



**viu**

**Universidad  
Internacional  
de Valencia**

## **Aplicación web y móvil multiplataforma para incentivar la donación de sangre en Panamá**

Titulación:  
Máster en Desarrollo de  
Aplicaciones y Servicios  
Web

Curso Académico  
2023-2024

Alumno:  
**GARCÍA GONZÁLEZ,  
Demóstenes**  
DNI: 8-808-595 (PA)

Director del TFM:  
**PÉREZ IBÁÑEZ, Rubén**

Convocatoria:

**PRIMERA**

# Índice general

<b>Índice de figuras</b>	<b>3</b>
<b>Índice de cuadros</b>	<b>3</b>
<b>Glosario de términos</b>	<b>7</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación del proyecto . . . . .	2
<b>2 Objetivos</b>	<b>3</b>
2.1 Objetivos tecnológicos . . . . .	3
2.2 Requisitos del proyecto . . . . .	4
2.2.1 Requerimientos no funcionales . . . . .	5
<b>3 Marco tecnológico</b>	<b>7</b>
3.1 Métodos existentes para donar hoy en día en Panamá . . . . .	8
3.2 Tecnologías utilizadas . . . . .	9
3.2.1 <i>Framework backend</i> : Laravel . . . . .	9
3.2.2 Motor de Base de Datos: PostgreSQL . . . . .	10
3.2.3 Librería <i>frontend</i> : React . . . . .	11
3.2.4 <i>Framework móvil</i> : React Native . . . . .	12
3.2.5 Expo . . . . .	13
3.2.6 Git y Gitlab . . . . .	13
3.2.7 Inertia . . . . .	15
3.3 Diseño de la interfaz . . . . .	15
3.3.1 Sistema de Diseño: Plaqueta DS . . . . .	15
<b>4 Metodología</b>	<b>19</b>
4.1 Dirección y gestión del proyecto . . . . .	20
4.1.1 Reglas establecidas . . . . .	21
4.2 Herramienta para la dirección del proyecto: Jira Software . . . . .	21
4.3 Épicas e historias de usuario . . . . .	22
4.3.1 Épicas . . . . .	23
4.3.2 Historias de Usuario: <i>Backlog</i> del primer <i>Sprint</i> . . . . .	23
4.3.3 <i>Roadmap</i> de producto . . . . .	25
4.3.4 Cálculo de velocidad del equipo . . . . .	25
4.4 Diagrama de la Base de Datos . . . . .	25
<b>5 Resultados obtenidos</b>	<b>28</b>
5.1 Gestión y dirección del proyecto . . . . .	28
5.1.1 Resultado de velocidad del equipo . . . . .	28
5.2 Ambiente, despliegue y CI/CD . . . . .	28
5.2.1 CI/CD con GitLab CI/CD . . . . .	29
5.3 Documentación del API . . . . .	32
5.4 Pruebas automatizadas en <i>Backend / API</i> . . . . .	33

5.4.1 Pruebas E2E . . . . .	35
5.5 Panel de Administración . . . . .	35
5.6 Notificaciones por correo . . . . .	36
5.7 Inicio de sesión con cuentas sociales . . . . .	37
5.7.1 Inicio de sesión en móvil: Google (iOS y Android) . . . . .	38
5.7.2 Inicio de sesión en móvil: Apple (iOS) . . . . .	40
5.8 Queues (Colas) . . . . .	43
5.8.1 Máquinas de Estado en <i>Backend</i> . . . . .	44
5.9 Mapas: OpenStreetMap con Leaflet, Google Maps (Android) y Apple Maps (iOS) . . . . .	45
5.9.1 Leaflet con marcadores personalizados . . . . .	46
5.10 Seguridad . . . . .	47
5.10.1 <i>Throttle</i> . . . . .	47
5.10.2 CAPTCHA . . . . .	48
5.10.3 Validaciones . . . . .	48
5.11 Notificaciones Push . . . . .	49
5.11.1 Configuración de <i>backend</i> para notificaciones <i>push</i> . . . . .	50
5.11.2 Notificaciones en Android . . . . .	51
5.11.3 Notificaciones en iOS . . . . .	53
5.12 Optimizaciones de desempeño . . . . .	54
5.12.1 Caché en el móvil . . . . .	55
5.12.2 Otras mejoras de rendimiento . . . . .	55
5.13 Eliminación de cuenta y protección de privacidad . . . . .	57
5.13.1 Confirmación en dos pasos (2FA) . . . . .	58
5.14 Pruebas en dispositivos reales . . . . .	60
5.14.1 Dispositivos para pruebas . . . . .	60
5.14.2 Proceso de pruebas general . . . . .	61
5.14.3 Pruebas para Android . . . . .	62
5.14.4 Pruebas para iOS . . . . .	62
5.15 Sitio informativo (Brochure) . . . . .	63
5.16 Escalabilidad de la solución: caso práctico . . . . .	65
<b>6 Conclusiones</b>	<b>66</b>
<b>7 Trabajo futuro</b>	<b>67</b>
<b>A Apéndice A: Épicas</b>	<b>68</b>
<b>B Apéndice B: Historias de usuario para el <i>Sprint 1</i></b>	<b>72</b>
<b>C Apéndice C: Reglas del Proyecto Ágil</b>	<b>85</b>
<b>D Apéndice D: <i>Roadmap</i> del producto</b>	<b>87</b>
<b>E Apéndice E: Velocidad evidenciada</b>	<b>88</b>

<b>F Apéndice F: Supervisor de colas</b>	89
<b>G Apéndice G: Marcadores personalizados con Leaflet en JavaScript</b>	90
<b>H Apéndice H: Archivo de migración para soportar notificaciones Push</b>	91
<b>I Apéndice I: Implementación de <i>Lazy Loading</i> en React</b>	92
<b>J Apéndice J: Expresión regular de cédula panameña</b>	94
<b>K Apéndice K: Eventos a través de máquinas de estado</b>	95
K.1 Estados en citas . . . . .	95
K.2 Transiciones en citas . . . . .	96
K.3 Configuración de máquina de estado con el modelo . . . . .	96
K.4 Notificaciones Push con Expo . . . . .	97
<b>L Apéndice L: Configuración de Firebase en Expo</b>	98
<b>M Apéndice M: Aplicación de escritorio con Tauri</b>	99
<b>N Apéndice N: Otras características del sitio informativo</b>	103
N.1 Sitio responsive . . . . .	103
N.2 Etiquetas para SEO y OpenGraph . . . . .	104
N.3 Auditoría de mejores prácticas . . . . .	105
N.4 Archivo para robots . . . . .	107
N.5 Sistema público de validación de citas de donación . . . . .	107
N.5.1 Obtener la información del QR . . . . .	109
<b>Ñ Apéndice Ñ: Auditoría de seguridad</b>	111
Ñ.1 Resumen . . . . .	111
Ñ.1.1 Alcance . . . . .	111
Ñ.2 Vulnerabilidades identificadas . . . . .	111
Ñ.2.1 Vulnerabilidad 1: Ausencia de CSP . . . . .	112
Ñ.2.2 Vulnerabilidad 2: Información del servidor expuesta . . . . .	114
Ñ.2.3 Vulnerabilidad 3: Ausencia de X-Frame-Options ( <i>clickjacking</i> ) .	115
Ñ.2.4 Vulnerabilidad 4: Ausencia de <i>HTTP Strict Transport Security</i> (HSTS) . . . . .	116
Ñ.3 Vulnerabilidades bajas . . . . .	116
<b>O Apéndice O: Consideraciones para puesta en producción</b>	117
O.1 Costos asociados . . . . .	117
<b>Bibliografía</b>	118

# Índice de figuras

2.1	Requerimientos de muy alto nivel. Fuente: Elaborado por el autor. . . . .	5
3.1	Popularidad librerías FE. Fuente Stack Overflow (2023). . . . .	11
3.2	Pull Request en Gitlab. Fuente: Elaborado por el autor. . . . .	14
3.3	Commitizen-cli en su uso. Fuente: Elaborado por el autor. . . . .	14
3.4	Logotipo de Dono Sangre. Fuente: Elaborado por el autor. . . . .	16
3.5	Proceso Tokens - Fuente: Elaborado por el autor. . . . .	17
3.6	Productividad con Design System. Fuente: Callahan (2021). . . . .	17
4.1	Sprint en Jira. Fuente: Elaborado por el autor. . . . .	21
4.2	Jira - Integración con GitLab. Fuente: Elaborado por el autor. . . . .	22
4.3	Diagrama de Base de Datos. Fuente: Elaborado por el autor. . . . .	26
4.4	Diagrama de Base de Datos (continuación). Fuente: Elaborado por el autor. . . . .	27
5.1	CI/CD en GitLab. Fuente: Elaborado por el autor. . . . .	30
5.2	Integración con Slack. Fuente: Elaborado por el autor. . . . .	31
5.3	OpenAPI <i>specification</i> . Fuente: Elaborado por el autor. . . . .	32
5.4	Tests con PHPUnit. Fuente: elaborado por el autor. . . . .	33
5.5	Panel de Admin - Citas. Fuente: Elaborado por el autor. . . . .	36
5.6	Notificación por correo. Fuente: Elaborado por el autor. . . . .	37
5.7	OAuth con Google, Parte 2. Fuente: Elaborado por el autor. . . . .	38
5.8	OAuth con Google en móvil. Fuente: Elaborado por el autor. . . . .	39
5.9	Inicio con Apple. Fuente: Elaborado por el autor. . . . .	41
5.10	Inicio con Apple desde iOS. Fuente: Elaborado por el autor. . . . .	42
5.11	Queue de cita. Fuente: Elaborado por el autor. . . . .	44
5.12	Marcador personalizado. Fuente: Elaborado por el autor. . . . .	47
5.13	hCaptcha en formulario de login. Fuente: Elaborado por el autor. . . . .	48
5.14	Registro de notificaciones. Fuente: Elaborado por el autor. . . . .	52
5.15	Recibo de notificación. Fuente: Elaborado por autor. . . . .	53
5.16	Estadísticas por caché. Fuente: Elaborado por el autor. . . . .	54
5.17	Verificador de conexión. Fuente: Elaborado por el autor. . . . .	57
5.18	Eliminar cuenta. Fuente: Elaborado por el autor. . . . .	59
5.19	Eliminar cuenta (Confirmar) - Fuente: Elaborado por el autor. . . . .	59
5.20	Equipos de prueba. Fuente: Elaborado por el autor. . . . .	61
5.21	Sitio web de bienvenida. Fuente: Elaborado por el autor. Ilustraciones por DrawKit.com . . . . .	64
5.22	Aplicación de escritorio en Tauri. Fuente: Elaborado por el autor. . . . .	65
D.1	Product Roadmap. Fuente: Elaborado por el autor. . . . .	87

M.1	Aplicación de Tauri en listado de apps. Fuente: Elaborado por el autor.	102
N.1	Sitio responsive (móvil, tableta y escritorio). Fuente: Elaborado por el autor.	103
N.2	Sitio responsive. Fuente: Elaborado por el autor	104
N.3	Métricas en Google Lighthouse. Fuente: Elaborado por el autor.	105
N.4	Métricas en GTMetrix. Fuente: Elaborado por el autor.	106
N.5	Desglose de información en QR. Fuente: Elaborado por el autor.	108
N.6	Resultado de validación de cita. Fuente: Elaborado por el autor.	109
N.7	Lector físico de códigos QR. Fuente: Elaborado por el autor.	110
Ñ.1	Vulnerabilidad 2 con Burp Suite. Fuente: Elaborado por el autor.	114
Ñ.2	Vulnerabilidad 2 (mitigación). Fuente: Elaborado por el autor.	115

# Índice de cuadros

3.1	Métodos actuales de donación . . . . .	8
3.2	Distintos frameworks en PHP . . . . .	9
3.3	Bases de Datos - Licencias . . . . .	10
3.4	Librerías FE - Licencias . . . . .	12
5.1	SPA-22 - Historia . . . . .	34
5.2	Notificaciones del sistema (push y mail) . . . . .	49
5.2	Notificaciones del sistema (push y mail) . . . . .	50
5.3	Listado de equipos de pruebas . . . . .	60
A.1	SPA-2 / Web - <i>Architecture</i> . . . . .	68
A.2	SPA-27 / Mobile - <i>Architecture</i> . . . . .	68
A.3	SPA-29 / Web - <i>Brochure site</i> . . . . .	68
A.4	SPA-28 / Web - <i>User Management</i> . . . . .	69
A.5	SPA-5 / Mobile - <i>User Management</i> . . . . .	69
A.6	SPA-30 / Mobile - <i>Information/Brochure</i> . . . . .	69
A.7	SPA-33 / Web - <i>Donations</i> . . . . .	69
A.8	SPA-92 / Mobile - <i>Donations</i> . . . . .	69
A.9	SPA-35 / Web - <i>Appointments</i> . . . . .	70
A.10	SPA-34 / Mobile - <i>Appointments</i> . . . . .	70
A.11	SPA-36 / Web - <i>Donation Centers</i> . . . . .	70
A.12	SPA-37 / Mobile - <i>Donation Centers</i> . . . . .	70
A.13	SPA-40 / Web - <i>Gamification</i> . . . . .	71
A.14	SPA-41 / Mobile - <i>Gamification</i> . . . . .	71
A.15	SPA-42 / Web - <i>Statistics</i> . . . . .	71
A.16	SPA-76 / Web - <i>Request for donors</i> . . . . .	71
A.17	SPA-77 / Mobile - <i>Request for donors</i> . . . . .	71
B.1	Sprint Backlog - Sprint 1 . . . . .	73
C.1	Reglas del proyecto ágil . . . . .	85
E.1	Velocidad (evidenciada) del equipo . . . . .	88

# Glosario de términos

**accesibilidad** La accesibilidad para la web señala que los sitios, herramientas y tecnologías deben ser desarrolladas y diseñadas para que las personas con capacidades especiales puedan utilizarlas (W3C, 2023). 105

**API** Un API o *Application Programming Interface* (Interfaz de Programación de Aplicaciones) es una capa expuesta para la comunicación con un módulo o software. 15, 25, 28, 29, 32, 36, 46, 47, 65, 95, 99

**background** En las programaciones de tareas (*job scheduling*), el *background* es el ambiente en donde las tareas que no requieren interacción con el usuario final o de baja prioridad son ejecutadas (IEEE, 1990). 44, 57

**CAPTCHA** *Completely Automated Public Turing test to tell Computers and Humans Apart* (CAPTCHA) es un mecanismo que se utiliza para poder distinguir entre humanos y sistemas automatizados como bots, mediante el despliegue de desafíos visuales. 48

**CDN** Un *Content Delivery Network* (Red de Entrega de Contenido) es un grupo o conjunto de servidores desplegados en distintas regiones que permiten la entrega de contenido estático (como imágenes). 56, 66

**CI/CD** *Continuous Integration / Continuous Delivery* (Integración Continua / Despliegue Continuo) es una práctica de software en donde el incremental del software se despliega continuamente y frecuentemente al usuario final de forma automatizada. 7, 13, 29, 66

**componente** Elemento mínimo del software que podría ser probado aisladamente (ISTQB, 2023). 5, 7, 31

**criterios de aceptación** Los criterios de aceptación son el listado de criterios que un sistema debe satisfacer para ser aceptados por el usuario, cliente u otra entidad autorizada (IEEE, 1990). 23, 33

**Cummulative Layout Shift** *Cummulative Layout Shift* (Cambio de diseño acumulado) es una métrica para definir la cantidad de elementos que cambian en un espacio específico. 56, 106

**Design System** Un *Design System* (sistema de diseño) es una librería local de componentes listos, estilizados y uniformes que permiten agilizar el proceso de desarrollo y al mismo tiempo hacerlo más uniforme. 15, 47, 63, 66

**DevOps** DevOps una metodología donde se integra el desarrollo (**Development**) con la parte operativa (**Operations**) del software. 1, 13

**donación por reposición** Las donaciones por reposición son donaciones donde el donante realiza la donación para reponer una unidad de sangre o plaqueta para un paciente específico y no para proveerlo abiertamente al centro. 1, 2, 58

**DRY** DRY o *Don't Repeat Yourself* (en español, "no te repitas"), es un principio de diseño que busca la disminución de código repetido dentro de una fuente de código. 55

**interoperabilidad** La capacidad de un software para interactuar con uno o más componentes específicos u otros sistemas (ISTQB, 2023). 7, 39

**JWT** JSON Web Token o JWT, es un estándar parte de RFC 7519 que permite compartir, entre dos actores y de forma segura, la identidad de un usuario dentro de un sistema. 41, 42, 65, 99

**lazy loading** *Lazy loading* (o carga en retraso) es una técnica de programación de carga de componentes de forma demorada (o retrasada) y según la necesidad actual del usuario. 92

**mantenibilidad** El grado de facilidad en que un producto de software puede ser modificado para corregir defectos o cumplir con nuevos requerimientos (ISTQB, 2023). 6

**mejores prácticas** Un método o práctica de carácter superior que contribuye a mejorar el desempeño de una organización bajo un contexto establecido, generalmente conocido como "mejor" por otras organizaciones (ISTQB, 2023). 1, 20, 40, 66, 105, 106

**máquina de estado finita** Modelo conceptual que describe procesos y comportamientos de un sistema a partir de los distintos estados (finitos) que puede tener un flujo (IEEE, 1990). 44, 51, 66

**OWASP** La OWASP, u *Open Worldwide Application Security Project*, es una comunidad internacional que produce documentos y metodologías asociadas con la seguridad informática (OWASP, 2001). 6

**PMI** El *Project Management Institute* es una organización global dedicada a guiar y normar estándares de gestión de proyectos. 19

**PoC** *Proof of Concept* (Prueba de Concepto) es una práctica en software en donde se desarrolla un producto o proyecto para demostrar la viabilidad de dicho proyecto, mostrando las características generales de la misma y sin esperar necesariamente su puesta en producción. 3, 5, 40, 66, 67

**prioridad** El nivel de importancia de un elemento desde la perspectiva del negocio (ISTQB, 2023). 4, 5, 67

**prueba funcional** Un requerimiento que especifica la funcionalidad esperada de un componente o sistema (ISTQB, 2023). Son pruebas que verifican si el sistema realiza la tarea esperada desde la perspectiva del usuario. 33–35

**prueba unitaria** *Test* (prueba) de una unidad o grupo individual de elementos de un software o hardware (IEEE, 1990). Son pruebas que se escriben en aislamiento y verifican funcionalidad específica en una unidad o parte individual del código. 33, 35

**pruebas automatizadas** El uso de software para ejecutar o soportar actividades relacionadas con las pruebas. Por ejemplo: administración de pruebas, diseño de pruebas, ejecución de pruebas y verificación de pruebas (ISTQB, 2023). 1, 33

**pruebas de penetración** Tipo especializado de pruebas enfocadas en atacar los vectores y vulnerabilidades listados como los más habituales según la OWASP (OWASP, 2001). 49, 111

**queue** Las *queues* (colas) son un listado de elementos que son agregados al final de la lista y recogidos desde la primera posición de la lista (IEEE, 1990). 43, 44, 66, 95, 96

**rama** La rama (*branch*) es un bloque básico de un todo que puede ser seleccionado para su ejecución. Dicha selección se realiza entre un listado de posibilidades disponibles (ISTQB, 2023). 30

**rendimiento** El rendimiento (*performance*) es el grado en el que un sistema o componente cumple con las funciones designadas dentro de un plazo determinado (IEEE, 1990). 1, 5, 54, 55, 105

**requerimiento** Una condición o capacidad requerida por un usuario para resolver un problema o para alcanzar un objetivo que debe ser cumplida o poseída por el sistema o el componente de un sistema para satisfacer el contrato, estándar, especificación u otro documento formalmente impuesto (ISTQB, 2023). 4, 49, 57, 111

**SaaS** Software as a Service (Software como servicio) es un modelo de licenciamiento para software donde se ofrece un servicio final, a través de una aplicación o software, y no se tiene acceso de administrar los recursos. El precio se paga usualmente por suscripción. 36, 117

**Scrum** Scrum es un marco de trabajo orientado de proyectos orientados en agilidad que ayuda a los equipos desarrollar proyectos en torno a valores, principios y prácticas definidas. 19, 20

**Single Source of Truth** El principio de *Single Source of Truth* (Única Fuente de Verdad) es un concepto utilizado para definir una sola fuente única de datos, la cual se considerará la fuente de datos autorizada y veraz. 17

**skeleton loaders** Los *skeleton loaders* (esqueletos de carga) son componentes visuales que intentan reservar el espacio que ocuparán los componentes una vez finalicen su carga. 56

**soft delete** El *soft delete* (borrado suave) es una acción para deshabilitar un registro sin eliminarlo, mediante el cual se agrega una bandera al registro y el mismo no es parte de los cálculos ni listas. 58

**SSR** *Server Side Rendering* (Renderizado a nivel de servidor) es una técnica donde los componentes visuales se renderizan en el servidor y son enviados listos para su carga al cliente. 56

**usabilidad** La usabilidad es el grado en que un programa o componente pueden ser utilizados por usuarios específicos para alcanzar metas específicas dentro de un contexto de su uso (ISTQB, 2023). 6

**velocidad** La velocidad es una métrica que analiza la cantidad de trabajo que puede ser entregado durante una iteración (Beck, 2013). 25, 28

**VPS** Un VPS o *Virtual Private Server* (Servidor Virtual Privado) es un tipo de servicio donde se despliegan elementos virtualizados (servidores) proporcionando un servidor virtual dedicado. 29, 66, 117

**XP** "eXtreme Programming" (XP) es un marco de trabajo para el desarrollo ágil de proyectos desarrollado por Kent Beck en la década de 1990 (Beck, 2013). 19

## Resumen

La donación de sangre, un acto esencial y altruista en el ámbito de la salud, a menudo enfrenta desafíos derivados de la falta de comprensión por parte de la población. A pesar de las diversas campañas emprendidas en Panamá para fomentar la donación de sangre, la persistente escasez de unidades y centros vacíos constituye una preocupación constante. Este Trabajo de Fin de Máster se centra en abordar estas problemáticas mediante el desarrollo de una Prueba de Concepto para una solución basada en desarrollo de software destinada a la web y dispositivos móviles.

La propuesta tecnológica abarca el desarrollo de varios sistemas que conformarían una solución integral en torno a esta temática. Incluye una capa de servicios web, un panel administrativo y una aplicación web responsive, todos desarrollados en el *framework* Laravel. Asimismo, aplicaciones móviles multiplataforma para dispositivos Android e iPhone, desarrolladas con React Native.

Este trabajo aborda todos los aspectos relacionados con el desarrollo de software desde una perspectiva realista, incorporando prácticas para la gestión de proyectos con enfoque ágil, técnicas de DevOps, implementación de mejores prácticas de desarrollo, integración de diversas tecnologías y soluciones, entrega de documentación conforme a estándares, pruebas automatizadas, y adhesión a mejores prácticas y estilos de código. Además, se incluyen pruebas en dispositivos reales, mejoras de rendimiento, conceptos de seguridad en distintos entornos, la protección de datos personales según la legislación nacional y el seguimiento de mejores prácticas de desarrollo de interfaces.

**Palabras clave:** donación de sangre, Panamá, multiplataforma, aplicación móvil, aplicación web, laravel, react, react native

# 1. Introducción

La atención médica y las transfusiones de sangre están estrechamente relacionadas. Los pacientes que se someten a cirugías o tratamientos que implican pérdida de sangre, así como aquellos con lesiones, enfermedades u otros trastornos sanguíneos, a menudo necesitan transfusiones de sangre para recuperarse.

Para llevar a cabo estas transfusiones, es esencial contar con donantes de sangre. En el Sistema de Salud Pública de Panamá (Ministerio de Salud y Caja de Seguro Social), los procesos de donación son coordinados por los Bancos de Sangre. Estos bancos reciben apoyo de organizaciones sin fines de lucro (ONG) y entidades públicas privadas que se dedican a fomentar la donación voluntaria de sangre y plaquetas en Panamá.

La donación de sangre es un acto completamente voluntario y altruista en el que el donante contribuye con su propia sangre, para que sea almacenada en los bancos de sangre. Posteriormente, esta sangre se utiliza en los procedimientos de transfusión. Esta práctica es fundamental para la atención médica, ya que contar con suficiente suministro de sangre es vital para la supervivencia de muchos pacientes.

Al donar sangre, los individuos desempeñan un papel crucial en el proceso de curación y recuperación de quienes enfrentan condiciones médicas graves. Su generosidad contribuye directamente a salvar vidas y a garantizar que los servicios médicos puedan atender de manera efectiva a quienes más lo necesitan.

A nivel de la donación de sangre, existen dos tipos de donaciones: donación por reposición y donación voluntaria. Los donantes por reposición son aquellos que donan para reponer las unidades suministradas o por suministrar a un paciente en específico. Por otro lado, los donantes voluntarios son aquellos que donan para la incorporación de dichas unidades al sistema central o los bancos de sangre.

Para suplir la demanda de sangre continua, es necesario contar con más donantes voluntarios, los cuales permiten tener unidades de sangre disponibles en casos de urgencia, emergencia o cuando el paciente requiera una transfusión que no ha sido planificada con anterioridad.

En este contexto, este **Trabajo de Fin del Máster Universitario en Desarrollo de Aplicaciones y Servicios Web**, se enfoca en analizar la creación de una aplicación que incentive, promueva e incremente la donación de unidades de sangre y plaquetas por voluntarios en Panamá. La propuesta incluye el desarrollo de una capa de servicios web utilizando el *framework* Laravel (Laravel, LLC, 2011) y una

Base de Datos PostgreSQL (The PostgreSQL Global Development Group, [2023](#)). Esta aplicación funcionará como un sistema centralizado de comunicación para una aplicación multiplataforma en Android y iOS, desarrollada con React Native (Meta Platforms, Inc., [2023](#)).

## 1.1. Motivación del proyecto

En Panamá, **solo 7 % de los donantes son voluntarios** (Gaceta Oficial, República de Panamá, [2020](#)), lo que crea una constante escasez de unidades que puedan ser utilizadas en aquellos pacientes que ingresan por una urgencia o emergencia.

**El restante 93 % corresponde a donación por reposición.** Esto quiere decir, que solo el 3 % de la sangre que ingresa a los bancos de sangre públicos es suplida a través de donantes voluntarios y que no están orientadas a un paciente en particular.

En la actualidad, si bien existen aplicaciones móviles que incentiven la donación de sangre (American Red Cross, [2020](#)), es importante recalcar que **no existe ninguna aplicación similar orientada al mercado panameño ni que siga las regulaciones ni los procedimientos establecidos según la legislación panameña** (Gaceta Oficial, República de Panamá, [1986](#)).

El déficit en los Bancos de Sangre en Panamá es un tema recurrente, y las autoridades han alertado sobre el peligro para la salud pública (Gaceta Oficial, República de Panamá, [2020](#)) (Editorial Panamá América, [2022](#)). Uno de los mayores desafíos es convertir a personas en nuevos donantes voluntarios y fomentar la donación recurrente de los que han sido aptos con anterioridad.

El proceso de donar puede resultar engorroso para algunos, ya que no existe una forma efectiva para agilizar la programación de citas, los requisitos son confusos y no hay un método de recordatorio para promover nuevas donaciones recurrentes. La solución propuesta busca desmitificar la donación de sangre, facilitar la programación de citas, permitir solicitudes de donantes y digitalizar la confirmación de las donaciones realizadas.

La solución no solo ofrecerá distintas plataformas tecnológicas accesibles a través de múltiples canales, sino que también será una herramienta de valor que proporcionará información sobre los centros de donación y ofrecerá cápsulas informativas sobre el proceso, todo ello manteniendo la seguridad y privacidad de los usuarios.

## 2. Objetivos

El objetivo principal del presente Trabajo de Fin de Máster es desarrollar una PoC (Prueba de Concepto) para una aplicación móvil multiplataforma, desplegada tanto para dispositivos Android como iOS, que permita la adopción de nuevos donantes voluntarios de sangre y plaquetas en Panamá, mediante el despliegue y verificación de los requisitos de donación, permitiendo la generación de citas en los centros de donación.

Además de los donantes nuevos, se requiere incorporar funcionalidades orientadas a donantes voluntarios anteriores, para que los mismos puedan convertirse en donantes recurrentes, considerando la legislación actual.

Además, desarrollar otros mecanismos para incentivar, promover, fidelizar y eliminar algunos mitos sobre el proceso de donación, como la muestra de cápsulas informativas sobre los beneficios de la donación de sangre y generación de estadísticas.

El resultado final será crear de una Prueba de Concepto (PoC) de sistemas interdependientes que podrían ser utilizados por las autoridades, centros particulares y demás fundaciones sin fines de lucro para promover la donación de sangre, según las regulaciones y reglamentaciones emitidas por el Ministerio de Salud de Panamá y en consonancia con las Leyes de Protección de Datos.

### 2.1. Objetivos tecnológicos

Como propuesta para el desarrollo de este proyecto, se escogerán distintas tecnologías actuales, tanto a nivel del *backend* como a nivel del *frontend*. Se optará por tecnologías libres y abiertas sobre privativas o que requieran licencia.

Por ello, para el desarrollo del *backend* se ha decidido utilizar Laravel, un *framework* en PHP que nos permitirá el desarrollo de una capa de servicios. Dicha capa de servicios se conectará a una base de datos PostgreSQL, la cual almacenará de forma centralizada todos los datos de la aplicación.

A nivel del desarrollo *frontend* se ha escogido React y React Native, desarrollados utilizando lenguaje TypeScript. React es una librería que permite el desarrollo de interfaces visuales utilizando JavaScript o TypeScript. React Native, por su parte,

es un *framework* destinado al desarrollo de aplicaciones e interfaces orientadas a dispositivos móviles, lo que nos permitirá a través de un solo lenguaje base acceder al desarrollo para distintas plataformas.

## 2.2. Requisitos del proyecto

Para los requerimientos y el alcance de este Trabajo de Fin de Máster se han utilizado requerimientos de alto nivel y se han priorizado los mismos aplicando la técnica MoSCoW, que define los siguientes niveles de prioridad (Anand & Dinakaran, 2017):

- **Must have** (Tiene que tener): No negociable, debe ser parte.
- **Should have** (Debe tener): Importante, no vital, pero que añade valor.
- **Could have** (Puede tener): Ideal, pero no reduce el valor si no se tiene.
- **Will not have** (No tendrá): No son prioridad.

Estos temas son de muy alto nivel, ya que al proponerse una gestión con pensamiento ágil, los mismos serán evaluados con mayor detalle nuevamente a medida que mayor entendimiento y conocimiento sobre los mismos sea refinado.

Desde la perspectiva de **valor generado en torno a los usuarios**, se han identificado los siguientes temas o requerimientos de muy alto nivel, mostrados en la figura 2.1:

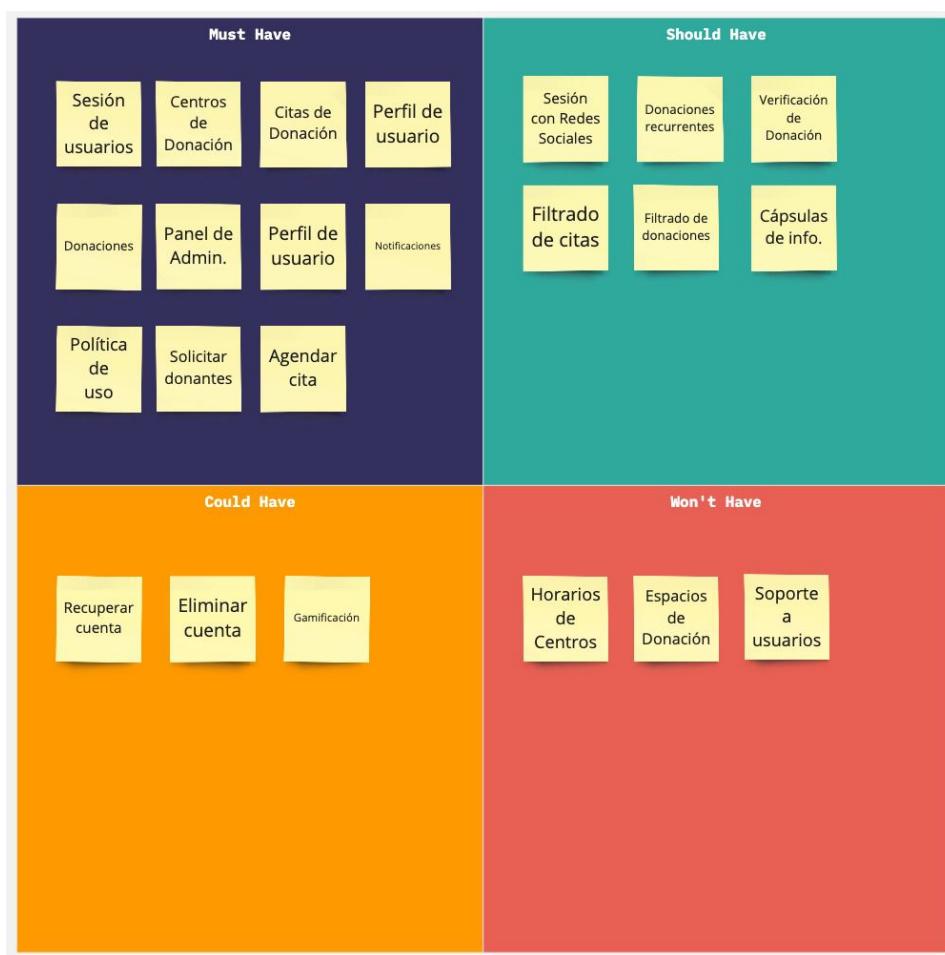


Figura 2.1: Requerimientos de muy alto nivel. Fuente: Elaborado por el autor.

Estos temas serán compartidos por todos los componentes que son parte de la PoC. Los mismos serán desglosados en un modelo ágil e incremental de desarrollo, donde la prioridad de los mismos, en torno a valor, referenciará el tema definido.

## Requerimientos no funcionales

Adicionalmente al segmento de objetivos funcionales de la solución, deben ser analizados los requerimientos no funcionales que se esperan de la implementación. Estos son requerimientos que no están relacionados con las funcionalidades del sistema y qué debería hacer el mismo (ISTQB, 2023).

- 1. Rendimiento (percibido):** La carga inicial de cualquier página, o vista, dentro del sistema debe ser inferior a 2 segundos para una conectividad estable y

de banda ancha. Si bien la carga final pueda tomar más, la experiencia inicial deberá ser menor a 2 segundos.

- 2. Seguridad:** La aplicación debe cumplir con estándares de seguridad, velando por proteger ante ataques de inyección de SQL, denegación de servicio ante las rutas de acceso público, las rutas seguras deben requerir un token de sesión y cualquier acción destructiva del usuario deberá ser validada por un factor de doble autenticación. Un reporte de pruebas penetración ante las 10 vulnerabilidades más comunes, según la OWASP, al año de la fecha del reporte deberá ser incluido.
- 3. Mantenibilidad:** Todos los componentes del sistema deben mantener uniformidad en su estructura, por lo cual se deberán implementar mecanismos para asegurar el estilo de código, el cual deberá ser de uso obligatorio.
- 4. Experiencia uniforme y consistente:** Con el fin de garantizar una experiencia coherente, es crucial implementar mecanismos que aseguren que los elementos y componentes mantengan uniformidad en sus características generales. Esto implica compartir de manera consistente atributos como colores, formas, tamaños, fuentes, espacios y bordes. Todos los sistemas que componen la solución deberán compartir estas características y solo se podrán utilizar las mismas al momento de su uso.
- 5. Legislación y privacidad:** Cualquier implementación que se realice deberá estar alineada a las legislaciones actuales de la República de Panamá en temas de salud pública y protección de datos personales. Cualquier librería por utilizar deberá orientarse en la protección a la privacidad de los usuarios.
- 6. Seguir las guías oficiales:** Las aplicaciones móviles deben seguir las guías oficiales de cada una de las plataformas. En el caso de iOS deben seguirse los *Human Interface Guidelines*<sup>1</sup> y en Android los *Design and Plan Guidelines*<sup>2</sup>.
- 7. Usabilidad en web:** El desarrollo web deberá seguir las mejores prácticas de usabilidad, definidos a través del “Web Content Accessibility Guidelines WCAG 2.1” (W3C, 2023).

<sup>1</sup><https://developer.apple.com/design/human-interface-guidelines>. Consultado el 14/01/2024.

<sup>2</sup><https://developer.android.com/design>. Consultado el 14/01/2024.

### 3. Marco tecnológico

La plataforma por desarrollar se compone de distintos componentes o subsistemas: una capa de servicios (*backend*), un sitio informativo (*frontend*), una aplicación móvil (*frontend* en móvil) y un sistema de base de datos.

Adicional a las tecnologías a utilizar para estos subsistemas, se escogerán tecnologías o plataformas de colaboración, como plataformas de repositorios, para la gestión del proyecto en línea y para el proceso de integración y despliegue continuo (CI/CD).

Para cada uno de los subsistemas se debe escoger tecnologías aptas, luego de pasar por un proceso de *benchmark* con otras tecnologías, por lo que se tomará en cuenta los siguientes criterios para su selección, alineados a los requerimientos no funcionales:

- 1. Licenciamiento:** Se escogerá software libre sobre software privativo, entendiéndose por software libre aquel que permite a sus usuarios ejecutar, copiar, distribuir, cambiar y mejorar el software (Stallman, 2020).
- 2. Mantenimiento, soporte y comunidad:** Se considerará si la tecnología se encuentra en constante actualización, si cuenta con soporte (por lo menos a nivel de comunidad) y si la comunidad y la calidad de documentación disponible. Para medir este punto, se utilizarán estadísticas vigentes de GitHub<sup>1</sup> y StackOverflow<sup>2</sup>, dos plataformas comunitarias para desarrolladores de software.
- 3. Interoperabilidad** como los subsistemas deben trabajar en conjunto y entre sí, es importante que exista una buena interoperabilidad entre ellos. La misma se entiende como la capacidad de dos o más componentes de software para cooperar a pesar de las diferencias entre las tecnologías, interfaces y plataformas de ejecución (Wegner, 1996).
- 4. Familiaridad:** la familiaridad con la tecnología también jugará un rol importante, siendo el caso que es fundamental escoger una tecnología con la que el proceso de desarrollo sea ágil y continuo.
- 5. Seguridad.**

<sup>1</sup><https://github.com>. Consultado al 30/01/2024.

<sup>2</sup><https://stackoverflow.com>. Consultado al 30/01/2024.

### 3.1. Métodos existentes para donar hoy en día en Panamá

En la actualidad existen mecanismos para cuando una persona desea o requiere realizar una donación de sangre o plaquetas en Panamá, ya sea de reposición o para agregarla al banco central de una institución o centro de donación.

No obstante, dichos mecanismos no interoperan entre ellos y la información se encuentra dispersa y en distintas formas. En el siguiente cuadro se muestra una comparativa de distintos métodos actuales:

	Fundación Dona Vida	Caja de Seguro Social	Centros Privados
<b>Requisitos en línea</b>	✓	✓	✓
<b>Agenda en línea</b>	✓	✗	✗
<b>Agenda presencial</b>	✓	✓	✓
<b>Solicitud de donantes</b>	✗	✗	✗
<b>Notificaciones sobre cita</b>	✗	✗	✗
<b>Notificaciones sobre donación</b>	✗	✗	✗
<b>Aplicación móvil</b>	✗	✗	✗
<b>Múltiples centros</b>	✓ <sup>1</sup>	✗ <sup>2</sup>	✗
<b>Boleta de donación digital</b>	✗ <sup>3</sup>	✗ <sup>3</sup>	✗ <sup>3</sup>

Cuadro 3.1: Métodos actuales de donación

<sup>1</sup>En Fundación Dona Vida permiten donar tanto al Instituto Oncológico Nacional como al Hospital Santo Tomás (ambos en Ciudad de Panamá).

<sup>2</sup>En el Centro de Donación de la Caja de Seguro Social se puede donar a cualquier policlínica u hospital de la Caja de Seguro Social (a nivel nacional).

<sup>3</sup>La boleta de confirmación se emite en formato físico (papel), propenso al deterioro y a la pérdida de este. Asimismo, no es posible validar una donación. Las boletas de donación se utilizan para validar una donación, ya sea para la reposición de una

unidad o para el permiso de trabajo por parte del empleador.

Como se puede evaluar, **no hay ningún método en la actualidad y disponible en Panamá** que brinde mecanismos para conocer sobre todos los centros de donación de sangre a nivel nacional, agendar citas en línea en cualquiera de ellos, recibir la boleta de confirmación de donación de forma digital ni tampoco solicitar ayuda para la donación en un caso de emergencia médica.

## 3.2. Tecnologías utilizadas

### **Framework backend: Laravel**

Los *frameworks* de backend están asociados principalmente con los lenguajes de programación que puedan ser ejecutados a nivel del servidor. Java, PHP, Python, JavaScript, son lenguajes de programación comunes a nivel del *backend*.

Dicho esto, se ha escogido PHP como lenguaje de referencia, ya que PHP es un lenguaje ampliamente utilizado, analizado y probado para el desarrollo de aplicaciones en el lado del servidor.

Cuadro 3.2: Distintos frameworks en PHP

Framework	Licencia	Estrellas en GitHub	Interoperabilidad	Seguridad
Laravel	MIT	Más de 75,000	✓	✓
Symfony	MIT	Más de 28,800	✓	✓
CodeIgniter	MIT	Más de 18,200	✓	✓

Nota: Estrellas en repositorios consultados al 17/10/2023.

Los tres frameworks poseen características similares, siendo Laravel el escogido por presentar una comunidad más grande, mejor documentación y una interoperabilidad nativa con otros sistemas.

Laravel es un *framework* backend de código abierto y libre para el desarrollo de aplicaciones web que utiliza el lenguaje PHP. Fue creado por Taylor Otwell y su primera versión fue presentada en junio de 2011 (Surguy, 2013).

Adicionalmente, Laravel es considerado seguro (Vanderlei et al., 2021) contra los ataques y vulnerabilidades más comunes.

A diferencia de CodeIgniter y Symfony, Laravel soporta nativamente algunas librerías (Vue y React) para el desarrollo de frontend (Laravel, LLC., 2023).

## Motor de Base de Datos: PostgreSQL

A nivel del motor de base de datos, es importante considerar los motores de bases de datos compatibles con el *framework* escogido (Laravel).

Laravel soporta distintos motores como MySQL, MariaDB, PostgreSQL, SQLite y SQL Server.

Para el análisis se escogen MySQL, MariaDB, PostgreSQL y SQL Server (2017).

Partiendo con el análisis, se realiza una verificación de los distintos tipos de licenciamiento para los sistemas de bases de datos.

Cuadro 3.3: **Bases de Datos - Licencias**

Motor	Licenciamiento	Costo?	Compatible con Laravel?
MySQL 5.7+	GPLv2 o propietario	Depende	✓
MariaDB 10.10+	GPLv2	✗	✓
PostgreSQL 3.8.8+	PostgreSQL License	✗	✓
SQL Server 2017+	Privativo	✓	✓

Partiendo por el licenciamiento, se descarta a SQL Server. Si bien es cierto que la base de datos SQL Server (Microsoft) cuenta con un rendimiento óptimo, representaría costos a nivel de uso y su licencia es privada.

La base de datos MySQL cuenta con dos tipos de licenciamiento y costos. Para la versión de la comunidad (Community Edition) el licenciamiento es GPLv2 y el uso es libre. MariaDB es un *fork* de MySQL y con características más recientes.

A nivel del análisis de rendimiento, PostgreSQL, en entornos iguales, suele tener mejor rendimiento al compararse contra MySQL y MariaDB (Kroc et al., 2020).

Al ser PostgreSQL un motor de bases de datos con un gran rendimiento, con licenciamiento libre / abierto (The PostgreSQL Global Development Group, 1996), la cual está admitida como una licencia libre y abierta aprobada (Open Source Initiative, 2009), además de no contar costos asociados para su utilización, se escoge el mismo para su implementación.

PostgreSQL es un sistema para la gestión de bases de datos, distribuido bajo licencia abierta y libre, el cual utiliza multiprocesos (en vez de multihilos) lo que lo hace garantizar la estabilidad del sistema y lo hace fuerte ante fallas (Ordóñez et al., 2017). Su primera versión fue presentada en 1996.

## Librería *frontend*: React

A nivel de librerías para el frontend, se tiene un amplio segmento de posibilidades: React, Angular, Vue y Svelte se consideran las librerías más utilizadas en la actualidad para el desarrollo a nivel de frontend (Stack Overflow, 2023a).

Si se analiza su popularidad, en la figura 3.1 es posible ver el porcentaje de preguntas orientadas a una librería específica en Stack Overflow por año (Stack Overflow, 2023b), entre los años 2008 a 2023.

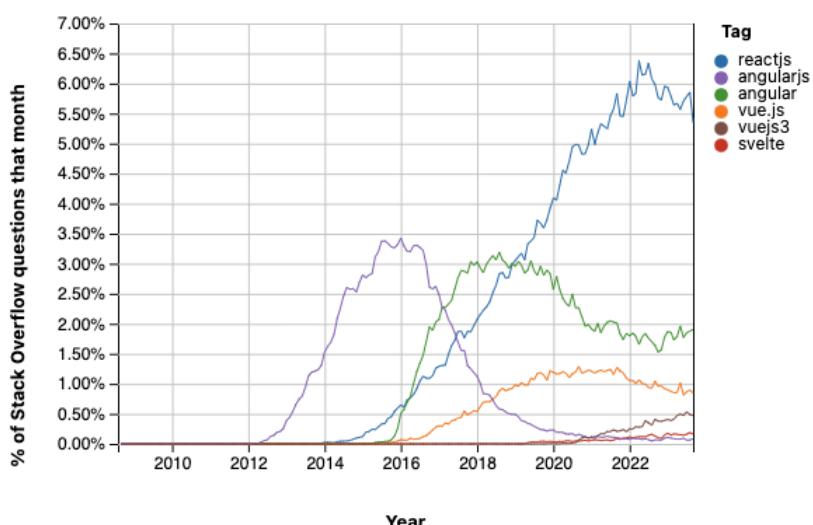


Figura 3.1: Popularidad librerías FE. Fuente Stack Overflow (2023).

Partiendo desde la popularidad de estas librerías, y al seleccionar las tres más populares (React, Angular y Vue), se pueden analizar sus licencias e interoperabilidad. Estas tres librerías permiten su desarrollo con TypeScript o JavaScript.

De estas tres librerías, se descarta AngularJS al ser la única que no posee una integración nativa con Laravel. Tanto Vue como React poseen una integración nativa, desde cero, con Laravel (Laravel, LLC., 2023).

Entre estas dos tecnologías restantes, se escoge React al ser más popular según

Cuadro 3.4: **Librerías FE - Licencias**

Librería	Licenciamiento	Interopera con Laravel
ReactJS	MIT	✓(nativamente)
AngularJS	MIT	✓(no nativamente)
Vue	MIT	✓(nativamente)

lo mostrado en los puntos anteriores.

React fue desarrollada por Meta (Facebook) y su primera versión fue presentada en 2013.

## **Framework móvil: React Native**

Los tipos de aplicaciones móviles pueden ser: aplicaciones web, aplicaciones nativas o aplicaciones híbridas (Delía et al., 2013).

1. Las **aplicaciones nativas** son aquellas que se ejecutan dentro de una plataforma o dispositivo específico y que son desarrolladas en lenguaje nativo (o compilado a lenguaje nativo).
2. Las **aplicaciones web** requieren su ejecución a través de un explorador web y utilizan tecnologías web (HTML, CSS, JavaScript).
3. Las **aplicaciones híbridas** son aplicaciones web que se incorporan a través de un contenedor web sobre el dispositivo móvil.

A nivel de aplicaciones nativas, estas se desarrollan en el lenguaje propio de la plataforma (por ejemplo, Swift u Objective-C para iOS) o en un lenguaje intermedio que permite el despliegue multiplataforma.

En este Trabajo de Fin de Máster (TFM), se necesita desarrollar una aplicación multiplataforma, lo que implica el despliegue en iOS y Android con un único código fuente para mayor eficiencia.

Entre los *frameworks* más conocidos para el desarrollo nativo multiplataforma se encuentran React Native<sup>3</sup> y Flutter<sup>4</sup>.

<sup>3</sup><https://reactnative.dev/>

<sup>4</sup><https://flutter.dev/>

Dado que React Native utiliza React (no ReactDOM) y los lenguajes preseleccionados en este trabajo, junto con sus ventajas como una curva de aprendizaje más baja, una buena experiencia de desarrollo y la facilidad de compartir entre 80-90 % del código en iOS y Android (Calixto et al., 2019), se elige React Native como el framework para el desarrollo de las aplicaciones móviles.

React Native fue desarrollado por Meta (Facebook), es entregado bajo licencia MIT y su primera versión fue desplegada en 2015.

## Expo

Expo (650 Industries, Inc., 2023) es una plataforma para el desarrollo de aplicaciones nativas en iOS y Android, desarrollada por 650 Industries, Inc. Sirve como capa intermedia para aplicaciones construidas con React Native, facilitando el acceso a funciones que de otra manera requerirían extenso desarrollo y pruebas.

Expo destaca en la capacidad para agregar características, paquetes y subsistemas adicionales que mejoran significativamente la experiencia de desarrollo móvil (Seignard, 2023), haciendo tareas como empaquetado, firmado y pruebas considerablemente más simples y accesibles.

## Git y Gitlab

Para el manejo del control de versiones se utilizará Git (Linus Torvalds, 2007). Git es un sistema de control de versiones libre y de código abierto desarrollado por Linus Torvalds y publicado por primera vez en 2007.

Como plataforma de administración y gestión del sistema de control de versiones Git se escogió Gitlab (Gitlab, Inc., 2014), una plataforma de DevOps desarrollada por Gitlab, Inc. y presentada públicamente en 2014. Además, GitLab posee una versión de código libre y abierto.

Gitlab, además de tener una interfaz para la gestión del sistema de control de versiones, posee módulos y funcionalidades orientadas a la integración y entrega del software de forma iterativa e incremental (CI/CD).

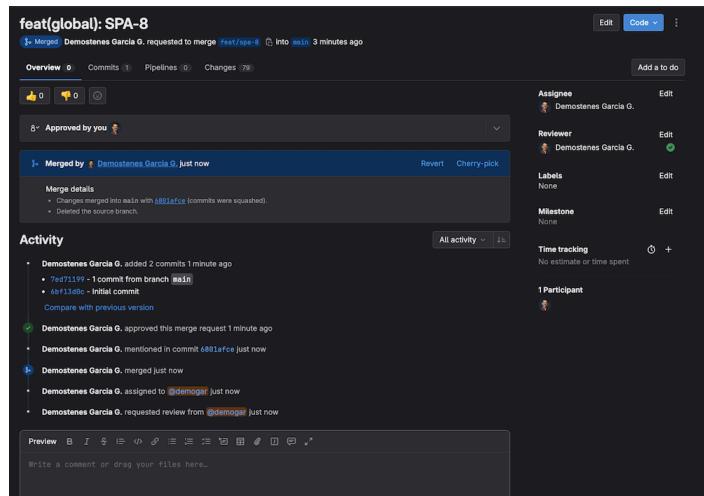


Figura 3.2: Pull Request en Gitlab. Fuente: Elaborado por el autor.

### **Conventional Commits**

Todos los repositorios se adherirán e implementarán *Conventional Commits*. *Conventional Commits* (*Commits convencionales*) es una especificación técnica para la creación de mensajes de *commit* con una estructura establecida. Estos mensajes proporcionan una forma de identificar el propósito de un cambio dado.

```
~/projects/masw-tfm/donasangrepanama-be git:(feat/spa-25) ✘70
git cz
cz-elliot4.3.0, cz-conventional-changelog3.3.0
? Select the type of change that you're committing: (Use arrow keys)
> feat: A new feature
fix: A bug fix
docs: Documentation only changes
style: Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc)
refactor: A code change that neither fixes a bug nor adds a feature
perf: A code change that improves performance
test: Adds missing tests or improves existing tests
(Move up and down to reveal more choices)
```

Figura 3.3: Commitizen-cli en su uso. Fuente: Elaborado por el autor.

Para asegurar su implementación, todos los repositorios utilizarán Commitizen<sup>5</sup> como herramienta de apoyo, disponible a través del comando:

```
1 yarn commit
```

Esto inicia un proceso para el *commit* que se muestra en la figura 3.3.

<sup>5</sup><https://github.com/commitizen/cz-cli>

## Inertia

La aplicación del *backend* funcionará, tanto para gestionar la capa de servicios como para mostrar una versión web de la aplicación y el panel de administración.

El API que se expondrá a través del *backend* tendrá el único propósito de proporcionar una capa de servicios para la aplicación móvil.

Para dicha arquitectura se escogerá una arquitectura de monolito. Una arquitectura de monolito es una arquitectura donde un sistema está compuesto de varios componentes distintos que trabajan en conjunto para definirlo como un todo (ELGHERIANI & AHMED, 2022).

Para que el monolito interactúe entre el *backend* y el *frontend* se ha escogido Inertia, una tecnología web que funciona con Laravel y que no requiere el uso de un API para la comunicación entre el *backend* y el *frontend* (Inertia.js, 2019).

### 3.3. Diseño de la interfaz

La interfaz de la aplicación es un elemento fundamental ya que es el punto de entrada hacia nuestros usuarios. Para un diseño de interfaz, según (Sosa-Tzec, 2021), un diseño debe ser agradable, utilizable, confiable y funcional.

Es por ello por lo que para el diseño preliminar se inició por el desarrollo de un **sistema de diseño (Design System)**, que es una colección de elementos documentados que incorporan los principios y reglas de diseño (Macdonald & Putnam, 2023).

El Design System va a compartir una serie de *design tokens*: una serie de llaves-valores donde se definen variables y constantes relacionadas a atributos del diseño como colores, espacios, tamaños de letras, entre otros. Esto asegura el cumplimiento del requerimiento no funcional para mantener una experiencia uniforme.

### Sistema de Diseño: Plaqueta DS

El sistema de diseño móvil se basará en Tamagui<sup>6</sup>, un sistema de diseño para móvil con componentes en React que incorpora elementos predefinidos para

---

<sup>6</sup><https://tamagui.dev>. Consultado el 28/10/2023.

construcción de interfaces de usuario, el cual está desarrollado en React y React Native.

Por su parte, el diseño para el sitio informativo y el panel del usuario (web) se basará en DaisyUI<sup>7</sup>, una librería de componentes para la web que despliega componentes en React.

Este **sistema de diseño** se llamará **Plaqueta DS**<sup>8</sup> y se incorporará en los distintos proyectos como una dependencia interna que tendrá dos versiones: una versión para React Native (móvil) y una versión que funciona con React DOM (web).



Figura 3.4: Logotipo de Dono Sangre. Fuente: Elaborado por el autor.

A nivel del desarrollo se debe configurar un repositorio específico y se utilizará NPM<sup>9</sup> para la publicación de este paquete de forma pública<sup>10</sup>.

Se hará uso de la librería style-dictionary<sup>11</sup>, desarrollada por el equipo de ingeniería de Amazon, para ayudarnos en el proceso de transformar un archivo JSON con los tokens a distintos archivos consumibles en nuestros distintos sistemas.

A partir de estos tokens se desarrollan componentes en cada plataforma para mantener la uniformidad en cada ambiente pero una experiencia global uniforme. Por ejemplo, se desarrollan componentes como PlaquetaButton (web y móvil), PlaquetaTextInput (web y móvil), PlaquetaScrollView (móvil).

<sup>7</sup><https://daisyui.com/>. Consultado el 28/10/2023.

<sup>8</sup>De Plaqueta Design System

<sup>9</sup>Node Package Manager

<sup>10</sup><https://www.npmjs.com/package/@demogar/plaqueta-ds>. Consultado el 12/11/2023.

<sup>11</sup><https://amzn.github.io/style-dictionary>. Consultado el 28/10/2023.

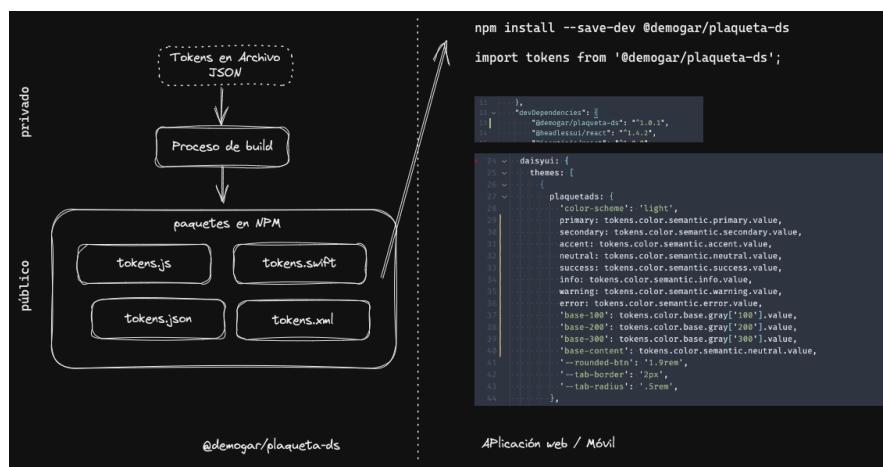


Figura 3.5: Proceso Tokens - Fuente: Elaborado por el autor.

En la figura 3.5 se aprecia el proceso de build y consumo de estos tokens en los sistemas. Gracias a este proceso se puede tener uniformidad entre el diseño presentado y las distintas plataformas, al consumir una única fuente de información (principio de Single Source of Truth).

De este modo, si se cambia el valor de alguno de estos valores el mismo se desplegará en **todas las plataformas que implementen el token** de forma automática.

### The Design System Efficiency Curve

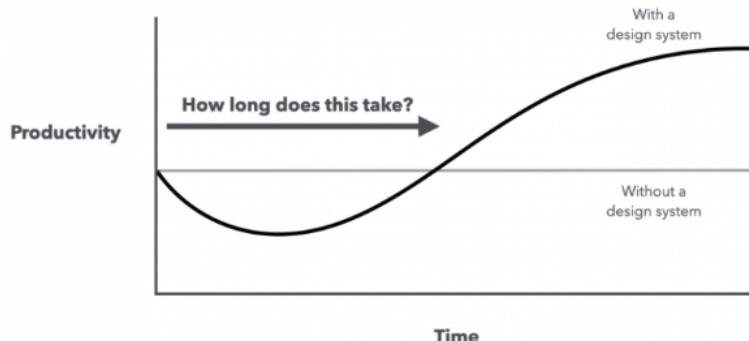


Figura 3.6: Productividad con Design System. Fuente: Callahan (2021).

Los beneficios de utilizar un sistema de diseño son amplios, aunque bien la productividad no se alcanza hasta cuando el sistema de diseño alcanza una cierta etapa

del desarrollo (Callahan, 2021), tal como lo muestra la 3.6 donde la productividad (eje y) es inicialmente menor con respecto al tiempo y el tiempo (eje x) , pero a medida que el tiempo avanza, y el sistema de diseño alcanza cierta madurez, la productividad incrementa.

Los tokens de diseño de PlaquetaDS se han desplegado como un paquete público en el repositorio de NPM<sup>12</sup>.

---

<sup>12</sup><https://www.npmjs.com/package/@demogar/plaqueta-ds>. Consultado el 13/01/2024.

## 4. Metodología

Un proyecto, orientado a la definición del PMI, es un **emprendimiento temporal** que se lleva a cabo para crear un **producto o servicio**. Es un proceso, con una **duración determinada** y un **fin concreto**, compuesto por actividades y tareas diferentes, que puede ser elaborado de manera gradual (Schwalbe, 2009).

Los proyectos deben ser gestionados (procesos) y administrados (recursos). Para la gestión y administración de proyectos de software existen dos grandes vertientes: proyectos estilo tradicional y proyectos ágiles.

Los proyectos ágiles se basan en adaptabilidad, por lo que se tienen ciertas ventajas sobre los proyectos tradicionales. Poseen múltiples beneficios, como lo son una mejor satisfacción del cliente con el producto final, mejor cooperación y flexibilidad dentro del proceso, lo que conlleva a reducir y mitigar ciertos riesgos dentro del proyecto (Hooda et al., 2023).

**Es importante recalcar que agilidad no se debe confundir con una metodología** (Doshi & Virparia, 2023) o un **marco de trabajo**, por lo que sería incorrecto utilizar el término "metodología ágil". Agilidad corresponde a un **enfoque** (*mindset*) de cómo se deben abordar los proyectos a partir de un manifiesto: el manifiesto ágil (Beck et al., 2001a).

Este manifiesto declara cuatro valores (Beck et al., 2001a) y doce principios para el desarrollo ágil de proyectos (Beck et al., 2001b), a partir de los cuales han surgido distintos marcos de trabajo como Scrum (Schwaber, 1997) o XP (Beck, 1999). De igual forma, sería incorrecto mencionar que cualquier marco de trabajo ágil es una metodología (Vazifeh-Noshafagh et al., 2022).

Para la gestión y administración de este proyecto se utilizarán enfoques ágiles y se utilizará la **base** del marco de trabajo Scrum, adaptándolo según las necesidades propias de este desarrollo<sup>1</sup>.

<sup>1</sup>La guía oficial de Scrum menciona distintos roles y ceremonias que deben realizarse. Este Trabajo de Fin de Máster ha sido elaborado por un único integrante, por lo que sería imposible cumplir con todo lo establecido en la guía.

## 4.1. Dirección y gestión del proyecto

Al cumplirse uno de los principios de agilidad ("Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo") se definirá un *backlog* inicial del proyecto y cada iteración (*Sprint*) del proyecto se refinará (*Refinement*), puntuará (*Estimations*) de forma individual.

Asimismo, al definirse un marco de trabajo basado en Scrum, se han definido como base los siguientes eventos (*Scrum Events*) y artefactos (*Scrum Artifacts*):

1. **Sprint** (Evento): Ciclo iterativo durante el cual el equipo trabajará comprometido para lograr la meta establecida durante la planificación. En nuestro caso, se ha establecido que el Sprint tendrá una duración de 1 semana calendario.
2. **Sprint Planning** (Evento): Correspondrá al proceso donde se definirá los entregables que el equipo se compromete a entregar. Se define una meta para el Sprint.
3. **Sprint Retrospective** (Evento): Buscará mejorar y aumentar la calidad y la efectividad del desarrollo, enfocándose en encontrar mecanismos para incrementar ambos. Se realizará en formato introspectivo.
4. **Product Backlog** (Artefacto): Correspondrá a la serie de historias de usuario que se han identificado para la solución final.
5. **Sprint Backlog** (Artefacto): Correspondrá a la serie de historias de usuario que permitirá alcanzar la meta establecida en el Sprint.
6. **Increment** (Artefacto): Incrementable del trabajo entregado que soporta la meta establecida y que demuestra progreso hacia la meta final del producto<sup>2</sup>.

Debido a la naturaleza de este trabajo, se han obviado algunos roles (*Product Owner*) y algunos eventos (*Daily Scrum*, *Sprint Review*).

**Más que una gestión de proyectos con el marco de Scrum, se podría decir que se realizará una gestión de proyectos con pensamiento y mejores prácticas ágiles.** Se podrán utilizar algunos términos del marco de trabajo Scrum, y de otros, pero no se deberá confundir con la implementación "estricta" de Scrum.

---

<sup>2</sup>Es importante recalcar que, según los principios de agilidad (Beck et al., 2001b), la medida principal del progreso es el software funcional, es por ello que la meta principal de cada Sprint estará asociada a entregar partes funcionales.

## Reglas establecidas

Al Scrum ser un marco de trabajo, el mismo plantea recomendaciones para su implementación, pero el equipo determinará la forma más eficiente de trabajo.

En el contexto de este trabajo, se han establecido las reglas mostradas en el Apéndice C.

## 4.2. Herramienta para la dirección del proyecto: Jira Software

Para la gestión del proyecto se ha seleccionado **Jira Software** (Atlassian, 2002). Jira es una aplicación web, creada por Atlassian y que ha sido desarrollada para la dirección de proyectos de todo tipo. Jira Software es la versión de Jira orientada a la gestión de proyectos de software.

Atlassian ofrece una versión gratuita de Jira Software, la que incluye características para la gestión de historias de usuario, épicas, *roadmap* (*timeline*), gestión del backlog de un Sprint (ver figura 4.1) e incluso integraciones con otras herramientas, como una integración nativa con GitLab (ver figura 4.2).

The screenshot shows a Jira Software interface for a sprint backlog. The sprint is titled "SPA Sprint 1" and spans from "24 Oct" to "31 Oct". There are 13 issues listed, each with a small green square icon, a title, a component, a status, and a due date. The components include "MOBILE - INFORMATION", "WEB - ARCHITECTURE", and "MOBILE - USER MANAGEMENT". The statuses include "TO DO", "IN PROGRESS", and "PENDING". The due dates range from "24 Oct" to "31 Oct". A "Create issue" button is visible at the bottom left.

Figura 4.1: Sprint en Jira. Fuente: Elaborado por el autor.

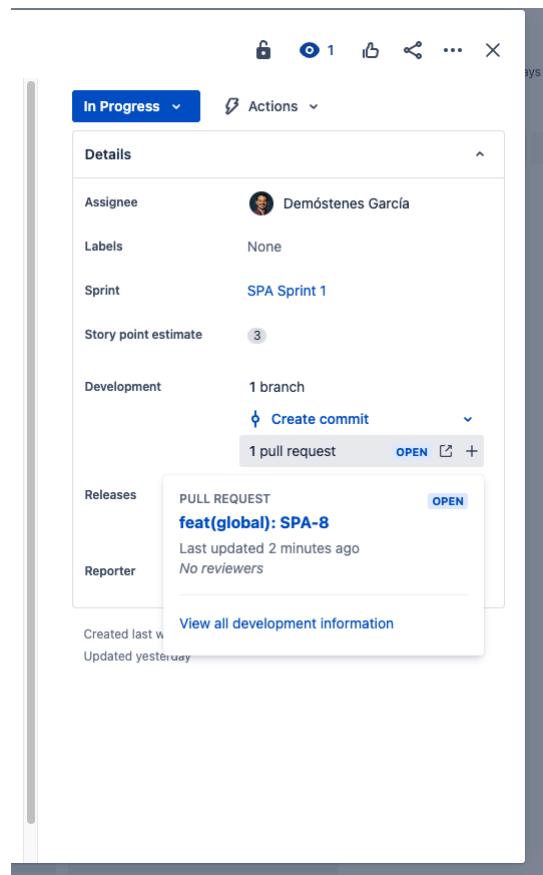


Figura 4.2: Jira - Integración con GitLab. Fuente: Elaborado por el autor.

En esta plataforma se creará la estructura del proyecto, partiendo desde las épicas, siguiendo hacia las historias de usuarios y a partir de estos dos se establecerá un *Roadmap* del producto con fechas (tentativas) establecidas.

### 4.3. Épicas e historias de usuario

Una historia de usuario describe funcionalidad que sería valiosa para el usuario del producto final y está compuesta por tres aspectos, según Cohn (2004):

1. Una descripción escrita de la historia de usuario, utilizada para planificar y como un recordatorio (de lo acordado).
2. Una conversación, donde se plasman los detalles de la historia.

3. Las pruebas o los criterios de aceptación mediante los cuales se puede considerar una historia como completada.

Cohn (2004) también señala que, cuando una historia es demasiado larga (historias cuya duración de trabajo exceden un tiempo que el equipo considere prudente), las mismas se considerarán como épicas.

En términos para el trabajo propuesto, se dividirá el trabajo en dos grandes grupos:

1. **Épicas:** constituirán grandes bloques de trabajo que podrán ser desmenuzados en otras tareas más pequeñas (historias de usuario). Las épicas, al constituirse en un grupo de historias de usuario, podrán abarcar distintos *Sprints*. Las mismas no tendrán estimación.
2. **Historias de usuario:** constituirán bloques de trabajo pequeños y específicos, cuya funcionalidad esté descrita, con criterios de aceptación específicos, refinados y estimados que se incorporarán a un *Sprint* en particular.

## Épicas

Para el Trabajo de Fin de Máster, se han definido las épicas descritas en el apéndice A. Estas constituyen grupos de historias de usuario. Estas épicas podrán no tener un identificador secuencial en la práctica ya que dependerá del momento en que han sido creadas en la herramienta que se ha definido para el trabajo, la cual se abordará más adelante. Dichas épicas son el resultado de los "temas" discutidos en los objetivos de alto nivel mostrados en la unidad 2.2.

El nombre del proyecto dentro de Jira es '**Sangre Panamá**' y su identificador único, que será utilizado para todos los prefijos (tanto de las épicas como de las historias), será '**SPA**' (de **Sangre Panamá**).

## Historias de Usuario: *Backlog* del primer *Sprint*

Tal como lo señalan los doce principios de agilidad (Beck et al., 2001b), **la agilidad abraza el cambio** (primer principio) y a intervalos regulares de tiempo **las personas que desarrollan el proyecto entienden mejor de él** (último principio).

Asimismo, se acepta que los requisitos cambien, incluso en etapas tardías del desarrollo (**segundo principio**).

En consecuencia, si bien se ha creado un *Backlog* del Producto inicial, se asume que el *Backlog* es una entidad viva que seguirá creciendo y cambiando a medida que el equipo de desarrollo entienda más del proyecto (Frederik Fowler, 2018). Dicho *Backlog* se mantendrá priorizado (**valor**) y se refinarán las historias durante el proceso de refinamiento del *Backlog* antes de un *Sprint*. En el Apéndice B se señala la definición del *Backlog* del *Sprint 1*.

Como bien lo señala Cohn (2004), **las historias de usuarios deben representar funcionalidad que sería valiosa para los usuarios**. Por ello, el formato en que se escribirán las mismas serán desde la perspectiva del valor que introducen al ser completadas.

Se agregarán los puntos (*Story Points* o estimaciones), los cuales serán relativos entre ellos.

Las historias de usuario no deberán, igualmente, representar todos los requerimientos dentro de la aplicación, si no lo puntos importantes de la conversación alrededor. En el Apéndice B se muestra el listado de historias priorizadas para el *Sprint 1*. Se ha asumido una carga inicial de puntos durante el *Sprint 1* y a medida que se siga desarrollando este valor se irá ajustando para representar la velocidad (se aborda durante la unidad 4.3.4 y 5.1) del equipo.

Las historias deben cumplir con la regla mnemotécnica<sup>3</sup> **INVEST** (Wake, 2003), proveniente de eXtreme Programming, la cual también es descrita por Cohn (2004):

- **Independiente:** Como sea posible, las historias no deberán tener otras dependencias.
- **Negociable:** No son requerimientos ni contratos, son conversaciones.
- **Valiosa:** Deben agregar algún valor al usuario.
- **Estimable:** Deben poderse estimar (en esfuerzo).
- **Small** (Pequeña): Si una historia es muy grande es mejor dividirla en pequeñas historias.
- **Testable** (Comprobable): Toda historia debe poderse probar o verificar de alguna forma.

<sup>3</sup>La Real Academia de la Lengua Española define una mnemotecnia como “procedimiento mental para facilitar el recuerdo de algo”. <https://dle.rae.es/mnemotecnia>. Consultado el 14/01/2024.

## Roadmap de producto

Para establecer la secuencia de la habilidad de las características (*features*) durante el desarrollo del proyecto se ha establecido un **Product Roadmap**.

Un *Product Roadmap* es un **documento estratégico** (por ende, establece los objetivos alineados a la organización) de alto nivel que alinea las características de un producto con la visión y dirección de un producto a lo largo del tiempo (ProductPlan, 2013).

Para nuestro **Product Roadmap** se utilizan las épicas como agrupadores. El mismo puede ser consultado a mayor detalle en el Apéndice D.

## Cálculo de velocidad del equipo

Inicialmente, al desconocer la cantidad de esfuerzo que puede ser entregado durante un *Sprint*, se agregarán las historias que se consideren puedan ser entregadas siguiendo las reglas y estimaciones de las historias.

A medida que se desarrollen nuevos *Sprints*, y se obtenga información de cuántos puntos realmente se están entregando por *Sprint*, se ajustará la velocidad del *Sprint* para que sea más realista.

## 4.4. Diagrama de la Base de Datos

Para almacenar y persistir la información entre sesiones y para mantener centralizada la misma entre distintos componentes que componen el proyecto, se ha hecho necesaria tener una estructura de base de datos, la cual se alimentaría a través de los distintos sistemas, ya sea a través de la aplicación web o a través de la capa de servicios del API.

Este sistema está basado en PostgreSQL, según lo discutido en la sección 3.2.2. En el mismo se muestran las relaciones entre distintas tablas como la existencia de entidades polimórficas, lo que conllevaría relaciones con distintos tipos de tablas, asociadas según las propiedades establecidas en su diseño. Las figuras 4.3 y 4.4 muestran los diagramas de clases establecidos.

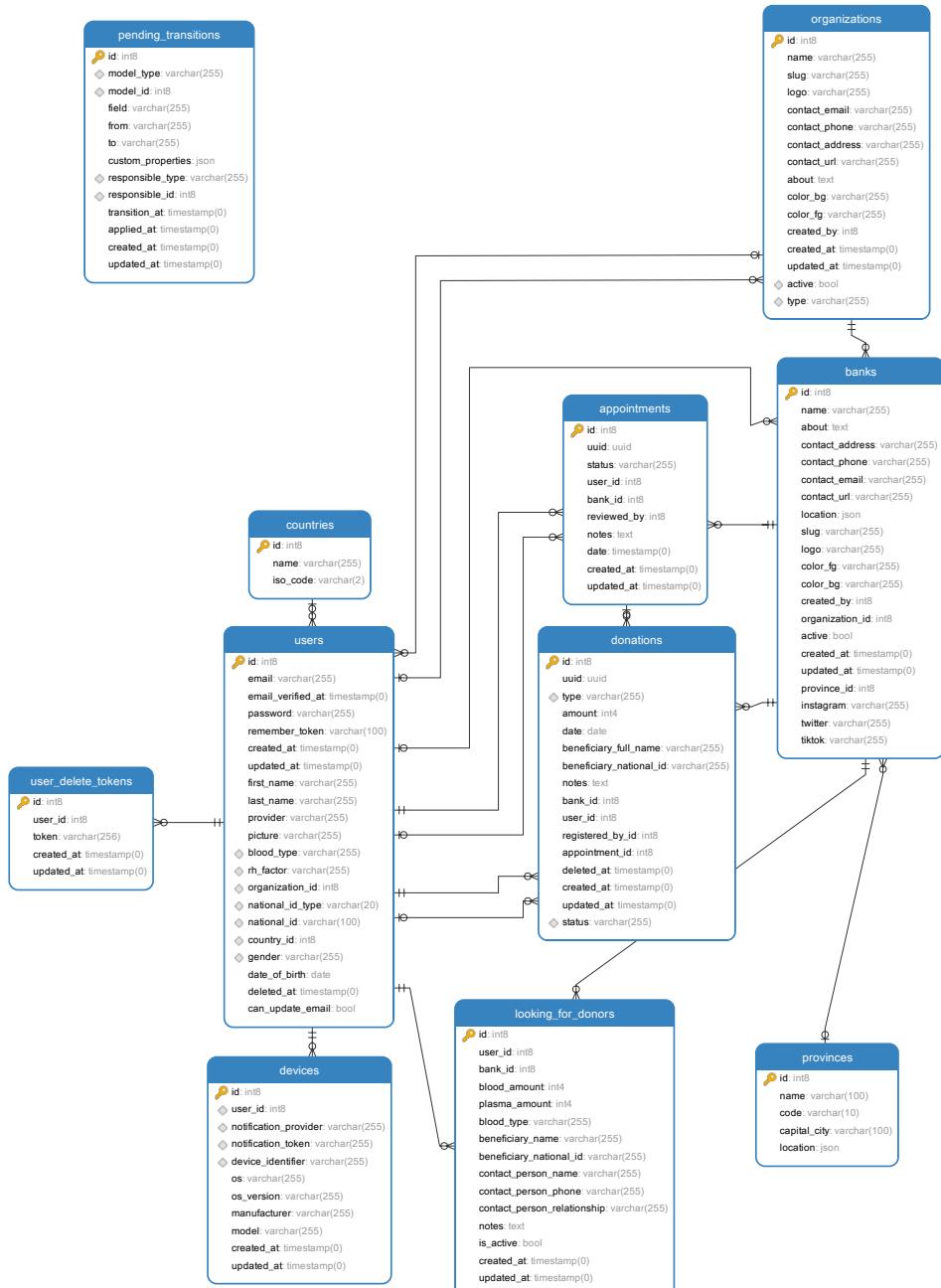


Figura 4.3: Diagrama de Base de Datos. Fuente: Elaborado por el autor.



Figura 4.4: Diagrama de Base de Datos (continuación). Fuente: Elaborado por el autor.

## 5. Resultados obtenidos

A continuación, se procederá a exponer lo realizado durante el proceso de desarrollo del alcance de este trabajo, durante todas sus fases:

1. Resultados en la **gestión y dirección** del proyecto.
2. Desarrollo del **sitio informativo y panel de administración**.
3. Desarrollo del **API**.
4. Desarrollo de la **aplicación móvil**.
5. **Despliegue** en entorno de prueba.

### 5.1. Gestión y dirección del proyecto

#### Resultado de velocidad del equipo

La velocidad del *Sprint* no puede ser calculada inicialmente puesto que se desconoce la relación entre esfuerzo calculado (estimación) y esfuerzo ejecutado por *Sprint*.

Como se señaló en 4.3.4, la velocidad se calcularía una vez los primeros *Sprints* fuesen ejecutado y a medida que el equipo perfeccionara las estimaciones.

Luego de ejecutar 7 *Sprints* se estableció con certeza que el resto del **proyecto se ejecutaría a una velocidad estimada de 23 puntos**. La evidencia de los 7 *Sprints* se muestra en el Apéndice E.

### 5.2. Ambiente, despliegue y CI/CD

Para el despliegue de la aplicación en un entorno de pre producción que asimile la puesta en producción, se han configurado los siguientes elementos:

- Se ha adquirido un dominio, que albergará el sitio web informativo, el panel de administración y será un punto de acceso al API. Dicho dominio es [yodonosangre.com](https://yodonosangre.com).
- Se ha configurado un VPS, el cual tendrá tres distintos servicios o servidores en funcionamiento:
  - Un servidor web, en nuestro caso Nginx<sup>1</sup>.
  - Un servidor de Base de Datos, en nuestro caso PostgreSQL.
  - Un servidor para caché, en nuestro caso Redis<sup>2</sup>.
- Dicho servidor se configurará en Linux, utilizando la distribución de Ubuntu Server 23.10 (Mantic)<sup>3</sup>.
- Se ha configurado un certificado HTTPS/SSL con Let's Encrypt<sup>4</sup> y Certbot<sup>5</sup>.
- Se ha configurado un subdominio (<https://dev.yodonosangre.com>) y se ha configurado el servidor HTTP (Nginx) con sus respectivos mapas.
- Se ha configurado un flujo de integración y despliegue continuo a través de GitLab y GitLab CI/CD, que se discutirá en el próximo apartado.

## CI/CD con GitLab CI/CD

Al ya tener acceso a GitLab, es posible utilizar el servicio gratuito de GitLab CI/CD para la implementación de un flujo de integración y despliegue continuo (CI/CD, por sus siglas en inglés).

El flujo de CI/CD constaría de tres *stages* (etapas):

- **build** (construcción): Se encargaría de la limpieza del entorno de integración continua y de la instalación de los paquetes y dependencias.
- **test** (prueba): Se encargaría de la ejecución de las pruebas del sistema de BE y de la verificación (*lint* en TypeScript y *code sniffer* en PHP) de los estilos.
- **deploy** (despliegue): Ejecutaría el proceso del despliegue de los servicios y partes del sistema en el entorno que le corresponde y cuando le corresponde.

<sup>1</sup><https://www.nginx.com/>. Consultado el 09/11/2023.

<sup>2</sup><https://redis.io/>. Consultado el 09/11/2023.

<sup>3</sup><https://releases.ubuntu.com/mantic/>. Consultado el 09/11/2023.

<sup>4</sup><https://letsencrypt.org/>. Consultado el 09/11/2023.

<sup>5</sup><https://certbot.eff.org/>. Consultado el 09/11/2023.

Cuando se menciona que en la etapa de **deploy** se hará el despliegue cuando corresponde, se hace referencia que solo se busca hacer despliegue cuando la integración del código es dentro del branch (rama) principal, en este caso en `main`.

En la figura 5.1 se muestra el flujo de CI/CD integrado en el proyecto desde GitLab.

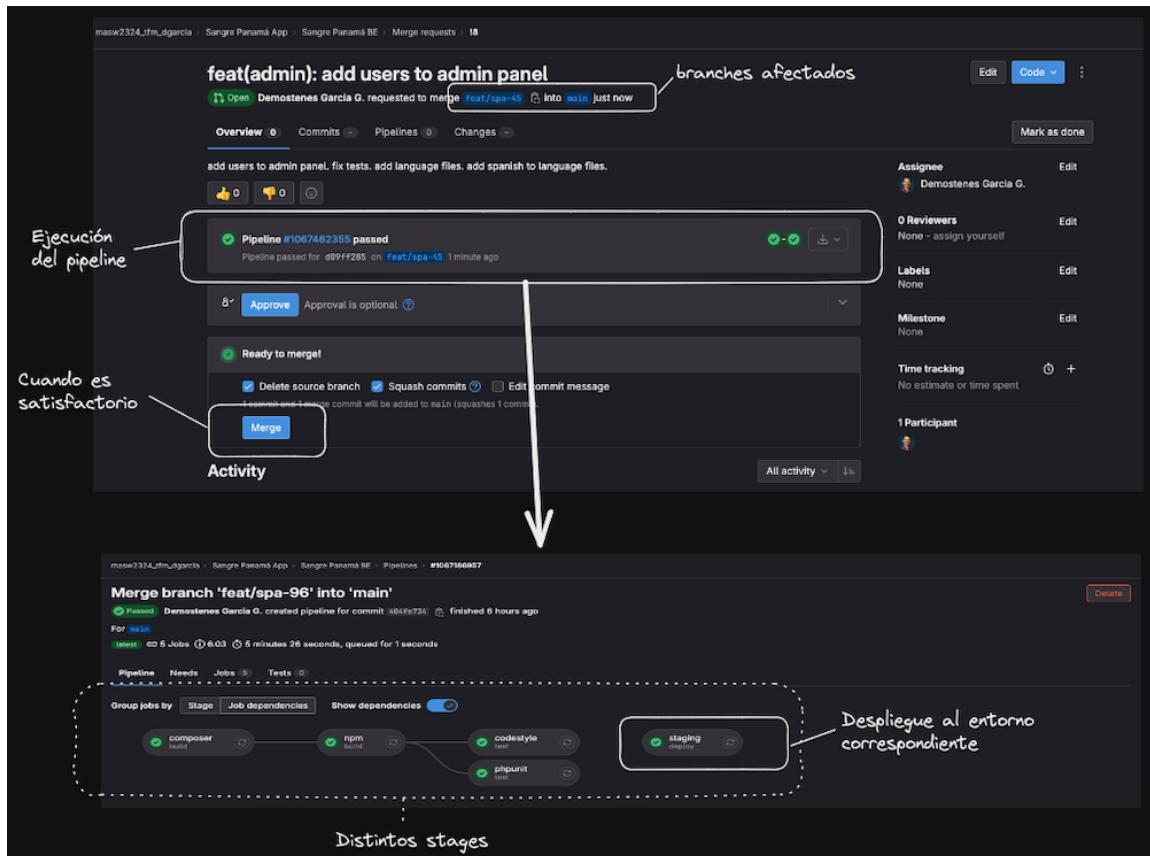


Figura 5.1: CI/CD en GitLab. Fuente: Elaborado por el autor.

Al implementar un flujo de CI/CD como este, se puede tener la certeza que:

- Todo el código se prueba localmente, antes de cualquier despliegue.
- Los estilos del código corresponden a los configurados dentro del proyecto.
- El entorno de pre-producción siempre cuenta con la última versión que se encuentra disponible en la rama `main`.

Para validar el último punto, se ha seguido un mecanismo de versionamiento

compatible con **Semantic Versioning**<sup>6</sup>.

*Semantic Versioning* es un mecanismo para comunicar la compatibilidad entre distintos componentes de un sistema (Lam et al., 2020). Al momento de cualquier despliegue, el sistema incrementará el build de referencia (un incremental), además de contar con un versionamiento que sigue el patrón de v{MAJOR} . {MINOR} . {PATCH}, por ejemplo v3.2.1:

- MAJOR: correspondería cuando se introducen cambios incompatibles con el desarrollo previo.
- MINOR: cuando se agregan nuevas funcionalidades, manteniendo compatibilidad con la funcionalidad previa.
- PATCH: mayormente cuando se despliegan cambios menores compatibles con el resto del código existente.

Otros puntos interesantes a recalcar en esta integración serían la verificación de los estilos en código y las pruebas unitarias.

Para la verificación de los estilos de código se utilizó **ESlint**<sup>7</sup> y **Prettier**<sup>8</sup> para el código de JavaScript y TypeScript, mientras tanto para el código escrito en PHP se utilizó **Laravel Pint**<sup>9</sup> con PHP Code Sniffer.

Las pruebas automatizadas del código se evaluarán en la sección 5.4.

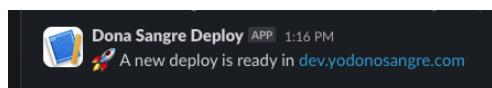


Figura 5.2: Integración con Slack. Fuente: Elaborado por el autor.

A nivel del despliegue, o puesta en staging, se ha utilizado Laravel Envoy<sup>10</sup>, una herramienta nativa de Laravel que permite el despliegue de las aplicaciones a través de bloques definidos. Se ha incorporado, adicionalmente, una tarea conectada a Slack para poder anunciar cuando un despliegue ha sido realizado, como se muestra en la figura 5.2.

<sup>6</sup><https://semver.org/>. Consultado el 09/11/2023.

<sup>7</sup><https://eslint.org/>. Consultado el 09/11/2023.

<sup>8</sup><https://prettier.io/>. Consultado el 09/11/2023.

<sup>9</sup><https://laravel.com/docs/10.x/pint>. Consultado el 09/11/2023.

<sup>10</sup><https://laravel.com/docs/10.x/envoy>. Consultado el 11/11/2023.

## 5.3. Documentación del API

Al ser una aplicación multiplataforma (web y móvil) se vuelve indispensable la creación de un API para la comunicación entre los servicios y la aplicación móvil.

Para mantener una integración adecuada, se ha hecho oportuno que todos los *endpoints* que corresponden al API estén bien documentados y probados.

En consecuencia, se ha documentado el API utilizando la especificación OpenAPI (Swagger)<sup>11</sup>, una referencia actual para la descripción de APIs, lo que permite una documentación limpia y concisa, además de permitir la importación de dichas referencias a otras herramientas (Casas et al., 2021), como lo sería Postman<sup>12</sup>.

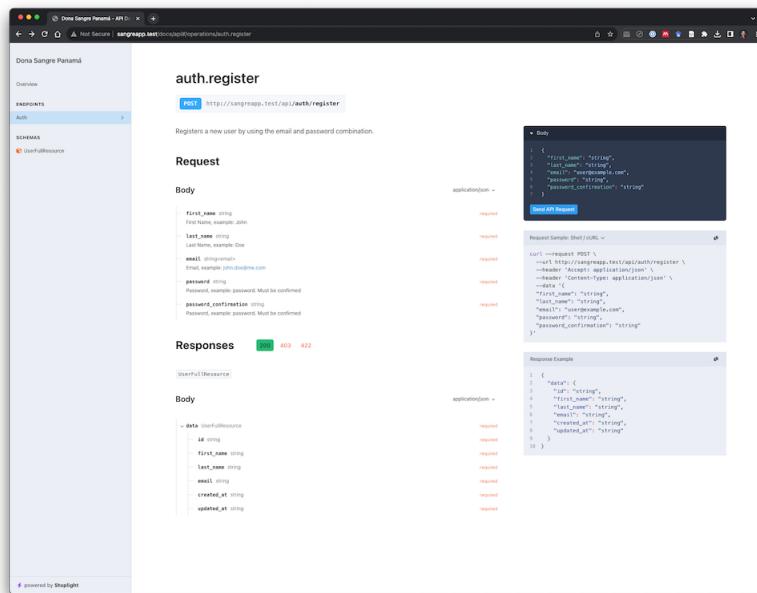


Figura 5.3: *OpenAPI specification*. Fuente: Elaborado por el autor.

En la figura 5.3 se puede ver la página de documentación del API, la cual solo está disponible en el ambiente de desarrollo (*dev*) a través de la dirección /docs dentro del explorador.

<sup>11</sup><https://swagger.io/specification/>. Consultado el 09/11/2023.

<sup>12</sup><https://www.postman.com/>, Consultado el 09/11/2023.

## 5.4. Pruebas automatizadas en *Backend / API*

```
~/projects/masw-tfm/donasangrepanama-be git:(feat/spa-22) ±16 (2.539s)
php artisan test
  PASS Tests\Feature\Auth\EmailVerificationTest
    ✓ email verification screen can be rendered
    ✓ email can be verified
    ✓ email is not verified with invalid hash
      0.02s
      0.02s
      0.02s

  PASS Tests\Feature\Auth>PasswordConfirmationTest
    ✓ confirm password screen can be rendered
    ✓ password can be confirmed
    ✓ password is not confirmed with invalid password
      0.02s
      0.06s
      0.22s

  PASS Tests\Feature\Auth>PasswordResetTest
    ✓ reset password link screen can be rendered
    ✓ reset password link can be requested
    ✓ reset password screen can be rendered
    ✓ password can be reset with valid token
      0.02s
      0.06s
      0.06s
      0.16s

  PASS Tests\Feature\Auth>PasswordUpdateTest
    ✓ password can be updated
    ✓ correct password must be provided to update password
      0.15s
      0.06s

  PASS Tests\Feature\Auth\RegistrationTest
    ✓ registration screen can be rendered
    ✓ new users can register
      0.02s
      0.09s

  PASS Tests\Feature\ExampleTest
    ✓ the application returns a successful response
      0.02s

  PASS Tests\Feature\Http\Controllers\Api\Auth\RegisterUserControllerTest
    ✓ can register a new account
    ✓ cannot register a new account with invalid data
    ✓ cannot register a new account with an existing email
    ✓ cannot register an account if user is logged in
      0.08s
      0.02s
      0.02s
      0.02s

  PASS Tests\Feature\Http\Controllers\Api\Auth\SessionControllerTest
    ✓ login user when valid credentials are provided and user is active
    ✓ does not login when credentials are wrong
    ✓ does not login when user is not active
      0.08s
      0.22s
      0.06s

  PASS Tests\Feature\ProfileTest
    ✓ profile page is displayed
    ✓ profile information can be updated
    ✓ email verification status is unchanged when the email address is unchanged
    ✓ user can delete their account
    ✓ correct password must be provided to delete account
      0.02s
      0.02s
      0.02s
      0.06s
      0.06s
```

Figura 5.4: *Tests* con PHPUnit. Fuente: elaborado por el autor.

Para las pruebas automatizadas, que es parte también de una de las etapas del flujo de CI/CD, se implementan pruebas de tipo funcional (prueba funcional) y unitario (prueba unitaria) a través de PHPUnit<sup>13</sup>, un framework de pruebas para PHP que cuenta con integración nativa en Laravel.

En la figura 5.4 se muestra el *runner* de los *tests* para el BE y los distintos *endpoints* de la API. Gracias a estas pruebas automatizadas, es posible verificar el correcto funcionamiento del código fuente además de comprobar que los criterios de aceptación de las historias se cumplen en cada una de las partes del código asociado con sus respectivas historias.

Un ejemplo sería la historia de la tabla 5.1.

<sup>13</sup><https://phpunit.de/>. Consultado el 10/11/2023.

Cuadro 5.1: SPA-22 - Historia

Título	Descripción
(SPA-22) [Web] Implement register flow (over the app and API)	<p><b>Historia de Usuario</b></p> <p>Como un usuario registrado, quiero poder iniciar sesión con mi correo y contraseña para así poder tener acceso a las características de la aplicación.</p> <p><b>Criterios de Aceptación</b></p> <ul style="list-style-type: none"><li>■ El usuario no debe tener una sesión activa actualmente.</li><li>■ Las credenciales (correo y contraseña) deben coincidir con las credenciales del usuario.</li><li>■ El usuario debe haber validado su correo electrónico anteriormente.</li><li>■ El usuario debe pertenecer al rol <code>user</code>, por ende, los usuarios con roles <code>superadmin</code> o <code>admin</code> no podrán iniciar sesión a través del API, puesto que son utilizados solo a través del panel de administración.</li></ul>

Partiendo de dichos criterios de aceptación, se han escrito las siguientes pruebas de tipo prueba funcional:

- `logins_user_when_valid_credentials_are_provided_and_user_is_active`
- `does_not_login_when_credentials_are_wrong`
- `does_not_login_when_user_is_not_active`
- `does_not_login_when_user_is_admin_or_superadmin`

Como se puede apreciar, dichas pruebas soportan lo esperado por nuestro sistema al momento de su implementación, por lo que corresponden a una forma automatizada de validar que nuestro sistema funciona tal como ha sido pensado y diseñado.

## Pruebas E2E

Además de las pruebas de tipo prueba unitaria y prueba funcional se han incluido pruebas E2E<sup>14</sup>, que son pruebas para demostrar la funcionalidad de extremo a extremo.

Para esto se utiliza Laravel Dusk<sup>15</sup> que es una librería de Laravel que implementa ya el ChromeDriver<sup>16</sup> y Selenium<sup>17</sup> para ejecutar funcionalidad automatizada desde un explorador real y a través de una aplicación funcional.

A través de estas pruebas se puede no solo probar las rutas, sino también la **aplicación corriendo en un entorno controlado y desde un explorador**, asemejando al uso que le daría un usuario real.

## 5.5. Panel de Administración

Uno de los subsistemas del proyecto es el panel administrativo. Dentro del sistema se han definido tres tipos de roles: `superadmin`, `admin` y `user`:

- `superadmin`: Administra todos los recursos, incluyendo los usuarios.
- `admin`: Administra todos los recursos, pero no administra otros usuarios. Solo puede consultar los recursos de una organización asociada a su usuario.
- `user`: Es un usuario funcional de la plataforma y no tiene acceso para administrar otros recursos más que los propios y fuera del panel de administración.

Para acceder al Panel de Administración es requerido contar con un rol `superadmin` o `admin`.

Para facilitar el desarrollo del Panel de Administración se ha instalado el paquete de *Backpack for Laravel*<sup>18</sup>, el cual proporciona mecanismos simples para el desarrollo de un Panel de Administración a través de los modelos definidos en Laravel.

En la figura 5.5 se puede apreciar el Panel de Administración en funcionamiento para el listado de las citas de donación registradas.

<sup>14</sup> *End to End*, o pruebas de extremo a extremo

<sup>15</sup> <https://laravel.com/docs/10.x/dusk>. Consultado el 15/11/2023.

<sup>16</sup> <https://chromedriver.chromium.org/home>. Consultado el 15/11/2023.

<sup>17</sup> <https://www.selenium.dev/>. Consultado el 15/11/2023.

<sup>18</sup> <https://backpackforlaravel.com/>. Consultado el 11/11/2023.

Citas De Donación							
+ Agregar cita de donación							
RANGO DE FECHAS	ESTADO	BANCO DE SANGRE	USUARIO	ESTADO	REVISADO POR	FECMA	ACTUALIZADO EN
9b139329-12b...	Aprobado	Demostenes Apple	Ciudad de la Salud - Banco de Sa...	Super Admin	en 4 días	12 ene. 2024, 13:22	➡ Vista previa ⚡ Editar ✓ Realizada
9b106875-c58...	En revisión	Carlos Aguilar	Complejo Hospitalario Dr. Arnulf...	-	hace 2 días	10 ene. 2024, 23:35	➡ Vista previa ⚡ Editar ✓ Aprobar ⚡ Rechazar
9b103d4e-6e0...	Aprobado	Demostenes Apple	Ciudad de la Salud - Banco de Sa...	Super Admin	en 1 día	10 ene. 2024, 21:39	➡ Vista previa ⚡ Editar ✓ Realizada
9b103c61-eed...	Aprobado	Demostenes Apple	Ciudad de la Salud - Banco de Sa...	Super Admin	en 3 días	10 ene. 2024, 21:39	➡ Vista previa ⚡ Editar ✓ Realizada
9b10392f-ba3...	Realizado	Demostenes Apple	Hospital Aquilino Tejera	Super Admin	en 1 semana	10 ene. 2024, 21:23	➡ Vista previa
9b0d9afo-bd3...	En revisión	Demostenes Developer	Centro de Donación de Sangre CSS...	-	en 1 día	9 ene. 2024, 13:24	➡ Vista previa ⚡ Editar ✓ Aprobar ⚡ Rechazar
9b0c2981-9a5...	Realizado	Demostenes Apple	Complejo Hospitalario Dr. Arnulf...	Super Admin	en 1 semana	8 ene. 2024, 20:57	➡ Vista previa
9b0c2f1f-baa...	Realizado	Demostenes Apple	Centro de Donación de Sangre CSS...	Super Admin	en 1 día	8 ene. 2024, 20:58	➡ Vista previa
9b0c20a8-f6e...	Realizado	Demostenes Apple	Fundación Dona Vida	Super Admin	en 1 semana	8 ene. 2024, 21:02	➡ Vista previa
9b0c206d-71d...	Realizado	Demostenes Apple	Fundación Dona Vida	Super Admin	en 2 días	8 ene. 2024, 21:03	➡ Vista previa
9b0c202a-561...	Aprobado	Demostenes Apple	Fundación Dona Vida	Super Admin	en 1 día	8 ene. 2024, 20:30	➡ Vista previa ⚡ Editar ✓ Realizada
9b0c1fb8-70d...	Aprobado	Demostenes Apple	Complejo Hospitalario Dr. Arnulf...	Super Admin	en 2 días	8 ene. 2024, 20:29	➡ Vista previa ⚡ Editar ✓ Realizada
9b0c1f21-e91...	Aprobado	Demostenes Apple	Fundación Dona Vida	Super Admin	en 4 días	8 ene. 2024, 20:27	➡ Vista previa ⚡ Editar ✓ Realizada
9b0c1dd9-f64...	Aprobado	Demostenes Apple	Hospital Pacifica Salud	Super Admin	en 1 día	8 ene. 2024, 20:23	➡ Vista previa ⚡ Editar ✓ Realizada
9b0c1d5f-540...	Aprobado	Demostenes Apple	Centro de Donación de Sangre CSS...	Super Admin	en 2 días	8 ene. 2024, 20:22	➡ Vista previa ⚡ Editar ✓ Realizada
9b0c1cbc-92c...	Aprobado	Demostenes Apple	Fundación Dona Vida	Super Admin	en 1 semana	8 ene. 2024, 20:18	➡ Vista previa ⚡ Editar ✓ Realizada
9b0c1bb1-b26...	Aprobado	Demostenes Apple	Centro de Donación de Sangre CSS...	Super Admin	en 4 días	8 ene. 2024, 20:17	➡ Vista previa ⚡ Editar ✓ Realizada
9b0c1b27-c68...	Aprobado	Demostenes Apple	Hospital Santo Tomás	Super Admin	en 4 días	8 ene. 2024, 20:16	➡ Vista previa ⚡ Editar ✓ Realizada
9b0cta55-Qeb...	Aprobado	Demostenes Apple	Hospital Pacifica Salud	Super Admin	hace 2 días	8 ene. 2024, 20:14	➡ Vista previa ⚡ Editar ✓ Realizada

Figura 5.5: Panel de Admin - Citas. Fuente: Elaborado por el autor.

## 5.6. Notificaciones por correo

El sistema requiere el envío de ciertas notificaciones por correo electrónico. Un ejemplo sería la verificación del correo electrónico suministrado al momento de registrar una cuenta a través de correo electrónico y contraseña.

A nivel de desarrollo, y para no requerir una dependencia de un servidor SMTP<sup>19</sup>, se utilizó Mailpit<sup>20</sup> que actúa y simula un servidor SMTP de forma local.

Para el servidor de staging sí es requerido un servidor real de SMTP, por lo que se podría utilizar un servicio disponible o configurar el propio. Para esta aplicación se ha escogido utilizar Resend<sup>21</sup>, un SaaS que expone funcionalidades SMTP a través de un API y que cuenta con un nivel (*tier*) gratuito<sup>22</sup>.

<sup>19</sup> Simple Mail Transfer Protocol

<sup>20</sup> <https://github.com/axllent/mailpit>. Consultado el 12/11/2023.

<sup>21</sup> <https://resend.com/>. Consultado el 12/11/2023.

<sup>22</sup> Hasta 3,000 correos mensuales o 100 correos diarios

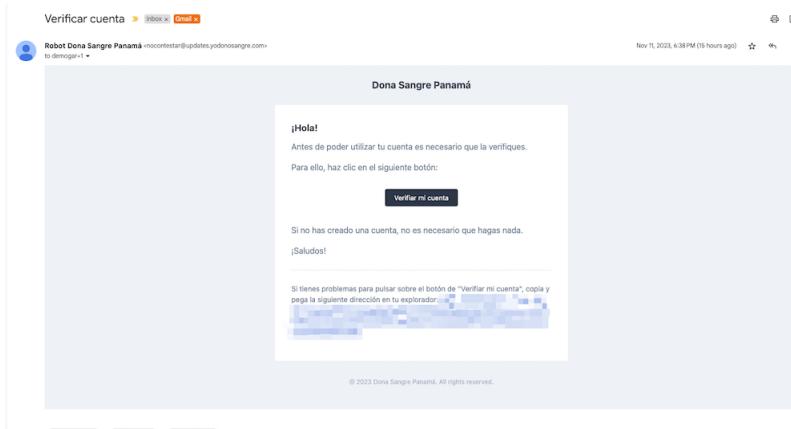


Figura 5.6: Notificación por correo. Fuente: Elaborado por el autor.

Para configurarlo correctamente y evitar problemas al entregar los correos, es necesario realizar algunas configuraciones en el DNS<sup>23</sup> del dominio y aplicar tres registros en el subdominio que se utilizará para este tipo de correos:

- MX: Registro de intercambio de correo para especificar los servidores de correo electrónico responsables de los correos en un dominio.
- TXT de tipo SPF<sup>24</sup>: Registro para evitar la falsificación de correos y otorga un grado de protección y validación a nuestro dominio.
- TXT de tipo SPF: Registro para guardar la llave pública del servicio de correo.

Con esta configuración, se pueden recibir los correos verificados en nuestra bandeja de entrada, como se muestra en la figura 5.6.

## 5.7. Inicio de sesión con cuentas sociales

Laravel provee el paquete llamado Laravel Socialite<sup>25</sup> que permite la configuración inicial para el inicio de sesión y autenticación a través de cuentas de otros proveedores que implementen el estándar OAuth<sup>26</sup>, como lo serían Google o Facebook<sup>27</sup>.

<sup>23</sup>Domain Name System.

<sup>24</sup>Sender Policy Framework.

<sup>25</sup><https://laravel.com/docs/10.x/socialite>. Consultado el 14/11/2023.

<sup>26</sup>Open Authorization.

<sup>27</sup>No se implementará Facebook como parte de este Trabajo de Fin de Máster.

Con su soporte, se ha podido configurar el inicio de sesión con una cuenta de Google, por lo que se tendría el mecanismo de registro, a través de una cuenta de Google o a través de usuario y contraseña. En la figura 5.7 se muestra el proceso.

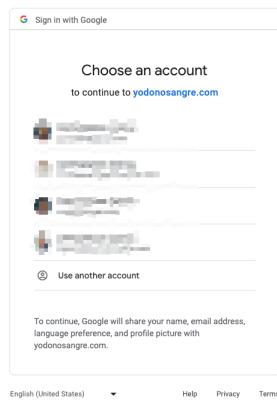


Figura 5.7: OAuth con Google, Parte 2. Fuente: Elaborado por el autor.

Para la aplicación web, en la página de inicio de sesión se muestra un botón de Iniciar Sesión con Google, lo que procede a generar el proceso de autorización.

## Inicio de sesión en móvil: Google (iOS y Android)

El inicio de sesión a través de la aplicación móvil, en conjunto con la capa de servicios, sigue un formato similar, aunque no idéntico al de la aplicación web. Esto se debe a que **la aplicación web es un monolito que se comunica directamente con el frontend a través de Inertia**.

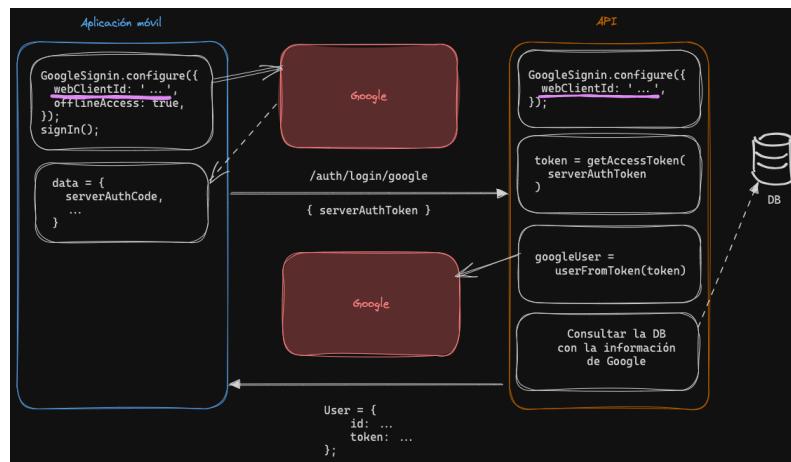


Figura 5.8: OAuth con Google en móvil. Fuente: Elaborado por el autor.

Para la aplicación móvil en React Native / Expo se utilizará la librería RN Google Sign In<sup>28</sup> y se continúa la configuración utilizando un proyecto de Firebase para las aplicaciones, tanto en iOS como en Android.

Como parte de esta configuración, es necesario la creación de los proyectos de forma individual en la consola de Firebase. Esto generará los archivos google-services.json (Android) y GoogleService-Info.plist (iOS) a los cuales deben relacionar en el proyecto de Expo (ver Apéndice L).

El flujo de inicio de sesión en el móvil y su interoperabilidad con el BE, visto en la figura 5.8, está descrito de la siguiente forma:

- Tanto a nivel del API (*Backend*) como a nivel de la aplicación móvil es importante que se comparta el mismo Client ID de Google (generado a través de Firebase). Sin esto, no se puede validar el token generado.
- El proceso inicia a través de la aplicación móvil. Se genera, a través de la librería RN Google Sign In, el inicio de sesión y es importante que se declare el uso de la llave offline=true y que el webClientId sea el mismo que se comparte a nivel del BE.
- Esto generará un serverAuthCode, además de la información del usuario.
- El serverAuthCode se envía al API a una ruta definida, en este caso particular, /api/login/google.

<sup>28</sup><https://react-native-google-signin.github.io/docs/original>. Consultado el 20/11/2023.

- En el API/Backend, se utiliza este mismo serverAuthCode para con Laravel Socialite conseguir el authToken y con este conseguir la información del usuario en el Backend.
- Una vez obtenida esta información se puede crear o actualizar el usuario localmente y generar un nuevo token para nuestro API.

Este token local es el que permite firmar las peticiones y consultas en el API y es requerido en los endpoints, puesto que los mismos están firmados utilizando dicho token.

## Inicio de sesión en móvil: Apple (iOS)

Si bien es cierto que el alcance de este Trabajo de Fin de Máster es elaborar una Prueba de Concepto (PoC), se debe también considerar que es importante cumplir con los estándares y requisitos establecidos por las mejores prácticas y guías de las tiendas de aplicaciones.

En el caso de dispositivos iOS y a partir con la llegada de iOS13, la guía de Apple señala que se hace obligatorio la implementación de inicio de sesión con una cuenta de Apple para cumplir con las regulaciones de la tienda<sup>29</sup>.

Para la implementación se ha utilizado el paquete de Expo de Apple Authentication<sup>30</sup> y se han generado los certificados y configuraciones en el proyecto, tal cual como se detalla en su guía.

El flujo satisfactorio de inicio de sesión con una cuenta de Apple es similar al mostrado en el diagrama de la figura 5.9:

---

<sup>29</sup><https://developer.apple.com/sign-in-with-apple/>. Consultado el 02/01/2024.

<sup>30</sup><https://docs.expo.dev/sdk/apple-authentication/>. Consultado el 02/01/2024.

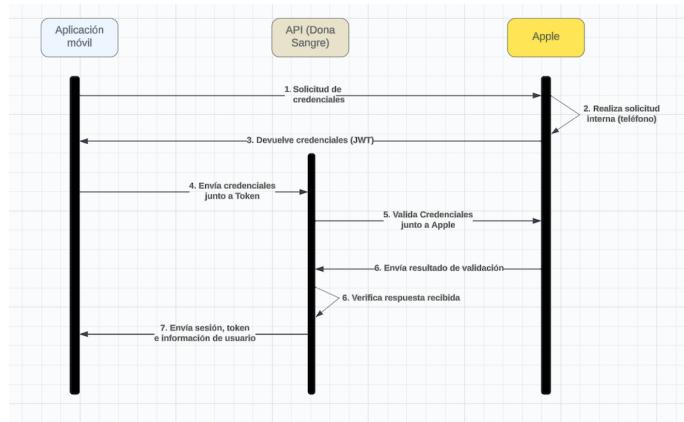


Figura 5.9: Inicio con Apple. Fuente: Elaborado por el autor.

El inicio de sesión no es controlado por la aplicación. El sistema operativo (iOS, en este caso) expone funcionalidades para realizar el inicio de sesión y ejecutar el mismo contra los servidores de Apple.

Cuando el proceso finaliza, Apple envía de vuelta a la aplicación un payload con los siguientes datos:

- `authorizationCode`: Es un token de corta duración que se utiliza para comunicarse con el servidor de Apple.
- `identityToken`: Es un token en formato JWT que posee información básica sobre el usuario y la sesión.
- `fullName`: Es un objeto en formato JSON con la información de la cuenta. Es importante recalcar que esta información puede venir vacía, puesto que Apple respeta el anonimato.
- `user`: Es un identificador único para la cuenta de Apple y funciona para poder identificar al usuario frente a Apple.
- `email`: Puede ser el correo real del usuario, un correo de tipo proxy o puede estar vacío (`null`).

Toda esta información se recopila y envía al API y se utiliza la información recibida a través del `authorizationCode`, el `identityToken` y el `user` para validar nuevamente el token frente a los servidores de Apple.

Si alguna información está vacía<sup>31</sup>, se crean datos intermedios. En el caso del correo se crea uno aleatorio, similar a `user.[idapple]@...`, y el usuario puede actualizar este correo desde la aplicación, junto con los otros datos intermedios.

Alguno de los mecanismos de seguridad y protección contra ataques y suplantación de identidad que se utilizan son los siguientes:

- Se utilizan las llaves públicas de Apple<sup>32</sup> para verificar de manera criptográfica la información del token.
- Una vez verificado, se verifica que el `iss` (el *issuer*) del certificado haya sido Apple.
- También se verifica que la audiencia (`aud`) es el paquete de nuestra aplicación.
- Se verifica que el tiempo de expiración del token es menor a la hora actual.

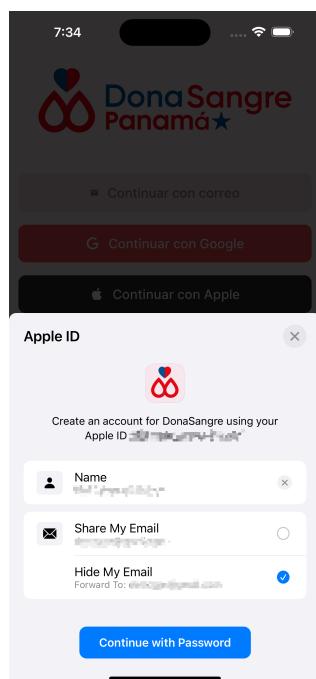


Figura 5.10: Inicio con Apple desde iOS. Fuente: Elaborado por el autor.

Si todos estos datos son correctos, automáticamente se inicia sesión, se crea un nuevo token propio en formato JWT y se envía de vuelta a la aplicación junto con la sesión e información del usuario.

<sup>31</sup>Apple implementa mecanismos de protección de privacidad, por lo que algunos datos pueden regresar vacíos.

<sup>32</sup><https://appleid.apple.com/auth/keys>. Consultado el 02/01/2024.

## 5.8. Queues (Colas)

Las colas de mensaje, o *Message Queues*, son un mecanismo para que distintos componentes de un software puedan comunicarse entre ellos, a través de un canal de comunicación, de forma asíncrona (Maharjan et al., 2023).

Para ello es necesario utilizar un sistema de colas (Queue System). En este caso se utiliza Redis<sup>33</sup> ya que ha mostrado baja latencia y buen rendimiento, según el precitado artículo.

Laravel ya posee una forma nativa para el manejo de colas<sup>34</sup>, por lo que se ha implementado el mismo junto con Laravel Horizon<sup>35</sup>, que es una plataforma integrada para la visualización de las colas en tiempo real.

Para la ejecución del *worker* en el servidor de *staging* es necesario un supervisor de las tareas en ejecución, como supervisor<sup>36</sup>, el cual se debe configurar para el comando del *worker*, como para *horizon* (ver configuración en Apéndice F).

Esta configuración permite que los dos procesos se ejecuten en el background del Sistema Operativo de forma asíncrona y paralela.

Un ejemplo sería la tarea para agregar nuevas citas, similar a la figura 5.11:

- El controlador recibe la petición a través de la ruta. Una vez validada, crea una nueva tarea dentro de la cola.
- Además, el controlador devolverá la información de la nueva cita pre-procesada.
- El nuevo elemento de cola se procesará en el background y verificará a mayor detalle la cita.
- Una vez procesada, se emitirán dos nuevos jobs:
  - El primer job se encargará de notificarle al usuario.
  - El segundo hará otros procesamientos para los administradores y cambiará el estado de la cita.

<sup>33</sup><https://redis.io/>. Consultado el 24/11/2023.

<sup>34</sup><https://laravel.com/docs/10.x/queues>. Consultado el 24/11/2023.

<sup>35</sup><https://laravel.com/docs/10.x/horizon>. Consultado el 24/11/2023.

<sup>36</sup><http://supervisord.org/>. Consultado el 24/11/2023.

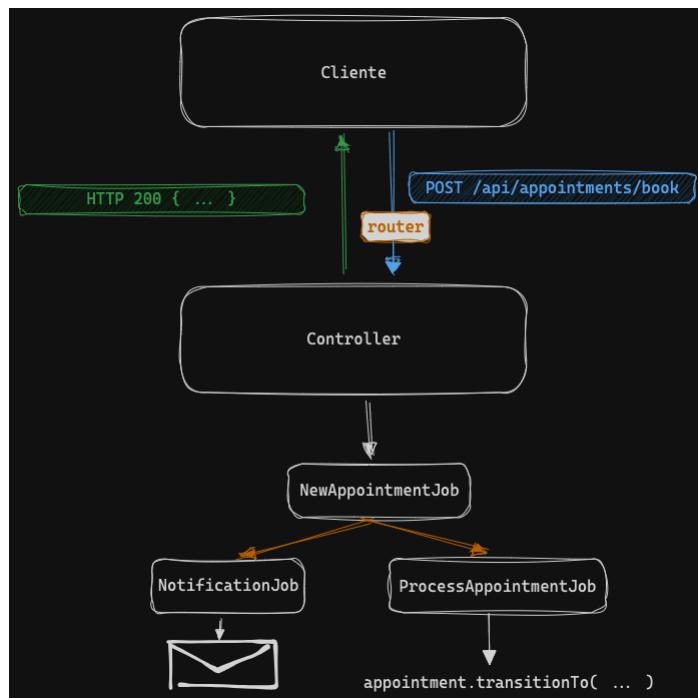


Figura 5.11: Queue de cita. Fuente: Elaborado por el autor.

Al incorporar queues y procesos en el fondo (background), **el sistema tiene la capacidad de procesar tareas de manera asíncrona, sin interferir con el flujo principal de la aplicación, incluso cuando estas tareas requieren un tiempo significativo**. Otra gran ventaja del uso de colas para estos procesos es que, si alguno falla, se reintenta en varias ocasiones.

Estas *queues* (colas) también administran todas las notificaciones (vistas en el apartado 5.6) que se emiten a través de la plataforma, por ejemplo, las notificaciones cuando una nueva cita es solicitada, una cita es aprobada y una nueva donación es recibida.

## Máquinas de Estado en *Backend*

Para facilitar el flujo en algunos casos y evitar que algunos elementos entren en un estado en el cuál no están listos para entrar (por ejemplo, para evitar que una cita sea aprobada sin antes ser verificada manualmente por un miembro del centro de donación) se ha implementado una máquina de estado finita a nivel del *Backend*.

Las colas (*queues*) se soportan fuertemente en estas máquinas de estado y los cambios de estado son verificados y ejecutados a través de la Máquina de estado

finita, lo que permite la transición controlada entre un estado a otro. Estas máquinas de estados se definen como máquinas de estado finitas, por lo que se componen de algunos elementos claves:

- **Estados:** Los distintos estados o condiciones en el cual un objeto puede encontrarse.
- **Transiciones:** Los distintos cambios de estado que pueden realizarse (de estado A a estado B, por ejemplo).
- **Eventos:** Señales que son emitidas al cambiar el estado (transición).
- **Acciones:** Operaciones que son ejecutadas cuando una transición es realizada.
- **Condiciones:** Criterios mediante el cual se determina cuando una transición puede ser ejecutada.

Para asegurar un flujo lineal entre los estados, las colas utilizan las máquinas de estado. La transición entre cada estado está asociada a distintas acciones que pueden incluir una notificación.

En el Apéndice K se muestran el listado de los distintos estados y las transiciones habilitadas para el modelo de cita (Appointment). También se muestra la conexión con el modelo y cómo estas mismas interactúan con otras partes del código.

## 5.9. Mapas: OpenStreetMap con Leaflet, Google Maps (Android) y Apple Maps (iOS)

OpenStreetMap es un proyecto comunitario, abierto y colaborativo que busca la creación de mapas más accesibles y editables por cualquier persona (OpenStreetMap, 2004). Al ser colaborativa y comunitaria, los datos son mantenidos por la comunidad. Dichos datos pueden ser consumidos y utilizados de forma gratuita.

Google Maps Platform, por su parte, es un servicio de pago de Google que permite la interacción con los mapas de Google. Cuentan con un plan gratuito<sup>37</sup> que permite la carga de hasta 28,500 mapas de forma mensual.

Si bien OpenStreetMap es robusto, su interacción y consumo directo no es simple, por lo que existen librerías como Leaflet<sup>38</sup> que permiten el consumo e interacción a través de JavaScript.

<sup>37</sup>Plan gratuito disponible al 05/12/2023.

<sup>38</sup><https://leafletjs.com/>. Consultado el 05/12/2023.

Para nuestra integración se ha escogido utilizar ambas plataformas de la siguiente forma:

- **Google Maps Platform:** Será utilizado únicamente en el Panel de Administración, mayormente debido a su Geocoding API<sup>39</sup>, lo que permite obtener fácilmente la latitud y longitud en base a una dirección escrita, algo que es muy difícil en Panamá.
- **OpenStreetMap + Leaflet:** Será utilizado para la aplicación web frontal, de cara al usuario. Esto debido a que ya se cuenta con la latitud y longitud precisa, además de una dirección con formato, por lo que solo se hace necesario mostrar el mapa con sus marcadores independientes.

A nivel de las aplicaciones móviles se utilizará React Native Maps<sup>40</sup>, que expone un API para utilizar los mapas nativos por plataforma (Apple Maps para iOS, Google Maps para Android).

A nivel de la Base de Datos, dichos campos se guardarán en una columna llamada location de tipo json que tendrá las siguientes llaves:

- lat: Latitud del punto.
- lng: Longitud del punto.
- formatted\_address: Dirección escrita y de fácil interpretación, según lo recibido a través del Geocoding API.

## Leaflet con marcadores personalizados

En el apartado web y la aplicación de usuario web, se han configurado marcadores personalizados para mejorar la experiencia del usuario final al determinar los distintos tipos de centros (privado o público).

Para poder diferenciar en el mapa los centros públicos/estatales de los privados se han agregado marcadores distintos con la ayuda del API en JavaScript de Leaflet (ver Apéndice G).

Esto permite mostrar marcadores en tonalidad **mostaza** para los centros **privados** y en una tonalidad **rosa** para los centros **públicos o estatales**, como se muestra

<sup>39</sup> <https://developers.google.com/maps/documentation/geocoding/overview>. Consultado el 05/12/2023.

<sup>40</sup> <https://github.com/react-native-maps/react-native-maps>. Consultado el 05/12/2023.

en la figura 5.12. Ambos colores están asociados a la paleta de colores descrita y expuesta por el Design System de PlaquetasDS, definido en la sección 3.3.

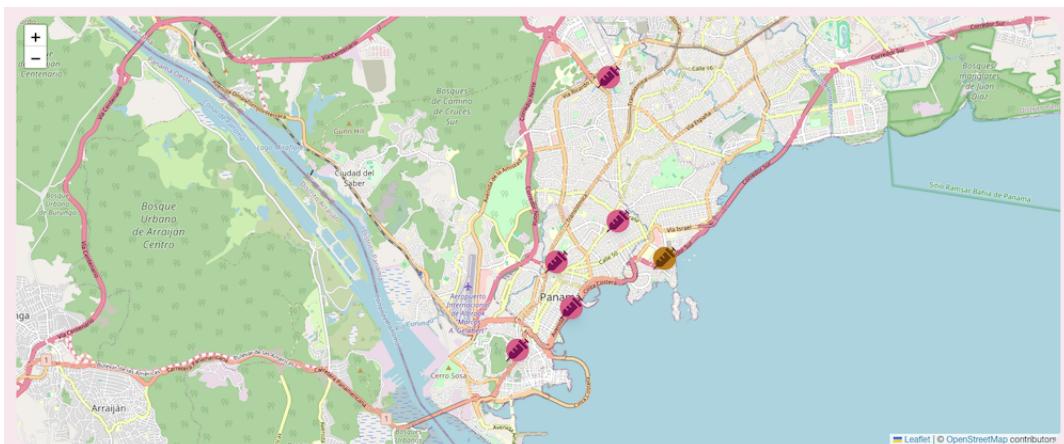


Figura 5.12: Marcador personalizado. Fuente: Elaborado por el autor.

## 5.10. Seguridad

Para la protección contra solicitudes maliciosas, además de proteger contra posibles ataques de denegación de servicio contra los servicios más básicos expuestos al usuario, se han implementado dos mecanismos principales y sencillos.

### **Throttle**

El primero es una configuración contra el **Throttle**, utilizando la configuración de *Rate Limitting* de Laravel<sup>41</sup>. La limitación de peticiones, o *Rate Limitting*, es una técnica sencilla mediante la cual se limita la frecuencia con la que algunas peticiones se realizan y es muy eficaz para detectar comportamientos inusuales (Jing et al., 2006).

Para ello toda la capa de APIs tienen una configuración de **no permitir más de 100 peticiones en un intervalo de 1 minuto**. Adicionalmente **los servicios relacionados con la autenticación no permiten más de 30 peticiones** en un intervalo de 1 minuto. Si algún usuario sobrepasa este límite, sus consultas serán automáticamente rechazadas.

<sup>41</sup><https://laravel.com/docs/10.x/rate-limiting>. Consultado el 27/12/2023.

## CAPTCHA

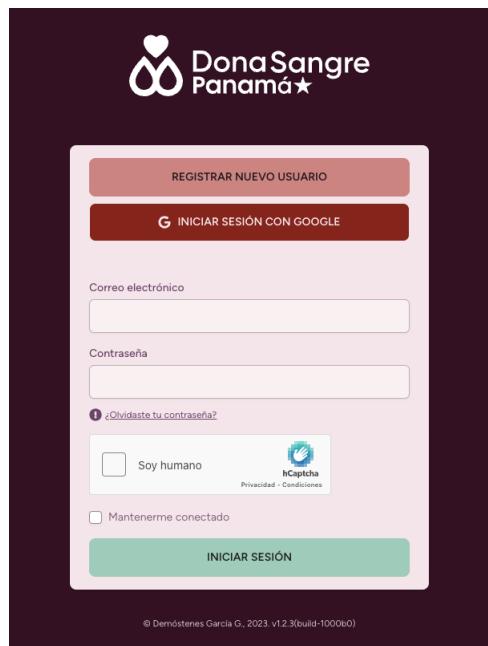


Figura 5.13: hCaptcha en formulario de login. Fuente: Elaborado por el autor.

Otra configuración, a nivel de la parte web, ha sido la implementación de un **CAPTCHA** en los formularios de inicio de sesión y registro de un usuario local (correo y contraseña), a través de la implementación de un servicio de hCaptcha<sup>42</sup>, que es un servicio gratuito<sup>43</sup> muy conocido por respetar ampliamente la privacidad de sus usuarios (Cloudflare, 2020).

## Validaciones

Para todas las consultas se han implementado validaciones en los *requests* utilizando Laravel y adicionalmente se han implementado algunos *middleware* para la verificación de roles, permisos y sesiones.

Igualmente se han implementado algunas reglas de validación personalizadas, entre las que se pueden mencionar las siguientes:

- **Validación de edad de donante:** Los donantes deben tener entre 18 y 65 años.

<sup>42</sup><https://www.hcaptcha.com/>. Consultado el 27/12/2023.

<sup>43</sup>Hasta 1 millón de consultas mensuales, a diciembre de 2023.

- **Validación de formato de cédula de identidad panameña (DNI):** El patrón de cédula debe ser correcto cuando el tipo de identificación es cédula (ver Apéndice J).
- **Validación de contraseña segura:** Al registrarse con correo y contraseña, los usuarios se les solicita una contraseña que tenga al menos 10 caracteres, al menos 1 carácter en mayúscula, al menos 1 carácter en minúscula, al menos 1 número y al menos 1 carácter especial.
- **Tiempo de espera entre donaciones<sup>44</sup>:** Si un donante solicita una nueva cita de donación sin esperar los **3 meses** que señala la legislación actual, el proceso no puede ser iniciado.

Adicionalmente, y en cumplimiento de los requerimientos no funcionales descritos en la Unidad 2.2 (Seguridad), se ha realizado una auditoría de seguridad a través de pruebas de penetración con su respectivo informe y soluciones, el cual ha sido documentado en el Apéndice N.

## 5.11. Notificaciones Push

Las notificaciones de tipo *Push* son mensajes de alerta o informativos que se emiten desde nuestros servicios hacia nuestros clientes, para informar sobre acciones o información que se ha generado en el *background* y no en la vista actual dentro de la aplicación.

En el caso de este proyecto, distintas acciones se realizan en formato asíncrono para el usuario, especialmente las relacionadas con las citas de donación y con las donaciones registradas. Es por ello por lo que se hace necesario implementar una capa de notificaciones y mecanismos para emitirlas y recibirlas.

Para las notificaciones de tipo *push* se han definido las mostradas en el cuadro 5.2, incluyendo las notificaciones por correo descritas en la sección 5.6:

Cuadro 5.2: **Notificaciones del sistema (push y mail)**

Notificación	Push	Correo	Reglas
Cita: Aprobada	✓	✓	Estado: aprobado

<sup>44</sup>Esta característica está detrás de un *Feature Flag*, que está completamente apagado en los entornos de desarrollo y pruebas.

Cuadro 5.2: Notificaciones del sistema (push y mail)

Notificación	Push	Correo	Reglas
Cita: Realizada	✓	✓	Estado: realizado
Cita: Cancelada (centro)	✗	✓	Estado: cancelado por centro
Cita: Cancelada (usuario)	✗	✓	Estado: cancelado por usuario
Donación: Registrada	✓	✓	Estado: finalizado
Eliminar cuenta	✗	✓	—
Donación recurrente	✓	✓	Última donación: mayor a 90 días
Nueva contraseña	✗	✓	—

Las notificaciones *push*, al igual que las notificaciones por correo, son manejadas a través de colas en el *background*, para evitar que las mismas bloqueen el hilo de ejecución principal de la aplicación.

## Configuración de *backend* para notificaciones *push*

Se ha generado primero una migración para crear una tabla de `devices`, relacionada con un usuario, donde se almacenarán dichos dispositivos y sus configuraciones para las notificaciones (ver Apéndice H).

Esto permite tener una relación de 1–n entre el Usuario (`users`) y los Dispositivos (`devices`) y se almacena información del dispositivo como el identificador único del mismo, junto al proveedor de notificaciones *push* y el token para dichas notificaciones.

Como ya se mencionó en el apartado 5.8, se ha implementado un sistema de colas para el procesamiento y envío de notificaciones por correo. Se ha podido utilizar el mismo sistema de colas y se ha agregado un canal adicional para manejar las mismas (ver la configuración en Apéndice K).

Para las pruebas, tanto de iOS como de Android, es necesario realizarlas en un dispositivo físico real. Este proceso se explicará en la sección 5.14.

## Notificaciones en Android

El registro de un dispositivo para recibir notificaciones se debe iniciar a partir del dispositivo móvil. Si bien se utiliza el paquete de Expo Notifications<sup>45</sup>, se necesita tener un proceso propio y adicionalmente:

- Se debe configurar el proyecto en Firebase. Para la configuración de Android, se debe habilitar Google Firebase Cloud Messaging (FCM) y generar un token de servidor de FCM.
- Este token de servidor de FCM se debe utilizar en la configuración del proyecto de Expo, adicionalmente de utilizar el archivo de configuración generado por Firebase.

La aplicación, intrínsecamente centrada en proporcionar a los usuarios solo la información y acciones solicitadas, requiere que las notificaciones se habiliten manualmente. Al iniciar este proceso, se activa una Máquina de estado finita que llevará a cabo los siguientes subprocessos:

- Obtener la información del dispositivo (sistema operativo, modelo, marca, identificador)
- Obtener permisos para recibir las notificaciones. Si el usuario rechaza el proceso, no se continúa.
- Obtener el token único de notificaciones de Expo.
- Recopilar y almacenar la información en el servidor, asociada al usuario.

Si todo este proceso se completa, un registro en Base de Datos (asociado al usuario) se generará y dicho token se vinculará, tanto al dispositivo como al usuario.

Esta información también se persistirá en el dispositivo y será eliminada (tanto a nivel del servidor como a nivel del dispositivo) si el usuario finaliza su sesión.

---

<sup>45</sup><https://docs.expo.dev/versions/latest/sdk/notifications/>. Consultado el 27/12/2023.

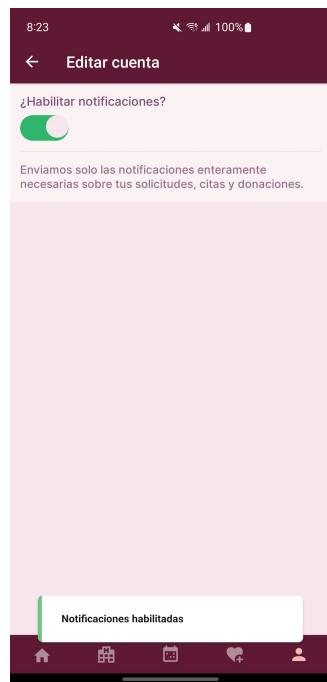


Figura 5.14: Registro de notificaciones. Fuente: Elaborado por el autor.

En la figura 5.14 se aprecia el fin del proceso de registro con las notificaciones. A este punto ya los puntos del proceso han sido completados.

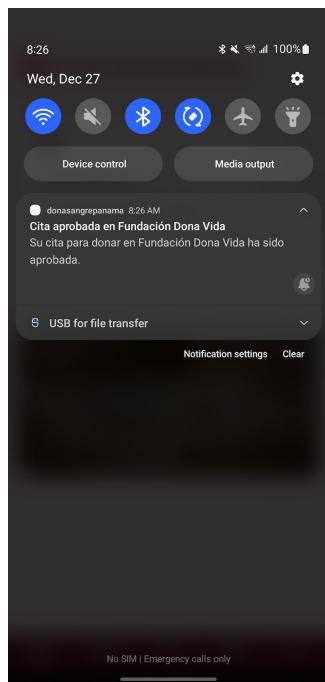


Figura 5.15: Recibo de notificación. Fuente: Elaborado por autor.

Cuando se ejecuta alguna acción que permite notificaciones dentro del proceso en paralelo, como por ejemplo, mediante una acción en el Panel de Administración, se enviará una notificación tipo Push al usuario. En la figura 5.15, se ilustra una notificación que se activa cuando una cita de donación ha sido aprobada.

## Notificaciones en iOS

La configuración en iOS se realiza a partir de Expo y a través del panel de desarrollo de la cuenta de Apple, por lo que es necesario contar con una afiliación (*membership*) de desarrollo anual de Apple como mínimo.

Luego de configurar la cuenta de desarrollo, se debe iniciar sesión a través del servicio de EAS de Expo<sup>46</sup> y con `eas cli`, el cual administra las credenciales.

Al requerir un despliegue en dispositivos reales para la prueba de esta característica, el proceso en cuestión es explicado en la unidad 5.14 (Pruebas en dispositivos reales).

<sup>46</sup><https://expo.dev/eas>. Consultado el 09/01/2024.

## 5.12. Optimizaciones de desempeño

Como se mencionó en la sección 5.2, la arquitectura del sistema incluye un servidor de caché con Redis para cumplir con el requisito de **Rendimiento (percibido)** detallado en la sección 2.2.1. Los datos que requieran alto procesamiento, tengan cambios poco frecuentes y utilicen muchos recursos se almacenan en Redis, utilizando la Caché de Laravel<sup>47</sup>. Dos implementaciones de caché son las siguientes:

- Estadísticas globales. Las estadísticas globales que se utilizan en el sistema de administración (ver 5.16).
- Estadísticas individuales por usuario.

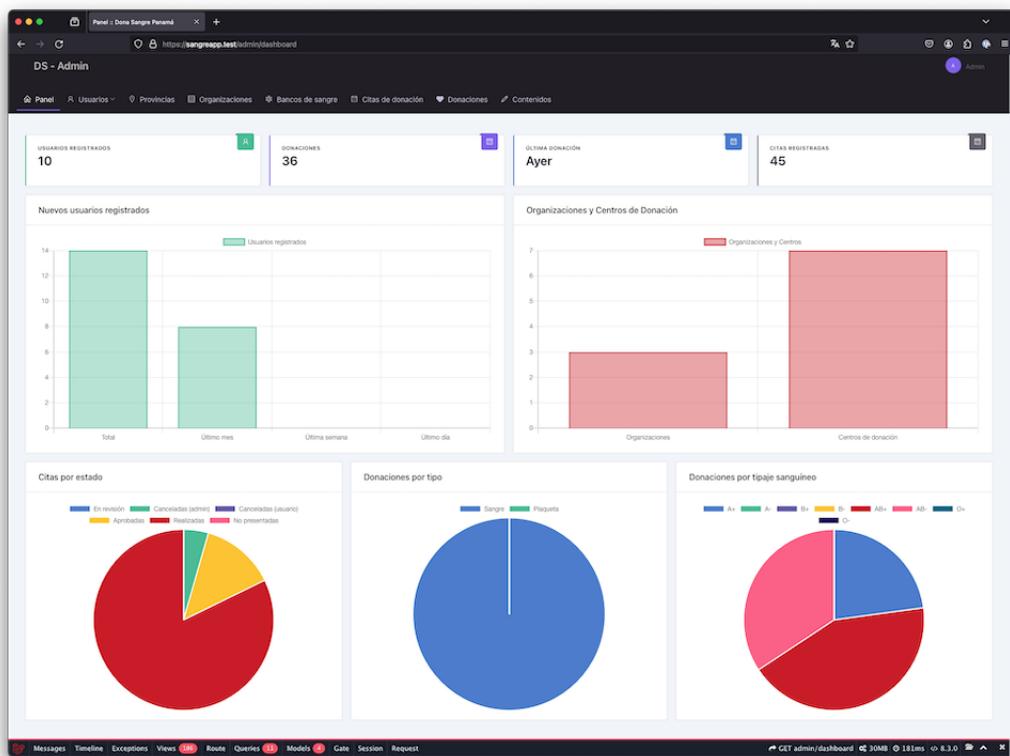


Figura 5.16: **Estadísticas por caché.** Fuente: Elaborado por el autor.

Por ejemplo, **estas estadísticas globales se almacenan en Redis y tienen un tiempo de vida de una hora**. Al paso de este tiempo y al momento de solicitarse nuevamente, se hace la solicitud y se vuelve a almacenar el resultado en caché.

<sup>47</sup><https://laravel.com/docs/10.x/cache>. Consultado el 27/12/2023.

Las estadísticas por usuario también se almacenan en Redis y se almacenan en forma indefinida hasta que el dato cambia.

Redis, al estar sus datos almacenados en memoria, proporcionan una latencia menor (Amazon, 2022), además de que reducirá la carga en la Base de Datos principal para datos que usualmente no varían.

Para implementar el principio de DRY, el código que maneja las estadísticas ha sido segregado en su propio *Service Container*<sup>48</sup>, y el proceso de gestión del tiempo de vida es el siguiente:

```
1 public function get(): array
2 {
3     return Cache::remember(
4         self::CACHE_KEY, // llave
5         self::CACHE_TTL_MINUTES * 60, fn () => $this->query() // 
6         TTL
7     );
}
```

## Caché en el móvil

Las aplicaciones móviles también han implementado un sistema de caché para evitar ejecutar las mismas consultas constantemente.

Esta implementación no utiliza Caché del BE, si no que se ha implementado React Query con QueryCache<sup>49</sup> y todas las consultas tienen un tiempo de vida a 5 minutos, a menos que alguna mutación<sup>50</sup> haya ocurrido, lo que invalida el caché.

## Otras mejoras de rendimiento

Además del uso de caché para mejorar el rendimiento de las aplicaciones, se han implementado algunos otros mecanismos para mejorar el rendimiento de la aplicación, al ser un requerimiento no funcional de calidad de la aplicación.

Si bien algunos aspectos pueden no ser controlados por nosotros, como degradación en la conectividad, el enfoque debe ser en el **rendimiento percibido por el usuario**.

<sup>48</sup><https://laravel.com/docs/10.x/container>. Consultado el 14/01/2024

<sup>49</sup><https://tanstack.com/query/latest/docs/react/reference/QueryCache>. Consultado el 27/12/2023.

<sup>50</sup><https://tanstack.com/query/v4/docs/react/guides/mutations>. Consultado el 27/12/2023.

## Optimizaciones en Web

Con la ayuda de Inertia<sup>51</sup>, algunos componentes (como las políticas de uso) son renderizados en el servidor y enviados listos para su consumo utilizando SSR.

Igualmente, las imágenes que se utilizan en el sitio web y la aplicación móvil son servidas a través de un CDN<sup>52</sup>, lo que permite la entrega de contenido más rápido y a través de un dominio secundario.

## Optimizaciones en Móvil

En algunas partes de la aplicación con listas que pueden ser muy grandes, se ha implementado FlashList<sup>53</sup> sobre FlatList<sup>54</sup>. FlashList recicla las vistas para mejorar el rendimiento<sup>55</sup> <sup>56</sup>.

Se han implementado Skeleton loaders para reducir el Cummulative Layout Shift. Con esto, el espacio que ocuparán se reserva mientras están en carga.

Se utilizan componentes “memoizados” y el uso de hooks como `useCallback`, `useMemo` y `React.memo` para no renderizar componentes que no han cambiado entre ciclos de renderizado.

Se implementa el motor de Hermes<sup>57</sup>, sobre el motor regular de JavaScriptCore, mediante el cual se reduce el tiempo de inicio y consumo de memoria de la aplicación.

En pantallas con componentes muy pesados, como en la pantalla de solicitud de donantes que se compone de un formulario con múltiples campos, se ha utilizado la técnica de (implementación detallada en el Apéndice I).

Como última mejora, se ha implementado una verificación de conectividad. Si bien es imposible controlar la conectividad (la calidad) de la aplicación a la red, al ser esta una aplicación que requiere conectividad para su funcionamiento se ha implementado un mecanismo de verificación de red.

<sup>51</sup> <https://inertiajs.com/server-side-rendering>. Consultado el 29/12/2023.

<sup>52</sup> En nuestro caso se utiliza <https://sirv.com/>. Consultado el 12/01/2024

<sup>53</sup> <https://shopify.github.io/flash-list/> Consultado el 29/12/2023.

<sup>54</sup> <https://reactnative.dev/docs/flatlist>. Consultado el 29/12/2023.

<sup>55</sup> <https://shopify.github.io/flash-list/docs/metrics>. Consultado el 29/12/2023.

<sup>56</sup> <https://shopify.github.io/flash-list/docs/recycling>. Consultado el 29/12/2023.

<sup>57</sup> <https://hermesengine.dev/>. Consultado el 06/01/2024.



Figura 5.17: Verificador de conexión. Fuente: Elaborado por el autor.

Para ello, el sistema, en segundo plano (*background*), verifica si ha habido algún cambio en los mecanismos de conectividad. Si se detectan variaciones, se activa una pantalla que bloquea el acceso al resto de la aplicación, tal como se representa en la figura 5.17.

## 5.13. Eliminación de cuenta y protección de privacidad

Uno de los puntos claves a nivel de los requerimientos no funcionales es la **protección de la privacidad** de los usuarios de la plataforma, orientado a la **Ley 81 de 2019 Sobre la protección de Datos Personales en la República de Panamá** (Gaceta Oficial, República de Panamá, 2019).

Uno de los derechos principales al que se ha hecho énfasis es el **principio de cancelación**. Para ello, la plataforma permite la eliminación de la cuenta en todo momento y sin mayor requerimiento que el de llenar un formulario.

Al momento de realizar esta acción, el usuario es desvinculado completamente con todos sus datos y los mismos son ofuscados, entendiéndose que para proteger

la integridad del sistema y por principios de seguimiento y trazabilidad no podrían ser eliminados completamente.

**La trazabilidad obedece a que los datos de donaciones y citas deben existir en todo momento, ya que pueden ser requeridos para la confirmación ante una Donación por reposición**

Al ejecutar la acción de cancelación de una cuenta, un proceso en background es iniciado el cual:

- Cambia todos los datos personales a asteriscos (\*\*\*)�.
- Cancela las citas que no hayan sido aceptadas hasta ese momento.
- Hace un Soft delete del usuario, además de eliminar su contraseña completamente.

Una vez esta acción es ejecutada, es imposible recuperar la cuenta a su estado previo. Nadie, ni los administradores de la plataforma, tendrían acceso a los datos anteriores.

## Confirmación en dos pasos (2FA)

Para proteger el uso no autorizado de esta funcionalidad **también se ha implementado un sistema de confirmación en dos pasos** (*Two-Factor Authentication* o 2FA), mediante el cual el usuario debe confirmar la acción mediante un código aleatorio de seis dígitos enviado al correo electrónico.

Esta funcionalidad se puede ver en las figuras 5.18 y 5.19.



Figura 5.18: Eliminar cuenta. Fuente: Elaborado por el autor.

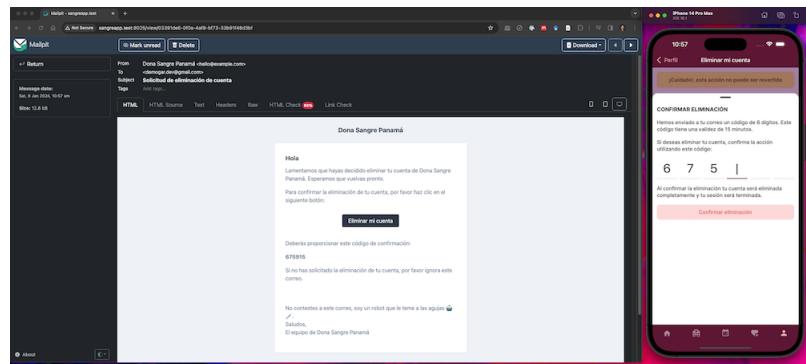


Figura 5.19: Eliminar cuenta (Confirmar) - Fuente: Elaborado por el autor.

Este flujo completo está accesible tanto en plataformas móviles, como en la web. El código numérico suministrado (de 6 dígitos) es válido durante 15 minutos.

Si se solicita un nuevo código, el anterior perderá su validez. Por lo tanto, el código tiene un único uso y está vinculado a la sesión y al usuario.

## 5.14. Pruebas en dispositivos reales

La naturaleza de ciertas características del desarrollo exige su validación en dispositivos reales, lo cual resalta la pertinencia y la necesidad de llevar a cabo pruebas en una variedad de dispositivos físicos reales, en contraposición a depender únicamente de emuladores y simuladores.

Igualmente, por la naturaleza de los dispositivos móviles, **se hace necesario probar en distintas resoluciones y densidades de pantallas**, sobre todo en el caso de Android donde existe gran número de gamas de teléfonos y de fabricantes (conocido como fragmentación).

### Dispositivos para pruebas

Para las pruebas mínimas se han seleccionado los dispositivos listados en el cuadro 5.3:

Cuadro 5.3: Listado de equipos de pruebas

Fabricante	Modelo	Sistema Operativo	Versión de S.O.
Apple	iPhone 12 Pro	iOS	16.6.1
Apple	iPhone 6S	iOS	15.7
LG	G6	Android	9 (Pie)
Samsung	S10+	Android	12 (Snow Cone)
Samsung	S22 Ultra	Android	14 (Upside Down Cake)

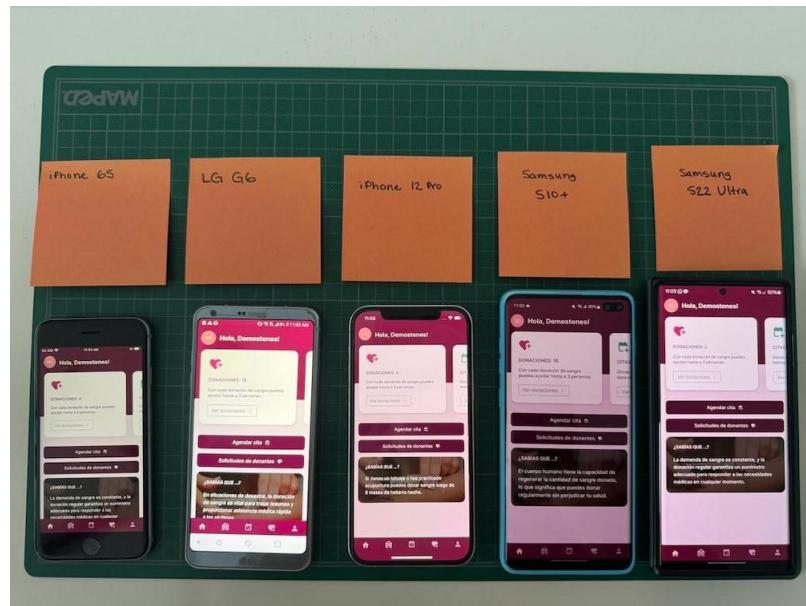


Figura 5.20: **Equipos de prueba.** Fuente: Elaborado por el autor.

La selección de dispositivos ha intentado abarcar distintas gamas de teléfonos, sistemas operativos, densidades y resoluciones.

En la figura 5.20 se pueden apreciar los 5 dispositivos, entre iOS y Android, con la aplicación funcional en un entorno controlado de pruebas local.

## Proceso de pruebas general

Para la configuración general del ambiente de pruebas se ha apalancado en el uso de `Expo dev-client`<sup>58</sup>, que permite empaquetar un `APK` (Android) o un `IPA` (iOS) que se conecte a un servidor de desarrollo para descargar el `bundle` de la aplicación.

Para el empaquetado de estos archivos se ha utilizado `Expo eas`<sup>59</sup>, que permite la compilación en los servidores de `Expo`. Se ha utilizado un perfil de `development`, lo que nos permite utilizar el `Metro Bundler`<sup>60</sup> como servidor del `bundle` de la aplicación.

Para el proceso de compilado del `APK` e `IPA` se puede realizar a través de `EAS`, como se ha mencionado:

<sup>58</sup><https://docs.expo.dev/develop/development-builds/introduction/>. Consultado el 08/01/2024.

<sup>59</sup><https://expo.dev/eas>. Consultado el 08/01/2024.

<sup>60</sup><https://docs.expo.dev/guides/customizing-metro/>. Consultado el 08/01/2024.

```
1 eas build --platform [ios|android] --profile development
```

## Pruebas para Android

Para las pruebas de Android el proceso es simple, pero se vuelve aún más importante debido al alto grado de fragmentación que existe en el ecosistema de Android (Kuroishi et al., 2024), lo que usualmente se puede transferir a problemas con las distintas densidades y resoluciones de pantallas de dichos dispositivos (Wei et al., 2016).

Para realizar las pruebas en dispositivos reales, se debe descargar o instalar el APK en el dispositivo y luego conectar al servidor de desarrollo local, el cual debe ser expuesto a través del comando:

```
1 npx expo start --dev-client --lan
```

Este comando expone el servidor en la red local y a través de un IP que es de acceso dentro de la red, por lo cual todos los dispositivos deben compartir la misma red.

## Pruebas para iOS

Las pruebas para dispositivos iOS es un tanto distinto. Por seguridad, Apple requiere que registremos los dispositivos físicos a través del portal de desarrollo de Apple<sup>61</sup> primero, luego agregar algunos dispositivos al perfil:

```
1 eas device:create
```

Posteriormente es posible agregar las credenciales, certificados y demás configuraciones:

```
1 eas credentials
2 ? Select platform - Use arrow-keys. Return to submit.
3   Android
4   iOS
```

Este proceso documentado en el sitio de Expo<sup>62</sup> nos guía en el proceso de cómo se deben generar todas las configuraciones y vincularlas a nuestro perfil de Expo.

<sup>61</sup> <https://developer.apple.com/>. Consultado el 08/01/2024.

<sup>62</sup> <https://docs.expo.dev/app-signing/managed-credentials/>. Consultado el 08/01/2024.

Una vez realizado esto, se puede compilar el IPA, tal como se ha explicado en el apartado 5.14.3.

El IPA estará disponible a través de Expo y el mismo se deberá descargar junto al perfil. Cuando se tenga el IPA en el dispositivo el proceso siguiente es la descarga del bundle local a través del Metro Bundler. **Este IPA está asociado únicamente a los dispositivos físicos agregados al momento de la firma de seguridad del empaquetado.**

## 5.15. Sitio informativo (Brochure)

No puede haber una aplicación web y un sitio móvil sin una página informativa básica sobre la aplicación. Es por ello por lo que parte del alcance incluyó la creación de un sitio web de bienvenida para mostrar las características principales de la aplicación y ser un sitio de partida para el resto de las aplicaciones.

Con el empleo de este Design System, abordado en la sección 3.3, se ha desarrollado un portal básico que exhibe las funciones generales de la aplicación y motiva a los usuarios a participar en ella. Se puede visualizar este portal en la figura 5.21. **El sitio informativo, al igual que la aplicación web, es completamente responsive.**

Asimismo, se presentan los términos de uso, políticas de privacidad y un procedimiento para verificar una donación, con medidas de protección de datos personales. Este último resulta de gran utilidad para que una persona pueda solicitar el día libre que establece la Ley 164 del 11 de septiembre de 2020<sup>63</sup>.

---

<sup>63</sup>Aplicable únicamente en el sector público.

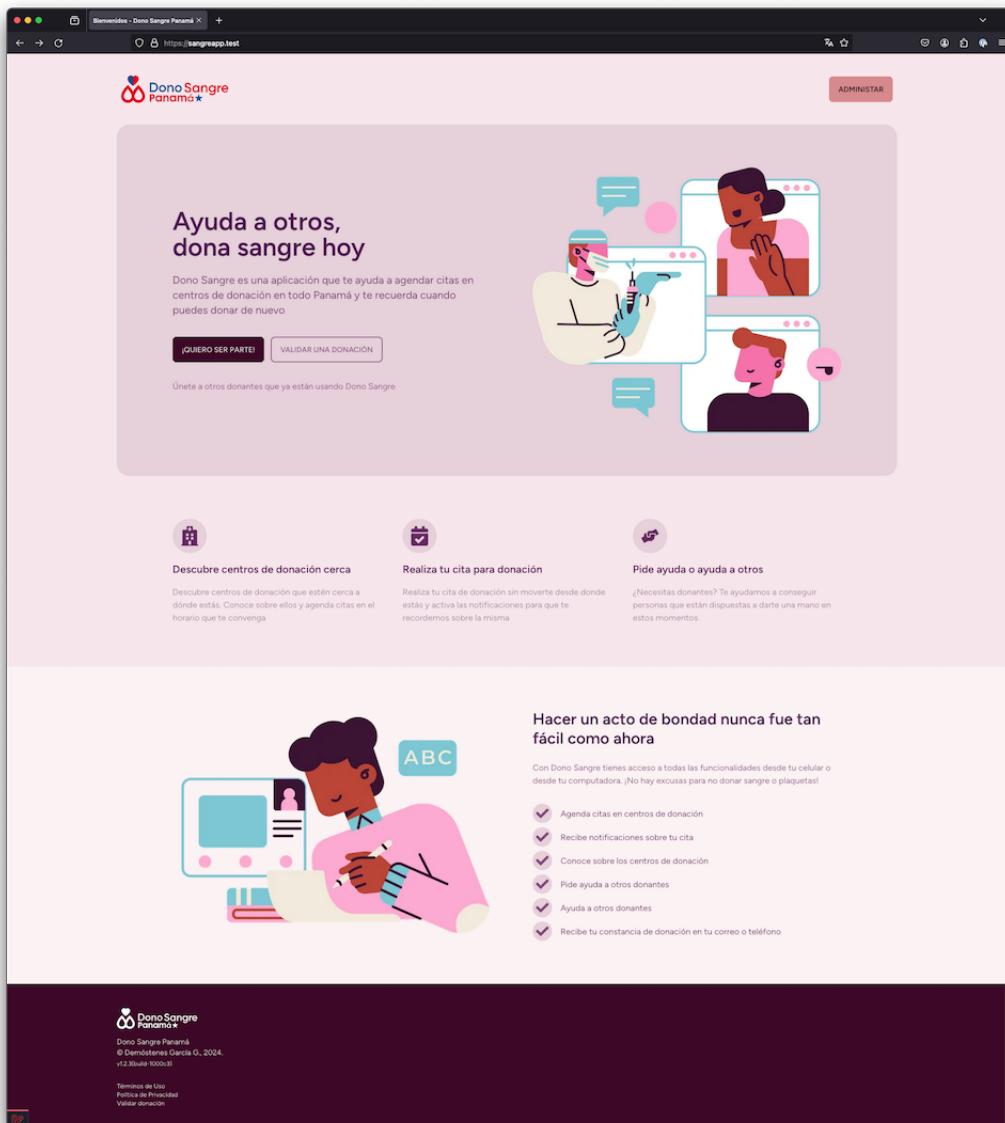


Figura 5.21: Sitio web de bienvenida. Fuente: Elaborado por el autor. Ilustraciones por DrawKit.com

Otras características y optimizaciones del sitio informativo se presentan en el Apéndice N.

## 5.16. Escalabilidad de la solución: caso práctico

Al pensar esta arquitectura, una de las partes modulares que componen la solución ha sido la incorporación de un API para facilitar la incorporación de nuevas características y la comunicación entre distintos factores.

Fuera del alcance original de este TFM, un caso práctico que se atacó para validar esta premisa fue la creación de una aplicación de escritorio que pudiese funcionar, para la verificación presencial a la presentación de una cita.

Para ello, se creó una aplicación de escritorio con el *framework* Tauri<sup>64</sup> y en lenguaje Rust<sup>65</sup> que se comunique con el API de forma segura, mediante una llave de JWT asociada a esa instalación.



Figura 5.22: Aplicación de escritorio en Tauri. Fuente: Elaborado por el autor.

En la figura 5.22 se aprecia la pantalla de esta aplicación donde se requiere el identificador de cita (un UUID), se valida la cita y se marca como presentada o realizada.

Al no ser parte de este alcance, el mismo se amplía con mayor detalle en el Apéndice M.

<sup>64</sup><https://tauri.app/>. Consultado el 19/01/2024.

<sup>65</sup><https://www.rust-lang.org/>. Consultado el 19/01/2024.

## 6. Conclusiones

A lo largo de este trabajo, se ha abordado la persistente problemática de la escasez de donantes en los bancos de donación a nivel nacional, un tema que resuena, tanto en las noticias nacionales como en las redes sociales de la población panameña. La situación, en algunos casos, alcanza niveles alarmantes, lo cual es inaceptable para un recurso vital como la sangre, esencial para mantener el funcionamiento del sistema de salud.

Al definir el alcance de este Trabajo de Fin de Máster, se destacó que este representaba una Prueba de Concepto (PoC), con la condición de ser completamente funcional y facilitar el proceso de donación para los posibles donantes.

En este proyecto, no se ha desarrollado simplemente una aplicación, sino múltiples aplicaciones que colaboran para proporcionar una solución integral. Esta solución abarca la agenda de citas, la gestión de donaciones, la exploración de centros de donación, la solicitud de ayuda para conseguir donantes para una causa específica y la posibilidad de convertir a los donantes en recurrentes.

Durante todo el desarrollo, se han seguido mejores prácticas, tales como el cumplimiento de guías de estilo con ESLint y PHP Code Sniffer (visto en 5.2) y la implementación de procesos de desarrollo ágiles e incrementales, como *Semantic Versioning*, *Conventional Commits* y un proceso de CI/CD (visto en 3.2 y 5.2). Además, se han aplicado técnicas avanzadas en FE y BE, como el desarrollo de un *Design System* para la interfaz de usuario (visto en 3.3), la implementación de una Máquina de estado finita con *Queues* (discutido en 5.8.1) o el uso de Caché para mejorar el rendimiento en algunas áreas (visto en 5.12). Todo este proceso no ha descuidado aspectos críticos como la seguridad del sistema (revisado en 5.10) y la privacidad del usuario (discutido en 5.13), subrayando la importancia de un desarrollo incremental, ágil y de cara a la generación de valor (visto en 4).

El desarrollo ha requerido múltiples lenguajes de programación<sup>1</sup>, tecnologías<sup>2</sup> y plataformas<sup>3</sup>, lo que se podría considerar un desarrollo bastante completo y donde se han podido atacar distintas áreas del desarrollo de software.

---

<sup>1</sup>PHP, TypeScript, JavaScript, Shell.

<sup>2</sup>Tailwind, Tamagui, DaysiUI, React Native, React, Laravel, PostgreSQL, Redis, PHPUnit, Selenium, Server Side Rendering, Throttle, InertiaJS.

<sup>3</sup>Expo, CDN, VPS, Google Firebase, Google Maps, Apple Development Account, hCAPTCHA, Slack.

## 7. Trabajo futuro

Si bien la solución se ideó como una Prueba de Concepto (PoC), se definieron las prioridades en base al proceso de evaluación previo, que se realizó en las etapas tempranas del proyecto, como se explicó en los Requisitos del Proyecto (ver 2.2).

En la etapa actual, la aplicación dispone todas funcionalidades estipuladas en los *must, should* y *could* (salvo la excepción de Gamificación), pero se ha dejado como trabajo futuro aquellas que se mencionan a continuación:

- **Horarios de los centros de donación:** En Panamá, el sector salud recibe a personas en su mayoría en horario de 7:00 a.m. a 2:00 p.m. (días de semana), hasta su cierre a las 3:00 p.m. Con el desarrollo de esta funcionalidad, los centros podrían tener sus propios días de trabajo y horarios de trabajo configurables a través del Panel de Administración.
- **Espacios de donación por centro y hora:** La afluencia a los centros en la actualidad es poca, por lo que se consideró que no era necesario crear sistemas de *slots* (espacios) de donación por hora. De implementarse esta funcionalidad, el Panel de Administración permitiría la configuración de espacios de donación relacionados en centro-hora.
- **Presentación a fundaciones y organismos dedicados a promover la donación:** Siendo la burocracia un tema importante en las instituciones públicas de Panamá, y tomando en consideración el tiempo establecido para el desarrollo de este proyecto, se decidió presentar este Trabajo de Fin de Máster a las instituciones.
- **Plataforma de gamificación:** La integración de sistemas de gamificación podría ofrecer a los usuarios la posibilidad de ganar puntos canjeables en comercios y otras recompensas al realizar donaciones. Sin embargo, es crucial destacar que esta propuesta se encuentra en una zona ambigua y requiere una evaluación cuidadosa en colaboración con las autoridades competentes. Dado que el acto de donar es inherentemente altruista, es esencial considerar otros enfoques éticos en la implementación de esta funcionalidad.
- **Publicación en tiendas (App Store y Google Play) y producción:** El proceso de publicación a tiendas y a producción, se ha decidido dejar como trabajo futuro, mayormente por el proceso que puede conllevar la aprobación (ver Apéndice O).

## A. Apéndice A: Épicas

En los siguientes cuadros, parte de este apéndice, se muestran las épicas (o historias de alto nivel) que componen la solución completa y que abarcarían los distintos sistemas.

Las mismas incluyen, de forma individual, un título, un identificador único y una descripción básica de qué engloba cada una.

Puede conocer más sobre el uso de estas épicas en la unidad 4.3.

Cuadro A.1: SPA-2 / Web - *Architecture*

Título	Web - <i>Architecture</i>
Identificador	SPA-2
Descripción	Mantener la estructura y todo lo relacionado a la aplicación Web y el API

Cuadro A.2: SPA-27 / Mobile - *Architecture*

Título	Mobile - <i>Architecture</i>
Identificador	SPA-27
Descripción	Mantener la estructura y todo lo relacionado a la aplicación móvil

Cuadro A.3: SPA-29 / Web - *Brochure site*

Título	Web - <i>Brochure site</i>
Identificador	SPA-29
Descripción	El <i>Brochure Site</i> se encargará de la parte informativa de la página

Cuadro A.4: SPA-28 / Web - *User Management*

Título	Web - <i>User Management (Register, Login, etc)</i>
Identificador	SPA-28
Descripción	Mantener todo lo relacionado a la administración y gestión de los distintos usuarios

Cuadro A.5: SPA-5 / Mobile - *User Management*

Título	Mobile - <i>User Management (Register, Login, etc)</i>
Identificador	SPA-5
Descripción	Mantener todo lo relacionado a las sesiones y usuarios dentro del App móvil

Cuadro A.6: SPA-30 / Mobile - *Information/Brochure*

Título	Mobile - <i>Information/Brochure</i>
Identificador	SPA-30
Descripción	Todo lo relacionado con la parte informativa, noticias, cápsulas de información y otros temas no directamente relacionados con los procesos dentro del App móvil

Cuadro A.7: SPA-33 / Web - *Donations*

Título	Web - <i>Donations</i>
Identificador	SPA-33
Descripción	Todo lo relacionado a mantener el proceso de donación y las donaciones como un ente

Cuadro A.8: SPA-92 / Mobile - *Donations*

Título	Mobile - <i>Donations</i>
Identificador	SPA-92

Descripción	Todo lo relacionado a mantener el proceso de donación y las donaciones como un ente dentro del app móvil
-------------	--

Cuadro A.9: SPA-35 / Web - *Appointments*

Título	Web - <i>Appointments</i>
Identificador	SPA-35
Descripción	Todo lo relacionado a mantener el proceso de citas para la donación

Cuadro A.10: SPA-34 / Mobile - *Appointments*

Título	Mobile - <i>Appointments</i>
Identificador	SPA-34
Descripción	Todo lo relacionado a mantener el proceso de citas para la donación

Cuadro A.11: SPA-36 / Web - *Donation Centers*

Título	Web - <i>Donation Centers</i>
Identificador	SPA-36
Descripción	Todo lo relacionado a mantener las entidades conocidas como "Bancos de Donación" (Centros de Donación)

Cuadro A.12: SPA-37 / Mobile - *Donation Centers*

Título	Mobile - <i>Donation Centers</i>
Identificador	SPA-37
Descripción	Todo lo relacionado a mantener las entidades conocidas como "Bancos de Donación" (Centros de Donación) dentro del app móvil

Cuadro A.13: SPA-40 / Web - *Gamification*

Título	Web - <i>Gamification</i>
Identificador	SPA-40
Descripción	Todo lo relacionado a las características de gamificación

Cuadro A.14: SPA-41 / Mobile - *Gamification*

Título	Mobile - <i>Gamification</i>
Identificador	SPA-41
Descripción	Todo lo relacionado a las características de gamificación dentro del App móvil

Cuadro A.15: SPA-42 / Web - *Statistics*

Título	Web - <i>Statistics</i>
Identificador	SPA-42
Descripción	Todo lo relacionado a las características de estadísticas

Cuadro A.16: SPA-76 / Web - *Request for donors*

Título	Web - <i>Request for donors</i>
Identificador	SPA-76
Descripción	Todo lo relacionado al proceso de solicitud de un donante

Cuadro A.17: SPA-77 / Mobile - *Request for donors*

Título	Web - <i>Request for donors</i>
Identificador	SPA-77
Descripción	Todo lo relacionado al proceso de solicitud de un donante dentro del app móvil

## B. Apéndice B: Historias de usuario para el *Sprint 1*

En el cuadro B.1 se muestran las historias que se crearon para el *Sprint 1*, indicando su identificador único, el estado inicial de dicha historia, su título, una descripción con los criterios de aceptación, los puntos estimados de esfuerzo y la épica a la que pertenece cada historia individual.

Puede conocer más sobre el fondo de estas historias de usuario en la unidad 4.3.

Cuadro B.1: Sprint Backlog - Sprint 1

Título	ID	Estado	Descripción	Puntos	Épica
[Web] Configure Laravel project	SPA-8	To Do	<p><b>Story</b></p> <p>As a user, I want to see a very basic page that at least prints hello world So I know something is being cooked</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"> <li>■ Gather a list of PHP / Laravel packages.</li> <li>■ Install the list of packages.</li> <li>■ Configure Laravel.</li> <li>■ Have the project running with:           <ul style="list-style-type: none"> <li>• PHP 8.2/8.3+</li> <li>• Laravel 10+</li> <li>• PostgreSQL</li> <li>• React</li> <li>• Inertia.js</li> <li>• Tailwind CSS</li> </ul> </li> </ul>	3	Web - Architecture

[Mobile] Configure RN	SPA-9	To Do	<p><b>Story</b></p> <p>As a user, I want to see at least a hello world message under an iOS or Android device I know something is in the oven</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ Explore Expo / React Native config.</li><li>■ Create a boilerplate for the mobile project.</li><li>■ Gather a list of packages we might need during the mobile project.</li></ul>	2	Mobile - Architecture
-----------------------------	-------	-------	---	---	-----------------------

[Mobile] Create mockup / design for login screen	SPA-10	To Do	<p><b>Story</b></p> <p>As a user I want a pretty login screen for the mobile application So that I can login easily by using my e-mail/password or my social app</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ User can login through social apps (google + apple accounts)</li><li>■ User can login through email + password combination</li><li>■ If a user tries to register through email + password but its already registered through social media, inform the user.</li><li>■ If there is any other error, inform the user.</li></ul>	1	Mobile - User Management
---	--------	-------	--	---	--------------------------

[Mobile] Create moc-kup / design for register screen	SPA-11	To Do	<p><b>Story</b></p> <p>As a user that wants to register for the app, I want to login into the app So that I have access to the other available features</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ User can register only through e-mail + password</li><li>■ User can also login through social media apps</li></ul>	1	Mobile - User Management
--	--------	-------	--	---	--------------------------

[Mobile] Create mockup / design for home screen	SPA-12	To Do	<p><b>Story</b></p> <p>As a user, I want a page that describes what the app is about So that I can understand the purpose of the app</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ Define the name for the application and create a logo</li><li>■ Create a wireframe for the home screen (guest user)</li><li>■ Create a design for the home screen (guest)</li><li>■ The guest screen should link the user to register, login and recover password screens.</li></ul>	2	Mobile - Informative / Brochure
---	--------	-------	---	---	---------------------------------

Explore mobile (React Native / Expo) design libraries	SPA-15	To Do	<p><b>Story</b></p> <p>As a user, I need the style for the application to be consistent and beautiful, So that everything looks similar and not like a Bootstrap like site</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ Select a design library or a component library that fit our needs:<ul style="list-style-type: none"><li>• It should be current (maintained and updated regularly)</li><li>• It should have access to provide a new / different color scheme or themes</li><li>• It should support basic components as Form (Inputs, Labels), Buttons, Cards, Text/Typography, etc.</li><li>• It should support React Native and Expo projects</li></ul></li></ul> <p>*Questions / Notes*</p>	1	Mobile - Informative / Brochure
---	--------	-------	---	---	---------------------------------

[Mobile] Define color palette for mobile application	SPA-16	To Do	<p><b>Story</b></p> <p>As a user, I want a refined color palette that represents the colors of the app So that all the UI elements are consistent and look similar</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ Define a color palette and other design tokens.</li><li>■ Configure this color palette to support Tailwind themes.</li><li>■ Explore the ability to have Storybooks to support our development process.</li></ul>	2	Mobile - Architecture
--	--------	-------	--	---	-----------------------

[API] Define User model	SPA-17	To Do	<p><b>Story</b></p> <p>As a user, I want to exist So that I can use the application and everything else is related to me</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ User should have a name</li><li>■ User should have a lastna-me</li><li>■ User should have an e-mail address</li><li>■ User should have a login mechanism (login+pass-word / social media)</li><li>■ User can have a password (if the users login mecha-nism is email+password)</li><li>■ User can have a birthdate</li></ul>	2	Web - User Management (Register, Login, etc.)
-------------------------	--------	-------	---	---	---

[Web] Create mockup / design for home screen	SPA-19	To Do	<p><b>Story</b></p> <p>As a guest user, I want to understand what the app is about So I have a better understanding about the application and its features</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ The design should be responsive.</li><li>■ It should include features about the application.</li><li>■ It should include install links to the Android and iOS apps.</li><li>■ It should have a footer with information related about the developers of the application.</li><li>■ It should include other useful links.</li></ul>	2	Web - Brochure site
--	--------	-------	--	---	---------------------

[Web] Create mockup / design for login + register screen	SPA-20	To Do	<p><b>Story</b></p> <p>As a user, I want to login or register to this cool application So that I can start using the application</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ Different login mechanisms (user+password vs. social media)</li><li>■ The user can register by using a user+password</li><li>■ User can reset the password</li></ul>	2	Web - User Management (Register, Login, etc.)
--	--------	-------	---	---	---

[FE] Configure basic libraries	SPA-24	To Do	<p><b>Story</b></p> <p>As a developer, I need a list of basic FE libraries supported by React So that I can start working on the other components and features</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"> <li>■ Curate the list gathered on SPA-8</li> <li>■ Install the packages</li> <li>■ Document the list of packages used</li> </ul>	1	Mobile - Architecture
[API/Web] Configure basic packages	SPA-25	To Do	<p><b>Story</b></p> <p>As a user, I want to be able to login with social media and have other features So that I can use the application with all the features I'm expecting</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"> <li>■ Gather a list of basic packages and install them</li> </ul>	1	Web - Architecture

[Web] Explore web (React) design libraries	SPA-26	To Do	<p><b>Story</b></p> <p>As a user, I want a very refined style and design So this application looks like it was built by people who know what they're doing</p> <p><b>Acceptance criteria</b></p> <ul style="list-style-type: none"><li>■ Select a design library / component library that fit our needs:<ul style="list-style-type: none"><li>• The component library should support Tailwind CSS.</li><li>• The component library should provide a way to set a custom theme / color scheme.</li><li>• The components should be built for React.</li></ul></li></ul>	1	Web - Architecture
--	--------	-------	---	---	--------------------

## C. Apéndice C: Reglas del Proyecto Ágil

En el cuadro C.1 se muestran las reglas que se establecieron para el proyecto en torno a la metodología para la gestión del mismo. Puede ver su funcionalidad en la unidad 4.1.

Cuadro C.1: **Reglas del proyecto ágil**

Regla
<b>Duración del Sprint</b> 1 semana
<b>Historia de Usuario - Definición de Completado</b> Una historia de usuario se considerará como completada cuando: <ul style="list-style-type: none"><li>- Tenga una descripción</li><li>- Tenga una estimación relativa (se utilizará Fibonacci para la estimación)</li><li>- Tenga unos criterios de aceptación</li><li>- El código que implemente la descripción y soporte los criterios de aceptación ha sido completado</li><li>- Posee pruebas automatizadas para validar los criterios de aceptación (dentro de lo posible)</li><li>- El código ha sido desplegado en el ambiente que corresponde</li></ul>
<b>Sprint - Definición de Completado</b> Un Sprint se considerará como completado cuando: <ul style="list-style-type: none"><li>- El tiempo dispuesto ha llegado a su fin</li><li>- Las historias completadas y aceptadas hayan sido desplegadas al ambiente que le(s) corresponde</li></ul>
<b>Estimaciones</b> <ul style="list-style-type: none"><li>- Las historias de usuario y sus estimaciones serán relativas y representarán su relación a nivel de esfuerzo vs. una historia previa existente. Jamás se utilizarán estimaciones de tiempo.</li></ul>

- Los posibles valores de estimación seguirán la secuencia de Fibonacci (1, 2, 3, 5, 8, ...) obligando que dichos valores siempre sean relativos.
- Si una historia de usuario tiene un valor de 5 puntos o más, deberá ser recortada en historias más pequeñas.

## D. Apéndice D: *Roadmap del producto*

El *Roadmap del Producto* (o *Product Roadmap*) constituye el documento estratégico mediante el cuál se establecen la entrega de los objetivos a lo largo del tiempo. En la figura D.1 se muestra el *Roadmap* del producto inicial para este proyecto. Puede ver su desarrollo en la unidad 4.3.3.

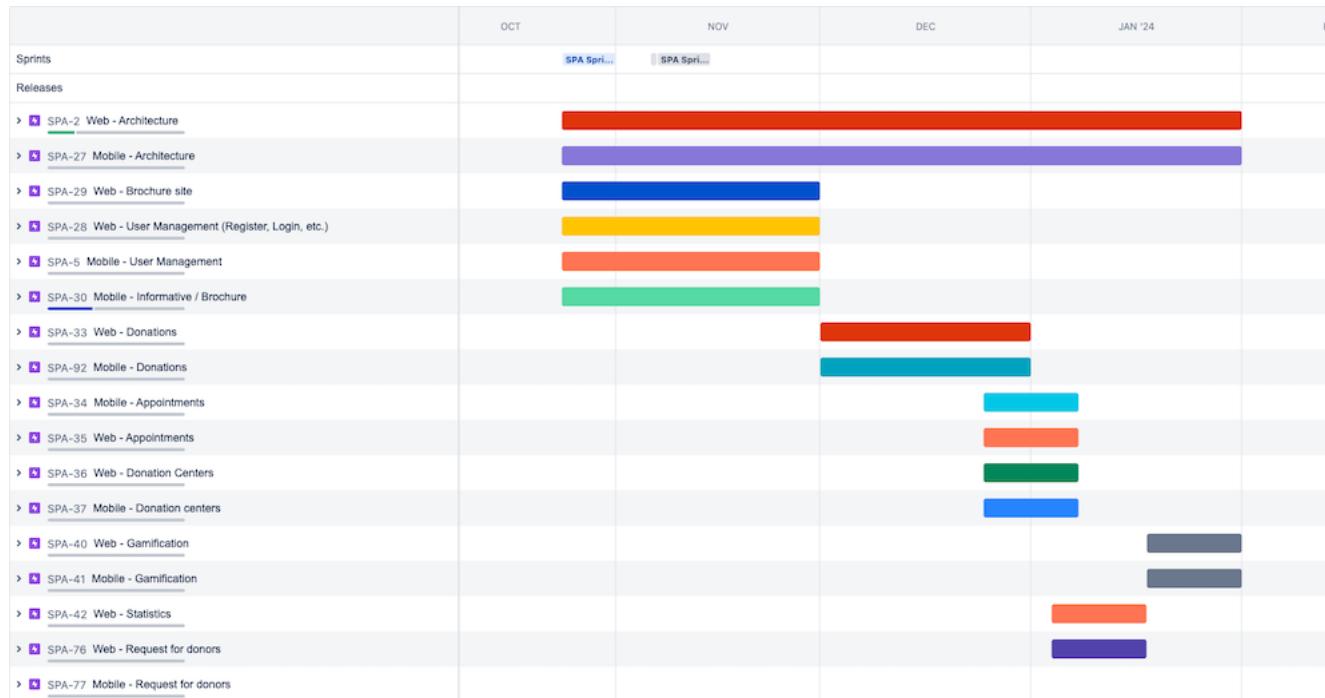


Figura D.1: Product Roadmap. Fuente: Elaborado por el autor.

## E. Apéndice E: Velocidad evidenciada

En el cuadro E.1 se muestra la evidencia recuperada luego de 7 *Sprints* donde se pudo determinar que el número correcto de puntos que podían ser atacados en un *Sprint* era de alrededor 23 puntos. Puede ver más sobre su función en la unidad 5.1.

Cuadro E.1: *Velocidad (evidenciada) del equipo*

Sprint	Planificado	Ejecutado	Notas de retrospectiva
Sprint 1	21	21	El Sprint se sintió pesado, pero mayormente a una sub-estimación de las historias (las historias requerían mayor estimación).
Sprint 2	14	14	El Sprint se sintió algo holgado, aunque hubo días festivos por festividades patrias. Las historias se estimaron mejor.
Sprint 3	14	17	El Sprint se sintió demasiado holgado. Si bien no deberían agregarse más historias al Sprint, se hizo con el solo fin de estimar velocidad.
Sprint 4	23	23	El Sprint y estimaciones se sintieron correctas. El último día quizás se pudo trabajar un poco más.
Sprint 5	24	22	El Sprint se sobre-estimó. Una historia se estimó incorrectamente al no tener datos exploratorios de la misma.
Sprint 6	24	21	El Sprint se sobre-estimó. Una historia se estimó incorrectamente al no tener datos exploratorios de la misma.
Sprint 7	23	23	El Sprint e historias se sintieron bien.

## F. Apéndice F: Supervisor de colas

El supervisor se encarga de la ejecución en *background* del inicio de ciertas tareas recurrentes y en paralelo que el sistema requiere. Estas tareas no bloquean el hilo principal de ejecución.

Para configurar un supervisor con la librería de supervisor es requerido hacer la misma a nivel del Sistema Operativo.

La configuración del supervisor<sup>1</sup> de colas (*Queues*) es la siguiente:

```
1 [program:donasangre-worker]
2 process_name=%(program_name)s_%(process_num)02d
3 command=php /home/staging/current/artisan queue:work --sleep=3 --
   tries=3 --max-time=3600
4 autostart=true
5 autorestart=true
6 stopasgroup=true
7 killasgroup=true
8 user=cirobot
9 numprocs=8
10 redirect_stderr=true
11 stdout_logfile=/home/staging/current/logs/worker.log
12 stopwaitsecs=3600
```

```
1 [program:donasangre-horizon]
2 process_name=%(program_name)s
3 command=php /home/staging/current/artisan horizon
4 autostart=true
5 autorestart=true
6 user=cirobot
7 redirect_stderr=true
8 stdout_logfile=/home/staging/current/logs/horizon.log
9 stopwaitsecs=3600
```

Esta configuración ha sido hecha en un Sistema Operativo Linux, distribución: Ubuntu Server 23.10 (Mantic).

Puede conocer más sobre el uso de supervisor, dentro del contexto de este Trabajo de Fin de Máster, en la unidad 5.8.

<sup>1</sup><http://supervisord.org/>. Consultado el 19/01/2024.

## G. Apéndice G: Marcadores personalizados con Leaflet en JavaScript

Para la generación de marcadores (*markers*) personalizados con Leaflet es requerido utilizar el API disponible en JavaScript, desde nuestro código en React.

Un ejemplo de este código es el siguiente:

```
1 const iconPrivate = L.divIcon({
2   className: 'custom-icon-private',
3   html: '<div class="w-8 h-8 bg-yellow-600 text-yellow-900 rounded-full shadow-lg text-center">${iconAsString}</div>',
4   iconSize: [20, 20],
5   iconAnchor: [10, 10],
6 });
7
8 const iconPublic = L.divIcon({
9   className: 'custom-icon-public',
10  html: '<div class="w-8 h-8 bg-pink-600 text-pink-900 rounded-full shadow-lg">${iconAsString}</div>',
11  iconSize: [20, 20],
12  iconAnchor: [10, 10],
13 });
14
15 // ...
16
17 <Marker
18   // ...
19   icon={
20     bank.organization.type === 'private' ? iconPrivate :
21     iconPublic
22   }
23 >
```

Este código utiliza también los tokens de diseño creados y expuestos a través de PlaquetaDS (ver 3.3).

Puede expandir sobre la implementación en la unidad 5.9.1.

## H. Apéndice H: Archivo de migración para soportar notificaciones Push

A través de los archivos de migración de Laravel se crea una nueva tabla llamada devices para permitir una relación 1-n (1 a muchos o HasMany) entre el usuario (user) y los dispositivos (devices).

Esta tabla permite que un usuario pueda tener múltiples dispositivos asociados.

```
1 Schema::create('devices', function (Blueprint $table) {
2     $table->id();
3     $table->foreignId('user_id')->constrained('users')->
4         cascadeOnDelete()->cascadeOnUpdate();
5     $table->string('notification_provider')->nullable();
6     $table->string('notification_token')->nullable();
7     $table->string('device_identifier')->nullable();
8     $table->string('os')->nullable();
9     $table->string('os_version')->nullable();
10    $table->string('manufacturer')->nullable();
11    $table->string('model')->nullable();
12    $table->timestamps();
13
14    $table->unique(['user_id', 'notification_provider', 'notification_token']);
15    $table->index(['notification_provider', 'notification_token']);
16    $table->index('user_id');
17    $table->index('device_identifier');
18});
```

Puede ver el del desarrollo en cuestión en la unidad 5.11.

# I. Apéndice I: Implementación de *Lazy Loading* en React

Lazy loading es una técnica para cargar los contenidos bajo demanda del usuario.

React ya implementa mecanismos para implementar *Lazy Loading* (a través de `React.lazy1`) y se ha utilizado `React.Suspense2` para mostrar un componente de carga mientras los componentes son importados.

Para ello, es necesario primero la importación de ellos de forma retrasada, a través de `lazy()`:

```
1 const RequestSection2 = lazy(() => import("./RequestSection2"));
2 const RequestSection3 = lazy(() => import("./RequestSection3"));
3 const RequestSection4 = lazy(() => import("./RequestSection4"));
```

Se utiliza `InteractionManager.runAfterInteractions3` para la carga después de la finalización de todas las interacciones:

```
1 ...
2 useEffect(
3     useCallback(() => {
4         if (!mounted) {
5             InteractionManager.runAfterInteractions(() => {
6                 setMounted(true);
7             });
8         }
9
10        return () => {
11            setMounted(false);
12        };
13    }, []);
14 );
15 ...
```

Se pinta el componente secundario solo cuando el componente principal ha terminado su renderizado, lo que dispara la carga retrasada de partes del formulario:

```
1 {!!mounted ? (
2     <Suspense fallback={<LoadingSection />}>
```

<sup>1</sup><https://react.dev/reference/react/lazy>. Consultado el 12/01/2024.

<sup>2</sup><https://react.dev/reference/react/Suspense>. Consultado el 12/01/2024.

<sup>3</sup><https://reactnative.dev/docs/interactionmanager>. Consultado el 12/01/2024.

```
3   <RequestSection2
4     control={control}
5     isLoading={isLoading}
6     errors={errors}
7     watch={watch}
8     setValue={setValue}
9   />
10  </Suspense>
11 ) : null}
```

---

Puede evaluar el uso de esta optimización a nivel de la aplicación en la unidad 5.12, junto a otras optimizaciones.

# J. Apéndice J: Expresión regular de cédula panameña

La “cédula” (o “Cédula de Identidad Personal”) compone el documento de identidad nacional para todo ciudadano de la República de Panamá.

El mismo se divide en tres grupos de datos. La primera parte indica dónde fue registrado el ciudadano, comúnmente conocido como “provincia”. Las otras dos partes corresponden al “folio” y al “asiento” del registro. Adicionalmente, algunas cédulas tienen algunas notaciones como sufijos (como AV para Antes de Vigencia o PI para Pueblos Indígenas).

La expresión regular completa como validación en Laravel es la siguiente:

```
1 // ...
2 public function validate(string $attribute, mixed $value, Closure
3     $fail): void
4 {
5     $regex = '/^P$|^([PE|EN|[23456789])(?:A|P)?|1[0123]?(?:A|P)?)$|^([PE|EN|[23456789])(?:AV|PI)?|1[0123]?(?:AV|PI)?)\-$|^([PE|EN|[23456789])(?:AV|PI)?|1[0123]?(?:AV|PI)?)-(\d{1,4})-(\d{1,6})$/i';
6     if (! preg_match($regex, $value)) {
7         $fail(__('validation.custom.national_id.cedula'));
8     }
9 }
```

El código ha sido basado en el desarrollo de Juan Merlos como autor principal del repositorio y disponible bajo licencia MIT. El mismo se encuentra disponible en <https://github.com/merlos/cedula-panama> (Consultado el 19/01/2024).

Puede consultar sobre la implementación de esta validación en la unidad 5.10.3.

# K. Apéndice K: Eventos a través de máquinas de estado

El *backend* del sistema, tanto a nivel del API como a nivel de la plataforma administrativa, se soporta tanto con las *Queues* (colas) como con las máquinas de estado.

A continuación, se expande sobre los distintos estados dentro de la máquina de estado de Citas (*Appointment*), sus transiciones, su configuración dentro de un modelo y como sus acciones se conectan con otras partes de la aplicación.

Puede conocer más sobre su uso en la unidad 5.8.1.

## K.1. Estados en citas

- **Recibida:** La solicitud ha sido registrada en el sistema. Es el estado inicial.
- **Procesada:** La solicitud ha sido procesada y las notificaciones han sido enviadas.
- **En revisión:** La solicitud se encuentra lista para ser revisada (manualmente) por personal del centro de donación.
- **Aprobada:** La solicitud fue aprobada por el centro de donación.
- **Cancelada por el usuario:** La solicitud fue cancelada por el usuario que la solicitó. Es un estado final.
- **Cancelada por el centro:** La solicitud fue cancelada por el centro de donación. Es un estado final
- **No presentada:** El usuario que hizo la solicitud no se presentó a la misma. Es un estado final.
- **Realizada:** La solicitud fue realizada y completada. Es un estado final

## K.2. Transiciones en citas

- **Desde Recibida:** Procesada.
- **Desde Procesada:** En revisión, Cancelada por el usuario.
- **Desde En revisión:** Aprobada, Cancelada por el centro.
- **Desde Aprobada:** Cancelada por el usuario, No presentada, Realizada.
- **Desde cancelada por el usuario:** — (estado final)
- **Desde cancelada por el centro:** — (estado final)
- **Desde No presentada:** — (estado final)
- **Desde Realizada:** — (estado final)

## K.3. Configuración de máquina de estado con el modelo

Las colas (*Queues*) soportan mucha de su funcionalidad de forma asíncrona a través de la incorporación de acciones y eventos conectados a las máquinas de estado.

Estas máquinas de estado permiten la ejecución de notificaciones y otras acciones a partir del cambio de estado en una instancia.

A continuación se muestra un ejemplo de cómo en un modelo la transición entre dos estados dispararía una notificación:

```
1 class Appointment extends Model
2 {
3     // ...
4     public function transitionBasedOnNextState(string $nextState)
5         // ...
6         } elseif ($nextState === AppointmentStatusStateMachine::APPROVED) {
7             $this->approve();
8         }
9         // ...
10    }
11
12    // ...
13 }
```

```
14     public function approve()
15     {
16         $this->status()->transitionTo(AppointmentStatusStateMachine
17 ::APPROVED);
18         $this->user->notify(new ApprovedAppointmentNotification(
19 $this));
20     }
21
22 // ...
```

El método `transitionBasedOnNextState` es el que se utiliza para la transición del estado de un objeto. Este método recibe el siguiente posible estado y verifica si es posible la transición.

Si es posible, ejecuta la acción (en este caso ejecuta el método `approve()` el cuál a su vez realiza la transición de estado y ejecuta una notificación (*Push* y/o correo).

## K.4. Notificaciones Push con Expo

Para las notificaciones Push, la configuración a nivel de la configuración es ligeramente distinta a las de correo similar a la siguiente:

```
1 class NewDonationNotification extends Notification implements
2 ShouldQueue
3 {
4
5     // ...
6
7     public function via(object $notifiable): array
8     {
9         return [ 'mail', ExpoChannel::class];
10    }
11
12    // ...
13
14    public function toExpo(object $notifiable): ExpoMessage
15    {
16        return ExpoMessage::create()
17            ->badge(1)
18            ->enableSound()
19            ->title('Nueva donación realizada')
20            ->body('Gracias por tu donación!');
21    }
22
23 // ...
```

## L. Apéndice L: Configuración de Firebase en Expo

Tal como se menciona en la unidad 5.7, el inicio de sesión a través de OAuth con Google requiere la configuración de un proyecto con Google Firebase. Para ello es requerido registrar el proyecto en Google Firebase y luego generar una configuración en nuestro proyecto con Expo.

La configuración de Firebase en el proyecto de Expo es similar a la siguiente, a través del archivo app.json:

```
1 ...
2   "ios": {
3     ...
4     "googleServicesFile": "./GoogleService-Info.plist",
5     "bundleIdentifier": "com.demogar.donasangrepanama"
6   },
7   "android": {
8     ...
9     "googleServicesFile": "./google-services.json",
10    "package": "com.demogar.donasangrepanama"
11  },
12 ...
```

# M. Apéndice M: Aplicación de escritorio con Tauri

Tauri<sup>1</sup> es un *framework* para el desarrollo de aplicaciones de escritorio que expone funcionalidad nativa del sistema operativo y que se trabaja en el lenguaje Rust<sup>2</sup>.

Para poder demostrar la escalabilidad a través del API de la solución planteada, se decidió crear como trabajo adicional fuera del enfoque de este Trabajo de Fin de Máster una aplicación en escritorio con este *framework*.

El enfoque de esta aplicación es muy básico y su uso sería exclusivo para ser utilizado como una validación en sitio de que un usuario se presentó físicamente a una cita de donación.

Tauri permite utilizar tecnologías web para el FE de las aplicaciones en escritorio, por lo que crearemos primero un proyecto React con NextJs<sup>3</sup>:

```
1 yarn create next-app --use-yarn
```

Y dentro de este proyecto crearemos un proyecto con Tauri:

```
1 yarn create tauri-app
```

La guía de Tauri explica a mayor detalle este proceso<sup>4</sup>.

Una vez configurado el proyecto base, se puede adjuntar la configuración del Sistema de Diseño (PlaquetaDS), que pasaría a ser parte de las dependencias del proyecto.

La pantalla inicial utiliza las mismas tecnologías web que se utilizaron para el resto del Trabajo de Fin de Máster. Las mismas dependen de dos variables de entorno que son injectados en el proyecto: la ruta para marcar una cita como atendida y la llave de JWT para firmar la solicitud.

```
1 export default function Home() {
2   const formRef = useRef<HTMLFormElement>(null);
3   const inputRef = useRef<HTMLInputElement>(null);
```

<sup>1</sup><https://tauri.app/>. Consultado el 19/01/2024.

<sup>2</sup><https://www.rust-lang.org/>. Consultado el 19/01/2024.

<sup>3</sup><https://nextjs.org/>. Consultado el 19/01/2024.

<sup>4</sup><https://tauri.app/v1/guides/getting-started/setup/next-js/>. Consultado el 19/01/2024.

```
4  const focusOnInput = useCallback(() => {
5    inputRef.current?.focus();
6  }, []);
7  async function onSubmit(event: FormEvent<HTMLFormElement>) {
8    event.preventDefault();
9    const formData = new FormData(event.currentTarget);
10   const response = await fetch(
11     env.process.URL,
12     {
13       method: "POST",
14       body: formData,
15       headers: {
16         Authorization: env.process.JWT_TOKEN,
17         Accept: "application/json",
18       },
19     }
20   );
21
22   formRef.current?.reset();
23   inputRef.current?.focus();
24
25   if (!response.ok) {
26     toast.error("No se ha podido finalizar su cita");
27     return;
28   }
29
30   if (response.ok) {
31     toast.success("Su cita ha sido finalizada, ¡Gracias por donar!");
32     return;
33   }
34 }
35
36 return (
37   <main
38     className="bg-gray-200 h-screen w-screen justify-center flex
39     flex-col text-gray-800 gap-10 select-none px-10"
40     onClick={focusOnInput}
41   >
42     <div className="justify-center flex" onClick={focusOnInput}>
43       <Image
44         src="/logo-full.svg"
45         alt="Vercel Logo"
46         width={300}
47         height={150}
48         priority
49       />
50     </div>
51     <div
52       className="flex justify-center flex-col text-center gap-3"
53       onClick={focusOnInput}
54     >
55       <h1 className="text-gray-800 text-3xl">
```

```
55         {"¡Gracias por presentarse a su cita!"}
56     </h1>
57     <p className="text-lg">
58         {"Si desea finalizar su cita, ingrese su identificador:"}
59     </p>
60   </div>
61   <form
62       className="flex justify-center flex-col gap-5"
63       onSubmit={onSubmit}
64       ref={formRef}
65   >
66     <div className="flex justify-center gap-5">
67       <label className="form-control w-full">
68         <div className="label">
69           <span className="label-text text-gray-900 text-lg">
70             Identificador de cita
71           </span>
72         </div>
73         <div className="relative">
74           <input
75             type="text"
76             name="uuid"
77             placeholder="####-####-####-####"
78             className="input input-bordered border-gray-600 w-
full input-lg pr-16"
79             autoFocus
80             ref={inputRef}
81           />
82           <div>
83             <button className="absolute top-5 right-5 btn btn-
xs btn-circle btn-secondary bg-gray-800 border-gray-800" type="button"
onClick={() => formRef.current?.reset()}>
84               /* reset */
85               <svg
86                 xmlns="http://www.w3.org/2000/svg"
87                 className="icon icon-tabler icon-tabler-x"
88                 width="16"
89                 height="16"
90                 viewBox="0 0 24 24"
91                 strokeWidth="1.5"
92                 stroke="currentColor"
93                 fill="none"
94                 strokeLinecap="round"
95                 strokeLinejoin="round"
96               >
97                 <path stroke="none" d="M0 0h24v24H0z" />
98                 <line x1="18" y1="6" x2="6" y2="18" />
99                 <line x1="6" y1="6" x2="18" y2="18" />
100               </svg>
101             </button>
102           </div>
103         </div>
104         <div className="label">
```

```
105          <span className="label-text-alt text-gray-700 text-lg">
106            " >
107              Ingrese el código único de su cita
108            </span>
109          </div>
110        </label>
111      </div>
112      <button
113        className="btn bg-brand-800 hover:bg-brand-600 text-brand-300 btn-lg uppercase"
114        type="submit"
115        >
116          {"Finalizar cita"}
117        </button>
118      </form>
119      <Toaster />
120      <p className="text-gray-500">{'I 2023-${new Date().getFullYear()} Dono Sangre.'}</p>
121    </main>
122  );
123}
```

Al compilar el proyecto con Tauri, se puede utilizar el comando:

```
1 yarn tauri build
```

Esto creará un ejecutable para instalar en la aplicación y la misma funcionaría como cualquier otra aplicación instalable en el sistema operativo. En la figura M.1 se ve la aplicación disponible en el listado de aplicaciones del sistema (Sistema operativo macOS Monterey).



Figura M.1: Aplicación de Tauri en listado de apps. Fuente: Elaborado por el autor.

# N. Apéndice N: Otras características del sitio informativo

El sitio informativo es la primera cara al usuario final y brinda un vistazo previo y breve a las características más esenciales de la plataforma y los sistemas que la incorporan.

Además de lo ya señalado en la unidad 5.15, el sitio informativo implementa algunas características que corresponden a mejoras visuales y no visuales dentro del sitio, los cuales se analizan en los próximos puntos.

## N.1. Sitio responsivo

Si bien se tiene acceso a aplicaciones móviles multiplataforma soportadas en React Native, el sitio web informativo y la aplicación web son totalmente responsivas, lo que hace que la aplicación pueda ser visualizada en formato optimizado para dispositivos móviles.

En la figura N.1 se puede apreciar el sitio web informativo en 3 distintas resoluciones demostrando 3 tipos de formatos distintos: vista para móvil, vista para tableta, vista para computadora de escritorio.

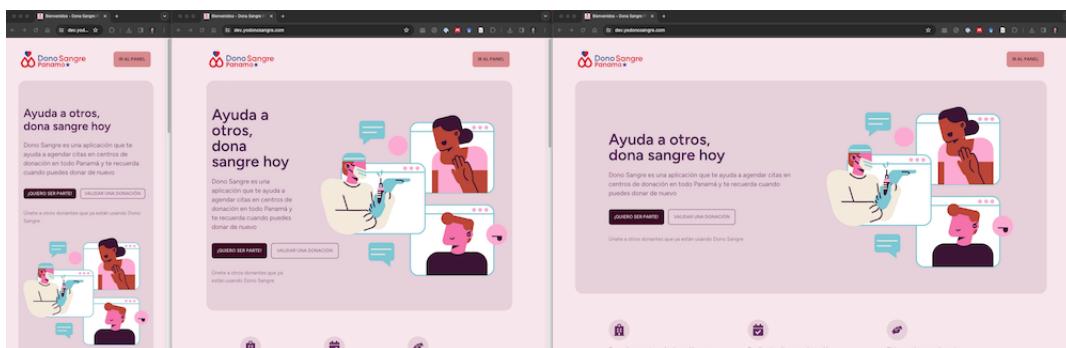


Figura N.1: Sitio responsivo (móvil, tableta y escritorio). Fuente: Elaborado por el autor.

## N.2. Etiquetas para SEO y OpenGraph

El protocolo OpenGraph<sup>1</sup> es un protocolo creado por Facebook / Meta para permitir la promoción de contenido más rico a través de redes sociales, mediante algunas etiquetas de tipo `meta` dentro del contenido.

Además de Facebook, otras plataformas como Twitter (X) o LinkedIn utilizan las etiquetas de este protocolo para poder desplegar mayor información rica en sus redes sociales.

Para ayudar con la optimización de los motores de búsqueda (SEO, por sus siglas en inglés) y mejor información al momento de compartir a través de Redes Sociales, se han verificado las etiquetas como el `title` o el `description` del encabezado de las respuestas, además de agregar favicons por plataforma y algunas etiquetas de OpenGraph como el `og:title`, `og:type`, `og:image` y `og:url`.

En la figura N.2 se puede apreciar como se ve esta información al compartirla a través de alguna red social que implemente el protocolo.



Figura N.2: Sitio responsivo. Fuente: Elaborado por el autor.

<sup>1</sup><https://ogp.me/>. Consultado el 26/01/2024.

## N.3. Auditoría de mejores prácticas

Se ha seguido las guías de recomendación de Google Lighthouse<sup>2</sup> como base para hacer una auditoría menor a través de las herramientas que proporcionan.

Estas mejores prácticas incluyen:

- Rendimiento: Se ha agregado `gzip` para todos los archivos de texto, como HTML (configurado en servidor de Nginx). También la carga previa de algunas imágenes.
- Accesibilidad: Se ha verificado el uso de algunas etiquetas o atributos, como el uso de `alt` para las imágenes, el uso de `title` en los enlaces y el uso de estricto marcado semántico.
- Mejores prácticas: Se ha implementado SSL, correcto `doctype` dentro del HTML, imágenes con resolución correcta.

En la figura N.3 se puede apreciar el resultado de la auditoría estos tres puntos, mientras que el cuarto punto se comentó en la sección anterior.

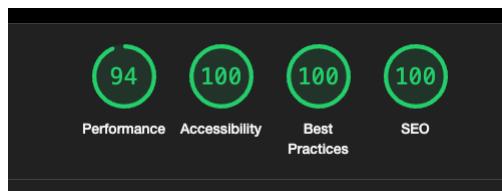


Figura N.3: Métricas en Google Lighthouse. Fuente: Elaborado por el autor.

Adicionalmente a las recomendaciones de Google Lighthouse, se han utilizado tres métricas para verificar los signos vitales de rendimiento (para la página inicial):

- LCP (*Largest Contentful Paint*) (en segundos): La cantidad de tiempo que demora la parte más grande del contenido en ser cargada y mostrada al usuario. Debe ser **1.2s** o menos.
- TBT (*Total Blocking Time*) (en milisegundos): La cantidad de tiempo que los *scripts* bloquean la interacción inicial del usuario. Debe ser de **150ms** o menos.

<sup>2</sup><https://developer.chrome.com/docs/lighthouse/overview>. Consultado el 26/01/2024.

- CLS (*Cummulative Layout Shift*): La cantidad de contenido que cambia a medida que el sitio está cargando. Debe ser **0.1** o menos.

Dadas estas métricas se ha conseguido la siguiente puntuación, según GTMetrix<sup>3</sup>, según se ve en la figura N.4.



Figura N.4: Métricas en GTMetrix. Fuente: Elaborado por el autor.

Esto se ha conseguido realizando algunas optimizaciones en el servidor y la forma en que los contenidos son entregados al usuario:

- Uso de compresión gzip<sup>4</sup>.
- Habilitación de HTTP/2 en Nginx<sup>5</sup>.
- Caché para recursos estáticos (CSS, JS y HTML, por ejemplo) de 30 días. Adicionalmente los recursos están versionados, por lo que si ocurre alguna actualización los mismos serán vistos como nuevos.
- Utilización de la regla de priority en los recursos<sup>6</sup>.

Esto indica que el sitio principal e informativo de bienvenida cumple con las mejores prácticas para la optimización de sitios web, según las sugerencias de Google Lighthouse y GTMetrix.

<sup>3</sup><https://gtmetrix.com/pages/dev.yodonosangre.com/FxBN4vXv/>. Consultado el 4/2/2024.

<sup>4</sup>Ver reporte en

<https://www.techrepublic.com/article/how-to-configure-gzip-compression-with-nginx/>. Consultado el 4/2/2024.

<sup>5</sup><https://www.nginx.com/resources/glossary/http2/>. Consultado el 2/4/2024.

<sup>6</sup><https://www.debugbear.com/blog/priority-hints>. Consultado el 2/4/2024.

## N.4. Archivo para robots

Se ha creado un archivo `robots.txt` con indicaciones para los robots de búsqueda de aquellas partes del sitio pueden ser indexadas por ellos.

Este archivo permite el indexado de todas las páginas, por todos los robots, menos aquellas rutas detrás de una sesión, como lo serían el panel de administración el panel de usuario.

## N.5. Sistema público de validación de citas de donación

Otra funcionalidad de interés implementada dentro del sitio público es el sistema público de validación de donaciones y de citas de donación.

El uso común que se le puede dar a esta funcionalidad es para validar si una cita de donación o una donación final son reales, lo que permitiría validar si una persona realmente se presentó a una cita o si una donación se realizó finalmente.

Hay dos usos potenciales para esta funcionalidad:

- **Validación por parte del empleador:** En Panamá, es común que los empleadores ofrezcan ciertos beneficios o permisos a sus colaboradores a la hora de realizar actos altruistas durante horas de trabajo, la donación de sangre no escapa de ellos. En el sector público incluso existe la Ley 164 de 2020 (Gaceta Oficial, República de Panamá, 2020) que manifiesta que el servidor público tendrá derecho a no asistir a su jornada laboral.
- **Validación por parte del beneficiario:** Las donaciones por reposición deben ser verificadas por el centro de donación y la institución médica que atiende al paciente. Es por ello que el beneficiario de dicha donación, quién requiere la misma, requerirá la validación de dicha donación.

Con respecto al primer punto, la precitada Ley menciona textualmente lo siguiente en su artículo tercero:

*"Artículo 3. Los servidores públicos que realicen la donación de sangre en los bancos oficiales de sangre reglamentados por el Ministerio de*

***Salud y la Caja de Seguro Social tendrán derecho a no asistir a su jornada laboral el día que efectúen la donación de sangre, siempre que presenten el documento que acredite su donación.***

Para este proceso es necesario validar tanto una cita de donación como una donación final. El hecho es que una cita de donación no necesariamente se convertirá en una donación, ya que el proceso de donación requiere una verificación del donante según las normas establecidas por el centro y el donante podrá ser rechazado o podrá ser remitido para una donación futura.

Tanto la cita de donación, como la donación, cuentan con identificadores únicos en formato UUID (alfanuméricos de 36 caracteres), al igual de presentar un código QR para facilitar el proceso de verificación. El sistema de verificación permite validarla con cualquiera de estos dos mecanismos.

Si analizamos el código QR de una donación podremos ver la información expuesta a través de la imagen N.5, en formato JSON:



Figura N.5: Desglose de información en QR. Fuente: Elaborado por el autor.

- **uuid:** Identificador único de la donación o cita. Código de tipo UUID.
- **app\_version:** Versión del lado del servidor, en formato *Semantic Versioning*, que contiene la versión que generó dicho código.
- **date:** Fecha en formato timestamp de la cita o donación.
- **type:** Tipo de registro (donation para Donación, appointment para Cita).

## Obtener la información del QR

Para obtener la información del QR se debe primero leer la información del QR de alguna forma. Para esto, se utiliza cualquier lector de QR disponible. Luego, se debe interpretar la cadena de texto correspondiente. Una vez obtenido, se debe reemplazar algunos caracteres básicos (como por ejemplo convertir ä en "a, algo que ocurriría en teclados en formato *U.S. International* en macOS) y se requiere verificar si la cadena de texto suministrada es un JSON. Si es un JSON se interpreta como tal, de lo contrario se interpreta como una cadena de texto. El código que lo soporta es el siguiente:

```
1 export function getUuidFromQr(qrCodeInfo: string) {
2   // first, we need to transform ä -> "a, è -> "e, etc
3   let qrInfo = qrCodeInfo.toLowerCase().trim();
4   qrInfo = qrInfo.replace(/ä/g, '"a').replace(/è/g, '"e').replace(
5     /ï/g, '"i').replace(/ö/g, '"o').replace(/ü/g, '"u');
6
7   // now we need to know if this is a JSON or just a UUID
8   const isJson = qrInfo.startsWith('{') && qrInfo.endsWith('}');
9
10  if (isJson) {
11    const json = JSON.parse(qrInfo);
12    return json.uuid;
13  }
14
15  return qrInfo;
}
```

A través de esto se puede limpiar la cadena de texto para solo quedar con el UUID, que es lo que interesa para esta implementación específica.

Una vez validado, se muestra el resultado al usuario, como puede identificarse en la figura N.6:

The screenshot shows a web page titled 'Validador de donación o cita de donación'. It asks if the user wants to validate a donation or appointment. Below is a QR code input field containing the identifier '9b1883a2-325d-48f6-8772-85b33cf6eb6'. A green button labeled 'VALIDAR' is present. A success message in a green bar says 'La donación es válida.' Below this, a section titled 'Detalles de la donación' displays the following information:

Identificador de donación	9b1883a2-325d-48f6-8772-85b33cf6eb6
Donante	Dña***
Identificador de donante (cédula o pasaporte)	*****3333
Fecha	Lunes, 15 de enero de 2024
Centro de donación	Fundación Dona Vida

Figura N.6: Resultado de validación de cita. Fuente: Elaborado por el autor.

Esta implementación funciona también con lectores de QR externos o físicos. En el caso del alcance de este Trabajo de Fin de Máster se ha probado la funcionalidad junto con un lector de códigos QR físico, lo que también podría ser útil al momento de aplicarlo con aplicaciones de verificación en sitio (ver Apéndice M).

Estos lectores funcionan como un teclado adicional en la máquina, por lo que el resultado de la lectura será un dispositivo de entrada. En la figura N.7 se aprecia el uso de este lector físico<sup>7</sup> de códigos QR, utilizando el código dispuesto a través de una pantalla dentro del sistema.

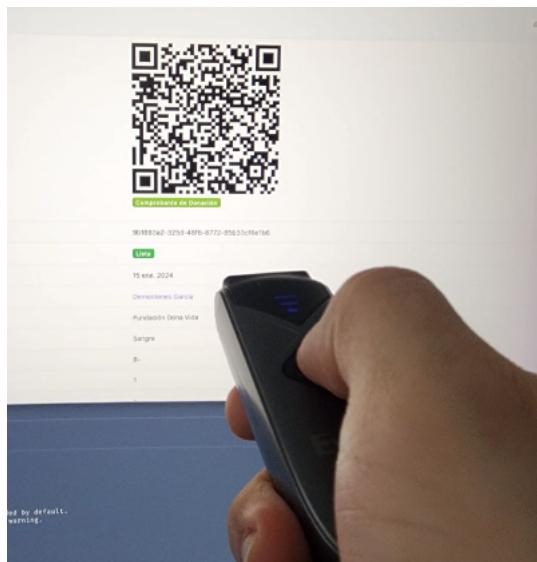


Figura N.7: Lector físico de códigos QR. Fuente: Elaborado por el autor.

<sup>7</sup>Modelo: EY015P, Marca: Eyoyo, YH1702.

# Ñ. Apéndice Ñ: Auditoría de seguridad

Para asegurar los requerimientos no funcionales, en especial el requerimiento no funcional de "Seguridad", se han realizado pruebas de penetración de la aplicación en su servidor de pruebas ([dev.yodonosangre.com](http://dev.yodonosangre.com)).

## Ñ.1. Resumen

Inicialmente, dichas pruebas arrojan una serie de vulnerabilidades identificadas, especialmente relacionadas a la ausencia de una implementación de CSP (*Content Security Policy*), exposición de datos sensitivos del servidor, entre otros que se listarán en este reporte.

### Alcance

Las vulnerabilidades que se expondrán en este reporte serán exclusivas de las pruebas de penetración cuyo objetivo es el servidor de pruebas `dev.yodonosangre.com` y únicamente las secciones que no requieren sesión. Se excluyen las aplicaciones móviles, zonas de usuario, panel de administración, capa de servicios y otras zonas que requieran un inicio de sesión.

De igual forma, se excluye cualquier otro dominio y subdominio que pueda requerirse para el funcionamiento completo de la solución.

## Ñ.2. Vulnerabilidades identificadas

En total se han identificado:

- 0 **vulnerabilidades críticas**.
- 1 **vulnerabilidad alta**.
- 3 **vulnerabilidades medias**.

- 3 vulnerabilidades bajas.

Esto da como resultado la identificación y corrección de un total de **7 vulnerabilidades** identificadas en total. **Como alcance de este TFM, se atacaron las vulnerabilidades altas y medias, dejando como trabajo futuro las vulnerabilidades bajas.**

## Vulnerabilidad 1: Ausencia de CSP

### Vulnerabilidad alta

El *Content Security Policy* (CSP) está diseñado para aumentar las defensas del sitio contra tipos específicos de ataques, como ataques de tipo *Cross-Site Scripting* (XSS) y ataques de *data injection* (inyección de datos).

Se ha encontrado que el sitio no implementaba el *header* de CSP dentro de la aplicación, lo que podría ocasionar que *scripts* no permitidos puedan ser ejecutados dentro de la aplicación.

### Mitigación

Para la mitigación se ha procedido a instalar el paquete Laravel CSP<sup>1</sup> y configurado para todas las rutas del *middleware* web, pero únicamente para las rutas públicas y de la aplicación web del usuario, exceptuando las del panel de administrador.

La razón para no realizarlo a nivel del Panel de Administrador es que el mismo es un código que implementa una librería externa de la cuál no se tiene acceso para modificar los encabezados (*headers*) para corregirlo.

La mitigación consiste en crear las directivas de los encabezados para los distintos tipos de datos, mostrado a continuación:

```
1 parent::configure();  
2  
3     $this->addDirective(Directive::BASE, config('app.url'));  
4     $this->addDirective(Directive::BLOCK_ALL_MIXED_CONTENT, Value::  
5     NO_VALUE);  
6     $this->addDirective(Directive::DEFAULT, Keyword::SELF);  
7     $this->addDirective(Directive::FORM_ACTION, Keyword::SELF);  
8     $this->addDirective(Directive::STYLE_ELEM, [
```

<sup>1</sup><https://github.com/spatie/laravel-csp>. Consultado el 30/01/2024.

```
8     Keyword::SELF,
9     Keyword::UNSAFE_INLINE,
10    'https://fonts.googleapis.com',
11    'https://fonts.bunny.net',
12  ]);
13 $this->addDirective(Directive::STYLE, [
14     Keyword::SELF,
15     Keyword::UNSAFE_INLINE,
16     'https://fonts.googleapis.com',
17     'https://fonts.bunny.net',
18  ]);
19 $this->addDirective(Directive::FONT, [
20     'https://fonts.gstatic.com',
21     'https://fonts.bunny.net',
22     'data:',
23     Keyword::SELF,
24  ]);
25 $this->addDirective(Directive::SCRIPT, [
26     Keyword::SELF,
27     Keyword::UNSAFE_EVAL,
28     Keyword::UNSAFE_INLINE,
29     'https://unpkg.com',
30     'http://127.0.0.1:5173',
31  ]);
32 $this->addDirective(Directive::SCRIPT_ELEM, [
33     Keyword::SELF,
34     Keyword::UNSAFE_INLINE,
35     'https://unpkg.com',
36     'http://127.0.0.1:5173',
37  ]);
38 $this->addDirective(Directive::IMG, [
39     Keyword::SELF,
40     'https://www.gstatic.com',
41     'data:',
42     'demogar.sirv.com',
43     '*.openstreetmap.org',
44     '*.googleusercontent.com',
45  ]);
46 $this->addDirective(Directive::CONNECT, [
47     Keyword::SELF,
48     Scheme::WS,
49     Scheme::WSS,
50  ]);
51 $this->addDirective(Directive::OBJECT, Keyword::NONE);
52 $this->addDirective(Directive::WORKER, Keyword::NONE);
53 $this->addDirective(Directive::FRAME_ANCESTORS, Keyword::SELF);
54 $this->addDirective(Directive::FRAME, Keyword::SELF);
55 $this->addDirective(Directive::CHILD, Keyword::NONE);
```

Esto configura los encabezados y permite recursos de dominios y direcciones específicas, para evitar que scripts maliciosos puedan ejecutarse en el sitio.

## Vulnerabilidad 2: Información del servidor expuesta

## Vulnerabilidad media

La información del tipo de servidor web y el sistema operativo se exponía anteriormente, como se puede evaluar en la figura N.1.

```
Pretty Raw Hex Hackvertor
Pretty Raw Hex Render Hackvertor
1 [HTTP / 1.1 200 OK]
2 Host: dev.yodanosangre.com
3 X-SRF-TOKEN=eyJpdiI6mtHTWCG2PbMtDjB59BTkGHOENDQ0EGPSIiIn2hHVl1joipZjNsM0QJMdL2JZJ4g5zY
4 eyjdI6l1kPkrV0CvpxlpoVzGz0uZbS0jMy3EmWkH0HZmUyMpwMsSdewH2bheepN0
5 FTQzJCY1LE6WNB0C2FnTHnczhvNgL0V3pg208WmalaFpUMxgrE5yHZEQ0p2ehbaXZJ7ZCMLGJ
6 MNQzZDFLMNg0UjTB7Iw1d6RfjnjoIn083D; dona_sangre_panama_session=
7 eyjdI6l1kPkrV0CvpxlpoVzGz0uZbS0jMy3EmWkH0HZmUyMpwMsSdewH2bheepN0
8 09PzL19TzJyQvkaB1YdmcTsDp2eVUHn94TkydEbh2zHFepeNsi1UWmfpk2d6uGvFpY
9 Vva3t3VtgPm11TtpSkpTj5B5leJ5ZGEHndnZQW0MgBHEKyzC1cplch2m9lLcGJ
10 tTM0M1oM1C1k2g5MNCMe0YTCZGMdMjY0MskM5mVntkkxH0Dm1dk1TzMeMcnTU0
11 HmQoQm4GQ5tV7N1wiLw1d6RfjnjoIn083D
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:12.0) Gecko/20100101 Firefox/12.0
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,* /;q=0.8
14 Accept-Language: en-US,en;q=0.5
15 Accept-Encoding: gzip, deflate
16 Upgrade-Insecure-Requests: 1
17 Sec-Fetch-Dest: document
18 Sec-Fetch-Mode: navigate
19 Sec-Fetch-Site: none
20 Sec-Fetch-User: ?1
21 Te: trailers
22 
```

Figura N.1: Vulnerabilidad 2 con Burp Suite. Fuente: Elaborado por el autor.

Esto puede ser perjudicial, ya que le da información al posible atacante sobre la versión actual de sistema operativo, por lo que podría explotar vulnerabilidades específicas del mismo.

## Mitigación

Para la mitigación, solamente se debe configurar el servidor web, en este caso Nginx. Luego, modificar el siguiente apartado dentro de la configuración global del mismo, al eliminar el comentario de la siguiente línea:

```
1 server_tokens off;
```

La diferencia, luego de la mitigación, se puede ver en la figura N.2.

```
~/projects/masw-tfm/donasangrepanama-be git:(main) ±6 (0.395s)
curl --head dev.yodonosangre.com
HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Wed, 31 Jan 2024 00:08:45 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive
Location: https://dev.yodonosangre.com/

~/projects/masw-tfm/donasangrepanama-be git:(main) ±6 (0.195s)
curl --head dev.yodonosangre.com
HTTP/1.1 301 Moved Permanently
Server: nginx/1.24.0 (Ubuntu)
Date: Wed, 31 Jan 2024 00:09:05 GMT
Content-Type: text/html
Content-Length: 178
Connection: keep-alive
Location: https://dev.yodonosangre.com/
```

Figura N.2: Vulnerabilidad 2 (mitigación). Fuente: Elaborado por el autor.

## Vulnerabilidad 3: Ausencia de X-Frame-Options (*clickjacking*)

### Vulnerabilidad media

La ausencia del encabezado (*header*) X-Frame-Options podría permitir a un posible atacante embeber dentro de su sitio a través de un *iframe* el sitio completamente. Esto podría permitir que el usuario interactúe con elementos ajenos al sitio web. Este tipo de ataque se conoce como *clickjacking*.

### Mitigación

La mitigación consiste en agregar el encabezado a nivel del servidor Nginx, tal como la mitigación anterior.

Para ello, se debe configurar el *server* dentro de la configuración particular en el *web server* de la siguiente manera:

```
1 add_header X-Frame-Options "SAMEORIGIN";
```

Esto obliga a que el único sitio en que pueda ser embebido sea dentro del mismo origen (el mismo dominio).

## Vulnerabilidad 4: Ausencia de *HTTP Strict Transport Security* (HSTS)

### Vulnerabilidad media

La ausencia de HSTS (*HTTP Strict Transport Security*), si bien se implementa un certificado de seguridad SSL para la comunicación segura, puede hacer que la aplicación sea susceptible ante ataques de tipo *Man-in-the-Middle* (MITM), al comprometer la habilidad de asegurar la conexión segura en todo momento.

### Mitigación

La mitigación requiere una nueva configuración a nivel del servidor Nginx, al agregar lo siguiente dentro de la configuración del servidor de este dominio:

```
1 add_header Strict-Transport-Security "max-age=31536000;  
includeSubDomains" always;
```

El `max-age` se configura a un año. Con esta configuración se obliga al cliente y al servidor a comunicarse sobre un canal protegido en todo momento.

## Ñ.3. Vulnerabilidades bajas

Las vulnerabilidades bajas (*Low*) encontradas, cuyas mitigaciones no son parte del alcance de este trabajo, son las siguientes:

- *Cookies sin HTTPONLY*. Estas *cookies*, al no tener la bandera de `HTTPONLY` podrían ser accedidas a través de otros *scripts*.
- *Cookies sin Secure Flag*. Estas *cookies*, al no tener la bandera de `Secure Flag` podría exponerlas para ataques de tipo MITM.
- Páginas en caché con posible contenido susceptible (requiere mayor investigación). Esta vulnerabilidad requiere mayor investigación puesto que la arquitectura actual del sistema no tiene ningún tipo de caché de información de la persona, pero al no tener las directivas `Cache Control: No Store` ni `Pragma: no cache` dentro de los encabezados de las respuestas, podría ser el caso que puedan guardarse datos sensativos.

# O. Apéndice O: Consideraciones para puesta en producción

Como parte del trabajo a futuro, es necesario la puesta en marcha en un entorno de producción de la solución descrita en el contexto de este Trabajo de Fin de Máster.

Para que se de el mismo, es de entera importancia considerar algunos puntos antes de evaluar el mismo, los cuales se describirán en los puntos a continuación.

## O.1. Costos asociados

Los costos asociados a la puesta en marcha de un entorno a producción serían los siguientes:

- **Infraestructura** (requerido): Si bien durante la elaboración de este TFM se realizó todo a través de un Servidor Virtual Privado (VPS), se haría posible verificar la implementación dentro de una nube pública, como *Amazon Web Services*.
- **Servidor de correo** (requerido): Se requeriría un servidor de correo (SMTP) para el envío de correos transaccionales, como lo serían el correo de bienvenida, de notificación de una nueva cita, entre otros.
- **Cuentas de tiendas** (requerido): Es necesario contar con una cuenta de desarrollador de Apple (*Apple Development Program*) y una para la tienda de *Google Play Store*.
- **Google API** (requerido): Los mapas de Google, si bien tienen una capa gratuita, pueden acarrear costos y los mismos deben ser considerados.
- **Content Delivery Network** (ideal): Para el despliegue de contenidos a través de un CDN (como imágenes), sería ideal contar con un servicio de tipo SaaS.
- **Otros SaaS** (ideal): Sería adecuado contar con otros servicios para la captura de errores, administración de tareas y el proyecto, entre otros.

# Bibliografía

- 650 Industries, Inc. (2023). Expo [Consultado el 21/10/2023]. <https://expo.dev/>
- Amazon. (2022). What is Redis? - Redis Benefits and Use Cases - AWS [Consultado el 27/12/2023]. <https://aws.amazon.com/redis/>
- American Red Cross. (2020). Blood Donor App - American Red Cross [Consultado el 14/10/2023]. <https://www.redcrossblood.org/blood-donor-app.html>
- Anand, R. V., & Dinakaran, M. (2017). Handling stakeholder conflict by agile requirement prioritization using Apriori technique. *Computers and Electrical Engineering*, 61, 126-136. <https://doi.org/https://doi.org/10.1016/j.compeleceng.2017.06.022>
- Atlassian. (2002). Jira: Unlock your team's best work with Jira Software [Consultado el 25/10/2023]. <https://jira.atlassian.com/>
- Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70-77.
- Beck, K. (2013). Extreme programming. <http://www.extremeprogramming.org>
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. (2001a). Manifesto for agile software development. <https://agilemanifesto.org>
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. (2001b). Principles for agile development. <https://agilemanifesto.org/principles.html>
- Calixto, R. N. L., González, L. Á. V., Díaz, D. E. B., & Guzmán, R. V. (2019). React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma. *Revista Digital Universitaria*, 20(5).
- Callahan, B. (2021, febrero). The Never-Ending Job of Selling Design Systems A List Apart [Consultado el 07/12/2023]. <https://alistapart.com/article/selling-design-systems/>
- Casas, S., Cruz, D., Vidal, G., & Constanzo, M. (2021). Uses and applications of the OpenAPI/Swagger specification: a systematic mapping of the literature. *2021 40th International Conference of the Chilean Computer Science Society (SCCC)*, 1-8.
- Cloudflare. (2020). Moving from reCAPTCHA to hCaptcha [Consultado el 27/12/2023]. <https://blog.cloudflare.com/moving-from-recaptcha-to-hcaptcha>
- Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.
- Delía, L. N., Galdamez, N., Thomas, P. J., & Pesado, P. M. (2013). Un análisis experimental de tipo de aplicaciones para dispositivos móviles. *XVIII Congreso Argentino de Ciencias de la Computación*.

- Doshi, M., & Virparia, P. (2023). Agile Development Methodology for Software Re-engineering. En *Advances in Information Communication Technology and Computing: Proceedings of AICTC 2022* (pp. 401-409). Springer.
- Editorial Panamá América. (2022). CSS alerta sobre déficit en bancos de sangre | Panamá América [Consultado el 17/10/2023]. <https://www.panamaamerica.com.pa/sociedad/css-alerta-sobre-deficit-en-bancos-de-sangre-1216116>
- ELGHERIANI, N. S., & AHMED, N. A. S. (2022). Microservices vs. monolithic architectures [the differential structure between two architectures]. *MINAR International journal of applied sciences and technology*, 4(3), 500-514. <https://doi.org/10.47832/2717-8234.12.47>
- Frederik Fowler. (2018). *The Product Backlog*. [https://doi.org/10.1007/978-1-4842-4164-6\\_9](https://doi.org/10.1007/978-1-4842-4164-6_9)
- Gaceta Oficial, República de Panamá. (1986). Ley 17 de 31 de julio de 1986, Por la cual se reglamentan los Bancos de Sangre y las transfusiones sanguíneas y se dictan otras medidas [Consultado el 15/10/2023]. <https://docs.panama.justia.com/federales/leyes/17-de-1986-aug-8-1986.pdf>
- Gaceta Oficial, República de Panamá. (2019). Ley 81 de 2019, Sobre Protección de Datos Personales [Consultado el 27/12/2023]. [https://www.asamblea.gob.pa/APPS/LEGISPAN/PDF\\_NORMAS/2010/2019/2019\\_645\\_300.pdf](https://www.asamblea.gob.pa/APPS/LEGISPAN/PDF_NORMAS/2010/2019/2019_645_300.pdf)
- Gaceta Oficial, República de Panamá. (2020). Ley 164 de 11 de septiembre de 2020, QUE INCENTIVA LA DONACION DE SANGRE EN EL TERRITORIO NACIONAL [Consultado el 17/10/2023]. [https://www.asamblea.gob.pa/APPS/SEG\\_LEGIS/PDF\\_SEG/PDF\\_SEG\\_2020/PDF\\_SEG\\_2020\\_P\\_267.pdf](https://www.asamblea.gob.pa/APPS/SEG_LEGIS/PDF_SEG/PDF_SEG_2020/PDF_SEG_2020_P_267.pdf)
- Gitlab, Inc. (2014). The DevSecOps Platform | GitLab [Consultado el 21/10/2023]. <https://about.gitlab.com/>
- Hooda, S., Sood, V. M., Singh, Y., Dalal, S., & Sood, M. (2023). *Agile Software Development: Trends, Challenges and Applications*. John Wiley & Sons.
- IEEE. (1990). IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990*.
- Inertia.js. (2019). Inertia.js - The Modern Monolith [Consultado el 21/10/2023]. <https://inertiajs.com/>
- ISTQB. (2023). Standard Glossary of Terms used in Software Testing. *Standard Glossary of Terms used in Software Testing, v4*.
- Jing, Y., Wang, X., Xiao, X., & Zhang, G. (2006). NIS04-5: Defending Against Meek DDoS Attacks By IP Traceback-based Rate Limiting. *IEEE Globecom 2006*, 1-5. <https://doi.org/10.1109/GLOCOM.2006.283>
- Krocz, K., Kizun, O., & Skublewska-Paszkowska, M. (2020). Performance analysis of relational databases MySQL, PostgreSQL, MariaDB and H2. *Journal of Computer Sciences Institute*, 14, 1-7. <https://doi.org/10.35784/jcsi.1565>
- Kuroishi, P. H., Maldonado, J. C., & Vincenzi, A. M. R. (2024). Towards the definition of a research agenda on mobile application testing based on a tertiary study. *Information and Software Technology*, 167, 107363. <https://doi.org/10.1016/j.infsof.2023.107363>

- Lam, P., Dietrich, J., & Pearce, D. J. (2020). Putting the semantics into semantic versioning. *Proceedings of the 2020 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, 157-179.
- Laravel, LLC. (2011). The PHP framework for Web Artisans [Consultado el 16/10/2023]. <https://laravel.com/>
- Laravel, LLC. (2023). Frontend - Laravel 10.x - Using Vue/React [Consultado el 10/17/2023]. <https://laravel.com/docs/10.x/frontend>
- Linus Torvalds. (2007). Git: free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. [Consultado el 21/10/2023]. <https://git-scm.com/>
- Macdonald, C., & Putnam, C. (2023). Design Systems: A scalable model for teaching design systems for UX. <https://doi.org/10.1145/3587399.3587403>
- Maharjan, R., Chy, M. S. H., Arju, M. A., & Cerny, T. (2023). Benchmarking Message Queues. *Telecom*, 4(2), 298-312. <https://doi.org/10.3390/telecom4020018>
- Meta Platforms, Inc. (2023). React Native: an open-source UI software framework for building multiplatform mobile apps [Consultado el 14/10/2023]. <https://reactnative.dev/>
- Open Source Initiative. (2009). The PostgreSQL Licence Open Source Initiative [Consultado el 18/10/2023]. [https://opensource.org/license/postgresql/](https://opensource.org/license/postgresql)
- OpenStreetMap. (2004). OpenStreetMap [Consultado el 05/12/2023]. <https://www.openstreetmap.org/about>
- Ordóñez, M. P. Z., Ríos, J. R. M., & Castillo, F. F. R. (2017). *Administración de Bases de datos con PostgreSQL* (Vol. 19). 3Ciencias.
- OWASP. (2001). OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation [Consultado el 29/01/2024]. <https://owasp.org/>
- ProductPlan. (2013). What is a Product Roadmap? The Ultimate Guide and Resources [Consultado el 25/10/2023]. <https://www.productplan.com/learn/what-is-a-product-roadmap/>
- Schwaber, K. (1997). Scrum development process. *Business Object Design and Implementation: OOPSLA95 Workshop Proceedings 16 October 1995, Austin, Texas*, 117-134.
- Schwalbe, K. (2009). *Introduction to project management*. Course Technology Cengage Learning Boston.
- Seignard, X. (2023, octubre). Our journey from React Native to Expo for mobile app development at Alan | by Xavier Seignard | Alan Product and Technical Blog | Medium [Consultado el 14/10/2024]. <https://medium.com/ala/our-journey-from-react-native-to-expo-for-mobile-app-development-at-alan-%EF%B8%8F-3b1569e8ab7c>
- Sosa-Tzec, O. (2021). Delight in the User Experience: Form and Place. *Congress of the International Association of Societies of Design Research*, 111-120.
- Stack Overflow. (2023a). Stack Overflow Developer Survey 2023 [Consultado el 18/10/2023]. <https://survey.stackoverflow.co/2023/>

- Stack Overflow. (2023b). Stack Overflow Trends - React, Angular, Svelte (2008-2023) [Consultado el 18/10/2023]. <https://insights.stackoverflow.com/trends>
- Stallman, R. (2020). La definición de software libre. *Comuniars. Revista de Imagen, Artes y Educación Crítica y Social*, 3, 151-154.
- Surguy, M. (2013). History of Laravel PHP framework, Eloquence emerging [Consultado el 15/10/2023]. <https://maxoffsky.com/code-blo/history-of-laravel-php-framework-eloquence-emerging/>
- The PostgreSQL Global Development Group. (1996). PostgreSQL: License [Consultado el 10/17/2023]. <https://www.postgresql.org/about/licence/>
- The PostgreSQL Global Development Group. (2023). PostgreSQL: The World's Most Advanced Open Source Relational Database [Consultado el 14/10/2023]. <https://www.postgresql.org/>
- Vanderlei, I., Araujo, J., Rocha, R., Silva, G., Pacheco, F., & Dantas, J. (2021). Analysis of Laravel Framework Security Techniques Against Web Application Attacks. *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-7. <https://doi.org/10.23919/CISTI52073.2021.9476475>
- Vazifeh-Noshafagh, S., Hajipour, V., Jalali, S., Di Caprio, D., & Santos-Arteaga, F. J. (2022). Maturing the Scrum Framework for Software Projects Portfolio Management: A Case Study-Oriented Methodology. *IEEE Access*, 10, 123283-123300.
- W3C. (2023). Web Content Accessibility Guidelines (WCAG) 2.1 [Consultado el 28/01/2024]. <https://www.w3.org/TR/WCAG21/>
- Wake, B. (2003). INVEST in Good Stories, and SMART Tasks - XP123 [Consultado el 25/10/2023]. <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
- Wegner, P. (1996). Interoperability. *ACM Computing Surveys (CSUR)*, 28(1), 285-287.
- Wei, L., Liu, Y., & Cheung, S.-C. (2016). Taming Android Fragmentation: Characterizing and Detecting Compatibility Issues for Android Apps. *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, 226-237. <https://doi.org/10.1145/2970276.2970312>