

Universidad de Lima
Facultad de Ingeniería y Arquitectura
Carrera de Ingeniería de Sistemas



APLICACIÓN CARPOOLING PARA EL TRANSPORTE DE PERSONAS QUE TRABAJAN EN EL CENTRO EMPRESARIAL DE SAN ISIDRO

Trabajo de suficiencia profesional para optar el Título Profesional de Ingeniero de
Sistemas

Ricardo Demetrio Amaro Meza
Código 20072124

Asesor

Edwin David Huaynate Mato

Lima – Perú
Setiembre de 2019





**CARPOOLING APPLICATION FOR THE
TRANSPORTATION OF PERSONS
WORKING IN THE SAN ISIDRO BUSINESS
CENTER**

TABLA DE CONTENIDO

RESUMEN	IX
ABSTRACT	X
CAPÍTULO I: INTRODUCCIÓN	1
CAPÍTULO II: CONCEPTOS Y FUNDAMENTOS	3
2.1. Carpooling	3
2.2. Cloud computing	3
2.2.1. Arquitectura serverless	5
2.2.2. Aws lambda	6
2.2.3. Amazon api gateway	7
2.3. Arquitectura de software	9
2.4. Aplicación móvil	10
2.4.1. Lenguaje de programación java.....	11
2.4.1.1. Javascript	12
2.4.2. Sistema operativo android	13
2.4.3. Base de datos mysql	14
2.4.4. Gitlab	15
2.5. Pasarela de pagos.....	16
CAPÍTULO III: FUNDAMENTACIÓN DEL PROYECTO.....	18
1.1. Fundamentación de la deseabilidad del proyecto.....	18
1.2. Fundamentación de la factibilidad del proyecto.....	22
1.3. Fundamentación de la viabilidad técnica	25
CAPÍTULO IV: DEFINICIÓN DEL PROYECTO.....	32
2.1. Objetivo general	34
2.2. Objetivos específicos.....	34
2.3. Diseño propuesta de valor	34
2.4. Modelo de negocio propuesto.	35
2.5. Pseudocódigo.....	36
2.6. Historias de usuarios	38
2.7. Diagramas de arquitectura de software	39
2.8. Modelo de datos de la aplicación	40
CAPÍTULO V: DESARROLLO DEL PROTOTIPO.....	42

3.1. Paso 1: empatizar.....	42
3.2. Paso 2: definir.....	42
3.3. Paso 3: idear	43
3.4. Paso 4: construir el prototipo.....	44
3.5. Paso 5: evaluar.....	53
CONCLUSIONES.....	54
RECOMENDACIONES	55
REFERENCIAS	56
BIBLIOGRAFÍA	59
ANEXOS	60



INDICE DE TABLAS

Tabla 1.2-1 Costos servicios AWS	24
Tabla 1.2-2 Tabla de costos y flujos de ingresos y egresos	25
Tabla 3.3-1 Lluvia de ideas, Design Thinking - Idear	43



INDICE DE FIGURAS

Figura 2.2.1	Arquitectura del Cloud Computing	4
Figura 2.2.1.1	Arquitectura Serverless	6
Figura 2.2.2.1	How it works - AWS Lambda	7
Figura 2.2.3.1	Funcionamiento de API Gateway.....	8
Figura 2.3.1	Componentes de una Arquitectura de software (J2EE).....	10
Figura 2.4.1.1	Java EE Platform APIs	12
Figura 2.4.2.1	Cuota de mercado del sistema operativo móvil en todo el mundo.....	14
Figura 2.4.4.1	Como se trabaja en un flujo Git.....	16
Figura 2.5.1	Como funciona una pasarela de pagos	17
Figura 1.1.1	The World's Worst Cities For Traffic Congestion	18
Figura 1.1.2	Los paraderos abarrotados del corredor Javier Prado	19
Figura 1.1.3	Tarifa dinámica de UBER en determinadas horas	20
Figura 1.1.4	Sector Financiero de San Isidro (en verde)	21
Figura 1.1.5	Mobile & Tablet Vendor Market Share Peru	21
Figura 1.2.1	Ingreso aproximado Te Jalo	23
Figura 1.3.1	Servicio de geolocalización de Google Maps	26
Figura 1.3.2	Amazon Web Services Arquitectura Serverless.....	27
Figura 1.3.3	Costos Solicitudes Lambdas.....	28
Figura 1.3.4	Arquitectura Node Js y Lambda.....	28
Figura 1.3.5	Culqi, pasarela de pagos online	30
Figura 2.1	Diagrama de procesos del proyecto.....	33
Figura 2.7.1	Diagrama de Aplicaciones.....	39
Figura 2.7.2	Diagrama de despliegue	39
Figura 2.7.3	Diagrama de Componentes	40
Figura 2.8.1	Modelo de datos del Proyecto Te Jalo.....	41
Figura 3.2.1	Canvas de propuesta de valor	43

ÍNDICE DE ANEXOS

Anexo 1 - Encuesta a los Trabajadores de San Isidro.....	61
--	----



RESUMEN

Actualmente el transporte en el distrito de San Isidro, Lima, y como en muchas otras ciudades, tiene muchos problemas e inconvenientes para aquellos trabajadores que desean moverse de manera cómoda y segura sobre todo en horas pico donde el tráfico empeora, además de lidiar con la problemática del transporte público, los elevados precios de taxis, la falta de seguridad en los colectivos o taxis, entre otros problemas, hace que empeore la situación de transporte para estas personas.

La solución que se propone es la implementación de una aplicación móvil basada en el carpooling y enfocada en las personas que trabajan en el centro empresarial de San Isidro, que quieran trasladarse desde sus centros de trabajo hacia sus hogares y que puedan conectar con otras personas que se movilizan en sus vehículos particulares y deseen generar un ingreso extra llevando a sus colegas de trabajo que compartan similar o misma ruta de retorno, y con ello poder reducir los tiempos y costos de viaje, transportándose de forma cómoda, segura y confiable, así como también reduciendo en cierto grado la huella de carbono.

Palabras Clave

Carpooling, Empresas, Aplicación móvil, Trabajadores, Distrito, Seguridad

ABSTRACT

Currently, transportation in San Isidro district in Lima, and as in many other cities, has many problems and inconveniences for those workers who want to travel comfortably and safely, especially at peak times where traffic increases, in addition to dealing with the Public transport problems, high taxis fares, lack of security in buses or taxis, among other problems, makes the transport situation worse for these people.

The solution proposed is the implementation of a mobile application based on carpooling and focused on people who work in the San Isidro Business Center, who want to mobilize from their homes and can connect with other people travelling in their private vehicles and wish to generate extra income by taking their work colleagues who share same return route, and thereby be able to reduce travel times and costs, transporting them comfortably, safely and reliably, as well as also reducing to a certain extent the carbon footprint.

Keywords

Carpooling, Business, Mobile App, Workers, District, Security

CAPÍTULO I: INTRODUCCIÓN

Actualmente en el distrito de San Isidro, en donde trabajan varias personas, existe muchas dificultades para el desplazamiento en toda la ciudad, este se convierte en todo un problema sobre todo para aquellas personas que salen a trabajar y llega el momento de retornar a sus hogares, debido al caos vehicular que existe actualmente, a las horas punta, el aumento continuo del parque automotor, pistas mal planificadas, obras en desarrollo, transporte no seguro, la inseguridad entre otros. En la actualidad Lima en Perú ocupa el tercer lugar en el mundo con un 57% de congestión vehicular "Un aspecto positivo de la congestión es que es un indicador de una economía global fuerte. Sin embargo, viene con un impacto ambiental masivo y cuesta a los viajeros grandes cantidades de tiempo" (Gestión, 2019). En promedio, un usuario pierde hasta 4 horas diarias en el tráfico de Lima. Según Aldo Bravo, experto en ingeniería de tránsito de la UPC, a lo largo de toda la vida de un usuario, las personas pueden pasar en el tráfico 12 años (Gestión, 2018).

De acuerdo con la policía de tránsito, San Isidro es uno de los puntos críticos en donde el tráfico se vuelve caótico sobretodo en horas punta ya que en esta existen centros empresariales grandes en donde el flujo de trabajadores es enorme. Existen soluciones que pueden atacar ciertos puntos negativos, como lo son el aumento en los carriles en avenidas principales, mejorar el transporte público o reduciendo el parque automotor quitando a los vehículos con determinada fecha de antigüedad, pero esto no ataca todos los problemas propuestos y que en cambio pueden afectar negativamente con el medioambiente. La antigüedad promedio del parque automotor avanza en la ruta equivocada. Hoy es de 15 a más años, y es que al año solo se renueva en 6%, cuando debería hacerse en 10%. Hay 12 personas por cada vehículo, en Colombia son 10, en Venezuela y Ecuador son 7, en Chile, Argentina, Brasil y México, son entre 3 y 3.7 afirmó el presidente de la Asociación Automotriz del Perú (AAP), Edwin Derteano (Gestión, 2019).

La congestión vehicular, conocida también como «resaca vehicular», ocurre tanto a nivel urbano como a nivel interurbano. Se trata de las condiciones de un flujo vehicular que se ve saturado debido al exceso de demanda de las vías, produciendo que los viajes sean hartamente largos y engorrosos. Este fenómeno se produce comúnmente en las horas pico y resulta frustrante para los automovilistas, ya que deriva en pérdidas de tiempo y deterioro del medioambiente, por la cantidad de esmog que expulsan los escapes de vehículos a costa del consumo de combustible (Riquelme, 2018).

Con el proyecto propuesto se busca brindar una solución y mejorar la forma en como las personas se trasladan desde sus centros de trabajo hacia sus domicilios de forma cómoda y segura, todo esto basado en el carpooling.

En una ciudad inteligente, los esfuerzos para reducir el consumo de energía se centran no solo en apoyar el desarrollo de sistemas de redes inteligentes o usuarios, sino también en reducir el tráfico. (Olszewski, Pałka, & Turek, 2018).

Las personas del centro empresarial de San Isidro que mediante el uso de una aplicación móvil pueden conectarse con aquellas personas que tengan similar ruta de retorno y así viajar de forma más efectiva, cómoda, confiable. En una ciudad donde cada día crece el parque automotor haciendo que en las horas punta el tráfico se vuelva cada vez más caótico es necesario soluciones que ayuden a mejorar la calidad de vida de las personas y ayudar a reducir la huella de carbono en lo posible.

La congestión del tráfico a menudo incurre en problemas ambientales. Una de las formas más efectivas de mitigar esto es el transporte compartido, que reduce sustancialmente las demandas de los automóviles. Debido a la popularización de los teléfonos inteligentes y las aplicaciones móviles, se puede acceder cómodamente a un servicio de transporte compartido a través del sistema inteligente de uso compartido del vehículo (Jiau & Huang, 2019).

CAPÍTULO II: CONCEPTOS Y FUNDAMENTOS

2.1. Carpooling

Carpooling es un sistema a través del cual los usuarios con rutas similares pueden usar un automóvil. Su objetivo principal es hacer coincidir a las personas que viajan diariamente al trabajo; por lo tanto, compartir el vehículo puede ser un método efectivo para aliviar los atascos de tráfico durante las horas pico (Olszewski, Pałka, & Turek, 2018).

El modelo de carpooling es la práctica dónde un conductor comparte su automóvil con otras personas con el objetivo de cumplir un viaje puntual para todos los pasajeros. El carpooling no es un nuevo método de transporte y se ha visto desarrollado en Europa por varios años. Compañías europeas y estadounidenses en particular, emplean sistemas para que sus empleados compartan el mismo vehículo en el momento de ingresar a su trabajo, y el auto compartido es incentivado con carriles exclusivos para personas que utilizan este sistema. Sin embargo, el rápido incremento de la población y el desarrollo estático de estas opciones de movilidad, no ha generado mejoras significativas desde su introducción.

Entre otros beneficios del auto compartido está específicamente el interés de descongestionar el tráfico donde se implementa, ya sea el caso en que los dueños de autos particulares se conviertan en pasajeros de otros vehículos o reduciendo la cantidad de gente circulando dentro del transporte público, mejorando la cooperación entre comunidades. Existen beneficios ambientales porque al remover autos de las calles algunos sectores tienen la oportunidad de reducir su huella de carbono. La movilidad o el transporte de personas es, junto a los alojamientos, el sector con más actividad y mayores volúmenes del consumo colaborativo (Vargas, Gaetan, & Saldaño, 2018).

2.2. Cloud Computing

La computación en la nube es un modelo para permitir el acceso a la red global, conveniente y bajo demanda a un conjunto de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que se puede aprovisionar y lanzar rápidamente con un mínimo esfuerzo de gestión o interacción del proveedor de servicios. (Mell & Grance, 2019)

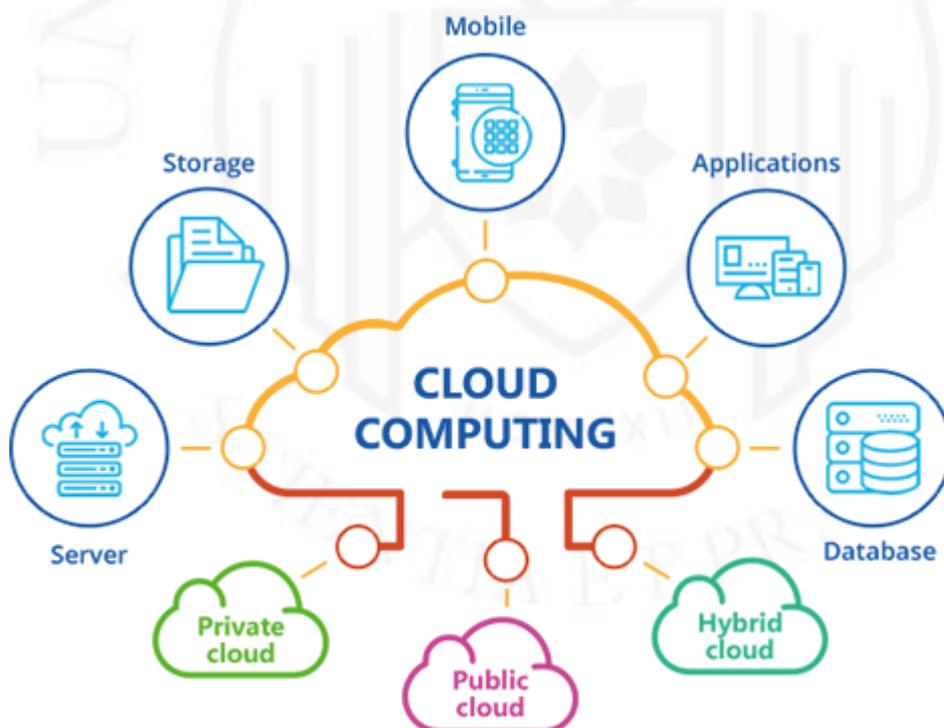
Casi todas las soluciones de TI están etiquetadas con el término computación en la nube o solo nube en la actualidad. Las palabras de moda como esta pueden ayudar a

las ventas, pero es difícil trabajar con ellas en un libro. Así que, en aras de la claridad, definamos algunos términos. La computación en la nube, o la nube, es una metáfora del suministro y consumo de recursos TI. Los recursos de TI en la nube no son directamente visibles para el usuario; existen capas de abstracción en el medio. El nivel de abstracción que ofrece la nube varía, desde ofrecer máquinas virtuales (VM) hasta proporcionar software como servicio (SaaS) en sistemas complejos distribuidos. Los recursos están disponibles bajo demanda en enormes cantidades, y usted paga por lo que usa. (Wittig & Wittig, 2018)

Cloud computing ofrece servicios de computación a todo nivel ya sean aplicaciones, espacio en disco, servidores, servicios para dispositivos móviles, base de datos entre otros, y dependiendo del proveedor (ya se Amazon Web Services, Azure, Google Cloud, u otro), esto permite menor esfuerzo y costo a la hora de implementar alguna solución deseada.

Figura 2.2.1

Arquitectura del Cloud Computing



Fuente: IT Travel Services (2014)

2.2.1. Arquitectura Serverless

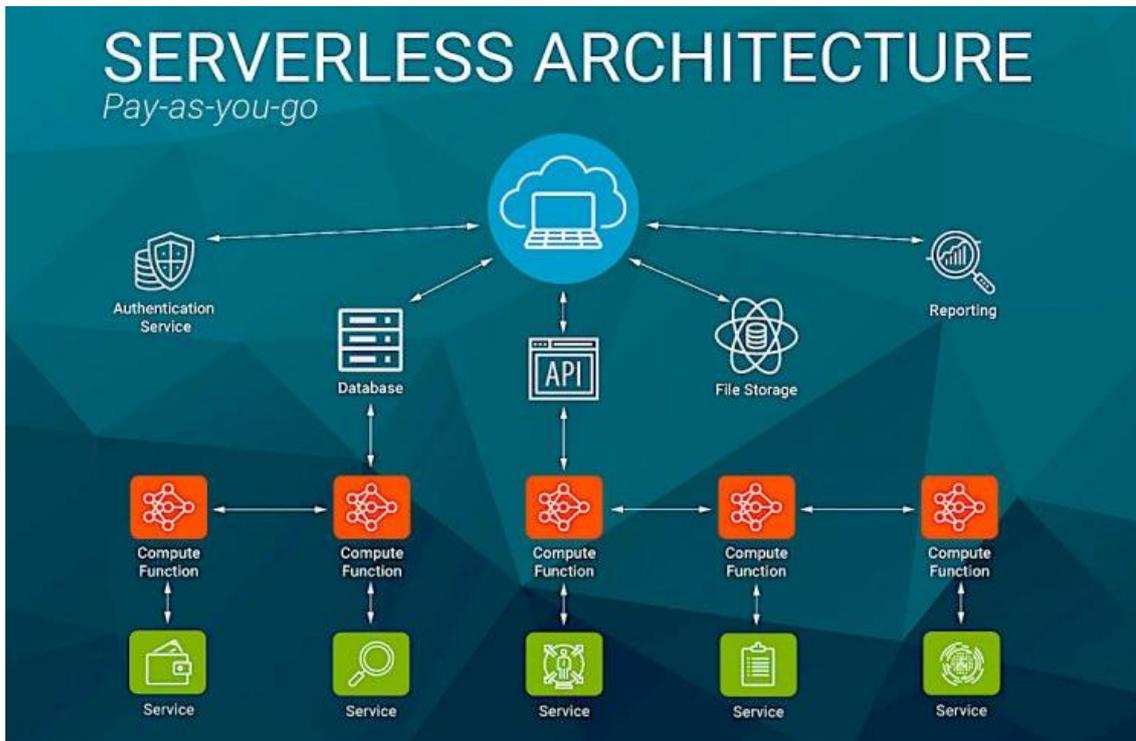
La informática sin servidor ha surgido recientemente como un nuevo paradigma en la virtualización y ya ha cambiado significativamente la economía de computación en la nube. Se considera una alternativa de baja latencia, eficiente en el uso de recursos y de rápida implementación a los enfoques de virtualización adicionales, como los basados en máquinas virtuales y contenedores. La informática sin servidor proporciona escalabilidad y reducción de costos, sin requerir ninguna sobrecarga de configuración adicional por parte del desarrollador. En este artículo, exploramos y analizamos cómo la informática sin servidor la tecnología puede ayudar a construir redes adaptables y escalables y mostrar los posibles peligros de hacerlo. (Aditya, y otros, 2019)

Serverless se basa en otro concepto antiguo: la contenedorización. Los contenedores son versiones simplificadas de máquinas virtuales, proporcionando un entorno dentro del cual una pieza de software pueda correr. Brenner llama a un contenedor "un sandbox para software" que no da al usuario acceso al hardware de la computadora. Lanzar una máquina virtual significa cargar en el sistema operativo todas las librerías necesarias, y el proceso puede tomar minutos, un contenedor puede ser lanzado en menos de un segundo, y el código se copió en menos de un segundo, por lo que está funcionando rápidamente. (Savage, 2018)

La computación en la nube es un servicio totalmente administrado por terceros y ya que esta arquitectura básicamente está controlada por eventos en servicios disponibles por el proveedor como espacio en disco, base de datos, autenticación, reportes, etc. Estos servicios resultan siendo escalables y solo utilizan recursos cuando ocurre un evento específico.

Figura 2.2.1.1

Arquitectura Serverless



Fuente: G2 Crowd Inc. (2015)

2.2.2. AWS Lambda

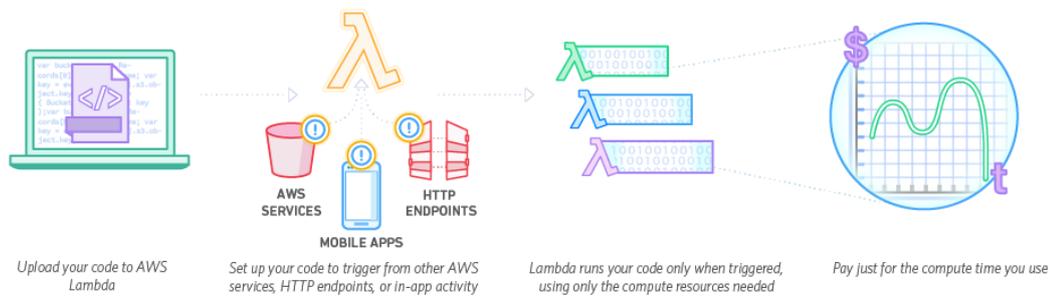
AWS Lambda permite construir código e implementarlo sin necesidad de configurar o administrar servidores subyacentes. Su unidad de despliegue es su código; no el contenedor que aloja el código o el servidor que ejecuta el código, sino simplemente el propio código. Desde el punto de vista de la productividad, hay beneficios obvios al descargar los detalles de dónde se almacena el código y cómo se gestiona el entorno de ejecución. La computación sin servidor también tiene un precio basado en métricas de ejecución, por lo que también hay una ventaja financiera. (Haines, 2017)

AWS Lambda es un servicio informático que le permite ejecutar código sin aprovisionar ni administrar servidores, ejecuta su código solo cuando es necesario y escala automáticamente, a partir de algunas solicitudes por día a miles por segundo. Paga solo por el tiempo de cómputo que consume: no hay cargo cuando su código no se está ejecutando. Con AWS Lambda, puede ejecutar código para prácticamente cualquier tipo de servicio de aplicación o backend, todo con cero administraciones. AWS Lambda ejecuta su código en una infraestructura informática de alta disponibilidad y realiza toda la administración de los recursos informáticos, incluido el mantenimiento del servidor y

del sistema operativo, el aprovisionamiento de capacidad y el escalado automático, monitoreo y registro de código. (AWS, 2019)

Figura 2.2.2.1

How it works - AWS Lambda



Fuente: Amazon Web Services (2017)

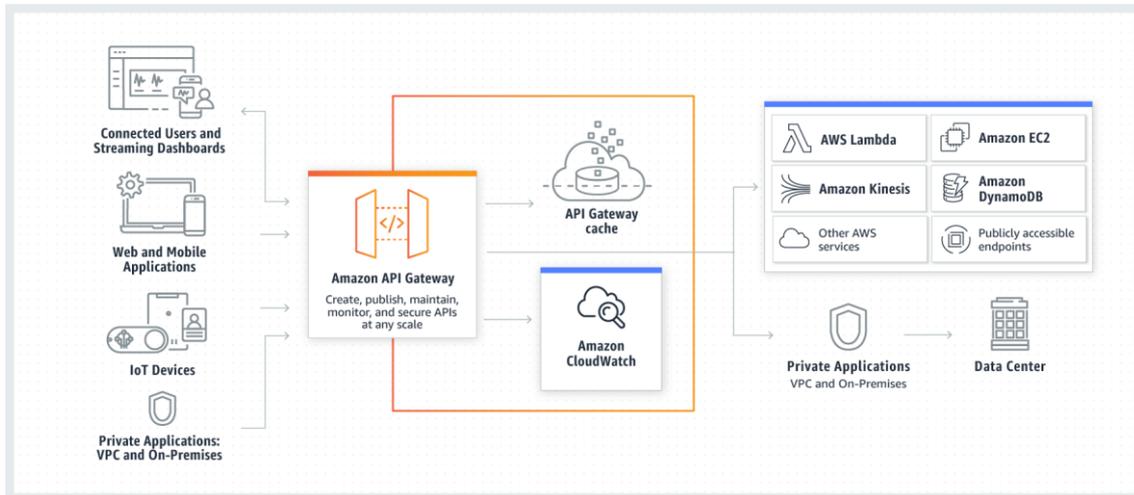
2.2.3. Amazon API Gateway

Es un servicio gestionado que facilita a los clientes de AWS crear, publicar, mantener, monitorear y proteger interfaces de programación de aplicaciones (API) a cualquier escala.

Permite a los clientes utilizar herramientas de seguridad de AWS familiares como AWS Gestión de Identidad y Acceso (IAM) para verificar y autenticar las solicitudes de API. Amazon API Gateway permite a las empresas ejecutar varias versiones de una API simultáneamente para que puedan desarrollar, implementar y probar nuevas versiones de sus API sin afectar las aplicaciones existentes. (NewsRx LLC, 2015)

Figura 2.2.3.1

Funcionamiento de API Gateway



Fuente: Amazon Web Services (2016)

La mayoría de las nuevas innovaciones están en la nube. La API Gateway es un ejemplo de esto, es un servicio paso a paso el cual permite a los usuarios desarrollar API's para sus propias aplicaciones para vincularlos con la infraestructura y los servicios de AWS. Las API's son críticas para escalar el software en la nube y son un gran negocio.

El desarrollo de API's, la seguridad y el mantenimiento continuo requieren mucho tiempo, por lo que Gateway se presenta en el momento adecuado. Se ocupará de gran parte del trabajo involucrado en la administración de API y aumenta la cantidad de aplicaciones que ahora se pueden integrar en la nube pública.

Desde la Cumbre de AWS en el Centro Javits, AWS anunció la disponibilidad general de tres productos anunciados anteriormente: el Catálogo de servicios de AWS, el servicio de entrega continua de AWS Code Pipeline y el repositorio de códigos de AWS CodeCommit.

También se anunció en la Cumbre el AWS Device Farm, un servicio de prueba que permitirá a los desarrolladores ejecutar sus aplicaciones, desde cualquier etapa de desarrollo, en una amplia variedad de teléfonos y obtener informes sobre fallas y métricas de rendimiento. El servicio está disponible solo para Android ahora, con una oferta de iOS en los próximos uno o dos meses, dijo el gerente de productos de AWS, Trent Peterson, en una sesión de presentación de AWS Device Farm en la Cumbre. Este servicio también estimulará el desarrollo porque las empresas pueden implementar

aplicaciones móviles más rápidamente y, con menos errores de compatibilidad que se envían con el software, también tienen menos usuarios insatisfechos. (Petersen, 2015)

2.3. Arquitectura de Software

Es el conjunto de decisiones significativas sobre la organización de un sistema de software que define los principios que guían el desarrollo, los componentes principales del sistema, sus responsabilidades y la forma en que se interrelacionan. La arquitectura de software es de especial importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema. Ejemplos de atributos de calidad son el desempeño, que tiene que ver con el tiempo de respuesta del sistema a las peticiones que se le hacen, la usabilidad, que tiene que ver con qué tan sencillo les resulta a los usuarios realizar operaciones con el sistema, o bien la modificabilidad, que tiene que ver con qué tan simple resulta introducir cambios en el sistema. (Reynoso, 2004)

La arquitectura no es el software operativo. Es una representación que permite: analizar la efectividad del diseño para cumplir los requerimientos establecidos, considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y reducir los riesgos asociados con la construcción del software. Esta definición pone el énfasis en el papel de los “componentes del software” en cualquier representación arquitectónica. En el contexto del diseño de la arquitectura, un componente del software puede ser algo tan simple como un módulo de programa o una clase orientada a objeto, pero también puede ampliarse para que incluya bases de datos y “middleware” que permitan la configuración de una red de clientes y servidores. Las propiedades de los componentes son aquellas características necesarias para entender cómo interactúan unos componentes con otros. En el nivel arquitectónico, no se especifican las propiedades internas (por ejemplo, detalles de un algoritmo). Las relaciones entre los componentes pueden ser tan simples como una invocación de procedimiento de un módulo a otro o tan complejos como un protocolo de acceso a una base de datos. (Pressman, 2010)

La arquitectura de software son las reglas, paradigmas, patrones que ayudan a construir, y probar una pieza de software. Es la experiencia práctica que se reduce al nivel abstracto. Arquitectura de software se basa en la ingeniería de sistemas y el método

A medida que aumenta el número de personas que poseen y usan teléfonos celulares, también lo hace el uso de teléfonos inteligentes. El 91% de la población adulta de los Estados Unidos posee actualmente un teléfono celular y de ese 91%, el 61% son teléfonos inteligentes.

Para ilustrar aún más la popularidad de los teléfonos inteligentes, se espera que se vendan mil millones en el próximo año, duplicando la cantidad de PC.

Con más de 10 mil millones de dispositivos de Internet móvil que se espera estén en uso para 2016, la industria de aplicaciones móviles crecerá enormemente para satisfacer la demanda y mantenerse al día con las tecnologías en constante evolución. (University of Alabama, 2015)

2.4.1. Lenguaje de programación Java

El lenguaje Java es un lenguaje de propósito general, de alto nivel, que utiliza el paradigma de orientación a objetos. Su sintaxis y tipos están basados principalmente en C++, sin embargo, las diferencias principales con éste son la administración de memoria, siendo ejecutada por la máquina virtual automáticamente y no por el código de cada programa, y el soporte de procesos livianos o threads a nivel del lenguaje, que ayuda a controlar la sincronización de procesos paralelos. Estas características dan al lenguaje Java las propiedades de robustez y seguridad, evitando por ejemplo problemas de bufferoverflow utilizados en ataques a sistemas. (Coronel Castillo, 2011)

Java EE (Java Platform Enterprise Edition) es una especificación orientada al desarrollo y ejecución de software con arquitectura de N capas distribuidas, que se basa en una arquitectura de componentes modulares que se ejecutan sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación, y se considera un estándar. Java EE incluye la especificación de distintas API's, y define como se deben coordinar las mismas. Esto permite crear aplicaciones empresariales escalables y portables entre distintas plataformas. (Ceballos Sierra, 2005)

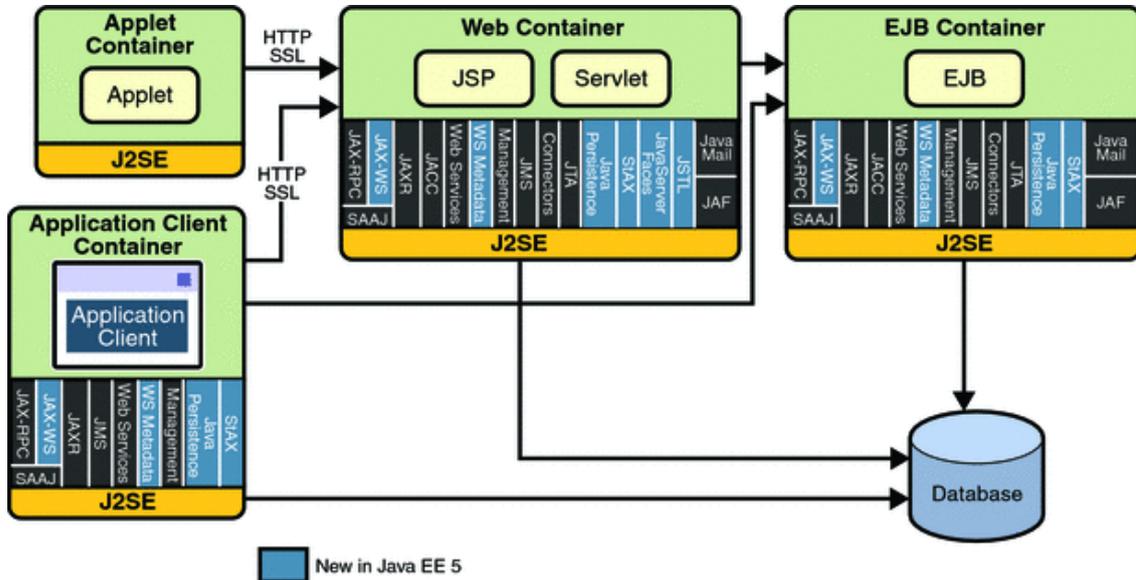
Características:

- Arquitectura basada en componentes
- Arquitectura multicapa
- Arquitectura distribuida
- Servicios ofrecidos por contenedores
- Multiplataforma
- Independencia del proveedor

- Organismo de control

Figura 2.4.1.1

Java EE Platform APIs



Fuente: Docs. Oracle (2003)

2.4.1.1. JavaScript

JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js, Apache CouchDB y Adobe Acrobat. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

El estándar de JavaScript es ECMAScript. Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1. Los navegadores más antiguos soportan por lo menos ECMAScript 3. El 17 de julio de 2015, ECMA International publicó la sexta versión de ECMAScript, la cual es oficialmente llamada ECMAScript 2015, y fue inicialmente nombrada como ECMAScript 6 o ES6. Desde entonces, los estándares ECMAScript están en ciclos de lanzamiento anuales. Esta documentación se refiere a la última versión del borrador, que actualmente es ECMAScript 2019.

JavaScript no debe ser confundido con el lenguaje de programación Java. Ambos "Java" y "Javascript" son marcas registradas de Oracle en Estados Unidos y otros países. Sin embargo, los dos lenguajes de programación tienen muchas diferencias en las sintaxis, semántica y usos. (Mozilla, 2019)

La programación del lado "cliente" de JavaScript permite añadir una cierta interactividad a las páginas web. En particular, las páginas podrán reaccionar a las acciones del usuario, como la selección en una lista desplegable, la selección de una casilla de selección o un clic en un botón de un formulario.

El código JavaScript, almacenado como archivos en un servidor web, es accesible, a través del protocolo HTTP, por el navegador del puesto "cliente", usando una llamada del usuario a través de la introducción de una URL o con un clic en un enlace de hipertexto. El código JavaScript es no compilado y es interpretado por un motor dedicado, incorporado al navegador.

El lenguaje JavaScript ha sufrido muchas evoluciones a lo largo del tiempo y lo soportan prácticamente todos los principales navegadores (Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari, Google Chrome...), y que están en los ordenadores independientemente de su sistema operativo (diferentes versiones de Microsoft Windows, Unix, Linux, Mac OS X). (Vigouroux, 2019)

2.4.2. Sistema Operativo Android

Android es un sistema operativo con una plataforma abierta para dispositivos móviles adquirido por Google y la Open Handset Alliance, su finalidad es satisfacer la necesidad de los operadores móviles y fabricantes de dispositivos, además de fomentar el desarrollo de aplicaciones, cualidad que ningún otro sistema operativo incluye en sus conceptos. (Google, 2010)

El sistema operativo Android ha evolucionado muy rápidamente: la progresión en el número de funcionalidades es similar a la de su cuota de mercado, simplemente impresionante. Estas funcionalidades, mejoras y correcciones de bugs han ido apareciendo con el tiempo en las sucesivas versiones.

Tras la versión Jelly Bean, el sistema Android está basado en el núcleo de Linux 3.0; las primeras versiones estaban basadas en el núcleo 2.6. Este núcleo tiene en cuenta la gestión de las capas inferiores, tales como los procesos, la gestión de la memoria, los permisos de usuario y la capa de hardware.

Sobre este núcleo, se sitúa la capa de bibliotecas del sistema proporcionadas por los fabricantes. Estas, de bajo nivel, están escritas en C o C++. Proporcionan los servicios esenciales, tales como la gestión de la visualización 2D y 3D, un motor de base de datos SQLite, la reproducción y la grabación de audio y vídeo, un motor de navegador web.

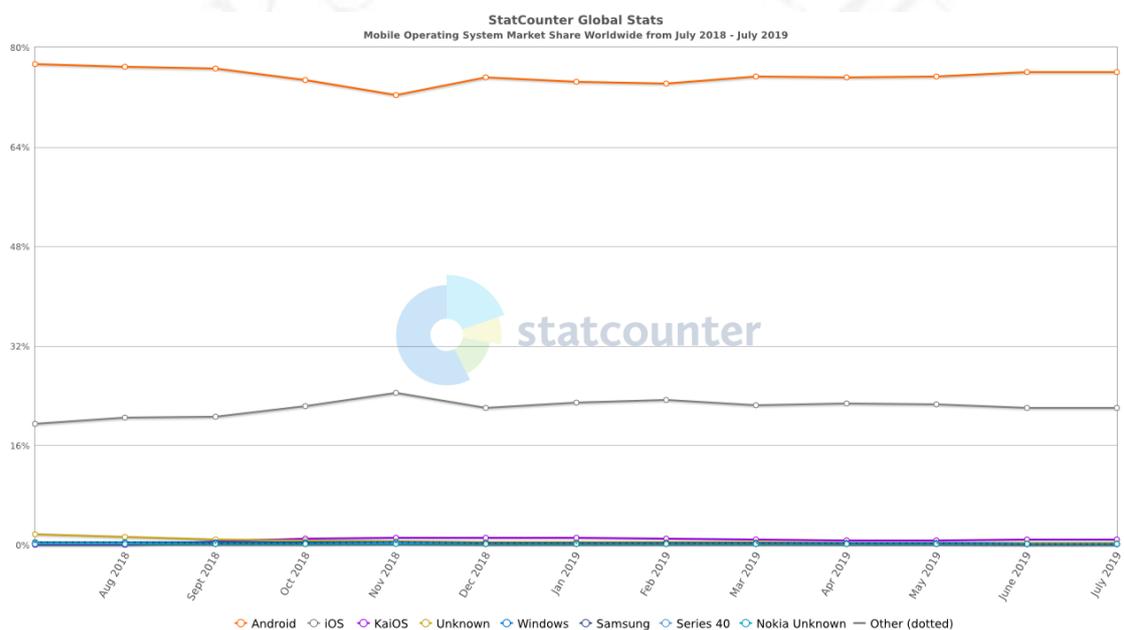
Las funcionalidades ofrecidas por estas bibliotecas las recoge y utiliza la capa superior bajo la forma de bibliotecas Java. Estas proporcionan bibliotecas y componentes

reutilizables específicos a dominios particulares. Encontramos, por ejemplo, bibliotecas de acceso a bases de datos, de telefonía, de localización geográfica, de comunicación sin contacto de corto alcance. (Hebuterne, 2016)

StatCounter es una plataforma que analiza el tráfico web del mundo, realiza estadísticas de usos, preferencias, dispositivos y/o sistemas operativos usados, entre otras cosas. Esto nos permite tener una idea aproximada de cuál es el dispositivo más usado según las preferencias de búsqueda que tengamos.

Figura 2.4.2.1

Cuota de mercado del sistema operativo móvil en todo el mundo



Fuente: Stat Counter (2019)

2.4.3. Base de datos MySQL

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2.

Lo que durante un tiempo se consideró como un sencillo juguete para su uso en sitios Web, se ha convertido en la actualidad en una solución viable y de misión crítica

para la administración de datos. Antes, MySQL se consideraba como la opción ideal para sitios Web; sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. MySQL supera desde hace tiempo a muchas soluciones comerciales en velocidad y dispone de un sistema de permisos elegante y potente, y ahora, además, la versión 4 incluye el motor de almacenamiento InnoDB compatible con ACID. (Gilfillan, 2003)

Como la base de datos de código abierto más popular del mundo, la velocidad, confiabilidad y facilidad de uso de MySQL han hecho que el contenido generado por los usuarios y las redes sociales estén disponibles para las masas. En los 18 meses transcurridos desde la adquisición de MySQL por parte de Oracle, Oracle continúa desarrollando MySQL internamente e integrando parches de la comunidad. Bajo la administración de Oracle, MySQL lanzó su primera nueva versión principal en 2 años. (Cabral, 2011)

2.4.4. **GitLab**

GitLab es un sistema auto hospedado para administrar su código. Fue lanzado por primera vez en octubre de 2011, y se actualiza cada veintidós días del mes desde entonces. Fue lanzado bajo la Licencia MIT.

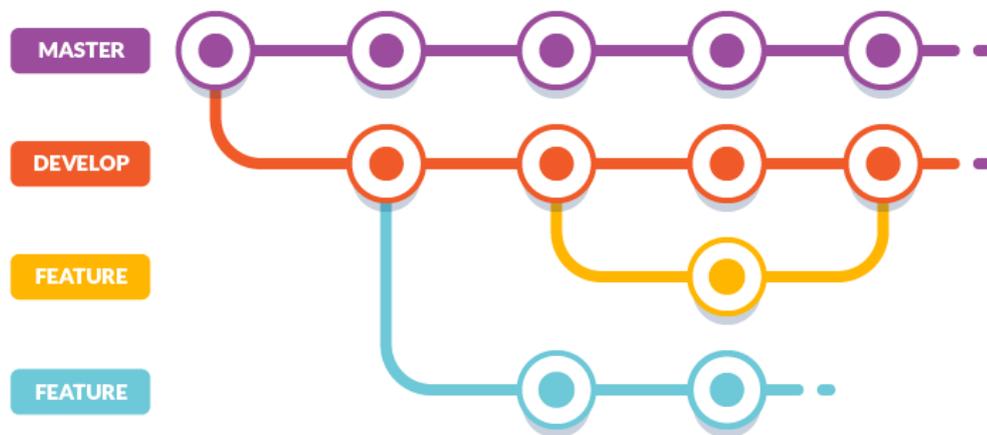
Solía estar alojado en GitHub, pero desde enero de 2014, su principal fuente de alojamiento es gitlab.com. La bifurcación de GitLab, que está alojada en GitHub, permanecerá activa como fuente donde puede presentar problemas y fusionar solicitudes. GitLab es un sistema complejo; tiene muchas partes móviles. Es muy útil entender la arquitectura de GitLab. La aplicación se basa en el marco de Ruby on Rails. La base de datos a la que se conecta depende de su configuración, pero GitLab le aconseja utilizar PostgreSQL.

GitLab no está conectado directamente a la Web, pero necesita un servidor de aplicaciones y un servidor web para conectarse al mundo exterior. Para el servidor de aplicaciones, se utiliza Unicorn. (van Baarsen, 2014)

En pocas palabras Git es un sistema de control de versiones, que registra los cambios que va sufriendo un archivo durante su desarrollo, es muy útil para trabajos en equipo ya que sigue todo un flujo de desarrollo manteniendo el completo control de lo que se desea mantener o desplegar.

Figura 2.4.4.1

Como se trabaja en un flujo Git



Fuente: GitLab (2018)

2.5. Pasarela de pagos

Las pasarelas de pagos son sistemas de pago electrónico que permiten la realización de pagos y transferencias entre tiendas electrónicas y entidades bancarias de manera segura. Se encargan de cifrar la información confidencial que se requiere para ejecutar transacciones bancarias por las redes. En el pago con tarjeta, la pasarela de pagos valida la veracidad de la tarjeta y organiza la transferencia del dinero de la cuenta del comprador a la cuenta del vendedor. Son llamadas terminales de punto de venta (TPV) virtual, pero no son realmente terminales de punto de venta porque estas últimas sí pertenecen al banco y el vendedor debe tener una cuenta en el banco en donde esté implementado el TPV virtual. Para el proceso de pago o transferencias bancarias es necesaria la utilización de tarjetas de crédito o débito como medios electrónicos de pago (Fonseca, Pérez, & Faurés, 2013).

Existen muchísimas alternativas en el mercado, pero nuestro país aún no cuenta con una buena cantidad de pasarelas de pago con una oferta atractiva.

Para poder elegir una se debe tener en consideración su viabilidad de uso en nuestro país, si acepta pagos en soles, con qué bancos y tarjetas trabaja y, sobre todo, cómo se van a transferir los fondos, ya que muchas veces esta transferencia no es factible porque se especializan solamente en ciertos mercados y países.

Muchas veces, los comercios ven conveniente ofrecer más de una forma de pago, con el propósito de facilitar la compra. En estos casos, el cliente puede elegir entre diferentes tarjetas u otras formas de pago. Por lo general, esto implica la contratación de

más de una pasarela de pagos, lo que se traduce en más cargos y comisiones (Vidal, 2015).

Figura 2.5.1

Como funciona una pasarela de pagos



Fuente: Pyme.pe (2010)

CAPÍTULO III: FUNDAMENTACIÓN DEL PROYECTO

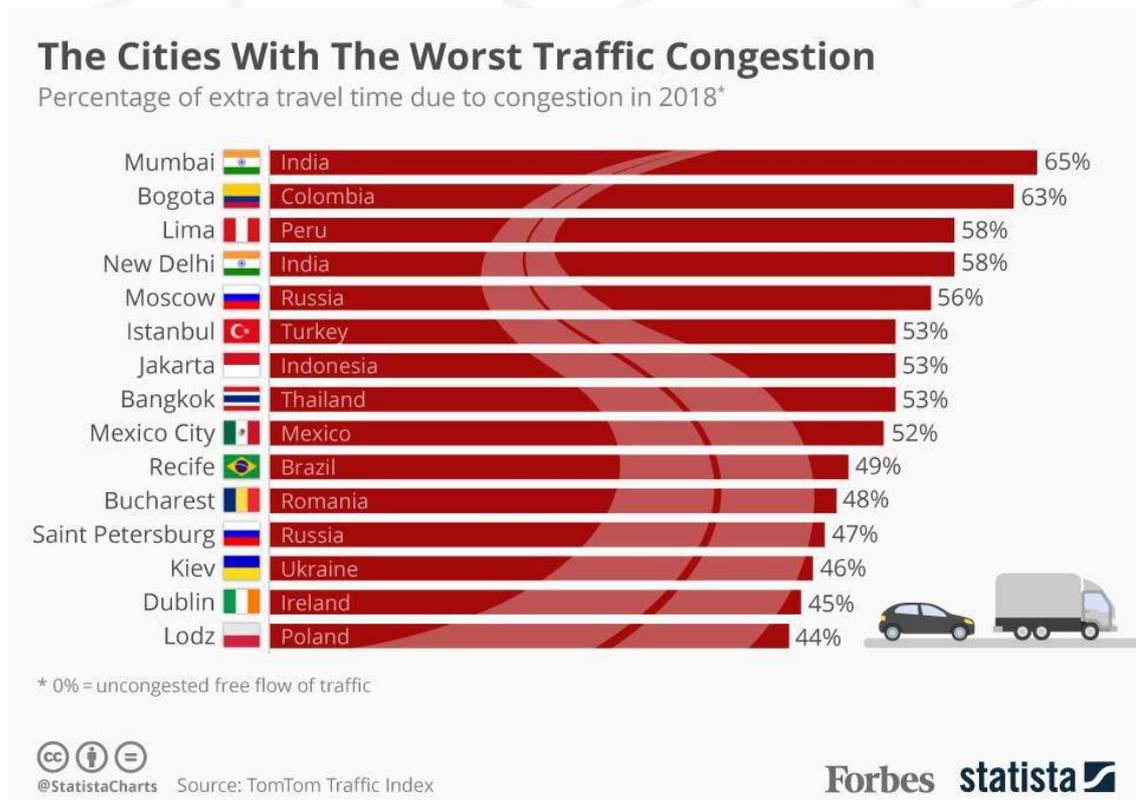
1.1. Fundamentación de la deseabilidad del proyecto

Actualmente uno de los problemas a resolver entre las personas que trabajan en el centro empresarial de San Isidro, es la falta de compartir la información de rutas a la hora de retornar desde su centro de trabajo hacia sus hogares, con el fin conocer personas que tengan rutas similares; y puedan así compartir viajes de una forma cómoda, confiable y segura.

De acuerdo con un informe de Forbes, el 75% de las ciudades en el índice han reportado niveles de congestión aumentados o estables entre 2017 y 2018, y entre estas la ciudad de Lima ocupa un tercer puesto entre ciudades en el mundo con mayor congestión vehicular.

Figura 1.1.1

The World's Worst Cities For Traffic Congestion



Fuente: Forbes (2018)

Cada día las insufribles colas en los paraderos de buses en los corredores de la Javier Prado entre muchos otros hacen que las personas tengan que esperar hasta 50

minutos para abordar un bus, esta situación genera mucho malestar entre los usuarios y que lastimosamente no tienen otra opción para movilizarse.

Figura 1.1.2

Los paraderos abarrotados del corredor Javier Prado



Fuente: El Comercio (2018)

Otras personas frente a esta situación optan por tomar colectivos informales lo cual es una opción más cómoda, pero a la vez muy insegura ya que cada día hay denuncias de personas que han sido asaltadas bajo esta modalidad debido a la carencia de filtros que puedan pasar los conductores de estos vehículos. Estos autos, que por lo general son coasters, autos o minivans, no tienen condiciones mínimas de seguridad. Algunos incluso circulan sin el SOAT que funciona ante accidentes.

Según la Policía de Tránsito, existen dos modalidades de servicios piratas (o servicios informales que no cuentan con autorización debida para realizar dicho servicio). Los que transitan sin los colores de ninguna empresa de transporte. A estos, por lo general, se les conoce como colectiveros. Según la GTU (Gerencia de transporte urbano), al cierre de diciembre del 2017 se han registrado 50 mil autos que ofrecen este servicio en Lima. Pero también están los que imitan el logo y los colores de una compañía formal. Para el usuario es difícil identificar a estas últimas. Tendrían que ingresar a la web de la GTU y verificar si la combi a la que pretenden subir se encuentra habilitada para circular en la capital.

Otra opción más segura pero más costosa son los taxis por aplicativo (UBER, Beat, Cabify), y en horas punta sube aún más esta tarifa debido a la alta demanda y al tráfico alto en dichas horas pudiendo llegar entre 20 a 30 soles para una ruta que normalmente costaría de 10 a 15 soles.

Figura 1.1.3

Tarifa dinámica de UBER en determinadas horas



Fuente: Uber (2019)

Una opción más accesible son los taxis compartidos por aplicativo (como UBER Pool), pero también conllevan algunos problemas, como la inseguridad por parte de las personas con las que uno viaja porque no existe “filtros para pasajeros”, además de la larga ruta que puede terminar realizando el taxi dejando a cada pasajero cerca a su lugar de destino y uno llegue a su destino en más tiempo que si hubiera tomado el bus.

El proyecto nace a partir del poco uso que se viene haciendo del carpooling en sectores empresariales concurridos como los que existen en el distrito de San Isidro, y a su vez se quiere aprovechar el uso cada vez más frecuente y masivo de los dispositivos móviles (Android) para así poder implementar una aplicación que permita conectar a los trabajadores del sector empresarial de San Isidro y que puedan compartir el mismo viaje.

Figura 1.1.4

Sector Financiero de San Isidro (en verde)

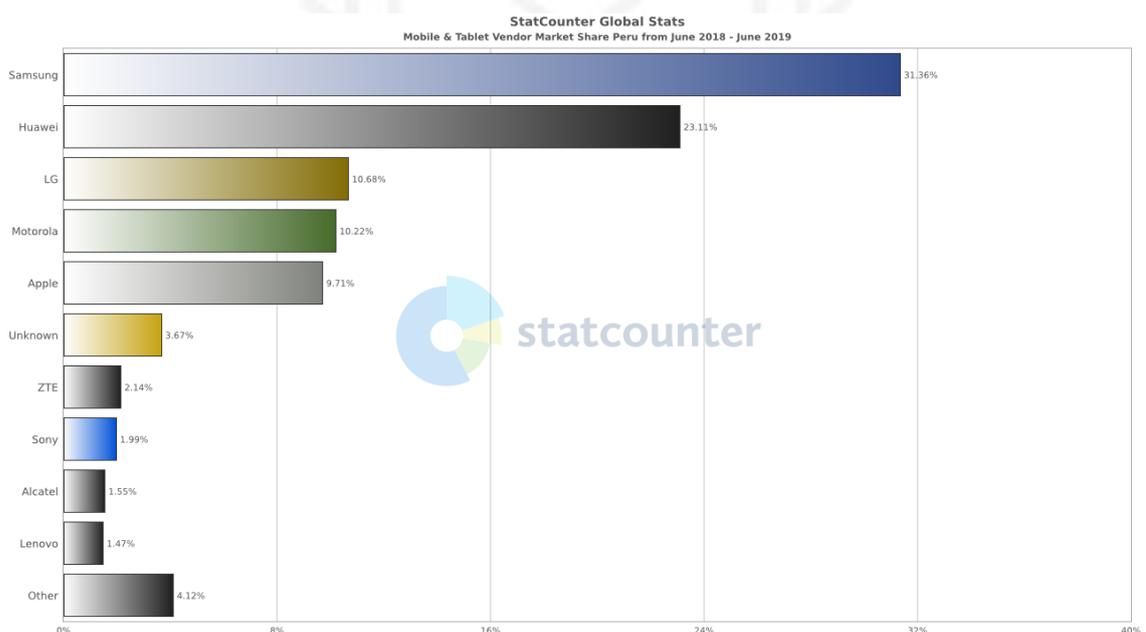


Fuente: Elaboración propia

En este sentido, se eligió por implementar la aplicación para esta primera versión para el sistema operativo Android ya que en el Perú son mucho más los usuarios que usan este sistema, y pueda ser aprovechado para que la aplicación pueda tener un mayor alcance hacia las personas. Según el gráfico adjunto que data de los últimos doce meses podemos ver la cantidad de usuarios de Android frente a los de IOS.

Figura 1.1.5

Mobile & Tablet Vendor Market Share Peru



Fuente: Stat Counter (2019)

Actualmente en el Perú se aprobó una ordenanza que restringe la circulación de vehículos particulares en determinadas vías principales de Lima según el último número de placa del vehículo durante las horas de mayor congestión. La misma entro en funcionamiento el 22 de julio del 2019 bajo la denominación de “Pico y Placa”.

Esta iniciativa aparte de ayudar con el excesivo congestionamiento vehicular en Lima, el proyecto presentado se podría apoyar de esta ordenanza ya que sería una alternativa de viaje para los conductores que normalmente llevaban sus vehículos al trabajo y que ahora ya no pueden llevarlo en determinados días. A su vez el tiempo de los trayectos se reducirían, debido al menor congestionamiento que se viene dando por las vías más transitadas en determinadas horas punta.

1.2. Fundamentación de la factibilidad del proyecto

Se decidió elegir este sector del distrito de San Isidro por ser uno de los principales ejes económicos del país. Entre 120 mil y 130 mil personas trabajan en el distrito, sobre todo en el centro financiero. La población flotante cada día es de 800 mil personas, que entran a San Isidro por negocios (Rodríguez, 2012). Además, tiene una ubicación geográfica que es estratégica para los fines de la aplicación.

Para que el proyecto pueda ser viable se optó por realizar el cobro de un porcentaje por cada ruta realizada por un usuario, en el gráfico Nro. 3.2.1 se muestra un ejemplo de una ruta desde San Isidro a la avenida La Molina, con un costo aproximado para nuestra aplicación de 3 a 5 soles por usuario (proporcional a la distancia recorrida), este monto aproximado se obtuvo en base a los costos de aplicaciones de taxi conocidas por el mismo trayecto y en la misma hora. Utilizamos la media de este rango de valores (S/4 soles) y con esto obtuvimos un total de 16 soles, lo cual se planea cobrar una comisión del 20% del total luego de aplicar la comisión de 4.20% (+\$0.30) que cobra Culqi (que es la pasarela de pagos que se optó a usar) con lo cual al final se obtendría un ingreso de S/2.87 soles.

Figura 1.2.1

Ingreso aproximado Te Jalo



Fuente: Elaboración propia

Estos ingresos servirán para el mantenimiento de la aplicación como costos por modificaciones o actualizaciones propias del App, así como de los costos por los servicios de Amazon Web Services tales como:

- Lambdas
El cual cobra en función del número de solicitudes para sus funciones y la duración, el tiempo necesario para ejecutar el código. (1 millón de solicitudes al mes sin cargo. luego 0,0000002 USD por solicitud).
- RDS (MySQL)
Se paga por la capacidad de cómputo a partir de la hora en que se ejecuta la instancia de base de datos. (Ej. Para una instancia db2.t2.medium un costo de 0.068 USD por hora).

Así como también se piensa invertir en publicidad en redes sociales (sobretudo Facebook), publicidad BTL y Publireportajes para que el producto sea conocido por las personas interesadas de dicho sector y así realizar la captación de más usuarios lo cual se traduce en más ingresos y así poder mejorar los servicios de la aplicación.

Se realizó una estimación del VAN (valor actual neto) en base a la inversión que se proyecta realizar y haciendo unos supuestos ingresos que se proyectan percibir a lo largo de 4 años.

Tabla 1.2-1
Costos servicios AWS

Region	Servicio	Mensual	Primeros 12 meses	Moneda	Configuración
South America (Sao Paulo)	Amazon RDS for MySQL	77.38	928.56	USD	Quantity (1), Storage for each RDS instance (General Purpose), Storage amount (30 GB), Instance type (db.t2.medium)
South America (Sao Paulo)	AWS Lambda	77.47	929.64	USD	Number of requests (5000000)
EU (Ireland)	Amazon Simple Email Service (SES)	21.8	261.6	USD	Email messages sent from EC2 (10 per month), Email messages sent from email client (100000)
South America (Sao Paulo)	Amazon Simple Notification Service (SNS)	21	252	USD	Requests (10 million per month)
			2371.8		

Fuente: AWS Pricing calculator (2019)

Se optó por utilizar la región de Sao Paulo porque se realizó un testeo de tiempos de latencia con la herramienta CloudPing a las regiones de AWS, obteniendo menores tiempos para esta región (aproximadamente 213 ms).

Se realizó un cálculo del costo anual de los servicios que se usaran en la aplicación con la herramienta de AWS Pricing calculator, obteniendo un monto total de \$2,371.80 por año, que multiplicándolo por 4 años daría un resultado final de \$9,487.20 (\$/31,782.12 soles), a esto le sumamos gastos de publicidad en general y obtenemos un gasto de inversión total de aproximadamente de S/80,000.

Con respecto a los ingresos si tomamos de referencia que en esta zona de San Isidro trabajan alrededor de 100 mil a 120 mil personas, de los cuales proyectamos que solo el 2% de personas usará la aplicación para el primer año y de los cuales estimamos que solo el 10% se convierten en trayectos completados y con una frecuencia de 8 veces al mes y un ingreso promedio para la aplicación de S/2.87 (calculado anteriormente) hace un ingreso de S/60,614 soles para el primer año.

Se estiman ingresos de S/70,000 para el segundo año, S/75,000 para el tercer año, y de S/80,000 para el cuarto año.

Asumimos una tasa de descuento del 10%.

Obtenemos el VAN: S/143,944.56.

El valor actual neto de la inversión es positivo, entonces conviene que realicemos la inversión.

Tabla 1.2-2

Tabla de costos y flujos de ingresos y egresos

	Horizonte de análisis				
	0	1	2	3	4
Ingresos					
Ingresos por trayectos completados		60,614	70,000	75,000	80,000
Egresos					
Gastos diversos (publicidad, mantenimiento)		-10,000	-10,000	-10,000	-10,000
Inversión inicial	-40,000				
Flujo de caja	-40,000	50,614	60,000	65,000	70,000

Fuente: Elaboración propia

1.3. Fundamentación de la viabilidad técnica

La solución a implementar será desarrollada para dispositivos Android y servicios en la nube de Amazon Web Services, así como también se manejará transacciones con el servidor de Culqi, todos estos aspectos se detallarán a continuación:

Aplicación Android

Para la creación de la aplicación se utilizará Android Studio con el lenguaje de programación java, se decidió por utilizar este es IDE ya que es el oficial de Android para el desarrollo de aplicaciones con gran soporte para diferentes dispositivos (móvil, Tablet, TV, etc.), y que también posee potentes herramientas para edición de código, análisis de errores de forma inmediata para así dar una solución más rápida a estos, así como también tiene herramientas integradas como la interfaz de usuario para el desarrollo y construcción de programas en Android.

Para el desarrollo de las funcionalidades se utilizarán las siguientes librerías:

- *Google Maps API*

Esta librería nos ayudará ampliamente en la gestión de rutas, el conductor verá el mapa de google en el aplicativo y colocará el origen y destino de su ruta. Así mismo el pasajero que desea usar el servicio verá las ruta trazadas en el mapa

creadas por los conductores e indicará su destino para que pueda ser visible por el conductor.

Figura 1.3.1

Servicio de geolocalización de Google Maps



Fuente: Google (2018)

- **Retrofit**

Es un cliente REST para Android y Java, que permite hacer peticiones GET, POST, PUT, PATCH, DELETE y HEAD. Se utilizará esta librería para las peticiones hacia los servicios creados en AWS.

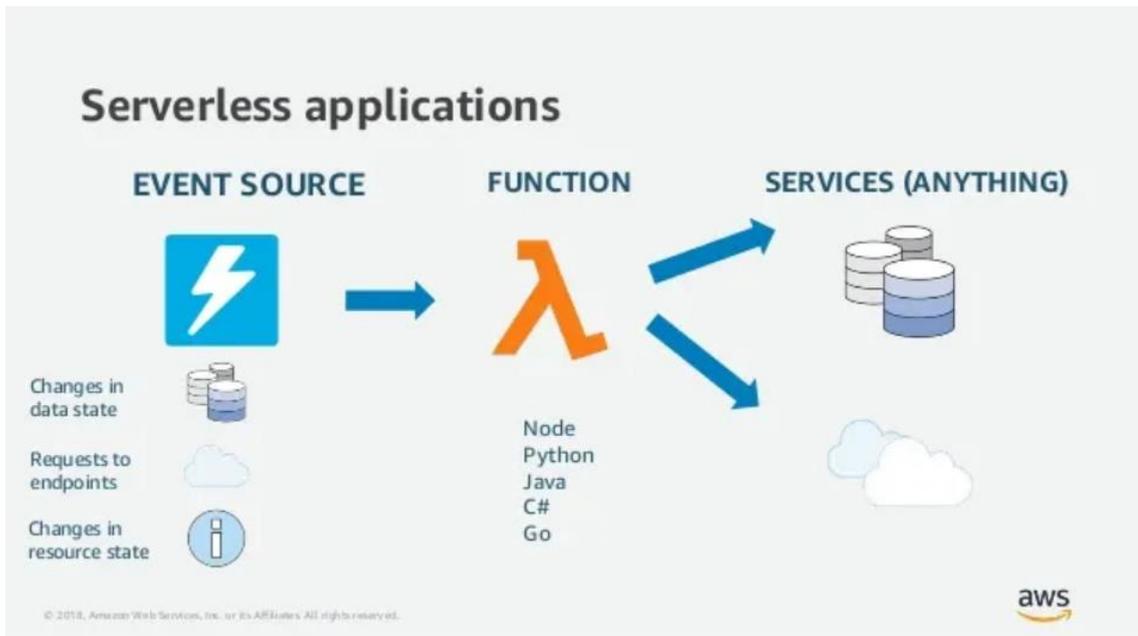
Servicios en la nube AWS

Serverless

Para el tema de micro servicios se utilizará la arquitectura *Serverless* que es un modelo de ejecución en el que el proveedor en la nube (AWS, Azure o Google Cloud) es responsable de ejecutar un fragmento de código mediante la asignación dinámica de los recursos. Aquí es donde se realizará la ejecución de los servicios que ejecutarán las operaciones de negocio, como las consultas de viajes, rutas disponibles, transacciones, pagos, etc.

Figura 1.3.2

Amazon Web Services Arquitectura Serverless



Fuente: Amazon Web Services (2018)

Lambda AWS

Se usará AWS Lambda ya que permite ejecutar código sin necesidad de administrar servidores (serverless) y se ejecuta solo cuando es necesario y es de gran escalabilidad, en donde solo se paga por lo que se consume, dejando de lado el mantenimiento y costos por tener administrando un servidor.

Figura 1.3.3

Costos Solicitudes Lambdas



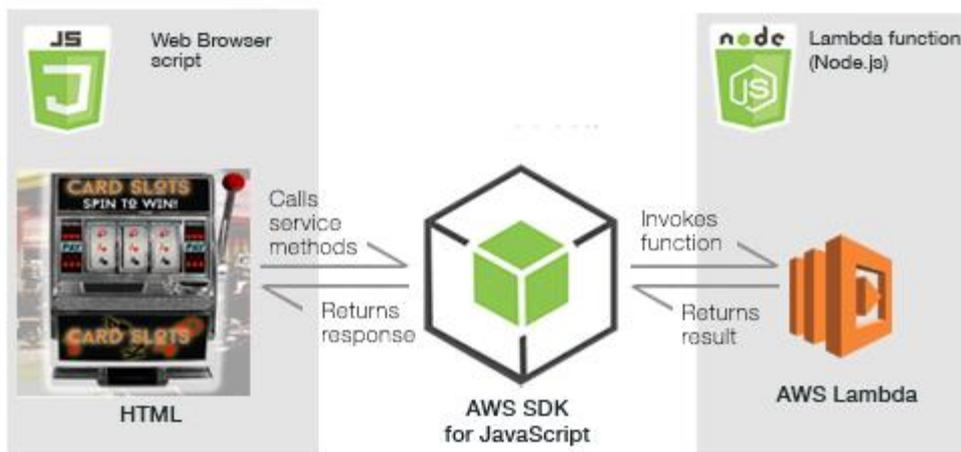
Fuente: Amazon Web Services (2019)

El lenguaje a ser usado en la creación de lambdas será con *Node Js* ya que permite aplicaciones mucho más rápidas en tiempo de respuesta debido a que utiliza el motor V8 de Google, además que su alto rendimiento permite manejar aplicaciones con alto tráfico de usuarios y eventos.

Para que las aplicaciones puedan tener acceso a los servicios se usará Amazon Api Gateway para la gestión segura de todos los procesos y peticiones simultaneas a estos, dando control y monitoreo a las API's que se lleguen a crear.

Figura 1.3.4

Arquitectura Node Js y Lambda



Fuente: Amazon Web Services (2018)

Amazon Relational Database Service (RDS)

Para guardar toda la información de los usuarios, así como las transacciones de viajes, registros, historiales, logs entre otros se hará uso de una base de datos, y para esto se utilizará un servicio de AWS, RDS (Relational Database Service) ya que es sencillo utilizar, configurar y escalar una base de datos en la nube. El servicio nos permite concentrarnos en nuestra aplicación para optimizarlas, darles mejores tiempos de respuesta, seguridad, etc. ya que AWS se encarga de lo demás. El motor de base de datos elegido es MySQL ya que es un motor confiable, robusto y sencillo de utilizar, además que esta optimizado para su uso en la nube.

Se describen algunas de las entidades más importantes:

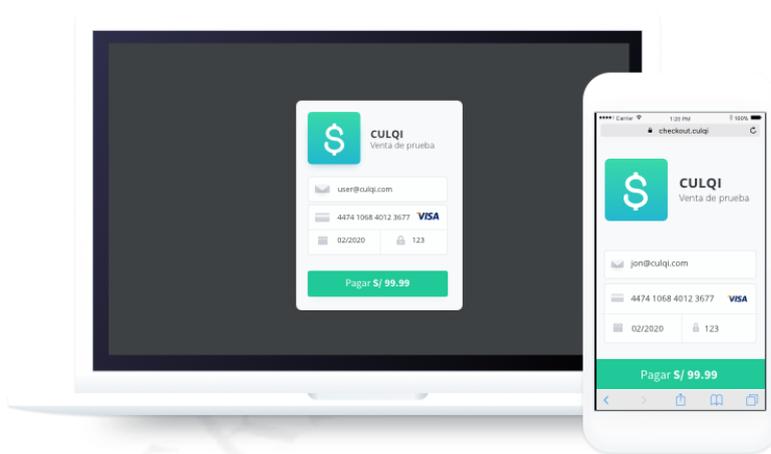
- *Usuario*
Corresponde al trabajador que se registrará con su correo corporativo y demás datos y tendrá acceso a cualquiera de los dos modos que tendrá la aplicación (conductor y pasajero).
- *Empresa*
Estarán registradas las empresas del sector financiero de San Isidro, así como su dominio de correo corporativo el cual será validado al momento del registro del usuario.
- *Ruta*
Esta será creada al momento de que un usuario conductor registre una ruta y podrá ser vista por los usuarios que deseen enrolarse a una.
- *Pasajero Ruta*
Esta será creada al momento de que un usuario pasajero desee enrolarse a alguna ruta que le convenga. Podrá cambiarlo o cancelarlo cuando lo desee.

Pasarela de pagos Culqi

Un tema importante son los ingresos ya que es un punto importante para la continuidad del proyecto y se optó por el cobro de un porcentaje por cada ruta finalizada por el usuario, y la tecnología que será usada para el cobro es la de una pasarela de pagos, la plataforma a ser usada será Culqi debido a que permite realizar todo tipo de pago por internet, su integración es simple y de manera rápida y además que es una empresa peruana de confianza.

Figura 1.3.5

Culqi, pasarela de pagos online



Fuente: Culqi (2019)

Culqi sirve para hacer y recibir pagos con tarjetas de crédito o de débito mediante smartphones. Funciona tanto para comercios que quieran contar con una forma extra de pago como para usuarios, que de esta manera evitan transportar dinero en efectivo para hacer compras.

El usuario al optar por una ruta disponible visualizará el monto del trayecto y si decide tomar la ruta, accederá a la pasarela de Culqi que estará integrado a la aplicación y podrá realizar el pago de su ruta con cualquier tarjeta de débito o crédito.

El proceso de pago se realizará de la siguiente manera:

- La aplicación móvil muestra el formulario de captura de datos de la tarjeta, usando bibliotecas en Android proporcionadas por Culqi.
- El cliente ingresa la información de su tarjeta.
- La información de la tarjeta es enviada de forma segura a los servidores de Culqi para ser tokenizada.
- Una vez que Culqi retorna el token a la aplicación móvil, este será enviado a nuestra aplicación para ser procesado.
- Desde la aplicación se hace una petición a los servidores Culqi para crear un Cargo inmediato o recurrente.
- Una vez recibida la respuesta de Culqi, se muestra de al cliente el resultado.

Arquitectura

Nuestra arquitectura se base en el modelo orientado a servicios ya que nos dará mayor flexibilidad y escalabilidad debido al desacoplamiento de servicios y procesos. Permite el desarrollo ágil de aplicaciones para que puedan ser manejables y más seguras, ya que proporciona una infraestructura y documentación común para desarrollar servicios con la posibilidad de añadir nuevas funcionalidades.

Metas de la arquitectura

- La aplicación “Te Jalo” podrá ser accedida en esta primera versión desde cualquier dispositivo Android con versión 5.0 en adelante.
- Para poder acceder a la aplicación se requiere contar con un correo corporativo del sector financiero de San Isidro registrado en la base de datos de la aplicación.
- El uso de la aplicación está limitado actualmente solo para los trabajadores del sector financiero de San Isidro.

Requisitos Funcionales del Sistema

- El Sistema deberá guardar la información de los usuarios, rutas, datos de contacto, medio de pago y los destinos de los usuarios.
- El sistema deberá validar que solo trabajadores del sector financiero de San Isidro se puedan registrar en la aplicación.
- El sistema deberá validar rutas y enrolamientos únicos por usuario, así como también validación periódica de que el correo aun siga estando activo.

Requisitos No Funcionales del Sistema

- La aplicación deberá tener un máximo de respuesta de 5 segundos para cualquier operación de consulta y/o transacción.
- La aplicación deberá correr sin errores para versiones Android 5.0 en adelante.
- La aplicación deberá guardar backup con la información del sistema con una periodicidad de un día.

CAPÍTULO IV: DEFINICIÓN DEL PROYECTO

El presente proyecto tiene como finalidad brindar una solución alternativa al problema que cada día vienen teniendo los trabajadores del sector empresarial de San Isidro al momento de retornar a sus destinos, ya sea por el tráfico caótico propia de la ciudad de Lima, al creciente parque automotor, pistas en mantenimiento, largas colas en los paraderos de buses en horas punta, informalidad, entre otros.

Se desea implementar una aplicación móvil en el sistema operativo Android por que como ya sabemos es el que mayor porcentaje de uso tiene entre las personas de nuestro público objetivo.

El proceso inicia desde el registro del usuario con su correo corporativo; el dominio de correo debe estar registrado en la base de datos, ya que debe pertenecer al sector empresarial de San Isidro), esto garantizará la seguridad para que solo las personas de este sector puedan registrarse. Una vez terminado el registro se enviará un mail con la contraseña temporal a la casilla del usuario para verificar que es el propietario del correo.

Una vez dentro del aplicativo seleccionará el tipo de usuario que es (conductor o pasajero), y según la elección tendrá determinadas opciones:

1. Conductor

- a. Podrá programar rutas, desde el punto de partida hasta su punto de llegada y la hora de partida.
- b. Podrá seleccionar el número de asientos disponibles.
- c. Tendrá opción a cancelar su ruta.

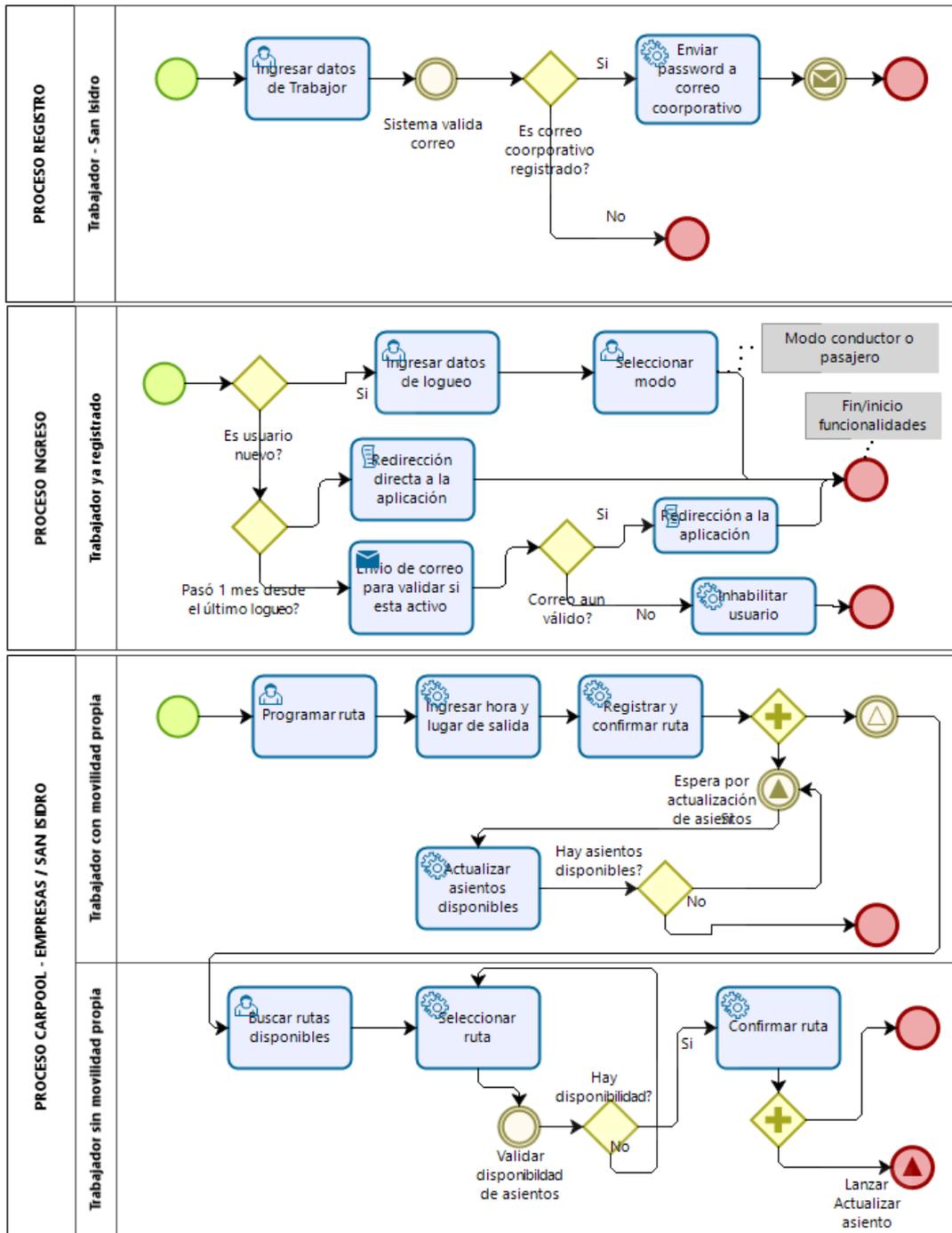
2. Pasajero

- a. Podrá realizar búsquedas de rutas según sea su conveniencia.
- b. Podrá seleccionar ruta y/o cancelarla en caso sea necesario.
- c. Podrá calificar al conductor.

En la figura 2.1 se presenta el diagrama para el modelo de procesos propuesto:

Figura 2.1

Diagrama de procesos del proyecto



2.1. Objetivo general

Desarrollar una solución que permita mejorar la forma en cómo los trabajadores en San Isidro puedan movilizarse de forma cómoda, segura, reduciendo tiempos de viaje y que permita que los trabajadores puedan conocer rutas similares y puedan compartir los viajes de retorno.

2.2. Objetivos específicos

- a) Reducir tiempos y costos para los trabajadores del centro financiero de San Isidro que normalmente usan colectivos, taxis o usan el transporte público.
- b) Generar seguridad y confianza al momento de transportarse desde sus centros de trabajo a su lugar de destino.
- c) Fomentar el uso del carpooling entre los trabajadores de una misma empresa.
- d) Optimizar el transporte con el fin de reducir costes (gasolina y/o estacionamientos) y reducir el impacto ambiental con menos autos circulando.

2.3. Diseño de la propuesta de valor

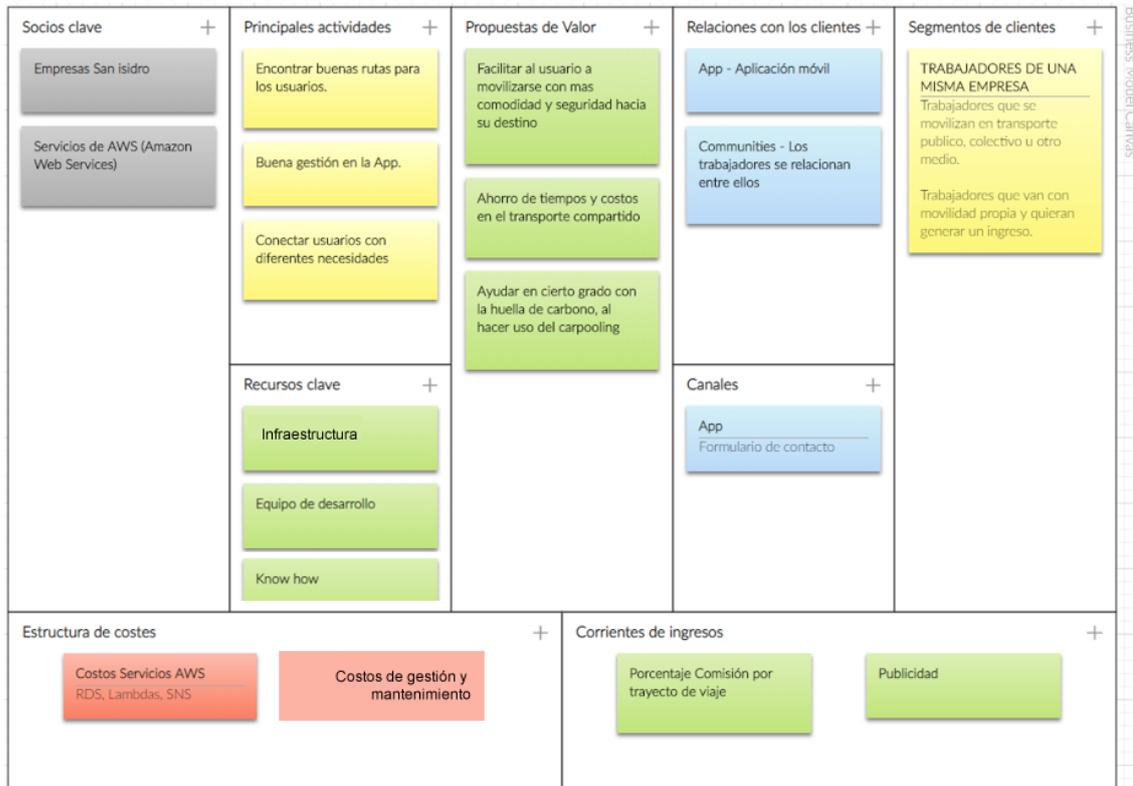
- Brindarle al usuario una nueva opción al momento de retornar a sus hogares desde sus centros de trabajo.
- Hacer que esta nueva opción sea percibida por el usuario como una forma más segura de transportarse ya que se sentirá más en confianza con personas de su mismo entorno laboral.
- Conectar trabajadores para viajes compartidos que está limitado solo a su ambiente laboral.
- Ahorro de costos en el transporte frente a taxis u colectivos con precios mayores sobretodo en horas punta.
- Mayor comodidad al transportarse entre los trabajadores que normalmente suelen movilizarse en buses de transporte público (colas en el paradero, buses llenos, etc.)
- Posibilidad que las personas con movilidad particular que viajan solas, puedan obtener ingresos sin desviarse de su ruta cotidiana y con total seguridad.
- Impactar en forma positiva en la huella de carbono, ya que al utilizar de manera eficaz un medio de transporte afectamos en la reducción de gases de efecto invernadero.

2.4. Modelo de negocio propuesto.

El modelo de negocio a usar será el Lean Canvas.

Figura 4.4

Modelo de negocio Lean Canvas



Fuente: Elaboración propia

- **Segmentos de clientes** - Son los trabajadores que trabajan en una misma empresa en el distrito de San Isidro ya que estamos creando el valor para este segmento de personas.
- **Relación con los clientes** - Mediante la aplicación los mismos trabajadores se relacionarán entre ellos, conectará a los conductores con los pasajeros.
- **Canales** - El canal con el que se quiere llegar a los clientes frente a cualquier eventualidad es mediante la misma Aplicación. Existirá una opción de ‘Contáctanos’.
- **Propuesta de valor** - Los usuarios podrán conocer las rutas disponibles para que puedan compartir un viaje en común con personas de su mismo entorno empresarial, haciendo de estos viajes más dinámicos, seguros y cómodos.

- **Principales actividades**

Las personas podrán programar los viajes para que puedan conectar personas que compartan similar ruta, a su vez los pasajeros podrán conocer las rutas que más le convenga y elegirla como ruta destino. La gestión del App para el correcto funcionamiento al momento de conectar personas debe ser realizada eficazmente.

- **Recursos clave**

Se especificó como recurso al equipo que desarrollará la aplicación que hará posible las necesidades de negocio que se quiere cubrir.

- **Socios clave**

Se identificó dos socios clave: las empresas ya que vienen ser parte del proceso del proyecto y con los cuales también se podría hacer alianzas estratégicas. Y por otra parte los servicios de AWS por un lado más técnico ya que con estos se hará posible todas las transacciones requeridas por el negocio.

- **Estructura de costes**

Los costos más importantes vendrían siendo por los servicios utilizados en AWS y posiblemente en publicidad.

- **Corrientes de ingresos**

Sabemos que actualmente los usuarios gastan cierta suma en transporte, harán un gasto menor con el uso de la aplicación y con esto se cobrará cierto porcentaje por cada ruta hecha. Adicionalmente se pondrá publicidad si se cree conveniente.

2.5. Pseudocódigo

Para describir en alto nivel el flujo de la funcionalidad principal de nuestra implementación haremos uso de Pseudocódigo para así explicar más eficazmente lo que a nivel de código podría resultar difícil y/o complejo de entender.

<i>Registro de trabajador</i>
Inicio
Trabajador registra sus datos: nombre apellido correoCorporativo
Si correo está dentro de (lista de correos corporativos registrado en BD) entonces enviarMailConClave()

validación ok
Si no
Mensaje “Correo corporativo no válido”
Fin

<i>Modo Conductor: Registro de Ruta</i>
Inicio
Conductor registra datos de su ruta: rutaInicio (coordenadas X, Y), rutaFin (coordenadas X, Y), asientos, hora, comentario, empresa
validarRuta(usuario, fecha); //valida en BD si existe una ruta activa para el usuario
Si conductor ya tiene una ruta creada para ese día entonces
Mensaje “Usted ya cuenta con una ruta activa para hoy”
Si no
registrarRuta(ruta); //registro en base de datos
Fin

<i>Modo Pasajero: Enrolamiento</i>
Inicio
listarRutasDisponibles(); //Lista de rutas disponibles (rutas y asientos disponibles)
Pasajero selecciona ruta
Pasajero registra datos de su ruta: rutaFinPasajero (coordenadas X, Y), empresa
validarEnrolamiento (usuario, fecha); //valida en BD si existe una ruta activa para el usuario
Si pasajero ya tiene una ruta asignada entonces
Mensaje “Usted ya cuenta con una ruta asignada hoy”
Si no
enrolamientoRuta(ruta); //registro en base de datos
actualizarAsientos();
Fin

2.6. Historias de usuarios

A continuación, se presentan las principales historias de usuario del flujo del proyecto presentado.

Historia	Registro
Como	Trabajador del sector financiero de San Isidro
Quiero	Registrarme en la aplicación
Para	Poder conocer rutas similares a mi destino con trabajadores del mismo sector y poder enrolarme a uno para viajar más cómodo y a la vez seguro.

Historia	Crear ruta
Como	Conductor
Quiero	Registrar mi ruta
Para	Mostrar mi trayecto a las personas que trabajan en el sector financiero de San Isidro y así puedan enrolarse las personas que les convenga mi trayecto. Y así no se pueda desperdiciar espacio en el auto y poder obtener alguna ganancia en ello.

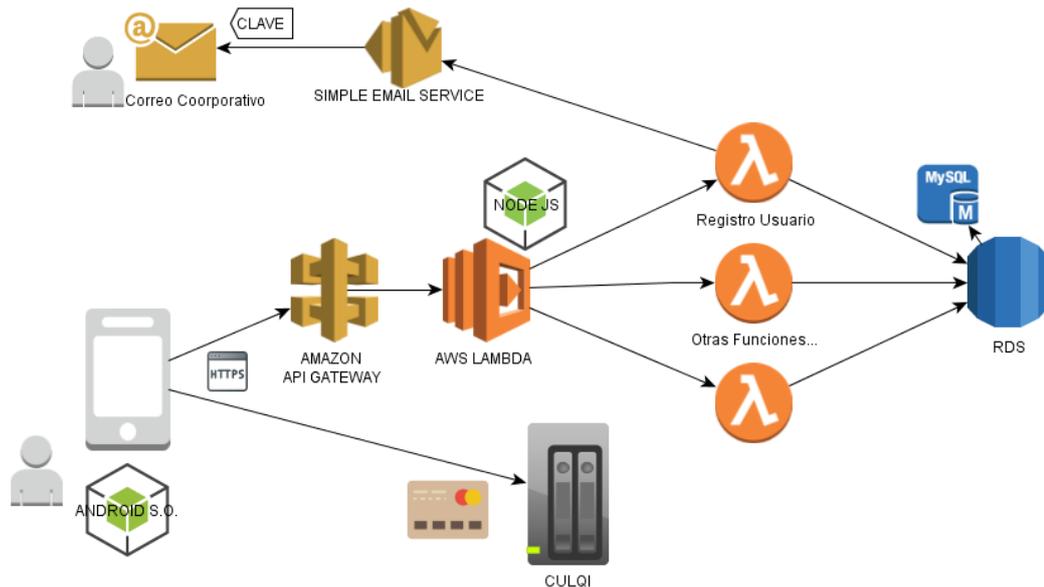
Historia	Búsqueda y Enrolamiento
Como	Pasajero
Quiero	Buscar rutas similares a mi destino
Para	Poder enrolarme a una ruta (habiendo previamente visualizado toda la información de esta como la hora de salida, punto de salida, precio, etc.) e indicar mi destino dentro del trayecto de la ruta del conductor.

2.7. Diagramas de arquitectura de software

A continuación, en las figuras 2.7.1, 2.7.2 y 2.7.3 se presentan los diagramas de aplicaciones, despliegue y de componentes del proyecto presentado.

Figura 2.7.1

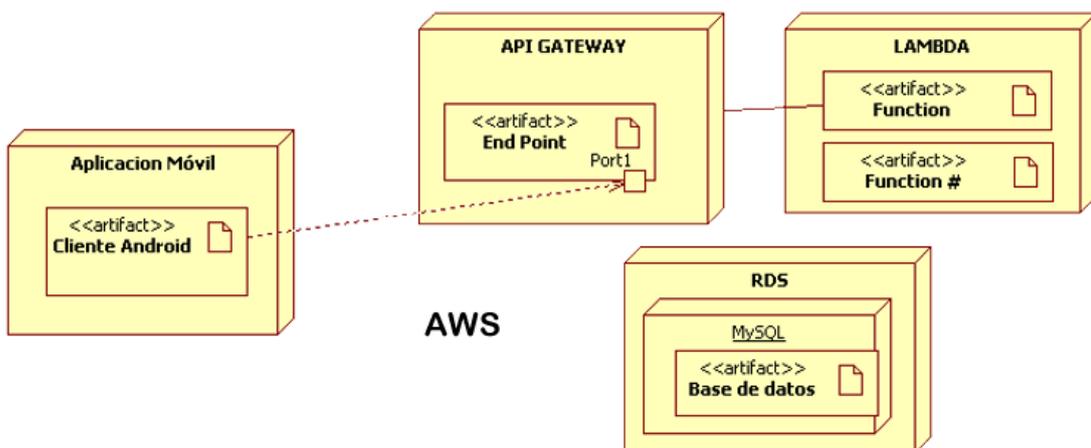
Diagrama de Aplicaciones



Fuente: Elaboración propia

Figura 2.7.2

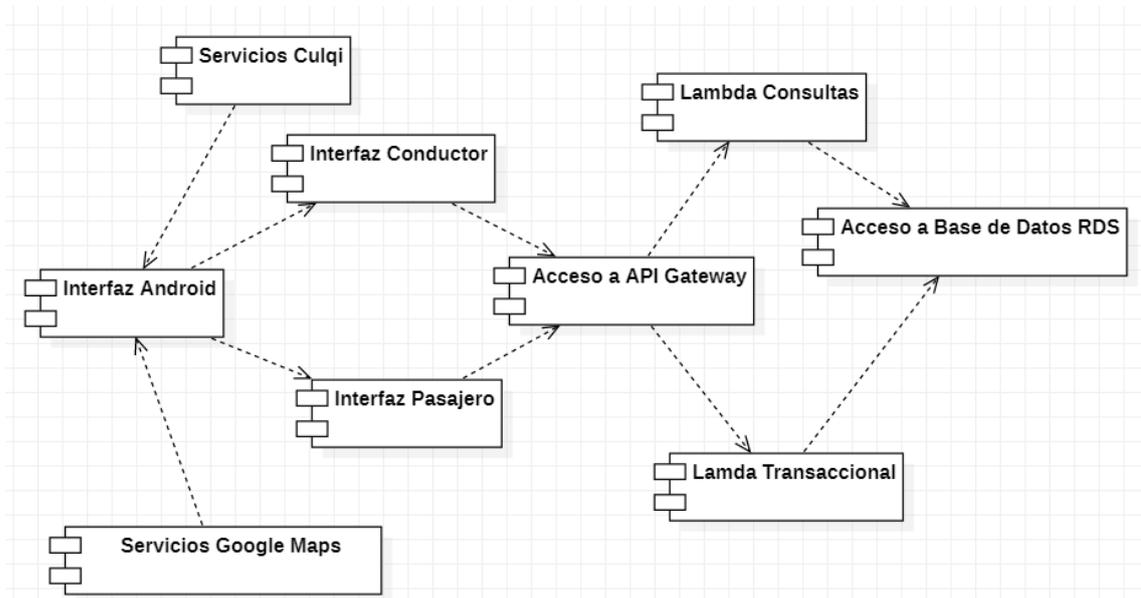
Diagrama de despliegue



Fuente: Elaboración propia

Figura 2.7.3

Diagrama de Componentes



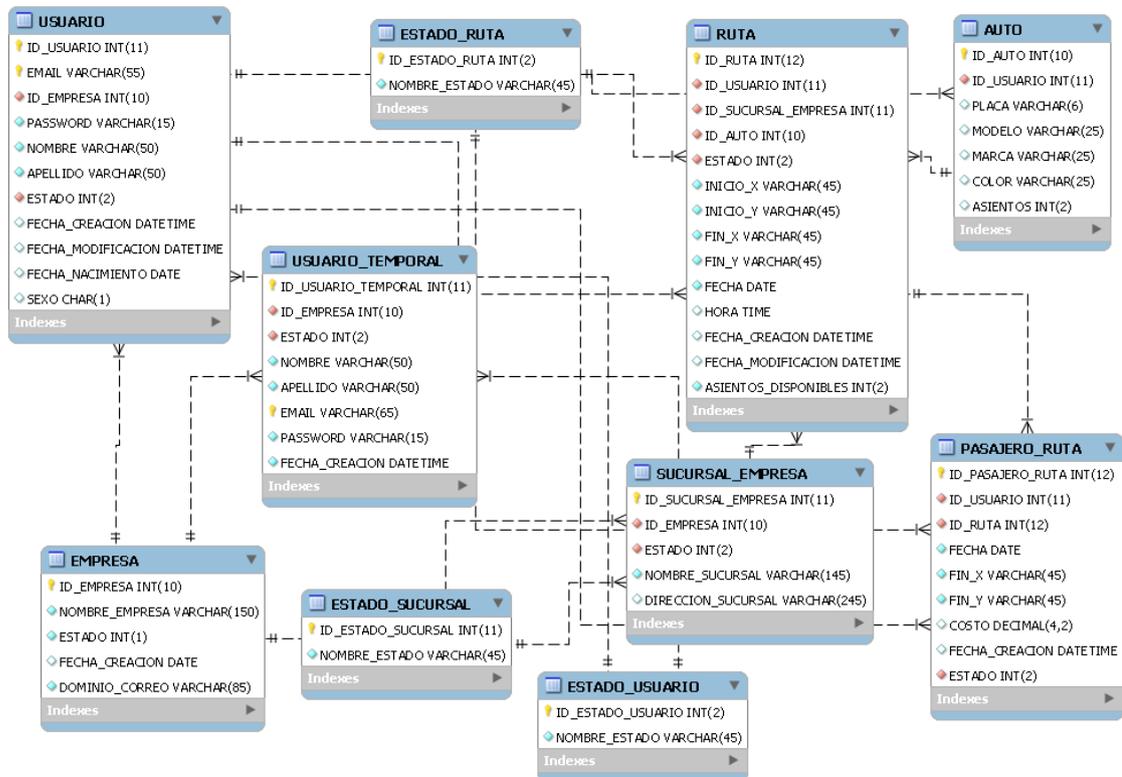
Fuente: Elaboración propia

2.8. Modelo de datos de la aplicación

A continuación, se muestra el diagrama del modelo de datos propuesto para el proyecto

Figura 2.8.1

Modelo de datos del Proyecto Te Jalo



Fuente: Elaboración propia

CAPÍTULO V: DESARROLLO DEL PROTOTIPO

La metodología elegida para documentar el despliegue del proceso de innovación es la de Design Thinking, ya que es una metodología para desarrollar la innovación centrada en las personas, ofreciendo una manera por la cual se pueden observar los retos, detectar necesidades y, finalmente solucionarlas. Se definen 5 pasos en esta metodología y cada paso cuenta con una variedad de técnicas, dinámicas y herramientas que elegiremos/adaptaremos a las necesidades del proyecto en el que se está trabajando.

3.1. Paso 1: Empatizar

Para este primer paso realizamos un mapa de empatía, porque nos permite tener una mejor comprensión de quienes son los clientes y su entorno, en que gastan su tiempo, que propuesta de valor esperan y cuanto podrían gastar en ello.

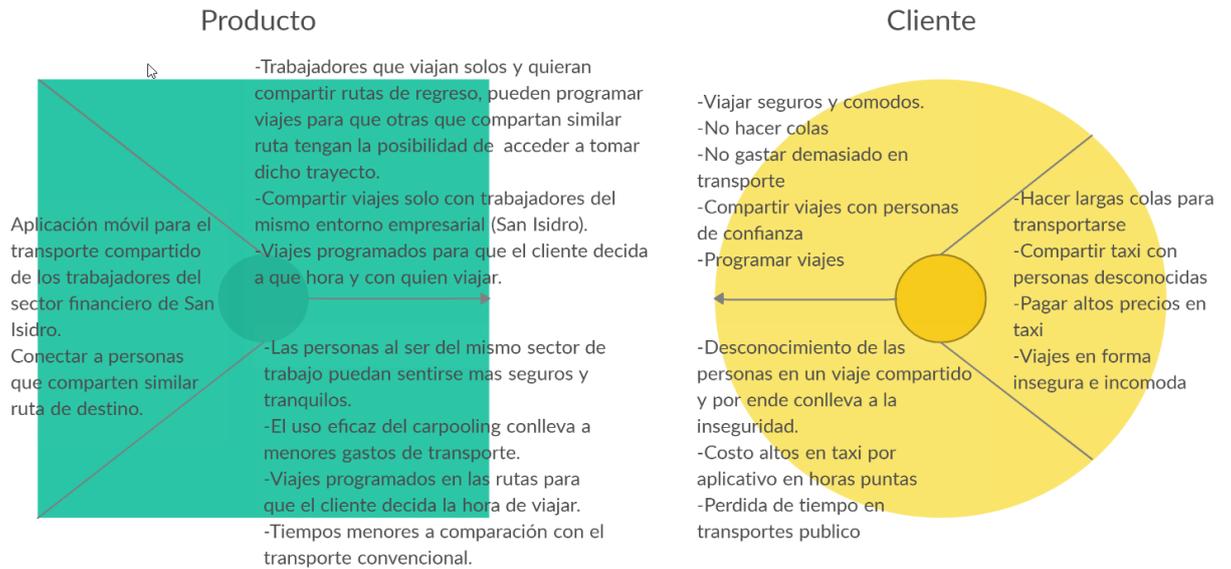
DICE <ul style="list-style-type: none">• Busca formas de movilizarse mejor.• El tráfico en esa zona en hora punta es caótico.• Los transportes públicos llenos y genera incomodidad.• No se encuentra transporte fácil.	PIENSA <ul style="list-style-type: none">• La seguridad y comodidad en el transporte es muy importante.• El costo de los taxis a esa hora es muy alto.• Deberían existir más alternativas para el transporte.
HACE <ul style="list-style-type: none">• Realiza frecuentemente largas colas para el transporte.• Las salidas tardes del trabajo es seguido, esto para esperar que baje la congestión.• Tomar más de un medio de transporte para llegar a su destino.	SIENTE <ul style="list-style-type: none">• Existe inseguridad si toma un taxi• Siente fastidio e incomodidad en cada viaje.• Confianza, si viaja con algún compañero en un taxi compartido.

3.2. Paso 2: Definir

Con la información recopilada en el paso 1, nos enfocaremos para ver dónde está el valor y los obstáculos. Para esto utilizaremos el canvas de propuesta de valor para que nos ayude a definir el problema que estamos tratando de solucionar y saber más sobre nuestro cliente y los problemas reales que tiene y como nuestro producto lo puede ayudar.

Figura 3.2.1

Canvas de propuesta de valor



Fuente: Elaboración propia

3.3. Paso 3: Idear

Aquí empezamos con el proceso de diseño y la generación de múltiples ideas, una herramienta en este proceso es la lluvia de ideas respecto a cierta pregunta definida y de las cuales surgieron las siguientes respuestas:

Tabla 3.3-1

Lluvia de ideas, Design Thinking - Idear

¿Cómo podríamos viajar de manera cómoda, segura y eficiente para retornar a nuestros destinos después del trabajo?				
Pool como colegio	Comentarios del conductor	Publicar disponibilidad	Informar las paradas	Clasificar zonas
Utilizar Pool Bus según ruta similar	Puntuación del conductor y del pasajero	Publicidad	Registrar el viaje	Chofer de reemplazo casi gratis
Uber moto		Alianza con la empresa	Compartir viaje	Estatizar el sistema de transporte e implementar cámaras y sensores de seguridad
Transporte empresarial			Nivel de comodidad	

Fuente: Elaboración propia

3.4. Paso 4: Construir el prototipo

Este paso nos permite convertir las ideas en algo más tangible, existen muchas formas de maquetar la solución, se optó por el dibujo en mockups para plasmar el proyecto.

- **Registro**

The mockup shows a smartphone screen with the title "Te Jalo" at the top. Below the title are two input fields: "Nombres" and "Correo corporativo". Underneath these fields is a checkbox labeled "Acepto los términos y condiciones". Below the checkbox is a "Registrarse" button. At the bottom of the screen, there is a link that says "Ya tienes cuenta?, ingresa aqui".

1. Paso previo al registro

El registro es un paso muy importante debido a que se verificará que el usuario sea un trabajador del sector empresarial de San Isidro, se valida que sea un correo corporativo registrado en la base de datos del sistema.

2. Validación de datos

Aparecerá un mensaje en caso no sea un correo corporativo registrado en la base de datos.

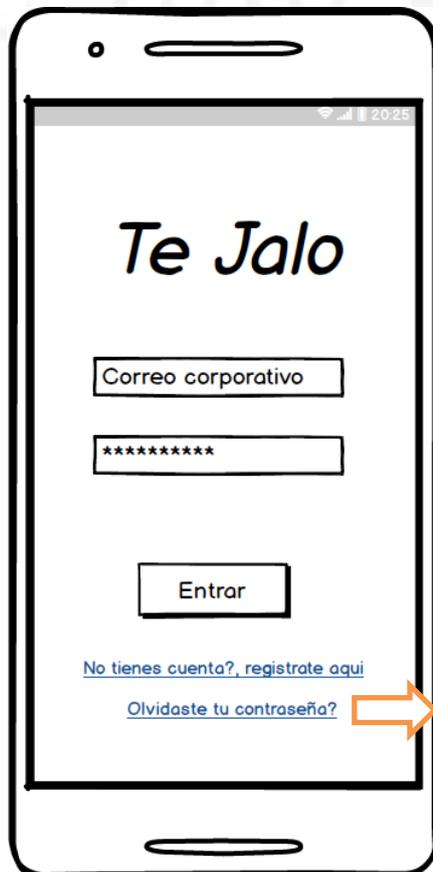
The mockup shows the same registration screen as the previous one, but with an error message displayed in an orange box at the top: "El correo ingresado no es un correo corporativo registrado". Below the error message, the "Nombres" field contains the text "Angel Remigio" and the "Correo corporativo" field contains "aremigio@correo.com". The "Registrarse" button is still present, along with the link "ya tienes cuenta?, ingresa aqui" at the bottom.



3. Registro exitoso de datos

En caso el registro sea exitoso verá un mensaje el cual indicará que se enviará la clave al correo corporativo registrado para que pueda ingresar a la aplicación.

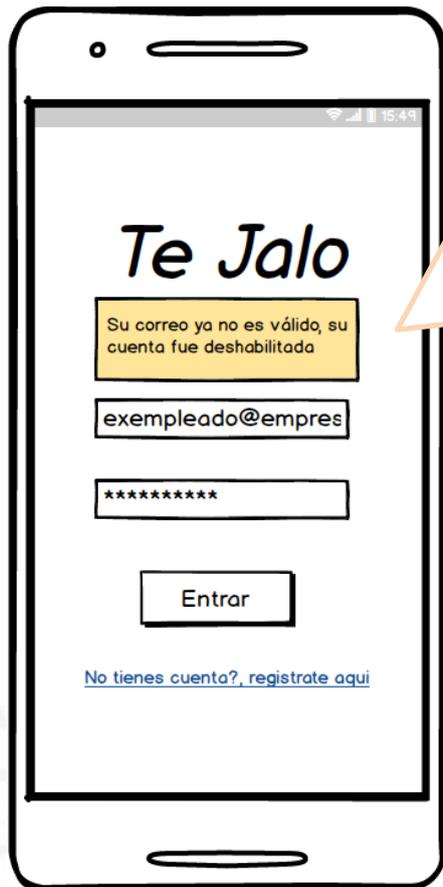
- Login



4. Logueo y olvido de clave

Una vez registrado el usuario podrá ingresar la primera vez con su correo corporativo. También tendrá la opción de recuperar su clave por si es que llega a olvidarse.





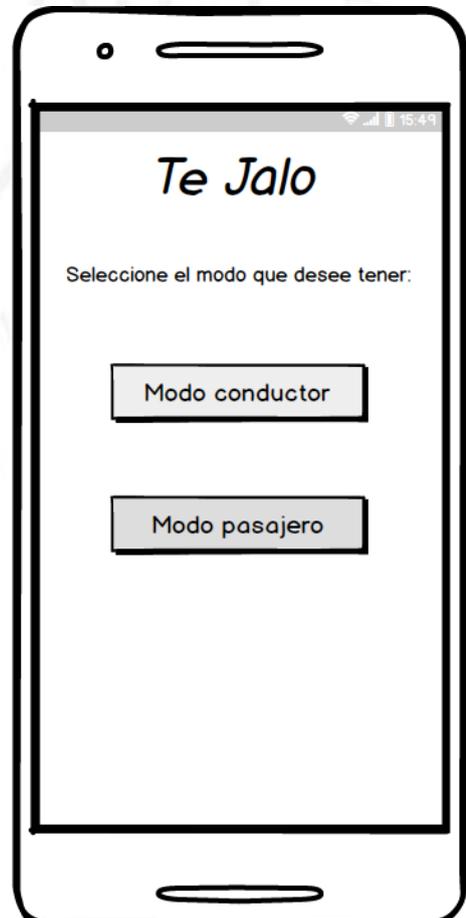
5. Validación de correo aun activo

En caso el trabajador ya no este laborando en el sector, al momento de volver a ingresar se le invalidará la cuenta.

Cada cierto tiempo la aplicación pedirá loguearse al usuario, esto con la finalidad de validar la casilla del correo corporativo, mediante el envío de un email, si este ya no existe devolverá un error (típicamente el 511). Con esto se comprueba que el usuario ha sido dado de baja.

6. Selección de modo

Al momento de ingresar el usuario podrá elegir el modo que tendrá su perfil, el cual también tendrá la opción de cambiarlo.



- **Modo conductor**

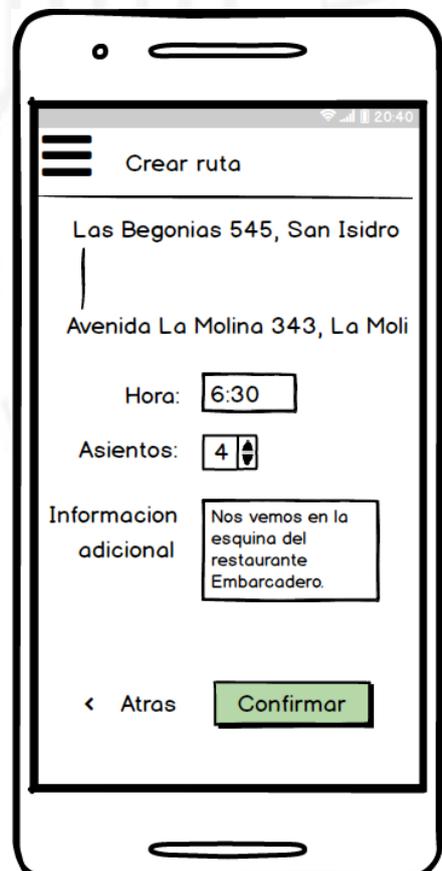


1. Trazar ruta inicio y fin

En caso el usuario escoja el modo conductor, deberá indicar su ruta de inicio y de fin para su trayecto de retorno.

2. Creación de Ruta

Luego de seleccionar su ruta, deberá programar su hora de salida, los asientos disponibles y algún dato adicional de interés si lo desea.





3. Opción menú – ver pasajeros enrolados

Una vez haya creado una ruta y la tenga activa, el usuario tendrá la opción de ver a los pasajeros que se han enrolado a la ruta que programó previamente.

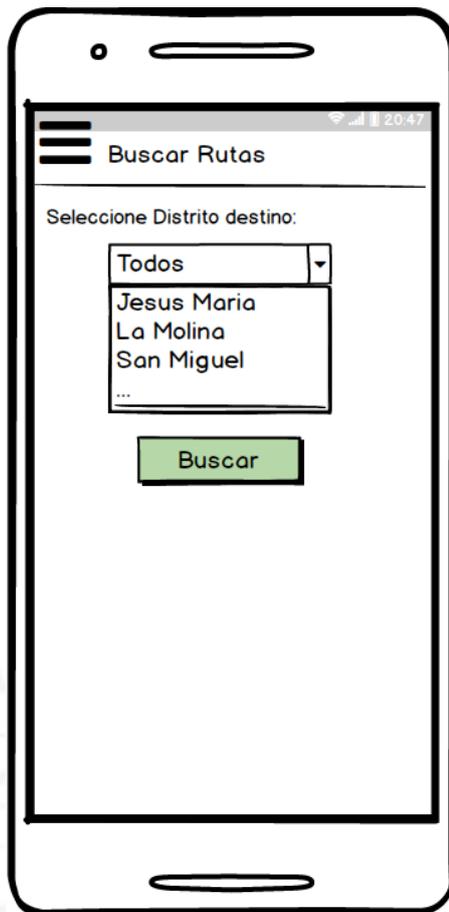
Podrá visualizarlo desde el Menú >

4. Confirmar o rechazar algún enrolamiento.

Podrá acceder al detalle de cada una de ellas y si en caso fuera podrá rechazar a alguno de ellos.



- **Modo pasajero**



1. Seleccionar distrito destino

En modo pasajero el usuario podrá buscar las rutas disponibles seleccionando su distrito destino, o en su defecto todas las rutas disponibles.

2. Ver rutas disponibles

Se puede observar las rutas disponibles con el resumen de cada una de ellas, con los datos de la ruta y del conductor.



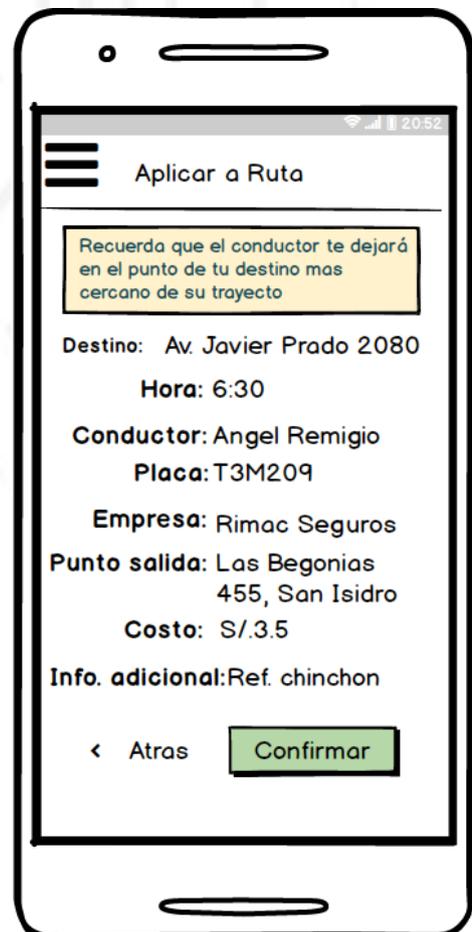


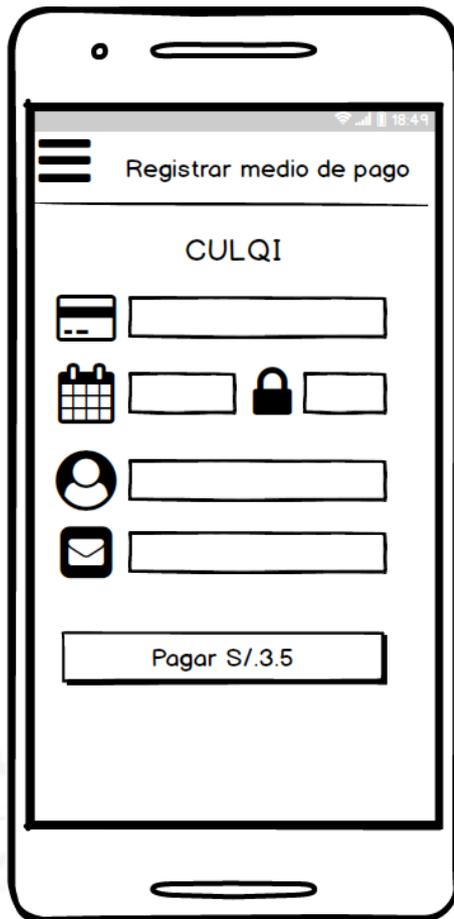
3. Indicar destino

Al seleccionar un trayecto, el usuario deberá seleccionar su destino, el cual deberá estar alrededor de la línea del trayecto del conductor.

4. Detalle de ruta a enrolarse

El usuario debe tener en cuenta que el conductor no se desviará de su trayecto normal. Se muestra el detalle de la ruta y el costo (que será proporcional a la distancia del trayecto de su ruta).



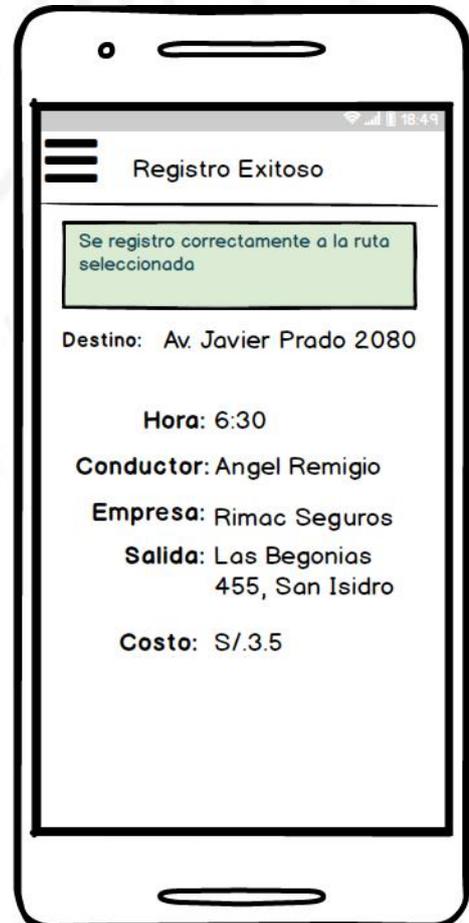


5. Registro de medio de pago

En caso el pasajero aun no tenga un medio de pago registrado, le aparecerá la pasarela de pagos Culqi para que pueda efectuar el pago de la ruta.

6. Enrolamiento exitoso

Al terminar el proceso se le mostrará un mensaje de que el enrolamiento a la ruta fue exitoso. Y también visualizará el detalle de la ruta.



- **Calificaciones y otros**



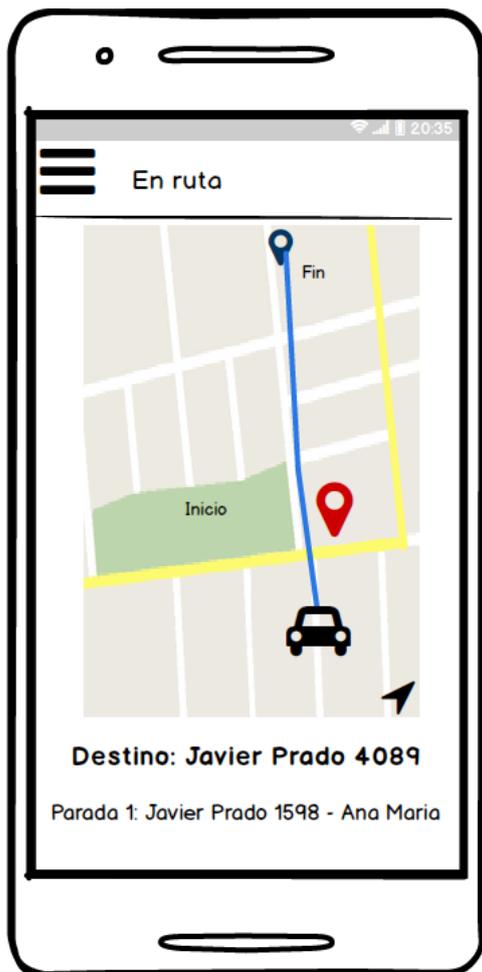
1. Calificación

Al terminar una ruta, el pasajero como el conductor podrán calificarse y si lo desea podrán enviar algún comentario adicional.

2. Registro datos conductor

Si el usuario utiliza el modo conductor, tendrá la opción de registrar los datos de su vehículo, así como sus datos personales (como su estado de cuenta) para que pueda realizar rutas y pueda recibir la comisión de sus trayectos finalizados.





3. En ruta

Tanto el conductor como los pasajeros podrán ver su ruta en tiempo real, así también el conductor verá las paradas que hará en su trayecto.

3.5. Paso 5: Evaluar

Se realizó la revisión de los prototipos con el asesor designado, así como también se solicitó a algunos trabajadores de nuestro público objetivo (a los que se les realizó la encuesta) que dieran sus opiniones y comentarios al respecto y se identificó algunos vacíos y observaciones que se fueron solucionando, así como también se realizaron mejoras en las funcionalidades. Así también se encontraron nuevos procesos que se incorporaron para que así la funcionalidad pueda estar alineada con los objetivos del proyecto.

CONCLUSIONES

Actualmente existe una demanda entre los trabajadores del sector financiero de San Isidro para la cual la aplicación desarrollada resultaría muy útil, ya que no solo ayudaría en el transporte eficaz de pasajeros, sino que sería una opción más, que ayudaría en el problema del tráfico que vivimos cada día.

En esta primera versión de la implementación, se desarrolló la aplicación móvil para el sistema operativo Android ya que actualmente es el más usado entre nuestro público objetivo, con lo que se piensa ahorrar tiempos en transporte, reducir costos y que, con el tiempo, tenga cada vez mayor aceptación entre los trabajadores de este sector. A pesar de esto se piensa desarrollar la versión en IOS para cubrir el porcentaje faltante de usuarios, para que así se tenga un mayor alcance y más personas del sector financiero puedan disfrutar de un transporte más seguro, cómodo y más accesible con trabajadores de su mismo entorno, y a su vez fomentar aún más el uso del carpooling entre estos usuarios e ir ampliando a otras zonas donde la población económicamente activa está presente, como lo es el centro histórico de Lima, la zona financiera de Miraflores, entre muchas otras. Y en paralelo podremos contribuir de manera paulatina en reducir el impacto ambiental por las emisiones de carbono de un desmesurado parque automotor.

RECOMENDACIONES

Para que la implementación de la aplicación móvil pueda llegar a la mayor cantidad de personas de nuestro público objetivo, se requerirá de una campaña de comunicación interna entre las empresas que conformen el sector empresarial, tales como sus canales disponibles ya sea en boletines, intranet u otro tipo comunicación disponible, así como crear incentivos para las personas que participen. También se desea invertir en publicidad BTL, publisreportajes y en anuncios en Facebook con segmentación enfocada ya que este es un canal en el que la mayoría de las personas dedica su tiempo, busca información, comparte sus experiencias, entre otras cosas. Con un creciente número de usuarios se deberá de mejorar la capacidad en los servicios de computación utilizados en la aplicación, analizar qué región en donde están los servicios de AWS ofrece la menor tasa de interrupciones de IISP y menores tiempo de latencia, así como también agregar servicios adicionales (como push notifications, sms, big data, etc.) para que así el usuario pueda tener una mejor experiencia a la hora de utilizar la aplicación.

REFERENCIAS

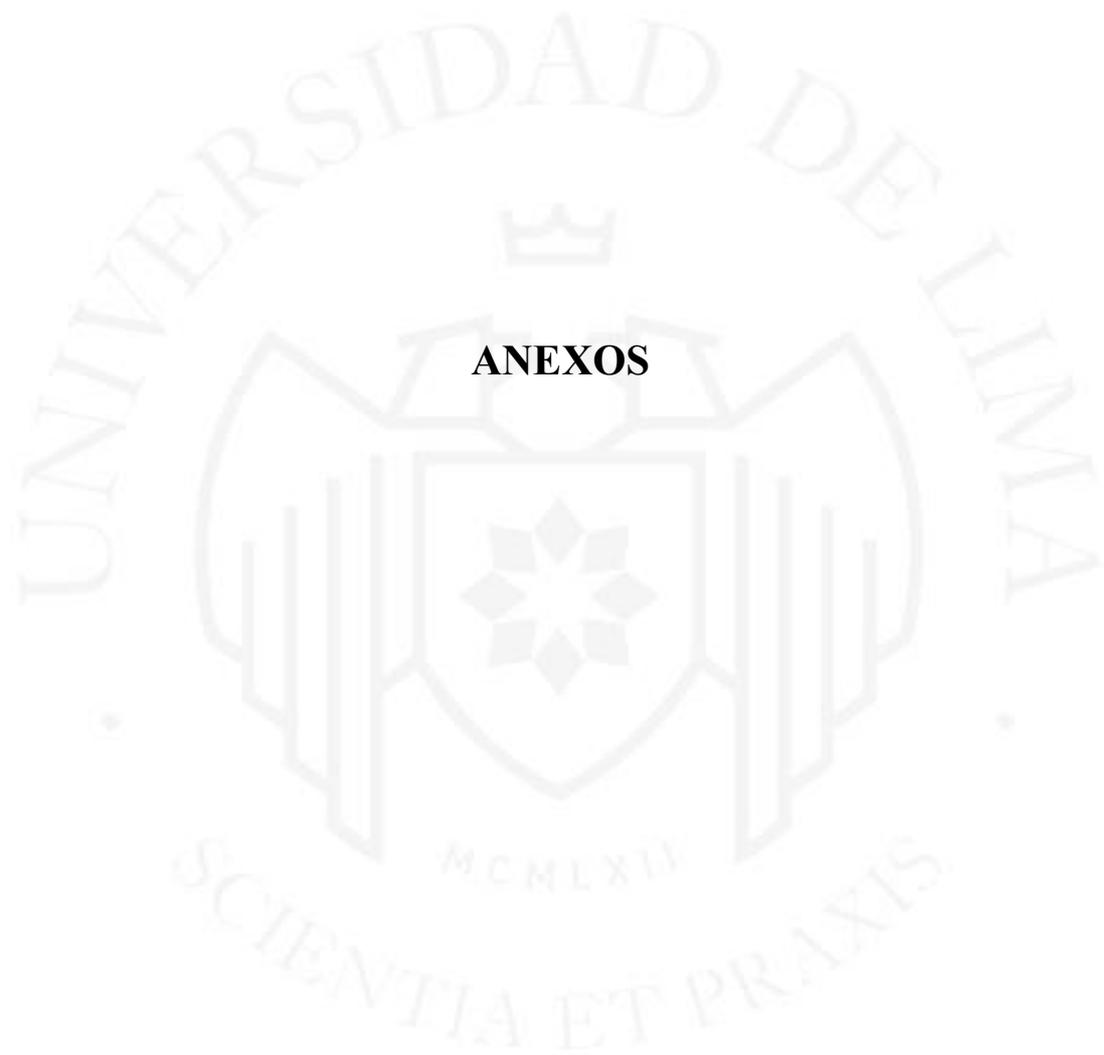
- van Baarsen, J. (2014). *GitLab Cookbook*. Packt Publishing.
- Aditya, Akkus, Beck, Chen, Hilt, Rimal, . . . Stein. (2019). Serverless Computing Revolutionize NFV? *Proceedings of the IEEE.*, 667-678.
- Andre, A. (2005). Software Architecture Overview. *Cornell University Library, arXiv.org*.
- AWS. (2019). *Amazon Web Services*. Obtenido de Amazon Web Services: <https://docs.aws.amazon.com/lambda/latest/dg/lambda-dg.pdf>
- Barletta, F., Pereira, M., Robert, V., & Yoguel, G. (2013). Argentina: dinámica reciente del sector de software y servicios informáticos. *Revista de la CEPAL*(110), 137-155. Obtenido de <http://www.cepal.org/publicaciones/xml/1/50511/RVE110Yoqueletal.pdf>
- Cabral, S. K. (2011). MySQL 5.5: Improving on the World's Most Popular Open Source Database. *Database Trends and Applications*, 30.
- Cantelon, M., Harter, M., Holowaychuk, T., & Rajlich, N. (2014). *Node.js in Action*. Manning.
- Ceballos Sierra, F. J. (2005).
- Choy, M., & Chang, G. (2014). *Medidas macroprudenciales aplicadas en el Perú*. Lima: Banco Central de Reserva del Perú. Obtenido de <http://www.bcrp.gob.pe/docs/Publicaciones/Documentos-de-Trabajo/2014/documento-de-trabajo-07-2014.pdf>
- Coronel Castillo, E. G. (2011). *Desarrollando soluciones con Java*. Lima: Macro.
- Fonseca, Pérez, & Faurés. (2013). Pasarela de pagos para la seguridad de transacciones bancarias en línea. *Revista de investigación. Editada por Área de Innovación y Desarrollo, S.L*, 10-10.
- García Nieto, J. P. (2013). *Constur ye tu Web comercial: de la idea al negocio*. Madrid: RA-MA.
- Gestión. (2018). *El 72% de los limeños termina estresado a causa del tráfico, según un estudio*. Obtenido de <https://gestion.pe/peru/lima-72-ciudadanos-estresa-causa-traffic-estudios-nndc-245062>

- Gestión. (16 de Marzo de 2019). *Parque automotor se renueva en 6% al año, cuando debería hacerlo en 10%*. Obtenido de <https://gestion.pe/economia/parque-automotor-renueva-6-ano-deberia-hacerlo-10-261551-noticia/>
- Gestión, R. (07 de 06 de 2019). <http://gestion.pe>. Obtenido de Gestión Perú: <https://gestion.pe/tendencias/lima-ciudades-peor-trafico-269483>
- Gilfillan, I. (2003). *La Biblia MySQL*. ANAYA.
- Google. (2010). *google.com*. Obtenido de Google.
- Haines. (2017). Serverless computing with AWS Lambda. *Network World Inc.*
- Hebuterne, S. (2016). *Android - Guía de desarrollo de aplicaciones Java para Smartphones y Tablets (3ª edición)*. ENI Ediciones.
- Jiau, & Huang. (2019). Self-Organizing Neuroevolution for Solving Carpool Service Problem With Dynamic Capacity to Alternate Matches. *IEEE Transactions on Neural Networks and Learning Systems*, 1048-1060.
- Mell, & Grance. (2019). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *Special Publication 800-145 - NIST*, 2-3.
- Mozilla. (2019). *mozilla.org*. Obtenido de mozilla.org: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- NewsRx LLC. (2015). Amazon Web Services Inc.; Amazon Web Services Announces Amazon API Gateway. *NewsRx*, 162.
- Olszewski, Pałka, & Turek. (2018). Solving “Smart City” Transport Problems by Designing Carpooling Gamification Schemes with Multi-Agent Systems: The Case of the So-Called “Mordor of Warsaw”. *Sensors*, 141.
- Petersen, S. (2015). AWS API Gateway, Device Farm Are Wins for Developers. *eWeek*, 1-1.
- Pressman, R. (2010). *Ingeniería del software*. México D.F.: The McGraw-Hill Companies.
- Reynoso, C. (2004).
- Riquelme. (2018). *Congestión vehicular*. Asuncion: ABC Color.
- Rodriguez, S. (23 de abril de 2012). *Publímetro*. Obtenido de Ocho datos de San Isidro en su aniversario: <https://publimetro.pe/actualidad/ocho-datos-san-isidro-su-aniversario-4840-noticia/>
- Savage, N. (2018). Going Serverless. *Communications of the ACM.*, 15-16.
- University of Alabama. (2015). *The Future of Mobile Application*. Birmingham.

- Vargas, F., Gaetan, G., & Saldaño, V. (2018). Usando Personas para Mejorar la Experiencia de Usuario de una Aplicación Móvil de Carpooling. *Informes Científicos Técnicos - UNPA*, 96-116.
- Vidal, C. (20 de Abril de 2015). *¿Qué es una pasarela de pagos y para qué sirve?*
Obtenido de <https://blog.pyme.pe/pasarela-de-pagos/>
- Vigouroux, C. (2019). *Aprender a desarrollar con JavaScript (3ª edición)*. ENI.
- Wikipedia. (2016). *Wikipedia*. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil
- Wittig, M., & Wittig, A. (2018). *Amazon Web Services in Action*. Manning Publications.
- Wittmann, R. (2006). ¿Hubo una revolución en la lectura a finales del siglo XVIII? En G. Cavallo, & R. Chartier, *Historia de la lectura en el mundo occidental* (págs. 435-472). México D.F.: Santillana.

BIBLIOGRAFÍA

- AWS. (2019). *Amazon Web Services*. Obtenido de Amazon Web Services:
<https://docs.aws.amazon.com/lambda/latest/dg/lambda-dg.pdf>
- Cantelon, M., Harter, M., Holowaychuk, T., & Rajlich, N. (2014). *Node.js in Action*. Manning.
- Coronel Castillo, E. G. (2011). *Desarrollando soluciones con Java*. Lima: Macro.
- García Nieto, J. P. (2013). *Consturye tu Web comercial: de la idea al negocio*. Madrid: RA-MA.
- Gestión. (2018). *El 72% de los limeños termina estresado a causa del tráfico, según un estudio*. Obtenido de <https://gestion.pe/peru/lima-72-ciudadanos-estresa-causa-traffic-estudios-nndc-245062>
- Gestión. (16 de Marzo de 2019). *Parque automotor se renueva en 6% al año, cuando debería hacerlo en 10%*. Obtenido de <https://gestion.pe/economia/parque-automotor-renueva-6-ano-deberia-hacerlo-10-261551-noticia/>
- Gestión, R. (07 de 06 de 2019). <http://gestion.pe>. Obtenido de Gestión Perú:
<https://gestion.pe/tendencias/lima-ciudades-peor-traffic-269483>
- Gilfillan, I. (2003). *La Biblia MySQL*. ANAYA.
- Google. (2010). *google.com*. Obtenido de Google.
- Hebuterne, S. (2016). *Android - Guía de desarrollo de aplicaciones Java para Smartphones y Tabletas (3ª edición)*. ENI Ediciones.
- Mozilla. (2019). *mozilla.org*. Obtenido de mozilla.org:
<https://developer.mozilla.org/es/docs/Web/JavaScript>
- Pressman, R. (2010). *Ingeniería del software*. México D.F.: The McGraw-Hill Companies.
- Vigouroux, C. (2019). *Aprender a desarrollar con JavaScript (3ª edición)*. ENI.
- Wittig, M., & Wittig, A. (2018). *Amazon Web Services in Action*. Manning Publications.
- Gutiérrez, L. E. (2010). *Arquitectura Software. Investigación Aplicada a la Construcción de Marcos de Trabajo-Bucaramanga, Colombia:(Sic) Editorial Ltda.*
- van Baarsen, J. (2014). *GitLab Cookbook*. Packt Publishing.



ANEXOS

Anexo 1 - Encuesta a los Trabajadores de San Isidro

Total de encuestas: 14

Número de preguntas: 7

Empresas: Rímac Seguros, Indra

- **Pregunta 1**

¿Qué movilidad normalmente usa para trasladarse de su centro de trabajo a su hogar?	
Opciones	Respuestas (#)
Corredor/Metropolitano	7
Bus	2
Colectivo	3
Taxi común	0
Taxi por aplicativo	0
Uber pool	2

Respuestas (#)	
Media	3
Error típico	1.471960144
Mediana	2.5
Desviación estándar	2.943920289
Varianza de la muestra	8.666666667
Curtosis	1.5
Coefficiente de asimetría	0.940660921
Rango	7
Mínimo	0
Máximo	7
Suma	12
Cuenta	4

- **Pregunta 2**

¿Cuál es el tiempo aproximado que demora/espera para tomar su movilidad?	
Opciones	Respuestas (#)
< 5 minutos	2
Entre 5 y 10 minutos	3
Entre 10 y 15 minutos	5
> 15 minutos	4

<i>Respuestas (#)</i>	
Media	3.5
Error típico	0.645497224
Mediana	3.5
Desviación estándar	1.290994449
Varianza de la muestra	1.666666667
Curtosis	-1.2
Rango	3
Mínimo	2
Máximo	5
Suma	14
Cuenta	4

- **Pregunta 3**

¿Qué aspectos cree usted que es más importante al momento de movilizarse a su destino? Marque las dos más importantes.	
Opciones	Respuestas (#)
Seguridad	9
Comodidad	9
Economía	6
Tiempo	4

<i>Respuestas (#)</i>	
Media	7
Error típico	1.224744871
Mediana	7.5
Moda	9
Desviación estándar	2.449489743
Varianza de la muestra	6
Curtosis	-2.944444444
Coefficiente de asimetría	-0.544331054
Rango	5
Mínimo	4
Máximo	9
Suma	28
Cuenta	4

- **Pregunta 4**

¿Realizaría viajes en conjunto con colegas de su misma empresa o sector empresarial?	
Opciones	Respuestas (#)
Si	10
No	0
Talvez	4

<i>Respuestas (#)</i>	
Media	4.666666667
Error típico	2.905932629
Mediana	4
Desviación estándar	5.033222957
Varianza de la muestra	25.33333333
Coefficiente de asimetría	0.585582726
Rango	10
Mínimo	0
Máximo	10
Suma	14
Cuenta	3

- **Pregunta 5**

¿Estaría interesado en utilizar nuestra aplicación?	
Opciones	Respuestas (#)
Si	12
No	0
Talvez	2

<i>Respuestas (#)</i>	
Media	4.666666667
Error típico	3.711842909
Mediana	2
Desviación estándar	6.429100507
Varianza de la muestra	41.33333333
Coefficiente de asimetría	1.545392526
Rango	12
Mínimo	0
Máximo	12
Suma	14
Cuenta	3

- **Pregunta 6**

¿Qué características de la aplicación cree que son las más importantes? Puede escoger más de una.	
Opciones	Respuestas (#)
Confianza	6
Viajar cómodo	7
Programar viaje	3
Tarifas bajas	6
Relaciones laborales	3
Medio ambiente	5

<i>Respuestas (#)</i>	
Media	5.5
Error típico	0.866025404
Mediana	6
Moda	6
Desviación estándar	1.732050808
Varianza de la muestra	3
Curtosis	2.888888889
Coefficiente de asimetría	-1.539600718
Rango	4
Mínimo	3
Máximo	7
Suma	22
Cuenta	4

- **Pregunta 7**

¿Cuánto estaría dispuesto a pagar por una ruta promedio (Ej. San Isidro – Av. La Molina)?	
Opciones	Respuestas (#)
< 3 soles	2
Entre 3 y 4 soles	6
Entre 3 y 5 soles	5
> 5 soles	1

<i>Respuestas (#)</i>	
Media	3.5
Error típico	1.190238071
Mediana	3.5
Desviación estándar	2.380476143
Varianza de la muestra	5.666666667
Curtosis	-4.339100346
Coefficiente de asimetría	0
Rango	5
Mínimo	1
Máximo	6
Suma	14
Cuenta	4

