



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PRÁCTICA PROFESIONAL

IMPACT MOBILE

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTADO POR:

11441141 JUAN CARLOS O'HARA LAÍNEZ

ASESOR: LIC. TANIA LUCILA MEZA

CAMPUS TEGUCIGALPA; MARZO, 2020

LISTA DE SIGLAS Y GLOSARIO

API Application Programming Interface

CSS Cascading Style Sheets

HTML Hyper Text Markup Language

SAGE Semi-Automatic Ground Environment

SQL Structured Query Language

UI User Interface

VCS Version Control System

Back-end: "La parte de un sistema computacional o aplicación a la cual no tiene acceso el usuario, es responsable de almacenar y manipular datos." (Front End: Definition of Front End by Lexico, s.f.)

Endpoint: El punto de comunicación entre un sistema y una API. (API Endpoints - What Are They? Why Do They Matter?, s.f.)

Front-end: "La parte de un sistema computacional o aplicación con la cual el usuario interactúa directamente." (Back End: Definition of Back End by Lexico, s.f.)

Transpilación: Convertir el lenguaje fuente original de un programa a otro lenguaje de programación. También conocido como compilación de fuente a fuente. (gurumoorthyP, arvindpdmn., 2019).

ÍNDICE DE CONTENIDO

I. Introducción	1
II. Generalidades de la empresa	2
2.1. Descripción de la Empresa.....	2
2.2. Descripción del Departamento	2
2.3. Objetivos de Puesto	2
2.3.1. Objetivo General.....	2
2.3.2. Objetivos Específicos	3
III. Marco Teórico.....	4
3.1. Redes y el internet	4
3.2. Software	4
3.3. Aplicaciones Web	5
3.4. Desarrollo Web	5
3.5. Marcos de Trabajo	6
3.6. Angular.....	7
3.7. NodeJS	7
3.8. Herramientas de Desarrollo	8
IV. Desarrollo.....	10
4.1. Descripción del trabajo realizado.....	10
4.1.1. Perfil de Usuario	10
4.1.2. Editar Perfil.....	12
4.1.3. Agregar Usuarios.....	14
4.1.4. Mostrar Usuarios	16

4.1.5. Cambios de Interfaz de Usuario en el perfil de Usuario.....	18
4.1.6. Solicitudes.....	19
4.1.7. Mensajería Instantánea.....	22
4.1.8. Notificaciones.....	23
4.1.9. Mejoras de UI y correcciones de errores.....	24
4.1.10. Desplazamiento infinito virtual.....	27
4.1.11. Motor aritmético.....	28
4.1.12. Módulo de arrastre.....	29
4.1.13. Módulo de Jornadas.....	30
4.1.14. Módulo de Jornada: Cálculo de Hora.....	31
4.1.15. Módulo de Turnos.....	32
4.1.16. Variables de Sistema/Usuarios.....	35
4.2. Cronograma de Actividades.....	36
V. Conclusiones.....	37
VI. Recomendaciones.....	38
Bibliografía.....	39

ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Plantilla base de Perfil.....	11
Ilustración 2 - Vista para visualizar perfil adaptado para el uso de la empresa.....	12
Ilustración 3 - Vista para editar perfil.....	13
Ilustración 4 - Las validaciones del formulario para editar perfil.....	14
Ilustración 5 - Vista para crear usuario.....	15
Ilustración 6 - Alerta de error en el caso de que ya exista un usuario con ese correo.....	15

Ilustración 7 - Alerta de éxito al momento de crear un nuevo usuario.....	16
Ilustración 8 - Vista para ver usuarios.	17
Ilustración 9 - La vista del perfil completo del usuario.....	17
Ilustración 10 – Nuevo UI de perfil de usuario.	18
Ilustración 11 – Nuevo menú para usuarios	19
Ilustración 12 – Componente de solicitudes.	20
Ilustración 13 – Agregar nueva solicitud.....	21
Ilustración 14 – Ver solicitud creada.....	22
Ilustración 15 – Mensajería instantánea.	23
Ilustración 16 – Notificaciones.	24
Ilustración 17 - Nuevo UI de Ver lista de usuarios.....	25
Ilustración 18 – Mejora de UI de sesiones.....	25
Ilustración 19 – Asignación de usuarios.	26
Ilustración 20 – Mensaje de elementos vacíos.....	27
Ilustración 21 – Tabla de datos.....	27
Ilustración 22 – Búsqueda en tabla.	28
Ilustración 23 – Motor aritmético.	29
Ilustración 24 – Módulo de Arrastre	30
Ilustración 25 – Resultado en Postman para obtener información de una jornada.....	31
Ilustración 26 – Resultado en Postman de los cálculos de horas de una jornada.	32
Ilustración 27 – Diseño de la base de datos para los turnos.....	33
Ilustración 28 –Resultado de insertar un acceso de un empleado.....	34
Ilustración 29 –Resultado de la suma de los totales de horas trabajadas de un empleado.	34

Ilustración 30 –Diseño de la base de datos para variables de sistema y de usuario..... 35

Ilustración 31 –Respuesta del servidor para crear variables de usuario..... 35

ÍNDICE DE TABLAS

Tabla 1- Diagrama de Gantt 36

RESUMEN EJECUTIVO

El presente informe se entrega como requisito para optar al título de Ingeniero en Sistemas Computacionales y muestra las actividades realizadas durante la práctica profesional realizada en el departamento de desarrollo de la empresa Impact Mobile durante el periodo de octubre del 2019 a marzo del 2020.

En los primero 3 meses en el departamento de desarrollo se trabajó en un sistema de boletas. Este tiene el rol de apoyar al departamento de diseño de Impact Mobile. En este sistema un cliente puede crear solicitudes, ver avances de los trabajos pedidos y hablar con la jefe de diseño y por el lado de departamento de diseño se puede asignar diseñadores a las solicitudes y darles tareas. El front end fue desarrollado con Angular 8, el back end en Javascript con el apoyo de nodeJS y la base de datos fue hecha en MySQL 8.

En los siguientes 2 meses y medio, se empezó a trabajar en un sistema para recursos humanos. Este sistema se encargará de manejar las horas de trabajo, hacer cálculos, crear y turnos entre otros. En este proyecto el front end utiliza la versión 9 de Angular mientras que el back end y la base de datos utiliza las mismas tecnologías del sistema de boletas.

I. INTRODUCCIÓN

La importancia del desarrollo web ha aumentado en los últimos años, ya que muchos si no todos los dispositivos actuales tienen acceso al internet. Su reciente auge se debe a que esta tecnología proporciona grandes ventajas competitivas y a la vez aumentan la productividad en muchas empresas. Donde antes se tenía que instalar un software en cada computadora de un negocio, verificar que estas computadoras cumplieran con los requisitos mínimos para instalar el software o de lo contrario tener que adquirir nuevos componentes para soportarlo, ahora solo se necesitaría un navegador de internet, una conexión a internet y credenciales de acceso. Las empresas disminuirán los gastos para tener estos servicios porque el mantenimiento del software y sus componentes físicos son responsabilidad de otros y el gasto se limitaría a la contratación de la licencia de estos servicios que se ofrecen por el internet.

Estos son los productos que se crean en el departamento de desarrollo de Impact Mobile. En esta área se da la oportunidad de aprender las tendencias en la tecnología web como desarrollar un portal en línea que maneje boletillas de trabajo o un sistema de pago de planillas para el área de recursos humanos. El objetivo de esta práctica es formar habilidades para desarrollar software robusto, seguro e intuitivo.

A continuación, se podrán ver los detalles de la empresa y su departamento de desarrollo, los conocimientos teóricos que se aplicaron como Angular, NodeJS, los sistemas desarrollados detallando sus componentes y por último las conclusiones de la práctica, así como las recomendaciones

II. GENERALIDADES DE LA EMPRESA

2.1. DESCRIPCIÓN DE LA EMPRESA

Impact Mobile es una empresa que brinda servicios de mercadeo y publicidad digital. Sus servicios se enfocan en facilitar a sus clientes la tarea de lanzar campañas publicitarias en medios digitales. Ofrecen servicios a través de mensajes de textos donde los usuarios pueden usar su propia base de datos de clientes para enviar sus promociones. Brindan servicios innovadores como el mercadeo a través de Wi-Fi, el cual consiste en la instalación de sistemas Wi-Fi en lugares con un alto número de personas donde los clientes pueden utilizarlo gratuitamente y así expandir las áreas de mercadeo de la empresa. Entre otros servicios, Impact Mobile se encarga de apoyar a muchas empresas en sus campañas de publicidad dando los servicios adecuados para garantizar el éxito de estos (WebMedia, H., s.f.).

2.2. DESCRIPCIÓN DEL DEPARTAMENTO

El departamento de desarrollo de Impact Mobile se encarga de crear el software adecuado de los servicios que se ofrecen, también se crea software que apoya a los otros departamentos de la empresa. Los desarrolladores utilizan el marco de trabajo de Angular 8.

El equipo está compuesto por cuatro desarrolladores de los cuales uno es el administrador de la base de datos, los otros dos son los desarrolladores del front-end y back-end y un desarrollador mayor que se encarga de dar tareas y supervisar el trabajo.

2.3. OBJETIVOS DE PUESTO

2.3.1. OBJETIVO GENERAL

- Fortalecer el departamento de desarrollo de software de Impact Mobile utilizando nuevas tecnologías web y buenas prácticas laborales para expandir los servicios ofrecidos por la empresa.

2.3.2. OBJETIVOS ESPECÍFICOS

- Cumplir con todas las tareas asignadas utilizando los estándares de calidad de la empresa al desarrollar software poco propenso a errores para garantizar servicios continuos y satisfactorios.
- Documentar el software desarrollado al detallar todas las características y funcionalidades de este para facilitar su uso y mantenimiento.
- Usar nuevas herramientas para el desarrollo de software en cada componente de que se esté realizando para el proyecto de boletas para poder utilizarlas en proyectos futuros de la empresa.
- Implementar las nuevas tendencias en el desarrollo web orientado al front-end al aplicar las nuevas versiones del marco de desarrollo web de Angular con el fin de que Impact Mobile esté al corriente con las nuevas tecnologías de la industria.

III. MARCO TEÓRICO

3.1. REDES Y EL INTERNET

La defensa aérea de los Estados Unidos desarrolló Semi-Automatic Ground Environment (SAGE) para poder conectar muchos de sus satélites en un sistema centralizado. En esa época la arquitectura era un computador central que interconectaba con terminales remotas a través de redes de teléfono. (Prieto, E. A., Lloris, R. A., y Torres, C. J. C., 2006).

Estas redes eran utilizadas principalmente por el departamento de Defensa de los Estados Unidos, universidades y agencias gubernamentales. (Norton, P., 2014). Eventualmente los costos de las computadoras empezaron a disminuir (Prieto, E. A. et al., 2006). lo que permitió la llegada de las redes a los hogares. Esta red es conocida como Internet y permite a sus usuarios conectarse a la World Wide Web en donde se puede compartir correos electrónicos, mensajes y otros medios entre sus usuarios. (Norton, P., 2014).

Internet revolucionó las computadoras y las telecomunicaciones. Es una red global que es conformada por miles de pequeñas redes privadas (O'Brien, J. A., 2006)., que se conectan a través de medios como cables de cobre, fibra óptica, entre otros. (Arenaza, V. A. E., Canseco, R. F. M., y Sosa, M. F., 2007). Cada una de las redes privadas puede operar independientemente del Internet.

Todo el contenido del Internet es transportado por distintos tipos de hardware de diferentes manufacturas y costos, pero la forma de presentar este contenido es por el software desarrollado.

3.2. SOFTWARE

El software es toda aplicación que permite a los usuarios controlar el hardware, crear contenido o calcular datos. Está formado por el código fuente que da instrucciones al hardware y algún tipo de ejecutable que permite interactuar con la aplicación. (Villarreal, D. L. G. S. (2007).

Para poder desarrollar una aplicación se necesita conocer algún tipo de lenguaje de programación para escribir instrucciones que controlan la interfaz que verá el usuario, los datos que se utilizarán y las funcionalidades que tendrá la aplicación. (Villarreal, D. L. G. S. (2007).

3.3. APLICACIONES WEB

El software se ha convertido en la principal herramienta para los negocios modernos. (Jones, C., 2008). Lo cual hace que la calidad del software desarrollado para la web sea bastante alta. Se han establecido cinco atributos de una aplicación web para determinar su calidad, estos son la usabilidad, funcionalidad, confiabilidad, eficiencia y el fácil mantenimiento. También hay que considerar la escalabilidad y el tiempo para llegar al mercado. Entre estos cinco atributos y las últimas consideraciones se determina el éxito de un sitio web. (Pressman, R., 2010).

Una aplicación web exitosa ofrece la oportunidad de hacer negocios a través del Internet, conocido como e-business. Esto significa que una compañía podrá conectar sistemas con los clientes, proveedores o el gobierno a sus quehaceres corporativos. (Cohen, K. D., 2009). Otras utilidades del software es proveer herramientas para mejorar la inteligencia del negocio gracias a los cambios de la tecnología. (Howson, C., 2009).

Las tecnologías del Internet proveen ventajas competitivas a distintas empresas en diferentes formas, ya sea como e-business, mercadeo o la inteligencia de negocio. Con la adopción rápida del Internet en el mundo (Cateora, P. R. X., Gilly, M. C., y Graham, J. L., 2014). se ha generado la necesidad de estudiar especialidades en tecnologías web. (Hallberg, B., 2007). El desarrollo web ha sido la principal fuente de innovación en software de soporte a negocios o entidades internacionales.

3.4. DESARROLLO WEB

El desarrollo web consiste principalmente en el uso del lenguaje de marcado (HTML), el uso de hojas de estilos en cascada (CSS) y el lenguaje de programación JavaScript. Los aspectos del desarrollo web son el diseño web, la programación web y el manejo de base de datos (Christensson, 2013).

El diseño web se enfoca en el uso de HTML y CSS para crear las interfaces de usuarios, la programación web usa lenguajes de programación como JavaScript para la creación de scripts que brindan funcionalidad al contenido de la página web y en el manejo de bases de datos se usa el Lenguaje de Consulta Estructurado (SQL) en el caso de ser una base de datos relacional.

En algunos casos, el uso de HTML, CSS y JavaScript son suficientes para la creación de páginas web sencillas y no hay necesidad de involucrar un sistema de almacenamiento como una base de datos, pero en la actualidad las páginas web ocupan ser más complejas para satisfacer las necesidades del mercado. Por lo tanto, se ha necesitado la creación de herramientas que extienden las funcionalidades de estas tres (Christensson, 2013).

En la industria del desarrollo web se suelen encontrar términos como front-end y back-end, esto se debe a que el desarrollo web necesita dividir las distintas características de una página web para facilitar el desarrollo de estas y para mejorar el mantenimiento de ellas. El front-end consiste en todo lo que involucre la interacción entre usuario y página web, aquí se pueden encontrar los diseñadores web, los diseñadores de interfaces de usuarios y los desarrolladores del front-end como tal. Por el otro lado, el back-end es todo lo que usuario no puede ver como los servidores o las bases de datos, es manejado por los desarrolladores de back-end y también los administradores de bases de datos. Entre esos dos, se encargan de la seguridad, la estructura y el contenido de la página web (Christensson, 2013).

Esta división evita que algún error se propague en las otras áreas y a la vez que estos sean resueltos efectivamente por la facilidad de aislar el problema.

3.5. MARCOS DE TRABAJO

Un marco de trabajo es una herramienta que provee funciones fundamentales para la creación de software. Los marcos de trabajo pueden tener clases, funciones predefinidas y librerías que agilizan la creación de software nuevo ya que los desarrolladores se pueden concentrar en cumplir con los objetivos sin tener que enfrentar problemas que ya tienen soluciones o tareas comunes en el desarrollo web (Difference Between Front-End & Back-End Developer, 2019).

Debido a que los sitios web son por naturaleza escalables, el uso de marcos de trabajos es sustancial para el mantenimiento y la creación de aplicaciones web. Actualmente se disponen de marcos de trabajo en el lado del front-end como Angular, Vue.js, ReactJS entre otros y en el back-end con Node.js, LARAVEL y otros (Server-side web frameworks, n.d.).

3.6. ANGULAR

Angular es parte de los marcos de trabajo para los front-end más populares. Inició como un proyecto personal llamado GetAngular desarrollado por Adam Abrons y Misko Hevery cuando ellos trabajaban para Google. Ambos desarrolladores propusieron reescribir la aplicación Google Feedback en GetAngular con el propósito de disminuir el tiempo de trabajo. La implementación se tardó más de lo que esperaban, sin embargo lograron reducir las 17,000 líneas del código fuente original a solo 1,500 líneas. Después de ese logro, su jefe Brad Green convirtió GetAngular a un nuevo proyecto llamado AngularJS para ser desarrollado y mantenido dentro de Google. (An Overview of AngularJS for Managers. 2014).

En el 2016 Google lanzó un sucesor conocido solo como Angular que reescribió mucho de su fuente original para adaptarse a un mercado digital más exigente. Angular permite manipular datos en el lado del cliente, controlar los elementos que se muestran en un navegador y manejar datos de entrada del usuario. (Gavigan, D, 2018).

Este marco de trabajo está escrito en Typescript que es un superconjunto de Javascript. Es un lenguaje de programación de código abierto y gratuito que fue desarrollado por Microsoft. Typescript adiciona soporte a la programación orientada a objetos basada en clases y permite la declaración explícita de tipos. El código desarrollado en Typescript es transpilado a su equivalente en Javascript por lo que se pueden crear aplicaciones de front-end y back-end. (TypeScript., 2016).

3.7. NODEJS

En el lado del back-end, los desarrolladores cuentan con el entorno de ejecución de código abierto y multiplataforma llamado NodeJS. Este es un marco de trabajo diseñado para mantener alta escalabilidad y tomar las ventajas de las tecnologías de Javascript para los navegadores. (Introduction to Node.js, s..).

NodeJS es una plataforma modular, implicando que muchas de las funcionalidades se encuentran en módulos externos. La cultura en NodeJS es la de crear y publicar módulos que resuelvan las necesidades imaginables del desarrollo web. (Dayley, B., & Dayley, B., 2017).

NodeJS corre el motor de Javascript V8 que es el componente principal del navegador Google Chrome. La mayor ventaja es que este motor es independiente del navegador, esta fue la característica clave del éxito de NodeJS. (The V8 JavaScript Engine, s.f.).

Una aplicación desarrollada con NodeJS corre en un solo proceso sin crear un nuevo hilo con cada llamado. NodeJS cuenta con periféricos de entrada y salida asíncronos evitando bloqueos innecesarios. El paradigma no bloqueante se utiliza frecuentemente en las librerías de NodeJS donde el bloqueo es una excepción y no la norma. (Introduction to Node.js, s..).

3.8. HERRAMIENTAS DE DESARROLLO

El desarrollo web cuenta con otras herramientas de apoyo además de los marcos de trabajo. Una de las herramientas más utilizada son los sistemas de control de versiones (VCS) o sitios web para compartir o utilizar paquetes de código abierto.

Un sistema de control de versiones es una herramienta que mantiene un histórico de los cambios hechos en activos digitales. Un activo digital puede ser una imagen, archivo, video o cualquier medio digital que pueda ser transformado. Un VSC permite a una persona u organización guardar una versión estable de un proyecto para poder experimentar con este. En el caso de no estar conforme con los cambios implementados se puede regresar a la versión más estable del proyecto. También es utilizado por equipos de trabajo en el que cada miembro se encarga de distintas tareas del mismo proyecto. (Jariwala, S., 2020).

Git es un sistema de control de versiones distribuida que es gratuito y de código abierto. Con Git, los proyectos se pueden manejar en ramas que son independientes entre ellas. Estas ramas pueden ser creadas, mezcladas y eliminadas en cuestión de segundos. El hecho de ser distribuido significa que cada desarrollador trabaja con una copia local del proyecto y esto permite implementar distintos flujos de trabajo. Estos flujos de trabajo pueden ser del estilo de subversiones del proyecto donde los desarrolladores envían sus cambios a una misma rama, del estilo de integración donde cada desarrollador trabaja en una rama y solicita a un administrador la integración de su trabajo a la rama principal y por último del estilo de "dictadores y tenientes" que se utiliza para proyectos masivos donde cada teniente se encarga de un subsistema y el dictador es quien envía los cambios a la rama principal. (About., s.f.).

Muchos de los módulos creados para NodeJS son compartidos por el proyecto npm. Este fue creado en el 2019 como un proyecto de código abierto. Actualmente cuenta con una vasta colección de paquetes de código fuente público para aplicaciones de front-end, servidores, aplicaciones web, enrutadores de red y robots. Es utilizado por desarrolladores de todos los continentes para proyectos de fuente abierta o por organizaciones que desarrollan de forma privada. (About npm, s.f.).

El proyecto Npm consiste en un sitio web para descubrir o subir nuevos paquetes, una interfaz de comando de línea CLI para interactuar con npm y el registro que transforma los paquetes de npm para poder leer la información contenida dentro de los estos implementando el registro de paquetes CommonJS. (registry The JavaScript Package Registry, s.f.).

IV. DESARROLLO

El proyecto Impact Tickets es un sistema que se encarga de administrar boletas de trabajo para el departamento de diseño en Impact Mobile. Fue desarrollado con funcionalidades para los clientes, los diseñadores y la supervisora del equipo de diseño. Los clientes pueden crear solicitudes, comunicarse con la jefa de diseño y ver los diseños solicitados, por el otro lado la supervisora se encarga de asignar diseñadores a una solicitud, crear tareas a dicha solicitud y aprobar el trabajo realizado. Los diseñadores reciben tareas las cuales deben ser subidas a un almacenamiento en línea. El trabajo de ellos es representado por los enlaces a los almacenamientos virtuales que contiene artes o diseños realizados.

4.1. DESCRIPCIÓN DEL TRABAJO REALIZADO

4.1.1. PERFIL DE USUARIO

En la primera semana, se asignó trabajar en el front-end con la tarea de crear la vista del perfil de un usuario utilizando una plantilla.

En la Ilustración 1 se puede observar la plantilla Minton para la vista de perfil. La plantilla despliega la información general, una tabla de proyectos, una sección con habilidades, una sección con experiencia laboral y la opción de modificar información del perfil.

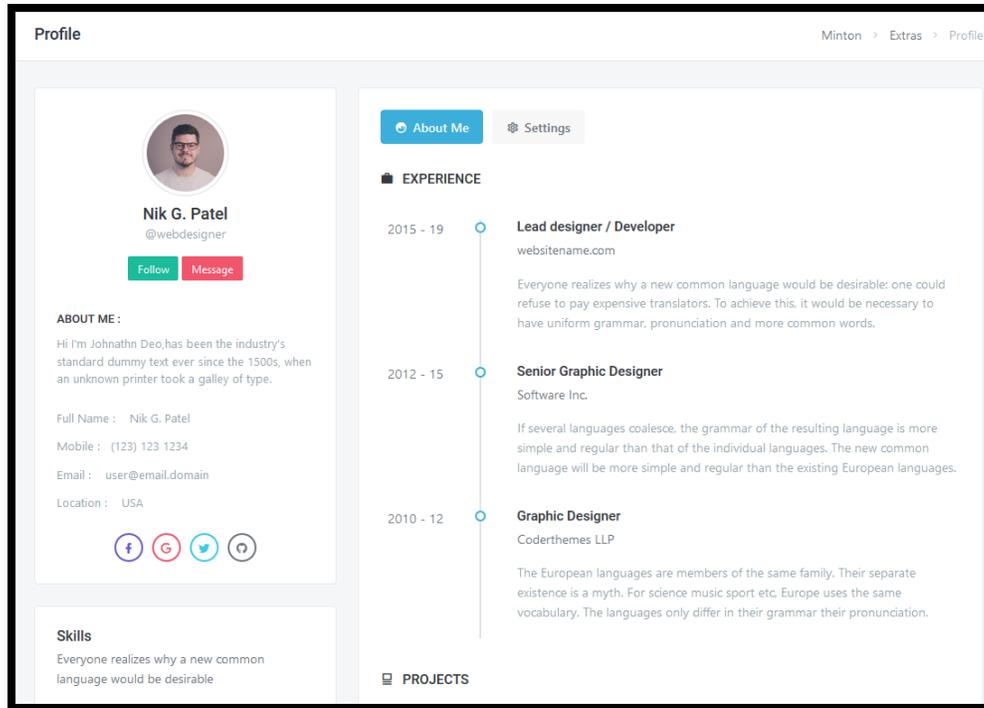


Ilustración 1 - Plantilla base de Perfil.

Fuente: (Coderthemes, s.f.)

Se cambió el proyecto con las boletas, el perfil desplegaría la información adecuada del usuario, se mostrarían sesiones en lugar de experiencias laborales y se eliminó la opción de modificar información y la sección de habilidades.

Todo esto se dividió en componentes para asegurar el reciclaje de código y facilitar el mantenimiento. Se adaptó el CSS a los requerimientos de la empresa.

En la Ilustración 2 se muestra la vista final de perfil que consistía en una sección de información, de registros de sesiones y de boletas. Se implementó un desplazamiento infinito vertical a la sección de boletas.

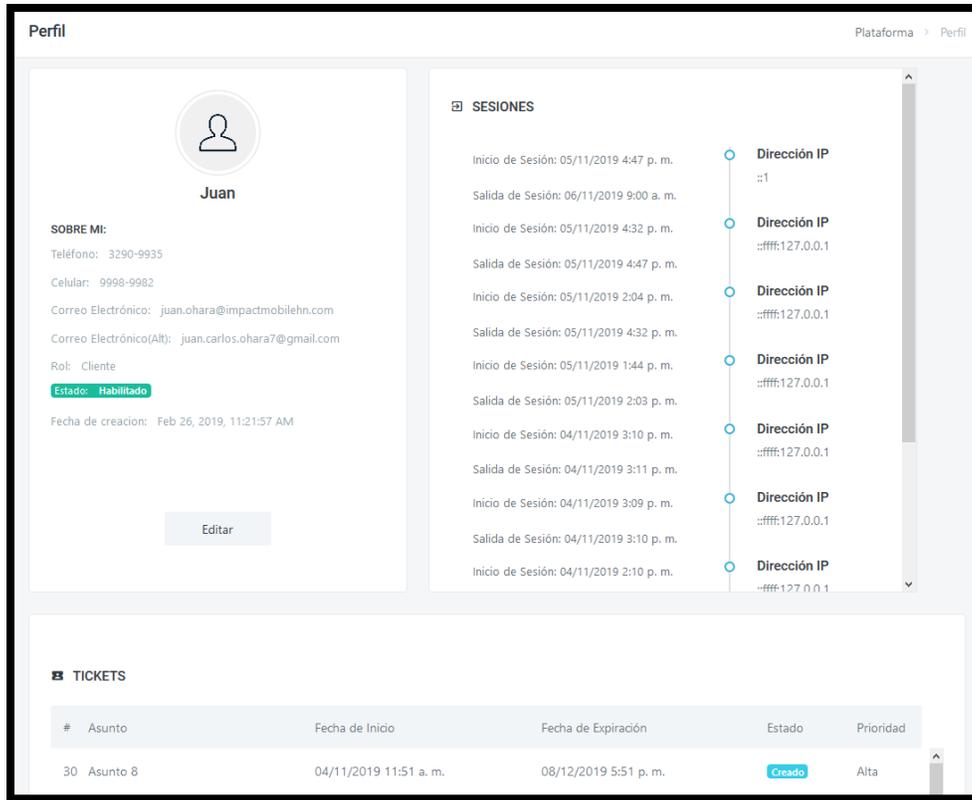


Ilustración 2 - Vista para visualizar perfil adaptado para el uso de la empresa

Fuente: Elaboración Propia

Al inicio se trabajó con datos estáticos para enfocarse en el diseño del perfil. Una vez aprobado, se otorgó acceso a la base de datos y al repositorio del back-end para trabajar con datos dinámicos.

El front-end utiliza el marco de trabajo Angular por lo cual se desarrolló principalmente en Typescript, HTML y CSS. Mientras que el desarrollador del back-end programó los endpoint en Javascript y los conectó a un base de datos de MySQL.

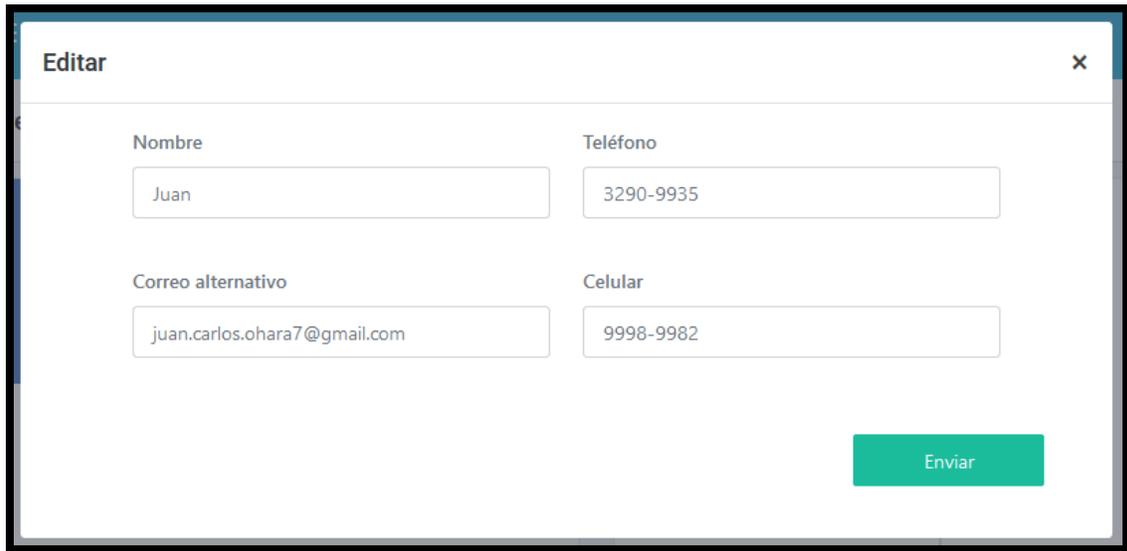
Las herramientas principales fueron Visual Code, Git Bash y MySQL Workbench.

4.1.2. EDITAR PERFIL

Naturalmente, la siguiente tarea fue permitir al usuario modificar su información, el sistema solo permite cambiar nombre, número de teléfono, número de celular y correo electrónico alternativo. El campo de rol no se debe editar para no permitir la modificación de los privilegios asignados a

un usuario y la empresa optó por no permitir cambiar el correo de autenticación. Los otros dos campos ya no eran relevantes para el usuario.

En la Ilustración 3 se ve el modal, que es un pequeño formulario, con la información original del usuario para poder ser editada.



The image shows a modal window titled "Editar" with a close button (X) in the top right corner. The form contains four input fields arranged in a 2x2 grid:

- Nombre:** Input field containing "Juan".
- Teléfono:** Input field containing "3290-9935".
- Correo alternativo:** Input field containing "juan.carlos.ohara7@gmail.com".
- Celular:** Input field containing "9998-9982".

At the bottom right of the modal is a green button labeled "Enviar".

Ilustración 3 - Vista para editar perfil

Fuente: Elaboración Propia

Cada campo es validado para disminuir la entrada de información errónea a la base de datos. Si el usuario se equivoca, se le señala el error con una alerta y esta desaparece una vez que se haya corregido. Si toda la información está en orden, los datos se envían al servidor del back-end para actualizar las tablas. En el lado del servidor, se creó el endpoint con la consulta que cambiará la información.

Se puede observar en la Ilustración 4 los mensajes de error en caso de omitir un dato o ingresar datos incorrectos.

The image shows a web form titled "Editar" (Edit) with a close button (X) in the top right corner. The form contains four input fields arranged in a 2x2 grid:

- Nombre:** An empty text input field with a red border and a red "X" icon on the right. Below it is the error message "Nombre no puede ser vacío" (Name cannot be empty).
- Teléfono:** A text input field containing the value "3290-9935".
- Correo alternativo:** A text input field containing the value "juan.carlos.ohara7gmail.com" with a red border and a red "X" icon on the right. Below it is the error message "Correo no válido" (Invalid email).
- Celular:** A text input field containing the value "9989-982" with a red border and a red "X" icon on the right. Below it is the error message "número de celular incompleto" (incomplete cell number).

At the bottom right of the form is a green button labeled "Enviar" (Send).

Ilustración 4 - Las validaciones del formulario para editar perfil

Fuente: Elaboración Propia

4.1.3. AGREGAR USUARIOS

La siguiente tarea es dar la opción a los supervisores de ingresar nuevos usuarios al sistema, se estableció que solo se puede agregar a alguien con un rol menor y que los clientes y los diseñadores no pueden añadir a nadie.

El componente requiere ingresar nombre, correo, rol y tipo. Siempre se realizan validaciones de datos y sus advertencias respectivas. El campo de tipo se habilita si el rol que se escoge no es el de cliente. En el caso de serlo, la opción quedará en blanco y se envía un parámetro por defecto.

La Ilustración 5 muestra el componente para agregar usuario.

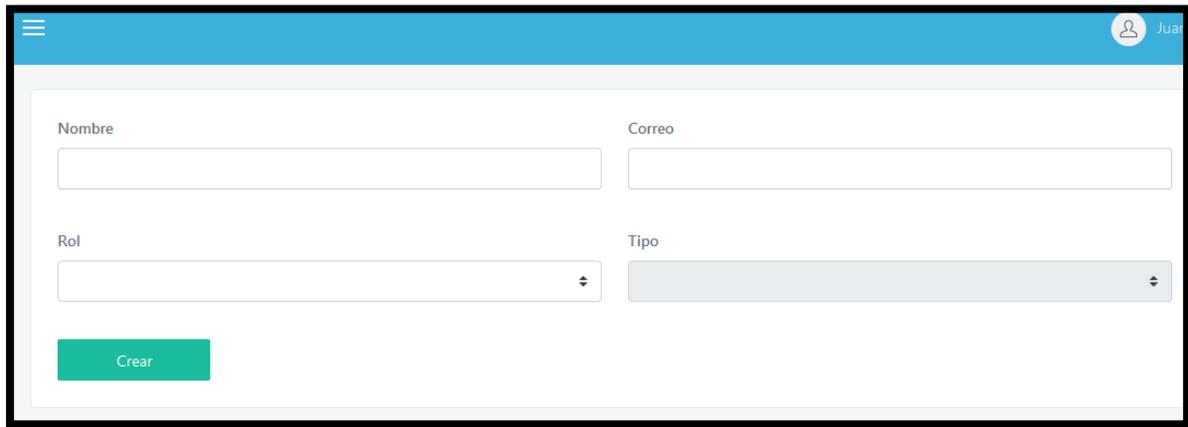
The image shows a web application interface for creating a user. It features a blue header bar with a menu icon on the left and a user profile icon labeled 'Juan' on the right. The main content area is white and contains four input fields: 'Nombre' (Name) and 'Correo' (Email) are text boxes; 'Rol' (Role) and 'Tipo' (Type) are dropdown menus. A green 'Crear' (Create) button is positioned at the bottom left of the form.

Ilustración 5 - Vista para crear usuario

Fuente: Elaboración Propia

El servidor verifica que el correo no esté repetido y este enviará una advertencia como se puede ver en la Ilustración 6 que ya existe un usuario con ese correo.



Ilustración 6 - Alerta de error en el caso de que ya exista un usuario con ese correo.

Fuente: Elaboración Propia

Con los datos en orden, se ejecuta la consulta de ingresar en la tabla de usuarios y se notifica el éxito de la operación como se puede ver en la Ilustración 7.

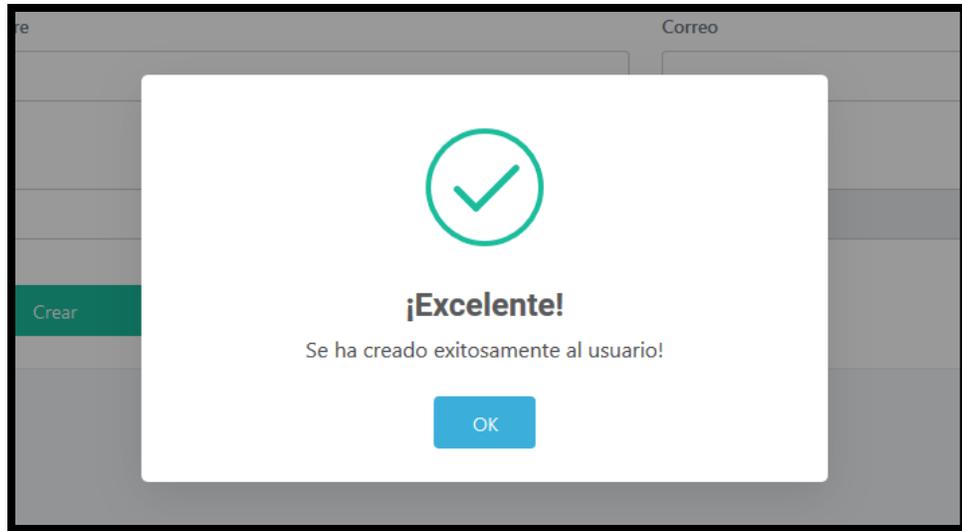


Ilustración 7 - Alerta de éxito al momento de crear un nuevo usuario.

Fuente: Elaboración Propia

4.1.4. MOSTRAR USUARIOS

Es importante que se puedan ver los usuarios para que los supervisores observen las boletas asignadas a los diseñadores o para que un cliente se ponga en contacto con algún diseñador.

Se creó un componente que muestra cartas de los perfiles con alguna información y un botón para ver el perfil completo, y en el caso de ser un supervisor las boletillas asignadas al usuario. Los usuarios de tipo cliente no pueden ver a otros clientes y se omiten las boletillas. Es en el lado del back end que se distingue a los tipos de usuarios.

Se puede ver en la Ilustración 8 el componente para ver otros usuarios

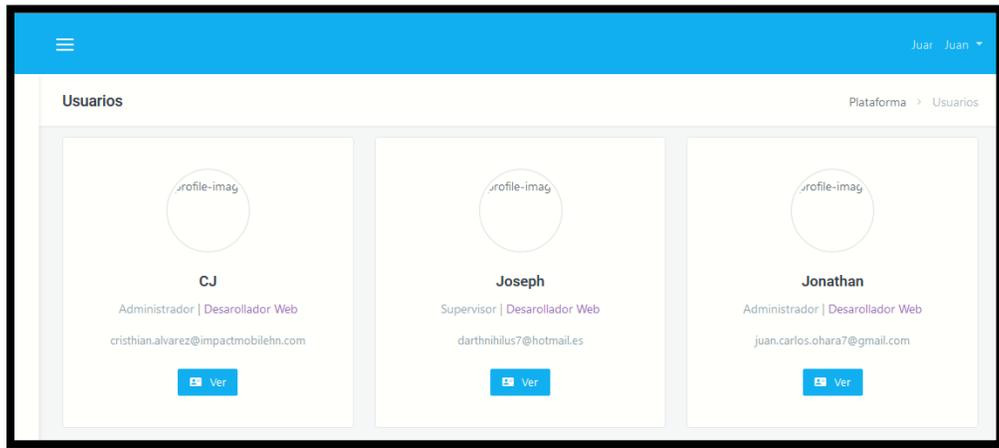


Ilustración 8 - Vista para ver usuarios.

Fuente: Elaboración Propia

En la Ilustración 9 se muestra el perfil de otro usuario.

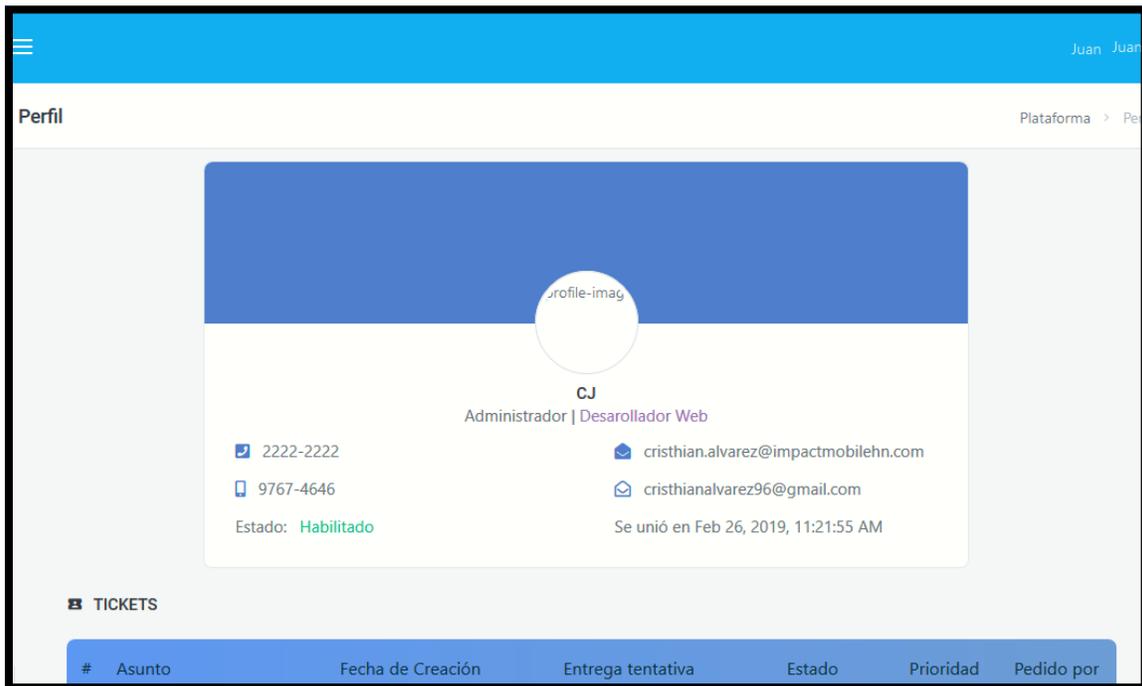


Ilustración 9 - La vista del perfil completo del usuario.

Fuente: Elaboración Propia

4.1.5. CAMBIOS DE INTERFAZ DE USUARIO EN EL PERFIL DE USUARIO

Después de presentar la página al equipo de diseño, se solicitaron cambios en funcionalidades y en la interfaz de usuario (UI) de distintos componentes, en el caso del perfil en la Ilustración 10 se puede ver que se agregó un banner azul, las letras cambiaron de posición y color, se modificó el botón de editar por un icono. Todo esto fue trabajado en el CSS. El componente de sesiones cambió de tamaño.

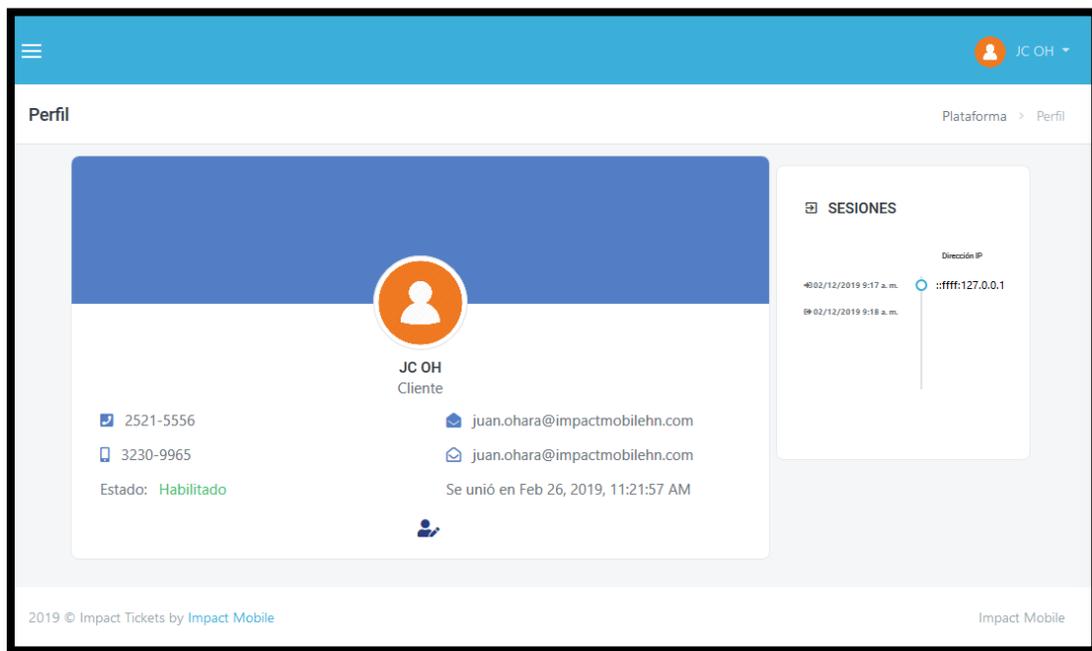


Ilustración 10 – Nuevo UI de perfil de usuario.

Fuente: Elaboración propia

En la barra de menú, se combinaron los botones de agregar y ver usuarios a uno solo. En el caso de ser cliente, él entraría al componente de ver usuarios inmediatamente, mientras que un supervisor vería el menú que se ve en la Ilustración 11.

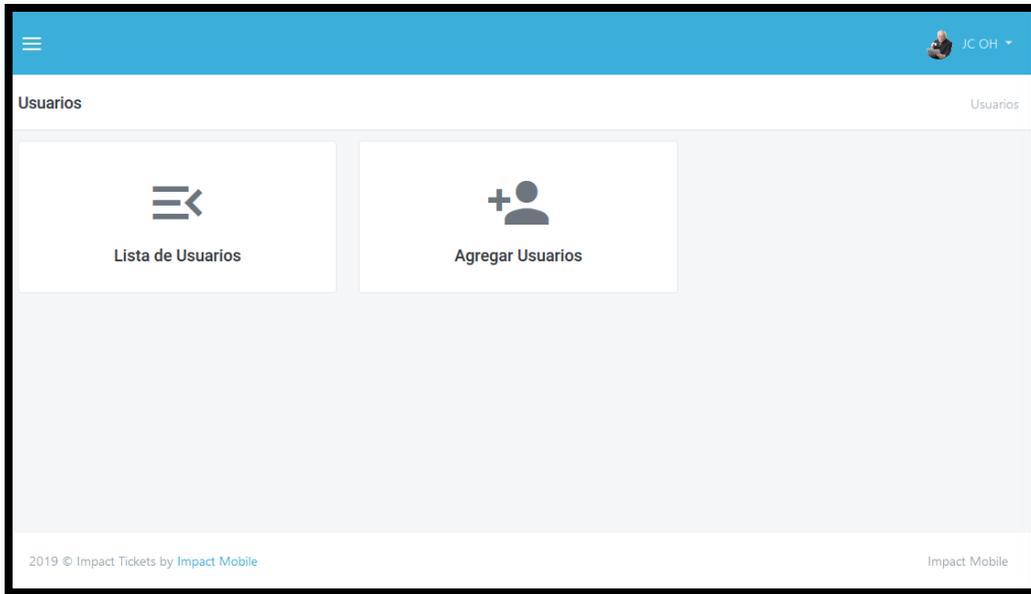


Ilustración 11 – Nuevo menú para usuarios

Fuente: Elaboración propia

4.1.6. SOLICITUDES

El cambio más fuerte fue la dinámica de crear boletillas, ahora la funcionalidad se controla por solicitudes. En la Ilustración 12 se ve la interfaz para los clientes que quieran crear una solicitud.

Se le da las opciones de filtrar solicitudes, crear solicitudes y ver las solicitudes archivadas. Las solicitudes archivadas son aquellas que ya fueron terminadas.

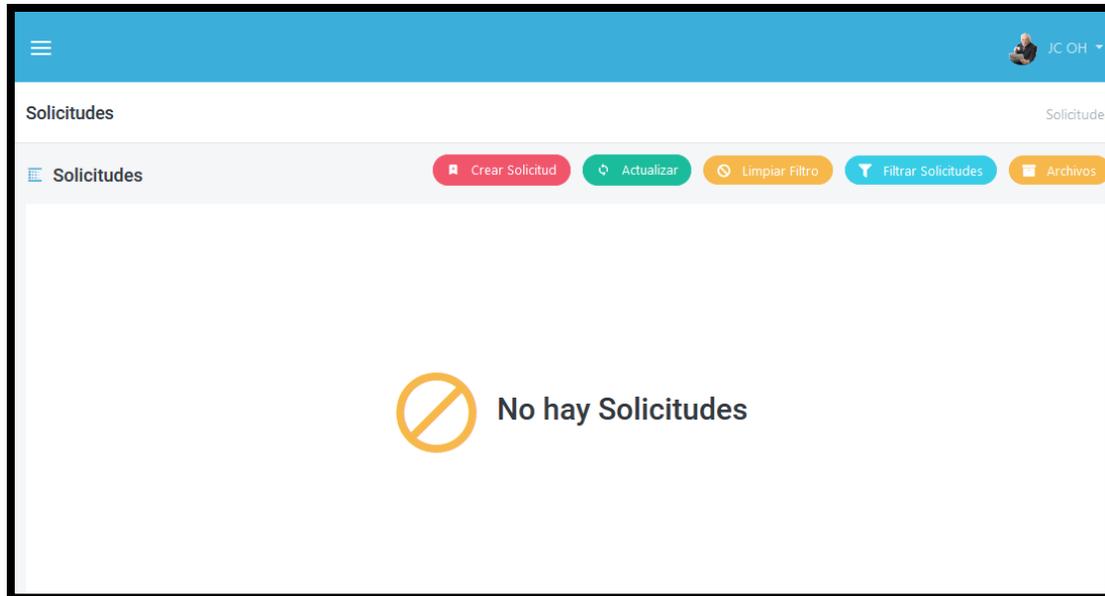


Ilustración 12 – Componente de solicitudes.

Fuente: Elaboración propia

En la Ilustración 13 se muestra el modal para crear una solicitud. Siempre se aplicaron las validaciones de datos, el usuario tiene la opción de subir archivos y elegir una fecha tentativa para la entrega de la solicitud.

El modal es un componente que avanza conforme el usuario llena los campos solicitados.

El servidor es quien se encarga de guardar los archivos que el usuario suba.

The image shows a web form for adding a new request. At the top, there is a progress indicator with three circular icons: an information icon, a person icon, and a calendar icon. Below this, the form has the following fields:

- Asunto:** A text input field.
- Tipo:** A dropdown menu.
- Descripción:** A large text area.
- Prioridad:** Three radio buttons labeled 'Alta', 'Media', and 'Baja'.

A 'Siguiete' button is located at the bottom right of the form.

Ilustración 13 – Agregar nueva solicitud.

Fuente: Elaboración propia

Por último, en la Ilustración 14 se ve la solicitud creada, ahí se ven todos los datos y los enlaces a los archivos subidos al servidor. Hay una sección de comentarios para que interactúen el cliente y el supervisor.

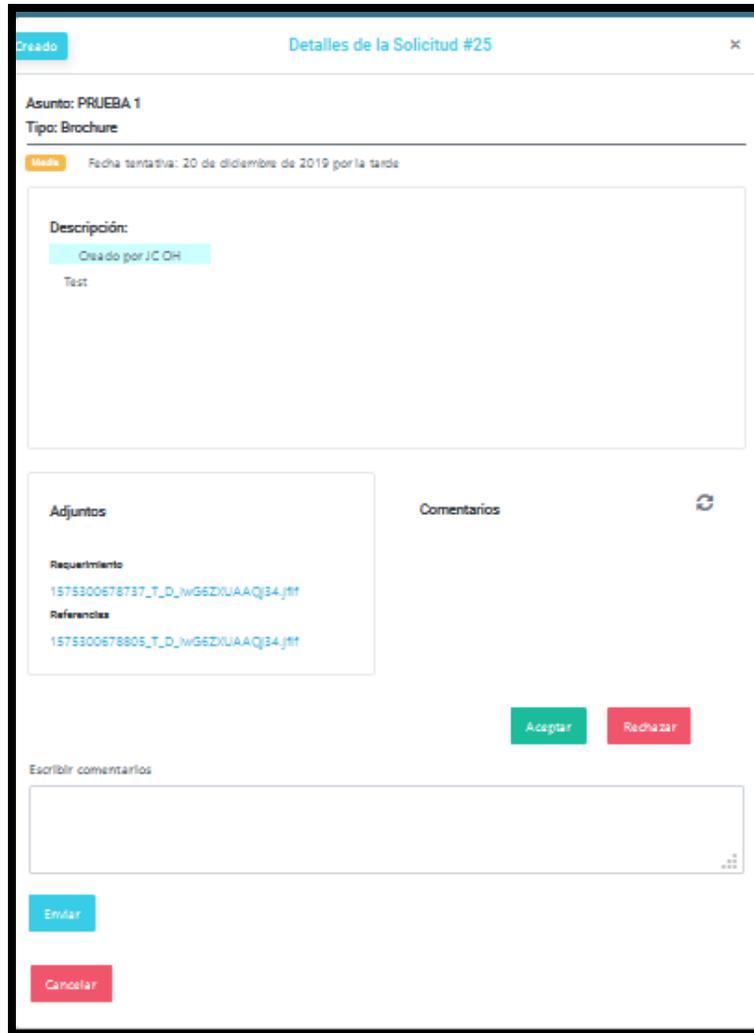


Ilustración 14 – Ver solicitud creada.

Fuente: Elaboración propia

4.1.7. MENSAJERÍA INSTANTÁNEA

En el transcurso del desarrollo del sistema, se decidió cambiar los comentarios estáticos en la vista de solicitudes por un servicio de mensajería instantánea. Esta conversación solo es entre un cliente y la supervisora. Se usó la librería de socketIO para manejar los mensajes de los usuarios y actualizar la base de datos. El usuario escribe su mensaje y ve como este aparece en la ventana superior y pasa lo mismo cuando se recibe un nuevo mensaje. El servicio fue desarrollado en el back end, ahí se crea el socket correspondiente a cada conversación entre usuarios a través de

sus endpoints. El front end utiliza los endpoint enviando la información del usuario actual. En la Ilustración 15 se muestra la interfaz de mensajería entre usuarios.

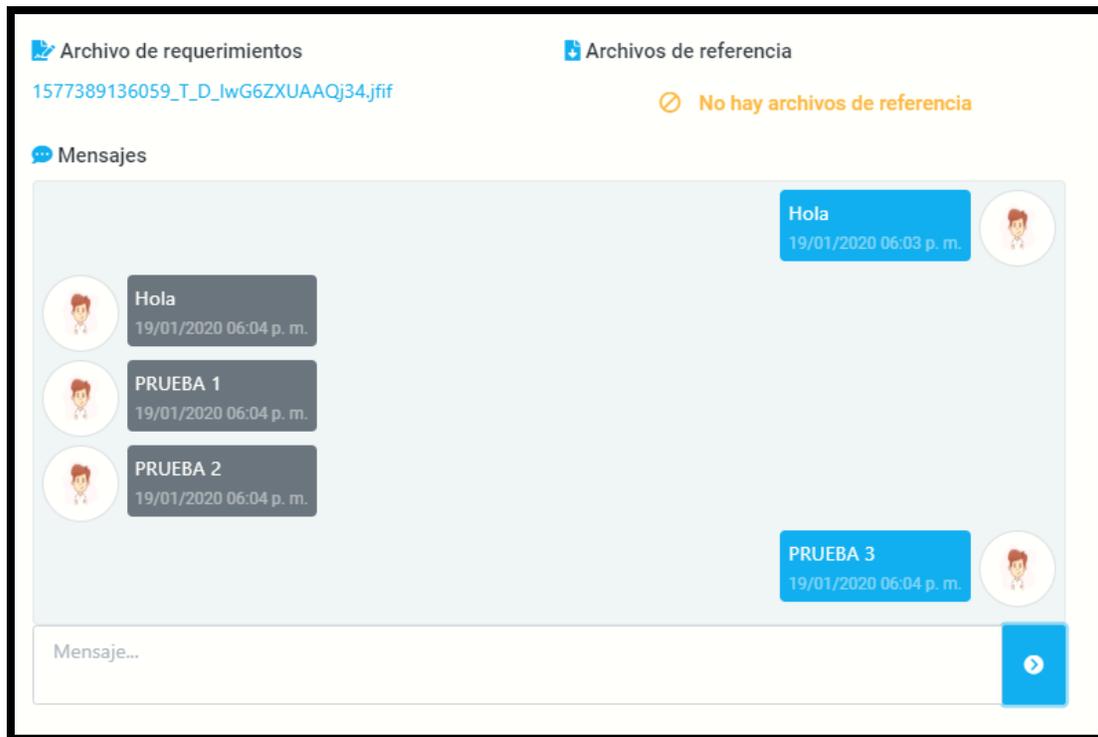


Ilustración 15 – Mensajería instantánea.

Fuente: Elaboración propia

4.1.8. NOTIFICACIONES

Debido a que todos los usuarios necesitan saber cuándo se ha realizado un cambio en una solicitud o cuándo han sido asignados una nueva tarea, se incorporó una pequeña ventana con notificaciones. Ahí se pueden ver mensajes, solicitudes y cambios, así mismo permite informar al usuario y evitar que este tenga que ir a buscar manualmente una solicitud. Se le indica cuántas solicitudes nuevas hay y el usuario puede hacer click en cada una para ver una solicitud completa. En la Ilustración 16 se observa la pequeña ventana de notificaciones.

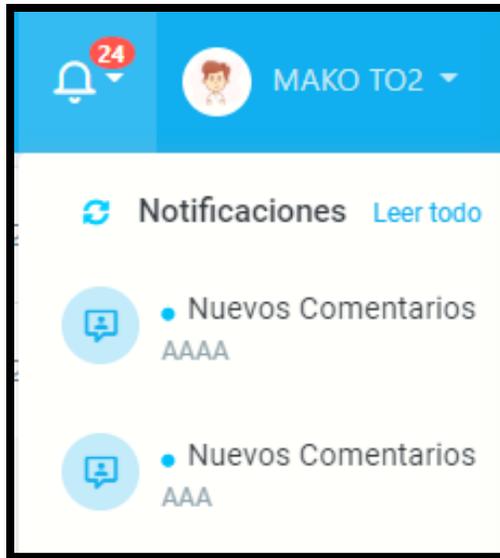


Ilustración 16 – Notificaciones.

Fuente: Elaboración propia

4.1.9. MEJORAS DE UI Y CORRECCIONES DE ERRORES

La implementación de la mensajería instantánea y las notificaciones fueron las últimas características que se añadieron al sistema de boletas. Las semanas restantes se concentraron en mejorar algunas vistas de la interfaz de usuario como se ve en la Ilustración 17 la vista de miembros del equipo. Ahora se hace un sombreado en los bordes de la imagen y se resalta el nombre del usuario cada vez que se pasa el cursor por encima de una carta de usuario. Se puede ver el perfil al hacer un click en el usuario, reemplazando el botón de ver en las versiones anteriores.

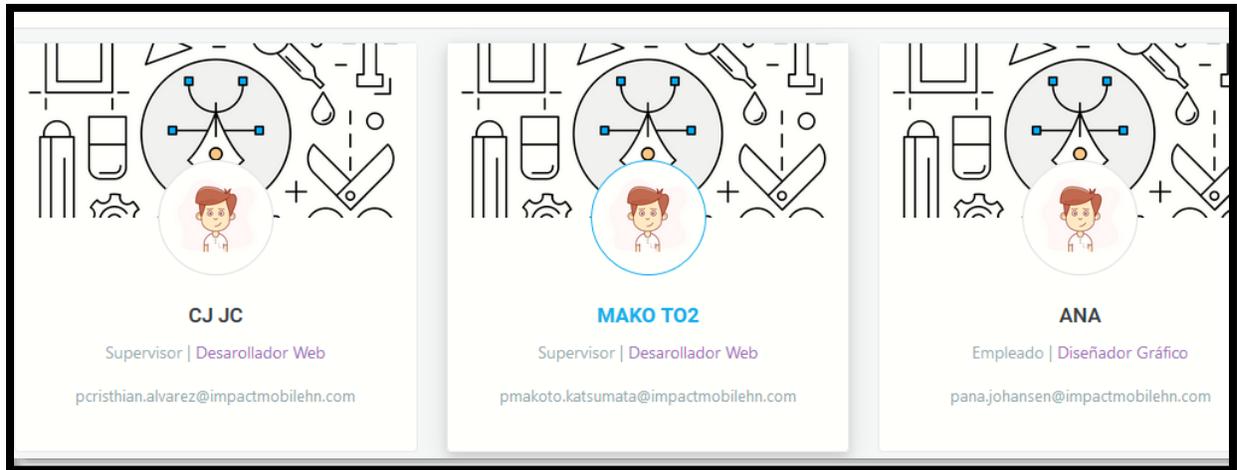


Ilustración 17 - Nuevo UI de Ver lista de usuarios.

Fuente: Elaboración propia

Las sesiones de usuarios que se veían al lado derecho del perfil en la Ilustración 10 pasaron a un modal visualizado en la Ilustración 18 que se abre al hacer click en el nuevo botón de sesiones, a este se le añadió el scroll infinito y se cambió la presentación de las sesiones.

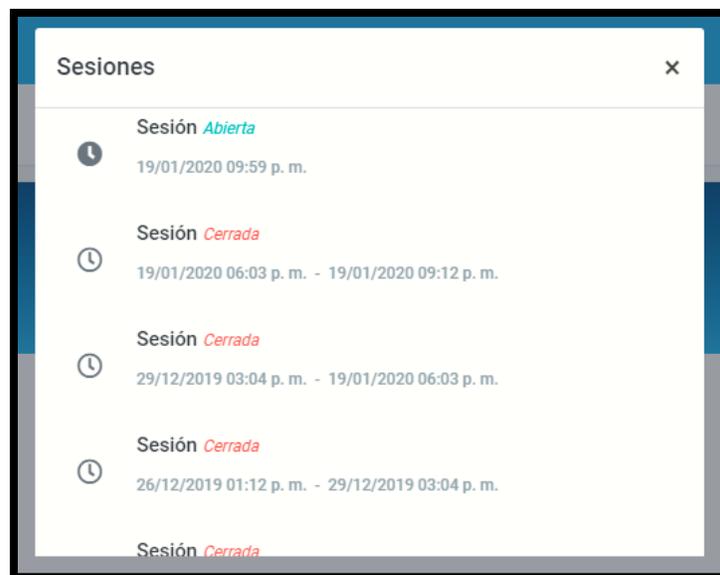


Ilustración 18 – Mejora de UI de sesiones.

Fuente: Elaboración propia

Se cambió la forma de asignar usuario de la vista creada por otro desarrollador, se reemplazó una lista que sombreaba a los usuarios asignados por dos listas que muestran los usuarios disponibles y los usuarios asignados. Cualquier click en los nombres hace que este se mueva de una lista a otra y viceversa. En la Ilustración 19 se puede ver una versión temprana de esta función, aún tenía las instrucciones de la lista original.

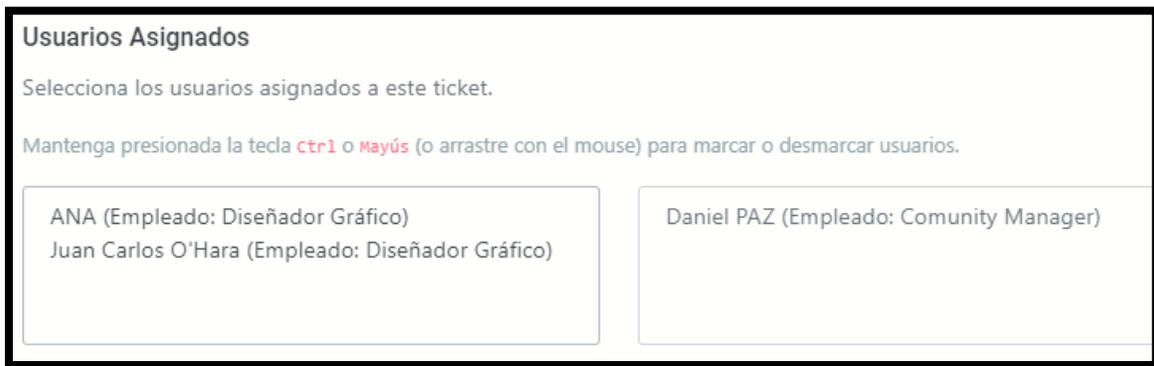


Ilustración 19 – Asignación de usuarios.

Fuente: Elaboración propia

Por último, el equipo de desarrollo empezó a buscar todos los errores que el sistema pudo presentar, principalmente asegurar que todo funcionaba de forma consistente y que hubiera armonía entre los componentes. Los errores más comunes involucraban a los estados de las boletas/solicitudes cada vez que se hacía una acción en ellos. No siempre se actualizaba a los estados correspondientes entonces se desarrollaron funciones que hicieron el cambio de estado correspondiente. Había errores con la paginación de solicitudes por conflictos de identificadores que fueron corregidos. Finalmente se agregaron mensajes que le indicaban al usuario cuando un elemento no existía en la base de datos. Se puede ver en la Ilustración 20 que no existen archivos de referencia y se le muestra el mensaje correspondiente para evitar que el usuario piense que dicho elemento no se ha cargado.

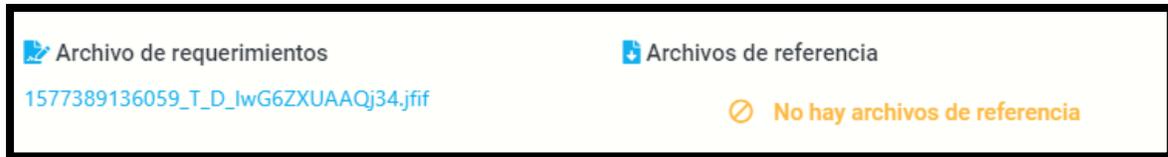


Ilustración 20 – Mensaje de elementos vacíos.

Fuente: Elaboración propia

4.1.10. DESPLAZAMIENTO INFINITO VIRTUAL

El siguiente módulo se creó como una prueba de concepto para un sistema de planilla que se estará desarrollando en los siguientes meses. La tabla que se ve en la Ilustración 21 muestra una lista de datos aleatorios. La idea principal de este módulo es de que el front end solo cargue una cantidad limitada de datos. Cada vez que se hace un desplazamiento vertical se eliminará una cantidad determinada de datos de la lista y se cargarán los datos siguientes o anteriores. En este pequeño programa se cuenta con 1000 registros y solo se cargan los primeros 50 datos. Cuando hace un desplazamiento hacia abajo se cargan los siguientes 50 hasta que se llega al límite de 200 registro por tabla. Llegar a este límite provoca que los primeros 50 datos sean borrados mientras que los 50 datos siguientes son añadidos a la tabla y viceversa al momento de hacer un desplazamiento hacia arriba.

ID	Asunto	Estado	Description	Fecha	Opciones
#1	facilisis	1	tincidunt adipiscing. Mauris molestie pharetra nibh. Aliquam ornare, libero at auctor ullamcorper, nisl arcu iaculis	Jan 23, 2019	Eliminar
#2	tellus	2	a, arcu. Sed et libero. Proin mi. Aliquam gravida mauris ut mi. Duis risus odio,	May 1, 2020	Eliminar

Ilustración 21 – Tabla de datos.

Fuente: Elaboración propia

El programa cuenta con funciones de borrar registros y agregar nuevos registros. Además, se cuenta con un pequeño buscador que resalta las palabras que concuerdan con la búsqueda como se puede ver en la Ilustración 22.

ID	Asunto	Estado	Description	Fecha	Opciones
#4	risus.	1	a, malesuada id, erat. Etiam vestibulum massa rutrum magna. Cras convallis convallis dolor. Quisque tincidunt	Jul 17, 2020	Eliminar
#14	Duis	3	in, dolor. Fusce feugiat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam auctor, velit	Jan 2, 2021	Eliminar

Ilustración 22 – Búsqueda en tabla.

Fuente: Elaboración propia

4.1.11. MOTOR ARITMÉTICO

Este pequeño módulo también es parte de unas pruebas de concepto para el sistema de planillas, la tarea era crear un motor aritmético que resuelve expresiones. El programa recibe las cadenas en el área de texto y crea tokens de esa cadena, estos tokens son evaluados para asegurar que sea una expresión matemáticamente correcta para que después pueda ser enviada al motor que hace el cálculo de la expresión. Como se observa en la Ilustración 23, se pueden declarar variables y hacer cálculos con dichas variables.



Ilustración 23 – Motor aritmético.

Fuente: Elaboración propia

4.1.12. MÓDULO DE ARRASTRE

Se solicitó crear una nueva interfaz para el endpoint del motor aritmético, esta nueva interfaz debe permitir arrastrar elementos para formar una expresión aritmética que se resolverá en el back-end.

Se implementó el módulo de DragnDrop de la librería de Angular/CDK para permitir arrastrar elementos en la aplicación que puede estar corriendo en un navegador web de escritorio o en uno móvil.

La interfaz en la Ilustración 24 muestra la aplicación que contiene una barra en donde se suelta los elementos. A su lado derecho se muestra el resultado de la expresión. Debajo de la barra hay botones para resolver la expresión, eliminar el último elemento, borrar la expresión o guardarla si el usuario lo desea.

En la parte inferior se ve un grupo para números, para operadores, para variables y para expresiones guardadas. Estos elementos pueden ser arrastrados y soltados en la barra superior.

Los elementos dentro de la barra pueden ser arrastrados para cambiar su posición.

Se puede formar una expresión con hacer click en uno de los elementos y hacer lo mismo a un elemento dentro de la barra lo elimina de esta.

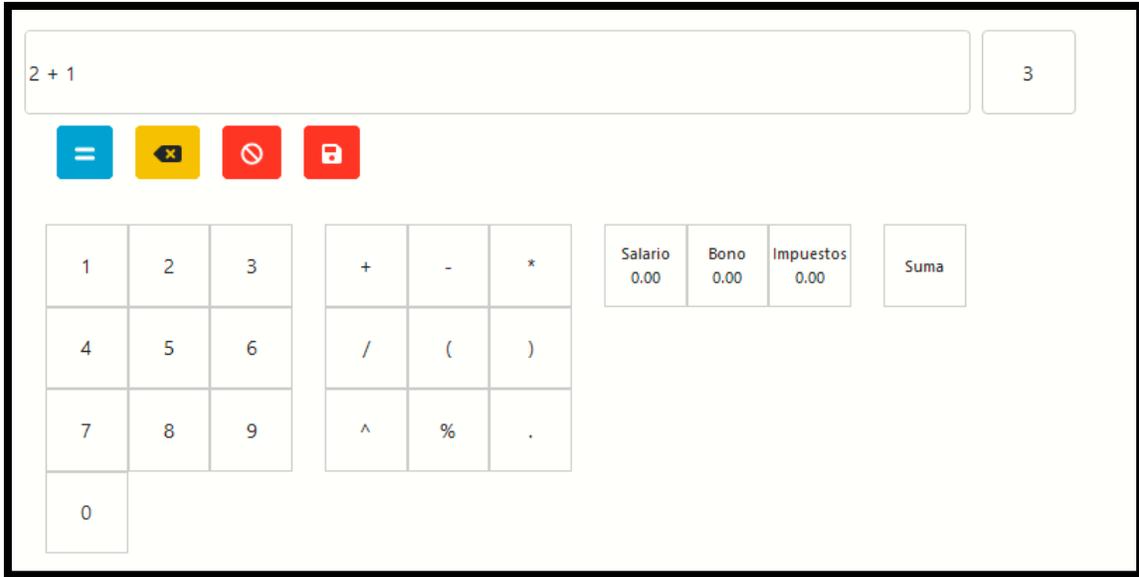


Ilustración 24 – Módulo de Arrastre

Fuente: Elaboración propia.

4.1.13. MÓDULO DE JORNADAS

Se pidió crear un endpoint que calcule el tiempo laborado dentro de una jornada y determinar el tipo de jornada a partir de la hora de entrada y de salida que se provea. La hora debe de estar en el formato de 24 horas con la forma de HH:MM:SS. El endpoint devuelve el total de horas trabajadas, el tipo de jornada y el total de horas extras, fue desarrollado en javascript y se encuentra dentro de un servidor creado con el módulo Express de NodeJS. La Ilustración 25 muestra el resultado en la herramienta de Postman que hace llamado del endpoint.

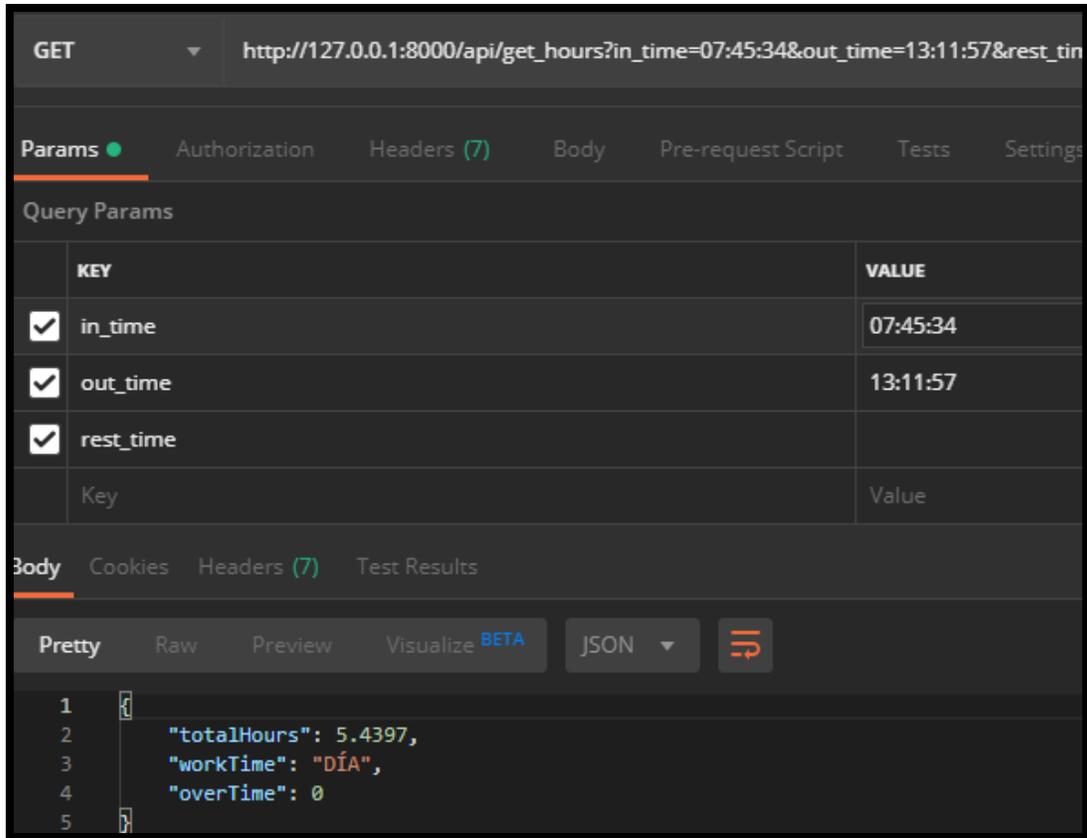


Ilustración 25 – Resultado en Postman para obtener información de una jornada.

Fuente: Elaboración propia.

4.1.14. MÓDULO DE JORNADA: CÁLCULO DE HORA

Se necesitó modificar el módulo anterior para que calculara y clasificara más detalladamente las horas en una jornada para poder guardar estas en una base de datos. Donde antes solo se mostraba el tiempo total ordinario y el extra ahora se calculan nuevos totales de acuerdo con los tipos de horas. Ese tiempo total puede ser de horas ordinarias diurnas o nocturnas, horas extras diurnas o nocturnas normales y nocturnas pero que pertenecen a una jornada diurna. Se agregó nuevas condicionales a la función que determina el tipo de jornada para que se basara en las reglas que se encuentran en el código de trabajo hondureño. El resultado de este cambio se refleja en la Ilustración 26.

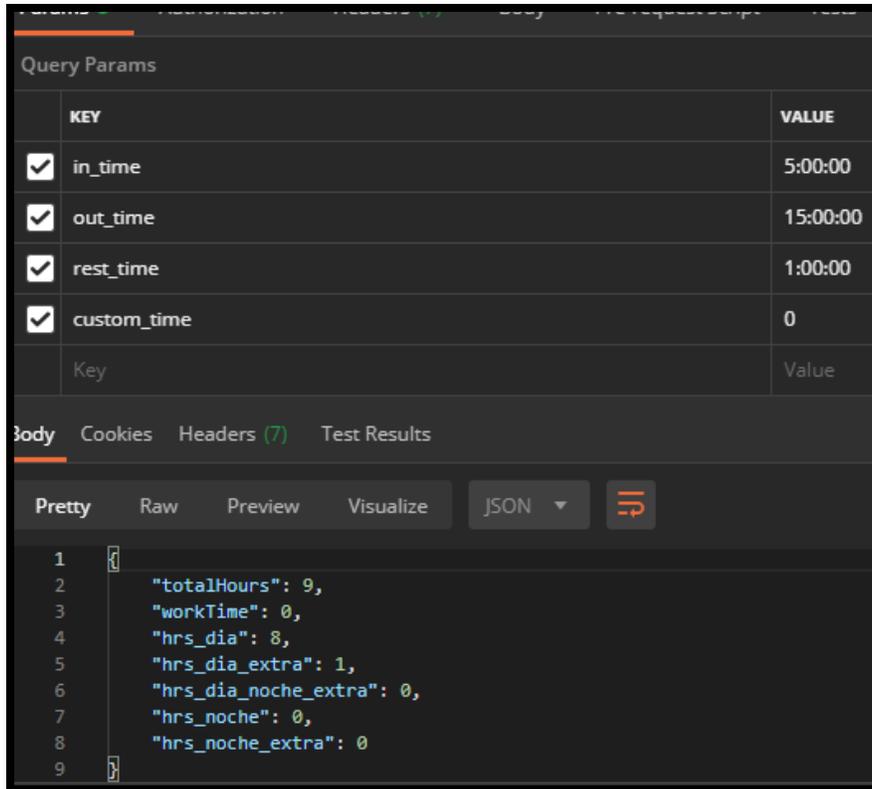


Ilustración 26 – Resultado en Postman de los cálculos de horas de una jornada.

Fuente: Elaboración propia.

4.1.15. MÓDULO DE TURNOS

La siguiente asignación fue diseñar un sistema que guardara los turnos y los tiempos de acceso de los empleados. En la Ilustración 27 se muestra el diseño de la base de datos que se creó para esta tarea. En el diseño, la tabla access_time guarda las horas trabajadas, la entrada y la salida de un empleado en el día, la tabla shifts se encarga de mantener el control de los turnos de los empleados, la tabla shifts_type ayuda a clasificar los turnos por tipos de jornadas, la tabla rest_time maneja los tiempos de descanso y por último la tabla de empleado que almacena la información de cada empleado. La base de datos fue creada con MySQL versión 8 y es manejada a través del programa Workbench.

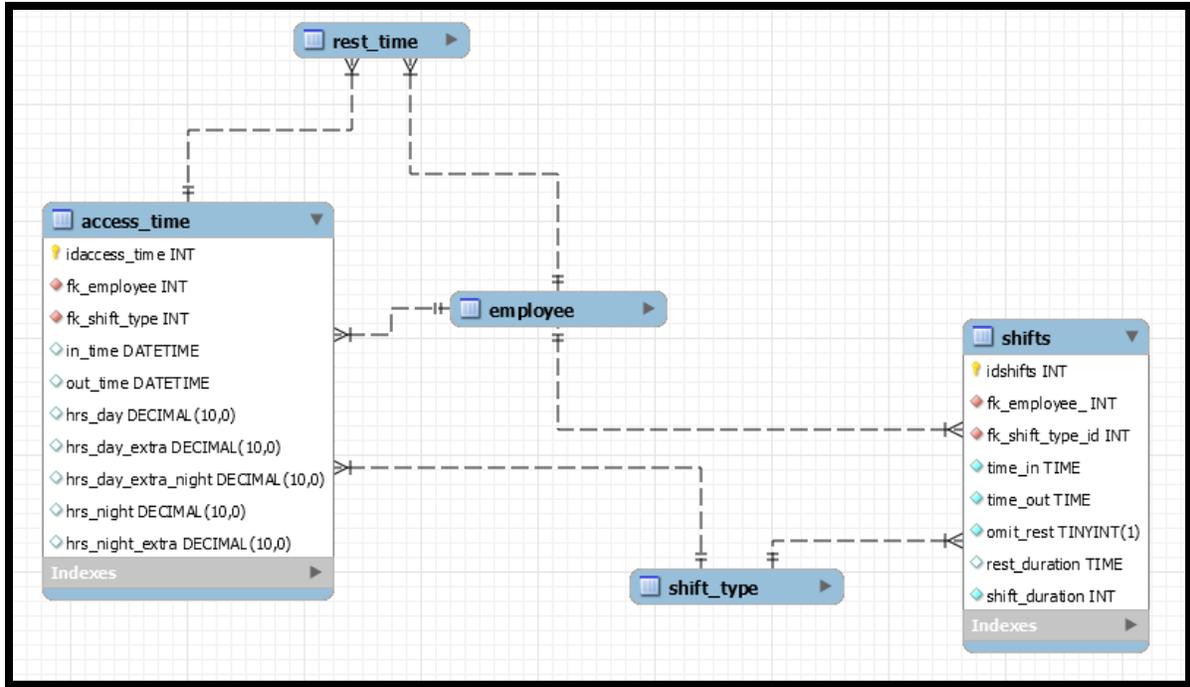


Ilustración 27 – Diseño de la base de datos para los turnos.

Fuente: Elaboración propia.

Se desarrolló en nodeJS todos los endpoints que se encargan de insertar, leer, actualizar y borrar filas en las tablas de la base de datos. Los endpoints que crean y actualizan filas de las tablas `access_time`, `shifts` y `rest_time` verifican que las fechas sean válidas para mantener la integridad de los datos. La Ilustración 28 muestra el resultado que se espera cuando se inserta un registro a la tabla `access_time`, este responde con un booleano que confirma si la ejecución del endpoint fue exitosa.

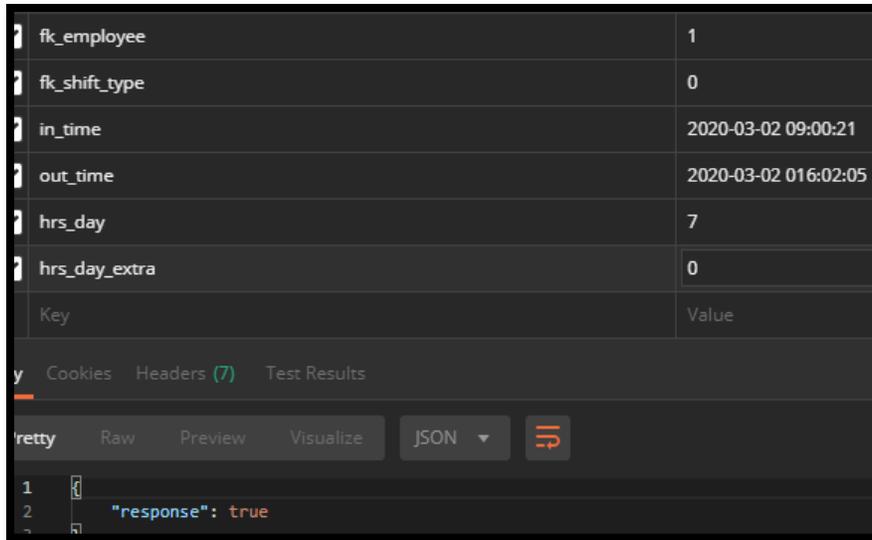


Ilustración 28 –Resultado de insertar un acceso de un empleado.

Fuente: Elaboración propia.

La tabla access_time cuenta con el endpoint para traer y sumar las horas totales de un empleado dentro de un rango de tiempo. Con el fin de dar la capacidad de crear reportes de un empleado. La Ilustración 29 muestra los totales de horas trabajadas de un empleado.

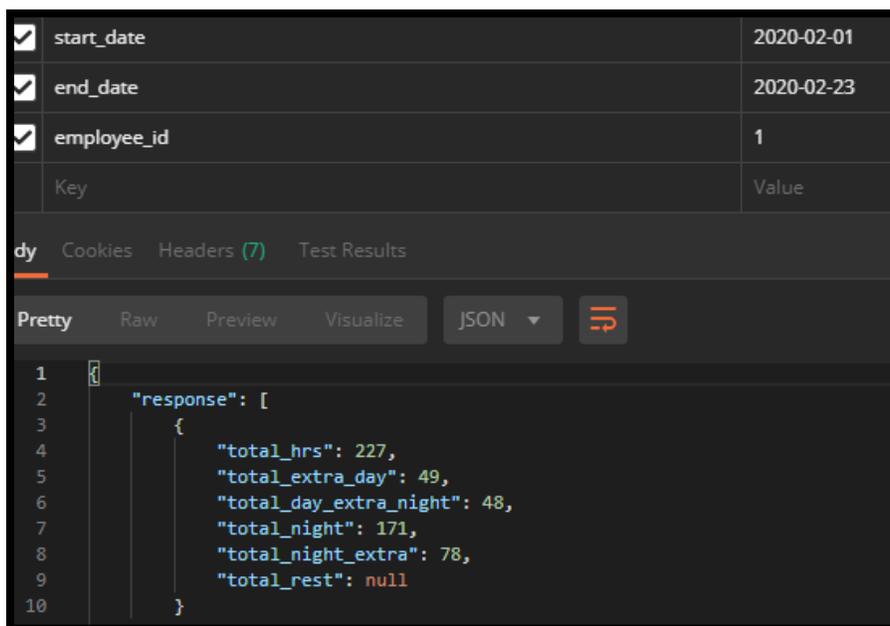


Ilustración 29 –Resultado de la suma de los totales de horas trabajadas de un empleado.

Fuente: Elaboración propia.

4.1.16. VARIABLES DE SISTEMA/USUARIOS

Esta asignación apoya al módulo aritmético para permitir guardar variables que un usuario pueda crear para sus fórmulas y para darle al mismo variables predeterminadas con las cuales él podrá hacer cálculos. La Ilustración 30 muestra el diseño de la base de datos, que consiste de la tabla `variable_system` que guardará las variables preestablecidas, la tabla `variable_user` para las variables de un usuario y la tabla de `variable_types` que determina por el momento si la variable es de tipo número o porcentual.

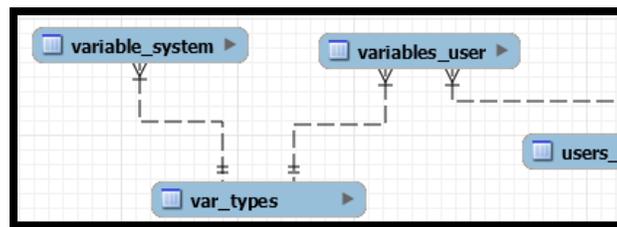


Ilustración 30 –Diseño de la base de datos para variables de sistema y de usuario.

Fuente: Elaboración propia.

Se crearon los endpoint para manipular las filas de las tablas en el servidor en nodeJS. Los endpoints para crear y actualizar una variable verifican que los nombres no se repitan. La Ilustración 31 muestra la respuesta del servidor de cuando se crea una variable de usuario.

La imagen muestra una herramienta de desarrollo (como Postman) que muestra la respuesta de un servidor. La respuesta es un objeto JSON con los siguientes campos:

Key	Value
<code>name_</code>	<code>TEST_RR</code>
<code>value</code>	<code>0.1</code>
<code>fk_id_user</code>	<code>1</code>
<code>fk_type</code>	<code>2</code>

Debajo de la tabla, se muestra la respuesta en formato JSON:

```
1 {  
2   "response": true
```

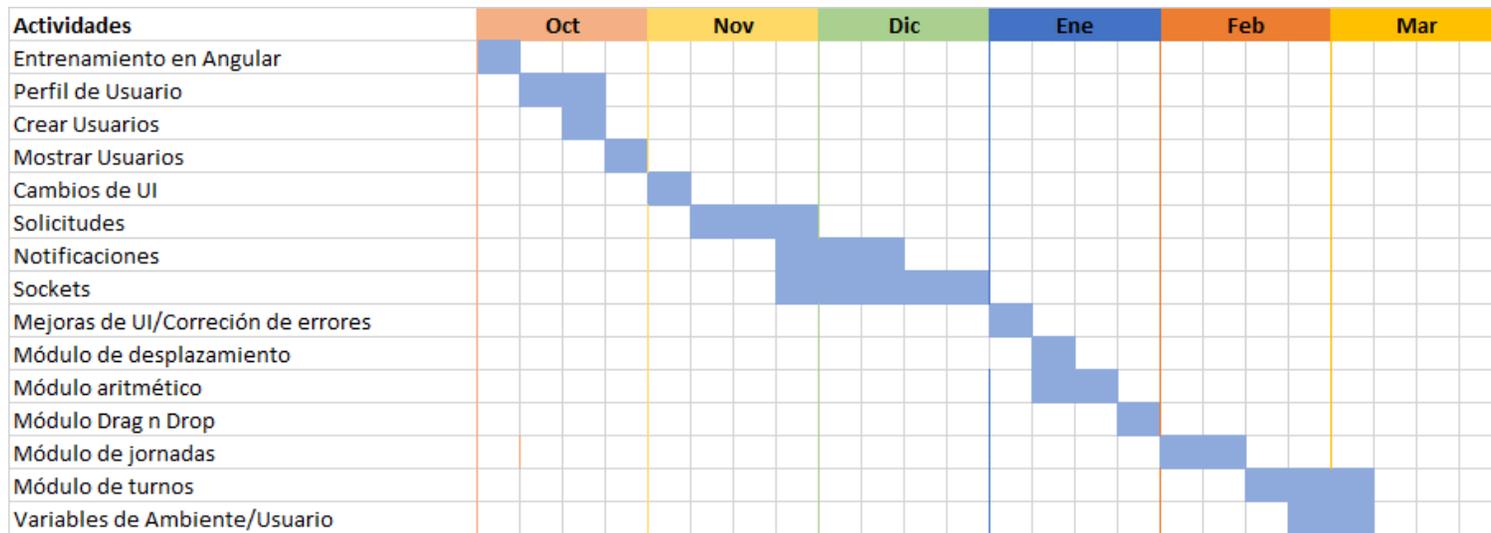
Ilustración 31 –Respuesta del servidor para crear variables de usuario.

Fuente: Elaboración propia.

4.2. CRONOGRAMA DE ACTIVIDADES

La Tabla 1 muestra el cronograma de actividades realizadas durante la práctica profesional

Tabla 1 - Diagrama de Gantt



Fuente: Elaboración propia

V. CONCLUSIONES

- Se fortaleció las competencias del departamento de desarrollo de software de Impact Mobile con el desarrollo del sistema de boletas y el sistema de recursos humanos de Impact Mobile.
- Se cumplió las tareas asignadas con buenos estándares de calidad utilizando buenas prácticas de programación y haciendo pequeñas pruebas para asegurar que el código desarrollado se comporte correctamente.
- Se documentó el código desarrollado en archivos de texto donde se describe el uso de las variables y funciones.
- Se usó nuevas herramientas como Postman para probar endpoints de servidores y MySQL Workbench para crear consultas a bases de datos.
- Se implementó la versión 8 de Angular para desarrollar el front-end de las aplicaciones desarrolladas.

VI. RECOMENDACIONES

- Tener diseñadores web para crear las interfaces de usuario y apoyar a los desarrolladores de back-end.
- Un equipo control de calidad para cerciorar que el código fuente cumple las políticas de la empresa o descubrir errores.
- Tener un equipo de control de calidad que se encargue de dar retroalimentación del código fuente desarrollado para apoyar el aprendizaje de los programadores.

BIBLIOGRAFÍA

1. About. (s.f.). Obtenida en el día 4 de febrero del 2020 desde <https://git-scm.com/about>
2. About npm. (s.f.). Obtenido en el día 4 de febrero del 2020 desde <https://docs.npmjs.com/about-npm/>
3. An Overview of AngularJS for Managers. (2014, agosto 14). Obtenido en el día 5 de febrero de 2020 desde <https://andrewaustin.com/an-overview-of-angularjs-for-managers/>
4. API Endpoints - What Are They? Why Do They Matter? (s.f.). Obtenido en el día 9 de noviembre del 2019 desde <https://smartbear.com/learn/performance-monitoring/api-endpoints/>.
5. Arenaza, V. A. E., Canseco, R. F. M., & Sosa, M. F. (2007). Informática i y ii. Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>.
6. Back End: Definition of Back End by Lexico. (s.f.). Obtenido en el día 9 de noviembre del 2019 desde https://www.lexico.com/en/definition/back_end.
7. Cateora, P. R. X., Gilly, M. C., & Graham, J. L. (2014). Marketing internacional (16a. ed.). Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>.
8. Coderthemes. (s.f.). Obtenido en el día 3 de noviembre del 2019 desde <https://coderthemes.com/minton/layouts/vertical/blue/index.html>.
9. Cohen, K. D. (2009). Tecnologías de información en los negocios (5a. ed.). Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>.
10. Christensson, P. (2013, febrero 5). Web Development Definition. Obtenido en el día 19 de octubre del 2019 desde <https://techterms.com>
11. Christensson, P. (2013, marzo 7). Framework Definition. Obtenido en el día 19 de octubre del 2019 desde <https://techterms.com>
12. Dayley, B., & Dayley, B. (2017). Node.js, MongoDB and Angular web development: the definitive guide to using the Mean stack to build web applications (2da ed.). Boston: Addison-Wesley.
13. Difference Between Front-End & Back-End Developer. (2019, October 9). Obtenido en el día 19 de octubre del 2019 desde <https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end>.

14. Front End: Definition of Front End by Lexico. (s.f Obtenido en el día 9 de noviembre del 2019 desde https://www.lexico.com/en/definition/front_end).
15. Gavigan, D. (2018, mayo 25). The History of Angular Obtenido en el día 5 de febrero del 2020 desde <https://medium.com/the-startup-lab-blog/the-history-of-angular-3e36f7e828c7>
16. gurumoorthyP, arvindpdmn. (2019, enero 24). Transpiler. Obtenido en el día 7 de febrero del 2020 desde <https://devopedia.org/transpiler>
17. Hallberg, B. (2007). Fundamentos de redes (4a. ed.). Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>.
18. Howson, C. (2009). Business intelligence: Estrategias para una implementación exitosa. Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>.
19. Introduction to Node.js (s.f.). Obtenido en el día 7 de febrero del 2020 desde <https://nodejs.dev/introduction-to-nodejs>
20. Jariwala, S. (2020, enero 18). What is Version Control and Version Control System? Obtenido en el día 4 de febrero del 2020 desde <https://www.toolsqa.com/git/version-control-system/>
21. Jones, C. (2008). Estimación de costos y administración de proyectos de software (2a. ed.). Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>.
22. Node.js. (s.f.). About. Obtenido en el día 3 febrero del 2020 desde <https://nodejs.org/en/about/>
23. Norton, P. (2014). Introducción a la computación (6a. ed.). Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>
24. O'Brien, J. A. (2006). Sistemas de información gerencial (7a. ed.). Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>
25. Pressman, R. (2010). Ingeniería del software: Un enfoque práctico (7a. ed.). Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>
26. Prieto, E. A., Lloris, R. A., & Torres, C. J. C. (2006). Introducción a la informática (4a. ed.). .). Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>

27. registry The JavaScript Package Registry. (s.f.). Obtenido en el día 4 de febrero del 2020 desde <https://docs.npmjs.com/misc/registry>
28. Server-side web frameworks. (s.f.). Obtenido en el día 19 de octubre del 2019 desde https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks.
29. The V8 JavaScript Engine (s.f.). Obtenido en el día 7 de febrero del 2020 desde <https://nodejs.dev/the-v8-javascript-engine>
30. TypeScript. (2016, agosto 11). Obtenido en el día 7 de febrero del 2020 desde <https://www.cleverism.com/skills-and-tools/typescript/>
31. Villarreal, D. L. G. S. (2007). Introducción a la computación: Teoría y manejo de paquetes (2a. ed.). Obtenido en el día 30 de noviembre del 2019 desde <https://ebookcentral.proquest.com>.
32. WebMedia, H. (s.f.). ¿Qué Ofrecemos? Obtenido en el día 19 de noviembre del 2019 desde <http://www.impactmobilehn.com/>.