

**Documento Final
Primer Corte**

Cesar Alejandro Antolínez Poveda
Angela Daniela García Díaz
Juan Felipe González Forero
Juan Pablo Lara Lara

Agosto 2018.

Fundación Universitaria Panamericana.
Facultad de Ingeniería.
Diseño de Procesos de Software

Being able to solve different daily problems is one of the purposes of the development of technologies. Therefore, this document proposes a solution to the problem experienced by residents of a residential complex regarding the administration of these by means of routine meetings that no longer fulfill their function and have become spaces for generation of conflicts between residents, causing an unnecessary waste of money and resources.

Throughout the present work, initially, the creation of a software development company is addressed where the object, the market segment and the current competition in our geographical space is determined. Then, the product that is going to be offered is described, identifying technical characteristics for the user, which make it a value proposition. Likewise, functionalities and later versions are presented.

Poder dar solución a distintos problemas cotidianos es uno de los propósitos del desarrollo de tecnologías. Por ello, el presente documento propone una solución al problema que viven los residentes de un conjunto residencial respecto a la administración de estos por medio de reuniones rutinarias que ya no cumplen su función y se han convertido en espacios de generación de conflictos entre los residentes, ocasionando un gasto de dinero y recursos innecesario.

A lo largo del presente trabajo se aborda, inicialmente, la creación de una empresa desarrolladora de software donde se determina el objeto, el segmento de mercado y la competencia actual en nuestro espacio geográfico. Seguido, se describe el producto que se va a ofrecer, identificando características técnicas para el usuario, que hacen de ello, una propuesta de valor. Así mismo, se presentan las funcionalidades y versiones posteriores.

SoftClient	1
¿Qué hacemos?	1
¿Cómo lo hacemos?	1
Segmento del Mercado	1
Metas de la Empresa	1
Competidores	2
Empresa PSL	2
Empresa Gammapeit	3
Producto	5
Definición del Problema	5
Causas	5
Consecuencia	5
Comercialización	6
Avances tecnológicos.....	6
Propuesta de Valor.....	7
Descripción del producto	7
Características del producto.....	7
Características técnicas	8
Mínimo Producto Viable.....	9
Mapa de ruta	10
Atributos técnicos	10
Requerimientos funcionales y no funcionales	10
Infraestructura	12
Desarrollo.....	16
Frontend	16
Backend.....	19
Pruebas.....	20
Prueba Unitaria	20
Pruebas de Integración.....	20
Pruebas del Sistema	21
Prueba de Aceptación	21
Promesa de valor.....	22
Versión 2	22
Versión 3	22
Versión 4.....	22
Versión 5	22
Versión 6.....	22
Lista de referencias	23
Anexos	24
Video.....	24
Presentación	24

Lista de figuras

iv

Figure 1 Servidores para aplicaciones más usados a 2017	14
Figure 2 Arquitectura de la aplicación móvil.	15
Figure 3 Ejemplo de botones flotantes.....	16
Figure 4 Ejemplo de Backdrop.	17
Figure 5 Ejemplos de campo de texto.....	18
Figure 6 Íconos integrados.....	18
Figure 7 Ejemplo de paleta de colores usada.....	19

SoftClient

¿Qué hacemos?

Nuestra empresa se enfoca en solucionar problemas de gestión administrativas mediante la creación de herramientas de software especializados en el problema de nuestro cliente.

¿Cómo lo hacemos?

Lo hacemos mediante la asesoría que le ofrecemos al cliente antes y después del proyecto para garantizar que no solo funcione el proyecto, sino que se acople a la perfección y solucione dicho problema del cliente, para ello nuestra empresa cuenta con un grupo de trabajo confortable y una serie de procesos que certificaran nuestra calidad.

Segmento del Mercado

El perfil de nuestro cliente tienen en común que tienen un problema a nivel administrativo o gerencial, se pueden generalizar en organizaciones o empresas a nivel micro o pequeño, estos clientes tienen la capacidad de pagar por nuestros productos entre 50 millones a 150 millones y la localización principal de estos clientes están en las zona de Suba, Kennedy, Engativá, Usaquén y Chapinero en Bogotá ya que en esta zona se centraliza los clientes a quienes les apuntamos, donde la cantidad de organizaciones, microempresas y empresas es alrededor de los 650.000 entidades pero no obstante en zona norte de Bogotá hay un menor número de clientes pero igualmente potenciales.

Metas de la Empresa

Las metas mínimas de nuestra empresa en ventas es entre 200 millones y 250 millones al año donde equivalen de 3 a 6 productos por año, nuestro ingreso principal son

los nuevos productos ya que nuestro personal se especializa en la creación de software pero no obstante tenemos la línea de mantenimiento a aquellos clientes que necesiten mejoras, no tenemos servicio de soporte ya que las aplicaciones que se ofrecen son de estructura básica y fácil usabilidad, para cumplir estas metas necesitamos un proyecto nuevo cada 4 meses por máximo de 65 millones, llegara el caso que no también es un buen indicador tener cada 6 meses pero sobre los 150 millones.

Competidores

Empresa PSL. Esta empresa se dedica a desarrollar software usando como técnica principal, el desarrollo ágil Scrum.

PSL está compuesta por dos grandes líneas de negocio: servicios de outsourcing de TI (desarrollo de software, aplicaciones web, aplicaciones móviles, aplicaciones de misión crítica, Software Factory, outsourcing de recursos, BPO) y el desarrollo de productos preprogramados de software (sistemas de gestión empresarial ERP, sistemas de gestión de garantías, plataformas web-banking, entre otras).

Uno de los productos que PSL sustituye del mercado es el software transaccional de los bancos. PSL propone un software que comercia con dinero a través de la web y una solución que también trabaja con dinero de un cliente de banco a través de su teléfono, a esto lo llama Mobile Banking, esto renueva un poco las formas en que las personas realizan las transacciones en un banco.

Barreras e inhibidores:

Existen dos puntos importantes a resaltar, el primero es que el tiempo de ejecución y lanzamientos de MVP es más corto ya que tiene productos reprogramados y

no tienen que generar desde 0, lo segundo es que el competidor trabaja con mayor experiencia en la gestión de clientes, pero permite a su vez trabajar varias gestiones de la empresa como el administrativo o gerencial.

Diferenciadores.

Por lo general los productos reprogramados tiene una debilidad y es que en cierta manera el cliente se debe adaptar, nosotros tomamos en cuenta que una empresa no es igual a otra por lo tanto nos enfocamos en que nuestro producto e adapte al problema y que a su vez nuestro cliente mejore.

Canal de comunicación.

PSL se enfoca en mostrar sus servicios a través de su página web, donde los posibles clientes pueden entrar a ver sus ventajas y otros clientes con los que han trabajado, también cuentan con un correo para dar soporte a información solicitada de una persona natural y sus líneas telefónicas.

Empresa Gammapeit. Esta empresa se dedica a desarrollar software con los servicios de construcción desde cero a base de tecnología web o móvil y ofrece asistencia e implementación de herramientas tecnológicas de última generación y de fábricas globales.

Barreras e inhibidores.

Cuenta con certificación ISO 9001:2015 y Oracle que permiten demostrar su calidad y eficiencia en los productos que generan, adicional a eso cuentan con un equipo con experiencia en las mejores y últimas tecnologías.

Diferenciadores.

Con respecto a nuestros clientes tienen la peculiaridad de demorarse más en los proyectos ya que le apuntan a cualquier clase de problema y no tendrán una experiencia solida dentro de un género de problemas, a cambio de nosotros que estamos enfocados a problemas administrativos le damos la posibilidad al cliente de no solo solucionar sino mejorar los resultados de su problema a gran escala.

Canal de comunicación.

Gammapeit tiene una página web que permite expresar su esencia, pero su modo de trabajo consiste en la búsqueda de clientes con problemas en la totalidad de Bogotá.

Producto**Definición del Problema**

Ineficiente gestión de la administración de un conjunto residencial debido a largas jornadas entre el administrador y los residentes de este, para la toma de decisiones que afectan directamente su forma de vivir (basuras, aseo, zonas comunes etc). Así mismo, el desperdicio de recursos como papel para la publicación de avisos importantes y gastos económicos innecesarios con instrumentos de sonido y demás para llevar a cabo estas reuniones.

Causas

- Ausencia de mecanismos que faciliten una mejor comunicación de avisos, nuevos reglamentos y demás cuestiones que interfieren en la dirección y desarrollo de un conjunto residencial.
- Falta de buena comunicación entre la administración y los residentes de un conjunto residencial, lo cual fomenta reuniones de extensa duración.

Consecuencia

- Las decisiones del conjunto residencial son tomadas por unos pocos debido a la poca disposición que presentan sus residentes para asistir a las reuniones con la administración.
- Falta de participación de los residentes de un conjunto frente a problemáticas o propuestas que los afectan directamente.
- Falta de conocimiento frente a las funciones de la administración de un conjunto residencial.

Comercialización

El sistema que estamos proponiendo tiene una buena probabilidad de venta, ya que brinda seguridad, disminución del tiempo empleado en cada asamblea, siendo totalmente transparentes con las votaciones y conteo de estas. Es un sistema amigable con el cliente y sus usuarios.

Avances tecnológicos

El conjunto residencial ha venido pasando desde nuevas contrataciones de personal para que la gestión en estas asambleas sea más efectiva.

Han implementado más gastos en papelería para que cada punto de la asamblea que conlleva una votación sea más claro. Generando un papel por punto de votación.

Propuesta de Valor

Gestión eficaz y eficiente de la administración de un conjunto residencial, de manera tal, que todas las personas residentes conozcan trámites políticos, económicos y sociales en tiempo real, evitando, extensas juntas de administración realizadas de forma recurrente.

Descripción del producto

Aplicación web o móvil a la medida del cliente, que tiene como objetivo gestionar de manera organizada las funciones que cumple la administración de un conjunto residencial, así como también, informar a las personas que residen allí, de nuevas políticas dentro del mismo, puedan participar de forma activa en la toma de decisiones y conozcan toda la información que concierne a su cartera.

Características del producto

La aplicación pretende:

- Personalizar un sitio web privado para la administración de un conjunto residencial.
- Permitir una comunicación directa entre administrador y residentes, evitando el gasto de recursos como el papel para la construcción de volantes para dar a conocer un aviso o incidencia importante.
- Realizar el pago de la administración de manera virtual y se notifique que la transacción ha sido satisfactoria.
- Llevar registro y control de los estados financieros de la administración del conjunto residencial, reconociendo la utilidad de los pagos que realizan

mensualmente los residentes del conjunto (ingresos, egresos, facturas, presupuestos, fondos, etc.).

- Dar a conocer el reglamento de propiedad horizontal que repetidamente el administrador debe dictar a los residentes del conjunto residencial, debido a que constantemente se actualiza.
- Realizar una buena organización de parqueaderos, de tal manera que se pueda tener un flujo vehicular dentro del conjunto, beneficiando tanto a propietarios como a visitantes.
- Tener un monitoreo de la vigilancia establecida, con el fin de dar prioridad a la misma y si es el caso adjuntar los documentos requeridos para el reconocimiento de los residentes del conjunto.
- Dejar constancia de las solicitudes, necesidades, quejas o reclamos que algún residente pueda tener y que a su vez sea atendida de manera inmediata.
- Poder reservar áreas comunes del conjunto residencial por medio de un calendario público.

Características técnicas

- Elaboración de usuarios y contraseñas para el acceso de la aplicación de cada residente del conjunto.
- Principales noticias o anuncios de la administración
- Sistema de votación de propuestas o personas a cargo de la administración
- Acceso al reglamento de propiedad horizontal o código nacional de policía, donde pueda identificarse y darse cuenta de multas impuestas

- Acceso a los estados financieros que permitan reconocer las personas que están en mora

- Acceso al sistema de parqueo para conocer su espacio
- Acceso al apartado de comunicación directamente con el administrador para generar nuevas propuestas, quejas o reclamos.

Mínimo Producto Viable

Aunque, son varias las necesidades de los residentes de un conjunto residencial, la aplicación abordará en principio las más importantes mientras se realiza el proceso de implementación. Luego, en base a demás requerimientos que solicite el cliente, se desarrollarán progresivamente las actualizaciones de la aplicación.

A continuación, se detallan las características esenciales de la aplicación:

- Reconocimiento del reglamento de propiedad horizontal para que los residentes puedan ponerlo en práctica.
- Realización de pago de la administración de manera virtual.
- Poder de votación frente a propuestas o elecciones de las personas a cargo de la administración
- Identificar el presupuesto y la utilidad del capital recopilado de la administración
- Acceso al sistema de parqueadero para ocupar un lugar en el mismo

Mapa de ruta

Atributos técnicos

Requerimientos funcionales y no funcionales El aplicativo deberá cumplir las siguientes funcionalidades separadas en funcionales y no funcionales, de la siguiente manera:

Funcionales:

- El aplicativo debe tener un sistema de inicio de sesión para los usuarios que usaran la aplicación.
- El aplicativo debe tener diferentes roles (administrador y usuario) para poder distinguir los responsables de publicar nuevas encuestas.
- El aplicativo debe guardar el nombre y la contraseña de los usuarios en el servidor de base de datos.
- El aplicativo debe persistir los datos de un usuario durante el tiempo que este requiera usarlo.
- El aplicativo debe tener la opción de generar encuestas por parte del administrador.
- El aplicativo debe tener la opción de responder las encuestas generadas por parte de los usuarios.
- El aplicativo debe guardar cada una de las encuestas, estadísticas, peticiones que haga el administrador o un usuario.
- El aplicativo debe generar reportes de cada encuesta que se genere y así mismo mostrar quien la respondió.

- El aplicativo debe responder a todas las solicitudes que haga los usuarios como consultas.

No funcionales:

- El aplicativo debe almacenar la información de cada usuario en una base de datos ORACLE ENTERPRISE EDITION.
- El aplicativo debe consumir un servicio web para la comunicación con el servidor de Base de datos.
- El aplicativo debe permitir listar a todos los usuarios que existan.
- El aplicativo debe permitir que el administrador pueda cerrar una encuesta.
- El dispositivo móvil debe contar con una conexión a internet al menos de 2 Megas.
- El aplicativo debe tener una disponibilidad de 24 horas x 7 días.
- El aplicativo debe usar menos de un 1gb de memoria RAM en sus transacciones.
- El aplicativo no debe permitir el acceso a usuarios no registrados.
- El aplicativo debe cifrar las contraseñas en el algoritmo de seguridad SHA o StringBASE64.
- El aplicativo debe estar desarrollado bajo la plataforma JAVA EE.
- El aplicativo debe usar software libre para su desarrollo.
- El código fuente de la aplicación debe estar documentado totalmente en español.
- El aplicativo debe responder en menos de 0.5 segundos a cada petición.
- El aplicativo debe ocupar menos de 2Gb de memoria interna en el móvil.

Infraestructura Para poder llevar a cabo la entrega de al menos el MVP, se deben desarrollar los siguientes puntos de manera secuencial:

1) Elección de motor de BD: Como se había mencionado anteriormente en los requerimientos una de las funcionalidades esenciales del aplicativo es guardar de forma permanente que personas pueden usar el aplicativo, esto como en toda aplicación, debe hacerse a través de un motor de bases de datos. Es nuestro caso usaremos la versión Enterprise de Oracle, de este modo podemos hacer el sistema transaccional mediante PL/SQL.

2) Elección de repositorio de código para las versiones del código: Es importante poder versionar el código que se realice tanto para la parte del servicio web que será el encargado de hacer la conexión a la base de datos, tanto para el código fuente de la aplicación móvil. Esto nos dará datos de quien hace cambios dentro del código y la facilidad de corregir errores. También se podrán hacer posteriores forks de cada aplicativo. En nuestro caso usaremos GITHUB.

3) Elección de arquitectura para desarrollo de Webservice: Como se ha mencionado anteriormente en los requerimientos otra de las funcionalidades esenciales del aplicativo es la habilidad de consumir un servicio web esto con el fin de que la capa del modelo y la capa vista-controlador queden separadas y no se pierda código de los métodos que conectan a la base de datos. Por esto es importante crear un servicio web que incluya múltiples métodos que hagan conexión a la base de datos. Hemos decidido usar el estándar de servicios web SOAP, esto es porque nos aporta un nivel de seguridad mayor al que puede ofrecer otro tipo de servicios web como los RESTFUL.

4) Creación de EndPoints: En este punto ya debemos tener claro que la tecnología que vamos a usar por lo que podemos usar métodos de autenticación como autenticación básica o autenticación por cabecera para dar seguridad al acceso de los servicios web.

Estos servicios web para el MVP deben tener al menos los siguientes métodos:

- Autenticación (Método con el que un usuario se podrá autenticar)
- ListarUsuarios (Método con el que se listaran los usuarios registrados)
- CrearUsuario (Método con el que se registrara un nuevo usuario en la base de datos)
- EditarUsuario (Método con el que se actualizara un usuario de la base de datos)
- EliminarUsuario (Método con el que se inactivara un usuario de la base de datos)
- CrearEncuesta (Método con el que se creara una encuesta en la base de datos)
- EditarEncuesta (Método con el que se actualizara una encuesta de la base de datos)
- FinalizarEncuesta (Método con el que se cerrará una encuesta)

5) Elección de servidor de aplicaciones para WebService: Una vez desarrollados, al menos los métodos anteriormente mencionados, se procederá a exponer los servicios web en un servidor de aplicaciones que soporte múltiples peticiones y que responda a cada una de las peticiones. Para nuestro caso usaremos Apache Tomcat, ya que según plumbr.io (plumbr.io, 2018), hasta el año 2017 seguía siendo de los más usados gracias a su robustez y su facilidad de ajustes en cuanto a acceso de puertos y número de hilos que puede usar.

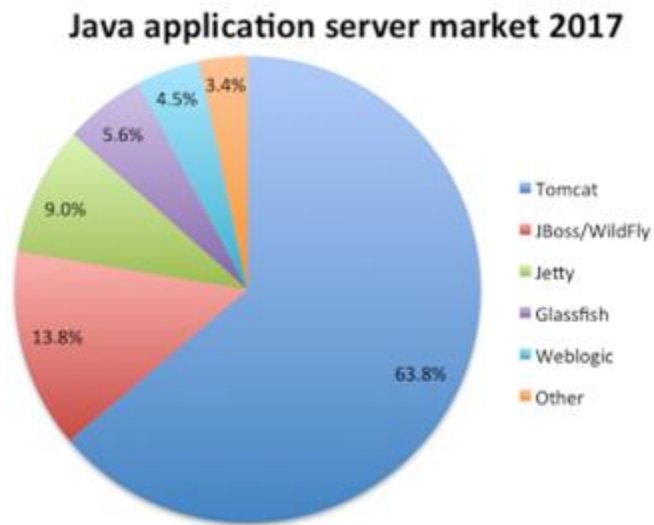


Figure 1 Servidores para aplicaciones más usados a 2017

6) Elección de IDE para desarrollar el aplicativo: Una vez expuesto los servicios que nos conectan a la base de datos, empezaremos a desarrollar la parte que conlleva el aplicativo como tal de móvil. Para eso nuestro equipo de desarrollo elegirá el ambiente de desarrollo que mejor se ajuste a nuestras necesidades. Para nuestro caso usaremos el ambiente Eclipse for Android Developers, esto porque es un ambiente familiar al de desarrollo JAVA y tiene soporte de Git Hub lo que nos facilita el proceso de versionamiento de código.

7) Elección de arquitectura para desarrollo de aplicación:

Con el fin de disminuir la complejidad de los módulos que se desarrollen, vamos a implementar patrones de diseño para nuestro desarrollo. El patrón de desarrollo que vamos a usar será el “patrón de fachada” lo que hace que cada petición se comunique con una interfaz y esta sea la encargada de hacer un redireccionamiento a la clase encargada de responder a cada solicitud.

El empaquetado de todas las clases se hará en base a cada funcionalidad de la aplicación, es decir, que todas las clases que se consideren “Entidades” irán en un paquete, las clases consideradas “Lógica” irán en otro paquete y así se procederá con todas las funcionalidades.

8) Creación módulos de registro y módulo de encuestas:

En un módulo que será exclusivo del registro de usuarios y otro modulo para la creación de encuestas se tendrán dentro de la aplicación móvil.

Estos módulos funcionaran del mismo modo para registrar, actualizar y eliminar información.

- Vista: La vista es la parte visual donde el administrador será el encargado de manipular los datos.
- Controlador: El controlador será el encargado de manipular los datos (fachada) y este enviar los datos las clases de lógica donde se validará la información.
- Lógica: Las clases “Lógica” serán las encargadas de validar que los datos sean acordes a los que se haya planteado en la base de datos.
- Llamado WS: estas clases invocaran al servicio web donde almacenara los datos.

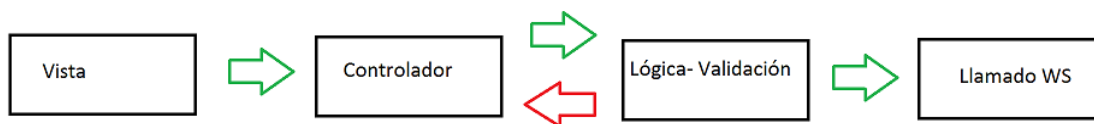


Figure 2 Arquitectura de la aplicación móvil.

9) Entrega MVP

Desarrollo

Es importante definir las bases con las que se desarrollara el frontend y el backend del aplicativo.

Frontend Como es habitual para las aplicaciones generadas como target Android, se usarán los temas visuales que tiene Google: Material Design.

Como particularidad usaremos la versión que fue lanzada en el mes de mayo/2018 de material design o Material2.

Particularmente, usaremos los botones flotantes en las partes inferiores de las barras de la aplicación para ejecutar acciones importantes como registrar un nuevo usuario, actualizar una encuesta, etc.

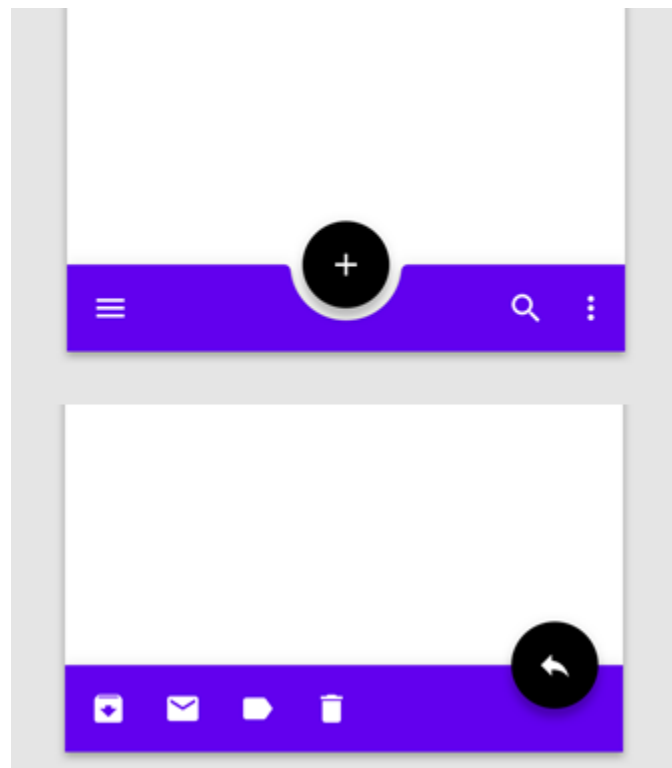


Figure 3 Ejemplo de botones flotantes.

Otra de las principales funcionalidades que usaremos de material design es el Backdrop, lo que nos ayudara a mostrar las opciones que puede contener un menú.

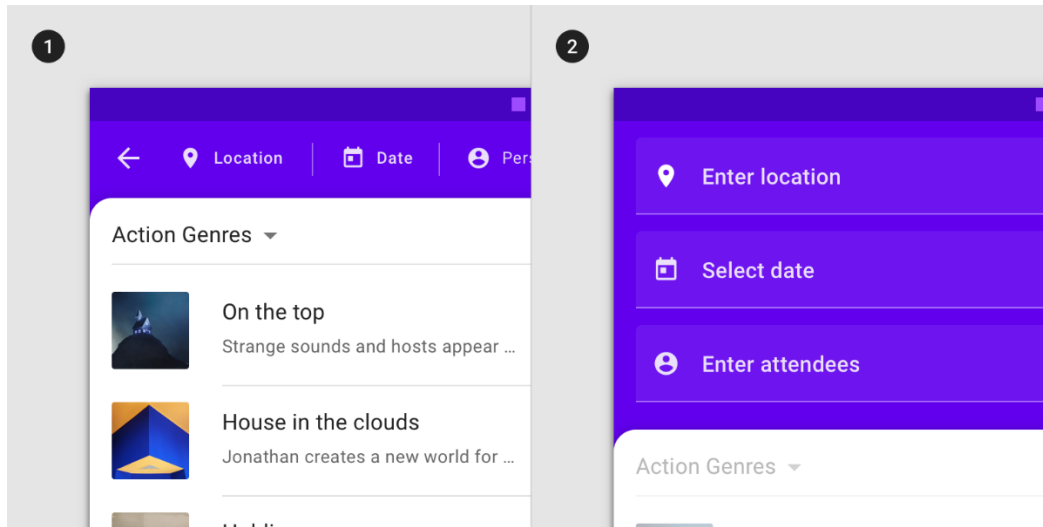


Figure 4 Ejemplo de Backdrop.

Otras de las funcionalidades que se tendrán en cuenta será los campos de texto para los usuarios, se usaran dos tipos de campos de textos el primero llama más la atención, con un sombreado oscuro que permite identificar donde se encuentra y destacarlo del resto de la interfaz.

El segundo, pensado para usarse en formularios y sitios con muchos campos a rellenar, llama menos la atención y ver muchos de ellos cansa menos la vista.

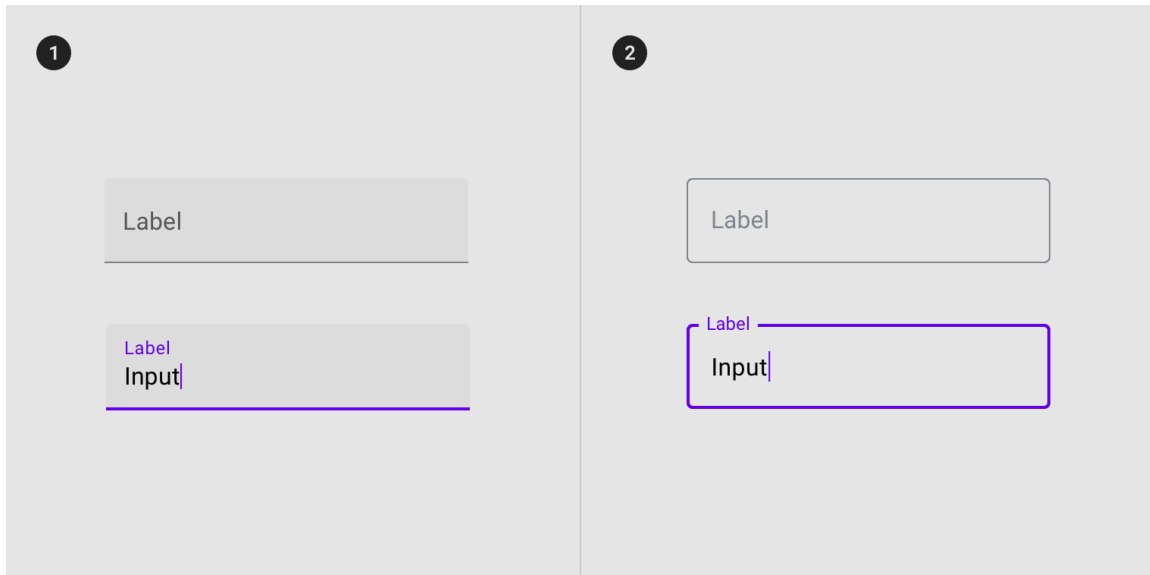


Figure 5 Ejemplos de campo de texto.

Los iconos que se usarán serán todos de la paleta de FontAwesome integrados con el plugin de material design2.

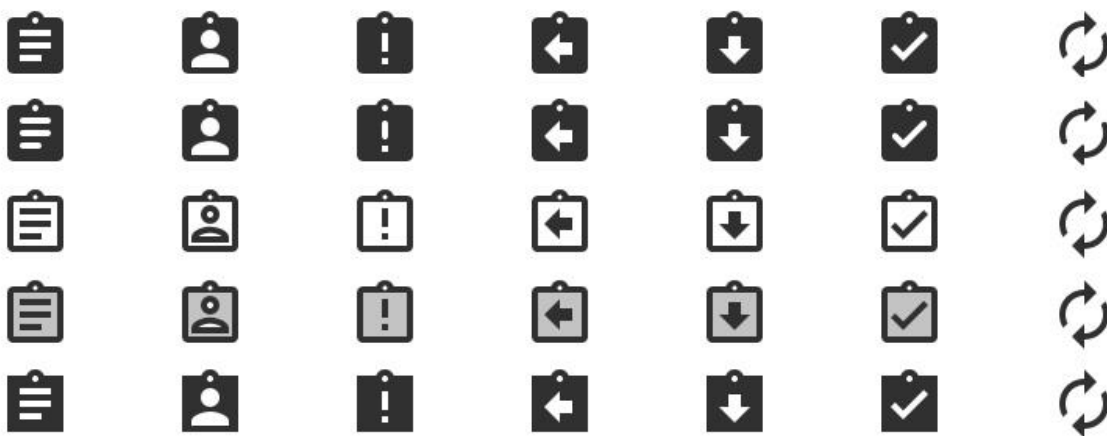


Figure 6 Íconos integrados.

Finalmente, la paleta de colores que se usara para este proyecto principalmente usara colores azules.

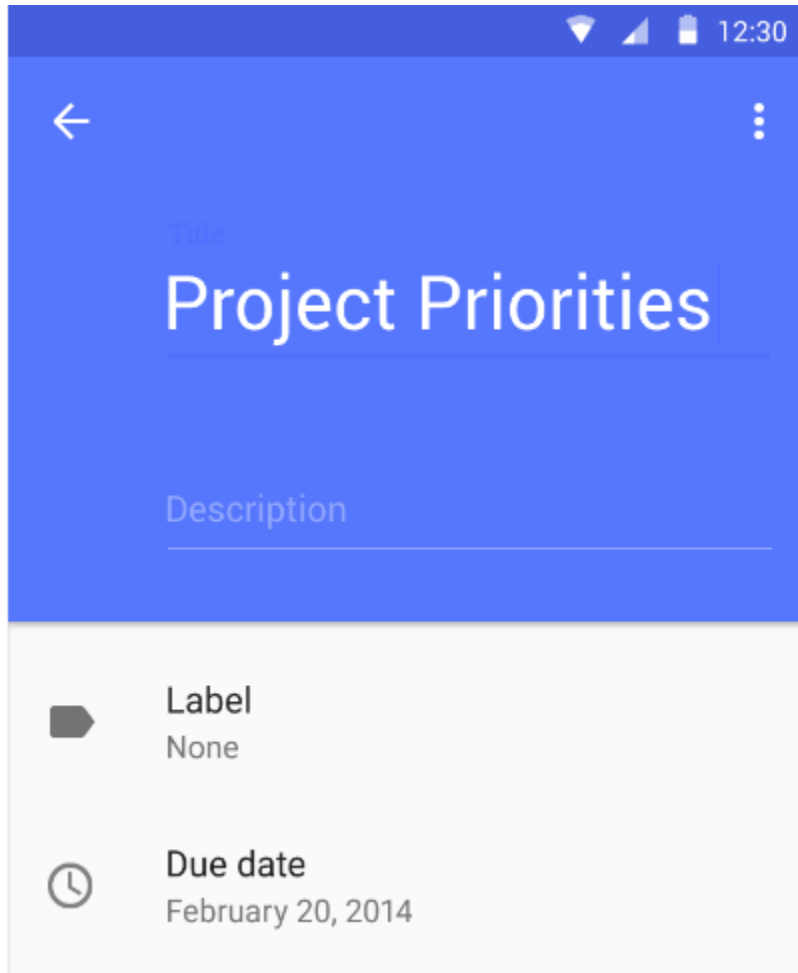


Figure 7 Ejemplo de paleta de colores usada.

Backend Para facilitarnos el desarrollo del backend, nuestra empresa usara el backend como servicio, lo que se conoce como Baas (backend as a service) que nos ayudan a concretarnos en la parte lógica de nuestro negocio.

Firebase de Google nos proporciona los siguientes puntos:

- SDK: reúne un grupo de herramientas que permiten la programación de aplicaciones móviles.

- Analíticas: Firebase nos da un conjunto de estadísticas que usuario imprescindible para el monitoreo del uso de la aplicación.
- Crash reporting: reportes necesarios para el control de bugs por parte de un usuario.
- Servicios de autenticación: Nos debe permitir la autenticación basada en usuario (email) / password, OAuth integrado con Google, Facebook, Twitter, etc. O integrar nuestro propio sistema de autenticación.
- Push notifications: Nos permite enviar notificaciones a la app cuando necesitamos comunicar algo al usuario o sincronizar algún dato.
- Dashboard: Control de todo el mBaaS, desde la creación de usuarios o los modelos de base de datos a la monitorización de las estadísticas.

Pruebas

Antes de entregar un desarrollo nuestro equipo de aseguramiento de calidad o QA, tendrán que realizar las siguientes pruebas para poder decir que el desarrollo se realizó de forma correcta:

Prueba Unitaria Estas pruebas se enfocan en validar cada módulo por separado, las técnicas que se usan para estas prueban son las de: comparar el resultado esperado con el resultado obtenidos, particionar los módulos en pruebas en unidades lógicas fáciles de probar.

Pruebas de Integración Estas pruebas se hacen con el fin que se identifican errores introducidos por la combinación de programas probados unitariamente. Verificar que las

interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente.

Pruebas del Sistema Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. Las pruebas del sistema deben enfocarse en requisitos que puedan ser tomados directamente de casos de uso y reglas y funciones de negocios.

Prueba de Aceptación Determinación por parte del cliente de la aceptación o rechazo del sistema desarrollado.

Promesa de valor

Para las futuras versiones de la aplicación se deberá tener al menos siguientes ítems:

Versión 2

El aplicativo tendrá un apartado donde los usuarios puedan tener conocimiento sobre alguna posible infracción que puedan tener.

Versión 3

El aplicativo tendrá un apartado donde cada usuario pueda proponer mejoras o algún plan de acción que se pueda aplicar dentro del conjunto residencial, esto haciéndose llegar al administrador del aplicativo como un escrito o una nota de voz.

Versión 4

El aplicativo tendrá un apartado donde se puedan ordenar por periodos de tiempos el espacio de los parqueaderos y así hacer que estos no se ocupen siempre por los mismos usuarios.

Versión 5

El aplicativo tendrá un apartado donde se pueda ordenar un calendario para la administración de espacio públicos como los salones comunales, sitios de recreación (si las tiene), entre otros.

Versión 6

El aplicativo tendrá un apartado para la administración de documentos entre el administrador y los usuarios. Esto se podrá administrar a través de los APIS de Google.

Lista de referencias

plumbr.io. (2018). *Most popular java application servers*. Obtenido de
<https://plumbr.io/blog/java/most-popular-java-application-servers-2017-edition>

Anexos

Video: [SoftClient - diseño procesos de software. Unipanamericana](#)

Presentación: [SoftClient](#)