

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



PROYECTO DE SISTEMA DE CONTROL DE FLUJO DE TRÁNSITO AÉREO PARA AEROPUERTOS

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

Alfredo Henry Rivera Herbozo

Código 20052077

Asesor

Edwin Manuel, Álvarez Valdivia

Lima – Perú

Abril de 2022

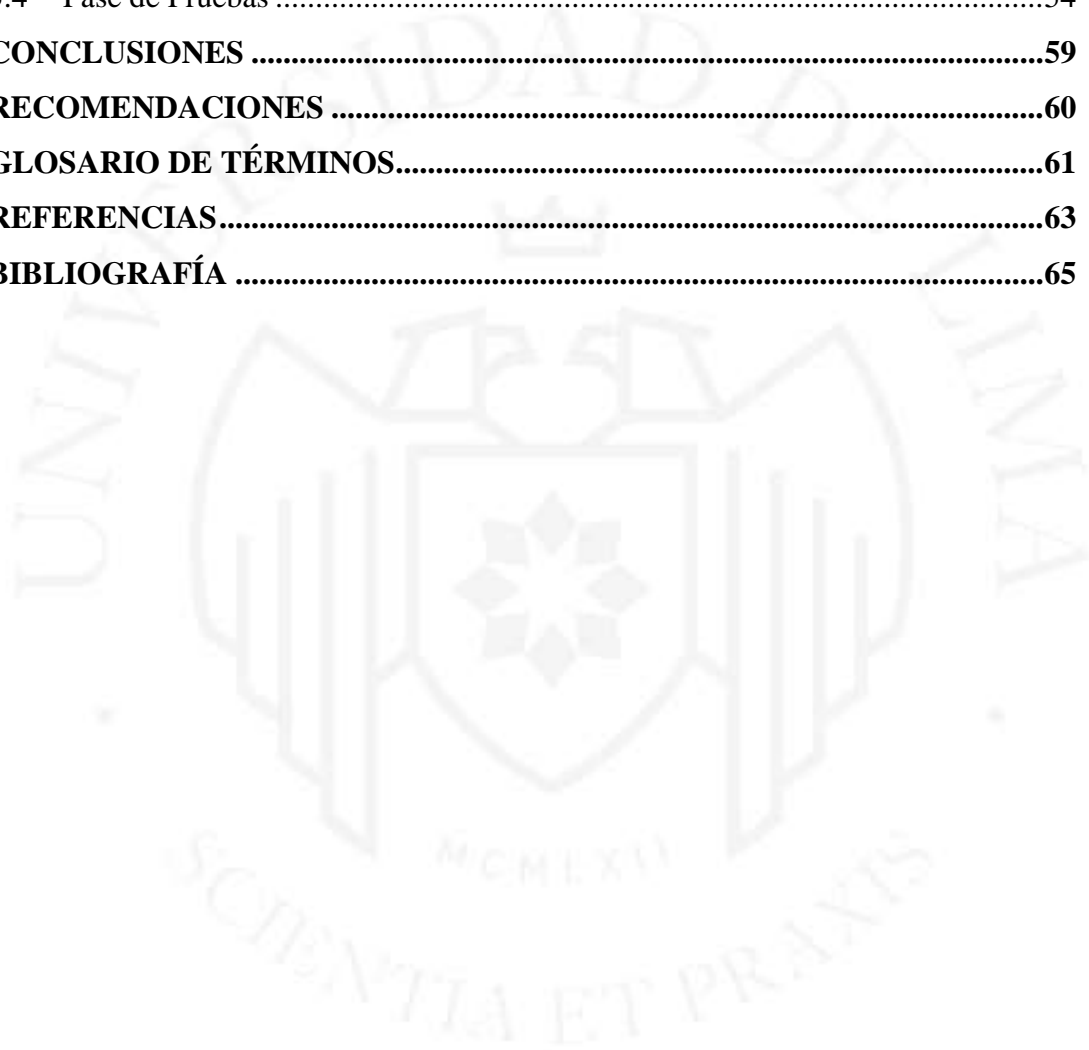


**AIR TRAFFIC FLOW MANAGEMENT
SYSTEM PROJECT FOR AIRPORTS**

TABLA DE CONTENIDO

RESUMEN	x
ABSTRACT	xi
CAPÍTULO I: INTRODUCCIÓN	1
CAPÍTULO II: FUNDAMENTOS TEÓRICOS	2
2.1 Capacidad de Pista.....	2
2.2 Cloud Computing	2
2.3 Api Restful	7
2.4 Design Thinking	8
CAPÍTULO III: DEFINICIÓN DEL MODELO DE NEGOCIO	10
3.1 Desarrollo de la Propuesta de Valor (VPC)	10
3.1.1 Fundamentación de la deseabilidad de la propuesta del valor.	12
3.1.2 Beneficios esperados por el cliente:	12
3.2 Desarrollo del Modelo de negocio (BMC).....	12
3.2.1 Modelo de negocio actual:	13
3.2.2 Modelo de negocio propuesto:	14
3.1. Viabilidad del nuevo modelo	14
3.2. KPI's de validación para continuidad	18
3.3. Límites de aceptación del Sistema	18
CAPÍTULO IV: DEFINICIÓN DEL PROYECTO	20
4.1 Definición del proyecto	20
4.2 Objetivos del Proyecto	21
4.2.1 Objetivo general	21
4.2.2 Objetivos específicos.....	21
4.3 Fundamento de la factibilidad del proyecto	22
4.4 Roles y responsabilidades del equipo del proyecto:.....	22
4.5 Cronograma y riesgos iniciales del proyecto	22
4.6 Recursos necesarios.....	25
CAPÍTULO V: DESARROLLO DEL PROYECTO	26
5.1 Desarrollo del prototipo.....	26
5.2 Mapa del Prototipo a alto nivel	26
5.2.1 Prototipo 1	27

5.2.2 Prototipo 2	33
5.2.3 Prototipo 3	35
5.3 Fase de Desarrollo del Sistema Web	37
5.3.1 Diagramas de Casos de Uso	38
5.3.2 Diagramas de secuencia	47
5.3.3 Pila de Producto.....	50
5.3.4 Planificación del Sprint	52
5.4 Fase de Pruebas	54
CONCLUSIONES	59
RECOMENDACIONES	60
GLOSARIO DE TÉRMINOS.....	61
REFERENCIAS.....	63
BIBLIOGRAFÍA	65



ÍNDICE DE TABLAS

Tabla 2.1 <i>Clasificación de proveedores SaaS</i>	4
Tabla 2.2 <i>Clasificación de proveedores PaaS</i>	5
Tabla 2.3 <i>Clasificación de proveedores IaaS</i>	5
Tabla 3.1 <i>Tabla de Ingresos</i>	14
Tabla 3.2 <i>Ingresos Mensuales del primer año</i>	15
Tabla 3.3 <i>Ventas de los 5 primeros años</i>	16
Tabla 3.4 <i>Costos asociados al proyecto</i>	16
Tabla 3.5 <i>Resultado de Ingresos y Gastos</i>	16
Tabla 3.6 <i>Evaluación</i>	17
Tabla 4.1 <i>Valorización de Riesgos</i>	23
Tabla 4.2 <i>Rango de Efectividad</i>	24
Tabla 4.3 <i>Controles y Riesgos</i>	24
Tabla 5.1 <i>Lista de Prototipos</i>	26
Tabla 5.2 <i>CU01 Iniciar Sesión con correo y contraseña</i>	39
Tabla 5.3 <i>CU02 Ver vuelos internacionales</i>	40
Tabla 5.4 <i>CU03 Ver detalle vuelo internacional</i>	41
Tabla 5.5 <i>CU04 Filtrar Solicitudes</i>	41
Tabla 5.6 <i>CU05 Agregar Vuelo</i>	42
Tabla 5.7 <i>CU06 Cancelar Solicitud</i>	42
Tabla 5.8 <i>CU07 Agregar Solicitud</i>	44
Tabla 5.9 <i>CU08 Aceptar Solicitud</i>	46
Tabla 5.10 <i>Roles Scrum</i>	50
Tabla 5.11 <i>Historias de Usuario</i>	51
Tabla 5.12 <i>Sprint 1</i>	52
Tabla 5.13 <i>Sprint 2</i>	52

ÍNDICE DE FIGURAS

Figura 2.1 <i>Tipos de Servicios</i>	3
Figura 2.2 <i>API y sus aplicaciones</i>	7
Figura 3.1 <i>Mapa de Actores</i>	10
Figura 3.2 <i>Usuario Persona</i>	11
Figura 3.3 <i>Propuesta de Valor</i>	11
Figura 3.4 <i>Modelo del Negocio Actual</i>	13
Figura 3.5 <i>Modelo de Negocio Propuesto</i>	14
Figura 4.1 <i>Cronograma de Actividades</i>	23
Figura 5.1 <i>Adobe XD como herramienta</i>	27
Figura 5.2 <i>Prototipo 1</i>	27
Figura 5.3 <i>Pantalla login</i>	28
Figura 5.4 <i>Pantalla de solicitudes Pro1</i>	28
Figura 5.5 <i>Pantalla de solicitudes expandida bloque naranja Pro1</i>	29
Figura 5.6 <i>Pantalla de solicitudes expandida bloque verde Pro1</i>	29
Figura 5.7 <i>Pantalla solicitudes aviso Pro1</i>	30
Figura 5.8 <i>Pantalla solicitudes cancelado Pro1</i>	31
Figura 5.9 <i>Pantalla solicitudes usuario Lima Pro1</i>	31
Figura 5.10 <i>Pantalla solicitudes expandida Lima Pro1</i>	32
Figura 5.11 <i>Pantalla solicitudes Lima alerta Pro1</i>	32
Figura 5.12 <i>Pantalla solicitudes Lima aceptado Pro1</i>	33
Figura 5.13 <i>Prototipo 2</i>	33
Figura 5.14 <i>Pantalla solicitudes Pro2</i>	34
Figura 5.15 <i>Pantalla solicitudes bloque2 Pro2</i>	34
Figura 5.16 <i>Pantalla solicitudes bloque1 Pro2</i>	35
Figura 5.17 <i>Prototipo 3</i>	35
Figura 5.18 <i>Pantalla Login Pro3</i>	36
Figura 5.19 <i>Pantalla solicitudes Pro3</i>	36
Figura 5.20 <i>Pantalla Agregar Solicitudes Pro3</i>	37
Figura 5.21 <i>Arquitectura General de la Aplicación</i>	38
Figura 5.22 <i>Caso de uso Autenticación de usuario</i>	39
Figura 5.23 <i>Funciones en Solicitudes</i>	40

Figura 5.24 <i>Caso de uso Agregar solicitud</i>	43
Figura 5.25 <i>Aceptar Solicitud</i>	45
Figura 5.26 <i>Diagrama de Secuencia Inicio de Sesión</i>	47
Figura 5.27 <i>Diagrama de Secuencia Registro de Solicitud</i>	47
Figura 5.28 <i>Diagrama de Secuencia Aceptar Solicitud</i>	48
Figura 5.29 <i>Modelo de Datos</i>	49
Figura 5.30 <i>Cloud Firestore</i>	49
Figura 5.31 <i>Jira Desarrollo del Software</i>	53
Figura 5.32 <i>Jira Pantalla Login</i>	53
Figura 5.33 <i>Interface Login</i>	54
Figura 5.34 <i>Jira Pruebas del Sistema</i>	54
Figura 5.35 <i>Jira pruebas funcionalidad Login Firestore</i>	55
Figura 5.36 <i>Cloud Firestore usuarios</i>	55
Figura 5.37 <i>Firestore Authentication</i>	56
Figura 5.38 <i>Credenciales Login</i>	57
Figura 5.39 <i>Header Nombre Aeropuerto</i>	57
Figura 5.40 <i>Lista de solicitudes</i>	58

ÍNDICE DE ANEXOS

Anexo 1 Creación de cuenta en Avionstack API.....	68
Anexo 2 Crear Objetos y clases para que el sistema interactúe con Firebase	69



RESUMEN

La constante demora en la confirmación de la hora de aterrizaje de los aviones, la falta de abastecimiento de personal de la torre de control para las coordinaciones, así como también la insuficiencia de documentación adecuada que genere una mayor facilidad de acceso a las gestiones aéreas, son las continuas dificultades que atraviesa el Aeropuerto Internacional. Teniendo en cuenta esta actual problemática, el presente proyecto tiene como objetivo general informar sobre el tráfico aéreo del Aeropuerto y como objetivos específicos facilitar y documentar el trabajo del área de *Flow Control* del Aeropuerto a la hora de gestionar las horas de aterrizaje y despegue de los vuelos que arriban desde los distintos aeropuertos del Perú. Para ello, se propone el desarrollo de un sistema *web* que mejore el proceso de trabajo realizado por los operarios a través del uso de Tecnología *Cloud Computing* usando una nube pública bajo el modelo PaaS (Plataforma como Servicio), debido a que la aplicación se encuentra en servidores que facilitan la gestión y desarrollo de las fuentes del sistema y sus datos. Para la elaboración de este proyecto se utilizó el marco de trabajo *Scrum* que permite gestionar tanto el desarrollo como las pruebas. Dentro de las mejoras obtenidas para con el cliente se presentó el desarrollo de tres prototipos que se fueron ajustando a la realidad, necesidad y visión del usuario. Si bien el proyecto tenía un alto grado de automatización en el proceso principal del encolamiento de solicitudes de horas de aterrizaje, este fue modificado a solicitud del usuario el cual requiere tener un mayor control sobre la edición de las solicitudes.

Palabras clave: IS Practicum, Data / Information Management, Operadores de vuelo, Aeropuertos Internacionales, gestión de vuelos

ABSTRACT

The constant delay in confirming the aircraft landing time, the lack of supply of control tower personnel for coordination purposes, as well as the insufficient adequate documentation that generates greater ease of access to air procedures, these are the continuous difficulties that the International Airport is going through. Taking into account this current problem, the general objective of this project is to inform about the air traffic of the Airport and as specific objectives to facilitate and document the work of the Flow Control area of the Airport when managing the hours landing and take-off of flights arriving from different airports in Peru. For this, the development of a web system is proposed that improves the work process carried out by the operators through the use of Cloud Computing Technology using a public cloud under the PaaS (Platform as a Service) model, because the application is on servers that facilitate the management and development of system sources and their data. For the development of this project, the Scrum framework was used, which allows managing both development and testing. Among the improvements obtained for the client, the development of three prototypes was presented that were adjusted to the reality, need and vision of the user. Although the project had a high degree of automation in the main process of the landing time request queuing, it was modified at the request of the user, which requires greater control over the editing of the requests.

Keywords: IS Practicum, Data / Information Management, Flight Operators , International Airports, flight management

CAPÍTULO I: INTRODUCCIÓN

Actualmente el Perú cuenta con dos aeropuertos internacionales, este proyecto se enfoca principalmente en uno de ellos, el cual en el año 2016 implementó un área llamada *Flow Control*, teniendo como razones principales para su implementación, la demora en la confirmación de la hora de aterrizaje de los aviones. los cuales se encontraban a la espera, volando en zonas aledañas al Aeropuerto, y la falta de abastecimiento de la torre de control del Aeropuerto Internacional para las coordinaciones, teniendo como tarea principal el mantener el orden del aterrizaje y poder generar rutas o vectores para los aviones en vuelo que se encuentran por aterrizar.

El área *Flow Control*, tiene como principal tarea la coordinación de los vuelos tanto nacionales e internacionales. La implementación de esta área fue efectiva cuando la cantidad de vuelos en el día no era considerable, manteniendo en promedio 20 vuelos por hora; sin embargo, actualmente el número de llamadas se ha visto incrementado y conforme pasa el tiempo se evidencia una mayor demanda de atención por lo que el aumento seguirá haciéndose más evidente generando inconvenientes y una inadecuada coordinación con los operarios de los otros aeropuertos del interior del país.

Dentro de los principales reclamos se pone en manifiesto la no disponibilidad ni facilidad de acceso a las líneas telefónicas que maneja el aeropuerto de Lima, las cuales se mantienen ocupadas durante largos periodos de tiempo por continuas coordinaciones con varios aeropuertos del interior del país o a la inadecuada gestión de la hora de despegue, no manteniéndose el horario establecido previamente lo que genera retraso en los vuelos.

Ante lo anteriormente mencionado, siendo el principal segmento de mercado los aeropuertos internacionales por la cantidad de vuelos que registra hacia y desde el interior del país, vemos la importancia de crear un sistema tecnológico de coordinaciones que permita concertar e indicar información de los vuelos a través de ella, permitiendo una información compartida con todos los aeropuertos con el fin de que se dé a conocer la fluidez del tránsito aéreo en todas las torres de control para una mejor gestión y coordinación.

CAPÍTULO II: FUNDAMENTOS TEÓRICOS

2.1 Capacidad de Pista

Debido al aumento constante del aforo del tránsito aéreo, se han realizado diversos estudios con el fin de poder definir las tasas reales de crecimiento y conocer la funcionalidad de los índices que detalla en su sistema de gestión de tránsito aéreo. Según (Comando de la Aeronáutica, 2009), toma importancia el concepto de equilibrio entre la capacidad y demanda, entendiéndose como la capacidad que presenta el sistema de poder absorber el volumen de tránsito aéreo pronosticado. La exigencia de obtener una estabilidad entre estos dos elementos ha generado que se desarrollen diferentes métodos de cálculo de capacidad que se ven afectadas por diversos factores tales como la configuración, la geometría del aeropuerto, la organización del espacio aéreo, entre otros. Por otro lado, se toma en cuenta que en caso la capacidad de pista sea insuficiente para poder dar solución y atender las solicitudes que se presentan, se generaría una saturación del sistema que traería como consecuencia un mayor nivel de retraso; es por ello, que el Sistema de Gestión de Afluencia de Tránsito Aéreo (AFTM) debe conocer los valores de capacidad y la demanda disponibles de los aeropuertos que quedan bajo su jurisdicción, con el fin que se cree un adecuado balance que permita una afluencia en el tránsito aéreo.

2.2 Cloud Computing

También conocido como “La Nube”, es un servicio que tiene como principal componente o núcleo del negocio a las computadoras o servidores que utilizan la internet para poder estar en continua comunicación con el usuario las veinticuatro horas del día, los siete días de la semana.

Según (Pérez, 2012), una plataforma de *Cloud Computing* dinámicamente suministra, configura, reconfigura y retira los servidores según las necesidades. *Cloud Computing* también describe aplicaciones que son extendidas para ser accesibles a través de Internet. Estas aplicaciones del *Cloud* usan grandes servidores para el almacenamiento de datos y poderosos servidores que alojan aplicaciones y servicios Web.

Cualquiera con una conexión adecuada a Internet y un navegador estándar puede acceder a una aplicación del Cloud.

Podemos definir que *Cloud Computing* se divide en tres tipos:

- Nubes Públicas: Según Domínguez (2017) este es el tipo de implementación más común. Todos los recursos, en las que incluimos componentes de *Software*, *Hardware* e Infraestructura, que se ofrecen pertenecen a un proveedor que serán los encargados de gestionar los servicios a través de internet. Usualmente, estos productos son compartidos por varios clientes lo cual crea cierta dependencia e incluso inconvenientes debidos a que los recursos necesarios para su sistema pueden tener conflictos.
- Nubes Privadas: Usa todos los procesos básicos de una nube pública con acceso exclusivo de una empresa u organización (Domínguez, 2017). En el caso de contratar el servicio de un proveedor, estas manejan redes privadas, así como *Software* y *Hardware* dedicado.
- Nubes Híbridas: Este tipo de nube se caracteriza porque combina infraestructura tanto pública como privada. La información que se aloje en una de ellas dada la estructura, puede viajar de un entorno al otro. Cuando una organización opta por este servicio es debido a las exigencias del comercio.

Como definición, *Cloud Computing* abarca varios conceptos, pero cuando se trata de ofrecer el servicio (véase figura 2.1), existen tres modalidades:

Figura 2.1

Tipos de Servicios



Nota. Adaptada de Tipos de Nubes, servicios y proveedores por Open Webinars, s.f., OpenWebinars.net (<https://openwebinars.net/blog/tipos-de-cloud-computing/>)

- SaaS (*Software* como Servicio), refiere a la aplicación implementada y alojada en la nube donde los usuarios pueden tener acceso a Internet, la clasificación de los proveedores es múltiple como se señala en la Tabla 2.1

Tabla 2.1

Clasificación de proveedores SaaS

TIPO DE SERVICIO	EJEMPLO
Aplicaciones como sitios Web	Box.net, Microsoft Office Live, Facebook, LinkedIn, Twitter, MySpace, Zillow, Google Maps.
Colaboración y aplicaciones de oficina	Cisco WebEx Weboffice, Google Docs, Google Talk, IBM BlueHouse, Microsoft Exchange Online, RightNow, Gmail, Microsoft Hotmail, Yahoo! Mail.
Servicios de pago	Amazon Flexible Payments Service (Amazon FPS), Amazon DevPay.
Software basado en Web integrable a otras aplicaciones	Flickr Application Programming Interface (API), Google Calendar API, Salesforce.com AppExchange, Yahoo! Maps API, Zembly.

Nota. De Colombia, por Rosales, E., 2010. UnaCloud: Infraestructura como Servicio para Cloud Computing oportunista.

- PaaS (Plataforma como Servicio), refiere a las máquinas implementadas de algún sistema operativo que el cliente solicite u otros requerimientos de la misma índole, la clasificación de los proveedores es múltiple como se señala en la Tabla 2.2

Sobre PaaS “Es un entorno de desarrollo e implementación completo en la nube, con recursos que permiten entregar todo, desde aplicaciones sencillas basadas en la nube hasta aplicaciones empresariales sofisticadas habilitadas para la nube. Usted le compra los recursos que necesita a un proveedor de servicios en la nube, a los que accede a través de una conexión segura a Internet, pero solo paga por el uso que hace de ellos” (Microsoft Azure, s.f., ¿Qué es Paas?)

Tabla 2.2*Clasificación de proveedores PaaS*

TIPO DE SERVICIO	EJEMPLO
Plataformas de desarrollo	Amazon Simple Queue Service (Amazon SQS), Amazon Simple Storage Service (Amazon S3), Google App Engine, GRIDS Lab Aneka.
Base de datos	Amazon SimpleDB, Big Table, Microsoft SQL Azure Database.
Cola de mensajes	Amazon Simple Queue Service (Amazon SQS).
Servidores de aplicaciones	NetSuite Business Operating System (NS-BOS).

Nota. De Colombia, por Rosales, E., 2010. UnaCloud, Infraestructura como Servicio para Cloud Computing oportunista.

- IaaS (Infraestructura como Servicio), refiere al hardware o máquinas físicas, del almacenamiento, de los equipos de red necesarios para el funcionamiento de los sistemas del cliente, , la clasificación de los proveedores es múltiple como se señala en la Tabla 2.3

Tabla 2.3*Clasificación de proveedores IaaS*

TIPO DE SERVICIO	EJEMPLO
Procesamiento	Amazon Elastic Compute Cloud (Amazon EC2), Sun Network.com (Sun Grid), ElasticHosts, Eucalyptus, Nimbus, OpenNebula, Enomaly.
Distribución de contenido a través de servidores virtuales	Akamai, Amazon CloudFront Beta.
Almacenamiento	Amazon Simple Storage Service (Amazon S3), Amazon SimpleDB, Amazon Elastic Block Store, Microsoft SkyDrive, Youtube, Nirvanix Storage Delivery Network, Microsoft Live Mesh Beta, Flickr.
Administración de sistemas	Elastra, Engine Yard, FlexiScalabl, Grid Layer, Joyent, Mosso, Savvis Virtual Intelligent Hosting.
Administración de alojamiento	Digital Realty Trust, GoDaddy.com, Layered Technology.
Alojamiento autónomo	Rackspace, Savvis Virtual Intelligent Hosting, Terremark Worldwide, FlexiScalable, 1&1 Internet.
VLAN (Virtual Local Area Network)	CohesiveFT.

Nota. De Colombia, por Rosales, E., 2010. UnaCloud: Infraestructura como Servicio para Cloud Computing oportunista.

Ventajas de adquirir un servicio en la nube:

- Para las nuevas empresas, adquirir recursos de tipo *Hardware* y *Software* suele ser bastante costoso, ya que requiere contratar personal capacitado para su implementación; así como también, hace referencia a la necesidad de espacios amplios que permita distribuir el equipo necesario teniendo en cuenta que a futuro estos pueden ir en aumento en base a la necesidad que se tenga.
- Es de rápida implementación. Si bien esto dependerá de la modalidad de servicio escogida, al ser comparada con la adquisición de equipos o servidores propios, va a presentar una gran diferencia como el tiempo en la adquisición de los equipos, el traslado, la ubicación y la adecuación de los mismos, la interacción de las computadoras con el servidor, la publicación del servidor en internet y en caso se desee que los usuarios puedan tener acceso desde el exterior se deberá realizar un soporte de la seguridad de la nueva implementación.
- La escalabilidad en la nube, hace que el servicio se vaya adecuando a la carga que tiene la aplicación como por la alta cantidad de usuarios conectados, así evitar que se encuentre fuera de servicio por largos periodos de tiempo lo que trae consigo pérdidas económicas.
- Soporte en la nube, que dependerá de la modalidad contratada, las responsabilidades se irán transfiriendo del cliente a la empresa contratada, y esta suele tener una duración de 24 horas, todos los días.
- Migración en la nube. Actualmente la mayoría de sus servicios maneja contenedores que permiten la migración de los sistemas independientemente de los sistemas operativos, evitando así problemas de compatibilidad con los nuevos sistemas o que se requiera de parches o nuevas versiones de SO.

Desventajas del servicio en la nube:

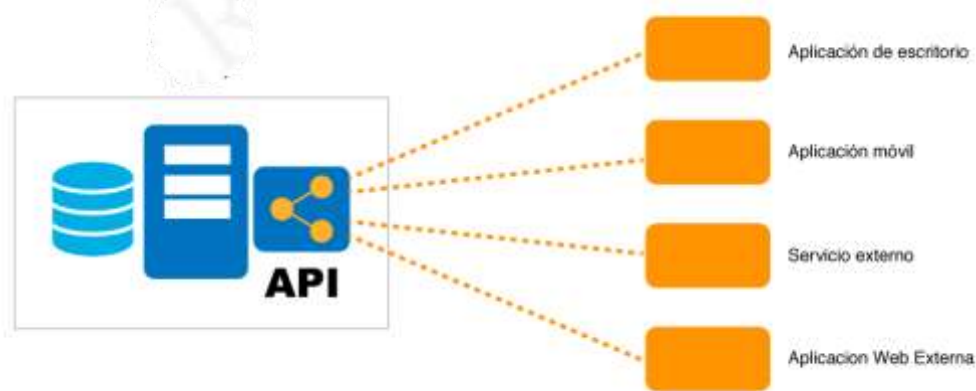
- La información que se guarda en los servidores de terceros no queda totalmente segura, ya que el acceso y alteración de los datos es un riesgo que todo cliente debe documentar para así tomar acciones correctivas.
- La dependencia con la empresa que brinda el servicio del *Cloud* va creciendo a medida que se va adquiriendo mayores servicios y que puede verse agravada cuando se desea migrar a otro *Cloud* que no maneje los mismos servicios o presente costos mayores.

2.3 Api Restful

“Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful” (Red Hat, 2020, ¿Qué es una API de REST?). Esta permite integrar distintos tipos de software donde cada uno de ellos puede estar desarrollado en diferentes lenguajes de programación (véase figura 2.2)

Figura 2.2

API y sus aplicaciones



Nota. Adaptada de RestApi de Wordpress, que es y cómo usarla por WebEmpresa, s.f., WebEmpresa.com (<https://www.webempresa.com/blog/rest-api-de-wordpress.html/>)

Tiene como principal objetivo el entregar datos o el ser consumida por algún sistema; para esto, las API'S se han clasificado en dos grupos: las API privadas y las API públicas, esto se da con el único fin de salvaguardar la seguridad de los servidores que alojan tanto los datos como los servicios.

La principal diferencia entre estos dos tipos de API, es que para una privada se es necesario una llave o token para poder consumirla o acceder a los datos. A diferencia de las claves de base de datos, estas tienen un tiempo de vida y la cantidad de caracteres es superior, lo cual, de cierta manera, dificulta poder descubrirlas y romper la seguridad que manejan.

En cuanto a REST, este no es un protocolo, es una arquitectura que trabaja sobre un protocolo, el cual es el HTTP. Esto puede generar una confusión en los términos ya que REST hereda los distintos métodos de comunicación: GET, POST, PUT, DELETE, más otros propios de REST.

Una de las características de una API REST es el código del estado de la solicitud por parte del consumidor; de esta manera, permitirá saber si la consulta es válida; es decir, si está bien elaborada, si el servidor está ocupado, así como también el conocer otros estados que han sido clasificados y divididos, y de esta manera poder mapear la solicitud en caso se hayan recibido errores.

La respuesta a una solicitud viaja en el cuerpo y usa el formato “json”, el mismo que actualmente es el estándar en temas de API REST.

2.4 Design Thinking

Es una metodología sencilla de entender, la misma que permite encontrar soluciones a los problemas desde una perspectiva centrada en la persona a los distintos escenarios que le puedan ocasionar dificultad a la hora de realizar sus necesidades ya sea en el trabajo o en su día a día.

Posee cinco etapas que no necesariamente son consecutivas, y que si es necesario se puede ir repitiendo con el fin de encontrar la solución más adecuada al objetivo trazado.

Las cinco etapas universalmente aceptadas son (García Ramos, 2017):

1. **Empatiza:** entender de manera profunda las necesidades y problemas de las personas y descubrir *insights* relevantes.
2. **Define:** sintetizar la información y determinar los objetivos, basándose en las distintas necesidades y los *insights* descubiertos.
3. **Idea:** idear y pensar cómo solucionar el problema. En esta fase la cantidad es más importante que la calidad, es importante contar con un equipo multidisciplinario que puedan aportar distintas visiones al proceso.
4. **Prototipo:** construir las ideas más resaltantes o con un alto potencial, ya sea en papel o usando herramientas tecnológicas, de tal manera que puedan ayudar a mejorar los resultados y en todo caso saber si vamos en la dirección correcta.
5. **Testea:** comprobar si nuestras soluciones realmente satisfacen las necesidades de las personas para las que diseñamos. Es una fase clave en el proceso, pues

nos entrega información sobre qué elementos fallan, dónde incidir y si hay que volver a una fase anterior.

Cada una de estas fases maneja distintas herramientas que ayudan con el proceso, la recolección de datos y la documentación para encontrar la solución al problema planteado.



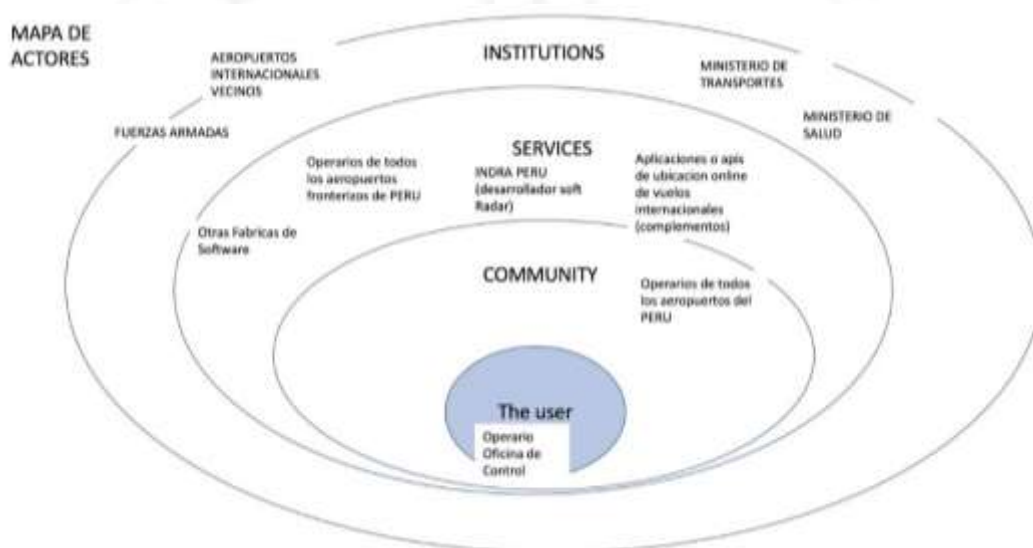
CAPÍTULO III: DEFINICIÓN DEL MODELO DE NEGOCIO

3.1 Desarrollo de la Propuesta de Valor (VPC)

Para el presente proyecto se realizaron entrevistas sobre el proceso actual de las coordinaciones con dos operarios del Aeropuerto, obteniéndose el Mapa de actores (véase figura 3.1)

Figura 3.1

Mapa de Actores



En la Figura 3.1, se muestra el mapa de actores, tomando como usuario al operario de oficina de control. Centrándonos en el usuario que es el operario de la Oficina de Control, podemos a continuación resolver nuestro diagrama de usuario persona, como se señala en la Figura 3.2

Figura 3.2

Usuario Persona

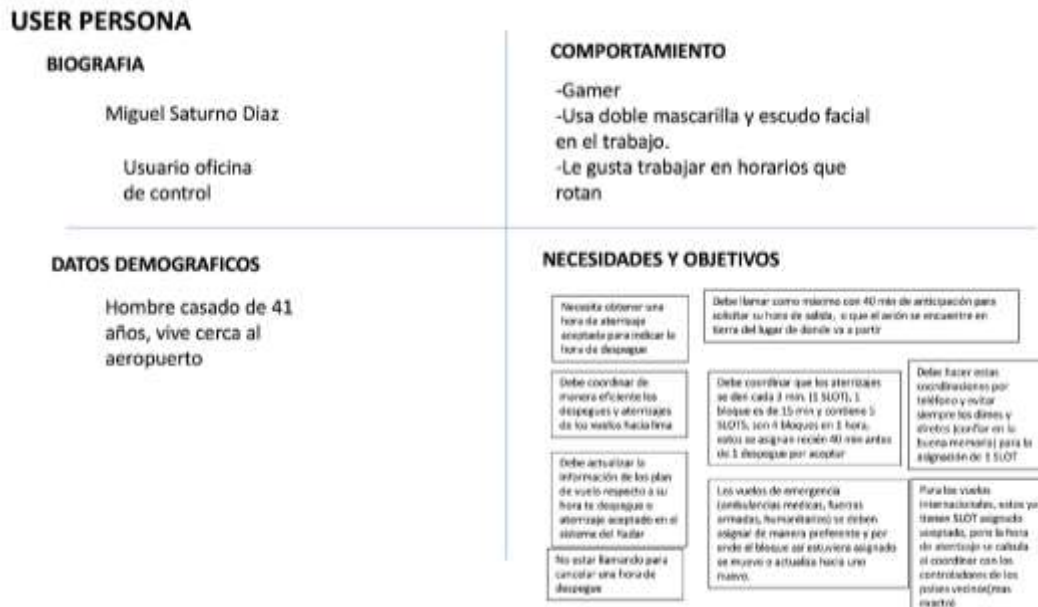


Figura 3.2, se muestra el usuario persona, con toda esta información recabada, se puede definir la propuesta de valor (véase figura 3.3) que es la aplicación web que permite registrar, aceptar y visualizar las solicitudes de hora de aterrizaje y despegue.

Figura 3.3

Propuesta de Valor

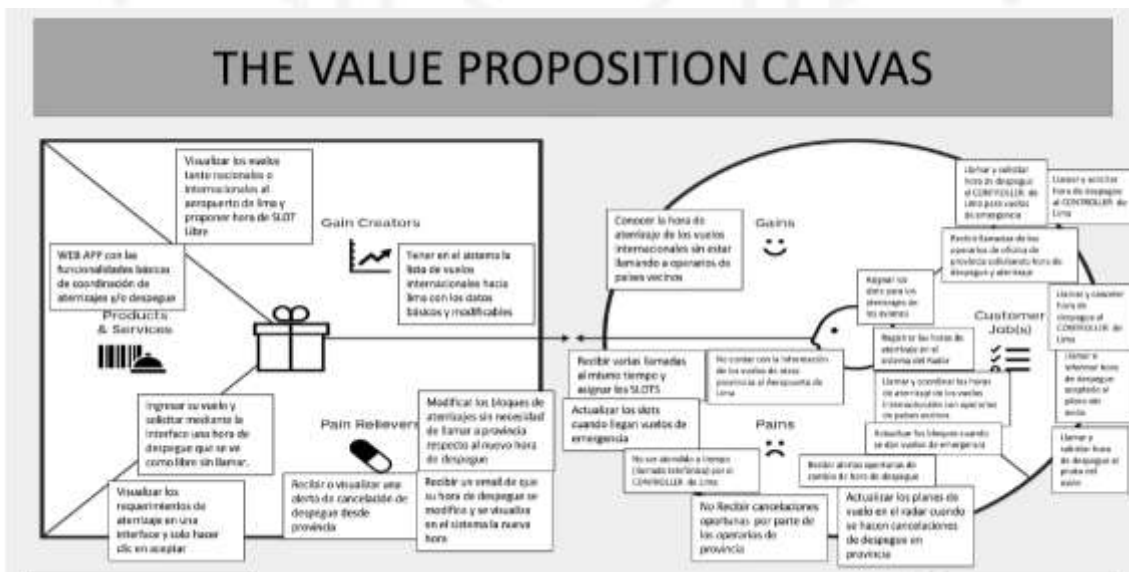


Figura 3.3, se muestra la propuesta de valor, podemos observar las tareas del usuario, los dolores que tiene al realizar sus actividades, que considera el cómo ganancias, a partir de

eso podemos darles soluciones a esos dolores, y a lo que él considera ganancias, finalmente la solución como tal.

3.1.1 Fundamentación de la deseabilidad de la propuesta del valor.

Si bien en la actualidad este proceso es manejado según sus recursos, se evidencia que existen reiteradas quejas y observaciones por parte de los operarios de los aeropuertos del interior del país, generando un malestar continuo debido al incremento de vuelos. Ante ello, se conoce que no existe un interés en contar con un mayor número de controladores; por el contrario, se prioriza la manera de ir automatizando el proceso e ir generando un historial de datos que ayuden con las estadísticas a partir de esta información.

3.1.2 Beneficios esperados por el cliente:

- El operario dejará de recibir extensas llamadas para coordinar las horas de despegue y/o aterrizaje.
- El operario podrá visualizar en el sistema las horas de aterrizaje de los vuelos internacionales.

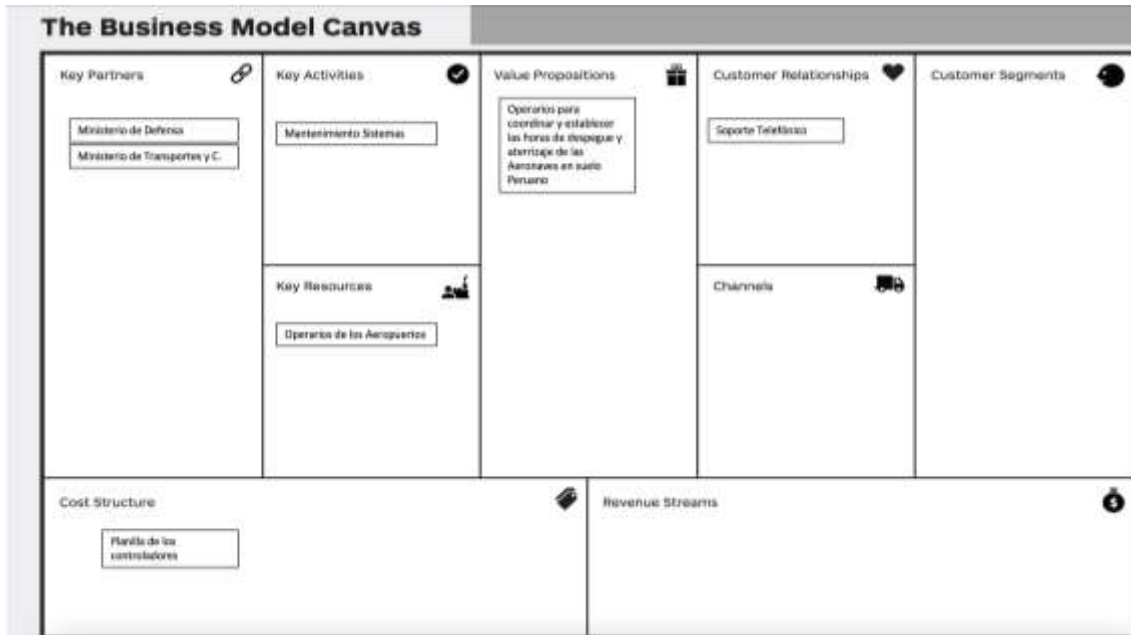
3.2 Desarrollo del Modelo de negocio (BMC)

Para el desarrollo del proceso de mejora, es necesario identificar el modelo de negocio actual en la Figura 3.4 y el modelo de negocio que se daría al implementar esta solución, ver Figura 3.5.

3.2.1 Modelo de negocio actual:

Figura 3.4

Modelo del Negocio Actual

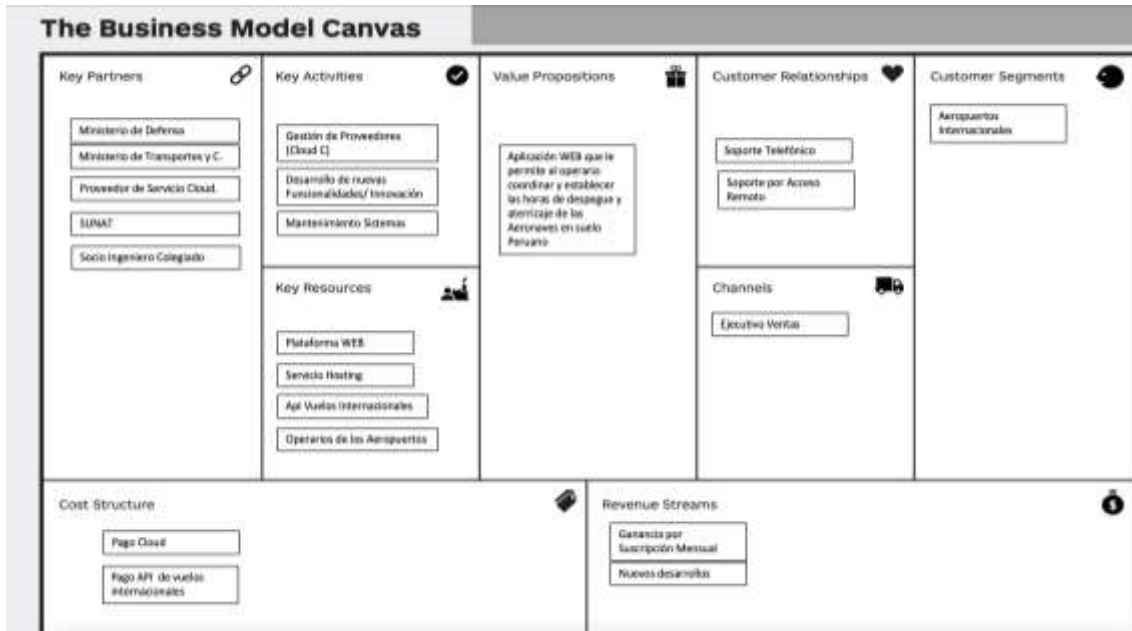


En la Figura 3.4 muestra el modelo actual donde el proceso es enteramente realizado por los operarios del aeropuerto.

3.2.2 Modelo de negocio propuesto:

Figura 3.5

Modelo de Negocio Propuesto



En la Figura 3.5 muestra el modelo propuesto donde el proceso es documento y coordinado desde la aplicación web que se propone en el presente trabajo.

3.1. Viabilidad del nuevo modelo

A continuación en la Tabla 3.1, se muestran los productos y servicios que forman parte de este proyecto. Se pone en conocimiento que la moneda utilizada es el sol peruano. La Tabla 3.1, muestra los productos ofrecidos y sus respectivos precios y unidades.

Tabla 3.1

Tabla de Ingresos

#	Descripción	Unidad Medida	Precio de Venta
1	Sistema Web de Coordinación	mensual	S/.3,500.00
2	Desarrollo de mejoras	mensual	S/.1,700.00
3	Suscripción aeropuertos Internacionales	mensual	S/.3,500.00

A continuación en la Tabla 3.2, se observa el detalle de las ventas en el primer año.

Tabla 3.2*Ingresos Mensuales del primer año*

Ventas en S/ Año 1	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12	Total
Ventas en Soles	S/ 0	S/ 0	S/ 0	S/ 3,500	S/ 3,500	S/ 3,500	S/ 5,200	S/ 5,200	S/ 5,200	S/ 5,200	S/ 5,200	S/ 5,200	S/ 41,700
Cobranza al contado	S/ 0	S/ 0	S/ 0	S/ 3,500	S/ 3,500	S/ 3,500	S/ 5,200	S/ 5,200	S/ 5,200	S/ 5,200	S/ 5,200	S/ 5,200	S/ 41,700
Total	S/ 0	S/ 0	S/ 0	S/ 3,500	S/ 3,500	S/ 3,500	S/ 5,200	S/ 5,200	S/ 5,200	S/ 5,200	S/ 5,200	S/ 5,200	S/ 41,700

En la Tabla 3.2 se puede observar que los primeros 3 meses no hay ingresos, del cuarto al sexto mes se estima solo realizar el cobro del sistema en sí, para luego del séptimo mes en adelante iniciar con desarrollos nuevos, sumando 1700 soles. El 100% de cobros es al contado.

A continuación, el detalle de las ventas en los 5 primeros años.

En la Tabla 3.3, para el segundo año se estima mantener 12 mensualidades con el aeropuerto de Lima, y 6 con el aeropuerto de Ecuador, para el tercer año 12 mensualidades con el aeropuerto de Lima y 12 con Ecuador (montos en soles)

Tabla 3.3

Ventas de los 5 primeros años

#	Ventas Anuales en Soles	Año 1	Año 2	Año 3	Año 4	Año 5
1	Sistema de Coordinación	31,500	63,000	84,000	105,000	126,000
2	Desarrollo de mejoras	10,200	20,400	40,800	40,800	40,800
3	Suscripción aeropuertos Internacionales	0	21,000	42,000	63,000	84,000
	Total Venta en Soles	41,700	104,400	166,800	208,800	250,800

En la Tabla 3.4, hemos detallado todos los costos del desarrollo del sistema.

Tabla 3.4

Costos asociados al proyecto

#	Descripción	Unidad Medida	Costo Unitario
1	API Vuelos internacionales	Servicio	S/. 120.00
2	Analista Desarrollador	Servicio	S/.1500.00
3	Heroku	Servicio	S/.262.00
4	Google Firebase	Servicio	S/.216.00

En la Tabla 3.5 se evidencia el resultado de los ingresos y gastos:

Tabla 3.5

Resultado de Ingresos y Gastos

	Año 1	Año 2	Año 3	Año 4	Año 5
Ventas	41,700	104,400	166,800	208,800	250,800
Costo de Ventas	9,432	18,864	25,152	31,440	37,728
Utilidad Bruta	32,268	85,536	141,648	177,360	213,072
Gastos de producción Fijos	800	800	800	800	800
Sueldos de Administración	2,400	2,400	2,400	2,400	2,400
Sueldos Comerciales	0	0	0	0	0
Cargas Sociales	0	0	0	0	0
Publicidad	0	0	0	0	0
Fletes	0	0	0	0	0
Impuestos y tasas	0	0	0	0	0

(continúa)

(continuación)					
Varios	14,400	14,400	14,400	14,400	14,400
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
Total Otros Gastos	17,600	17,600	17,600	17,600	17,600
Utilidad Antes de Intereses e	14,668	67,936	124,048	159,760	195,472
Impuestos					
Intereses	0	0	0	0	0
Utilidad Antes de Impuestos	14,668	67,936	124,048	159,760	195,472
Impuesto a las Ganancias	4,400	20,381	37,214	47,928	58,642
Utilidad Después de Impuestos	10,268	47,555	86,834	111,832	136,830

En la Tabla 3.6 se evidencia la tasa interna de retorno después de los 5 años, siendo este valor de 590.9%

Tabla 3.6

Evaluación

Tasa de Ganancia Requerida	30%					
Flujo del Proyecto	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Ingresos por Ventas		41,700	104,400	166,800	208,800	250,800
Egresos Operativos		26,032	40,064	62,333	85,454	102,456
		15,668	64,336	104,467	123,346	148,344
Inversión	-4,000					
Flujo del Proyecto	-4,000	15,668	64,336	104,467	123,346	148,344
					Valor Residual	494,480
Tasa Interna de Retorno	590.9%					
Valor Actual Neto	279,255					

3.2. KPI's de validación para continuidad

Actualmente el aeropuerto cuenta con dos personas que se encargan de las llamadas telefónicas, una encargada de la parte norte y otra de la parte sur del país, se registran aproximadamente un total de 400 llamadas diarias.

Como indicador, se busca reducir el número de llamadas diarias de los controladores de provincia a Lima en un 50% en el primer mes de implementado el sistema.

Para obtener esta reducción se expone que los usuarios de las distintas provincias, pueden visualizar el tráfico aéreo del aeropuerto en el sistema web propuesto, dada esta información, están en la capacidad de crear su solicitud sin necesidad de realizar la llamada para consultar los espacios, seguidamente el usuario de Lima puede editar, o cancelar esta solicitud, el sistema web maneja un servicio web que no requiere que el usuario tanto de lima como de provincia, se encuentren actualizando la página para visualizar si existe un registro por aceptar o si este ya fue aceptado, aparece automáticamente en su pantalla cuando un registro en la base de datos se actualiza.

3.3. Límites de aceptación del Sistema

Durante los tres primeros meses se tendrá como prioridad la implementación y desarrollo del sistema, por lo que no se contará con ingresos de manera momentánea. De igual manera, es importante mencionar que el producto debe encontrarse operativo en un plazo máximo de 3 meses posterior a la implementación.

El primer mes el sistema recibirá los datos de manera manual, es decir el usuario controlador ingresara sus vuelos del día al sistema y cuando corresponda generar la solicitud de hora de despegue/aterizaje lo realizará a partir de los datos anteriormente cargados, como control de los datos ingresados es el usuario de Lima el que se encarga de validar el requerimiento y tiene la capacidad de aceptar o cancelar la solicitud.

El segundo mes, los vuelos de todos los aeropuertos serán cargados al sistema en forma masiva, y cuando corresponda el usuario controlador solo generará la solicitud de hora de despegue/aterizaje.

El tercer mes es de acompañamiento, y de resolver los últimos problemas que podrían generarse como también de mejoras que el usuario pueda solicitar.

Se debe resaltar que durante los tres meses se estarán resolviendo los problemas suscitados y que el usuario pueda aceptar o no el producto.



CAPÍTULO IV: DEFINICIÓN DEL PROYECTO

4.1 Definición del proyecto

El Aeropuerto Internacional cuenta actualmente con una sola pista que se usa tanto para los aterrizajes y despegues, dándose ambos en una sola dirección.

En cuanto a los aterrizajes, parte del procedimiento es que todos los aviones antes de poder aterrizar pasen por un punto en común, llamado punto SIMUR.

La diferencia en tiempo entre los aviones va a depender del tránsito aéreo, pero se maneja un equivalente mínimo de 2 minutos. Para manejar esta diferencia, en algunos casos la torre de control procede en generar vectores o rutas para los aviones con el fin de tener una cola de espera con el mínimo en SIMUR.

Los vuelos de emergencia, de las Fuerzas Armadas y los vuelos humanitarios manejan una constante prioridad para el aterrizaje, por lo que en muchos casos no solicitan hora de despegue y estando en vuelo, solicitan hora de aterrizaje preferencial. En el sistema, estos vuelos solo se manejan de forma documental, ya que no requieren de la coordinación entre aeropuertos. Por otro lado, los vuelos internacionales no requieren solicitar hora de despegue desde su origen, estos manejan una prioridad mediana y sus horas de aterrizaje se estiman o confirman una vez hayan ingresado al espacio aéreo peruano con previo aviso del coordinador del aeropuerto del país vecino por donde cruza el vuelo.

La solicitud de hora de despegue y aterrizaje de un vuelo nacional se realiza siempre y cuando el avión ya se encuentre en tierra de la ciudad de origen. Aun así, la solicitud de despegue no puede ser mayor a 45 minutos desde el momento en que se da la misma. Es importante tener en cuenta que, a un vuelo, una vez en tierra, le toma cerca de 40 minutos estar listo para su nuevo despegue.

En el caso de los vuelos nacionales que solicitan hora de despegue y encuentran alto tránsito hasta en las próximas dos horas, la solicitud si puede ingresar y debería ser aceptada.

De igual manera, en el caso de que un avión ya se encuentre en vuelo en dirección al Aeropuerto Internacional con hora de despegue/aterrizaje aceptado y coincida con un

vuelo de emergencia que se encuentra con la misma hora de aterrizaje, el encargado de la torre de control puede colocarlo en cola (en el siguiente *Slot*) o ponerlo al final, mediante vectores de ruta, según sea su criterio.

La propuesta planteada para el área de *flow control* como principal beneficiado en el presente proyecto del Aeropuerto Internacional, consiste en reemplazar el proceso de coordinación de horas de despegue y aterrizaje de las aeronaves procedentes del interior del país mediante llamadas telefónicas, por un sistema web en el cual se podrá visualizar el tráfico aéreo y tener conocimiento de los espacios disponibles para solicitar las horas de despegue y aterrizaje.

Este sistema se estará alimentando constantemente por las solicitudes generadas por los operarios del interior del país para los vuelos nacionales y también por una API Web que maneja los datos de los vuelos internacionales.

4.2 Objetivos del Proyecto

4.2.1 Objetivo general

El proyecto tiene como principal objetivo la creación de una aplicación Web que permita mostrar el tráfico aéreo del Aeropuerto Internacional de esta manera los controladores manejen los datos y que se generen largas colas al momento de solicitar una hora de despegue y aterrizaje logrando ser eficientes en los tiempos de las coordinaciones tanto para solicitar y aceptar una.

4.2.2 Objetivos específicos

- 1 Documentar e informar sobre los vuelos internacionales al controlador de Lima, el sistema se alimentará de la información de los vuelos internacionales desde una API web que maneja distintos datos y entre ellos, la estimada hora de llegada del vuelo al Aeropuerto Internacional, de esta manera saber la ubicación de la aeronave en todo momento.
- 2 Permitir ingresar, aceptar solicitudes de hora de aterrizaje y despegue y ser visualizadas por todos los operarios que deseen solicitar un espacio u horario de aterrizaje de esta manera conocer la disponibilidad de la pista de aterrizaje.

4.3 Fundamento de la factibilidad del proyecto

Se propone el uso de Tecnología *Cloud Computing* usando el modelo PaaS (Plataforma como Servicio), ya que la aplicación se encuentra en un servidor web de la empresa Heroku, así como la base de datos se encuentra en un servidor de *Google Cloud* llamado *Firestore*.

Para este proyecto se está utilizando tecnología web con lenguaje de programación *JavaScript*, *HTML* con librerías de *ReactJS* para el *Frontend*.

Por otro lado, el *Backend* utiliza *restful* propios de la librería de Google con base de datos no relacional, conocida también como colecciones que utilizan el formato JSON.

4.4 Roles y responsabilidades del equipo del proyecto:

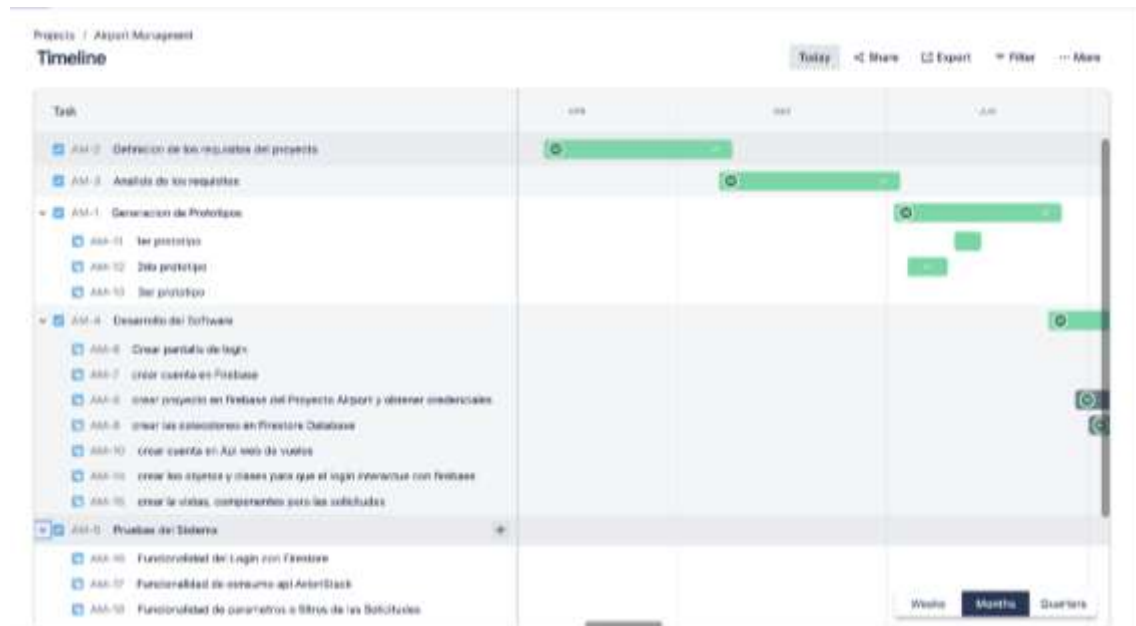
Se encontrarán dos roles, el administrador del sistema es el encargado de liderar el proyecto en sus distintas fases; así como también, es el responsable del diseño, desarrollo, implementación y soporte del sistema. Por otro lado, el usuario del sistema será el encargado de aprobar los diseños y funcionalidades básicas y reglas del sistema, de las pruebas de calidad y de aprobar la puesta en producción.

4.5 Cronograma y riesgos iniciales del proyecto

Para realizar el cronograma del proyecto se utilizó el programa de gestión de proyectos *Jira Software*, el mismo que maneja la herramienta de línea de tiempo que se inicia con la “Definición de los requisitos del proyecto” en el mes de abril y culmina con la actividad “Pruebas del Sistema” en el mes de agosto.

Figura 4.1

Cronograma de Actividades



La Figura 4.1 muestra las tareas generales desde el inicio del proyecto hasta finalizar con la fase de prueba, se puede observar las actividades realizadas en cada actividad principal.

En cuanto a los riesgos empezamos haciendo la valorización, los valores asignados van en el rango de 1 a 5, donde 1 representa insignificante, 2 baja, 3 media, 4 moderada y 5 alta

Tabla 4.1

Valorización de Riesgos

		FRECUENCIA		
		BAJO	MEDIO	ALTO
IMPACTO	ALTO	4	5	5
	MEDIO	3	3	5
	BAJO	1	2	4

Hacemos una escala de valoración de efectividad de controles con el siguiente rango:

Tabla 4.2*Rango de Efectividad*

Control	Efectividad
Ninguno	1
Bajo	2
medio	3
Alto	4
Destacado	5

Finalmente se calcula el riesgo neto

Tabla 4.3*Controles y Riesgos*

Nivel Riesgo	Riesgos	Control	Efectividad	Promedio Efectividad	Riesgo Neto
3	Api web deje de funcionar	Operador se comunica con aeropuertos de países vecinos de ruta de la aeronave	2	3	1
		Solicitar a Sistemas visualizar en otras Apis mediante herramientas	3		
		Abrir otras aplicaciones web y buscar por vuelo	4		
4	Servidores dejan de funcionar	Pasamos al proceso de llamadas telefónicas	3	3	1.333
2	Usuarios de provincia ingresen misma hora de solicitud	Operador de Lima cancela solicitud	4	4.5	0.444
		Operador de Lima edita y encola las solicitudes	5		
1	Usuario de Lima acepte solicitudes de la misma hora	solicitudes figuran en color rojo	5	4.5	0.222
		y operario se encarga de trazar ruta o vector (procedimiento que se maneja normalmente)	4		
Perfil de Riesgo:					0.75

Para este cálculo se asignó valores de acuerdo a las tablas presentadas, tanto para el nivel de riesgo como para la efectividad, con estos parámetros se calcula el promedio de efectividad que es la suma de los valores de efectividad en el riesgo planteado entre el número de controles, el Riesgo Neto es el resultado de la división entre nivel de riesgo / Promedio de efectividad calculado.

El perfil de riesgo es el Promedio de los riesgos netos y se considera un mismo peso de ponderación a cada uno de ellos.

Tal cual como se dieron los resultados, y por lo que respecta al análisis de las variables analizadas y estudiadas en la presente investigación, se señala que el perfil del riesgo es muy bajo (0.75) en la escala de riesgo de 1 a 5, es decir de menor a mayor riesgo.

4.6 Recursos necesarios

Para el presente proyecto, que maneja como principal funcionalidad el ingreso de solicitudes de hora de despegue, se necesitará que el usuario pueda visualizar tanto los vuelos nacionales como los internacionales. Para el caso de los vuelos internacionales, se requiere la Api de vuelos internacionales, ya que esa información no es ingresada por los controladores nacionales. De igual manera, se requerirá de un analista desarrollador para resolver los problemas que puedan presentarse, *Heroku* será el servidor que almacene la parte lógica del *backend* del sistema, *Firebase* será el repositorio de datos.

CAPÍTULO V: DESARROLLO DEL PROYECTO

5.1 Desarrollo del prototipo

Para el presente proyecto, se han manejado 3 prototipos antes del inicio del desarrollo del sistema propuesto, la cual se detalla en la Tabla 5.1

Tabla 5.1

Lista de Prototipos

N	Alcance	Beneficio	Objetivo	Aceptado
1	Desde la interfaz de ingreso al sistema con dos usuarios de distintos aeropuertos hasta la aceptación y cancelación de solicitudes de hora de aterrizaje	La interfaz se divide en bloques de colores, y cada uno de ellos de 5 espacios o slots para ingresar solicitudes Los colores para diferenciar la disponibilidad, rojo es ocupado, amarillo es casi ocupado, y verde casi libre	Conocer el Tráfico. El usuario pueda identificar mediante los colores que bloque está ocupado y no estar consultándolo detalladamente para ingresar una solicitud	No
2	Se agrega una barra de filtros con parámetros de fecha, hora inicial, hora final, y tamaño de bloque (tiempo horas)	Se muestra barra de filtros con fechas, hora y tamaño de bloque. La lista de Solicitudes se almacena en bloques dinámicos. El botón de solicitar se visualiza en el acordeón o bloque, solo se mostrará este botón si hay slots disponibles.	El usuario pueda tener control sobre los filtros de las solicitudes. Conocer Historial. Ser eficiente al momento de solicitar una hora de despegue, al solo hacer clic en el bloque, el sistema automáticamente le asignaba un espacio en el bloque.	No
3	Se quitan los bloques de tiempo, y se visualiza una sola pila de vuelos y solicitudes	Una sola lista de solicitudes filtradas por los parámetros de la barra de filtros	Conocer el tráfico de manera ordenada por hora de aterrizaje. Conocer Historial	Si

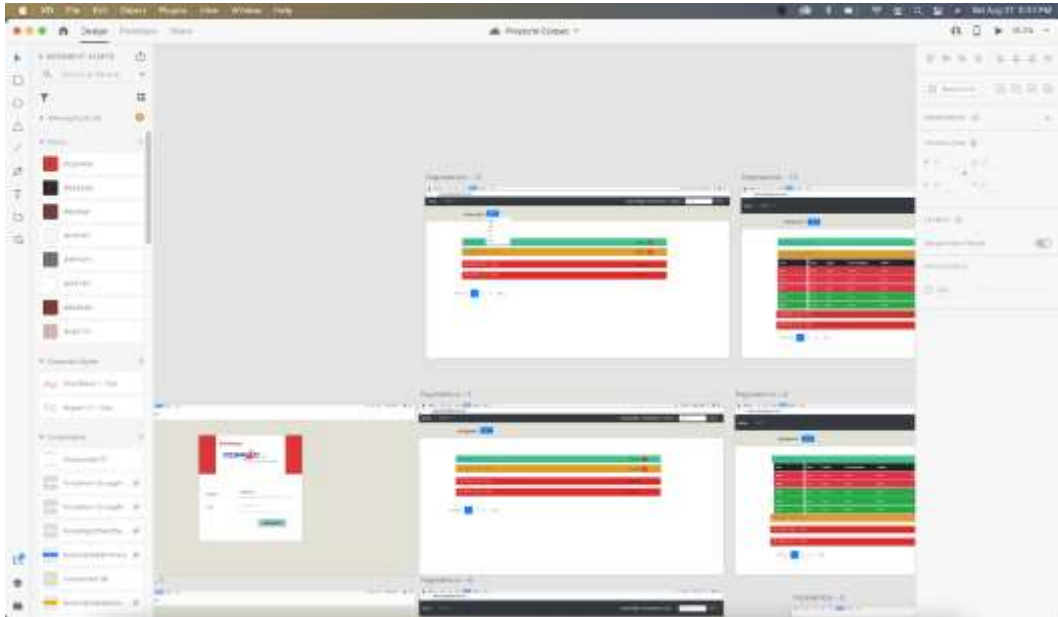
Del análisis realizado en conjunto con el cliente, se decidió continuar con el tercer prototipo, para ello se utilizó la herramienta Jira para el control y la documentación de las distintas actividades.

5.2 Mapa del Prototipo a alto nivel

Para todos los prototipos trabajados en este sistema se utilizó la herramienta Adobe XD

Figura 5.1

Adobe XD como herramienta.

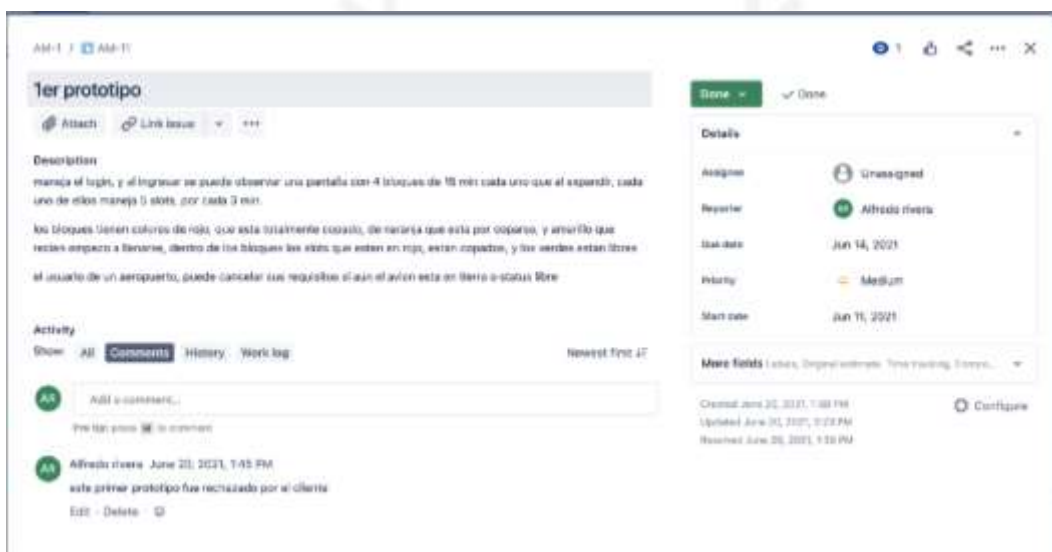


La Figura 5.1 muestra en un alto nivel el primer prototipo utilizando la herramienta Adobe XD.

5.2.1 Prototipo 1

Figura 5.2

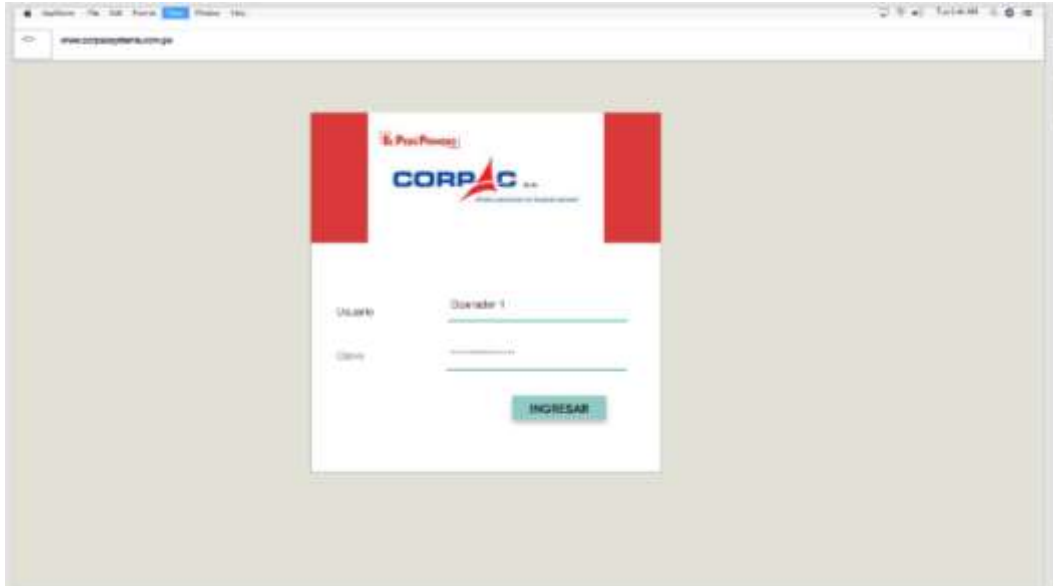
Prototipo 1



La Figura 5.1 muestra la descripción de la solicitud inicial del usuario. Esta imagen se obtuvo desde *Jira Software*.

Figura 5.3

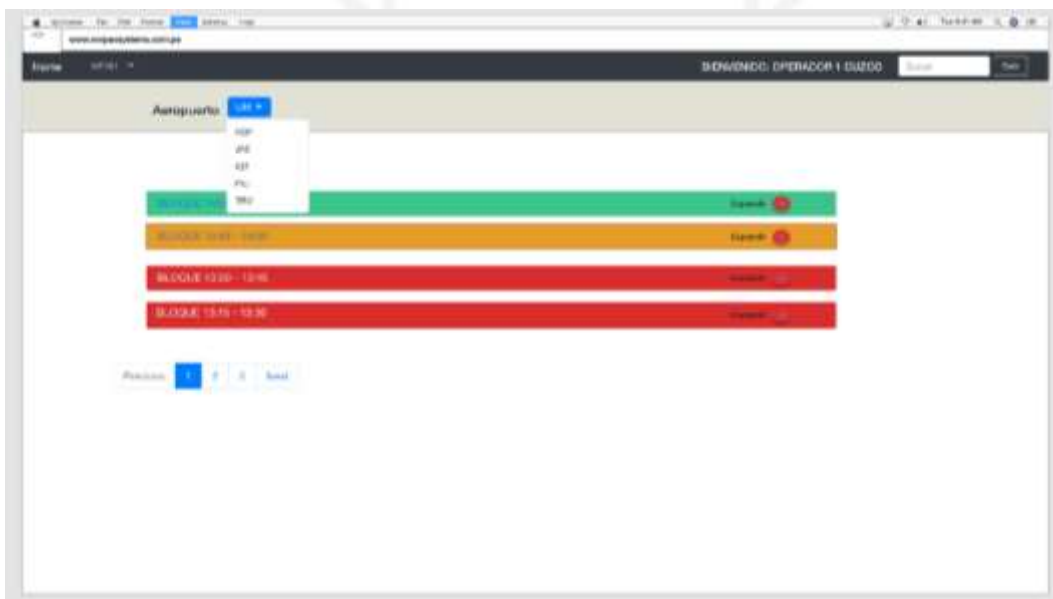
Pantalla login



En la Figura 5.3 se observa el ingreso a la aplicación web a través de una pantalla donde el usuario insertará sus credenciales.

Figura 5.4

Pantalla de solicitudes Pro1



La Figura 5.4, evidencia que una vez validada las credenciales, aparece una pantalla con cuatro barras o bloques, cada uno de ellos representa quince minutos en un espacio de tiempo.

Figura 5.5

Pantalla de solicitudes expandida bloque naranja Pro1

BLOQUE 13:45 - 14:15 Expandir					
BLOQUE 13:45 - 14:00 Completar					
Hora	Vuelo	Origen	Hora Despegue	Status	Acciones
13:57	K124	Cusco	13:00	Body	Cancelar
13:54	T124	Tarma	12:30	Body	
13:51	Body	Body	Body	Body	
13:48	Body	Body	Body	Body	Solicitar
13:45	U148	Body	Body	Body	Solicitar
BLOQUE 13:30 - 13:45 Expandir					

La Figura 5.5 muestra que al dar clic en la opción “expandir”, el usuario podrá visualizar los espacios ocupados y los que están libres, teniendo la oportunidad de poder solicitar este espacio.

Figura 5.6

Pantalla de solicitudes expandida bloque verde Pro1

Aeropuerto LM1					
BLOQUE 14:00 - 14:15 Completar					
Hora	Vuelo	Origen	Hora Despegue	Status	Acciones
14:00	A124	Cusco	13:00	Body	Cancelar
14:03	T124	Tarma	12:30	Body	
14:06	Body	Body	Body	Body	Solicitar
14:09	Body	Body	Body	Body	Solicitar
14:12	U148	Body	Body	Body	Solicitar
BLOQUE 13:45 - 14:00 Completar					

La Figura 5.6 muestra el color de las barras que representa la cantidad de espacios libres que se tiene en ese horario. El color rojo señala que ya no hay disponibilidad de espacios para una solicitud, mientras que el naranja señala que está casi lleno, y el verde que está casi o totalmente vacío.

Figura 5.7

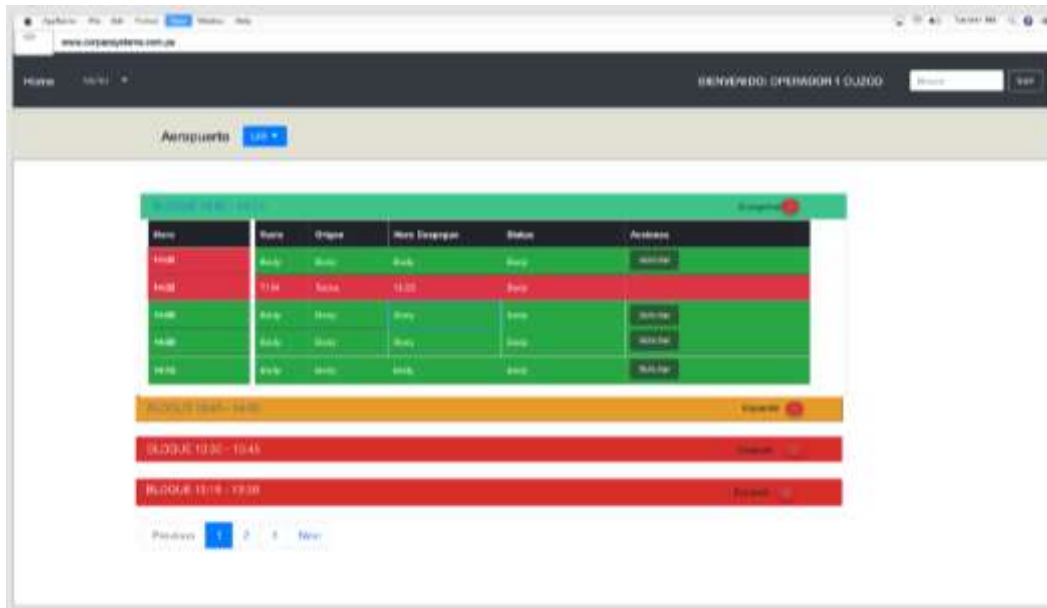
Pantalla solicitudes aviso Pro1



La Figura 5.7, muestra que el usuario puede hacer una solicitud o cancelar la hora de aterrizaje a la ciudad de Lima dándole clic al botón de su requerimiento dentro de la línea de la tabla, se mostrará un mensaje de confirmación.

Figura 5.8

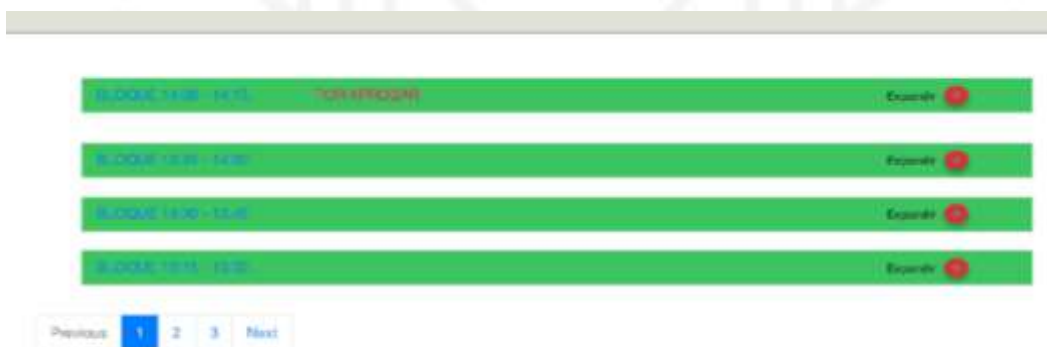
Pantalla solicitudes cancelado Pro1.



La Figura 5.8 muestra que inmediatamente después de cancelar la solicitud esta se torna en color verde como disponible.

Figura 5.9

Pantalla solicitudes usuario Lima Pro1



La Figura 5.9 muestra que, si un usuario de provincia ha ingresado una solicitud, el usuario de Lima, podrá visualizar que existe una por aprobar desde su plataforma.

Figura 5.10

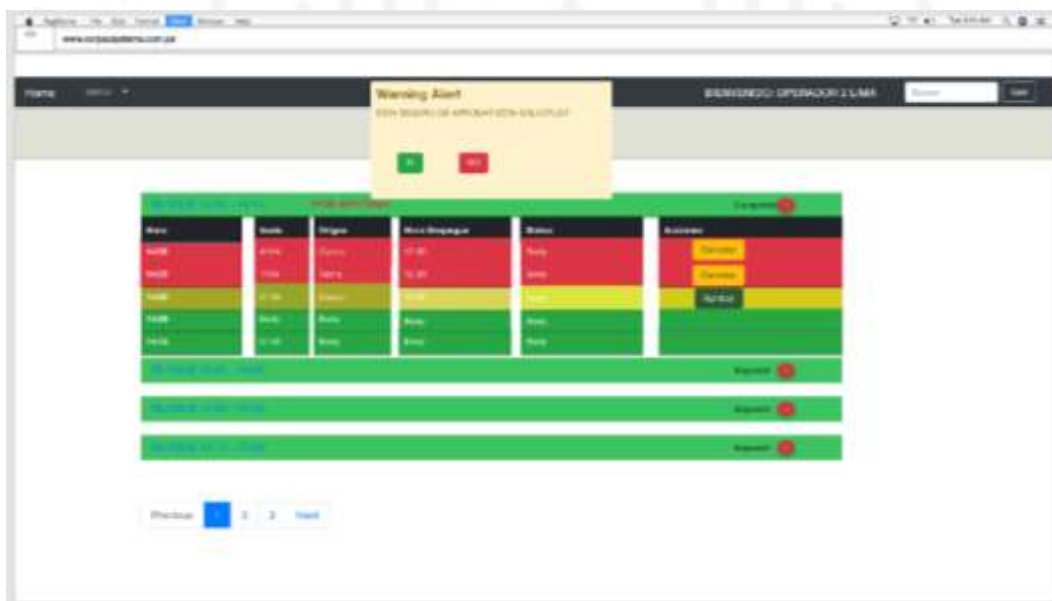
Pantalla solicitudes expandida Lima Pro1

Hora	Vuelo	Origen	Hora Despegue	Status	Acciones
14:00	A124	Caracas	15:00	Enviado	Cancelar
14:00	T134	Toronto	15:00	Enviado	Cancelar
14:00	C134	Caracas	15:00	Enviado	Aprobar
14:00	Enviado	Enviado	Enviado	Enviado	
14:00	Enviado	Enviado	Enviado	Enviado	

La Figura 5.10 muestra que al dar clic en “Expandir”, se podrá visualizar el estado de la solicitud, y podrá aceptarla o cancelarla.

Figura 5.11

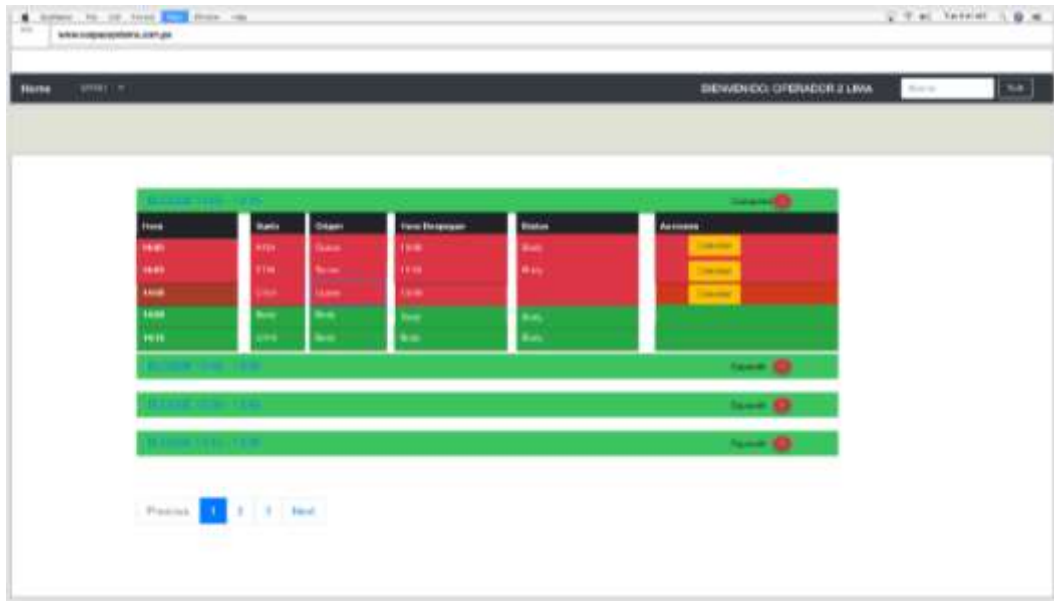
Pantalla solicitudes Lima alerta Pro1.



La Figura 5.11 muestra que al dar clic en el botón “Aprobar”, aparecerá un mensaje de validación.

Figura 5.12

Pantalla solicitudes Lima aceptado Pro1.

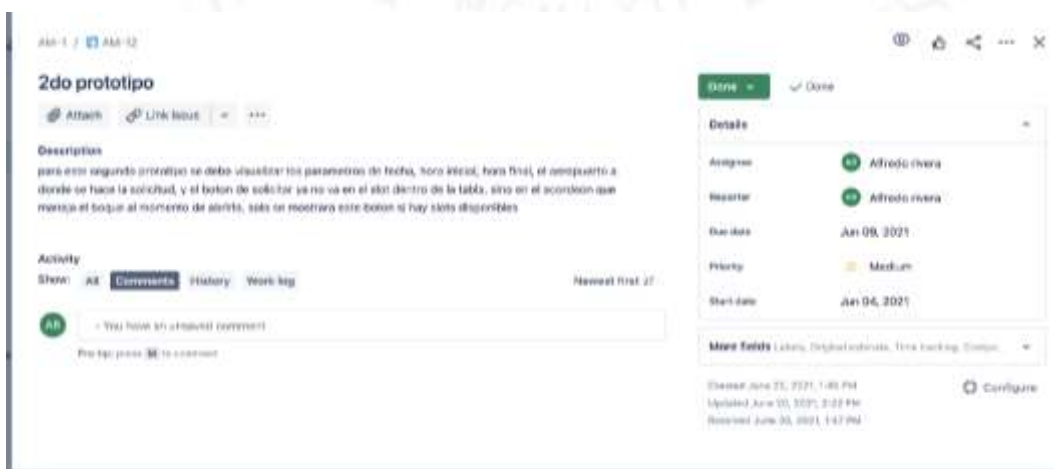


La Figura 5.12, muestra que luego de aprobada la solicitud esta cambia de estado y el color pasa de ser amarillo a rojo en señal de ocupado.

5.2.2 Prototipo 2

Figura 5.13

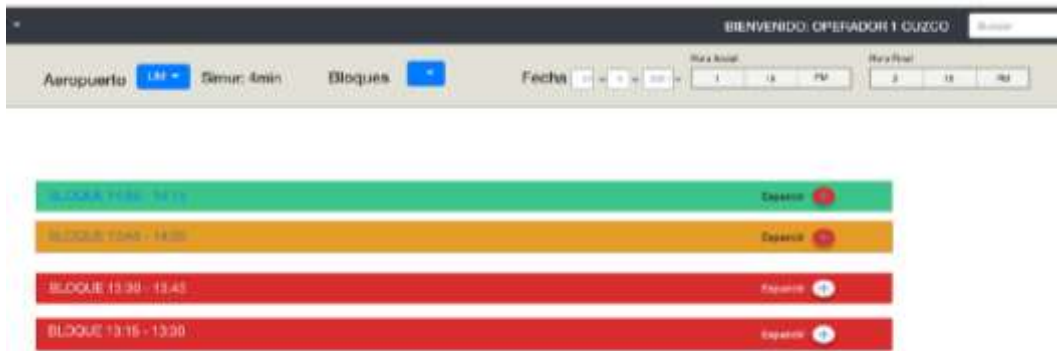
Prototipo 2



En la Figura 5.13 se evidencian las modificaciones y requerimientos para el Prototipo 2.

Figura 5.14

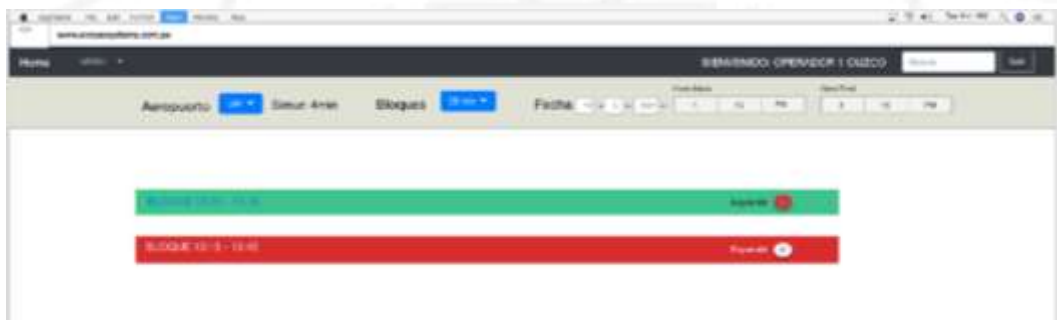
Pantalla solicitudes Pro2



En la Figura 5.14, muestra que tal como indica la solicitud; ahora la barra maneja varios parámetros para el filtro de los vuelos, siendo los bloques inicialmente de 15 minutos

Figura 5.15

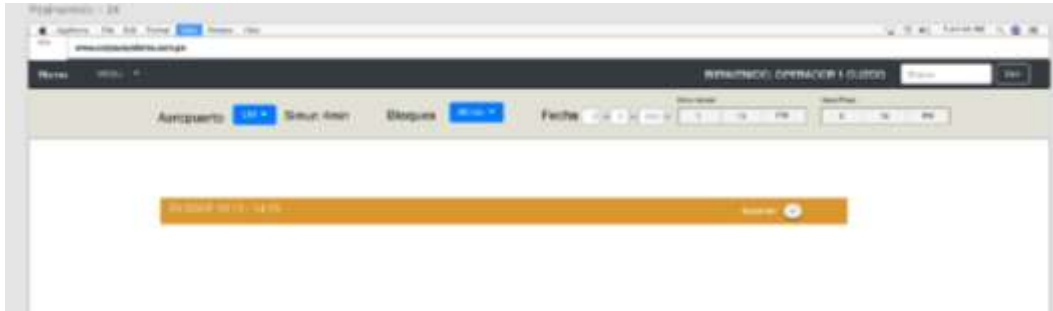
Pantalla solicitudes bloque2 Pro2



En la Figura 5.15 se muestra un cambio a bloque de 30 minutos.

Figura 5.16

Pantalla solicitudes bloque1 Pro2

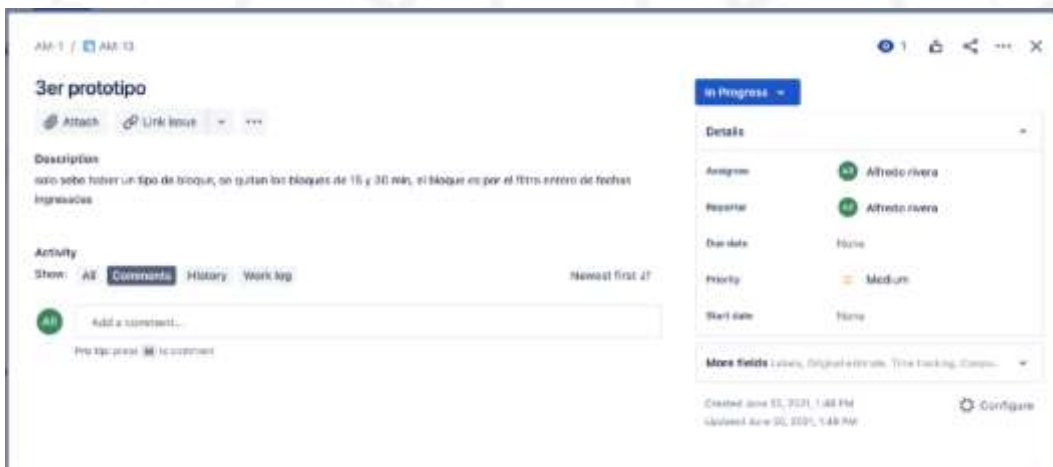


En la Figura 5.16, se muestra un cambio a bloque de 60 minutos.

5.2.3 Prototipo 3

Figura 5.17

Prototipo 3



La Figura 5.17 muestra que para este tercer prototipo se dejaron de lado los bloques y por ende los colores que lo diferenciaban según la disponibilidad de espacios o solicitudes dentro de ellas.

Figura 5.18

Pantalla Login Pro3

LOGIN

Usuario: Operador 1

Clave: *****

Ingresar

La Figura 5.18 muestra en la pantalla el *Login* donde el usuario deberá ingresar sus credenciales.

Figura 5.19

Pantalla solicitudes Pro3.

Programación - 28

Inicio Solicitudes Agregar Solicitudes BIENVENIDO OPERADOR 3 (LUCE)

Nombre: LU Fecha: 11/28/2011 Estado: 1 Hora: 11:00

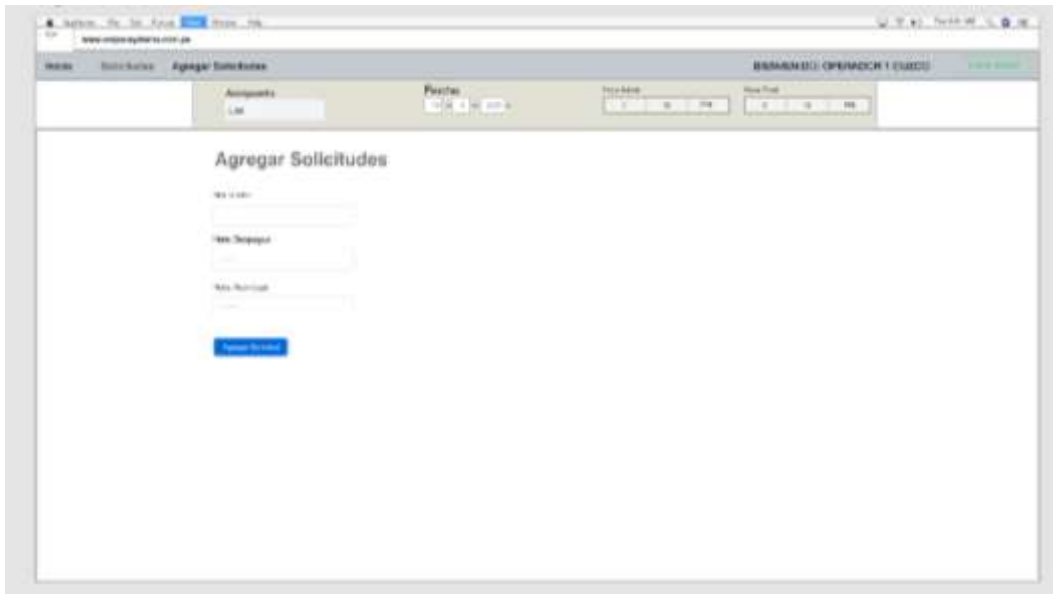
Solicitudes

#	Hora Despegue	Hora Aterrizaje	Vuelo	Origen	Destino	Acciones
1	11:00	0:00	LA127	CLJ	TORAJROBAT	Ver Eliminar
2	11:00	0:00	LA127	MBIA	INTERNACIONAL	Ver
3	11:00	0:00	LA127		INTERNACIONAL	Ver
4	11:00	0:00	LA127	CLJ	AMORDO	Ver
5	11:00	0:00	LA127		INTERNACIONAL	Ver
6	11:00	0:00	LA127		INTERNACIONAL	Ver
7	11:00	0:00	LA127	PUJ	FORAJROBAT	Ver

La Figura 5.19 muestra la pantalla con las solicitudes nacionales y los vuelos internacionales.

Figura 5.20

Pantalla Agregar Solicitudes Pro3.



La figura 5.20 muestra la pantalla de la posibilidad de agregar nuevas solicitudes.

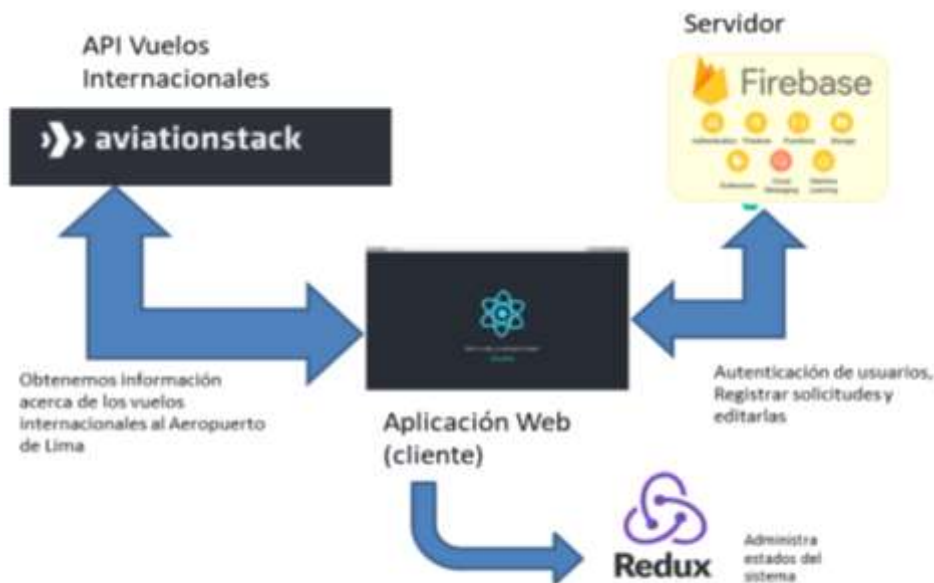
Con la aceptación del Tercer Prototipo, se empezó con el desglose de los distintos componentes que formaban parte de este.

5.3 Fase de Desarrollo del Sistema Web

En esta fase empezamos planteando la arquitectura del sistema.

Figura 5.21

Arquitectura General de la Aplicación



La Figura 5.21 muestra la arquitectura básica de la aplicación.

Es una arquitectura compleja, que incorpora las siguientes características:

- Llamadas a una API para obtener los datos de vuelos internacionales
- Uso de *Firebase* como servidor de base de datos, que nos brinda autenticar a los usuarios y el almacenamiento de los datos.
- Uso de *Redux* para Administrar los estados del sistema

Ahora nos toca detallar la estructura y el funcionamiento de la aplicación mediante un conjunto de diagramas UML.

5.3.1 Diagramas de Casos de Uso

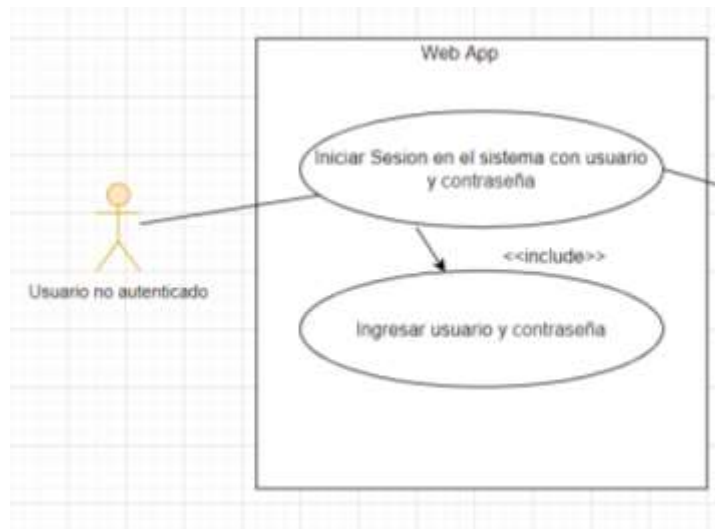
Los diagramas presentados describen las interacciones entre los actores y el sistema

5.3.1.1 Autenticación del usuario

Se muestra a continuación, el proceso de autenticar a un usuario en la aplicación. Con la participación del usuario que intenta autenticarse (véase Figura 5.22).

Figura 5.22

Caso de uso Autenticación de usuario



En la Tabla 5.2 se detallan las operaciones del caso de uso.

Tabla 5.2

CU01 Iniciar Sesión con correo y contraseña

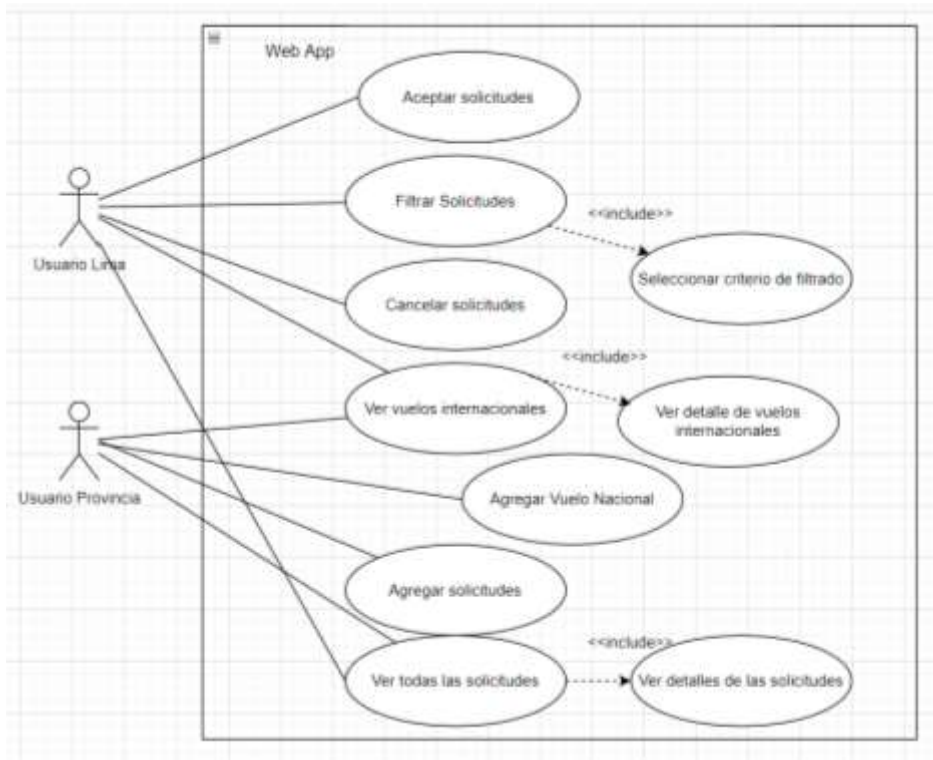
1.- Documento de caso de uso 01		
Elaborado por: Alfredo Rivera		
1.1 Nombre:	Iniciar sesión con correo y contraseña	
1.2 Descripción	Se detalla la acción de la aplicación para autenticar a un usuario que tiene una cuenta registrada en el sistema	
1.3 Flujo Principal	Evento Actor	Evento Sistema
	El usuario introduce su correo electrónico y contraseña	El módulo de autenticación de Firebase verifica correctamente las credenciales introducidas
1.4 Pre-condición	El usuario debe estar registrado en el sistema	
1.5 Post-condición	El usuario inicia sesión de manera exitosa en el sistema.	

5.3.1.2 Funciones en Solicitudes

Se muestra a continuación, las distintas operaciones que puede realizar un usuario que se encuentra dentro de la aplicación (véase Figura 5.23).

Figura 5.23

Funciones en Solicitudes



En las siguientes tablas se detallan las operaciones del caso de uso.

Tabla 5.3

CU02 Ver vuelos internacionales

1.- Documento de caso de uso 02		
Elaborado por: Alfredo Rivera		
1.1 Nombre:	Ver vuelos internacionales	
1.2 Descripción	Se detalla la acción de la aplicación para mostrar los vuelos internacionales	
1.3 Flujo Principal	Evento Actor	Evento Sistema
	El usuario solicita al sistema los vuelos	El sistema lee de API, y muestra los vuelos
1.4 Pre-condición	El usuario debe haber accedido al sistema, La API debe estar activa	
1.5 Post-condición	El usuario visualiza los vuelos internacionales a la ciudad de Lima	

Tabla 5.4*CU03 Ver detalle vuelo internacional*

1.- Documento de caso de uso 03		
Elaborado por: Alfredo Rivera		
1.1 Nombre:	Ver detalle de vuelos internacionales	
1.2 Descripción	Se detalla la acción de la aplicación para mostrar el detalle de un vuelo internacional	
1.3 Flujo Principal	Evento Actor	Evento Sistema
	El usuario solicita al sistema los vuelos, se da el caso de uso ver vuelos internacionales	El sistema lee de API, y muestra los vuelos
	El usuario solicita ver el detalle de uno de los vuelos	El sistema muestra el detalle
1.4 Pre-condición	El usuario debe haber accedido al sistema,, La API debe estar activa	
1.5 Post-condición	El usuario visualiza el detalle de un vuelo internacional	

Tabla 5.5*CU04 Filtrar Solicitudes*

1.- Documento de caso de uso 04		
Elaborado por: Alfredo Rivera		
1.1 Nombre:	Filtrar Solicitudes	
1.2 Descripción	Se detalla la acción de la aplicación para filtrar solicitudes en un lapso de tiempo	
1.3 Flujo Principal	Evento Actor	Evento Sistema
	El usuario selecciona un criterio de filtrado	El sistema lee de la API y de la base de datos y muestra los vuelos filtrados
1.4 Pre-condición	El usuario debe haber accedido al sistema, deben existir solicitudes	
1.5 Post-condición	El usuario visualiza sólo las solicitudes filtradas	

Tabla 5.6*CU05 Agregar Vuelo*

1.- Documento de caso de uso 05		
Elaborado por: Alfredo Rivera		
1.1 Nombre:	Agregar Vuelo	
1.2 Descripción	Se detalla la acción de la aplicación para agregar un vuelo nacional, y se puede visualizar en la lista que sirve para generar una solicitud.	
1.3 Flujo Principal	Evento Actor	Evento Sistema
	El usuario solicita al sistema agregar un vuelo.	El sistema agrega el vuelo como colección en la base de datos
1.4 Pre-condición	El usuario debe haber accedido al sistema	
1.5 Post-condición	El usuario visualiza el formulario de solicitudes	

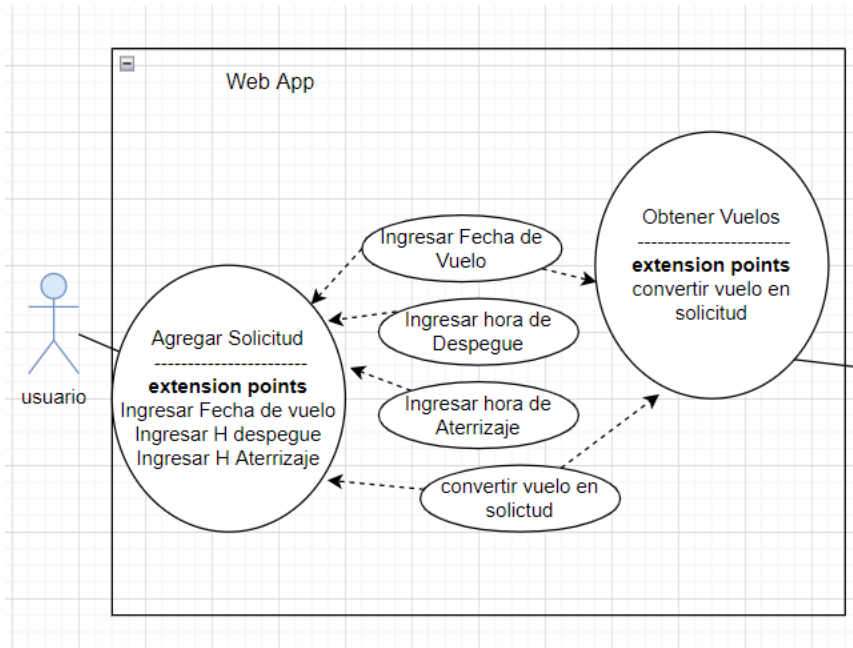
Tabla 5.7*CU06 Cancelar Solicitud*

1.- Documento de caso de uso 06		
Elaborado por: Alfredo Rivera		
1.1 Nombre:	Cancelar Solicitud	
1.2 Descripción	En este caso de uso se detalla el comportamiento del sistema para cancelar solicitudes de vuelo	
1.3 Flujo Principal	Evento Actor	Evento Sistema
	El usuario Lima solicita al sistema cancelar una solicitud	Se muestra la lista de solicitudes y aquellas con el estado "POR APROBAR" y un botón que hace mención a Detalles
	El usuario Lima selecciona Cancelar solicitud, se realiza el caso de uso Cancelar Solicitud	El sistema modifica el estado de solicitud a CANCELADO en la base de datos
1.5 Pre-condición	El usuario debe haber accedido al sistema	
1.6 Post-condición	El usuario visualiza las solicitudes	

Debido a que los casos de uso para Agregar Solicitudes y Aceptar Solicitudes son más complejos, detallamos estos casos de uso.

Figura 5.24

Caso de uso Agregar solicitud



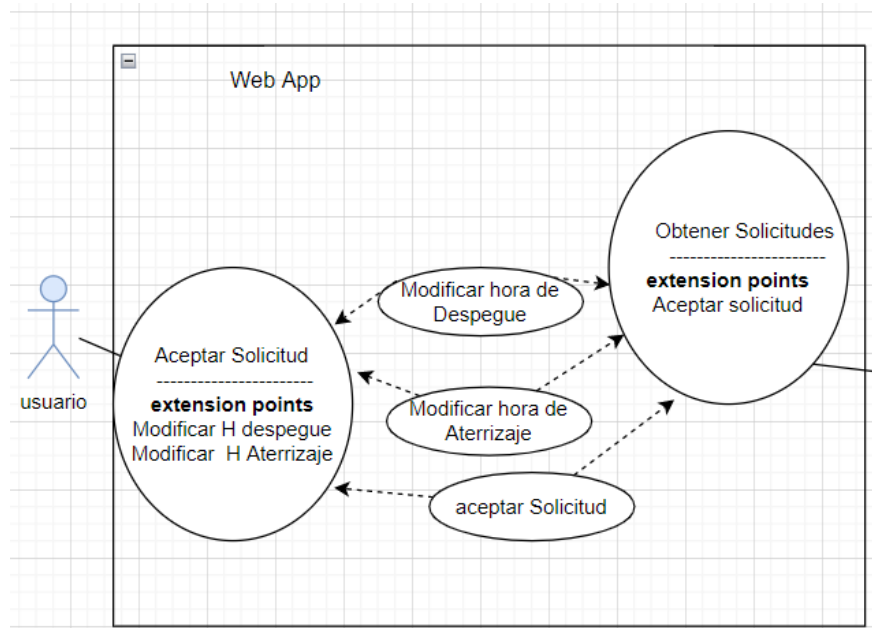
En la Figura 5.24 se detallan las operaciones del caso de uso Agregar Solicitud.

Tabla 5.8*CU07 Agregar Solicitud*

1.- Documento de caso de uso 07		
Elaborado por: Alfredo Rivera		
1.1 Nombre:	Agregar Solicitud	
1.2 Descripción	En este caso de uso se detalla el comportamiento del sistema para agregar solicitudes de vuelo	
1.3 Flujo Principal	Evento Actor	Evento Sistema
	El usuario provincia solicita al sistema agregar un solicitud	Se muestra el formulario a completar
	Se selecciona la fecha del vuelo, se realiza el caso de uso Ingresar Fecha vuelo	El sistema muestra los vuelos de ese día, del aeropuerto registrado del usuario
	Se ingresa Hora de Despegue, se realiza el caso de uso Ingresar Hora de Despegue	El sistema valida que la hora de aterrizaje sea mayor a la de despegue
	Se ingresa hora de Aterrizaje, se realiza el caso de uso Ingresar Hora de Aterrizaje	El sistema valida que la hora de aterrizaje sea mayor a la de despegue
	El usuario provincia selecciona agregar solicitud, se realiza el caso de uso Agregar Solicitud	El sistema agrega la solicitud en la base de datos
1.4 Pre-condición	El usuario debe haber accedido al sistema	
1.5 Post-condición	El usuario visualiza las solicitudes	

Figura 5.25

Aceptar Solicitud



En la Figura 5.25 se detallan las operaciones del caso de uso Aceptar Solicitud.

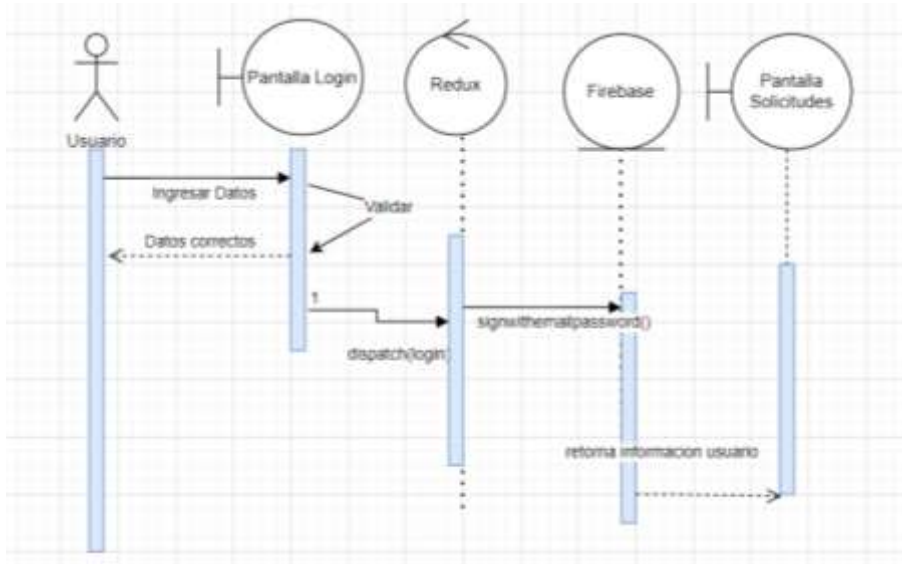
Tabla 5.9*CU08 Aceptar Solicitud*

1.- Documento de caso de uso 08		
Elaborado por: Alfredo Rivera		
1.1 Nombre:	Aceptar Solicitud	
1.2 Descripción	En este caso de uso se detalla el comportamiento del sistema para aceptar solicitudes de vuelo	
1.3 Flujo Principal	Evento Actor	Evento Sistema
	El usuario Lima solicita al sistema aceptar una solicitud	Se muestra la lista de solicitudes y aquellas con el estado "POR APROBAR" y un botón que hace mención a Aprobar
	El usuario Lima selecciona Aprobar solicitud, se realiza el caso de uso Agregar Solicitud	El sistema modifica el estado de solicitud a APROBADO en la base de datos
1.4 Flujo Secundario	El usuario Lima solicita al sistema aceptar una solicitud	Se muestra la lista de solicitudes y aquellas con el estado "POR APROBAR" y un botón que hace mención a Detalles
	Se Modifica Hora de Despegue, se realiza el caso de uso Modificar Hora de Despegue	El sistema valida que la hora de aterrizaje sea mayor a la de despegue
	Se Modifica hora de Aterrizaje, se realiza el caso de uso Modificar Hora de Aterrizaje	El sistema valida que la hora de aterrizaje sea mayor a la de despegue
	El usuario selecciona Aprobar solicitud, se realiza el caso de uso Aprobar Solicitud	El sistema modifica el estado de solicitud a APROBADO en la base de datos
1.5 Pre-condición	El usuario debe haber accedido al sistema	
1.6 Post-condición	El usuario visualiza las solicitudes Editadas	

5.3.2 Diagramas de secuencia

Figura 5.26

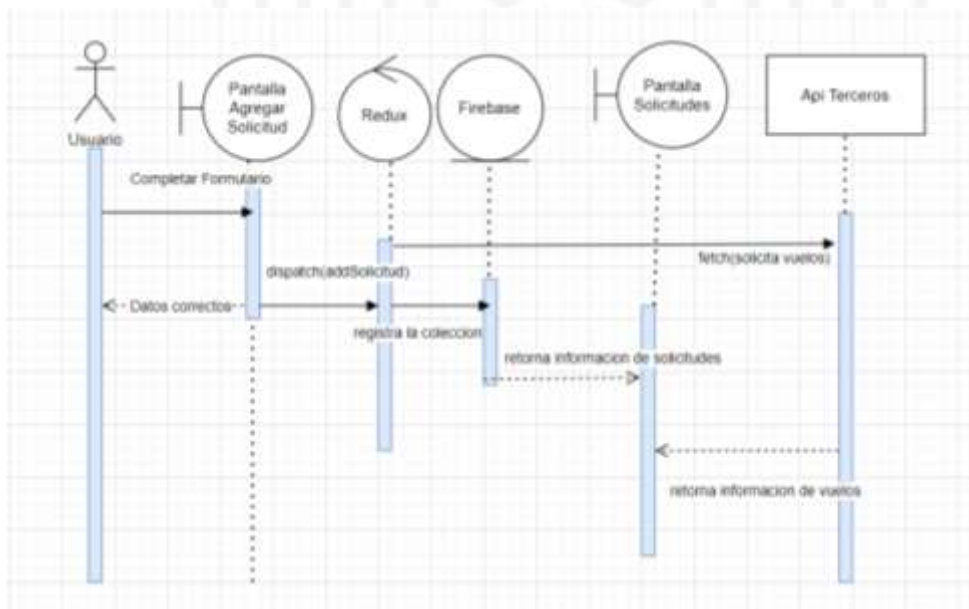
Diagrama de Secuencia Inicio de Sesión



En la Figura 5.26 se muestra el Diagrama de Secuencia de Inicio de Sesión.

Figura 5.27

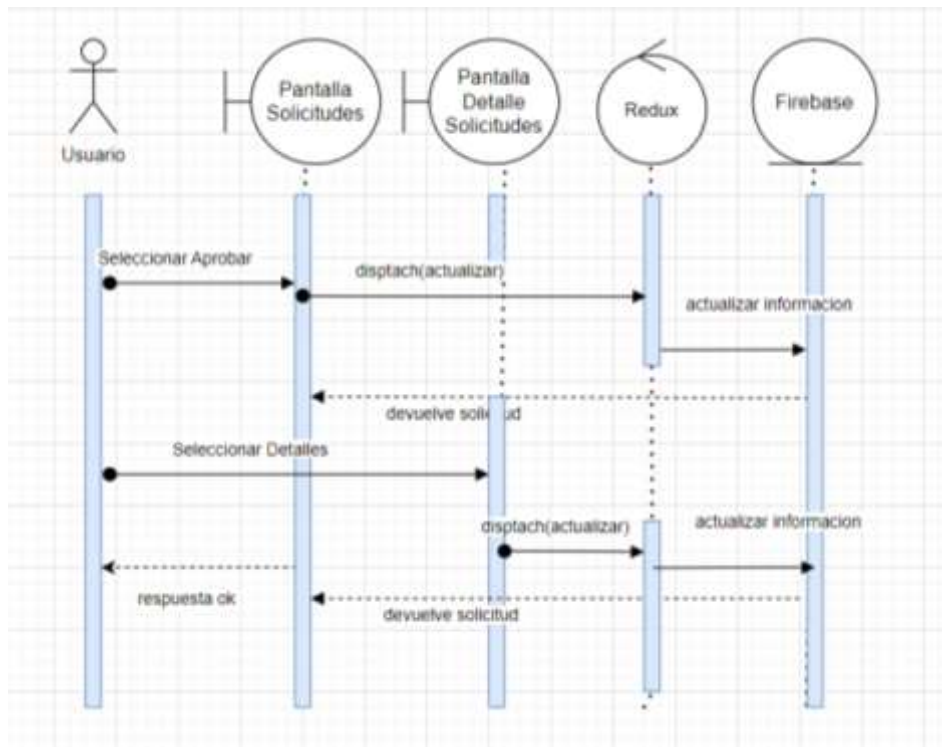
Diagrama de Secuencia Registro de Solicitud



En la Figura 5.27 se muestra el Diagrama de Secuencia Registro de Solicitud.

Figura 5.28

Diagrama de Secuencia Aceptar Solicitud



En la Figura 5.28 se muestra el Diagrama de Secuencia Aceptar Solicitud.

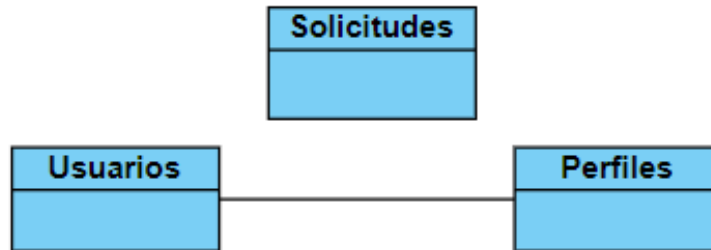
- Modelo Conceptual de Datos

En este proyecto hemos trabajado con 3 colecciones

En la Figura 5.29 se observa las 3 colecciones que se han manejado para este proyecto, siendo Solicitudes la que maneja los datos principales, la colección de usuarios y perfiles tiene una relación de 1 a 1, es decir un usuario solo puede tener un perfil, es necesario saber que los permisos son escalonados.

Figura 5.29

Modelo de Datos



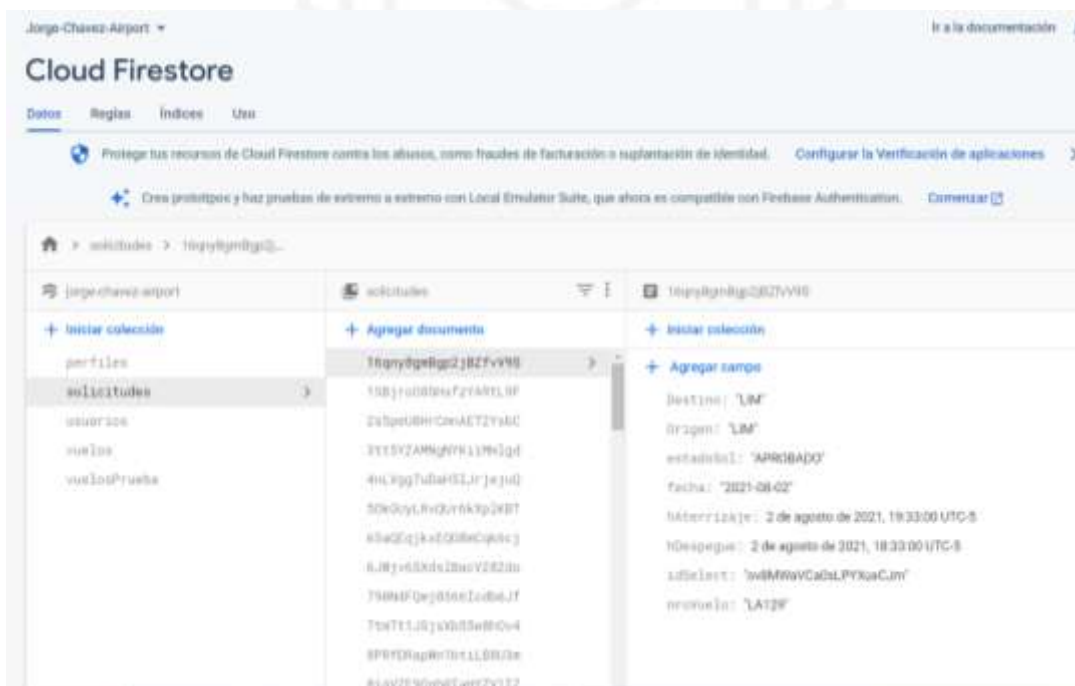
- Diagrama de base de Datos NoSQL

Todos los datos de Firebase se almacenan como objetos JSON. La base de datos puede conceptualizarse como un árbol JSON alojado en la nube. A diferencia de una base de datos de SQL, no hay tablas ni registro

En la Figura 5.30 se observa que todos los datos se almacenan en *Firebase* como objetos JSON, y se expresa como un árbol de JSON.

Figura 5.30

Cloud Firestore



La ejecución del proyecto se llevó a cabo mediante la metodología ágil *Scrum*

Tabla 5.10

Roles Scrum

Rol	Persona a cargo	Descripción
PRODUCT OWNER	Miguel Dionicio	Negocia decisiones sobre el producto con los usuarios.
SCRUM MASTER	Alfredo Rivera	Líder de equipo
SCRUM DEVELOPER	Alfredo Rivera	Diseñar y desarrollar la aplicación.

La Tabla 5.10 muestra los Roles Scrum.

5.3.3 Pila de Producto

La pila de producto fue proporcionada por el dueño del producto, son las distintas historias de usuarios (véase Tabla 5.11) que se van recopilando y se hacen cada vez más extensas.

Tabla 5.11*Historias de Usuario*

HISTORIA DE USUARIO				
ID	COMO...	QUIERO...	PARA...	Criterios de Aceptación
H001	Usuario Lima	iniciar sesión	identificarme como usuario	no pueden existir dos usuarios con el mismo email
H002	Usuario Lima	visualizar los vuelos internacionales	conocer hora de llegada al aeropuerto	solo se muestran vuelos en curso, no aterrizados
H003	Usuario Lima	visualizar los vuelos nacionales	aceptar directamente las solicitudes	el vuelo no debe tener misma hora de aterrizaje que otros vuelos
H004	Usuario Provincia	visualizar el tráfico de Lima	enviar solicitudes y ser aceptadas	los vuelos ordenados por hora de aterrizaje
H005	Usuario Provincia	registrar mis vuelos	presentar posibles solicitudes de hora de despegue	la hora de aterrizaje no puede ser menor a la de despegue
H006	Usuario Lima	filtrar mis vuelos por horas	conocer el tráfico en un espacio de tiempo	la hora fin debe ser mayor a la hora de inicio
H007	Usuario Lima	ver detalle de los vuelos nacionales	aceptar, editar o cancelar los vuelos	el registro puede aceptarse directamente o visualizar el detalle y modificarlo para aceptarlo
H008	Usuario Lima	visualizar vuelos nacionales registrados sin estar refrescando la página	evitar estar consultado cada momento	el usuario no debe refrescar la página, sólo espera si entra una solicitud
H009	Usuario Provincia	visualizar estados de solicitudes registrados sin estar refrescando la página	evitar estar consultado cada momento	el usuario no debe refrescar la página, solo está a la espera y se refresca si se aceptó o no.
H010	Usuario Provincia	iniciar sesión	identificarme como usuario	no pueden existir dos usuarios con el mismo email
H011	Usuario Provincia	filtrar mis vuelos por horas	conocer el tráfico en un espacio de tiempo	la hora fin debe ser mayor a la hora de inicio

5.3.4

5.3.5 Planificación del Sprint

El dueño del producto y el *Scrum developer* llegaron a un acuerdo para iniciar con los desarrollos y en qué tiempo serán terminados, se formaron los siguientes *Sprints*.

Sprint 1

Se estimó un total de 2 semanas para los siguientes ítems de la pila de productos.

Tabla 5.12

Sprint 1

HISTORIA DE USUARIO			
ID	COMO...	QUIERO...	PARA...
H003	Usuario Lima	visualizar los vuelos nacionales	aceptar directamente las solicitudes
H005	Usuario Provincia	registrar mis vuelos	presentar posibles solicitudes de hora de despegue
H007	Usuario Lima	ver detalle de los vuelos nacionales	aceptar, editar o cancelar los vuelos
H010	Usuario Provincia	iniciar sesión	identificarme como usuario
H011	Usuario Provincia	filtrar mis vuelos por horas	conocer el tráfico en un espacio de tiempo

Sprint 2

Se estimó un total de 3 semanas para los siguientes ítems de la pila de productos.

Tabla 5.13

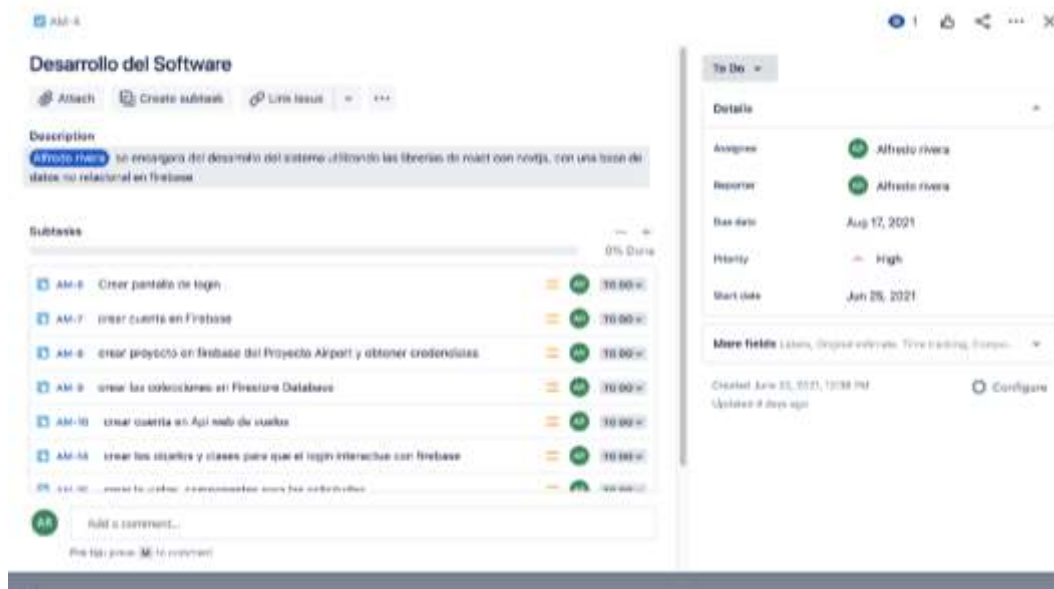
Sprint 2

HISTORIA DE USUARIO			
ID	COMO...	QUIERO...	PARA...
H001	Usuario Lima	iniciar sesión	identificarme como usuario
H002	Usuario Lima	visualizar los vuelos internacionales	conocer hora de llegada al aeropuerto
H004	Usuario Provincia	visualizar el tráfico de Lima	enviar solicitudes y ser aceptadas
H006	Usuario Lima	filtrar mis vuelos por horas	conocer el tráfico en un espacio de tiempo
H008	Usuario Lima	visualizar vuelos nacionales registrados sin estar refrescando la página	evitar estar consultado cada momento
H009	Usuario Provincia	visualizar estados de solicitudes registrados sin estar refrescando la página	evitar estar consultado cada momento

Nuevamente hacemos uso de la herramienta de Jira Software, para el control y seguimiento en esta etapa.

Figura 5.31

Jira Desarrollo del Software.



En la Figura 5.31 se visualiza la lista de tareas por realizar.

Figura 5.32

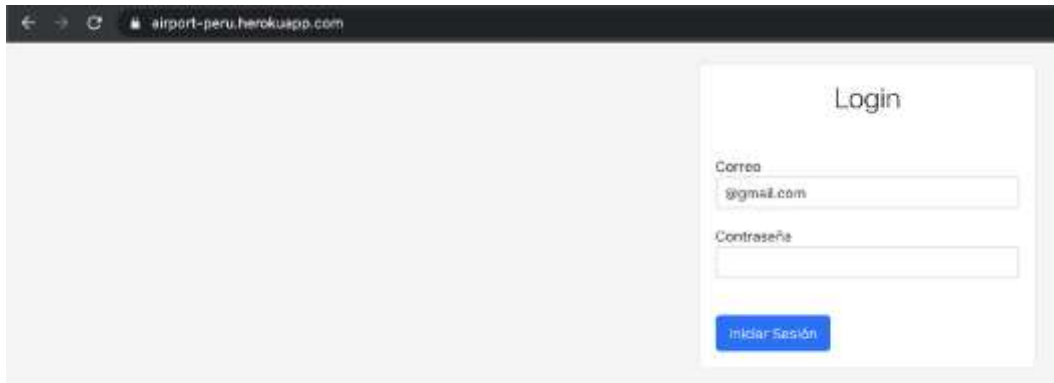
Jira Pantalla Login.



En la Figura 5.32, como primer ejemplo se muestra la primera sub-tarea realizada de esta actividad, crear la pantalla de *Login*.

Figura 5.33

Interface Login.



En la Figura 5.33, el resultado del desarrollo de esta tarea muestra la interfaz de Login.

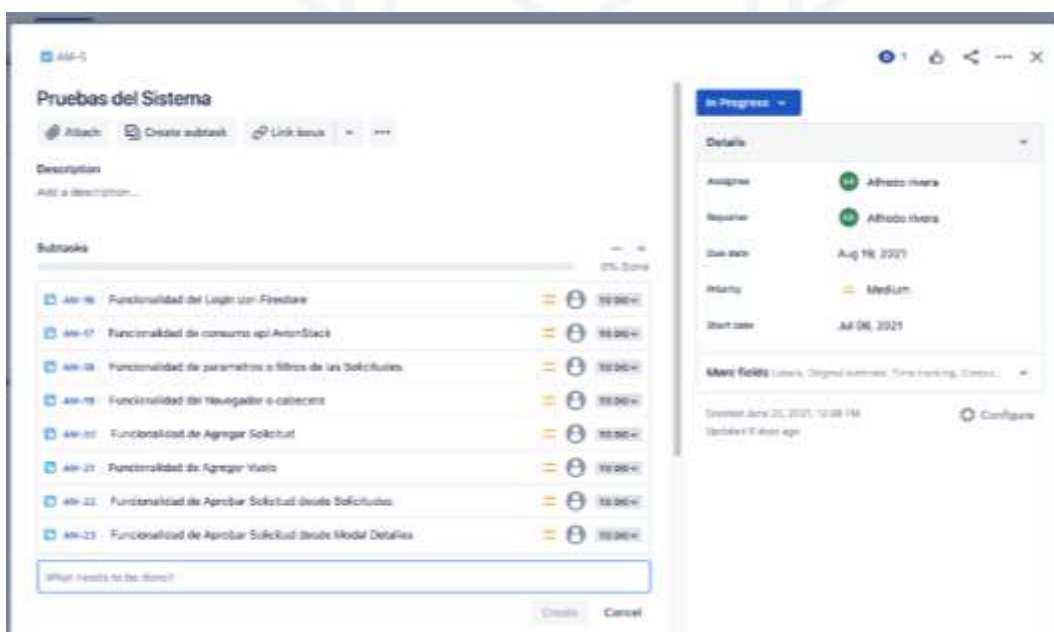
El resto de la documentación para este proyecto se elabora en base al mismo lineamiento (Anexo 1 y Anexo 2).

5.4 Fase de Pruebas

A continuación, se muestra la lista de las distintas pruebas realizadas a las funcionalidades desarrolladas dentro del sistema, usando la herramienta de *Jira Software*.

Figura 5.34

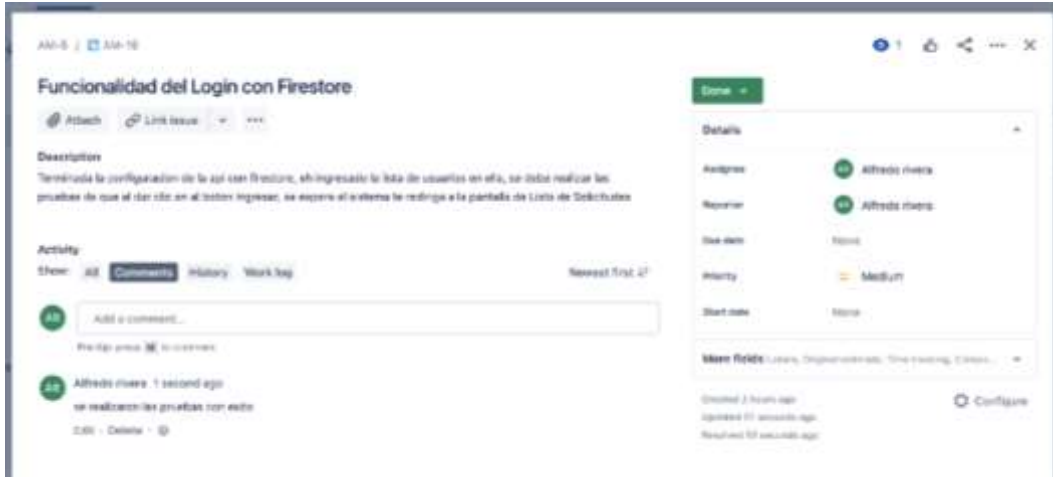
Jira Pruebas del Sistema.



En la Figura 5.34 se visualiza la lista de tareas por realizar.

Figura 5.35

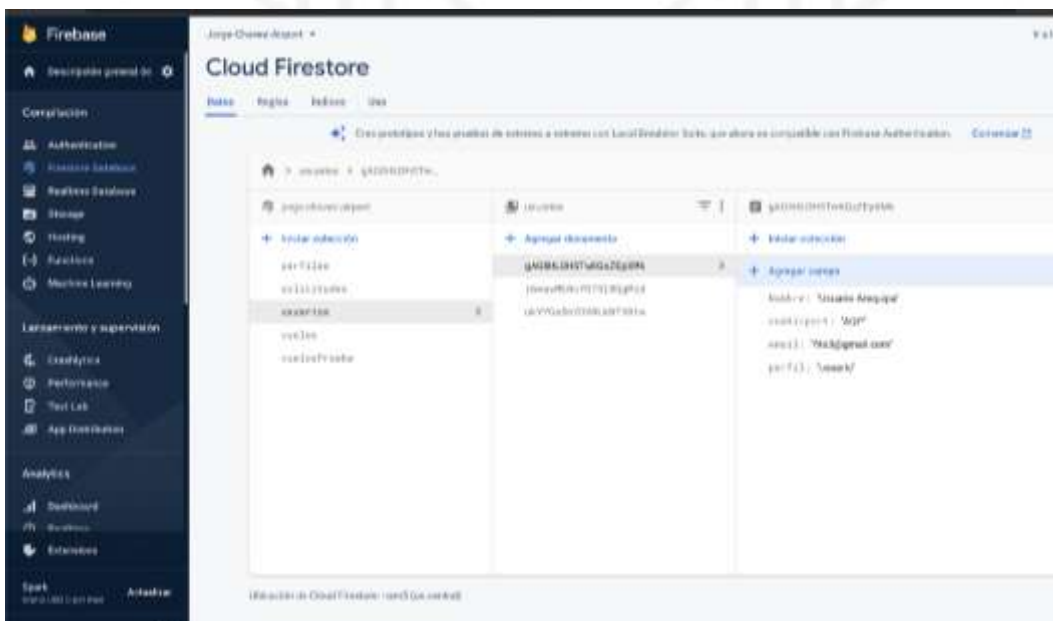
Jira pruebas funcionalidad Login Firestore.



En la Figura 5.35, se muestra como primer ejemplo la primera sub-tarea realizada de esta actividad. AM-16 Funcionalidad del *Login con Firestore*

Figura 5.36

Cloud Firestore usuarios



En la Figura 5.36 se puede observar la consola de *Cloud Firestore*, en ella se hace mención a la colección de usuarios, que maneja un ID proporcionado aleatoriamente por la plataforma y en ella a un usuario, por ejemplo:

Nombre: “Usuario Arequipa”

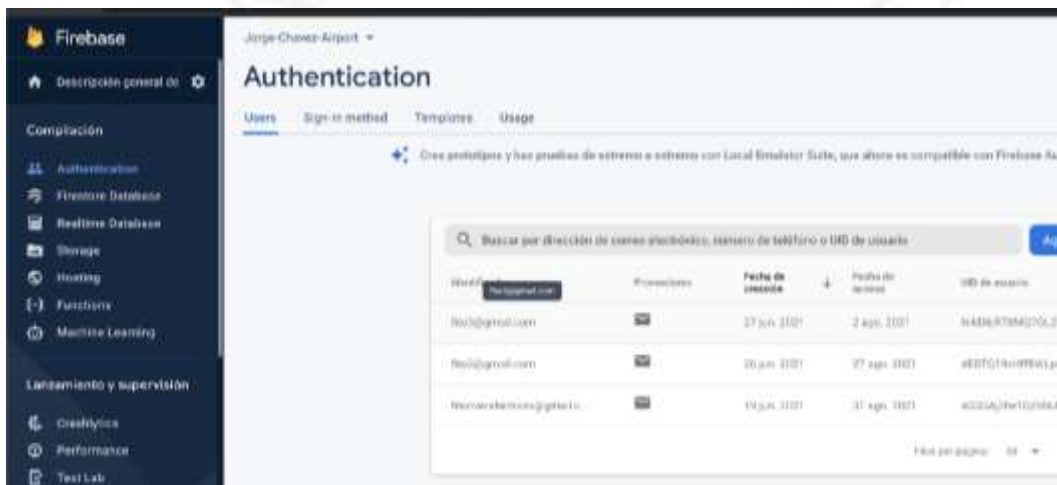
CodAirport: “AQP”

Email: “fito3@gmail.com”

Perfil: “usuario”

Figura 5.37

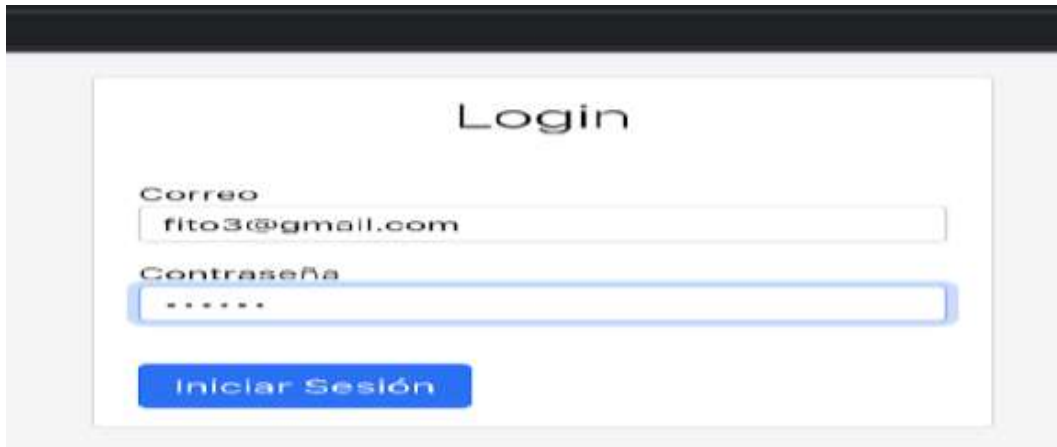
Firestore Authentication.



En la Figura 5.37 se muestra que, si bien esta lista de usuarios, que nos ayuda a salvaguardar cierta información sobre el usuario y que podamos ir ingresando nuevas propiedades o características, el tema de la seguridad se maneja desde el módulo de *Authentication* de Firebase.

Figura 5.38

Credenciales Login



Login

Correo
fito3@gmail.com

Contraseña
.....

Iniciar Sesión

En la Figura 5.38 se muestra las credenciales dentro de la pantalla del *Login*

Figura 5.39

Header Nombre Aeropuerto



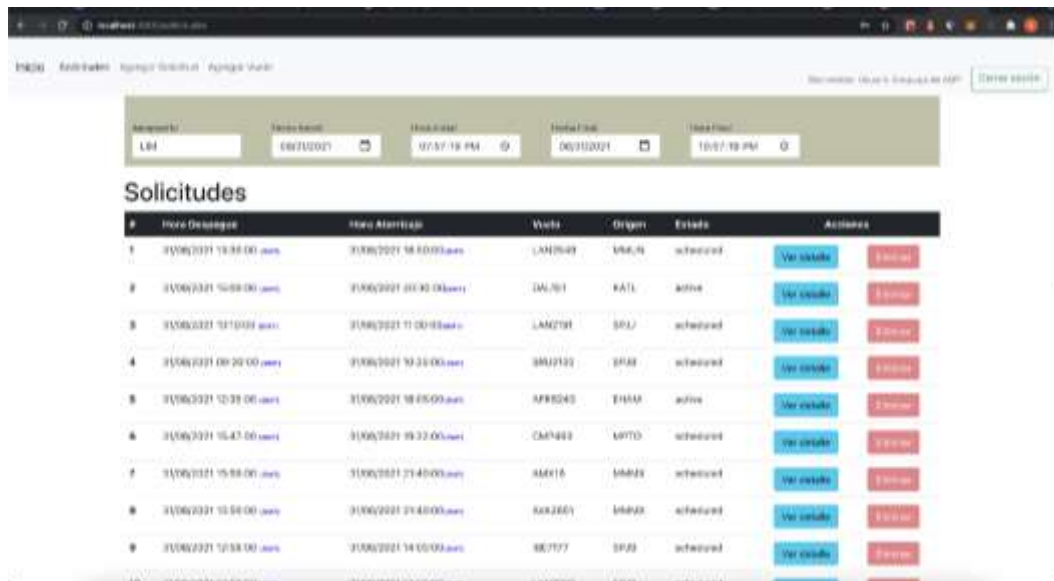
Bienvenido: Usuario Arequipa de AQP Cerrar sesión

Hora Final
10:57:19 PM

En la Figura 5.39 se muestra que al dar clic en el botón “Iniciar Sesión”, aparece la pantalla de solicitudes y en la cabecera o navegador se puede ver el nombre del usuario y código del aeropuerto asignado.

Figura 5.40

Lista de solicitudes



#	Fecha Diseñada	Fecha Asignada	Work	Origin	Estado	Acciones
1	31/06/2021 18:38:00 ver	31/06/2021 18:50:00 ver	LAN763	UMU-R	scheduled	Ver estado Eliminar
2	31/06/2021 19:58:00 ver	31/06/2021 20:43:00 ver	DA761	KATE	active	Ver estado Eliminar
3	31/06/2021 21:09:00 ver	31/06/2021 21:00:00 ver	LAN709	SPJ	scheduled	Ver estado Eliminar
4	31/06/2021 09:20:00 ver	31/06/2021 10:23:00 ver	SM712	SPJ	scheduled	Ver estado Eliminar
5	31/06/2021 12:38:00 ver	31/06/2021 18:45:00 ver	AFR545	ENR	active	Ver estado Eliminar
6	31/06/2021 16:43:00 ver	31/06/2021 18:32:00 ver	CM749	MTO	scheduled	Ver estado Eliminar
7	31/06/2021 15:58:00 ver	31/06/2021 21:45:00 ver	AM715	MAR	scheduled	Ver estado Eliminar
8	31/06/2021 15:58:00 ver	31/06/2021 21:45:00 ver	KK265	MAR	scheduled	Ver estado Eliminar
9	31/06/2021 12:38:00 ver	31/06/2021 14:10:00 ver	SK777	SPJ	scheduled	Ver estado Eliminar
10	31/06/2021 10:54:00 ver	31/06/2021 13:20:00 ver	LA706	SPJ	scheduled	Ver estado Eliminar

La Figura 5.40, se muestra la lista de solicitudes pendientes.

CONCLUSIONES

- Analizados todos y cada uno de los objetivos planteados se expone, en relación al primer objetivo, Mostrar el tráfico aéreo el sistema desarrollado cumple con el requerimiento del cliente, que es poder observar la lista de solicitudes de hora de aterrizaje/despegue de los operarios del interior del país, de esta manera los otros usuarios pueden visualizar el tráfico y en que horario no se debería registrar una solicitud.
- Los operarios pueden observar los distintos vuelos internacionales que están en camino hacia el aeropuerto de Lima.
- En cuanto a los objetivos específicos, los operarios tienen documentadas las solicitudes y pueden acceder a estos en cualquier momento del día, tanto para los actuales como los pasados.
- Los operarios pueden ingresar solicitudes sin necesidad de usar las líneas telefónicas y tener respuestas rápidamente, de esta manera ellos sabrán si sus solicitudes fueron aceptadas o no.

RECOMENDACIONES

- Se recomienda que, para el uso de API'S públicas, es necesario hacer pruebas iniciales con distintos tipos de filtros sin esperar llegar a la funcionalidad requerida dentro del desarrollo del sistema, ya que algunas API'S solo permiten uno o máximo dos filtros a la hora de consumirlas, esto puede ser un problema, ya que podría originar demoras, pues la información no llega correctamente filtrada y eso hace que el sistema deba hacerlo, generando demoras para resolver el problema.
- Se recomienda que, en cuanto a API'S, ya que muchas de ellas envían la información no encriptada es decir solo usando el protocolo HTTP y no HTTPS, puede funcionar adecuadamente de manera local o en fase de desarrollo; sin embargo, cuando ya pasamos a un ambiente productivo, con algún servicio en la nube, puede ocurrir que la seguridad de esta nube no permite respuestas de API'S con solo HTTP; en este caso, se debe optar por el servicio pagado que probablemente incluya HTTPS para las respuestas y solicitudes.
- Se recomienda que las páginas web se deben adecuar al tamaño de la pantalla del usuario.
- Se recomienda que al saber que este software es una herramienta de trabajo y que en caso de emergencia el usuario pueda usarlo desde su celular sin ningún inconveniente, a esto se le llama un diseño responsivo, lo recomendable es usar las librerías de *Bootstrap* para facilitar este diseño.
- En un ambiente productivo es necesario optar por la versión pagada de la API, ya que el servidor web donde se encuentra la aplicación sólo recibe respuestas en el protocolo HTTPS a las solicitudes generadas desde los formularios.
- La carga masiva utilizando la consola de google y su entorno de desarrollo, es beneficioso cuando se trata de proyectos con base de datos *Firebase*.
- El usuario final requiere que las aprobaciones y modificaciones de los vuelos están bajo su mando de manera momentánea y no sean automáticas del sistema.

GLOSARIO DE TÉRMINOS

- API: Application Programming Interface, es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones.
- HTTP: Es el protocolo de transferencia de hipertexto, sirve para comunicarse con un servidor web utilizando la internet.
- HTTPS: es el protocolo seguro de transferencia de hipertexto, es decir es la versión cifrada de HTTP utilizando SSL/TLS
- JAVASCRIPT: Es un lenguaje de programación interpretado y que permite crear contenido de actualización dinámica, orientado a objetos.
- HTML: Lenguaje de marcas o etiquetas de hipertexto, es utilizado para el desarrollo de páginas web.
- REACTJS: Es una biblioteca del lenguaje JavaScript para construir interfaces de usuario
- SLOT: Es un espacio de tiempo que se le asigna a una aeronave para su aterrizaje o despegue, es único y solo puede ser ocupado por una aeronave.
- CONTROLLER: Es la persona encargada de gestionar y velar los aterrizajes de las aeronaves en los distintos aeropuertos del Perú.
- SIMUR: Es un punto común en el espacio por donde todas las aeronaves deben pasar antes de su aterrizaje.
- BLOQUE: Es un espacio de tiempo donde se pueden cargar varios Slots, el mínimo bloque que es de quince minutos maneja cinco Slots con una separación de tres minutos entre ellos.
- ATFM: Es el Sistema de Gestión de Afluencia de Tránsito Aéreo – *Air Traffic Flow Managemet*
- INSIGHT: Es la clave, la esencia que nos permite encontrar la solución a un problema. Un camino, un dato que nos sugiere cómo resolver cualquier ecuación

por compleja que sea. Un *insight* no es la solución, es simplemente el punto que nos lleva al camino de esa solución.



REFERENCIAS

- Comando de la Aeronáutica. (2009). Capacidad de Pista. <https://www.icao.int/SAM/eDocuments/10ATFM%20Curso%20Teoria%20Capacidad%20Pista%20Spa%20only.pdf>
- Domínguez, W. (2017). Modelo de continuidad de servicios de las tecnologías de la información y comunicación utilizando cloud computing en la empresa Américas Potash Perú S.A – Proyecto de Minería No Metálica en el año 2016. [Tesis para optar el grado académico de maestro en Ingeniería de Sistemas, Universidad Católica Los Ángeles de Chimbote] http://repositorio.uladech.edu.pe/bitstream/handle/123456789/2372/TECNOLOGIAS_DE_INFORMACION_Y_COMUNICACION_CLOUD_COMPUTING_%20DOMINGUEZ_OLIVA_WENDY.pdf?sequence=4
- Microsoft Azure. (s.f.) ¿Qué es PaaS? Plataforma como servicio. <https://azure.microsoft.com/es-es/overview/what-is-paas/>
- Openwebinars s.f. Cloud Computing: Tipos de Nubes, servicios y proveedores [Figura] <https://openwebinars.net/blog/tipos-de-cloud-computing/>
- Pérez, A. (2012) Implementación de tecnología de Cloud Computing para ofrecer servicios de infraestructura (IaaS) en la Facultad de Telemática [Tesis para optar el título de maestro en computación]. Universidad de Coloma.
- Red Hat. (8 de mayo del 2020) ¿Qué es una API de REST? [.https://www.redhat.com/es/topics/api/what-is-a-rest-api](https://www.redhat.com/es/topics/api/what-is-a-rest-api).
- Rosales, E. (2010). Unacloud: Infraestructura como servicio para cloud computing oportunista. Bogotá
- WebEmpresa s.f. RestApi de WordPress: que es y cómo usarla. [Figura] <https://www.webempresa.com/blog/rest-api-de-wordpress.html>

García Ramos, Rocío. (2017). Design Thinking en Español

<https://www.designthinking.es/home/index.php>



BIBLIOGRAFÍA

- Aston, B. (2015). A brief history of JavaScript. [online] Medium.
<https://medium.com/@benastontweet/lesson-1a-the-history-of-javascript-8c1ce3bffb17>
- Atlassian. (s. f.). DevOps. <https://www.atlassian.com/devops>
- AWS (s. f.). Accelerate Your Cloud Migration with AWS. AWS. [Acelera tu migración a AWS] <https://aws.amazon.com/events/migration/overview/>
- AWS. (s. f.). What is DevOps? [¿Que es Devops?] <https://aws.amazon.com/devops/what-is-devops/>
- Cloudeconomics (2012): The Business Value of Cloud Computing [El Valor Comercial de la Nube] <http://www.cloudeconomics.com/>
- CIANCUTTI, J. (2010). 5 Lessons We've Learned Using AWS. Medium.
<https://medium.com/netflix-techblog/5-lessons-weve-learned-using-aws-1f2a28588e4c>
- Firebase. (s.f.) Cloud Firestore [La nube de Firestore] <https://firebase.google.com/docs/firestore>
- GitHub Pages. (s. f.). [paginas con GitHub] <https://pages.github.com/>
- GORBOUJ, H. (2018). Edge Computing Demystified: Understand the status and the future of Edge Computing. [Entendiendo el estado y el futuro de la era computacional]
- Hackernoon. (2018). Ten years of DevOps. What changed?[diez años de Devops, que ha cambiado?] <https://hackernoon.com/ten-years-of-devops-what-changed-6d0e9de1a60>
- Heroku. (s.f.) Documentation[Documentación] <https://devcenter.heroku.com/categories/reference>

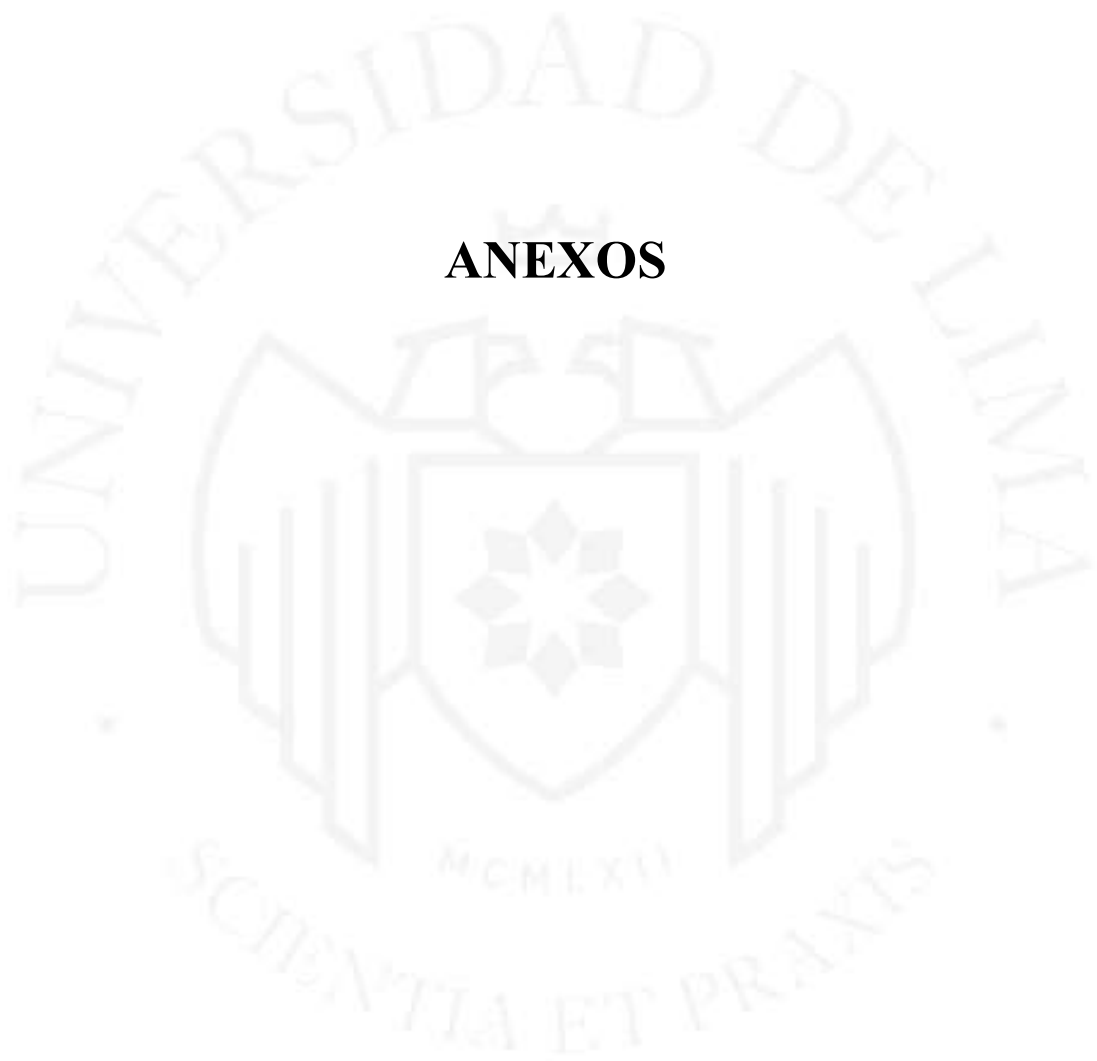
Loukides, M. (2017). The evolution of DevOps. O'Reilly. [La evolución de Devops]<https://www.oreilly.com/ideas/the-evolution-of-devops>

NADELLA, S. (2018). Pulsa actualizar. Madrid: Harper Collins

Wexler, J. (2018). [online]. How to Build a Simple Web Server with Node.js. Sitepoint. [Como construir de manera simple un servidor web con Node.js]
<https://www.sitepoint.com/build-a-simple-web-server-with-node-js/>



ANEXOS



**LOS ANEXOS NO ESTÁN DISPONIBLES POR
CONTENER INFORMACIÓN CONFIDENCIAL**

