UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PRÁCTICA PROFESIONAL

ACKLEN AVENUE | HERO UNIT

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTADO POR:

11641022 JUAN LUIS GUEVARA MONDRAGÓN

ASESOR: LICDA. TANIA LUCILA MEZA AMADOR

CAMPUS TEGUCIGALPA; ABRIL, 2021

RESUMEN EJECUTIVO

El presente informe tiene como objetivo describir el trabajo realizado durante el transcurso de la práctica profesional de la carrera de Ingeniería en Sistemas Computacionales, que se llevó a cabo en el área de desarrollo de software de Acklen Avenue | Hero Unit desde octubre del 2020 hasta marzo del 2021.

Acklen Avenue | Hero Unit ha trabajado en el desarrollo de soluciones de software para muchos clientes externos, de igual manera, también se ha trabajado en el desarrollo de distintos proyectos internos entre los cuales se encuentra Acera. Este es un proyecto interno cuyo objetivo es entrenar a sus empleados y mantener un seguimiento del progreso del entrenamiento realizado por las personas que trabajan para la empresa.

Durante la práctica profesional se trabajó completamente dentro del departamento de desarrollo de software en el proyecto interno Acera, esto se realizó de forma exclusivamente remota ya que esta empresa trabaja de esta manera. En este periodo de tiempo se realizaron múltiples actividades asignadas tales como desarrollos de nuevas características, modificaciones y mejoras a características existentes, pruebas sobre componentes desarrollados, integraciones con herramientas de seguimiento de errores externas, uso de funciones en la nube, entre otras. Esto fue de mucha utilidad de cara al aprendizaje adquirido en estas áreas en las cuales no se poseía una experiencia muy vasta. Dentro del proyecto interno Acera se utilizaron las tecnologías de React, Node.js, Firebase y Gitlab.

Todas las actividades realizadas en este tiempo fueron revisadas y probadas por el asesor de calidad asignado en el equipo del proyecto interno Acera, de esta forma se validó la funcionalidad y calidad de estas.

ÍNDICE DE CONTENIDO

ÍNDICE DE ILUSTRACIONES

LISTA DE SIGLAS Y GLOSARIO

API	Interfaz de Programación de Aplicaciones
CRAI	Centro de Recursos para el Aprendizaje y la Investigación
DOM	Modelo de Objetos del Documento
HTML	Lenguaje de Marcas de Hipertexto
HTTP2	Protocolo de Transferencia de Hipertexto, versión 2
JSON	Notación de Objeto de JavaScript
JSX	Extensión de JavaScript
SQL	Lenguaje de Consulta Estructurada
SSL	Capa de Conexión Segura
SVG	Gráficos Vectoriales Escalables
TPS	Sistema de Producción de Toyota

Firebase	Es una plataforma digital que se utiliza para facilitar el desarrollo de aplicaciones web o móviles de una forma efectiva, rápida y sencilla. (¿Qué es Firebase?, 2019)
JavaScript	Es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web. (¿Qué es JavaScript?, 2020)
Kanban	Método de administración de tareas y flujos de trabajo usado especialmente en las empresas que trabajan en desarrollo de software. (Significado de Kanban, 2016)

Node.js	Entorno de código abierto y multiplataforma que ejecuta código de JavaScript. (¿Qué es Node.js y para qué sirve?, 2020)
Postman	Una herramienta dirigida a desarrolladores web que permite realizar peticiones HTTP a cualquier API. (¿Qué es Postman?, 2020)
React	Es una librería de JavaScript utilizada en el desarrollo web para la creación de elementos interactivos en sitios web (Tech 101: ¿Qué es React?)
Scrum	Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. (Qué es Scrum)
Scrum Master	Facilitador de proyectos, es la figura que lidera los equipos en la gestión ágil de proyectos. Su misión es que los equipos alcancen su objetivo, eliminando cualquier dificultad que puedan encontrar en el camino. (Definición y Características del Scrum Master, 2015)
Software	Término informático que hace referencia a un programa o conjunto de programas de cómputo, así como dato, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático. (Significado de Software, 2019)
Sprint	Mini proyecto de no más de un mes, cuyo objetivo es conseguir un incremento de valor en el producto. (Metodología 'scrum': ¿Qué es un Sprint?, 2019)

Web	Sistema de gestión de información más popular para la transmisión de datos a través de internet. (Significado de Web)

I. INTRODUCCIÓN

En la sociedad, la tecnología se ha convertido en un pilar muy importante, y ésta se encuentra en todas partes, desde lo más cotidiano como el entretenimiento, en el transporte, en las comunicaciones, y hasta en las industrias. Aquí reside la importancia del software y su desarrollo, esto debido a que cada día la demanda en la calidad de los servicios informáticos aumenta, por lo tanto, el software debe mantenerse en continuo desarrollo para satisfacer las necesidades latentes. La importancia del desarrollo de software se puede apreciar en los grandes sistemas que se actualizan constantemente.

Desde un punto de vista empresarial, el software es imprescindible para alcanzar el éxito, ya que éste es de gran ayuda para facilitar los trabajos que se realizan, en ocasiones, brinda la capacidad de poder automatizar procesos, lo que ayuda a las empresas a reducir costos y a poseer una mayor eficiencia y ventaja competitiva en el rubro.

Acklen Avenue | Hero Unit, una empresa situada en Nashville, Tennessee. Empresa especializada en el desarrollo de software de calidad cuya principal prioridad son sus clientes. La práctica profesional se realizará en el departamento de desarrollo de la empresa, con el objetivo de asistir con el diseño, desarrollo e implementación de soluciones de software, con la oportunidad de trabajar en proyectos internos de la empresa. Proyectos que realizan con distintos objetivos, siendo uno la reducción de costos, al desarrollar soluciones propias y así no hacer uso de plataformas de pago de terceros, como lo es el caso del proyecto interno Acera.

El presente informe describe las generalidades de la empresa Acklen Avenue | Hero Unit, así como el departamento donde se realizará la práctica profesional y los objetivos generales y específicos que se planean realizar.

El marco teórico contiene información sobre todas las metodologías, herramientas y tecnologías utilizadas en el desarrollo del proyecto planteado.

En la sección de desarrollo se describen todas las actividades realizadas durante la práctica profesional.

II. GENERALIDADES DE LA EMPRESA

2.1 DESCRIPCIÓN DE LA EMPRESA

Acklen Avenue | Hero Unit crea productos digitales que se adaptan a las empresas con ambición. El equipo de ágiles profesionales del desarrollo de software es un apasionado del código limpio y las experiencias de usuario sencillas. Gracias a los heroicos esfuerzos del equipo y las increíbles ideas de los clientes, Acklen Avenue | Hero Unit entrega con amor software diariamente desde Nashville y Honduras a socios de todo el mundo. En cada compromiso, se entiende tan bien el producto y los objetivos del cliente que cada miembro del equipo podría dar su discurso de ascensor en su nombre. Se consideran dueños del éxito de los clientes. (Acklen Avenue: Acerca de)

¿Qué define a Acklen Avenue | Hero Unit hoy?

Integridad, se hará lo correcto, incluso si duele, al no participar ni apoyar la deshonestidad, la corrupción o la injusticia hacia ningún grupo de personas. Se hará lo mejor para los clientes, incluso si eso significa decir adiós. Se será honesto en todo lo que se haga.

Dominio, ser mejores de lo que se fue ayer al encontrar formas de mejorar cada día y dar la bienvenida a mejores formas de hacer las cosas.

Calidad, se crearán productos de calidad, por dentro y por fuera, asumiendo el control del éxito de los clientes. Se dedicará el tiempo necesario para hacer un gran trabajo para ayudar a los clientes a obtener lo que realmente necesitan, no solo lo que dijeron que necesitaban hace 6 meses.

Agilidad, se entregarán productos valiosos, flexibles y fáciles de mantener con la mayor frecuencia posible al asociarse con los clientes para maximizar el aprendizaje y el valor del producto mediante el uso de un enfoque ágil para el desarrollo de software.

Amor, se cuidará de los compañeros de trabajo, tanto a nivel profesional como personal al mismo tiempo que se preocupará por los clientes y su éxito. (Defining Acklen Avenue)

2.2 DESCRIPCIÓN DEL DEPARTAMENTO

En el departamento de desarrollo de Acklen Avenue | Hero Unit, se trabaja de forma completamente remota en aplicaciones y soluciones de software, tanto para el uso de la empresa, como para clientes y socios de todo el mundo.

2.3 OBJETIVOS DEL PUESTO

2.3.1 OBJETIVO GENERAL

Asistir con el diseño, desarrollo e implementación de soluciones de software en el proyecto interno Acera mediante el uso de herramientas, procesos y métricas de software para que este pueda ser usado por la empresa y sus empleados.

2.3.2 OBJETIVOS ESPECÍFICOS

- Desarrollar componentes funcionales para su integración en el proyecto interno Acera.
- Asistir con la documentación y el mantenimiento de la funcionalidad del proyecto interno Acera para facilitar el entendimiento de este y garantizar que todas sus características funcionen correctamente.
- Utilizar tecnologías como React y Node.js en los desarrollos del proyecto interno Acera.

III. MARCO TEÓRICO

Primero, se debe conocer cómo trabaja el equipo de desarrollo de Acklen Avenue | Hero Unit en los proyectos de desarrollo de software que poseen, a continuación, se definirán las prácticas de las cuales hacen uso.

En Acklen Avenue | Hero Unit, se utiliza la metodología ágil para desarrollar el software funcional, pero más que una metodología, se toma como una filosofía, siguiendo distintos principios, entre los cuales se encuentran:

- La mayor prioridad es satisfacer al cliente entregando software con valor de manera temprana y continua.
- Se aceptan que los requisitos del producto cambien, incluso en etapas bastante avanzadas del proyecto. Los procesos ágiles aprovechan estos cambios para proporcionar una ventaja competitiva al cliente.
- Se entrega software funcional de manera frecuente, entre dos semanas y un mes, siempre tratando de hacer uso del periodo de tiempo más corto.
- Los responsables del negocio y equipo de desarrollo se reúnen cotidianamente durante todo el proceso del proyecto.
- Los proyectos se desarrollan entorno a individuos motivados, se debe proporcionar un ambiente adecuado para ellos.
- La principal medida de progreso es el software funcionando.
- Los procesos ágiles promueven el desarrollo sostenible. Las personas involucradas en el proceso deben ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la agilidad del proceso.
- La simplicidad es esencial para entregar un producto de calidad.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
- El equipo regularmente reflexiona sobre cómo ser más efectivos para ajustar y perfeccionar su comportamiento en consecuencia.

Estos son los 12 principios encontrados en el manifiesto ágil. (Principles behind the Agile manifesto, 2001)

En la empresa, se utilizan algunos aspectos de la metodología ágil Scrum. Scrum es un marco de referencia en el cual se abordan problemas complejos de adaptación, Scrum es ligero, simple de entender y difícil de dominar. Es un marco de proceso que se ha utilizado para gestionar el trabajo en productos complejos desde los principios de la década de 1990. Scrum no es un proceso, sino un marco en el cual se pueden emplear varios procesos y técnicas. (Scrum Guide, 2020)

Scrum no es un proceso en el que se siguen metódicamente una serie de pasos secuenciales que garanticen un producto que satisfaga a los clientes, a tiempo y dentro del presupuesto, en su lugar, Scrum es un marco de referencia para organizar y manejar trabajo. Está basado en una serie de principios, valores y prácticas que proveen la base para la implementación de los enfoques del marco de referencia. (Rubin, 2012)

Scrum se basa en procesos empíricos o en el empirismo. Esto confirma que el conocimiento viene de la experiencia adquirida con el trabajo, así como la toma de decisiones en base a lo que ya se conoce. Se emplea un enfoque iterativo con el propósito de optimizar la previsibilidad y poder controlar el riesgo. Existen tres pilares que sustentan la implementación de los procesos empíricos, estos son: adaptación, inspección y transparencia. (Scrum Guide, 2020)

El equipo de Scrum está conformado de un propietario de producto, de un Scrum Master y del equipo de desarrollo. Los equipos que trabajan con Scrum deben ser autoorganizados y multifuncionales, ya que pueden elegir la mejor manera de realizar su trabajo, en lugar de ser dirigidos por alguien externo al equipo. Y los equipos multifuncionales tienen todas las capacidades necesarias para realizar el trabajo sin depender de personas externas al equipo.

En Scrum existen distintas prácticas o eventos, el evento principal es el Sprint, todo en scrum gira en torno a él, éste es un espacio de tiempo de un mes o menos en el cual se crea un incremento de producto utilizable y potencialmente liberable. Existe la planificación del Sprint, en la cual se planifica el trabajo a realizar durante el Sprint, se escogen los artículos que se realizarán y se planea la forma de trabajarlos. Luego está el Scrum diario, que no es más que una reunión diaria del equipo en la cual se discute lo realizado durante el día, y lo que se planea realizar hasta la

siguiente reunión. Al finalizar un Sprint, se realiza una revisión de este, en la cual el equipo y los interesados en el proyecto colaboran sobre lo que se hizo en el Sprint, y se discute qué artículos podrían realizarse a continuación. Por último, se realiza una retrospectiva del Sprint, en la cual el equipo se evalúa a sí mismo, con el propósito de conocer lo que salió bien y lo que no durante el Sprint, para tratar de mejorar para el siguiente. (Scrum Guide, 2020)

Adicionalmente, la empresa usa características de otra metodología ágil llamada Kanban, así combinando características de ambas metodologías y logrando una armonía para encontrar el flujo de trabajo óptimo.

Kanban es un método para gestionar el trabajo que nació en el Sistema de Producción de Toyota, (TPS) por sus siglas en inglés. A finales de los años 40, se implementó este sistema en el cual la producción se basa en la demanda de los clientes y no en la práctica tradicional de fabricar productos e intentar venderlos en un mercado. Su propósito fundamental es minimizar los desperdicios sin afectar la producción. Su objetivo principal es crear más valor para el cliente sin generar más gastos.

La palabra Kanban viene del japonés y traducida literalmente significa tarjeta con signos. El tablero más básico de Kanban posee tres columnas: "Por hacer", "Haciendo" y "Hecho". AL aplicar este formato correctamente, sirve como una fuente de información porque demuestra dónde existen cuellos de botella en el proceso, lo que impide un flujo de trabajo continuo e interrumpido, que es la característica principal de Kanban. (Qué es Kanban: Definición, Características y Ventajas, 2020)

En el proyecto Acera se utiliza React, esta es una librería de JavaScript utilizada en el desarrollo web para la creación de elementos interactivos en sitios web. JavaScript es un lenguaje de secuencias utilizado para crear y controlar contenido web dinámico, este incluye elementos como gráficos animados, presentaciones de fotos y formularios interactivos. En los sitios web donde las cosas se mueven, se actualizan o cambian de alguna manera en la pantalla sin la necesidad de recargar la página de forma manual, es muy probable que JavaScript sea el que lo hace posible. (Tech 101: ¿Qué es React?)

Las librerías de JavaScript son colecciones de código previamente escrito que se pueden usar para tareas comunes, lo que permite evitar el proceso de codificación manual ya que esto requiere mucho tiempo. (Tech 101: ¿Qué es React?)

React se basa en un paradigma llamado programación orientada a componentes en el que cada uno es una pieza con la que el usuario puede interactuar. Estas se crean usando una sintaxis llamada Extensión de JavaScript (JSX) por sus siglas en inglés, la cual permite escribir código de Lenguaje de Marcado de Hipertexto (HTML) dentro de objetos JavaScript. Estos componentes son completamente reutilizables y se combinan para crear componentes más grandes hasta configurar una red completa. Esta es la forma de combinar HTML con toda la funcionalidad de JavaScript. (¿Qué es React y para qué sirve?, 2020)

React provee muchas ventajas frente a la forma clásica de desarrollar una web, las facilidades de desarrollo unido al rendimiento, la flexibilidad y organización del código la convierten en una gran opción para el desarrollo. Una de las razones por las cuales esto es posible es el uso del Modelo de Objeto de Documento (DOM) virtual, React puede generar el DOM virtual de forma dinámica, haciendo los cambios en memoria y comparándolos con los actuales, de esta forma se evita renderizar toda la página cuando ocurra un cambio, simplemente se aplica el cambio al componente al que corresponda la actualización. De esta forma se mejora la experiencia de usuario y se tiene un excelente rendimiento y una gran fluidez. (¿Qué es React y para qué sirve?, 2020)

En el proyecto se utilizó Firebase, esta es una plataforma en la nube para la elaboración de aplicaciones web y móviles, está disponible para distintas plataformas, por lo cual es más rápido trabajar en el desarrollo. Comenzó como una base de datos en tiempo real, pero se añadieron más funciones a lo largo del tiempo. (Firebase: ¿qué es y para qué sirve?, 2020)

La función principal de Firebase es hacer más sencillo el proceso de crear aplicaciones web y móviles, facilitando su desarrollo y acelerando la producción del trabajo sin necesidad de renunciar a la calidad requerida. Las herramientas que posee son muy variadas y fáciles de usar, esto porque su agrupación simplifica las tareas de gestión a una misma plataforma. Las finalidades de estas herramientas se pueden dividir en 4 grupos: desarrollo, crecimiento, monetización y análisis. (Firebase: ¿qué es y para qué sirve?, 2020)

El grupo de desarrollo incluye los servicios necesarios para crear un proyecto de aplicación móvil o web, haciendo que el proceso sea más rápido, ya que muchas de las actividades que se necesitan realizar en el proceso de desarrollo son dejadas en manos de Firebase. Las herramientas más destacadas e importantes que posee son las bases de datos en tiempo real. Estas se alojan en la nube y no pertenecen al Lenguaje de Consulta Estructurada (SQL) por sus siglas en inglés, y almacenan los datos como Notación de Objeto de JavaScript (JSON). Estas permiten alojar y disponer de los datos e información de la aplicación en tiempo real, de esta forma se mantienen los datos actualizados en todo momento, aunque el usuario no realice ninguna acción dentro de la aplicación. De igual forma Firebase envía automáticamente eventos a las aplicaciones cuando los datos cambian, almacenando los datos actualizados en el disco. De esta forma, aunque un usuario no posea conexión, sus datos estarán disponibles para el resto de los usuarios, y estos cambios se sincronizan cuando la conexión sea restablecida. (Firebase: ¿qué es y para qué sirve?, 2020)

Firebase también ofrece la autenticación de usuarios, su uso es necesario en la mayoría de los casos en las aplicaciones web y móviles. Este proceso permite el registro mediante el ingreso de credenciales tales como correo electrónico y contraseña, así como utilizando perfiles de otras plataformas externas, algo muy útil para los usuarios por la diversidad de opciones que ofrece. De igual manera, Firebase puede guardar en la nube estos datos de inicio de sesión con total seguridad, esto evita que un usuario deba identificarse cada vez que ingrese a la aplicación.

De igual manera, ofrece un servidor para alojar las aplicaciones de manera rápida y fácil con un alojamiento estático y seguro. Este proporciona certificados de seguridad de Capa de Conexión Segura (SSL) y de Protocolo de Transferencia de Hipertexto versión 2 (HTTP2) de manera gratuita y automática para cada dominio, otorgando la seguridad en la navegación (Firebase: ¿qué es y para qué sirve?, 2020)

En recopilación, Firebase presenta numerosos beneficios para los desarrolladores que hagan uso de ella, entre los cuales se encuentran:

- Altamente recomendable para aplicaciones que necesiten compartir datos en tiempo real.

- Ofrece numerosos documentos y tutoriales introductorios e informativos para que su implementación sea mucho más fácil.
- Posee soporte gratuito vía correo electrónico, esto sin importar si el desarrollador utiliza la versión gratuita o de pago.
- Ofrece escalabilidad porque los inicios son gratuitos, pero permite adaptarse a las necesidades de la aplicación con distintos planes de pago. (Firebase: ¿qué es y para qué sirve?, 2020)

Una de las funciones más importantes del proyecto es que permite la visualización de una representación gráfica de la ruta que se está siguiendo, en la cual se muestran las medallas necesarias para completarla, así como los requerimientos de cada nodo que contenga medallas. Para esto, se utiliza D3. (¿Qué es D3.js?, 2020)

D3 o también conocida como D3.js significa documentos basados en datos. Es una biblioteca de JavaScript de código abierto que crea visualizaciones de datos interactivas en un navegador mediante el uso de Gráficos Vectoriales Escalables (SVG), Lenguaje de Marcas de Hipertexto (HTML) y Hojas de Estilo de Cascada (CSS). Esta librería le permite vincular datos arbitrarios a un DOM para luego poder aplicar transformaciones basadas en datos al documento. (¿Qué es D3.js?, 2020)

Algunas características de la librería D3 son:

- Usa estándares web modernos para crear visualizaciones de datos, por lo tanto, la curva de aprendizaje es bastante baja.
- Genera dinámicamente elementos a partir de datos y les aplica estilos. Un elemento puede ser desde una tabla, un gráfico y hasta otro elemento HTML.
- Posee visualizaciones personalizadas que brindan un control completo sobre sus funciones de visualización.
- Proporciona soporte para animaciones suaves, posee una función de transición que permite interpolar lógicamente los valores en los datos y encontrar estados intermitentes.
- Al ser liviano y funcionar directamente con los estándares web, es muy rápido y funciona bien con grandes conjuntos de datos.

- Es de código abierto, por lo que se puede editar el código fuente y agregar funciones propias del desarrollador. (¿Qué es D3.js?, 2020)

Para realizar ciertas pruebas, se utilizó Postman, esta es una herramienta dirigida a desarrolladores que permite hacer peticiones HTTP a cualquier Interfaz de Programación de Aplicaciones (API). Es muy útil para realizar pruebas porque con ella se puede comprobar que las aplicaciones estén funcionando correctamente. En general, Postman es muy útil para cualquier escenario en el cual se deba trabajar con una API. (¿Qué es Postman?, 2020)

Con esta herramienta, se pueden probar, consumir y depurar las API, además de monitorizarlas, escribir pruebas automatizadas, documentarlas, simularlas, etc. Es una de las herramientas más favorable para equipos con poca experiencia en programación por la facilidad que brinda. Postman genera una documentación bastante interesante y atractiva, con ejemplos de código que ayudan a comprender el funcionamiento de una API en particular. (¿Qué es Postman y para qué sirve?, 2019)

Algo muy importante para los desarrolladores y para el equipo del proyecto en general es la capacidad de detectar los errores de la aplicación de manera eficiente para poder resolverlos lo más rápido posible, para lograr esto el equipo utiliza Rollbar. Esta es una solución de monitoreo y seguimiento de errores basada en la nube que sirve para organizaciones de todos los tamaños. Soporta múltiples lenguajes de programación y marcos de trabajo como JavaScript, Node.js, etc. Es posible implementar esta solución en instalaciones de usuario. (Rollbar Software, s.f.)

Rollbar es capaz de proporcionar agrupación automática de errores basada en la causa de un error y también brinda a los usuarios la habilidad de personalizar las reglas de agrupación para poseer una lista de errores más específica. También ayuda a los usuarios a filtrar notificaciones y establecer prioridades en los tipos de error, así como la capacidad de etiquetar los problemas como activos, resueltos o silenciados. También permite a los usuarios realizar un seguimiento de las implementaciones, ver la información en forma de gráficos, administrar la línea de tiempo y mantener un historial detallado. (Rollbar Software, s.f.)

Entre las principales ventajas que brinda el uso de Rollbar se encuentran:

- Puede analizar, diagnosticar y arreglar errores.

- Su configuración es sumamente fácil y brinda resultados instantáneos.
- Posee precios asequibles para problemas costosos.
- Permite a los desarrolladores encontrar y corregir errores críticos rápidamente.
- Es totalmente compatible con los estándares de la industria. (Rollbar - Una Herramienta para detectar, diagnosticar y eliminar errores web , s.f.)

IV. DESARROLLO

4.1 DESCRIPCIÓN DEL TRABAJO REALIZADO

4.1.1 INDUCCIÓN A LA EMPRESA

En la Ilustración 1 se puede observar el proceso de inducción que se sigue dentro de la empresa.

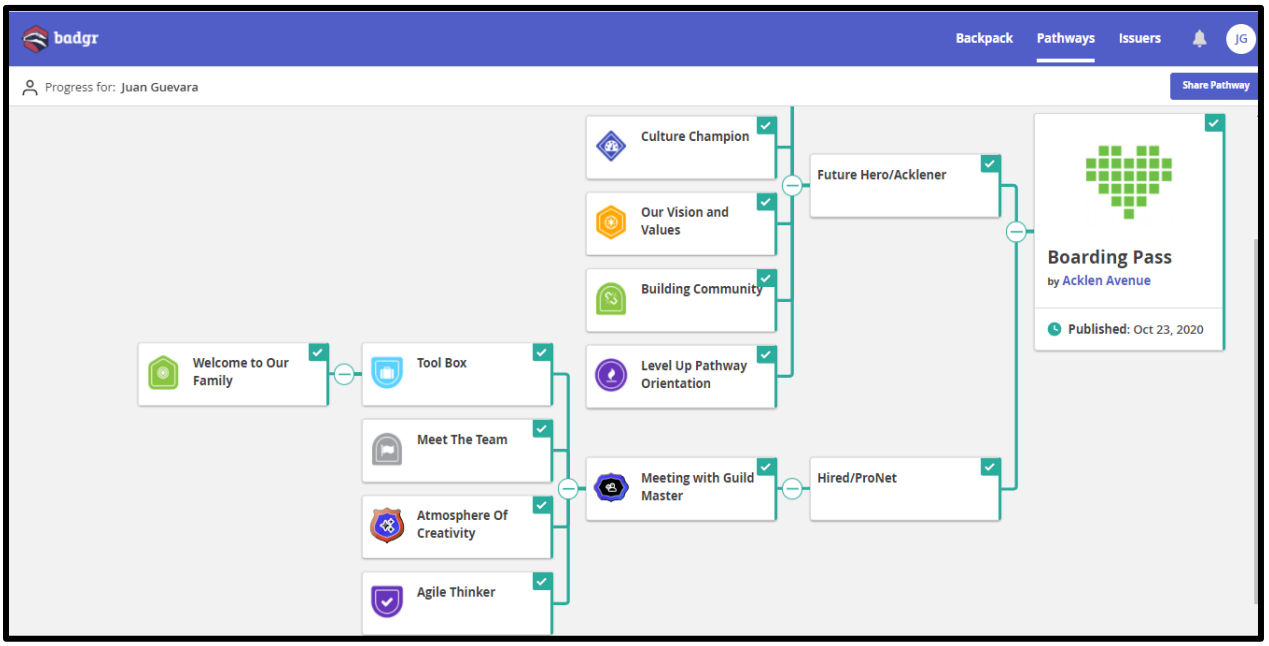


Ilustración 1 - Ruta de Inducción de la empresa

Fuente: (Badgr, s.f.)

Dentro de la empresa, se maneja un sistema de medallas, estas medallas se les otorgan a los empleados por distintas razones: adquirir un nuevo conocimiento o habilidad, cumplir con los objetivos descritos en la medalla, contribuir a la comunidad, presentar un proyecto, una evaluación con un experto en la materia, enseñar sobre un tema, o asistir a un evento patrocinado por Acklen Avenue | Hero Unit.

Como primer paso, para llegar a conocer la empresa y la forma en la que trabajan, se realizó la ruta de inducción, que consiste en una serie de medallas que poseen diversos objetivos, tales como:

- Conocer las herramientas que usa la empresa, configurarlas y aprender a usarlas. Herramientas como: el calendario de Google, Github, Trello, Peeplo y Slack.
- Conocer a ciertas personas que trabajan en la empresa, como los maestros de gremio, que son los encargados de gestionar las actividades y los recursos humanos de cada departamento.
- Realizar evaluaciones con expertos en la materia, se realizaron 2 evaluaciones: La primera sobre el cómo crear una atmosfera creativa y la segunda sobre metodologías ágiles, en específico Scrum.
- Agendar una reunión con el maestro del gremio designado, en este caso, del departamento de desarrollo, para puntualizar futuras expectativas.
- Hacer uso del material informativo que proporciona la empresa, material que muestra sus valores, sus prácticas y formas de crecer dentro de la empresa.
- Realizar una evaluación del material mencionado.

4.1.2 Integración al Proyecto Acera

Acera es un proyecto interno de la empresa, cuyo objetivo es replicar la funcionalidad de la plataforma Badgr, plataforma de terceros usada para mantener las diversas rutas que otorgan las medallas utilizadas por la empresa. Con la funcionalidad agregada de poder ser una plataforma común en la cual mantener las medallas de otras plataformas de terceros que ofrezcan medallas también.

Se realizó una reunión con el equipo del proyecto para la discusión de las herramientas que se utilizan, así como el estado actual del proyecto. Adicionalmente, se realizó una reunión más para la explicación y configuración de credenciales y permisos en las distintas herramientas necesarias para trabajar en el proyecto. Estas herramientas son: la base de datos en tiempo real de Firebase, el tablero de Trello con las actividades del trabajo a realizar, el repositorio en GitLab que contiene el código fuente del proyecto. En esta reunión se observó el funcionamiento del proyecto, las funciones utilizadas en Firebase, el código fuente que ejecutan los distintos componentes y el recorrido que un usuario puede tomar dentro de la aplicación.

En la Ilustración 2 se observa un ejemplo de una ruta de medallas dentro de la aplicación.

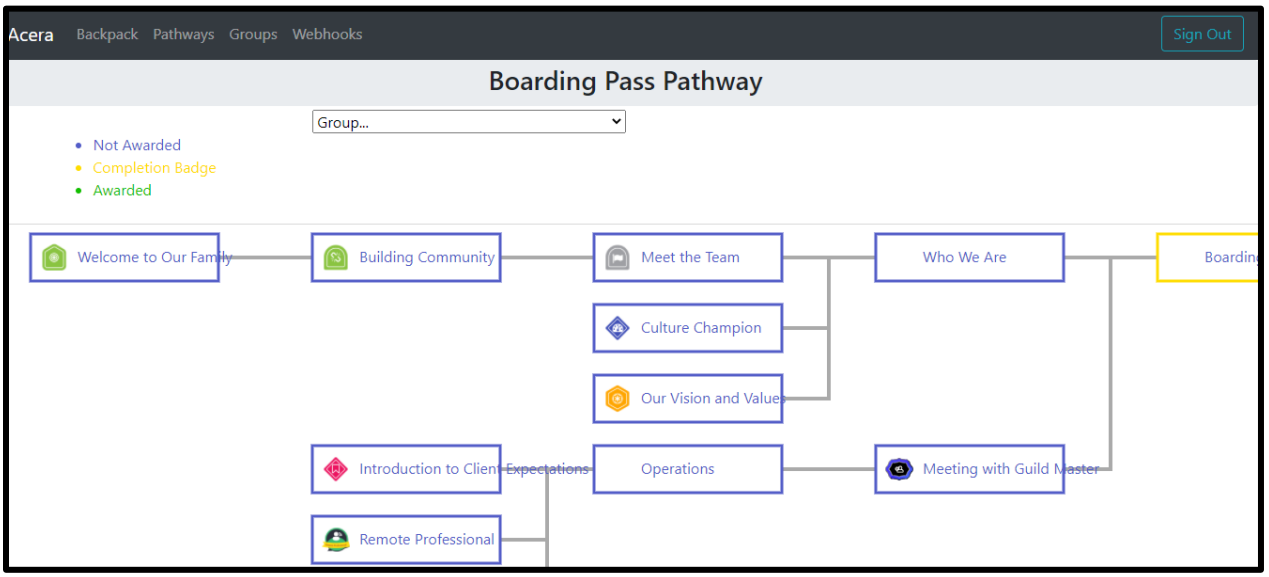


Ilustración 2 - Ejemplo de Ruta de Medallas en la Aplicación

Fuente: (Elaboración Propia)

Como se puede observar, es muy similar al diagrama mostrado en la Ilustración 1. Esto porque el proyecto Acera utiliza la API de Badgr para obtener los objetos conocidos como nodos, estos contienen las medallas necesarias para completar ese nodo, así como sus requerimientos.

4.1.3 Desarrollo de Componente de Otorgamiento Automático

Dentro de la aplicación Acera, un nodo puede contener una medalla de finalización o una medalla requerida. La primera se utiliza para indicar que ese nodo ha sido completado, y esto sucede cuando todas las medallas requeridas de las cuales depende sean completadas.

En la Ilustración 3 se puede observar un ejemplo del caso descrito.

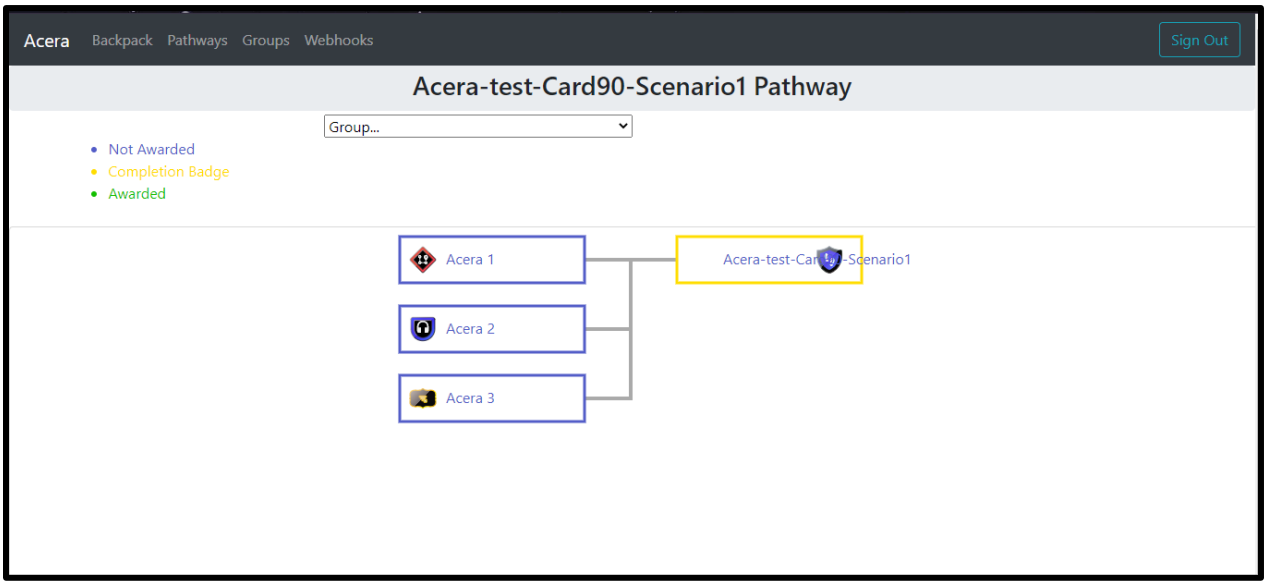


Ilustración 3 - Ejemplo de Nodos de Finalización y Requeridos

Fuente: (Elaboración Propia)

En este caso, los 3 nodos que se encuentran en la parte izquierda del diagrama son nodos que contienen una medalla requerida, esto se representa colocando el ícono de la medalla a la izquierda del nodo. Y en la parte derecha del diagrama se encuentra el nodo de nivel superior que posee una medalla de finalización, representada por mostrar el ícono a la izquierda del nodo y tener un borde color amarillo.

Por lo general las medallas requeridas necesitan ser reclamadas, al hacer esto, se envía un correo electrónico a la persona responsable de entregar la medalla, para revisión de los requisitos de esta. El propósito de esta actividad era evitar que el usuario tuviera que reclamar la medalla de finalización, cuyos requisitos son completar las medallas requeridas de las cuales depende. Por esto se codificó una función que revisa el estado de las medallas requeridas de una ruta, verificando que se cumplan los requisitos de la medalla de finalización correspondiente y así otorgarla al usuario de forma automática.

En la Ilustración 4 se observa el código fuente que realiza la revisión de las medallas.

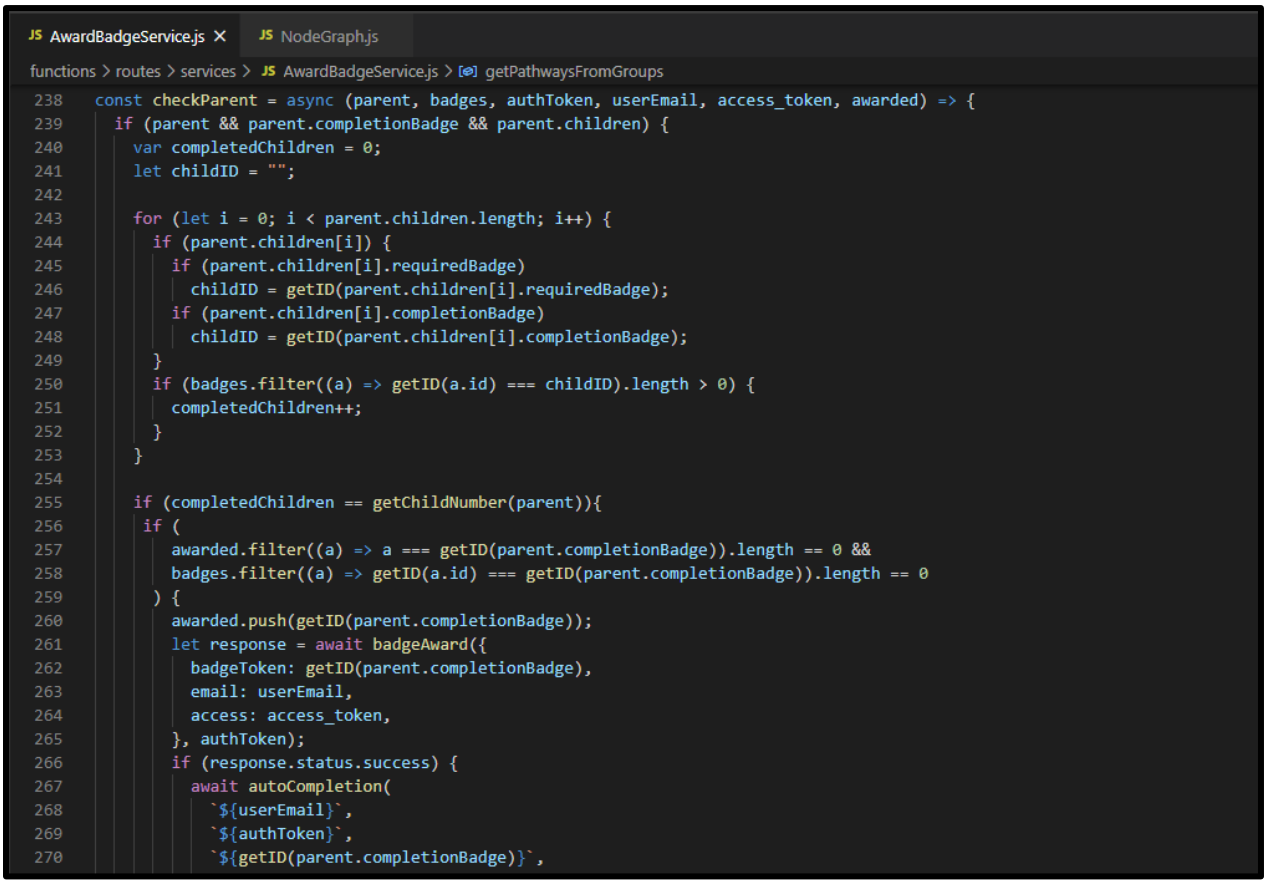


Ilustración 4 - Función de Otorgamiento Automático de Medallas

Fuente: (Elaboración Propia)

Esta funcionalidad se encuentra en las funciones de Firebase, y se ejecuta cada cierto intervalo de tiempo, se realizaron pruebas sobre la misma con distintas cantidades de tiempo, comenzando en 5 minutos. Como resultado de estas pruebas, se obtuvo un error en el cual las medallas se asignaban múltiples veces al mismo usuario. Se descubrió que el error se daba ya que la mochila que contiene las medallas de Badgr del usuario se borraba de la base de datos en tiempo real debido a un protocolo de seguridad que evitaba accederla de manera continua, al hacer la correspondiente validación, el problema se solucionó.

Adicionalmente se agregó una regla nueva en las características de los nodos en la cual a un nodo se le permite poseer hasta 2 medallas, haciendo que puedan tener tanto medalla de finalización como requerida.

4.1.4 Sobrescribir las Características de una Ruta Hija

Es posible anidar rutas dentro de la aplicación, para que en el caso de que una ruta contenga las mismas medallas que otra, pero con medallas adicionales, sólo se deba incluir una ruta dentro de otra.

En la Ilustración 5 se observa un ejemplo del escenario descrito.

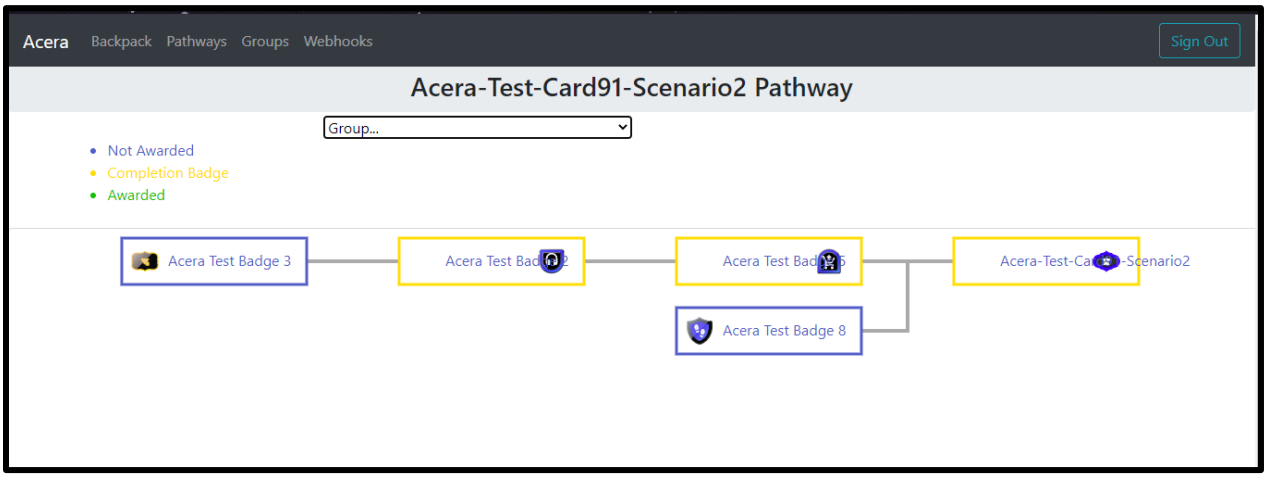


Ilustración 5 - Escenario de Rutas Anidadas

Fuente: (Elaboración Propia)

En este caso, el nodo llamado “Acera Test Badge 5” es el nodo de alto nivel de una ruta distinta, por lo que esta ruta sólo contiene 2 hijos, el nodo requerido inferior “Acera Test Badge 8” y una ruta anidada. La tarea consistió en sobrescribir las características de la ruta anidada con las del nodo de alto nivel de la ruta actual en momento de ejecución, es decir, se cambió su título y su lista de nodos hijos. Esto con el propósito de optimizar el proceso de otorgamiento automático de medallas.

En la Ilustración 6 se observa la función en el código fuente que logra este comportamiento.

```
function modifyaux(pathway, pathways){
  var newPathway = pathway;
  if(pathway){
    var newChildren = []
    var oldChildren = []
    if(pathway.pathwayURL && pathway.pathwayURL!==""){
      var childPathway = pathways.filter(path => getID(path.completionBadge) === getID(pathway.pathwayURL))
      if(childPathway.length > 0 && childPathway[0].children){
        newPathway.title = (childPathway[0].title).split('-').join(' ')
        newPathway.completionBadge = childPathway[0].completionBadge ? childPathway[0].completionBadge : ""
        newPathway.requiredBadge = childPathway[0].requiredBadge ? childPathway[0].requiredBadge : ""
        if(!newPathway.children){
          newPathway.children = childPathway[0].children;
        } else {
          oldChildren = newPathway.children;
          newPathway.children = childPathway[0].children;
          newPathway = addChildrenAtDeep(oldChildren, newPathway)
        }
      }
    }

    if(newPathway.children){
      for(let index = 0; index < newPathway.children.length; index++){
        var newChild = modifyaux(newPathway.children[index], pathways)
        if(newChild)
          newChildren.push(newChild)
      }
      newPathway.children = newChildren
    }
  }

  return newPathway
}
```

Ilustración 6 - Función que Sobrescribe las Características de una Ruta Anidada

Fuente: (Elaboración Propia)

4.1.5 Modificar la Representación Gráfica del Diagrama

Con la creación de la nueva regla en la que un nodo puede poseer 2 medallas, se tuvo que realizar modificaciones al componente encargado de crear el diagrama para su visualización, haciendo uso de la biblioteca de JavaScript D3.js.

La Ilustración 7 muestra la visualización gráfica actualizada del diagrama.

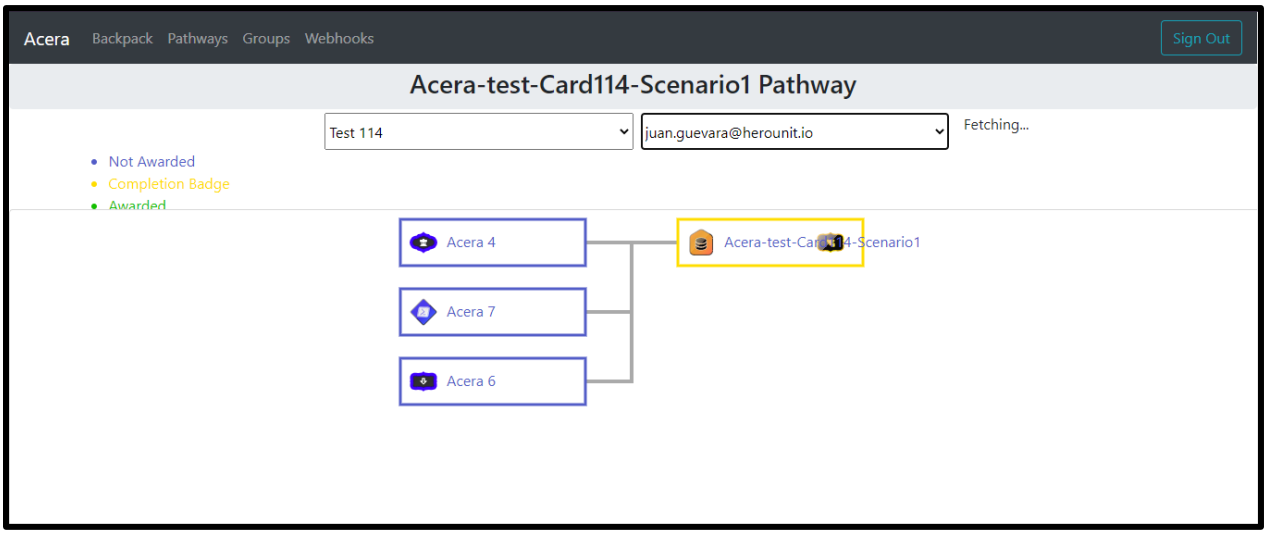


Ilustración 7 - Visualización Gráfica de Nodos con 2 Medallas

Fuente: (Elaboración Propia)

Con el cambio realizado se obtuvo el resultado mostrado, en el cual un nodo contiene ambas medallas, teniendo la medalla requerida a la izquierda del nodo y la medalla de finalización a la derecha. De igual manera el borde del nodo es de color amarillo para representar que es un nodo de finalización.

La implementación de la regla en la que un nodo puede tener 2 medallas produjo problemas en las pruebas de integración, ya que causó que el mismo componente para visualizar el diagrama tuviera un fallo en el momento de mostrar qué medallas ya han sido asignadas. Se añadió la condición para el caso en el que el nodo contuviera ambas medallas, tomando como punto de referencia la medalla de finalización, ya que para que la misma sea completada deben completarse todas las requeridas que tenga como hijas.

En la Ilustración 8 se puede apreciar el comportamiento que muestra cuando una medalla ya fue asignada.

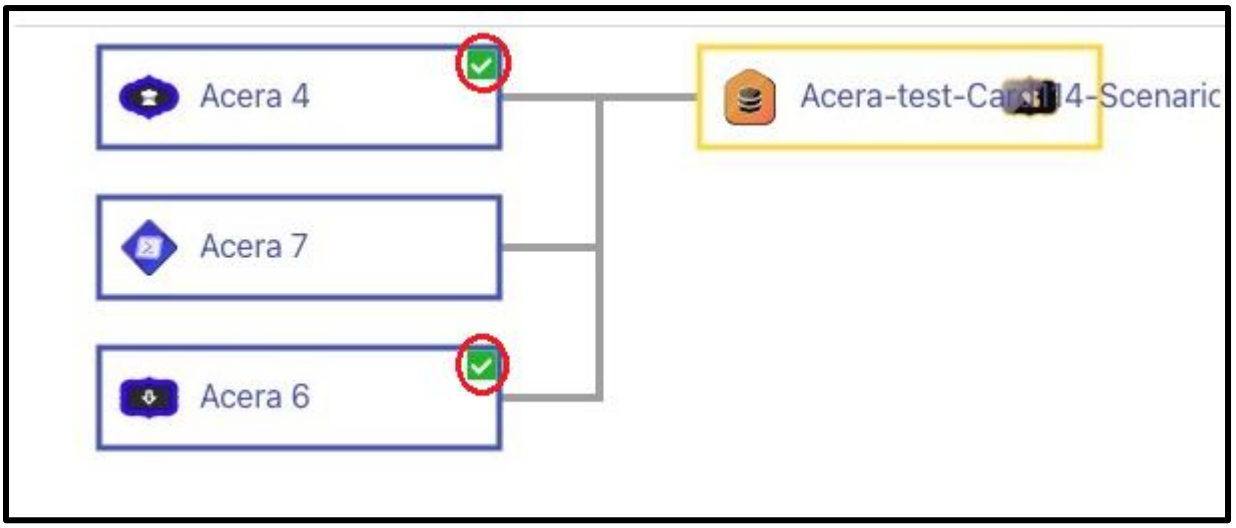


Ilustración 8 - Visualización de Medallas Asignadas

Fuente: (Elaboración Propia)

Adicionalmente, se solicitó otro requerimiento que se produjo por la nueva regla en la que un nodo puede contener 2 medallas, con el propósito de mostrar el progreso, así como sucede en los demás nodos. Para esto se utilizó la misma funcionalidad que muestra el símbolo de verificación verde en la esquina de los nodos, pero en el caso específico en el que contenga una medalla requerida y una de finalización.

La Ilustración 9 muestra el cambio realizado a los nodos con ambas medallas.

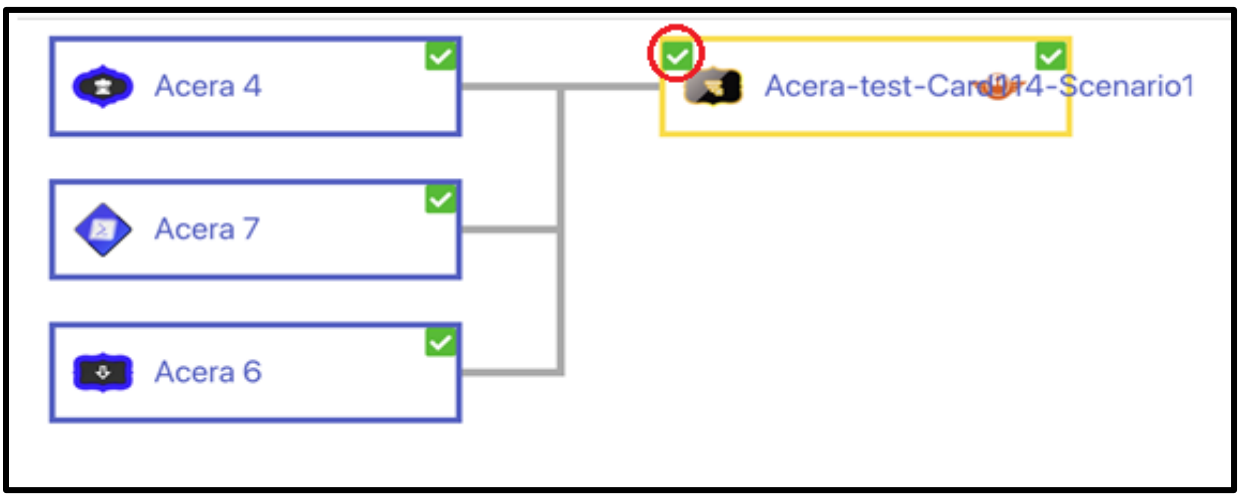


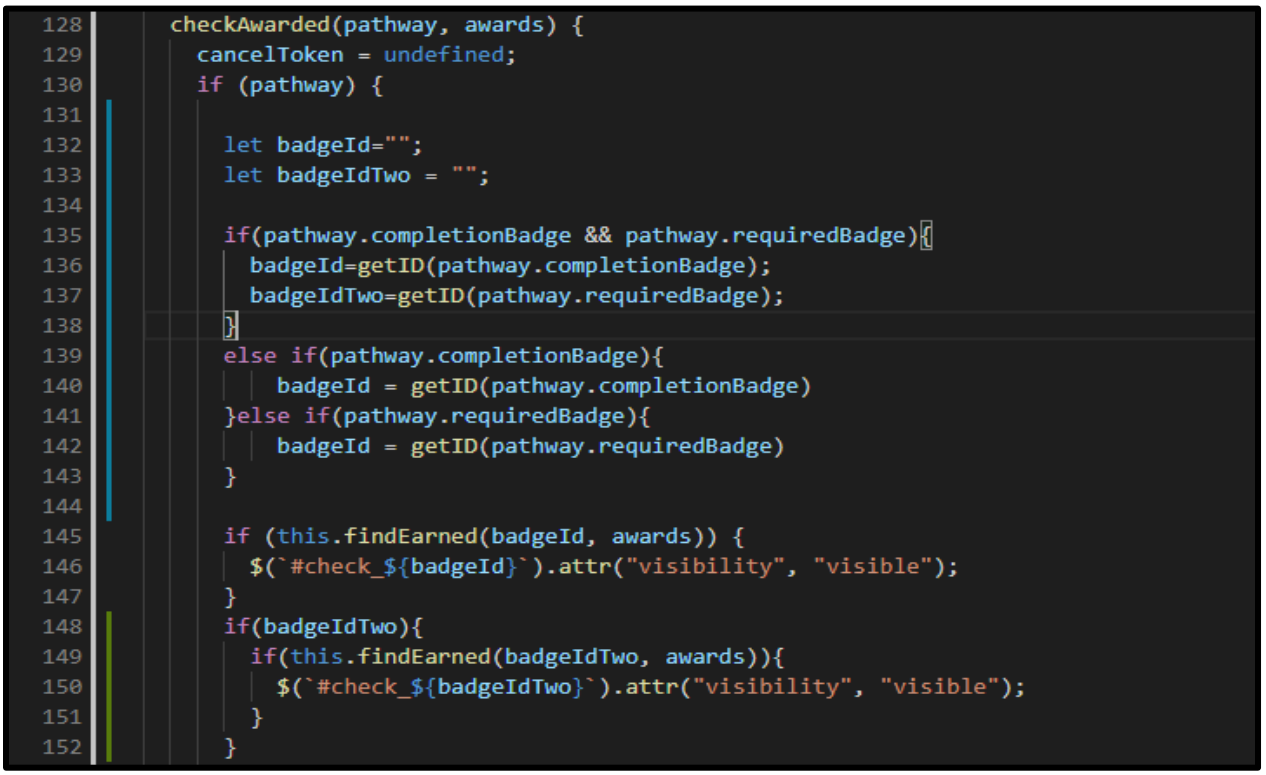
Ilustración 9 - Visualización Gráfica de Nodos con 2 Medallas versión 2

Fuente: (Elaboración Propia)

Se puede observar el cambio realizado, en el cual se agregó un símbolo de verificación verde en la esquina superior izquierda del nodo que contiene ambas medallas, representando que la medalla requerida que se encuentra en la parte izquierda del nodo ha sido otorgada, de igual manera manteniendo el símbolo en la esquina superior derecha para representar que la medalla de finalización del nodo fue otorgada.

Para cumplir con este requerimiento se realizaron modificaciones al código fuente del componente, en su mayoría duplicando las variables existentes y colocando las validaciones necesarias para el caso de un nodo con ambas medallas en el cual se debe revisar cada una para verificar si ya ha sido otorgada.

La Ilustración 10 muestra una porción del código fuente modificado.



```
128     checkAwarded(pathway, awards) {
129       cancelToken = undefined;
130       if (pathway) {
131
132         let badgeId="";
133         let badgeIdTwo = "";
134
135         if(pathway.completionBadge && pathway.requiredBadge){
136           badgeId=getID(pathway.completionBadge);
137           badgeIdTwo=getID(pathway.requiredBadge);
138         }
139         else if(pathway.completionBadge){
140             badgeId = getID(pathway.completionBadge)
141         }else if(pathway.requiredBadge){
142             badgeId = getID(pathway.requiredBadge)
143         }
144
145         if (this.findEarned(badgeId, awards)) {
146           $(`#check_${badgeId}`).attr("visibility", "visible");
147         }
148         if(badgeIdTwo){
149           if(this.findEarned(badgeIdTwo, awards)){
150             $(`#check_${badgeIdTwo}`).attr("visibility", "visible");
151           }
152         }
```

Ilustración 10 - Función para Revisar el Estado de los Nodos

Fuente: (Elaboración Propia)

4.1.6 PRUEBAS A LA API DE BADGR

La aplicación de Acera se conecta a la API de Badgr para recuperar información de las medallas y las mochilas de los usuarios. Se presentaron errores en los cuales la aplicación no cargaba correctamente la información por lo que se procedió a realizar pruebas directas a la API haciendo uso de la herramienta Postman.

La Ilustración 11 muestra una prueba realizada con la herramienta Postman.

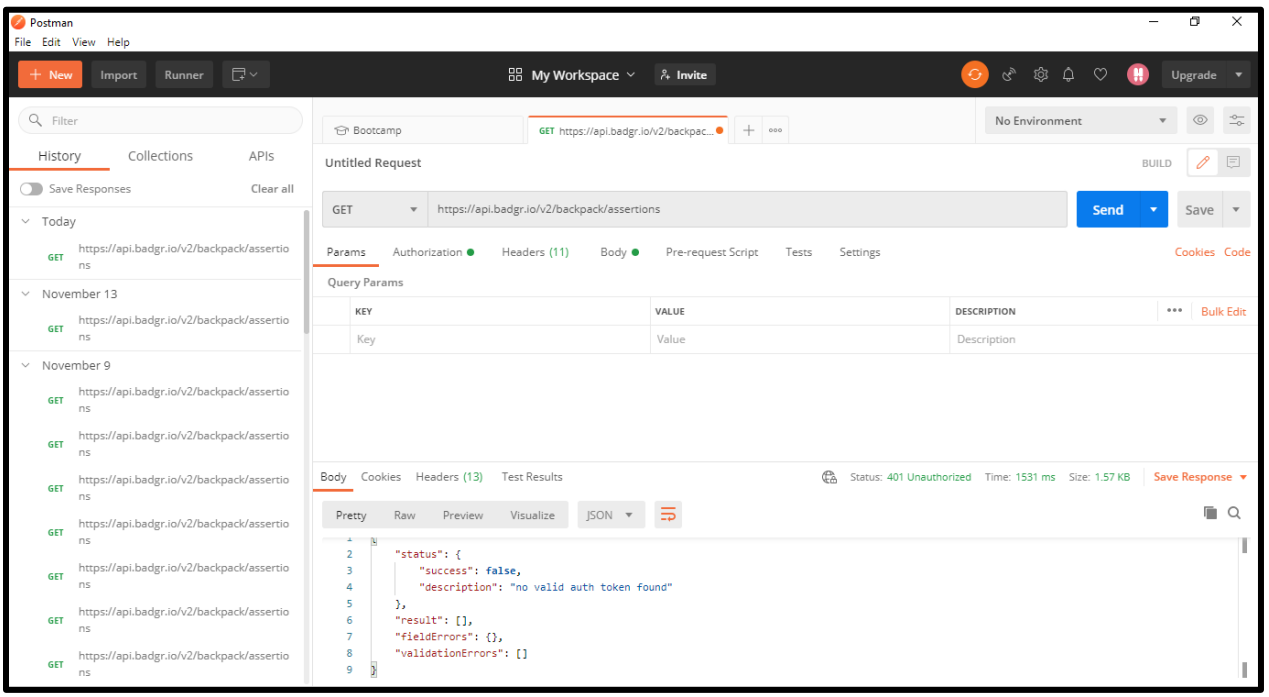


Ilustración 11 - Pruebas a la API de Bagdr

Fuente: (Postman, s.f.)

Mediante estas pruebas se pudo comprobar que la API de Badgr no estaba funcionando correctamente y que los errores no eran por parte de la aplicación.

4.1.7 CAMBIOS EN EL GUION DE DESPLIEGUE

La aplicación consta de distintos ambientes, entre ellos están el de desarrollo, puesta en escena y producción. Cada nuevo desarrollo debe seguir el flujo de trabajo establecido y necesita pasar por los primeros 2 ambientes antes de llegar al ambiente de producción, que sería el que se

muestra al público. En la etapa de desarrollo se necesitan realizar despliegues constantes hacia el ambiente de desarrollo, para realizar pruebas y verificar que los cambios generados funcionen correctamente. Para esto, se debían realizar al menos 5 comandos de forma individual, lo que tendía a ser tedioso y más propenso a errores humanos por la posibilidad de escribir incorrectamente un comando.

Para solucionar este problema, se creó un guion para realizar el conjunto de comandos en una simple línea, estableciendo las variables de entorno necesarias y ejecutando los comandos que se encuentren en el guion, esto le brinda una gran escalabilidad, ya que de ser necesario un nuevo comando para el despliegue, basta con agregarlo al final del guion.

En la Ilustración 12 se puede observar la estructura del guion de despliegue.

Ilustración 12 - Guion de Despliegue

Fuente: (Elaboración Propia)

4.1.8 Cambio de Método de Recuperación de Objeto llamado Backpack

Al ingresar a la aplicación Acera se ejecuta un método que se conecta al API de Badgr y recupera la información de la mochila que contiene las medallas del usuario, se presentó un error que al cargar la página enviaba demasiadas solicitudes al API causando que esta contestara con una respuesta fallida. Mediante pruebas se encontró la fuente del error, esta era la forma de generar las peticiones ya que se usaba un ciclo iterativo, el cual enviaba todas las peticiones al mismo tiempo.

Se decidió implementar una función recursiva con la lógica antes contenida en el ciclo iterativo ya que esta se puede ejecutar de manera asíncrona por lo que cada petición espera a que la

anterior termine para poder ejecutarse, así evitando el problema de enviar demasiadas peticiones concurrentes al API.

En la Ilustración 13 se puede observar el fragmento de código que contiene la función recursiva.

```
126   async refreshToken(attempt, email, token) {
127     if(attempt < 2){
128       const new_access = await axios.post(`/users/refresh`, {
129         token: token.data.refresh_token,
130       });
131       const [error, infoError] = saveBackpackToken(
132         email.email,
133         new_access.data,
134         this.state.email
135       );
136
137       if (error) {
138         this.setState({attemptMsg: true})
139         attempt++;
140         console.error("Error", infoError, attempt);
141       } else {
142         this.setState({ isAuthenticated: true });
143         const badges = await axios.post(`/users/backpack`, {
144           token: new_access.data.access_token,
145         });
146         if (badges && badges !== {}) {
147           this.saveBackpackLocalStorage(badges);
148           return;
149         }
150       }
151       attempt++
152       this.setState({attempt:attempt})
153       await this.refreshToken(attempt, email, new_access.data)
154     }
155   }
```

Ilustración 13 - Método de Recuperación de Mochila

Fuente: (Elaboración Propia)

4.1.9 Integración con Rollbar

Con el propósito de poseer una documentación de errores accesible y fácil de interpretar se solicitó integrar el proyecto con una herramienta de seguimiento de errores. Entre las opciones se optó por utilizar Rollbar por su extensa documentación y facilidad de implementación para los marcos de trabajo utilizados en el proyecto.

Para su implementación, fue necesario realizar la instalación de la dependencia de Rollbar y crear un archivo que contenga la configuración y credenciales necesarias. Este archivo se importó en cada componente en el que fue necesario y así hacer uso de las funciones para registrar los errores. Esto le da escalabilidad a esta funcionalidad ya que el archivo simplemente se debe importar en futuros componentes.

En la Ilustración 14 se muestra el archivo creado con la configuración de Rollbar.

functions > JS rollbar.js > ...

```
1   const Rollbar = require('rollbar');
2   const envs = require('./env.json');
3
4   const ROLLBAR_USE = envs.service.rollbar_use;
5   const ROLLBAR_ACCESS_TOKEN = envs.service.rollbar_accessToken;
6   const ROLLBAR_CAPTURE_UNCAUGHT = envs.service.rollbar_captureUncaught;
7   const ROLLBAR_CAPTURE_UNHANDLED = envs.service.rollbar_captureUnhandledRejections;
8
9   var rollbar = new Rollbar({
10    accessToken: ROLLBAR_ACCESS_TOKEN,
11    captureUncaught: ROLLBAR_CAPTURE_UNCAUGHT,
12    captureUnhandledRejections: ROLLBAR_CAPTURE_UNHANDLED
13  });
14
15  //Use a function in case wee need to add more logic for check this process
    Complexity is 4 Everything is cool!
16  const isActivated = () => {
17    if(ROLLBAR_USE){
18      let message = "Rollbar activated succesfully"
19      rollbar.log(message)
20      console.log(message)
21      return true
22    }
23    console.log("Rollbar it's not active")
24    return false
25  }
```

Ilustración 14 - Configuración de Rollbar

Fuente: (Elaboración Propia)

Rollbar contiene varias categorías de errores por los cuales puede agrupar por defecto, estos son: un error común, una advertencia, un registro informacional, un registro de depuración y un error crítico. Estos errores se pueden filtrar de varias maneras y se pueden visualizar en forma de

gráficas o listas. Para la primera implementación se colocaron todos los registros como errores comunes.

En la Ilustración 15 se observan los primeros registros realizados por Rollbar en la etapa de pruebas.

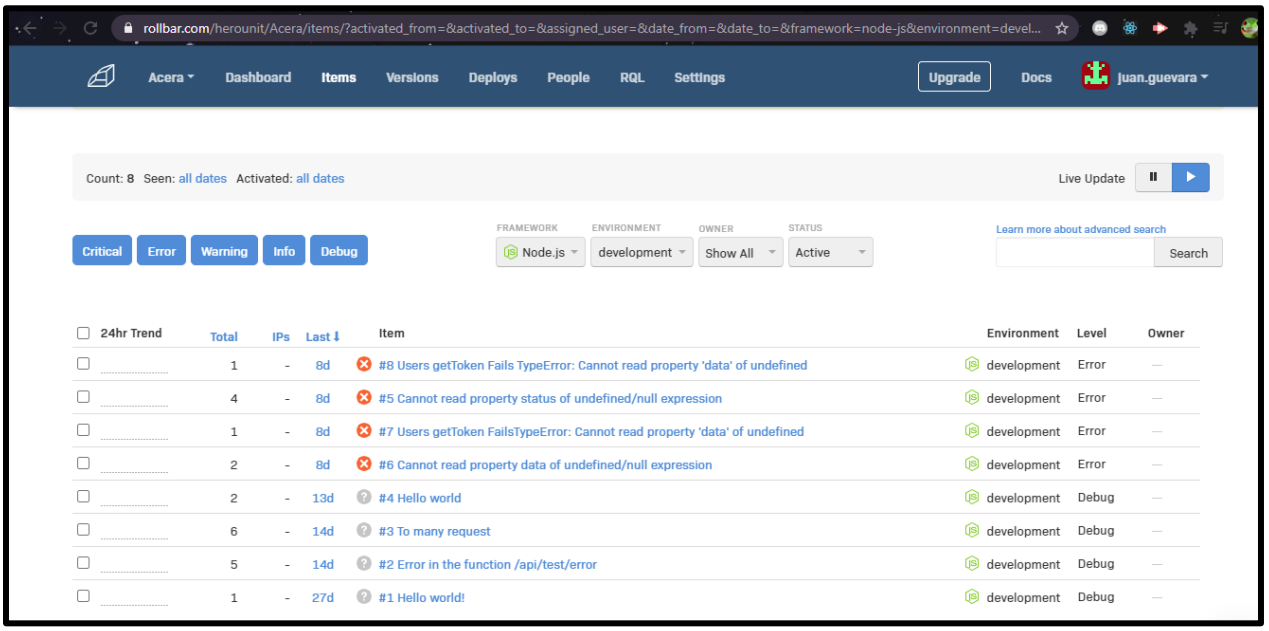


Ilustración 15 - Registros en página de Rollbar

Fuente: (Rollbar, s.f.)

4.1.10 Carga de Progreso de Almacenamiento Local

Al ingresar a la aplicación Acera, los usuarios normales y administradores pueden ingresar a la sección de rutas, donde se encuentra el conjunto de rutas al que pertenecen. Estas rutas son conformadas por distintos nodos interconectados, cada uno de ellos puede contener hasta 2 medallas y los usuarios pueden visualizar su progreso mediante símbolos de verificación encontrados en las esquinas de los nodos.

Con el propósito de brindar una mejor experiencia a los usuarios, se solicitó guardar la información del objeto backpack que contiene las medallas del usuario en el almacenamiento local del navegador, esto para que al momento de cambiar de una ruta a otra no se deba realizar

una petición al API de Badgr solicitando toda la información del usuario, lo cual suele tardar unos segundos en los que el usuario debe esperar para poder visualizar su progreso.

Por este motivo se implementó el uso del almacenamiento local del navegador, se guardaron los datos de las medallas obtenidas para cargar el progreso rápidamente cuando un usuario entre a una ruta, así mismo, se realizó una petición al API de Badgr en el fondo, para actualizar los datos del usuario, ya que se puede dar el caso en el cual el usuario obtenga una medalla y los datos almacenados en el navegador deban ser actualizados.

La Ilustración 16 muestra un fragmento del código fuente que genera este comportamiento.

```
Complexity is 5 Everything is cool!
276  getUserToken(email) {
277    this.cleanError()
278    let badgesStored = getLocalStorage("user_progress");
279    if (email !== "0") {
280      if (
281        this.readCookies("user_email") === window.localStorage.getItem("progress_owner") &&
282        badgesStored
283      ) {
284        this.loadBackpackIds(badgesStored);
285        this.getUserProgress(email)
286      } else {
287        this.getUserProgress(email)
288      }
289    }
290  }
291
292
```

Ilustración 16 - Carga de Datos de Almacenamiento Local

Fuente: (Elaboración Propia)

4.1.11 Recuperación de Rutas desde Firebase

En el proyecto interno Acera, las rutas que contienen los nodos y sus respectivas medallas se encontraban inicialmente almacenadas dentro de una carpeta en el código fuente del proyecto. Estas se almacenaban en formato JSON y contenían la estructura que formaba una ruta.

La Ilustración 17 muestra un ejemplo de una ruta en formato JSON.

Ilustración 17 - Ruta de Nodos en Formato JSON

Fuente: (Elaboración Propia)

Esta forma de almacenar las rutas generaba un problema al trabajar en distintas ramas ya que se debían agregar en cada una de ellas y en ocasiones generaba conflictos al unir las ramas, adicionalmente se generaba un error en las funciones en la nube de Firebase donde solicitaba archivos JSON que se encontraban en ramas que habían sido desplegadas anteriormente pero que no se encontraban en la rama que estaba siendo desplegada.

Por estos motivos se solicitó cambiar la forma en la que se accedía a estas rutas, se decidió eliminar por completo la carpeta que las contenía en el código fuente del proyecto y guardarlas en la base de datos en tiempo real de Firebase, para que todos los despliegues accedieran a los mismos datos.

La Ilustración 18 muestra un ejemplo de una ruta en la base de datos en tiempo real de Firebase.

Ilustración 18 - Ruta de Nodos en Base de Datos de Firebase

Fuente: (Firebase, s.f.)

Para este desarrollo se crearon las respectivas funciones necesarias, tales como agregar una ruta a la base de datos, recuperar todas las rutas y obtener una ruta en específico mediante su identificador. Con las funciones creadas se revisó el código fuente para encontrar todos los lugares donde se hacía uso de los archivos en formato JSON y se procedió a reemplazar este código con las funciones, para así utilizar las rutas recuperadas desde la base de datos en tiempo real de Firebase.

La Ilustración 19 muestra el código fuente de la función para guardar una ruta en la base de datos de Firebase.

```
async function AddPathwayToDatabase(pathway){
    let res;
    let error;
    const pathwayID = Object.keys(pathway)[0]
    const pathwayValues = Object.values(pathway)[0]

    try{
        await database.ref(`/pathways/${pathwayID}`).set(pathwayValues)
        res = `Pathway ${pathwayValues.title} | ${pathwayID} it's now in DB`;
        error = false;
    }catch(e){
        console.log(e)
        logger.error(e,{structuredData:true})
        res = e;
        error = true
    }
    console.log('res:', res)
    return [error,res];
}
```

Ilustración 19 - Función para Guardar Rutas en Base de Datos

Fuente: (Elaboración Propia)

4.1.12 Cambios en Componente de Otorgamiento Automático

A medida que el número de usuarios y rutas aumentaba dentro de la base de datos del proyecto Acera se identificaron problemas con respecto al proceso de otorgamiento automático de medallas que sucedía en las funciones en la nube de Firebase. El error más notorio fue un desbordamiento de pila durante la ejecución de la función.

Mediante pruebas en el flujo del proceso de otorgamiento automático de medallas se descubrió que el desbordamiento de la pila era causado porque se poseía demasiada recursión y que las llamadas recursivas aumentaban considerablemente al aumentar el número de usuarios o rutas, por lo que se decidió cambiar el enfoque para el proceso.

Se optó por optimizar el preprocesamiento de los datos necesarios para realizar el otorgamiento automático, de igual manera se mejoraron los filtros para no hacer iteraciones redundantes como revisar varias veces la misma ruta en un usuario.

Se desarrolló un algoritmo para darle un mejor formato a las rutas mediante una función recursiva que recorre la estructura de la ruta y genera un arreglo de una dimensión con cada nodo y las medallas necesarias del mismo, esto facilitaría el resto del proceso ya que este arreglo simplemente se debe iterar con un ciclo básico y no con muchas funciones recursivas que generan una carga de memoria y provocan el desbordamiento de la pila.

La Ilustración 20 muestra el código fuente de la función para darle formato a las rutas.

```
434  const reformatChildren = (children) => {
435    for(let child of children){
436      if(child.needs !== undefined){
           Complexity is 5 Everything is cool!
437        let arr = child.needs.map(function(a){
438          if(a.id){
439            return a.id;
440          }else{
441            return a;
442          }
443        });
444        let obj = {
445          id: child.id,
446          needs: arr
447        }
448        if(obj.needs[0] !== ''){
449          values.push(obj)
450        }
451      }
452    }
453    for(let child of children){
454      if(child.needs){
455        num++
456        reformatChildren(child.needs)
457      }
458    }
459  }
```

Ilustración 20 - Función de Formato de Rutas

Fuente: (Elaboración Propia)

4.2 CRONOGRAMA DE ACTIVIDADES

En la Ilustración 21 se observa el Cronograma de Actividades realizadas hasta la fecha.

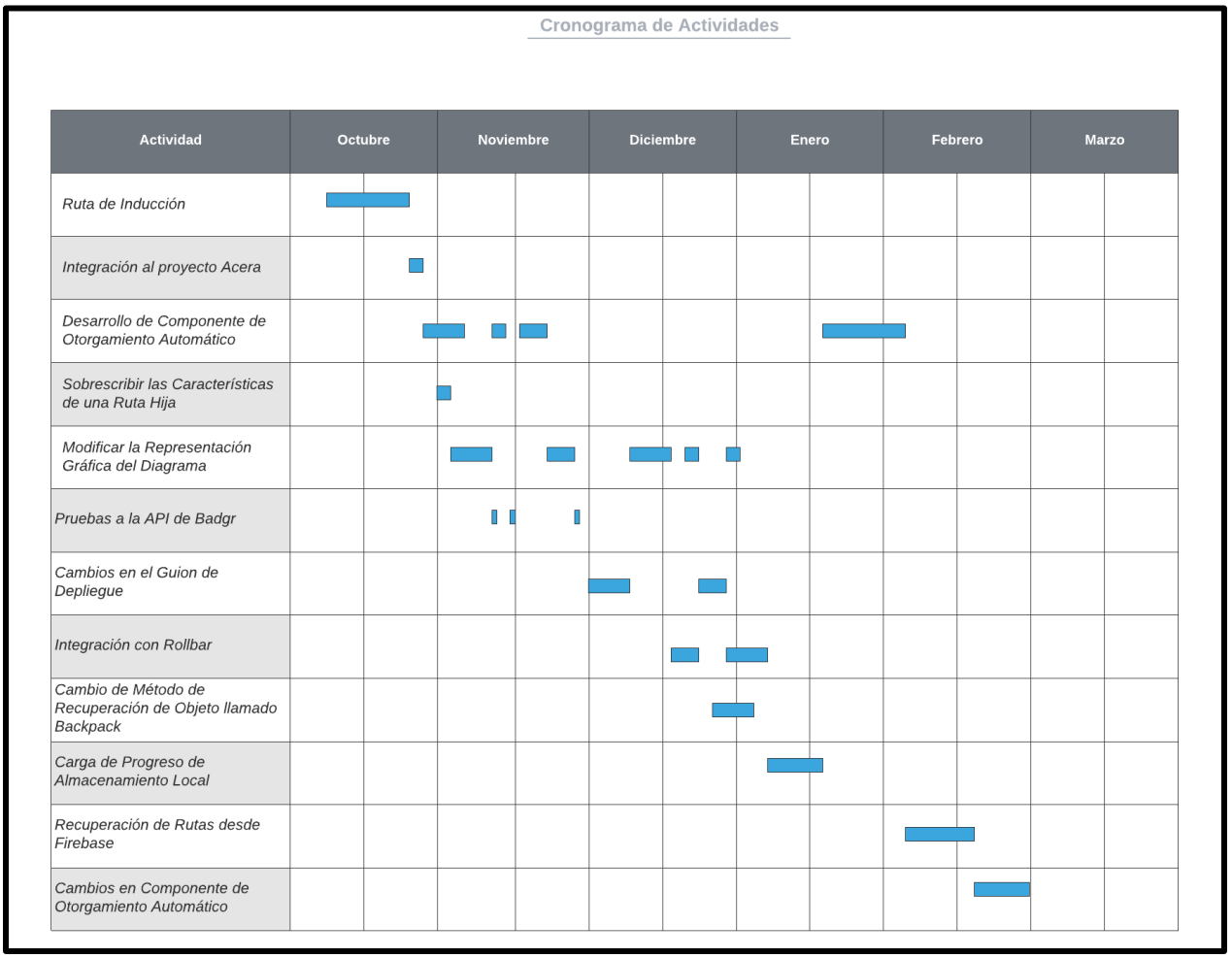


Ilustración 21 - Cronograma de Actividades

Fuente: (Elaboración Propia)

V. CONCLUSIONES

- Durante el transcurso de la práctica profesional se logró participar en el diseño de distintos diagramas de flujo sobre el proceso de trabajo, así como en el desarrollo e implementación de múltiples soluciones de software que influyeron positivamente en el avance para el lanzamiento del proyecto interno Acera. Todo esto con la ayuda de las herramientas y procesos utilizados por la empresa y manteniendo los estándares de calidad adecuados.
- Se desarrollaron y mejoraron diversos componentes funcionales los cuales fueron integrados con éxito en el proceso de desarrollo seguido por el equipo del proyecto interno Acera. Estos componentes fueron revisados y aprobados por el asegurador de calidad asignado, de esta manera se validó el trabajo realizado.
- Se asistió con la documentación en el código implementado y de distintos diagramas sobre el proyecto interno Acera, de igual manera se dio mantenimiento a distintos componentes funcionales que presentaron fallos y se mejoraron las características existentes.
- Se utilizaron las tecnologías React y Node.js para realizar los distintos desarrollos solicitados dentro del proyecto interno Acera durante el transcurso de la práctica profesional.

VI. RECOMENDACIONES

- Se debe reducir lo más posible la dependencia de APIs externas ya que esto puede ralentizar el flujo de trabajo esperado en los proyectos. De igual forma se deben conocer las herramientas y procesos que se utilizan dentro de los proyectos para trabajar de manera óptima.
- Se deben seguir todos los pasos del proceso de trabajo y despliegue que posea el proyecto en el cual se trabaja para evitar generar y propagar errores en los distintos ambientes manejados en el desarrollo.
- Se debe generar y mantener una buena documentación para que los futuros desarrolladores posean más facilidades de comprender el funcionamiento del proyecto, de igual manera, se deben realizar los mantenimientos necesarios en las aplicaciones para garantizar su funcionalidad.
- Se recomienda siempre poseer información sobre las actualizaciones y documentación de la tecnología usada en el proyecto para conocer las características más recientes y trabajar de manera óptima.

BIBLIOGRAFÍA

1. *¿Qué es D3.js?* (2020). Recuperado el 14 de noviembre de 2020, de edspresso: https://www.educative.io/edpresso/what-is-d3js?utm_source=Google%20AdWords&aid=5082902844932096&utm_medium=cpc&utm_campaign=kb-dynamic-edpresso&gclid=Cj0KCQiAnb79BRDgARIsAOVbhRpIQUEcBPgHwEsg-_ycH592hFzU473IGAaLPMLm7jheglJgx_bv40IaAtsiEALw_wcB

2. *¿Qué es Firebase?* (16 de abril de 2019). Recuperado el 15 de noviembre de 2020, de RockContent: https://rockcontent.com/es/blog/que-es-firebase/

3. *¿Qué es JavaScript?* (8 de agosto de 2020). Recuperado el 15 de noviembre de 2020, de MDN web docs: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript

4. *¿Qué es Node.js y para qué sirve?* (26 de octubre de 2020). Recuperado el 16 de enero de 2021, de Gacelaweb: https://www.gacelaweb.com/que-es-nodejs-y-para-que-sirve/

5. *¿Qué es Postman y para qué sirve?* (3 de junio de 2019). Recuperado el 14 de noviembre de 2020, de OpenWebinars: https://openwebinars.net/blog/que-es-postman/

6. *¿Qué es Postman?* (2020). Recuperado el 14 de noviembre de 2020, de La Madriguera Bit: https://lamadriguerabit.com/articulos/que-es-postman/

7. *¿Qué es React y para qué sirve?* (27 de enero de 2020). Recuperado el 13 de noviembre de 2020, de Drauta: https://www.drauta.com/que-es-react-y-para-que-sirve

8. *Acklen Avenue: Acerca de*. (s.f.). Recuperado el 23 de octubre de 2020, de LinkedIn: https://www.linkedin.com/company/acklen-avenue

9. *Badgr*. (s.f.). Recuperado el 15 de noviembre de 2020, de Badgr: https://badgr.com/

10. *Definición y Características del Scrum Master*. (10 de Septiembre de 2015). Recuperado el 25 de octubre de 2020, de IeBS: https://www.iebschool.com/blog/definicion-y-caracteristicas-del-scrum-master-agile-scrum/

11. *Defining Acklen Avenue*. (s.f.). Recuperado el 23 de octubre de 2020, de AcklenAvenue: https://acklenavenue.com/blog/defining-acklen-avenue

12. *Firebase*. (s.f.). Recuperado el 20 de febrero de 2021, de Firebase: https://console.firebase.google.com/

13. *Firebase: ¿qué es y para qué sirve?* (17 de mayo de 2020). Recuperado el 14 de noviembre de 2020, de Digital55: https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/

14. *Metodología 'scrum': ¿Qué es un Sprint?* (1 de Marzo de 2019). Recuperado el 25 de octubre de 2020, de BBVA: https://www.bbva.com/es/metodologia-scrum-que-es-un-sprint/

15. *Postman*. (s.f.). Recuperado el 15 de noviembre de 2020, de Postman: https://www.postman.com/

16. *Principles behind the Agile manifesto*. (2001). Recuperado el 24 de octubre de 2020, de Agile Manifesto: https://agilemanifesto.org/iso/es/principles.html

17. *Qué es Kanban: Definición, Características y Ventajas*. (2020). Recuperado el 25 de octubre de 2020, de Kanbanize: https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban

18. *Qué es Scrum*. (s.f.). Recuperado el 25 de octubre de 2020, de proyectos agiles.org: https://proyectosagiles.org/que-es-scrum/

19. *Rollbar*. (s.f.). Recuperado el 12 de enero de 2021, de Rollbar: https://rollbar.com

20. *Rollbar - Una Herramienta para detectar, diagnosticar y eliminar errores web* . (s.f.). Recuperado el 10 de enero de 2021, de Unweaving the Web: https://estebanromero.com/herramientas-emprender-desarrollar-proyectos/rollbar-una-herramienta-para-detectar-diagnosticar-y-eliminar-errores-web/

21. *Rollbar Software*. (s.f.). Recuperado el 10 de enero de 2021, de Software Advice: https://www.softwareadvice.com/bug-tracking/rollbar-profile/

22. Rubin, K. S. (2012). *Essential Scrum*. Arbor: Addison-Wesley.

23. *Scrum Guide*. (2020). Recuperado el 23 de octubre de 2020, de Scrum Guides: https://scrumguides.org/scrum-guide.html

24. *Significado de Kanban*. (2016). Recuperado el 25 de octubre de 2020, de Significados: https://www.significados.com/kanban/

25. *Significado de Software*. (1 de Agosto de 2019). Recuperado el 25 de octubre de 2020, de Significados: https://www.significados.com/software/

26. *Significado de Web*. (s.f.). Recuperado el 15 de noviembre de 2020, de Significados: https://www.significados.com/web/

27. *Tech 101: ¿Qué es React?* (s.f.). Recuperado el 13 de noviembre de 2020, de SkillCrush: https://skillcrush.com/blog/what-is-react-js/#toc