapa es donde se aplican procesos como filtros, balance de blancos, corrección gamma, etc., que son dependientes de cada dispositivo. Las cámaras digitales cuentan con gran portabilidad y calidad de imagen, dependiendo del modelo, sin embargo, están influenciadas por la luminosidad ambiental. Esto implica que para medir el color con precisión se deban tener controladas las condiciones de luminosidad ambientales o realizar procesos de calibración previos [58–60].

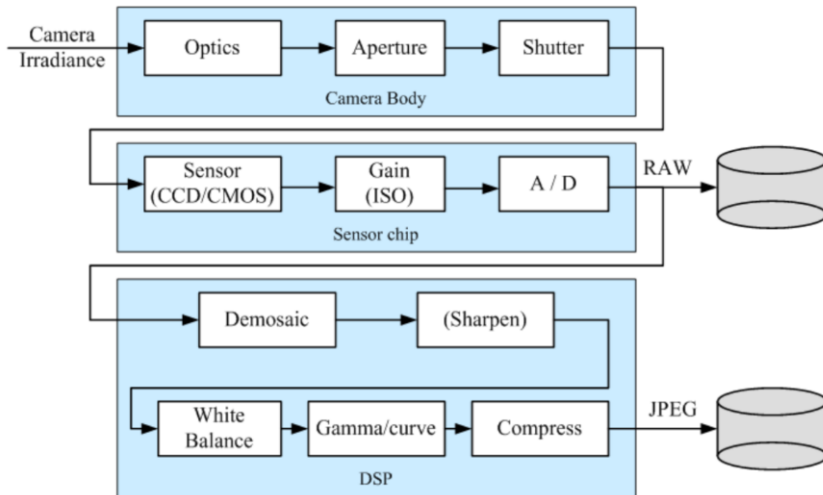


Figura 2.10. Diagrama de bloques funcional de una cámara digital [61].

Gracias a la irrupción de los teléfonos inteligentes en esta última década y la integración de cámaras digitales en estos dispositivos, se empezaron a ver estos aparatos como sustitutos de las cámaras digitales convencionales. En la actualidad los *smartphones* ofrecen una excelente portabilidad y pueden contar con cámaras de una calidad media o incluso de alta calidad en algunos modelos [62–64]. Los *smartphones* cuentan habitualmente con sensores de imagen basados en tecnología CMOS. Actualmente se pueden encontrar sensores de imagen de alta calidad como la tecnología PureCel de OmniVision [65], ISOCell de Samsung [66] y la serie IMX500 de Sony Semiconductor [67]. Estos sensores presentan una gran resolución, un alto rango dinámico y alguno de ellos cuentan con funcionalidades de procesamiento con Inteligencia Artificial. Una de las principales ventajas de los *smartphones* en la actualidad es la combinación de las cámaras digitales integradas con las aplicaciones móviles, disponibles en el dispositivo. Esto posibilita usar aplicaciones para medir el color directamente en el *smartphone* e incluso diseñar aplicaciones para propósitos más específicos aprovechando la capacidad de procesamiento de estos dispositivos. Sin embargo, una de las principales problemáticas del uso de los *smartphones* para la medición del color es que no se pueden controlar todos los parámetros de la cámara. Hoy en día muchos de ellos presentan algoritmos de autoenfoco o embellecimiento de la imagen que hace difícil obtener resultados reproducibles en ciertos experimentos. Aun así, el uso de *smartphones* para medición del color es una solución muy explorada en la literatura [2–4,9,11,68] debido a la alta penetración de estos, su facilidad de uso y su bajo coste en comparación con colorimétricos comerciales.

2.3 Procesamiento de imagen digital

A partir de la imagen digital puede ser necesario aplicarle algún procesamiento digital para mejorar la calidad, realizarle alguna mejora estética o comprimirla para que ocupe menos espacio. En esta tesis se busca extraer información relevante de los sensores químicos captados en la imagen. El objetivo es que un dispositivo identifique automáticamente características de interés en la imagen del sensor para detectar su posición, extraer su intensidad o calcular la concentración del analito en cuestión. Para ello se aplican técnicas de procesamiento de imagen digital (PID) y visión por computador. Por este motivo merece la pena dedicar algunas páginas de este trabajo a explicar algunos aspectos teóricos de estas disciplinas y profundizar en los algoritmos que se han empleado en este trabajo.

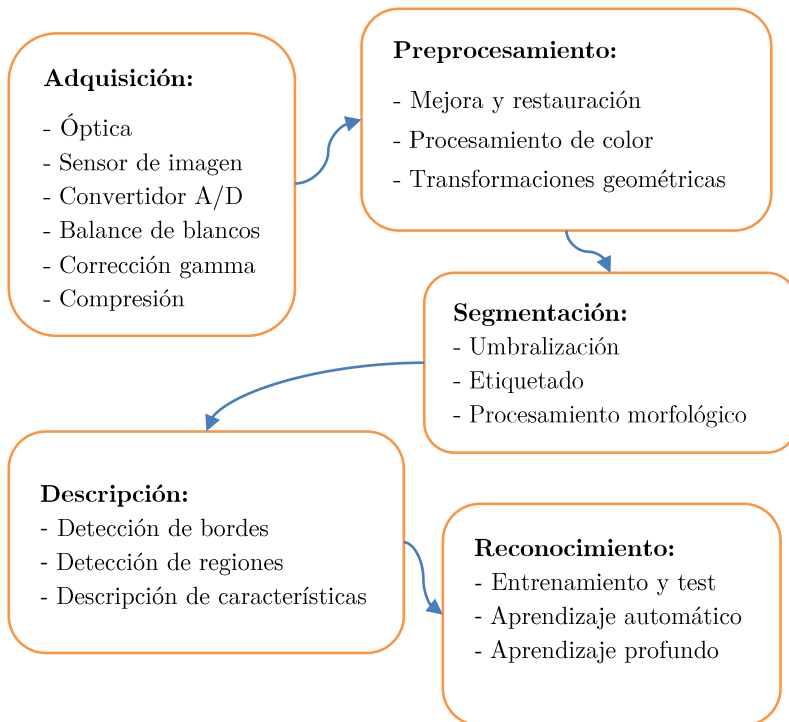


Figura 2.11. Esquema de las etapas del procesamiento de imagen digital y visión por computador.

Para clarificar un poco toda esta información, en la Figura 2.11, se presenta un esquema que organiza las etapas del PID y la visión por computador mencionando algunos ejemplos de algoritmos que se pueden encontrar en cada etapa [69]. La primera etapa del procesamiento de imágenes es la adquisición,

que consta de todos los dispositivos y métodos que van destinados a captar una escena real y convertirla a una imagen 2D. En este proceso están involucradas las herramientas de captación (óptica, sensores y convertidores A/D), mencionadas en la sección 2.2. y otros algoritmos destinados a la preparación de la imagen para mostrarla al usuario, normalmente dependientes del dispositivo. La siguiente etapa es la de preprocesamiento, en la que se aplican procesos dirigidos a mejorar la calidad de la imagen, del color o incluso transformaciones geométricas de la imagen. Después, se encontraría la etapa de segmentación. Esta es la primera etapa donde los métodos están destinados a la extracción de información de la imagen, como los algoritmos de umbralización y de análisis morfológico que permiten detectar regiones de interés en la imagen. La siguiente etapa es la de descripción, en la que se agrupan métodos que emplean algún tipo de magnitud para describir las regiones detectadas. Entre ellos se encuentran la detección de bordes, formas o esquinas y la extracción de atributos y características a partir de un elemento de una imagen. Finalmente, en algunos casos es de utilidad entrar en la siguiente etapa, reconocimiento, en la que a partir de un conjunto de imágenes somos capaces de asignar una etiqueta sobre un concepto que buscamos clasificar. En esta etapa entrarían los algoritmos de aprendizaje automático y redes neuronales, sin embargo, estos quedan fuera de los resultados obtenidos en este trabajo. A continuación, en las próximas secciones, se van a presentar y formular algunos casos concretos de algoritmos de procesamiento de imagen que son útiles para entender este trabajo.

2.3.1 Algoritmos para la adquisición

Una vez es digitalizada una imagen, sobre esta se aplican una serie de algoritmos que van destinados a mejorar el proceso de adquisición y presentación de la imagen. Uno de los casos más interesantes es el de la corrección gamma. Antiguamente los televisores analógicos de tubos de rayos catódicos tenían una respuesta no lineal entre el brillo y el voltaje de la señal de entrada. Esta relación venía determinada por una función caracterizada por un número llamado *gamma*, γ . Para compensar este efecto se realizaba una corrección en el televisor a través de una función aplicando una *gamma* inversa. En la Figura 2.12 se puede ver un ejemplo de la función *gamma* para distintos valores de γ . En la actualidad se sigue aplicando esta corrección *gamma* en cámaras y *smartphones* a fin de que los colores en la pantalla sean más naturales para el ser humano [70]. Esta corrección presenta la desventaja de que los valores RGB no son proporcionales a la intensidad de la luz recibida de forma lineal lo que puede ocasionar dificultades según qué aplicaciones [61]. Sin embargo,

no es tan sencillo anular esa corrección *gamma* ya que muchos dispositivos ocultan la función exacta que usan para realizar dicha corrección.

Otro algoritmo que se realiza de forma interna en las cámaras de forma automática es el balance de blancos. Este es un proceso que busca compensar la iluminación externa moviendo el blanco de la imagen a un blanco puro (referenciado a un sistema de iluminación estándar como el iluminante D65). Esta compensación puede ser especialmente significativa cuando la iluminación externa está muy coloreada [61]. El balance de blancos es un proceso que se suele realizar de forma automática en las cámaras, aunque en la mayor parte de dispositivos puede controlarse para simular una iluminación determinada. En la literatura existen diferentes métodos para realizar dicha corrección [71,72]. Sin embargo, para dispositivos comerciales es complejo conocer las operaciones exactas que se aplican en este proceso ya que suele ser información que las empresas guardan de forma privada.

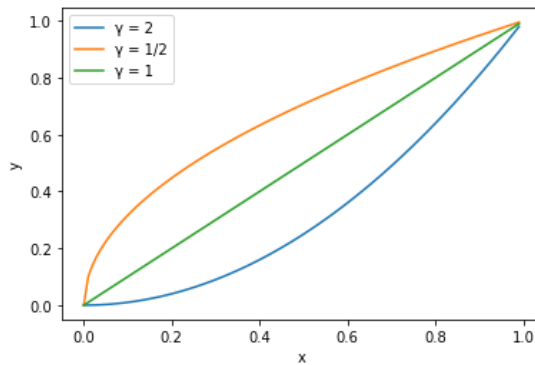


Figura 2.12. Representación de la función *gamma*, para distintos valores de γ .

La última etapa de esta fase de adquisición de la imagen suele ser la aplicación del estándar JPEG [73] la cual implica la compresión de la imagen para reducir el tamaño del archivo final. Las cámaras digitales permiten, normalmente, obtener las imágenes en formato JPEG o en formato RAW. Este último es usado habitualmente por fotógrafos y diseñadores ya que contiene la información en bruto de la imagen sin ningún procesamiento. Sin embargo, este formato ocupa mucho más espacio que el estándar JPEG y requiere de un procesamiento posterior (similar a un revelado) para poder mostrar la imagen final. En *smartphones* se usa habitualmente el formato JPEG a partir del espacio de color YUV, o variantes de este, ya que ocupan mucho menos espacio y permite un procesamiento posterior más rápido. El formato JPEG

realiza una compresión basada en la transformada discreta del coseno (*Discrete Cosine Transform*, DCT) que permite reducir el tamaño de la imagen reduciendo las altas frecuencias de la imagen, a las cuales el ser humano es menos sensible [61].

2.3.2 Algoritmos para el preprocesamiento

En esta etapa se busca procesar la imagen y adecuarla a la aplicación específica que se esté realizando. Este procesamiento se puede realizar de forma manual mediante programas de procesamiento de imagen o pueden implementarse *scripts* que apliquen estos métodos de forma automática. Hay distintas técnicas que son usadas en esta etapa y se pueden clasificar según el tipo de operación realizada en la imagen; en esta sección se van a abordar algunas de ellas.

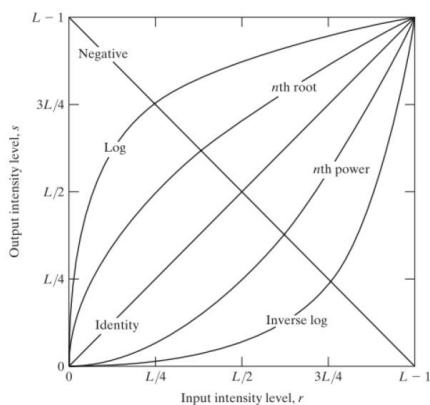


Figura 2.13. Funciones básicas de transformación de intensidad en una imagen [69].

Por un lado, se encuentran las transformaciones de intensidad que operan directamente sobre los niveles de intensidad de la imagen en RGB o en escala de grises. Estas transformaciones aplican un mapeo de los niveles de intensidad pixel a pixel mediante una función [69]. Dependiendo del tipo de transformación utilizada se pueden dar distintas técnicas de procesamiento, véase la Figura 2.13. Hay dos transformaciones que tienen especial importancia en el procesamiento de imágenes en baja luminosidad como pueden ser el aumento del brillo o el ajuste del contraste. La primera permite aumentar el nivel de intensidad de los pixeles mediante una transformación aditiva mejorando la visualización de ésta. Por otro lado, el ajuste del contraste permite

aumentar el rango dinámico de la intensidad de la imagen dando lugar a una mejor visualización y aprovechamiento del rango total de niveles. En la Figura 2.14 puede verse un ejemplo de un ajuste de brillo y otro de contraste. Estos algoritmos se han aplicado sobre la imagen de Lenna (una imagen de test ampliamente reconocida); debajo de las imágenes se presentan las funciones de transformación aplicadas en cada caso empleando el *software* de procesamiento de imágenes GIMP.

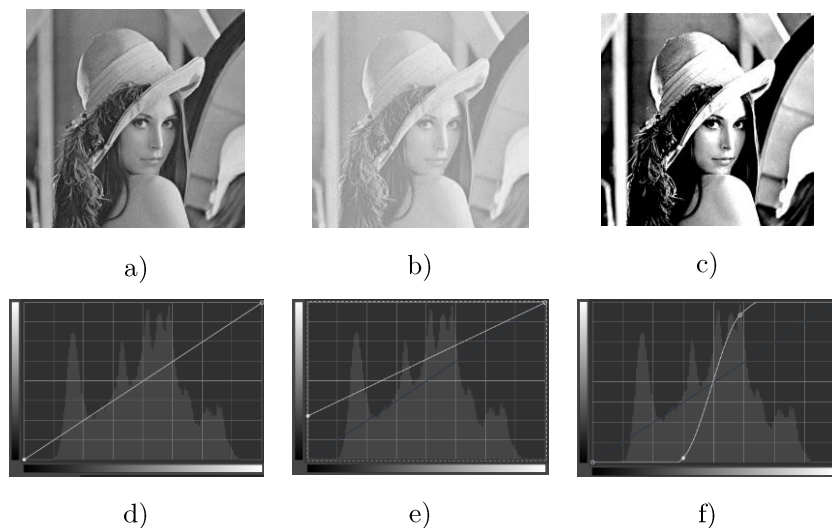


Figura 2.14. Ejemplos de ajustes de brillo y contraste, a) imagen original, b) aumento del brillo, c) y d) aumento del contraste, d), e) y f) funciones de transformación.



Figura 2.15. Transformaciones de intensidad aplicadas sobre cada canal RGB por separado, a) imagen original, b) imagen de una transformación de histograma por canal.

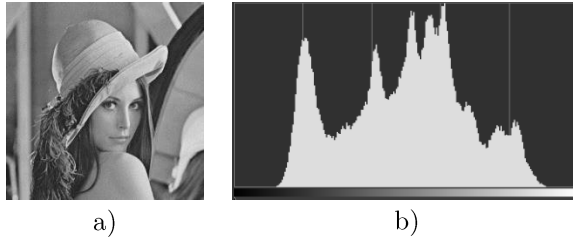


Figura 2.16. a) Imagen de test, b) histograma entre 0 y 255.

También es posible aplicar una modificación de intensidad a cada canal RGB por separado como puede verse en la Figura 2.15. Por otro lado, se tienen las técnicas de procesamiento basado en histograma. Un histograma es una representación de la distribución de los píxeles para cada nivel de intensidad del 0 al 255. En la Figura 2.16 puede verse un ejemplo para la imagen de Lenna. Dentro de las técnicas de procesamiento por histograma se encuentran la ecualización de histogramas o el emparejamiento de histogramas, que son muy útiles para aplicar mejoras automáticas de intensidad modificando el histograma de una imagen a partir de otro conocido del que se quiera replicar su distribución de niveles. En la Figura 2.17 se puede ver un ejemplo de esta técnica en la que una imagen es oscurecida y usando el histograma original como referencia se puede restaurar la imagen obteniendo un nuevo histograma semejante al original. En el procesamiento de imagen existen también las transformaciones geométricas, que modifican la posición de los píxeles de la imagen sin modificar sus niveles de intensidad. Estas funciones se aplican pixel a pixel sobre el dominio de la imagen modificando su forma y disposición. En la Figura 2.18 se presenta un ejemplo de los distintos tipos de transformaciones geométricas. La transformación proyectiva tiene especial relevancia ya que permite modificar la perspectiva de la imagen. Esta operación es usada por ejemplo en la detección de códigos QR para poder reconocer éste desde distintos ángulos de medición o en la construcción de fotos panorámicas a partir de varias fotografías. Finalmente se encuentran las operaciones de filtrado que consisten en aplicar una máscara 2D de un tamaño pequeño mediante una operación de convolución sobre la imagen. Dependiendo del tipo de máscara utilizada se dan distintos tipos de filtrado: de promediado, de gradiente, de mediana, etc. Con este tipo de operaciones se puede conseguir el emborronamiento de imágenes, la detección de bordes o incluso la eliminación de ruido de la imagen [61].

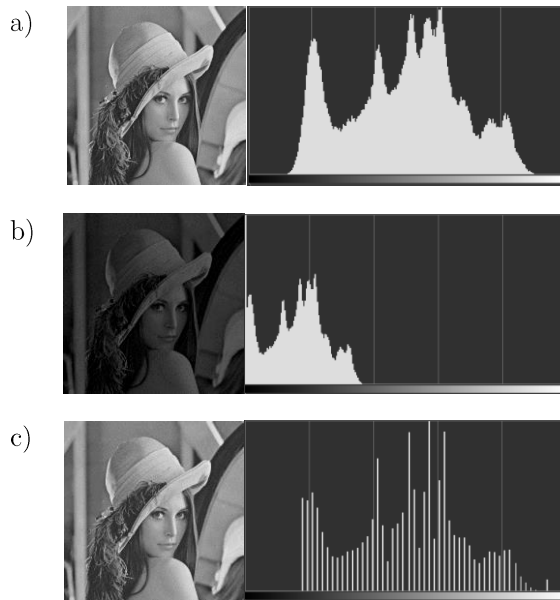


Figura 2.17. Técnica de emparejamiento de histograma, a) imagen original y su histograma, b) imagen oscurecida y su histograma, c) imagen restaurada histograma.

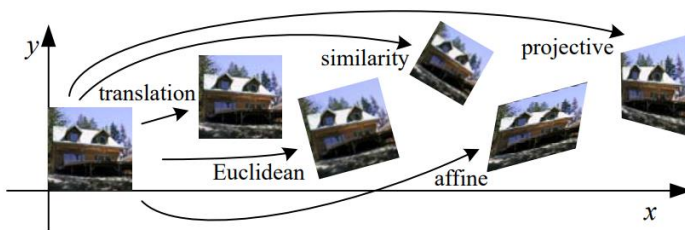


Figura 2.18. Ejemplos de transformaciones geométricas aplicadas a una imagen [61].

2.3.3 Algoritmos para la segmentación

Una vez superada la etapa de preprocesamiento se encuentra la de segmentación. Esta consiste en la identificación de regiones de interés en la imagen. Estas regiones pueden hacer referencia a cualquier tipo de objeto o evento que ocurra en la imagen. La segmentación es el primer paso para poder obtener atributos de interés a partir de una imagen, como el cálculo de su tamaño o el nivel de intensidad. Los algoritmos de segmentación se basan en dos propiedades básicas de los valores de intensidad: discontinuidad y similitud. En la primera categoría se encuentran algoritmos que buscan cambios abruptos de intensidad en la imagen. En la segunda categoría se realiza la partición de

la imagen en regiones que guardan similitud respecto algún conjunto de criterios predefinidos [69]. Si se quiere priorizar la simplicidad de implementación y la velocidad computacional los algoritmos de umbralización son los más adecuados. Estos consisten en la segmentación de regiones basándose en la agrupación de pixeles cuya intensidad está acotada en un rango definido por uno o varios umbrales. Uno de los casos más sencillos se obtienen usando una umbralización por histograma, como se ve en la Figura 2.19. En los casos donde las intensidades de los pixeles de una imagen estén agrupadas en dos o más conjuntos dominantes se pueden identificar los umbrales, T , en los valles de dichas agrupaciones. La función de umbralización $g(x, y)$ para el caso de un único umbral T puede formularse como se indica en la ecuación (16).

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > T \\ 0 & \text{si } f(x, y) \leq T \end{cases} \quad (16)$$

Donde $f(x, y)$ es la función de intensidad de la imagen para cada pixel. El resultado de esta umbralización es una imagen binaria donde los 1s representan las regiones de interés y los 0s el resto de la imagen.

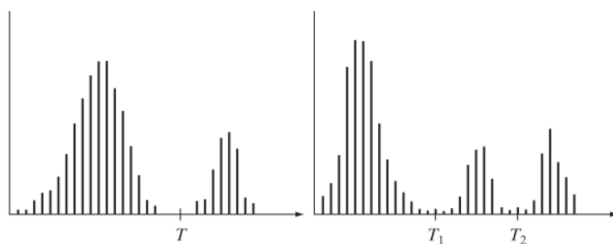


Figura 2.19. Ejemplo de la identificación de umbrales, T , para la umbralización por histograma [69].

Los umbrales se aplican de forma global a toda la imagen, aunque hay casos donde este valor de T puede cambiar a lo largo de la imagen. En estos casos es preferible aplicar un umbral variable mediante la aplicación de un algoritmo de umbralización local. En la Figura 2.20 se puede ver un ejemplo de la umbralización por histograma aplicada en una imagen de huella dactilar. De izquierda a derecha se puede ver: imagen original, histograma de la imagen e imagen umbralizada. Se puede notar perfectamente la agrupación de los pixeles en dos conjuntos, los pixeles de la huella y del fondo.

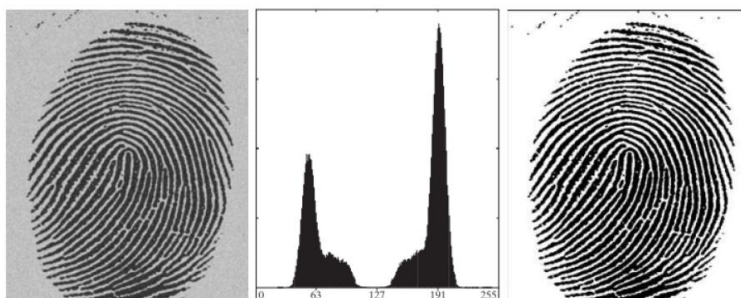


Figura 2.20. Ejemplo de umbralización por histograma aplicada al caso de una huella dactilar [69].

En el procesamiento de imágenes también existe lo que se conoce como operaciones morfológicas, que aplicadas sobre imágenes binarias pueden transformar la forma de las regiones identificadas tras un proceso de umbralización. Puede haber casos en los que tras un proceso de umbralización se requiera ampliar o reducir la región de interés detectada, pulir su forma o incluso eliminar errores en el proceso de detección de estas regiones; para todos estos casos se requiere del uso de operaciones morfológicas.

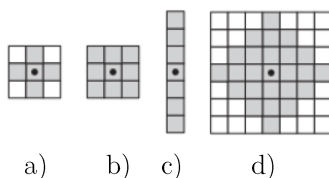


Figura 2.21. Tipos de formas de un elemento estructural, a) cruz, b) cuadrado, c) lineal, d) disco [69].

El fundamento matemático de estas operaciones se basa en la teoría de conjuntos, siendo un conjunto cualquier objeto identificado en la imagen. Estas operaciones basan su funcionamiento en desplazar sobre la imagen un elemento estructural que se aplica pixel a pixel mediante operaciones de conjuntos [69]. Este elemento estructural puede adquirir distintas formas lo que afectará al resultado final de la operación aplicada. Se pueden ver en la Figura 2.21 de izquierda a derecha las formas de los elementos: cruz, cuadrada, línea y disco. Existen dos operaciones morfológicas básicas: la erosión y la dilatación. Estas son la base de otras más complejas como la apertura y el cierre [69].

En la Figura 2.22 se puede ver un ejemplo sencillo de la aplicación de la operación de erosión y dilatación sobre un cuadrado. En la Figura 2.22a se presenta conjunto A , un cuadrado de lado d , y el conjunto B un elemento estructural cuadrado de lado $d/4$. La erosión puede formularse de la siguiente forma:

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (17)$$

lo que indica que la erosión es el resultado de desplazar el elemento estructural B sobre el conjunto A dando como resultado un conjunto de los desplazamientos z de B que pertenezcan al conjunto A [69]. En la Figura 2.22b se observa cómo la operación de erosión reduce las dimensiones del cuadrado A respecto al original (línea punteada) en función del tamaño y forma del elemento estructural B . Por otro lado, la dilatación se formula como:

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\} \quad (18)$$

indicando que la dilatación da como resultado el conjunto de desplazamientos z de B en los cuales al menos un punto de este solape con A [69]. En la Figura 2.22c se nota cómo la dilatación genera un aumento de las dimensiones del cuadrado respecto al original (línea punteada) a diferencia de la erosión. Estas operaciones son muy utilizadas en el tratamiento de imagen ya que permiten mejorar la detección de objetos en la segmentación. En la Figura 2.23 se muestra un ejemplo para el caso de reconocimiento de texto en la que la dilatación aumenta el grosor de las letras detectadas facilitando su legibilidad y descripción.

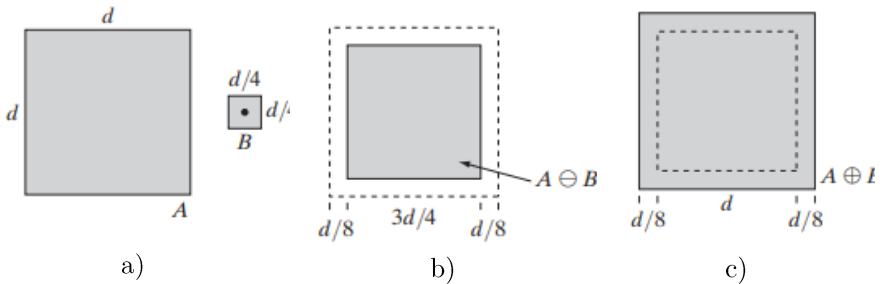


Figura 2.22. Ejemplo de aplicación de las operaciones de erosión y dilatación, a) conjunto A y elemento B , b) erosión, c) dilatación [69].

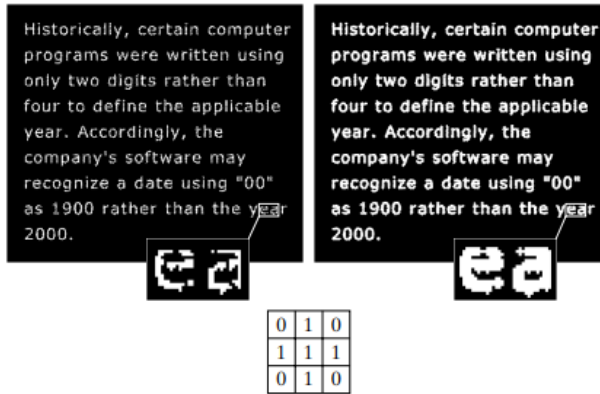


Figura 2.23. Aplicación de una operación de dilatación con un elemento estructural de cruz sobre el resultado de una detección de texto [69].

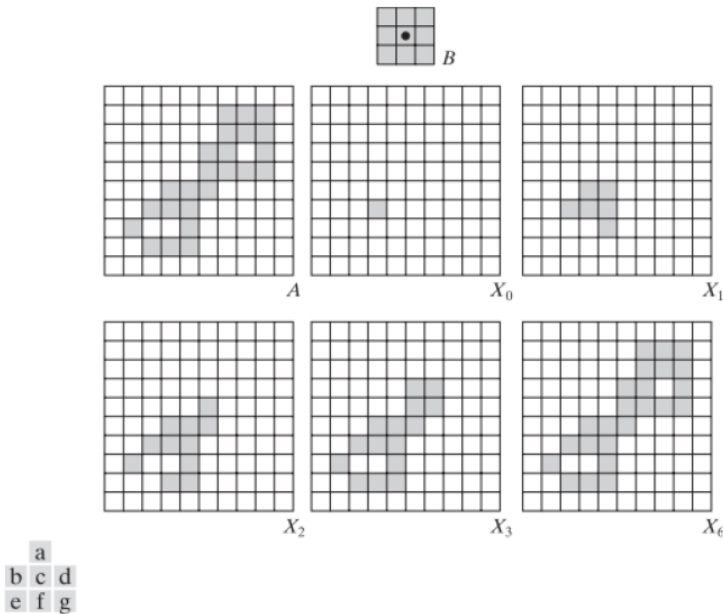


Figura 2.24. Ilustración del procedimiento de identificación de las componentes conectadas [69].

En el caso de que haya varios objetos detectados tras la umbralización es de interés identificar cada uno de ellos por separado. Para esto se requiere la aplicación de un algoritmo de identificación de componentes conectadas que es capaz de agrupar en una imagen binaria aquellos pixeles de valor uno que estén conectados de forma adyacente. Suponiendo que A es un conjunto que contiene una o más componentes conectadas y forma una agrupación X_0 (del mismo tamaño A) cuyos elementos son 0s excepto una localización de una

componente conectada, que será 1s. Empezando por X_0 se quiere encontrar todas las componentes conectadas mediante el siguiente procedimiento iterativo [69].

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots \quad (19)$$

donde B es un elemento estructural deseado que define el tipo de conectividad de que guardan los pixeles. El procedimiento termina cuando $X_k = X_{k-1}$. En la Figura 2.24 se muestra una ilustración para ejemplificar el algoritmo presentado. En la Figura 2.24a se presenta el elemento estructural empleado. En la Figura 2.24b se presenta el conjunto que contiene todas las componentes conectadas. En la Figura 2.24c se puede ver el primer paso del algoritmo en el que se tiene un conjunto con una única componente conectada. Posteriormente se va iterando este procedimiento dando como resultado los conjuntos de la Figura 2.24d-f. Finalmente en la Figura 2.24g se obtiene como resultado todas los pixeles conectados de dicho conjunto.

2.3.4 Algoritmos para la descripción y reconocimiento

Para terminar esta sección se van a presentar brevemente en qué consisten las últimas dos etapas del procesamiento de imagen y la visión por computador. La etapa de descripción es la primera en la que el resultado de esta no es una imagen procesada sino un conjunto de datos de salida que describen los objetos detectados en la etapa de segmentación [69]. Para ello se emplean lo que se conoce como descriptores, herramientas matemáticas y algoritmos que describen algún tipo de característica del objeto identificado. Estos pueden describir los bordes de la imagen, su forma, su contenido y textura. En este trabajo se hará uso de descriptores que son capaces de determinar el área de un objeto y su centro, así como otros que determinen momentos estadísticos de los pixeles del objeto como su media o desviación típica. En el caso de la determinación de sensores químicos colorimétricos la descripción sería el paso previo a la determinación de la concentración del sensor.

Finalmente, la etapa de reconocimiento engloba un conjunto de herramientas matemáticas que buscan reconocer patrones más complejos en una imagen: objetos, personas, animales o incluso expresiones faciales. No se va a profundizar sobre los algoritmos presentados en esta etapa ya que no han sido utilizados durante el desarrollo de este trabajo. Sin embargo, merece la pena mencionar algunos de ellos con la idea de tener una visión global del campo de la visión por computador. Esta etapa está ligada al concepto de aprendizaje automático que pretende lograr que las maquinas puedan aprender a clasificar

patrones complejos a partir de una entrada de datos, como puede ser una imagen. Los modelos de aprendizaje automático se basan en la utilización de conjuntos de datos de entrenamiento y test. Con los datos de entrenamiento se parametriza el modelo que posteriormente es validado mediante el conjunto de test. Los algoritmos de aprendizaje automático se pueden clasificar en dos grandes grupos: supervisados y no supervisados. Los supervisados son aquellos que requieren que los conjuntos de datos estén etiquetados previamente con los patrones que se quieren clasificar. Los no supervisados utilizan herramientas estadísticas para establecer agrupaciones en los datos sin un etiquetado previo. Un caso especial son las redes neuronales que pertenecen a las herramientas de aprendizaje profundo. Estas herramientas son básicas en el desarrollo de la inteligencia artificial y hacen uso de un gran número de datos para obtener resultados más precisos. Todas estas herramientas son ampliamente usadas para el reconocimiento de patrones en imágenes sin embargo no han sido aplicadas en este trabajo ya que se ha priorizado el desarrollo de algoritmos de un coste computacional bajo y sin necesidad de usar un gran número de datos para que puedan usarse en dispositivos móviles de forma nativa y offline.

2.4 Códigos bidimensionales

Los códigos bidimensionales (2D) son una parte esencial de este trabajo ya que las soluciones aportadas derivan directamente del uso de códigos QR, un tipo de códigos bidimensionales. Los códigos bidimensionales son un tipo de código visual que evoluciona de los conocidos códigos de barras. A diferencia de estos últimos, en un código 2D es posible codificar información visualmente en dos dimensiones aumentando la capacidad de información almacenada. Estos códigos son usados extensamente en etiquetado y seguimiento de productos, entretenimiento, restaurantes, etc. Su uso se extendió recientemente a campos como la química o la medicina donde estos códigos se combinan con dispositivos biosensores y μ PAD para dar como resultado códigos 2D que pueden modificarse dinámicamente [26,27,74-76]. Por este motivo, en la siguiente sección se va a profundizar en las características, de los códigos bidimensionales.

2.4.1 Tipos de códigos 2D

Existen diferentes tipos de códigos bidimensionales que ofrecen distintas características de uso. Entre los más utilizados en la actualidad se encuentra

el código QR debido a su rápida detección y su gran capacidad de almacenamiento. En la Tabla 2.1 se presenta una tabla comparativa de la capacidad de los códigos 2D que se pueden encontrar en la actualidad. Se puede notar cómo el código QR es uno de los que tiene mayor capacidad. El código *DataMatrix* es capaz de almacenar gran cantidad de información en un tamaño muy reducido, lo que lo hace ideal en aplicaciones industriales donde se puedan usar cámaras con mayor resolución. Otra de las razones de la gran penetración de los códigos QR es que es un estándar abierto, lo que facilita su implementación en un gran número de aplicaciones [77].

Capacidad (bits)	QR code	DataMatrix	MaxiCode	PDF417	Aztec
Númérica	7089	3116	138	2710	3832
Alfanumérica	4296	2355	93	1850	3067
Binaria	2953	1556	-	1108	1914
Kanji	1817	778	-	554	957

Tabla 2.1. Comparativa de la capacidad de códigos bidimensionales [77].

2.4.2 Características del código QR

El código QR cuenta con diferentes versiones que van de la 1 a la 40, cada una de ellas con diferente configuración y número de módulos. Un módulo es cualquier punto blanco o negro presente en el código. Cada módulo se interpreta como un bit siendo el “0” y el “1” equivalentes al blanco y al negro, respectivamente. Hay un tipo especial de símbolo QR que invierte estos valores, lo que permite imprimirlo sobre fondos oscuros. La configuración de módulos hace referencia al número de módulos contenidos en un símbolo QR, yendo de la versión 1 (21×21 módulos) hasta la 40 (177×177 módulos) [78]. En la Figura 2.25 se muestra un ejemplo de la evolución de las versiones de un código QR.

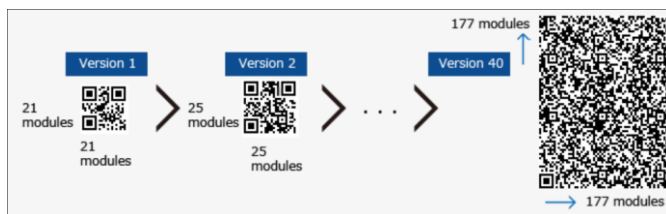


Figura 2.25. Evolución de las versiones de un símbolo QR [78].

Un código QR presenta un conjunto de características que han facilitado su extensa penetración en la sociedad [79]. Tiene capacidad de corrección de errores lo que le permite ser detectado correctamente a pesar de haber sido dañado, véase la Figura 2.26. El código QR permite hasta 4 niveles de corrección de errores dependiendo del nivel de robustez que se quiera tener sobre los datos, véase la Tabla 2.2.



Figura 2.26. Casos de símbolos QR dañados que pueden ser detectados gracias a su capacidad de corrección de errores [79].

Nivel de corrección	Errores corregidos
L	7%
M	15%
Q	25%
H	30%

Tabla 2.2. Niveles de corrección de errores del código QR y porcentaje máximo de errores corregidos.

Dependiendo de los datos almacenados y de la corrección de errores empleada un código QR puede llegar a alcanzar una capacidad de 23 kb para su versión mayor. El símbolo QR permite almacenar una gran capacidad de información en un tamaño de impresión más pequeño que un código de barras convencional. Además, es capaz de almacenar datos numéricos, alfanuméricos y hasta caracteres Kanji y Kana. El símbolo puede leerse desde casi cualquier orientación, pudiéndose imprimir en superficies curvas, compensando así la distorsión en la detección. El código presenta una capacidad de acoplamiento estructurado en la que se pueden juntar varios símbolos QR en uno, disminuyendo su área de impresión, véase la Figura 2.27.

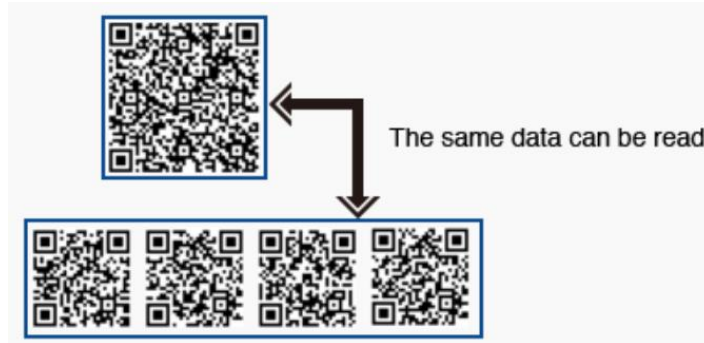


Figura 2.27. Ejemplo de acoplamiento estructurado de 4 símbolos QR en uno [79].

2.4.3 Diseño del símbolo QR

Para alcanzar todas las ventajas mencionadas anteriormente, el símbolo QR está construido por un conjunto de patrones sobre los que se implementa el proceso de detección del código. En la Figura 2.28 se presenta una imagen con las distintas partes que conforman el código.

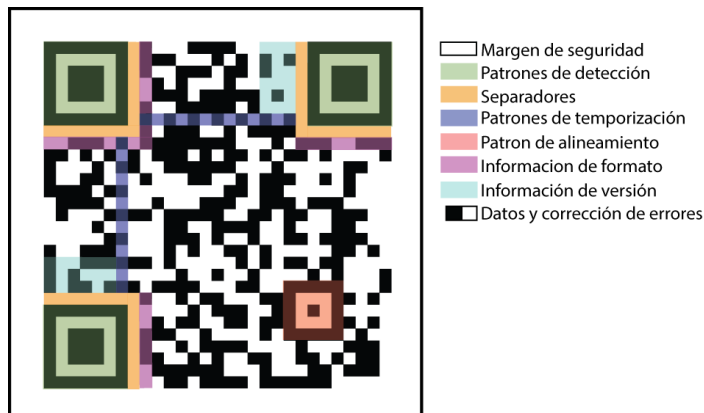


Figura 2.28. Diseño del símbolo del código QR.

El símbolo está rodeado por un margen blanco que sirve como margen de seguridad entre el código y lo que se encuentre a su alrededor. Seguidamente se nota que el código presenta unos patrones de detección en las dos esquinas superiores y en la esquina inferior izquierda, que son los encargados de ayudar a la detección del código. Estos patrones están separados del código por unos separadores blancos. Una vez se detectan los patrones de detección se usa el patrón de alineamiento para normalizar el código al tamaño, la orientación y el ángulo de visualización. También están los patrones de temporización que

ayudan a determinar la densidad de símbolo, las coordenadas de los módulos y el área de información de la versión [80]. Dentro de los datos del código QR hay zonas reservadas para datos necesarios para la decodificación del código como son la información de formato y la información de versión. La primera almacena los datos necesarios para realizar el proceso de enmascaramiento y el tipo de corrección de errores. Cuando se codifica un código QR se realiza un proceso para aleatorizar los módulos y evitar que dentro del símbolo pudieran aparecer patrones que puedan confundir durante la detección. Este proceso consiste en la aplicación de una máscara sobre los datos del código durante la codificación, por ello es necesario almacenar en el código cuál fue la máscara utilizada para poder revertir el proceso durante la decodificación. En la Figura 2.29 se puede ver de dónde se extrae la información de la máscara utilizada y los tipos de máscara existentes. En la información de formato también se encuentra el nivel de corrección de errores usado en la codificación, como se ve en la Figura 2.29.

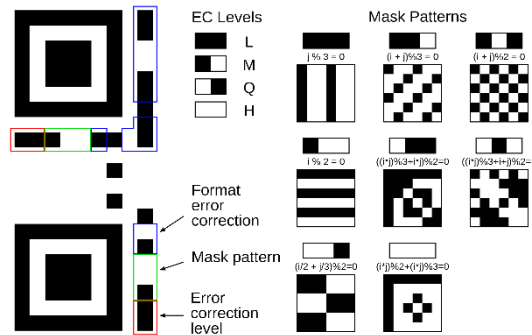


Figura 2.29. Trama de bit de la información de formato y tipos de patrones de máscara [81].

La información de la versión almacena la configuración de los módulos del QR y si se ha realizado un entrelazado en los símbolos de la trama de bits. En algunas versiones se realiza un entrelazamiento entre los símbolos de datos y de corrección de errores para darle más robustez al código frente a errores de ráfaga. El resto del símbolo lo componen los datos codificados y los datos añadidos para la corrección de errores. Estos datos se organizan en símbolos de 8 bits que se van disponiendo a lo largo del código QR, como se muestra en la Figura 2.30. Se escoge la agrupación en símbolos de 8 bits ya que estos son sobre los que se construye la codificación de errores Reed-Solomon. En la Figura 2.30 se nota cómo los símbolos de datos (D1-D26) y los de corrección

de errores (E1 - E44) se entrelazan para evitar errores de ráfaga. Es importante que el proceso de decodificación del código se haga en el mismo orden de codificación ya que si no se incurriría en errores en la lectura.

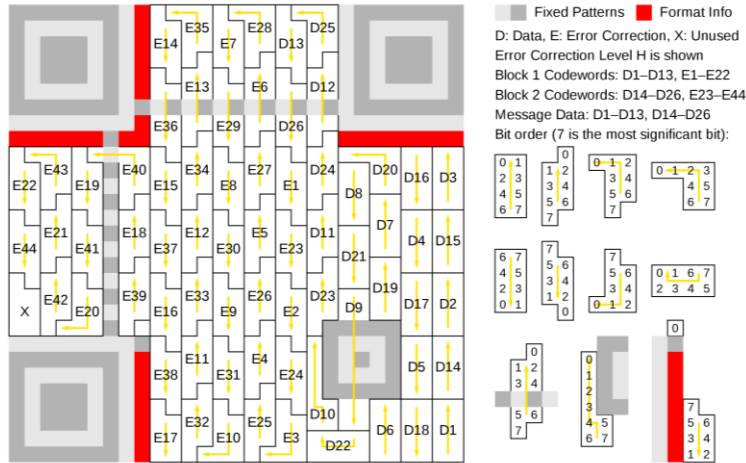


Figura 2.30. Trama de bit del símbolo QR y dirección de lectura de bits [81].

2.5 Códigos de corrección de errores Reed-Solomon

Los datos del código QR presentan una corrección de errores Reed-Solomon para que este pueda ser identificado en caso de que estén dañados. El Reed-Solomon es un código de corrección de errores multibit. En el caso del código QR, los bits se agrupan en símbolos de 8 bits sobre los que opera este código de corrección de errores. En esta sección se van a presentar los fundamentos matemáticos de la codificación Reed-Solomon y los detalles necesarios para entender su implementación *software*.

2.5.1 Definición del código Reed-Solomon

El código Reed-Solomon es un código de bloque, es decir, divide el mensaje transmitido en bloques de datos. El código también es sistemático, lo que quiere decir que la información de protección es añadida de forma separada en bloques aparte, véase Figura 2.31. En un código sistemático el proceso de codificación de errores no altera el símbolo del mensaje original. El código Reed-Solomon es un código lineal (la suma de dos palabras código da como resultado otra palabra código) y cíclico (el desplazamiento cíclico de los símbolos produce otra palabra código) [82]. Este forma parte de la familia de los códigos Bose-Chaudhuri-Hocquenghem (BCH) [83,84], pero opera a nivel de

símbolo, en el que cada símbolo está formado por un bloque de bits. Esto implica que este código sea muy robusto ante errores de ráfaga ya que, aunque todos los bits de un símbolo sean erróneos, el código sería capaz de corregirlo porque solo contaría como un error en un símbolo. Un código Reed-Solomon puede describirse como un código (n, k) donde n es el número de símbolos que tiene la palabra código y k es el número de símbolos de información del mensaje. Hay $n - k$ símbolos de paridad añadidos al mensaje y se pueden corregir como máximo t errores, donde $t = (n - k)/2$. Finalmente cada bloque está formado por símbolos de m bits.

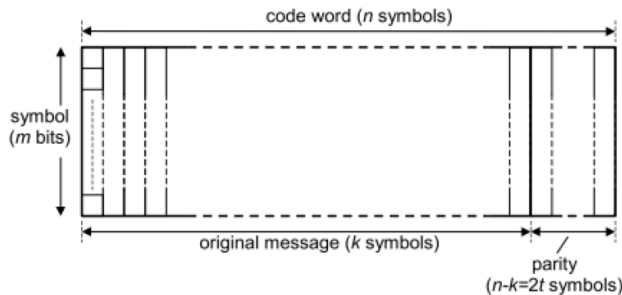


Figura 2.31. Definición de un código Reed-Solomon [82].

2.5.2 Construcción de un código Reed-Solomon

El código Reed-Solomon no se construye en base a la aritmética decimal tradicional sino que se define en base a la aritmética de los campos de Galois de conjuntos finitos [82]. Un campo de Galois es un conjunto de elementos basados en un elemento primitivo α y puede tomar los valores: $0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{N-1}$ para formar un conjunto de 2^m elementos donde $N = 2^m - 1$. Este campo es conocido como $GF(2^m)$. Los elementos de un campo de Galois son finitos y pueden generarse a través de la definición de un polinomio generador del campo $p(x)$ de grado m el cual es irreducible. Por ejemplo, para el caso de $GF(16)$ un posible polinomio generador del campo podría ser $p(x) = x^4 + x + 1$. En un campo de Galois todas las operaciones aritméticas tienen que dar como resultado otro elemento dentro del campo de Galois. La suma y la resta de los elementos del campo son equivalentes y consiste en una operación XOR de los elementos en binario. La multiplicación se realiza mediante una división sobre el polinomio generador para obtener un resto que pertenezca a un elemento del campo. Finalmente, la división sigue un proceso de división larga de números binarios usando la aritmética del campo (suma y resta son equivalentes a un XOR). No se va a profundizar mucho en el desarrollo matemático que definen las operaciones de un campo de Galois, pero sí es importante entender que esta aritmética es la base de que un código

Reed-Solomon sea lineal y cíclico. Se pueden consultar [82] para profundizar más sobre la base teórica de los campos de Galois en relación a los códigos Reed-Solomon.

De igual forma que los elementos de un campo de Galois son generados por un polinomio generador $p(x)$ para construir las palabras código en una codificación Reed-Solomon es necesario un polinomio generador. Un código Reed-Solomon (n, k) se construye por medio de un polinomio generador $g(x)$ que contenga $n - k = 2t$ factores cuyas raíces sean elementos de un campo de Galois. Escoger elementos consecutivos del campo como raíces del polinomio maximiza las propiedades de distancia del código [82]. Por tanto, un polinomio generador tomaría la siguiente forma:

$$g(x) = (x + \alpha^b)(x + \alpha^{b+1}) \dots (x + \alpha^{b+2t-1}) \quad (20)$$

donde se escoge $b = 0$ para reducir la complejidad en la implementación *hardware*. La elección del polinomio generador de los elementos del campo de Galois determina los coeficientes del polinomio generador del código Reed-Solomon.

2.5.3 Codificación Reed-Solomon

Los k símbolos de información del mensaje codificados como un bloque se pueden representar como un polinomio del mensaje $M(x)$ de orden $k - 1$, tal que:

$$M(x) = M_{k-1}x^{k-1} + \dots + M_1x + M_0 \quad (21)$$

donde cada coeficiente M_{k-1}, \dots, M_1, M_0 son los símbolos del mensaje de m bits, elementos de un $GF(2^m)$ [82]. A partir de este polinomio mensaje se puede formar la palabra código transmitida $T(x)$ como sigue:

$$T(x) = M(x)x^{n-k} + r(x) \quad (22)$$

Esta operación consiste en el desplazamiento de los bits del mensaje $n - k$ posiciones a la izquierda y en la suma del término $r(x)$, que se obtiene de la siguiente forma:

$$\frac{M(x)x^{n-k}}{g(x)} = q(x) + \frac{r(x)}{g(x)} \quad (23)$$

donde $r(x)$ es el resto resultante de dividir $M(x)x^{n-k}$ entre el polinomio generador del código $g(x)$. Este procedimiento de construcción del código es del que deriva que el código sea sistemático.

2.5.4 Decodificación Reed-Solomon

Una vez se ha generado la palabra código a transmitir $T(x)$ esta es enviada por algún canal que añade errores al mensaje. Por tanto el polinomio recibido $R(x)$ será dado por:

$$R(x) = T(x) + E(x) \quad (24)$$

donde $E(x)$ es un polinomio de errores que se define como:

$$E(x) = E_{n-1}x^{n-1} + \dots + E_1x + E_0 \quad (25)$$

donde E_{n-1}, \dots, E_1, E_0 son los valores del error en símbolos de m bits que son elementos de un $GF(2^m)$. La posición de los símbolos de error está determinada por el grado de x de cada término [82]. Si hay más de $t = (n - k)/2$ valores de E no nulos se habrá excedido la capacidad de corrección del código y no se podrán corregir los errores.

Para realizar la decodificación del código Reed-Solomon es importante definir el concepto de síndrome. El mensaje recibido $R(x)$ es proporcional al polinomio generador y por lo tanto puede dividirse por los factores de este $(x + \alpha^i)$ tal que:

$$\frac{R(x)}{x + \alpha^i} = Q_i(x) + \frac{S_i}{x + \alpha^i} \quad \text{para } i \leq 2t - 1 \quad (26)$$

donde el síndrome, S_i , es el resto resultante de la división de $R(x)$ entre cada factor del polinomio generador [82]. De las propiedades de los síndromes se deriva que:

$$R(\alpha^i) = E(\alpha^i) = S_i \quad (27)$$

Esto significa que los valores de cada síndrome dependen exclusivamente del patrón de error y que si no ocurre ningún error todos los síndromes serán cero. Para calcular los síndromes se pueden usar varios métodos aunque uno de los más empleados en implementación *hardware* es el Método de Horner [82]. En este método parte de la siguiente descripción de síndrome:

$$S_i = (\dots (R_{n-1}\alpha^i + R_{n-2})\alpha^i + \dots + R_1)\alpha^i + R_0 \quad (28)$$

donde R_{n-1}, \dots, R_0 son los símbolos de la palabra código recibida. Este método tiene la ventaja de que la multiplicación siempre se hace por el mismo valor α^i en cada etapa. A partir de aquí se puede reescribir la ecuación del error (25) para definirla a partir de los síndromes [82]:

$$S_i = E(\alpha^i) = Y_1\alpha^{ie_1} + Y_2\alpha^{ie_2} + \dots + Y_v\alpha^{ie_v} \quad (29)$$

donde e_1, \dots, e_v identifican las localizaciones de los errores en la palabra código (las potencias de x en (25)) mientras que Y_1, \dots, Y_v representan los valores de error de dichas localizaciones. Asumiendo que han ocurrido $v \leq t$ errores. Se puede reescribir la ecuación tal que:

$$S_i = E(\alpha^i) = Y_1X_1^i + Y_2X_2^i + \dots + Y_vX_v^i \quad (30)$$

siendo $X_1^i = \alpha^{ie_1}, \dots, X_v^i = \alpha^{ie_v}$ los localizadores del error. El objetivo de la decodificación es encontrar la posición de los errores e_1, \dots, e_v en la palabra código y su magnitud Y_1, \dots, Y_v para poder corregirlos. Para poder resolver este problema es necesario construir lo que se conoce como polinomio localizador de errores [82]. Este polinomio, $\Lambda(x)$ se puede definir como:

$$\Lambda(x) = (1 + X_1x)(1 + X_2x) \dots (1 + X_vx) \quad (31)$$

desarrollando los factores se deriva que:

$$\Lambda(x) = 1 + \Lambda_1x + \dots + \Lambda_{v-1}x^{v-1} + \Lambda_vx^v \quad (32)$$

El principal problema que se encuentra para determinar los coeficientes del polinomio es que no conocemos el número de errores que han ocurrido v . Para

calcular los coeficientes de este polinomio se pueden usar varios métodos [82]: el método directo, el algoritmo de Euclides o el algoritmo de Berlekamp. Uno de los más interesante es el algoritmo de Berlekamp [85,86] que consiste en una técnica iterativa para resolución de ecuaciones (aún sin conocer v) que cuenta con una implementación *hardware* muy eficiente [82].

Tras la definición completa del polinomio de localización de errores se procede a averiguar las posiciones de la palabra de código que contienen errores. Para ello es necesario resolver el polinomio localizador de errores mediante la búsqueda de Chien [87]. Este método consiste en un procedimiento de prueba y error en el que se sustituyen todas las posibles raíces del polinomio (los elementos del campo $\alpha^i, 0 \leq i \leq n - 1$) en la ecuación (32) y se evalúan los resultados. Si el resultado es cero entonces el valor de la raíz identifica la posición del símbolo [82]. Cuando las localizaciones del error $X_1^i = \alpha^{ie_1}, \dots, X_v^i = \alpha^{ie_v}$ son calculadas se procede a obtener los valores del error Y_1, \dots, Y_v resolviendo las ecuación (30). Un método para resolver este sistema se basa en el algoritmo de Forney que hace uso de la derivada del polinomio localizador de errores [82]. Finalmente, teniendo localizados los símbolos erróneos, X_j , y calculado su valor, Y_j , se pueden corregir los errores del mensaje recibido, $R(x)$, sumándole el polinomio de error, $E(x)$.

2.6 Librería *Zebra Crossing (Zxing) Project*

En esta sección se va a profundizar en la implementación *software* de los detectores de códigos QR. En este trabajo se ha hecho uso de la librería *Zebra Crossing (Zxing) Project* de código abierto [88] como base para el desarrollo de los lectores de códigos QR desarrollados. En cada capítulo se detallarán las modificaciones que se hayan realizado sobre esta librería, que dependerán en cada caso de la aplicación móvil implementada. En esta sección se realizará un esquema general de la librería (la versión para Android) y se detallarán los procedimientos de detección, decodificación y corrección de errores del código QR. Para entender mejor la arquitectura de la aplicación se ha preparado un diagrama simplificado de su arquitectura en la Figura 2.32. Se pueden agrupar los módulos de interés de la librería en cuatro funcionalidades: extracción de la imagen, detección del código QR, decodificación del código QR y visualización de los resultados. Cada funcionalidad engloba un conjunto de clases java que contienen funciones que ayudan a realizar dicha función. En esta sección se va a describir el funcionamiento de cada etapa y se detallarán los algoritmos empleados en el proceso de segmentación, detección y decodificación del QR.

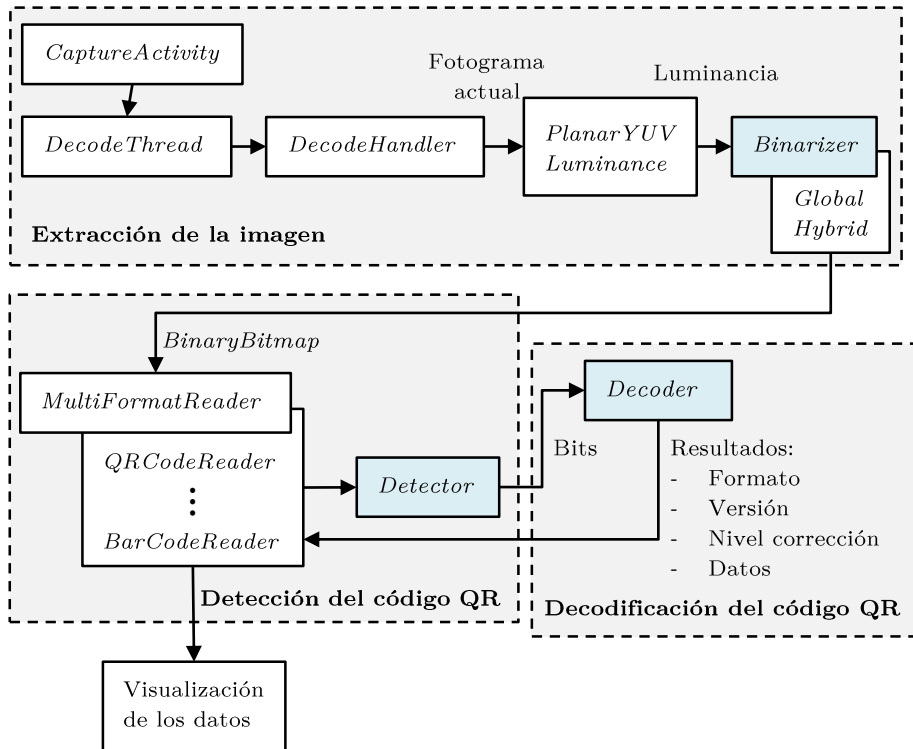


Figura 2.32. Arquitectura simplificada de la librería Zxing.

2.6.1 Extracción de imagen

En esta funcionalidad se agrupan todas las clases que tienen algún tipo de responsabilidad sobre la extracción de la imagen sobre la que se va a detectar el código QR. Este procedimiento puede realizarse de múltiples formas: desde una imagen estática, un vídeo o una vista de lo que ve la cámara en vivo. Esta librería cuenta con un conjunto de clases que permiten la detección de un código QR mediante una vista de la cámara. Hay varias clases que gestionan esta funcionalidad, aunque una de las principales es la clase *CaptureActivity*. Esta se encarga de abrir, configurar la cámara y ofrecer una vista al usuario de lo que ve la cámara. Además, es la responsable de generar un hilo paralelo donde ejecutar el escaneo del QR sin bloquear la vista de la aplicación mediante la clase *DecodeThread*. Posteriormente la clase *DecodeHandler* obtiene el fotograma actual de la cámara en formato YUV. La librería usa el módulo *PlanarYUV Luminance* para extraer el canal de luminancia del fotograma actual equivalente al nivel de gris del fotograma. Para terminar esta

etapa la clase *Binarizer* realiza el proceso de umbralización de la imagen para obtener un objeto del tipo *BinaryBitmap* que consiste en una imagen binaria sobre la que se realiza posteriormente el proceso de detección del QR. Esta umbralización se puede llevar a cabo mediante dos algoritmos: *GlobalHistogramBinarizer* y el *HybridBinarizer*. El primero hace uso de una umbralización por histograma que obtiene un umbral global para todo el QR. Este algoritmo es útil para móviles de baja capacidad de procesamiento, pero puede ser deficiente cuando haya zonas del símbolo QR que se encuentren en sombra. El segundo método hace uso de un algoritmo de umbralización local en el cual se divide la imagen en bloques y se procede a estimar un umbral para cada bloque dependiendo del rango dinámico de sus píxeles. En el caso de que los píxeles de la región presenten un alto contraste el umbral se calculará como el promedio de los píxeles del bloque. Por el contrario, si los píxeles que componen el bloque presentan un rango dinámico pequeño se tomará la mitad del valor mínimo de este. Una vez estimados los umbrales de cada bloque se procederá a umbralizar cada región, pero modificando su umbral por el promedio de este con sus bloques vecinos.

2.6.2 Detección de código QR

Usando la imagen binaria del tipo *BinaryBitmap* la librería comienza un proceso de detección para identificar alguno de los formatos disponibles usando la clase *MultiFormatReader*. En el caso del símbolo QR se usa un detector que forma parte del módulo *QRCodeReader*. A partir de la imagen del tipo *BinaryBitmap* se genera un objeto *BitMatrix* que consiste en la matriz de bits de la imagen (con valores 0 y 1) sobre la que se realiza el proceso de detección. La clase encargada de detectar el código QR se llama *Detector* y se ayuda de otras como *FindPatternFinder* o *FindPatternInfo* para extraer la posición de los patrones de detección y alineamiento. En la Figura 2.33 se muestra un diagrama explicativo sobre el funcionamiento de la detección. Para empezar, se hace una búsqueda de los patrones de detección buscando en la imagen patrones de intensidad del tipo negro/blanco/negro/blanco/negro que siga las proporciones 1/1/3/1/1, tal y como indica la Figura 2.33a este proceso se realiza verticalmente, horizontalmente y de forma diagonal hasta comprobar que los patrones detectados coinciden con el deseado en las tres realizaciones.

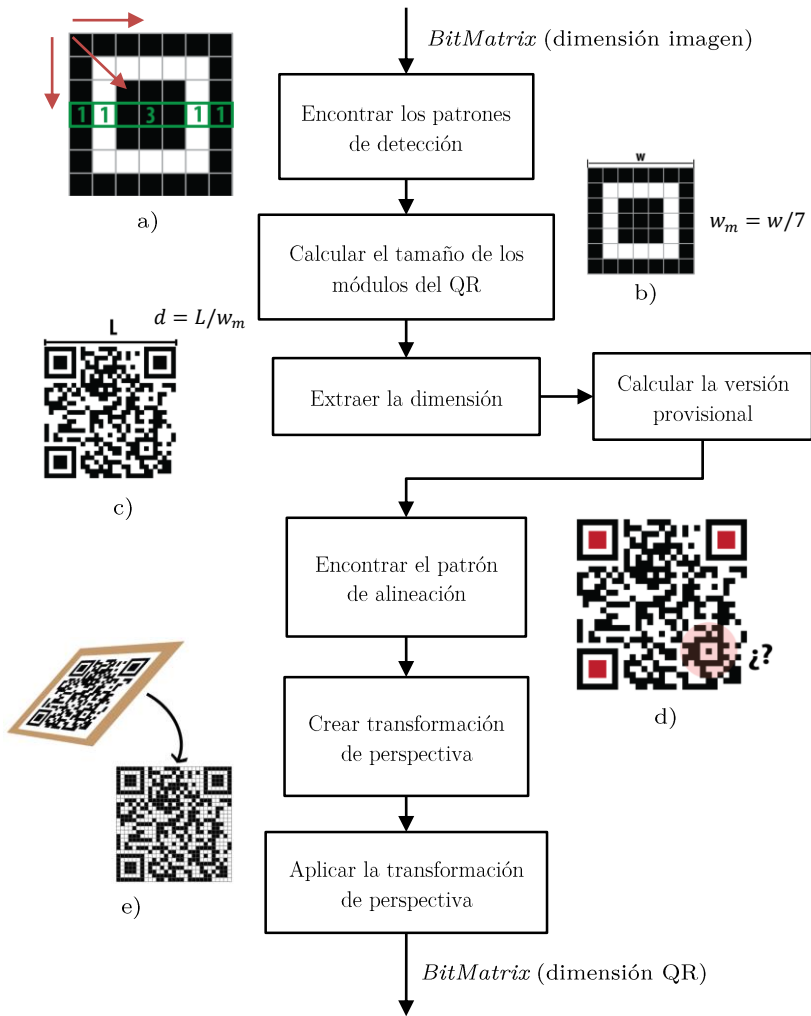


Figura 2.33. Diagrama de funcionamiento del detector, a) patrón de detección, b) tamaño del módulo, c) dimensión, d) patrón de alineamiento, e) transformación proyectiva.

A partir de estos patrones puede determinarse el tamaño del módulo del QR en píxeles conociendo la distancia total del patrón y el número de módulos en una línea del patrón, véase la Figura 2.33b. A continuación, se estima la dimensión del QR a partir de la distancia entre los patrones y el tamaño de píxeles de los módulos, como indica la Figura 2.33c. Una vez conocida la dimensión puede estimarse la versión del símbolo QR de forma provisional debido a que cada versión tiene un número fijo de módulos. El siguiente paso consiste en la detección de los patrones de alineamiento. Este proceso hace una búsqueda del patrón de intensidad de alineamiento en una zona esperada

que viene determinada tras conocer las posiciones de los tres patrones de detección, como se nota en la Figura 2.33c. Los códigos QR de versiones mayores que la 0 presentan patrones de alineamiento. Las últimas dos etapas se encargan de realizar una transformación de perspectiva para mapear los píxeles del QR a una matriz del tamaño de la dimensión del símbolo QR. Este proceso se realiza por parte de las clases *PerspectiveTransform* y *DefaultGridSampler*. La primera modela la transformación de perspectiva deseada y la segunda se encarga de aplicar esta transformación y dar como resultado un objeto *BitMatrix* de la dimensión del código QR.

2.6.3 Decodificación de código QR

Como resultado de la detección del símbolo QR se obtiene un objeto del tipo *BitMatrix* del tamaño de la dimensión del código QR y que cada punto de la matriz es 1 y 0 según sean negro o blanco respectivamente. En la Figura 2.34 se muestra el resultado para la detección de un QR de versión 1.

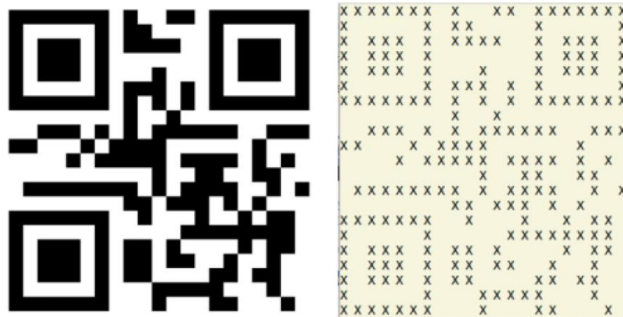


Figura 2.34. Resultado de la detección de un símbolo QR (izquierda) en un objeto *BitMatrix* (derecha) la ‘X’ y ‘ ’ representan el 1 y 0 respectivamente [89].

El siguiente paso es extraer los bits de la matriz del código QR de forma ordenada para obtener su información. Para ello se usa la clase *BitMatrixParser* que implementa un conjunto funciones como *readFormatInformation()* y *readVersión()* que extraen la información de formato (máscara y tipo de corrección de errores) y su versión. Esta información también se encuentra añadida de forma redundante en el código implementando un código de redundancia cíclica (CRC) por lo que estas funciones también realizan esa corrección de errores. Una vez se detecta esta información se procede a leer los símbolos del código usando la función *readCodewords()*. Antes de leer las palabras código es necesario realizar el desenmascaramiento usando la máscara extraída de la información de formato. Tras realizar la extracción de

cada bloque de bits del código QR se procede a corregir los posibles errores mediante un decodificador Reed-Solomon implementado en la clase *ReedSolomonDecoder*. Esta clase emplea las funciones de otro módulo llamado *GenericGF* que implementa la aritmética de los campos de Galois para poder trabajar en la decodificación del código.

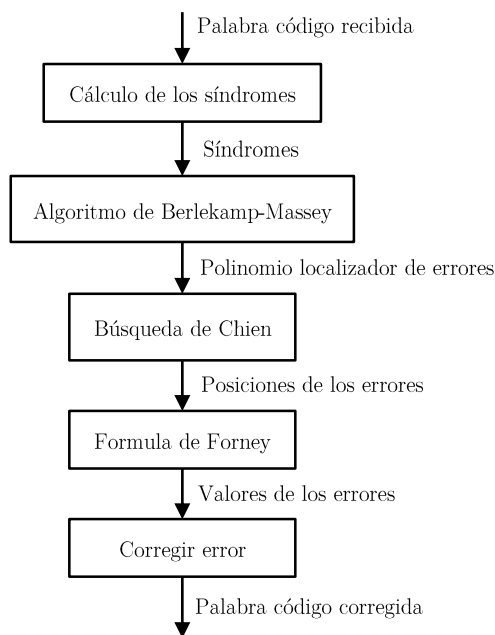


Figura 2.35. Esquema del proceso de corrección de errores en la librería Zxing.

La Figura 2.35 muestra un esquema con el proceso de corrección de errores que implementa la librería Zxing que está basado en la teoría de corrección de errores explicada en la sección 2.5. El primer paso sería calcular los síndromes a partir de la palabra código recibida que permiten calcular los coeficientes del polinomio de localización de errores mediante el algoritmo de Berlekamp-Massey. Posteriormente se aplica la búsqueda de Chien para encontrar la posición de los símbolos erróneos, esto se realiza mediante la función *findErrorLocations()*. La magnitud del error es calculada por medio de la Formula de Forney como se explicó anteriormente. Finalmente se corrige el bloque de código recibido mediante un XOR con la magnitud del error. Para terminar los bits corregidos son codificados según el tipo de dato almacenado (numérico, alfanumérico, byte o Kanji) y se mostraría al usuario los resultados de la detección. La visualización de los resultados dependerá de la aplicación específica que se esté implementando.

Capítulo 3

Sensor colorimétrico integrado en un código QR con corrección de errores

En este capítulo se presenta un sistema de detección de sensores colorimétrico integrados en un código QR mediante el uso de corrección de errores. En primer lugar, se hará una descripción del sistema a nivel conceptual y se profundizará en las soluciones que se proponen para resolver los problemas derivados de la detección de errores colorimétricos en el QR. Seguidamente, se detallará el proceso de implementación *software* del sistema, destacando el procedimiento de extracción y detección de errores. La descripción de la metodología y los resultados experimentales será abordada a continuación, donde el sistema será optimizado bajo distintas condiciones de medición. Finalmente, se presentarán las conclusiones que resultan del desarrollo de este sistema.

3.1 Introducción

El uso de códigos QR en combinación con sistemas microfluídicos y μ PAD es una metodología en plena investigación debido a las ventajas que esta supone. El uso de códigos QR como base del sustrato de estos análisis da la posibilidad de mejorar la trazabilidad de la muestra ya que puede estar directamente conectada con un individuo en concreto. De esta forma se podría tener en un mismo dispositivo la información estática del QR y la información dinámica del resultado del sensor. Esto unido al uso de dispositivos móviles para la lectura de estos códigos reduce la subjetividad del análisis de los sensores colorimétricos incrustados en el QR. Esta innovación supondría un impulso en las tecnologías de etiquetas inteligentes *SmartTags* que puede tener un amplísimo número de aplicaciones en diferentes áreas como: medicina, alimentación, logística, medioambiente, etc.

Una de las aplicaciones de sensores colorimétricos basados en códigos QR consiste en usar el desarrollo de sistemas de control que puedan detectar si el analito introducido en el código QR es positivo o negativo. A la hora de combinar estos dos elementos existen diferentes métodos en la literatura en los que se pueden introducir sensores colorimétricos dentro del QR o fuera de éste. El procedimiento más extendido consiste en retirar parte de un QR para que sea indetectable por un lector convencional e imprimir en esa área un

sensor colorimétrico cuya aparición haga que el QR sea detectable dando como resultado un análisis positivo [26,75,90–92]. En la mayoría de estos casos la detección del QR solo permite la detección de un analito al mismo tiempo, sin embargo, en [75] el dispositivo presentado permite el análisis multianalito borrando varias partes del QR que aparecen cuando los analitos son positivos. Además, en [27], se realiza el desarrollo de una aplicación móvil que es capaz de analizar un conjunto de módulos del QR para medir su concentración.

También se puede encontrar en la literatura el uso de varios códigos QR para la detección de analitos a través de la detección completa de uno o varios códigos QR [74,93]. Concretamente en, [94] se combina este método con un sistema de enmascaramiento gradual de forma que permite detectar concentraciones del analito mediante un procedimiento semi-cuantitativo. Un uso bastante innovador que se puede encontrar en la literatura es la implementación de un sistema de codificación de códigos QR en el que cambiando los últimos datos de la URL almacenada se puedan generar pares de códigos diferentes. De esta forma se generan dos pares de códigos que contienen dos URLs distintas que difieren entre sí solo por un conjunto pequeño de módulos. En el símbolo QR se imprimen estos módulos con tinta reactiva de forma que si estos módulos reaccionan se genera un código con una URL y en caso contrario se construye el otro código QR. De esta forma el resultado del análisis influirá en la información contenida en el código QR [76,95]. Aunque este proceso es interesante por el hecho de que cualquier lector de QR puede decodificar el código presenta el inconveniente de que la información estática del QR se vea modificada por el análisis. Sería más relevante que estos dos elementos se pudieran leer de forma independiente. Un enfoque muy útil para nuestro trabajo es el que propone [96], en el que se hace una detección del QR en canales separados RGB para la detección de información en múltiples canales.

Por lo general los avances actuales que se pueden encontrar en la literatura van destinados al desarrollo de sistemas específicos para cada analito lo que dificulta su escalabilidad para el análisis multianalito. En este trabajo se busca desarrollar un sistema de detección de sensores colorimétricos basados en el QR que sea capaz de aplicarse a distintos tipos de analitos y que solo requiera la utilización de un QR. Se busca además que el sistema de lectura pueda ser ejecutado en una aplicación móvil sin necesitar uso de conexión de internet (aunque pueda tener opción a ello para buscar actualizaciones) para que pueda ser accesible en lugares donde no hay cobertura de red. El objetivo es

que esta aplicación tenga una base de datos de los códigos QR colorimétricos que puede analizar y sea capaz de hacerlo in-situ.

El método que se presenta en este trabajo para la lectura de códigos QR colorimétricos está basado en la identificación de módulos aprovechando la capacidad de detección de errores que presentan los códigos QR. Esta es una aproximación muy novedosa que no se ha usado en la literatura y que facilitaría la detección multianalito en un mismo QR. El desarrollo se va a enfocar desde el punto de vista de cómo influyen los módulos impresos en color en un código QR considerando los errores que es capaz de detectar. De esta forma, se puede generalizar el sistema para su uso con distintos colores, que es lo que finalmente generan los analitos, un cambio de color.

3.2 Conceptualización del sistema

En esta sección se va a presentar el sistema propuesto para dar solución a los problemas planteados con anterioridad. La idea principal consiste en desarrollar una aplicación móvil que sea capaz de detectar un QR modificado químicamente por un analito y determinar si este último resultó de un diagnóstico positivo o negativo. Se verá cómo el color de los analitos va a ser el principal hándicap que hay que resolver, ya que habrá colores que no se detectarán correctamente. En este capítulo se presentan dos soluciones que pueden tener distintas utilidades dependiendo del color de los analitos analizados, sin embargo, ambas aprovechan la capacidad de corrección de errores del QR para detectar los analitos. Primero se hará una descripción general del sistema que explicará aquellos elementos que comparten ambas soluciones. Posteriormente se realizará una conceptualización de cada solución y se profundizará en sus diferencias destacando sus ventajas y desventajas.

3.2.1 Esquema general de un sistema de diagnóstico basado en códigos QR

Un sistema de diagnóstico de sensores químico-ópticos basado en códigos QR consta de dos partes principales: un código QR con módulos modificados químicamente y un sistema de lectura capaz de detectar el contenido del QR y leer la información dinámica aportada por los analitos. La detección del estado de los analitos (si está presente o no) es crucial para realizar correctamente un diagnóstico. Si se quiere detectar la variación de un reactivo binario (positivo o negativo) hay que tener en cuenta que este puede variar de color según una paleta específica de color,

Figura 3.1 muestra un ejemplo ficticio de una paleta de color para un analito. Normalmente esta paleta puede variar de un color cercano al blanco hacia uno más oscuro, variando su saturación, S, por ejemplo. Pero también podría darse el caso contrario, un cambio de color oscuro a claro, o incluso un cambio de tono, H. A partir de este momento se va a suponer que los analitos tratados van a presentar cambios en la saturación, como en la Figura 3.1, de forma que si el analito aparece con un tono similar al blanco se le catalogará como “negativo” pero si aumenta la saturación lo suficiente por encima de un umbral este será “positivo”. Este umbral de diagnóstico del analito es definido normalmente por un especialista que realiza la caracterización y calibración de dicho reactivo. Además de este umbral es importante tener en cuenta el papel del control, que es un reactivo que se coloca en este tipo de análisis, y que será positivo siempre y cuando el análisis esté bien realizado. En la Figura 3.1 también se puede ver una muestra del color que tomaría el control en este supuesto analito.

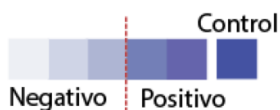


Figura 3.1. Ejemplo de una hipotética paleta de color de un analito. La línea punteada roja marca el umbral de detección. El color que representa el control se marca separado.

La idea principal de este sistema es depositar en algunos módulos (blancos) del QR este analito para que, cuando éstos sean positivos, el QR los detecte como un error en el código usando su corrección de errores, Reed Solomon. Para que los resultados sean fiables y reproducibles es necesario que los analitos y el control sean replicados en varios módulos del QR y, en el caso de que todos (o un alto porcentaje de estos) sean detectados, se diagnosticará como positivo (o negativo en caso contrario). Por otro lado, si los módulos de control no son detectados el análisis sería erróneo o indeterminado.

Para que este proceso se pueda llevar a cabo es necesario que el QR esté diseñado según las necesidades del analito a analizar y que el sistema sea capaz de detectar y decodificar la información estática del QR, detectar la posición de sus errores y determinar el resultado del diagnóstico comparando el patrón de error detectado con el supuesto patrón que debería presentar dicho QR. En la Figura 3.2 se muestra un esquema general de un sistema de detección de QR colorimétrico basado en la detección de la posición de los errores. Para implementar este sistema en una aplicación móvil se usará como

base la librería Zxing en Android para detectar y decodificar códigos QR. Esta librería usa el canal de luminancia del modelo YUV para analizar el QR, lo que facilita la detección de un código QR normal, pero introduce una serie de dificultades cuando se quieren analizar analitos de colores incrustados en el QR. Además, una de las premisas que se ha tratado de mantener en el desarrollo de la aplicación es que esta pueda ejecutarse *offline* y que no se necesite un dispositivo de altas prestaciones para ejecutarse.

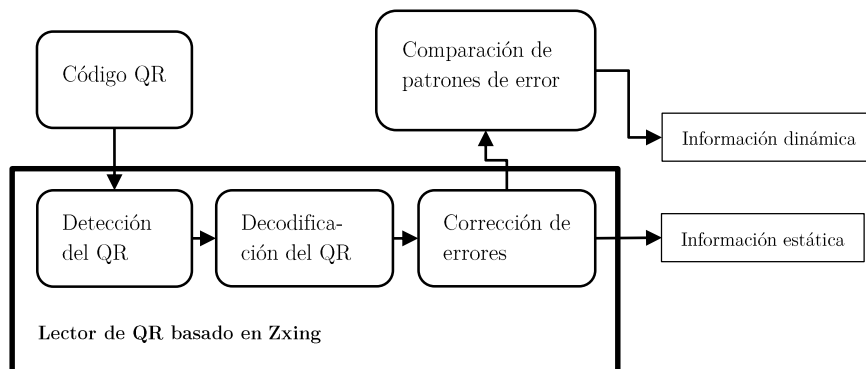


Figura 3.2. Esquema general de un sistema de diagnóstico de sensores químico-ópticos basado en códigos QR.

El problema que se quiere resolver puede abordarse de diferentes formas y en este capítulo se presentan dos métodos que podrían tener su utilidad dependiendo de los analitos usados y de las necesidades del usuario final. Las soluciones propuestas son las siguientes:

- La detección en el canal de luminancia de un código QR impreso con el reactivo de control y los analitos repartidos por el QR.
- La detección en los canales RGB por separado de un código QR impreso en negro con los controles y los analitos repartidos por los módulos del QR.

3.2.2 QR de color como control y detección en luminancia

Como se explicó anteriormente, los lectores de códigos QR captan las imágenes en YUV y usan el canal Y (luminancia) para detectar el código QR. Esto implica un problema cuando se quieren detectar errores de módulos de colores ya que en ocasiones sus valores de luminancia (equivalente al nivel de gris) no son detectados por el lector. El algoritmo de detección usado en la

librería Zxing utiliza los patrones de las esquinas para determinar el umbral de detección del QR. Si el patrón de las esquinas tiene un contraste muy alto (blanco y negro) y el control a detectar muestra una luminancia por encima de ese umbral (un color más claro) este es obviado y no es detectado como un error. Para solucionar este problema se propuso modificar el diseño del símbolo QR para que este sea impreso con una tinta reactiva que sirva como control del ensayo a realizar. Por otro lado, se imprimirían los módulos biosensores con el reactivo, que en caso de reaccionar positivamente serían detectados por la corrección de errores. Al estar todo el símbolo QR impreso con la tinta de control, si el control reacciona de forma correcta, todo el símbolo QR aparecerá y podrá ser detectado. Si el control no reacciona, el código QR no se detectaría debido a algún error en el experimento y no sería necesario comprobar el estado de los módulos biosensores. Además, el umbral generado a partir de las esquinas de este nuevo diseño producirá un umbral más alto (más claro) que favorecerá la detección correcta de los errores generados por los módulos biosensores. En la Figura 3.3 se muestra un ejemplo de un símbolo QR impreso con el color del control de la paleta de color del reactivo. También, se presenta un ejemplo de las posiciones en las que se podrían depositar los analitos biosensores replicados (módulos de borde rojo). El diseño del QR cambiará dependiendo del analito que se quiera analizar ya que el color del control y el número de réplicas del analito deberían calibrarse para cada caso.



Figura 3.3. Diseño de un QR impreso con el color del control de la paleta de color. Los módulos de borde rojo indican la posición de los módulos biosensores.

El sistema propuesto deberá por tanto ser capaz de analizar la información estática y dinámica del QR teniendo en cuenta qué tipo de QR estamos tratando. El usuario final podría seleccionar en la aplicación el tipo de QR a analizar de una lista de QR soportados por dicha versión. En la Figura 3.4 se muestra un esquema conceptual de la solución propuesta para tener en cuenta este requisito. El sistema es capaz de analizar códigos QR a partir de una

imagen de un solo canal. En este caso se extrae el canal de luminancia de la imagen a partir del cual se realiza la detección de las esquinas y se calcula el umbral de detección usado para umbralizar la imagen y detectar el QR. Tras esta etapa se obtiene una matriz binaria de la dimensión del QR en la que los 1s y 0s representan los módulos negros y blancos respectivamente. Posteriormente esa matriz es decodificada y se obtiene una trama de bits que es corregida mediante el algoritmo Reed Solomon. El siguiente paso es extraer la posición de los errores en la trama de bits y mapearlos a la posición que tendrían en la matriz del QR (para facilitar al usuario final su verificación). Tras la corrección de los errores se contaría con una información estática dada por el contenido del QR y las posiciones de los errores en el código QR.

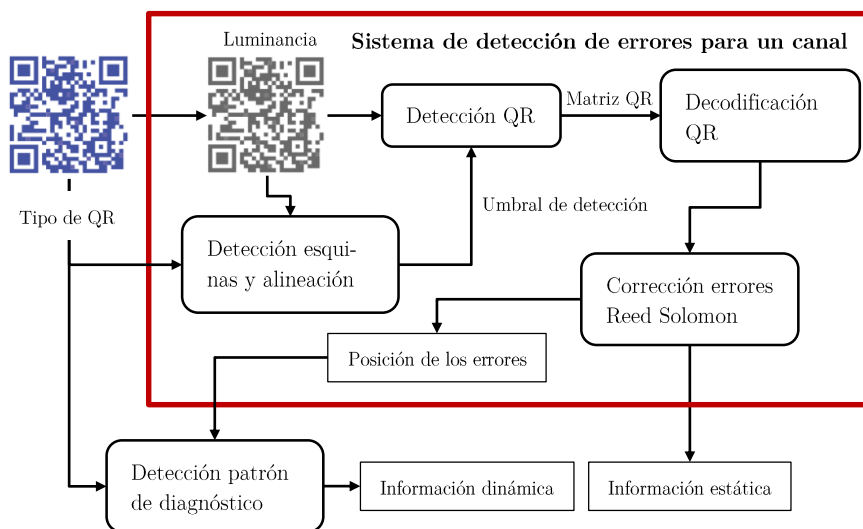


Figura 3.4. Esquema conceptual del funcionamiento de la solución basada en el análisis del QR en luminancia.

El siguiente paso sería definir unos patrones de diagnóstico en base a las posiciones de los errores esperados en el código. Estos patrones de diagnóstico dependerán del analito o analitos que se estén analizando. El caso más simple sería la detección de un analito y comprobar si se detectan como errores todos los módulos biosensores depositados en el símbolo QR. En este caso el patrón de error de diagnóstico positivo sería la detección de todos los módulos biosensores y el patrón de error de diagnóstico negativo que no se detectara ningún error. Podría haber casos más complejos en los que se quiera diferenciar dos analitos en el mismo símbolo QR, lo que conllevaría definir patrones positivos y negativos para cada analito. Más adelante, en la sección 3.5.9, se

verá un ejemplo de detección multianalito. La ventaja de este sistema radica en su simplicidad y en aprovechar las capacidades que ya aporta la librería Zxing para decodificar el QR sin añadir procesamiento extra. Sin embargo, es posible que se puedan encontrar dificultades en la detección de códigos QR que tengan controles con muy bajo contraste, que impedirían que éste pudiera detectarse.

3.2.3 Análisis RGB de la imagen y detección de errores

La siguiente propuesta viene a dar solución a algunos de las dificultades que podría tener la solución anterior. En este caso, en vez de modificar el diseño del QR para que pueda ser interpretado por el lector, se van a implementar cambios en la aplicación para que pueda detectar errores de colores distintos al negro. Para explicarlo se va a suponer que se tiene un sensor con una paleta de color que va del blanco al amarillo, como se muestra en la Figura 3.5a.

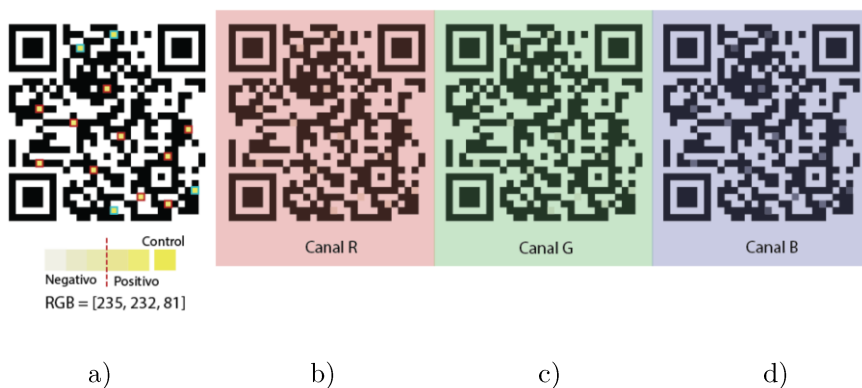


Figura 3.5. a) Diseño de un QR sensor, b) detección del QR en el canal R, c) detección del QR en el canal G, d) detección del QR en el canal B.

Los controles y los biosensores son impresos con tinta reactiva como módulos repartidos por el símbolo QR, como puede verse en la Figura 3.5a. Este diseño presenta cuatro controles (módulos de borde azul) y diez réplicas del analito (módulos de borde rojo). En este caso, si se imprimiera un QR con el control de este analito es probable que no fuera detectado por el lector ya que su luminancia (nivel de gris) está muy cercana al blanco debido a que sus componentes R y G son muy altas. Sin embargo, se puede notar cómo el canal B tiene un valor mucho más bajo, lo que implica un contraste mucho mayor respecto al blanco. De este hecho surge la necesidad de analizar el QR por

separado en cada canal RGB, ya que hay colores en los que un canal da más información que la combinación de los tres. En la Figura 3.5 se puede observar cómo en este caso en los canales R y G no se aprecian los módulos erróneos (marcados en el primer QR) sin embargo en el canal B se aprecian perfectamente. A causa de este fenómeno, el diseño del QR con esta solución se simplifica ya que éste se puede imprimir en negro.

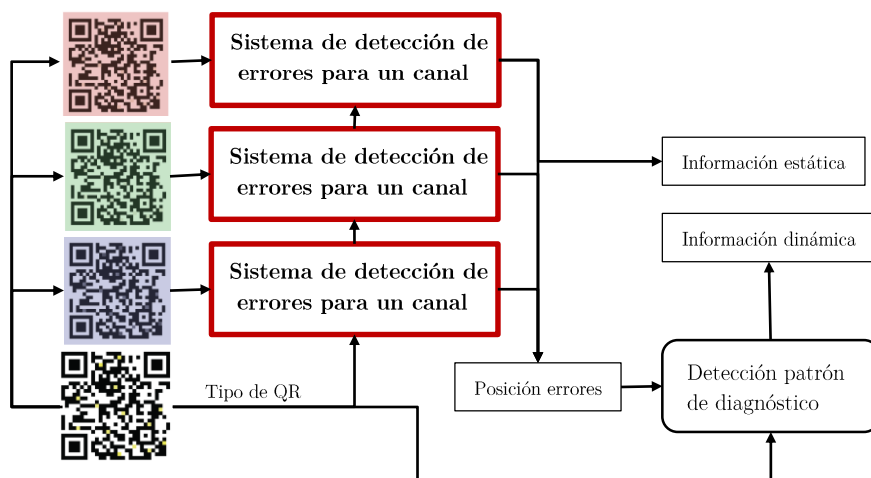


Figura 3.6. Esquema conceptual del funcionamiento de la solución 2 para el análisis del QR en canales separados RGB.

Para implementar la detección en cada canal RGB se va a modificar el sistema propuesto en la sección 3.2.2 para agregar esta funcionalidad. En la Figura 3.6 se puede ver un esquema conceptual de esta solución donde se hace uso del sistema de detección de errores de un canal presentado en la sección 3.2.2. En este caso, a partir de la imagen RGB del QR se extrae cada canal y se procesa por separado. Esto implica que se van a obtener tres patrones de error detectados, uno por cada canal RGB. Una vez realizada la extracción de la información estática y el patrón de errores, en base a la posición de los errores detectados, en cada canal se pasa a realizar la comparación de patrones de error. En este sistema esta etapa tiene un poco más de complejidad ya que se comparan los patrones en los tres canales y el proceso de verificación puede complicarse. Si por ejemplo se quiere detectar un patrón de error para un tono rojo, será necesario comprobar que este aparece en el canal R, pero no en los otros canales. La complejidad de este proceso dependerá del número de análisis analizados y del número de patrones de diagnóstico que sea necesario

definir para realizar un diagnóstico en concreto. El resultado de esta comparación podrá determinar la presencia o no de uno o varios analitos, la información dinámica. Este sistema es más complejo de implementar y puede ocasionar un aumento del tiempo de computación, pero podría aumentar notablemente la gama de colores de analitos detectados. Además, este sistema permite el análisis de varios analitos en el mismo QR añadiendo distintos controles y analitos.

3.2.4 Ajuste de la umbralización

Un analito es detectado como un error por el QR cuando su valor está por debajo del umbral de detección fijado por el sistema. Sin embargo, puede haber ocasiones en las que el umbral de diagnóstico a partir del cual el analito es catalogado como positivo puede ser distinto del umbral de detección, ya que el umbral de diagnóstico del analito es determinado por un especialista. En la Figura 3.7 puede verse cómo en la paleta de colores del analito hay una diferencia entre ambos umbrales que puede ocasionar que el sistema pase por alto un diagnóstico positivo, teniendo lugar así un falso negativo. Es por esto que resulta de interés que haya un umbral de detección asociado a cada analito o tipo de QR. Sin embargo, puede ser problemático definir un valor absoluto del umbral ya que éste podría verse modificado cuando las condiciones de luminosidad cambien. Por esta razón se propone definir un valor llamado *porcentaje de cambio del umbral*, $U(\%)$, que hace referencia al porcentaje de cambio que debería darse sobre el umbral de detección para alcanzar al umbral del analito. Este valor depende del umbral de detección que es calculado en cada análisis del QR en las condiciones de luminosidad actuales por lo que el porcentaje de cambio se verá afectado de forma consistente. Como el valor del umbral del analito es un valor desconocido y fijado de forma perceptual por un especialista se necesita recurrir a un proceso de calibración iterativo para definir $U(\%)$. Por este motivo cuando se diseña un QR para un analito específico es importante analizarlo con la aplicación y modificar este porcentaje reiterativamente hasta que la detección positiva y negativa se discrimine correctamente. En la sección 3.5.8 se da un ejemplo de esta calibración y en la sección 3.5.10 se evalúa cómo se ve afectado por los cambios de luminosidad.

Como se ha podido ver el valor de $U(\%)$ es un valor que depende de cada analito o tipo de QR analizado. Por este motivo, es necesario que esta información sea introducida en la etapa de umbralización en las soluciones propuestas previamente. Este fenómeno ya se tuvo en cuenta en los esquemas

generales de las soluciones del sistema donde se puede apreciar que la información sobre el tipo de QR no solo se introduce en la etapa de comparación de patrones de error sino también en la etapa de detección de esquinas y alineación donde se calcula el umbral de detección, véase Figura 3.4.

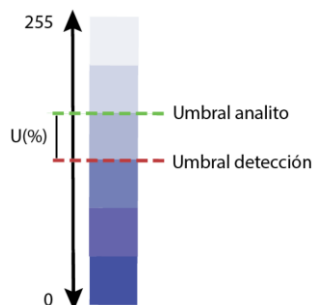


Figura 3.7. Comparación entre el umbral de detección del QR y el umbral de diagnóstico del analito. Definición del porcentaje de cambio del umbral $U(\%)$.

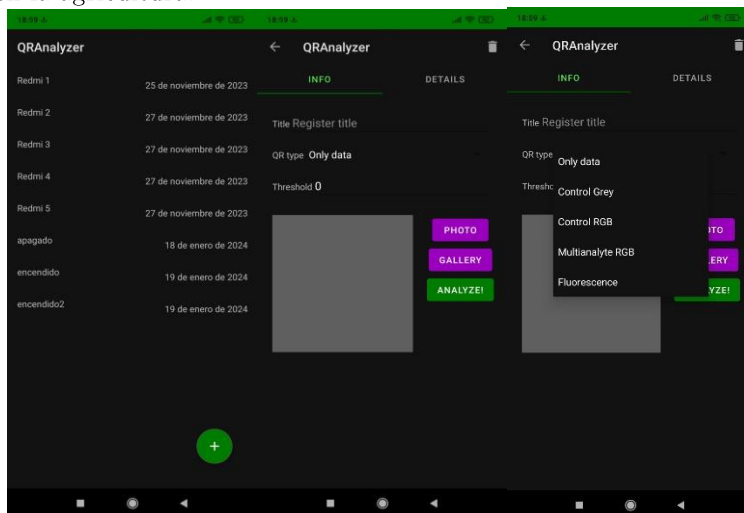
3.3 Implementación *software*

En esta sección se abordarán todos los pasos llevados a cabo para el desarrollo de una aplicación de diagnóstico de analitos químico-ópticos basados en códigos QR. Por un lado, se presentará el diseño de la aplicación y sus principales funcionalidades para el usuario final. En segundo lugar, se expondrá la arquitectura de la aplicación basada en un patrón de diseño modelo-vista-controlador. Posteriormente se profundizará en el proceso de lectura y extracción de errores, para acabar abordando el método de comparación de patrones de error.

3.3.1 Diseño de la aplicación

Esta aplicación está destinada principalmente para personal especializado (médicos, enfermeros, técnicos de laboratorio, químicos, etc.) para facilitar ensayos de diagnóstico y mejorar la trazabilidad de los resultados del análisis con los datos del paciente. Es fundamental mantener una sencillez en el desarrollo y que el usuario final no tenga excesivos requerimientos ni aparatos externos para obtener unos resultados fiables y reproducibles. Además, otro requisito es que sea capaz de funcionar de forma *offline*, aunque pudiera mantenerse actualizada de forma *online*, para poder dar soporte en lugares donde los recursos son más bajos o en zonas rurales donde el acceso a internet está más restringido. Aunque esta aplicación está principalmente diseñada para el ámbito del diagnóstico, podría adaptarse fácilmente para la realización de

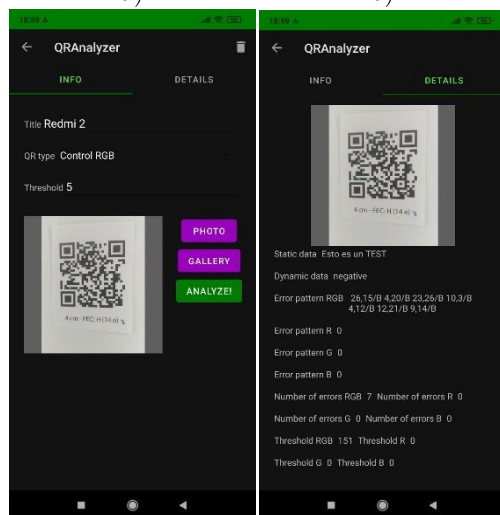
cualquier análisis químico-óptico como en el sector alimenticio, medioambiental o en la agricultura.



a)

b)

c)



d)

e)

Figura 3.8. Capturas de pantalla de la aplicación implementada, a) lista de registros inicial, b) introducir un nuevo registro, c) tipos de códigos QR, d) registro preparado para análisis, e) resultados del análisis.

Al iniciar la aplicación, el usuario se encuentra con una página de inicio que muestra una lista de registros de análisis, identificados con un nombre y una fecha, como se muestra en la Figura 3.8a. Además, esta ventana permite

al usuario crear un nuevo registro o seleccionar uno de la lista. Si seleccionamos un nuevo registro la aplicación nos llevará a una ventana con dos pestañas, como se puede ver en la Figura 3.8b. La pestaña de *Info* nos indica la información general del código QR antes del análisis y la pestaña *Details* nos muestra los resultados del análisis realizado. En la pestaña *Info* es donde tendremos que introducir los datos del QR a analizar. Se nos pedirá el nombre del registro y que seleccionemos de una lista el tipo de QR que estamos analizando (Figura 3.8c). Actualmente la aplicación es capaz de realizar los siguientes análisis:

- Lectura de datos estáticos del QR
- Diagnóstico de un analito a partir del QR en luminancia
- Diagnóstico de un analito con varios patrones en RGB
- Diagnóstico de dos analitos en RGB

A futuro la idea es que cada tipo de QR este identificado con un código de análisis y que la aplicación te permita escoger entre los distintos códigos para saber qué tipo de análisis tiene que hacer. En esa misma ventana está la opción de hacer una foto a un QR o tomar una foto de la galería para posteriormente ser analizada con el botón correspondiente. La aplicación permite además establecer el porcentaje de cambio del umbral $U(\%)$ (opción que se muestra en la Figura 3.8d) para hacer el proceso de calibración más sencillo. En la aplicación del usuario final este valor debería tomarse de la base de datos de códigos QR una vez conocido el tipo de QR. En la pestaña *Details* nos mostrará los resultados del análisis realizado que dependerán del tipo de análisis a realizar. En la Figura 3.8e se muestran los resultados de un análisis de detección de errores RGB. En la versión actual de la aplicación el diagnóstico de analitos nos muestra el contenido del QR, los errores identificados en cada canal y el resultado del diagnóstico. Algunos de estos resultados pueden ser innecesarios dependiendo del usuario final que use la aplicación, pero son muy útiles para realizar las calibraciones de los códigos QR y comprobar los resultados. Todos estos datos son guardados en una base de datos en el dispositivo.

3.3.2 Arquitectura de la aplicación

La aplicación ha sido desarrollada en Android usando Java, aunque se valoró el uso de otros lenguajes multiplataforma como React Native. Finalmente se optó por usar Java ya que la librería Zxing estaba basada en ese lenguaje. Nuestra aplicación está implementada usando un patrón de diseño conocido

como Modelo-Vista-Controlador [97]. Éste busca separar las funciones de nuestra aplicación en tres áreas para que sea más escalable y más sencillo hacer modificaciones futuras. En esta implementación las tareas de visualización y diseño se llevan a cabo mediante ficheros *xml* donde se define cuál va a ser la disposición de los elementos visuales en la ventana. Las tareas del modelo son realizadas por un conjunto de clases que se encargan de definir la estructura de datos y objetos de la aplicación. Finalmente, las tareas del controlador son delegadas a clases de las *actividades* y los *fragmentos* de la aplicación que se encargan de controlar qué datos se visualizan y los eventos en pantalla (desplazamiento de la pantalla, eventos de clic, entradas de texto, navegación, etc.). El controlador es el que se encarga de conectar la parte de visualización con los datos. Para entender el funcionamiento de la aplicación un poco mejor se han agrupado sus funcionalidades en paquetes, donde cada paquete es un conjunto de clases que se encarga de cierta funcionalidad de la aplicación. A continuación, se van a presentar los paquetes implementados en la aplicación y algunas clases de interés que serán explicadas más adelante:

- **Activities.** Este paquete agrupa todas las actividades y fragmentos que controlan el funcionamiento de la aplicación.
- **Model.** Aquí se agrupan las estructuras de datos que son usadas en la aplicación. En este caso contamos con una clase *QRCodeData* que guarda los datos y los resultados cada vez que se realiza un análisis. Se puede encontrar también la clase *ErrorPattern* que se encarga de almacenar los patrones de error de cada tipo de QR soportado y las funciones para comparar patrones de errores.
- **Database.** Este paquete lo componen tres clases que se utilizan para definir y gestionar la funcionalidad de una base de datos basada en SQLite.
- **Tools.** En este paquete se agrupan clases que se encargan de distintas utilidades de interés. *PictureUtils* acoge métodos relacionados con el tratamiento de imágenes. Por otro lado, *QRCodeUtils* se encarga de conectar la aplicación con las funcionalidades de la librería Zxing.
- **Zxing.** En esta librería se encuentran todas las clases que permiten la detección y decodificación de los códigos QR. En este paquete también se implementó la clase *ErrorsExtraction* que se encarga de la extracción de la posición y el estado de los errores.

Esta arquitectura permite una agrupación de sus funcionalidades y una mayor posibilidad de escalabilidad en caso de que se quieran añadir nuevas características.

3.3.3 Lectura, extracción de errores y comparación de patrones de error

Para que la aplicación pueda realizar la detección y decodificación del código QR se usa una función de la clase *QRCodeUtils* llamada *getQRFromImage()* la cual es capaz de extraer la información estática de un QR a partir de una imagen de un canal (ya sea luminancia o cada canal RGB por separado). Esta función hace uso de los métodos de la librería *Zxing* para detectar y decodificar la información del código QR. En esta sección solo se van a describir aquellas clases que se han tenido que modificar o implementar para obtener la posición el estado de los errores en el código QR. Una vez el QR es detectado la aplicación genera y guarda una matriz del código QR usando la clase *BitMatrix*. A partir de esta matriz la clase *BitMatrixParser* es capaz de extraer la información del código QR. Para ello usa las funciones *readFormatInformation()* y *readVersion()* que son capaces de extraer la información de formato y versión del QR que queda inalterada tras el enmascaramiento tal y como se explicó en la sección 2.6. A partir de esta información se extrae el tipo de máscara que tiene que usar el sistema para desenmascarar el código y el nivel de corrección de errores usada. Sabiendo el nivel de corrección de errores se puede saber la cantidad de palabras de código de 8 bits correspondientes a los datos y a la corrección de errores y es posible hacer una lectura de los datos bit a bit, como se explicó en la sección 2.4.3

La clase *ReedSolomonDecoder* realiza el proceso de corrección de errores detectando los símbolos erróneos y corrigiendo los resultados. La extracción de la posición de los errores en cada símbolo se ha implementado inspirándose en la solución aportada en [89]. Además, se ha realizado el mapeo de la posición de los errores en cada símbolo a su posición del módulo dentro de la matriz del código QR. Tanto la posición de los errores como el umbral de detección de cada análisis son almacenados en una variable de la clase *InfoErrors* que luego se utiliza para mostrar los resultados y realizar la comparación de los patrones de diagnóstico. Para realizar la comparación de los errores detectados con los patrones de error de referencia y saber si un análisis es positivo o negativo se ha implementado la clase *ErrorPatterns*. En esta clase se almacenan los patrones de error de los distintos tipos de símbolos QR y se implementan las funciones para comparar patrones de errores simples,

patrones multiresultado o patrones multianalito. Un patrón de error está compuesto por la detección de la posición de un conjunto de errores derivados de la apariencia del analito. Se podría considerar que un patrón ha sido detectado cuando se identifican todos los errores que lo definen. Sin embargo, en un contexto real esto puede ocasionar muchos falsos negativos ya que el patrón está compuesto por réplicas del mismo sensor y alguna podría fallar por motivos experimentales. Por esto, es necesario determinar un porcentaje mínimo de errores detectados para clasificar un patrón como positivo. En nuestro caso se va a definir el porcentaje mínimo de errores detectados como el 80%. En este se asumen problemas en la impresión de las réplicas o cambios en las condiciones de luminosidad. Sin embargo, este porcentaje dependerá en última instancia del analito concreto que se esté analizando y de las indicaciones de los especialistas que realizaron su calibración. También, hay que tener en cuenta la opción de que no se detecte ningún error en las posiciones de control, lo que ocasionaría un análisis erróneo. Una vez detectado el patrón de control si no se detecta el patrón positivo se consideraría un patrón de diagnóstico negativo.

3.4 Metodología experimental

En la siguiente sección se va a presentar la metodología experimental seguida en este capítulo. Se presentarán los reactivos y materiales usados en los experimentos, así como el *hardware* y *software* empleados a lo largo del trabajo desarrollado. Finalmente se describirá el proceso de medición seguido para realizar los experimentos de validación en la sección 3.5.

3.4.1 *Hardware y software*

Para la toma de las imágenes de los códigos QR se han llegado a usar tres dispositivos distintos. El dispositivo principal donde se ejecuta la aplicación es un Pocophone F1 (Xiaomi Corporation, Hidian, China) con una cámara dual de 12 + 5 megapíxeles con enfoque automático de doble píxel y sensor CCD. La distancia focal de la cámara trasera es f/1.9 y cuenta con 6 GB de RAM y procesador Snapdragon 845 [98]. En la sección 3.5.11 se hace una comparativa del sistema con otros dos dispositivos principales. También se ha usado un dispositivo Xiaomi Redmi 7A con cámara dual de 12 + 5 megapíxeles y distancia focal de la cámara trasera de f/2.0. Cuenta con 2 GB de RAM y con un procesador Snapdragon 439 [99]. Además, se trataron imágenes de una cámara Canon EOS 70D [100]. Esta cámara cuenta con un sensor

CMOS de 20.2 megapíxeles con enfoque automático de doble pixel. Las imágenes fueron tomadas usando una caja de luz Duclus Photography de 22×23×24 cm que cuenta con dos tiras de led de 6500K [101]. Los símbolos QR se imprimieron en una impresora de inyección de tinta Epson XP-2200 [102]. Para el desarrollo *software* se empleó la herramienta Android Studio y los símbolos QR se diseñaron usando Adobe Illustrator CS6.

3.4.2 Procedimiento de medida experimental

El procedimiento general de medida consiste en introducir un QR en la caja de luz y tomar las imágenes desde la abertura circular con la que cuenta la caja en el techo. De esta forma la distancia de medición sería de unos 22 cm y el ángulo sería perpendicular. Para reducir la distancia de medición se utiliza un soporte de fabricación manual, donde se coloca el QR sobre un papel blanco, con el que se puede ajustar la distancia de medición. Los códigos QR utilizados varían según el objetivo del experimento y se definirán en cada caso. Una vez se toman las fotos del QR y se analizan in-situ por la aplicación los datos son extraídos para su análisis y las imágenes son guardadas por si hicieran falta para una posterior revisión. En algunos experimentos se modifican algunas condiciones en la medición como la distancia o las condiciones de luminosidad para comprobar su influencia en los resultados. El dispositivo utilizado para realizar el análisis será exclusivamente el Pocophone F1, excepto en la sección 3.5.11 donde se comparan éste con otros dispositivos.

3.5 Resultados y discusión

En esta sección se realizan un conjunto de experimentos que van destinados a validar las hipótesis enunciadas anteriormente y comprobar el rendimiento de las soluciones propuestas. Esta sección está organizada de forma que favorece el discurso para explicar las decisiones tomadas en el diseño a raíz de las problemáticas que van apareciendo. Por ello, al principio se hace un estudio de la capacidad de detección de errores de un QR convencional con un análisis en luminancia. Se analizarán las posiciones que pueden ocupar los errores o el número máximo que se puede detectar. Posteriormente se estudiará cómo influyen las condiciones de medición distancia y luminosidad en la detección de errores. También, se hace un estudio de la influencia del color en la detección de errores y se evalúan las dos soluciones propuestas para solucionar los problemas que puedan aparecer. Seguidamente se realizará una calibración del porcentaje de cambio del umbral, explicado en la sección 3.2.4. Finalmente,

se comprobará el rendimiento del sistema para detectar patrones de errores específicos y se valorará su rendimiento bajo distintas condiciones de medida.

3.5.1 Evaluación de la capacidad de corrección y la disposición de errores

En este primer experimento se quiere hacer una evaluación de cómo influye la posición de los errores en la capacidad de corrección de los errores y en el número máximo de errores que se pueden detectar. Para ello se ha usado un código QR impreso en negro de 5×5 cm con un tamaño de módulo de 2.5 mm. En el código QR se han ido pintando gradualmente los módulos blancos con un rotulador negro en distintas posiciones para generar errores en el código. El objetivo de este experimento es determinar cuál es el número de errores que se pueden detectar dependiendo de la disposición de éstos en el QR. En la Figura 3.9 se pueden ver las imágenes de los símbolos QR y la disposición de los errores en el QR. Las disposiciones de errores de izquierda, derecha y centro se basan en la idea de agrupar todos los errores lo más juntos posibles. La disposición distribuida trata de inducir los errores en bytes separados en la trama de datos de esta forma solo habría un error en cada símbolo de 8 bits. Este experimento se ha realizado con un QR con la capacidad de corrección más baja (L) y otro con la más alta (H). Las imágenes se han tomado en una caja de luz a una distancia de 22 cm y un ángulo de medición perpendicular. El dispositivo usado para ejecutar la aplicación y tomar las fotos fue Pocophone F1. El código QR ha sido analizado con la aplicación desarrollada usando el método de detección en el canal de luminancia.

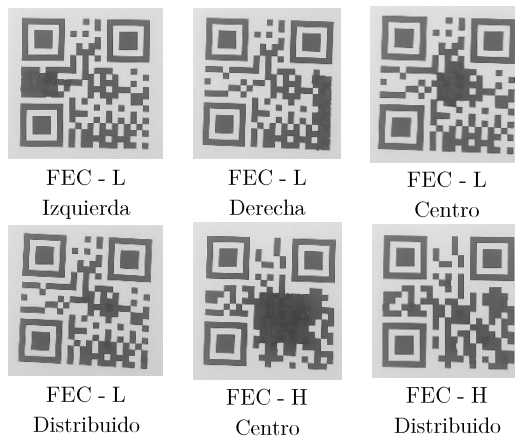


Figura 3.9. Imagen de la disposición de los errores en el QR y su nivel de corrección de errores (FEC).

En la Tabla 3.1 se pueden ver los resultados del experimento realizado. Se puede observar que el aumento de la capacidad de corrección de errores aumenta el número de errores máximos que se pueden detectar para todas las disposiciones. En el caso de los errores distribuidos por el código con nivel de corrección H se nota un aumento de la corrección de más del doble de los errores respecto al nivel L. Además, se puede notar un fenómeno muy significativo en relación a la disposición de los errores. La disposición distribuida supone una reducción muy notable del número máximo de errores detectados. Esto es debido a que el tipo de corrección de errores que utiliza el código es Reed Solomon. Este tipo de corrección funciona a nivel de símbolo por tanto se generan palabras de código para cada símbolo de 8 bits. Esto provoca que el código detecte errores a nivel de símbolo y que tenga un máximo de símbolos que puede corregir en cada caso. Sin embargo, dentro del símbolo puede darse más de un bit erróneo ya que seguirá contando como un símbolo erróneo. Por ese motivo los errores agrupados permiten una mayor detección del número de módulos erróneos ya que tienen la posibilidad de agrupar varios errores dentro del mismo símbolo. La diferencia de errores detectados en las disposiciones agrupadas radica principalmente en la cantidad de módulos blancos que haya en proximidad para pintar y si estos han ocupado algún símbolo adyacente aumentando el número de símbolos erróneos. Aunque el sistema distribuido es el que tiene menor capacidad de corrección es ideal como punto de partida para generar errores ya que permite que los errores estén mejor distribuidos espacialmente y que no se vean excesivamente afectados por sombras en el código. Por este motivo la disposición utilizada a partir de ahora será la distribuida, pero con la posibilidad de añadir más errores dentro de cada símbolo erróneo si fuera necesario.

Disposición	FEC	Errores
Izquierda	L	16
Derecha	L	14
Centro	L	12
Distribuido	L	3
Centro	H	30
Distribuido	H	8

Tabla 3.1. Número de errores máximos detectados según la posición del error en el QR y su nivel de corrección de errores (FEC).

3.5.2 Evaluación de la detección de errores con la distancia

En este apartado se va a realizar un estudio de la influencia de la distancia de medición en la detección de errores del código QR. Para ello se definirá un nuevo código QR basándonos en una disposición distribuida en el que se han inducido 14 errores de color negro, véase la Figura 3.10. En la Figura 3.10a los módulos marcados en rojo indican la posición de los errores identificados por un número. En la Figura 3.10b puede verse a qué posición corresponde cada identificador en el código QR y su estado previo al error (B es blanco y N es Negro). El código tiene un tamaño de 4×4 cm con un tamaño de módulo de 1.3 mm, haciéndolo más manejable y realista. El símbolo QR se detecta en el canal de luminancia con los parámetros de la cámara por defecto.

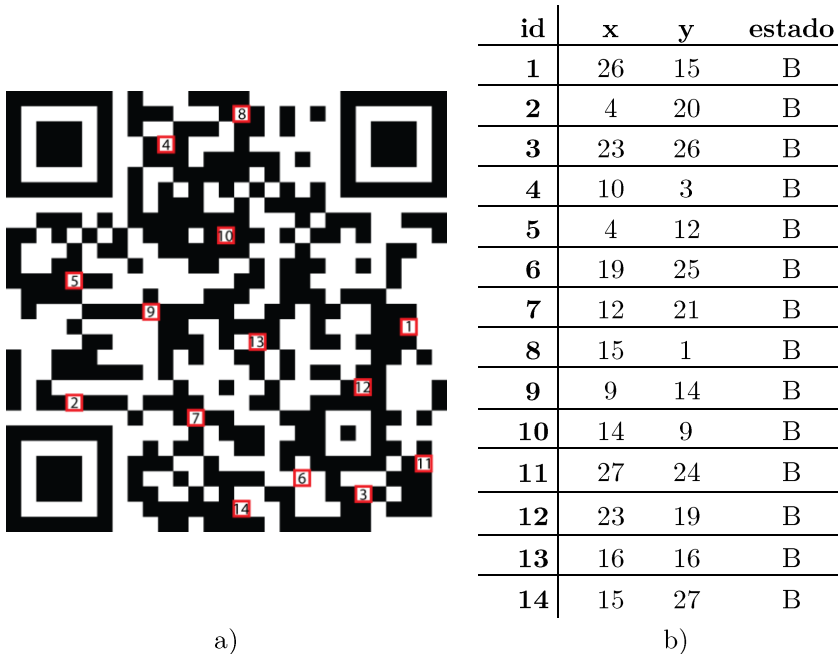


Figura 3.10. a) Diseño del QR con disposición de errores distribuida, b) tabla de identificación de los errores.

La distancia de medición variará desde los 5 cm hasta los 60 cm para hacer una detección del símbolo QR en cada posición. En la Tabla 3.2 se pueden ver los resultados de este experimento indicando el número de errores detectados, donde NR indica que el código QR no fue reconocido. Se puede apreciar cómo para el caso de errores de color negro la capacidad de errores no varía con la distancia ya que en todos los casos se ha podido detectar el máximo de

errores, 14. Se puede apreciar que para los casos más extremos 5 cm y 60 cm el QR ya no es detectable. A una distancia de 5 cm el dispositivo no puede enfocar correctamente el QR y éste no es detectable. Esta distancia podría variar según el dispositivo utilizado. Para el caso de 60 cm el QR es tan pequeño que no se pueden apreciar correctamente en la imagen. Se podría definir una distancia óptima de medición como la distancia más cercana en la que el QR se pueda enfocar correctamente. Esta sería la distancia óptima para detectar módulos de color y además sería más práctica para el usuario final ya que no es necesario alejarse ni acercarse mucho para realizar el análisis.

Distancia (cm)	5	6	9	12	14	17	19	21	22	24	40	60
Errores	NR	14	14	14	14	14	14	14	14	14	14	NR

Tabla 3.2. Análisis de la capacidad de detección de errores cambiando la distancia de medición.

3.5.3 Condiciones de luminosidad y dispositivos

Para que un sistema pueda operar de forma práctica es fundamental que sus resultados sean fiables bajo distintas condiciones de luminosidad. Para testear la influencia de los cambios de luz en el sistema se van a modificar los parámetros de la cámara para emular distintas condiciones de luminosidad. Variando la ISO se podrán emular entornos con distinta intensidad lumínica. Por otro lado, si se cambia el balance de blanco se podrán emular entornos con fuentes de luz de distinta temperatura de color. En este caso se evaluará la influencia de las condiciones de luminosidad en la detección de errores en negro del QR diseñado en la sección 3.5.2. La distancia de medición se va a fijar a 22 cm y el QR se analizará en el canal de luminancia como se ha hecho anteriormente.

En la Tabla 3.3 se puede ver el resultado de analizar la capacidad de corrección de errores del sistema variando el balance de blancos entre 8000K y 2000K con un tiempo de exposición de 1/500 s y una ISO de 200. Se puede notar cómo el balance de blancos no tiene ninguna influencia en la detección de errores ya que el sistema siempre es capaz de detectar el número máximo de errores. Seguidamente se comprueba la influencia de variar la ISO entre 100 y 3200 en la capacidad de corrección, véase la Tabla 3.4. En este caso se fijó el tiempo de exposición a 1/1000 s y el BW a 5800 para poder emular unas condiciones de muy poca luminosidad para la ISO más baja. En este

caso sí se puede ver cómo en los casos de ISO más extremos el QR no llega a detectarse porque hay muy poca luz o se genera mucho reflejo por una luminosidad muy alta. A pesar de tener estos buenos resultados para el caso de QR con errores impresos en negro se puede entrever que esto no será así si los errores están impresos en algún color. Los errores en color se verían más afectados a los cambios de luminosidad que el color negro porque éste no refleja la luz. Este análisis se hará más adelante cuando se evalúe el sistema final bajo distintas condiciones de luminosidad.

BW (K)	8000	6000	5200	4400	3600	2800	2000
Errores	14	14	14	14	14	14	14

Tabla 3.3. Evaluación de la capacidad de corrección de errores variando el balance de blancos (BW).

ISO	100	200	400	800	1600	3200
Errores	NR	14	14	14	14	NR

Tabla 3.4. Evaluación de la capacidad de corrección de errores variando la ISO. NR = No reconocido.

3.5.4 Influencia de los errores en color. Análisis en luminancia

En esta sección se va a abordar un aspecto fundamental de nuestro sistema, su rendimiento cuando los errores son impresos en colores distintos al negro. Este procedimiento es fundamental ya que en la realidad los analitos no dan cambios de color tan sencillos como de un blanco a un negro, sino que pueden variar de gamas de color dependiendo de los reactivos que se usen. Para todos los análisis que se lleven a cabo, a continuación se va a utilizar el código QR descrito en la sección 3.5.2, pero modificando los colores de los errores impresos identificados en la Figura 3.10. También se va a fijar una distancia de medición a 9 cm, que supone una distancia más cercana la que se usaría en unas condiciones de uso real. Los parámetros de la cámara también se van a mantener por defecto para evitar tener un excesivo control sobre estos en condiciones reales. En este primer caso se va a analizar el código QR en luminancia y se va a estudiar el efecto de emplear errores en color. Para ello hay que tener en cuenta, como se explicó en la sección 3.2.2, que la imagen captada por la aplicación es convertida al espacio YUV y de este canal se extrae el valor Y de luminancia que es un valor equivalente a la conversión de la imagen a nivel de gris. Por ello se va a recordar que para convertir un color RGB a nivel de gris en la aplicación se usa la siguiente ecuación:

$$Y = \text{redondear}(0.299R + 0.587G + 0.114B) \quad (33)$$

En este experimento se van a utilizar 16 códigos QR y en cada uno de ellos se van a imprimir errores de un color distinto, identificándolos con un número del 1 al 16.

Color (id)	Color (R,G,B)			Gris	Errores	Posiciones
1	239	59	57	113	14	Todas
2	229	93	101	135	14	Todas
3	233	134	135	164	3	8, 10, 14
4	69	83	164	88	14	Todas
5	116	127	183	130	5	1, 3, 10, 12, 13
6	174	181	210	182	2	4, 10
7	56	197	243	160	3	5, 12, 14
8	145	214	234	196	1	9
9	200	229	236	221	0	0
10	116	192	68	155	12	Todas - 2, 10
11	157	206	121	182	4	2, 4, 8, 12
12	200	226	173	212	2	6, 9
13	245	235	15	213	4	1, 4, 9, 10
14	240	233	146	225	0	0
15	238	234	198	231	1	14
16	0	0	0	0	14	Todas

Tabla 3.5. Comparación del número de errores y patrones de error específicos detectados para distintos colores.

En la Tabla 3.5 se presenta una muestra de los colores junto con su número de identificación, el valor en RGB que se imprimió en el *software* de diseño. Los colores han sido repartidos en 5 tonos distintos y de cada tono se han generado 3 colores variando su saturación. De esta forma, para cada tono se obtiene una variación de saturación similar a la que podríamos encontrar en

un analito real, color de control, color positivo y color negativo. En cada tonalidad el color de mayor saturación simula el control, el de saturación media simula el positivo y el de saturación baja simula el negativo. Por ejemplo, para el caso del tono rojo el control sería el color 1, el positivo el color 2 y el negativo el color 3. En la Tabla 3.5 también se muestra el nivel de gris teórico calculado para cada color. Este valor es puramente orientativo ya que puede variar dependiendo de la luminosidad y la impresora utilizada, pero será útil para dar explicación a los fenómenos que tienen lugar en este experimento.

En la Tabla 3.5 se puede observar cuántos errores han sido identificados y, en el caso de que no se hayan identificado todos los errores, se indican las posiciones de error que se han detectado, usando el sistema de identificación basado en la numeración mostrada en la Figura 3.10. En el caso del QR con color (id) igual a 10, se han indicado las dos posiciones de error que no se han detectado. Se va a considerar una detección correcta del patrón de error si se pueden detectar más del 80% de los errores impresos, que se han resaltado en la tabla con las celdas sombreadas en verde. Un primer objetivo de la aplicación sería detectar aquellos colores con una saturación de color más alta (que podrían simular los controles) para posteriormente tratar de detectar el siguiente escalón de saturación (equivalente a un positivo del analito). En cualquier caso, se requiere que el sistema no detecte los colores de baja saturación (similares al analito negativo).

En los resultados se observa cómo el sistema solo es capaz de detectar correctamente 3 de los 5 colores que simulan un control más el negro. Además, solo en el caso del tono rojo se identifica el color que estaría asociado a un positivo, color (id) igual a 2, que tiene una saturación menor. Sin embargo, el resto de colores que simulan positivos en el resto de tonalidades, colores (id) igual a 5, 8, 11, 14, no son detectados. Esto puede deberse principalmente a que el nivel de gris equivalente de ese color es más bajo que el umbral de detección del QR. Para estos análisis, el umbral de detección extraído de la aplicación está en torno a 152, por este motivo muchos colores que perceptualmente vemos claramente cuando se toma su luminancia pierden mucho contraste porque varias de sus componentes RGB están muy cercanas a 255, como puede ser el caso del color 13. Para resolver este problema se van a evaluar las soluciones propuestas en las Secciones 3.2.2 y 3.2.3.

3.5.5 Detección en luminancia del código QR impreso del color del control.

En esta solución se va a buscar ajustar el umbral de detección según los niveles de gris de los controles utilizados. Para este caso se requería que todo el QR fuera impreso con el reactivo del control de forma que si éste aparecía diera el análisis como correcto a falta de detectar los módulos erróneos dentro del QR. Para emular este procedimiento se van a imprimir los códigos QR diseñados en la sección 3.5.2 completamente con un color que emule al del reactivo de control, incluyendo sus errores. Se sigue analizando el QR en el canal de luminancia de forma que no se modifica el proceso de detección original del lector de códigos QR. En la Tabla 3.6 se presenta una comparativa de los resultados obtenidos con esta solución y de los resultados de la sección anterior, donde solo se imprimen en color los módulos erróneos. En esta tabla se ha añadido, además del nivel de gris de cada color, el umbral de detección usado por la aplicación para detectar ese QR.

Color (id)	Gris	Umbral	Color en módulos		Color QR completo	
			Errores	Posiciones	Errores	Posiciones
1	113	144	14	Todos	14	Todas
2	135	136	14	Todas	15	Todas + 11/16N
3	164	0	3	8, 10, 14	NR	NR
4	88	136	14	Todas	14	Todas
5	130	0	5	1, 3, 10, 12, 13	NR	NR
6	182	0	2	4, 10	NR	NR
7	160	0	3	5, 12, 14	NR	NR
8	196	0	1	9	NR	NR
9	221	0	0	0	NR	NR
10	155	0	12	Todas - 2, 10	NR	NR
11	182	0	4	2, 4, 8, 12	NR	NR
12	212	0	2	6, 9	NR	NR
13	213	136	4	1, 4, 9, 10	14	Todas

14	225	0	0	0	NR	NR
15	231	0	1	14	NR	NR
16	0	152	14	Todas	14	Todas

Tabla 3.6. Comparación del número de errores y los patrones de error detectados aplicando el color solo en los módulos de error o al QR completo (NR: No Reconocido)

Se puede observar que no hay diferencias sustanciales en la detección correcta de los patrones de error para este experimento. Sin embargo, sí se puede ver que la solución no es capaz de reconocer el QR, marcado como NR en la fila “Error” de esta tabla, cuando el color tiene una saturación más baja. Se puede notar que el sistema que usa color sólo en los módulos erróneos sí que es capaz de detectar el código QR y algunos errores en este caso. En el caso de que la solución propuesta detecte el patrón, ésta identifica todos los errores, incluyendo algún falso negativo como en el caso del color (id) 2. En este caso, se ha detectado un falso negativo en la posición (11,16) del QR indicando un error de negro a blanco, identificado como 11/16N en esta tabla. Es bastante probable que esta solución sea muy dependiente de la luminosidad, ya que puede que haya fuentes de luz que permitan que ciertos códigos QR se identifiquen mejor. Esta solución, siendo sencilla, está muy restringida a colores que tengan un contraste muy alto y por este motivo está muy limitada. Es importante tener un sistema que sea capaz de detectar el mayor número de colores para así permitir una mayor detección de analitos.

3.5.6 Detección de errores del código QR en canales separados RGB

Esta implementación busca dar solución al problema encontrado en las secciones anteriores mediante una detección por canales separados en RGB, tal y como se explicó en la sección 3.2.3. En este caso, el código QR es impreso en negro y los módulos erróneos se imprimen en distintos colores, pero en este caso se va a realizar un análisis del QR en cada canal RGB. De esta forma se espera que, al menos en algún canal, el sistema sea capaz de detectar los errores impresos de los colores de control y positivos. Estos experimentos son realizados con el código QR presentado en la sección 3.5.2.

En la Tabla 3.7 se muestran los resultados del experimento realizado donde se presentan el número de errores detectados (E) y sus posiciones de error (P) en cada canal RGB. Se puede observar que, si se combinan las detecciones en

cada canal R, G o B por separado, se logra detectar correctamente todos los colores que emularían los controles del analito (colores 1, 4, 7, 10, 13, 16). En el caso de los colores que emularían los resultados positivos, se han identificado 2 de los 5 posibles colores (2, 11) con una efectividad del 80%. Los colores positivos 5 y 14 han mostrado una detección muy alta, pero sin llegar al estándar de detección correcta. En ningún caso se han detectado más del 80% de los colores negativos, que emulan la no presencia del analito. El color 13 es un ejemplo de que el sistema funciona correctamente ya que en el canal R y G el sistema no es capaz de realizar una detección de errores correcta, pero en el canal B sí puede porque en esa componente el color tiene un mayor contraste respecto al blanco. Este fenómeno se repite para cada color, demostrando así una mejor detección que en las soluciones anteriores. A pesar de los buenos resultados, esta solución aún no es capaz de identificar correctamente todos los errores de ciertos colores positivos (5 y 14). Por este motivo se va a realizar una evaluación del rendimiento del sistema combinando las dos soluciones, detección en canales RGB con códigos QR impresos con el color del control.

Color (id)	(R,G,B)			R		G		B	
				E	P	E	P	E	P
1	239	59	57	0	0	14	Todas	14	Todas
2	229	93	101	0	0	14	Todas	14	Todas
3	233	134	135	0	0	6	2, 6, 8, 10, 11, 14	7	2, 6, 8, 10, 11, 13, 14
4	69	83	164	14	Todas	14	Todas	0	0
5	116	127	183	9	Todas - 2, 4, 6, 7, 11	5	1, 3, 10, 12, 13	0	0
6	174	181	210	3	4, 10, 14	2	4, 10	2	4, 10
7	56	197	243	12	Todas - 2, 9	1	14	0	0
8	145	214	234	2	7, 9	0	0	0	0
9	200	229	236	0	0	0	0	0	0

10	116	192	68	12	Todas - 2, 10	6	4, 5, 6, 8, 13, 14	14	Todas
11	157	206	121	3	2, 4, 8	0	0	12	Todas - 13, 14
12	200	226	173	2	6, 9	1	9	6	1, 4, 6, 9, 11, 14
13	245	235	15	0	0	2	9, 10	14	Todas
14	240	233	146	0	0	0	0	8	Todas - 2, 4, 5, 8, 9, 12
15	238	234	198	0	0	1	14	1	11
16	0	0	0	14	Todas	14	Todas	14	Todas

Tabla 3.7. Análisis del número de errores (E) y las posiciones de error (P) detectados para cada color analizando el código QR en los canales RGB de forma independiente.

En la Tabla 3.8 se pueden ver los resultados de este experimento, en el que se puede observar que ocurre algo similar a lo ocurrido en la sección 3.5.5. No se detectan tantos códigos QR como en el experimento anterior y además este procedimiento favorece la aparición de muchos más falsos negativos.

Color (id)	(R,G,B)			R		G		B	
				E	P	E	P	E	P
1	239	59	57	NR	NR	14	Todas	14	Todas
2	229	93	101	NR	NR	15	Todas + 11/16N	15	Todas + 11/16N
3	233	134	135	NR	NR	NR	NR	NR	NR
4	69	83	164	14	Todas	14	Todas	NR	NR
5	116	127	183	NR	NR	NR	NR	NR	NR
6	174	181	210	NR	NR	NR	NR	NR	NR

7	56	197	234	17	Todas + 11/15N, 12/14N, 15/10N	NR	NR	NR	NR
8	145	214	234	NR	NR	NR	NR	NR	NR
9	200	229	236	NR	NR	NR	NR	NR	NR
10	116	192	68	NR	NR	NR	NR	14	Todas
11	157	206	121	NR	NR	NR	NR	NR	NR
12	200	226	173	NR	NR	NR	NR	NR	NR
13	245	235	15	NR	NR	NR	NR	14	Todas
14	240	233	146	NR	NR	NR	NR	NR	NR
15	238	234	198	NR	NR	NR	NR	NR	NR
16	0	0	0	14	Todas	14	Todas	14	Todas

Tabla 3.8. Análisis del número de errores (E) y las posiciones de error (P) detectados para cada color analizando el código QR en los canales RGB de forma independiente. NR = No reconocido.

3.5.7 Comparativa de las soluciones

Para resumir la evaluación de las soluciones propuestas se va a realizar una comparativa de los experimentos realizados y se discutirá brevemente cuál es la solución que puede aportar más beneficios y con la que se va a realizar la experimentación con detección de patrones de diagnóstico. En la Tabla 3.9 se puede ver una comparativa de todas las soluciones evaluadas. Se aprecia que el análisis en RGB es el que aporta mayor número de detecciones correctas, siendo el caso del QR impreso en negro el que mejor rendimiento da. Además del número de errores detectados, la solución del QR impreso en negro y el análisis en RGB aporta una mayor flexibilidad en el análisis de los analitos y más facilidad para la fabricación del código QR. Esta solución también da la posibilidad de realizar el análisis de varios analitos en el mismo símbolo QR, ya que podrían imprimirse los controles y los analitos en módulos separados sin tener que imprimir el QR completo del color del control. Esta solución también supone un coste computacional superior porque requiere la realización de 3 detecciones por cada QR. Sin embargo, el tiempo de ejecución no aumenta excesivamente pudiendo llegar como máximo a 1 segundo por encima

del registrado en el análisis del código QR en luminancia. Este sistema también daría la posibilidad de analizar cambios de color no solo en saturación sino en tono, ya que podríamos analizar cada canal por separado y establecer las diferencias entre ellos.

Tipo QR	Tipo de análisis	Colores detectados (80% de los errores)
QR negro - módulos negros	En luminancia	5
QR color - módulos color	En luminancia	5
QR negro - módulos color	En RGB	8
QR color - módulos color	En RGB	7

Tabla 3.9. Comparativa de las soluciones en el rendimiento con color.

Por todo esto se puede afirmar que la solución con el QR impreso en negro, los errores y control en color y el análisis en RGB es la que mejor rendimiento aporta. Sin embargo, aún queda en el aire el fallo que este sistema dio en la detección de algunos colores positivos. Para resolver este problema se va a implementar la calibración del umbral expuesta en la sección 3.2.4.

3.5.8 Calibración del umbral para el ajuste de la detección de analitos

En este apartado se va a realizar la calibración del umbral de detección para cada tono de color, que simula a un analito con un control, un positivo y un negativo. El objetivo de esta calibración es definir un umbral que permita la detección correcta del control y el positivo, sin detectar el negativo. Para ello se ha repetido el análisis del QR con módulos impresos con análisis en RGB pero iterando el proceso hasta encontrar un porcentaje de cambio del umbral $U(\%)$ que produzca los resultados deseados.

Color (id)	U(%)	nº errores			
		Gris	R	G	B
1	0	14	0	14	14
2	0	14	0	14	14
3	0	3	0	6	7

4	10	14	14	14	7
5	10	10	12	11	4
6	10	3	3	3	1
7	17	10	14	11	1
8	17	4	12	3	0
9	17	2	4	4	3
10	5	13	13	7	14
11	5	6	3	2	13
12	5	1	1	2	5
13	5	14	0	2	14
14	5	0	0	0	12
15	5	1	0	1	1
16	0	14	14	14	14

Tabla 3.10. Ajuste del incremento del umbral para mejorar la detección de los errores de color del QR.

En la Tabla 3.10 se pueden ver los resultados de la calibración donde se demuestra que el ajuste del umbral para cada analito favorece la detección correcta de éste y permite cierto nivel de ajuste en casos donde los colores de los analitos no lleguen a ser detectados por el sistema. Una vez comprobada la funcionalidad del control del umbral conviene destacar que este experimento se realizó en condiciones de luminosidad muy concretas y que este umbral podría verse modificado cuando las condiciones de luminosidad cambien drásticamente. Más adelante se hará un estudio de este fenómeno cuando se analice la influencia de las condiciones de luminosidad en el sistema final.

3.5.9 Detección de patrones de diagnóstico con uno y varios analitos.

Una vez definida cuál va a ser la implementación final del sistema para la detección de errores en color es momento de evaluar el rendimiento de ésta cuando se quieren comparar patrones de error concretos. Para este experimento se ha diseñado un símbolo QR como el de la Figura 3.11. En ésta se han definido 4 posiciones que se imprimirán con el color del control (marcadas

en rojo) 5 posiciones donde irán los colores que simulan positivos (marcadas en verde) y 5 posiciones donde se imprimirán los colores que no deben detectarse, que simularían los negativos (marcadas en azul). Este experimento se realizará para los 5 tonos de color definidos en los experimentos anteriores. Cada grupo de color del patrón de diagnóstico consta de tres colores en el mismo tono (un control, un positivo y un negativo) donde el negativo es el único que no se tiene que detectar. La detección positiva del color se da cuando se han detectado al menos el 80% de las réplicas impresas.



Figura 3.11. Diseño de un QR para el testeo de la detección de patrones de diagnóstico. Módulos con borde rojo = posiciones de control, módulos con borde verde = posiciones positivas y módulos con borde azul = posiciones negativas.

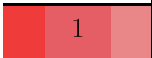
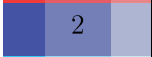


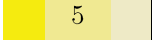
Color (id)	Control	Positivo	Negativo
 1	OK	OK	X
 2	OK	OK	X
 3	OK	OK	X
 4	OK	OK	X
 5	OK	OK	X

Tabla 3.11. Comparación de la detección de patrones de diagnóstico para distintos colores.

En la Tabla 3.11 se presentan los resultados de este experimento en la que se indica con un OK si se ha detectado el patrón del control, del positivo o del negativo y con una X en caso de que el patrón no se haya detectado correctamente (número de errores este por debajo del 80%). En cada color se ha usado el ajuste de umbral definido en la sección anterior. Se puede ver que

todas las detecciones han sido correctas para los 5 colores, ya que en todos los casos solo se han detectado el control y el positivo.

A continuación, se hará un experimento adicional para mostrar la capacidad de análisis multianalito que puede mostrar este sistema. Se ha diseñado un QR para la detección de 2 analitos ficticios (analito 1 y analito 2) simultáneamente como se puede ver en la Figura 3.12. Cada analito cuenta con 12 módulos para su detección, 4 para el control y 8 para el analito. En la Figura 3.12 el analito 1 se depositaría en las posiciones marcadas en verde y su control en las posiciones marcadas en rojo. Por otro lado, el analito 2 se depositaría en las posiciones marcadas en azul y su control en las posiciones marcadas en amarillo. Para aumentar el número de módulos erróneos detectables se han agrupado módulos con sustancia similar de forma adyacente para que los errores que se hayan detectado formen parte del mismo símbolo, como se explicó en la sección 3.5.1. En este experimento se van a analizar cuatro códigos QR para abarcar todas las posibilidades de detección (solo analito 1 positivo, solo analito 2 positivo, todos positivos, todos negativos). Para simular este experimento se van a imprimir las posiciones marcadas con dos paletas de color usadas previamente. En este caso se usaron los colores 1 y 4 para imprimir el control 1 y el control 2. El analito 1 y el analito 2 se imprimieron con los colores 2 y 5 para simular la detección positiva y con los colores 3 y 6 para simular su detección negativa, respectivamente. Se utilizó un ajuste de umbral del 5% en ambos casos. En la Tabla 3.12 se muestran los resultados de este análisis, mostrando la imagen del QR detectado e indicando debajo el patrón específico que representa y si su detección fue correcta, marcada con un OK. Se puede ver cómo todas las detecciones han sido realizadas correctamente.



Figura 3.12. Diseño de un QR para el análisis de dos analitos simultáneamente. Módulos de borde rojo = posiciones control 1, borde amarillo = posiciones control 2, borde verde = posiciones analito 1 y borde azul = posiciones analito 2.





			
Analito 2 positivo	Analito 1 positivo	Todos positivos	Todos negativos
OK	OK	OK	OK

Tabla 3.12. Resultados de los análisis para un código QR multianalito.

Este sistema permite actualmente una detección de analitos que compartan el mismo ajuste de umbral. En caso de que no fuera así, se dificultaría el análisis, aunque a futuro podría valorarse la posibilidad de resolver esta limitación. El número de analitos a detectar puede aumentar sin problema siempre que se distribuyan los errores para que haya más errores por símbolo. Modificando el tamaño del QR, su nivel de corrección de errores y la disposición de los errores, se podrían llegar a analizar una gran cantidad de analitos en el mismo código QR. Es importante notar que hemos trabajado con los analitos que generan errores de blanco al negro, sin embargo, este sistema sería igualmente válido para errores generados de negro a blanco. Establecer el patrón de error correspondiente para realizar el diagnóstico sería suficiente para que funcionara correctamente.

3.5.10 Influencia de las condiciones de luminosidad en la detección de patrones de diagnóstico

El sistema propuesto solo se ha evaluado en condiciones de luminosidad muy concretas. Por lo tanto, es necesario realizar un análisis de su rendimiento cuando las condiciones de luminosidad son modificadas. Se va a analizar el código QR propuesto en la Figura 3.11 bajo distintas condiciones de luminosidad (variación de balance de blancos y variación de ISO) para los 5 tonos principales de color. Se define una detección correcta si el patrón de control y positivo es detectado pero el patrón negativo no. Se recuerda que una detección correcta de un patrón conlleva la detección de al menos el 80% de los errores de éste. Primero se hará una variación del balance de blancos para un tiempo de exposición de 1/500 s y una ISO de 200. Los resultados de muestran en la Tabla 3.13. En este caso la detección sí se ve afectada por la temperatura de color de la fuente de luz, ya que en el caso del color 3 la

detección no es correcta para ningún balance de blancos. Para el color 4 y 5 se dan detecciones incorrectas para ciertas temperaturas de color. Sin embargo, este resultado cambia si se modifica el ajuste del umbral. Si definimos el ajuste del umbral al 25%, 10% y 15% para los colores 3, 4 y 5, respectivamente, todas las detecciones son correctas. Este resultado podría indicar que es necesario que la calibración del umbral se haga bajo distintas condiciones de luminosidad y establecer un rango del umbral óptimo en el que este código QR pueda ser analizado.

		Balance de blancos				
Color (id)	U(%)	8000	5200	3600	2000	
1	0	OK	OK	OK	OK	
2	5	OK	OK	OK	OK	
3	17	X	X	X	X	
4	5	OK	X	OK	OK	
5	5	OK	OK	X	X	

Tabla 3.13. Comparación de la detección de patrones de diagnóstico cambiando el balance de blancos de la cámara. OK = Detección correcta. X = Detección incorrecta.

		ISO						
Color (id)	U(%)	100	200	400	800	1600	3200	
1	0	X	OK	OK	OK	OK	X	
2	5	X	OK	OK	OK	OK	X	
3	17	X	OK	OK	OK	OK	X	
4	5	X	OK	OK	OK	OK	X	
5	5	X	OK	OK	OK	OK	X	

Tabla 3.14. Comparación de la detección de patrones de diagnóstico ISO de la cámara. OK = Detección correcta. X = Detección incorrecta.

Seguidamente, se realiza la evaluación de la influencia de la ISO en el rendimiento del sistema. En este caso el tiempo de exposición de la cámara se fija a 1/1000 s y el balance de blancos se mantiene en 5800 K. En la Tabla

3.14 se muestran los resultados derivados de este experimento, en la que se puede apreciar cómo para los valores de ISO considerado, en el rango de 200 a 1600 la detección se realiza correctamente, mientras que para los valores de ISO extremos a este rango, 100 y 3200, el QR no se detecta.

3.5.11 Análisis con otros dispositivos.

Para terminar la evaluación de nuestro sistema se va a realizar un análisis de la influencia del dispositivo utilizado para la captación de las imágenes en el proceso de diagnóstico. Se va a analizar el código QR anterior en tres dispositivos distintos, bajo condiciones de luminosidad estándar y usando la caja de luz con una temperatura de color de 6500 K. En la Tabla 3.15 se observan los resultados de este análisis comparando los resultados obtenidos con el dispositivo Pocophone F1 con obtenidos con un dispositivo Xiaomi Redmi note A1 y una cámara Canon EOS 70D. Se puede observar cómo los resultados son correctos para todos los dispositivos si se mantiene el ajuste del umbral calibrado previamente. De este modo, se puede concluir que nuestro sistema no depende del dispositivo utilizado para captar las imágenes.






Color (id)	Pocophone F1	Redmi note A1	Canon EOS 70D
 1	OK	OK	OK
 2	OK	OK	OK
 3	OK	OK	OK
 4	OK	OK	OK
 5	OK	OK	OK

Tabla 3.15. Comparación del rendimiento del sistema entre distintos dispositivos.

3.6 Conclusiones

En este capítulo se ha desarrollado un sistema de detección de sensores colorimétricos incrustados en códigos QR mediante el uso de *smartphones*. El método de detección empleado está fundamentado en la identificación de los errores que la aparición del analito introduce en el código junto con la detección del QR en distintos canales RGB. Esta metodología permite la detección de módulos de distintos colores incluyendo la posibilidad de detectar colores distintos en el mismo código QR simulando una detección multianalito. Se ha desarrollado una aplicación móvil capaz de ejecutar este algoritmo in-situ sin

necesidad de conexión a internet y con la capacidad de funcionar en distintos dispositivos y bajo distintas condiciones de luminosidad sin necesidad de usar la luz del flash del dispositivo. Esta aplicación permite el análisis de múltiples códigos QR guardando los resultados en una base de datos en el dispositivo.

Este sistema presenta una solución escalable a distintos tipos de analitos mediante la creación de una base de datos de códigos QR estandarizados que definan patrones de error específicos dependiendo de la necesidad del usuario. Como trabajo futuro es fundamental poder testear este sistema con el mayor número de muestras reales y estandarizar los distintos tipos de códigos QR colorimétricos que se pueden analizar. Finalmente, sería necesario implementar una arquitectura en la nube que aloje la aplicación y los datos necesarios para que ésta se mantenga actualizada, además de implementar un servicio de registro y una capa de seguridad para proteger los datos de la aplicación.

Capítulo 4

Biosensor multianalito basado en un código QR colorimétrico

Este capítulo comenzará presentando la conceptualización del sistema y de los algoritmos diseñados para determinación de la concentración del analito mediante códigos QR. Posteriormente, se describirá la metodología experimental seguida y el procedimiento de implementación del *software*. El capítulo continuará abordando los resultados obtenidos del testeo y optimización del sistema bajo distintas condiciones de medición y luminosidad. También se realizará la determinación de la concentración de glucosa y lactato en muestras reales de suero y sudor mediante el sensor colorimétrico integrado en el código QR. Finalmente, se presentarán las conclusiones, las líneas de trabajo futuro y las contribuciones derivadas de este trabajo.

4.1 Introducción

En el capítulo anterior se presentó un sensor colorimétrico basado en códigos QR que era capaz actuar como un test de diagnóstico. Este sistema estaba basado en la detección de errores dentro del código en el que se demostró que se podían detectar cambios de distintos colores bajo distintas condiciones de luminosidad. Sin embargo, no se implementó la determinación de la concentración del analito detectado en el código QR. Uno de los principales problemas que se presentan para determinar la concentración de un analito con un dispositivo móvil es la diferencia entre las condiciones de luminosidad en el momento de la medición y en el momento de la caracterización del analito. Los estudios sobre el análisis de sensores colorimétricos mediante el uso de *smartphones* están muy presentes en la literatura, mostrando diferentes soluciones ante el problema de la calibración de las condiciones de luminosidad.

Una de las correcciones de luminosidad más empleadas consiste en la normalización de los valores RGB del analito en función de referencias impresas de color blanco o negro, ya que éstas tienen un valor conocido 255 y 0 respectivamente [103–105]. También puede encontrarse en la literatura calibraciones de luminosidad basadas en modelos de *machine learning*. En estos métodos es necesario realizar un entrenamiento del sistema introduciendo un gran nú-

mero de imágenes del sensor colorimétrico bajo distintas condiciones de luminosidad para que éste pueda ser reproducible bajo distintas condiciones. En, [106] se realiza un entrenamiento de un modelo *Support Vector Machine* (SVM) combinándolo con el uso de una carta de color. También, en [107] se presenta un método que combina el uso de un modelo *Random Forest* junto con un algoritmo de constancia de color. Otro de los métodos más representados en la literatura es la corrección y calibración del color usando rectas de referencia. Cada analito cuenta con una recta de caracterización de color en la que cada color está asociado a una concentración determinada. En este caso el objetivo es tomar la recta de referencia y el analito bajo las mismas condiciones de luminosidad y posteriormente utilizar un algoritmo de distancia para determinar la concentración en función de la recta de referencia. En el caso de [23] este proceso se realiza mediante dos fotos, una con las rectas de calibración y otra con el analito combinado con un algoritmo de mínima distancia aplicado sobre el espacio de cromaticidad xy . Un avance muy relevante es el presentado por [21], en el que se diseña una tira reactiva en la que el elemento reactivo es colocado en el centro de un círculo donde se disponen las referencias de dicho analito. Esta innovación permite la realización de la calibración y la medición con una sola fotografía.

En este capítulo se pretende integrar un sensor colorimétrico dentro de un código QR de forma que facilite el proceso de detección del sensor haciendo uso de los patrones de posicionamiento y alineación del símbolo QR. Esto permite, igual que en el capítulo anterior, tener un código QR capaz de almacenar una información estática y otra dinámica, la concentración del analito. Se pueden encontrar publicaciones en las que se diseñan sensores colorimétricos integrados en patrones de posicionamiento de un símbolo QR para facilitar su detección y procesamiento [28,29]. Este diseño, aunque innovador, pierde la posibilidad de detectar la información estática del código QR, ya que ésta es eliminada completamente para integrar el sensor colorimétrico. En [26] se presenta un sensor colorimétrico basado en la activación mediante una fuente de luz integrado en un código QR. En este caso la aparición del analito posibilita la detección del código QR, manteniendo la información estática del código. En este caso la concentración del analito es determinada en función del tiempo que tarda en detectarse el código QR. Por otro lado, [27] presenta sensor colorimétrico para la detección de malaria impreso en un código QR y detectado mediante una aplicación móvil. En este sistema todo el símbolo QR está impreso con el reactivo, sin embargo, la aplicación móvil es capaz de

detectar una línea de módulos dentro del código QR que sirven para normalizar la luminosidad y determinar un valor de luminosidad normalizado.

En este trabajo se propone integrar un sensor colorimétrico en un código QR, pero manteniendo su información estática. Se va a implementar una aplicación móvil capaz de detectar la posición de los módulos del símbolo QR donde se integra el sensor colorimétrico y realizar la corrección de luminosidad. Esta aplicación combina los métodos de corrección basados en las referencias blancas y negras junto con una recta de referencias de color. En este capítulo se propone un método para la corrección de la luminosidad basado en la corrección geométrica de la recta de referencias respecto a una calibración estándar. Este sistema se desarrolla para la determinación de la concentración de glucosa y lactato en un mismo símbolo QR, lo que posibilita una capacidad de detección multianalito. La aplicación móvil realiza la detección y el almacenamiento de la información en el dispositivo sin conexión de red, sin embargo, también se presenta el diseño de una arquitectura en la nube donde se integraría este sistema.

4.2 Conceptualización del sistema

En esta primera sección se presentará un esquema general de la solución propuesta para proporcionar una vista general de las funcionalidades del sistema. Posteriormente, se mostrarán las características que debe tener el diseño del símbolo QR y se explicarán los algoritmos desarrollados para realizar la corrección del color y obtener una medida precisa de la concentración. Finalmente, se presentará una propuesta conceptual de una arquitectura de un sistema en la nube que podría ser implementada para la solución propuesta.

4.2.1 Esquema general del sistema

En la Figura 4.1 se presenta un esquema general del sistema que servirá para explicar sus elementos, funcionalidades y cómo interactúan entre ellos. La solución propuesta comienza con el diseño y fabricación de un código QR que actúa como biosensor. Éste se diseña superponiendo tres capas: la capa de soporte para los datos codificados del código QR (información estática), los módulos de referencia para la corrección de la señal (elementos de corrección) y los módulos biosensores para la detección analítica (información dinámica). El código QR está integrado en un dispositivo de sujeción, para facilitar su manipulación y evitar daños al código y al biosensor. Esto también facilita

el análisis desde cualquier ángulo, una característica deseable cuando se utilizan *smartphones* como elemento de lectura.

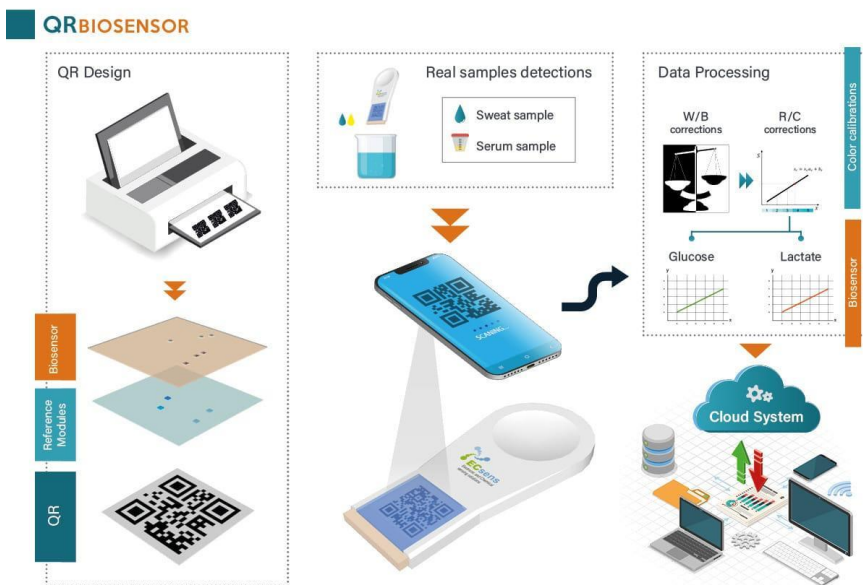


Figura 4.1. Esquema general del sistema [108].

En el *smartphone* se implementa un *software* de detección de códigos QR que es capaz de decodificar los datos del código (información estática) y la información del analito (información dinámica) que contiene, además de corregir los datos mediante los módulos de referencia (elementos de corrección). El sistema que se ha propuesto siguiendo este esquema es capaz de determinar simultáneamente la glucosa y el lactato a partir de muestras de suero y sudor, permitiendo así analizar múltiples analitos. La información estática obtenida no se ve afectada por cuestiones como la luz, el posicionamiento del *smartphone* o diferencias entre los dispositivos empleados, debido a que el algoritmo de corrección de errores inherente a los códigos QR, como el de Reed Solomon, compensa todos estos posibles errores. Una vez que se detecta el código QR, el *software* desarrollado e implementado en el dispositivo móvil extrae el color del analito afectado por la influencia de la luz. Para determinar correctamente la concentración del analito utilizando las curvas de caracterización analíticas, es necesario calibrar el color medido. Los algoritmos desarrollados en esta solución para la corrección de color emplean dos métodos diferentes: corrección con blanco y negro (método W/B) y correcciones de la curva de color de referencia (método R/C). Estos métodos se explicarán en profundidad más adelante. Tras esta corrección el sistema sería capaz de medir la concentración

de cada analito compensando las variaciones de luminosidad ambiente. Aunque el sistema está preparado para funcionar *offline*, lo que permite su uso en lugares donde no haya acceso a Internet, se conceptualizó una plataforma de computación en la nube con el fin de compartir e interconectar los resultados obtenidos por la aplicación móvil con cualquier otro servicio en línea.

4.2.2 Diseño del código QR

El diseño del código QR debe cumplir las especificaciones requeridas por el sistema presentado en la sección 4.2.1. El código QR debe contener una capa de datos estática, unos módulos de referencia como elementos de corrección y módulos del biosensor (la información dinámica). La Figura 4.2 presenta el diseño final del código QR. Suponiendo que la luz es espacialmente invariante, lo cual es una suposición razonable para los tamaños de códigos QR utilizados aquí, 22 mm, la posición de los módulos específicos no sufrirá grandes cambios por la variación espacial de la luz. Sin embargo, las posiciones de los módulos de referencia como la de los biosensores están distribuidas en el código QR para tener una mejor información de la distribución espacial de la luz en éste. Las regiones marcadas con bordes cuadrados verdes se utilizan como referencias en blanco y negro para el método de corrección W/B. Estas regiones permanecen constantes en cada código QR, independientemente del tamaño de los datos.

Por otro lado, los módulos de referencia, coloreados en azul en la Figura 4.2, corresponden a una curva lineal en el mismo espacio de color que la respuesta colorimétrica del biosensor. En nuestro caso, la glucosa y el lactato presentan una respuesta colorimétrica con variación en la saturación de color en el espacio de color HSV. De esta forma, la curva de referencia impresa modela una función lineal entre la saturación medida, S_m , (extraída por el *smartphone*) y la saturación impresa, S_m , (saturaciones ideales desde el 10% hasta el 90%). En condiciones no estandarizadas, esta curva de referencia se desplaza con respecto a la curva tomada en condiciones estandarizadas (misma condiciones en las que se realizan las curvas de caracterización de los biosensores). Asumiendo que la desviación entre ambas curvas es la misma que el desplazamiento del analito medido en condiciones no estandarizadas, la saturación del analito se puede calibrar corrigiendo la variación entre ambas curvas de referencia (estandarizadas y no estandarizadas); este procedimiento se explicará en profundidad en la sección 4.2.4. Finalmente, los módulos marcados con “pequeñas moléculas” en la Figura 4.2 son los módulos biosensores, los cuales están replicados para obtener resultados más precisos. En el diseño

propuesto se tienen 3 réplicas para cada biosensor (glucosa y lactato). Esta solución permite medir un conjunto de analitos con cambios colorimétricos similares en el sistema TMB utilizando la misma curva de referencia para corregirlos, lo que permite tener la capacidad de medir analitos distintos simultáneamente.

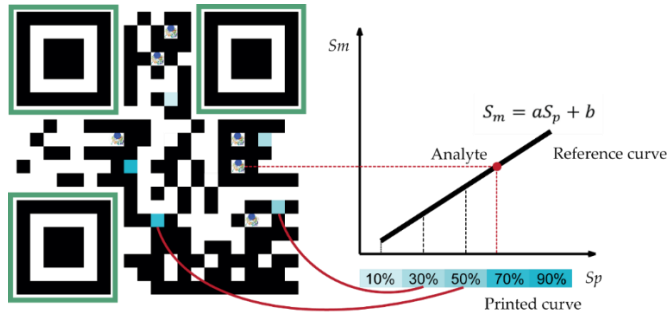


Figura 4.2. Diseño del código QR colorimétrico junto con su curva de referencia.

4.2.3 Corrección W/B

A continuación, se va a detallar el funcionamiento de uno de los métodos de compensación de la luz ambiental usados, la corrección W/B. Para apoyar la explicación de los métodos de corrección se va a hacer referencia a la Figura 4.3, donde se presenta un diagrama de bloques del analizador de códigos QR colorimétricos junto con el estado de la señal colorimétrica en cada etapa del proceso.

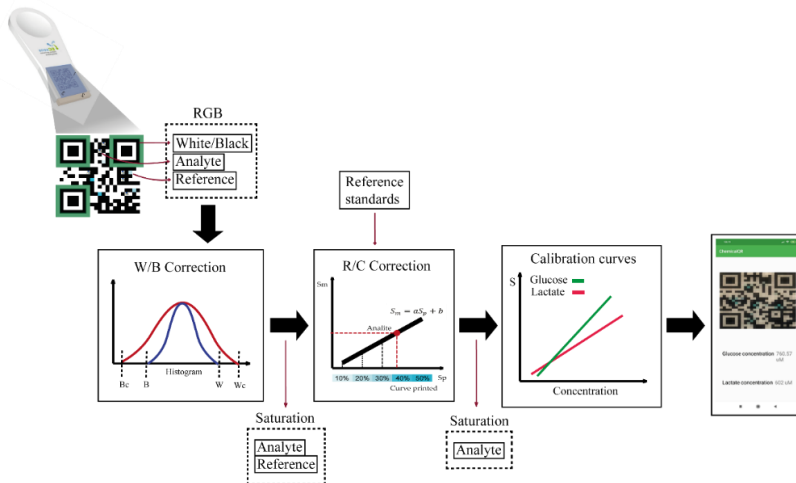


Figura 4.3. Diagrama de bloques del analizador de códigos QR colorimétrico.

Como se comentó en la sección 4.2.2, el QR está diseñado para contener tres elementos de interés para el funcionamiento de la aplicación: los módulos blancos y negros (los bordes marcados en verde en las esquinas en la Figura 4.2), los módulos de referencia (que representan la curva de referencia lineal) y los módulos de biosensor (que son objeto de nuestro análisis). Cuando el *smartphone* analiza el código QR, extrae los valores RGB de estos elementos obteniendo una señal RGB separada para cada elemento (señal de blanco/negro, señal de referencia, señal del analito). Una vez se tienen las señales en RGB el método W/B realiza una primera corrección de la señal de referencia y del analito usando las referencias blanco/negro. Posteriormente los valores corregidos son transformados a un espacio de color de interés según la aplicación específica. En este caso la saturación (S) del espacio HSV es el canal que muestra una variación con la concentración de glucosa y lactato, como se mencionó en la sección 4.2.2.

El método W/B, como se denomina en este trabajo, es una técnica típica de corrección de luminosidad que se ha aplicado anteriormente en la literatura [22,27,103]. Este método utiliza referencias de blanco y negro para corregir las señales RGB y obtener nuevos valores RGB corregidos, como en la siguiente ecuación:

$$rgb_c = \frac{255}{(rgb_w - rgb_b)} (rgb_m - rgb_b) \quad (34)$$

donde rgb_c y rgb_m son las señales RGB corregidas y medidas, respectivamente, rgb_b y rgb_w son las señales RGB de blanco y negro extraídas de los módulos dentro de las regiones marcadas en verde en las esquinas del QR como muestra la Figura 4.2. Este método se aplica tanto para las señales RGB del analito como de referencia (ver Figura 4.3). A continuación, la conversión de las señales corregidas RGB a señales S se realiza utilizando la ecuación de conversión de color espacial RGB a HSV (ecuación (11)) y extrayendo el canal de saturación de cada señal. El método de corrección W/B es especialmente importante en nuestro sistema, como muestra la Figura 4.4. Los valores de saturación de la curva de referencia están muy influenciados por la temperatura de color de la fuente de luz, deformando la tendencia de la curva, y el

método W/B corrige perfectamente este error. Para aplicar este método se debe procurar que no haya zonas sombrías en la zona del QR.

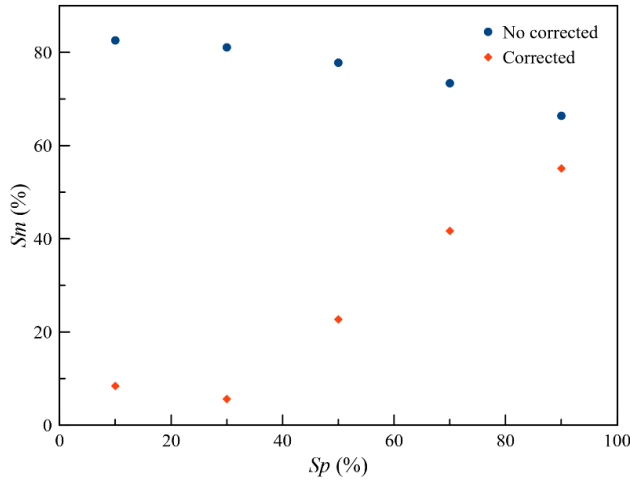


Figura 4.4. Evaluación de la corrección W/B para la curva de referencia bajo una temperatura de color de 7000 K.

4.2.4 Corrección R/C

Una vez realizada la corrección W/B el sistema cuenta con una señal de saturación corregida que podríamos introducir en la curva de caracterización analítica (corregida bajo este mismo método) para determinar la concentración del analito en cuestión (glucosa o lactato). Sin embargo, la aplicación de este método por sí solo arroja resultados para la concentración estimada poco precisos cuando las condiciones ambientales de medición (no estandarizadas) cambian respecto a las que se usaron para medir la curva de caracterización (condiciones estandarizadas). Además, la saturación es una magnitud muy variable y pequeños cambios en la luminosidad o en el dispositivo pueden ser difíciles de compensar con este método.

A raíz de estos resultados, se propuso un proceso adicional de corrección que permitiera compensar mejor estas variaciones, el método R/C (corrección por curva de referencia). El objetivo de esta corrección es utilizar una curva de referencia para mapear la saturación de los analitos tomados en condiciones no estandarizadas a unas condiciones estandarizadas. Esta curva de referencia ya fue mencionada en la sección 4.2.2 donde se vio cómo esta curva lineal se integraba dentro del QR. Para que este método funcione correctamente hace falta realizar un proceso de calibración previo, detallado en la sección 4.3.5,

que consistirá en medir la curva de referencia y las curvas de caracterización analíticas de cada analito (glucosa y lactato) bajo las mismas condiciones ambientales estandarizadas (luz, dispositivo, posición). De esta manera, el analito se puede medir en condiciones no estandarizadas siempre que la curva de referencia se mida de manera similar, por este motivo la curva de referencia y los analitos se encuentran en el mismo símbolo QR. Con este método se puede mapear el analito sobre la curva de referencia en condiciones no estandarizadas, corregirlas con el método R/C y llevar los valores a las condiciones estándar para poder calcular la concentración del analito en la curva de caracterización de éste.

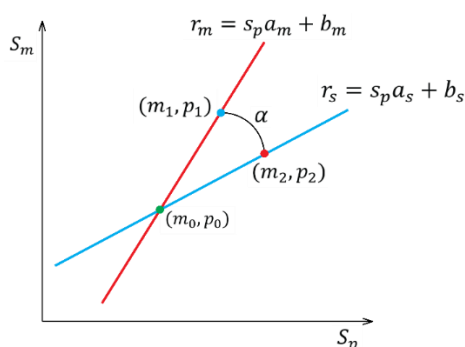


Figura 4.5. Modelado matemático de la desviación entre la curva de referencia no estándar, r_m , (curva roja) y la estándar, r_s , (curva azul).

A continuación se detallará el funcionamiento del método de corrección R/C partiendo de que todas las saturaciones (del analito y la curva de referencia) han sido corregidas con el método W/B, como se ilustra en la Figura 4.3. El objetivo de la corrección R/C es compensar la desviación entre las curvas de referencia medida en condiciones no estándar, r_m , y la medida en condiciones estándar, r_s . Estas dos curvas, que son modeladas como una recta por mínimos cuadrados en la práctica, sufren entre ellas una desviación debido al cambio de las condiciones de medida. En la Figura 4.5 se puede ver el modelado matemático de este fenómeno, el cual se representaría con una rotación entre las curvas por un ángulo α . Teniendo en cuenta que los analitos medidos y la curva de referencia son tomadas bajo las mismas condiciones no estándar, si se logra calcular el ángulo α , se podría mapear el analito medido a una saturación similar a la tomada en condiciones estándar. El método R/C calcula esta rotación usando la Ecuación (35). Esta rotación es aplicada al analito medido mapeado en la recta, r_m (m_1, p_1) para obtener el analito estimado en condiciones estándar (m_2, p_2) usando las Ecuaciones (36) y (37).

Donde (m_0, p_0) es el punto de corte entre las rectas r_m y r_s . Con este método m_2 sería una estimación de m_1 que puede ser usado en la curva analítica (de glucosa o lactato) guardada en la aplicación móvil para obtener la concentración corregida.

$$\alpha = \tan^{-1} \left(\frac{b_m - b_s}{1 + a_m a_s} \right) \quad (35)$$

$$p_2 = (p_1 + p_0) \cos(-\alpha) - (m_1 - m_0) \sin(-\alpha) + p_0 \quad (36)$$

$$m_2 = (p_1 - p_0) \sin(-\alpha) + (m_1 - m_0) \cos(-\alpha) + p_0 \quad (37)$$

4.3 Materiales y métodos

En esta sección se describen todos los métodos experimentales y los materiales utilizados. En primer lugar, se presentarán los reactivos y materiales utilizados con fines experimentales, así como el *hardware* y *software* empleados para medir e implementar los algoritmos. Posteriormente, se detallará el proceso de fabricación del código QR biosensor. Finalmente, se abordará el proceso de medición de las muestras, incluidas muestras reales, y el proceso de calibración seguido para obtener las curvas de caracterización y de referencia estandarizadas.

4.3.1 Reactivos y materiales

En el desarrollo experimental de este capítulo se utilizarán los siguientes reactivos adquiridos de Sigma Aldrich: glucosa, fructosa, urea, ácido úrico, lactato, ácido pirúvico, cloruro de sodio (NaCl), cloruro de potasio (KCl), cloruro de amonio (NH₄Cl), cloruro de magnesio (MgCl₂), cloruro de calcio (CaCl₂), sulfato de calcio (CaSO₄), hidrogeno carbonato de sodio (NaHCO₃), sodio dihidrógeno fosfato (NaH₂PO₄), DL-cisteína, peroxidasa de rábano (HRP), glucosa oxidasa (GOx), lactato oxidasa (LOx), 3,3',5,5'-tetrametilbenzidina (TMB) y etanol. Todos los reactivos eran de grado reactivo analítico a menos que se indique lo contrario. Para imprimir el código QR se utiliza papel de cromatografía Whatman No 1 de Sigma-Aldrich. Todas las soluciones acuosas se prepararon utilizando agua de calidad del tipo de ósmosis inversa (estación Milli-RO 12 plus Milli-Q de Millipore, conductividad 18,2 M Ωcm).

4.3.2 *Hardware y software*

Para imprimir los códigos QR se utiliza una impresora de cera digital Xerox ColorQube 8580 (Xerox Corporation, Norwalk, CT, EE. UU.). Además, se usa una impresora 3D de Formlabs (Form-labs, Form 3, EE. UU.) para fabricar un dispositivo de plástico para poder manipular el código QR de detección. Los códigos QR son captados con tres dispositivos diferentes. Para el proceso de calibración se utiliza una cámara digital Sony Cyber-shot DSC-HX 300 (Sony, Tokio, Japón). Para mantener las condiciones de luminosidad estándar durante la adquisición de imágenes se utiliza una Cube Light Box [109] con la cámara colocada frente a una caja casera de madera blanca con dos lámparas LED de 6500 K colocada a 45° con respecto a la cámara digital, con el μ PAD en una posición fija dentro de la caja para aislarlo de la radiación externa. Los *smartphones* utilizados para decodificar los códigos QR y la información alojada son los siguientes: Pocophone F1 (Xiaomi Corporation, Hidian, China) y Huawei PSmart (Huawei Technologies Co. Ltd., Shenzhen). El dispositivo Pocophone F1 tiene una cámara incorporada de 12 + 5 megapíxeles con enfoque automático de doble píxel y sensor CCD; 6 GB de RAM y procesador Snapdragon 845 [98]. El dispositivo Huawei PSmart tiene una cámara incorporada de 13 + 2 megapíxeles con enfoque automático de doble píxel y sensor CCD; 3 GB de RAM y procesador Kirin 659 [110].

El *software* que se usa para desarrollar el trabajo se presenta a continuación. Todos los algoritmos implementados en los dispositivos teléfonos inteligentes se validan previamente mediante la ejecución de simulaciones y pruebas de conceptos en Matlab. Por otro lado, Android Studio ha sido el entorno de desarrollo utilizado para programar la aplicación de Android para el análisis colorimétrico de códigos QR. Para realizar la detección de códigos QR se usa una conocida biblioteca de código abierto llamada Zxing [88]. Esta biblioteca fue el código base sobre el que se trabajó para desarrollar la aplicación móvil para detectar códigos QR colorimétricos. Además, se ha usado Adobe Illustrator para diseñar los códigos QR colorimétricos, y un *software* de tratamiento de imagen ImageJ para extraer los datos de la recta de calibración.

4.3.3 *Fabricación y preparación del código QR biosensor*

En esta sección se describirá el proceso de fabricación y preparación del código QR biosensor. Se diseñó y fabricó un dispositivo de plástico para manipular el papel sensor durante los experimentos en una impresora 3D, como se muestra en la Figura 4.6a. Este dispositivo de manipulación 3D se compone

de tres partes, como se puede ver en la Figura 4.6b: (1) el código QR impreso en papel, que contiene los reactivos para la determinación de glucosa y lactato en los módulos seleccionados, (2) una almohadilla absorbente para muestreo, y (3) un panel transparente que se ensambla con la almohadilla absorbente y el código QR. El código QR, previamente diseñado, se imprime en papel de cromatografía Whatman No 1 mediante una impresora de cera Xerox Color-Qube 8580. Este diseño incluye tres capas, como se muestra en la Figura 4.6b: la capa de información estática con el código QR en blanco y negro, los módulos impresos de la curva de referencia y los módulos del biosensor. Los módulos del biosensor se preparan de la siguiente manera: usando una micropipeta, se prepararon 0.2 μL de TMB (20 mM) con 0.2 μL de HRP (156 U/mg) y 0.2 μL de GOx (160 U/mL) en 1 mg/mL de quitosano en solución salina tamponada con fosfato (PBS; pH. 7.4, 10 mM) que se añaden a los módulos biosensores del código QR para la detección de glucosa. Los módulos para la detección de lactato también se preparan según este procedimiento, pero utilizando LOx (150 U/mL) en lugar de GOx. Después de cada adición, el código QR se seca durante 5 minutos.

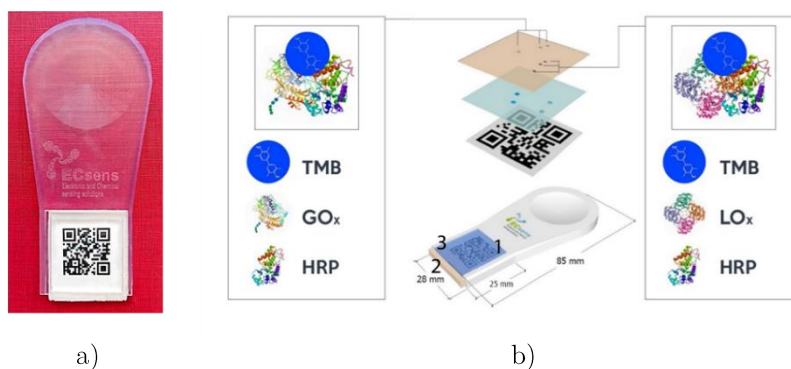


Figura 4.6. a) Accesorio de soporte del código QR, b) proceso de preparación de los módulos biosensores del código QR y elementos del accesorio de soporte.

4.3.4 Procedimiento de medida experimental

En un proceso típico para la determinación simultánea de glucosa y lactato, el dispositivo de manipulación se sumerge en soluciones que contienen glucosa y/o lactato. La muestra es recogida por la almohadilla absorbente y avanza hacia el código QR por acción capilar, donde se realiza el análisis colorimétrico. Una vez que la muestra llega al código QR, se mantiene a temperatura ambiente durante 15 minutos para completar la reacción colorimétrica. Una vez

completada esta reacción, aparece un color azul en los módulos de medición de glucosa y lactato. Para optimizar las condiciones del ensayo colorimétrico, el código QR se coloca en una caja de luz cúbica en una posición fija y luego se digitaliza utilizando uno de los teléfonos inteligentes descritos en la sección 4.3.2. Una vez el QR es analizado por la aplicación, ésta es capaz de estimar la concentración de glucosa y lactato simultáneamente in-situ.

4.3.5 Calibración de las curvas de caracterización de glucosa y lactato.

Para investigar la correlación entre la concentración de glucosa y lactato y el cambio de color registrado, se imprimieron círculos de papel de 10 mm de diámetro en papel de filtro de celulosa mediante una impresora de cera; las barreras de cera se generaron calentando el papel a 125 °C durante 5 minutos en un horno. A continuación, se prepararon 5 µL de TMB (20 Mm), 5 µL de HRP (156 U/mg) y 5 µL de GOx (160 U/mL) o LOx (150 U/mL) en 1 mg/mL de quitosano en tampón fosfato. Se dejó caer solución salina (PBS; pH 7.4, 10 mM) sobre los círculos de papel. El ensayo colorimétrico se llevó a cabo utilizando soluciones estándar de glucosa y lactato con niveles de concentración predeterminados (150-4000 µM) y se dejó reaccionar sobre papel durante 15 minutos. El sustrato de peroxidasa TMB utilizado para probar la oxidación catalítica en la detección de glucosa y lactato tiene el inconveniente de generar una distribución de color no homogénea en el papel de celulosa por lo que para solucionar este inconveniente se ha incluido una solución de quitosano. El quitosano se adhiere fuertemente al papel de celulosa mediante absorción electrostática y enlaces de hidrógeno, ralentizando el movimiento del TMB y aumentando la homogeneidad del color en el sustrato del papel, como se indica en trabajos anteriores [8,111,112].

Las curvas de caracterización de glucosa y lactato son tomadas con la cámara Sony Cyber-shot DSC-HX 300 y cada punto se promedia mediante tres ensayos. Se utilizó la coordenada S del espacio de color HSV como parámetro analítico porque se pudo observar que la saturación era directamente proporcional a la concentración de las muestras enriquecidas con ambos analitos (glucosa y lactato), como se muestra en la Figura 4.7. Para analizar cuantitativamente los resultados del ensayo, tomamos una fotografía del círculo de papel después de la reacción con estándares de glucosa o lactato y digitalizamos el cambio de color de las imágenes obtenidas usando el *software* ImageJ. Después de 15 minutos, el cambio de color se pudo distinguir claramente. En este ensayo, el rango lineal de respuesta estuvo entre 150 y 1000 µM para

glucosa y entre 150 y 1500 μM para lactato, con un coeficiente de regresión (R^2) de 0.9755 y 0.9584 respectivamente. Las ecuaciones estándar para las gráficas de calibración de glucosa y lactato se incluyen en la Figura 4.7. De acuerdo con estos resultados, el ensayo mostró una respuesta lineal en un amplio rango de concentraciones de glucosa y lactato.

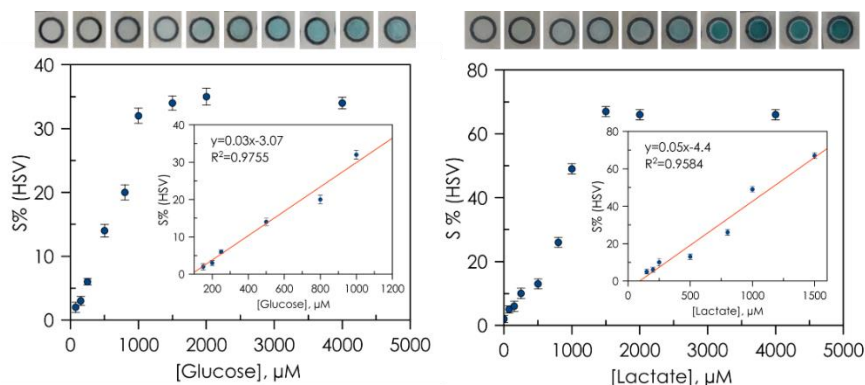


Figura 4.7. Curvas de caracterización de glucosa y lactato.

4.3.6 Análisis de muestras biológicas

Para establecer la idoneidad y eficiencia de la aplicación en entornos reales, se realizó la determinación simultánea de glucosa y lactato en dos fluidos biológicos complejos (suero y sudor artificial). Se analizó suero humano obtenido de voluntarios sanos anónimos. Por otro lado, las muestras de sudor artificial se preparan según la siguiente composición: 0.08 g/L MgCl_2 , 4.67 g/L NaCl , 0.45 g/L KCl , 0.27 g/L NH_4Cl , 0.13 g/L CaSO_4 , 0.22 g/L NaHCO_3 , 0.005 g/L NaH_2PO_4 , 0.60 g/L urea, 0.004 g/L ácido úrico, 0.07 g/L ácido pirúvico y 0.002 g/L ácido ascórbico. Después de la preparación, el pH se ajusta a 6.5 [113]. Las muestras se enriquecen con una solución estándar fija de glucosa y lactato. Después de eso, las muestras se filtran a través de un filtro de $0.22 \mu\text{m}$ y se diluyen cien veces con agua purificada. Finalmente, la detección de ambos analitos se realiza siguiendo el procedimiento de medición, y se calcula el contenido de glucosa y lactato en las muestras según la función de caracterización a partir de las soluciones estándar.

4.4 Implementación *software*

En esta sección, se abordan varias características del proceso de implementación del *software*. Los algoritmos desarrollados se modelaron en Matlab, como etapa de prueba de concepto. Luego, los algoritmos se implementaron

en Android Studio, personalizando una aplicación existente para la detección de códigos QR con el fin de agregar funcionalidades de medición y calibración de color. A continuación, se explica la arquitectura del *software* desarrollado y sus funcionalidades.

4.4.1 Pruebas de concepto en Matlab, algoritmos y métodos.

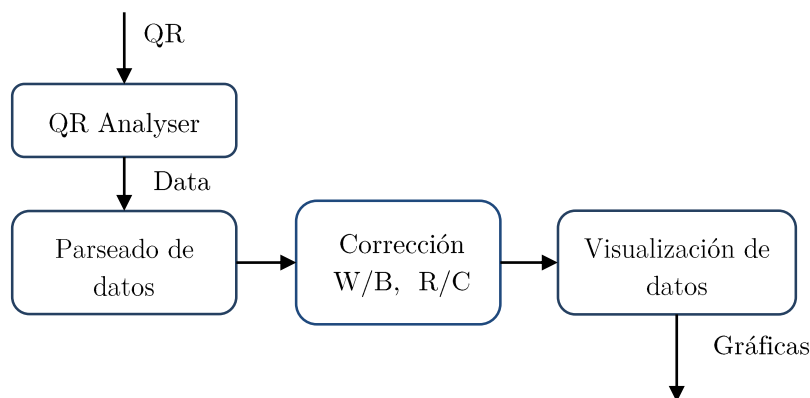


Figura 4.8. Flujo de trabajo de las pruebas de concepto realizadas con Matlab.

Matlab fue el *software* que se usó para realizar las pruebas de conceptos de los métodos W/B y R/C antes de ser implementados en Android. También se recurrió a Matlab para realizar técnicas de minería de datos y de visualización de estos. En la Figura 4.8 se muestra un esquema del flujo de trabajo seguido para realizar las pruebas de concepto con Matlab. Este procedimiento parte de la extracción de los datos de interés (curva de referencia, módulos biosensores y medición de referencias de blanco y negro) por parte de la aplicación QR Analyser, que es capaz de detectar el QR y extraer los valores RGB de aquellos módulos de interés del código QR colorimétrico. Como resultado de este proceso se obtiene un archivo de datos CSV con los datos de los módulos de interés en RGB, HSV y escala de grises. La posición de estos módulos es pregrabada en la aplicación para facilitar el proceso de extracción de datos. Posteriormente, estos datos son parseados en Matlab y se utilizan para realizar las pruebas de concepto de los métodos de corrección W/B y R/C. Una vez se ha realizado una primera implementación en Matlab de estos métodos y se ha comprobado su correcto funcionamiento, los algoritmos son implementados en Android basándose en el desarrollo realizado en Matlab, teniendo en cuenta las diferencias entre los lenguajes. Posteriormente se utilizó Matlab para realizar el parseado y la visualización de los datos de los

experimentos que se realizaron para validar el rendimiento de los algoritmos, como se verá en la sección 4.5

4.4.2 Implementación de la aplicación QR Analyser en Android

Como resultado de este trabajo se desarrolló una aplicación en Android que integrara los modelos de corrección propuestos y fuera capaz de medir la concentración de los analitos propuestos. Para el desarrollo de esta aplicación se plantearon los siguientes requisitos:

- **Fácil uso:** es necesario que la aplicación tuviera una interfaz sencilla y que pudiera ser usada tanto por personal autorizado como para el usuario final.
- **Accesible y rápida:** durante el planteamiento teórico de los modelos de corrección se procuró que los modelos desarrollados tuvieran un coste computacional bajo para que la potencia del *smartphone* no fuera impedimento para su uso.
- **Offline:** otro requisito inicial fue que la aplicación pudiera funcionar *offline*, para que se pudiera usar en zonas donde no hubiera acceso a Internet o la calidad de este fuera muy baja.
- **Precisa y fiable:** uno de los requisitos más importantes es que tanto los modelos de corrección como los procesos experimentales fueran fiables y dieran resultados precisos.

Además, se necesita que esta aplicación sea capaz de realizar las siguientes funcionalidades:

1. Detectar y obtener información de un código QR.
2. Ser capaz de detectar un QR in-situ o desde una imagen estática tomada previamente.
3. Acceder al valor RGB de los módulos de un código QR y extraerlo.
4. Almacenar los datos en el móvil y ser capaz de exportar esa información en un archivo CSV.
5. Almacenar varios registros de códigos QR de forma local.
6. Aplicar el método de corrección R/C y W/B.
7. Mostrar y visualizar algunos datos de interés en la aplicación para mejorar el flujo de trabajo.

Para evitar realizar todo este trabajo desde cero se utilizó una biblioteca de código abierto para la detección de códigos QR Zxing [88], explicada en la sección 2.6, con la que se pudo dar solución al primer requisito. Los siguientes requisitos fueron solucionados modificando partes de esta biblioteca o desarrollando implementaciones originales. Dentro de esta implementación se desarrolló la clase *RGBMatrix*, que gestiona todas las funcionalidades para generar una matriz de valores RGB de la dimensión del código QR analizado. Como la librería Zxing no cuenta con esta funcionalidad tuvieron que modificarse un conjunto de clases para que hicieran uso de la implementación desarrollada. Esta clase es fundamental para extraer los valores RGB de los módulos de interés que son guardados en la aplicación y que dependen de cada sensor diseñado. Esta información debería estar almacenada además en un servidor para que la aplicación pudiera actualizar su información periódicamente, según se fueran añadiendo nuevos tipos de sensores colorimétricos. Otra clase que tiene gran importancia es la llamada *ResultQR*, que es la encargada de guardar toda la información referida a un análisis de QR concreto y conectar la información con la base de datos de la aplicación y con la visualización de resultados. Los algoritmos de corrección propuestos se implementan dentro de esta clase donde se accede a los valores RGB de las referencias y los analitos a través de un objeto de los resultados arrojados por *RGBMatrix*

4.4.3 Aplicación final. Experiencia de usuario (UX)

La aplicación finalmente desarrollada debía cumplir con los requisitos expuestos en la sección 4.4.2. Esta aplicación para teléfonos inteligentes, llamada QR Analyser, es fácil de usar y no requiere ninguna formación previa para capturar las muestras. La Figura 4.9 muestra capturas de pantalla del diseño visual de la aplicación para analizar su experiencia de usuario. Primero, en la Figura 4.9a, se presenta una lista de registros de análisis realizados donde un usuario puede agregar un nuevo análisis u obtener información sobre cada uno. El usuario tiene dos opciones para agregar un nuevo registro QR, desde una imagen en la galería o tomando una fotografía in situ (Figura 4.9b). La Figura 4.9c muestra un ejemplo de una vista previa del análisis donde el usuario puede nombrar el registro y comprobar que la imagen es adecuada. Una vez detectado el código QR, se puede acceder al registro realizado donde se proporciona una vista previa del código QR colorimétrico y la concentración medida para muestras de glucosa y lactato, como se muestra en la Figura 4.9d. Durante el desarrollo de la aplicación se añadieron funcionalidades como la de visualizar la curva de referencia, obtener el módulo de pixel y otros datos de interés para el desarrollador.

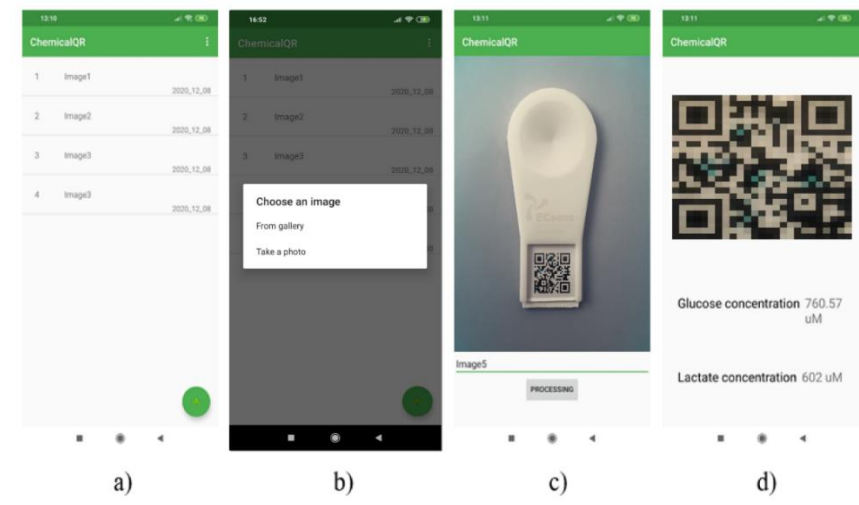


Figura 4.9. Capturas del diseño visual de la aplicación QR Analyser. a) lista de registros, b) captura del código QR, c) captura del código desde una foto, d) visualización de resultados.

4.4.4 Arquitectura de un sistema en la nube

Como parte de esta solución se propone un diseño de un sistema de computación en la nube que utiliza nuestra propuesta para proporcionar un conjunto de servicios *backend*, principalmente para rastrear y compartir mediciones de muestras, creando un repositorio. La Figura 4.10 muestra los diferentes componentes de *hardware* y *software* desarrollados en este estudio. Estos componentes se clasifican en tres capas principales: aplicación, plataforma e infraestructura. La capa de aplicación contiene componentes para el usuario final: una aplicación móvil, una aplicación web de gestión, códigos QR y un analizador de códigos QR diseñado a medida. La capa de plataforma contiene un conjunto de servicios de alto nivel que son utilizados directamente por los componentes de la capa de aplicación. Un servicio de notificación permite enviar mensajes a los usuarios finales cada vez que ocurre un evento importante en todo el sistema (disponibilidad de nuevas pruebas, noticias importantes, etc.). Un sistema de información química almacena datos estructurados y multimedia sobre las diferentes pruebas químicas que ofrece el sistema. Además, el servicio de identificación y autorización proporciona acceso de edición al sistema de información química. La capa de infraestructura está compuesta por un conjunto de servicios de bajo nivel para almacenar información y archivos estructurados o para acceder a los servicios internos ofrecidos por el proveedor de computación en la nube. Finalmente, se imponen

un conjunto de políticas de seguridad entre las diferentes capas: cifrado de datos y conexiones a Internet, copias de seguridad automatizadas, privacidad del usuario, etc.

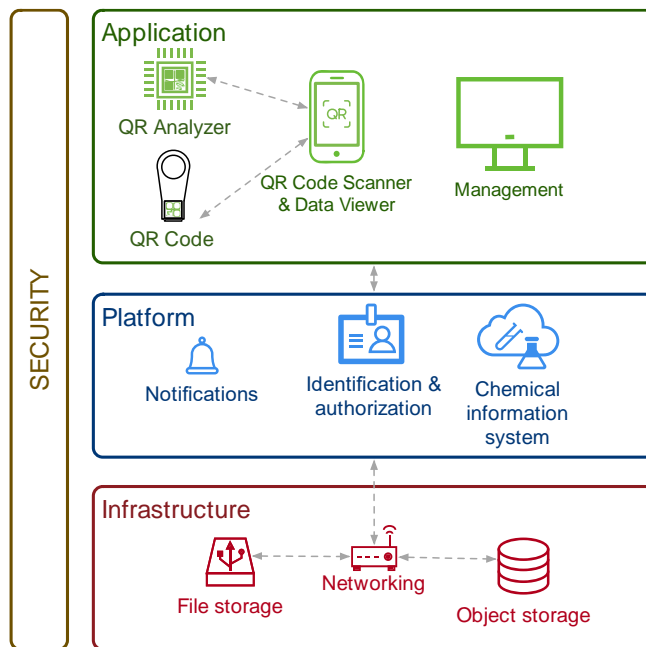


Figura 4.10. Diseño de una arquitectura de computación en la nube basado en nuestra aplicación QR Analyser.

4.5 Resultados y discusión

En esta sección se realizan un conjunto de pruebas y experimentos para evaluar los elementos del sistema propuesto. Primero se realiza la caracterización analítica del QR colorimétrico para los analitos de interés (glucosa y lactato). Posteriormente se medirá la curva de referencia impresa bajo distintas condiciones y se evaluará la capacidad de corrección del método W/B y R/C. Finalmente, se mostrarán los resultados del análisis sobre las muestras biológicas (sudor y suero) aplicados al QR.

4.5.1 Caracterización analítica del QR colorimétrico

En la sección 4.3.5 se demostró la relación lineal entre la saturación y la concentración de la glucosa y el lactato. En esta sección se pretende realizar

la caracterización analítica del QR colorimétrico, que consiste en establecer la relación entre la curva de referencia impresa y las curvas de calibración de glucosa y lactato de la sección 4.3.5. Para realizar esta caracterización se procederá a fotografiar el QR colorimétrico diseñado, con la curva de referencia impresa, depositando los analitos correspondientes en los módulos biosensores tal y como se especificó en la sección 4.2.2 para distintas concentraciones. Este experimento se realiza bajo las mismas condiciones estándar descritas en la sección 4.3.5. De esta forma la curva de referencia y la de caracterización están tomadas bajo las mismas condiciones de luminosidad dando la posibilidad de relacionar las referencias impresas con la concentración del analito. Este proceso es clave para que cuando se mida el QR en condiciones no estandarizadas pueda aplicarse el método de corrección R/C previamente explicado y puedan obtenerse los valores de concentración fuera de las condiciones estándar de la calibración.

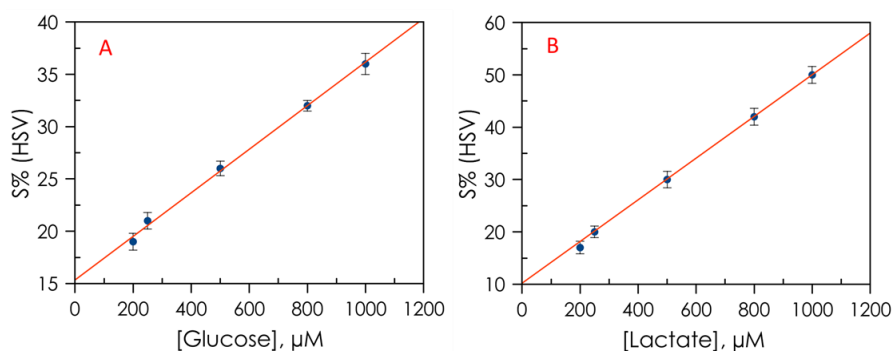


Figura 4.11. Calibración de las curvas de detección de glucosa y lactato dentro del código QR colorimétrico medidas en condiciones ambientales estándar. Datos promediados para tres ensayos.

La función analítica se obtiene mediante un set de calibración compuesto por 5 estándares de glucosa y lactato. Los rangos de concentración de glucosa y lactato son de 200 a 1000 μM, utilizando 15 minutos como tiempo de reacción. La curva de calibración resultante para glucosa y lactato se presenta en la Figura 4.11. La dependencia entre el valor de S y la concentración de glucosa o lactato obtenida presentó un coeficiente de correlación de $R^2=0,9529$ para glucosa y $R^2=0,9051$ para lactato, indicando una amplia respuesta lineal para ambos compuestos en cada caso. Utilizando un ajuste lineal, la ecuación de regresión para la glucosa y el lactato es $y = 0.05x + 15.63$ e $y = 0.05x + 11.37$, respectivamente. El límite de detección (LOD) se define mediante la ecuación $LOD = (3\sigma/s)$ donde σ es la desviación estándar de las señales en

blanco y s es la pendiente de la curva de calibración. En base a esta ecuación, se calcula que el LOD para la glucosa es $65.21 \mu\text{M}$ y $43.16 \mu\text{M}$ para el lactato. También se estudió la reproducibilidad del sistema de detección propuesto para glucosa y lactato. La desviación estándar relativa (RSD) es 1.20% y 0.80% para cada compuesto a $200 \mu\text{M}$ para 6 mediciones repetidas, respectivamente. En la Tabla 4.1 se muestra un resumen de las características analíticas. Estas curvas analíticas se aplican en la última etapa de la arquitectura del sistema (Figura 4.3). Por lo tanto, una vez que a los módulos biosensores (analitos) se le aplican las correcciones W/B y R/C, el valor de saturación obtenido se ingresa la curva analítica correspondiente, (glucosa o lactato) y se obtiene la concentración real.

Parámetros analíticos	Glucosa	Lactato
Rango de medida (μM)	197.60 – 1000.00	130.78 – 1000.00
Pendiente (a)	0.05	0.05
Ordenada en el origen (b)	15.63	11.37
LOD (μM)	65.21	43.16
Precisión (%) 200 μM	1.20	0.80
Tiempo de vida (días)	30	30

Tabla 4.1. Parámetros de la caracterización analítica del QR colorimétrico para glucosa y lactato en condiciones estándar.

4.5.2 Estudio de la curva de referencia para diferentes *smartphones* y condiciones de iluminación

La curva de referencia impresa es uno de los elementos primordiales para la corrección de los valores de saturación de los analitos. Por este motivo es necesario realizar un estudio en profundidad sobre la medición de esta curva bajo distintas condiciones de luminosidad y distintos dispositivos. La Figura 4.12a presenta un promedio de 24 curvas de referencia (saturación medida frente a saturación impresa) medidas a partir de códigos QR colorimétricos

con los dos teléfonos inteligentes diferentes descritos en la sección 4.3.2. El tamaño del código QR es de 22 mm y la distancia de medición es de 8 cm con un ángulo de 90° respecto a la posición del QR. Para estos experimentos, los parámetros de la cámara son controlados automáticamente por los *smartphones* y se utiliza una lámpara LED (3000 K) como fuente de luz que simula las condiciones de luminosidad estándar. Se puede notar que el comportamiento de la curva de referencia es lineal en ambos dispositivos y se aprecia un ligero aumento en la pendiente de la curva tomada con el Huawei PSmart. La desviación estándar de saturación para ambos dispositivos es bastante similar, 5.6 para el dispositivo Pocophone F1 y 6.6 para el dispositivo Huawei PSmart.

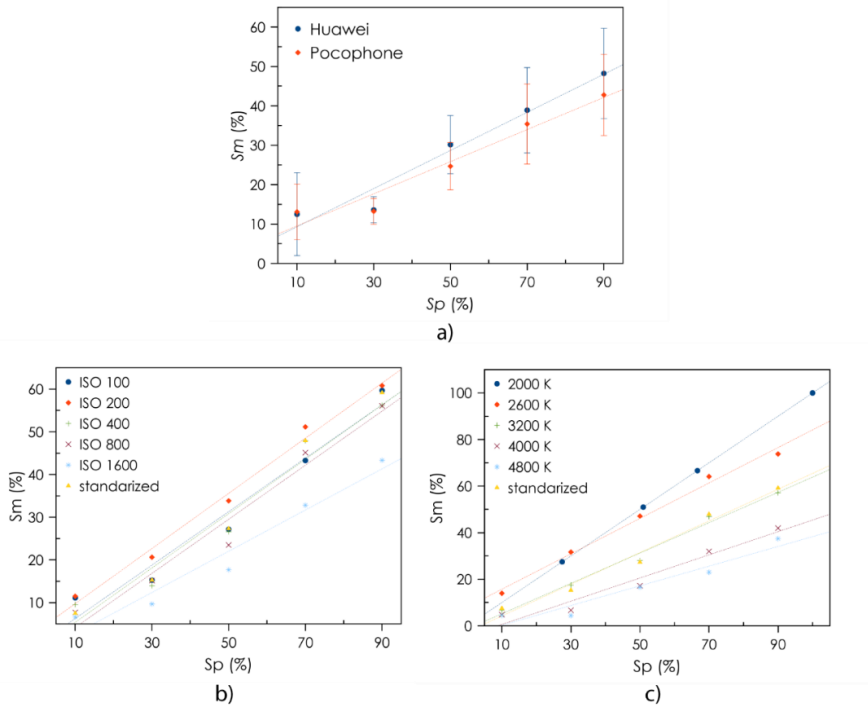


Figura 4.12. Curva de referencia impresa en el QR, a) promedio de las curvas de referencia con distintos dispositivos, b) curva de referencia medida bajo diferentes ISOs para el dispositivo Pocophone F1, c) curva de referencia medida con diferentes balances de blancos para el dispositivo Pocophone F1.

Las Figura 4.12b y Figura 4.12c presentan un estudio sobre los cambios que sufre la curva de referencia de la corrección W/B en diferentes condiciones de iluminación simuladas con los parámetros de la cámara del dispositivo Pocophone F1. En este caso se mantiene un balance de blancos automático y

un tiempo de exposición 1/125s. Cuando cambia la ISO hay un desplazamiento de la curva aumentando la ordenada en el origen conforme aumenta la ISO. Como muestra la Figura 4.12b, la corrección W/B puede corregir considerablemente las variaciones de la curva cuando la ISO no es muy alta o muy baja, casos de saturación de la imagen o de baja sensibilidad. Posteriormente se modificó el balance de blancos incrementando la temperatura del color con una ISO 400 y 1/125s de tiempo de exposición. En este caso la corrección W/B no es capaz de corregir estas desviaciones correctamente, como muestra la Figura 4.12c. Es en este caso cuando el uso de la corrección R/C es especialmente útil.

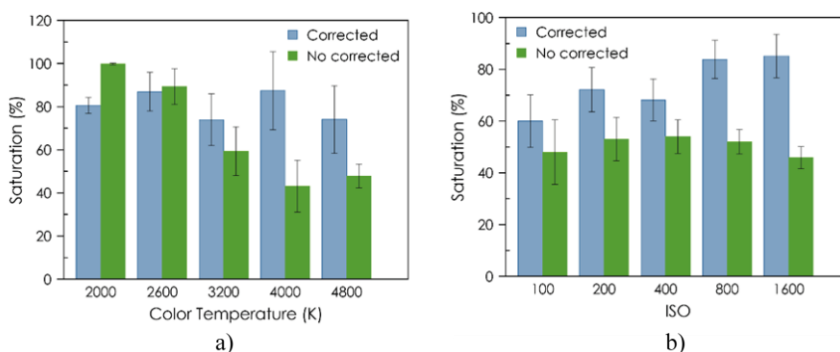


Figura 4.13. a) Efecto de la corrección R/C y W/B para cambios de balance de blancos, b) efecto de la corrección R/C y W/B para cambios de ISO.

A continuación, se va a comprobar el efecto de la corrección W/B y R/C sobre un conjunto de módulos de lactato a 800 μM . Tras ambas correcciones, este módulo de lactato debería presentar una saturación entorno al 80%, que es la medida en condiciones estándar después de aplicar la corrección W/B. La Figura 4.13a muestra la saturación medida a diferentes temperaturas de color, antes y después de ambos métodos de corrección (W/B y R/C). Las fotos han sido tomadas con un Pocophone F1 con ISO 400 y 1/125 s de tiempo de exposición. Se observa una disminución en la saturación a medida que aumenta la temperatura del color, como muestra la Figura 4.13a. Sin embargo, esta tendencia se corrige completamente después del método de corrección RC manteniéndose en torno al 80% de saturación, como se muestra en la Figura 4.13a. Como resultado, el error aumenta ligeramente después de este proceso. En la Figura 4.13b se realiza un estudio similar de la variación de saturación para diferentes ISO. Las fotos fueron tomadas con un balance de blancos automático y un tiempo de exposición de 1/125 s. En este caso, no existen variaciones significativas entre diferentes condiciones ISO. Sin embargo,

cuando se aplica la corrección R/C, la saturación tiende a aumentar hasta un 80%, que es el valor esperado de estos módulos medidos. Así, la combinación de ambos métodos mejora el rendimiento de la conocida corrección W/B para cambios de iluminación complejos basados en la modificación de las temperaturas de color. Además, los algoritmos de corrección reportados tienen baja complejidad. Como resultado de esto, estos algoritmos se pueden implementar fácilmente en otros dispositivos y aplicaciones de códigos QR basadas en diferentes lenguajes de programación.

4.5.3 Estudio del ángulo de medida y el tamaño del QR

A continuación, se realizará un estudio de optimización de las condiciones de medida basándonos en la medición de la curva de referencia (saturación medida, S_m , frente a saturación impresa, S_p) usando el dispositivo Huawei PSmart. Para realizar este experimento se analiza la curva de referencia impresa en el código QR colorimétrico para diferentes distancias de medición, tamaños de códigos QR y ángulos de medición. Además, se analiza el área de píxeles de cada módulo del código QR, para establecer un valor mínimo del área de píxel que debe tener cada módulo. Las imágenes han sido tomadas con un tiempo de exposición de 1/60 y una ISO 400.

La Figura 4.14a presenta un estudio de la relación entre el área del módulo y la distancia de medición para varios tamaños de códigos QR. Se observa una disminución en el área del módulo a medida que aumenta la distancia. La distancia mínima de medición se alcanza cuando el código QR ocupa todo el rectángulo de medición en la pantalla del teléfono inteligente. La distancia mínima suele ser la mejor distancia para medir el código QR, pero cuando la distancia de medición es menor que la distancia focal del teléfono inteligente, el área del módulo disminuye drásticamente. Esto ocurre cuando el tamaño del QR es inferior a aproximadamente 13 mm. Además, para este tamaño de código QR, éste no se puede detectar a distancias superiores a 6 cm.

La Figura 4.14b muestra la curva de referencia para las diferentes distancias de medición con un tamaño QR de 22 mm. Se observa una disminución en la pendiente de la curva de referencia a medida que aumenta la distancia de medición. La Figura 4.14c presenta la curva de referencia para diferentes áreas del módulo. Se observa un comportamiento similar en la curva de referencia para áreas de módulo superiores a 17 píxeles, aunque la pendiente de la curva de referencia disminuye significativamente por debajo de este valor. La Figura 4.14d presenta la curva de referencia para tres ángulos de medición

diferentes a la distancia mínima. No se notan variaciones en la curva de referencia a medida que cambia el ángulo, pero cuando el ángulo es mayor a 40° , el código QR no se detecta a la distancia mínima. Así, las condiciones óptimas de medición del sistema se presentan cuando el área del módulo en pixeles es superior a 17 pixeles. Esto puede lograrse mediante una compensación entre la distancia de medición y el tamaño del QR. Como recomendación se puede establecer que la mejor distancia sería la mínima posible (la que permite identificar todo el QR en la pantalla del *smartphone*) en la que se pueda enfocar correctamente. Este valor va a depender de la cámara del teléfono, ya que hay algunas que son capaces de enfocar a una distancia más corta. También se puede determinar un arco de medición óptimo entre -40° y 40° sobre la vertical del QR ya que fuera de esa distancia no es detectado.

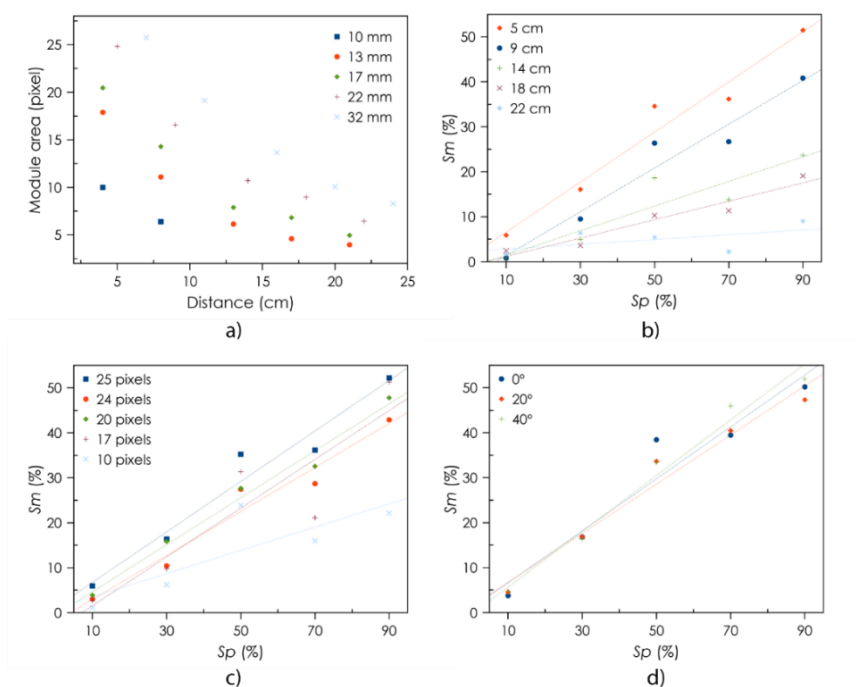


Figura 4.14. Curvas de referencia, a) área de los módulos frente a la distancia para diferentes tamaños de QR, b) curva de referencia variando la distancia de medición, c) curva de referencia en función del área del módulo, d) curva de referencia bajo diferentes ángulos de medición.

4.5.4 Aplicación con muestras biológicas

A continuación, se realiza un análisis del rendimiento de la aplicación desarrollada usando muestras biológicas de sudor y suero de forma simultánea. Para ello estas muestras se enriquecen con soluciones estándar de glucosa o

lactato de 200, 500 y 800 μM . Los experimentos se realizan por triplicado para cada concentración de glucosa y lactato usando los dos teléfonos inteligentes (Huawei PSmart y Pocophone F1). En este ensayo se comprueba si la concentración medida por la aplicación móvil desarrollada es similar a la estimada por otros métodos comerciales. Para ello se busca tener porcentajes de recuperación en torno al 100 %, ya que esto indicaría que la concentración del analito medida es similar a la suma de la concentración base y añadida. Los resultados se muestran en la Tabla 4.2 y la Tabla 4.3, para el Huawei PSmart y el Pocophone F1, respectivamente. Se puede observar que las recuperaciones obtenidas están en el rango del 100-117% para la glucosa y del 91-108% para el lactato. También se aprecia que no existen grandes diferencias entre los resultados obtenidos para ambos *smartphones*. Esto muestra que el sistema es capaz de funcionar en distintos dispositivos. Además, para probar las capacidades del ensayo de código QR desarrollado, se estimó la concentración de glucosa de estas muestras de suero y sudor mediante un glucómetro de sangre comercial. Asimismo, la concentración de lactato se midió utilizando un kit de ensayo de lactato comercial. Los resultados confirman que los valores de concentración obtenidos utilizando el método propuesto basando en el código QR mostraron una buena concordancia con los valores de los ensayos comerciales. Por tanto, nuestro método puede ser una alternativa viable para detectar glucosa y lactato en muestras reales sin interferencia del entorno biológico.

Samples	Standard method	Proposed method	Spiked	Total Found	Recovery %
Glucose content, μM					
Serum 1	11 ± 2.1	0.00	200	210.2	105.1
Serum 2	14 ± 0.9	0.00	500	516.6	103.3
Serum 3	11 ± 4.2	0.00	800	806.3	100.8
Sweat 1	134 ± 4.8	139.2	200	356.3	105.0
Sweat 2	138 ± 2.5	139.2	500	729.0	114.0
Sweat 3	138 ± 3.2	139.2	800	1037.7	110.5
Lactate content, μM					
Serum 1	235 ± 3.8	239	200	405.5	92.4
Serum 2	240 ± 1.5	239	500	742.4	100.5
Serum 3	241 ± 2.2	239	800	1075.8	103.6

Sweat 1	134 ± 2.1	135	200	321.0	95.8
Sweat 2	135 ± 1.2	135	500	602.0	94.8
Sweat 3	133 ± 4.0	135	800	859.4	91.9

Tabla 4.2. Determinación de glucosa y lactato en muestras biológicas usando el dispositivo Huawei PSmart.

Samples	Standard method	Proposed method	Spiked	Total Found	Recovery %
Glucose content, μM					
Serum 1	5 ± 1.1	0.00	200	235.0	117.5
Serum 2	9 ± 2.5	0.00	500	502.3	100.5
Serum 3	11 ± 2.1	0.00	800	816.3	102.0
Sweat 1	135 ± 3.8	139.2	200	423.7	105.6
Sweat 2	135 ± 4.1	139.2	500	783.2	111.7
Sweat 3	139 ± 1.1	139.2	800	1118.9	111.7
Lactate content, μM					
Serum 1	239 ± 1.2	239	200	405.5	92.4
Serum 2	238 ± 1.8	239	500	742.4	100.5
Serum 3	242 ± 1.0	239	800	1075.8	103.6
Sweat 1	136 ± 4.1	135	200	321.0	95.8
Sweat 2	131 ± 2.0	135	500	602.0	94.8
Sweat 3	135 ± 1.0	135	800	859.4	91.9

Tabla 4.3. Determinación de glucosa y lactato en muestras biológicas usando el dispositivo Pocophone F1.

4.5.5 Estudio de estabilidad y selectividad

Las muestras biológicas contienen diferentes sustancias como ácido úrico, NaCl, CaCl₂, MgCl₂, fructosa, cisteína, glucosa y lactato que podrían inducir interferencias en la determinación de glucosa y lactato. El estudio de interferencia evalúa los efectos de estas sustancias en la detección de glucosa y lactato. La Figura 4.15 muestra los resultados de selectividad medidos y procesados con el dispositivo Huawei PSmart. Las soluciones de las muestras que

contienen las interferencias se miden utilizando catorce códigos QR diferentes. Las concentraciones de las siete interferencias probadas son de 800 μM . Además, se estudió la selectividad del código QR para 800 μM de glucosa y lactato en presencia de 2000 μM de interferencias como se muestra en las Figura 4.15c y Figura 4.15d, respectivamente.

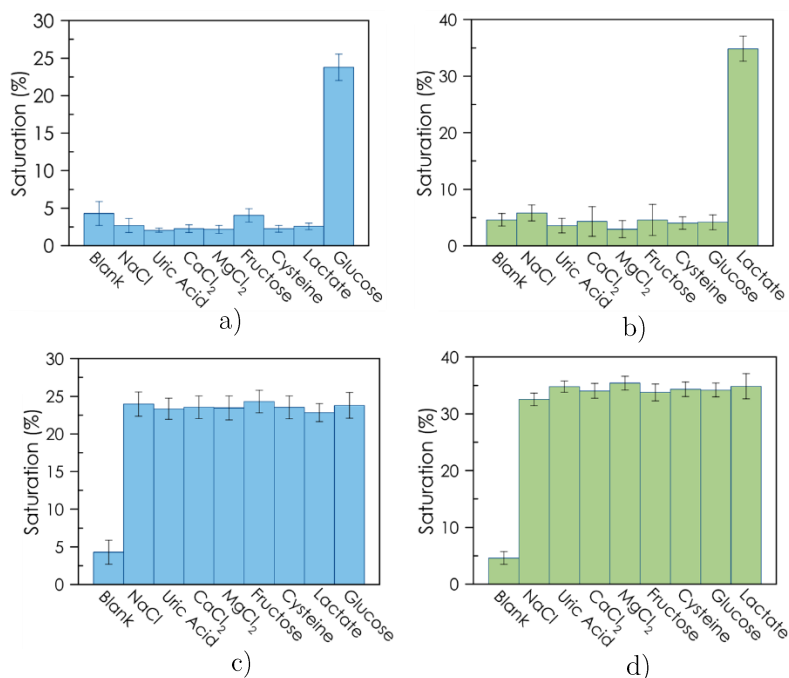


Figura 4.15. Respuesta del código QR para: a) glucosa a 800 μM , b) lactato a 800 μM , c) 800 μM glucosa en presencia de 2000 μM de interferentes, d) 800 μM lactato en presencia de 2000 μM de interferentes.

Los resultados demuestran que el valor de saturación no se vio afectado por la presencia de estas sustancias. En la Figura 4.15a se puede ver que el valor de saturación solo aumentó para la glucosa del mismo modo ocurre para el lactato, como se observa en la Figura 4.15b. Por lo tanto, se puede decir que el código QR desarrollado es un candidato prometedor para la detección selectiva de glucosa y lactato, ya que GOx y LOx son específicos de la glucosa y el lactato únicamente, no de otras sustancias. Por tanto, el método desarrollado basado en códigos QR presenta una alta selectividad y pueden aplicarse para la detección multiplexada de glucosa y lactato en mezclas. La buena selectividad se debe principalmente a la alta especificidad de las reacciones colorimétricas enzimáticas.

Finalmente, se realizó un estudio de la estabilidad de almacenamiento del código QR desarrollado para detectar glucosa y lactato durante trece días consecutivos. La glucosa y el lactato fueron testados a una concentración de 1000 μM . Los códigos QR se almacenan en condiciones oscuras, secas y selladas a 4°C y se prueban cada semana, durante 30 días. En la Figura 4.16 se muestra el estudio de estabilidad en la que todos los valores representados son la media y la desviación estándar de tres experimentos independientes. No se observan cambios significativos en las respuestas colorimétricas para ambos analitos durante al menos 30 días. Por lo tanto, la estabilidad de almacenamiento del código QR es aceptable y está disponible para diagnóstico clínico y pruebas en el lugar de atención.

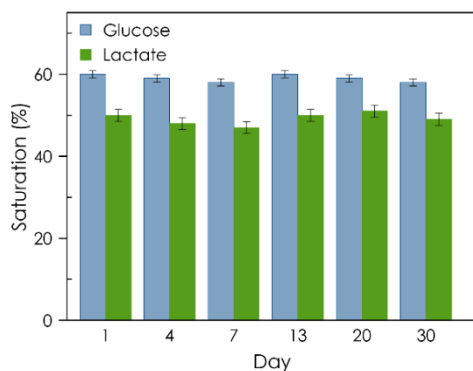


Figura 4.16. Test de estabilidad para el código QR desarrollado almacenado a 4°C en la oscuridad durante 30 días.

4.6 Conclusiones y contribuciones

El sistema propuesto en este trabajo presenta una solución para analizar POC basada en *smartphones* capaz de realizar análisis colorimétricos de biosensores integrados en un código QR. Este sistema propone un nuevo diseño de sensor colorimétrico basado en códigos QR para la determinación de glucosa y lactato. Además, se presenta un sistema de lectura que extrae los valores RGB de los módulos del código QR permitiendo un mayor grado de libertad para integrar los sensores en él sin perder la información estática almacenada en el código. Esta información estática pudiera estar asociada a un identificador del paciente de forma que la muestra identificada estuviera contenida en el código aumentando la trazabilidad de las muestras y dando la posibilidad de conectar toda esta información mediante una arquitectura en la nube. La aplicación móvil desarrollada también es capaz de ejecutarse

sin conexión a red lo que da la posibilidad a analizar y almacenar la información de los códigos QR in-situ de forma *offline*.

El sistema se ha testeado para la determinación de glucosa y lactato en muestras reales de sudor y suero en un mismo código QR lo que demuestra su capacidad de análisis multianalito. Unido a esta aplicación se ha diseñado, implementado y testeado un algoritmo de corrección de luminosidad en distintas condiciones de luminosidad. Este algoritmo está basado en la combinación de referencias blancas y negras junto con un método de corrección geométrica de una recta de referencia que, hasta donde sabemos, no se había presentado en la literatura con anterioridad. Se espera que en un futuro este sistema pueda combinarse con la implementación de esta arquitectura en la nube y estandarizar distintos sensores colorimétricos basados en este QR para que pueda testearse en el análisis de otros analitos. Una línea de trabajo futura relevante sería cambiar el método de corrección por curva de referencia por uno que cuente con referencias genéricas, como una carta de color, para poder aumentar la capacidad multianalito del sistema. Como resultado de este trabajo se ha contribuido con los siguientes trabajos científicos:

- Publicación en revista científica: Antonio Marín-Sánchez, Inmaculada Ortiz-Gómez, David Gallego Méndez, Carlos Rodríguez-Domínguez, Luís Fermín Capitán-Vallvey, Diego P. Morales, José Antonio Álvarez-Bermejo, Encarnación Castillo, Alfonso Salinas-Castillo. Smart Quick Response code for multianalyte determination. *Measurements*, 204 (2022). <https://doi.org/10.1016/j.measurement.2022.112035>
- Presentación de poster: Inmaculada Ortiz-Gómez, Antonio Marín-Sánchez, David Gallego Méndez, Carlos Rodríguez-Domínguez, Luís Fermín Capitán-Vallvey, Diego P. Morales, José Antonio Álvarez-Bermejo, Encarnación Castillo, Alfonso Salinas-Castillo. Colorimetric Smart Quick Response code for glucose and lactate determination. 21st International Symposium on Bioluminescence and Chemiluminescence and XIX International Symposium on Luminescence Spectrometry (2022).

- Comunicación oral: Antonio Marín-Sánchez, Inmaculada Ortiz-Gómez, David Gallego Méndez, Carlos Rodríguez-Domínguez, Luis Fermín Capitán-Vallvey, Diego P. Morales, José Antonio Álvarez-Bermejo, Encarnación Castillo, Alfonso Salinas-Castillo. Colorimetric Quick Response code biosensor. 21st International Symposium on Bioluminescence and Chemiluminescence and XIX International Symposium on Luminescence Spectrometry (2021).
- Comunicación oral: Antonio Marín-Sánchez, Inmaculada Ortiz-Gómez, David Gallego Méndez, Carlos Rodríguez-Domínguez, Luis Fermín Capitán-Vallvey, Diego P. Morales, José Antonio Álvarez-Bermejo, Encarnación Castillo, Alfonso Salinas-Castillo. Integration colorimetric biosensor in Quick Response code. I Congreso Virtual de la Red Nacional de Sensores y bioSensores Ópticos (2021)

Capítulo 5

Determinación de señales de fluorescencia ratiométricas en μ PAD

En este capítulo se va a presentar una aplicación móvil capaz de detectar el nivel de intensidad fluorimétrico de un μ PAD para la detección de biotioles propuesta en [114]. Para ello, primero se describirá el diseño del μ PAD y la conceptualización del algoritmo de procesamiento de imagen empleado junto con las características que presenta la implementación de la aplicación en Android. Posteriormente, se describirá la metodología experimental empleada, destacando el proceso de fabricación y preparación del μ PAD. Finalmente, se testeará y optimizará el sistema con imágenes tomadas a distinta ISO y se presentarán las conclusiones del trabajo.

5.1 Introducción

La medición de sensores fluorimétricos requiere en ocasiones condiciones de baja luminosidad que necesitan equipos especializados para realizar la medición de los sensores. Es por esto que, en los últimos años, los sistemas de medición de intensidad fluorimétrica de sensores químicos basados en *smartphones* han sido muy estudiados, abriendo así la posibilidad de evitar costosos equipos para realizar este tipo de análisis. Con ese propósito se desarrolló un algoritmo de procesamiento de imagen para la detección de los sensores del dispositivo analítico de forma automática. Este algoritmo se implementó en un dispositivo móvil de forma que pudiera usarse de forma portable. Además, se implementó una funcionalidad adicional para que el sistema fuera capaz de determinar si el μ PAD empleado requería de normalización ratiométrica y aplicarla en ese caso. Como en este caso no se usan códigos QR como base del sensor, será necesaria la implementación de un algoritmo de procesamiento de imagen desde cero para la detección del μ PAD empleado. Esta implementación es una primera aproximación del uso de *smartphones* para detección de señales fluorimétricas mediante dispositivos móviles.

5.2 Conceptualización e implementación del sistema

En esta sección se presentará brevemente el esquema general del sistema y su implementación software. Se abordarán las distintas partes del algoritmo

de procesamiento para la detección del sensor y del proceso de normalización ratiométrica empleado. Además, se presentará el diseño de la aplicación móvil implementada y se presentarán capturas de esta para mostrar sus funcionalidades.

5.2.1 Esquema general del sistema

El sistema consta de un dispositivo móvil en el que se implementa una aplicación para la detección del nivel de fluorescencia del μ PAD presentado en [114]. Con la cámara del móvil se toma una imagen del μ PAD que el sistema usa para identificar las regiones de la imagen que ocupan cada sensor y determinar la intensidad de color promedio de cada uno. En la Figura 5.1 se puede ver un ejemplo del esquema general del algoritmo de procesamiento propuesto. Para empezar, se realiza una conversión de la imagen a nivel de gris para realizar las operaciones sobre un solo canal. Posteriormente se realiza un proceso de umbralización por histograma con el que se obtiene una imagen binaria donde los círculos del μ PAD son detectados, convertidos a 1, y el fondo de la imagen a 0. Este proceso de umbralización puede introducir falsos positivos derivados de pequeños artefactos que aparezcan en la imagen. Otra fuente de error que se puede encontrar se debe a los bordes circulares de la estructura del μ PAD que tienen un valor de intensidad suficiente para ser detectados, pero generan error en el valor promedio de los píxeles del sensor. Para paliar estos errores se realiza un procedimiento morfológico de erosión sobre la imagen usando un elemento circular. Este proceso eliminará los artefactos suficientemente pequeños y reducirá el diámetro de los círculos detectados para evitar el error de los bordes. La siguiente etapa es la aplicación de un algoritmo de componentes conectadas que clasifica las componentes de píxeles que tengan conexión entre sí. Tras este procedimiento es posible que algún artefacto de mayor tamaño, pero no como los círculos del sensor, sea detectado. Este error es evitable mediante un proceso de discriminación de componentes por área. A través del cálculo del área relativa de cada componente se discriminan aquellas que sean menores a un umbral optimizado previamente en función del área promedio de los sensores detectados. Este proceso es posible ya que la imagen se posiciona a una distancia tal que los sensores queden enmarcados en un recuadro prefijado y que el área se calcula en relación a estas dimensiones para que no dependa de la resolución de la cámara utilizada. En la siguiente etapa se calcula el valor promedio de la intensidad de los píxeles de cada componente.

Finalmente, el algoritmo tiene la posibilidad de diferenciar si el μ PAD empleado cuenta con normalización ratiométrica o no, proveyendo así resultados adecuados en cada caso. El proceso de normalización usa el valor de intensidad de la fila superior del μ PAD como base para obtener un valor independiente de la ISO empleada en la cámara. Este algoritmo aplica la siguiente ecuación:

$$I_{norm} = \frac{I_{rat} - I_{sensor}}{255} \quad (38)$$

donde I_{norm} es la intensidad de fluorescencia normalizada, I_{rat} e I_{sensor} son las intensidades del ratiométrico y del sensor, respectivamente. Dependiendo de la sustancia analizada en la aplicación final se podría precargar una curva de caracterización para determinar la concentración de la sustancia.

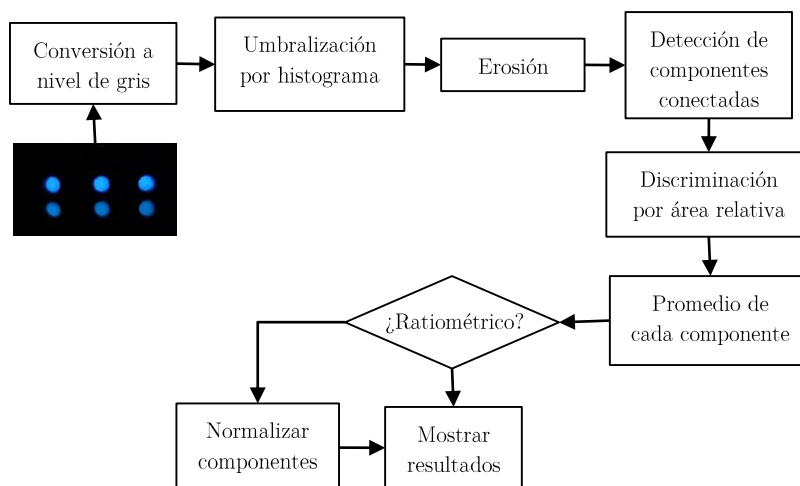


Figura 5.1. Esquema general del sistema para la determinación del nivel de fluorescencia de μ PAD fluorimétrico diseñado.

5.2.2 Implementación de la aplicación final

Los algoritmos presentados en la sección 5.2.1 han sido implementados apoyándose en la librería OpenCV para facilitar algunas funcionalidades de procesamiento de imágenes. La aplicación desarrollada busca, en primera instancia, ser de fácil uso y que pueda guardar los datos de forma *offline*. En esta sección se va a describir brevemente la arquitectura de esta aplicación y su flujo de

trabajo para un usuario final. La aplicación móvil desarrollada está estructurada en 3 paquetes en los que se reparten todas las funcionalidades del sistema basándose en un patrón MVC. En esta sección se van a describir brevemente las tareas que desenvuelven cada paquete y se profundizará en las clases que implementan los algoritmos presentados en la sección 5.2.1. En primer lugar, se tiene el paquete *activities*, que contiene todas las clases y funcionalidades que se encargan de ejecutar las vistas programadas en archivos .xml y conectarlas con los datos procesados. El paquete *model* es el encargado de almacenar los datos en el dispositivo y realizar todo el procesamiento de imagen requerido. Este paquete contiene la clase *ImageAnalyzed* que almacena todas las funciones relacionadas con el procesamiento de la imagen como la umbralización, erosión, extracción de componentes o comprobar si se requiere normalización o no. Esta aplicación actúa como puente entre la aplicación y los algoritmos de procesamiento de imagen del paquete OpenCV que facilita parte del trabajo desarrollado. Por otro lado, la clase *ResultImage* es la encargada de realizar el procesamiento completo de la imagen llamando a las funciones de *ImageAnalyzed* y de almacenar estos datos en un archivo *JSON* mediante la conveniente serialización de estos.

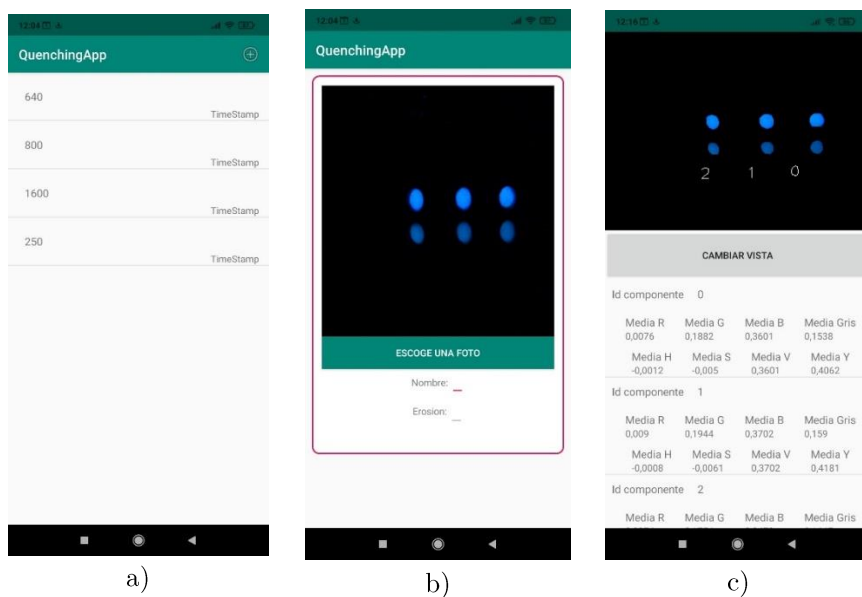


Figura 5.2. Capturas de pantalla de la aplicación QuenchingApp, a) lista de registro de análisis realizados, b) elección de la imagen, nombre y nivel de erosión, c) resultados obtenidos tras la detección de los sensores de la imagen.

En la Figura 5.2 se pueden ver capturas de pantalla de la aplicación desarrollada, llamada QuenchingApp. Cuando se ejecuta la aplicación ésta muestra una lista de registros en la que el usuario puede almacenar varios análisis, Figura 5.2a. Esta ventana le permite además borrar los registros que ya no necesiten. Si se quiere realizar un nuevo análisis se abre una ventana en la que el usuario puede tomar una foto in-situ o usar una de la galería tomada previamente para ser analizada, Figura 5.2b. Posteriormente, se puede ajustar el tamaño de pixeles del elemento usado para la erosión, cuanto mayor sea éste, mayor será su efecto. Una vez se ha analizado la imagen se puede ver una ventana de resultados en la que se muestran los valores medios de cada componente detectada en distintos espacios de color (RGB, Gris, y HSV), Figura 5.2c. Como funcionalidad adicional, se muestra un resultado de la imagen usada antes y después del procesamiento para comprobar si la erosión aplicada fue demasiado dura.

5.3 Metodología experimental

En esta sección se va a describir la metodología experimental empleada en este capítulo. Se describirán los materiales y reactivos usados en los experimentos y el procedimiento de fabricación y preparación del dispositivo analítico, μ PAD. Posteriormente se presentará el *hardware* y el *software* empleado para la realización de los ensayos y la implementación de la aplicación móvil desarrollada.

5.3.1 Reactivos y materiales

Los reactivos utilizados fueron los siguientes. L-Glutatión (GSH), L-cisteína (Cys), DL-homocisteína (Hcy), L-serina (Ser), L-histidina (His), L-galactosa (Gal), L-lisina (Lys), L -arginina (Arg), L-asparagina (Asn), D-manosa (Man), etanol, etanolamina, trietilamina (TEA), diclorometano (DCM), hidrogenofosfato de sodio, dihidrogenofosfato de sodio, hidrogenocarbonato de sodio, carbonato de sodio anhidro. ácido cítrico (CA), ácido yodoacético (IAA), divinilsulfona (DVS), polietilenimina ramificada de 25 kDa (25 kPEI); N, N-diciclohexilcarbodiimida (DCC) y N, N-diisopropiletetilamina (DIPEA) se adquirieron de Sigma-Aldrich [115]. Todos los reactivos eran de grado reactivo analítico a menos que se indique lo contrario. Para fabricar el μ PAD se utilizó papel de cromatografía Whatman grado 1 de Sigma-Aldrich. Todas las soluciones acuosas se prepararon utilizando agua de calidad del tipo de ósmosis inversa (estación Milli-RO 12 plus Milli-Q de Millipore, conductividad 18.2 M Ω cm).

5.3.2 *Hardware y software*

La intensidad de fluorescencia del dispositivo analítico fue medida con un dispositivo Pocophone F1 (Xiaomi Corporation, Hidian, China), que tiene una cámara incorporada de $12 + 5$ megapíxeles con enfoque automático de doble píxel y sensor CCD; 6 GB de RAM y procesador Snapdragon 845 [98]. Las imágenes fueron tomadas tras iluminar el sensor con un UV transilluminator 220V (MUV 21 series) [116]. Los dispositivos analíticos fueron fabricados en papel usando la técnica de *craft-cutting*, usando una grabadora laser CO₂ de 12W marca Rayjet [117]. Para fabricar la carcasa de plástico que aloja los dispositivos analíticos se utilizó una impresora 3D de Formlabs, Form 3, [118]. Se utilizó Matlab para realizar la prueba de concepto de los algoritmos, y la aplicación fue desarrollada con Android Studio con la biblioteca OpenCV.

5.3.3 **Fabricación y preparación del μ PAD**

Para la realización de los ensayos se requirió el diseño y fabricación de un soporte en 3D que contiene el dispositivo analítico. El papel requiere de un proceso de funcionalización de para que éste pueda producir una señal fluorescente mediante *Carbon Dots* (CD) cuando los analitos sean depositados. Los CD producen la fluorescencia del papel y posteriormente se usa la técnica de *quenching* que apaga la fluorescencia para que sea recuperada en función de la concentración de la sustancia depositada. Tanto la fabricación como la preparación del μ PAD fueron extensamente explicados en [114]. No se va a profundizar más en el proceso de preparación y caracterización de los analitos ya que en este trabajo de investigación solo se va a abordar el algoritmo de procesamiento de imagen para la medición de la señal fluorimétrica.

5.4 **Resultados**

En esta sección se han realizado un conjunto de experimentos para testear la aplicación empleada. Primero se realizarán pruebas para la determinación de la intensidad de fluorescencia del μ PAD sin normalización ratiométrica empleando distintas ISOs. Posteriormente se aplicará el uso del dispositivo con normalización ratiométrica para comprobar su estabilidad bajo distintas ISOs.

5.4.1 Determinación de intensidad de fluorescencia variando la ISO

En este experimento se han tomado un conjunto de fotos del dispositivo analítico sin normalización ratiométrica con el *smartphone* para distintas ISOs. En la Figura 5.3a se puede ver una imagen de ejemplo del μ PAD tomada con una ISO 2500 y el resultado de la detección de los 3 sensores, antes y después del análisis. Se puede observar que la aplicación es capaz de detectar las tres componentes perfectamente. En la Figura 5.3b se aprecia una reducción del radio de los sensores debido al proceso de erosión.

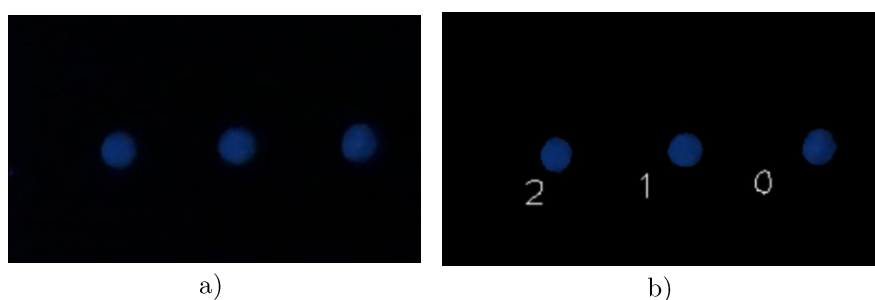


Figura 5.3. Imágenes del μ PAD captada a ISO 2500, a) antes de la detección, b) después de la detección y de aplicar la erosión.

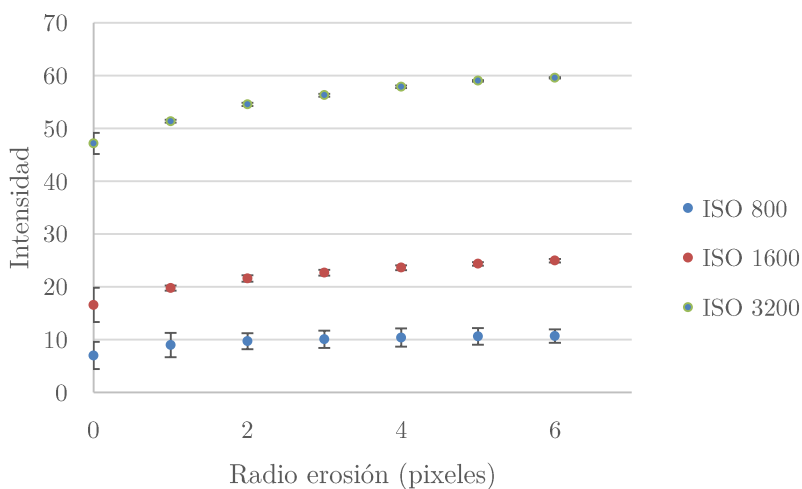


Figura 5.4. Intensidad de fluorescencia de las componentes detectadas en función del radio de erosión para diferentes ISOs.

Un aspecto a tener en cuenta es cómo afecta el tamaño del elemento de erosión en el proceso de detección de los sensores. Si éste es muy pequeño, la

erosión no será muy fuerte y la intensidad de los bordes afectará a la media del sensor. Por este motivo se va a realizar una optimización del tamaño del elemento con el que se realiza la erosión. Este elemento es un disco circular cuyo radio, medido en píxeles, determina su tamaño. Se ha realizado un experimento tomando imágenes a distintas ISOs y realizando varios análisis para cada imagen cambiando el radio del elemento de erosión. En la Figura 5.4 se muestran los resultados de este experimento donde se aprecia cómo el nivel de intensidad de fluorescencia, equivalente al nivel de gris, aumenta con el tamaño del disco de erosión. A partir de un radio de 3 píxeles, la intensidad tiende a estabilizarse y el error en la medición se reduce. Para la ISO 1600 y 3200 las rectas siguen creciendo, aunque con una pendiente menor que para tamaños del disco de erosión más bajos. Para poder usar este método de erosión en otros dispositivos solo habría que determinar el radio del elemento en relación a la resolución del dispositivo. De esta forma podríamos aplicar un porcentaje de erosión que se mantendría fijo entre dispositivos.

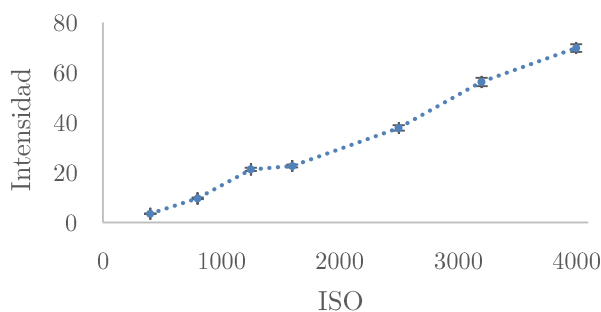


Figura 5.5. Intensidad de fluorescencia, equivalente al nivel de gris, en función de la ISO empleada en la captura de imágenes.

A continuación, se van a comparar los valores de intensidad de fluorescencia en función de la ISO empleada. Todos los análisis se han realizado con un radio del disco de erosión de 3 píxeles, ya que es el que vimos que estabilizaba la medida o hacía que el crecimiento fuera menos pronunciado. En la Figura 5.5 se puede ver cómo el nivel de intensidad aumenta con la ISO de forma lineal. Para poder analizar los valores extraídos a distintas ISOs es necesario normalizar estos datos usando el dispositivo con la línea de valores ratiométricos, que muestran la intensidad de fluorescencia del papel previa a la técnica de *quenching*. En la Figura 5.6 se pueden ver los resultados obtenidos tras analizar este dispositivo empleando distintas ISOs. Se puede observar que la intensidad de fluorescencia normalizada permanece estable a diferencia de la intensidad sin normalizar, que tiene una tendencia creciente. Este resultado

valida el procedimiento de normalización visto para poder usar este algoritmo empleando distintas ISOs.

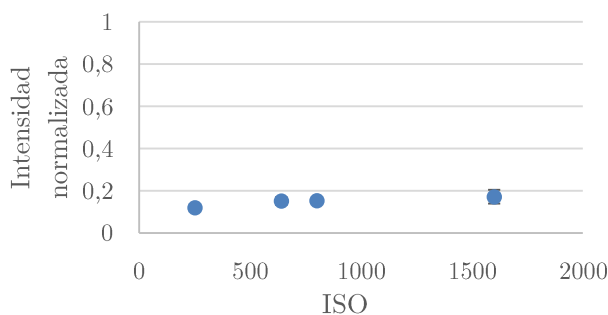


Figura 5.6. Intensidad de fluorescencia normalizada en función de la ISO.

5.5 Conclusiones y contribuciones

En este capítulo se ha presentado una aplicación móvil capaz de medir la intensidad de fluorescencia de un sensor fluorimétrico μ PAD. Este sensor usa la técnica de *quenching* para modular la fluorescencia emitida bajo una fuente de luz UV. La aplicación desarrollada es capaz de detectar los sensores del dispositivo y medir el valor medio de intensidad de forma eficaz. Además, permite identificar si el sensor fluorimétrico cuenta con línea de normalización ratiométrica o no. Todos los datos se pueden guardar en el teléfono y permite la realización de más de un análisis. Esta aplicación puede ejecutarse en cualquier dispositivo Android, lo que evita el uso de equipos más costosos para la determinación de la intensidad del sensor. Como trabajo futuro se quiere dar conexión online a la aplicación y que sea capaz de determinar la concentración de la sustancia medida en función del ensayo realizado. Esto requeriría de la precarga de las curvas de caracterización del analito y la elección del espacio de color usado en dicha calibración. El algoritmo desarrollado fue parte de la siguiente publicación científica en revista:

- Ortiz-Gomez, M. Ortega-Muñoz, A. Marín-Sánchez, I. de Orbe-Payá, F. Hernandez-Mateo, L.F. Capitan-Vallvey, F. Santoyo-Gonzalez, A. Salinas-Castillo, A vinyl sulfone clicked carbon dot-engineered microfluidic paper-based analytical device for fluorometric determination of biothiols, *Microchim. Acta.* 187 (2020) 421. <https://doi.org/10.1007/s00604-020-04382-9>.

Capítulo 6

Código QR fluorescente para la detección de cianuro

En este capítulo se presentará una aplicación móvil para la detección de QR fluorescentes para la determinación de la concentración de cianuro. Esta solución se abordará combinando la lectura de la información estática y dinámica del código QR. Además, se testeará un algoritmo de apilamiento de imágenes para tratar de mejorar la sensibilidad del dispositivo. Se presentará también el algoritmo desarrollado en Matlab para la detección y apilamiento de μ PADs fluorescentes.

6.1 Introducción

Los detectores de fluorescencia basados en teléfonos inteligentes están atrayendo la atención en campos como el de la salud y la seguridad alimentaria [2,119]. Por este motivo, el desarrollo de detectores de fluorescencia en imagen precisos, portátiles y de bajo coste es un campo de investigación muy prometedor. En el caso de la seguridad ambiental, la detección sencilla de alta sensibilidad y las técnicas de control remoto de bajo coste despiertan un gran interés. En esta línea, la detección de aniones de cianuro en el agua es importante para garantizar la seguridad en los residuos. Es por esto que la Agencia de Protección Ambiental determinó que el límite de detección de iones de cianuro en el agua potable tiene que ser de 2 ppm [120].

Para la determinación de cianuro se han desarrollado diferentes métodos basados en voltamperometría, electroquímica y cromatografía, entre otros [121–123]. Desafortunadamente, estas técnicas requieren de instrumentación costosa y complicada para realizar un análisis en tiempo real. Por este motivo, el diseño de sondas eficientes y rentables para detectar la contaminación por iones de cianuro, como los sensores colorimétricos y fluorescentes, se convierten en una solución muy prometedora. En [124] se reporta un método para la detección de cianuro usando la fluorescencia inducida por iones de cianuro y CD. La fluorescencia de los CD es apagada mediante *quenching* y es recuperada mediante la presencia de cianuro según su concentración. Además, se puede combinar el cambio de fluorescencia inducido por iones de cianuro con el procesamiento de las imágenes adquiridas con un teléfono inteligente [120].

En este Otra estrategia de diagnóstico interesante en aplicaciones ambientales es el desarrollo de dispositivos de análisis en papel (PAD) [125]. Debido a las características naturales del papel, como su bajo coste, flexibilidad, adaptabilidad, peso ligero, respetuoso con el medio ambiente, desechable, químicamente activo para la funcionalización y su posibilidad de hacer fluir líquido de forma natural, los PAD son candidatos ideales para el desarrollo de herramientas de análisis ambiental.

Las aplicaciones de detección de señales de fluorescencia mediante teléfonos inteligentes se basan en la implementación algoritmos de procesamiento de imágenes que detecten la señal del μ PAD. El uso de códigos QR como base para el desarrollo del μ PADs fluorescentes ha sido reportado en la literatura [91,92,96]. Sin embargo, estas investigaciones no exploran la línea de investigación de esta tesis doctoral, que consiste en combinar la detección de la información estática del código QR con la información dinámica del sensor. La mayoría de las cámaras integradas de los teléfonos inteligentes ofrecen una baja sensibilidad y un alto ruido de fondo en comparación con los equipos de imágenes más caros de laboratorio [126]. Además, en condiciones de baja luminosidad, donde se realizan los ensayos de fluorescencia, la sensibilidad de los *smartphones* es aún menor. El incremento de sensibilidad se puede lograr mejorando las características ópticas de las cámaras y realizando algoritmos de procesamiento de imágenes [126]. La primera aproximación se puede conseguir adquiriendo un *smartphone* que tenga mejores características ópticas, pero esto supondrá una solución más cara. La otra opción será aplicar algoritmos de procesamiento de imágenes que aumenten la sensibilidad. Uno de los algoritmos más prometedores que se encuentran en la literatura es la aplicación del apilamiento de imágenes [127]. Más adelante se testeará este algoritmo para la detección de cianuro y se comprobará si mejora la sensibilidad en la detección. A continuación, se procederá a explicar la conceptualización e implementación del sistema propuesto.

6.2 Conceptualización del sistema e implementación *software*

Se pretende implementar una aplicación móvil capaz de detectar un símbolo QR fluorescente para la determinación de cianuro. La idea es similar a la expuesta en capítulos anteriores, se quiere usar un símbolo QR para detectar información estática en el código e información dinámica en el color de los módulos para determinar la concentración del analito que se está midiendo.

En este caso, se va a desarrollar una aplicación móvil para la detección de cianuro en un código QR. La determinación de la concentración de cianuro se realiza haciendo uso del fenómeno de la fluorescencia, por lo que se hace necesario el diseño de un código QR fluorescente. A diferencia de las propuestas descritas en el Capítulo 3 y Capítulo 4, en las que los símbolos QR eran medidos bajo luz visible, para la propuesta presentada en este capítulo, la medición se realiza en unas condiciones de oscuridad e iluminación con luz ultravioleta. Esta diferencia implica modificaciones tanto en el diseño del símbolo QR, como en los sistemas presentados en el Capítulo 3 y en el Capítulo 4. De forma similar a lo presentado en el Capítulo 5, se quiere medir la fluorescencia aplicando un valor ratiométrico para que éste sea más robusto. La determinación de cianuro a concentraciones bajas puede ser un problema debido a que la señal de fluorescencia medida es muy débil, incluso por debajo de la sensibilidad del sistema. Como ya se mencionó anteriormente, para solucionar esta problemática se va a testear un algoritmo de mejora de la sensibilidad basado en el apilamiento de imágenes [127]. De este modo, en esta sección se va a presentar el diseño del código QR fluorescente propuesto para la determinación de cianuro a través de un ratiométrico. Además, se describirá el algoritmo diseñado para la detección del código QR fluorescente y los aspectos principales de su implementación en una aplicación móvil. Posteriormente, se presentará una visión final de la aplicación y de su funcionamiento de cara al usuario final. Finalmente, se abordará el algoritmo de apilamiento de imágenes desarrollado.

6.2.1 Diseño del QR fluorescente

El código QR de fluorescencia fue diseñado con el fin de medir la intensidad de fluorescencia aplicando un valor ratiométrico que será invariante. En la Figura 6.1 se puede ver un ejemplo de la apariencia del símbolo QR fluorescente visto bajo luz visible (Figura 6.1a) y bajo luz ultravioleta (Figura 6.1b). En este caso, la información estática del código QR se imprime con una tinta que contiene isotiocianato de rodamina B, que emite fluorescencia de un tono rojo. Los módulos de los analitos, donde se deposita el cianuro, marcados con borde azul claro en la Figura 6.1a, adquirirán un color azul bajo luz ultravioleta, como se aprecia en la Figura 6.1b, que dependerá de la concentración de cianuro. El valor para la corrección ratiométrica se extrae a partir del color rojo de los patrones de detección del símbolo QR bajo luz ultravioleta, marcados con borde verde en la Figura 6.1b. También se requiere extraer la fluorescencia del fondo del papel, que se obtendrá a partir del color azul oscuro de los

patrones de detección del QR, recuadros verdes en la Figura 6.1b. Esta medida servirá para mejorar el método de normalización ratiométrica, que se verá en la sección 6.4.2. Esta configuración del símbolo QR implica que se tiene que extraer información en canales RGB distintos. El símbolo QR es de un color rojo, que es perfectamente detectable en el canal R, sin embargo, en este canal no se detectarían los módulos de cianuro fluorescentes en color azul. Por lo tanto, es necesario extraer la información del canal B de los módulos de cianuro teniendo en cuenta que el código QR no es detectable en este canal porque es de color rojo. En la siguiente sección se expondrá el sistema propuesto para dar solución a esta problemática.

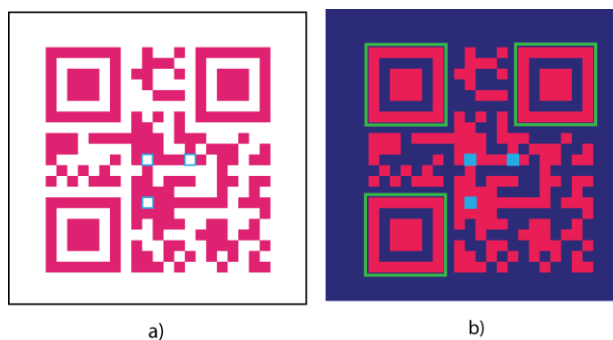


Figura 6.1. Ejemplo de diseños de un símbolo QR fluorescente para detección de cianuro bajo, a) luz visible, b) luz ultravioleta.

6.2.2 Esquema general del sistema de detección de códigos QR fluorescentes

En esta sección se va a profundizar sobre el procedimiento de detección de códigos QR fluorescentes. En la Figura 6.2 se presenta un esquema general del sistema que se usará para explicar la solución propuesta. El trabajo desarrollado en este capítulo toma como base la librería Zxing, explicada en la sección 2.6. Además, el sistema también se apoya en los desarrollos realizados en la sección 3.2.3 y 4.4.2. Para realizar la detección del código QR fluorescente, se parte de la imagen RGB, sin embargo, se requiere realizar la detección en el canal R, donde se visualiza claramente el código QR (Figura 6.2). Para ello se usa la implementación de lectura de códigos QR en distintos canales que se desarrolló en la sección 3.2.3, solo que en este caso la detección del código QR se hace solo en el canal R. En la sección 2.6.2, se explicó que el detector de símbolos QR de la librería Zxing realiza un proceso de umbralización para detectar los módulos negros como 1s y los blancos como 0s. Sin embargo, en el caso del QR fluorescente, en la Figura 6.2 se puede observar

cómo la imagen del canal R representa un código QR invertido, provocando un error en la detección. Para resolver este problema, hay que modificar el procedimiento de umbralización invirtiendo la asignación; en este caso los módulos blancos serían 1s y los negros serían 0s. Una vez se ha realizado la detección del símbolo QR, se obtiene un objeto *BitMatrix* de la dimensión del QR. A partir de este objeto, se decodifica la información estática del código QR, como se explicó en la sección 2.6.2.

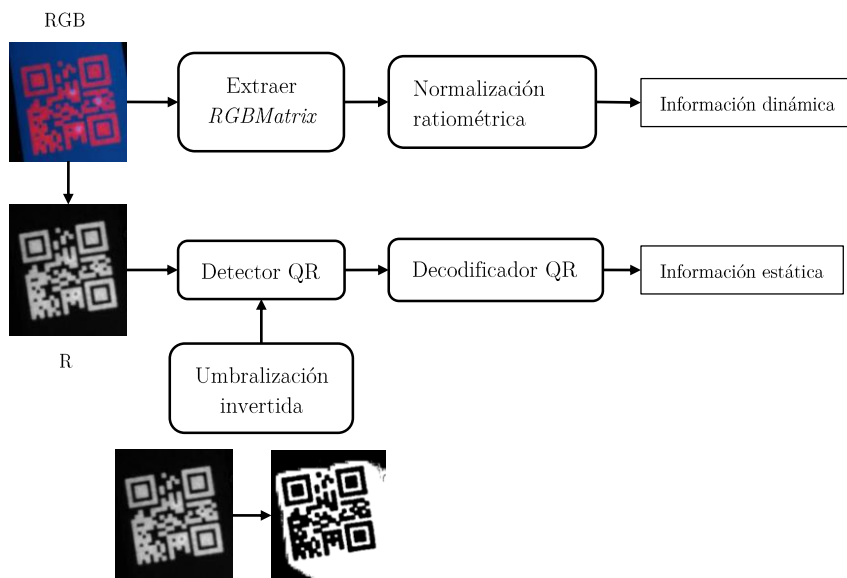


Figura 6.2. Esquema general del sistema de detección de códigos QR fluorescentes.

Cómo se explicó anteriormente, en la imagen extraída del canal R, no es posible detectar los módulos del cianuro que aparecen en color azul. Por este motivo, es necesario extraer esta información del canal B teniendo en cuenta que el código no se visualiza en este canal. Para ello, se usa la implementación propuesta en la sección 4.4.2, en la que se emplea la clase *RGBMatrix* para extraer el valor RGB de los módulos del símbolo QR. La clase *RGBMatrix* necesita usar la información de la transformación de perspectiva obtenida en el detector QR, a partir del canal R, para poder generar una matriz RGB de la dimensión del QR. De esta forma, se conoce el valor RGB de cada uno de los módulos del símbolo QR. Posteriormente, se realiza la extracción del valor RGB de los módulos donde se deposita el cianuro y de los módulos con el valor ratiométrico en el patrón de detección del QR. Las posiciones de los módulos de cianuro y del patrón del QR son definidas y precargadas previamente en el sistema. Estas referencias se definen a partir del tipo de código

QR analizado, dando la posibilidad al sistema de poder almacenar distintas configuraciones dependiendo del analito que se quiera analizar y del tamaño del QR empleado. Una vez extraídos estos valores, se procede a la normalización de los valores en función del ratiométrico. Como el color del ratiométrico es distinto al del analito, es necesario realizar un estudio para determinar qué fórmula y sobre qué canales hay que aplicar este cálculo; este estudio se realizará en la sección 6.4.2. Finalmente, se procede al cálculo de la concentración del analito a partir de su curva de caracterización, que se determina en un proceso de calibración previo; el estudio de la curva de caracterización se abordará en las secciones 6.4.1 y 6.4.2. Como los datos se normalizan con un valor ratiométrico, los valores del QR y de la curva de calibración pueden compararse entre sí.

6.2.3 Arquitectura de la aplicación y experiencia de usuario

Esta implementación se ha llevado a cabo a partir de la aplicación desarrollada en el Capítulo 3. Se definió una nueva opción de análisis para códigos QR fluorescente dentro de la aplicación y se implementaron nuevas funcionalidades para actualizar el *software*. Dentro de las funcionalidades añadidas se encuentra la implementación de la clase *RGBMatrix* en el código, que permite la construcción de una matriz RGB de las dimensiones del símbolo QR. La inclusión de esta clase implica realizar modificaciones en varias clases de la librería Zxing para permitir la utilización de objetos del tipo *RGBMatrix*. Además, se implementaron las clases *QRCodeReferences* y *QRCodePatternFormat*, que almacenan las posiciones de los módulos de los analitos, las curvas de caracterización precargadas y las posiciones de los patrones de detección de cada tipo de código QR. Todas estas nuevas implementaciones se agregaron en el paquete *tools*, manteniendo el código organizado. También, se hicieron modificaciones en la clase *HybridBinarizer* para permitir usar la umbralización invertida solo en el caso de que se estén analizando códigos fluorescentes. El cálculo del valor del ratiométrico y de la concentración se realizan en una clase llamada *QRCodeAnalysis*, que se implementa en el paquete *model*. La experiencia de usuario de la aplicación es similar a la mostrada en la sección 3.3.1, ya que la detección de códigos QR se implementa como una funcionalidad extra de dicha aplicación. La única diferencia es que en los resultados mostrados se indica la concentración detectada del analito.

6.2.4 Algoritmo de apilamiento

En casos donde la concentración de cianuro es demasiado baja, la sensibilidad de las cámaras de los *smartphones* no es suficiente para detectar correctamente los cambios en la concentración. En esta tesis doctoral se desarrolló y testeó un algoritmo de apilamiento para aumentar la sensibilidad del sistema basada en los trabajos propuestos en [126–128]. Concretamente, en [128] se presenta una disminución del LOD de un factor 10, gracias a la aplicación del apilamiento de imágenes. La base de este algoritmo es tomar un vídeo del sensor fluorescente con una cámara fija y promediar los fotogramas del vídeo para aumentar la sensibilidad de la medición. Se pretende implementar este algoritmo en la aplicación de detección de QR fluorescentes para detección de cianuro. Sin embargo, previamente se va a realizar una prueba de concepto en Matlab para testear si este algoritmo aumenta la sensibilidad para la detección de cianuro. El sistema tomará como entrada un vídeo del μ PAD diseñado para medir la recta de caracterización de cianuro. A partir de dicho vídeo, se desarrollará un algoritmo de detección automática del μ PAD, partiendo del implementado en la sección 5.2.1. De este modo, el nuevo algoritmo integrará la técnica de apilamiento de imágenes para comprobar si éste mejora la sensibilidad del dispositivo. Para evaluar el rendimiento de dicho algoritmo se va a medir la SNR de la señal fluorescente y el LOD de la recta como indican las ecuaciones (39) y (40):

$$SNR = \frac{I_{analito}}{\sigma} \quad (39)$$

$$LOD = \frac{3\sigma}{a} \quad (40)$$

siendo $I_{analito}$ la señal de fluorescencia del analito medido, σ la desviación de los pixeles de la señal de fluorescencia del papel y a la pendiente de la recta de caracterización del cianuro, cuyo cálculo se realizará en la sección 6.4.1.

En la Figura 6.3 se muestra un esquema del algoritmo de apilamiento desarrollado en Matlab. Éste parte de un vídeo del μ PAD tomado con el dispositivo en una posición fija. Al estar el dispositivo de captación en una posición fija, es posible aplicar el apilamiento, ya que el μ PAD siempre permanece en la misma posición. A partir de este vídeo, se hace un promedio de todos los fotogramas para obtener una imagen apilada. Esta imagen apilada se verá más emborronada, pero se espera que la desviación entre los pixeles disminuya, reduciendo el error en la medida. A partir de esta imagen apilada se realiza

la detección de cada uno de los círculos del μ PAD y la extracción de la intensidad de fluorescencia promedio para cada punto. El resultado esperado de esta técnica es que, al promediar la imagen, la desviación del error, definida como la desviación estándar del punto de control blanco, debería disminuirse. Para realizar este procedimiento se implementó un código en Matlab basado en el algoritmo presentado en la sección 5.2.1. La diferencia de esta implementación es que, para evaluar el apilamiento, no se toma el valor ratiométrico, sino la intensidad de la señal de fluorescencia. Una vez detectada la señal ratiométrica de cada sensor del μ PAD, se extrae la recta de caracterización haciendo un promedio de los sensores donde se haya depositado la misma concentración. Si se quisiera extraer el valor ratiométrico de la recta, habría que tomar la intensidad del punto ratiométrico, en rojo, y aplicar el apilamiento sobre la recta apilada. Además de la intensidad promedio de cada sensor, también se extrae su desviación estándar, incluyendo la del punto de control. Con estos valores es posible medir la SNR y el LOD del sistema para un número de fotogramas específico.

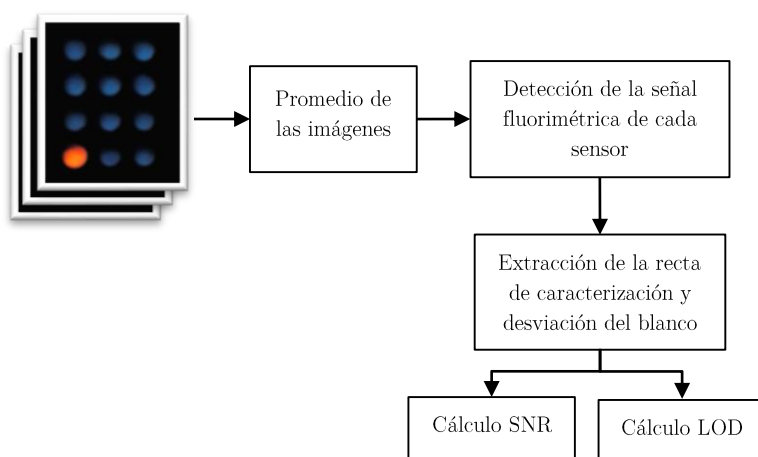


Figura 6.3. Esquema de funcionamiento del algoritmo de apilamiento.

6.3 Materiales y métodos

En esta sección se presentarán los materiales y métodos empleados durante el desarrollo del trabajo incluido en este capítulo. Primero se presentarán los reactivos y materiales con los que se llevaron a cabo los experimentos realizados, así como el *hardware* y el *software* utilizado para realizar las medidas e implementar los algoritmos diseñados. Posteriormente, se presentará el proceso de fabricación y preparación del μ PAD usado para caracterizar la recta

de cianuro. Finalmente, se mostrará el proceso de fabricación y preparación del QR fluorescente, así como el procedimiento de medida para la extracción de la señal colorimétrica.

6.3.1 Reactivos y materiales

Los reactivos químicos empleados son de calidad analítica y se utilizaron tal y como se recibieron, sin purificación adicional, adquiridos de Sigma-Aldrich [115]. Se prepararon soluciones de menor concentración mediante una dilución adecuada con agua Milli-Q. Se emplearon los siguientes reactivos: hidróxido de sodio 98%, cianuro de sodio, ácido bórico, trietilamina (TEA), hidrogenofosfato de sodio, dihidrogenofosfato de sodio, hidrogenocarbonato de sodio, carbonato de sodio, ácido cítrico anhidro (CA), ácido yodoacético (IAA), divinilsulfona (DVS), polietilenimina ramificada 25 KDa (25 Kpei), N,N-diciclohexilcarbodiimida (DCC), N,N-diisopropyletilamina (DIPEA), isotiocianato de rodamina B, tensioactivo Triton X-100 y quitosano. Se usó una bolsa de diálisis con peso molecular: 1 KDa. El papel de filtro para cromatografía Whatman grado 1 se adquirió de Sigma-Aldrich [115]. El cartucho de tinta recargable para Epson se compró en Epson Corporation [129]. Todas las soluciones acuosas se prepararon utilizando agua de calidad del tipo de ósmosis inversa (estación Milli-RO 12 plus Milli-Q de Millipore, conductividad 18.2 M Ω ·cm).

6.3.2 Hardware y software

La intensidad de fluorescencia del dispositivo analítico de papel funcionalizado se midió con un teléfono inteligente Pocophone F1 [98]. Este dispositivo tiene una cámara integrada de 12 + 5 megapíxeles con enfoque automático de doble píxel y sensor CCD; 6 GB de RAM y procesador Snapdragon 845. El dispositivo Pocophone F1 también fue usado para alojar la aplicación móvil para la decodificación de códigos QR implementada. Las imágenes fueron tomadas después de la iluminación con un transiluminador UV de 220 V (serie MUV 21) [116]. Los espectros de fluorescencia se obtuvieron utilizando el espectrofotómetro de fluorescencia UV-Vis Cary Eclipse [130]. Se utilizó un medidor de pH (Crison micro pH 2000) para medir los valores de pH de las soluciones acuosas [131]. El procedimiento de impresión se realizó en una impresora de inyección de tinta Epson ME-10 [132]. Para caracterizar la tinta se utilizaron el viscosímetro Visco Basic Plus (Fungilab, Barcelona, España) [133] y el tensiómetro de superficie Theta Lite Attention (Biolin Scientific AB, Estocolmo, Suecia) [134]. El tamaño, la morfología y la caracterización de los

CD se llevó a cabo mediante microscopía electrónica de transmisión de alta resolución (HRTEM) utilizando un microscopio FEI TITAN G2 que funciona a 200 kV [135]. Las mediciones de los espectros XPS se realizaron en una espectroscopia de fotoelectrones de rayos X (Kratos Axis ultra-DLD) para la caracterización del papel funcionalizado y los CD [136]. Todos estos estudios se llevaron a cabo en el Centro de Instrumentación Científica (Universidad de Granada, España).

La recta de caracterización de cianuro fue extraída manualmente usando el *software* GIMP de procesamiento de imágenes y automáticamente usando un *script* desarrollado en Matlab. La aplicación móvil fue desarrollada en Android mediante Android Studio. Para la implementación de la aplicación se empleó la librería Zxing para facilitar la lectura de códigos QR [88] y la librería OpenCV de procesamiento de imagen [137]. Se empleó la aplicación móvil openCamera, que permite adquirir vídeos con diferentes parámetros de la cámara (ISO, tiempo de exposición, etc.). Finalmente, se usó Adobe Illustrator para diseñar los códigos QR fluorimétricos.

6.3.3 Preparación de las tintas

En esta sección se describirá el procedimiento para la preparación de las tintas usadas para imprimir el QR. Se emplearon dos formulaciones diferentes de tintas en la construcción del código QR fluorescente. La formulación de tintas se preparó para obtener una tensión superficial y una viscosidad apropiadas en el rango de 30-50 mN/m y 2-5 mPas, que se requiere para una impresión de inyección de tinta suave y continua [138]. La tinta utilizada para imprimir el QR con información estática (tinta 1) se preparó a partir de una solución hidroalcohólica de agua y etanol, TritonX-100 (2:2:4% en peso) e isotiocianato de rodamina B en una concentración de 1 mg/ml. La tinta utilizada para imprimir los módulos sensores del código QR para la detección de cianuro (tinta 2) se preparó a partir de agua, solución Triton X-100 (2:4 % en peso) y puntos cuánticos de carbono sintetizados (1.72 ng/mL) [114]. Las tintas preparadas se desgasificaron mediante un baño ultrasónico para evitar el gas disuelto, que inhibe el chorro. A continuación, las tintas se filtraron a través de un filtro de 0.45 μm antes de llenar el cartucho para una inyección óptima. Después de cada experimento de impresión, se realizó un lavado automático de la impresora para evitar el bloqueo de la boquilla por agregados de moléculas.

6.3.4 Fabricación del código QR

El código QR fluorescente se fabricó mediante el método de impresión por inyección. Se agregaron 4 ml de las tintas mencionadas anteriormente en cartuchos separados. La tinta 1 se añadió al cartucho magenta y la tinta 2 se añadió al cartucho azul, véase Figura 6.4a. El sustrato de papel utilizado para imprimir el código QR fue funcionalizado siguiendo el procedimiento descrito en [114], con pequeñas modificaciones con el fin de obtener una superficie activa para la modificación covalente de la celulosa durante la impresión.

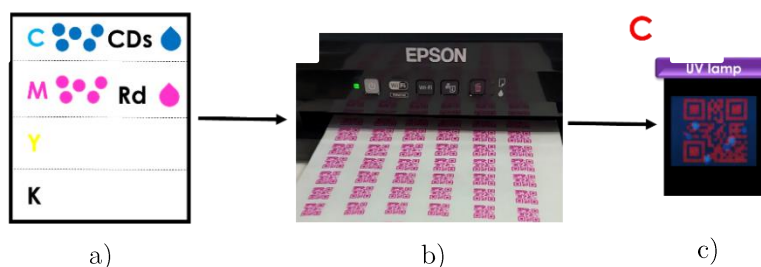


Figura 6.4. Procedimiento de fabricación del código QR fluorescente, a) esquema de las tintas en los cartuchos de impresión, b) impresión de los códigos QR, b) código QR iluminado con la lámpara UV.

Se sumergió un papel de filtro Whatman grado 1 (hoja de $203 \times 254 \text{ mm}^2$) en una solución de DVS al 10% (v/v carbonato de sodio 0.5 M, pH 12) para activar la celulosa. La mezcla se agitó con un agitador a 250 rpm durante 12 horas a temperatura ambiente. Después de la activación, la hoja de papel se lavó con etanol/agua 1:1 y se secó a temperatura ambiente. Posteriormente, la hoja de papel se colocó en papel de aluminio en condiciones ambientales para protegerla de la luz y se almacenó en una atmósfera seca de nitrógeno hasta su uso. Por medio del proceso de impresión por inyección de tinta, se rociaron gotas de soluciones de tinta (tinta 1 y tinta 2) sobre la superficie del papel de celulosa activada para impresión, lo que permitió la inmovilización covalente de CD y rodamina en la superficie del papel antes de la evaporación completa de las gotas durante la impresión del código QR fluorescente. Los CD y los patrones de rodamina impresos en papel de celulosa fueron diseñados con precisión por una computadora que controla la impresora. Los códigos QR fluorescentes resultantes se lavaron con agua Milli-Q, se secaron al vacío a temperatura ambiente y se almacenaron en un ambiente seco en la oscuridad hasta su uso. En la Figura 6.4b y Figura 6.4c se puede ver el proceso de

impresión de los códigos QR mediante la impresora y la imagen tomada del QR tras iluminarla con la lámpara de UV (λ excitación = 365 nm).

6.3.5 Funcionalización del código QR en papel y μ PAD para análisis de cianuro

En esta sección se describirá el proceso de funcionalización del código QR en papel para el análisis de cianuro mediante CD. Para realizar este proceso es necesario inmovilizar los CD en los puntos del QR en los que se vaya a medir la señal fluorescente. Esta señal es apagada con una solución de cobre que recuperará su fluorescencia en función del contenido de cianuro que contenga la muestra analizada. Los CD se sintetizaron utilizando ácido cítrico como fuente de carbono y polietilenimina (PEI) como fuente de nitrógeno mediante un proceso hidrotermal. La síntesis se realizó según el procedimiento descrito en [139]. Los códigos QR fluorescentes impresos se funcionalizaron para la detección de cianuro en agua. Para ello, en las posiciones de detección del código QR fluorescente donde los CD estaban inmovilizados covalentemente se dejaron caer en 0.5 μ L de solución de cobre 2 μ M para apagar la fluorescencia de los puntos de carbono [139]. El código QR se dejó secar a temperatura ambiente durante 5 minutos. Para la determinación de iones cianuro en agua, el código QR fluorescente se sumergió en una solución que contenía cianuro (0.1 y 2 μ M) preparada en tampón borato (pH 10). El proceso de funcionalización del μ PAD de la recta de caracterización se realizó mediante la inmovilización covalente de los CD [114]. El proceso de apagado de la fluorescencia y preparación del cianuro fue similar al seguido para el código QR.

6.3.6 Procedimiento de medida y extracción de la señal colorimétrica

Para la caracterización de la recta de cianuro se captan los vídeos a distintas ISOs (200, 400 y 800) por medio de la aplicación OpenCamera en el dispositivo Pocophone F1. El dispositivo móvil fue colocado en un soporte fijo para que éste se mantuviera inmóvil durante la captura del vídeo y favorecer el algoritmo de apilamiento. La recta de caracterización de la sección 6.4.1 fue extraída manualmente a partir del fotograma 32 del vídeo de ISO 400, después del primer segundo de duración, usando el *software* de procesamiento de imágenes GIMP. Los vídeos a distintas ISOs fueron usados para evaluar el algoritmo de apilamiento presentado en la sección 6.2.4. El código QR fluorescente fue analizado mediante la aplicación móvil presentada en la sección

6.2.3. Las imágenes del QR fluorescente se tomaron con el dispositivo Pocopone F1 a una ISO 400. Tanto el μ PAD como el QR fluorescente fueron captados en condiciones de oscuridad, usando como fuente de luz una lámpara de rayos UV.

6.4 Resultados

Para evaluar el rendimiento del sistema propuesto se han realizado un conjunto de experimentos que se describen en la siguiente sección. Para empezar, se presentarán los resultados de la caracterización de los CD y de la recta de caracterización de cianuro. Posteriormente, se optimizará el sistema para determinar qué canal RGB del sensor ratiométrico es el más indicado para comparar los resultados entre distintas ISOs. Finalmente, se evaluará el rendimiento del algoritmo de apilamiento sobre la recta de caracterización de cianuro evaluando su SNR y su LOD en función del número de fotogramas apilados.

6.4.1 Caracterización de la recta de cianuro

En esta sección se va a presentar la caracterización de la recta de cianuro a partir del μ PAD. En la Figura 6.5 se muestra una imagen de la recta de caracterización del μ PAD tomada con el dispositivo móvil a una ISO de 800. Esta recta está formada por dos réplicas de cada concentración depositada, que van desde los $0.1 \mu\text{M}$ a los $2 \mu\text{M}$. Además, se cuenta con el punto de control de la fluorescencia del papel y el punto con el que se aplicará el ratiométrico de rodamina (en rojo). La equivalencia de los círculos de la imagen del μ PAD y su concentración es la siguiente: 1 y 2 ($2 \mu\text{M}$); 3 y 4 ($1 \mu\text{M}$); 5 y 6 ($0.5 \mu\text{M}$); 7 y 8 ($0.2 \mu\text{M}$); 9 y 10 ($0.1 \mu\text{M}$); 11 (papel); 12 (ratiométrico de rodamina).

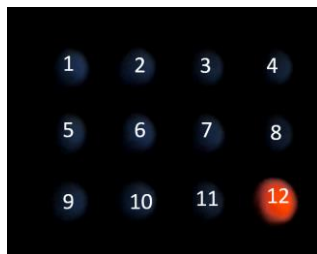


Figura 6.5. Imagen de la que se extrae la recta de caracterización de cianuro a ISO 400.

A partir de esta imagen se extrae la recta de caracterización de cianuro para cada canal RGB y el nivel de gris mostrada en la Figura 6.6. En la recta se ha eliminado el punto de 1 μM debido a que éste salió erróneo por el procedimiento experimental. Se puede observar que la recta de caracterización tiene un comportamiento lineal en todos los canales y que el canal que presenta mayor fluorescencia es el canal B. La curva de caracterización puede ajustarse a una ecuación de una recta $I = aC + b$ donde, I es la intensidad y C es la concentración de cada punto de la recta. Los parámetros de la recta a y b hacen referencia a la pendiente y a la ordenada en el origen, respectivamente.

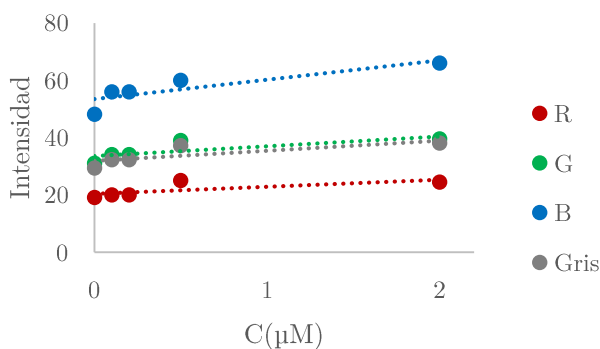


Figura 6.6. Rectas de caracterización de cianuro en los canales RGB y nivel de gris.

	R	G	B	Gris
a	2.4495	3.4012	6.7887	3.6306
b	20.345	33.598	53.433	3.5027
σ	2.5122	3.2903	4.6718	3.0951
LOD (μM)	3.0768	2.9022	2.0648	2.7300

Tabla 6.1. Resultados de la recta de caracterización en cada canal.

En la Tabla 6.1 se presentan los resultados de la recta de caracterización para cada canal. Se puede ver cómo la recta que presenta más pendiente es la del canal B. En la tabla también se calcula la desviación estándar, σ , de los píxeles del punto de control (la fluorescencia del papel) y el LOD. Se puede observar que el canal B es el que presenta una mayor desviación del punto de control, pero al tener mayor pendiente, es el canal que presenta un LOD más bajo. Por este motivo se escoge el canal B para medir la fluorescencia del cianuro. Se puede ver que el LOD del canal B, aun siendo el más bajo, está en torno a 2 μM , lo que nos indica que el sistema presenta una sensibilidad

demasiado baja para medir correctamente concentraciones bajas de cianuro. Es por esto que en la sección 6.4.4 se testea la aplicación de un algoritmo de apilamiento para intentar mejorar la sensibilidad.

6.4.2 Empleo del ratiométrico R a distintas ISOs

En el dispositivo analítico diseñado para medir cianuro se ha colocado un punto ratiométrico de rodamina que emite fluorescencia en el canal rojo. Aplicando este punto en la recta correctamente, se puede obtener un valor ratiométrico de la recta que pueda ser invariante a los cambios de ISO. Por otro lado, es importante poder saber qué canal es el más adecuado para medir el ratiométrico y cómo debe ser aplicado sobre la recta para que ésta sea invariante.

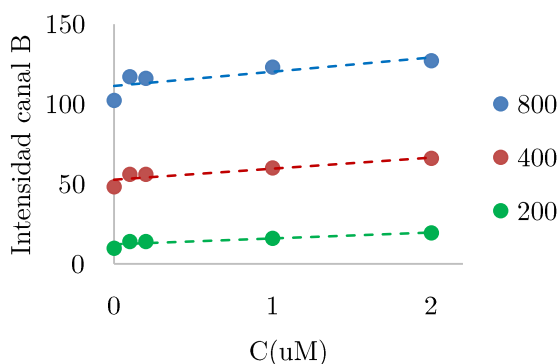


Figura 6.7. Rectas de caracterización del cianuro del canal B en distintas ISOs.

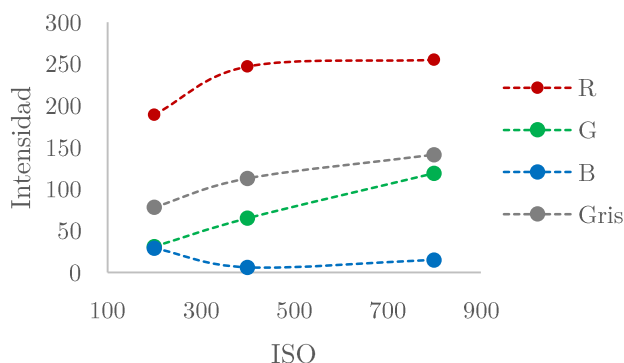


Figura 6.8. Evolución de la intensidad del ratiométrico en función de la ISO para cada canal RGB y nivel de gris.

En la Figura 6.7 se muestran las rectas de caracterización del cianuro medidas bajo distintas ISOs para el canal B. Se puede ver cómo el aumento de la ISO genera un aumento en la intensidad de los píxeles elevando la recta. Por otro lado, para determinar qué canal del ratiométrico es el más indicado, es necesario saber qué canal es el que representa mejor la variación de la ISO. En la Figura 6.8 se presenta una gráfica de la evolución de la intensidad del ratiométrico para cada canal en función de la ISO. Se puede observar que la intensidad del ratiométrico aumenta en función de la ISO para todos los canales excepto para el canal B. Es por esto que los principales candidatos serían los canales R, G y el nivel de Gris. En nuestro caso, se va a escoger el ratiométrico en el canal R, ya que es el que más luz refleja y puede representar mejor los cambios del aumento de la ISO. Para aplicar el ratiométrico a la recta de caracterización se ha usado la siguiente ecuación:

$$I_{sensor}^{rat} = \frac{I_{sensor}^B - I_{papel}^B}{I_{rat}^R} \quad (41)$$

donde I_{sensor}^{rat} es la intensidad ratiométrica del sensor medido, I_{sensor}^B es la intensidad del sensor en el canal B, I_{papel}^B es la intensidad de fluorescencia del papel en el canal B y I_{rat}^R es la intensidad del ratiométrico medida en el canal R. La intensidad del sensor se mide en el canal B puesto que es el que mejor representa la caracterización del sensor de cianuro, como se pudo ver en la sección anterior. Se ha hecho uso de la intensidad del blanco del papel para eliminar la influencia de éste en la medida. El blanco del papel es el punto de concentración 0 en la recta de la Figura 6.7 y ésta guarda relación con el aumento de la ordenada en el origen. Por otro lado, la intensidad del ratiométrico nos va a modificar ligeramente las pendientes de las rectas para que puedan ser comparadas entre sí.

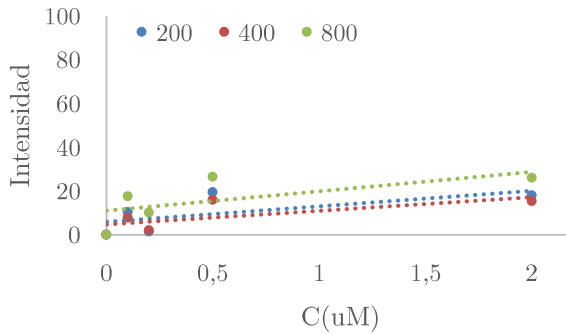


Figura 6.9. Recta de caracterización ratiométrica del cianuro a distintas ISOs.

ISO	200	400	800
a	7.1373	6.1851	8.858
b	5.8428	4.7291	11.04
σ	1.6496	4.7844	11.5406
LOD (μM)	0.6934	2.3206	3.9085

Tabla 6.2. Datos de la recta de caracterización ratiométrica a distintas ISOs.

En la Figura 6.9 se pueden ver las rectas ratiométricas de caracterización del cianuro a distintas ISOs, normalizadas entre 0 y 255. Se puede ver cómo las rectas se han ajustado al mismo rango dinámico, lo que permite que se puedan comparar los datos extraídos a diferentes ISOs. La recta tomada a ISO 800, presenta una variación mayor respecto a las otras. Esto puede ser debido a que el ratiométrico del canal R sature el sensor de la cámara para esa ISO, como puede verse en la Figura 6.8, y que no ajuste correctamente los datos. Además, en la Tabla 6.2, se pueden ver los parámetros de la recta de caracterización para cada ISO. Para la detección de la concentración de las muestras de cianuro a partir de la intensidad de fluorescencia es necesario extraer la recta de caracterización inversa, $C = Ia' + b'$ y donde $a' = 1/a$ y $b' = b/a$. En la Tabla 6.3 se presentan los parámetros de la recta de caracterización inversa para ISO 400 que se utilizarán en la sección 6.4.3 para evaluar la determinación de la concentración de cianuro en el QR fluorescente.

ISO	400
a'	0.1616
b'	-0.7646

Tabla 6.3. Parámetros de la recta de caracterización inversa a ISO 400.

6.4.3 Detección del QR fluorescente

Para evaluar el rendimiento de la aplicación móvil para la detección de códigos QR fluorescentes, se va a testear el sistema midiendo un par de símbolos QR en los que se ha depositado cianuro en tres de sus módulos. La fabricación y funcionalización de los códigos QR fueron presentadas previamente en las secciones 6.3.4 y 6.3.5. En la Figura 6.10 se presenta un conjunto de imágenes que representan el proceso de detección de ambos códigos QR en el sistema presentado. La Figura 6.10a muestra la imagen original del símbolo QR en el que se ha depositado 0.1 μM de cianuro en tres módulos del símbolo QR. En la Figura 6.10b se presenta la imagen umbralizada con el método de umbralización invertida presentado en la sección 6.2.2. Posteriormente, se

realiza la extracción del objeto *RGBMatrix* que representa una matriz del tamaño de la dimensión del código QR. Cada punto de esta matriz es el promedio de los píxeles de cada módulo del símbolo QR, véase la Figura 6.10c. Las Figura 6.10d, 6.10e y 6.10f representan el mismo proceso, pero para una concentración de cianuro de 2 μM . Paralelamente, la aplicación detecta la información estática codificada en el código QR, dando como resultado en ambos casos el texto “01234”.

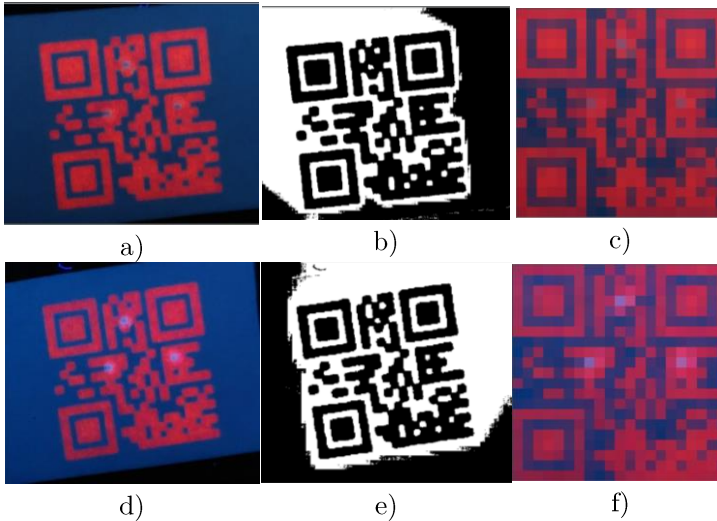


Figura 6.10. Imágenes del QR fluorescente con cianuro depositado en distintas etapas de la detección a ISO 400. Concentraciones de cianuro a)-c) 0.1 μM , d)-f) 2 μM .

La información dinámica es extraída a partir de los puntos de la matriz RGB del código QR de la Figura 6.10c. A partir de las posiciones de los módulos de interés precargados en la aplicación se puede obtener: la intensidad del ratiométrico, la intensidad del papel y la intensidad de los tres módulos con cianuro. Con estos valores se puede determinar la intensidad ratiométrica del sensor y calcular la concentración de cianuro, como se explicó en la sección 6.4.2. En la Tabla 6.4 se presenta la concentración estimada del cianuro a partir de la información extraída del código QR. La medición del sensor, I_{sensor}^B , se obtiene promediando la intensidad de los tres módulos de cianuro. Los valores de I_{papel}^B y I_{rat}^R se extraen como promedio de los módulos azules y rojos de los patrones de detección, respectivamente, como se explicó en la sección 6.2.1. El valor de intensidad ratiométrica del sensor se calcula según la ecuación (41). En la Tabla 6.4 se puede observar que la concentración estimada, C_{est} , es superior a la concentración depositada, C_d . Para la concen-

tración de 2 μM la diferencia es más acuciante, obteniendo un valor de concentración estimada hasta cinco veces mayor. Este desajuste puede deberse principalmente a fallos en la ejecución del experimento con el QR o con la recta de caracterización. Esto puede comprobarse comparando la intensidad de fluorescencia del cianuro en el QR y la de la recta de caracterización tomada a la misma ISO, que fue 400. Se puede observar cómo en el código QR se mide una intensidad de 160, sin embargo, en la recta de caracterización, para la misma concentración, se tiene una intensidad de 66. Para mejorar esto sería necesario realizar la medición de la recta de caracterización en el mismo símbolo QR y llevar a cabo varias réplicas del experimento para asegurar su reproducibilidad. A pesar del desajuste en los resultados obtenidos en la concentración final, se ha demostrado que el sistema es capaz de extraer la información estática y dinámica de un símbolo QR fluorescente.

$C_d(\mu\text{M})$	I_{papel}^B	I_{rat}^R	I_{sensor}^B	$I_{\text{sensor}}^{\text{rat}}$	$C_{\text{est}}(\mu\text{M})$
2	117	173	160	63.3815	9.4829
0.1	92	188	97.3333	7.234	0.405

Tabla 6.4. Datos de la determinación de la concentración de cianuro en el QR fluorescente.

6.4.4 Evaluación del algoritmo de apilamiento

En esta sección se va a evaluar el funcionamiento del algoritmo de apilamiento presentado en la sección 6.2.4. Este apilamiento se va a realizar sobre los vídeos tomados de la recta de caracterización a diferentes ISOs. El objetivo del apilamiento es conseguir que se reduzca la desviación estándar de la intensidad de fluorescencia del papel tomada como punto de control. Si ésta se reduce, se espera que la SNR aumente y que el LOD disminuya. En las Figura 6.11a y 6.11b, se muestra la evolución del triple de la desviación estándar del papel y la pendiente de la recta de caracterización en función del número de fotogramas apilados para ISO 800 y 400. Se puede observar que la desviación estándar del papel disminuye ligeramente y que la pendiente de la recta permanece prácticamente constante. La desviación del blanco presenta un rango de variación entre 0.5 y 1.5 que es bastante poco teniendo en cuenta que se muestra el valor triplicado. En las Figura 6.11c y 6.11d se presenta el LOD para ambas ISOs en función del número de fotogramas. Se puede apreciar que el LOD presenta una tendencia confusa en cada ISO. En ambos casos para más de 150 fotogramas apilados presenta una tendencia descendente con picos de subida en algunos puntos. Además, la variación entre el LOD de partida

y el LOD al que tiende el apilamiento es muy pequeño, alrededor de $0.3 \mu\text{M}$, que deja el LOD tras el apilamiento muy por encima del necesario para medir concentraciones bajas de cianuro. En las Figura 6.11e y 6.11f se muestra la evolución de la SNR de cada punto de la recta con el número de fotografías. Se puede ver que la SNR aumenta, pero muy ligeramente, alcanzando un aumento en torno a 0.5 unidades para ambas ISOs.

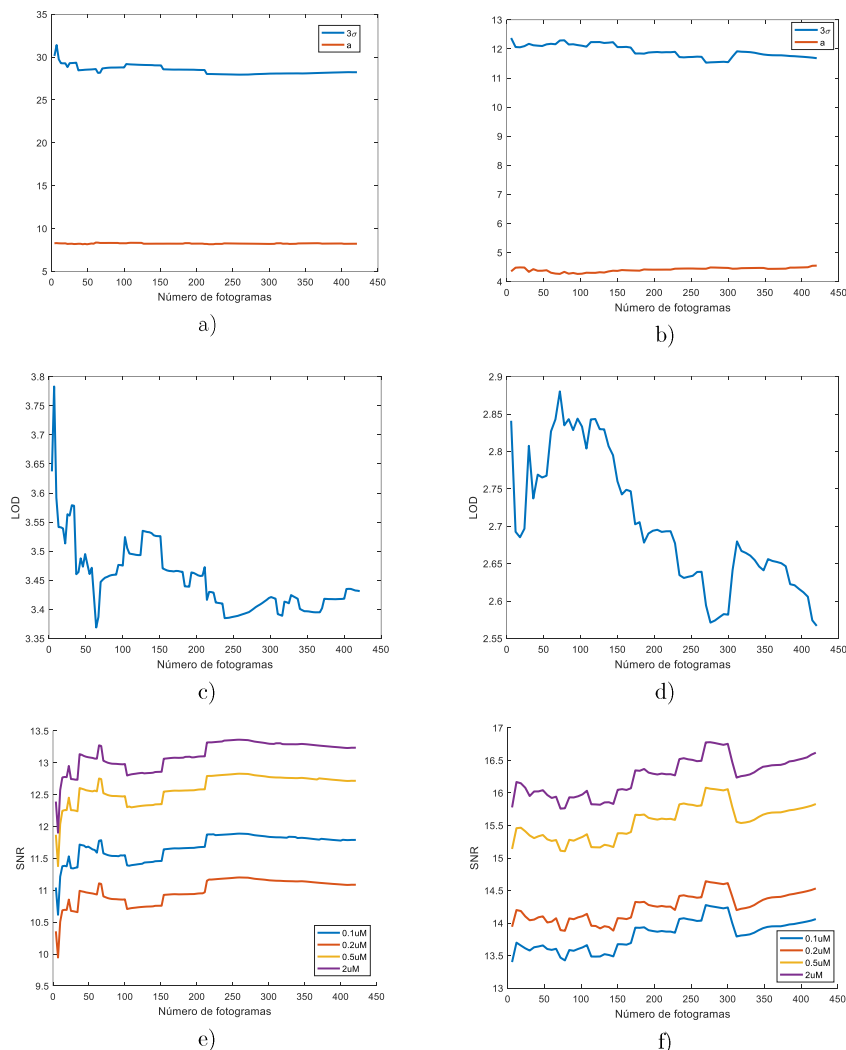


Figura 6.11. El triple de la desviación estándar y la pendiente de la recta para ISO, a) 400 y b) 800, LOD para ISO c) 400 y d) 800, SNR para ISO e) 400 y f) 800.



Figura 6.12. Imagen de la recta de caracterización a ISO 200.

Un caso particular se puede extraer de la recta de caracterización medida a ISO 200. En este caso, la imagen fue tomada bajo condiciones muy extremas de luminosidad (Figura 6.12). En este caso, el sistema es capaz de detectar la señal de fluorescencia de cada punto de la recta, pero los resultados derivados del apilamiento son distintos a los vistos en los casos anteriores.

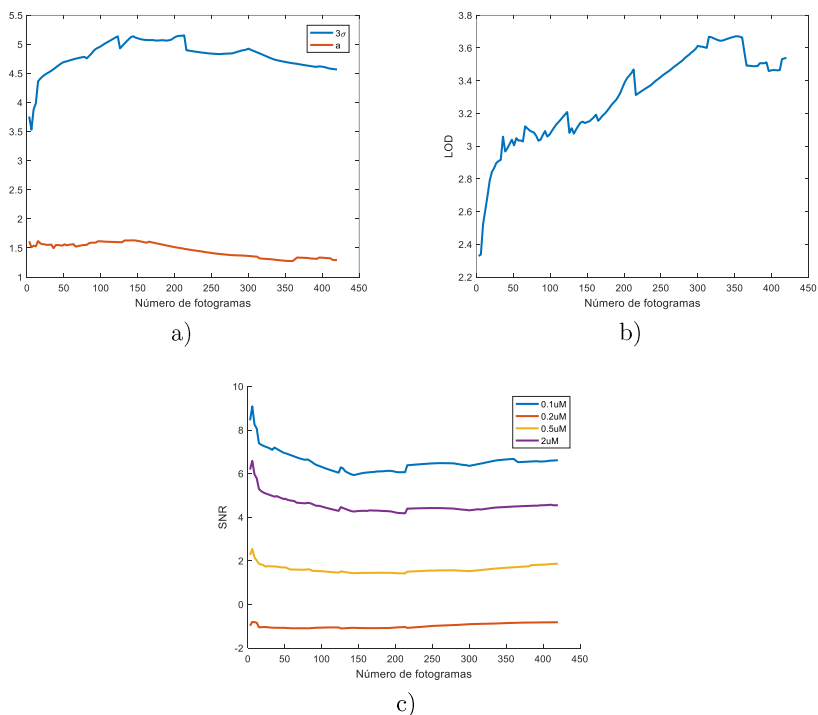


Figura 6.13. Evaluación del apilamiento para ISO 200, a) desviación estándar del papel y pendiente de la recta, b) LOD, c) SNR.

En la Figura 6.13a, se observa cómo la desviación estándar del papel aumenta con el número de fotogramas, con una variación de hasta 1.5, similar a lo ocurrido para ISO 800 y 400. Este resultado deriva en el aumento del

LOD y la disminución en la SNR para cada punto con el número de fotogramas. Estos resultados son contrarios a lo esperado, pero pueden tener una explicación derivada del proceso de detección de la señal. Debido a la baja intensidad de la señal, las variaciones en la desviación estándar del papel son comparables a los valores de intensidad de la señal. Esto provoca que las variaciones en la SNR y LOD tengan una escala mucho mayor que la mostrada en ISO 400 y 800. Por otro lado, el cambio de tendencia de los resultados puede deberse a una mala detección de los puntos. En este caso, los círculos tienen una intensidad muy baja y se detectan con muy pocos píxeles, lo que puede provocar que la detección se vea más influenciada por los bordes negros. Esta influencia puede ocasionar que el emborronamiento de la imagen debida al apilamiento haga que la región detectada contenga cada vez más píxeles 0. Este fenómeno podría estar ocasionando ese aumento de la desviación estándar del papel y el consecuente cambio de tendencia en el LOD y la SNR. Los resultados derivados de la aplicación de este algoritmo plantean la necesidad de mejorar la sensibilidad añadiendo otros enfoques como puede ser óptico o de mejora del rango dinámico de la imagen [128].

6.5 Conclusiones y contribuciones

En este capítulo se ha presentado una aplicación para la detección de códigos QR fluorescentes capaz de detectar conjuntamente un QR fluorescente junto con la señal del sensor analizado. Para mejorar los resultados en la determinación de la concentración de cianuro, se propone repetir los ensayos de la recta de caracterización en el mismo símbolo QR para reducir las variables de error durante los ensayos experimentales. A pesar de esto, se demostró que el sistema es capaz de detectar la señal correctamente sin que la información estática del código QR se viera afectada. Además, se ha desarrollado un sistema de apilamiento de imágenes de μ PAD fluorescentes en Matlab que ha demostrado presentar una mejora en la sensibilidad insuficiente para los requerimientos de la detección de cianuro. Por ello, se propone mejorar el algoritmo mediante una restauración previa de la imagen que aumente su rango dinámico. Algunos resultados de este capítulo fueron incluidos en los siguientes trabajos científicos:

- Comunicación de un póster: Inmaculada Ortiz-Gómez, Antonio Marín-Sánchez, Encarnación Castillo, Diego P. Morales, Luis Fermín Capitán-Vallvey, Alfonso Salinas-Castillo. Fluorogenic smart quick response code for cyanide detection. I Reunión Científica del

Grupo Especializado en Ciencia y Tecnologías (Bio)Analíticas (GCTbA) (2022).

- Comunicación oral: Inmaculada Ortiz-Gómez, Antonio Marín-Sánchez, Encarnación Castillo, Diego P. Morales, Luis Fermín Capitán-Vallvey, Alfonso Salinas-Castillo. Turn-on fluorescence smart quick response code for cyanide detection. 21st International Symposium on Bioluminescence and Chemiluminescence and XIX International Symposium on Luminescence Spectrometry (2022).

Capítulo 7

Conclusiones

La investigación realizada durante esta tesis doctoral consiste en el desarrollo de algoritmos para análisis de imágenes de sensores bioquímicos de respuesta óptica y su aplicación en dispositivos móviles. Las principales conclusiones que se extraen son las que se detallan a continuación:

1. Todas las soluciones aportadas en este trabajo llevaron consigo la implementación de una aplicación móvil en Android. Esto supuso que los algoritmos fueran optimizados para su uso en dispositivos móviles. Esto trae consigo que todas las implementaciones propuestas sean totalmente portables y de bajo coste.
2. El desarrollo de sensores colorimétricos y fluorescentes basados en códigos QR ha sido explorado en profundidad durante todo el trabajo. Todos los detectores de códigos QR eran capaces de mantener la integridad de la información estática almacenada y extraer la información dinámica del sensor químico que contuviera.
3. Las aplicaciones implementadas permitieron el desarrollo e integración de distintos métodos de análisis aplicados a distintos ensayos químicos. Estas implementaciones *software* están estructuradas de forma similar, lo que permite una integración total de distintos tipos de análisis en una sola aplicación móvil, facilitando su usabilidad.
4. Se ha explorado el uso de las herramientas de codificación de errores del código QR como base para los algoritmos de detección de analitos en el dispositivo analítico. Se implementó una aplicación

móvil capaz de discriminar entre patrones de diagnósticos positivos y negativos en base a los errores identificados en el código QR. Se ha testeado el sistema bajo distintas condiciones de luminosidad, discriminando correctamente los distintos patrones, salvo en condiciones muy extremas. Se ha permitido una calibración del umbral de diagnóstico para adaptarlo según el analito analizado. El sistema permite integrar detección multianalito discriminando correctamente entre analitos de distintos colores gracias a su detección en distintos canales RGB.

5. Se ha presentado un sistema de determinación de glucosa y lactato basado en códigos QR colorimétricos. Este sistema permite extraer los valores RGB de las posiciones de los analitos depositadas en el código QR y corregirlas mediante un algoritmo de corrección de la luminosidad. El algoritmo de corrección de luminosidad permite corregir el canal de saturación (S) de los módulos sensores haciendo uso de un conjunto de módulos de referencia. Se ha testeado el sistema con muestras reales de suero y sudor, demostrando resultados de recuperación de entre el 90% y el 100%.
6. La detección de señales fluorimétricas mediante el uso de aplicaciones móviles se ha testeado en profundidad en esta tesis doctoral. Se ha desarrollado un algoritmo para de detección de un μ PAD para determinación fluorimétrica basada en el uso de un valor ratiométrico.
7. La detección de códigos QR fluorescentes ha sido otro punto de interés en esta investigación, presentando una aplicación móvil capaz de detectar QR fluorescentes y extraer la información dinámica de los analitos depositados en éste. Se testeó el sistema para la detección de cianuro en el código QR, demostrando su capacidad para detectar la presencia del analito en el dispositivo. Se ha estudiado el problema de la baja sensibilidad del sistema en condiciones de baja luminosidad. Para tratar de solucionar este problema se ha

trabajado en el desarrollo de un algoritmo de apilamiento de imágenes que por sí solo no atribuye una mejora sustancial en la sensibilidad.

Es importante aprovechar el interés de las aplicaciones móviles para el desarrollo de instrumentación portable para detección de sensores químicos colorimétricos y fluorescentes. La línea que se ha desarrollado en el trabajo de investigación integrando los códigos QR con los μ PAD y sus múltiples formas de detección supone una puerta abierta a seguir investigando y profundizando en este campo. En lo que respecta al trabajo realizado en esta tesis doctoral, algunas soluciones deben ser testeadas en mayor profundidad. El sistema de detección de códigos QR colorimétricos basado en corrección de errores debe testarse con una aplicación química para evaluar su funcionamiento en condiciones reales. En el caso de la detección de códigos QR fluorescentes, es necesario realizar un ensayo experimental con más muestras de cianuro para valorar su reproducibilidad en la determinación de este analito.

En lo que respecta al algoritmo de apilamiento se hace necesario abordar el problema desde otros puntos de vista tratando de mejorar el estado original de la señal mediante algún algoritmo de restauración de la imagen. Una línea realmente interesante es la aplicación de referencias de color fijas independientes del analito depositado en el código QR para realizar la calibración de la luminosidad con referencias estandarizadas y universales. Uno de los puntos más importantes de esta investigación fue mantener la facilidad de uso de la aplicación y dotarla de funcionalidades que pudieran ser útiles en un contexto de uso real. En esa línea se buscaría realizar una integración de los distintos tipos de análisis en la misma aplicación y dotar a ésta de una arquitectura en la nube para poder recibir actualizaciones frecuentes. El objetivo final al que se quiere llegar es desarrollar un producto fácil de usar, fiable y de bajo coste que pueda integrar distintos tipos de funcionalidades con un buen rendimiento en un contexto real.

Referencias

- [1] A. Al Wahaibi, R. Al Shehi, A. Al Maani, S. Al-Abri, Out-of-hospital mortality as an indicator of health care system saturation: Two waves retrospective analysis, *Travel Med. Infect. Dis.* 43 (2021) 102121. <https://doi.org/10.1016/j.tmaid.2021.102121>.
- [2] I. Hernández-Neuta, F. Neumann, J. Brightmeyer, T. Ba Tis, N. Madaboosi, Q. Wei, A. Ozcan, M. Nilsson, Smartphone-based clinical diagnostics: towards democratization of evidence-based health care, *J. Intern. Med.* 285 (2019) 19–39. <https://doi.org/10.1111/joim.12820>.
- [3] K. Yang, H. Peretz-Soroka, Y. Liu, F. Lin, Novel developments in mobile sensing based on the integration of microfluidic devices and smartphones, *Lab Chip.* 16 (2016) 943–958. <https://doi.org/10.1039/C5LC01524C>.
- [4] G. Rateni, P. Dario, F. Cavallo, Smartphone-Based Food Diagnostic Technologies: A Review, *Sensors.* 17 (2017) 1453. <https://doi.org/10.3390/s17061453>.
- [5] H. Yousefi, H.M. Su, S.M. Imani, K. Alkhalidi, C.D. Filipe, T.F. Didar, Intelligent Food Packaging: A Review of Smart Sensing Technologies for Monitoring Food Quality, *ACS Sensors.* 4 (2019) 808–821. <https://doi.org/10.1021/acssensors.9b00440>.
- [6] A.P.F. Turner, Biosensors: Sense and sensibility, *Chem. Soc. Rev.* 42 (2013) 3184–3196. <https://doi.org/10.1039/c3cs35528d>.
- [7] C.M. Pandey, S. Augustine, S. Kumar, S. Kumar, S. Nara, S. Srivastava, B.D. Malhotra, Microfluidics Based Point-of-Care Diagnostics, *Biotechnolohy J.* 13 (2018). <https://doi.org/10.1002/biot.201700047>.
- [8] A.K. Yetisen, M.S. Akram, C.R. Lowe, Paper-based microfluidic point-of-care diagnostic devices, *Lab Chip.* 13 (2013) 2210–2251. <https://doi.org/10.1039/c3lc50169h>.
- [9] D. Quesada-González, A. Merkoçi, Mobile phone-based biosensing: An emerging “diagnostic and communication” technology, *Biosens. Bioelectron.* 92 (2017) 549–562. <https://doi.org/10.1016/J.BIOS.2016.10.062>.
- [10] A. Roda, E. Micheleni, M. Zangheri, M. Di Fusco, D. Calabria, P. Simoni, Smartphone-based biosensors: A critical review and perspectives, *TrAC Trends Anal. Chem.* 79 (2016) 317–325. <https://doi.org/10.1016/J.TRAC.2015.10.019>.
- [11] S. Kanchi, M.I. Sabela, P.S. Mdluli, Inamuddin, K. Bisetty,

- Smartphone based bioanalytical and diagnosis applications: A review, *Biosens. Bioelectron.* 102 (2018) 136–149. <https://doi.org/10.1016/j.bios.2017.11.021>.
- [12] L.F. Capitán-Vallvey, N. López-Ruiz, A. Martínez-Olmos, M.M. Erenas, A.J. Palma, Recent developments in computer vision-based analytical chemistry: A tutorial review, *Anal. Chim. Acta.* 899 (2015) 23–56. <https://doi.org/10.1016/j.aca.2015.10.009>.
- [13] M.M. Calabretta, A. Lopreside, L. Montali, M. Zangheri, L. Evangelisti, M. D’Elia, E. Micheli, Portable light detectors for bioluminescence biosensing applications: A comprehensive review from the analytical chemist’s perspective, *Anal. Chim. Acta.* 1200 (2022) 339583. <https://doi.org/10.1016/j.aca.2022.339583>.
- [14] H.J. Chun, Y. Park, Y.D. Han, Y. Jang, H. Yoon, Paper-based glucose biosensing system utilizing a smartphone as a signal reader, *Biochip J.* 8 (2014) 218–226. <https://doi.org/10.1007/s13206-014-8308-7>.
- [15] V. Oncescu, D. O’Dell, D. Erickson, Smartphone based health accessory for colorimetric detection of biomarkers in sweat and saliva, *Lab Chip.* 13 (2013) 3232–3238. <https://doi.org/10.1039/c3lc50431j>.
- [16] N.K. Thom, G.G. Lewis, K. Yeung, S.T. Phillips, Quantitative fluorescence assays using a self-powered paper-based microfluidic device and a camera-equipped cellular phone, *{RSC} Adv.* 4 (2014) 1334–1340. <https://doi.org/10.1039/C3RA44717K>.
- [17] V.K. Rajendran, P. Bakthavathsalam, P.L. Bergquist, A. Sunna, Smartphone detection of antibiotic resistance using convective PCR and a lateral flow assay, *Sensors Actuators B Chem.* (2019). <https://doi.org/10.1016/j.snb.2019.126849>.
- [18] M. Xiao, Z. Liu, N. Xu, L. Jiang, M. Yang, C. Yi, A Smartphone-Based Sensing System for On-Site Quantitation of Multiple Heavy Metal Ions Using Fluorescent Carbon Nanodots-Based Microarrays, *ACS Sensors.* 5 (2020) 870–878. <https://doi.org/10.1021/acssensors.0c00219>.
- [19] M. Zangheri, L. Cevenini, L. Anfossi, C. Baggiani, P. Simoni, F. Di Nardo, A. Roda, A simple and compact smartphone accessory for quantitative chemiluminescence-based lateral flow immunoassay for salivary cortisol detection, *Biosens. Bioelectron.* 64 (2015) 63–68. <https://doi.org/10.1016/j.bios.2014.08.048>.
- [20] K. Wang, J. Yang, H. Xu, B. Cao, Q. Qin, X. Liao, Y. Wo, Q. Jin, D. Cui, Smartphone-imaged multilayered paper-based analytical device for colorimetric analysis of carcinoembryonic antigen, *Anal. Bioanal.*

- Chem. 412 (2020) 2517–2528. <https://doi.org/10.1007/s00216-020-02475-1>.
- [21] R. Moonsoo, M. Mannan Saeed, L. Chawei, H. Sehui, J. Chansung, K. Whoi-Yul, Smartphone-Based Point-of-Care Urinalysis under Variable Illumination, *IEEE J. Transl. Eng. Heal. Med.* 6 (2018) 2800111. <https://doi.org/http://dx.doi.org/10.1109/JTEHM.2017.2765631>.
- [22] J. Il Hong, B.-Y. Chang, Development of the smartphone-based colorimetry for multi-analyte sensing arrays, *Lab Chip.* 14 (2014) 1725–1732. <https://doi.org/10.1039/c3lc51451j>.
- [23] A.K. Yetisen, J.L. Martinez-Hurtado, A. Garcia-Melendrez, F. da Cruz Vasconcellos, C.R. Lowe, A smartphone algorithm with inter-phone repeatability for the analysis of colorimetric tests, *Sensors Actuators B Chem.* 196 (2014) 156–160. <https://doi.org/10.1016/j.snb.2014.01.077>.
- [24] J. Choi, A.J. Bandodkar, J.T. Reeder, T.R. Ray, A. Turnquist, S.B. Kim, N. Nyberg, A. Hourlier-Fargette, J.B. Model, A.J. Aranyosi, S. Xu, R. Ghaffari, J.A. Rogers, Soft, skin-integrated multifunctional microfluidic systems for accurate colorimetric analysis of sweat biomarkers and temperature, *ACS Sensors.* 4 (2019) 379–388. <https://doi.org/10.1021/acssensors.8b01218>.
- [25] N. Lopez-Ruiz, V.F. Curto, M.M. Erenas, F. Benito-Lopez, D. Diamond, A.J. Palma, L.F. Capitán-Vallvey, Smartphone-based simultaneous pH and nitrite colorimetric determination for paper microfluidic devices, *Anal. Chem.* 86 (2014) 9554–9562. <https://doi.org/10.1021/ac5019205>.
- [26] M. Yuan, Q. Jiang, K.-K. Liu, S. Singamaneni, S. Chakrabartty, Towards an Integrated {QR} Code Biosensor: Light-Driven Sample Acquisition and Bacterial Cellulose Paper Substrate, *{IEEE} Trans. Biomed. Circuits Syst.* 12 (2018) 452–460. <https://doi.org/10.1109/tbcas.2018.2801566>.
- [27] T.F. Scherr, S. Gupta, D.W. Wright, F.R. Haselton, An embedded barcode for “connected” malaria rapid diagnostic tests, *Lab Chip.* 17 (2017) 1314–1322. <https://doi.org/10.1039/c6lc01580h>.
- [28] J.A. M. Conrado, R. Sequinel, B.C. Dias, M. Silvestre, A.D. Batista, J.F. da S. Petrucci, Chemical QR Code: A simple and disposable paper-based optoelectronic nose for the identification of olive oil odor, *Food Chem.* 350 (2021) 129243. <https://doi.org/10.1016/j.foodchem.2021.129243>.

- [29] L. Engel, I. Benito-Altamirano, K.R. Tarantik, C. Pannek, M. Dold, J.D. Prades, J. Wöllenstein, Printed sensor labels for colorimetric detection of ammonia, formaldehyde and hydrogen sulfide from the ambient air, *Sensors Actuators, B Chem.* 330 (2021). <https://doi.org/10.1016/j.snb.2020.129281>.
- [30] R.S. Berns, *Principles of Color Technology*, Wiley, 2000. <https://learning.oreilly.com/library/view/billmeyer-and-saltzmans/9781119367222/c10.xhtml#head-2-126>.
- [31] P. Kay, R.S. Cook, World Color Survey, *Encycl. Color Sci. Technol.* (2015) 1–8. <https://doi.org/10.1007/978-3-642-27851-8>.
- [32] R.G. Kuehni, A. Schwarz, *Color Ordered: A Survey of Color Order Systems from Antiquity to the Present*, Oxford University Press, 2008. <https://doi.org/https://doi.org/10.1093/acprof:oso/9780195189681.01.0001>.
- [33] R.W.G. Hunt, M.R. Pointer, *Measuring color*, John Wiley & Sons, Incorporated, 2011. <https://ebookcentral.proquest.com/lib/ugr/reader.action?docID=693754>.
- [34] J. Guild, The colorimetric properties of the spectrum, *Philos. Trans. R. Soc. London.* 230 (1931) 149–187. <https://doi.org/https://doi.org/10.1098/rsta.1932.0005>.
- [35] W.D. Wright, A re-determination of the mixture curves of the spectrum, *Trans. Opt. Soc.* 31 (1930) 201–218. <https://doi.org/10.1088/1475-4878/31/4/303>.
- [36] G. Wyszecki, W.S. Stiles, *Color Science*, 2nd editio, Wiley, New York, 1982.
- [37] Y.L. Shishkin, S.G. Dmitrienko, O.M. Medvedeva, S.A. Badakova, L.N. Pyatkova, Use of a scanner and digital image-processing software for the quantification of adsorbed substances, *J. Anal. Chem.* 59 (2004) 102–106. <https://doi.org/10.1023/B:JANC.0000014733.32082.4b>.
- [38] A. Abbaspour, M.A. Mehrgardi, A. Noori, M.A. Kamyabi, A. Khalafi-Nezhad, M.N.S. Rad, Speciation of iron(II), iron(III) and full-range pH monitoring using paptode: A simple colorimetric method as an appropriate alternative for optodes, *Sensors Actuators, B Chem.* 113 (2006) 857–865. <https://doi.org/10.1016/j.snb.2005.03.119>.
- [39] P. Jongwon, H. Wonhak, K. Chang-Soo, Color intensity method for hydrogel oxygen sensor array, *IEEE Sens. J.* 10 (2010) 1855–1862. <https://doi.org/10.1109/JSEN.2010.2049837>.
- [40] C. Poynton, *Frequently Asked Questions about Color*, (2020). <https://doi.org/10.1201/9781003108887-9>.

- [41] T. Bishop, G. Lee, TAOS Colorimetry Tutorial “The Science of Color,” 2006.
- [42] SharkD, RGB color solid cube, Wikipedia. (n.d.). https://commons.wikimedia.org/wiki/File:RGB_color_solid_cube.png (accessed January 15, 2024).
- [43] BenRG, CIE 1931 xy color space, Wikipedia. (n.d.). https://commons.wikimedia.org/wiki/File:CIE1931xy_blank.svg (accessed January 15, 2024).
- [44] SharkD, HSV color solid cylinder, Wikipedia. (n.d.). https://es.m.wikipedia.org/wiki/Archivo:HSV_color_solid_cylinder.png (accessed January 15, 2024).
- [45] R. Jacob, Munsell color system, Wikipedia. (n.d.). <https://commons.wikimedia.org/wiki/File:Munsell-system.svg> (accessed January 15, 2024).
- [46] K. Cantrell, M.M. Erenas, I. De Orbe-Payá, L.F. Capitán-Vallvey, Use of the hue parameter of the hue, saturation, value color space as a quantitative analytical parameter for bitonal optical sensors, *Anal. Chem.* 82 (2010) 531–542. <https://doi.org/10.1021/ac901753c>.
- [47] A.R. Smith, Color gamut transform pairs, *Proc. 5th Annu. Conf. Comput. Graph. Interact. Tech. SIGGRAPH 1978.* (1978) 12–19. <https://doi.org/10.1145/800248.807361>.
- [48] K.E. Burchett, The early development of the Munsell system, *Color Res. Appl.* 27 (2002) 20–27. <https://doi.org/10.1002/col.10002>.
- [49] S. Cotton, Colour, colour spaces and the human visual system, *Univ. Birmingham, Sch. Comput. Sci. ...* (1995). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.7962&rep=rep1&type=pdf>.
- [50] T.E. Madden, E.J. Giorgianni, *Digital Color Management: Encoding Solutions: Second Edition*, 2010. <https://doi.org/10.1002/9780470994375>.
- [51] Y.-H. Shin, M. Teresa Gutierrez-Wing, J.-W. Choi, Review—Recent Progress in Portable Fluorescence Sensors, *J. Electrochem. Soc.* 168 (2021) 017502. <https://doi.org/10.1149/1945-7111/abd494>.
- [52] A. Einstein, Concerning an heuristic point of view toward the emission and transformation of light, *Am. J. Phys.* 33 (1905) 1–16.
- [53] L.F. Capitán-Vallvey, A.J. Palma, Recent developments in handheld and portable optosensing—A review, *Anal. Chim. Acta.* 696 (2011) 27–46. <https://doi.org/10.1016/J.ACA.2011.04.005>.
- [54] V.B.V. MacIel, C.M.P. Yoshida, T.T. Franco, Development of a

- prototype of a colourimetric temperature indicator for monitoring food quality, *J. Food Eng.* 111 (2012) 21–27. <https://doi.org/10.1016/j.jfoodeng.2012.01.037>.
- [55] T. Kuroda, *Essential principles of image sensors*, 1st ed., CRC Press, 2015. <https://doi.org/https://doi.org/10.1201/b17411>.
- [56] D. Litwiller, *CCD vs. CMOS: Facts and fiction*, *Photonics Spectra*. 35 (2001) 154–158.
- [57] Dat-Science, *Cómo funciona un escáner*, (n.d.). <https://dat-science.com/como-funciona-el-escaner/> (accessed January 15, 2024).
- [58] R.W.. Hunt, *The Reproduction of color*, 6th Editio, John Wiley & Sons, Ltd, 1995. <https://doi.org/10.5040/9781501335402.ch-029>.
- [59] P. Jackman, D.W. Sun, G. ElMasry, Robust colour calibration of an imaging system using a colour space transform and advanced regression modelling, *Meat Sci.* 91 (2012) 402–407. <https://doi.org/10.1016/j.meatsci.2012.02.014>.
- [60] M.A. Barbero-Álvarez, J.M. Menéndez, J.A. Rodrigo, An adaptive colour calibration for crowdsourced images in heritage preservation science, *IEEE Access.* 8 (2020) 185093–185111. <https://doi.org/10.1109/ACCESS.2020.3017390>.
- [61] R. Szeliski, *computer vision: algorithms and applications*, Springer, 2010.
- [62] P. Dempsey, *TheTeardown Apple iPhone Pro I3 smartphone*, (2021) 68–69.
- [63] P. Dempsey, *TheTeardown Samsung Galaxi S21 Ultra smartphone*, (2021) 70–71.
- [64] P. Dempsey, *TheTeardown Google Pixel 3a smartphone*, (2019) 82–83.
- [65] Omnivision, *PureCel Plus*, (2016). <https://www.ovt.com/technologies/purecel-plus/> (accessed May 17, 2023).
- [66] Samsung, *ISOCELL HP2*, (2023). <https://semiconductor.samsung.com/image-sensor/mobile-image-sensor/isocell-hp2/> (accessed May 17, 2023).
- [67] Sony, *Semiconductors & Image Sensor*, (2020). <https://www.sony.com/en/SonyInfo/CorporateInfo/History/sonyhistory-o.html> (accessed May 17, 2023).
- [68] A. Roda, E. Micheli, M. Zangheri, M. Di Fusco, D. Calabria, P. Simoni, Smartphone-based biosensors: A critical review and perspectives, *TrAC - Trends Anal. Chem.* 79 (2016) 317–325. <https://doi.org/10.1016/j.trac.2015.10.019>.

- [69] R.C. Gonzalez, R.E. Woods, Digital Image Processing, 3rd ed., Pearson Prentice Hall, New Jersey, 2008.
- [70] Y. Soda, E. Bakker, Quantification of Colorimetric Data for Paper-Based Analytical Devices, *ACS Sensors*. 4 (2019) 3093–3101. <https://doi.org/10.1021/acssensors.9b01802>.
- [71] M. Affi, M.A. Brubaker, M.S. Brown, Auto White-Balance Correction for Mixed-Illuminant Scenes, *Proc. - 2022 IEEE/CVF Winter Conf. Appl. Comput. Vision, WACV 2022*. (2022) 934–943. <https://doi.org/10.1109/WACV51458.2022.00101>.
- [72] M. Farghaly, R.F. Mansour, A.A. Sewisy, Two-stage deep learning framework for sRGB image white balance, *Signal, Image Video Process*. 17 (2023) 277–284. <https://doi.org/10.1007/s11760-022-02230-2>.
- [73] D.S. Taubman, M.W. Marcellin, JPEG2000: Standard for interactive imaging, *Proc. IEEE*. 90 (2002) 1336–1357. <https://doi.org/10.1109/JPROC.2002.800725>.
- [74] S.M. Russell, A. Doménech-Sánchez, R. de la Rica, Augmented Reality for Real-Time Detection and Interpretation of Colorimetric Signals Generated by Paper-Based Biosensors, *{ACS} Sensors*. 2 (2017) 848–853. <https://doi.org/10.1021/acssensors.7b00259>.
- [75] T. Wang, G. Xu, W. Wu, X. Wang, X. Chen, S. Zhou, F. You, A novel combination of quick response code and microfluidic paper-based analytical devices for rapid and quantitative detection, *Biomed. Microdevices*. 20 (2018) 1–8. <https://doi.org/10.1007/s10544-018-0325-1>.
- [76] N. Gligoric, S. Krco, L. Hakola, K. Vehmas, S. De, K. Moessner, K. Jansson, I. Polenz, R. Van Kranenburg, SmartTags: IoT Product Passport for Circular Economy Based on Printed Sensors and Unique Item-Level Identifiers, *Sensors (Basel)*. 19 (2019). <https://doi.org/10.3390/s19030586>.
- [77] Denso-Wave, What is a 2D code?, (2023). <https://www.denso-wave.com/en/adcd/fundamental/2dcode/2dcode/> (accessed January 4, 2024).
- [78] Denso-Wave, Information capacity and versions QR code, (2024). <https://www.qrcode.com/en/about/version.html> (accessed January 4, 2024).
- [79] Denso-Wave, What is a QR code?, (2024). <https://www.qrcode.com/en/about/> (accessed January 4, 2024).
- [80] S. Tiwari, An Introduction to QR Code Technology, in: 2016 Int. Conf.

- Inf. Technol., IEEE, 2016: pp. 39–44. <https://doi.org/10.1109/ICIT.2016.021>.
- [81] Wikipedia, QR code, (2024). https://en.wikipedia.org/wiki/QR_code (accessed January 4, 2024).
- [82] C.K.P. Clarke, Reed-Solomon error correction, 2002.
- [83] R.C. Bose, D.K. Ray-Chaudhuri, On a class of error: Correcting binary group codes, *Inf. Control.* 3 (1960) 68–79. <https://repository.lib.ncsu.edu/items/e51ec8ca-6490-4cf2-89be-d4784707e58e>.
- [84] A. Hocquenghem, Codes correcteurs d’erreurs, *Chiffres.* 2 (1959).
- [85] E.R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, 2015.
- [86] M. Purser, An introduction to error-correcting codes, London, 1995. <https://doi.org/10.1201/b14977-8>.
- [87] R. Chien, Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes, *IEEE Trans. Inf. Theory.* IT-10 (1964) 357–363. <https://www.semanticscholar.org/paper/Cyclic-decoding-procedures-for-Bose-codes-Chien/d813ccedb50668b843f5ae9a28ac52b4e0139016>.
- [88] S. Owen, Zxing Project, (2014). <https://github.com/zxing/zxing> (accessed December 9, 2020).
- [89] W.G. Melero, Sistema de reconocimiento de cambios en códigos QR, 2017. <http://ddd.uab.cat/pub/landes/11394218v7p196.pdf>.
- [90] M. Yuan, K. Liu, S. Singamaneni, S. Chakrabarty, Self-Powered Forward Error-Correcting Biosensor Based on Integration of Paper-Based Microfluidics and Self-Assembled Quick Response Codes, *{IEEE} Trans. Biomed. Circuits Syst.* 10 (2016) 963–971. <https://doi.org/10.1109/tbcas.2016.2580156>.
- [91] A.Q. Xie, J. Guo, L. Zhu, S. Chen, Carbon dots promoted photonic crystal for optical information storage and sensing, *Chem. Eng. J.* 415 (2021) 128950. <https://doi.org/10.1016/j.cej.2021.128950>.
- [92] S. Han, H.J. Bae, J. Kim, S. Shin, S.-E. Choi, S.H. Lee, S. Kwon, W. Park, Lithographically Encoded Polymer Microtaggant Using High-Capacity and Error-Correctable QR Code for Anti-Counterfeiting of Drugs, *Adv. Mater.* 24 (2012) 5924–5929. <https://doi.org/10.1002/adma.201201486>.
- [93] M. Edinger, D. Bar-Shalom, N. Sandler, J. Rantanen, N. Genina, QR encoded smart oral dosage forms by inkjet printing, *Int. J. Pharm.* 536 (2018) 138–145. <https://doi.org/10.1016/j.ijpharm.2017.11.052>.
- [94] A. Katoh, K. Maejima, Y. Hiruta, D. Citterio, All-printed semiquantitative paper-based analytical devices relying on QR code array readout, *Analyst.* 145 (2020) 6071.

- <https://doi.org/10.1039/d0an00955e>.
- [95] A. Burklund, H.K. Saturley-Hall, F.A. Franchina, J.E. Hill, J.X.J. Zhang, Printable QR code paper microfluidic colorimetric assay for screening volatile biomarkers, *Biosens. Bioelectron.* 128 (2019) 97–103. <https://doi.org/10.1016/j.bios.2018.12.026>.
- [96] J.F.C.B. Ramalho, S.F.H. Correia, L. Fu, L.M.S. Dias, P. Adão, P. Mateus, R.A.S. Ferreira, P.S. André, Super modules-based active QR codes for smart trackability and IoT: a responsive-banknotes case study, *Npj Flex. Electron.* 4 (2020) 1–9. <https://doi.org/10.1038/s41528-020-0073-1>.
- [97] B. Phillips, B. Hardy, *Android Programming: The Big Nerd Ranch Guide*, 2017. <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Android+Programming+:+The+Big+Nerd+Ranch+Guide#0>.
- [98] Xiaomi, Pocophone F1, (2020). <https://www.mi.com/es/pocophone-fl/specs/> (accessed April 26, 2020).
- [99] Xiaomi, Redmi A1, (2023). <https://www.mi.com/es/product/redmi-a1/> (accessed December 1, 2023).
- [100] Canon, EOS 70D, (2023). https://www.canon.es/for_home/product_finder/cameras/digital_slr/eos_70d/ (accessed December 1, 2023).
- [101] Duclus, Photo Light Box, (2023). <https://www.duclus.com/products/duclus-mini-photo-studio-light-box-photo-shooting-tent-kit-portable-folding-photography-light-tent-kit-with-40pcs-led-light-6-kinds-color-backgrounds-for-small-size-products> (accessed December 1, 2023).
- [102] EPSON, Epson XP-2200, (2023). https://www.epson.es/es_ES/productos/impresoras/inkjet/consumo/expression-xp-2200/p/34101?cid=sh:google:P916-es:111::&cid=ps:google:gen23-es:111::&aff_code=EUPSG100171&gclid=CjwKCAiApaarBhB7EiwAYiMwqiVsPOPm0jLCtCYHkTNr6xA-khfUuUwTcreUjIQ-r82TTD998QyYRoCB (accessed December 1, 2023).
- [103] M.-Y. Jia, Q.-S. Wu, H. Li, Y. Zhang, Y.-F. Guan, L. Feng, The calibration of cellphone camera-based colorimetric sensor array and its application in the determination of glucose in urine, *Biosens. Bioelectron.* 74 (2015) 1029–1037. <https://doi.org/10.1016/j.bios.2015.07.072>.
- [104] J.L.D. Nelis, L. Bura, Y. Zhao, K.M. Burkin, K. Rafferty, C.T. Elliott, K. Campbell, The efficiency of color space channels to quantify color and color intensity change in liquids, pH strips, and lateral flow assays

- with smartphones, *Sensors*. 19 (2019). <https://doi.org/10.3390/s19235104>.
- [105] Y. Chen, G. Fu, Y. Zilberman, W. Ruan, S.K. Ameri, Y.S. Zhang, E. Miller, S.R. Sonkusale, Low cost smart phone diagnostics for food using paper-based colorimetric sensor arrays, *Food Control*. 82 (2017) 227–232. <https://doi.org/10.1016/j.foodcont.2017.07.003>.
- [106] A.Y. Mutlu, V. Kılıç, G.K. Özdemir, A. Bayram, N. Horzum, M.E. Solmaz, Smartphone-based colorimetric detection via machine learning, *Analyst*. 142 (2017) 2434–2441. <https://doi.org/10.1039/C7AN00741H>.
- [107] M.E. Solmaz, A.Y. Mutlu, G. Alankus, V. Kılıç, A. Bayram, N. Horzum, Quantifying colorimetric tests using a smartphone app based on machine learning classifiers, *Sensors Actuators B Chem*. 255 (2018) 1967–1973. <https://doi.org/10.1016/j.snb.2017.08.220>.
- [108] A. Marín-Sánchez, I. Ortiz-Gómez, D. Gallego-Méndez, C. Rodríguez-Domínguez, L. Fermín Capitán-Vallvey, D. P. Morales, J. Antonio Álvarez-Bermejo, E. Castillo, A. Salinas-Castillo, Smart quick response code for multianalyte determination, *Meas. J. Int. Meas. Confed*. 204 (2022). <https://doi.org/10.1016/j.measurement.2022.112035>.
- [109] M. Ariza-Avidad, A. Salinas-Castillo, M.P. Cueílar, M. Agudo-Acemel, M.C. Pegalajar, L.F. Capitán-Vallvey, G. Spain, Printed Disposable Colorimetric Array for Metal Ion Discrimination, *Anal. Chem*. 86 (2014) 16. <https://doi.org/10.1021/ac501670f>.
- [110] Huawei, Huawei, (2020). <https://consumer.huawei.com/es/support/phones/p-smart/> (accessed April 24, 2020).
- [111] M. Ariza-Avidad, A. Salinas-Castillo, L.F. Capitán-Vallvey, A 3D μ PAD based on a multi-enzyme organic-inorganic hybrid nanoflower reactor, *Biosens. Bioelectron*. 77 (2016) 51–55. <https://doi.org/10.1016/j.bios.2015.09.012>.
- [112] I. Ortiz-Gómez, A. Salinas-Castillo, A.G. García, J.A. Álvarez-Bermejo, I. de Orbe-Payá, A. Rodríguez-Diéguez, L.F. Capitán-Vallvey, Microfluidic paper-based device for colorimetric determination of glucose based on a metal-organic framework acting as peroxidase mimetic, *Microchim. Acta*. 185 (2018). <https://doi.org/10.1007/s00604-017-2575-7>.
- [113] C. Callewaert, B. Buyschaert, E. Vossen, V. Fievez, T. Van de Wiele, N. Boon, Artificial sweat composition to grow and sustain a mixed human axillary microbiome, *J. Microbiol. Methods*. 103 (2014) 6–8.

- <https://doi.org/10.1016/j.mimet.2014.05.005>.
- [114] A. Ortiz-Gomez, Inmaculada Ortega-Muñoz, Mariano Marín-Sánchez, Antonio de Orbe-Payá, Ignacio Hernandez-Mateo, Fernando Capitan-Vallvey, Luis Fermin Santoyo-Gonzalez, Francisco Salinas-Castillo, A vinyl sulfone clicked carbon dot-engineered microfluidic paper-based analytical device for fluorometric determination of biothiols, *Microchim. Acta.* 187 (2020) 421. <https://doi.org/10.1007/s00604-020-04382-9>.
- [115] Sigma-Aldrich, Sigma-Aldrich, (2023). <https://www.sigmaaldrich.com> (accessed December 22, 2023).
- [116] M. Science, Transilluminator UV, (2023). <https://www.majorsci.com/ec99/rwd1661/product.asp?prodid=A60> (accessed December 22, 2023).
- [117] Rayjet, Grabadora laser, (n.d.). www.rayjetlaser.com (accessed January 29, 2024).
- [118] Formlabs, Form 3, (n.d.). <https://formlabs.com/es/> (accessed January 29, 2024).
- [119] J.L.D. Nelis, A.S. Tsagkaris, M.J. Dillon, J. Hajslova, C.T. Elliott, Smartphone-based optical assays in the food safety field, *TrAC - Trends Anal. Chem.* 129 (2020) 115934. <https://doi.org/10.1016/j.trac.2020.115934>.
- [120] S. Erdemir, S. Malkondu, On-site and low-cost detection of cyanide by simple colorimetric and fluorogenic sensors: Smartphone and test strip applications, *Talanta.* 207 (2020) 120278. <https://doi.org/10.1016/j.talanta.2019.120278>.
- [121] P.L. Breuer, C.A. Sutcliffe, R.L. Meakin, Cyanide measurement by silver nitrate titration: Comparison of rhodanine and potentiometric end-points, *Hydrometallurgy.* 106 (2011) 135–140. <https://doi.org/10.1016/j.hydromet.2010.12.008>.
- [122] K.K.W. Mak, H. Yanase, R. Renneberg, Cyanide fishing and cyanide detection in coral reef fish using chemical tests and biosensors, *Biosens. Bioelectron.* 20 (2005) 2581–2593. <https://doi.org/10.1016/j.bios.2004.09.015>.
- [123] G. Ding, T. Lin, R. Zhou, Y. Dong, J. Xu, X. Lu, Electrofluorochromic Detection of Cyanide Anions Using a Nanoporous Polymer Electrode and the Detection Mechanism, *Chem. - A Eur. J.* 20 (2014) 13226–13233. <https://doi.org/10.1002/chem.201403133>.
- [124] Y. Hu, X. Lu, X. Jiang, P. Wu, Carbon dots and AuNCs co-doped electrospun membranes for ratiometric fluorescent determination of

- cyanide, *J. Hazard. Mater.* 384 (2020) 121368. <https://doi.org/https://doi.org/10.1016/j.jhazmat.2019.121368>.
- [125] C.C. Tseng, C. Te Kung, R.F. Chen, M.H. Tsai, H.R. Chao, Y.N. Wang, L.M. Fu, Recent advances in microfluidic paper-based assay devices for diagnosis of human diseases using saliva, tears and sweat samples, *Sensors Actuators, B Chem.* 342 (2021) 130078. <https://doi.org/10.1016/j.snb.2021.130078>.
- [126] R. Rasooly, H. Bruck, J. Balsam, B. Prickril, M. Ossandon, A. Rasooly, Improving the Sensitivity and Functionality of Mobile Webcam-Based Fluorescence Detectors for Point-of-Care Diagnostics in Global Health, *Diagnostics*. 6 (2016) 19. <https://doi.org/10.3390/diagnostics6020019>.
- [127] J. Balsam, H.A. Bruck, Y. Kostov, A. Rasooly, Image stacking approach to increase sensitivity of fluorescence detection using a low cost complementary metal-oxide-semiconductor (CMOS) webcam, *Sensors Actuators, B Chem.* 171–172 (2012) 141–147. <https://doi.org/10.1016/j.snb.2012.02.003>.
- [128] J. Balsam, R. Rasooly, H.A. Bruck, A. Rasooly, Thousand-fold fluorescent signal amplification for mHealth diagnostics, *Biosens. Bioelectron.* 51 (2014) 1–7. <https://doi.org/10.1016/j.bios.2013.06.053>.
- [129] EPSON, Epson Corporation, (2023). <https://www.epson.es/> (accessed December 22, 2023).
- [130] Agilent, Fluorescence Spectrofotometer, (2023). <https://www.agilent.com/en/product/molecular-spectroscopy/fluorescence-spectroscopy/fluorescence-systems/cary-eclipse-fluorescence-spectrophotometer> (accessed December 22, 2023).
- [131] C. Instruments, Crison Instruments, (2023). <https://www.crisoninstruments.com/site/index.html> (accessed December 22, 2023).
- [132] Epson, Epson ME-10, (2023). https://www.epson.com.ph/ME-Series/Epson-Expression-ME-10/s/SPT_C11CC22401.
- [133] Instrumentación Científica Técnica, Visco Basic Plus, (2023). <https://www.ictsl.net/productos/01d63694a80f7db0d/01d63695130981d28.html> (accessed December 22, 2023).
- [134] Biolin Scientific, Optical Tensiometer, (2023). https://www.biolinscientific.com/attension/optical-tensiometers/theta-lite?utm_term=theta+lite+optical+tensiometer&utm_campaign=BR-Optical+Tensiometers-G-World&utm_source=adwords&utm_medium=ppc&hsa_acc=1094516957&hsa_cam=1397419159&hsa_grp=55227851236&hsa (accessed

- December 22, 2023).
- [135] Thermo Fisher Scientific, Thermo Fisher Scientific, (2023). <https://www.thermofisher.com/es/es/home/electron-microscopy/products/transmission-electron-microscopes.html> (accessed December 22, 2023).
- [136] Jenck, Kratos, Axis Ultra DLD, (2023). <https://www.jenck.com/productos/producto/axis-ultra-dld> (accessed December 22, 2023).
- [137] OpenCV Team, OpenCV, (2023). <https://opencv.org/> (accessed December 22, 2023).
- [138] M. Hou, H. Zhao, Y. Feng, J. Ge, Synthesis of patterned enzyme–metal–organic framework composites by ink-jet printing, *Bioresour. Bioprocess.* 4 (2017). <https://doi.org/10.1186/s40643-017-0171-7>.
- [139] X. Wu, L. Wu, X. Cao, Y. Li, A. Liu, S. Liu, Nitrogen-doped carbon quantum dots for fluorescence detection of Cu²⁺ and electrochemical monitoring of bisphenol A, *RSC Adv.* 8 (2018) 20000–20006. <https://doi.org/10.1039/c8ra03180k>.