



Requerimientos para el desarrollo de Software

Plataforma BOOM

PREFACIO

Este documento describe los requerimientos para la plataforma BOOM haciendo uso de una solución que comprende de un sitio web, una aplicación móvil, una backend de administración, una api y una base de datos. El objetivo principal es que los usuarios puedan consultar todo tipos de eventos según sus preferencias.

Alcance

Este documento de requerimientos es la base del desarrollo de la plataforma, y será la directriz en cuanto a los requerimientos funcionales, la arquitectura de software, diseño de la base de datos, definición de los endpoint de la api.

HISTORIA DEL DOCUMENTO

Fecha	Versión	Comentarios	Autor
05 abril de 2022	0.1	Versión inicial	Quipro

TABLA DE CONTENIDOS

Introducción	4
Objetivo General del Proyecto	4
Estándares	4
Documentación	4
Requerimientos Software y Desarrollo	5
Requerimientos Funcionales	8
Requerimientos de Testing y de Calidad	15
Matriz de Acciones vs Actores	17
Matriz Requerimientos Funcionales vs. Requerimientos de Testing	20
Requerimientos de Ambiente	23
Requerimientos de Ambiente de Desarrollo	23
Hardware de Desarrollo	23
Desarrollo de Software	23
Recursos Humanos para el Desarrollo	23
Integraciones	24
Requerimientos de Ambiente de Testing	25
Hardware de Testing	25
Software de Testing	25
Testing Automatizado	25
Restricciones	27
Restricciones Hardware	27
Restricciones Software	27
Restricciones de Interfaz	27
Arquitectura del Software	28
Modelo Arquitectónico	28
Seguridad del Software	29
Escalabilidad del Software	30
Requerimientos de Desarrollo	31
Requerimientos Participación Cliente	31
Requerimientos de Comunicación	31
Requerimientos de Infraestructura	31

Requerimientos Post Desarrollo	32
Requerimientos de Entrenamiento	32
Requerimientos de Mantención	32
Diseño de la Base de Datos	33
Estructura	33
Diccionario de Datos	35
API	42
End Points	44
Documentación de Soporte	52

1 Introducción

1.1 Objetivo General del Proyecto

Desarrollar una plataforma para la consulta de todo tipo de eventos de entretenimiento enfocado en las preferencias del usuario. Además de recolección de datos para su explotación para obtener conocimiento de valor.

1.2 Estándares

El sistema será implementado y gestionado a través de herramientas que faciliten el desarrollo colaborativo en línea por un grupo de programadores. Se usará un gestor de tareas que facilite el control y evolución de la solución propuesta. Así mismo, se contará con un sistema de control de versiones (git) donde estará disponible el repositorio de archivos conforme la solución. Se emplearán metodologías de desarrollo ágil para acortar el tiempo de obtención de la solución final y el marco de trabajo para la gestión de entregables parciales será SCRUM.

1.3 Documentación

- Existentes:
 - Versión preliminar del Brief del proyecto.
- Entregables como parte de este documento:
 - Descripción de los requerimientos detectados.
 - Propuesta de solución tecnológica.
 - Diagrama de la estructura de la base de datos.
 - Diagrama de la arquitectura de software.
 - Definición de Api.

2 Requerimientos Software y Desarrollo

Para dar solución al objetivo planteado, se propone el diseño e implementación de una plataforma que cubra cada una de las necesidades identificadas. La plataforma propuesta es un sistema que cuente con los siguientes elementos:

- Website.
- Aplicación móvil para iOS y Android.
- Backend.
- Api.
- Base de datos.

Modelo del negocio

En el negocio se han identificado como actores principales a boomers (usuarios interesados en explorar los eventos), editores y administradores. Cada uno debe de poder hacer las siguientes funciones generales:

El usuario tiene como función principal el consultar eventos, además de poder comprar boletos en alguna de las versiones de la plataforma.

El editor tiene la responsabilidad de cargar la información que se mostrará en el sitio web y en las aplicaciones móviles.

El administrador tiene la facultad de cargar y/o autorizar la publicación del contenido que cargue el editor, poder gestionar las cuentas de los usuarios y la posibilidad de poder hacer reporte.

Análisis de requisitos del sistema

Además de los roles previamente identificados en la modelación del sistema, se propone la inclusión del rol de Super Admin que será el superusuario de la herramienta informática obtenida con la solución propuesta.

En la siguiente tabla se enumeran los roles identificados así como las funciones, que serán detalladas más adelante, que debe desempeñar cada uno en la plataforma:

Rol	Funciones
Boomers	<ol style="list-style-type: none">1. Podrá consultar eventos.2. Podrá agendar sus eventos.3. Poder elegir marcar eventos con ciertas etiquetas (favoritos, asistiré, etc.).4. Compartir eventos a sus contactos.5. Comprar boletos de eventos.
Editor	<ol style="list-style-type: none">1. Cargar y editar información de los eventos que aparecerán

Rol	Funciones
	<p>en la web y app.</p> <ol style="list-style-type: none"> 2. Cargar y editar información de las promociones que aparecerán en la web y app. 3. Cargar y editar información para las push notifications de la aplicación móvil. 4. Cargar de formularios para encuestas que aparecerán en la aplicación móvil.
Administrador	<ol style="list-style-type: none"> 1. Cargar y editar información de los eventos que aparecerán en la web y app. 2. Cargar y editar información de las promociones en la web y app. 3. Cargar y editar información para las push notifications en la web y app. 4. Administrar usuarios y editores. 5. Hacer reportes de la información en la base de datos 6. Hacer reportes de las encuestas. 7. Exportar información de módulos específicos. 8. Gestión de cuentas de usuarios, editores y administradores.

Módulos de la plataforma que permitirán lograr las funcionalidades detectadas.

Tipo de Usuario	Descripción	Tecnologías/Módulo
Usuario	Preview de eventos	Sitio Web
Usuario	Preview eventos y consulta de eventos.	App
Usuario	Agendar eventos	App/Google Calendar y Outlook
Usuario	Marcar eventos con ciertas etiquetas.	App
Usuario	Compartir eventos a contactos.	App/ Whats app, Facebook, Twitter
Usuario	Compra de boletos para los eventos.	App y/o Sitios de compra

Tipo de Usuario	Descripción	Tecnologías/Módulo
Editor	Carga y edición de información de los eventos que aparecerán en la web y app.	Backend
Editor	Carga y edición de información de las promociones en la web y app.	Backend
Editor	Carga y edición de información para las push notifications en la web y app.	Backend
Editor	Carga de encuestas para mostrar en la app.	Backend
Administrador	Carga y edición de la información de los eventos que aparecerán en la web y app.	Backend
Administrador	Carga y edición información de las promociones en la web y app.	Backend
Administrador	Carga y edición información para las push notifications que mostrará el app.	Backend
Administrador	Administrar cuentas del rol usuarios y editores.	Backend
Administrador	Hacer reportes de la información en la base de datos.	Backend
Administrador	Hacer reportes de las encuestas.	Backend
Administrador	Exportar información de módulos específicos.	Backend

2.1 Requerimientos Funcionales

A continuación el listado de requerimientos funcionales que debe tener el sistema. Esta lista refleja las posibilidades principales que debe ofrecer el producto final.

Requerimientos funcionales específicos para el rol usuario boomers.

Requerimiento (RF)	Descripción
1	La plataforma debe permitir registrarse con al menos correo electrónico y contraseña, por medio de la aplicación móvil y el sitio web. Además de poder recuperar contraseña.
2	La plataforma debe permitir registrarse en la aplicación móvil y en el sitio web con facebook login.
3	La plataforma debe permitir registrarse en la aplicación móvil y en el sitio web con Google login.
4	La plataforma debe permitir registrarse en la aplicación móvil y en el sitio web con Spotify.
5	La plataforma debe permitir registrarse en la aplicación móvil y en el sitio web con Apple login.
6	El usuario debe poder configurar en la aplicación móvil, la privacidad de su información para que nadie pueda ver mi información.
7	El usuario debe poder configurar en la aplicación móvil, la privacidad de su información a sólo mis amigos puede ver mi información.
8	El usuario debe poder configurar en la aplicación móvil, la privacidad de su información para que todos puedan ver mi información.
9	La aplicación móvil debe de permitir cambiar la contraseña de acceso
10	El usuario debe poder ver su perfil en la aplicación móvil.
11	El usuario debe poder editar su perfil en la aplicación móvil.

Requerimiento (RF)	Descripción
12	La plataforma debe permitir en el sitio web, elegir la ciudad donde quiere ver los eventos.
13	La plataforma debe permitir en la aplicación móvil, elegir la ciudad donde quiere ver los eventos.
14	La plataforma debe permitir en el sitio web, ver las categorías de los eventos.
15	La plataforma debe permitir, en la aplicación móvil, ver las categorías de los eventos.
16	Después de ser instalada la aplicación móvil, el usuario debe de poder ver una introducción de cómo funciona el mapa en el app.
17	Después de ser instalada la aplicación móvil, el usuario debe de poder ver una introducción de cómo funciona el feed en el app.
18	La plataforma debe permitir en el sitio web, mostrar los eventos de forma cronológica.
19	La plataforma debe permitir en la aplicación móvil, mostrar los eventos de forma cronológica.
20	La plataforma debe permitir en la aplicación móvil, con un solo touch consultar preview de eventos.
21	La plataforma no debe permitir ver información completa de los eventos, sólo mostrar preview cuando el usuario usa el sitio web.
22	La plataforma sólo debe permitir ver información completa de los eventos cuando el usuario está logueado en la aplicación móvil.
23	El usuario debe de poder agendar eventos en google calendar sólo si está logueado en en la aplicación móvil.
24	El usuario debe de poder agendar eventos en Outlook calendar sólo si está logueado en la aplicación móvil.
25	El usuario debe de poder agendar eventos en iCalendar sólo si está logueado en la aplicación móvil.
26	La plataforma debe permitir mostrar en un mapa los eventos cercanos al usuario sólo si está logueado en en la aplicación móvil.

Requerimiento (RF)	Descripción
27	El usuario debe de poder ver en el mapa los eventos con colores dependiendo la cercanía sólo si está logueado en en la aplicación móvil.
28	El usuario debe de poder marcar un evento como favorito sólo si está logueado en la aplicación móvil..
29	El usuario debe de poder marcar un evento al que asistirá sólo si está logueado en la aplicación móvil.
30	El usuario debe de poder marcar un evento al que asistió sólo si está logueado en la aplicación móvil.
31	El usuario debe de poder editar (cambiar/eliminar) la marca que puso en algún evento sólo si está logueado en la aplicación móvil.
32	La plataforma debe permitir mostrar en un mapa los eventos que etiquetó o marcó el usuario, sólo si está logueado en la aplicación móvil..
33	El usuario debe de recibir una notificación (con información del evento) cuando etiqueta un evento al cual va a asistir.
34	El usuario (sólo si está logueado en la aplicación móvil) debe poder invitar a otros usuarios de la app a asistir a algún evento.
35	El usuario (sólo si está logueado en la aplicación móvil) debe de poder invitar a otros usuarios de facebook a asistir a algún evento.
36	El usuario (sólo si está logueado en la aplicación móvil) debe de poder compartir un evento a otros usuarios de facebook.
37	El usuario (sólo si está logueado en la aplicación móvil) debe de poder compartir un evento a otros usuarios de whatsapp.
38	El usuario (sólo si está logueado en la aplicación móvil) debe de poder compartir un evento a otros usuarios de Twitter.
39	La plataforma, en la aplicación móvil, debe permitir que el usuario reciba notificaciones sobre algún cambio en un evento programado.
40	La plataforma, en la aplicación móvil, debe permitir que el usuario reciba notificaciones sobre eventos que coincidan con sus búsquedas previas.

Requerimiento (RF)	Descripción
41	La plataforma, en la aplicación móvil, debe permitir que el usuario reciba notificaciones promociones dependiendo de su ubicación en un evento.
42	La plataforma, en la aplicación móvil, debe permitir ver información de usuarios que se encuentran en el mismo evento.
43	La plataforma debe de permitir acumular puntos al asistir a cada evento.
44	La plataforma debe de permitir acumular puntos al contestar formularios.
45	La plataforma debe de permitir entregar distintivo a los usuarios por asistir a cada evento.
46	La plataforma debe de permitir entregar distintivos a los usuarios por contestar formularios.
47	La plataforma debe permitir a los usuarios la compra de boletos. (etapa 2)
48	El sitio web debe de permitir a los usuarios llenar formularios de contacto
49	El usuario debe de poder hacer logout

Requerimientos funcionales específicos para el rol usuario editor.

Requerimiento	Descripción
50	La plataforma debe permitir hacer login en el módulo de administración. Además de poder recuperar contraseña.
51	La plataforma debe de permitir crear categorías de eventos.
52	La plataforma debe de permitir crear tipo de eventos.
53	La plataforma debe de permitir crear categorías de eventos.
54	La plataforma debe de permitir crear campos en las descripciones de los eventos.
55	La plataforma debe de permitir eliminar campos en las descripciones de los eventos.

Requerimiento	Descripción
56	El usuarios debe de poder ver la lista de eventos
57	El usuario debe de poder dar de alta un evento, el cual debe ser aprobado por el administrador.
58	El usuario debe de poder editar un evento, el cual debe ser aprobado por el administrador.
59	El usuario debe de poder eliminar un evento, lo cual debe ser aprobado por el administrador.
60	La plataforma debe de permitir subir la información correspondiente al preview de cada evento.
61	La plataforma debe de permitir subir las coordenadas de cada evento.
62	La plataforma debe de permitir seleccionar los colores dependiendo de la cercanía del usuario a una locación de un evento.
63	El usuario debe de poder ver en una lista las cuentas de los usuarios de la app.
64	El usuario debe de gestionar las cuentas de los usuarios de la app (excepto contraseñas).
65	El usuario debe de por generar reportes con la información contenida de la base de datos (Edad, género, categorías, eventos, asistencia, invitados y uso de la app)
66	El usuario debe de generar reportes haciendo uso o integrando los datos obtenidos por analytics en el sitio web.
67	El usuario debe de poder cargar la información para las notificaciones (push notifications).
68	El usuario debe de poder seleccionar el target para mandar las notificaciones (push notifications).
69	El usuario debe de poder cargar las promociones que se mostrarán en la aplicación móvil.
70	El usuario debe de poder cargar las ciudades para categorizar los eventos por la ciudad.

Requerimientos funcionales específicos para el rol administrador.

Requerimiento	Descripción
50	La plataforma debe permitir hacer login en el módulo de administración. Además de poder recuperar contraseña.
51	La plataforma debe de permitir crear categorías de eventos.
52	La plataforma debe de permitir crear tipo de eventos.
53	La plataforma debe de permitir crear subcategorías de eventos.
54	La plataforma debe de permitir crear campos en las descripciones de los eventos.
55	La plataforma debe de permitir eliminar campos en las descripciones de los eventos.
56	El usuarios debe de poder ver la lista de eventos
57	El usuario debe de poder dar de alta un evento.
58	El usuario debe de poder editar un evento.
59	El usuario debe de poder eliminar un evento.
60	La plataforma debe de permitir subir la información correspondiente al preview de cada evento.
61	La plataforma debe de permitir subir las coordenadas de cada evento.
62	La plataforma debe de permitir seleccionar los colores dependiendo de la cercanía del usuario a una locación de un evento.
63	El usuario debe de poder ver en una lista las cuentas de los usuarios de la app.
64	El usuario debe de gestionar las cuentas de los usuarios de la app (excepto contraseñas).
65	El usuario debe de generar reportes con la información contenida de la base de datos (Edad, género, categorías, eventos, asistencia, invitados y uso de la app)
66	El usuario debe de generar reportes haciendo uso o integrando los datos obtenidos por analytics en el sitio web.
67	El usuario debe de poder cargar la información para las notificaciones

Requerimiento	Descripción
	(push notifications).
68	El usuario debe de poder seleccionar el target para mandar las notificaciones (push notifications).
69	El usuario debe de poder cargar las promociones que se mostrarán en la aplicación móvil.
70	El usuario debe de poder cargar las ciudades para categorizar los eventos por la ciudad.
71	El usuario debe de poder gestionar las cuentas de usuarios de toda la plataforma.

2.2 Requerimientos de Testing y de Calidad

Para garantizar la integridad de la solución obtenida con el desarrollo de la aplicación se proponen la implementación de una serie de pruebas (tests) que se enumeran a continuación.

ST	Descripción
1	Un usuario puede acceder al sistema sólo con su identificador de usuario y su contraseña
2	Un usuario puede registrarse en la aplicación móvil sólo con los datos completos.
3	Un usuario solo puede cambiar datos relacionados con su cuenta.
4	Ningún usuario que no tenga los permisos adecuados podrá ingresar al sistema.
5	Ningún usuario que no tenga los privilegios adecuados puede modificar información para mostrar en las aplicaciones.
6	Un usuario solo puede acceder a su perfil.
7	Un usuario solo puede editar su perfil.
8	Los cambios realizados por el usuario deben de actualizar la información de manera dinámica.

ST	Descripción
9	Ningún usuario puede acceder a la plataforma sin credenciales
10	Ningún usuario puede acceder a la plataforma con credenciales no válidas.
11	Ningún usuario se puede cambiar la privacidad de la información de otro usuario
12	Un usuario sólo debe de ver la información de su perfil en la aplicación móvil.
13	Un usuario no puede ver eventos que no pertenezcan a la categoría o tipo indicada.
14	Un usuario no puede ver eventos que no pertenezcan a la ciudad indicada.
15	Un usuario no puede ver la información completa del evento en el sitio web
16	Un usuario no puede ver la información completa del evento en la aplicación móvil a menos de que esté logueado.
17	La introducción de cómo funciona el mapa/feed sólo debe de poderse ver una vez instalada la aplicación móvil
18	No deben de mostrarse eventos en desorden, sólo por fecha cronológica
19	Un usuario no logueado sólo puede ver previews de eventos.
20	Un evento no puede ser calendarizado en varios días y horas
21	El usuario debe de poder editar y etiquetar parámetros de sus eventos asociados.
22	La aplicación móvil debe de mostrar correctamente los eventos en el mapa
23	El usuario no puede realizar acciones, más que preview de eventos, si no está logueado.
24	La plataforma debe enviar push notifications al target seleccionado.
25	El usuario solo puede compartir eventos que tiene relacionados con su cuenta.

ST	Descripción
26	La aplicación debe mostrar eventos según las preferencias del usuario.
27	La plataforma debe asignar puntos o distintivos sólo al usuario que le corresponda.
28	La plataforma debe de permitir la compra de boletos para eventos

2.3 Matriz de Acciones vs Actores

Acciones	Actores				Plataformas								Otros			
	Usr	Ed	Adm	Super User	Web	App	Back	Google Calendar	Outlook	Whatsapp	FB	Twitter	Pasarela	Push N.	Report	Filtro eventos
Instalación de app	x					x										
Introducción de uso de mapas	x					x										
Introducción del feed en app	x					x										
Registro y admon usuario	x															
Registro de usuario (correo, google, etc.)	x				x	x										
Administración de usuario	x L					x										
Configuración de variantes de app	x L					x										
Definir nivel de privacidad	x L					x										
Elegir ciudad	x L				x	x										
Llenar formularios de contacto	x L				x											
Login	x				x	x										
Logout	x L				x	x										
Eventos						x										
Consulta de eventos. (Por colores)	x				x	x										x
Ver preview completa de evento	x L				x											
Ver preview parcial de evento	x				x											
Mostrar en mapa eventos cercanos	x L					x										
Podrá agendar sus eventos.	x L					x	x	x	x							
Elegir categorías de eventos	x				x	x										
Poder elegir marcar eventos a los que asistirá.	x L					x										
Poder elegir marcar eventos a los que asistió.	x L					x										
Poder editar marcas de eventos	x L					x										
Poder ver marcas de eventos en el mapa	x L					x										

Aprobar evento dado de alta, de baja o edición de evento			x				x										
Subir información del preview de cada evento		x	x				x										
Subir coordenadas del evento		x	x				x										
Seleccionar simbología de colores en función de cercanía de evento		x	x				x										
Ver lista de cuentas de usuarios		x	x				x										
Gestionar cuentas de usuarios (excepto claves de acceso)		x	x				x										
Generar reportes de usuarios filtrados por parámetros de la base de datos		x	x				x										
Generar reportes a partir de los datos de analytics		x	x				x										
Cargar información para push notifications		x	x				x							x			
Seleccionar el target de push notifications		x	x				x							x			
Cargar promociones		x	x				x										
Cargar ciudades		x	x				x										
introducción del feed en app		x		x			x										
Administrar editores.			x	x			x										
Hacer reportes de las encuestas.			x	x			x										
Gestión de cuentas de usuarios, editores y administradores.				x			x										

2.4 Matriz Requerimientos Funcionales vs. Requerimientos de Testing

En la siguiente matriz se muestra de qué manera se pueden vincular las pruebas identificadas con cada uno de los requerimientos funcionales identificados en el levantamiento de especificaciones.

S T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
R F																												
1	x																											
2	x																											
3	x																											
4	x																											
5		x	x	x	x	x	x	x																				
6		x	x	x	x	x	x	x																				
7		x	x	x	x	x	x	x																				
8		x						x																				
9			x		x						x																	
10			x		x		x				x	x																
11													x	x	x			x				x				x		
12													x	x	x			x				x				x		
13												x		x				x				x						
14															x								x					
15																x												
16																x												
17														x			x	x					x					
18														x			x	x					x					
19																		x	x				x					
20															x			x	x				x					
21									x	x					x			x	x				x					
22									x	x											x	x		x				
23									x	x											x	x		x				

Requerimientos de Ambiente

2.5 Requerimientos de Ambiente de Desarrollo

A Continuación se hace una propuesta sobre el hardware, software y recursos humanos para el desarrollo del proyecto. La evolución del proyecto podría motivar a hacer cambios en algunos de los puntos que se presentan, es por eso que el desarrollo no queda limitado a lo que se propone, pero si se recomienda tomar como referencia o base lo que se describe en esta sección.

2.5.1 *Hardware de Desarrollo*

Para el desarrollo del sistema se usarán computadoras personales actuales, no hay un requerimiento particular. Simplemente que sea capaz de correr un servidor web y un sistema gestor de bases de datos relacional.

2.5.2 *Desarrollo de Software*

Se propone que la plataforma sea desarrollada con los siguientes frameworks y/o tecnologías:

- React Javascript con Next.js o con Blades de Laravel, para Sitio Web.
- Ionic o Flutter para el desarrollo de las aplicaciones móviles.
- Django REST o Laravel para el backend.
- Api Rest para el API.
- MySQL o PostgreSQL para la base de datos.

2.5.3 *Recursos Humanos para el Desarrollo*

Para este proyecto se recomienda tener un equipo de desarrollo con las siguientes tareas generales:

- Project Manager.
 - Gestionar el proyecto para lograr los objetivos y alcances en los tiempos y forma establecidos.
- Desarrollador Backend
 - Programar la base de datos.
 - Programar la parte lógica del negocio.
 - Programar el Api.
- Desarrollador Frontend
 - Desarrollar del Sitio Web
 - Desarrollar la aplicación para dispositivos móviles con sistema operativo Android.

- Desarrollar el app para dispositivos móviles con sistema operativo iOS
- Tester
 - Encargado de las pruebas para QA (Quality Assurance).
- Technical writers
 - Encargado de hacer la documentación necesaria para la plataforma.

2.5.4 Integraciones

Aparte de lo mencionado anteriormente se requieren ciertas tecnologías para lograr llevar a acabo los siguientes procesos:

Proceso	Tecnologías sugeridas
Compra de boletos .	Por definir.
Push notifications.	<ul style="list-style-type: none"> ● Firebase. ● Frizbit. ● SendPulse.
Reporteo.	<ul style="list-style-type: none"> ● Google Data. ● Metabase. ● Tableau.
Definir eventos para mostrarle al usuario con base a sus preferencias.	<ul style="list-style-type: none"> ● Algoritmo básico basado en reglas. ● Naive Bayes. ● Gaussian Naive Bayes. ● Multinomial Naive Bayes. ● Bayesian Network.
Marketing	<ul style="list-style-type: none"> ● Hubspot

2.6 Requerimientos de Ambiente de Testing

2.6.1 *Hardware de Testing*

Será necesario el empleo de computadoras personales y dispositivos móviles que permitan probar la disponibilidad de la aplicación, así como su usabilidad y diseño responsivo.

2.6.2 *Software de Testing*

Para este tipo de pruebas se emplearán diversas versiones de los navegadores web más usados en la actualidad. Además de diferentes tipos de dispositivos móviles con sistemas operativos Android y iOS.

2.6.3 *Testing Automatizado*

Se deben desarrollar tests automatizados que comprueben las funcionalidades de lógica de negocio de la plataforma. Estas pruebas se corren bajo PHPUnit, Framework de Testing para aplicaciones PHP. Las pruebas automatizadas forman parte de las buenas prácticas del desarrollo del software y garantizan un nivel de confianza antes de realizar nuevas liberaciones de software, puesto que los test se ejecutan de forma automática y si cualquier prueba falla podría ser identificada y en caso de ser un falla ser corregida antes de una nueva liberación de software.

Las tareas mínimas que se deben de cumplir para el testeo son las siguientes:

- Diseñar o generar nuevas matrices de pruebas, con base a la que se incluye en esta documentación.
- Ejecutar análisis de flujos o escenarios alternos.
- Encontrar brechas en la funcionalidad de la aplicación.
- Proponer mejoras en la funcionalidad de la aplicación.
- Pruebas de optimización de la aplicación.
 - Velocidad de carga de las páginas.
 - Velocidad de respuesta de las funcionalidades.
- Pruebas de Usabilidad.
 - Análisis de la experiencia de usuario, considerando los perfiles de usuario de la aplicación.
 - Accesibilidad.
- Pruebas de Funcionalidad.
 - Verificar que se encuentren implementadas las funcionalidades establecidas para la plataforma.

- Verificar que las funcionalidades se ejecuten correctamente.
- Diseño y aplicación de pruebas de casos extremos para poder identificar errores.
- Pruebas exploratorias.
 - Diseño e implementación de baterías de pruebas exploratorias para crear diferentes escenarios de usabilidad y de funcionalidad.

3 Restricciones

3.1 Restricciones Hardware

Dado que la solución es una aplicación cliente-servidor, hay que ver las restricciones desde ambas partes. Desde el punto de vista del servidor, el equipo que se emplee debe tener la capacidad computacional para ejecutar de manera eficiente un servidor web como el Apache o nginx. Además, si se piensa desplegar el servidor de bases de datos en el mismo nodo (no es requerido) debe tenerse en cuenta la concurrencia en cuanto al número de peticiones que pueda atender la aplicación.

Desde el punto de vista del cliente, sólo será necesario que el dispositivo desde el cual se accede a la aplicación, tenga la capacidad de instalar y ejecutar aplicaciones móviles, así como alguno de los navegadores modernos.

3.2 Restricciones Software

El sistema debe ser lo suficientemente robusto como para evitar ataques malintencionados y fuga de información. El lenguaje y entorno de desarrollo y despliegue debe tener por tanto mecanismos de encriptación de datos, evitar ataques de SQL Injection y xss, entre otros. Además se debe garantizar que cada usuario con acceso al sistema pueda ver/modificar solamente la información asociada a sus funciones.

El sistema debe proveer mecanismos que garanticen su escalabilidad y extensibilidad, de manera que se puedan incorporar módulos posteriores al mismo sin romper la integridad de la versión en explotación.

3.3 Restricciones de Interfaz

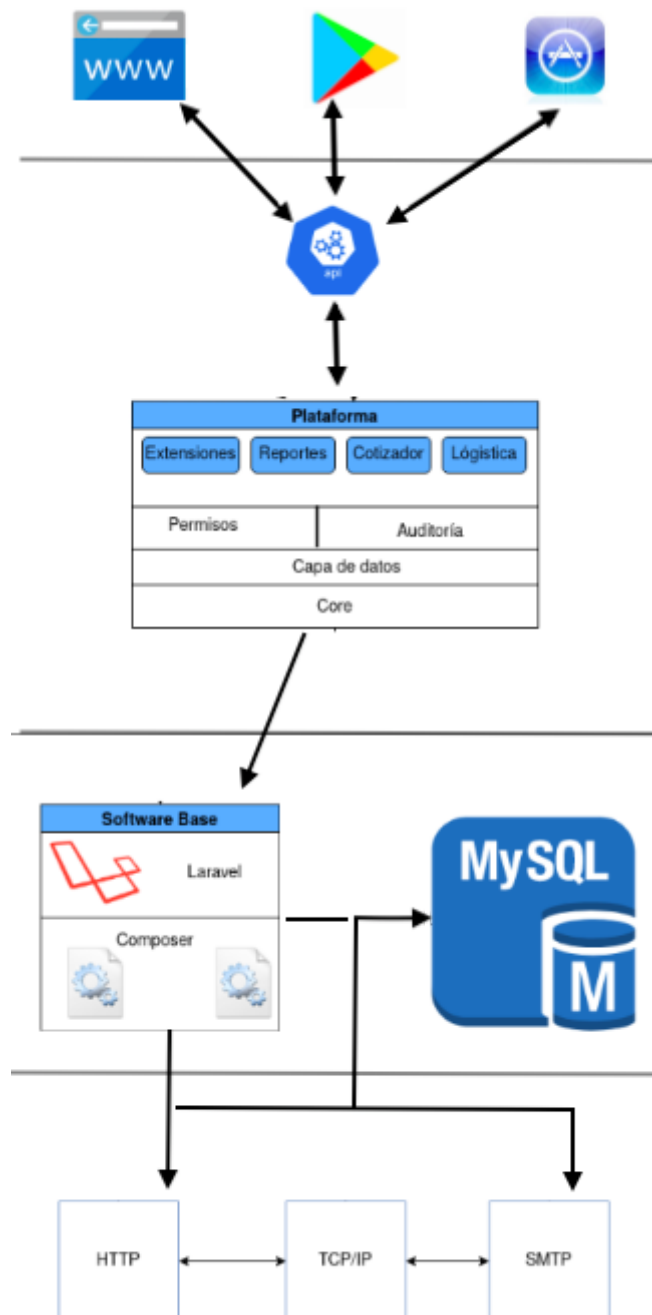
Las interfaces deben ser claras, simples y adaptables al dispositivo desde el cual se acceda a la plataforma. Se recomienda hacer primero un **mockup** de la página web y de la aplicación móvil, lo cual permitirá ver con antelación los flujos de cada proceso y requerimientos identificados, tomando muy en cuenta la experiencia de usuarios (UX).

4 Arquitectura del Software

4.1 Modelo Arquitectónico

El sistema estará estructurado en capas y usando un enfoque de reutilización para facilitar que los componentes sean agrupados en correspondencia con su nivel de reusabilidad.

A continuación se describen las capas a ser implementadas y que forman parte de la arquitectura del Software.



Protocolos: Como la plataforma tiene un sitio web y aplicaciones móviles, el protocolo principal a utilizar es el protocolo HTTP. El protocolo SMTP se utilizará para realizar el envío de correos electrónicos (por ejemplo para la solicitud de restablecimiento de contraseña). El protocolo TCP/IP se utiliza internamente para comunicar componentes como de software como el motor de base de datos.

Software Stack: Se refiere a las tecnologías de software principales a utilizar las cuales consisten en:

- Composer: Manejador de librerías de PHP.
- Laravel: Framework PHP.
- MySQL: Motor de base de datos.

Plataforma lógica: Contiene las capas internas del software.

- Core: Capa de más bajo nivel de la aplicación. Contiene configuración de las implementaciones a utilizar, basado en el modelo de inyección de dependencias lo que permite el desacoplamiento y atomicidad de la aplicación.
- Capa de datos: Contiene la lógica para acceder / modificar los datos.
- Permisos: Módulo encargado de autorizar o denegar las acciones de usuarios según su nivel de acceso.
- Auditoría: Componente de Software interceptor que permite programar qué acciones almacenar en cualquier medio (por ejemplo base de datos o un archivo de log).
- Módulos plataforma: Contiene la lógica del negocio.
 - Reportes: Permite filtrar información y exportarla a Excel.
 - Módulo administrativo.
 - Extensiones: Integraciones con otros servicios (por ejemplo servicios de envío de SMS).

4.2 Seguridad del Software

Como se mencionó, en el caso en que la plataforma sea desarrollada utilizando como framework Laravel, tendrá algunas características que se deben de considerar en el desarrollo e implementación de este proyecto para garantizar la seguridad.

- Ataques XSS. Para mitigar esta posible vulnerabilidad se utilizan tokens CSRF generados desde el servidor para cada formulario que el cliente puede enviar de regreso. Se valida que el token haya sido generado por el servidor lo cual agrega una capa extra de seguridad que evita los ataques denominados cross-domain.

- Inyección SQL. Laravel provee métodos que sanitizan los datos de entrada evitando los posibles ataques de inyección SQL. Además, se debe de implementar las mejores prácticas utilizando un ORM para acceder a la base de datos, evitando usar SQL directamente y proporcionando escalabilidad ya que existe la posibilidad de cambiar el motor de base de datos cambiando únicamente el driver a nivel de configuración, sin necesidad de ajustar sentencias SQL.
- CORS. Se provee de la funcionalidad de añadir seguridad a nivel de CORS lo que garantiza que la capa de API REST pueda ser consultada por diferentes servidores de manera segura.
- Encriptación de datos. Los datos sensibles como contraseñas se almacenan de forma encriptada, utilizando BCrypt; modelo de encriptación recomendado por la robustez en la generación de hashes. Laravel utiliza adicionalmente una llave generada de forma única por cada proyecto y que sirve como semilla de encriptación. Esto garantiza que una misma contraseña genere un hash distinto para cada proyecto.

4.3 Escalabilidad del Software

El software se debe desarrollar bajo estándares de programación de calidad, implementando una serie de patrones de desarrollo que garantizan la atomicidad, escalabilidad y desacoplamiento del proyecto.

Se deben de utilizar algunos patrones de desarrollo como Singleton, Builder, Factoría e inyección de dependencias. Utilizar patrones de desarrollo como la inyección de dependencias permitirá generar interfaces e implementaciones, evitando el acoplamiento de nuestra aplicación a una tecnología en concreto. Por ejemplo, este patrón de desarrollo nos permite cambiar el motor de base de datos simplemente cambiando el driver de base de datos (la inyección de dependencias se encarga de “inyectar” la implementación hacia el motor de base de datos correcto en base al driver configurado).

Los beneficios de aplicar estas prácticas de desarrollo se traducen en una aplicación escalable, fácil de mantener y fácil de testear con pruebas automatizadas.

5 Requerimientos de Desarrollo

5.1 Requerimientos Participación Cliente

Durante el proceso de diseño el personal de logística estará vinculado a la evolución de la obtención del producto. En la primera etapa su participación será crucial para una mejor comprensión de la lógica del negocio. De igual manera, durante la etapa de desarrollo habrá participación en el proceso de pruebas y revisiones de evolución del proyecto. A partir de la pila de necesidades a resolver se irá realizando un proceso iterativo que facilite la resolución de problemas hasta llegar a la solución final.

5.2 Requerimientos de Comunicación

La comunicación entre las partes debe ser muy productiva para la obtención de un mejor producto que responda a las expectativas de la solución final. Es por ello que se deben planificar las actividades de retroalimentación con el cliente durante todo el proceso de diseño y desarrollo de la solución.

5.3 Requerimientos de Infraestructura

Para la solución final no se requerirán elementos de infraestructura novedosos, simplemente conexión a internet y una comunicación fluida entre las partes.

6 Requerimientos Post Desarrollo

6.1 Requerimientos de Entrenamiento

Para una mejor comprensión de todo el sistema, se recomienda la planificación de una sesión de entrenamiento al término del desarrollo. El entrenamiento estará orientado al personal que operará el sistema (los trabajadores del negocio más el administrador del sistema). El período de entrenamiento podría ser de dos semanas.

6.2 Requerimientos de Mantención

Para el desarrollo del sistema se ha considerado la inclusión de una herramienta que garantice llevar un control de cambios sobre los archivos de código que componen la solución. Este sistema estará disponible para todo el equipo de desarrollo a través de internet y de manera privada en (gitlab.com). El personal tendrá acceso a la pila de necesidades, los *sprints* que se programen y además contarán con la capacidad de solicitar nuevas funcionalidades o mejoras a las ya disponibles.

6.3 Requerimientos de Ambiente de Pruebas

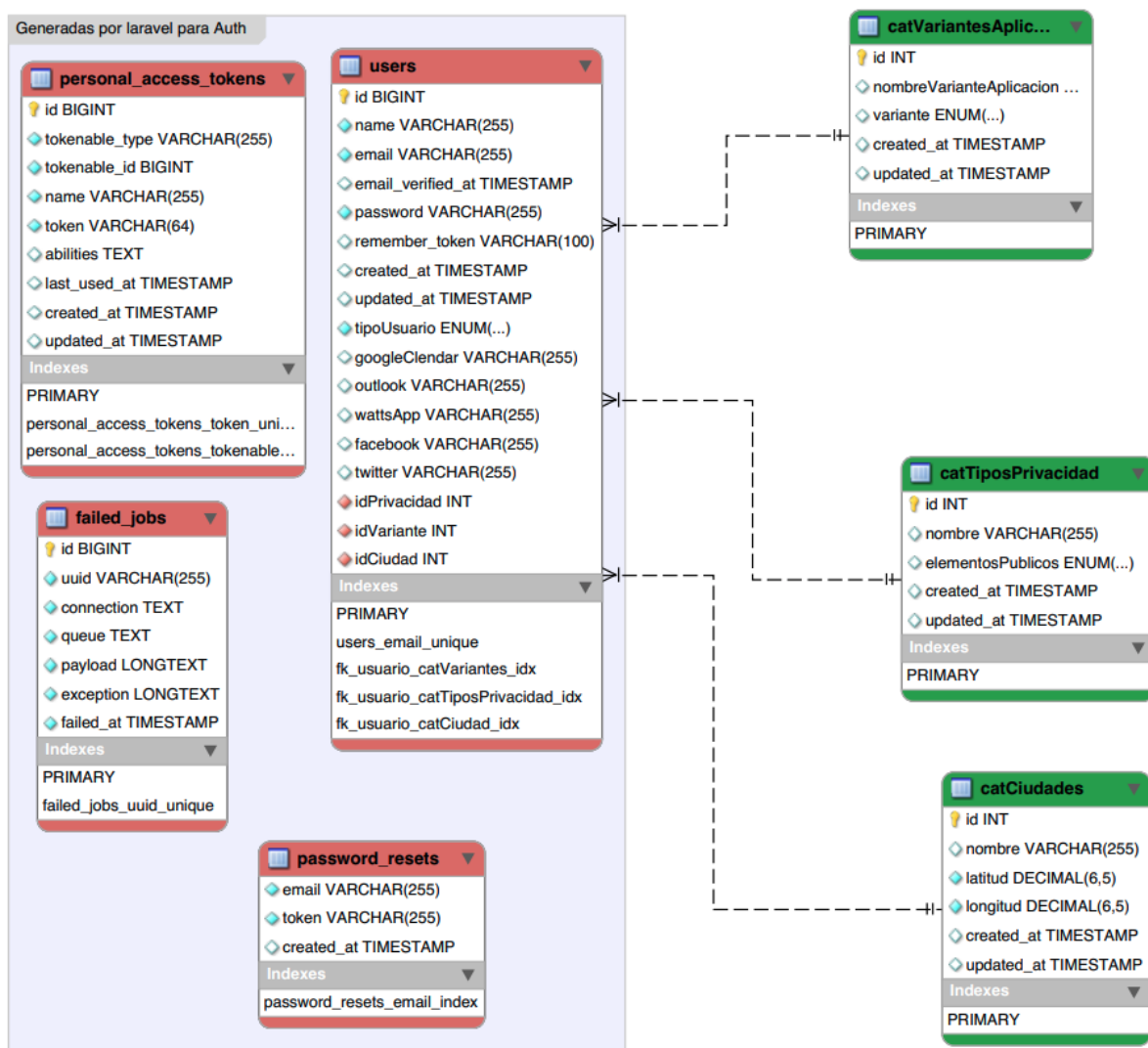
Para levantar las nuevas implementaciones en el ambiente de producción se recomienda hacer una matriz de testing que incluya las nuevas funcionalidades. Además de hacer un ambiente de pruebas que contenga los nuevos cambios o nuevas implementaciones.

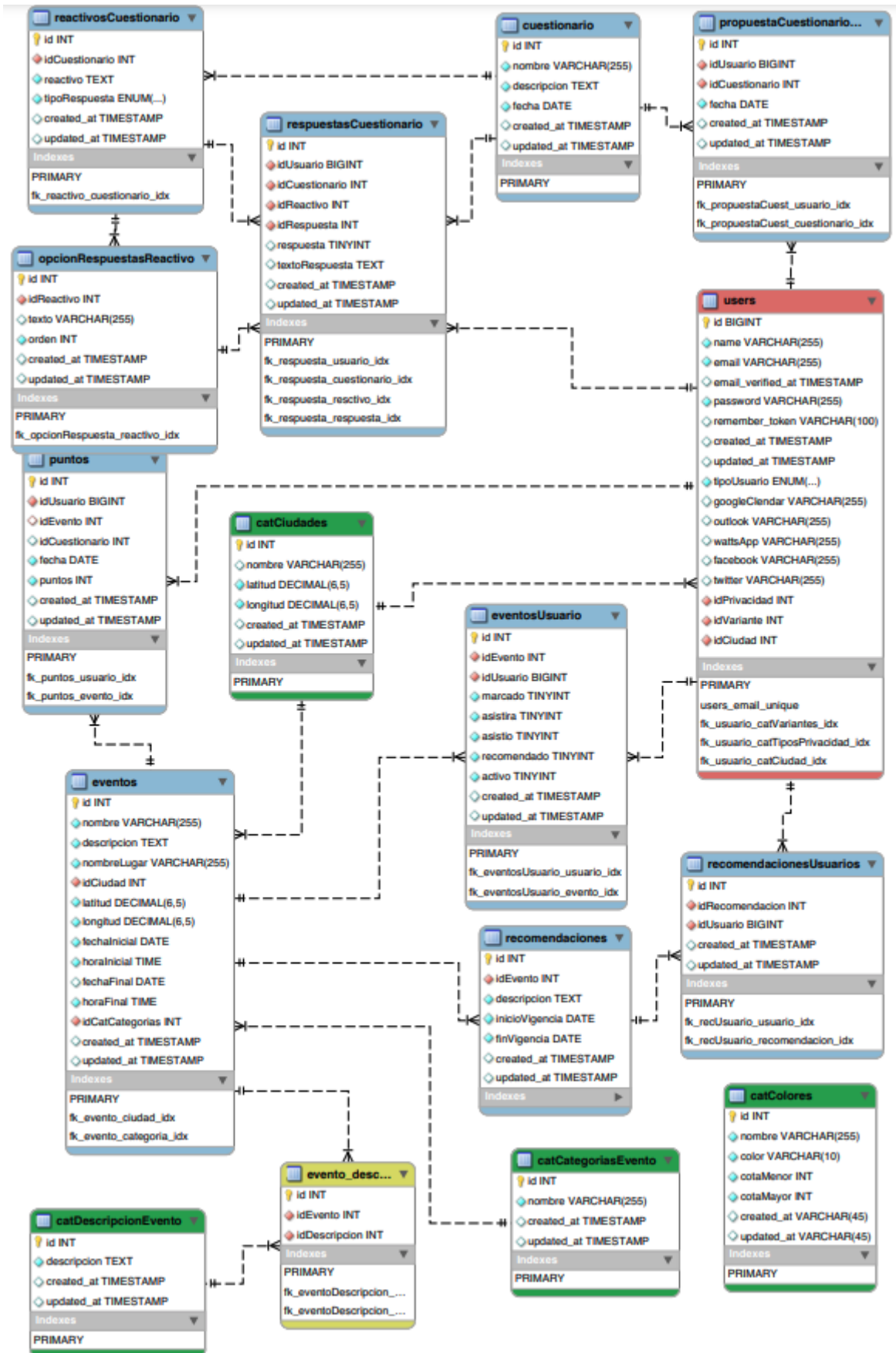
7 Diseño de la Base de Datos

A Continuación se muestra la estructura inicial de la base de datos, tablas y descripción de cada uno de los campos que conforman las tablas. Este diseño de la base de datos podría tener cambios al identificarse nuevas necesidades del producto.

7.1 Estructura

El diseño de base de datos inicial cuenta con 21 tablas relacionadas con las estructuras para el almacenamiento de datos.





7.2 Diccionario de Datos

En las siguientes tablas se describen los campos de cada tabla que componen la base de datos.

catCategoriasEvento						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
nombre	varchar(255)	\N		NO		Nombre de categoría
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
catCiudades						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
nombre	varchar(255)	\N		YES		Nombre de ciudad
latitud	decimal(6,5)	\N		NO		Latitud expresada en valor decimal
longitud	decimal(6,5)	\N		NO		Longitud expresada en valor decimal
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
catColores						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
nombre	varchar(255)	\N		NO		Nombre del color
color	varchar(10)	\N		NO		Valor hexadecimal con 8 caracteres para incluir transparencia
cotaMenor	int unsigned	\N		NO		Distancia usuario-evento minima
cotaMayor	int unsigned	\N		NO		Distancia usuario-evento maxima
created_at	varchar(45)	\N		YES		Fecha de creación
updated_at	varchar(45)	\N		YES		Fecha de última edición
catDescripcionEvento						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria

descripcion	text	\N		NO		Texto de descripción
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición

catTiposPrivacidad

Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
nombre	varchar(255)	\N		YES		Nombre del tipo de privacidad
elementosPublicos	enum('todos','parciales','ninguno')	parciales		YES		Tipo de privacidad
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición

catVariantesAplicacion

Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
nombreVarianteAplicacion	varchar(255)	\N		YES		Nombre de variante
variante	enum('publica','privada')	\N		YES		Variante de tipo de aplicación a desplegar
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición

cuestionario

Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
nombre	varchar(255)	\N		NO		Nombre del cuestionario
descripcion	text	\N		NO		Descripción del cuestionario
fecha	date	\N		NO		Fecha de elaboración
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición

evento_descripcionEvento

Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
idEvento	int unsigned	\N	MUL	NO		Llave foránea a tabla eventos

idDescripcion	int unsigned	\N	MUL	NO		Llave foránea a tabla descripción
eventos						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
nombre	varchar(255)	\N		NO		Nombre del evento
descripcion	text	\N		NO		Descripción del evento
nombreLugar	varchar(255)	\N		NO		Lugar del evento
idCiudad	int unsigned	\N	MUL	NO		Llave foránea a tabla catCiudades
latitud	decimal(6,5)	\N		NO		Latitud expresada en valor decimal
longitud	decimal(6,5)	\N		NO		Longitud expresada en valor decimal
fechaInicial	date	\N		NO		Fecha de inicio del evento
horaInicial	time	\N		NO		Hora de inicio del evento
fechaFinal	date	\N		YES		Fecha final del evento (si es pertinente)
horaFinal	time	\N		NO		Hora final del evento (si es pertinente)
idCatCategorias	int unsigned	\N	MUL	NO		Llave foránea a tabla catCategorias
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
eventosUsuario						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
idEvento	int unsigned	\N	MUL	NO		Llave foránea a tabla eventos
idUsuario	bigint unsigned	\N	MUL	NO		Llave foránea a tabla users
marcado	tinyint	0		NO		Booleano si el usuario lo marcó
asistira	tinyint	1		NO		Booleano si el usuario asistirá
asistio	tinyint	0		NO		Booleano si el usuario asistió
recomendado	tinyint	0		NO		Booleano si el usuario lo recomendó

activo	tinyint	1		NO		Booleano si el evento está activo para el usuario
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
failed_jobs						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	bigint unsigned	\N	PRI	NO	auto_increment	Generada por Laravel
uuid	varchar(255)	\N	UNI	NO		
connection	text	\N		NO		
queue	text	\N		NO		
payload	longtext	\N		NO		
exception	longtext	\N		NO		
failed_at	timestamp	CURRENT_TIMESTAMP		NO	DEFAULT_GENERATED	
opcionRespuestasReactivo						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
idReactivo	int unsigned	\N	MUL	NO		Llave foránea a tabla reactivos
texto	varchar(255)	\N		YES		Texto de la respuesta
orden	int unsigned	\N		NO		Orden en el que se representa
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
password_resets						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
email	varchar(255)	\N	MUL	NO		Generada por Laravel
token	varchar(255)	\N		NO		
created_at	timestamp	\N		YES		
personal_access_tokens						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	bigint unsigned	\N	PRI	NO	auto_increment	Generada por Laravel
tokenable_type	varchar(255)	\N	MUL	NO		
tokenable_id	bigint unsigned	\N		NO		
name	varchar(255)	\N		NO		
token	varchar(64)	\N	UNI	NO		

abilities	text	\N		YES		
last_used_at	timestamp	\N		YES		
created_at	timestamp	\N		YES		
updated_at	timestamp	\N		YES		
propuestaCuestionarioUsuario						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
idUsuario	bigint unsigned	\N	MUL	NO		Llave foránea a tabla users
idCuestionario	int unsigned	\N	MUL	NO		Llave foránea a tabla cuestionario
fecha	date	\N		NO		Fecha en la que se propone al usuario
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
puntos						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
idUsuario	bigint unsigned	\N	MUL	NO		Llave foránea a tabla users
idEvento	int unsigned	\N	MUL	YES		Llave foránea a tabla eventos
idCuestionario	int unsigned	\N		YES		Llave foránea a tabla cuestionario
fecha	date	\N		NO		Fecha de asignación de puntos
puntos	int unsigned	\N		NO		Puntos asignados
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
reactivosCuestionario						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
idCuestionario	int unsigned	\N	MUL	NO		Llave foránea a tabla cuestionario
reactivo	text	\N		NO		Texto de la pregunta
tipoRespuesta	enum('unico','multiple','abierto')	unico		NO		Tipo de respuesta a la pregunta
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición

recomendaciones						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
idEvento	int unsigned	\N	MUL	NO		Llave foránea a tabla evento
descripcion	text	\N		NO		Descripción de la recomendación
inicioVigencia	date	\N		NO		Fecha de inicio de vigencia de recomendación
finVigencia	date	\N		NO		Fecha de fin de vigencia de recomendación
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
recomendacionesUsuarios						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
idRecomendacion	int unsigned	\N	MUL	NO		Llave foránea a tabla recomendaciones
idUsuario	bigint unsigned	\N	MUL	NO		Llave foránea a tabla users
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
respuestasCuestionario						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	int unsigned	\N	PRI	NO	auto_increment	Llave primaria
idUsuario	bigint unsigned	\N	MUL	NO		Llave foránea a tabla users
idCuestionario	int unsigned	\N	MUL	NO		Llave foránea a tabla cuestionario
idReactivo	int unsigned	\N	MUL	NO		Llave foránea a tabla reactivos
idRespuesta	int unsigned	\N	MUL	NO		Llave foránea a tabla respuestas
respuesta	tinyint	1		YES		Respuesta booleana
textoRespuesta	text	\N		YES		Respuesta en forma de texto
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición

users						
Columna	Tipo	Default	Llave	Nulo	Otros	Descripción
id	bigint unsigned	\N	PRI	NO	auto_increment	Llave primaria
name	varchar(255)	\N		NO		Nombre de usuario
email	varchar(255)	\N	UNI	NO		Correo
email_verified_at	timestamp	\N		YES		Fecha de verificación de correo
password	varchar(255)	\N		NO		Clave de acceso
remember_token	varchar(100)	\N		YES		Token
created_at	timestamp	\N		YES		Fecha de creación
updated_at	timestamp	\N		YES		Fecha de última edición
tipoUsuario	enum('usuario','editor','administrador','superusuario')	\N		NO		Tipo de usuario
googleClendar	varchar(255)	usuario		YES		Liga a Google calendar
outlook	varchar(255)	\N		YES		Liga a outlook
whatsApp	varchar(255)	\N		YES		Liga a whatsApp
facebook	varchar(255)	\N		YES		Liga a facebook
twitter	varchar(255)	\N		YES		Liga a twitter
idPrivacidad	int unsigned	\N	MUL	NO		Llave foránea a tabla de catPrivacidad
idVariante	int unsigned	\N	MUL	NO		Llave foránea a tabla catVariantes
idCiudad	int unsigned	\N	MUL	NO		Llave foránea a tabla catCiudades

8 API

La API debe ser construida bajo los estándares de REST. Por lo que se utilizarán los siguientes métodos:

- POST: Crear un recurso.
- PUT: Actualizar un recurso.
- GET: Obtener recurso / varios recursos.
- DELETE: Eliminar un recurso.

La nomenclatura de las URLs debe de ser clara, simple, además de existir un endpoint para cada recurso. Un ejemplo de endpoint es [/api/users](#) el cual permite crear (POST), actualizar (PUT), obtener (GET) y eliminar (DELETE) recursos (usuarios en este caso). Para completar cada solicitud se debe enviar una petición HTTP con el verbo respectivo según la acción que se desee ejecutar sobre el recurso y utilizando las cabeceras:

- Content-Type: application/json
- Accept: application/json

Adicionalmente, deberá contar con un cuerpo que deberá contener los datos necesarios para crear el recurso. Si los datos no son correctos, la API debe de ofrecer información de validación de datos, respondiendo con el código HTTP 422 (Bad Request). Un ejemplo de un request con validación de datos fallida es el siguiente:

Request

```
POST /api/users
{
  "first_name": "Miguel"
}
```

Response

```
{
  "errors": {
    "email": [
      "Campo es requerido"
    ],
    "last_name": [
      "Campo es requerido"
    ],
    "password": [
      "Campo es requerido"
    ],
  }
}
```

```

        "role": [
            "Campo es requerido"
        ]
    },
    "message": "The given data was invalid."
}

// status code: 422
// Host: localhost:8000
// Date: Mon, 08 Nov 2021 09:26:14 GMT, Mon, 08 Nov 2021 09:26:14
GMT
// Connection: close
// X-Powered-By: PHP/7.4.3
// Cache-Control: no-cache, private
// Content-Type: application/json
// X-RateLimit-Limit: 60
// X-RateLimit-Remaining: 59
// Access-Control-Allow-Origin: *

```

Un ejemplo de un request de creación de usuario exitoso es el siguiente:

Request

POST [/api/users](#)

Content-Type: application/json

Accept: application/json

```

{
    "first_name": "Miguel",
    "last_name": "Ramirez",
    "email": "miguelramirez@correo.com",
    "password": "12345678",
    "role": "ADMIN"
}

```

Response

```

{
    "data": {
        "created_at": "2021-11-08T09:31:35.000000Z",
        "email": "miguelramirez@correo.com",
        "first_name": "Miguel",
        "last_name": "Ramirez",
        "role": "ADMIN",
        "updated_at": "2021-11-08T09:31:35.000000Z"
    },
    "message": "Usuario creado correctamente"
}

```

```

// status code: 201
// Host: localhost:8000
// Date: Mon, 08 Nov 2021 09:31:35 GMT, Mon, 08 Nov 2021 09:31:35
GMT
// Connection: close

```

```
// X-Powered-By: PHP/7.4.3
// Cache-Control: no-cache, private
// Content-Type: application/json
// X-RateLimit-Limit: 60
// X-RateLimit-Remaining: 59
// Access-Control-Allow-Origin: *
```

Otros códigos HTTP que deben ser implementados y la API los utiliza según el caso:

- 200 - Ok
- 201 - Creado
- 404 - No encontrado
- 422 - Solicitud malformada
- 403 - No autorizado
- 401 - No autenticado
- 500 - Error del sistema

Las respuestas van acompañadas de un mensaje (message) que contiene una corta descripción de la acción ejecutada.

Además, las solicitudes hacia los endpoint principales del sistema sólo deben ser accesibles mediante un usuario autenticado y que cuente con los permisos correctos para acceder / modificar dicho recurso. De lo contrario el usuario recibirá un mensaje indicando que no cuenta con los permisos o que no está autenticado junto con el código HTTP correspondiente. Los endpoint core del sistema serán accedidos únicamente por los frontend desarrollados.

8.1 End Points

A continuación se enlistan los endpoints detectados en primer lugar para los que serán usados por el frontend y en segundo por el backend.

Acciones	Acces point	Descripción
Instalación de app	/api/downloadApp	Download app
introducción de uso de mapas	/api/introduccionMapas	Devuelve datos introducción uso mapas
introducción del feed in app	/api/introduccionFeed	Devuelve datos introducción uso feed
Registro y admon usuario		
Registro de usuario (correo, google, etc.)	/api/registro	Envía datos de registro y devuelve usuario agregado
Administración de usuario	/api/usuario/datos	Devuelve datos de usuario para su edición
Actualización de datos de	/api/usuario/update	Envía datos editados de usuario y

Acciones	Acces point	Descripción
usuario		devuelve usuario modificado
Configuración de variantes de app	/api/usuario/variantes	Devuelve datos de variantes de configuración de usuario para su edición
Actualización de datos de variantes de usuario	/api/usuario/variantes/update	Envía datos editados de variantes de usuario y devuelve variantes modificadas
Editar nivel de privacidad	/api/usuario/privacidad/update	Envía datos de nivel de privacidad de usuario y devuelve nivel de privacidad modificado
Editar ciudad	/api/usuario/ciudad	Envía datos de ciudad de usuario y devuelve ciudad modificada
Login	/api/login	Envía datos de login y devuelve usuario accedido o error
Logout	/api/logout	Envía datos de logueo y devuelve resultado del logout
Eventos		
Consulta de eventos. (Por colores)	/api/eventos	Devuelve lista de eventos y su clasificación de color segun filtro
Mostrar en mapa eventos cercanos	/api/eventos/cercanos	Envía coordenadas de ubicación de usuario y devuelve conjunto de eventos y sus coordenadas en un radio de X km
Elegir categorias de eventos	/api/eventos/categorias	Envía categorías seleccionadas y devuelve categorías modificadas
Podrá agendar un evento	/api/eventos/agendar	Envía datos de evento, envía notificaciones a google o outlook y devuelve lista de eventos agendados o error
Poder elegir marcar eventos a los que asistirá.	/api/eventos/siguientes/marcar	Envía datos de evento, envia push notification y devuelve lista de eventos a los que quiere asistir o error
Poder elegir marcar eventos a los que asistió.	/api/eventos/asistencias	Envía datos de evento al que asistió y devuelve lista de eventos a los que asistió o error
Poder editar marcas de eventos	/api/eventos/marcar	Envia marca de evento, envía notificación push y devuelve

Acciones	Acces point	Descripción
		resultados
Poder ver marcas de eventos en el mapa	/api/eventos/marcados	Envía coordenadas de ubicación de usuario y devuelve conjunto de eventos marcados y sus coordenadas en un radio de X km
Compartir eventos a sus contactos. Invitar otros usuarios al evento al que asistirá	/api/eventos/compartir	Envía eventos a compartir y red social para compartir, envía notificaciones a las redes seleccionadas del evento
Recibir notificaciones de cambio en un evento	/api/eventos/modificaciones	Devuelve información nueva de evento
Recibir notificaciones sobre eventos que coincidan con sus búsquedas previas	/api/eventos/similares	Devuelve lista de eventos similares a las búsquedas
Recibir promociones sobre eventos según ubicación	/api/eventos/promociones	Devuelve lista de promociones
Permitir ver información de usuarios que se encuentran en el mismo evento	/api/eventos/asistentes	Devuelve lista de usuarios en evento
Comprar boletos de eventos.	/api/eventos/comprar	Envía datos evento y usuario, usa pasarela de compra y devuelve resultado

Acciones	Access point	Método	Descripción
Administración			
Login			
Login módulo administración	/admin/login	post	Envía datos de login y devuelve usuario logeado o error
Administración de cuentas de usuarios			
Listar	/admin/usuarios/	get	Devuelve lista de usuarios
Seleccionar uno	/admin/usuarios/{id}	get	Devuelve datos de usuario
Crear	/admin/usuarios/nuevo	post	Envía datos de nuevo usuario y devuelve usuario insertado o error
Editar	/admin/usuarios/update/{id}	put	Envía datos editados de

Acciones	Access point	Método	Descripción
			usuario y devuelve usuario modificado
Borrar	/admin/usuarios/delete/{id}	delete	Devuelve éxito o error en el borrado
Administración de cuentas de editores			
Listar	/admin/usuarios/editores/	get	Devuelve lista de editores
Seleccionar uno	/admin/usuarios/editores/{id}	get	Devuelve datos de editor
Crear	/admin/usuarios/editores/nuevo	post	Envía datos de nuevo editor y devuelve editor insertado o error
Editar	/admin/usuarios/editores/update/{id}	put	Envía datos editados de editor y devuelve editor modificado
Borrar	/admin/usuarios/editores/delete/{id}	delete	Devuelve éxito o error en el borrado
Administración de cuentas de administradores y super usuarios			
Listar	/admin/usuarios/administradores/	get	Devuelve lista de administradores
Seleccionar uno	/admin/usuarios/administradores/{id}	get	Devuelve datos de administrador
Crear	/admin/usuarios/administradores/nuevo	post	Envía datos de nuevo administrador y devuelve administrador insertado o error
Editar	/admin/usuarios/administradores/update/{id}	put	Envía datos editados de administrador y devuelve administrador modificado
Borrar	/admin/usuarios/administradores/delete/{id}	delete	Devuelve éxito o error en el borrado
Ciudades			
Listar	/admin/ciudades	get	Devuelve lista de ciudades
Seleccionar una	/admin/ciudades/{id}	get	Devuelve datos de ciudad
Crear	/admin/ciudades/nueva	post	Envía datos de nueva ciudad y devuelve ciudad insertada o error

Acciones	Access point	Método	Descripción
Editar	/admin/ciudades/update/{id}	put	Envía datos editados de ciudad y devuelve ciudad modificada
Borrar	/admin/ciudades/delete/{id}	delete	Devuelve éxito o error en el borrado
Categorías			
Listar	/admin/categorias	get	Devuelve lista de categorías
Seleccionar una	/admin/categorias/{id}	get	Devuelve datos de categoría
Crear	/admin/categorias/nueva	post	Envía datos de nueva categoría y devuelve categoría insertada o error
Editar	/admin/categorias/update/{id}	put	Envía datos editados de categoría y devuelve categoría modificada
Borrar	/admin/categorias/delete/{id}	delete	Devuelve éxito o error en el borrado
Eventos			
Listar	/admin/eventos	get	Devuelve lista de eventos
Seleccionar una	/admin/eventos/{id}	get	Devuelve datos de evento
Crear	/admin/eventos/nueva	post	Envía datos de nuevo evento y devuelve evento insertado o error
Editar	/admin/eventos/update/{id}	put	Envía datos editados de evento y devuelve evento modificado
Borrar	/admin/eventos/delete/{id}	delete	Devuelve éxito o error en el borrado
Aprobar evento dado de alta, de baja o editado	/admin/eventos/publicar/{id}	put	Marca evento como publicado y devuelve evento modificado
Subir información del preview de cada evento	/admin/eventos/preview/agregar/{id}	put	Envía información del preview y devuelve evento modificado
Subir coordenadas del evento	/admin/eventos/coordenadas/editar/{id}	put	Envía coordenadas del evento y devuelve evento modificado

Acciones	Access point	Método	Descripción
Campos de descripción de eventos			
Listar	/admin/eventos/camposDescripcion	get	Devuelve lista de campos de descripción de eventos
Seleccionar una descripción	/admin/eventos/camposDescripcion/{id}	get	Devuelve datos de campo de descripción de evento
Crear nueva descripción	/admin/eventos/camposDescripcion/nuevo	post	Envía datos de nuevo campo de descripción de evento y devuelve campo de descripción insertado o error
Editar descripción	/admin/eventos/camposDescripcion/update/{id}	put	Envía datos editados de campo de descripción de evento y devuelve campo de descripción de evento modificado
Borrar descripción	/admin/eventos/camposDescripcion/delete/{id}	delete	Devuelve éxito o error en el borrado
Asociar campo de descripción a evento	/admin/eventos/camposDescripcion/asociar/{id}	post	Envía datos de nuevo campo de descripción de evento y de evento, asocia el campo al evento y éxito o error
Simbología de colores según distancia			
Listar	/admin/simbologia	get	Devuelve lista de colores
Crear	/admin/simbologia/nueva	post	Envía datos de nuevo color y devuelve color insertado o error
Editar	/admin/simbologia/update/{id}	put	Envía datos editados de color y devuelve color modificado
Borrar	/admin/simbologia/delete/{id}	delete	Devuelve éxito o error en el borrado
Notificaciones Push			
Listar	/admin/notificacion	get	Devuelve lista de notificaciones
Crear	/admin/notificacion/nueva	post	Envía datos de nueva

Acciones	Access point	Método	Descripción
			notificación y devuelve notificación insertado o error
Editar	/admin/notificacion/update/{id}	put	Envía datos editados de notificación y devuelve notificación modificada
Borrar	/admin/notificacion/delete/{id}	delete	Devuelve éxito o error en el borrado
Seleccionar el targets de notificación	/admin/notificacion/seleccionar/{id}	put	Envía lista de usuarios asociados a notificación
Enviar notificación	/admin/notificacion/enviar/{id}	put	Envía id de notificación, envía notificación y devuelve éxito o error
Promociones			
Listar	/admin/promociones	get	Devuelve lista de promociones
Crear	/admin/promociones/nueva	post	Envía datos de nueva promoción y devuelve promoción insertada o error
Editar	/admin/promociones/update/{id}	put	Envía datos editados de promoción y devuelve promoción modificada
Borrar	/admin/promociones/delete/{id}	delete	Devuelve éxito o error en el borrado
Publicar promoción	/admin/promociones/publicar/{id}	put	Marca promoción como publicable y devuelve promoción modificada
Encuestas			
Listar	/admin/encuestas	get	Devuelve lista de encuestas
Crear	/admin/encuestas/nueva	post	Envía datos de nueva promoción y devuelve promoción insertada o error
Editar	/admin/encuestas/update/{id}	put	Envía datos editados de promoción y devuelve promoción modificada
Borrar	/admin/encuestas/delete/{id}	delete	Devuelve éxito o error en el

Acciones	Access point	Método	Descripción
			borrado
Seleccionar el target de notificación	/admin/encuestas/seleccionar/{id}	put	Envía lista de usuarios asociados a encuesta
Publicar encuesta	/admin/encuestas/publicar/{id}	put	Marca encuesta como publicable y devuelve encuesta modificada
Reportes			
Listar	/admin/reportes	get	Devuelve lista de reportes
Crear	/admin/reportes/nuevo	post	Envía datos de nuevo reportes y devuelve reporte insertado o error
Editar	/admin/reportes/update/{id}	put	Envía datos editados de reporte y devuelve reporte modificado
Borrar	/admin/reportes/delete/{id}	delete	Devuelve éxito o error en el borrado
Solicitar reporte	/admin/reportes/solicitar	post	Envía id de reporte y devuelve campos a filtrar
Generar reporte	/admin/reportes/generar	post	Envía datos de filtro y devuelve datos de reporte

9 Documentación de Soporte

Para la administración, gestión y mantenimiento de la plataforma se debe considerar comentar el código por parte de los desarrolladores, además de que durante el desarrollo del proyecto se debe generar como mínimo las siguiente documentación:

- Manual de usuario para el sitio web.
- Manual de usuario para la aplicación móvil.
- Manual de usuarios editor.
- Manual del administrador.