	GESTIÓN DE SERVICIOS ACADÉMICOS Y BIBLIOTECARIOS		CÓDIGO	FO-GS-15	
			VERSIÓN	02	
	ESQUEMA HOJA DE RESUMEN			FECHA	03/04/2017
				PÁGINA	1 de 1
ELABORÓ		REVISÓ		APROBÓ	
Jefe División de Biblioteca		Equipo Operativo de Calidad		Líder de Calidad	

RESUMEN TRABAJO DE GRADO

AUTOR(ES): NOMBRES Y APELLIDOS COMPLETOS

NOMBRE(S): CRISTIAN LEONARDO APELLIDOS: PEÑARANDA MORA

FACULTAD: FACULTAD DE INGENIERÍAS

PLAN DE ESTUDIOS: INGENIERÍA DE SISTEMAS

DIRECTOR:

NOMBRE(S): ALVARO APELLIDOS: SALAMANCA LANDINEZ

NOMBRE(S): MILTON JESÚS APELLIDOS: VERA CONTRERAS

TÍTULO DEL TRABAJO (TESIS): DESARROLLO DE UN MVP (MINIMUM VIABLE PRODUCT) FUNCIONAL PARA LA SOLICITUD Y SIMULACIÓN DE CRÉDITOS EN LA EMPRESA ZERFTECH LLC.

Desarrollo de un MVP funcional de un simulador de créditos con el fin de solucionar la problemática de eficiencia y eficacia de las solicitudes de crédito presentadas con sistemas de terceros que dificultan las integraciones y la escalabilidad de las aplicaciones internas del cliente de la empresa ZERFTECH LLC.

PALABRAS CLAVE: MVP, simulador, solicitud, créditos

CARACTERÍSTICAS:

PÁGINAS: 49 PLANOS: ___ ILUSTRACIONES: ___ CD ROOM: ___

DESARROLLO DE UN MVP (MINIMUM VIABLE PRODUCT) FUNCIONAL PARA LA
SOLICITUD Y SIMULACIÓN DE CRÉDITOS EN LA EMPRESA ZERFTECH LLC.

CRISTIAN LEONARDO PEÑARANDA MORA

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

SAN JOSÉ DE CÚCUTA

2022

DESARROLLO DE UN MVP (MINIMUM VIABLE PRODUCT) FUNCIONAL PARA LA
SOLICITUD Y SIMULACIÓN DE CRÉDITOS EN LA EMPRESA ZERFTECH LLC.

CRISTIAN LEONARDO PEÑARANDA MORA

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS

DIRECTOR

ALVARO SALAMANCA LANDINEZ

DOCENTE

CODIRECTOR

MILTON JESÚS VERA CONTRERAS

DOCENTE

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

SAN JOSÉ DE CÚCUTA

2022



ACTA DE SUSTENTACIÓN DE TRABAJO DE GRADO

FECHA: 12 DE JULIO DE 2022

HORA: 4:00 P. M.

LUGAR: meet.google.com/opq-ymvo-iii

PLAN DE ESTUDIOS: INGENIERÍA DE SISTEMAS

TÍTULO DEL TRABAJO DE GRADO: "DESARROLLO DE UN MVP (MINIMUM VIABLE PRODUCT) FUNCIONAL PARA LA SOLICITUD Y SIMULACIÓN DE CRÉDITOS EN LA EMPRESA ZERFTECH LLC."

Jurados

ING. MARCO ANTONIO ADARME JAIMES

ING. FREDY HUMBERTO VERA RIVERA

ING. CARLOS RENE ANGARITA SANGUINO

DIRECTOR: ÁLVARO SALAMANCA LANDINEZ

CODIRECTOR: MILTON JESÚS VERA CONTRERAS.

NOMBRE DEL ESTUDIANTE	CÓDIGO	CALIFICACIÓN	
		NÚMERO	LETRA
CRISTIAN LEONARDO PEÑARANDA MORA	1151489	4,8	CUATRO, OCHO

MERITORIA

FIRMA DE LOS JURADOS

ING. MARCO ANTONIO ADARME JAIMES

ING. FREDY HUMBERTO VERA RIVERA

ING. CARLOS RENE ANGARITA SANGUINO

JUDITH DEL PILAR RODRIGUEZ TENJO
Coordinadora Comité Curricular

Tabla de contenido

Introducción	8
1. Presentación de la empresa	9
2. Planteamiento del problema	11
3. Objetivos	13
3.1. General	13
3.2. Específicos	13
4. Intereses de la pasantía.	14
4.1. Interés académico que tuvo la pasantía para el estudiante	14
4.2. Interés profesional que tuvo la pasantía para el estudiante	14
4.3. Interés laboral que tuvo la pasantía para el estudiante	15
4.4. Interés y aporte productivo para la organización donde se desarrolló la pasantía	15
5. Desarrollo del proyecto	16
5.1. Cumplimiento de la propuesta	16
5.1.1. Primera Entrega	18
5.1.2. Segunda Entrega	25
5.1.3. Entrega final	31
5.2. Técnicas y herramientas de desarrollo	32
5.3. Procedimientos utilizados en el desarrollo de la pasantía	34
5.4. Elementos innovadores de la propuesta	38
6. Resultados	41
6.1. Productos generados	41
6.2. Experiencia adquirida durante el desarrollo de la pasantía	42
7. Recomendaciones	44
8. Bibliografía	47
9. Anexos	48

Listado de tablas

Tabla 1 Resultados esperados por SPRINT y su completitud

16

Listado de figuras

Figura 1 Logo ZERF LLC, 2022, (https://www.zerf.com.ar/)	9
Figura 2 Integrantes equipo kayak	19
Figura 3 Mockups Simulador de créditos en Figma	22
Figura 4 Diagrama creación de la simulación y meta datos (deltas)	23
Figura 5 Cálculo de pasos del simulador de créditos	24
Figura 6 Diagrama recuperación de simulaciones viejas	25
Figura 7 Diagrama envió correo de bienvenida	26
Figura 8 Diagrama envió de correo del aval del crédito	26
Figura 9 Mockups de los templates de los correos	27
Figura 10 Diagrama de infraestructura de despliegue de kayak frontend	28
Figura 11 Monitoreo de errores kayak-backend API	29
Figura 12 Monitoreo de errores en Sentry de kayak-frontend en React Js	30
Figura 13 Diagrama propuesta de asignación avales	30
Figura 14 Integración continua kayak-back	31
Figura 15 Despliegue continuo en el servicio de AWS CodePipeline	32
Figura 16 Línea de tiempo de tecnologías usadas a lo largo de los sprints	34
Figura 17 Etapas metodología de desarrollo ágil	35
Figura 18 Diagrama comunicación deltas múltiples	40
Figura 19 Métricas de mejora de rendimiento con deltas múltiples	40
Figura 20 Kayak en producción (https://micredito.migrante.com/)	41

Introducción

La empresa ZERFTECH LLC es una casa de software enfocada desde el desarrollo inicial de un MVP (Minimum Viable Product), pasando por todo el proceso de identificación de necesidades y problemas de las empresas, alineados con los objetivos de las mismas, hasta el proceso de validación del MVP usando indicadores KPI (Key Performance Indicator) según las metas que desea alcanzar la compañía con dicho producto.

El presente documento quiere dar a conocer las actividades a realizar en la empresa ZERFTECH LLC para construir un MVP funcional de un sistema para la solicitud y simulación de créditos, partiendo desde el inicio del proceso de identificación de necesidad de la empresa, pasando por la implementación del MVP haciendo uso de la metodología ágil SCRUM, hasta al final llegar a la comprobación del éxito del mismo en base a indicadores KPI.

1. Presentación de la empresa

La empresa ZERFTECH LLC es una casa de software multinacional consolidada en Miami, Florida que se enfoca desde el desarrollo inicial de un MVP (Minimum Viable Product), pasando por todo el proceso de identificación de necesidades y problemas de las empresas, alineados con los objetivos de las mismas, hasta el proceso de validación del MVP usando indicadores KPI (Key Performance Indicator).

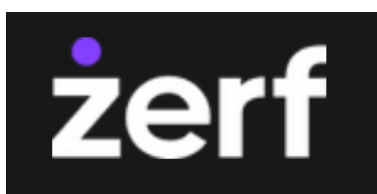


Figura 1 Logo ZERF LLC, 2022, (<https://www.zerf.com.ar/>)

Misión

Desarrollar la sociedad, identificando, comprendiendo y solucionando estratégicamente problemas de personas, organizaciones y/o comunidades, transmitiendo el valor de la aplicación de soluciones tecnológicas de calidad.

Visión

Formar una empresa con una identidad marcada por el compromiso de replantear los paradigmas de la sociedad en la que vivimos y modificarlos desarrollando productos de calidad, con equipos interdisciplinarios y profesionales que se sientan orgullosos y felices de pertenecer.

Destacar por el nivel de las soluciones brindadas, transmitiendo el valor de la aplicación de nuevas tecnologías a la resolución u optimización de desafíos presentes en la sociedad.

Conformar un grupo humano sano y un ambiente de desarrollo individual ideal para que las personas puedan crecer personal y profesionalmente dentro de la organización.

Valores institucionales

- Innovación
- Responsabilidad
- Confianza
- Diversidad
- Diversión
- Timing

Visión compartida

Deseamos trabajar con personas y/o organizaciones que compartan nuestros valores y tengan desafíos claros que quieran abordar en conjunto, involucrándose de manera completa al proceso de construcción de la solución.

Queremos transmitir la importancia de la utilización de la tecnología y sus aplicaciones a problemas industriales o sociales que puedan ser resueltos u optimizados, siendo responsables de comunicar eficazmente nuestros objetivos como empresa.

2. Planteamiento del problema

Tomando en cuenta la deficiencia en el proceso de solicitud de créditos de los Migrantes a Chile; actualmente estos requieren de todo tipo de solicitud de créditos para poder adquirir nuevos bienes como motos, carros u otros tipos de crédito. El medio por donde realizan esta solicitud de crédito; es a través, de un formulario de una aplicación de terceros llamada "123Formbuilder", el cual permite construir formulario de manera sencilla y rápida capaz de disponibilizar la solicitud de información a clientes potenciales de manera rápida; sin embargo el uso de esa aplicación de formularios de 123Formbuilder genera inconvenientes en la experiencia de usuario al momento de hacer la simulación o solicitud de su crédito, Siendo el mismo, un formulario muy estático y poco flexible en la carga de documentación e información, lo que vuelve el proceso tedioso y lento, reduciendo la tasa de conversión de la plataforma y reduciendo la cantidad de créditos colocados y por ende, las utilidades de la compañía. También al hacer integraciones de 123Formbuilder con la aplicación principal de la empresa ARC (Es donde se centraliza toda la información de las simulaciones y clientes) se vuelve una tarea compleja, debido a la poca flexibilidad de esta herramienta. Otros problemas que también presenta son la mantenibilidad y escalabilidad que tiene actualmente esta herramienta enfoque.

Ante este escenario, resulta fundamental, es necesario buscar una solución que permita aumentar el flujo de clientes que simulan créditos, volviendo más ameno este proceso para usuarios finales, y también, reducir el tiempo y el costo del proceso de aceptación de créditos por parte de la empresa, ya que con la herramienta 123Formbuilder se genera re-trabajo a la hora de la solicitud de información a los clientes, por esto se requiere construir un MVP (Minimum Viable Product) que pueda reemplazar totalmente la utilización de 123Formbuilder, que permita al equipo de tecnología tener más control sobre lo que se le solicita al cliente para la simulación de

su crédito, y permita un mejor mantenimiento y escalamiento con el fin de alcanzar los objetivos de crecimiento de la organización a largo plazo.

3. Objetivos

3.1. General

Desarrollar un MVP (Minimum Viable Product) web funcional para la solicitud y simulación de créditos de los clientes migrantes a Chile, a partir de los diseños UX/UI propuestos por ZERFTECH.

3.2. Específicos

- Analizar el proceso de solicitud de créditos, respecto a la integración con la aplicación principal en base a los lineamientos y estándares de la empresa ZERFTECH.
- Implementar una API que permita consumir los datos de los clientes y las simulaciones almacenadas en la aplicación principal de la empresa.
- Configurar entornos de pruebas(staging) y de producción del MVP funcional que permita la integración de la API con la Single Web Page Application.

4. Intereses de la pasantía.

4.1. Interés académico que tuvo la pasantía para el estudiante

El interés académico que se tuvo en esta pasantía fue el de aplicar los conocimientos adquiridos en la carrera de ingeniería de sistemas durante 10 semestres académicos, entre estos conocimientos están el de análisis y diseño de software, que permitió modelar los problemas presentados como historias de usuario en la pasantía, en donde se modelaban usando diagramas UML (Unified Modeling Language) lo que permitió ver un panorama general y específico de distintas propuestas para implementación de soluciones, despliegue de aplicaciones y a su vez, dejar documentación sobre el proyecto que será de utilidad para cuando ingresen nuevos desarrolladores al proyecto de Kayak.

4.2. Interés profesional que tuvo la pasantía para el estudiante

El interés profesional que se tuvo en esta pasantía fue el de crecer como desarrollador de software ya que en oportunidades anteriores había construido diferentes aplicaciones web, pero en esta pasantía, se contó con la oportunidad de aprender temas de infraestructura con AWS para el despliegue de aplicaciones frontend de manera manual y automática (Despliegue Continuo) usando diferentes servicios de Amazon como lo son CodePipeline, CodeBuild, CodeDeploy, CodeCommit, S3, CloudFront, entre otros. También se pudo conocer lo referente a la dinámica de trabajo en equipo con una metodología como Scrum para el desarrollo ágil del software de Kayak.

En general, mis habilidades tecnológicas y blandas se enriquecieron gracias a la pasantía que tuve con la empresa ZERFTECH LLC, gracias a que tuve la oportunidad y libertad de diseñar soluciones a partir de las necesidades y requisitos propios del proyecto de solicitud de créditos.

4.3. Interés laboral que tuvo la pasantía para el estudiante

El interés laboral que tuvo la pasantía fue adquirir experiencia en el uso de servicios Cloud con AWS, desarrollo e integración de APIs, también en el uso de GitHub Actions para automatización de procesos como Integración continua, y, por último, el de dejar una buena imagen en la empresa ZERFTECH para negocios o recomendaciones laborales a futuro.

4.4. Interés y aporte productivo para la organización donde se desarrolló la pasantía

El interés y aporte productivo para ZERFTECH LLC fue el desarrollo y puesta en producción de un MVP funcional para la solicitud y simulación de créditos de migrantes en Chile con el fin de reemplazar el uso del formulario hecho en la plataforma FormBuilder123.

El desarrollo del producto Kayak, permitió mejorar la experiencia UX/UI de los clientes que solicitan créditos y también mejorar el proceso de evaluación de los ejecutivos de dichos créditos, ya que ahora se disponibiliza toda la información referente a documentación e información personal del cliente en el sistema core de la empresa llamado Arc.

5. Desarrollo del proyecto

5.1. Cumplimiento de la propuesta

A continuación, se muestra la tabla de lo esperado en cada SPRINT de la pasantía, en los cuales de 5 SPRINTs el porcentaje global de completitud de las tareas que se esperaban en cada uno de ellos fue de un 100%.

Tabla 1

Resultados esperados por SPRINT y su completitud

Fases	Resultados esperados (Hitos)	Completitud
SPRINT 1	Mockup del MVP que se desea implementar para la aplicación de solicitud de crédito llamada Kayak Diagrama de comunicación entre las diferentes aplicaciones de las que depende Kayak.	100%
SPRINT 2	API de Kayak desarrollada en Ruby on Rails en su respectivo repositorio de GitHub. Documentación de la API de Kayak en Postman con las instrucciones necesarias para poder consumirla.	100%
SPRINT 3	Diseños (Mockups) de la estructura de los mails que se desean enviar a los clientes de la aplicación de solicitud de crédito.	100%

	<p>Diagramas de flujo donde se especifique en qué parte del flujo de la solicitud de crédito se van a enviar cada uno de los Mails que se van a crear.</p>	
SPRINT 4	<p>Ambiente de Staging disponible a través de un dominio público y alojado en el servicio de amazon S3 para el Frontend de la aplicación en ReactJs.</p> <p>Ambiente de Production disponible a través de un dominio público y alojado en el servicio de amazon S3 para el Frontend de la aplicación en ReactJs.</p> <p>Captures de aplicaciones creadas en el Servicio de Sentry, tanto para la API como para el Frontend de la aplicación.</p> <p>Diagrama del flujo que se utilizará para asignación de avales en el proceso de solicitud de crédito.</p>	100%
SPRINT 5	<p>Diagrama de manejo de estos de la solicitud de crédito hechas desde Kayak en Pipedrive.</p> <p>Captures del Continuos integration funcionando en Github.</p> <p>Captures del Continuous Deployment funcionando en AWS CodePipeline</p>	100%

A continuación, se pretende mostrar un consolidado de toda la información sobre las 3 grandes entregas que se hicieron a lo largo de la pasantía en la empresa ZERFTECH LLC, y a su vez, la información detallada de las actividades que se realizaron a lo largo de los 5 sprints de desarrollo del MVP del simulador de créditos.

5.1.1. Primera Entrega

En la primera entrega se consolida la información correspondiente al primer y segundo sprint realizados entre el 4 y el 29 de abril de 2022.

Durante el primer mes de ejecución de la pasantía, se alcanzaron los siguientes objetivos:

- Analizar la estructura existente del proceso de solicitud de créditos respecto a la integración con la aplicación principal en base a los lineamientos y estándares de la empresa ZERFTECH.
- Implementar una API que permita consumir los datos de los clientes y las simulaciones almacenadas en la aplicación principal de la empresa, llamada a partir de ahora 'Kayak'.

Para el cumplimiento de los anteriores objetivos se contaba con una base inicial de código acorde a la arquitectura que es utilizada en otros proyectos de la compañía. Antes de empezar con el diseño y el desarrollo de la API, fue necesario estudiar y aprender cómo funciona la arquitectura de la aplicación y los protocolos de comunicación que son utilizados con el fin de comunicarse con la aplicación principal, la cual recibe el nombre de Arc. Una vez familiarizado con la estructura de la aplicación Arc, fue necesario profundizar acerca de la estructura del proyecto de Kayak, el cual fue construido usando el framework de programación Ruby on Rails.

Con el fin de empezar el desarrollo del proyecto y alcanzar la construcción del PMV de solicitud de créditos en el tiempo estimado, se consolidó el equipo de trabajo Kayak con la siguiente composición:

- Un product Owner (PO)
- Una diseñadora UX/UI (UX/UI)
- Tres desarrolladores backend (DEVS)
- Un desarrollador frontend (DEV)
- Un facilitador agil (SM)

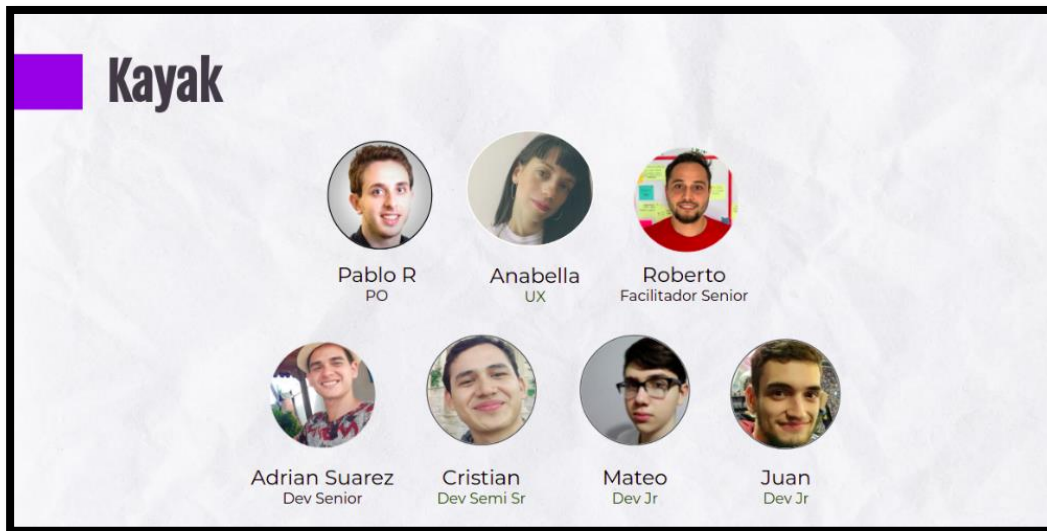


Figura 2 Integrantes equipo kayak

Responsabilidades de los roles

Product owner

- Se encarga de validar las necesidades del negocio y asegurarse de que aportemos valor al mismo.
- Generar un backlog y estar a cargo de la planificación del proyecto.

- Priorizar tareas.
- Refinar tareas.
- Definir KPI (Key Performance Indicators).

Diseñadora UX/UI

- Comprender al usuario, sus necesidades, emociones y características.
- Diseñar cómo el usuario va a interactuar con el producto.
- Estudia, mide y optimiza aplicaciones (itera).
- Analiza la competencia.
- Prototipos y Wireframes.

Desarrollador Backend

- Construir soluciones escalables y mantenibles en el tiempo que solucionen las problemáticas del negocio.
- Programación backend utilizando Ruby on Rails y otras tecnologías según las necesidades del negocio.
- Manejo de Base de datos PostgreSQL.
- Manejo de plataformas cloud como Heroku o AWS.

Desarrollador Frontend

- Programación frontend utilizando React JS, Redux y tecnologías a fines.
- Construir soluciones frontend con alto nivel de usabilidad y mantenible en el tiempo.
- Buen manejo de componentes en React Js que sirva para reusarse en diversos proyectos.
- Manejo de plataformas cloud como Heroku o AWS.

Facilitador Ágil

- Busca impulsar e inspirar formas innovadoras de hacer las cosas, fomentando la evolución del ecosistema.
- Ayuda a la organización estableciendo prácticas ágiles para una gestión y gobernanza efectiva del portafolio (desde que nace una idea hasta que se materializa en un producto).
- Fomenta la cultura de la agilidad enseñando, facilitando, asesorando y entrenando a la gerencia de tecnología y en un futuro al negocio.

Una vez se tuvieron los roles claros, se procedió con el inicio del proyecto en el cual la empresa definió un flujo de trabajo personalizado con el fin de tener un proceso de desarrollo ágil que nos permita llegar a alcanzar los objetivos del proyecto en los tiempos acordados. Según el equipo en donde se esté trabajando, se tiene una variación en las etapas del proceso de desarrollo del producto, pero en este caso, para el equipo Kayak se define como se muestra en la (Figura 17 Etapas metodología de desarrollo ágil)

Actividades Realizadas

Diseño solución de steps del proceso de solicitud de créditos

En esta primera actividad, con ayuda del PO y la diseñadora UX/UI, se hizo una bajada de requerimientos en base a cómo funciona el simulador de créditos con la plataforma FormBuilder, y se diseñó el nuevo flujo en la herramienta de diseño Figma en base a esos descubrimientos con el fin de solucionar los problemas que se presentaban en formbuilder tanto de usabilidad, como de persistencia de información.

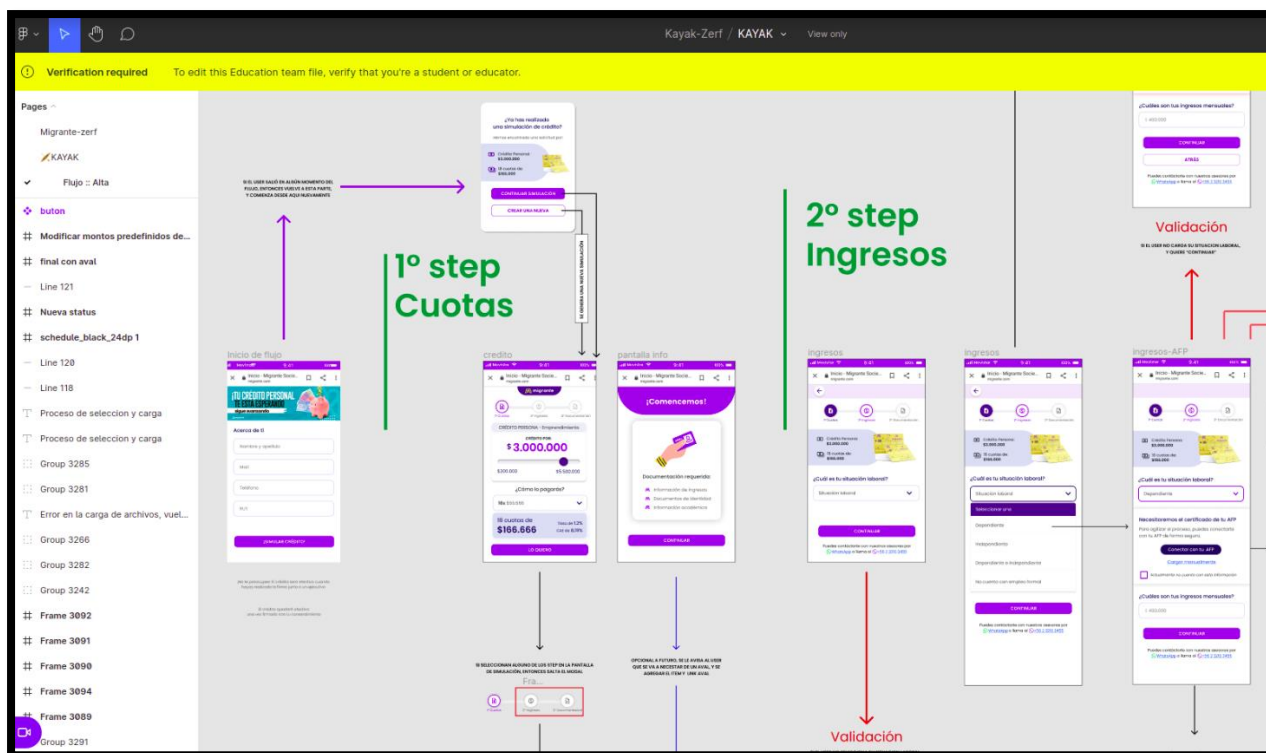


Figura 3 Mockups Simulador de créditos en Figma

Guardado de campos, secciones y creación de la simulación

Basado en la arquitectura hecha en el framework Ruby on Rails, en los diseños y requerimientos obtenidos en el diseño del proceso de solicitud de créditos, se desarrolló la lógica de guardado de campos y secciones que involucra la simulación de crédito, esto se hizo en base a la implementación de micro cambios (deltas) que se van agregando a la simulación a lo largo del proceso de carga de datos.

Estos micro cambios (deltas) se usan para hacer micro actualizaciones independientes al modelo simulaciones como actualización de montos, de datos del cliente, entre otros campos que buscan ir actualizando la información de manera asíncrona usando las colas de Sidekiq que implementan internamente Redis para persistir sus datos.

Por otra parte, estos micro cambios nos permiten tener la capacidad de crear la simulación de crédito a partir de los datos del cliente y los datos de la simulación sin tenernos que preocupar por la cantidad de datos extra que queremos agregarle a esta.

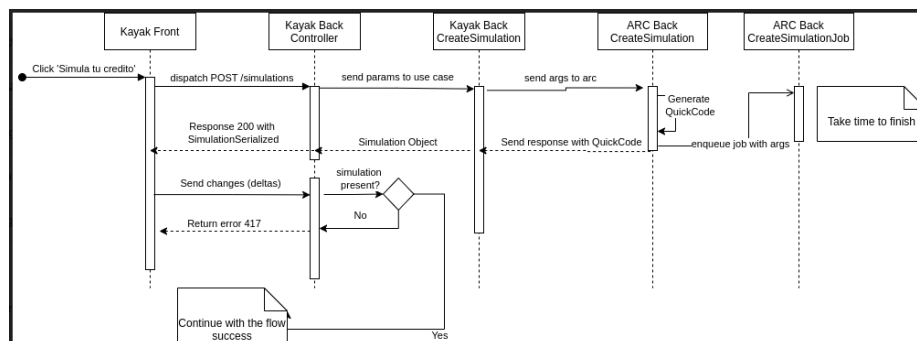


Figura 4 Diagrama creación de la simulación y meta datos (deltas)

Cálculo de completitud de secciones

De acuerdo a la información que se va completando y actualizando en el proceso de simulación del crédito, se hizo necesario saber que tan completo o no estaba el proceso de simulación basado en el diseño de steps que se hizo inicialmente, por esto se establecieron cuáles son los criterios de completitud de cada paso del flujo y luego se implementó el endpoint que se encarga de realizar esta acción.

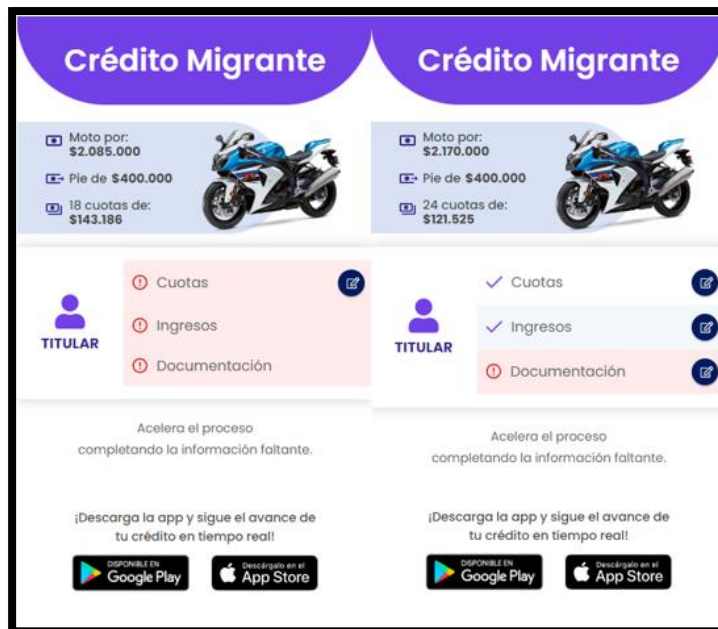


Figura 5 Cálculo de pasos del simulador de créditos

Recuperación de Simulaciones viejas

Con el fin de que el cliente pueda acceder a la última simulación que tiene activa en el proceso de solicitud de crédito y evitar la generación de simulaciones duplicadas, se implementó un endpoint capaz de recuperar la última simulación activa del cliente en un periodo de tiempo de 2 semanas.

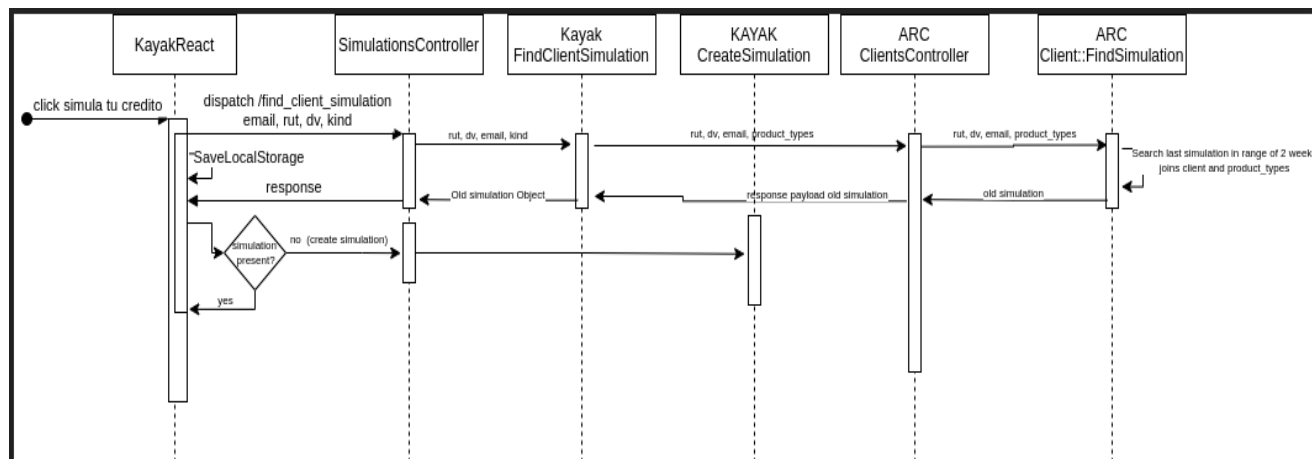


Figura 6 Diagrama recuperación de simulaciones viejas

5.1.2. Segunda Entrega

En la segunda entrega se consolida la información correspondiente al tercer y cuarto sprint realizados entre el 2 y el 27 de mayo de 2022.

Durante el segundo mes de ejecución de la pasantía, se alcanzaron los siguientes objetivos:

- Implementar una API que permita consumir los datos de los clientes y las simulaciones almacenadas en la aplicación principal de la empresa, llamada a partir de ahora 'Kayak'.
- Configurar entornos de pruebas(staging) y de producción del MVP funcional que permita integrar la aplicación con la Single Web Page Application.

Actividades Realizadas

Diseño de lógica de envío de correos y Armado de templates de mailing para comunicación con usuarios.

En esta primera actividad, con ayuda del PO y la diseñadora UX/UI se hizo una bajada de en qué parte del flujo de kayak se querían enviar los correos para notificarle a los clientes sobre el estado de su solicitud de crédito.

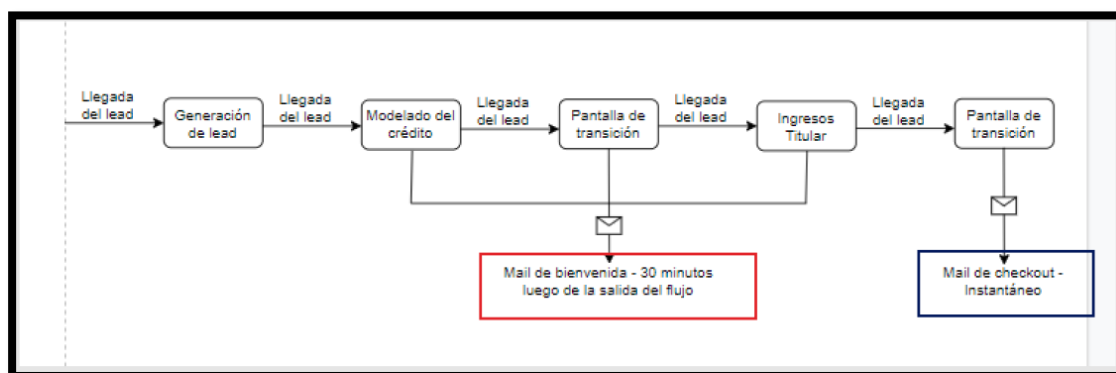


Figura 7 Diagrama envió correo de bienvenida

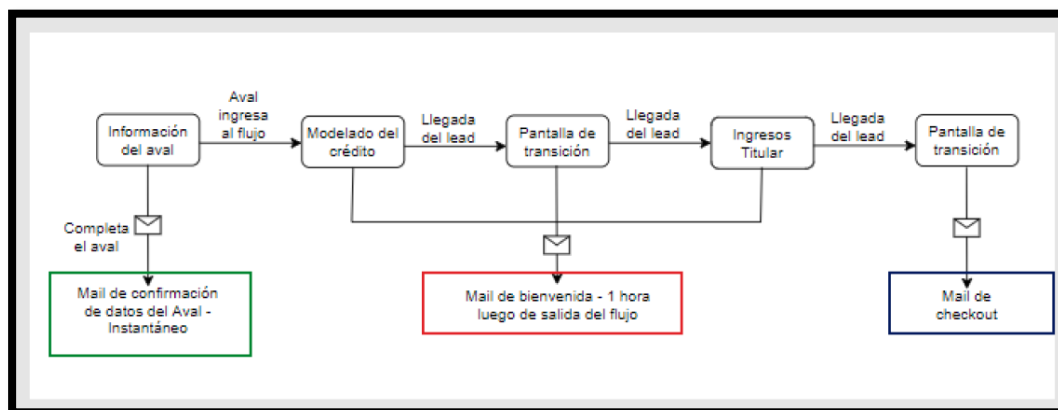


Figura 8 Diagrama envió de correo del aval del crédito

La diseñadora UX/UI realizó los diseños (mockups) respectivos de cada uno de los correos que se querían enviar a los usuarios, para así poder avanzar con las actividades posteriores de armado de templates en Mailgun para el envío de estos correos.

Basado en los mockups entregados por la diseñadora UX/UI, se procedió a armar el template base, el cual se usó para armar la estructura de los distintos mails que Kayak envía al cliente y al aval:

- Mail de bienvenida
- Mail checkout titular
- Mail checkout aval
- Mail de confirmación del aval
- Mail de rechazo de documentación

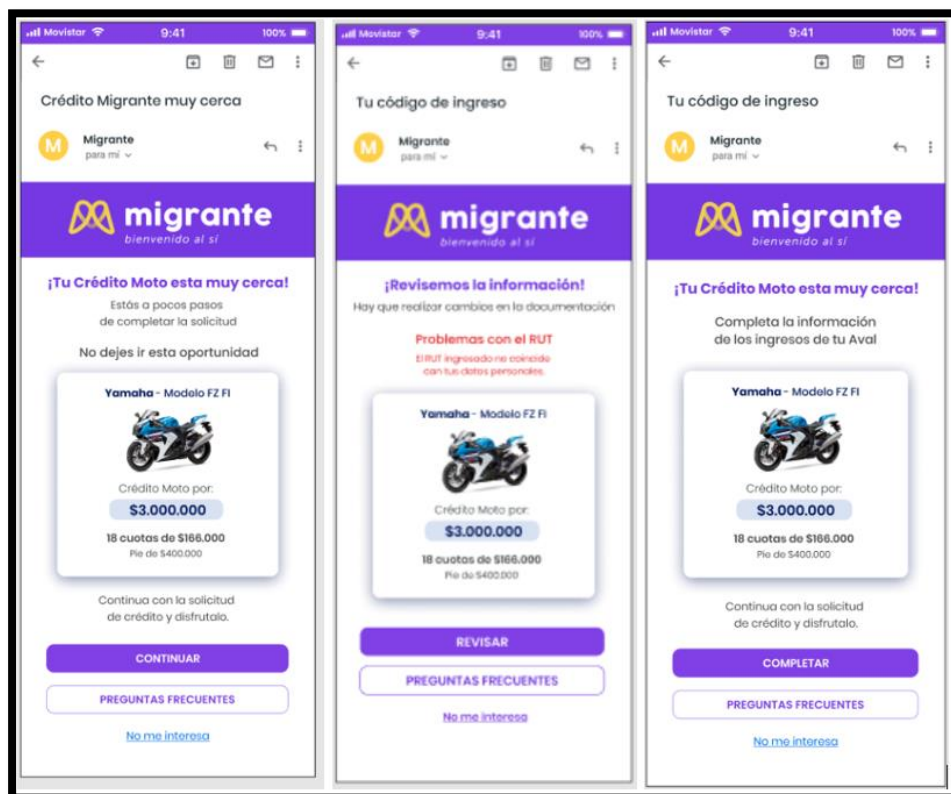


Figura 9 Mockups de los templates de los correos

Se usó HTML en formato de tablas para implementar los mockups propuestos por la diseñadora en la herramienta Figma, ya que es el formato que los proveedores de correos

electrónicos permiten utilizar para mostrar la información de manera correcta. Una vez se implementaron los templates de los mails, se subieron al servicio de Mailgun para poder utilizarlos y enviarlos desde la aplicación principal Arc.

Ambiente Staging y Production en AWS para Aplicación Frontend en React Js

Se despliego al ambiente de Staging y Production la aplicación Frontend de Kayak hecha en React Js, pero para llegar a cumplir esta tarea, primero se investigó cómo estaba la infraestructura de despliegue de las aplicaciones frontend en la empresa haciendo uso de la base de conocimiento de deajo el antiguo DevOps de ZERFTECH LLC. Una vez se entendió la forma en cómo funcionaba el despliegue en aplicaciones similares en la empresa, se procedió a hacer el despliegue de la aplicación frontend en un bucket de Amazon S3, el cual es accesible mediante el servicio Amazon CloudFront.

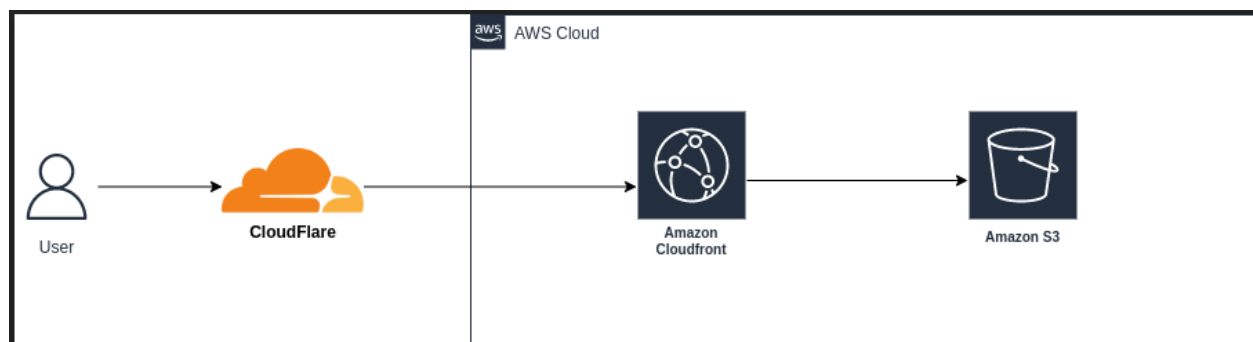
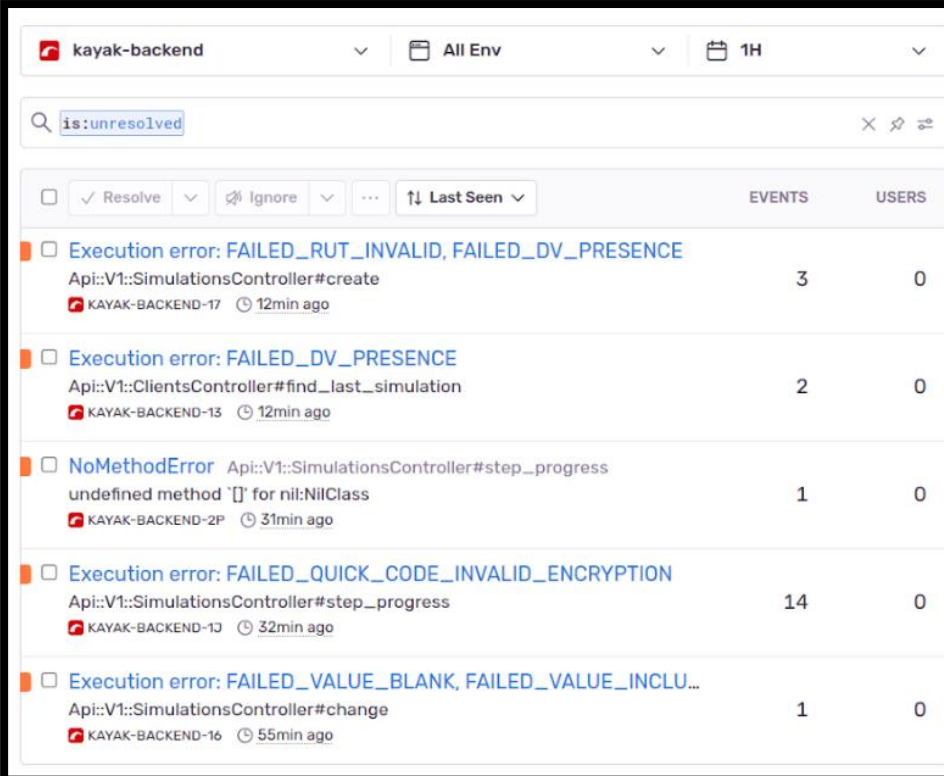


Figura 10 Diagrama de infraestructura de despliegue de kayak frontend

La API de Kayak backend no hubo necesidad de desplegarla, ya que se contaba con anterioridad con un Pipeline en la plataforma como servicio de Heroku, en donde estaban desplegados los entornos de staging y producción.

Monitoreo de errores en la API usando Sentry service

Una vez se desplegaron las aplicaciones de kayak frontend y kayak backend API, se vio la necesidad de monitorear los errores que ocurrieran en cada uno de los ambientes, con el fin de poder tomar acciones correctivas lo antes posible. Por lo anterior, se usó el servicio de Sentry, para monitorear las aplicaciones de kayak backend hecha en Ruby on Rails y la aplicación kayak frontend en React Js.



	EVENTS	USERS
<input type="checkbox"/> Execution error: FAILED_RUT_INVALID, FAILED_DV_PRESENCE Api::V1::SimulationsController#create KAYAK-BACKEND-17 12min ago	3	0
<input type="checkbox"/> Execution error: FAILED_DV_PRESENCE Api::V1::ClientsController#find_last_simulation KAYAK-BACKEND-13 12min ago	2	0
<input type="checkbox"/> NoMethodError Api::V1::SimulationsController#step_progress undefined method `[]' for nil:NilClass KAYAK-BACKEND-2P 31min ago	1	0
<input type="checkbox"/> Execution error: FAILED_QUICK_CODE_INVALID_ENCRYPTION Api::V1::SimulationsController#step_progress KAYAK-BACKEND-1J 32min ago	14	0
<input type="checkbox"/> Execution error: FAILED_VALUE_BLANK, FAILED_VALUE_INCLU... Api::V1::SimulationsController#change KAYAK-BACKEND-16 55min ago	1	0

Figura 11 Monitoreo de errores kayak-backend API

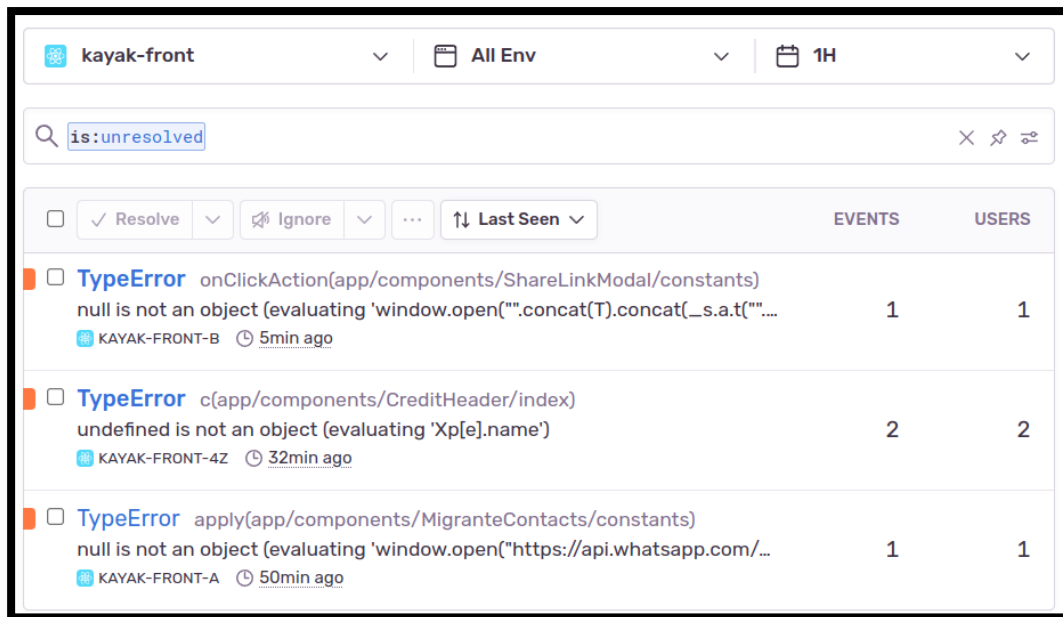


Figura 12 Monitoreo de errores en Sentry de kayak-froned en React Js

Lógica de asignación de avales a la simulación

Anteriormente, kayak manejaba la asignación de avales de manera directa sin tener en cuenta poder asignarlos o actualizarlos directamente en la simulación, por esto, surgió la necesidad de replantear el diseño de esta solución de tal manera que el cliente pudiese asignar un aval, actualizar sus datos personales o en caso de necesitar, reemplazar un aval por otro.

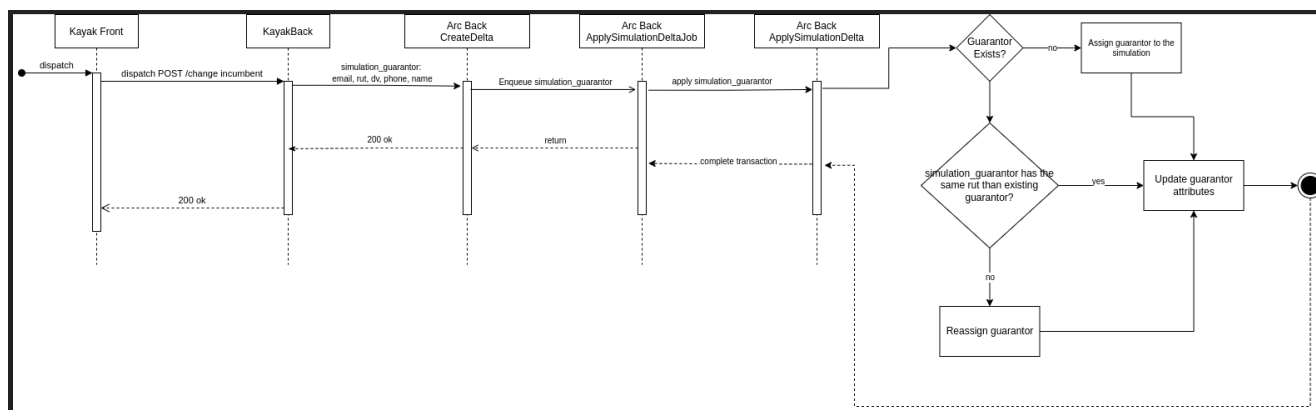


Figura 13 Diagrama propuesta de asignación avales

5.1.3. Entrega final

En la entrega final se consolida la información correspondiente al quinto sprint realizado entre el 3 de mayo al 10 de junio de 2022.

Durante el sprint número 5 (el final) se realizaron las actividades relacionadas a la integración y despliegue continuo de la aplicación de kayak.

Actividades Realizadas

Integración continúa configurada en Github Actions.

Durante esta tarea se configuró en GitHub Actions la integración continua de la API de Kayak backend tomando como referencia la configuración existente en la aplicación principal de Arc.

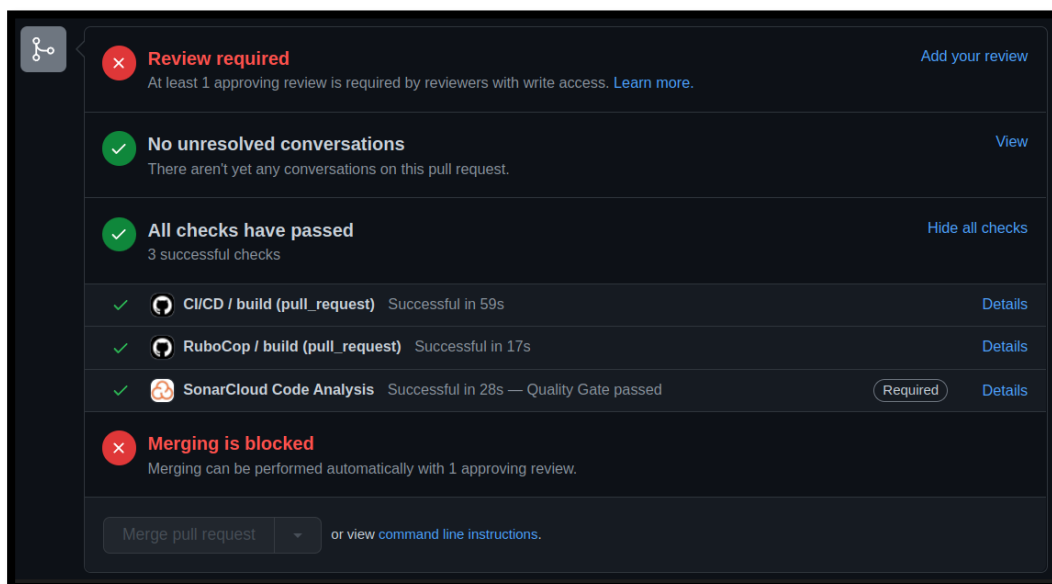


Figura 14 Integración continua kayak-back

Despliegue continuo configurado en el servicio de AWS CodePipeline

Durante esta tarea se hizo la configuración del despliegue continuo de la aplicación de Kayak frontend hecha en React Js, para esta tarea se hizo una investigación de cómo realizar tareas de CI/CD (Continuous integration/Continuous deployment) en amazon en el servicio de CodePipeline de amazon, en la cual se configuró el pipeline de amazon, con una integración con GitHub por medio del servicio de Star Connections, lo que permitió realizar despliegue continuo ya que cada cambio en la rama master o staging del repositorio de Kayak notificará a CodePipeline en tiempo real lo que permite automatizar el despliegue.

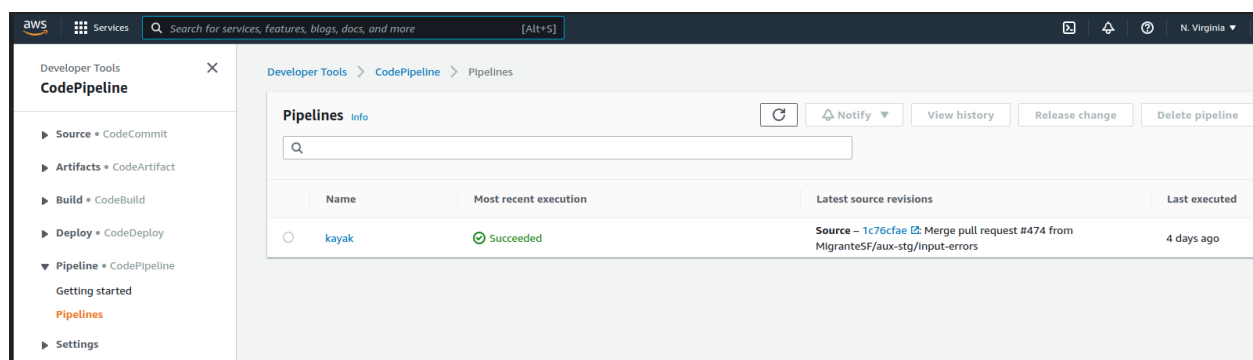


Figura 15 Despliegue continuo en el servicio de AWS CodePipeline

5.2. Técnicas y herramientas de desarrollo

- ClickUp:** ClickUp es una plataforma cloud-based colaborativa y una herramienta para la administración del proyecto que sirve para compañías de todo tamaño, desde pequeñas hasta grandes empresas y sin importar la industria o campo específico de acción de la empresa. Dentro de sus características más importantes destacan las herramientas de comunicación y colaboración dentro de los equipos de las empresas, pudiendo así crear tareas, asignar estados, personas, establecer deadlines del trabajo, entre otras características. (“ClickUp™ | One app to replace them all,” 2022)

- **Heroku:** Es una plataforma en la nube que nos permite construir, hacer despliegue, monitoreo y escalar aplicaciones de manera rápida gracias a su modelo de Plataforma como servicio (PaaS) (“What is Heroku | Heroku,” 2022)
- **AWS:** Amazon Web Services es la plataforma cloud más grande y con más servicios en la nube que otros proveedores de servicios cloud, entre estos están S3, CodePipeline, CloudFront, entre otros que nos ayudan con el escalamiento, mantenimiento, y gestión de distintos tipos de aplicaciones según las necesidades respectivas de cada una de ellas. (“What is AWS,” 2022.)
- **Pipedrive:** Pipedrive es un CRM (Customer Relationship Management) diseñado para ayudar a los equipos de ventas y marketing a la gestión de los clientes, prospectos y nuevos tratos de la organización. En este servicio podrás crear nuevos tratos o Deal con las características de los productos de ventas de la organización con el fin de llevar embudos con los tratos que se crean por cada cliente o nuevo prospecto del negocio. (“¿Qué es CRM? (Customer Relationship Management) | Pipedrive,” 2022.)
- **Zapier:** Es una plataforma que permite conectar todo tipo de aplicaciones e integrarlas entre sí, lo que hace posible automatizar flujos de trabajo de manera rápida y fácil. Cuenta con automatizaciones con aplicaciones de terceros predefinidas, como lo son CRM como Pipedrive, Gmail, Mailgun, entre otras aplicaciones de terceros, pero a su vez, las aplicaciones con las que no tiene integración directa, se puede automatizar las tareas en base a consumo de servicios REST API. (“Zapier | The easiest way to automate your work,” 2022.)
- **GitHub:** Es un repositorio online usado para el almacenamiento, seguimiento y colaboración en distintos proyectos de software, el cual permite que el desarrollador suba su propio código fuente para que posteriormente pueda ser usado por sus colaboradores y

también para automatizar acciones referentes a qué hacer con ese código, como, por ejemplo, integración y despliegue continuo usando GitHub Actions. (“What is GitHub? (¿And What Is It Used For?)”, 2022.)

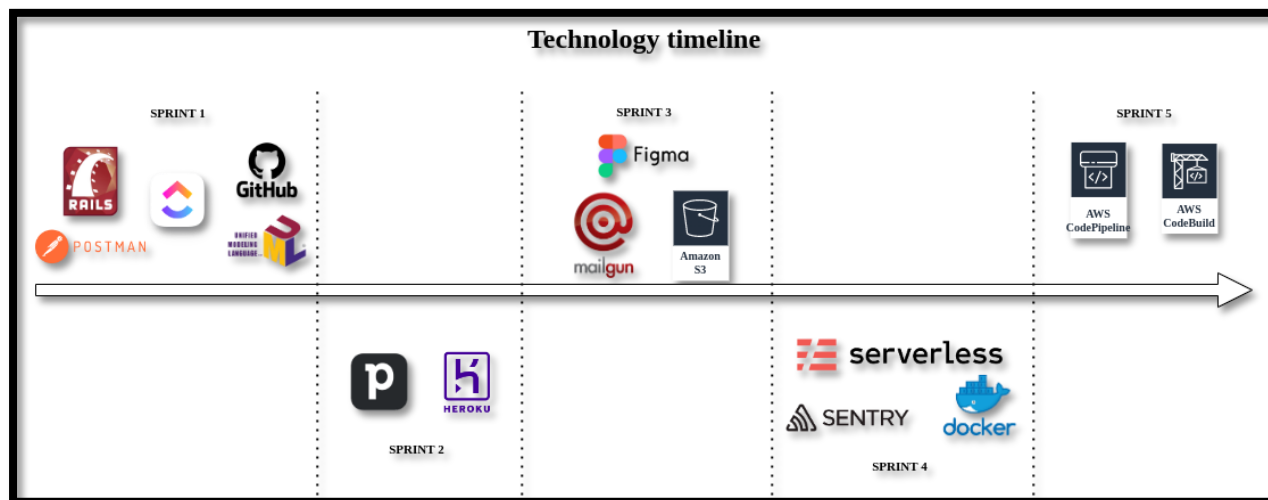


Figura 16 Línea de tiempo de tecnologías usadas a lo largo de los sprints

5.3. Procedimientos utilizados en el desarrollo de la pasantía

La metodología o el marco de trabajo que se usó con el fin de alcanzar el objetivo de construir el MVP funcional fue Scrum, pero no un Scrum puro, sino que también se optó por tomar una combinación entre Scrum y Kanban como una nueva propuesta de desarrollo ágil para todos los equipos de trabajo en la empresa, ya que se percataron que los tableros de tareas de algunos equipos son muy variantes y por ende se quiso aplicar la flexibilidad de toma de tareas que aporta Kanban durante el ciclo de desarrollo.

Las fases que tiene la nueva metodología que se planteó en base a las necesidades de cada equipo se muestran en la siguiente figura:

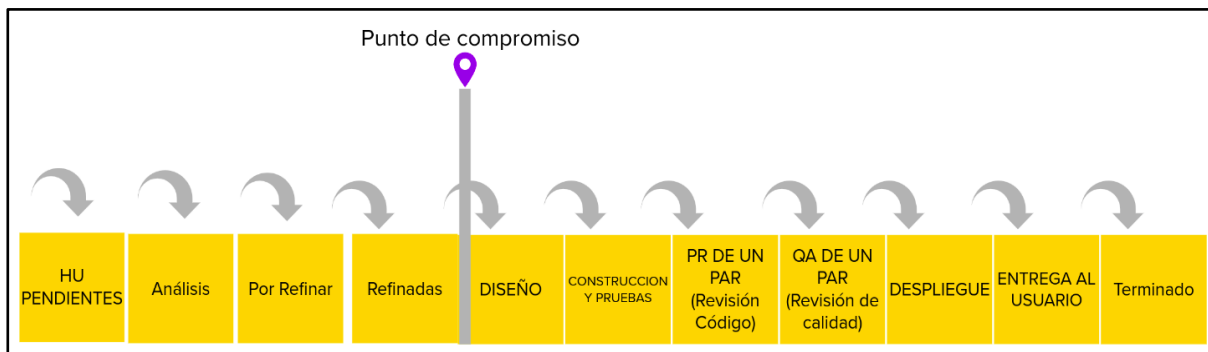


Figura 17 Etapas metodología de desarrollo ágil

HU (Historia de usuario) Pendientes

En esta primera fase, se dejaban las tareas que se iban creando en base a las necesidades o problemas que manifestaban los involucrados o interesados del proyecto Kayak, por ende, estas tareas se creaban sin tanta definición, y más enfocadas a cómo presentan el problema los involucrados al product owner (PO) del equipo.

Análisis

Una vez se tienen HU pendientes, y antes de pasar a ser candidatas de refinamiento, el equipo debe hacer un análisis general de lo que se busca de la HU y de lo que se necesita para llevar a cabo, esto con el fin de conocer la dimensión de la tarea y saber que tanto esfuerzo puede llevar realizarla al momento de llegar al refinamiento.

Por Refinar

En esta fase, caen las tareas que ya cuentan con un análisis previo y que han sido priorizadas de manera macro, es decir, según las necesidades del proyecto y de los involucrados el PO (Product Owner) toma la decisión de dejarla como candidata a refinar o no para que el equipo tome la tarea y la desarrolle.

Refinadas

En esta fase caen las tareas que se refinaron previamente, es decir, se cuenta con ceremonias específicas de refinamiento de tareas según la cantidad de tareas por refinar que existan y el trabajo ya tomado por el equipo. La idea principal de esta fase, es entender lo que se quiere lograr en la tarea de forma específica, darle su respectivo análisis técnico y de negocio con el fin de darnos una idea macro del esfuerzo que requiere realizar la tarea, y, por último, asignar a la persona idónea que puede tomar la nueva tarea refinada.

Diseño

En esta fase comienza el punto de compromiso del desarrollador, es decir, desde que se pasa la tarea de estado refinado a diseño comienza a correr el tiempo para medir cuál fue el tiempo de entrega de la tarea, esto con el fin de saber que tan bien estamos estimando el esfuerzo y saber a qué nos podemos comprometer a futuro con nuevos desarrollos que lleguen.

En esta fase de diseño se espera dar un análisis y diseño más técnico de la solución a historia de usuario, generalmente según el diseño que se proponga, este se aplica exactamente en la fase de construcción y prueba con el fin de no emplear tanto esfuerzo en esa última fase.

Construcción y pruebas

En esta fase se empieza a implementar la solución planteada en la fase de diseño y también se realizan test automatizados en caso de requerirse.

Revisión de código

En esta fase se espera a que otro desarrollador de la empresa revise el código que se hizo con el fin de que siga los estándares de código de la organización y que cuente con las

optimizaciones necesarias para que no coloquen lento al sistema principal. En esta fase también es un requisito dejar un video probando la funcionalidad o corrección de error de lo que se hizo, con el fin de tener más seguridad al momento de mezclar el código a la rama principal del repositorio en GitHub.

Revisión de calidad

Una vez se haya aprobado la fase anterior, el código se mezcla a la rama principal del repositorio en GitHub, y se hace el despliegue al entorno de pruebas o como le llaman entorno de “Staging”, en este caen todas las nuevas funcionalidades con el fin de que el desarrollador u otra persona pruebe la funcionalidad nueva que se hizo, y si todo anda como debería, se puede pasar al entorno productivo para que quede disponible para el usuario final.

Despliegue

Se llega a esta fase si el código fue debidamente probado en Staging y desplegado en el entorno productivo, es decir, esta es la fase en donde queda la funcionalidad en el entorno productivo accesible para todos los usuarios finales.

Entrega al usuario

Una vez se ha puesto en producción la nueva funcionalidad, es deber del equipo avisar a los involucrados a los que les afecta o interesa este nuevo desarrollo con el fin de que puedan estar al tanto de lo nuevo que se desarrolla o alguna corrección de algún error de alguna funcionalidad que los afecte directa o indirectamente.

Terminado

Esta es la fase final del ciclo de vida del desarrollo, en el que una tarea cae siempre y cuando se haya hecho el respectivo aviso a los usuarios finales o involucrados en caso de necesitarse.

Aplicando esta nueva metodología de trabajo ágil en el equipo, fue posible organizarnos de una mejor manera y así se pudo sacar el MVP funcional del simulador de créditos. Cabe mencionar que esta es una nueva metodología que se aplica en la empresa y ya que la transición es un proceso algo complejo, entonces la aplicación de este nuevo proceso se fue haciendo gradualmente con el fin de que no afectará la forma en que se llevaban a cabo los proyectos anteriormente.

5.4. Elementos innovadores de la propuesta

Carga de archivos a AWS de manera asíncrona

En kayak se estaban presentando muchos errores de 499 Client Request Interrupted, debido a que los archivos que cargaban los clientes de manera manual, a veces pesaban mucho y hacía que el tiempo de carga en el servidor de la API y de la API al servidor de AWS S3 incrementara los tiempos de respuesta a la aplicación frontend y esta interrumpiera la conexión entre ambos. Por lo anterior, se diseñó una solución que se baja en la carga asíncrona en AWS S3 en la siguiente secuencia:

- Se sube el archivo de kayak backend API
- Se guarda en la base de datos Postgresql
- Se encola un llamado asíncrono para procesar el archivo y subirlo a AWS
- Se procesa el archivo y se sube a AWS S3

- y, por último, se borra el archivo de la base de datos de Postgresql para que no consuma espacio de almacenamiento.

Mejora de rendimiento de Kayak en la aplicación de micro cambios en la simulación de crédito

La arquitectura base que se había diseñado en kayak para la actualización de la simulación de manera asíncrona aplicando micro cambios (deltas) contaba con problemas de rendimiento debido a que todos esos micro cambios se encolaban en una cola de procesos asíncronos y al momento de salir a entorno productivo estos problemas de rendimiento fueron más notorios ya que se llegaron a encolar más de 80 mil procesos solo para aplicar cambios a la simulación, esto ocasiona que los tiempos de respuestas de la aplicación se incrementaran, y que también, debido a la cantidad de hilos procesando al mismo tiempo y accediendo a la base de datos, que esta última se quedara sin recursos suficientes para acceder a ella y por ende, que colapsara la aplicación.

Por lo anterior, se hizo una evaluación de cómo estaba la arquitectura de los deltas en kayak, y posteriormente se elaboró una propuesta que se presentó a los líderes técnicos de la empresa. Esta propuesta se basaba en la solución de manejar micro cambios por lotes, lo cual hacía que se redujeran la cantidad de procesos en cola, la cantidad de transacciones de la base de datos y mejorar considerablemente los tiempos de respuesta de la aplicación.

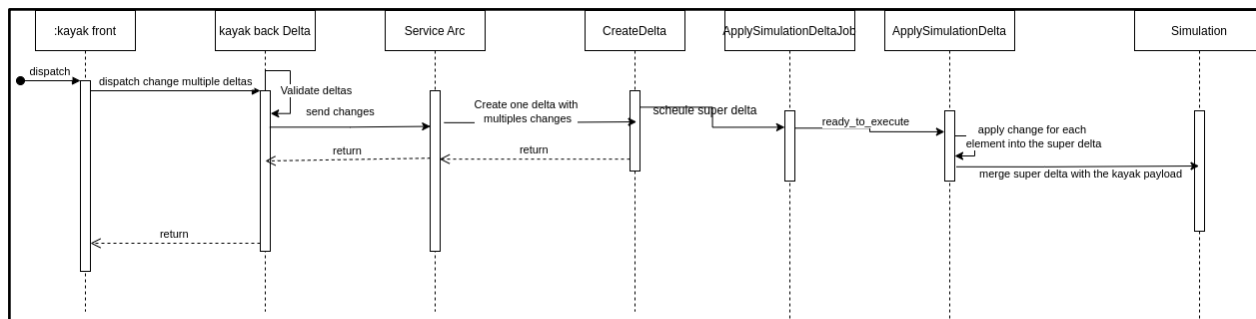


Figura 18 Diagrama comunicación deltas múltiples

La propuesta que se hizo se aprobó y se implementó, sus resultados fueron los siguientes:

Step	single deltas	multiple deltas
find_last_simulation	500ms	272.56ms
create simulation	2.71s	2.02s
/change after create simulation	1.84s	607.60ms
update simulation	1.13s	331.49ms
change after update simulation	860.96ms	526.57ms
change after incomes pages	1.76s	518.12ms
change after documentation page	1.64s	350.02ms
change after confirmation modal client	702.31ms	589.84ms
change after "seguir sin aval" button	1.20s	546.56ms
change after "entiendo" status page button modal	830.73ms	389.77ms
change after "Completar info aval" in status page	737.77ms	369.68ms
change after info titular	1.52s	821.60ms
change after incomes aval	1.93s	497.98ms
change after documentation aval	1.52s	305.51ms
change after transition page final aval	1.03s	543.94ms
Created deltas in all the flow	92 deltas	38 deltas

Figura 19 Métricas de mejora de rendimiento con deltas múltiples

Como se muestra en la figura anterior, hubo una reducción de micro cambios simples a micro cambios múltiples por lotes de casi un 60%, esto hizo que todo el flujo de simulación de créditos fuese más rápido, y que la aplicación core de la compañía optimizara recursos computacionales.

6. Resultados

6.1. Productos generados

El producto que generó la propuesta fue Kayak, una plataforma web para solicitud y simulación de créditos para migrantes en Chile, dentro de los créditos que pueden simular están el crédito consumo, moto, auto, electro movilidad, entre otros donde existe la posibilidad de simular usando un aval o no, según la necesidad del cliente y el tipo de crédito al cual desea acceder.

Este producto es uno de los más importantes de la empresa ya que Kayak es la entrada principal de clientes de la empresa Migrante SF, ya que no solamente simulan su solicitud de crédito, sino que directamente solicitan el acceso al crédito subiendo todo lo que es su información personal como documentos de identificación nacional o información financiera como documentos de liquidación de sueldo, deudas, entre otros. Toda esta información permite que la empresa evalúe las solicitudes de crédito, y evalúe si un cliente es apto o no para recibir el crédito.

The image displays four sequential screens of the Kayak credit application process for a motorcycle loan. Each screen features a progress indicator at the top with three steps: 1º Crédito, 2º Ingresos, and 3º Documentación.

- Screen 1 (Crédito Moto sin Aval):** Shows a total value of \$2,000,000. The user selects 'En cuotas' (in installments) with a pie of \$400,000. The installment amount is set to 24x \$117,678. The interest rate is 2.79% and the CAE is 48.23%. A 'CONTINUAR' button is at the bottom.
- Screen 2 (¿Cuál es su situación laboral?):** Asks for the user's labor situation. It requires a CMF certificate and offers to connect via 'Clave Única' or 'Cargar manualmente'. A 'CONTINUAR' button is at the bottom.
- Screen 3 (¿Tienes cédula de identidad?):** Asks if the user has an ID card. A 'CONTINUAR' button is at the bottom.
- Screen 4 (Crédito Migrante):** A summary screen showing the loan details (Moto por \$2,000,000, Pie de \$400,000, 24 cuotas de \$117,678). It lists 'TITULAR' with checked items: Cuotas, Ingresos, and Documentación. It includes a thank-you message and a call to action to download the app from Google Play or the App Store.

Figura 20 Kayak en producción (<https://micredito.migrante.com/>)

6.2. Experiencia adquirida durante el desarrollo de la pasantía

Durante el desarrollo de la pasantía se logra un alcance significativo en el aprendizaje de habilidades blandas y tecnológicas, en primera instancia habilidades blandas, como lo son habilidades de comunicación que mejoraron gracias al trabajo en equipo de la célula o equipo de trabajo de Kayak, ya que se pudo compartir propuestas de solución a problemas, también debatir sobre soluciones que se querían implementar en la aplicación, por otra parte si alguno del equipo tenía alguna duda sobre lo que se estaba desarrollando, todos los miembros del equipo ayudaban a que se pudiese destrabar ese desarrollo o tarea con el fin de lograr alcanzar los objetivos propuestos del proyecto.

Por otra parte, las habilidades tecnológicas también mejoraron, ya que hubo oportunidad para el aprendizaje de muchas cosas nuevas durante el desarrollo del proyecto como lo son:

- Docker: Dockerización de APIs desarrolladas en Ruby on Rails usando Docker y Docker Compose con el fin de que, si algún desarrollador nuevo llega al proyecto, le sea transparente colocar a funcionar el proyecto localmente y no tenga que colocarse a la tarea de instalar dependencias del proyecto y configuración de las mismas.
- Despliegue de aplicaciones frontend: Despliegue de aplicaciones que no requieren de un servidor para procesamiento de datos, es decir, aplicaciones netamente frontend usando el servicio de Simple Storage Service (S3) de AWS y también, a disponibilizar para el acceso al público usando CloudFront de AWS y CloudFlare.
- Despliegue continuo en AWS: Automatización del despliegue de aplicaciones frontend en AWS S3 usando CodePipeline que a su vez usa los servicios de CodeCommit, CodeBuild y CodeDeploy.
- Serverless en AWS: Se aprendió de manera general cómo funciona el panorama serverless en Amazon y haciendo uso del framework Serverless en Ruby usando el framework Sinatra,

debido a que se necesita de un framework ligero que permita a la Lambda en AWS disponibilizar la aplicación de manera rápida con el fin de poder procesar las solicitudes de los usuarios.

- Integración continua: Integración continúa haciendo uso de GitHub Actions para aplicaciones en Ruby on Rails, en donde se corren todos los test de la aplicación y todos los validadores de estándares de código (Linters), asegurando que la aplicación funcione correctamente y estén los estándares básicos de códigos aplicados en esta.

En general, se logró consolidar el aprendizaje inicial o básico sobre el uso de varios servicios de Amazon Web Services (AWS), ya que estos servicios tienen una amplia gama de documentación relevante para su configuración, buenas prácticas, ejemplos de uso, y también de formas de integrarlos con muchos otros servicios con el fin de que interactúen en el ecosistema Cloud de AWS. No obstante, lo que se alcanzó a hacer y a aprender en el desarrollo de la pasantía es enriquecedor para la experiencia profesional, ya que, gracias a esto, en un futuro nos podremos defender fácilmente en temas de despliegue continuo e integración continua de aplicaciones y seguramente aprender mejores estándares para llevar a cabo todas estas prácticas.

7. Recomendaciones

Al momento de hacer una pasantía, lo más importante es estar abierto a aprender, no importa que tan difícil o complicado que sea algo, sabe que tendrá el apoyo de las personas que están en la empresa, o en mi caso, las personas que son parte del equipo. Es bueno apoyarse en ellos ya que la idea de estos espacios de pasantías es que el estudiante aprenda a desenvolverse en el ámbito laboral, y que mejor forma de hacerlo que estar resolviendo dudas y ver cómo resuelven los problemas las personas con experiencia del equipo y de la organización en general.

Por otro lado, también es importante consultar información en inglés, ya que, la mayoría de los problemas que nos pasan al momento de desarrollar software, suelen haber sucedido en otros proyectos, y generan foros, documentación, artículos, en donde te ayudan resolviendo dudas, errores o te enseñan algo nuevo que te permita implementar o mejorar algo del software en el que estás trabajando. Así que hablando desde la experiencia adquirida, si nos quedamos bloqueados en algún aspecto del desarrollo o de diseño de alguna solución, que no nos de pena alzar la mano y preguntar a alguna persona del equipo para que nos enseñe desde su experiencia o ir a consultar documentación en inglés en busca de resolver este problema, diría que el orden lógico sería, ir a consultar primero y si aún no pudo resolver el problema, ahí sí, pedir ayuda a alguien con mayor experiencia que te pueda enseñar y te ayude a desbloquearte.

También recomiendo tomar una postura proactiva durante el desarrollo de alguna práctica o pasantía empresarial, ya que no muchas veces las tareas van a estar definidas al detalle entonces en esos casos tendrás que tener criterio y decidir qué es lo mejor en base a lo que hay definido en la tarjeta, y si requieres de alguien más para completar la tarea, trata de dejar la mayor cantidad de insights posibles, de modo que la persona que vaya a tomar esa dependencia pueda

solucionarlo de una forma más rápida gracias a que dejaste todo listo para que el tomara la tarea o si ves que puedes hacerlo sin esa persona, también puedes ayudarlo con esa parte del trabajo, ya que recuerda, tú no eres solo responsable de tus tareas, eres responsable de que un proyecto entero salga adelante, y eso implica que también debes apoyar a tus compañeros para que esto sea posible.

Por último, mucho de lo aprendido a lo largo de la carrera de ingeniería de sistemas fue útil para aplicarlo durante el desarrollo de la pasantía, la experiencia en maratones de programación ya que desarrolla más la algoritmia del estudiante, el análisis y diseño de software haciendo uso de diagramas UML, entre otras áreas que fueron de ayuda para este proceso. Pero sin embargo hay algunas cosas en las que no se hace énfasis en la carrera que sería bueno aprender con el fin de estar a la par del mercado laboral existente como, por ejemplo,

- Profundizar sobre temáticas de metodologías de desarrollo de software ágil que comúnmente usan en la industria, como Scrum, kanban, ShapeUp, etc.
- Preparar al estudiante para presentar entrevistas para cargos en áreas de tecnologías, tal vez esto pueda sonar algo muy trivial, pero tener experiencia al momento de llevar a cabo una entrevista laboral ayudará al estudiante a que consiga su primer empleo, ya que hay una gran variedad de tipos de entrevista según la empresa a la que vaya a aplicar.
- Impulsar el desarrollo en diferentes lenguajes de programación durante la carrera permitirá al estudiante tener un panorama más amplio sobre los lenguajes más demandados por empresas de tecnología o startups en el mercado laboral.
- Impulsar o promover en algún punto de la carrera de ingeniería de sistemas un plan de carrera para el estudiante a modo materias optativas u organizar

bootcamps con el fin de que él pueda especializarse en algún área específica del mundo de desarrollo de software según las preferencias de cada estudiante, como, por ejemplo, Backend, Frontend, Full Stack, DevOps, entre otras áreas. Esto podría ayudar al estudiante a tener un horizonte fijo y así poder tener mejores oportunidades en el mercado laboral.

8. Bibliografía

Zapier. *Zapier | The easiest way to automate your work.* (s.f.). <https://zapier.com/>

Pipedrive. *¿Qué es CRM? (Customer Relationship Management) | Pipedrive.* (s.f.).
<https://www.pipedrive.com/es/blog/que-es-crm>

Amazon Web Services. *What is AWS?* (s.f.). <https://aws.amazon.com/what-is-aws/>

Heroku. *What is Heroku | Heroku.* (s.f.). <https://www.heroku.com/what>

123 Form Builder. *123 Form Builder: Crear Formularios Online Gratis.* (s.f.).
<https://www.123formbuilder.com/es/>

ClickUp. *ClickUpTM | One app to replace them all.* (s.f.). <https://clickup.com/>

ClickUp. *ClickUp Reviews, Demo & Pricing - 2022.* (s.f.).
<https://www.softwareadvice.com/project-management/clickup-profile/>

HubSpot. *What is GitHub? (¿And What Is It Used For?)* (s.f.).
<https://blog.hubspot.com/website/what-is-github-used-for>

9. Anexos

Anexos primera entrega

Mockups Kayak:

https://drive.google.com/drive/folders/19R_QhqhjmT2nsJu5j1jC8RnypBoDko6O?usp=sharing

Documentación backend API y diseños:

<https://drive.google.com/drive/folders/1w9WsJIgoskJDeYL8byvzwFzE7GaITfVR?usp=sharing>

Ceremonias, reuniones y otras evidencias:

<https://drive.google.com/drive/folders/1iLE3mksBxWwB9YmpZzVCG-nMVsIi40e?usp=sharing>

Anexos segunda entrega

Ceremonias, reuniones y otras evidencias:

<https://drive.google.com/drive/folders/1IDs1ujSoQCPUm23Anf-kaYiauP2iN-eq?usp=sharing>

Diagramas de kayak y demo aplicación funcionando:

https://drive.google.com/drive/folders/12s_o5Y9FRNiEmeZPaJQ9Nm1bBX2nIypr?usp=sharing

Mockups del mailing para kayak:

<https://drive.google.com/drive/folders/1w9WsJIgoskJDeYL8byvzwFzE7GaITfVR?usp=sharing>

Evidencias monitoreo de errores con Sentry:

https://drive.google.com/drive/folders/1NPFzZyGBFC7csHiyw-tzRe8AYsAt_qfJ?usp=sharing

Anexo entrega final

Evidencias entrega final, diagramas, cartas de finalización y evaluación de la pasantía:

https://drive.google.com/drive/folders/1ZFKAPdPhcZERrzP20ysEto_u0QzfDiLS?usp=sharing

Anexo acta finalización de la pasantía

Acta de finalización:

https://docs.google.com/document/d/17IfgCYK6wSfLkk_SaLJGqPGMZK2JZ6R3/edit?usp=sharing&oid=114629095929776377718&rtpof=true&sd=true

Anexo repositorio en Github

README en GitHub con información de la pasantía:

<https://github.com/cristianleonardop/zerf-internship>