



PLANNING



APP para gestión de tus productos financieros

APP para gestión de tus productos financieros

Presentado por:

Edwin Rodríguez y Daniel Murcia

Asesores:

Ingeniera Dianalin Neme Prada. Coordinadora Académica

Ingeniero Iván Rodrigo Romero Flórez. Docente Arquitectura de Software

Proyecto de grado como requisito para optar al título de:
Especialista en Ingeniería de Software

Universidad Antonio Nariño

Facultad de Ingeniería de Sistemas

Especialización Ingeniería de Software

Bogotá D.C.
2020

Tabla de Contenidos

Contenido

Capítulo 1	6
Capítulo 2	7
Marco Teórico	7
Capítulo 3	15
Estado del Arte	15
Capítulo 4	17
Formulación del Problema	17
Capítulo 5	19
Objetivos	19
5.1. Objetivo General.	19
5.2. Objetivos Específicos.	19
Capítulo 6	21
Metodología	21
Capítulo 7	25
Proceso de Software	25
7.1. Requerimientos Funcionales.	25
7.2. Requerimientos No Funcionales.	26
7.3 Diseño y arquitectura	27
7.3.1 Diagrama de despliegue.	27
7.3.2 Diagrama Casos de Uso.	28
7.3.3 Diagrama de Secuencia	29
7.3.4 Arquitectura de Alto Nivel	30
7.3.5 Diagrama de Clases	31
7.3.6 Diagrama de Clases	32
7.4 Construcción	33
7.5 Pruebas	36
7.4.1 Instalación y configuración	41
Conclusiones	43
Referencias Bibliográficas	44
Anexos	45

Lista de figuras

Ilustración 1	20
Ilustración 2	22
Ilustración 3	25
Ilustración 4	26
Ilustración 5	27
Ilustración 6	28
Ilustración 7	29
Ilustración 8	30
Ilustración 9	31
Ilustración 10	34
Ilustración 11	36
Ilustración 12	38
Ilustración 13	39

Capítulo 1

Introducción

La educación en Colombia así como en muchos más países latinoamericanos no cuenta con una educación financiera. No existe educación dirigida a formarse financieramente. Nuestra aplicación, busca dar hacer entender a los usuarios, cuáles compras son beneficiosas y cuáles no, a través de la tecnología, buscamos ayudar a las personas entre 20 y 28 años, concientizarlos de hacer compras responsables con los productos de créditos que ofrecen y suministran las entidades financieras.

Capítulo 2

Marco Teórico

2.1. Finanzas

Finanzas son el área de la economía que estudia el funcionamiento de los mercados de dinero y capitales, las instituciones que operan en ellos, las políticas de captación de recursos, el valor del dinero en el tiempo y el coste del capital. 1. Las finanzas se pueden dividir de varios tipos: Finanzas personales, finanzas empresariales y finanzas públicas.

2.2. Finanzas Personales

Las finanzas personales se interpretan como la generación de ahorros y obtener recursos financieros adicionales por parte de individuos o núcleos familiares. Del mismo modo, abarca las decisiones para optimizar la inversión de excedentes de recursos, todo relacionado a una cantidad de tiempo.

Las finanzas personales se pueden dividir en áreas tales como:

Ganancia de recursos: En esta área se especifica cómo se va a ganar el dinero, estrategias para ganancia de dinero como emprendimientos pequeños o sueldo base y donde más responsabilidad se sienta, donde incluye desarrollo profesional, personal, oportunidades de empleo.

Gasto: Es la salida de dinero que cambia para acreditar la adquisición de un servicio o un producto, donde se definirá qué tipos de gastos se podrán adquirir sea de esparcimiento, día a día, vivienda.

Planificación: La planificación de finanzas es un proceso de análisis, donde evalúa qué tipos de ingresos, cuáles son las expectativas en determinados tiempos, jubilación, etc. Establecer objetivos, identificar y priorizar los gastos. Boceto y definición de estrategias de cómo llevar a cabo los objetivos y cuáles son las herramientas financieras apropiadas. Ejecución donde se lleva a cabo plan financiero considerando ajuste en caso de imprevistos. Monitoreo de gastos y reevaluación en caso tal que existan cambios relevantes en las condiciones iniciales.

2.3. Crédito

El crédito es un préstamo de dinero que una parte facilita a la otra, con el compromiso futuro, de quien lo recibe devolverá dicho préstamo de forma gradual, parcial o un solo pago mediante el pago de cuotas y con un interés adicional que compensa a la parte que presta, por el tiempo que no tuvo ese dinero.

Existen diferentes tipos de créditos que otorgan diferentes beneficios tratando que sea el más adecuado en concordancia con la cultura financiera del cliente:

- Crédito de consumo: es un préstamo realizado a una persona, para adquirir bienes o servicios, como la compra de electrodomésticos, remodelaciones, vehículos, productos para el hogar, viajes, actividades de entretenimiento, entre otros. El crédito de consumo puede ser de corto y mediano plazo.

- Crédito hipotecario: es una alternativa de financiación para personas naturales mediante la cual pueden adquirir vivienda nueva o usada, construir, reparar, remodelar, o subdividir una casa que ya esté en uso. El crédito hipotecario es de largo plazo.

- Microcrédito: es un préstamo de monto reducido, que sirve principalmente para financiar proyectos que ya están en marcha, como el mejoramiento de producción de microempresas, o para las unidades productivas ya constituidas, y que no necesariamente estén registradas en la Cámara de Comercio. El microcrédito es de mediano plazo.

- Créditos especiales: son préstamos que apuntan a cubrir una necesidad empresarial o hacer inversión en una actividad primaria económica, como el crédito rural y agropecuario, el crédito para empresas o el capital de trabajo. 2

Todos los créditos poseen ciertas características comunes entre ellos:

- El monto, es la cantidad de dinero máxima que la entidad tiene establecida como tope a conceder para cada producto financiero.

- Plazo de devolución, tiempo en el que se presta el dinero y en el cual se regresa la misma mediante pagos parciales o pago total.
- El tipo de interés es el coste adicional que se debe pagar por recibir el dinero durante el periodo de tiempo que dure la operación (plazo de devolución).

2.4. Arquitectura de micro servicios.

Una arquitectura de micro servicios consta de una colección de servicios autónomos y pequeños. Los servicios son independientes entre sí y cada uno debe implantar una funcionalidad de negocio individual. En cierto modo, los micro servicios son la evolución natural de las arquitecturas orientadas a servicios, pero hay diferencias entre los micro servicios y estas arquitecturas.

2.5. Lenguaje Unificado de Modelado (UML).

El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos

reciclados de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

2.6. API.

Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones.

2.7. API REST.

REST cambió por completo la ingeniería de software a partir del 2000. Este nuevo enfoque de desarrollo de proyectos y servicios web fue definido por Roy Fielding, el padre de la especificación HTTP y uno los referentes internacionales en todo lo relacionado con la Arquitectura de Redes, en su disertación ‘Architectural Styles and the Design of Network-based Software Architectures’. En el campo de las API, REST (Representational State Transfer- Transferencia de Estado Representacional) es, hoy, la alfa y omega del desarrollo de servicios de aplicación

2.8. Base de datos MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y está a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee los derechos de autor de la mayor parte del código. Esto es lo que posibilita el esquema de doble licenciamiento anteriormente mencionado. La base de datos se distribuye en varias versiones, una Community, distribuida bajo la Licencia pública general de GNU, versión 2, y varias versiones Enterprise, para aquellas empresas que quieran incorporarlo en productos privativos. Las versiones Enterprise incluyen productos o servicios adicionales

tales como herramientas de monitorización y asistencia técnica oficial. En 2009 se creó un Fork denominado MariaDB por algunos desarrolladores (incluido algunos desarrolladores originales de MySQL) descontentos con el modelo de desarrollo y el hecho de que una misma empresa controle a la vez los productos MySQL y Oracle Database.

2.9. React Native

React Native es una de las tecnologías más de moda en este momento, y nadie puede negar que está revolucionando el desarrollo móvil. En esta entrada vamos a resumir cuáles son sus características principales y sus ventajas sobre los competidores actuales.

Hasta hace bien poco, había dos enfoques muy diferentes para desarrollar APPS y poder publicarlas en las tiendas oficiales:

- Desarrollos en plataformas nativas (Android-Java, Ios-Objective C/Swift)
- Desarrollos en plataforma web incrustada dentro una webview, o Web App (Cordova, Ionic, ...)

Cada enfoque tiene sus ventajas e inconvenientes, y dependiendo del proyecto, podría ser más beneficioso utilizar uno u otro.

Las APPS nativas tienen un aspecto profesional y su rendimiento es considerable, pero es necesario conocer cada plataforma a fondo y son propensas a tener una infinidad de fallos (y sus inevitables retrasos en la entrega). Otro aspecto a tener en cuenta es que al desarrollar en nativo, se tiene la sensación de estar repitiendo el trabajo una y otra vez en cada plataforma.

Capítulo 3

Estado del Arte

El mundo de las finanzas personales posee una variedad de productos para facilitar a todos los usuarios realizar la validación de sus estados financieros, simulaciones de compras y créditos, hasta hojas de cálculos. Entre esos productos se encuentran aplicaciones móviles para celulares tales “como haga cuentas”, esta aplicación ofrecida por el banco Davivienda el usuario puede gestionar y elaborar presupuestos mensuales, comparar valores reales de su presupuesto, programar alertas para el manejo de su presupuesto, comparar a través de gráficas o tablas sus presupuestos personales de los 3 o 6 meses anteriores. También aplicaciones como “presupuesto Rápido” ofrecida en la tienda para Android (Play Store) con más de un millón de descargas donde ofrece la aplicación tiene una página de resumen completamente personalizable para ver toda la información importante en un vistazo, categorías, tarjetas de crédito, transacciones, presupuestos, gráficas y calendario.

Ningún simulador cuenta con una revisión activa de gastos, sino una revisión pasiva que se limita a registrar los gastos y no como a proceder si este gasto es rentable o inteligente. Los simuladores de crédito que ofrecen entidades Bancarias como Davivienda, Bancolombia, entre otros bancos, son de bastante utilidad para realizar la simulación de compras independiente ya que se hace uso de servicios que la entidad ofrece. Sin embargo, el simulador se limita a evaluar el pago de las cuotas pero no a evaluar si esta compra fue inteligente o no.

3.1. Impacto.

La elaboración de “planning” va a permitir favorecer la adquisición de conocimientos y habilidades para el manejo de finanzas personales. Para empezar, se trata de una aplicación que muestra que posee los componentes comunes para elaboración de balance de gastos, donde el sistema basándose en criterios de manejo de finanzas busca explicar al usuario si las compras que realiza son o no una compra inteligente. Dada esto se presentan nuevos escenarios de aprendizaje, se fortalece el proceso de construcción del conocimiento.

3.2. Componente de Innovación.

Todas las herramientas detalladas anteriormente todas poseen un comportamiento similar, los egresos sean compras, gastos, entre otros, son registrados de forma que el usuario no obtiene un retorno de cómo deberían ser los egresos y la intención de "planning" es que el usuario obtenga un retorno “una opinión” de cómo se están realizando los gastos si la compra del usuario de por ejemplo un celular de 700.000,00 llevada hasta 36 cuotas porque “son más económicas” pero el usuario no toma en cuenta interés, cuotas de manejo, inclusive seguros.

Capítulo 4

Formulación del Problema

La educación financiera de Colombia se encuentra en números rojos, así lo demuestra el estudio realizado por David Cuervo director de Wealth en Mercer Colombia. Donde muestra que Colombia tiene un bajo índice de bienestar financiero, el 70 % de los Colombianos destinan más de la mitad de sus ingresos en el pago de las deudas como son la hipoteca, arriendo y en la atención de sus necesidades básicas.

Los colombianos que viven en Bogotá, Medellín y Cali tienen por lo general más de dos tarjetas de crédito que utilizan para efectuar avances en el pago de la otra tarjeta (Pagar una deuda con otra).

Esto, conlleva a que las personas cada vez se encuentren en un estado de déficit financiero y estas situaciones se están trasladando a ambientes familiares y laborales, donde David Cuervo comenta que, “el estrés financiero es un factor que incide directamente en la productividad y eficiencia de los empleados toda vez que el 52 % de los encuestados admitió que semanalmente ocupan más de 5 horas laborales resolviendo dificultades relacionadas con dinero y obligaciones financieras.”.

4.1. Descripción del Problema.

La vida crediticia de los jóvenes entre 20 y 28 años se ve asediada por entidades financieras, debido a su desconocimiento, acceden a productos financieros que al final no pueden pagar debido a los altos intereses que se deben pagar y en muchas de las ocasiones se desconocen por la falta de experiencia financiera. “Planing”, tiene como meta concientizar a la población al momento de usar productos financieros de forma inteligente, teniendo a la mano una aplicación que pueda realizar simulaciones de compras.

Capítulo 5

Objetivos

5.1. Objetivo General.

Desarrollar una aplicación móvil para sistema operativo ANDROID que permita el ingreso de gastos mensuales, posea la funcionalidad de realizar la simulación de compras con tarjetas de crédito y créditos, para lograr llevar el control de gastos y egresos, dirigido a población entre 20 y 28 años.

5.2. Objetivos Específicos.

- Crear una aplicación móvil con interfaces intuitivas dentro de los dos semestres de duración de la Especialización en la Universidad.
- Crear el módulo de ingresos durante la primera fase del ciclo de desarrollo.
- Creación del módulo de Egresos durante la segunda fase del ciclo de desarrollo.
- Desarrollar un simulador de compras con ayuda contextual de términos financieros para la tercera fase del ciclo de desarrollo.
- Implementar el módulo de calificación de la aplicación que determine la experiencia de usuario con la misma al finalizar los ciclos de desarrollo.
- Realizar glosario de términos básicos en finanzas como módulo final del ciclo de desarrollo.

- Permitir al usuario observar un balance general por medio de gráfico estadístico, cuál es su estado según los datos ingresados en el mes actual en el que se encuentre el usuario.
- Permitir que la aplicación pueda operar, hacer simulaciones en línea, durante el desarrollo del proyecto, sin necesidad de estar conectado a la red de internet.

Capítulo 6

Metodología

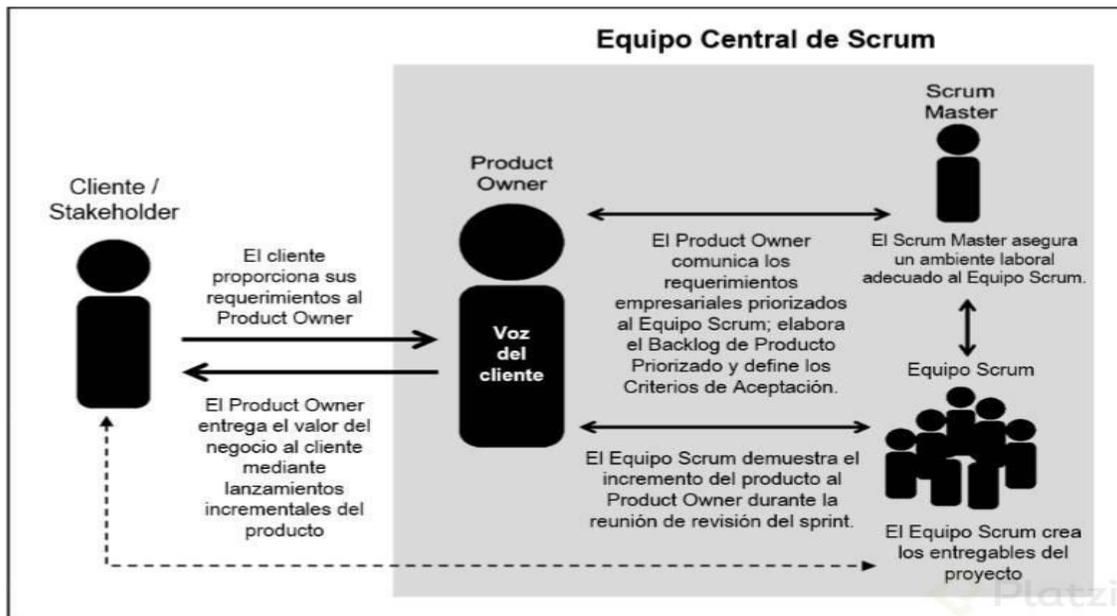
La metodología seleccionada por el equipo del proyecto es SCRUM. Esta metodología nos permite cumplir con el cronograma de actividades declaradas en el cronograma de trabajo, así como también cumplir con cada uno del sprint planteado para la etapa de Construcción. Adicional, para el tiempo de implementación corto que tenemos es la mejor opción.

6.1 Selección de la metodología.

Se ha seleccionado SCRUM entre otras metodologías ágiles, basado en varios factores técnico y productivo, las metodologías analizadas son AGILE, SCRUM y XP (Extreme Programming), la siguiente tabla comparativa nos muestra que de las tres metodologías analizadas Extreme Programming es la que más riesgo y complejidad representa, SCRUM, es un Marco de trabajo que simplifica la metodología Agile requiere menos funciones.

Ilustración 1

Descripción de procesos de la metodología



Fuente: <https://testingbares.com/2015/01/26/metodologias-agiles-scrum/>

6.2 Implementación de Metodología.

La metodología seleccionada, se implantara de la siguiente forma adaptando la metodología a la necesidad de nuestra aplicación:

- Inicio de Ciclo: El ciclo inicia con la reunión de interesados (stakeholders) en la que se crea la descripción de la visión del proyecto. Luego el product Owner crea la lista priorizada del producto (prioritized product backlog) que contiene la lista de los requerimientos en orden de prioridad para el negocio y el proyecto en forma de historias de usuario. Se realizarán entrevistas a los

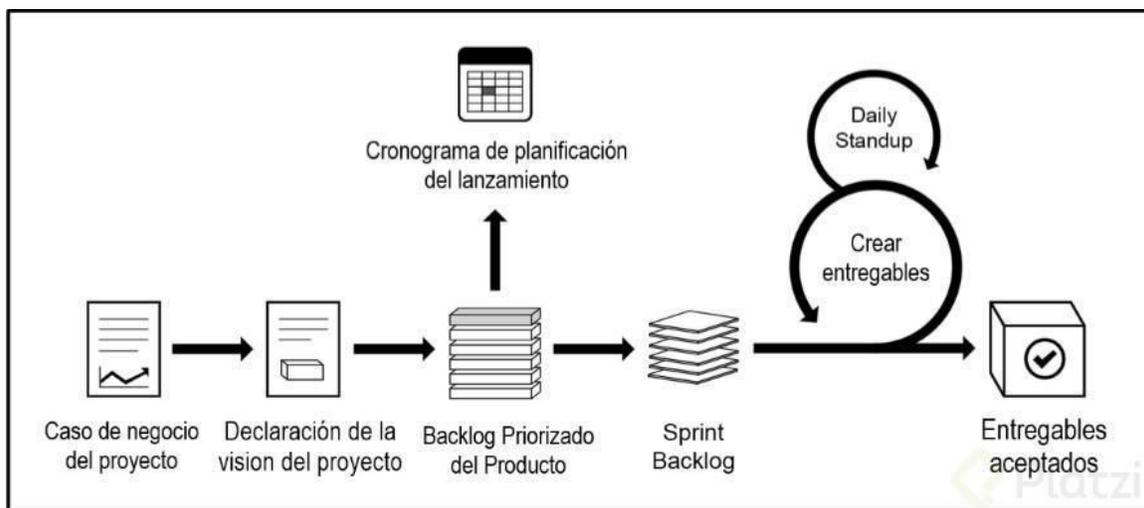
usuarios objetivos de nuestras empresas en donde laboramos actualmente, con el fin de recoger requerimientos funcionales que sean esperados por la aplicación.

- **Sprint:** Es una de las cajas de tiempo de SCRUM, tiene duración de 1 a 6 semanas en las que el equipo de SCRUM trabaja en la creación de los entregables, el Sprint inicia con la reunión de planeación del sprint (Sprint planning meeting) esta reunión tiene una duración de 8 horas para un sprint de 4 semanas, tiempo en el que se analizan las historias de usuario y de acuerdo con la prioridad se incluyen en el Sprint Backlog que es el listado de tareas que se van a implantar durante el sprint que inicia. Luego. Se realizará el desarrollo por cada módulo de la aplicación por sprints, asegurándonos que al final del desarrollo se realizará la correspondiente integración de los mismos con el nivel de calidad requerido y ya probado.
- **Daily Standup:** Durante el sprint se realizan reuniones diarias llamadas Daily Standup, durante 15 minutos máximo vía SKYPE u otro sistema de llamada vía internet, el equipo de SCRUM, Daniel Murcia y Edwin Rodríguez se reúnen para discutir el progreso diario y si hay impedimentos. Al final del sprint se realiza una reunión de revisión del Sprint (Sprint Review Meeting) en la que se hace una demostración de los entregables desarrollados al product Owner y a los stakeholders relevantes. Entregables:

- Luego de cada sprint se reunirá el equipo SCRUM para validar si el entregable cumple con los requisitos y requerimientos diseñados si esto es correcto se elaborarán lecciones aprendidas para implantar en un nuevo sprint y así hasta lograr la finalización del proyecto.

Ilustración 2

Proceso Metodología SCRUM



Fuente: <https://testingbares.com/2015/01/26/metodologias-agiles-scrum/>

Capítulo 7

Proceso de Software

7.1. Requerimientos Funcionales.

- El sistema permitirá al usuario iniciar sesión con un usuario y contraseña previamente creada o por medio de una cuenta GOOGLE+ o FACEBOOK.
- El usuario podrá visualizar su balance general por medio de un gráfico estadístico que muestre porcentajes visibles.
- El sistema permite ingresar los ingresos que percibe el usuario registrado en valores enteros y positivos.
- El usuario permite ingresar los tipos de egresos que el usuario registrado pueda tener.
- El sistema permite realizar la simulación de compras con medios de pago por crédito, el usuario puede ingresar datos específicos de la compra como la tasa de interés, efectiva mensual o anual, número de cuotas para compras con tarjetas de crédito, seguro que deba pagar la tarjeta o la utilización del crédito bancario.
- El sistema permite al usuario conocer la descripción de cada término financiero por medio de una ayuda contextual, con el fin que sepa cómo se calcula su compra.

7.2. Requerimientos No Funcionales.

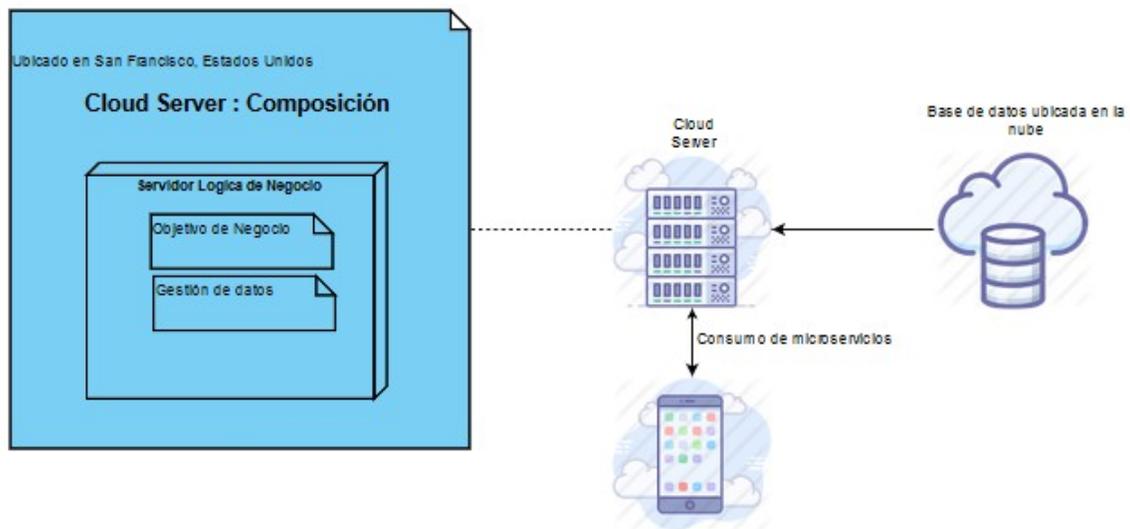
- El sistema controlará el acceso a la aplicación y lo permitirá solamente a usuarios autorizados.
- El sistema estará diseñado para sistema operativo ANDROID.
- El sistema deberá soportar los 50 usuarios iniciales conectados al tiempo, realizando simulaciones de compras.
- El sistema deberá estar diseñado para soportar una escalabilidad, para cuando la aplicación sea publicada en el PLAY STORE.

7.3 Diseño y arquitectura

7.3.1 Diagrama de despliegue.

Ilustración 3

Diagrama de despliegue

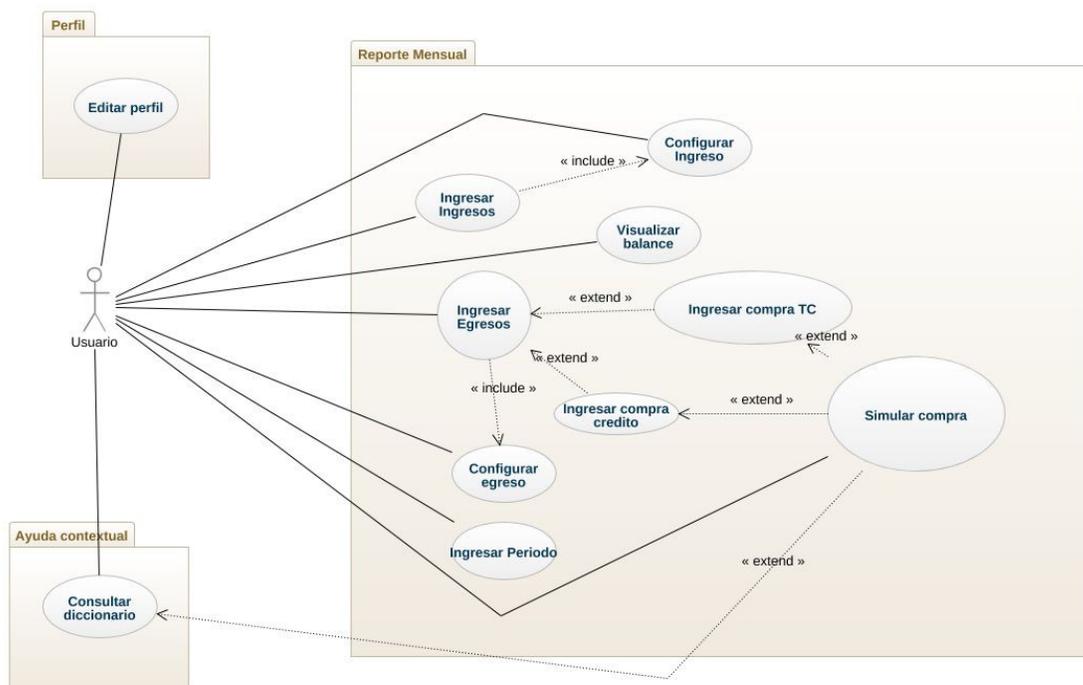


Fuente: Elaboración Propia

7.3.2 Diagrama Casos de Uso.

Ilustración 4

Diagrama de casos de uso



Fuente: Elaboración Propia

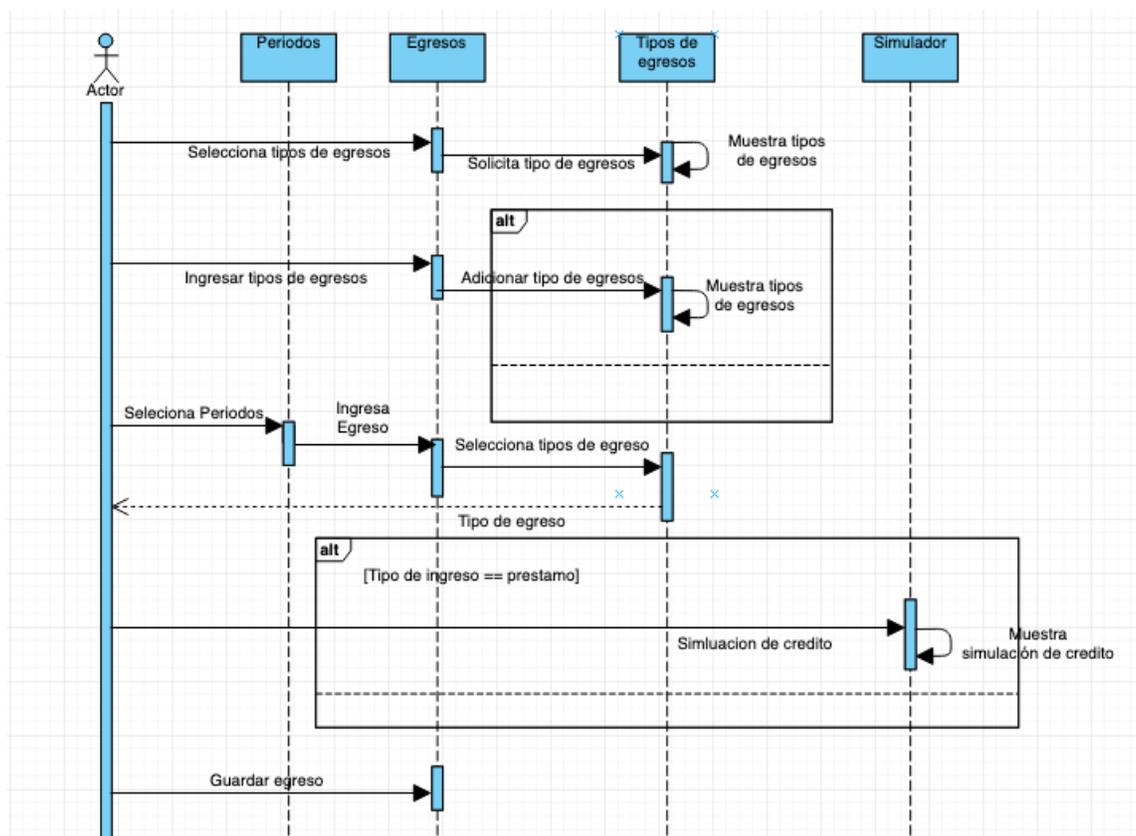
7.3.2.1 Caso de uso Ingresar Egresos.

Ver Anexo 1.

7.3.3 Diagrama de Secuencia

Ilustración 5

Diagrama de Secuencia

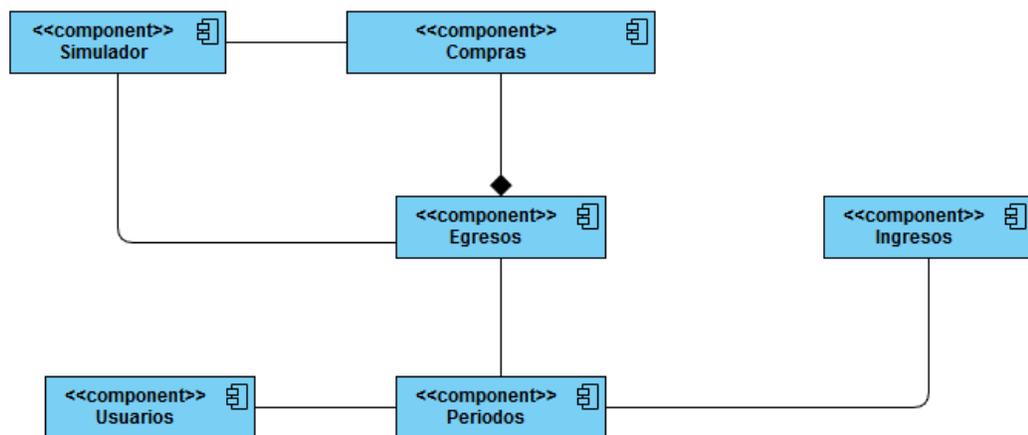


Fuente: Elaboración Propia

7.3.4 Arquitectura de Alto Nivel

Ilustración 6

Arquitectura de Alto Nivel

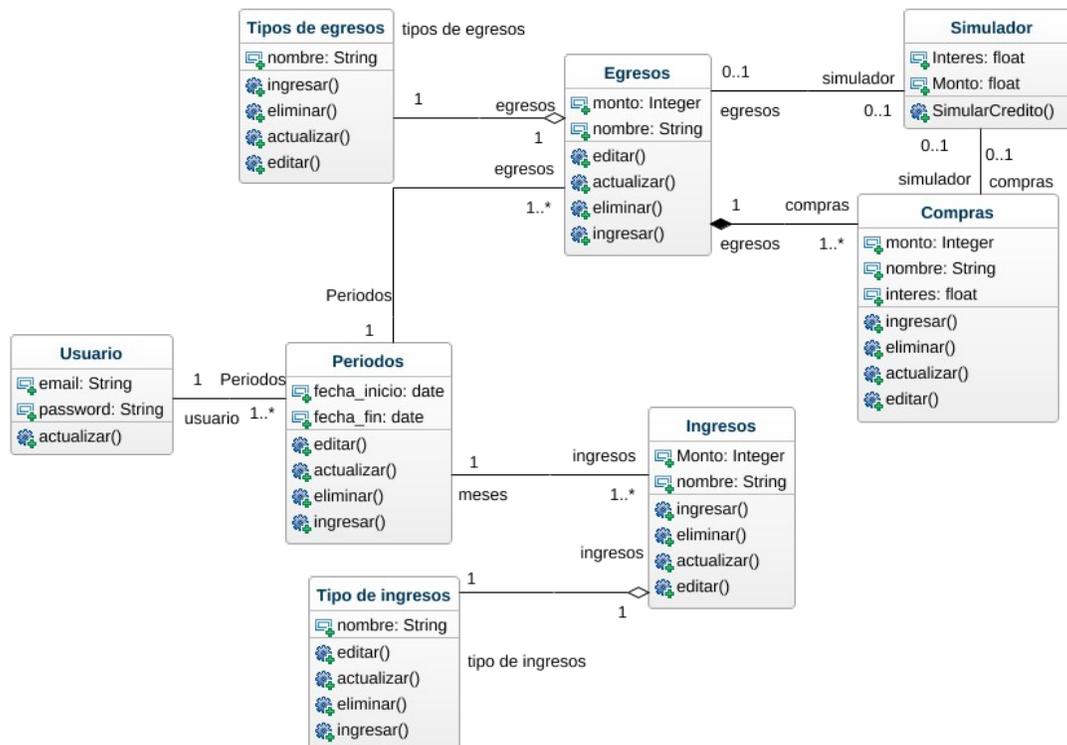


Fuente: Elaboración Propia

7.3.5 Diagrama de Clases

Ilustración 7

Diagrama de Clases

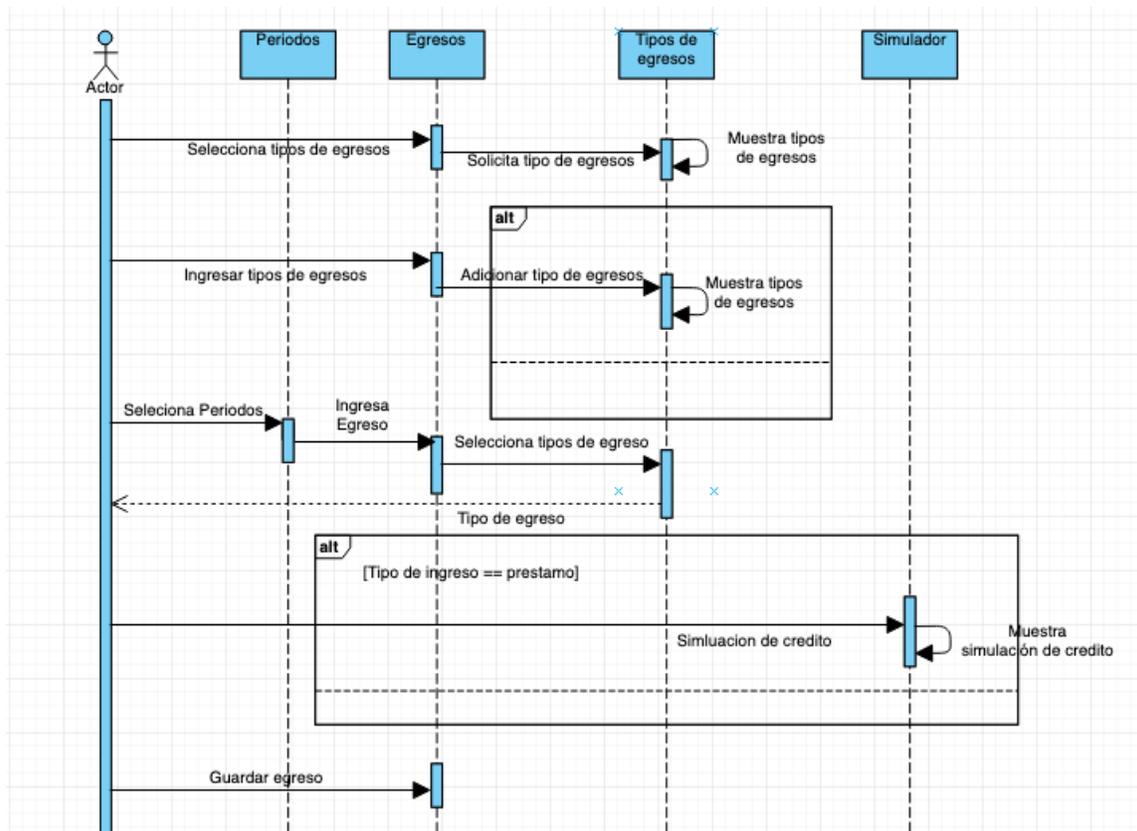


Fuente: Elaboración Propia

7.3.6 Diagrama de Clases

Ilustración 8

Diagrama de clases



Fuente: Elaboración Propia

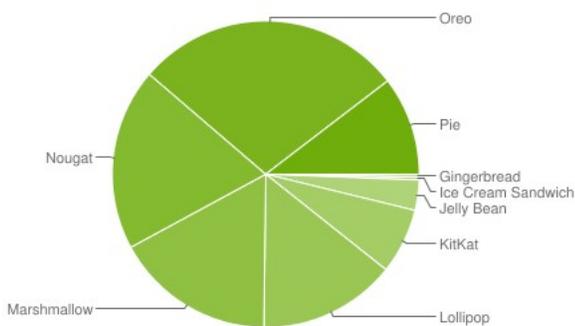
7.4 Construcción

Planning es una aplicación móvil que necesita de una base de datos para realizar la traza datos para su funcionalidad, además de ello puede ser accedido desde múltiples dispositivos lo cual hace necesario una base de datos, por tanto se elige una base de datos relacional como lo es mysql, que si bien es una de las más usada también posee la ventaja de que correrá bajo una licencia pública general (gratuita).

La aplicación se correrá solamente en dispositivos android que su sistema operativo sea superior a la versión de Android 4.1 Jelly Bean, aquellos sistemas operativos por debajo de esta versión no serán tenidos en cuenta.

Ilustración 9

Compatibilidad con versiones de plataforma diferentes.



Datos recopilados durante un período de 7 días hasta el May 1, 2020.

No se muestran versiones con una distribución inferior al 0.1%.

Fuente: <https://developer.android.com/about/dashboards>

Dadas estas circunstancias se decide que la mejor tecnología para desarrollar esta aplicación es React Native , es un framework desarrollado por facebook de código abierto y licencia abierta desarrollado en javascript pero se transcribe a código nativo en dispositivos android y despliega la aplicación de forma nativa real (no simulada).

Ahora bien se necesita “un puente” un sistema de comunicación entre la aplicación y la base datos y además estandarizar la arquitectura del mismo. Por tanto se decidió una arquitectura basada en micro servicios debido a que estas arquitecturas poseen ventajas como una aplicación lista para comercializarse más rápidamente, una mayor facilidad de implementación, Accesibilidad debido a que las funcionalidades están divididas en pequeñas piezas fáciles de comprender, actualizar y mejorar fácilmente.

Una vez definida la arquitectura, se establece que el protocolo de comunicación a usar es Transferencia de Estado Representacional (REST) debido a que fácilmente se puede separar la comunicación cliente - servidor siendo el cliente la aplicación móvil ,además de ello es independiente del tipo de plataformas o lenguajes lo cual un elemento que se acopla muy bien con unas de las ventajas de la arquitectura basada en micro servicios.

Los micro servicios deben tener una comunicación REST y pueden ser multiplataforma, dado esto, se considera que codeigniter un framework de PHP sería una de las soluciones más interesantes para manejar el backend del mismo. Codeigniter definido por la empresa que desarrollo el framework como : “Su objetivo es permitir que los desarrolladores

puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas.” *Fuente del comentario:* https://web.archive.org/web/20141006100806/https://ellislab.com/codeigniter/user-guide/overview/at_a_glance.html/ .

Además de ello es un framework rápido, escalable, ordenado y de fácil despliegue.

Con respecto a la infraestructura será desplegada en la nube los micro servicios son 5, Egresos, ingresos, periodos, ahorros, usuarios, todos funcionan de forma independiente, todos desarrollados en codeigniter y desplegados en una Droplet, una Droplet es la forma como denomina a una instancia en DigitalOcean que funciona de forma muy similar a un EC2 de Amazon Web Services solo que a menor precio. Digital Ocean es una empresa de computación en la nube de alto crecimiento con presencia en más de 10 países y con clientes como wework, nodeDB entre otros.

7.5 Pruebas

7.5.1 Propósito

Definir la estrategia y alcance que será cubierta durante el proceso de pruebas por parte de los estudiantes creadores del software Planning. Este plan incluye tipos de pruebas a realizar, documentación que se tomará como insumo, requerimientos a ser probados según el alcance definido, metodología, herramientas y criterios de aceptación. Se verificará la hipótesis planteada al comienzo del proyecto, verificar y comprobar que el software almacena la información de manera adecuada y correcta en la base de datos.

7.5.2 Definiciones

Ilustración 10

Definiciones y términos de pruebas de software

Término	Definición
Caso de prueba	Condiciones de ejecución y salidas esperadas de un ítem de prueba.
Criterio de Aceptación	Condición que determina si un entregable está listo para ser utilizado en el ambiente de producción.
Defecto	Inconsistencia de un entregable encontrado durante la etapa de pruebas.
Entregable	Resultado tangible (documento plan de pruebas, ejecución casos de prueba, software, etc.) de una tarea

	completada.
Incidencia	Defecto, Pregunta o Sugerencia reportada.

Fuente: Elaboración Propia

7.5.3 Alcance General de pruebas

El plan de pruebas propuesto en este documento, abarca las pruebas que se realizarán al proyecto Planning. Teniendo en cuenta que las pruebas se realizarán bajo un ambiente de desarrollo. Las pruebas presentadas se realizarán en paralelo a los desarrollos realizados. El plan de pruebas cuenta con un set de casos de pruebas referentes a los casos de uso relevantes realizados con los Stakeholders, también, se cuenta con casos de prueba que hacen referencia a las funcionalidades del sistema. Además de resolver las incidencias que pudieran presentarse al momento de realizar las validaciones por los Ingenieros responsables del proyecto.

7.5.4 Tipos de pruebas a ejecutar y su alcance

Los siguientes son los tipos de pruebas que hacen parte del alcance de este plan de pruebas.

Ilustración 11

Estrategia de pruebas

Nombre de la prueba:	Pruebas Funcionales
Objetivo de la prueba:	Se asegura el trabajo apropiado de los requisitos funcionales, entrada de datos, procesamiento, obtención de resultados, actualizaciones a las estructuras de datos y que los eventos de base de datos se ejecutaron en forma correcta. También involucra pruebas de usabilidad y facilidad de uso del sistema.
Alcance de la prueba:	Se validarán los flujos y campos básicos con que cuenta la herramienta.
Insumos a utilizar:	Se tomaran como insumos los casos de uso levantados para el proyecto.
Estrategia a utilizar:	Se validarán los flujos básicos de los casos de uso y seguidamente se procederá a mirar los flujos alternos, ingresando datos inválidos para comprobar la validación que realiza el software.
Estrategia para generación de datos de prueba:	Se utilizarán datos aleatorios para el caso de las pruebas, se ingresaran fotografías tomadas por la cámara del sistema y fotos previamente creadas para verificar efectividad de reconocimiento de caracteres.
Herramientas a	N/A

utilizar:	
------------------	--

Fuente: Elaboración propia

7.5.5 Tipos de pruebas

Se implementaran las pruebas de caja negra al software Planning, se generara un reporte de las pruebas y su respectivo caso de prueba para el caso de uso relevante.

7.4.1 Instalación y configuración

7.4.1.1 Despliegue

Antes de realizar el despliegue se especifica que el sitio necesita de apache en un versión 2.4.49 en adelante para PHP, codeigniter, para el normal funcionamiento de este se recomienda que se use una versión de php 5.6 o más reciente.

Para desplegar solo se debe ubicar los repositorios completos dentro de la carpeta contenedora de esta manera.

Ilustración 12

Repositorios

Nombre de archivo	Tamaño d...	Tipo de arc...	Última modific...	Permisos	Propietario...
..					
Expense		Carpeta de...	9/05/2020 8:53:...	drwxr-xr-x	root root
Income		Carpeta de...	9/05/2020 9:09:...	drwxr-xr-x	root root
Periods		Carpeta de...	2/05/2020 8:48:...	drwxr-xr-x	root root
Saves		Carpeta de...	9/05/2020 8:16:...	drwxr-xr-x	root root
Users		Carpeta de...	23/04/2020 9:5...	drwxr-xr-x	root root
.htaccess	488	Archivo HT...	23/04/2020 8:1...	-rw-r--r--	root root
index.html	193	Archivo HT...	23/04/2020 8:1...	-rw-r--r--	root root

Fuente: Elaboración propia

- Editar archivos de configuración base de datos.

Ubicando la carpeta application/config se encuentran los archivos de configuración entre ellos database.php, este archivo gestiona la conexión a la base de datos. Aca se ingresan las credenciales de forma sencilla.

Ilustración 13

Configuración base de datos

```
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => 'root',  
    'database' => 'db_escueladirectores',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
    'cache_on' => FALSE,  
    'cachedir' => '',  
    'char_set' => 'utf8',  
    'dbcollat' => 'utf8_general_ci',  
    'swap_pre' => '',  
    'encrypt' => FALSE,  
    'compress' => FALSE,  
    'stricton' => FALSE,  
    'failover' => array(),  
    'save_queries' => TRUE  
);
```

Fuente: Elaboración propia

- Configuración url

Ubicando la carpeta application/config se encuentran los archivos de configuración entre ellos config.php, aca principalmente se configura el base_url, consiste en ubicar la url del sitio.

Conclusiones

Cumplir con los objetivos planteados.

Aplicar cada una de las materias vistas durante la especialización a la APP que se desarrollaba en paralelo.

Lograr afianzar el método de aprendizaje, implementando en la APP cada concepto teórico visto.

Implementar una arquitectura de software a la APP.

Por medio de un muestreo con el uso de la herramienta por 20 personas logramos identificar que la interfaz es intuitiva y los usuarios no tuvieron necesidad de manuales o ayudas contextuales para su correcta operación.

Referencias Bibliográficas

Diccionario de Economía. (2018). Finanzas: qué es.

<https://www.eleconomista.es/diccionario-de-economia/finanzas>

Grupo Bancolombia (2019). Tipos de crédito.

<https://www.grupobancolombia.com/wps/portal/personas/aprender-es-facil/como-manejar-dinero/endeudamiento-responsable/tipos-credito>

Microsoft (2019). Estilo de arquitectura de micro servicios.

<https://docs.microsoft.com/es-es/azure/architecture/guide/architecture-styles/microservices>

Grupo BBVA (2018) . API Rest.

<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>.

Platzi (2017). Qué es SCRUM y los roles en SCRUM

https://platzi.com/blog/que-es-scrum-y-los-roles-en-scrum/?utm_source=google&utm_medium=paid&utm_campaign=intention&utm_content=&adgroup=71299386940&gclid=EA1aIQobChMli-DAv-nQ5AIVF4rICh352AHzEAAyAAEgIuvfD_BwE

Wikipedia (2008). MySQL.

<https://es.wikipedia.org/wiki/MySQL>

Anexos

1. Caso de uso arquitectural relevante.
2. SDL - Microsoft – Planning (Microsoft Security Development Lifecycle)
3. Visión general del proyecto Caso de Negocio.