



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PRÁCTICA PROFESIONAL

DESARROLLADOR DEL DEPARTAMENTO DE IT, SEGURIDAD MÓVIL GPS

PREVIO A LA OBTENCIÓN DEL TÍTULO:

INGENIERÍA EN SISTEMAS COMPUTACIONALES

PRESENTADO POR:

21521086 DAVID EDGARDO NÚÑEZ MOLINA

ASESOR: ING. ROQUE EDGARDO LÓPEZ

CAMPUS SAN PEDRO SULA; JULIO, 2022

Dedicar a mi familia este nuevo logro en mi vida, en especial a mis padres, David Núñez y Joaquina Molina, quienes hicieron hasta lo imposible por ayudarme, no solo económicamente sino en todo aspecto a lo largo de vida. Asimismo por estar pendientes en todo momento de mi carrera, por su sacrificio y esfuerzo por darme una carrera para un mejor futuro y por creer en mi capacidad y sobre todo por brindarme su amor.

AGRADECIMIENTOS

Primeramente agradezco a Dios por guiarme en el camino y estar conmigo en cada paso que doy, por darme la Inteligencia, sabiduría necesaria y darme fortaleza en tiempos duros.

Agradecer a todos los integrantes del departamento de IT de Seguridad Móvil GPS que me recibieron con los brazos abiertos, ayudándome a desempeñarme de la mejor manera y a familiarizarme con todas las tecnologías, métodos de desarrollo y estándares dentro de la empresa.

De igual manera agradecer a la Universidad Tecnológica Centroamericana, juntamente con los honorable catedráticos por transmitirme sus diversos conocimientos especialmente del campo y de los temas que corresponden a mi profesión.

Y a todas aquellas personas que estuvieron presente a lo largo de mi carrera universitaria.

RESUMEN EJECUTIVO

El presente informe de práctica profesional tiene como objetivo detallar el desarrollo de la práctica profesional realizada para optar la Ingeniería Sistemas Computacionales en la Universidad Tecnológica Centroamericana (UNITEC).

Los objetivos principales de la práctica profesional son desarrollar las habilidades, destrezas y conocimientos adquiridos durante la carrera Ingeniería en Sistemas Computacionales, trabajar en equipo y adquirir experiencia en el mundo laboral.

La práctica profesional fue desarrollada en Seguridad Móvil GPS ya que es una de las corporaciones más grandes, modernas y activas de Honduras enfocada a los servicios de posicionamiento global y rastreo activo.

La práctica fue ejecutada específicamente en el departamento de IT, en el periodo entre el 17 de enero al 24 de junio del presente año como desarrollador web. Este departamento se encarga de garantizar el buen funcionamiento del software y desarrollar nuevas características y funcionalidades para el usuario final.

Entre las actividades principales que tuvo, en el proyecto contable de la empresa, el desarrollo de nuevas características en la edición y creación de facturas, proformas, cotizaciones, notas de débito y crédito. Ya que esta acción anteriormente se tenía que realizar manual mente en la base de datos, con el peligro que eliminar o editar el registro incorrecto. De igual manera se desarrolló la funcionalidad de generación de estados de cuentas sobre las facturas y notas de débito, en el cual se resumen las facturas y notas de débito, asimismo dando la posibilidad de compartirlo vía correo electrónico.

Como actividades de mejora al proyecto, el practicante colaboro en la creación de pruebas automatizadas utilizando Cypress, asimismo se encargó de la optimización y actualización de los gráficos que generan los reportes sobre datos obtenidos por parte de los GPS, y entre otras actividades que lograron la mejora y automatización del proyecto.

ÍNDICE DE CONTENIDO

Lista de Siglas	1
Glosario	2
I. Introducción.....	3
II. Generalidades de la Empresa	5
2.1 Descripción de la Empresa	5
2.1.1 Misión	6
2.1.2 Visión	7
2.2 Descripción del Departamento o Unidad	7
2.3 Objetivos del Puesto.....	8
2.3.1 Objetivos General	8
2.3.2 Objetivos Específicos.....	8
III. Marco Teórico	9
3.1 Tecnologías Utilizadas.....	9
3.1.1 ReactJs	9
3.1.1.1 Componentes de clase.....	9
3.1.1.2 Componentes funcionales.....	10
3.1.1.3 Hooks	10
3.1.2 NodeJs	11
3.1.3 Visual Studio Code	12
3.1.4 NPM	12
3.1.5 Redux	13
3.1.6 ESLint	13

3.1.7	Ant Design.....	14
3.1.8	BizCharts.....	14
3.1.9	CSS	15
3.1.9.1	SASS	15
3.1.10	Loopback	16
3.1.11	Datagrip.....	17
3.1.12	Git.....	18
3.1.12.1	GitHub.....	19
3.1.12.2	GitKraken.....	19
3.1.13	Slack	20
3.1.14	Trello.....	21
3.1.15	Cypress.....	22
3.1.16	PuTTY	23
3.2	Buenas Practicas.....	24
IV.	Desarrollo	25
4.1	Proyectos de desarrollo.....	25
4.1.1	Optimus GPS	25
4.1.2	IronHide.....	25
4.2	Cronología de Desarrollo	25
4.2.1	Semana 1 (17 de enero a 21 de enero).....	26
4.2.2	Semana 2 (24 de enero a 28 de enero).....	26
4.2.3	Semana 3 (31 de enero a 4 de febrero).....	27
4.2.4	Semana 4 (7 de febrero a 11 de febrero)	28

4.2.5	Semana 5 (14 de febrero a 18 de febrero).....	29
4.2.6	Semana 6 (21 de febrero a 25 de febrero).....	29
4.2.7	Semana 7 (28 de febrero a 4 de marzo).....	30
4.2.8	Semana 8 (7 de marzo a 11 marzo).....	30
4.2.9	Semana 9 (14 de marzo a 18 marzo).....	31
4.2.10	Semana 10 (21 marzo a 25 de marzo).....	32
4.2.11	Semana 11 (28 de marzo a 1 de abril).....	32
4.2.12	Semana 12 (4 de abril a 8 de abril).....	33
4.2.13	Semana 13 (11 de abril de 15 de abril).....	35
4.2.14	Semana 14 (18 de abril a 22 de abril).....	36
4.2.15	Semana 15 (25 de abril a 29 de abril).....	37
4.2.16	Semana 16 (2 de mayo a 6 mayo).....	38
4.2.17	Semana 17 (9 de mayo a 13 de mayo).....	39
4.2.18	Semana 18 (16 de mayo a 20 de mayo).....	40
4.2.19	Semana 19 (23 de mayo a 27 de mayo).....	41
4.2.20	Semana 20 (30 de mayo a 3 de junio).....	43
4.2.21	Semana 21 (6 de junio a 10 de junio).....	45
4.2.22	Semana 22 (13 de junio a 17 de junio).....	46
4.3	Diagrama de Gantt.....	48
V.	Conclusiones.....	49
VI.	Recomendaciones.....	50
	Bibliografía.....	51

ÍNDICE DE ILUSTRACIONES

Ilustración 1, Ejemplo de Componente de clase.....	10
Ilustración 2, Ejemplo de Componente funcional.....	10
Ilustración 3, Ejemplo de Hooks useState.....	11
Ilustración 4, Ejemplo de Interfaz VSC	12
Ilustración 5, Ejemplo de Componentes de Ant Design.....	14
Ilustración 6, Ejemplo gráficos de Bizcharts.....	15
Ilustración 7, Comparación de Sintaxis SCSS con CSS	16
Ilustración 8, Ejemplo de Interfaz de Loopback	16
Ilustración 9, Ejemplo de Datagrip con PostgreSQL.....	17
Ilustración 10, Ejemplo Interfaz de GitKraken	19
Ilustración 11, Ejemplo Interfaz de Slack.....	20
Ilustración 12, Ejemplo Interfaz de Trello.....	21
Ilustración 13, Ejemplo de código de Cypress	23
Ilustración 14, Ejemplo ejecución de prueba.....	23
Ilustración 15, Reporte de Control de Combustible, Estilo "single sheets"	27
Ilustración 16, Ejemplo de nota de crédito	33
Ilustración 17, Relación de la tabla "payment"	34
Ilustración 18, Ejemplo de desestructuración de props.	34
Ilustración 19, Ejemplo estado de cuenta generado.	36
Ilustración 20, Ejemplo de mock API (Carriers).....	38
Ilustración 21, Implementación del método debounce.	39
Ilustración 22, Ejemplo de formulario con opción "clientes varios"	40

Ilustración 23, Problema de implementación bizcharts v3	41
Ilustración 24, Optimización e implementación bizcharts v4.....	42
Ilustración 25, Mock del comando de login	43
Ilustración 26, Ejemplo de nuevo método waitOn	44
Ilustración 27, Tiempo de ejecución (Sin mockear)	46
Ilustración 28, Tiempo de ejecución (Mockeados).....	46
Ilustración 29, Plugin Cypress Screenshot Upload	47

ÍNDICE DE TABLAS

Tabla 1. Términos claves de Git	18
Tabla 2. Semana 1, Actividades diarias	26
Tabla 3. Semana 2, Actividades diarias	27
Tabla 4. Semana 3, Actividades diarias	28
Tabla 5. Semana 4, Actividades diarias	28
Tabla 6. Semana 5, Actividades diarias	29
Tabla 7. Semana 6, Actividades diarias	29
Tabla 8. Semana 7, Actividades diarias	30
Tabla 9. Semana 8, Actividades diarias	31
Tabla 10. Semana 9, Actividades diarias.....	31
Tabla 11. Semana 10, Actividades diarias	32
Tabla 12. Semana 11, Actividades diarias	33
Tabla 13. Semana 12, Actividades diarias	35
Tabla 14. Semana 13, Actividades diarias	36
Tabla 15. Semana 14, Actividades diarias	37
Tabla 16. Semana 15, Actividades diarias	38
Tabla 17. Semana 16, Actividades diarias	39
Tabla 18. Semana 17, Actividades diarias	40
Tabla 19. Semana 18, Actividades diarias	41
Tabla 20. Semana 19, Actividades diarias	42
Tabla 21. Semana 20, Actividades diarias	44
Tabla 22. Semana 21, Actividades diarias	45

Tabla 23. Semana 22, Actividades diarias 47

LISTA DE SIGLAS

API Application Programming Interfaces

GPS Global Positioning System

HTTP Hypertext Transfer Protocol

IT Information Technology Department

REST Representational State Transfer

SOAP Simple Object Access Protocol

SSH Secure Shell

VSC Visual Studio Code

NPM Node Package Manager

SASS Syntactically Awesome Styles Sheets

HTML Hypertext Markup Language

XML Extensible Markup Language

GLOSARIO

Back-end Es un sistema corporativo que se utiliza para ejecución de sitios web de una organización, como sistema de gestión de inventario y procesamientos de suministros.

Front-end Es la parte del sitio web con la que el usuario interactúa para recibir o utilizar las capacidades de back-end del sistema anfitrión.

Geocercas Perímetro o cerca virtual alrededor de una ubicación física.

JavaScript Lenguaje de programación de computadora orientado a objetos que se usa comúnmente para crear efectos interactivos dentro de los navegadores web.

TypeScript Es un lenguaje de programación sintáctico estricto de JavaScript el cual agrega escritura estática opcional al lenguaje.

Framework Es un esquema que ofrece una estructura base para elaborar un proyecto con objetivos específicos, simplificando las tareas o procesos.

Package.Json Es un archivo con extensión JSON, que contiene información importante sobre el proyecto así como metadatos funcionales.

I. INTRODUCCIÓN

En el presente informe se detallará la experiencia obtenida en la empresa Seguridad Móvil GPS en la cual el practicante David Edgardo Nuñez Molina ejercerá sus conocimientos obtenidos al largo de la carrera de Sistemas Computacionales. Brindando los detalles del desarrollo de las actividades realizadas en los proyectos que formo parte, así como descripciones detalladas sobre cada actividad que se realizó tanto semanalmente como diariamente.

El área donde se realizará la práctica profesional será en el departamento de IT, bajo la supervisión del Ingeniero Alex Fernández (Gerente de IT). La cual tendrá como objetivo mejorar y desarrollar nuevas funcionalidades bajo demanda de los clientes, para los diferentes proyectos que cuenta la empresa. Asimismo la implementación de pruebas de integración con el fin de verificar el funcionamiento de los diferentes componentes, así como un mecanismo para asegurarse que los cambios que se llegasen a realizar no afecten el funcionamiento de la aplicación. Actualizar o sustituir paquetes dentro de los proyectos con el fin de mantener bajo vanguardia las aplicaciones web y prevenir riesgos de seguridad por paquetes desactualizados que poseen errores de seguridad conocidos.

El informe se dividirá en cinco capítulos que nos permita detallar en su totalidad la práctica profesional del estudiante.

El capítulo II detallará la información general de la empresa Seguridad Móvil GPS, dando una breve explicación sobre los tipos servicios que provee la empresa, así mismo se brindará una descripción acerca del departamento de IT el cual se realizará la práctica profesional, de igual manera describirá de manera breve los proyectos en los que participo y los equipos de trabajo que conformó y los objetivos personales de la práctica profesional.

En el capítulo III detallará el marco de teórico, asimismo se explicará el funcionamiento de las herramientas y las tecnologías fundamentales utilizadas para el desarrollo de los proyectos como fueron ReactJs, NodeJs entre muchas otras.

En cuanto el capítulo IV este presentara el desarrollo de las actividades de los distintos proyectos en el cual formara parte dentro de la empresa Seguridad Móvil GPS durante el tiempo de práctica

profesional. De esta manera detallando cada una de las actividades llevando un flujo de trabajo y procesos realizados de los proyectos.

El capítulo V se reducirá todo el informe en base a las conclusiones obtenidas sobre la experiencia laboral.

Y finalmente en el capítulo VI se presentará las recomendaciones que pueda sugerir el practicante acerca de los procedimientos de desarrollo actuales.

II. GENERALIDADES DE LA EMPRESA

2.1 DESCRIPCIÓN DE LA EMPRESA

Seguridad Móvil GPS es una empresa 100% hondureña, pionera en el servicio de rastreo de vehículos, cargamento y personal mediante dispositivos de posicionamiento global (GPS) y localización constante mediante satélites. La cual poseen de una gran cartera de clientes compuesta por prestigiosas empresas e individuos, lo que los hace ser la empresa más sólida en el rubro a nivel nacional.

Actualmente en Seguridad Móvil GPS cuenta con 2 aplicaciones para la disposición de los clientes las cuales son Optimus GPS versión web y versión móvil, las cual hacen uso de la herramienta de Google maps que permiten al cliente visualizar con precisión los dispositivos en los cuales se instaló un dispositivo de posicionamiento global.

Al contratar el servicio de rastreo, la empresa le asigna un usuario y contraseña único al cliente para poder tener accesos a la aplicación web como móvil. Donde podrá administrar y visualizar todos los vehículos que cuenten con una unidad GPS y que están enlazados con su cuenta.

Este sistema de Optimus provee numerosos datos sobre los vehículos como pueden ser:

- Actualizaciones en tiempo real sobre la posición actual, velocidad y dirección que se dirige el vehículo.
- Botón de pánico y apagado de motor en caso de emergencias a través de la plataforma web o app.
- Alertas sobre excesos de velocidad.
- Estadísticas de los recorridos como el tiempo en movimiento o estacionado, kilometraje recorrido, puntos de interés y un historial de las rutas recorridas.
- Creación de eventos de Geocercas: Permiten definir un rango en el mapa donde se pueden registrar las entradas y salidas de vehículos y realizar diferentes eventos como notificaciones por correo.

La empresa también ofrece servicios especiales como pueden ser:

- Monitoreo de rutas de los vehículos el cual permite saber:
 - El porcentaje de eficiencia de las visitas.
 - Los clientes visitados y no visitados definidos por colores.
 - Invasión de rutas por otros camiones.
- Sensor de combustible el cual permite saber:
 - Cantidad de combustible en el tanque del vehículo en todo momento.
 - Consumo de combustible por rango de tiempo y distancia.
 - Llenados de combustible (cantidad de combustible ingresado en la bomba).
 - Consumos irregulares.
 - Reportes y graficas de combustible por rango de fechas.
- Cámaras fotográficas que permiten:
 - Definir eventos para registrar fotografías por vehículo.
 - Solicitar una fotografía por demanda.
- Sensor de temperatura el cual permite saber:
 - La temperatura durante los recorridos del vehículo.
 - Alertas sobre temperaturas máximas y/o mínima definida por el usuario
 - Reportes y graficas de temperatura por rango de fechas.

De igual manera cuentan con una aplicación de facturación llamada Ironhide la cual se utiliza de manera interna para realizar todos los procesos contables y administrativos de la empresa como puede ser la facturación, reportes, cotizaciones, notas de débito y crédito; entro otras funcionalidades contables.

2.1.1 MISIÓN

Proveer servicios especializados en el gestión y monitoreo satelital de transportes a través de tecnología de vanguardia. Con el fin de brindar a nuestros clientes un incremento en la seguridad, control y optimización del área logística de sus empresas.

2.1.2 VISIÓN

Posicionarnos como la empresa líder en monitoreo satelital en toda la región, distinguirnos por ser una empresa que se reinventa permanentemente con el compromiso de crear a nuestros clientes, asociados y personales una experiencia satisfactoria.

2.2 DESCRIPCIÓN DEL DEPARTAMENTO O UNIDAD

La práctica profesional se realizó en el departamento de IT. Este departamento, es la división encargada de la administración de los recursos tecnológicos. Entre las tareas que el área asignada está; el prestar soporte a los inconvenientes tecnológicos de la compañía, desarrollar nuevas características y funcionalidades según soliciten los clientes, dar mantenimiento a aplicaciones y software necesarios para el funcionamiento operativo de la compañía. De igual manera se encarga de asegurar y dar almacenamiento a los datos de la organización. Siendo uno de los principales motores del éxito de una compañía radica en lograr hacer de manera automatizada y eficiente las tareas de rutina.

Debido a las asignaciones solicitadas, se realizaron tareas en el proyecto de IronHide y Optimus versión web. En el proyecto de rastreo se tuvo como líder al jefe del departamento de IT, el cual se encargaba de distribuir las asignaciones necesarias a realizar, así como revisar los avances realizados de cada integrante. El equipo de encargado del proyecto de Optimus estaba conformado por 7 programadores, incluyendo al jefe del departamento. A diferencia del proyecto de facturación tuvo como un jefe inmediato un miembro del equipo de Optimus el cual se encargaba resolver las dudas, revisar los avances, dividir las asignaciones y reunirse con el propietario de la empresa el cual solicitaba las nuevas funcionalidades o mejoras en los procesos que estaban presentes. El proyecto de facturación estaba conformado por 2 programadores incluyéndonos.

2.3 OBJETIVOS DEL PUESTO

2.3.1 OBJETIVOS GENERAL

Aplicar los conocimientos, habilidades y destrezas adquiridos a través de la educación universitaria, asimismo fortalecer mis conocimientos sobre el desarrollo web dando mantenimiento y desarrollando nuevas funcionalidades a los proyectos de IronHide (facturación) y Optimus (servicio rastreo) de la empresa privada Seguridad Móvil GPS.

2.3.2 OBJETIVOS ESPECÍFICOS

1. Desarrollar e implementar en IronHide nueva funcionabilidad de edición de registros para diferentes módulos de la aplicación utilizando patrones de diseño y mejores prácticas de codificación, en los cuales anteriormente si se requerían algún cambio a un registro este se debía de realizar de manera manual en la base de datos.
2. Añadir pruebas de integración de Cypress a diferentes módulos de la aplicación e implementar interceptores para capturar las solicitudes al servidor y regresar respuestas simuladas con el fin de mantener la integridad de la base de datos y evitar generar carga al servidor cada vez que ejecuten las pruebas de integración.
3. Realizar mantenimiento y correcciones de los defectos detectados en el proyecto de Optimus GPS y IronHide.

III. MARCO TEÓRICO

3.1 TECNOLOGÍAS UTILIZADAS

El marco teórico, que se desarrolla a continuación, permite conocer los conceptos básicos necesarios para el entendimiento de las tecnologías y bibliotecas fundamentales para desarrollo de las actividades en los proyectos de IronHide y Optimus. Asimismo conoceremos las razones por la cual utilizar buenas prácticas a la hora de codificar son necesarias en el ámbito laboral y personal.

3.1.1 REACTJS

JavaScript es uno de los lenguajes de programación más populares del mundo. Debido a que impulsado a millones de sitios web y ha atraído a muchos desarrolladores para crear funciones para la web, lo que conlleva a la creación del framework ReactJs.

ReactJs es una biblioteca de JavaScript que se implementa para crear interfaces de usuario, para desarrollar componentes de interfaz de usuario reutilizables. Cuenta con una amplia comunidad al estar basado en JavaScript lo que conlleva a contar con una amplia cantidad de librerías externas las cuales facilitaran el desarrollo de la aplicación.

React básicamente permite el desarrollo de aplicaciones web grandes y complejas que pueden cambiar sus datos sin actualizaciones de página posteriores. Ofreciendo así una aplicación sencilla, robusta y de alto rendimiento experiencia de desarrollo. React se basa en componentes que pueden ser de clase o funcionales, los cuales manejan su propio estado que utilizan para guardar y acceder a datos. React normalmente renderiza principalmente en el lado del servidor usando NodeJs y soporte para aplicaciones móviles nativas se ofrece usando React Native. (Aggarwal, 2018)

3.1.1.1 Componentes de clase

Un componente de clase es una clase de JavaScript que extiende las clases de componente de ReactJs. El cual nos permite guardar su estado y controlar lo que ocurre durante su ciclo de vida exponiéndonos a métodos como `componentDidMount` o `componentWillUnmount`. (React, 2022)

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Ilustración 1, Ejemplo de Componente de clase

Fuente: (React, 2022).

3.1.1.2 Componentes funcionales

Esta función es un componente React válido porque acepta un solo argumento de objeto "props" (que significa propiedades) con datos y devuelve un elemento React. Llamamos a estos componentes de función porque son literalmente funciones de JavaScript. El cual utiliza Hooks para imitar las cualidades de un componente de clase. *(React, 2022)*

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Ilustración 2, Ejemplo de Componente funcional

Fuente: (React, 2022).

3.1.1.3 Hooks

Los Hooks son una nueva incorporación en React 16.8. Que permiten usar el estado y otras características de React sin escribir una clase. Los Hooks brindan una API más directa a los conceptos de React que ya conoce: accesorios, estado, contexto, referencias y ciclo de vida.

Algunos de los hooks con los que cuenta ReactJs son:

- Hooks Básicos
 - useState
 - useEffect
 - useContext

- Hooks Adicionales
 - useRef
 - useReducer
 - useMemo

Entre otros hooks adicionales y de biblioteca. (React, 2022)

```
function Counter({initialCount}) {
  const [count, setCount] = useState(initialCount);
  return (
    <>
      Count: {count}
      <button onClick={() => setCount(initialCount)}>Reset</button>
      <button onClick={() => setCount(prevCount => prevCount - 1)}>-</button>
      <button onClick={() => setCount(prevCount => prevCount + 1)}>+</button>
    </>
  );
}
```

Ilustración 3, Ejemplo de Hooks useState

Fuente: (React, 2022).

3.1.2 NODEJS

NodeJs está diseñado para crear aplicaciones de red escalables en tiempo de ejecución de JavaScript asíncrono basado en eventos. Sin embargo NodeJs no se considera un framework debido a que el tiempo de ejecución se construye sobre un lenguaje de programación y ayuda a la ejecución de los propios framework, Siendo NodeJs y ReactJs siendo idóneas una para la otro, dominando el mercado debido a la capacidad de trabajar del lado del servidor y de los clientes a la vez.

Lo cual contrasta con el modelo de concurrencia más común de hoy en día, en el que se emplean subprocesos de sistemas operativos. NodeJs es similar en diseño e influenciado por sistemas como Event Machine de Ruby y Twisted de Python. NodeJs lleva el modelo de eventos un poco más allá presentando un bucle de eventos como una construcción en tiempo de ejecución en lugar de como una biblioteca. (Node.Js, 2021)

3.1.3 VISUAL STUDIO CODE

Visual Studio Code es un editor de código potente, desarrollador para multiplataformas. El cual incluye soporte para la depuración integrado de Git, igualmente viene con soporte incorporado para JavaScript y NodeJs. Lo cual lo hace el perfecto editor de código a la hora de utilizar ReactJs y NodeJs ya que los dos parten de JavaScript. (*Visual Studio Code*, 2022)

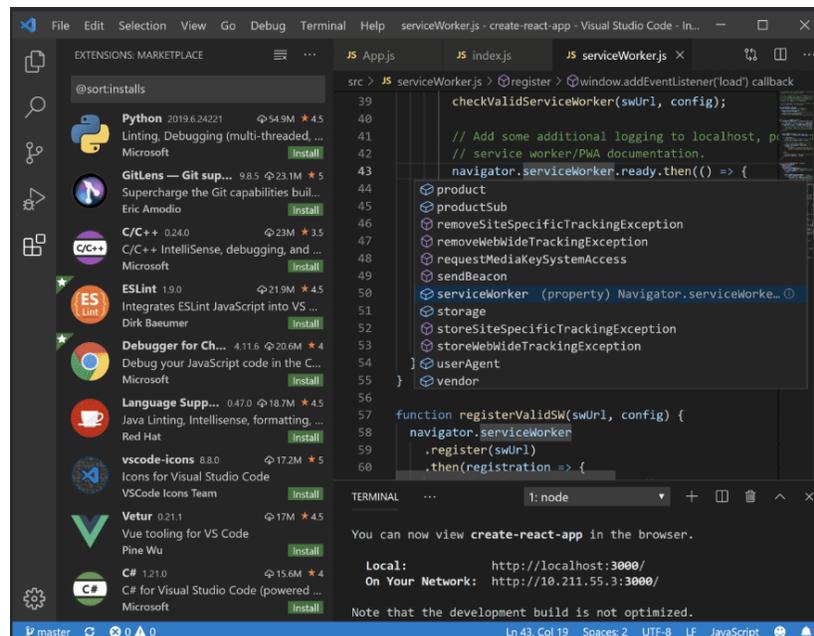


Ilustración 4, Ejemplo de Interfaz VSC

Fuente: (Visual Studio Code, 2022)

Asimismo Visual Studio Code cuenta con una lista inacabable de extensiones que se pueden utilizar para JavaScript lo permiten al desarrollador configurar el editor de texto según la necesidad o el enfoque, así como se puede observar en la *ilustración 4*.

3.1.4 NPM

NPM siendo unas de las extensiones que facilitan el trabajo a la hora de utilizar JavaScript debido a que esta se centra en los archivos package.json, ayudando con la revisión de sintaxis y detectando posibles problemas de instalaciones de dependencias del proyecto o sus versiones. Brindando utilidades para ejecutar scripts npm, cuando son acciones rutinarias. (*Npm*, 2022)

3.1.5 REDUX

Redux es una de las extensiones más populares para React debido a que esta permite describir la interfaz de usuarios como una función de estado. Redux es un contenedor de estado predecible que emite actualizaciones de estados en respuesta a acciones,

Esto significa que la causa del cambio de estado se puede identificar fácilmente. Redux usa una tienda única en lugar de muchas tiendas diferentes como otros flujos las bibliotecas inspiradas lo hacen. Esto a menudo se denomina "fuente única de verdad", ya que la tienda es el único lugar donde se puede acceder al estado. A medida que las aplicaciones se vuelven más complejas, tienen para administrar más estado. Redux hace que los cambios de estado sean predecibles. Lo logra estableciendo las siguientes reglas:

- El estado debe ser inmutable.
- En segundo lugar la información sobre los cambios de estado debe ser contenidos por acciones.
- Por último el uso de reductores, que son funciones puras, especificando cómo las acciones transforman el estado. (Caspers, 2017)

3.1.6 ESLINT

Eslint es un depurador que analiza estáticamente el código JavaScript para encontrar problemas rápidamente. Siendo Eslint una extensión integrada en Visual Studio Code y entre muchos otros más editores de texto. Ahorrando muchos problemas debido a que ESLint soluciona las malas prácticas automáticamente ya que procesa previamente el código y usa analizadores totalmente configurables donde se puede definir criterios (llamados reglas) para que utilice y cuales se prefieren que ignore, adaptándose a las necesidades de la empresa, programador o el equipo de trabajo. (ESLint, 2022)

Eslint se ha establecido como herramienta estándar, por encima de otras como JSHint o JSLint, las cuales han quedado prácticamente obsoletas a su lado. La cual requiere de un mínimo de Node 6.15 y NPM 3+ para funcionar.

3.1.7 ANT DESIGN

Ant Design es un framework que fue diseñado para su uso en el desarrollo de aplicaciones web utilizando ReactJs. Ant Design cuenta con varios valores, principios y patrones de diseño, así como componentes y demos de alta calidad que permiten la creación y desarrollo de aplicaciones web interactivas, llamativas y eficientes. Ant Design tiene como fin facilitar el diseño de interfaces gráficas y mejorar la experiencia del usuario final.

Algunas características que posee Ant Design:

- Potente personalización del tema en cada detalle.
- Soporte de internacionalización para decenas de idiomas.
- Es un paquete completo de recursos de diseño y herramientas de desarrollo.
- Escrito en TypeScript con tipos estáticos predecibles.
- Es una interfaz de usuario de clase empresarial diseñada para aplicaciones web.
- Un conjunto de componentes React de alta calidad listos para usar. (*Ant Design, 2022*)

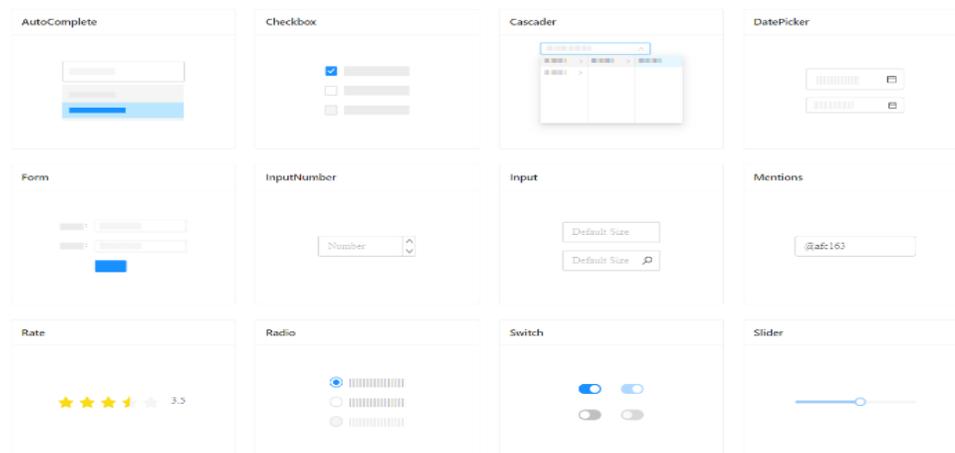


Ilustración 5, Ejemplo de Componentes de Ant Design

Fuente: (Ant Design, 2022).

3.1.8 BIZCHARTS

Es una biblioteca de componentes de gráficos comerciales desarrollada por Alibaba para satisfacer la visualización de los datos generado por los reportes de manera eficiente, profesional

y convenientes. Siendo una librería de fácil uso y que satisface la implementación de gráficos convencionales y gráficos altamente personalizados. (Bizcharts, 2022)

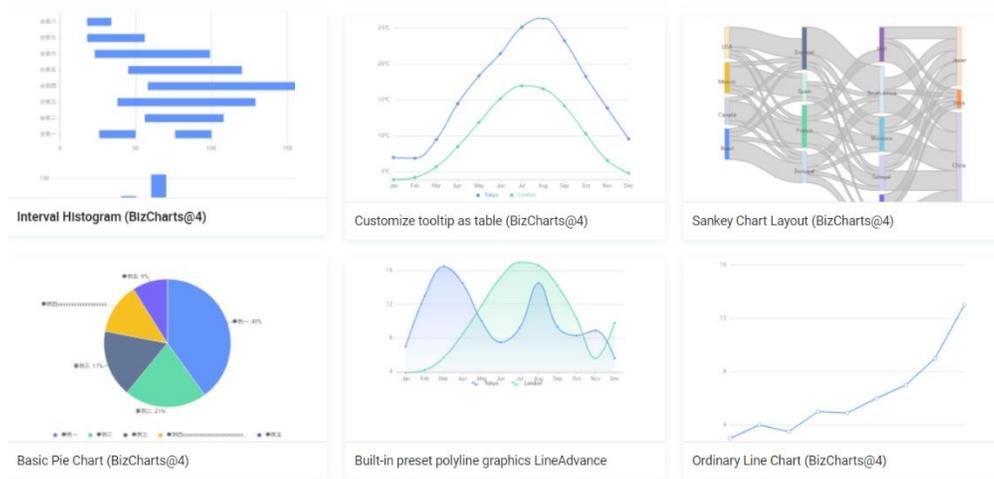


Ilustración 6, Ejemplo gráficos de Bizcharts

Fuente: (Bizcharts, 2022)

3.1.9 CSS

Cascading Style Sheets (CSS), es un lenguaje de estilos poderoso ya que afecta la presentación de un documento o una colección de documentos HTML o XML. Los navegadores utilizan CSS para modificar la presentación del propio navegador. Siendo uno de los lenguajes más importantes para ordenar las instrucciones referentes a la apariencia de un sitio y presentar los contenidos de una página de forma atractiva. (Meyer, 2006)

3.1.9.1 SASS

Es un preprocesador de la sintaxis de CSS, que implementa una nueva sintaxis llamada Sassy CSS, la cual utiliza llaves para denotar bloques de código y punto y coma para separar reglas dentro de un bloque, Los archivos SCSS reciben normalmente la extensión de scss. De igual forma que cada hoja de estilo CSS valida es un archivo SCSS valido con el mismo significado.

Las hojas de estilo cada vez son más grandes y complejas y más difíciles de mantener, lo cual hace que un preprocesador sea de utilizadas. Ya que brindan nuevas características que aún no cuenta CSS, como anidamientos, mezclas, herencias entre otras ventajas. (Sass, 2022)

SCSS	CSS
<pre> nav { ul { margin: 0; padding: 0; list-style: none; } li { display: inline-block; } a { display: block; padding: 6px 12px; text-decoration: none; } } </pre>	<pre> nav ul { margin: 0; padding: 0; list-style: none; } nav li { display: inline-block; } nav a { display: block; padding: 6px 12px; text-decoration: none; } </pre>

Ilustración 7, Comparación de Sintaxis SCSS con CSS

Fuente: (Sass, 2022)

3.1.10 LOOPBACK

Loopback es un framework de código abierto altamente extensible y basado en Express para Node.js. El cual permite crear rápidamente API y microservicios compuestos por sistemas de backend, como bases de datos y servicios SOAP o REST. Loopback nos permite crear modelos (representan tablas en una base de datos), permite el manejo de las tablas y sus campos, y crear relaciones entre los modelos. (LoopBack, 2021)

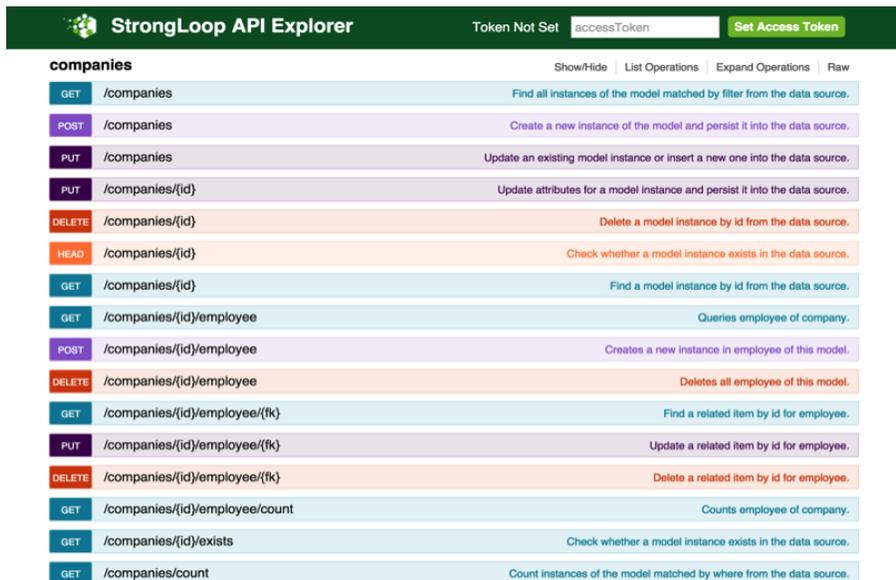


Ilustración 8, Ejemplo de Interfaz de Loopback

Fuente: (LoopBack, 2021).

En la Ilustración 8 se muestra la interfaz gráfica de Loopback. Cuando se crea un modelo en Loopback este crea 4 tipos de transacciones por defecto del protocolo HTTP los cuales son GET, PUT, POST y DELETE. Además, Loopback brinda la opción de crear métodos personalizados o métodos remotos. Los cuales se pueden utilizar en la interfaz gráfica de igual manera que los por defecto.

3.1.11 DATAGRIP

Pasando a la gestión de los datos, Datagrip es un entorno de gestión de bases de datos para desarrolladores SQL. Está diseñado para consultar, crear y administrar bases de datos. Las bases de datos pueden funcionar localmente, en un servidor o en la nube. Admite MYSQL, PostgreSQL, Microsoft SQL Server, Oracle y más.

Algunas características que posee son las siguientes:

- Finalización de código inteligente,
- Inspecciones de código
- Resaltado de errores sobre la marcha
- Arreglos rápidos y capacidades de refactorización.

Lo cual ahorra tiempo al hacer que el proceso de escritura de código SQL sea más eficiente.

(DataGrip, 2022)

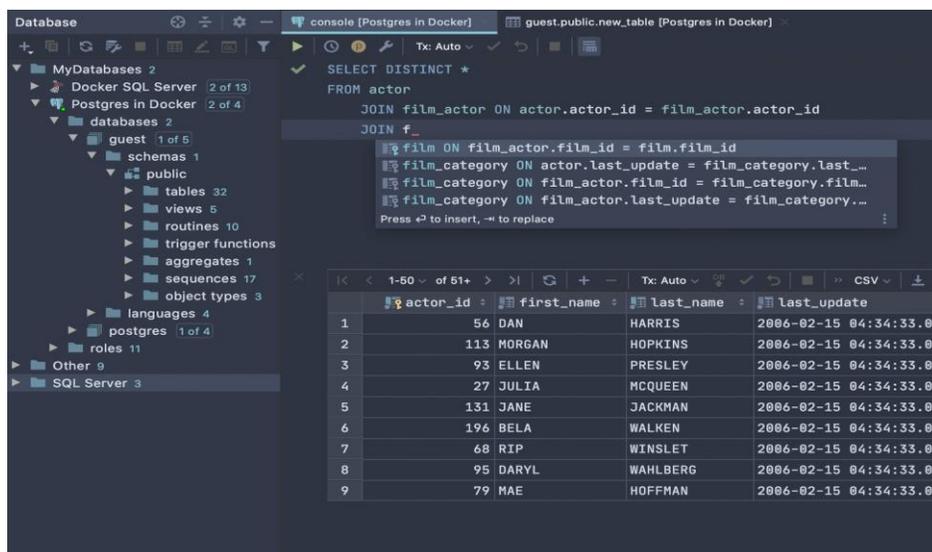


Ilustración 9, Ejemplo de Datagrip con PostgreSQL

Fuente: (DataGrip, 2022).

3.1.12 GIT

Git es el sistema utilizado para el control y rastreo de cambios que se realizan en el día a día o cualquier conjunto de archivos distribuido. Es gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes de organizaciones, con rapidez y eficiencia.

Algunas características que posee:

- Compatibilidad con sistemas y protocolos existentes.
- Manejo eficiente de grandes proyectos.
- Fuerte apoyo para el desarrollo no lineal.
- Fácil coordinación para el desarrollo de código. (Git, 2022)

Tabla 1. Términos claves de Git

Termino	Definición
Git Commit	Un commit representa una captura del repositorio en un punto específico de tiempo. Un commit se realiza en Git como un método de aplicar cambios locales a un repositorio remoto.
Git Clone	El comando de Git clone se utiliza para hacer una copia de un repositorio existente a un directorio local.
Git Merge	El comando de Git Merge permite incorporar los cambios realizados en una rama hacia cualquier otra rama.
Git Branch	Una branch es un puntero a un commit específico. El puntero se mueve junto a cada nuevo commit que se realiza y solo diverge en el grafico si un commit es realizado desde un commit anterior a él.
Git Push	El comando Git Push es utilizado para subir el contenido de un repositorio local a un repositorio remoto.
Git Pull	El comando Git Pull toma los cambios de un repositorio remoto y los incorpora en la rama local en la cual se esté trabajando.
Git Rebase	El comando Git Rebase adquiere los commits, o un grupo de commits, desde una rama fuente y los aplica encima de la rama objetivo.
Git Stash	El comando Git Stash toma el estado del directorio actual en el que se está trabajando y los guarda en un stack de cambios sin terminar que pueden ser reaplicados en cualquier momento.

Fuente: (GitKraken, 2022).

3.1.12.1 GitHub

GitHub es una plataforma en la cual se lleva el desarrollo colaborativo. Que sirve de almacén para aloja los proyectos de los desarrolladores acerca de sus aplicaciones y herramientas utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código por parte de los desarrolladores de manera individual o colaborativa en la cual los demás desarrolladores tienen la capacidad de descargar la aplicación y colaborar con el desarrollo.

GitHub cuenta con las siguientes características fundamentales:

- Repositorios: Lugar virtuales alojados en la donde nube donde el desarrollador almacena el código de su aplicación.
- Pull request: También conocido por solicitudes de admisión de código.
- Issues: Propuestas estos tienen una gran similitud a los hilos de algunos foros y redes sociales, teniendo como objetivo resolver una cuestión en específico. (GitHub, 2022)

3.1.12.2 GitKraken

GitKraken es una herramienta que ayuda a manejar Git de manera sencilla con interfaz de grafica potente y amigable al usuario lo cual transforma el trabajo un poco más eficiente de realizar. Siendo multi plataformas para distinto sistemas operativos del mercado actual. Contando con integraciones a GitHub, gitlab y bitbucket. (GitKraken, 2022)

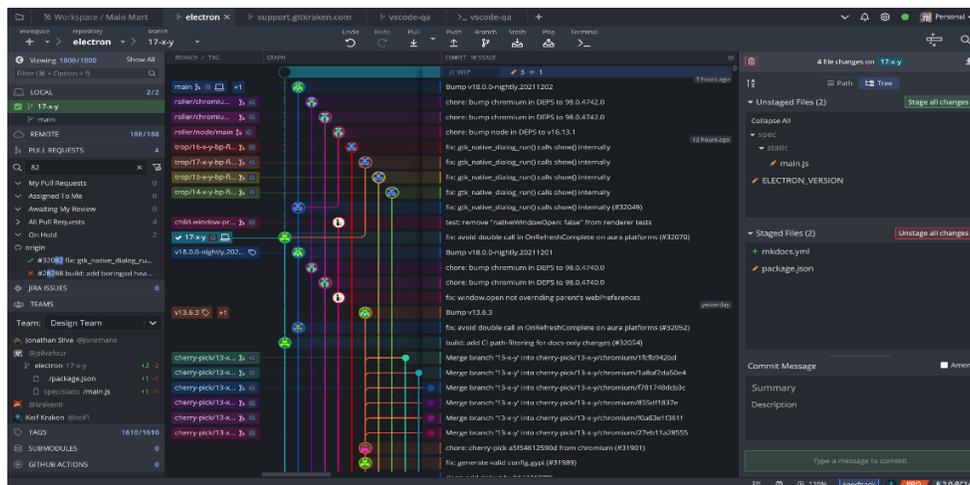


Ilustración 10, Ejemplo Interfaz de GitKraken

Fuente: (GitKraken, 2022).

3.1.13 SLACK

Slack es una herramienta flexible e inclusiva de mensajería entre miembro de la empresa, facilitando la colaboración para los miembros de la organización. Debido a las opciones que ofrece ya sea compartiendo documentos, videos y reuniones interactivas. Compartiendo la pantalla del escritorio o una aplicación en específico dando la opción de a los integrantes de escribir en la pantalla. Slack utiliza los espacios de trabajo para dividir y tener un mejor manejo de los grupos de trabajo. Estos espacios de trabajo son centros compartidos donde los miembros de una organización pueden comunicarse entre ellos. Asimismo Slack posee canales, los cuales permiten al grupo compartir información entre sí, permitiendo que todos tengan acceso a la misma información para que los miembros estén en sincronía.

Beneficios por los cuales utilizar Slack:

- Colaboración en tiempo real.
- Compartir código de manera eficiente.
- Cuentas con muchas integraciones de aplicación que suelen utilizarse. (Slack, 2022)

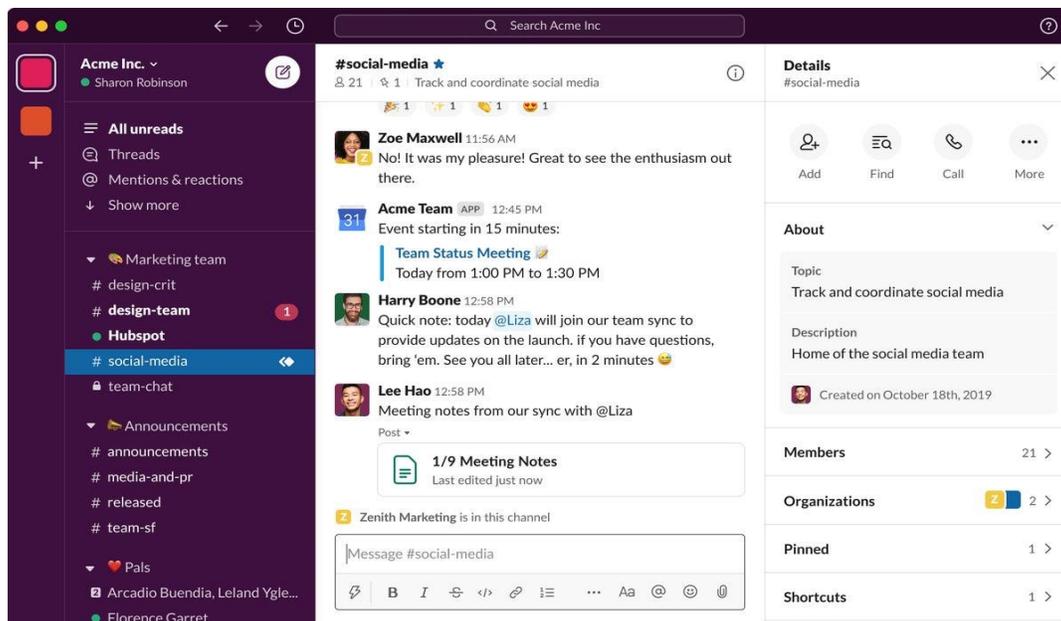


Ilustración 11, Ejemplo Interfaz de Slack

Fuente: (Slack, 2022).

3.1.14 TRELLO

Trello es una herramienta de gestión de proyectos que facilita la organización interna y el trabajo remoto. En donde los integrantes del equipo pueden colaborar, organizar el flujo de trabajo y hacer respectivo seguimiento del proceso de una manera visual y amigable. (Trello, 2022)

Trello se basa en una idea sencilla, para gestionar proyectos de manera eficaz. Teniendo la posibilidad de crea lista de to-do, doing, done entre muchas más opciones personalizables. Siendo sus principales atributos la facilidad de uso y su versatilidad, que permiten organizar desde una lista básica hasta grandes proyectos. Asimismo tiene la posibilidad de crear graficas Gantt y tableros Kanban, logrando sacar el mayor provecho para proyectos personales y profesionales.

Logrando que esta aplicación resuelva problemas como:

- Falta de organización y transparencia.
- Clarificación de las asignaciones de cada miembro.
- Exceso de reuniones.
- Olvidos y omisiones de punto importante del proyecto.

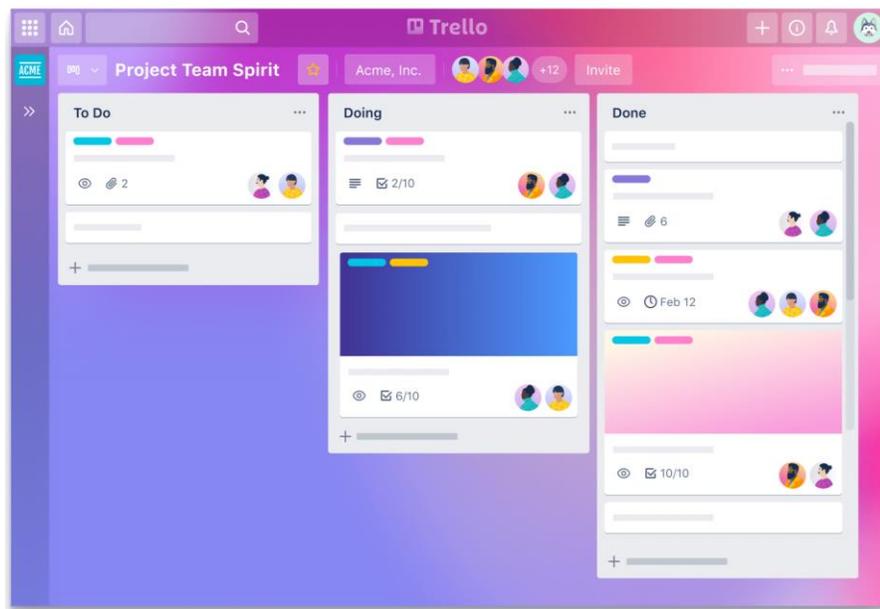


Ilustración 12, Ejemplo Interfaz de Trello

Fuente: (Trello, 2022).

3.1.15 CYPRESS

Cypress es una herramienta de pruebas de front-end de siguiente generación para la web moderna. La cual se utilizó para trata de abordar los puntos clave que los desarrolladores e ingenieros de control de calidad enfrentan al momento de probar aplicaciones modernas. Cypress puede probar cualquier cosa con tal se ejecute en un navegador.

Cypress le permite escribir todo tipo de pruebas:

- Pruebas de extremo a extremo.
- Pruebas unitarias.
- Pruebas de integración.

Características fundamentales que ofrece Cypress son:

- Capturas de pantalla y videos: Lo que permite ver los videos y capturas de pantalla automatizadas en caso de falla.
- Viaje en el tiempo: Cypress durante la ejecución de la prueba toma capturas, siendo un registro de comandos para ver exactamente que paso a detalle.
- Resultados consistentes: Pruebas rápidas, consistentes y confiables.
- Espera automatiza: Permite evitar agregar opciones de espera a las pruebas, Cypress espera automáticamente a que los comandos y las afirmaciones terminen antes de continuar. (Cypress, 2022)

```
it("Should navigate to client-admins screen", () => {
  const getCompaniesIdentifier = "getCompaniesClients";
  const getIClientUsersIdentifier = "getClientsUsers";
  const getAccountsRolesIdentifier = "getAccountsRoles";
  cy.fixture("/client-admins/getCompanyClientsResponse.json").then(
    (response) => {
      cy.mockGetCompanyClients(response, getCompaniesIdentifier);
    }
  );
  cy.fixture("/client-admins/getClientsUsersResponse.json").then(
    (response) => {
      cy.mockGetClientsUsers(response, getIClientUsersIdentifier);
    }
  );
  cy.fixture("/client-admins/getAccountsRolesResponse.json").then(
    (response) => {
      cy.mockGetAccountsRoles(response, getAccountsRolesIdentifier);
    }
  );
  cy.navigate("submenu-admin", "/client", "menu-client-admins");
  cy.waitOn(getCompaniesIdentifier);
  cy.waitOn(getIClientUsersIdentifier);
  cy.waitOn(getAccountsRolesIdentifier);
});

it("Should validate add client-admins button is visible", () => {
  cy.get("[data-testid=client-admin-add-btn]").should("be.visible");
});
```

Ilustración 13, Ejemplo de código de Cypress

Fuente: (Elaboración propia)

Las pruebas de integración de Cypress se ejecutan dentro del navegador controlándolo a través de un proceso backend en Node.js. Con lo que permite a nivel de red la interceptación de todo el tráfico entrante y saliente de la aplicación web. Siendo una ejecución rápida y interactiva a cada uno de los pasos y las acciones ejecutadas durante la prueba.

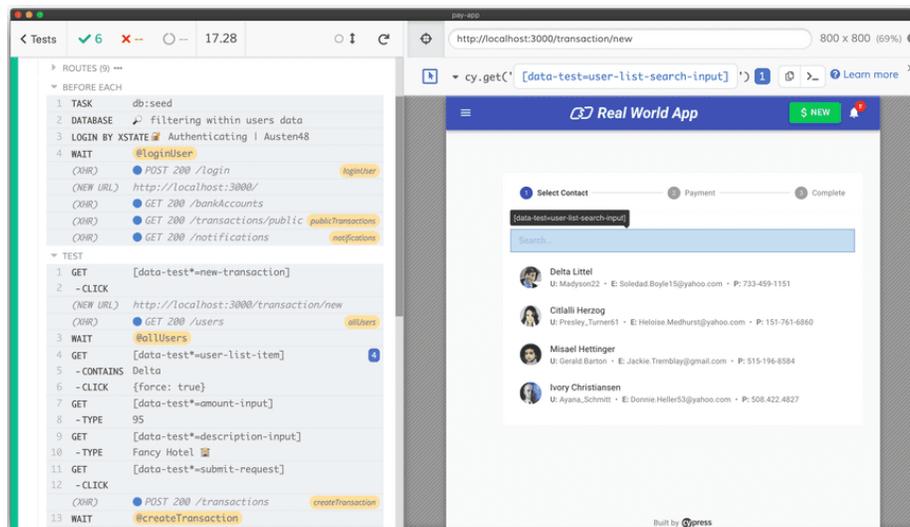


Ilustración 14, Ejemplo ejecución de prueba

Fuente: (Cypress, 2022)

3.1.16 PuTTY

PuTTY es un software multiplataformas de código abierto que está disponible con código fuente y está desarrollado y respaldado por un grupo de voluntarios, siendo una herramienta de uso gratuito y en constante desarrollo. La cual es utilizada para emulación de terminal que soporta diferentes protocolos de red como SSH, Telnet entre otros. (PuTTY, 2022)

Basicamente el uso de PuTTY es debido a facilidad que nos brinda para iniciar una sesión remota SSH con otro máquina usando un nombre de usuario y contraseña o este caso al servidor de desarrollo de los proyectos de la empresa para tener acceso a la base de datos.

Opciones que integra PuTTY:

- Soporta IPv6
- Respuestas de puertos
- Guarda las preferencias de conexión para establecerlas rápidamente.
- Cuenta con versiones portables lo que permite evitar instalaciones en el sistema operativo.

3.2 BUENAS PRACTICAS

A la hora de codificar se puede de hacer de diferentes maneras y que cumpla la función que se espera, sin embargo con la utilización de buenas prácticas podemos obtener un código limpio, reutilizable, escalable y dinámico, el cual nos ayuda a disminuir el tiempo de dedicación en las tareas, a prevenir y sortear errores comunes durante la codificación. En si las buenas prácticas son un conjunto de técnicas, metodologías que debemos implementar en nuestro software para que se vuelva fácil, rápido y seguro de desarrollar.

Algunas de las buenas prácticas que se utilizan en JavaScript:

- Utilizar === en lugar de ==
- Utilizar siempre el punto y coma al final de cada sentencia.
- Definir estándares para la declaración de sentencias switch, for, map entre otros métodos.
- Utilizar Camel Case en los nombres de las variables.
- Cuando la longitud de la línea de código supera alrededor de los 100 – 120 caracteres se deben de realizar un salto de línea, para evitar líneas demasiada s largas.
- Utilizar una extensión para depurar el código, la cual solo debas realizar un script y rápidamente escaneara en busca de problemas y errores notables en el código.

Entre muchas otras buenas prácticas que hagan que el código sea mucho más legible y limpio, ahorrando tiempo para mejorar las capacidades como programador.

IV. DESARROLLO

4.1 PROYECTOS DE DESARROLLO

4.1.1 OPTIMUS GPS

Optimus es un proyecto en el cual se proporciona el servicio de rastreo de los GPS por parte de la compañía. Estas son algunas de las características proporcionadas en la aplicación:

- Un mapa de Google en el cual se pueden visualizar los vehículos que tienen instalado un GPS en tiempo real. Así como un historial de toda la ruta realizada por el vehículo en cualquier momento del día.
- Sección de reportes que permite generar diferentes tipos de reportes como lo pueden ser de distancia, eventos, excesos de velocidad, manejo, combustible entre otros, a su vez proporcionando la opción de exportar estos reportes en un archivo de Excel.
- Configurar mantenimientos a los vehículos dependiendo de 3 tipos configurables: distancia, tiempo y recurrencia en la que se puede determinar un mantenimiento cada X día, semana o mes.
- Definir diferentes tipos de Alertas por vehículo las cuales pueden ser: en movimiento, motor encendido, consumo irregular de combustible, puerta abierta entre otros.

4.1.2 IRONHIDE

Ironhide es un proyecto interno de la compañía en el cual se maneja un sistema de facturación, proformas, notas de crédito, notas de debido y cotizaciones para los clientes de la aplicación de Optimus (servicio de rastreo). Asimismo se lleva la contabilidad de la empresa para tener mayor conocimiento de los ingresos y egresos de la organización.

4.2 CRONOLOGÍA DE DESARROLLO

Durante esta práctica profesional realizada en Seguridad Móvil GPS, se logró desempeñar el puesto de Desarrollador. En la cual se me designo de trabajar en el proyecto interno utilizado por la compañía, de igual forma se me designo a formar parte del equipo de desarrollo que se encarga

del proyecto que provee el servicio de rastreo, lo cual me ayudo a familiarizarme con los proyectos y lograr capacitarme con las tecnologías y estándares que eran requeridos para el desarrollo dentro de la empresa.

El tiempo final invertido como practicante en la empresa Seguridad Móvil fue de 22 semanas, comenzando el 17 de enero hasta el 17 de junio del año 2022.

4.2.1 SEMANA 1 (17 DE ENERO A 21 DE ENERO)

En la primera semana se tuvo múltiples reuniones con un miembro del equipo del desarrollo y el generen de IT para aclarar dudas y ayudar a configurar el ambiente de trabajo para la aplicación de Optimus GPS.

Tabla 2. Semana 1, Actividades diarias

Fecha	Actividades
17 de enero	<u>Optimus:</u> Configuración de accesos a Slack y Trello. Creación del correo empresarial.
18 de enero	<u>Optimus:</u> Instalación y configuración de la aplicación y el API para correrlos de manera local. Configuración de la herramienta Datagrip para poder realizar las operaciones necesarias en la base de datos.
19 de enero	<u>Optimus:</u> Realizar pair Programming con un miembro del área de IT con el cual revisamos el bug de eliminar comandos de un modelo y agregar un botón para mostrar el modal para compartir fotografías por correo electrónico.
20 de enero	<u>Optimus:</u> Crear un custom end point en el API para enviar correos electrónicos y publicar una nueva versión del proyecto de traducciones Intl.
21 de enero	<u>Optimus:</u> Adjuntar foto del automóvil mediante AWS, para que el usuario pueda acceder al enlace de descarga de la foto y agregar un pop up que le indique al cliente que se envió el email con éxito.

Fuente: (Elaboración propia)

4.2.2 SEMANA 2 (24 DE ENERO A 28 DE ENERO)

Durante la semana 2, el gerente de IT creo tarjetas en la plataforma de Trello sobre que componentes se les debía de hacer una migración. Se definió que los componentes que salen en la lista se les debería de realizar los siguientes cambios:

- Realizar una prueba de Cypress al componente para verificar su funcionalidad antes de pasar a realizar cambios en el.
- Pasar el componente de clase a componente funcional para hacer uso de los Hooks que era una nueva funcionalidad proporcionada por la versión de React 16.8.0.

Tabla 3. Semana 2, Actividades diarias

Fecha	Actividades
24 de enero	<u>Optimus:</u> Reunión con un miembro de IT para mostrarme el funcionamiento de las pruebas de Cypress, para empezar a realizar mi primera prueba de Cypress.
25 de enero	<u>Optimus:</u> Realizar la migración de componente de clase a componente funcional del módulo de modelos de los dispositivos.
26 de enero	<u>Optimus:</u> Realizar la prueba de Cypress para comprobar el funcionamiento del formulario de comandos.
27 de enero	<u>Optimus:</u> Crear una prueba de Cypress para revisar el funcionamiento del formulario de agregar y editar modelos y empezar con la migración del componente.
28 de enero	<u>Optimus:</u> Continuar la migración del componente de agregar y editar modelos a componente funcional.

Fuente: (Elaboración propia)

4.2.3 SEMANA 3 (31 DE ENERO A 4 DE FEBRERO)

La semana 3, se trabajó en mejorar la generación y exportación de los reportes de control, lo que llevo implementar el estilo de "Single Sheet" para el documento de Excel para mejorar la vista del documento y optimizar el documento.

1	Unidad	Solicitante	Gasolinera	Dirección	Día	Volumen	Valor	Comentar
2	Camion 3	Alex Fernandez	Uno	105 Brigada	04/10/2021 06:00:00 PM	14,69 L	874	
3	Camion 3	Mabel Izaguirre	Gasolinera#2	Ni idea	20/10/2021 06:00:00 PM	57,32 L	265 asdaawda	
4	Camion 3					72,01 L	1139	
5	Carro 2	Oscar Galindo	Dippsa	ExpoCentro	06/10/2021 06:00:00 PM	20,00 L	541 F	
6	Carro 2	Jose David	Puma#340	El Progreso	06/10/2021 10:56:42 AM	1,00 L	10 Test comment	
7	Carro 2					21,00 L	551	
8	Eco 254	Juan Venegas	Puma	Guamilito	08/10/2021 06:00:00 PM	22,00 L	365	
9	Eco 254					22,00 L	365	
10	AAI0736	Juan Venegas	Dippsa	ExpoCentro	08/10/2021 02:00:00 PM	54,00 L	1062	
11	AAI0736					54,00 L	1062	
12	Camion 2	Oscar Galindo	Texaco#2	Barrio Arriba	12/10/2021 06:00:00 PM	19,00 L	321 Llenado quincenal	
13	Camion 2					19,00 L	321	
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								

Ilustración 15, Reporte de Control de Combustible, Estilo "single sheets"

Fuente: (Elaboración propia)

Tabla 4. Semana 3, Actividades diarias

Fecha	Actividades
31 de enero	<u>Optimus:</u> Terminar la migración del formulario de agregar y editar modelos. Publicar nuevas traducciones en inglés y español para los formularios migrados.
1 de febrero	<u>Optimus:</u> Mostrar la fecha con la hora en la tabla de facturas y tabla de reporte de control de combustible, igualmente se agregó a la hora de exportar el Excel.
2 de febrero	<u>Optimus:</u> Agregar estilo a la hoja de Excel sobre el reporte de control de combustible.
3 de febrero	<u>Optimus:</u> Realizar el end point para función de exportar el reporte de control de combustible con el estilo solicitado.
4 de febrero	<u>Optimus:</u> Corregir la conversión de la zona horaria a la hora de exportar el reporte. Trabajar en "info Windows" para que mostrar la foto del dispositivo.

Fuente: (Elaboración propia)

4.2.4 SEMANA 4 (7 DE FEBRERO A 11 DE FEBRERO)

Se tuvo una reunión con el dueño de la empresa y el gerente de IT en cual me solicitaron preparar el ambiente de trabajo para el proyecto IronHide y me asignaron mis primeras card en trello.

Tabla 5. Semana 4, Actividades diarias

Fecha	Actividades
7 de febrero	<u>Optimus:</u> Se realizar los cambios solicitados en el pull request del componente migrado de los Modelos.
8 de febrero	<u>Optimus:</u> Corregir las pruebas de Cypress para que el pull request pueda permitir hacer Merge con la branch de desarrollo.
9 de febrero	<u>IronHide:</u> Reunión con el dueño de la empresa para discutir las primeras asignaciones en IronHide.
10 de febrero	<u>IronHide:</u> Configurar el ambiente de trabajo para el proyecto de IronHide.
11 de febrero	<u>IronHide:</u> Terminar las configuraciones para IronHide y realizar pruebas de conexión de SQL Server con Datagrip.

Fuente: (Elaboración propia)

4.2.5 SEMANA 5 (14 DE FEBRERO A 18 DE FEBRERO)

Durante la semana 5, se trabajó en la primera asignación requerida por parte del dueño de la empresa, en la se solicitó la desarrollar una nueva funcionabilidad que permita deshabilitar clientes y se agregó una columna a la tabla para mostrar el siguiente mes a pagar del cliente, para facilitar la identificación por parte de los miembros del área de contabilidad.

Tabla 6. Semana 5, Actividades diarias

Fecha	Actividades
14 de febrero	<u>IronHide:</u> Empezar a trabajar con la primera asignación de IronHide, que consistía en crear un pop confirm para deshabilitación de un o muchos clientes.
15 de febrero	<u>IronHide:</u> Agregar dos nuevas columnas una drop down botón con las acciones de la tabla de clientes aprobados y la segunda para indicar la fecha a pagar del siguiente cliente.
16 de febrero	<u>IronHide:</u> Agregar combo box para poder seleccionar varias filas de la tabla de clientes aprobados y aparezca una opción de deshabilitar todos a la vez.
17 de febrero	<u>IronHide:</u> Continuar trabajando en la funcionabilidad de deshabilitar clientes.
18 de febrero	<u>IronHide:</u> Finalizar la funcionabilidad de deshabilitar unos o muchos clientes a la vez.

Fuente: (Elaboración propia)

4.2.6 SEMANA 6 (21 DE FEBRERO A 25 DE FEBRERO)

Durante la semana 6, se desarrolló la segunda asignación solicitada, la cual requería utilizar la librería de Redux debido a que es una herramienta para la gestión de estados en apps JavaScript, facilitándome la asignación para almacenar la búsqueda en las diferentes tablas.

Tabla 7. Semana 6, Actividades diarias

Fecha	Actividades
21 de febrero	<u>IronHide:</u> Mantener la búsqueda en la tabla de facturas utilizando Redux.
22 de febrero	<u>IronHide:</u> Revisar y corregir la tabla 'InvoiceDevice' en la base de datos, para que me permita revisar el detalle de las facturas.
23 de febrero	<u>IronHide:</u> Mantener la búsqueda en las tablas de Proformas, cotizaciones y notas de crédito.
24 de febrero	<u>IronHide:</u>

	Se realizaron los cambios solicitados en el pull request de las columnas extras agregadas en tabla de clientes aprobados.
25 de febrero	<u>IronHide:</u> Identificar y corregir las llaves foráneas faltante en la tabla de payment.

Fuente: (Elaboración propia)

4.2.7 SEMANA 7 (28 DE FEBRERO A 4 DE MARZO)

Durante la semana 7, se corrigió el comportamiento de la selección de los clientes a deshabilitar debido a que el comportamiento esperado era poder seleccionar clientes alrededor de todas las paginaciones de la tabla de clientes aprobados, lo cual detecte que la migración de Ant Desing v3 a v4 lo había ocasionado y que era necesarios añadir una nueva propiedad en el componente de la tabla en ambos proyectos, que permitiría la selección de la fila por las diferentes 'paginas' de la tabla.

Tabla 8. Semana 7, Actividades diarias

Fecha	Actividades
28 de febrero	<u>IronHide:</u> Guardar las selecciones de clientes aprobados a la hora de cambiar de página o realizar una nueva búsqueda. Agregarle estilo al pop over de deshabilitación de clientes.
1 de marzo	<u>Optimus:</u> Permitir que las tablas en el proyecto de Optimus guarden la selección de filas de diferentes páginas.
2 de marzo	<u>Optimus:</u> Realizar code review a pull request de otros integrantes del área de IT.
3 de marzo	<u>Optimus:</u> Eliminar la default-case rule acerca de la declaración switch y agregar un default case a todos los switches del proyecto de Optimus.
4 de marzo	<u>IronHide:</u> Reunión con el dueño de la empresa para discutir las siguientes asignaciones en IronHide.

Fuente: (Elaboración propia)

4.2.8 SEMANA 8 (7 DE MARZO A 11 MARZO)

Durante la semana 8, se empezó a desarrollar la funcionabilidad de editar las facturas en IronHide, la cual se dividió en asignaciones más pequeñas debido a que el formulario de las facturas está conformado por varias funcionabilidades. La primera funcionabilidad por trabajar fue la acción

edición de los productos ingresados en la factura, luego añadir más productos a la factura editando y por último la eliminar productos de la factura.

Tabla 9. Semana 8, Actividades diarias

Fecha	Actividades
7 de marzo	<u>IronHide:</u> Empezar a trabajar en la nueva funcionabilidad de editar las facturas en IronHide.
8 de marzo	<u>IronHide:</u> Crear una nueva acción en la APP y un end point en el API para obtener un arreglo de los dispositivos de la factura de clientes.
9 de marzo	<u>IronHide:</u> Continuar trabajando en las acciones de la tabla de productos de la factura y dividir los productos en diferentes arreglos según su acción.
10 de marzo	<u>IronHide:</u> Implementar la funcionabilidad del modal de editar la cantidad de productos en la factura, en caso de incremento de precio del producto se desplegará un input para definir la razón del incremento.
11 de marzo	<u>IronHide:</u> Implementar hotfix, en el cual se corrigió el filtrado ascendente y descendente de la columna "Fecha para el siguiente pago"

Fuente: (Elaboración propia)

4.2.9 SEMANA 9 (14 DE MARZO A 18 MARZO)

La semana se basó en desarrollar la funcionabilidad implementada de editar los productos de la factura en la cual cuando ocurre un cambio de precio en el producto se habilita un input para solicitar la razón del aumento. Lo que requirió realizar un método end point en el API con la finalidad de recibir los cambios del APP y modificarlos registros en la base de datos.

Tabla 10. Semana 9, Actividades diarias

Fecha	Actividades
14 de marzo	<u>IronHide:</u> Implementar nuevo hotfix, en el cual se solicitó revisar la funcionabilidad de salvar la búsqueda para las tablas (Facturas, proformas, cotizaciones y notas de debito), y mantener la página de tabla.
15 de marzo	<u>IronHide:</u> Trabajar en las funcionabilidades de agregar y eliminar los productos en la edición de las facturas.
16 de marzo	<u>IronHide:</u> Realizar cambios solicitados sobre almacenar la paginación y búsqueda.
17 de marzo	<u>IronHide:</u>

	Desarrollar un nuevo end point que permita la edición de las facturas.
18 de marzo	<u>IronHide:</u> Continuar en la implementación de un nuevo end point que pueda editar las facturas por completo.

Fuente: (Elaboración propia)

4.2.10 SEMANA 10 (21 MARZO A 25 DE MARZO)

Se verificaron que los cálculos se estuvieran realizando de manera de correcta ahora de editar la factura. Igualmente se probó el funcionamiento en los diferentes escenarios del formulario y que se estuvieran cumpliendo con todas las validaciones aplicados. En el resto de la semana 10, se desarrolló la funcionabilidad de editar las cotizaciones y proformas.

Tabla 11. Semana 10, Actividades diarias

Fecha	Actividades
21 de marzo	<u>IronHide:</u> Verificar el funcionamiento del detalle de la tabla de movimientos de la factura concuerde después de realizar la edición de la factura.
22 de marzo	<u>IronHide:</u> Empezar a desarrollar la funcionabilidad de editar las proformas, validando que cierto tipo de usuario puede realizar esta acción.
23 de marzo	<u>IronHide:</u> Trabajar en la funcionabilidad de aplicar exención de impuesto en la proforma y poder programar una fecha para la proforma.
24 de marzo	<u>IronHide:</u> Reunión con el dueño de la empresa para aclarar las validaciones realizadas Crear nuevo método end point que reciba los datos editados de la proforma
25 de marzo	<u>IronHide:</u> Desarrollar la funcionabilidad de editar las cotizaciones y crear nuevo método end point en el API para actualiza el registro en base de datos.

Fuente: (Elaboración propia)

4.2.11 SEMANA 11 (28 DE MARZO A 1 DE ABRIL)

El 30 de marzo, se empezó a desarrollar la funcionabilidad de editar las notas de crédito, las cuales están enlazadas con las facturas y sus pagos. En lo cual se tuvo que agregar una nueva columna en la tabla de notas de crédito para identificar si el pago se realizó en lempiras o dólares.

Ilustración 16, Ejemplo de nota de crédito

Fuente: (Elaboración propia)

Tabla 12. Semana 11, Actividades diarias

Fecha	Actividades
28 de marzo	<u>IronHide:</u> Corregir/ aclarar el funcionamiento del input number de "day for payment" para cotizaciones. Realizar cambios sugeridos en el pull request de la edición de las facturas.
29 de marzo	<u>IronHide:</u> Realizar cambios para el pull request de proformas y cotizaciones y verificar que toda la funcionabilidad de cotizaciones.
30 de marzo	<u>IronHide:</u> Empezar a desarrollar la funcionabilidad de editar notas de crédito: - Obtener un record de las notas de créditos a la hora de la edición y colocarlo en su input correspondiente. - Obtener las facturas pendientes del cliente con la cantidad de la nota de crédito que se colocó en la creación y seleccionarla en la tabla.
31 de marzo	<u>IronHide:</u> Condicionar el cambio de lempiras a dólares cuando se seleccione la opción y condicionar la acción cuando sea editar o agregar. Agregar un nuevo campo a la base de datos, para guardar cuando el cliente allá pagado en dólares o en lempiras, al modelo API.
1 de abril	<u>IronHide:</u> Empezar a desarrollar el método end point para realizar la edición de las notas de crédito.

Fuente: (Elaboración propia)

4.2.12 SEMANA 12 (4 DE ABRIL A 8 DE ABRIL)

Durante la semana 12 se retomó el desarrollo de la funcionabilidad de editar las notas de debito en la cual detecte un problema con las relaciones de las tablas 'payment' y 'creditNote' en la base

de datos lo cual se consultó con el gerente de IT cual sería la mejor manera de abordar el problema, se concordó que lo mejor sería agregar una columna a la tabla de payment llamada creditNotelId la cual es una llave foránea del id de la nota de crédito.

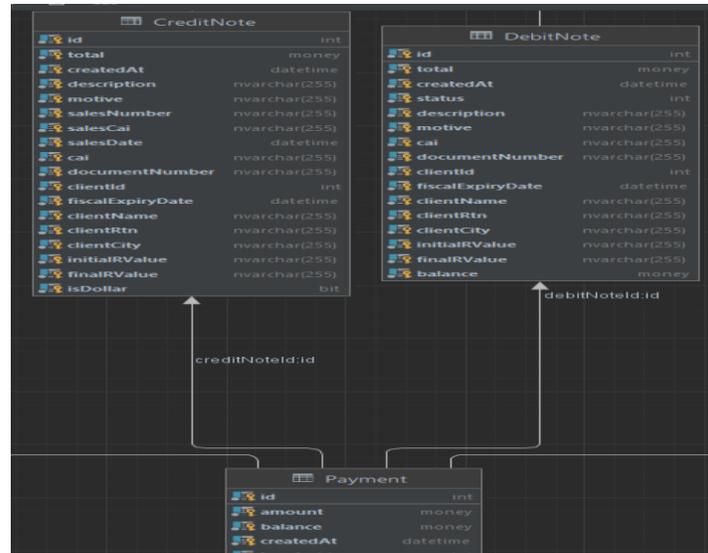


Ilustración 17, Relación de la tabla “payment”

Fuente: (Elaboración propia)

La desestructuración es una propiedad simple que se usa para hacer que el código sea mucho más claro y legible, principalmente cuando pasamos accesorios en React. Lo cual facilita al desarrollador a asignar las variables y ayuda reducir la cantidad de código utilizado. Lo cual mejora la sostenibilidad y la legibilidad del código.

```

28  renderProductField = () => {
    -   const { props, getIntl } = this;
    -   const { form, currentProduct } = props;
    -   const { getFieldDecorator } = form;
    -   const { product } = currentProduct;
    -   const { id, description } = product;
29+  const { props: { form: { getFieldDecorator }, currentProduct: { product: { id, description } }}, getIntl } = this;
30  return (

```

Ilustración 18, Ejemplo de desestructuración de props.

Fuente: (Elaboración propia)

El 7 de abril se empezó a desarrollar la funcionalidad de notas de débito, de la misma manera que se desarrolló para las notas de débito, debido a que en su momento se utilizaba de la misma manera que las notas de crédito.

Tabla 13. Semana 12, Actividades diarias

Fecha	Actividades
4 de abril	<u>IronHide:</u> Reunión con el gerente de IT para discutir un problema que detecte en el cual necesitaría crear una relación en la base de datos entre los pagos y las notas de crédito.
5 de abril	<u>IronHil:</u> Editar el método end point de agregar las notas de crédito para cuando escojan la opción de dólares, siempre se haga la conversión a lempiras antes de guardar y que se genere el pago en lempiras.
6 de abril	<u>IronHide:</u> Agregar un nuevo campo a la tabla de payment llamada creditNoteld la cual es una llave foránea del id de la nota de crédito. Limpiar el código en los componentes que edite para sea más legible y ordenado.
7 de abril	<u>IronHil:</u> Desarrollar la funcionabilidad de editar las notas de débito: - Agregar la acción de edición - Obtener el récord de la nota de débito y hacer que muestre donde corresponde cada dato.
8 de abril	<u>IronHil:</u> Empezar a desarrollar el método end point para realizar la edición de las notas de débito.

Fuente: (Elaboración propia)

4.2.13 SEMANA 13 (11 DE ABRIL DE 15 DE ABRIL)

La semana comenzó con una reunión con el dueño de la empresa y el jefe inmediato de IronHide, en la cual se me solicito posponer la funcionabilidad de la edición de las notas de debito. Solicito deslazarla las notas de débito con las facturas, ya que las notas de débito lo querían manejar como un documento individual sin ninguna relación a las facturas y que tuviera sus propios pagos como las facturas. Lo que llevo a modificar el formulario de las notas de débito, el método end point y eliminar las relaciones de la tabla de facturas con la de notas débito.

Se habilito la opción de realizar pagos de notas de débito, ya que el desarrollador anterior de IronHide solo agrego la opción pero no finalizo la funcionabilidad. Lo cual tuve que verificar que los cálculos se realizaran de manera de correcta en la base de datos.

Tabla 14. Semana 13, Actividades diarias

Fecha	Actividades
11 de abril	<u>IronHide:</u> Reunión con el dueño de la empresa y el jefe inmediato para posponer la funcionabilidad de las notas de débito, ya que solicitaron cambios al formulario de las notas de débito y que se desenlazaras de las facturas.
12 de abril	<u>IronHide:</u> Trabajar en el nuevo formulario para las notas de débito el cual se desvincularía a las facturas y modificar el método end point para agregar las notas de débito.
13 de abril	<u>IronHide:</u> Habilitar la opción de pagar notas de débito en el formulario de pagos.
14 de abril	Feriado Nacional
15 de abril	Feriado Nacional

Fuente: (Elaboración propia)

4.2.14 SEMANA 14 (18 DE ABRIL A 22 DE ABRIL)

En la semana 14, se remoto el desarrollo de la edición de las notas de débito después de realizar los cambios solicitados por el dueño de la empresa.



Seguridad Móvil GPS
COL. TARA BLOQUE 11 CASA 9, SAN PEDRO SULA, HONDURAS
www.smovilgps.com

Estado de Cuenta

Cliente: **Ejemplo**
Representante: Sr. Test
RTN:
Dirección:
Ciudad: San Pedro Sula, Cortés
Fecha: 22/04/2022
Saldo hasta la fecha: L 16,536.80 / \$ 682.30

Factura 000-001-01-00022807
Fecha de facturación 29/03/2022
Fecha maxima de pago 13/04/2022
Días de mora 0
Tasa de cambio 24

Codigo	Descripción	Cantidad	Precio	Total
E0002	Unidad localización GPS portátil	2	\$ 109.00	\$ 218.00
A0003	Cámara de seguridad	3	\$ 55.00	\$ 165.00
A0010	Caja magnética portátil	1	\$ 15.00	\$ 15.00
A0007	Arnés	1	\$ 20.00	\$ 20.00
Balance	L 11,536.80		\$ 480.70	

Nota de débito 001-001-01-00000014
Fecha de facturación 11/04/2022
Motivo Cheque rebotado

Balance L 5,000.00 \$ 201.80

Ilustración 19, Ejemplo estado de cuenta generado.

Fuente: (Elaboración propia)

La siguiente asignación que determino fue de enlazar los estados de cuenta de las facturas y notas de debito en un mismo documento generado por la aplicación, en el cual se secciono para dividir las notas de debito y las facturas como se muestra en *Ilustración 19*, ahorrando a los miembros del área de contaduría generar varios documentos para el cliente.

Tabla 15. Semana 14, Actividades diarias

Fecha	Actividades
18 de abril	Por motivos personales se solito el día libre.
19 de abril	<u>IronHide:</u> Retomar el desarrollo de la funcionabilidad de edición de las notas de débito y verificar que funcione correctamente.
20 de abril	<u>IronHide:</u> Enlazar los estados de cuentas entre las facturas y notas de débito. Agregar una nueva columna en la tabla de estados de cuenta, la cual sirve para diferenciar el tipo del documento.
21 de abril	<u>IronHide:</u> Trabajar en la función del backend para que genere las notas de débito en el documento para enviarlo por email.
22 de abril	<u>IronHide:</u> Corregir la función de vista previa del documento, ya que no tenía el comportamiento deseado y editar la función de descargar el documento para que funcione con las notas de débito.

Fuente: (Elaboración propia)

4.2.15 SEMANA 15 (25 DE ABRIL A 29 DE ABRIL)

Para concluir con las asignaciones solicitadas para el proyecto de IronHide, se habilito la opción de exención de impuesto para las cotizaciones.

Durante el resto de esta semana me asignaron realizar mock api para las pruebas de Cypress en el proyecto de Optimus, los cuales son utiles para mejorar la velocidad de ejecución de las pruebas Cypress, ya que crean atajos en las pruebas, y tener una gestión de datos simple, haciendo uso de datos mockeados.

Los datos mockeados, son datos que puede que no provengan de un backend o de un servicio o, si provienen de ellos, igualmente son considerados datos "falsos" ya que los escribimos a nuestro antojo, no son datos reales que provienen de la propia aplicación, y se utilizan para el propio desarrollo o automatización de pruebas.

```

8
9
10+ it("Should navigate to sims", () => {
11+   const carriersIdentifier = "getCarriers";
12+   cy.mockGetCarriers([], carriersIdentifier);
13+   cy.navigate("submenu-admin", "/sims", "menu-sims");
14+   cy.waitOn(carriersIdentifier);
15+ });
16
17+ it("Should click Carrier tab in Sims page", () => {
18+   const carriersIdentifier = "getCarriers";
19+   const carriersCountIdentifier = "getCountCarriers";
20+   cy.fixture("/carriers/getCarriers.json").then((response) => {
21+     cy.mockGetCarriers(response, carriersIdentifier);
22+   });
23+   cy.fixture("/carriers/carrier.json").then((response) => {
24+     cy.mockGetCountCarriers(
25+       response.getCountCarriers,
26+       carriersCountIdentifier
27+     );
28+   });
29+   cy.get('[data-testid="carrier-tab"]').should("be.visible").click();
30+   cy.waitOn(carriersCountIdentifier);
31+   cy.waitOn(carriersIdentifier);
32+ });

```

Ilustración 20, Ejemplo de mock API (Carriers).

Fuente: (Elaboración propia)

Con el objetivo de optimizar la ejecución de las pruebas ya que nos permitirán establecer un contexto adecuado a cada escenario de prueba, sin la necesidad de realizar acciones reales a la base de datos.

Tabla 16. Semana 15, Actividades diarias

Fecha	Actividades
25 de abril	<u>IronHide:</u> Agregar opción de habilitar exención de impuestos a las cotizaciones.
26 de abril	<u>Optimus:</u> Llamada vía Slack con un miembro del equipo, el cual me explico como implementar un mock API para desarrollar el mock API del módulo Carriers.
27 de abril	<u>Optimus:</u> Aclarar dudas con un miembro del equipo acerca de cuál es la mejor manera de realizar un mock API. Realizar los cambios que me solicito para el mock api de Carriers.
28 de abril	<u>Optimus:</u> Desarrollar el mock api para la prueba Cypress del módulo de Devices
29 de abril	<u>Optimus:</u> Desarrollar el mock api para la prueba Cypress del módulo de Clients.

Fuente: (Elaboración propia)

4.2.16 SEMANA 16 (2 DE MAYO A 6 MAYO)

Durante la semana 16 se dedicó más que todo para realizar cambios que se solicitaron por parte de los encargados a los pull request de los mock api de devices y clients. Sin embargo se realizó un hotfix para revisar por que la no se estaba guardando la paginación de la tabla de cotizaciones

en Redux. Lo cual detecte que por culpa de un mal rebase, se habían borrado un par de líneas de código en el reducer del módulo de las cotizaciones.

```
emailExist = debounce(async ({ rule, value }) => {
  const { data } = this.props;
  if (data && data.user && data.user.email === value) return value;
  await this.emailVerification(rule, value);
}, 1500);

doesEmailExist = (rule, value) => this.emailExist({ rule, value });
```

Ilustración 21, Implementación del método debounce.

Fuente: (Elaboración propia)

Se implementa el método debounce como se muestra *Ilustración 21*, con el fin de facilitar el mock api y evitar que el backend reciba request innecesarios a la hora de validar el email y usuario del cliente.

Tabla 17. Semana 16, Actividades diarias

Fecha	Actividades
2 de mayo	<u>Optimus:</u> Implementar el método debounce para el formulario de clients-admins en los campos de email y usuario, ya que estos campos se validan con los registros de la base de datos.
3 de mayo	<u>Optimus:</u> Finalizar el nuevo enfoque para el mock api de la prueba Cypress del módulo client-admins.
4 de mayo	<u>Optimus:</u> Agregar a la prueba de devices, la prueba Cypress al formulario de client-devices y revisar por que falla la ejecución de las pruebas en GitHub.
5 de mayo	<u>IronHide:</u> Llamada vía Slack con el jefe de inmediato de IronHide para notificarme que realizara un hotfix ya que habían reportado que la paginación de la tabla de cotización estaba fallando.
6 de mayo	<u>Optimus:</u> Realizar los cambios solicitado del pull request de la implementación del mock api de clients-admin y devices.

Fuente: (Elaboración propia)

4.2.17 SEMANA 17 (9 DE MAYO A 13 DE MAYO)

La semana 17 se basó en desarrollar los mocks api para los módulos de Sims, User y Edit Tank Configuration, de igual forma en realizar los cambios solicitados por parte de los encargados a los pull request de los mocks.

* Cliente: Clientes Varios * Tasa de Cambio: 24.5569 * Tipo: Contado **Crédito** * Dias para Pagar: 15

* Cliente: * RTN: * Representante:
 Un cliente es requerido

* Departamento: * Ciudad: * Dirección:

Producto:	Cantidad:	Precio:	Total:
<input type="text"/>	1	\$ 0	\$ 0.00

Ilustración 22, Ejemplo de formulario con opción “clientes varios”.

Fuente: (Elaboración propia)

Sin embargo se realizó un hotfix para IronHide, ya que no se había tomado en cuenta la funcionalidad de “Clientes varios” los permite al usuario definir un nuevo cliente en el mismo formulario de la factura, proforma y cotizaciones. Lo que conlleva a habilitar la opción para el servidor de desarrollo y revisar que el formulario cargara bien los datos a la hora de editar una factura que contara con opción de “clientes varios”.

Tabla 18. Semana 17, Actividades diarias

Fecha	Actividades
9 de mayo	<u>IronHide:</u> Realizar un hotfix para IronHide (La opción de "Clientes Varios" estaba deshabilitada y no la había tomado en cuenta a la hora de editar las Cotizaciones, proformas y facturas).
10 de mayo	<u>Optimus:</u> Desarrollar el mock api para la prueba Cypress del módulo de Sims.
11 de mayo	<u>Optimus:</u> Desarrollar el mock api para la prueba Cypress del módulo de User.
12 de mayo	<u>Optimus:</u> Reestructurar la prueba de Cypress del módulo “Edit Tank Configuration” y realizar el mock api del módulo.
13 de mayo	<u>Optimus:</u> Realizar los cambios sugerido de los pull request de mock api Users y sims.

Fuente: (Elaboración propia)

4.2.18 SEMANA 18 (16 DE MAYO A 20 DE MAYO)

Durante la semana 18, el gerente de IT me solicito enlistar en una tarjeta de la plataforma de Trello los gráficos de cada reporte. Se definió que después de actualizar el package de Bizcharts los

gráficos que salen en la lista se les deberán revisar las propiedades que se utilizaron han pasado a ser incompatibles con la v4 y sustituir por las nuevas propiedades y componentes.

Tabla 19. Semana 18, Actividades diarias

Fecha	Actividades
16 de mayo	<u>Optimus:</u> Enlistar los gráficos de cada uno de los reportes de la aplicación Investigar y leer la documentación de bizcharts v4 acerca los componentes y props que no ya no son compatible con la v4.
17 de mayo	Por motivos de salud se solicitó el día libre.
18 de mayo	<u>Optimus:</u> Actualizar el paquete de bizcharts v3 a v4.
19 y 20 de mayo	<u>Optimus:</u> Empezar a eliminar o sustituir los componentes y props que ya no son compatibles con la v4 en el gráfico del reporte de velocidad.

Fuente: (Elaboración propia)

4.2.19 SEMANA 19 (23 DE MAYO A 27 DE MAYO)

La semana 19, se basó en eliminar o sustituir los componentes y props que ya no son compatibles con la v4 para todos los gráficos que se enlisto en la tarjeta de Trello. Sin embargo detecte y corregí el problema de rendimiento en la implementación de bizcharts v3 la cual ocasionaba que el explorador del usuario se pusiera lento.

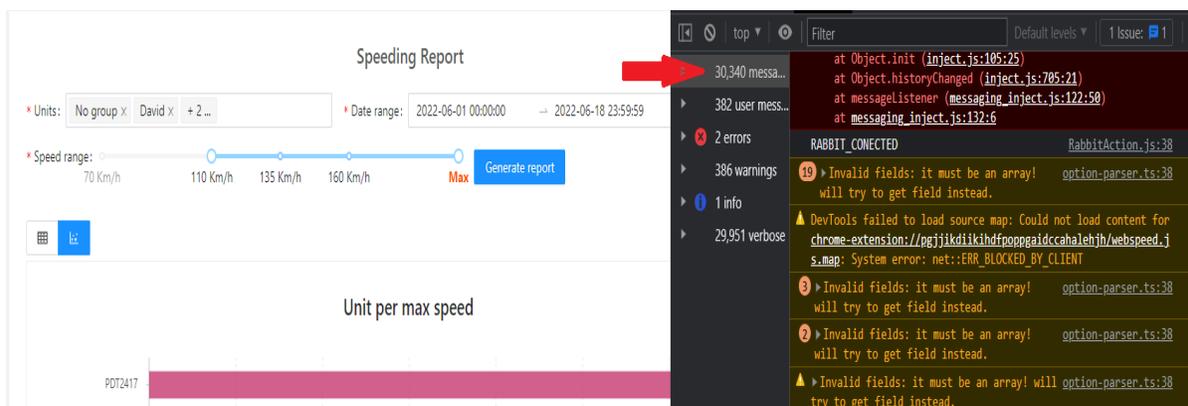


Ilustración 23, Problema de implementación bizcharts v3

Fuente: (Elaboración propia)

Este problema se ocasiono en la migración del componente de clase a componente funcional donde se encontraba el grafico de velocidad. En el cual se implementó un hook useEffect con la

dependencia de los “grupos de los dispositivos” el cual estaba en constante cambio lo que realizaba que el explorador se sobre cargar.



Ilustración 24, Optimización e implementación bizcharts v4

Fuente: (Elaboración propia)

Sin embargo la documentación del bizcharts menciona que la actualización v4 se mejoró el rendimiento y optimaciones con las nuevas propiedades que desarrollaron como se muestra en la *Ilustración 24*.

Tabla 20. Semana 19, Actividades diarias

Fecha	Actividades
23 de mayo	<u>Optimus:</u> Resolver un problema de rendimiento que identifique en la implementación del grafico del reporte de velocidad.
24 de mayo	<u>Optimus:</u> Empezar a eliminar o sustituir los componentes y props que ya no son compatibles con la v4 en el gráfico de reporte combustible.
25 de mayo	<u>Optimus:</u> Trabajar en eliminar o sustituir los componentes y props que ya no son compatibles con la v4 en los gráficos del reporte de manejo: <ul style="list-style-type: none"> • Distancia/Tiempo/Velocidad por unidad • Distancia/Tiempo/Velocidad por día • Distancia/Tiempo/Velocidad de todas las unidades.
26 de mayo	<u>Optimus:</u> - Actualizar componentes y props que ya no son compatibles con la v4 en los gráficos del Reporte de parqueo: <ul style="list-style-type: none"> • Unidad por Tiempo • Total paradas por día • Gráfico general por unidad Y el Reporte de control de Combustible: <ul style="list-style-type: none"> • Volumen • Valor
27 de mayo	<u>Optimus:</u>

	<p>Empezar a eliminar o sustituir los componentes y props que ya no son compatibles con la v4 en el gráfico de los widgets:</p> <ul style="list-style-type: none"> • Puntos de Interés • Velocidad top 5 • Nivel de Batería • Estado actual
--	---

Fuente: (Elaboración propia)

4.2.20 SEMANA 20 (30 DE MAYO A 3 DE JUNIO)

La semana 20 se comenzó realizando el mockeado del módulo de Login, asimismo se trabajó en el mockear el comando de login que implementaron con el fin de utilizarlo en la mayoría de las pruebas y evitar copiar código. El comando de login permite utilizar las variables de entorno para poder realizar los que es el ingreso a la aplicación con las credenciales que sean necesarias para las pruebas. Con lo cual el mockear lograría optimizar el tiempo de ejecución de las pruebas de Cypress teniendo en cuenta las diferentes credenciales que cuenta la aplicación.

```

Cypress.Commands.add('login', (username, password) => {
  cy.visit('/login');
  cy.get('[data-testid=form-login]').should('be.visible');
  cy.get('[data-testid=input-username]').should('be.visible').type(username)
  cy.get('[data-testid=input-password]').should('be.visible').type(`${password}`);
  cy.wait(10000);
  cy.url().should('not.include', '/login');
});

27+ Cypress.Commands.add("login", (username, password) => {
28+   const postIdentifier = "postAccountLogin";
29+   const getAccountAlerts = "getAccountAlerts";
30+   const getAccountSession = "getAccountSession";
31+   const getMapDevices = "getMapDevices";
32+   const postS3Bucket = "postS3Bucket";
33+
34+   let responseJson = "";
35+   if (username === "gps") {
36+     responseJson = "/login/postGPSResponse.json";
37+   } else if (username === "admin") {
38+     responseJson = "/login/postMasterResponse.json";
39+   } else {
40+     responseJson = "/login/postResponse.json";
41+   }
42+   cy.mockPostAccountLogin({ fixture: responseJson }, postIdentifier);
43+   cy.mockPostS3Buckets({}, postS3Bucket);
44+   cy.mockGetAccountSession({ status: 200 }, getAccountSession);
45+   cy.mockGetAccountAlerts(
46+     { fixture: "/login/getReponseAlerts.json" },
47+     getAccountAlerts
48+   );
49+   cy.mockGetMapDevices(
50+     { fixture: "/report/getMapDevicesResponse.json" },
51+     getMapDevices
52+   );
53+   cy.visit("/login");
54+   cy.get("[data-testid=form-login]").should("be.visible");
55+   cy.get("[data-testid=input-username]").should("be.visible").type(username);
56+   cy.get("[data-testid=input-password]").should("be.visible")
57+     .type(`${password}`);
58+   cy.waitOn([
59+     getAccountAlerts,
60+     getAccountSession,
61+     postS3Bucket
62+   ]);

```

Ilustración 25, Mock del comando de login

Fuente: (Elaboración propia)

Asimismo se me solicitó implementar un nuevo método waitOn el cual permitiera recibir un arreglo de identificadores, con el fin de mantener un código más legible en las pruebas de

Cypress. El cual hace uso del método wait de Cypress el cual espera una cantidad de milisegundos o espera a que se resuelva un request con un identificador antes de pasar al siguiente comando.

```

cy.waitOn(getTripIdentifier);
cy.waitOn(getCancelTripIdentifier);
cy.waitOn(patchTripIdentifier);
});

cy.waitOn([
  getTripIdentifier,
  getCancelTripIdentifier,
  patchTripIdentifier,
]);
});

```

Ilustración 26, Ejemplo de nuevo método waitOn

Fuente: (Elaboración propia)

Tabla 21. Semana 20, Actividades diarias

Fecha	Actividades
30 de mayo	<u>Optimus:</u> Desarrollar el mock api para la prueba Cypress del módulo de Login de la aplicación y comando de la login Cypress que utilizan todas las demás pruebas.
1 de junio	<u>Optimus:</u> Implementar un nuevo metodo waitOn que permita enviar los identifiers de los request como un arreglo. Descubrir una nueva sintaxis de implementar los fixtures para mockear las respuestas de los request.
2 de junio	<u>Optimus:</u> Probar cada una de las pruebas de Cypress y enlistar las pruebas que falla con la razón del porque fallan, igualmente enlistar las pruebas que aún faltan por mockear.
3 de junio	<u>IronHide:</u> Llamada vía Slack con el jefe de inmediato de IronHide, para mostrarme los pasos a seguir cuando un miembro del área de contabilidad me solicite revertir un pago de alguna factura. Realizar los cambios sugeridos para el pull request de bizcharts, los cuales indicaban eliminar la interpolación innecesaria en el código y sustituir los headers tag de html con typography levels de ant design.
4 de junio	<u>Optimus:</u> Realizar dos hotfix: <ol style="list-style-type: none"> 1. Corregir el mal comportamiento del slider el sobre el anchor de la geocercas. 2. Los eventos personalizados de las entradas digitales, no se mostraban con la personalización configurada, ni con el icono configurado definido por el usuario.

Fuente: (Elaboración propia)

4.2.21 SEMANA 21 (6 DE JUNIO A 10 DE JUNIO)

La semana 21, se basó en la gran parte en realizar los mock api de los reportes faltantes los cuales eran:

- Command History
- Current State
- Distance
- Driving
- Gas Consumption
- Geofences
- Idle
- Maintenance
- Parking
- Photos
- Point of Interest
- Speeding
- Temperature
- Units per-Client

Asimismo se trabajó en realizar los mock api para los módulos faltantes:

- Edit Device Massive
- Themometer Configuration
- Top Menu Profile
- Google Earth
- Gas Station

Con el fin de optimizar el tiempo de ejecución de las pruebas de Cypress, debido a que cuando se crea un pull request GitHub ejecuta todas las pruebas para revisar la funcionalidad de la aplicación, con lo cual si llega a fallar una prueba no se podría realizar "merge" al servidor de desarrollo, con lo cual el desarrollador tendría que esperar nuevamente que se ejecuten todas las pruebas lo que puede llegar a mucho tiempo, debido a la cantidad de pruebas y acciones que realizan cada uno de ellos.

Tabla 22. Semana 21, Actividades diarias

Fecha	Actividades
6 de junio	<u>Optimus:</u> Trabajar en el bug de Restablecer mantenimiento del GPS, validar que el device y el maintenancelid no se envíen indefinidos a rabbitMQ.
7 de junio	<u>Optimus:</u> Mover la opción "Street view" a la parte inferior izquierda del mapa. Realizar los cambios sugerido de los pull request de mock api Login.
8 de junio	<u>Optimus:</u>

	<p>Crear una tarjeta en la plataforma de trello para enlistar los reportes que tenga la prueba de Cypress sin mockear.</p> <p>Realizar los mock api de los reportes enlistados en trello.</p>
9 y 10 de junio	<p><u>Optimus:</u></p> <p>Corregir un problema que detecte con el modal de agregar y editar la configuración del termómetro.</p> <p>Realizar los mock api de los módulos faltantes</p>

Fuente: (Elaboración propia)

4.2.22 SEMANA 22 (13 DE JUNIO A 17 DE JUNIO)

Durante la semana 22, se finalizó los mock api de los módulos faltantes enlistado la semana anterior y se corrigió los mock api que presentaban fallan en la ejecución, lo que concluyo con la implementación de mock api para todas las pruebas Cypress de la aplicación de Optimus GPS, logrando una reducción destacada del tiempo de ejecución de las pruebas.

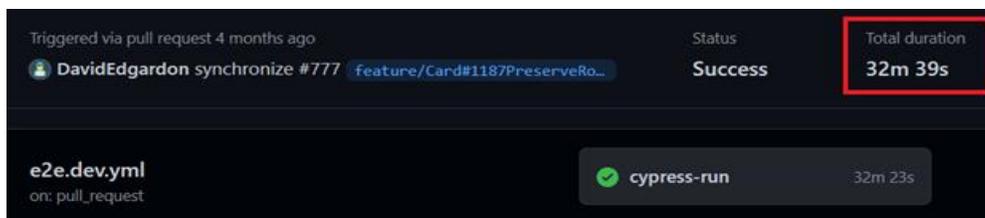


Ilustración 27, Tiempo de ejecución (Sin mockear)

Fuente: (Elaboración propia)

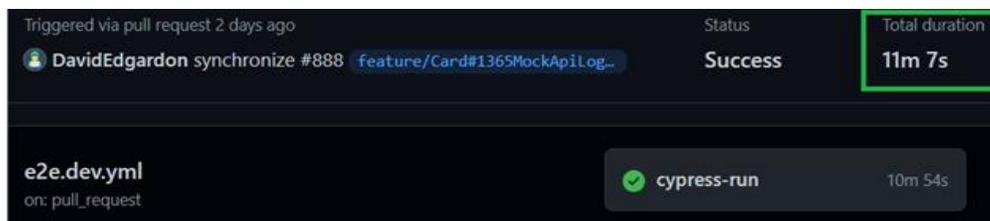


Ilustración 28, Tiempo de ejecución (Mockeados)

Fuente: (Elaboración propia)

Como se muestra en la *Ilustración 27* y *28*, la comparativa es bastante evidente con respecto al tiempo de ejecución de las pruebas. De igual forma se logró eliminar el proceso de limpiado de las tablas de cada módulo después de la ejecución de las pruebas Cypress, debido a que las pruebas Cypress mockeadas se falsifican las respuestas del back end evitando que se realicen solicitudes reales al servidor y a la base de datos.



Ilustración 29, Plugin Cypress Screenshot Upload

Fuente: (Elaboración propia)

Para finalizar la semana se tuvo una reunión con el gerente de IT y un miembro del equipo de desarrollo para implementar un nuevo plugin para Slack el cual tenga la función de subir screenshot de las pruebas que fallen durante la ejecución en GitHub como se muestra en la *ilustración 29*.

Tabla 23. Semana 22, Actividades diarias

Fecha	Actividades
13 de junio	<u>Optimus:</u> Finalizar los mock api de los módulos faltantes.
14 de junio	<u>Optimus:</u> Realizar más cambios para el pull request de los gráficos de bizcharts, los cuales indicaban, revisar y solucionar las escalas de los gráficos: <ul style="list-style-type: none"> • Unidad por velocidad máxima • Reporte de manejo por vehículo • Reporte de punto de interés (Visitas general y por vehículo) • Reporte de temperatura.
15 y 16 de junio	<u>Optimus:</u> Corregir los mock api de los módulos de Cypress que estaban fallando: <ul style="list-style-type: none"> • Trips • Device • Fuel Control • ModelProfile • Carriers • ShareDevices • Extra Information Form • Vehicle Information Form
17 de junio	<u>Optimus:</u> Realizar un nuevo hotfix, acerca de la barra de búsqueda la cual estaba funcionando para la tabla de staff.

	Reunión con el gerente de IT y otro integrante para integrar un nuevo plugin de Slack es cual mandara screen shot cuando una prueba de Cypress falle su ejecución en GitHub.
--	--

Fuente: (Elaboración propia)

4.3 DIAGRAMA DE GANTT

Asignacion	Fecha de inicio	Fecha de finalización	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8	Semana 9	Semana 10	Semana 11	Semana 12	Semana 13	Semana 14	Semana 15	Semana 16	Semana 17	Semana 18	Semana 19	Semana 20	Semana 21	Semana 22
Configuración del ambiente de trabajo Optimus	17.01.2022	18.01.2022	█																					
Funcionabilidad de compartir foto vía correo electrónico	19.01.2022	21.01.2022	█	█																				
Migración de componentes y creación de pruebas Cypress	24.01.2022	31.01.2022		█	█																			
Estilo single sheet para reporte de control de combustible	01.02.2022	04.02.2022			█	█																		
Realizar cambios en pull request	07.02.2022	07.02.2022				█																		
Corregir pruebas de Integracion	08.02.2022	08.02.2022					█																	
Configuración del ambiente de trabajo IronHide	09.02.2022	11.02.2022						█																
Deshabilitación de uno o muchos clientes.	14.02.2022	18.02.2022							█															
Mantener la búsqueda y paginación diferentes tablas de IronHide	21.02.2022	23.02.2022								█														
Tabla de Facturas	21.01.2022	22.01.2022																						
Tabla de Proformas, cotizacion y notas de debito	23.01.2022	23.01.2022																						
Realizar cambios en pull request	24.01.2022	24.01.2022																						
Cambios en la base de datos de IronHide	25.01.2022	25.01.2022																						
Guardar la seleccion de filas entre la paginacion de la tabla	28.01.2022	01.03.2022																						
Code Review	02.03.2022	02.03.2022																						
Eliminar la default-case rule	03.03.2022	04.03.2022																						
Funcionabilidad de editar facturas.	07.03.2022	21.03.2022																						
Funcionabilidad de editar proformas	22.03.2022	24.03.2022																						
Funcionabilidad de editar cotizaciones	25.03.2022	29.03.2022																						
Funcionabilidad de editar notas de credito	30.03.2022	06.04.2022																						
Funcionabilidad de editar notas de debito	07.04.2022	08.04.2022																						
Desenlazar las notas de débito y nuevas funcionabilidades	11.04.2022	20.04.2022																						
Generar estados de cuentas	20.04.2022	25.04.2022																						
Desarrollar mock API para los módulos Carriers, Devices y Clients	25.04.2022	04.05.2022																						
Desarrollar mock API para los módulos Sims, Users y Edit Tank Configuration	10.05.2022	13.05.2022																						
Actualizar paquete de Bizcharts v4 y los gráficos de los reportes de la app	16.05.2022	27.05.2022																						
Desarrollar mock API para el login y el comando de login.	30.05.2022	02.06.2022																						
Enlistar las pruebas de Cypress que fallaban y faltaban mockear	03.06.2022	03.06.2022																						
Realizar hotfix del slider de geocercas y eventos personalizados	04.06.2022	04.06.2022																						
Mover la opcion del street view del mapa	06.06.2022	07.06.2022																						
Corregir bug de los diferentes modales	07.06.2022	10.06.2022																						
Realizar hotfix del slider de geocercas y eventos personalizados	13.01.2022	17.01.2022																						

Fuente: (Elaboración propia)

V. CONCLUSIONES

1. En conclusión la práctica profesional me ayudo a desarrollar y utilizar mis conocimientos adquiridos a lo largo de mi carrera, a la vez mis habilidades y aptitudes encontrando la mejor solución a los determinados problemas y para desarrollar nuevas funcionalidades en los distintos proyectos de la empresa.
2. Se desarrollo nuevas funcionalidades en el módulo de facturación para la edición de facturas, proformas, cotizaciones, notas de crédito y débito. Lo que conlleva a solventar que estas acciones se realizaran de forma manual a la base de datos. A su vez se enlazo los estados de cuenta entre facturas y notas de débito generando un documento seccionado con ambas partes, el cual brinda la opción de ser compartido vía correo electrónico.
3. Se implementaron pruebas de integración utilizando Cypress para verificar la funcionalidad de los diferentes componentes que forman parte de la aplicación web Optimus. Esto logrando la simulación de los resultados esperados de las peticiones para evitar realizar operaciones reales o complejas y reduciendo el tiempo de ejecución a 11 minutos.
4. Se realizo servicio de mantenimiento tanto a las aplicaciones internas como externas de la empresa, actualizando paquetes y solucionando bugs encontrados por parte de los clientes, logrando una optimización en el rendimiento y una mejor experiencia de usuario.

VI. RECOMENDACIONES

Se recomienda a la empresa Seguridad Móvil GPS crear y definir un equipo respectivo al área de pruebas, los cuales se responsabilicen de declarar la calidad final del código y realizar las pruebas integraciones de las nuevas funcionalidades y presentes. Lo cual agilizaría el proceso de la asignación para llegar a producción. Definir estándares de buenas prácticas los cuales se deban seguir al momento de codificar en las diferentes aplicaciones de la empresa con el fin de evitar problemas de funcionamiento y entendimientos para los demás desarrolladores de la empresa.

En la actualidad, el desarrollo de aplicaciones está ganando gran popularidad en el mercado laboral, lo que lleva al uso de nuevas tecnologías que crecen día a día. Se recomienda a los estudiantes de la carrera de Sistemas Computacionales a optar por metodologías de autoaprendizaje debido que la formación continua es imprescindible en la profesión, lo cual se recomienda el autoaprendizaje para mantener al tanto de las nuevas tecnologías, reforzar las habilidades y lograr ser una persona competitiva en el ámbito laboral y de emprendimiento.

De igual forma se recomienda a los estudiantes priorizar el aprendizaje de buenas practicas a la hora de codificar y desarrollar habilidades de resolución de problemas, ya que es fundamental en el ámbito laboral debido a que se debe tener la capacidad de identificar los problemas, tomar las medidas lógicas para encontrar una solución deseada y ejecutar dicha solución.

Se recomienda a la universidad implementar más programas o métodos de interacción con las organizaciones para que el estudiante se vaya familiarizando con escenarios reales dentro de una empresa, ya que solo se cuenta con una clase en la se interactúa con una empresa externa a lo largo de la carrera universitaria.

BIBLIOGRAFÍA

Aggarwal, S. (2018). *Modern Web-Development using ReactJS*. 5(1), 5.

Ant Design—The world's second most popular React UI framework. (2022). <https://ant.design/>

Bizcharts. (2022). <https://bizcharts.taobao.com/>

Caspers, M. K. (2017). *Rich Internet Applications w/HTML and Javascript*. 36.

Cypress. (2022). JavaScript End to End Testing Framework | Cypress.io Testing Tools.
<https://www.cypress.io/>

DataGrip: The Cross-Platform IDE for Databases & SQL by JetBrains. (2022). JetBrains.
<https://www.jetbrains.com/datagrip/>

ESLint. (2022). ESLint - Pluggable JavaScript Linter. <https://eslint.org/>

Git. (2022). <https://git-scm.com/>

GitHub. (2022). GitHub. <https://github.com>

GitKraken. (2022). <https://www.gitkraken.com>

LoopBack. (2021). <https://loopback.io/>

Meyer, E. A. (2006). *CSS: The Definitive Guide: The Definitive Guide*. O'Reilly Media, Inc.

Node.js. (2021). <https://nodejs.org/en/>

Npm. (2022). <https://www.npmjs.com/>

PuTTY. (2022). <https://www.putty.org/>

React – A JavaScript library for building user interfaces. (2022). <https://reactjs.org/>

Sass. (2022). <https://sass-lang.com/>

Slack. (2022). <https://slack.com/intl/es-la/>

Trello. (2022). <https://trello.com>

Visual Studio Code. (2022). <https://code.visualstudio.com/>