



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DESARROLLO DE UN SISTEMA PARA LA IDENTIFICACIÓN DE PERSONAS
MEDIANTE ASIGNACIÓN Y USO DE CÓDIGO QR POR PERFIL

Autor

Wolfgang Andrés Darquea Berger

Año
2017



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DESARROLLO DE UN SISTEMA PARA LA IDENTIFICACIÓN DE PERSONAS
MEDIANTE ASIGNACIÓN Y USO DE CÓDIGO QR POR PERFIL

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Sistemas de Computación
e Informática

Profesor Guía

Msc. Pedro Manuel Nogales Cobas

Autor

Wolfgang Andrés Darquea Berger

Año

2017

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

.....

Pedro Manuel Nogales Cobas

Máster en Ciencias

CC: 175676028-4

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

.....
Santiago Ramiro Villarreal Narváez
Máster en Ciencias
CC: 171398007-4

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

.....
Wolfgang Andrés Darquea Berger

CC: 200007945-5

AGRADECIMIENTOS

Agradezco a Dios, mis padres y mi esposa por ser los pilares en mi vida, por enseñarme el camino a seguir y confiar en mis decisiones, por creer en mis capacidades para concluir mis estudios universitarios y en mí, como persona.

DEDICATORIA

Deseo dedicar este trabajo a Dios, mis padres y mi esposa quienes me han apoyado incondicionalmente a través del largo camino de la vida en todos los aspectos, adicionalmente dedico los conocimientos plasmados en la investigación a aquellas personas que buscan su propio camino hacia el éxito mediante el esfuerzo y el trabajo honesto.

RESUMEN

Un código QR es una tecnología global que permite almacenar información y realizar acciones con la misma, tal como páginas *web*, el envío de correos, mensajes de texto, llamadas telefónicas, almacenar eventos en agenda e incluso la ubicación de la posición geográfica.

La siguiente investigación muestra el desarrollo de una aplicación *web*, cuya finalidad es brindar asistencia médica a un individuo según lo requiera. Mediante la lectura del código QR del portador con un dispositivo inteligente, el cuál desplegará información sobre el portador y adicionalmente le permitirá reportar la ubicación geográfica de ese evento, esto con la finalidad poder proveer información básica sobre el portador a quien realice la lectura y reportar la ubicación del portador a los perfiles asociados al suyo.

La propuesta se desarrolló utilizando el *framework* Laravel, programado en PHP, utilizando MySQL como base de datos y sus páginas *web* en HTML5, adicionalmente se realizaron pruebas unitarias e integral sobre de cada uno de los módulos que contiene el prototipo.

ABSTRACT

QR code it's a global technology that allows to store and work with the information itself, by doing tasks like redirecting to *web* pages, send emails, text messages, phone calls, store events on the agenda or the geographic position.

The following investigation details a *web* application development which has as objective to offer medical assistance to the user according to its needs through the carrier's QR code reading with a smartphone, which will show information about the carrier and additionally the user will be able to report the event's geographic location, in order to be able to provide basic information about the carrier to whoever reads and to report the event to the carrier's associated profiles.

The proposal was developed by using Laravel *framework*, programmed based on PHP with MySQL as database and its *web* pages were based HTML5, additionally there were unitary and integration tests done on each of the module's prototype.

ÍNDICE

1. INTRODUCCIÓN.....	1
1.2. Antecedentes.....	1
1.3. Alcance.....	3
1.4. Justificación.....	4
2. MARCO TEÓRICO	5
2.1. Código QR.....	5
2.1.1. Desarrollo del código qr.....	8
2.1.2. Características del código qr	9
2.1.3. Estructura del código QR.....	15
2.1.4. Especificaciones del código QR	16
2.1.5. Conclusión sobre el uso del código QR.....	18
2.1.6. Conclusión parcial	18
2.2. Metodologías de desarrollo de software	18
2.2.1. Áreas de conocimiento en la ingeniería de	18
software	18
2.2.2. Metodologías tradicionales de desarrollo de software	21
2.2.3. Metodologías ágiles de desarrollo	24
software	24
2.2.4. Conclusión sobre las metodologías de	28
desarrollo de software	28
2.2.5. Conclusión parcial	28
2.3. Tecnologías de desarrollo	29
2.3.1. Base de datos.....	29
2.3.2. Computación en la nube (cloud computing).....	34
2.3.3. Lenguajes de programación	38
2.3.4. Marcos de trabajo	38
2.3.5. Software móvil	44
2.3.6. Conclusión sobre las tecnologías de	47
desarrollo.....	47

2.3.7. Conclusión parcial	47
3. DESARROLLO DE LA PROPUESTA.....	47
3.1. Base de datos.....	49
3.2. Product backlog	50
3.3. Sprints	54
3.4. Capturas del desarrollo	60
3.4.1. Pantalla de inicio o bienvenida	60
3.4.2. Pantalla de registro de nuevo usuario	60
3.4.3. Pantalla de inicio de sesión	61
3.4.4. Pantalla de usuario conectado.....	61
3.4.5. Presentación de datos de usuario	62
3.4.6. Pantalla de ingreso visitante de perfil y	62
creación de reporte.....	62
3.4.7. Pantalla de selección sobre reporte de	63
geolocalización de perfil personal y conocidos.....	63
3.4.8. Pantalla de visualización de registro de	63
evento.....	63
3.4.9. Pantalla manejo del perfil en peticiones y	64
contactos	64
3.4.10. Pantalla para manejo de solicitudes de	64
contacto	64
3.4.11. Pantalla de contactos registrados	65
3.4.12. Pantalla ingreso de nueva información	65
general	65
3.4.13. Pantalla ingreso de nueva información de	66
salud	66
3.5. Pruebas unitarias y de integración	67
3.5.1. Registro e inicio de sesión de usuario	67
3.5.2. Perfil social	69
3.5.3. Geolocalización	75
3.5.4. Lectura qr.....	77

4. CONCLUSIONES Y RECOMENDACIONES.....	80
4.1. Conclusiones	80
4.2. Recomendaciones.....	80
REFERENCIAS	82

1. INTRODUCCIÓN

1.2. Antecedentes

La persona como un activo de la sociedad es importante proteger, para lo cual existe en Ecuador el Servicio Integrado de Seguridad ECU911, que pone a disposición de la ciudadanía una variedad de servicios que ofrecen una asistencia única y permanente. (ECU911, s.f.)

Aunque al no contar con documentos o una manera de proveer información en momentos de asistencia por parte del servicio ECU911, puede dificultar la atención del mismo, lo ideal sería contar una herramienta adicional que pueda brindar información personal. Una de las formas para contar con ese recurso, es el tener un código QR individual asignado.

Los códigos QR, siendo un tipo de código de barras bidimensional, permiten que la información se encuentre codificada dentro de un cuadrado, permitiendo el almacenamiento de información alfanumérica.

En un principio los códigos QR tuvieron como finalidad cubrir áreas como la publicidad, campañas de marketing, facilitar la comercialización, implementarse en el diseño gráfico, también estar presente en internet como *blogs* y páginas *web*.

Las acciones que se pueden generar automáticamente mediante los códigos QR pueden ser:

- Abrir URLs
- Leer textos
- Envío de correos electrónicos
- Enviar SMS
- Llamadas telefónicas
- Generar eventos de agenda en el teléfono inteligente

- Ubicación geográfica en plataformas como Google

(Dawson, 2012)

“(…) la lectura de códigos QR es en extremo global y universal hoy en día, ya que la mayoría de celulares inteligentes cuentan con “lectores QR”, sin embargo, en caso de no contar con dicha herramienta en el celular, se puede adquirir cualquier lector QR gratuito de preferencia.” (QR CODES, 2012).

Un estudio realizado por Pitney Bowes acerca del “(…) nivel de uso de los códigos QR sobre lugares impresos versus lugares digitales como pantallas o televisión, en Francia, Alemania, Reino Unido y Estados Unidos.” (Emarketer, 2013).

Young Adults in Select Countries Who Have Scanned a QR Code, by Media, 2012							
% of respondents							
	Magazine	Poster	Mail	Packaging	Website	Email	TV
France							
18-24	21%	18%	17%	18%	13%	10%	3%
25-34	18%	12%	10%	19%	5%	4%	3%
Germany							
18-24	27%	21%	18%	22%	13%	6%	4%
25-34	23%	15%	14%	14%	11%	4%	5%
UK							
18-24	26%	23%	16%	18%	15%	8%	10%
25-34	23%	16%	18%	17%	12%	8%	7%
US							
18-24	39%	35%	38%	32%	21%	14%	9%
25-34	36%	25%	36%	31%	17%	16%	12%
Total (18-34)	27%	21%	21%	21%	13%	9%	7%
Source: Pitney Bowes, "Getting Ahead of the Emerging QR Code Marketing Trend," Jan 15, 2013							
150347				www.eMarketer.com			

Figura 1. Nivel de uso del Código QR en los países seleccionados.

Tomado de (Emarketer, 2013)

Dicho estudio, demuestra que el nivel de aceptación y uso de código QR impreso se usa aún más, en el caso de éste prototipo, el código QR podría estar ubicado bajo criterio del usuario en la zona más visible o accesible ante un evento de requerir su lectura.

En cuanto a los portadores de teléfonos inteligentes de acuerdo con el estudio aportado por el instituto Nacional de Estadística y Censos (INEC), demuestra un crecimiento exponencial.

“En 2 años el acceso a teléfonos inteligentes se ha duplicado pasando de los 500 mil usuarios en 2011 a 1 millón 200 mil en el 2013, señalando un crecimiento aproximado de un 140%(...)” (Espinosa, 2014)



Figura 2. Porcentaje de personas con teléfono inteligente en Ecuador. Tomado de (Espinosa, 2014)

1.3. Alcance

Este proyecto tiene como alcance lograr un prototipo de aplicación *web* que permita brindar oportunamente información al usuario que desee ayudar al propietario del código, dar la ubicación geográfica del lugar en donde se hizo la lectura del código, impulsar disparadores de notificaciones a las personas relacionadas con el perfil y brindar teléfonos de contacto de la persona afectada, además de la creación de una aplicación móvil para la lectura de código QR

Se podrá acceder a los perfiles por medio de la lectura del código QR desde cualquier ubicación con acceso a internet con la aplicación móvil, ya sea para visitar un perfil o crear uno nuevo, contar con la opción de enviar notificaciones a perfiles relacionados con el perfil que se está visitando brindando su ubicación geográfica además de anunciar la lectura del mismo.

1.4. Justificación

Debido a que la idea está orientada a cubrir un tema de salud con tópicos específicos en caso que el usuario no cuente con la información personal consigo o no pueda facilitarla, este, a través de la lectura del código QR individual asignado se podría hacerlo, para lo cual adicionalmente se consideró el desarrollo de un aplicativo móvil.

Citando cifras sobre el uso de los códigos QR, “en la actualidad se cuenta con 1,76 billones de usuarios de teléfonos inteligentes, estimando una cantidad de 2,73 billones para el 2018” (QR Code eBook, 2014)

Alrededor del mundo se ocupa el código QR y a su vez obtienen un gran beneficio, “(...) en Alemania más del 75% de usuarios de teléfonos inteligentes sabe acerca del código QR.” (Estateqr codes, 2014)

Unos de los beneficios que puede traer el uso del código QR son:

- Versatilidad
- Pueden ser usados para cualquier cosa
- Son beneficiosos para el usuario y negocio
- Evita gastos innecesarios a la empresa
- De extrema fácil distribución

(Estateqr codes, 2014)

El tiempo de ejecución está directamente vinculado al tiempo de planificación estratégica sobre sus servicios y procesos, el diseño de políticas, creación de plataforma, manejo de los principales pilares de la información, implementación de forma correcta del uso de la información, entre otras.

Debido a la facilidad de obtención de información, recursos y que se puede contratar servidores para poder cumplir con la totalidad funcional del proyecto

además que la información requerida sobre plataformas y código QR se encuentra tanto en libros como en internet, el proyecto se puede realizar.

1.2. Objetivo General

Identificar personas mediante la creación de perfiles y asignación de un código QR individual.

1.3. Objetivos específicos

- Compartir la ubicación del código QR de una cuenta particular a partir de su lectura, con perfiles vinculados al mismo.
- Implementar el diseño responsivo en la aplicación *web*
- Desarrollar e implementar prototipo de aplicación móvil que permita la creación de perfiles de usuarios, además de la lectura de código QR.
- Realizar pruebas de integración y unitarias del prototipo con la finalidad de asegurar su correcto funcionamiento.
- Identificar la información básica y necesaria e incorporarla como un requerimiento a contar al momento de la creación de perfiles.

2. MARCO TEÓRICO

2.1. Código QR

“El código QR, *QR Code* o *Quick Response Code* (Código de Respuesta Rápida, por sus siglas en inglés), es un símbolo bidimensional, inventado por Denso en 1994 y aprobado como un estándar nacional ISO (ISO/IEC18004) en el año 2000”. (Narayanan, 2012)

Creado con la intención de ser utilizado en el control de partes de automóviles, en la actualidad es visto y usado en cualquier lugar a cualquier hora por las siguientes razones:

- Características superiores a los códigos de barra lineales, por ejemplo, soporta Kan ji/Chino, una densidad de datos más alta.

- La patente fue liberada al público, por lo que su uso es gratuito.
- Equipos alrededor del mundo equipados con cámaras que puedan leer el código QR pueden acceder direcciones de internet automáticamente simplemente con la lectura del URL codificado en el código QR.

Hoy la tecnología basada en código QR se ha vuelto indispensable en muchos aspectos, especialmente para aquellos que pretendan llegar a los usuarios de teléfonos inteligentes.

Al comparar la seguridad de un código de barras (1D) dimensional es inferior a los códigos de barras 2D. Los códigos de barras 1D son muy fáciles de leer mediante el escaneo de las líneas y los espacios. Sin embargo, los códigos de barras 2D no son distinguibles ante los ojos humanos. Con respecto a la legibilidad, los códigos de barras 1D deben escanear a lo largo de una sola dirección. Si el ángulo de una línea de exploración no encaja dentro de un rango, los datos no serán leídos correctamente. La diferencia entre los dos códigos es la cantidad de datos que puede compartir, ya que los códigos lineales o 1D solo pueden contener hasta 20 dígitos, mientras que los códigos QR, son de dos dimensiones, “pueden contener hasta 7.089 caracteres numéricos y 4.296 alfanuméricos, además de 1.817 caracteres kan ji de información” (Chuang, Hu, & Ko, 2010, pág. 22)

Con el código QR es posible mantener una serie de beneficios, en especial para los teléfonos inteligentes, ya que según indica:

“Al leer un código QR con su celular inteligente u otra cámara, puede vincular el *Smartphone* a los contenidos digitales en la *web*; activar una serie de funciones del teléfono, incluyendo correo electrónico, mensajería instantánea y SMS; y conectar el dispositivo móvil a un navegador *web*.” (Korhan, 2011, pág. 69)

Estas y otras ventajas es posible lograr, como abrir más información con respecto a un determinado artículo con solo acercar el teléfono e identificar el

código QR, dar una dirección específica, números de teléfono, acceso hacia un determinado sitio *web* e incluso acceder a una zona de redes inalámbricas.

Como ejemplo hay una aplicación reciente de los códigos QR, es el proyecto de la India 'Aadhar' que le da un número de identificación único para los ciudadanos de la India tanto como el número de seguro social (SSN) en EE.UU. También por primera vez en la aviación india (*Jet Airways*) utiliza los códigos QR en sus productos y servicios. (Korhan, 2011, pág. 69)

Los principales usos que se les da a los códigos QR son:

- Publicidad: El mensaje puede ser una URL de una empresa o información sobre un producto suyo.
- Papelería corporativa: Puede contener la información de una tarjeta de visita sin la necesidad de imprimirla.
- Gestión de stocks y almacenes: Puede contener la información necesaria sobre un artículo en una estantería, incluso indicar si este se encuentra agotado, con la finalidad de realizar un nuevo pedido o si, por el contrario, existen demasiados y es recomendable promocionarlos por medio de ofertas.
- Concursos: Se puede hacer intervenir al instante a un consumidor en una competición, llevándole, incluso, a una página *web* donde puede inscribirse para ganar premios.
- Asesoramiento al consumidor: Puede proporcionar una información adicional del producto con el que viene acompañado.

(Pliego, 2013)

Así, cada vez son más las aplicaciones que se pueden dar a los códigos QR, que de una u otra manera permiten el acceso a información de acuerdo a su requerimiento y bajo su propio riesgo.

2.1.1. Desarrollo del código QR

“Un código QR se distingue por ser un cuadrado con tres pequeños cuadros en tres esquinas y gráficamente se puede identificar varias regiones adicionales”. (Diputación de Cádiz, 2011)

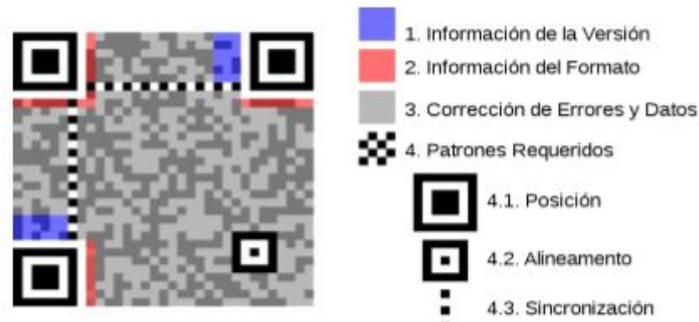


Figura 3. Componentes del Código QR.

Tomado de (Diputación de Cádiz, 2011)

El código QR lo componen cuatro elementos, los cuadros azules representan la información de la versión del código, los fragmentos rojos significan la información con referencia al formato de la información, mientras que la parte ploma indica la corrección de errores y los datos, que a su vez aquí es donde se mantiene realmente la seguridad ya que no podrá ser plagiada, y en los cuadros pequeños negros indican los patrones requeridos tanto de posición, alineación y sincronización.

A lo largo de la historia se han generado diversos códigos para simbolizar dígitos, empezando por los símbolos UPC en 1970 que son ampliamente ocupados en sistemas POS (*Points of Sale*), luego el Código 39 que puede codificar aproximadamente 30 dígitos de caracteres alfanuméricos, posterior tenemos códigos similares conocidos como Código 16k y Código 49, debido al proceso de evolución en la información, se fueron presentando requerimientos de símbolos que puedan almacenar más información y a su vez, representar otro idioma distinto al inglés, como resultante, el código QR, con una capacidad de 7000 dígitos incluyendo caracteres Kan ji fue desarrollado en 1994 (Narayanan, 2012)

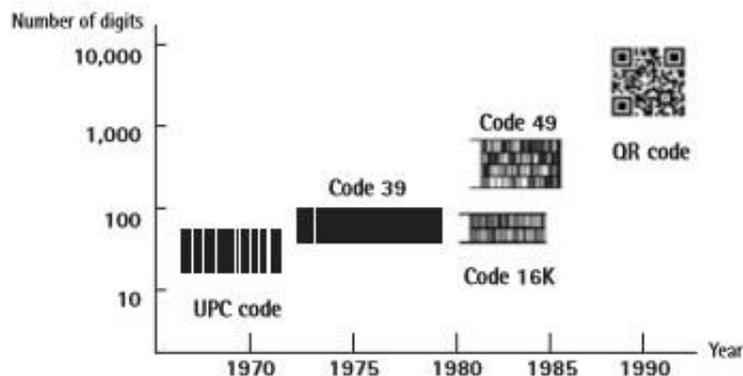


Figura 4. Historia de los símbolos.

Tomado de (Narayanan, 2012)

El código QR como símbolo matricial, permite la impresión de alta densidad de datos matriciales y la lectura de alta velocidad de código Maxi. Los símbolos de dos dimensiones generalmente contienen mucha información cuando son comparados con símbolos lineales (aproximadamente 100 veces más), por lo que el tiempo de proceso y lectura es mucho más complejo, algo que fue considerado al momento de buscar la lectura rápida en el código. (Soon, 2008, p.3)

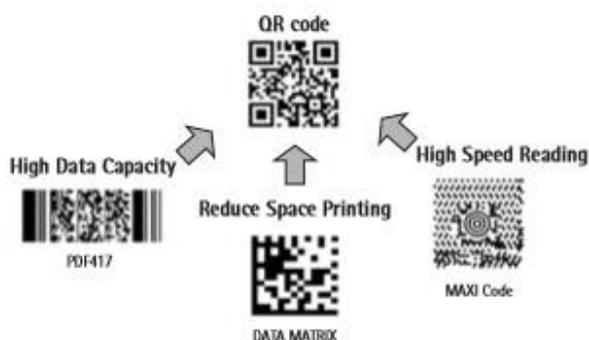


Figura 5. Desarrollo del código QR

Tomado de (Nayayanan, 2012)

El código QR incluye un mayor número de características con el objetivo de hacerlo seguro, rápido de lectura y sobre todo con mayor contenido de información.

2.1.2. Características del código QR

Dentro de las características principales de los códigos QR se encuentran:

- “Alta capacidad de codificación de datos: hasta 7.089 caracteres numéricos o 2.953 bytes.
- Decodificación sencilla y a alta velocidad: desde lectores hardware o aplicaciones *software*.
- Mayor densidad de datos y poco espacio necesario para impresión del código: en torno a 1/10 respecto al código de barras tradicional.
- Adaptabilidad del código a los datos: tamaño en puntos de la matriz según contenido almacenado.
- Soporte de múltiples lenguajes y códigos de caracteres: numéricos, alfanuméricos, binarios, escrituras Kanji, Kana, Hiragana, o cualquier formato de datos mediante la definición de extensiones.
- Permite otras variantes como Micro QR o hasta 16 estructuras añadidas.
- Capacidad de corrección de errores: restauración de hasta un 30% de los datos.” (Luque, 2010, pág. 9)

Adicional a las características más notables de los símbolos bidimensionales también cuenta con superioridad en aspectos como rendimiento y funcionalidad; es preciso mencionar que el código QR utiliza máscaras para los datos, es decir hay mayor diferenciación de niveles claros y oscuros, además facilita la lectura del código ya que existe detección aunque no se lo oriente de manera correcta, incluso puede detectarse si existe distorsión, estructuras en espejos; sin contar con la confidencialidad que permite facilitar el cifrado del código.

2.1.2.1. Lectura de alta velocidad en todas las direcciones

Tras la lectura del código matricial mediante un sensor de área (CCD), son almacenados en la memoria y ocupados por el *software* los datos capturados del escaneo, se analiza e identifica patrones y posición/ángulo/tamaño del símbolo detectado, empieza un proceso de decodificación, mientras que los códigos lineales tienen la ventaja sobre los códigos bidimensionales en cuanto al tiempo de lectura e identificación, con el código QR son la excepción, debido a que

cuenta con patrones en tres de las cuatro esquinas del código. (Soon, 2008, pág. 64)

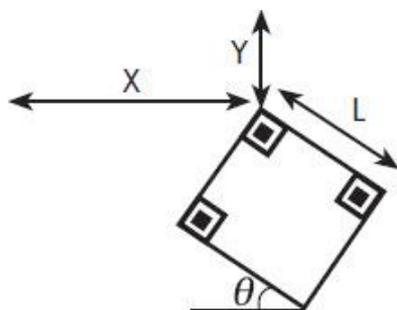


Figura 6. Identificación del código QR.

Tomado de (Soon, 2008)

Para demostrar lo antes dicho es preciso mencionar que los códigos QR son bidimensionales, es decir es posible que contenga información en sentido horizontal y vertical, a diferencia de los códigos de barras tradicionales los cuales están direccionados para ser leídos solo de manera horizontal.



Figura 7. Código QR vs. Código de Barras tradicionales

Tomado de (Luque, 2010)

2.1.2.2. Resistente a la distorsión de los símbolos

Los símbolos están siempre sujetos ante la forma o tipo de superficie en la que se encuentran por lo que el código QR contiene arreglos de patrones de alineación con el intervalo regular en el rango del símbolo. La variación entre la

posición del centro y el patrón de alineación estimados desde la forma del símbolo y la posición del centro son calculados para realizar el mapeo para identificar el centro de cada celda, lo cual permite que los símbolos distorsionados y no lineares se vuelvan legibles. (Soon, 2008, pág. 62)



Figura 8. Corrección de símbolos distorsionados.

Tomado de (Soon, 2008)

2.1.2.3. Restauración de funcionalidad de datos (resistente a símbolos dañados)

El Código QR tiene distintos niveles de corrección de errores, cada uno de ellos son implementados de acuerdo al tipo de daño (7%, 15%, 25%, 30% del área). Se lo hace mediante la utilización del código Reed-Salomon que es altamente resistente a los errores y permite la lectura correcta hasta el nivel de corrección aceptable, dicho nivel de corrección es configurable por el usuario al momento de crear el símbolo. (Soon, 2008, pág. 63)



Figura 9. Símbolos dañados.

Tomado de (Soon, 2008)

2.1.2.4. Eficiencia en la codificación de caracteres kan ji y kana

Debido a que el código QR fue desarrollado con la premisa que iba a ser utilizado en Japón, tiene un 20% de eficiencia en la codificación contra otros códigos bidimensionales. (Soon, 2008, pág. 64)

2.1.2.5. Funcionalidad de símbolos enlazados

El código QR permite representar un símbolo en varios, dividiéndolos hasta un máximo de 16 símbolos. Cada símbolo parte del enlace cuenta un indicador de cuantos símbolos enlazados ha sido dividido el símbolo original, adicional a dicha característica, no es necesario un orden de lectura de los símbolos enlazados, con la funcionalidad de enlace, se puede imprimir códigos enlazados en áreas donde por lo general uno mayor no cabría. (Soon, 2008, pág. 64)

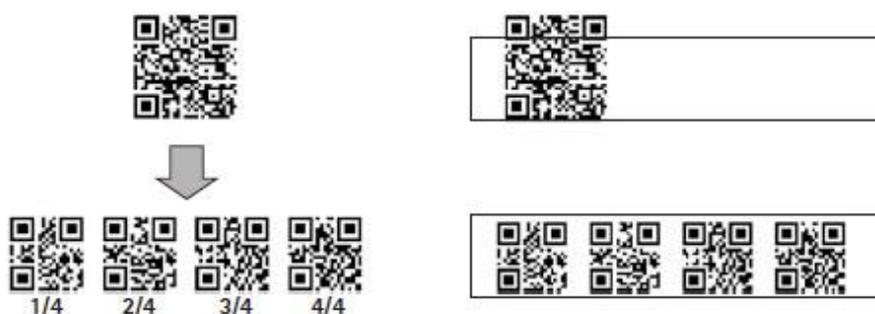


Figura 10. Enlace de símbolos.

Tomado de (Soon, 2008)

2.1.2.6. Proceso de enmascaramiento

Debido a que el código QR cuenta con un arreglo de celdas negras y blancas con el objetivo de analizar los datos captados es necesario que estén bien balanceados, para lo cual se ocupa el cálculo EXOR, encargado de juzgar el patrón a elegir para la información a almacenar en el código. (Soon, 2008, pág. 64)

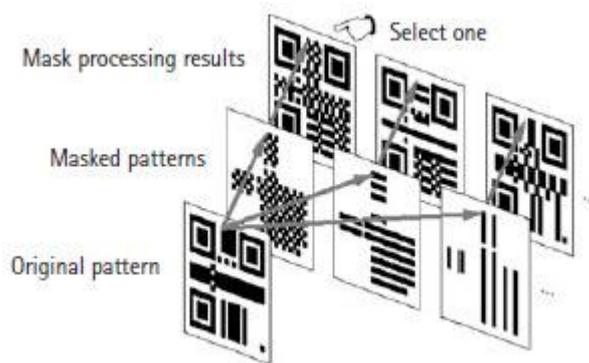


Figura 11. Proceso de enmascaramiento.

Tomado de (Soon, 2008)

2.1.2.7. Confidencialidad del código

Se puede generar una relación entre el tipo de carácter y la información almacenada para un uso especial, logrando una encriptación, únicamente si se descifra la tabla entre el tipo de carácter y la información almacenada, no se podría realizar la lectura del código. (Soon, 2008, pág. 68)

2.1.2.8. Marcación directa

Permite al código no solo ser cuadrado, también circular, de color negro y blanco invertidos, adicionalmente de manera frontal o trasera. (Soon, 2008, pág. 68)

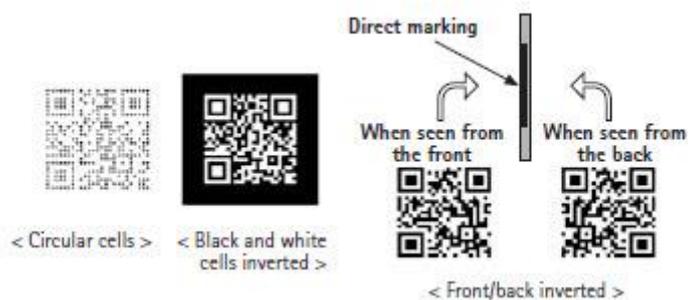


Figura 12. Marcación directa.

Tomado de (Soon, 2008)

2.1.3. Estructura del código QR

Está dada por una caracterización bidimensional, denominada símbolos, mientras que los símbolos forman módulos; es decir, son los cuadros negros, y estos están a su vez codificados de forma binaria (por 0 y 1).

Los módulos ubicados en bloques dentro estructura cuadrada: los patrones de función y la región de codificación. En cada símbolo existen un conjunto de módulos que no contienen datos codificados, sino información necesaria para su decodificación. Son los denominados patrones de función (Luque, 2010)

Para permitir que la lectura del código sea fácil cuenta con patrones de funcionamiento y zonas, los cuales son:

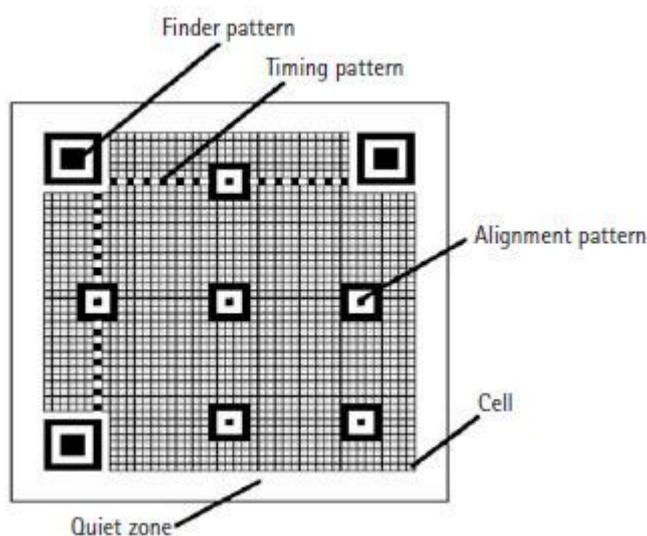


Figura 13. Estructura del código QR.

Tomado de (Soon, 2008)

2.1.3.1. Patrón descubridor

Detecta la posición del código QR, “mediante el arreglo del patrón en 3 esquinas del símbolo, posición, tamaño y ángulo pueden ser detectados en 360 grados”. (Soon, 2008, pág. 68)

2.1.3.2. Patrón temporizador

“El patrón de función que permite re-sincronizar las coordenadas de mapeo del símbolo ante posibles distorsiones moderadas” (Luque, 2010, pág. 11)

Identifica la coordinación central de cada celda en el código QR con los patrones blancos y negros en el arreglo, permitiendo la corrección de los datos de las celdas cuando el símbolo se encuentra distorsionado o cuando el tono de alguna celda tiene algún error. (Soon, 2008, pág. 68)

2.1.3.3. Patrón de alineamiento

“Secuencia alternada de módulos blancos y negros que ayuda a calcular las coordenadas de los módulos del símbolo”. (Luque, 2010, pág. 11). Utilizado para la corrección de la distorsión del código QR, altamente efectivo al momento de corregir distorsiones no lineales.

2.1.3.4. Zona silenciosa

“Para lograr una lectura mucho más fácil por el sensor CCD, se deja un margen de espacio blanco de 4 o más celdas”. (Soon, 2008, pág. 68)

2.1.3.5. Área de datos

Basados en la regla de codificación los datos son guardados en números binarios de “1” y “0” representados por celdas negras y blancas en el arreglo del área de datos en el código QR.

2.1.4. Especificaciones del código QR

Hay dos modos de apertura, por medio del internet o local:

- 1) Modo en línea: Los códigos QR y sus contenidos se almacenan en Internet, siendo accesibles desde cualquier ordenador.
- 2) Modo local: Los códigos QR y sus contenidos se almacenan en el mismo ordenador con el que trabajas, por lo que no serán visibles desde el exterior.

(Fundación Orange, 2010)

2.1.4.1. Tamaño del símbolo

Depende directamente del volumen de datos almacenados y el método de lectura, el símbolo se incrementa por 4 celdas verticales y horizontales en dirección (celdas de 21x21, 25x25, 29x29), también se cuenta con 40 tipos distintos de conjuntos en celdas de 177x177. Por ejemplo, en el caso que sea 45x45 celdas, si un cuadro mide 0.25mm, un lado mide 11.25mm, la zona silenciosa (de un mínimo de 4 celdas) también debe ser agregada, vamos a tener un cuadrado de 12.25mm de lado. (Soon, 2008, pág. 66)

2.1.4.2. Volumen y tipo de información

El código QR puede manejar varios tipos de código, básicamente cualquier carácter soportado por ISO/IEC 646 e ISO/IEC 10646.

Tabla 1.

Especificaciones del código QR

Symbol size	Min. 21x21 cell - Max. 177x177 cell (with 4-cells interval)	
Information type and volume	Numerical characters	7,089 characters at maximum
	Alphabets, signs	4,296 characters at maximum
	Binary (8 bit)	2,953 characters at maximum
	Kanji characters	1,817 characters at maximum
Conversion efficiency	Numerical characters mode	3.3 cells/character
	Alphanumeric/signs mode	5.5 cells/character
	Binary (8 bit) mode	8 cells/character
	Kanji character mode (13 bit)	13 cells/character
Error correction functionality	Level L	Approx. 7% of the symbol area restored at maximum
	Level M	Approx. 15% of the symbol area restored at maximum
	Level Q	Approx. 25% of the symbol area restored at maximum
	Level H	Approx. 30% of the symbol area restored at maximum
Linking functionality	Possible to be divided into 16 symbols at maximum	

Tomado de (Soon, 2008, pág. 66)

2.1.5. Conclusión sobre el uso del código QR

El autor de este trabajo, considera que el código QR, en base a todos los fundamentos presentados, permiten al lector comprender las razones sobre las cuáles se basa para elegir el código QR como la tecnología más destacada para este proyecto.

2.1.6. Conclusión parcial

Se empleó el código QR como tecnología para esta investigación, logrando facilitar y cumplir con los objetivos propuestos, tales como estar en capacidad de asignar perfiles de usuarios a códigos QR de manera dinámica.

2.2. Metodologías de desarrollo de software

El concepto surge como una relación entre ciencias de la computación con conocimientos o disciplinas como administración/manejo e ingeniería de sistemas, convirtiéndose en parte esencial de los proyectos de *software* que da inicio a la definición del ciclo de vida de *software* y las metodologías de desarrollo de *software*, en el cuál se define cómo será estructurado, planeado y controlado el *software*. (SIMAO, 2009)

2.2.1. Áreas de conocimiento en la ingeniería de software

Se mantiene como alcance las siguientes áreas:

2.1.6.1. Requerimientos de software

Definida como la razón del problema real que debe ser exhibido por el *software* para el cual tuvo el propósito de ser, los requerimientos pueden definir funcionalidad del sistema, detalles técnicos o automatización de tareas, en sí, las propiedades del producto.

2.1.6.2. Diseño de software

Es el proceso de definición de las características del sistema, componentes y su expectativa de resultado. Dichas características comprenden su arquitectura, componentes, interfaz, entre otros. Puede ser dividido en dos actividades, la

descripción de la estructura superior, organización y descripción de su rendimiento; y el diseño a detalle donde los componentes son descritos en más detalle para permitir la debida construcción de dicho componente.

2.1.6.3. Construcción de software

Definido como la creación detallada de *software* con resultados, relacionado al diseño, testeo de *software* y soportado por el uso de metodologías, enfatizando en diversas tareas realizadas con la programación, como verificación, desarrollo, integración, testeo unitario, testeo de integración y resoluciones. Con la finalidad de minimizar la complejidad, anticipación de cambio, uso de estándares y verificación de la construcción.

2.1.6.4. Testeo de software

Actividad en la que se evalúa la calidad del producto, mediante la identificación de problemas y defectos, más conocidos como bugs, además de mejorar el producto tras la corrección de los errores detectados. Adicionalmente las pruebas tienen que ser una actividad para prevenir fallas durante el desarrollo y es más oportuno su corrección.

2.1.6.5. Mantenimiento de software

Definido como las actividades requeridas para proveer soporte efectivo a un producto de *software*, estas actividades no solo concierne en cuanto a modificación de *software* después de la entrega, sino también la corrección de errores, mejoras de la calidad del *software* y planeación de entrega con la finalidad de sustentar el *software* en el ciclo de vida sobre el producto, además del registro, seguimiento y medición de impacto de modificaciones o cambios requeridos, de lo cual el resultado es código.

El mantenimiento puede ser dividido en:

- Mantenimiento correctivo, o mejor conocido como modificaciones reactivas, debido a que está orientado a corregir problemas descubiertos tras la entrega del producto.

- Mantenimiento adaptativo, abarca las modificaciones que previenen que el *software* se vuelva inusable debido a cambios del ambiente.
- Mantenimiento perfectivo, apunta a las modificaciones que proveen mejoras de la calidad, por lo general rendimiento.
- Mantenimiento preventivo, modificaciones para detectar y corregir posibles potenciales errores del producto, antes de tornarse errores.

2.1.6.6. Configuración del manejo de software

Actividad de soporte en el desarrollo de *software* donde la configuración, está definido como la suma de los componentes funcionales y características físicas del sistema. Manejo de lanzamiento y entrega de *software*, actividad relacionada con la entrega de productos y su manejo, además de tratar la construcción de librerías de *software* y manejo de versiones.

2.1.6.7. Procesos de ingeniería de software

En esencia es la definición del ciclo de vida del *software*, la implementación, evaluación, mejoras, medición y cambio.

- Proceso de implementación y cambio, involucra cambios en la infraestructura, modelos, actividades y prácticas del *software*.
- Proceso de evaluación, generalmente realizado bajo modelos y métodos de evaluación, los cuales están basados en mejores prácticas acorde a las diferentes variables que se presenten en el proyecto, como por ejemplo *ITIL* o *CMMI*, es importante considerar que dichos métodos deben proveer puntajes cuantificables, los cuales caracterizan la capacidad del proceso.
- Mediciones del producto y proceso, los procesos pueden ser medidos por sus fortalezas y debilidades, o por el nivel de impacto que produce su resultado, mientras que el producto puede ser medido por el tamaño del producto, es decir, cantidad de código, documento con las

especificaciones, su estructura, el nivel de diseño, diversas características que pueden ser puestas en consideración.

2.1.6.8. Herramientas y métodos de ingeniería de software

Usados para asistir en las actividades de ingeniería de *software* durante el ciclo de vida, con objetivos como la simplificación de actividades realizadas en el caso de herramientas, mientras que los métodos están divididos en el tipo de acercamiento hacia el proyecto, la metodología puede aconsejar el uso de herramientas y métodos cuando se hace referencia a las prácticas, logrando proveer ayuda en las actividades de la ingeniería de *software*.

2.1.6.9. Calidad

Cuando se realiza la ingeniería de *software*, se debe tener presente que no solo el producto debe ser de calidad, sino también las actividades realizadas, considerando la ética adicionalmente. Usualmente la calidad va acompañada de costos en prevención, evaluaciones, fallas internas y externas. Cada producto cuenta con su costo previsible para cada una de las posibilidades tomadas en cuenta, por ello, es necesario establecer un balance entre la calidad y costos para lograr los requerimientos de *software* y a su vez mantener la calidad a lo largo del desarrollo y producto. (SIMAO, 2009, pág. 11)

2.2.2. Metodologías tradicionales de desarrollo de software

“Las metodologías tradicionales o formales se focalizan en documentación, planificación y procesos. (Plantillas, técnicas de administración, revisiones)”. (Uñoja, 2011)

“Teniendo en cuenta la filosofía de desarrollo de las metodologías, aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales o Pesadas”. (Ticona, 2011)

Para lograr manejar las metodologías tradicionales es preciso generar requisitos adecuados de trabajo sobre el proceso de desarrollo del *software*; para lo cual

es necesario planificar todo el proyecto, luego de que todo está totalmente claro y detallado, es posible comenzar el ciclo de desarrollo del *software*. Las metodologías tradicionales tienen la característica principal de que no se adaptan a cambios pues son procesos rígidos, y es así que los requisitos completos no es posible predecirlos o pueden variar.

Entre las metodologías tradicionales más utilizadas se puede detallar a:

- RUP (*Rational Unified Process*)
- MSF (*Microsoft Solution Framework*)

2.2.2.1. Metodología RUP

Es conocida de concepto en como “Forma la disciplina de asignar tareas y responsabilidades en una empresa de desarrollo, aquí se define quién hace, qué hace, cuándo y cómo” (Ticona, 2011)

La metodología de procesos en RUP, se fundamenta en el análisis y diseño, requerimiento y prueba desarrollo. Así en cuanto a metodología de datos se establecen a través de: “casos de uso, paquetes, modelo conceptual, diagrama de secuencias” (Ticona, 2011)

Se dice que la metodología RUP se da para un tamaño pequeño de grupo de desarrolladores, máximo hasta cuatro personas, aquí la carga de trabajo es mayor ya que deben dividirse tareas para un número pequeño de personas, se requiere gran documentación, la duración de los proyectos al aplicar esta metodología es de larga duración, así como la evaluación del Estado del Proyecto es en largos periodos; y se mantiene poca accesibilidad al código fuente por parte del cliente.

Las cuatro fases del ciclo de vida RUP son:

- Concepción
- Elaboración
- Construcción

- Transición

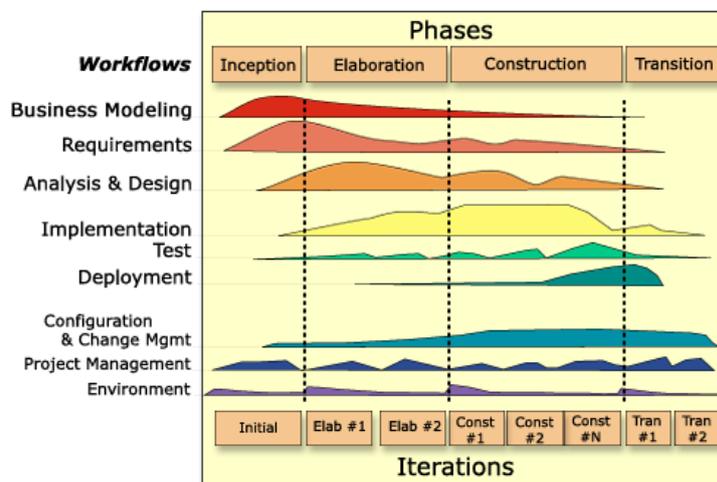


Figura 14. Proceso con metodología RUP.

Tomado de (Ticona, 2011)

Tenemos como las fortalezas de RUP las siguientes:

- En las fases hay una evaluación, permitiendo así, el cambio de objetivos.
- Se adapta perfectamente a proyectos innovadores.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el *software*.
- El seguimiento en cada una de las fases es detallado.

RUP cuenta con los siguientes puntos que pueden considerarse como debilidades:

- El proceso de evaluación de riesgos es complejo.
- En determinados proyectos la flexibilidad puede ser excesiva.
- Es necesario que el cliente este en la capacidad de detallar y comprender el alcance del proyecto en un gran nivel de detalle. (Uñoja, 2011)

2.2.2.2. Metodología microsoft solution framework (msf)

Esta metodología tradicional se enfoca en las mejores prácticas administrativas de proyectos de *software*. Esta no es una metodología únicamente rígida, está más enfocada en modelos adaptables a cualquier proyecto de TI.

En este caso las fases del proyecto son:

- Planificación.
- Desarrollo.
- Estabilización.
- Implantación.

(Uñoja, 2011)

Las primeras metodologías de desarrollo de *software* o tradicionales se basan en modelos de *software* que tienen como definición el uso del ciclo de vida del *software*, lo cual mejoró el nivel de efectividad y exactitud, pero también incrementó el nivel de rigidez y se volvió menos flexible.

2.2.3. Metodologías ágiles de desarrollo de software

Mejor conocido como la nueva tendencia del desarrollo de *software*, fuertemente relacionado con el modelo de desarrollo rápido de aplicaciones, con un gran diferenciador del modelo tradicional, el cual es la auto-organización del equipo y el uso de las habilidades de gestión entre equipos funcionales. A continuación, unas de las metodologías ágiles de desarrollo más notables.

2.2.3.1. Desarrollo adaptativo de software (das)

Técnica propuesta por Jim Highsmith con el fin de elaborar *software* y sistemas complejos, con un enfoque hacia la organización y colaboración del equipo, está compuesta de las siguientes capas:

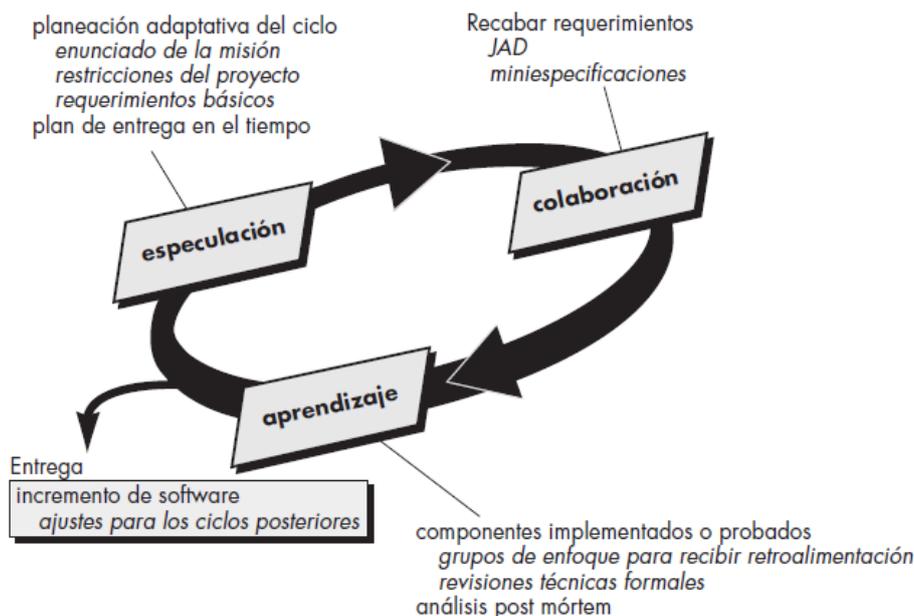


Figura 15. Proceso XP

Tomado de (Pressman, 2005, p.68)

Especulación; Etapa de inicio del proyecto en la cual se conduce la planeación adaptativa del ciclo, usando información como misión de los clientes, restricciones y requerimientos básicos del proyecto para establecer el conjunto de ciclos de entrega en los incrementos de *software*. Debido a que la información concretada en el primer ciclo puede variar, el plan puede ser revisado y ajustado, manteniendo siempre un ambiente de colaboración.

Colaboración; Son los componentes para incrementar la producción creativa con factores como evitar enojos en base a críticas, ayudar sin resentimiento, trabajo duro, contar con las aptitudes para la contribución al trabajo y comunicar problemas o preocupaciones.

Aprendizaje; El ciclo adaptativo empieza el momento en que el desarrollo de los componentes inicia, con ello, el aprendizaje, etapa en la que se mejora el nivel de entendimiento real tanto de proyecto como proceso y tecnología empleada. “Los equipos DAS aprenden de tres maneras: grupos de enfoque, revisiones técnicas y análisis post mórtem del proyecto.” (Pressman, 2005, pág. 9)

2.2.3.2. Scrum

Scrum, original de un término estratégico de rugby en la que se denota el trabajo en equipo, sin requerir o proveer un método o práctica de desarrollo de *software* específico, pero una cierta cantidad de prácticas de gestión y herramientas en distintas etapas de *Scrum* para evitar problemas relacionados a la complejidad o falta de dirección.

Iterativo e incremental durante el proceso de desarrollo para un producto o gestión de cualquier trabajo, enfocado en cómo los miembros del equipo deberían funcionar para producir un sistema flexible en un ambiente de cambio constante, con la finalidad de que al culminar cada iteración se produzca un grupo potencial de funcionalidades. (Schwaber & Sutherland, 2013)

A continuación, las actividades de *Scrum* son:

- Planificación de la Iteración (*Sprint planning*); Compuesto de dos reuniones, la primera de máximo de 4 horas, para que el cliente presente al equipo encargado del proyecto una lista de requisitos priorizada, en la cual se presenta los requisitos prioritarios, lista a la cual el equipo revisa, pregunta y añade condiciones de satisfacción, además de seleccionar objetivos y requisitos de los cuales el equipo se compromete a completar en la iteración. En la segunda reunión a si mismo con un tiempo máximo de 4 horas, el equipo planifica la primera iteración en base a una táctica, obteniendo como resultado la definición de las tareas necesarias para poder completar los objetivos y requerimientos, todo esto plasmado en una lista de tareas de la iteración.
- Ejecución de la Iteración (*Sprint*); La forma de ejecución del proyecto es por bloques temporales cortos y fijos, es decir con duración variable de dos semanas hasta un mes, en cada iteración se proporciona un resultado completo, con un incremento del producto y volviéndolo potencialmente entregable, durante las iteraciones y el tiempo establecido, adicionalmente el equipo a cargo del proyecto tiene reuniones para sincronizar las inspecciones de trabajo entre los miembros, además de

actualizar sobre la iteración la lista de tareas, los gráficos de trabajo pendiente. También el facilitador se asegura de que el equipo mantenga o mejore su ritmo de productividad y en cumplimiento con la iteración.

- Reunión diaria de sincronización del equipo (*Scrum daily meeting*); En las reuniones diarias, cada miembro del equipo inspecciona el trabajo que los otros integrantes realizan, como tareas, progresos de la iteración, obstáculos, con el objetivo de transferir la información, aumentar la colaboración del equipo, por ende, la productividad.
- Demostración de los requisitos completados (*Sprint review*); Consta de una reunión informal donde los requisitos completados en la iteración se presentan al cliente por parte del equipo.
- Retrospectiva (*Sprint retrospective*); Actividad en la que se analiza la forma y calidad del trabajo del equipo en conjunto con el facilitador durante la iteración, analizar como consiguieron o no cumplir con los objetivos, con el objetivo de poder aumentar su productividad y la calidad del producto en desarrollo.
- Refinamiento de la lista de requisitos y cambios en el proyecto (*Product backlog refinement*); etapa de trabajo del cliente con el equipo sobre la lista de objetivos y requisitos priorizados del proyecto, añadiendo, modificando, eliminando y reasignando prioridades a los requisitos, conforme se acerque el momento del desarrollo. (Schwaber & Sutherland, 2013)

Las responsabilidades que destacan en esta metodología son:

- Cliente (*Product Owner*); es el representante de las personas interesadas en los resultados del proyecto, define los objetivos del proyecto, además de dirigir los resultados y maximizar el ROI (retorno de inversión).
- Facilitador (*Scrum Master*); Líder del equipo, involucra al equipo con valores, principios ágiles, reglas, el proceso de *Scrum*, mejora y guía la colaboración entre el equipo y con el cliente.

- Equipo (*Team*); Grupo auto organizado, de 5 a 9 personas que desarrollan de manera conjunta el producto del proyecto, con un objetivo común, compartir la responsabilidad del trabajo por iteración en el proyecto.

Las principales herramientas ocupadas en el proceso del proyecto son:

- Lista de requisitos priorizada (*Product Backlog*); representación de la visión y expectativas sobre los objetivos y entregas del proyecto, esta lista es responsabilidad del cliente respecto a su creación y gestión, con ayuda del facilitador y equipo.
- Lista de tareas de la iteración (*Sprint Backlog*); Es la lista de tareas elaborada por el equipo en la reunión de planificación de la iteración, para lograr cumplir con los objetivos y requisitos comprometidos para demostrar al final de la iteración al cliente.
- Gráficos de trabajo pendiente (*Burndown Chart*); Gráfico que muestra la velocidad a la que se completa progresivamente los objetivos y requisitos, permitiendo comprender al equipo si se podrá completar el trabajo en el tiempo estimado. (Schwaber & Sutherland, 2013)

2.2.4. Conclusión sobre las metodologías de desarrollo de software

El desarrollo de este aplicativo *web* se lo realizará empleando la metodología *Scrum* por su capacidad iterativa e incremental, con un enfoque realista sobre el esfuerzo utilizado en el mismo.

2.2.5. Conclusión parcial

Mediante el empleo de la metodología *Scrum* se puede obtener un nivel detallado del trabajo realizado en el transcurso de la investigación, información y material como los *product backlog* y los *sprints*, que contienen el seguimiento y desarrollo de las tareas propuestas.

2.3. Tecnologías de desarrollo

2.3.1. Base de datos

Las bases de datos son “un conjunto de datos comunes que se almacenan sin redundancia para ser útiles en diferentes aplicaciones, es decir es un fondo común de información almacenada en una computadora para cualquier persona o programa autorizado para acceder a ella”. (Galiano, n.d.)

Adicionalmente hay que considerar los sistemas gestores de bases de datos también llamados DBMS (*DataBase Management System*), “(...) es un *software* con capacidad para definir, mantener y utilizar una base de datos; lo que permite definir estructuras de almacenamiento, acceder a los datos de forma eficiente y segura”. (Galiano, n.d.)

Dentro de las características más notables en una base de datos están:

Tabla 2.

Ventajas de Bases de Datos

Característica	Ventajas
Independencia física	- Simplifica el desarrollo de las aplicaciones, que no han de preocuparse del almacenamiento de los datos. - Permite realizar cambios en la estructura física para optimizar el uso de la BD sin tener que modificar las aplicaciones.
Independencia lógica	- Facilita la realización de cambios en las aplicaciones sin tener que modificar la base de datos. - Cada aplicación organiza los datos en función de sus propias necesidades.
Lenguajes de consulta	- Facilitan al usuario el acceso a los datos.
Centralización	- Gestión automática de accesos concurrentes.
Consistencia	- Mantenimiento de la integridad de los datos. p.ej. Datos duplicados, valores incorrectos...
Seguridad	- Control sobre el acceso a los datos particulares. - El uso de bases de datos permite que los datos se

Tomado de (DECSAI, 2010)

2.3.1.1. Tipos de base de datos comunes

- **Modelo Jerárquico.** - El primer modelo que apareció de bases de datos, y se caracteriza por sus estructuras en forma de registros interrelacionados en forma de árbol.

- **Modelo en Red.** - Estos aparecieron a principios de los años setenta y consiste en que es parecido al modelo jerárquico, sin embargo, hay registros e interrelaciones, pero un registro ya no está limitado a ser el “hijo” de un solo registro tipo sino en varios. (Rafael Camps Paré, 2005)
- **Modelo Relacional.** - Aparece gracias a IBM, propuesto en 1969, donde se utiliza el lenguaje SQL en todos los SGBD, la mayoría de los sistemas informáticos que actualmente se encuentran en funcionamiento utilizan una SGBD relacional. (Camps, 2005, págs. 22-26)
- **Modelo Transaccional.** - Bases de datos no tan conocidas orientadas al análisis de calidad o datos de producción con el objetivo de envío y recepción de una cantidad determinada de datos a gran velocidad.
- **Modelo Multidimensional.** - Técnica para formalizar y visualizar modelos de datos como un grupo de medidas que son definidas por aspectos de los procesos del negocio, formada por hechos (colecciones de estadísticas de negocio) y dimensiones (parámetro sobre el cuál se realizará el análisis de los hechos y datos). (Agrawal, 2015, págs. 119-123)
- **Modelo Orientado a objetos.** - También conocido como sistema gestor de base de datos orientada a objetos (ODBMS) y tiene como objetivo almacenar objetos en vez de datos como enteros, cadenas o números reales, el mencionado modelo consiste de atributos, que representa las características de un objeto y métodos, que representa el comportamiento de un objeto, mejor conocidos como procedimientos o funciones. (Allen, 2001)

2.3.1.2. Gestores de bases de datos

También conocidos como el conjunto de programas enfocados hacia el almacenamiento, modificación y extracción de información de un repositorio conocido como base de datos, adicionalmente dichos programas permiten

realizar acciones como la modificación, creación, eliminación y análisis de datos, esto, mediante herramientas que permiten interactuar con los datos.

Los gestores de base de datos adicionalmente cuentan con métodos para asegurar que los datos se mantengan íntegros, a su vez, también permiten administrar el acceso de los usuarios a los datos y ofrecen recuperar la información en caso que el sistema se corrompa. Otra de las bondades de los sistemas que se describe, es que suelen contar con un generador de informes en variados formatos e incluso la presentación de la información en gráficos o tablas. (Bachman, 2013).

2.3.1.3. Tipos de gestores de bases de datos

Es posible nombrar algunos tipos de sistemas para bases de datos que cumplen dichas características y ocupan un tipo de base de datos, por ejemplo: Oracle, IBM DB2, Microsoft SQL Server.

2.3.1.3.1. Oracle

“Definido como una colección de datos tratados todos ellos como una unidad que está formada por diversos tipos de ficheros dentro de un sistema operativo. Físicamente, se trata la Base de Datos como un conjunto de ficheros de base de datos y ficheros de traza. Lógicamente, como un conjunto de diccionarios, tablas de usuarios y ficheros de traza conteniendo datos de recuperación de errores. Adicionalmente, requiere uno o más ficheros de control. Dichos ficheros contienen aquella información que identifica y describe el resto de la Base de Datos.” (Vives, 2007, pág. 2)

ORACLE es considerado un sistema compartido debido a que puede es posible acceder desde múltiples mediante el código de acceso que brinde ORACLE. (Vives, 2007, pág. 2)

“ORACLE como sistema consiste en un área de memoria llamada Área Global del Sistema (SGA) que permite una comunicación entre los procesos, al menos

cinco procesos en segundo plano (SMON, PMON, DBWR, LGWR y ARCH) utilizados por los usuarios de ORACLE.” (Vives, 2007, pág. 2)

A continuación, una representación del funcionamiento general de ORACLE, en el área global del sistema:

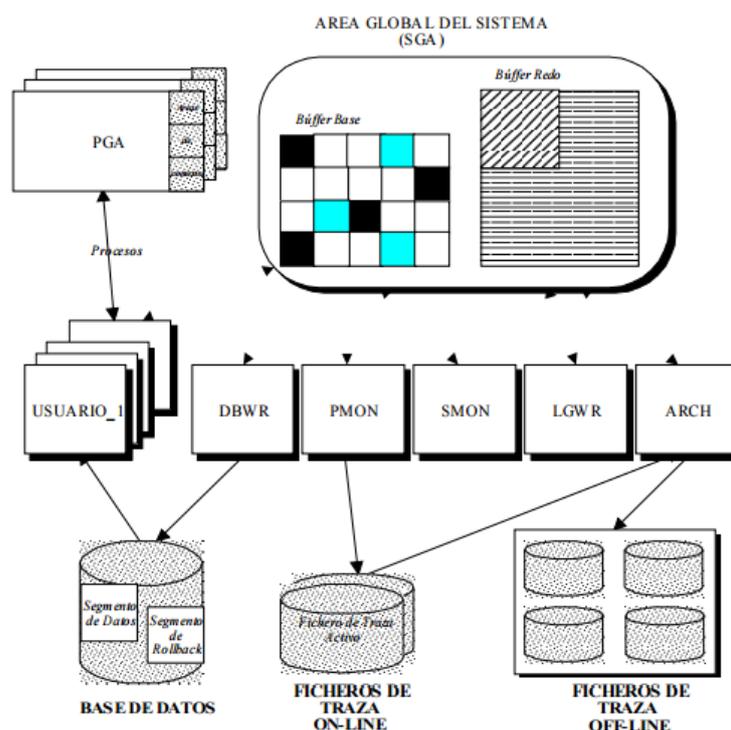


Figura 16. Funcionamiento General del SGBDR ORACLE.

Tomado de (Vives, 2007)

2.3.1.3.2. PostgreSQL

“Implementa las características necesarias para competir con cualquier otra base de datos comercial, con la ventaja de tener una licencia de libre distribución BSD. La migración de bases de datos alojadas en productos comerciales a PostgreSQL se facilita gracias a que soporta ampliamente el estándar SQL. PostgreSQL cuenta con una serie de características atractivas como son la herencia de tablas (clases), un rico conjunto de tipos de datos que incluyen arreglos, BLOB, tipos geométricos y de direcciones de red.” (Ginestà & Mora, 2006)

PostgreSQL incluye procedimientos en lenguajes propios de bases de datos comerciales como PL/SQUL a PL/PGSQL. Sin embargo, PostgreSQL aún está en evolución, ya que se espera que en futuras versiones se pueda incluir nuevas características y mejoras en el diseño interno.

Las bases de datos en PostgreSQL, siendo un modelo relacional pueden a veces integrarse con el modelo orientado a objetos, incluso trabajar con ambos modelos. Sin embargo, es importante considerar que la coexistencia de ambos modelos requiere de un equilibrio, evitando conflictos y cumpliendo con los conceptos fundamentales para ambos. (Ginesta & Pérez, 2013)

Existen dos enfoques para asociar el concepto de clase con el modelo relacional:

2.3.1.3.3. MySQL

MySQL se distribuye libremente en Internet bajo la licencia GPL que le da beneficios como la estabilidad y desarrollo. Basado en C/C++ cuenta con una gran cantidad de ejecutables para alrededor de 19 plataformas distintas. Su API puede ser encontrado en TCL, Ruby, Python, Perl, Java, Eiffel, C, y C++. Orientado hacia la optimización en equipos con múltiples procesadores, con un tiempo de respuesta efectivo, se lo puede ocupar para estructuras cliente-servidor o aplicaciones. (Guia SGBD, 2014)

Para quienes utilizan el lenguaje C/C ++, se hace muy sencillo el uso de MySQL. Pues su diseño e interfaz es similar; y se mantiene en la idea de crear herramientas específicas para cada cosa. En cuanto a UNIX también se puede observar que se mantiene la sintaxis, es así que en la utilización de expresiones la diferenciación de funciones va entre paréntesis, los valores lógicos 0 y 1, el uso del tabulador para completar sentencias, son algunas similitudes. Es decir, el uso de MySQL estaría totalmente adaptado a la plataforma de C/C++ y UNIX; pero con una mejor interfaz y rapidez en sus procesos.

2.3.2. Computación en la nube (cloud computing)

“El concepto de servicios en la nube o *Cloud Computing* está revolucionando el modo en el que las organizaciones implementan su infraestructura de Tecnologías de la Información (TI). Los diferentes usos y servicios ofertados bajo el paraguas de este nuevo paradigma afectan, por un lado, a las organizaciones y particulares que observan cómo la prestación de servicios en la nube les reporta ventajas en términos de eficiencia, flexibilidad y disminución del esfuerzo inversor y, por otro, a las empresas tecnológicas y operadoras tradicionales que ven una oportunidad de ampliar su negocio.” (Management Solutions, 2012, pág. 3)

2.3.2.1. Almacenamiento en la nube

Surge como idea en 1960, con el objetivo de brindar espacio de almacenamiento virtual, también conocido como *cloud storage*, basado en redes de computadoras. (computerweekly, 2009)

Es evidente que se ha hecho esfuerzos cada vez mayores por compilar y almacenar datos de manera segura, además es una alternativa idónea empresarial que se ajusta a las necesidades del mercado.

2.3.2.2.1. Características de la computación en la nube (cloud computing)

Su principal característica es la abstracción del usuario final de los datos respecto a la infraestructura de las Tecnologías de la Información que está accesible para el mismo. La tecnología tanto en hardware como *software* se ofrece por el proveedor como un servicio de red; la capacidad y rendimiento de la información aumenta, de forma escalable y modulable, flexibilizando el coste en función al uso que se hace de los servicios. A través de este modelo posible generar la oferta de productos, servicios de consumo y empresariales por medio de plataformas en internet

De acuerdo al modelo de desarrollo e implementación es posible la entrega de información en tiempo real, así como de productos, servicios y soluciones en

Internet. También se cuenta con la compartición de recursos que se apoya en una virtualización, de lo cual es posible desprender servicios elásticos para el cliente. (Management Solutions, 2012, pág. 12)

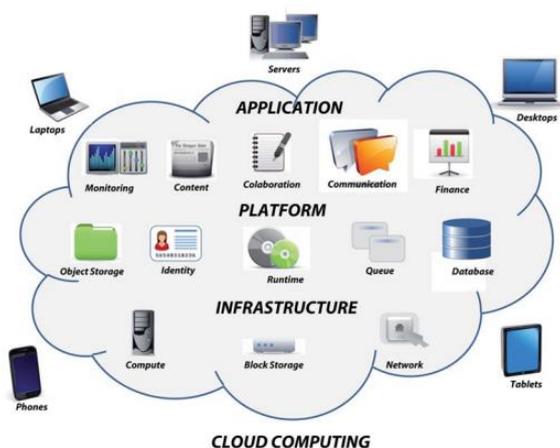


Figura 17. Ecosistema de la nube.

Tomado de (Management Solutions, 2012)

“(…) puede afirmarse que el término *Cloud Computing* no hace referencia a una tecnología concreta sino a un modelo de implementación de servicios de TIC a través de Internet”. (Management Solutions, 2012, pág. 11) lo cual proyecta a los usuarios se decidan por generar su información a través del uso del *cloud computing* o *clouding*, gracias a su modelo de servicios ajustado al 100% a la *web*, con flexibilidad y adaptados a los recursos tecnológicos que necesite el cliente.

A continuación, se muestra una gráfica comparativa sobre el modelo tradicional de almacenamiento de datos, vs el modelo de *cloud*:

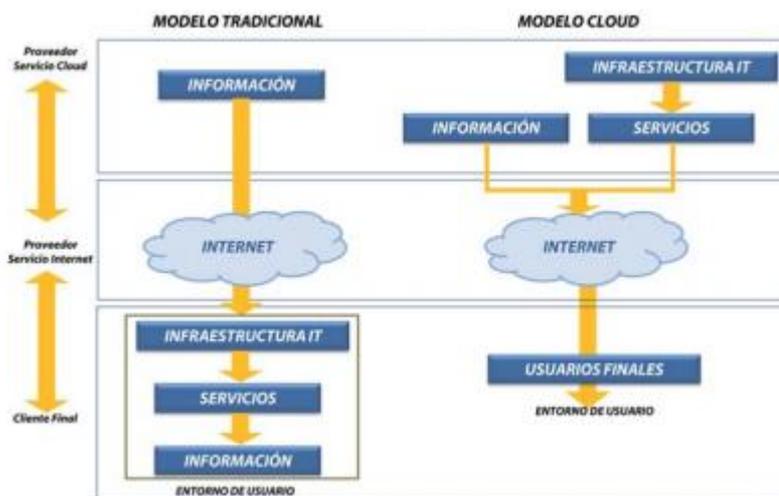


Figura 18. Modelo Tradicional vs. Modelo Cloud.

Tomado de (Management Solutions, 2012)

2.3.2.2.2. Soluciones de cloud computing

“(…) se definen tres características fundamentales que marcan la clasificación de las soluciones *cloud*: familias, formas de implementación y agentes intervinientes. Estas tres características, junto con sus diferentes tipos de soluciones asociadas, se pueden representar en un cubo de tres dimensiones (…)” (Alberto U, 2012, pág. 17)

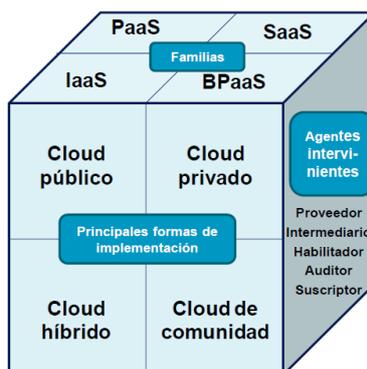


Figura 19. Cubo de clasificación de Soluciones.

Tomado de (Alberto U, 2012)

2.3.2.2.3. Modelos de servicio por familias

- Infraestructura como servicio (IaaS). - Es aquella que pone a disposición del cliente el uso de la infraestructura adicional a los servicios de lo que

contenga se contenga en la misma, aplicable cuando el cliente desea evitar adquirir directamente los recursos necesarios para ejecutar servicios como el servidor, base de datos, equipos de red, entre otros. (Alberto U, 2012)

- *Software* como un servicio (SaaS). - Consiste de la entrega de *software* para el cliente mediante descargas y ejecución de aplicaciones en los servidores *web* del cliente bajo un sistema temporal o perpetuo dependiendo del acuerdo. (Management Solutions, 2012, pág. 7)
- Plataforma como un servicio (PaaS). - Entrega en forma de servicio a un cliente el entorno de desarrollo de aplicaciones, proporcionando red, servidores y almacenamiento. (Alberto U, 2012)

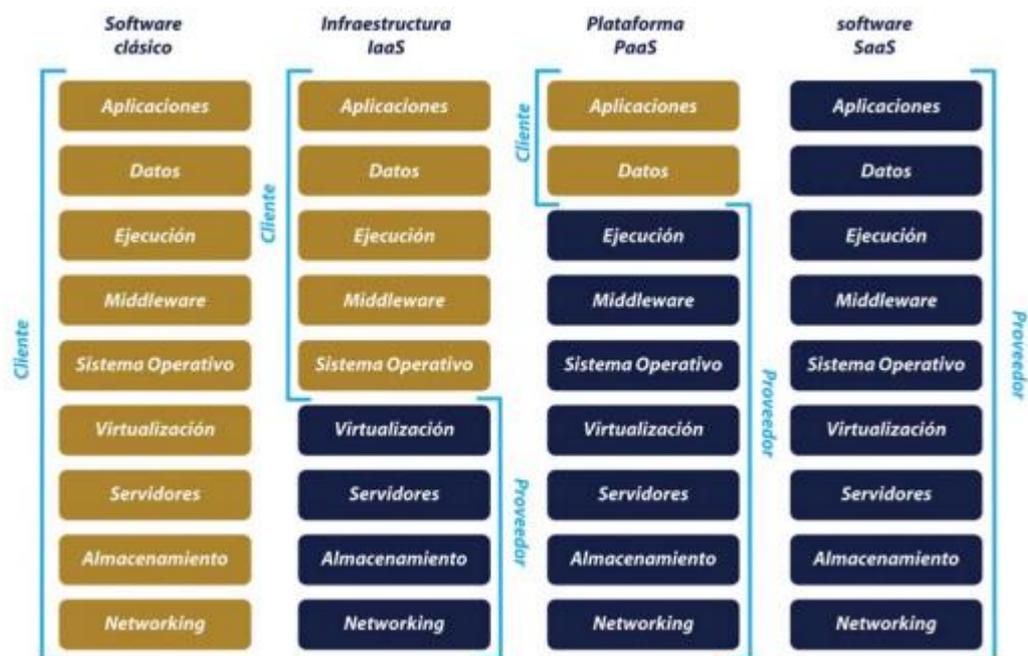


Figura 20. Modelos de Servicio.

Tomado de (Management Solutions, 2012)

2.3.3. Lenguajes de programación

2.3.3.1. Php

PHP fue creado con el principal objetivo de desarrollar contenido *web* dinámico, adicionalmente popular al momento de generar documentos XML, gráficas, animaciones flash, archivos PDF, entre otros.

Cuenta con la ventaja de soportar bases de datos como MySQL, PostgreSQL, Oracle, Sybase, MS-SQL, DB2 y ODBC, además de bases de datos NO-SQL como SQLite y MongoDB. Las páginas de PHP son generalmente páginas de HTML con comandos de PHP embebido, lo cual explica el alto nivel dinámico que pueden presentar estas páginas. (Tatroe, 2003, págs. 4, 7)

2.3.4. Marcos de trabajo

Al hablar de un marco de trabajo o *framework*, es como referirse a una estructura de componentes editables e intercambiables al momento del desarrollo de una aplicación.

“(...) en general, con el término *framework*, nos estamos refiriendo a una estructura *software* compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.” (Gutierrez, 2010, pág. 1)

2.3.4.1. Funcionamiento de un framework

La mayor parte de los *frameworks* trabajan con un patrón denominado MVC (Modelo-Vista-Controlador), se dice que dicho patrón está en capacidad de organizar el aplicativo en tres modelos, el modelo, que comprende las reglas del negocio y datos del aplicativo, la vista, trata sobre el conjunto de vistas que permiten representar los formularios de entrada y salida con la información y el tercero, conocido como controlador, encargado de manejar peticiones por parte de los usuarios y flujos del sistema. (Gutierrez, 2010, pág. 2)

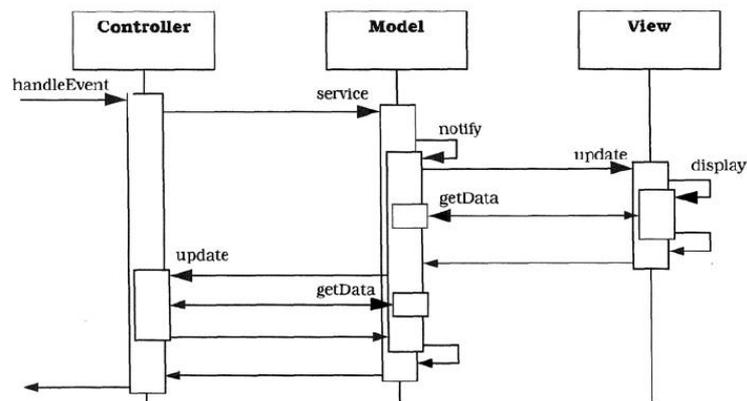


Figura 21. Modelo MVC

Tomado de (Gutierrez, 2010)

2.3.4.2. Características de un framework

A continuación, características más notables de un *framework*:

Tabla 3.

Características de un framework

Abstracción de URLs y sesiones.	No es necesario manipular directamente las URLs ni las sesiones, el <i>framework</i> ya se encarga de hacerlo.
Acceso a datos.	Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc....
Controladores.	La mayoría de <i>frameworks</i> implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser

	fácilmente adaptables a las necesidades de un proyecto concreto.
Autenticación y control de acceso	Incluyen mecanismos para la identificación de usuarios mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios.

Tomado de (Gutierrez, 2010)

2.3.4.3. Tipos de framework

Existen diferentes tipos de *framework* cada uno adaptado a la necesidad del usuario del sistema, y la interfaz que requiere mantener disponible, “la mayor parte de *frameworks* se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC ofreciendo mecanismos importantes que facilitan la integración de otras herramientas para la implementación de las capas de negocio y presentación”. (Gutierrez, 2010, pág. 3)

Entre los ejemplos de *Framework* a exponer en este trabajo se encuentran Django, Symfony y Laravel:

2.3.2.2.4. Django

Framework web escrito en Python, con un inicial objetivo de ser un administrador de contenido para sitios de noticias, hasta que se definió que podía cubrir una gran cantidad de tipos de aplicaciones, nombrado Django en honor a Django Reinhart.

“(…) proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos. Por ejemplo, en lugar de requerir que los desarrolladores escriban controladores y vistas para las áreas de administración de la página, Django proporciona una aplicación incorporada para administrar los contenidos (*django.contrib.admin*), que puede incluirse como parte de cualquier página hecha con Django y que

puede administrar varias páginas hechas con Django a partir de una misma instalación; la aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios (incluyendo una asignación detallada de permisos).” (Defossé & Marcos, 2009, págs. 10, 11)

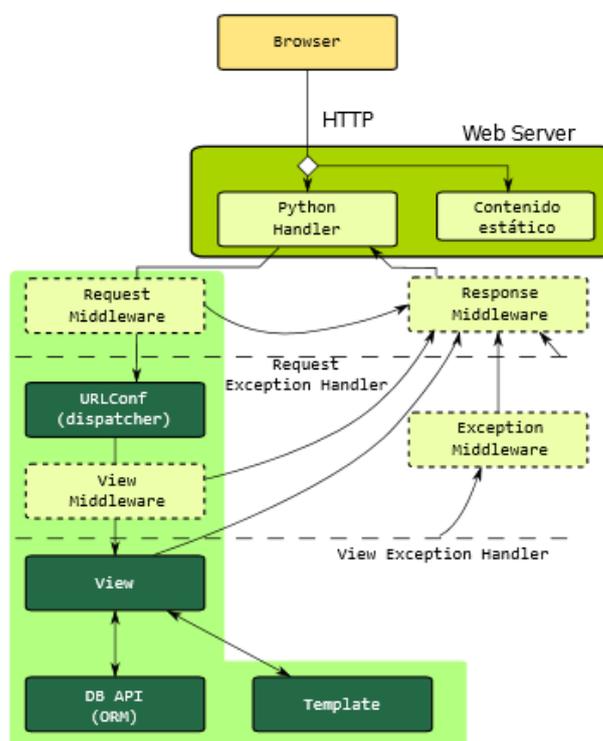


Figura 22. Esquema de Flujo de una aplicación Django.

Tomado de (Defossé & Marcos, 2009)

2.3.2.2.5. Symfony

“Symfony es un conjunto de componentes PHP para desarrollar aplicaciones web, una filosofía de trabajo y una comunidad que funcionan conjuntamente en armonía.” (symfony.com, 2011) Además de ser compatible con la mayoría de gestores de base de datos, entre ellos están Microsoft SQL Server, Oracle, PostgreSQL y MySQL. También como punto adicional su ejecución sobre las plataformas como Linux y Microsoft es factible.

Symfony es un *framework* que permite construir aplicaciones *web*, mantiene además un conjunto de herramientas y utilidades que simplifican el desarrollo de la aplicación *web*.

2.3.2.2.5.1. Características y funciones

Considerado como “(...) un conjunto de servicios y operaciones disponibles bajo un determinado nombre de dominio y que comparten el mismo modelo de objetos”. (Fabien Potencier) Dentro de un proyecto; se puede obtener operaciones que se agrupan de manera lógica normalmente, una aplicación se ejecuta de forma independiente respecto de otras aplicaciones del mismo proyecto.

- Lo habitual es que un proyecto contenga dos aplicaciones: → Parte pública → Parte de gestión
- Formado por uno o más módulos, de los cuales un módulo normalmente representa a una página *web* o a un grupo de páginas con un propósito relacionado.
- Los módulos almacenan las acciones, que representan cada una de las operaciones que se puede realizar en un módulo. Por ejemplo, el módulo “carritoCompra” puede definir acciones como añadir, mostrar y actualizar. (Llonch, 2008)

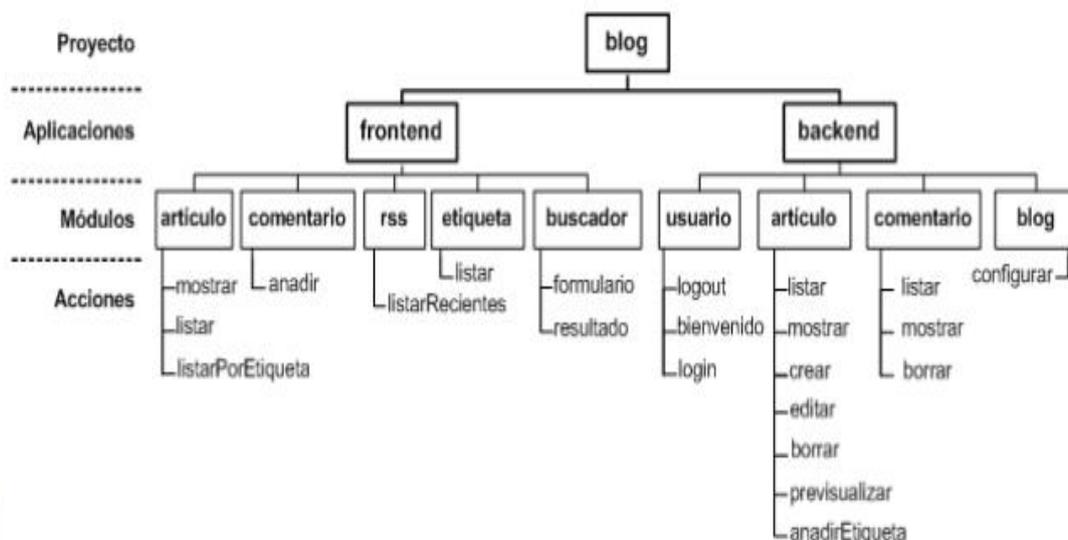


Figura 23. Proyecto con Symfony.

Tomado de (Llonch, 2008)

2.3.2.2.6. Laravel

Laravel es un *framework* de aplicaciones para PHP 5.3 escrito por Taylor Otwell. Fue escrito con las características de PHP 5.3 en mente. La combinación de esas características y su expresiva sintaxis le han granjeado al *framework* bastante popularidad. (Rees & Laguna, 2012) Permite trabajar con bases de datos como MySQL, Postgres, SQLite y SQL Server.

Cuenta con mucha energía, documentación atractiva y con alto contenido, además de ofrecer la funcionalidad necesaria para el desarrollo de aplicaciones totalmente interactivas, modernas y lo realiza de manera fácil y segura.

2.3.2.2.6.1. Características y funciones

- Ser modular y extensible ya que permite agregar toda la información que se requiera través de su directorio Packalyst; donde es posible ver más de 5.500 paquetes.
- Micro servicios: A Laravel se le considera como un enfoque con lean development, es decir permite el desarrollo de micro servicios, y APIs de rendimiento óptimo de los proyectos.

- Caché: Las aplicaciones contarán con un sistema robusto en el que es posible ajustar para que las aplicaciones se carguen de manera rápida.
- Autenticación: En cuanto a seguridad se refiere Laravel puede implementar autenticación de manera nativa, e incluso incluye la opción de recordar al usuario.
- Tareas Automatizadas: Gracias al API Elixir, Laravel que permite definir tareas de Gulp en donde es posible ver el uso de preprocesadores para comprimir.
- Eventos: “Definir, registrar y escuchar eventos en tu aplicación es muy sencillo. Mediante una lista de todos los eventos registrados en tu aplicación en un evento. Usa como llave el nombre del evento y como valor los *listeners* registrados a él”. (Anton, 2011)

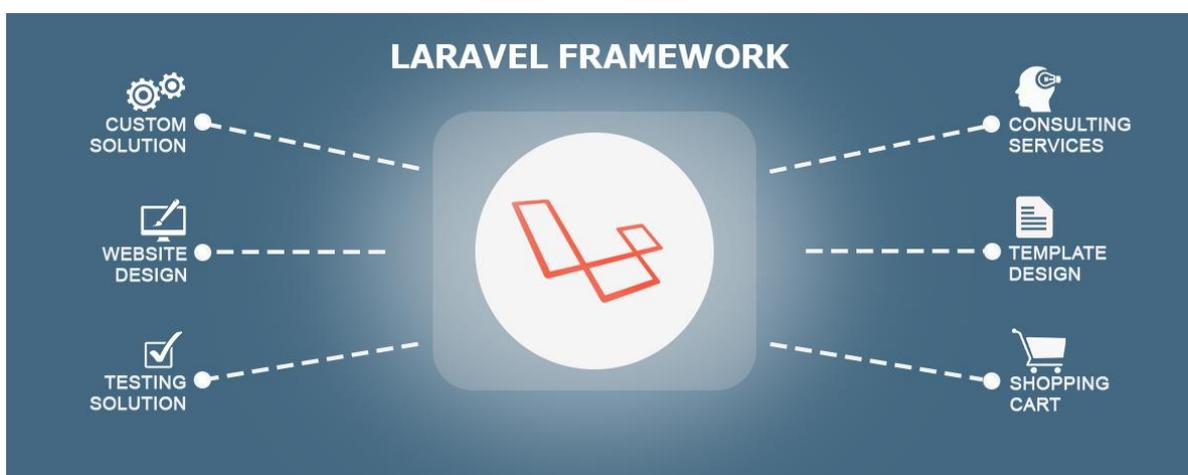


Figura 24. Framework Laravel

Tomado de (Anton, 2011)

2.3.5. Software móvil

2.3.5.1. Android

“Basado en Linux enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es

desarrollado por *la Open Handset Alliance*, liderada por Google. Las aplicaciones se desarrollan habitualmente en el lenguaje Java con *Android Software Development Kit (Android SDK)*, Existen otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C, C++ u otros lenguajes de programación.” (Garrido, 2013, pág. 6)

Se estiman que existen más de 700.000 aplicaciones que en la actualidad pueden ser compatibles con la plataforma de Android, y de ellas 1´000.000 de teléfonos móviles se activan diariamente.

2.3.5.1.1. Características de android

Sus aplicaciones están orientadas a objetos, ya que es posible acceder al núcleo de la biblioteca de Java, también ocupa bibliotecas en lenguaje C, bases de datos relacionales como SQLite, interfaz API gráfica, motor gráfico SGL, una biblioteca estándar de C Bionic. (Garrido, 2013, pág. 6)

Cuenta con una interfaz amigable que brinda Android convirtiéndolo en adecuado para la elaboración de proyectos, a continuación, se presenta un gráfico donde se puede apreciar las diferentes aplicaciones y librerías disponibles en Android:

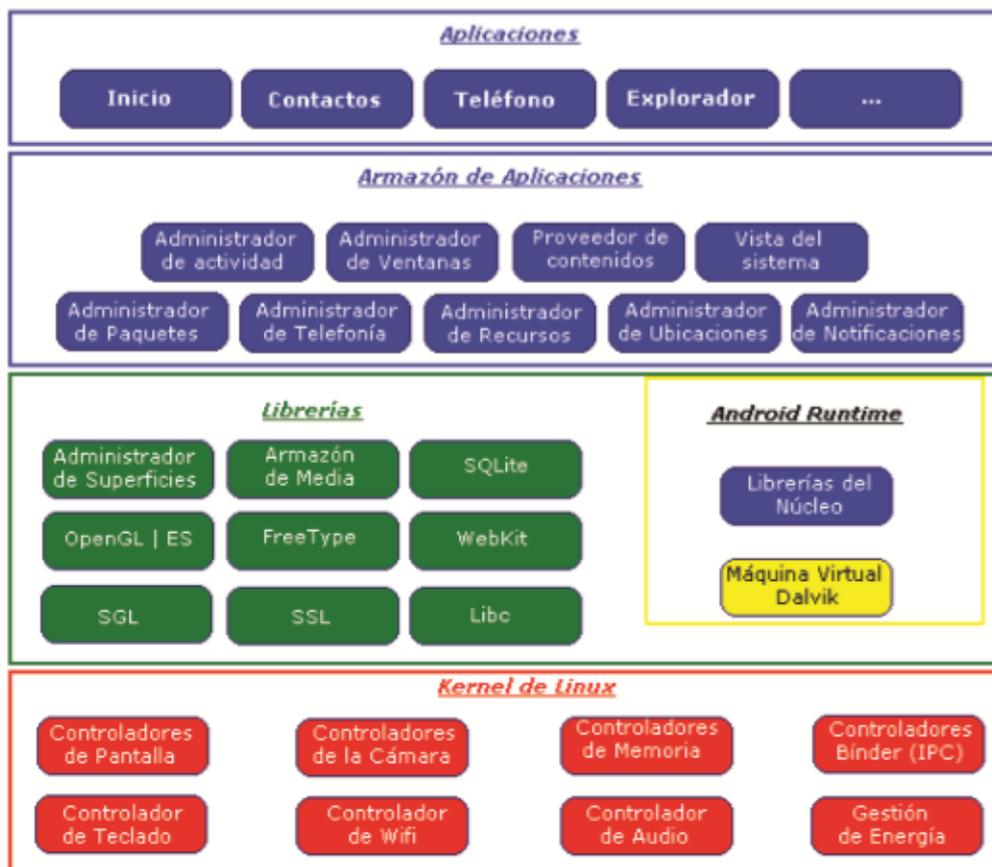


Figura 25. Aplicaciones y Librerías disponibles Android.

Tomado de (Báez, y otros, 2011)

En Android cada aplicación utiliza un proceso propio, con un ID propio conocido como un identificador para que solo dicha aplicación tenga acceso a sus archivos.

“Los dispositivos tienen un único foco, la ejecución principal, que es la aplicación que está visible en la pantalla, pero puede tener varias aplicaciones en un segundo plano, cada una con su propia pila de tareas. La pila de tareas es la secuencia de ejecución de procesos en Android. Se componen de actividades que se van apilando según son invocadas, y solo pueden terminarse cuando las tareas que tiene encima están terminadas, o cuando el sistema las destruye porque necesita memoria, por lo que tienen que estar preparadas para terminar en cualquier momento. El sistema siempre eliminará la actividad que lleve más tiempo parada. En caso de que el sistema necesitase mucha memoria, si la

aplicación no está en el foco, puede ser eliminada por completo a excepción de su actividad principal.” (Sanz, 2013)

A continuación, una descripción gráfica sobre la plataforma de actividades secuenciales generada en multitarea:



Figura 26. Actividad en capas.

Tomado de Secuencia de actividades en Android (Báez, y otros, 2011)

2.3.6. Conclusión sobre las tecnologías de desarrollo

En base a las tecnologías presentadas en este documento, el desarrollo del aplicativo *web* propuesto fue realizado mediante el marco de trabajo Laravel que permite integrar PHP para la parte de programación, MySQL para la gestión y uso de base de datos y HTML5 para la integración del aplicativo *web*.

2.3.7. Conclusión parcial

Al ocupar las tecnologías de desarrollo previamente indicadas, se logró obtener un aplicativo *web* que no discrimina el tipo de navegador que se desee usar, popular por el marco de trabajo en el que está desarrollado, con una amplia gama de soporte en línea y de fácil edición por el lenguaje de programación.

3. DESARROLLO DE LA PROPUESTA

Como resultado del desarrollo de la propuesta queda en evidencia la documentación generada al emplear la metodología *SCRUM*, es importante clarificar que los roles existentes en dicha metodología son cubiertos todos por el autor de este trabajo, dicha información está compuesta del *product backlog*,

el cual es una lista de requisitos priorizada sobre los objetivos y entregas del proyecto.

Adicionalmente se anexan los *sprints* que detalla la forma de la ejecución mediante bloques temporales y el *sprint backlog*, el cual contiene las tareas elaboradas que fueron planificadas para cada iteración.

Se ha agregado imágenes ilustrativas sobre el aplicativo *web* y la aplicación móvil, de igual manera el diagrama de entidad-relación sobre el modelo de la base de datos ocupado para el aplicativo.

3.1. Base de datos

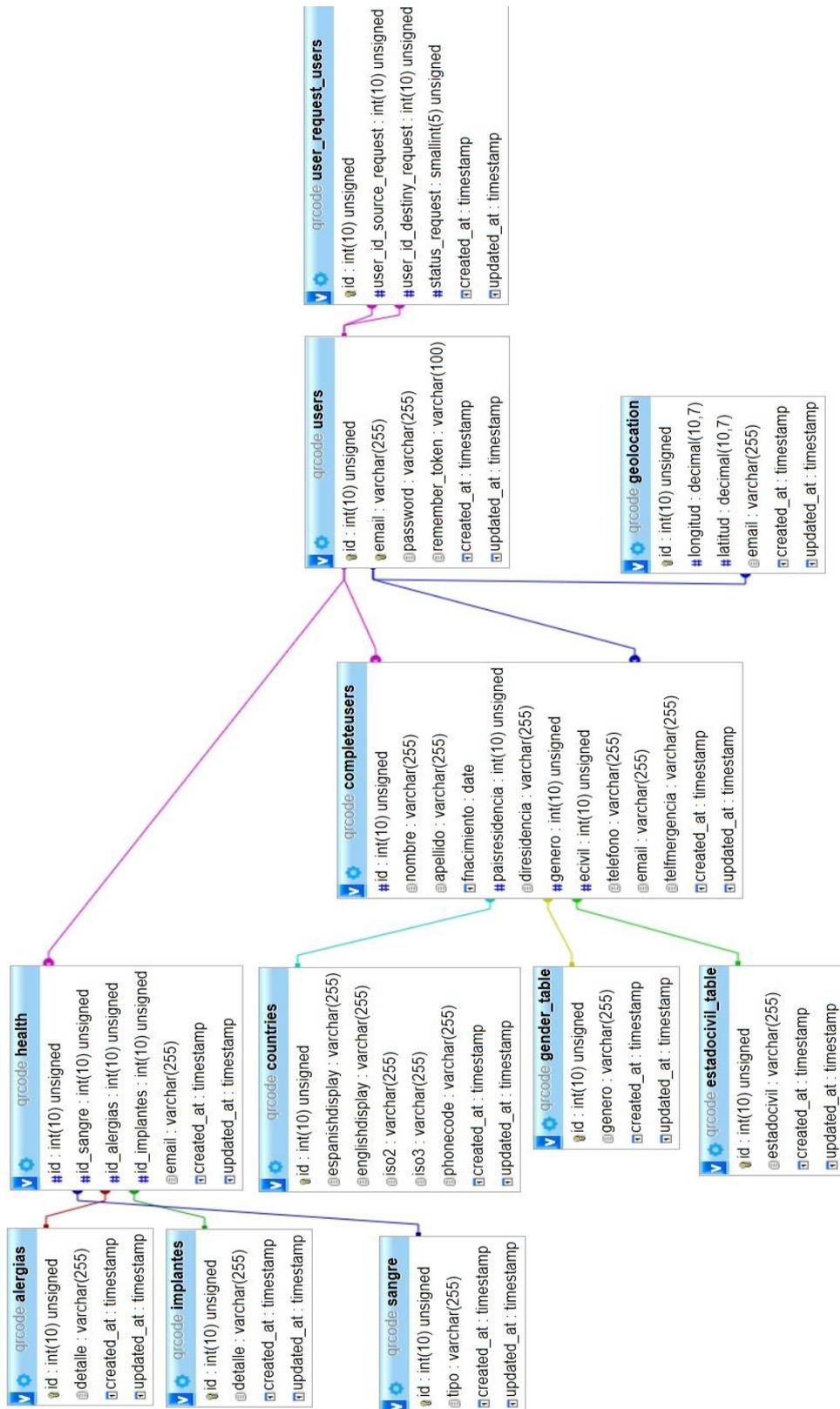


Figura 27. Tabla entidad-relación

3.2. Product backlog

Tabla 4.

Product Backlog

Título: Product Backlog						
ID	Prioridad	Nombre	Historia de Usuario	Criterio de Validación	Estimado	Por
1 C U 01	Muy Alta	Creación de Usuario	Como usuario me gustaría poder generar mi cuenta donde pueda administrar las propiedades que sean posibles bajo mi usuario.	El usuario debe requerir de un email y un password con los requerimientos básicos de validación como son 6 caracteres.	30	AD
2 M U 01	Alta	Definición de campos generales de usuario	Como usuario, tras crear mi cuenta me gustaría poder ingresar mis datos generales para ser visualizados e n caso de que se acceda a	Debido a que no todos los usuarios les interesan ingresar sus datos generales, no es mandatorio al momento de crear la cuenta, sino es opcional, sin embargo, si se desea agregar información a los campos generales, es mandatorio llenarlos todos, de los	60	AD

			ellos por otro usuario o mi persona	<p>cuales son los siguientes:</p> <ul style="list-style-type: none"> -Nombres -Apellidos -Fecha de nacimiento -País de Residencia -Género -Estado Civil -Dirección de Residencia -Teléfono personal -Teléfono de emergencia 		
2 M U 02	Alta	Definición de campos de salud de usuario	<p>Como usuario, tras crear mi cuenta me gustaría poder ingresar mis datos relacionados a mi salud para ser visualizados en caso de que se acceda a ellos por otro usuario o mi persona</p>	<p>Debido a que no todos los usuarios les interesan ingresar sus datos de salud, no es mandatorio al momento de crear la cuenta, sino es opcional, sin embargo, si se desea agregar información a los campos de salud, es mandatorio llenarlos todos, de los cuales son los siguientes:</p> <ul style="list-style-type: none"> -Implantes -Alergias -Tipo de sangre 	60	AD
3 Q R 01	Muy Alta	Módulo de código QR	<p>Como usuario, al momento de registrarme, debo contar con un código QR</p>	<p>Los códigos QR serán dinámicos y generados a partir de la creación de un perfil</p>	70	AD

		<p>dinámico ya asignado a mi cuenta de manera dinámica.</p> <p>La funcionalidad de este código, es la de brindar una vista a mis datos generales y salud a aquellos que lean el código QR</p>				
4 R C 01	Muy Alta	<p>Definición de relación entre usuarios y utilidad</p>	<p>Como usuario, se debería poder agregar contactos para poder visualizar cualquier tipo de reporte en la aplicación sobre ese usuario, sin embargo, para ello debería ser en base a una solicitud que solo funciona habilitando al aplicante, más no ambos</p>	<p>Determinar la funcionalidad de una solicitud de contacto para permitir observar reportes del usuario que acepte la solicitud de contacto.</p> <p>Hay dos planteamientos funcionales:</p> <p>Usuario registrado -> solicita acceso a reportes generados sobre usuario que recepta y acepta solicitud</p> <p>Usuario registrado -></p>	60	AD

			usuarios (aplicante y receptor).	solicita acceso a reportes generados y otorga acceso sobre usuario que recepta y acepta solicitud		
4 R C 02	Muy Alta	Reportes de Geolocalización	Como usuario o visitante, debería poder reportar la ubicación de la lectura del código QR que a su vez está anexo a un usuario registrado a dicho código.	El visitante o usuario podrá ingresar al perfil del código QR que lea y reportar su ubicación, otorgando permisos de ubicación en el dispositivo que realice la lectura del código.	65	AD
5 A P 01	Media	APP Lector de códigos QR	Como portador de teléfono inteligente con acceso a internet debo poder hacer la lectura de los códigos QR en general con la finalidad de contar con una app propia	El visitante o usuario podrá ingresar al perfil del código QR que lea desde la APP	30	AD

3.3. Sprints

Tabla 5.

Sprint #1

Título: Sprint #1				Fecha 01/04/2016 02/05/2016	Ejecución: –
Categoría	ID hist	ID Tarea	Tarea	Responsable	Estimado Puntos
Análisis	1CU01	1CU01T1	Análisis de campos requeridos para la creación de usuario	AD	5
Desarrollo	1CU01	1CU01T2	Desarrollo de login de usuario	AD	15
Pruebas	1CU01	1CU01T3	Pruebas de creación e ingreso de usuario	AD	5
Deploy	1CU01	1CU01T4	Aplicar al proyecto y generar respaldo de proyecto	AD	5
Análisis	2MU01	2MU01T1	Definición de tipos de datos y campos	AD	5
Desarrollo	2MU01	2MU01T2	Incorporación en HTML y PHP para CRUD de proyecto	AD	45

Pruebas	2MU01	2MU01 T3	Pruebas de CRUD en proyecto sobre usuarios	AD	5
Deploy	2MU01	2MU01 T4	Aplicar al proyecto y generar respaldo de proyecto	AD	5
Análisis	2MU02	2MU02 T1	Desarrollo de tipos de datos y campos en salud	AD	5
Desarrollo	2MU02	2MU02 T2	Incorp en HTML y PHP para CRUD de proyecto	AD	45
Pruebas	2MU02	2MU02 T3	Pruebas de CRUD en proyecto sobre usuarios	AD	5
Deploy	2MU02	2MU02 T4	Aplicar al proyecto y generar respaldo de proyecto	AD	5

Tabla 6.

Sprint Backlog #1

Sprint Backlog #1						PROYECTO			
Categoría	ID	ID Tarea	Tarea	Responsible	Estimado en Puntos	Estado	INICIO	FIN	JORNADA Pts.
Historia	1CU01	T1	Análisis de campos requeridos para la	AD	5	Completo	abr-05	abr-05	150
Desarrollo	1CU01	T2	Desarrollo de login de usuario	AD	15	Completo	abr-06	abr-06	150
Pruebas	1CU01	T3	Pruebas de creación e ingreso de usuario	AD	5	Completo	abr-07	abr-07	150
Deploy	1CU01	T4	Aplicar al proyecto y generar respaldo de	AD	5	Completo	abr-08	abr-08	150
Análisis	2MU01	T1	Definición de tipos de datos y campos	AD	5	Completo	abr-09	abr-09	150
Desarrollo	2MU01	T2	Incorporación en HTML y PHP para CRUD de	AD	45	Completo	abr-10	abr-10	150
Pruebas	2MU01	T3	Pruebas de CRUD en proyecto sobre usuarios	AD	5	Completo	abr-11	abr-11	150
Deploy	2MU01	T4	Aplicar al proyecto y generar respaldo de	AD	5	Completo	abr-12	abr-12	150
Análisis	2MU02	T1	Desarrollo de tipos de datos y campos en salud	AD	5	Completo	abr-13	abr-13	150
Desarrollo	2MU02	T2	Incorporación en HTML y PHP para CRUD de	AD	45	Completo	abr-14	abr-14	150
Pruebas	2MU02	T3	Pruebas de CRUD en proyecto sobre usuarios	AD	5	Completo	abr-15	abr-15	150
Deploy	2MU02	T4	Aplicar al proyecto y generar respaldo de	AD	5	Completo	abr-16	abr-16	150

Esfuerzo Pendiente			Progreso Sprint		
Tareas Pendientes	Esfuerzo Pendiente	Progreso Sprint	Tareas Pendientes	Esfuerzo Pendiente	Progreso Sprint
	12	148		12	148
	11	140		11	140
	11	135		11	135
	11	130		11	130
	9	125		9	125
	9	125		9	125
	8	120		8	120
	7	115		7	115
	7	110		7	110
	7	105		7	105
	7	100		7	100
	7	95		7	95
	7	90		7	90
	7	85		7	85
	7	80		7	80
	7	75		7	75
	7	70		7	70
	5	65		5	65
	4	60		4	60
	3	55		3	55
	3	45		3	45
	3	35		3	35
	3	25		3	25
	3	16		3	16
	2	8		2	8
	1	4		1	4
	0	0		0	0
	0	0		0	0

Tabla 7.

Sprint #2

Título: Sprint #2				Fecha Ejecución: 01/04/2016 – 02/05/2016	
Cate goría	ID historia	ID Tarea	Tarea	Respon sa ble	Estimado en Puntos
Análi sis	3QR01	3QR01 T1	Establecer mejor método sobre código QR dinámico en proyecto	AD	20
Desa rrollo	3QR01	3QR01 T2	Incorporación de módulo de QR dinámico a proyecto	AD	35
Prue bas	3QR01	3QR01 T3	Realizar pruebas de URLs y acceso a perfiles con el código	AD	10
Deplo y	3QR01	3QR01 T4	Actualizar proyecto y respaldo del mismo	AD	5
Análi sis	4RC01	4RC01 T1	Decidir entre estilo de solicitud y como establecer relación	AD	10

Desarrollo	4RC01	4RC01 T2	Incorporación en HTML, PHP y BDD	AD	40
Pruebas	4RC01	4RC01 T3	Pruebas sobre usuarios creados de prueba	AD	5
Deploy	4RC01	4RC01 T4	Incorporar al proyecto y actualizar respaldo de proyecto	AD	5
Análisis	4RC02	4RC02 T1	Definir forma y tipo de reporte	AD	10
Desarrollo	4RC02	4RC02 T2	Geolocalización para usuarios	AD	40
Pruebas	4RC02	4RC02 T3	Pruebas de ubicación en base a evento generado de botón	AD	10
Deploy	4RC02	4RC02 T4	Actualización de proyecto y respaldo con nuevo código	AD	5
Desarrollo	5AP01	5AP01 T1	Desarrollo de aplicativo lector de códigos QR	AD	30

Tabla 8.

Sprint Backlog #2

Categoría	ID historia	ID Tarea	Tarea	Respon sable	Estimado en Puntos	Estado	PROYECTO		
							INICIO	FIN	JORNADA Pis.
							Mayo 3, 2016	Junio 5, 2016	225
Sprint Backlog #2							Tareas Pendientes		
Estuerzo Pendiente							189	13	may-03
Progreso Sprint							183	13	may-04
Progreso Sprint							177	13	may-05
Progreso Sprint							171	12	may-06
Progreso Sprint							165	12	may-07
Progreso Sprint							159	12	may-08
Progreso Sprint							153	12	may-09
Progreso Sprint							147	12	may-10
Progreso Sprint							140	12	may-11
Progreso Sprint							134	12	may-12
Progreso Sprint							128	9	may-13
Progreso Sprint							122	8	may-14
Progreso Sprint							116	8	may-15
Progreso Sprint							110	7	may-16
Progreso Sprint							114	7	may-17
Progreso Sprint							108	7	may-18
Progreso Sprint							102	7	may-19
Progreso Sprint							96	7	may-20
Progreso Sprint							90	7	may-21
Progreso Sprint							84	7	may-22
Progreso Sprint							78	7	may-23
Progreso Sprint							72	7	may-24
Progreso Sprint							66	5	may-25
Progreso Sprint							60	4	may-26
Progreso Sprint							54	3	may-27
Progreso Sprint							49	3	may-28
Progreso Sprint							43	3	may-29
Progreso Sprint							37	3	may-30
Progreso Sprint							31	3	may-31
Progreso Sprint							23	3	jun-01
Progreso Sprint							17	3	jun-02
Progreso Sprint							11	3	jun-03
Progreso Sprint							5	1	jun-04
Progreso Sprint							0	0	jun-05
Desarrollo	5AP01	5AP01 T1	Desarrollo de aplicativo lector de códigos QR	AD	30	Completo	30	5	
Deploy	4RC02	4RC02 T4	Actualización de proyecto y respaldo con nuevo	AD	5	Completo	27	5	
Pruebas	4RC02	4RC02 T3	base a evento generado	AD	10	Completo	24	5	
Desarrollo	4RC02	4RC02 T2	Geolocalización para usuarios	AD	40	Completo	21	5	
Análisis	4RC02	4RC02 T1	Definir forma y tipo de reporte	AD	10	Completo	18	5	
Deploy	4RC01	4RC01 T4	Incorporar al proyecto y actualizar respaldo de	AD	5	Completo	15	5	
Pruebas	4RC01	4RC01 T3	Pruebas sobre usuarios creados de prueba	AD	5	Completo	12	5	
Desarrollo	4RC01	4RC01 T2	Incorporación en HTML, PHP y BDD	AD	40	Completo	9	5	
Análisis	4RC01	4RC01 T1	Decidir entre estilo de solicitud y como	AD	10	Completo	6	5	
Deploy	3QR01	3QR01 T4	Actualizar proyecto y respaldo del mismo	AD	5	Completo	5	5	
Pruebas	3QR01	3QR01 T3	Realizar pruebas de URLs y acceso a perfiles con el	AD	10	Completo	4	5	
Desarrollo	3QR01	3QR01 T2	Incorporación de módulo de QR dinámico a	AD	35	Completo	3	5	
Análisis	3QR01	3QR01 T1	Establecer mejor método sobre código QR	AD	20	Completo	2	5	

3.4. Capturas del desarrollo

3.4.1. Pantalla de inicio o bienvenida

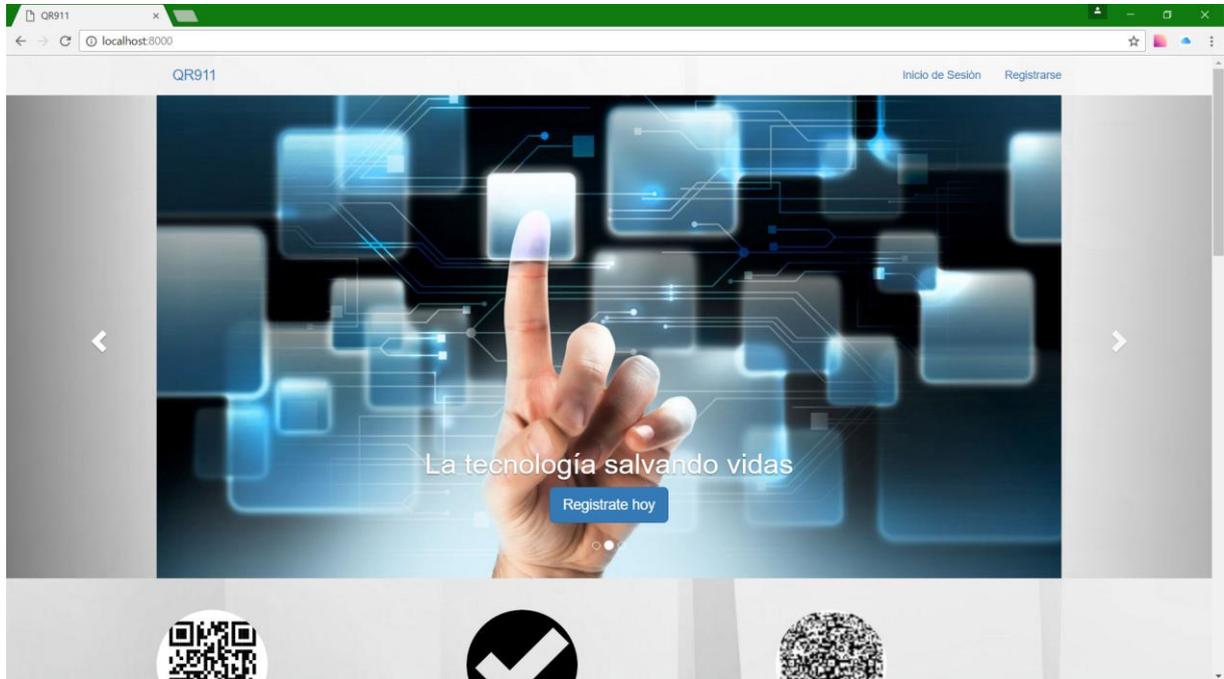


Figura 28. Pantalla de inicio o bienvenida

3.4.2. Pantalla de registro de nuevo usuario

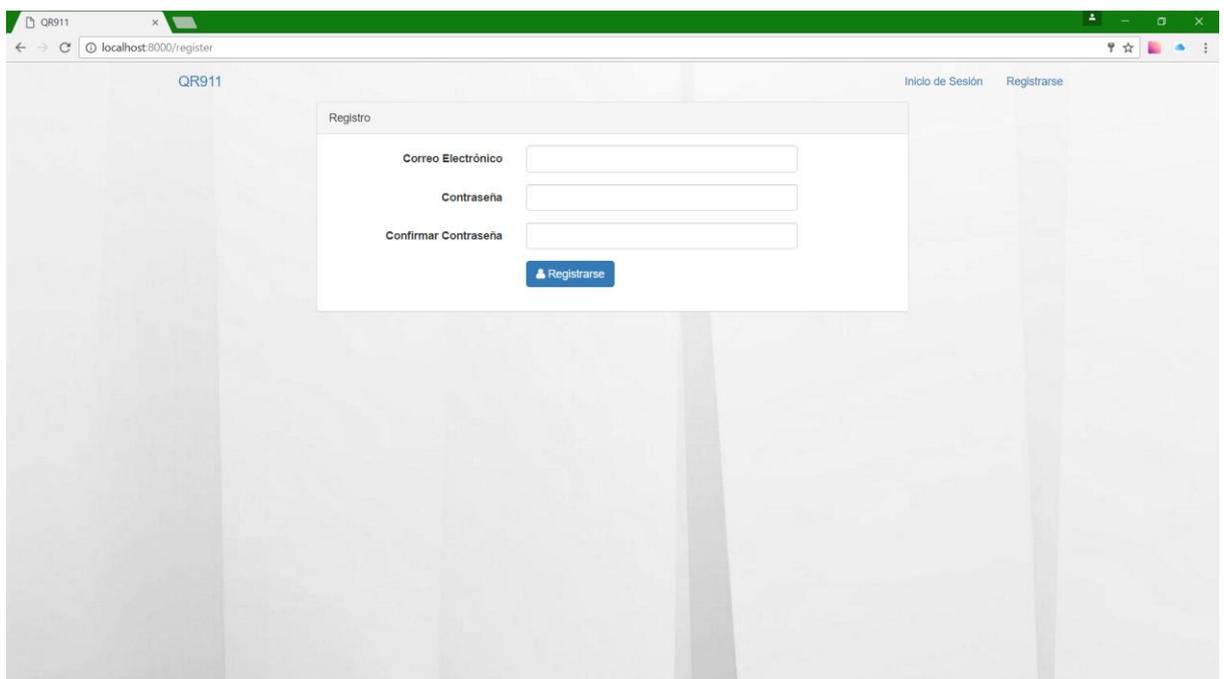


Figura 29. Pantalla de registro de nuevo usuario

3.4.3. Pantalla de inicio de sesión

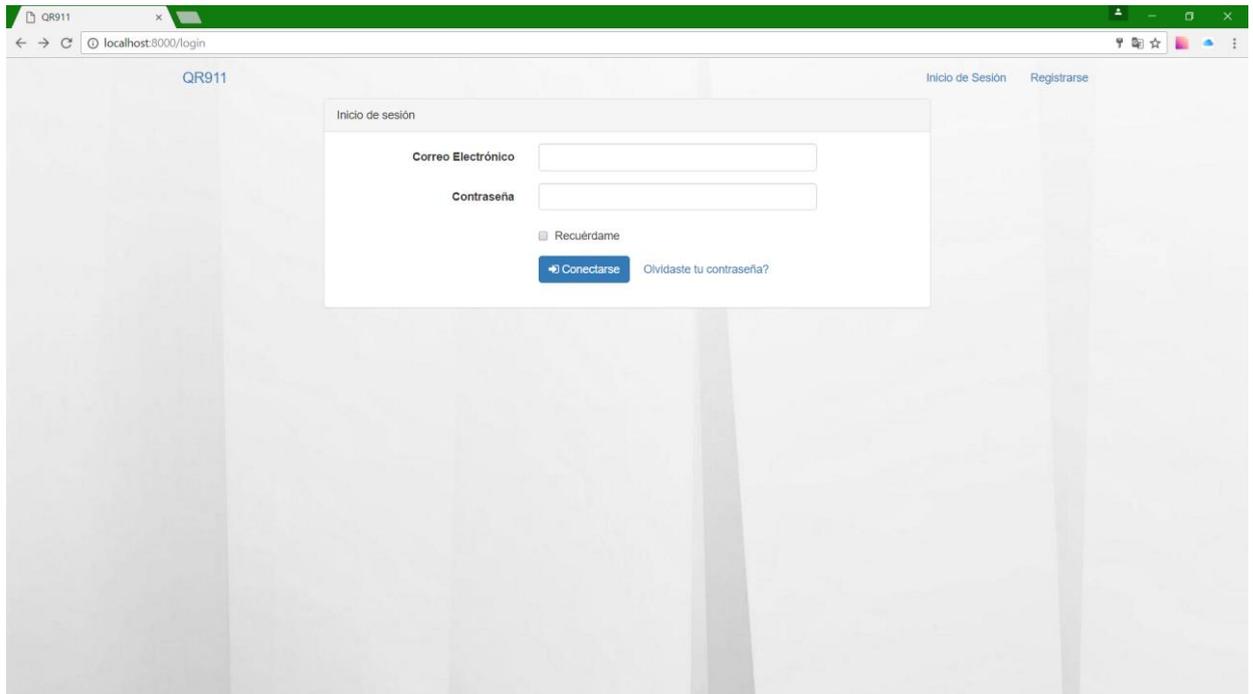


Figura 30. Pantalla de inicio de sesión

3.4.4. Pantalla de usuario conectado

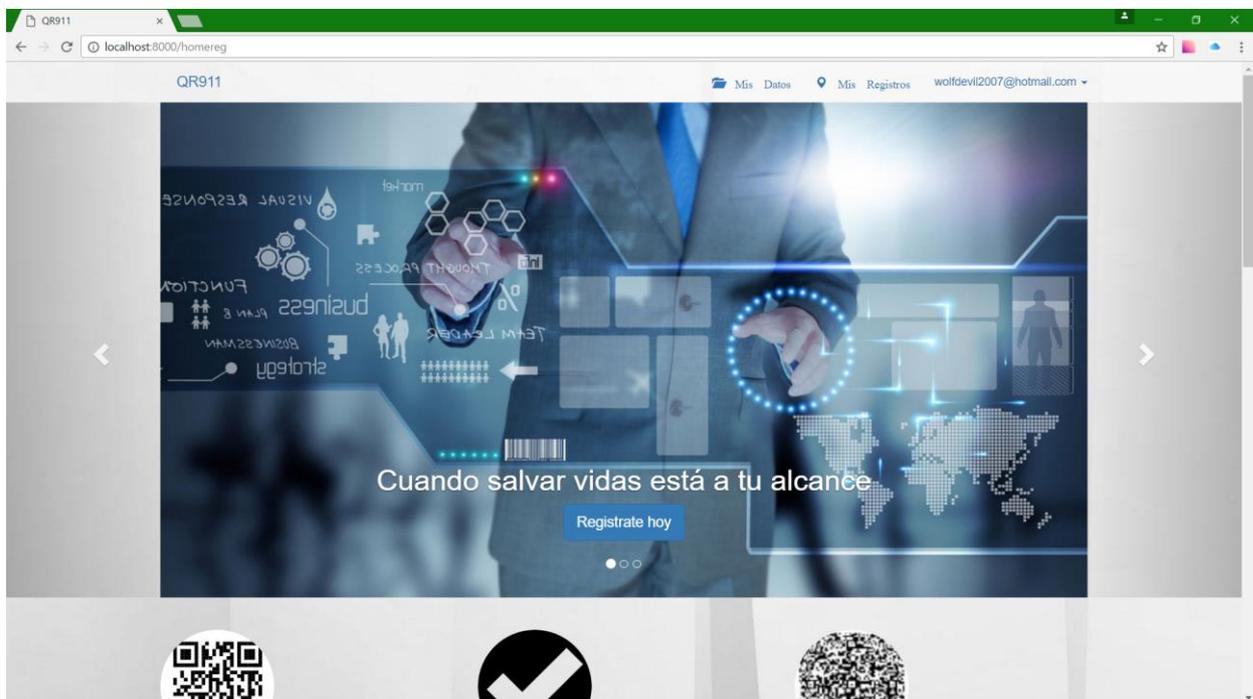


Figura 31. Pantalla de usuario conectado

3.4.5. Pantalla de presentación de datos de usuario

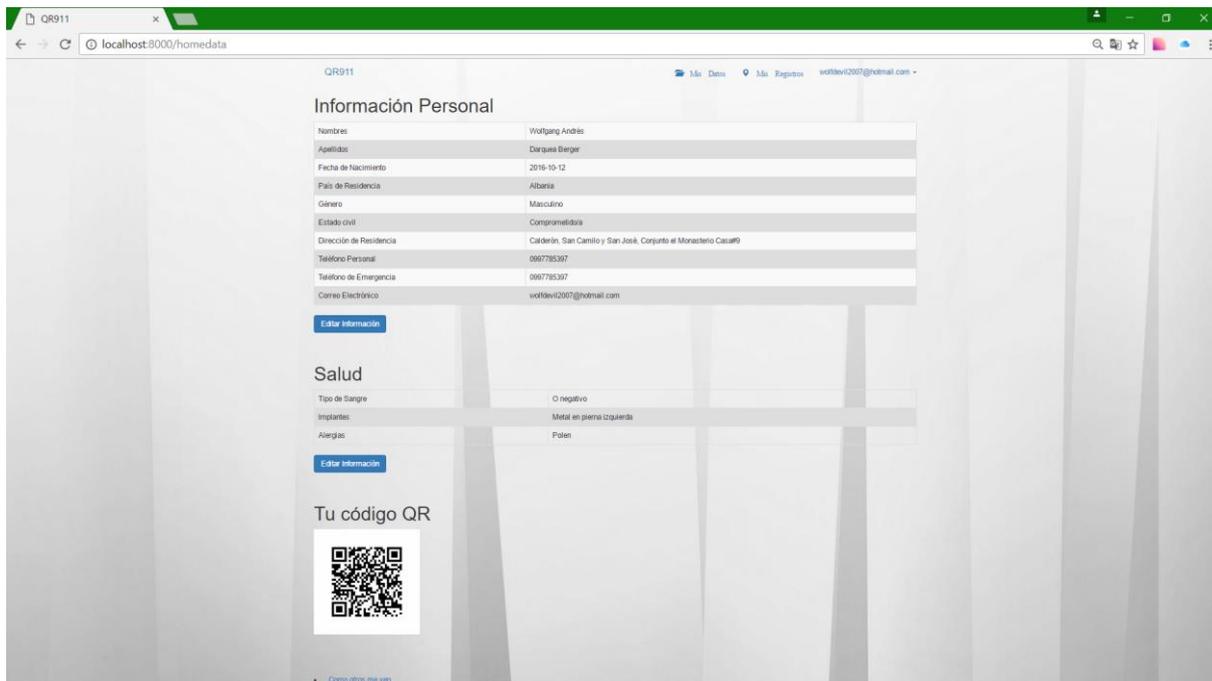


Figura 32. Pantalla de presentación de datos de usuario

3.4.6. Pantalla de ingreso visitante de perfil y creación de reporte

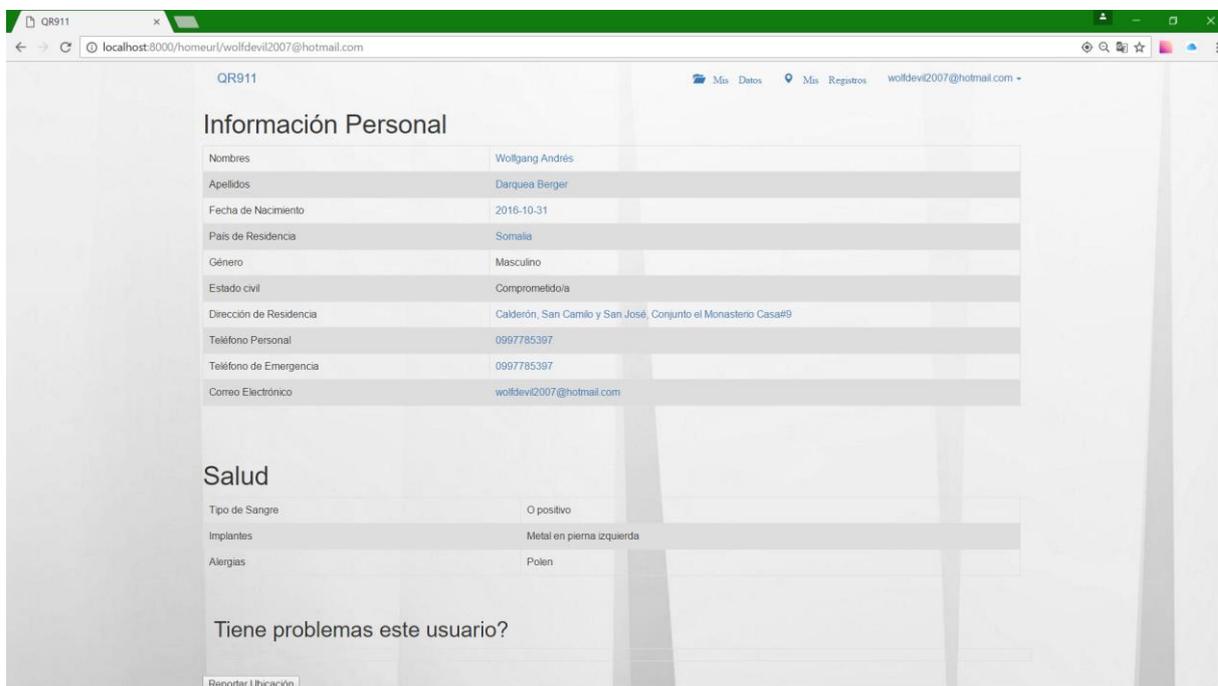


Figura 33. Pantalla de ingreso para visitante de perfil y creación de reporte

3.4.7. Pantalla de selección sobre reporte de geolocalización de perfil personal y conocidos

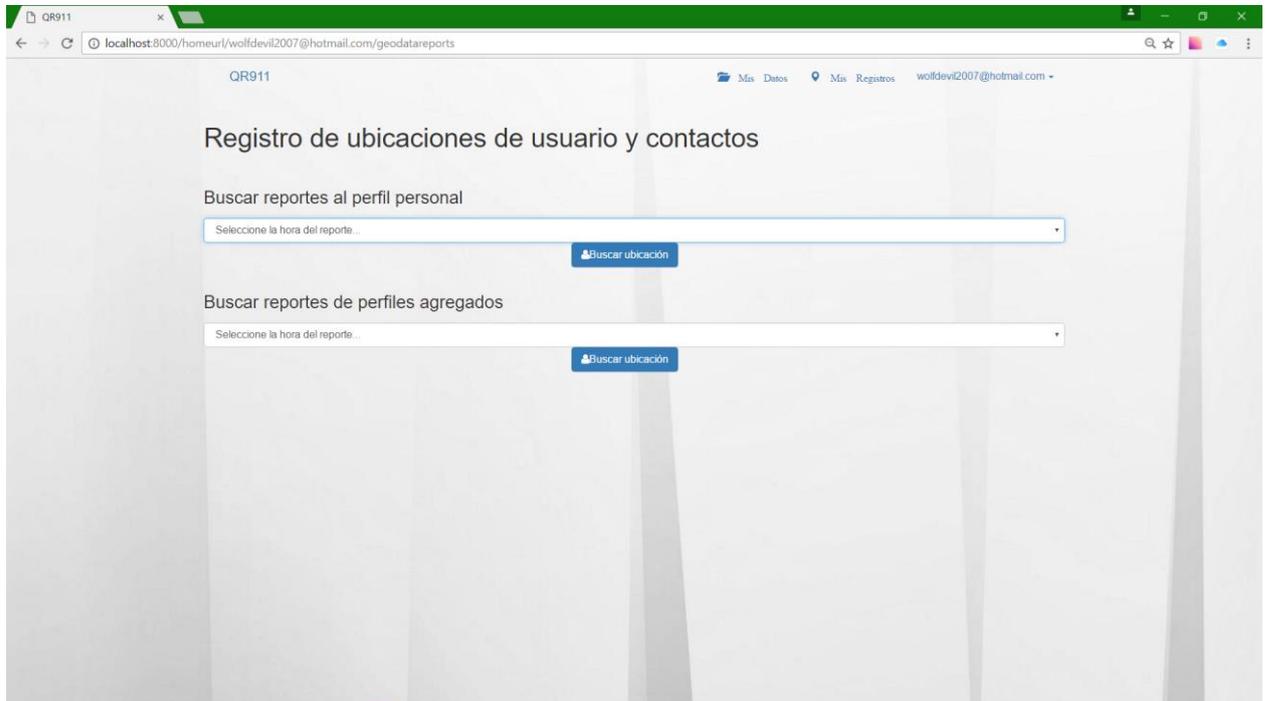


Figura 34. Pantalla de selección sobre reporte de geolocalización de perfil personal y conocidos

3.4.8. Pantalla de visualización de registro de evento

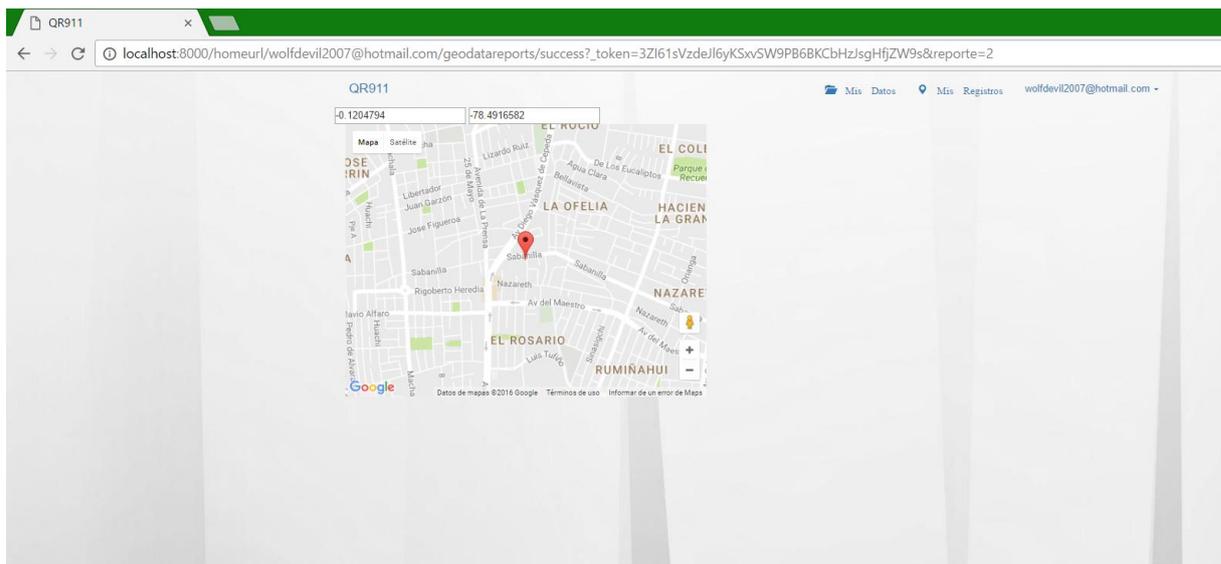


Figura 35. Pantalla de visualización de registro de evento

3.4.9. Pantalla manejo del perfil en peticiones y contactos

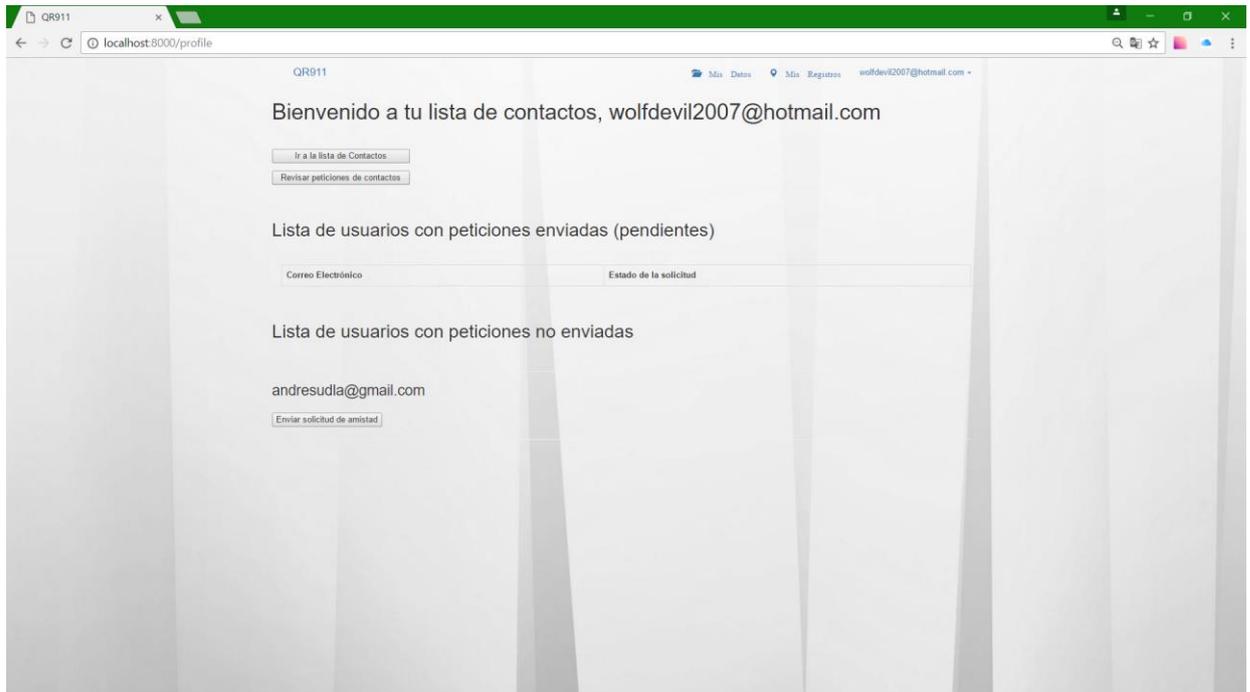


Figura 36. Pantalla para manejo del perfil social como peticiones y contactos

3.4.10. Pantalla para manejo de solicitudes de contacto

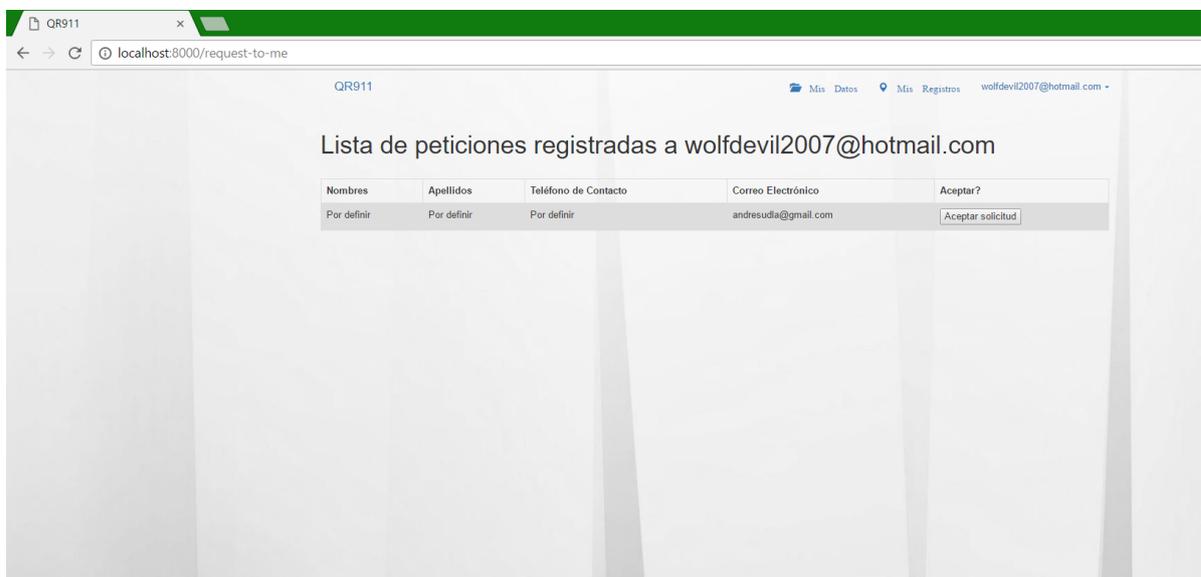


Figura 37. Pantalla para manejo de solicitudes de contacto

3.4.11. Pantalla de contactos registrados

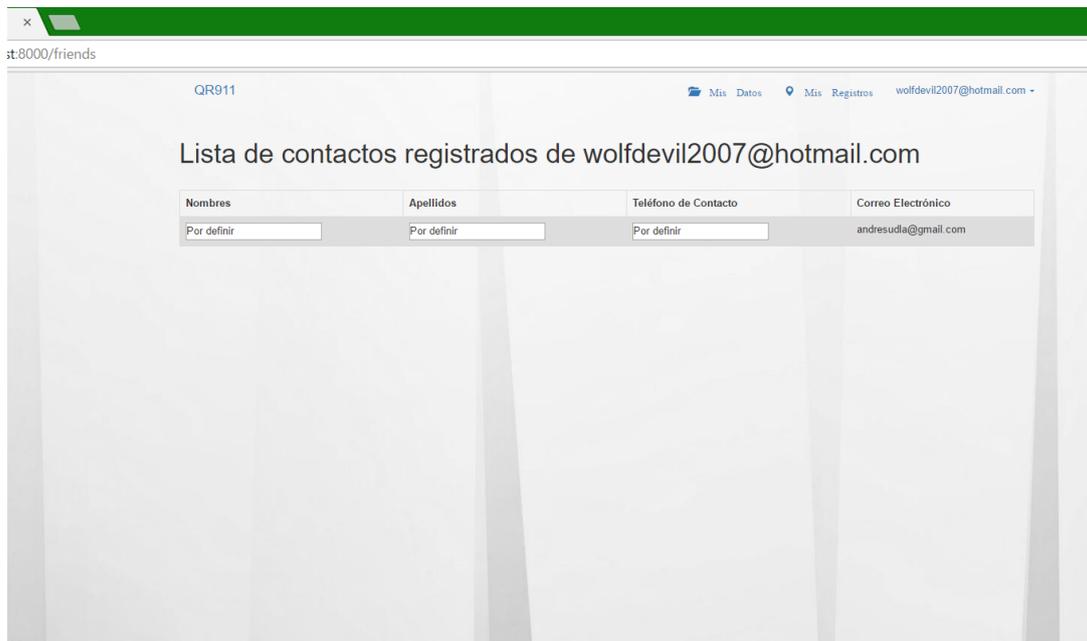


Figura 38. Pantalla de contactos registrados

3.4.12. Pantalla ingreso de nueva información general

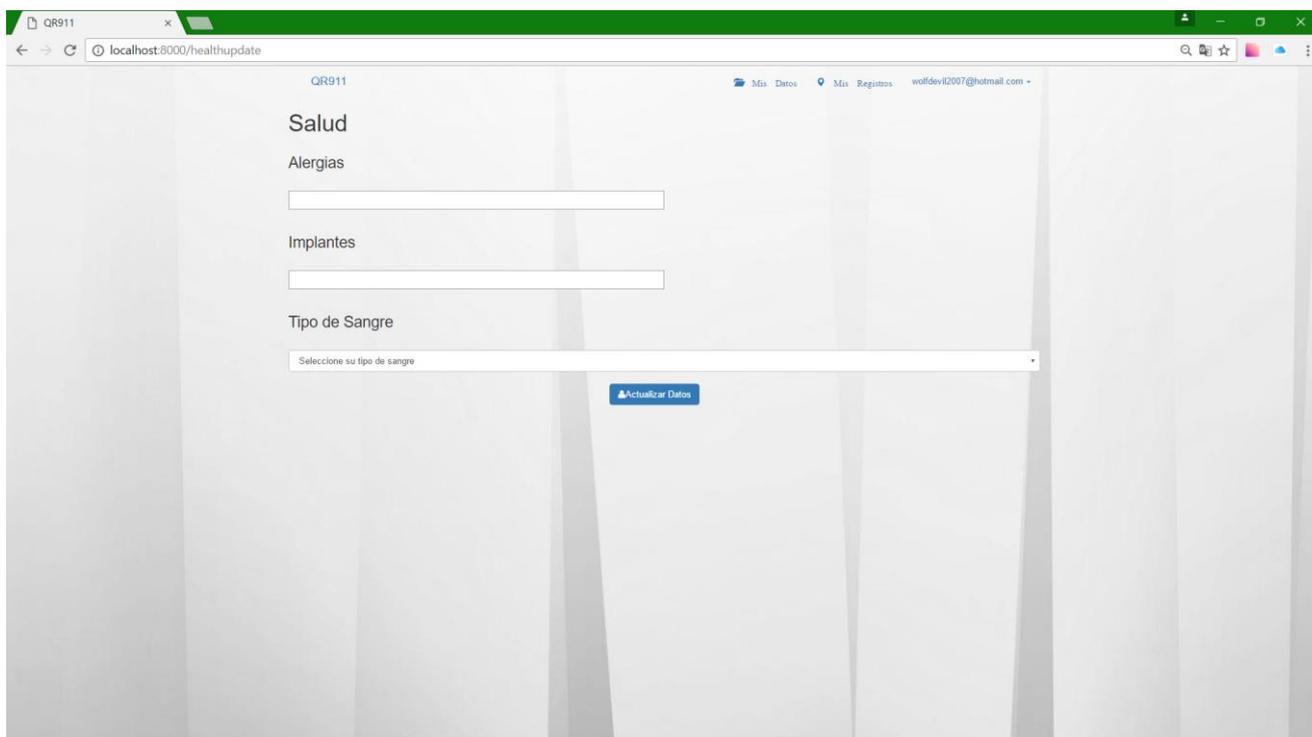
The screenshot shows a web browser window with the address bar displaying 'localhost:8000/updatedata'. The page title is 'QR911'. The main heading is 'Información General'. The form contains the following fields:

- Nombres:
- Apellidos:
- País de Residencia:
- Género:
- Estado Civil:
- Fecha de Nacimiento:
- Dirección de Residencia:
- Teléfono Personal:
- Teléfono de Emergencia:

At the bottom right of the form is a blue button labeled 'Actualizar Datos'.

Figura 39. Pantalla para ingreso de nueva información general

3.4.13. Pantalla ingreso de nueva información de salud



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/healthupdate'. The page title is 'QR911'. The main content area is titled 'Salud' and contains three input fields: 'Alergias', 'Implantes', and 'Tipo de Sangre'. The 'Tipo de Sangre' field is a dropdown menu with the text 'Seleccione su tipo de sangre'. A blue button labeled 'Actualizar Datos' is positioned below the form fields. The browser's address bar and navigation icons are visible at the top.

Figura 40. Pantalla para ingreso de nueva información de salud

3.4.14. Pantalla de aplicativo android para lectura de código QR

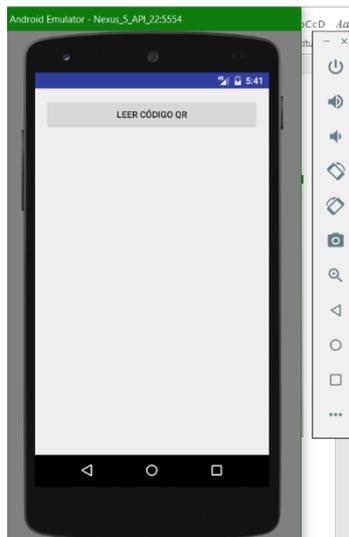


Figura 41. Pantalla de aplicativo Android para lectura de código QR

3.4.15. Pantalla de navegador tras lectura de código QR

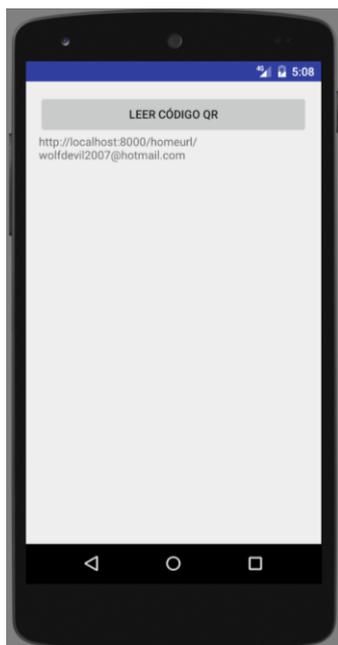


Figura 42. Pantalla de navegador tras lectura de código QR

3.5. Pruebas unitarias y de integración

Las pruebas unitarias consisten en verificar el correcto funcionamiento junto a una bitácora sobre las acciones realizadas en cada uno de los módulos presentes en el prototipo y la prueba integral es el conjunto de todos los módulos trabajando de manera integral, se puede identificar un módulo en la aplicación Android; Lectura QR y tres módulos en el aplicativo *web*; Registro e inicio de sesión de usuario, perfil social y Geolocalización.

3.5.1. Registro e inicio de sesión de usuario

Tabla 9.

Bitácora #1

BITÁCORA #1		
REGISTRO E INICIO DE SESIÓN DE USUARIO		
ACCIÓN	DETALLE	RESULTADO
Ingreso a página principal	Mediante cualquier explorador se ingresa a la página localhost:8000	OK

Navegación e ingreso de la página para registro de usuario	Se generan credenciales y crea el usuario exitosamente	OK
Ingreso de usuario ya existente a su cuenta	En la página de inicio se ingresa a la opción de inicio de sesión e ingresa las credenciales exitosamente	OK
Cierre de sesión de usuario	Se procede a cerrar sesión de usuario exitosamente	OK

Se crea el usuario andres.darquea@test.com con una contraseña genérica

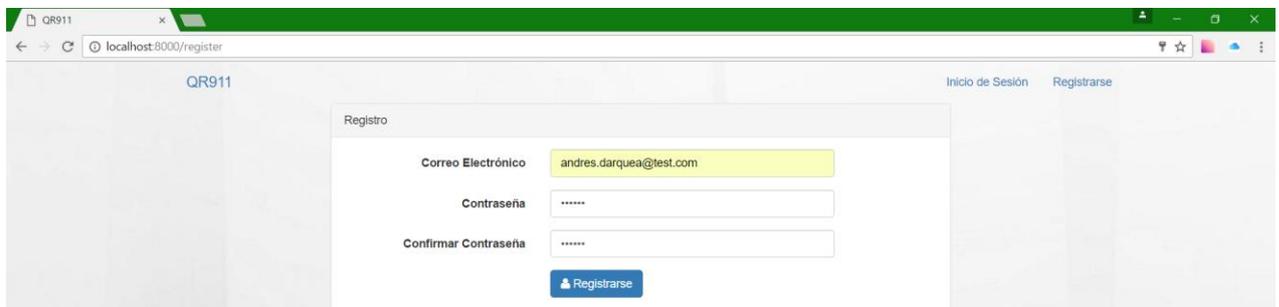


Figura 43. Creación de usuario

Tras registrarse el usuario, se visualiza el inicio de sesión exitoso



Figura 44. Verificación de creación de usuario

Se procede a cerrar la sesión para esta vez solo iniciarla y no registrar un nuevo usuario

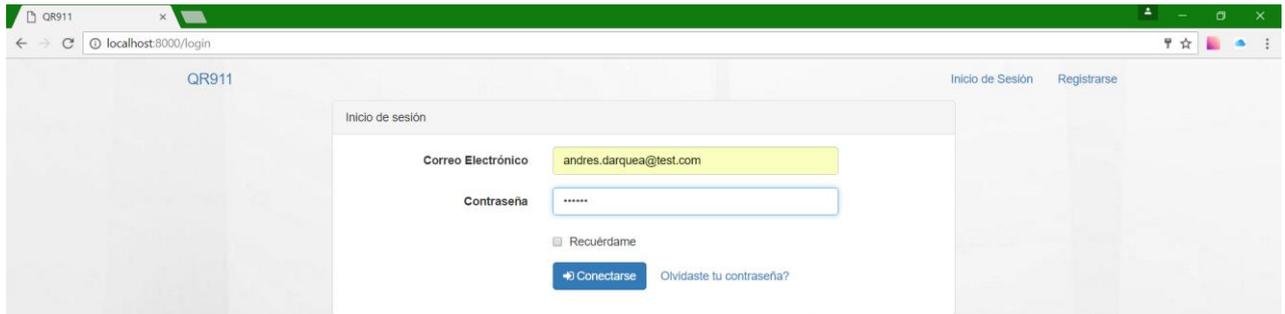


Figura 45. Inicio de sesión de usuario

El inicio de sesión fue exitoso



Figura 46. Inicio de sesión exitoso

3.5.2. Perfil social

Tabla 10.

Bitácora #2

BITÁCORA #2		
PERFIL SOCIAL		
ACCIÓN	DETALLE	RESULTADO
Revisión de datos generales y de salud pertenecientes al usuario registrado	Se ingresa desde la página de inicio a la página de "Mis datos" y verifica que los datos estén desplegados de manera correcta	OK
Actualización de datos generales y de salud pertenecientes al usuario registrado	Desde la barra superior se ingresa a los datos generales y salud para modificarlos exitosamente y retornar a la página de inicio	OK

Verificación de datos actualizados	Se retorna a la página de "Mis datos" para verificar que la información del usuario esté registrada exitosamente	OK
Revisión de usuarios relacionados al perfil	Se ingresa al link de "Perfil Social" ubicado dentro del nombre del usuario activo en la sesión en la barra superior de navegación, para posteriormente verificar peticiones de acceso a su perfil y peticiones enviadas	OK
Agregar usuarios a vista del perfil	Se elige uno de los usuarios adicionales creados y envía una petición de acceso al usuario, esta petición de acceso es aceptada por el otro usuario, permitiéndole así acceso a reportes de geolocalización	OK

El perfil social consiste en la información que el usuario desee brindar a aquellas personas que hagan la lectura de su código QR y le permite administrar quién puede observar dichos reportes adicionalmente.

The screenshot displays a user profile interface for a user named 'QR911'. The page is divided into three main sections: 'Información Personal', 'Salud', and 'Tu código QR'. The 'Información Personal' section contains a table with the following data:

Nombres	
Apellidos	
Fecha de Nacimiento	
País de Residencia	Por configurar
Género	No configurado
Estado civil	No configurado
Dirección de Residencia	
Teléfono Personal	
Teléfono de Emergencia	
Correo Electrónico	andres.darquea@test.com

Below this table is a blue button labeled 'Editar Información'. The 'Salud' section contains a table with the following data:

Tipo de Sangre	No configurado
Implantes	No configurado
Alergias	No configurado

Below this table is another blue button labeled 'Editar Información'. The 'Tu código QR' section features a QR code. At the bottom left of the page, there is a small link that says 'Como otros me ven'.

Figura 47. Vista para usuario conectado sobre sus datos

Al momento de registrarse por defecto se establecen valores en los campos “país de residencia, género, estado civil, tipo de sangre, implantes, alergias”.

La edición de información personal y salud del usuario no es obligatoria, dicha información puede ser agregada mediante los botones visualizados en la pantalla y adicionalmente a través de la barra de inicio bajo el correo del usuario en la sección de datos.



Figura 48. Menú de opciones para usuario registrado

Se procede a editar la información personal y de salud

The screenshot shows the 'Información General' form. The fields are as follows:

- Nombres: Wolfgang Andrés
- Apellidos: Darquea Berger
- País de Residencia: Ecuador
- Género: Masculino
- Estado Civil: Comprometido/a
- Fecha de Nacimiento: 27/09/1989
- Dirección de Residencia: En la oficina
- Teléfono Personal: 0997785397
- Teléfono de Emergencia: 0997785397

An 'Actualizar Datos' button is located at the bottom right of the form.

Figura 49. Edición de Información general del usuario registrado

localhost:8000/healthupdate?

QR911

Mis Datos

Salud

Alergias

Polen, Polvo

Implantes

Marcapasos

Tipo de Sangre

A positivo

Actualizar Datos

Figura 50. Edición de información sobre salud del usuario registrado

De manera exitosa se registran los cambios generados en la cuenta

QR911

localhost:8000/updatehealth

QR911

Información Personal

Nombres	Wolfgang Andrés
Apellidos	Darquea Berger
Fecha de Nacimiento	1989-09-27
País de Residencia	Ecuador
Género	Masculino
Estado civil	Comprometido/a
Dirección de Residencia	En la oficina
Teléfono Personal	0997785397
Teléfono de Emergencia	0997785397
Correo Electrónico	andres.darquea@test.com

Editar Información

Figura 51. Confirmación de ingreso de información de usuario

El prototipo también cuenta con la sección para la administración de registro de usuarios que deseen acceder a los reportes generados en el usuario activo y registro de reportes de usuarios a los que el usuario activo desee acceder

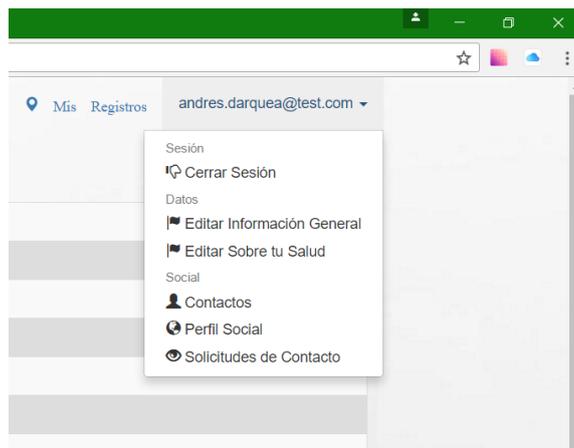


Figura 52. Menú de opciones para el usuario conectado

En la sección de perfil social es posible administrar las peticiones que se envíen, navegar hacia la lista de contactos y revisar peticiones de contactos con la información de los mismos en caso que la tengan agregada. Para efectos de prueba se envía la solicitud de acceso a reportes a uno de los usuarios visibles

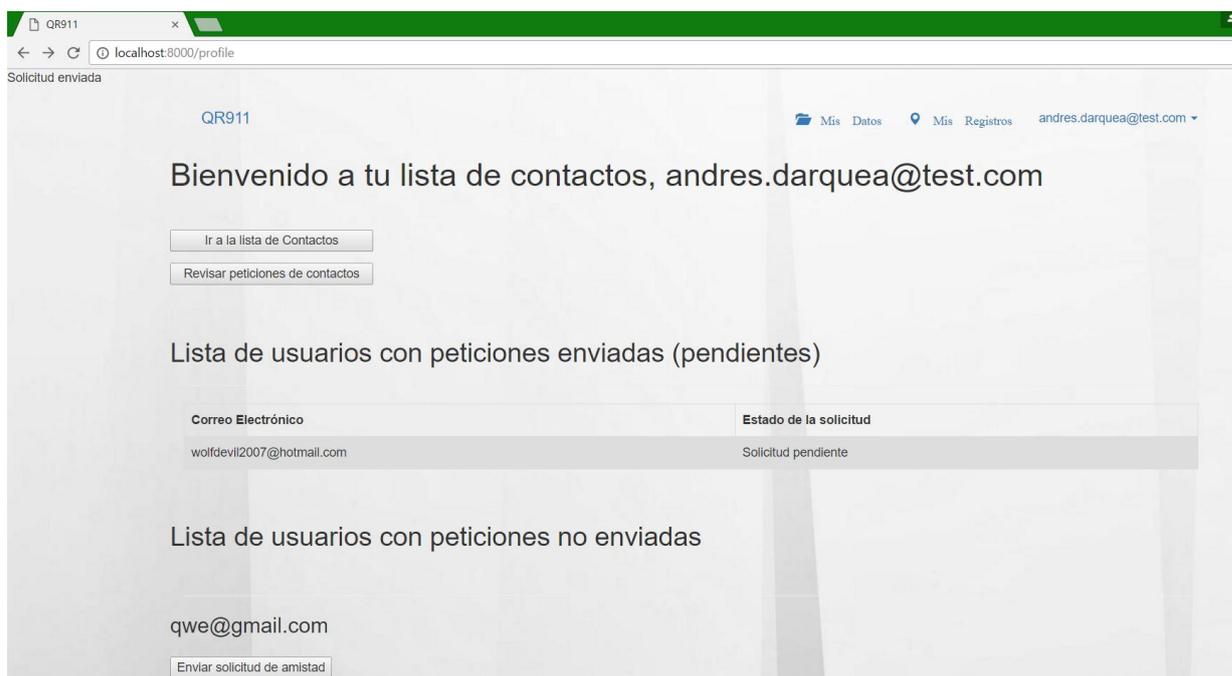


Figura 53. Vista de lista de contactos y peticiones al usuario conectado

Se verifica la solicitud desde el otro usuario y agrega exitosamente



Figura 54. Vista sobre lista de peticiones a usuario conectado

Al ingresar a la lista de contactos registrados desde el usuario de prueba se verifica que exitosamente ha sido agregado para poder visualizar los reportes generados en el otro usuario.

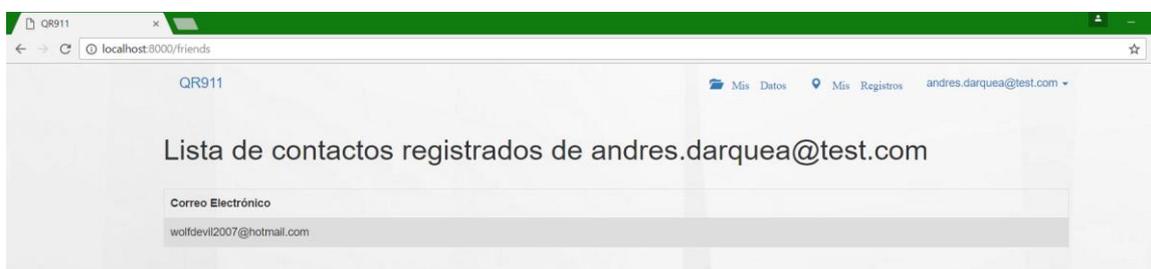


Figura 55. Vista de contactos con petición aceptada

3.5.3. Geolocalización

Tabla 11.

Bitácora #3

BITÁCORA #3		
GEOLOCALIZACIÓN		
ACCIÓN	DETALLE	RESULTADO
Generación de reporte de ubicación geográfica sobre usuario registrado	Se ingresa a la vista "como otros me ven" desde la página que despliega todos los datos del usuario activo para dar click en "reportar ubicación" y generar el evento	OK
		OK

Revisión de reporte de ubicación geográfica generado sobre perfil propio	Se ingresa desde el tab "Mis registros" para revisar el acceso a los reportes generados de la cuenta y los reportes de los perfiles a los que se tiene acceso	
Revisión de reporte de ubicación sobre perfil compartido	Se ingresa al tab "Mis Registros" y revisa los eventos generados de los usuarios que compartieron perfil con la cuenta activa	OK

Para realizar las pruebas en el módulo de geolocalización se ingresa a la vista "como otros me ven" en la página de "Mis datos" y genera un reporte

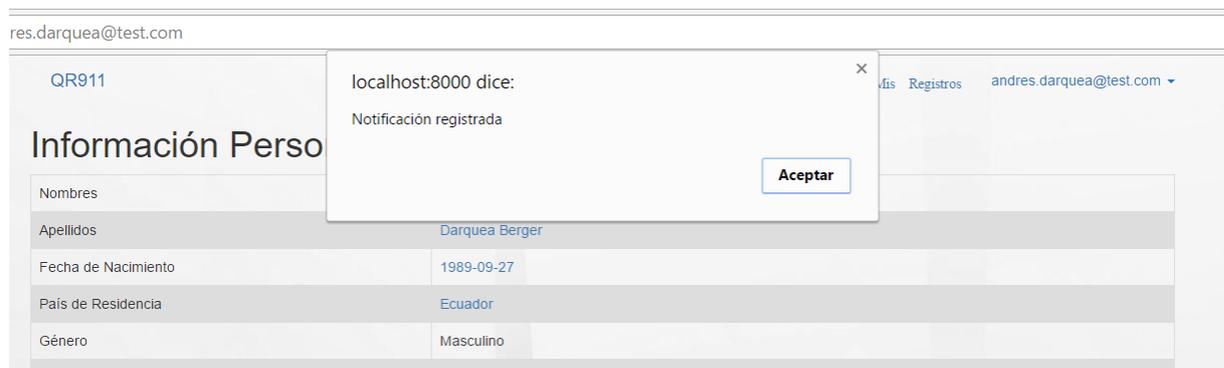


Figura 56. Registro posición geográfica

Se revisa el reporte generado previamente

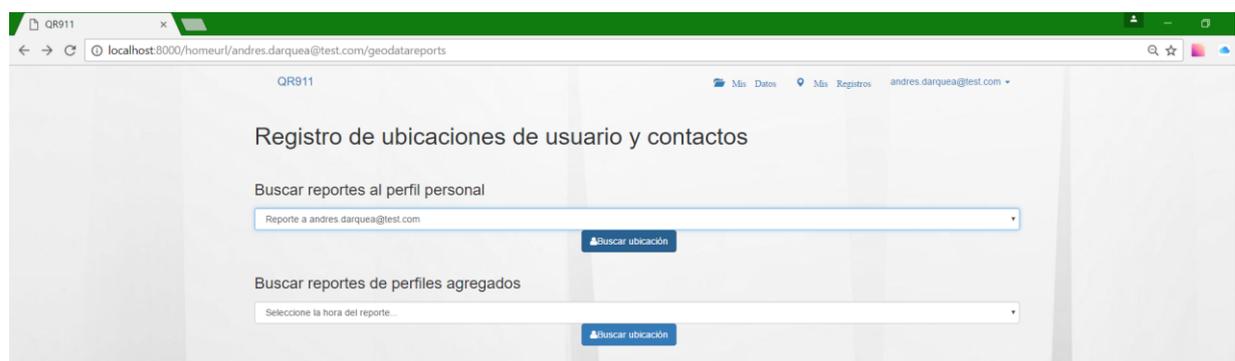


Figura 57. Pantalla para revisión de reportes personales y agregados

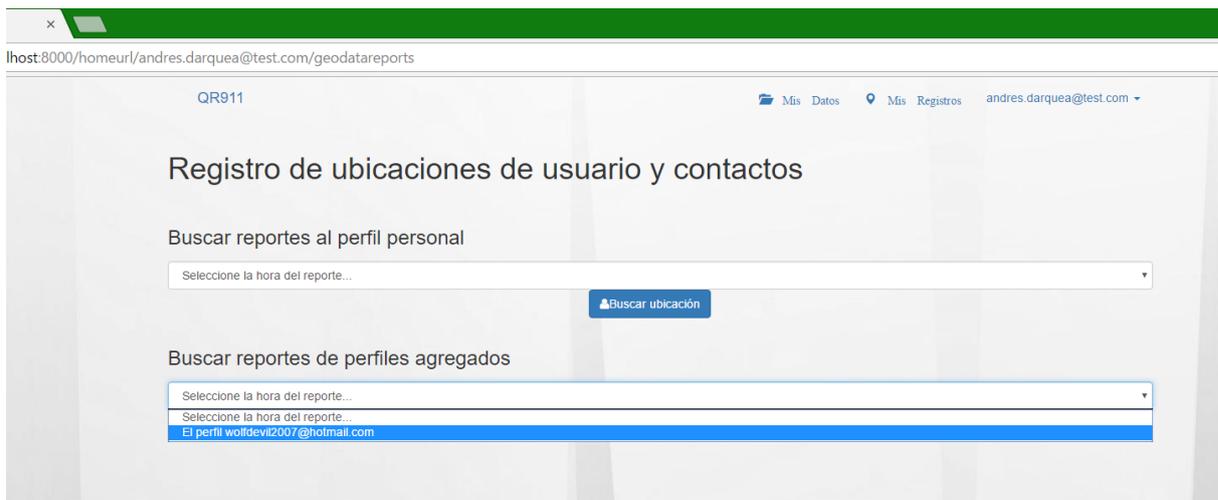


Figura 58. Registro de perfiles agregados

Al generar un reporte sobre el otro usuario previo a la autorización para el acceso a ese contacto se puede visualizar los reportes compartidos

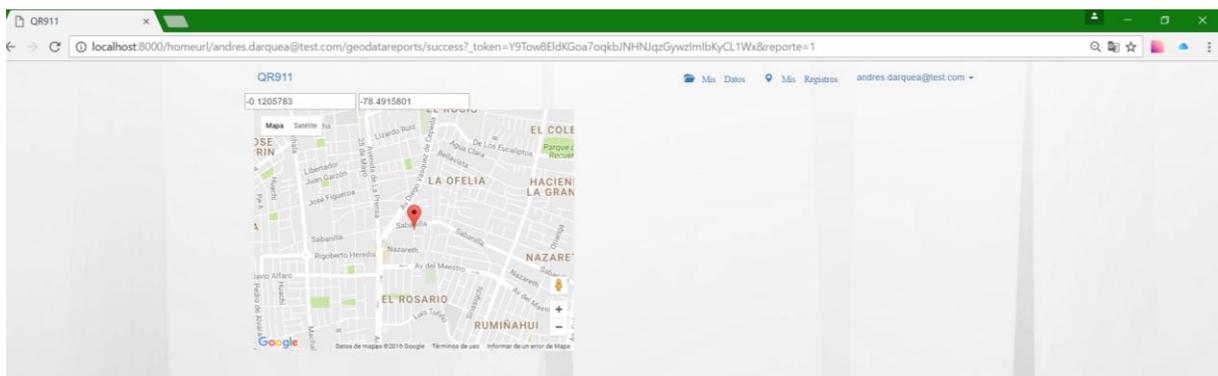


Figura 59. Pantalla indicando la posición geográfica del registro

3.5.4. Lectura QR

Tabla 12.

Bitácora #4

BITÁCORA #4		
LECTURA QR		
ACCIÓN	DETALLE	RESULTADO
Inicio de APP		OK

	Se inicia la aplicación para verificar que no hay problemas de versión o permisos	
Lectura de código QR con la aplicación de Android	Se ejecuta el botón "LEER CÓDIGO" y con la cámara se enfoca el código QR a leer	OK
Verificación de lectura de código QR	Posterior a la lectura del código y re direccionamiento de la página, en la misma actividad de la aplicación se retrocede para verificar el retorno correcto del contenido del código QR leído	OK

Para verificar el correcto funcionamiento del lector QR se procede a iniciar el dispositivo Android virtual

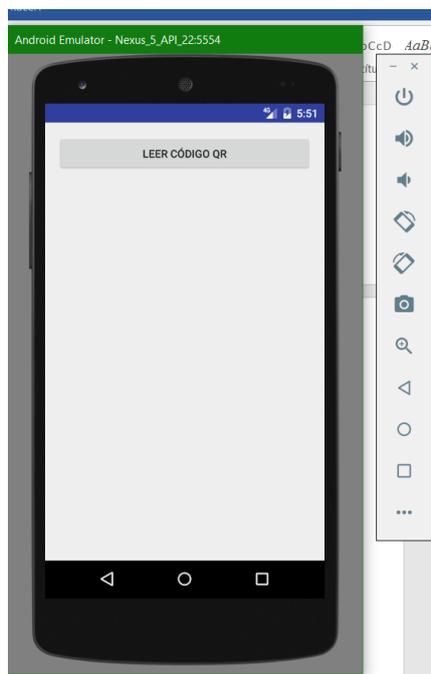


Figura 60. Pantalla del aplicativo Android para la lectura de códigos QR
Posteriormente se inicia la pantalla de lectura del código QR y escanea el objeto.



Figura 61. Código QR objetivo

Adicionalmente al retroceder de actividad se puede verificar mediante un texto plasmado el contenido del código QR leído

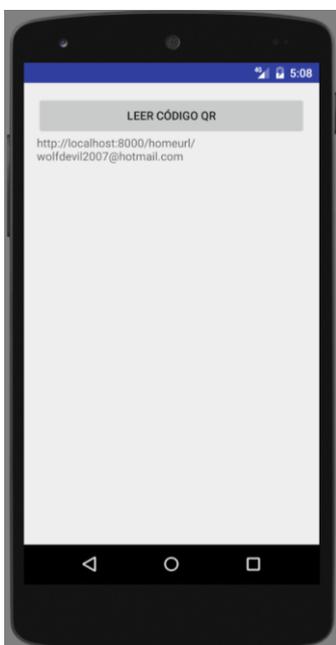


Figura 62. Lectura del Código QR

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Mediante la asignación dinámica de códigos QR a usuarios registrados en el aplicativo *web*, se puede acceder a los usuarios con tan solo la lectura del código QR asignado y acceso a internet, permitiendo brindar de manera oportuna la información del usuario requerida por el lector.

Fue posible compartir la ubicación geográfica de un evento en base al registro generado por el visitante en el perfil, mediante la obtención de la ubicación del dispositivo que está realizando el reporte al usuario leído.

Se logró obtener un aplicativo *web* que no discrimina el tipo de navegador que se desee usar, esto debido al popular marco de trabajo en el que está desarrollado, con una amplia gama de soporte en línea y de fácil edición por el lenguaje de programación.

Se identificó la información básica para situaciones en que se requiere de asistencia médica, permitiendo así a los usuarios del aplicativo *web* proveer dicha información al visitante en el perfil mediante la lectura del código QR, indicando previamente que la información ingresada por el usuario es bajo su propio riesgo.

En base a las pruebas unitarias e integración entre módulos se pudo verificar que el aplicativo *web* cumple con los requisitos planteados en un inicio en esta investigación, permitiendo así un flujo de navegación e interacción para el visitante o usuario.

4.2. Recomendaciones

Al aplicativo *web* se le puede incorporar módulos adicionales, que a su vez incrementen el nivel de funcionalidad y alcance, dichos módulos pueden incluir un manejo en el perfil social como la forma de búsqueda y hallazgo de perfiles ya generados, a través de cualquiera de los campos de información personal del usuario registrado.

La creación de un módulo que implemente un proceso de notificaciones sobre perfiles compartidos tras la generación de un evento o reporte de ubicación geográfica en el perfil compartido con los usuarios que tienen acceso a dichos reportes.

En el reporte de la ubicación geográfica se puede generar acciones posibles a realizar con el evento.

Para poder acceder a la información de un perfil de usuario es requerido el acceso a internet, pero los códigos QR también permiten contener información similar a la que se está registrando en el aplicativo *web*, es un módulo que podría integrarse de manera favorable al aplicativo.

En la actualidad el marco de trabajo Laravel está en constante evolución bajo demanda, lo que genera que aparezcan nuevas versiones, es recomendable revisar la documentación antes de proceder con la actualización de versión al proyecto.

REFERENCIAS

- Agrawal, A. &. (2015). Analysis. En A. &. Agrawal, Analysis of Multidimensional Modeling Related To Conceptual Level. Recuperado el 1 de septiembre de 2016 de <http://www.ijrra.com/Vol2issue1/IJRRRA-02-01-26.pdf>,
- Alberto U, A. F. (2012). El Estudio Cloud Computing. Retos y Oportunidades. Recuperado el 3 de septiembre de 2016 de http://www.academia.edu/20019300/1-_estudio_cloud_computing_retos_y_oportunidades_vdef
- Allen, M. (2001). The CTDP Database Guide Version 0.2.1. Recuperado el 1 de septiembre de 2016 de <http://www.comptechdoc.org/independent/database/begin/>
- Anton, C. (2011). Platzi. Recuperado el 2 de Julio de 2016 de <https://platzi.com/blog/laravel-framework-php/>,
- Awad, M. A. (2005). A Comparison between Agile and Traditional. Recuperado el 3 de Julio de 2016 de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.464.6090&rep=rep1&type=pdf>
- Bachman, C. (2013). The programmer as navigator. Recuperado el 23 de Agosto de 2016 de <https://pdfs.semanticscholar.org/f371/d196bf0e7b43df6dcbbc44de461925a21709.pdf>
- Báez, M., Borrego, Á., Cordero, J., Cruz, L., González, M., Hernández, F., . . . Llera, J. R. (2011). Introducción a Android. Recuperado el 4 de Julio de 2016 de <http://www.it-docs.net/ddata/18.pdf>
- Belmonte, O. (2004). Introducción al lenguaje de programación Java. Recuperado el 20 de Junio de 2016 de <http://www.pdf-manual.es/programacion/java/62-introduccion-al-lenguaje-de-programacion-java.html>

- Camps, R. (2005). Introducción a las bases de datos. Recuperado el 20 de febrero de 2016 de <http://www.uoc.edu/masters/oficiales/img/913.pdf>
- Chuang, J. C., Hu, Y. C., & Ko, H. J. (2010). A Novel Secret Sharing Technique Using QR Code, International Journal of Image Processing. Recuperado el 27 de septiembre de 2016 de <http://www.cscjournals.org/manuscript/Journals/IJIP/Volume4/Issue5/IJIP-263.pdf>
- computerweekly. (2009). computerweekly. Recuperado el 21 de septiembre de 2016 de <http://www.computerweekly.com/feature/A-history-of-cloud-computing>.
- Dawson, A. (20 de Febrero de 2012). sixrevisions. Recuperado el 21 de septiembre de 2016 de <http://sixrevisions.com/web-development/qr-codes-uri-schemes/>
- Defossé, N., & Marcos, v. H. (2009). Sistemas Web Desconectados. Recuperado el 5 de septiembre de 2016 de <http://docplayer.es/4271841-Grupo-de-sistemas-y-comunicaciones.html>
- Diputación de Cádiz. (2011). Servicio de Sociedad de la Información. Recuperado el 2 de agosto de 2016 de http://www.dipucadiz.es/sociedad_de_la_informacion
- djangoproject. (2008). djangoproject. Recuperado el 15 de septiembre de 2016 de www.djangoproject.com.
- ECU911. (s.f.). ECU911. Recuperado el 2 de mayo de 2016 de <http://www.ecu911.gob.ec/programas-y-servicios/>.
- emarketer. (28 de Juni de 2013). <http://www.emarketer.com>. Recuperado el 4 de mayo de 2016 de <http://www.emarketer.com/Article/US-Ahead-of-Western-Europe-QR-Code-Usage/1009631>.
- Espinosa, C. (20 de Mayo de 2014). Smartphones en Ecuador: Acceso se duplico en 2 años. Recuperado el 10 de mayo de 2016

<http://www.coberturadigital.com/2014/05/20/smartphones-en-ecuador-acceso-se-duplico-en-2-anos/>.

Estateqr codes. (2014). Advantages and Disadvantages of QR code. Recuperado el 15 de mayo de 2016 de <http://www.estateqr codes.com/advantages-disadvantages.html>.

Fabien Potencier, F. Z. (s.f.). Symfony 1.2, la guía definitiva. Recuperado el 20 de mayo de 2016 de http://librosweb.es/libro/symfony_1_2/.

Fernández, M. (2009). Generador e intérprete de QR code. Recuperado el 3 de marzo de 2016 de <https://es.scribd.com/document/72506982/Documentacion-proyecto-QR-Code>

Fundación Orange. (2010). Manual SPQR. Recuperado el 15 de mayo de 2016 de <https://www.specialqr.org/index.php/proyecto-spqr.html>

Galiano, F. B. (s.f.). Fernando Berzal. Recuperado el 26 de mayo de 2016 de <http://elvex.ugr.es/>

Garrido, J. (2013). TFC Desarrollo de Aplicaciones Móviles. Recuperado el 13 de marzo de <http://www.revistalatinacs.org/070/paper/1063/RLCS-paper1063.pdf>

Ginestà, M. G., & Mora, O. P. (2006). Bases de datos en PostgreSQL. Recuperado el 2 de febrero de 2016 de <http://www.uoc.edu/masters/oficiales/img/913.pdf>

Ginesta, M., & Pérez, O. (2013). Bases de Datos en PostgreSQL. UOC. <http://www.uoc.edu/masters/oficiales/img/914.pdf>

Guia SGBD. (2014). Sistemas gestores de bases de datos. Recuperado el 6 de septiembre de 2016 de <http://assets.mheducation.es/bcv/guide/capitulo/8448148797.pdf>.

- Gutierrez, J. (2010). ¿Qué es un framework web? Recuperado el 25 de Junio de 2016 de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- Korhan, J. (2011). How QR Codes Can Grow Your Business. Recuperado el 4 de Junio de 2016 de <http://www.socialmediaexaminer.com/how-qrcodes-can-grow-your-business/>
- Llonch, J. (05 de 2008). Curso Framework Symfony. Recuperado el 7 de Junio de 2016 de <http://www.laigu.net/wp-content/uploads/2009/08/Curso-Symfony.pdf>.
- Luque, J. (2010). Códigos QR. Madrid: ACTA. Recuperado el 7 de Junio de 2016
- Management Solutions. (2012). La nube: oportunidades y retos para los integrantes de la cadena de valor. España: Management Solutions.
- Proyectos Agiles Org. Como funciona Scrum. Recuperado el 10 de Junio de 2016 de <http://proyectosagiles.org/como-funciona-scrum/>
- Narayanan, S. (July de 2012). QR Codes and Security Solutions. Recuperado el 10 de septiembre de 2016 de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.467.9840&rep=rep1&type=pdf>.
- Pedrozo, G. (2012). Sistemas Operativos en Dispositivos móviles. Recuperado el 24 de Mayo de 2016 de http://exa.unne.edu.ar/informatica/SO/Sistemas_Operativos_en_Dispositivos_Moviles.pdf
- Pliego, C. (2013). Desarrollo de una aplicación generadora y lectora de códigos QR seguros en Android . Madrid: Universidad Carlos III de Madrid.
- Pressman, R. S. (2005). SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH SEVENTH EDITION. Palgrave Macmillan.

QR Code eBook. (2014). The complete Guide to Marketing. Recuperado el 20 de abril de 2016 de <http://www.qr-code-generator.com/qr-code-marketing/why-should-i-use-qr-codes>

QR CODES. (12 de 2 de 2012). CODIGOS QR. Recuperado el 10 de abril de 2016 de <http://www.codigos-qr.com/>.

Rafael Camps Paré, L. A. (2005). Introduccion a la base de datos. Recuperado el 24 de julio de 2016 de <https://www.semanticscholar.org/paper/Introducci-n-a-Las-Bases-De-Datos-Par%C3%A9/f51dbd1b1460ca6055a9bb417351e71afbcefa95>

Rees, D., & Laguna, A. (2012). Desarrollo de Aplicaciones con el Framework de PHP Laravel. Recuperado el 23 de febrero de 2016 de <http://www.intercambiosvirtuales.org/libros-manuales/laravel-code-happy-es-desarrollo-de-aplicaciones-con-el-framework-de-php-laravel-para-principiantes>

Rouse, M. (2010). searchdatacente. Recuperado el 20 de marzo de 2016 de <http://searchdatacenter.techtarget.com/es/definicion/SQL-o-lenguaje-de-consultas-estructuradas>.

Sánchez, S. (2010). OS: para dispositivos móviles inalámbricos . Recuperado el 4 de enero de 2016 de http://exa.unne.edu.ar/informatica/SO/Sistemas_Operativos_en_Dispositivos_Moviles.pdf

Sanz, D. (2013). Conceptos Básicos Android. Recuperado el 23 de enero de 2016 de <http://diposit.ub.edu/dspace/bitstream/2445/48925/2/memoria.pdf>

Schwaber, K., & Sutherland, J. (2013). La Guía de Scrum. Recuperado el 20 de febrero de 2016 de <https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

- SIMAO, E. M. (20 de Octubre de 2009). Software Development Methodologies. Recuperado el 2 de septiembre de 2016 de http://scholar.googleusercontent.com/scholar?q=cache:RocXjEWmN0gJ:scholar.google.com/+Software+Development+Methodologies+SIMAO,+E+l%C4%B1sio+Maciel&hl=es&as_sdt=0,5
- Soon, T. J. (2008). QR Code, Synthesis Journal 2008. Recuperado el 13 de agosto de 2016 de https://foxdesignsstudio.com/uploads/pdf/Three_QR_Code.pdf.
- symfony.com. (2011). symfony.com. Recuperado el 5 de Agosto de 2016 de <http://symfony.com/es/what-is-symfony>.
- Tatroe, K. M. (2003). Programming Php. " O'Reilly Media, Inc."
- Ticona, S. (2011). Metodologías Tradicionales. Recuperado el 12 de marzo de 2016 de <https://tallerinf281.wikispaces.com/file/view/METODOLOG%C3%8DAS+TRADICIONALES.pdf>
- Uñoja, R. (2011). Metodologías tradicionales vs Metodologías ágiles. Recuperado el 15 de junio de 2016 de <https://prezi.com/xpo8pdv2ghqm/metodologias-tradicionales-vs-metodologias-agiles-Unoja/>
- Vives, B. (2007). Sistema Gestor de Base de Datos. Recuperado el 20 de julio de 2016 de http://dmi.uib.es/~labsoft/Labsg/4003_Labsg_Tema2-1.pdf

