



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PRÁCTICA PROFESIONAL

ACKLEN AVENUE / HERO UNIT

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTADO POR:

11641196 JOSUÉ DAVID RODRÍGUEZ HERNÁNDEZ

ASESOR: LIC. TANIA LUCILA MEZA AMADOR

CAMPUS TEGUCIGALPA; ENERO, 2021

Dedico esta tesis a mis padres, Fausto Martín y Lourdes Patricia, en agradecimiento al gran sacrificio que ellos han hecho para poder darme la oportunidad de recibir educación de calidad.

A mis hermanos, Martín Alberto y Fausto Daniel, por el apoyo incondicional y los consejos de vida que llevo guardados dentro de mí.

A mi tío Jorge Irula (QEPD) y mi tía Edith Rodríguez (QEPD), quienes siempre me recordaban lo orgulloso que los hacía sentir.

A mi mejor amiga, Elena Cossette, por el apoyo emocional durante los años que cursaba mi carrera universitaria.

Finalmente, a la ingeniera Laura Cabrera, al ingeniero Reynod Bocanegra, al ingeniero Román Pineda y a la licenciada Keren Vásquez, quienes fueron una parte fundamental en mi desarrollo personal, académico y profesional.

AGRADECIMIENTOS

Se agradece a Acklen Avenue por proporcionar un espacio de trabajo para el practicante Josué David Rodríguez Hernández y a UNITEC por dar soporte y asesoría en el transcurso de la práctica profesional.

RESUMEN EJECUTIVO

El presente informe se entrega como requisito para optar al título profesional de Ingeniero en Sistemas Computacionales, donde se describen las actividades desarrolladas durante la práctica profesional realizada de julio del 2020 a diciembre del 2020. La práctica se realizó en la empresa estadounidense Acklen Avenue en la división Hero Unit como desarrollador de software en la plataforma Peeplo, realizando las tareas de corrección de errores, desarrollo de nuevas características y administración de la base de datos.

En el año 2018 se comenzó el desarrollo de una plataforma con el objetivo de administrar empleados en una organización. En esta plataforma se puede ver la información detallada de cada uno de los empleados, registrar las horas trabajadas por los empleados y marcarlas como pagadas, controlar en qué proyectos o departamentos están actualmente asignados los empleados y llevar un control de los días libres que soliciten los empleados. Esta plataforma inicialmente se llamó Kiwi y luego fue renombrada a Peeplo, que es una variación de la palabra en inglés "People", que en español se traduce a "Personas".

El departamento de desarrolladores de software, donde se ha ejercido la práctica, es el encargado de manejar la mejora continua de la plataforma en cuestión, corrigiendo los errores reportados por los usuarios finales y por los ingenieros analistas de calidad y desarrollando las nuevas características indicadas por el dueño del producto. También, es el encargado de proveer todos los detalles técnicos ligados a las nuevas solicitudes del dueño del producto, así como proveer una estimación de tiempo que tardará el desarrollo de una tarea.

Actualmente, la plataforma está siendo usada por los miembros de Acklen Avenue y en los próximos meses se espera su lanzamiento oficial al mercado.

ÍNDICE DE CONTENIDO

I.	Introducción.....	1
II.	Generalidades de la Empresa.....	3
2.1	Descripción de la Empresa.....	3
2.2	Descripción del Departamento.....	4
2.3	Objetivos del Puesto.....	4
2.3.1	Objetivo General.....	4
2.3.2	Objetivos Específicos.....	4
III.	Marco Teórico.....	5
IV.	Desarrollo.....	17
4.1	Descripción del Trabajo Realizado.....	17
4.1.1	Curso Introductorio a la Empresa.....	17
4.1.2	Generalidades de la Plataforma Peeplo y de las Reuniones.....	19
4.1.3	Corrección de errores en la Plataforma Peeplo.....	23
4.1.4	Implementación de nuevas características en la Plataforma Peeplo.....	37
4.2	Cronograma de Actividades.....	55
V.	Conclusiones.....	56
VI.	Recomendaciones.....	57
	Bibliografía.....	58

ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Etapas de un Sprint	6
Ilustración 2 - Equipo Scrum.....	7
Ilustración 3 - Metodología Scrum.....	7
Ilustración 4 - Componente en React.....	8
Ilustración 5 - Actualización del DOM y DOM Virtual en React.....	9
Ilustración 6 - Archivo de configuración TypeORM.....	11
Ilustración 7 - Modelo de una entidad en TypeORM.....	11
Ilustración 8 - Definición de un DTO en TypeORM.....	12
Ilustración 9 - Comparación entre modelo orientado a objetos y modelo relacional	13
Ilustración 10 - PgAdmin Dashboard	14
Ilustración 11 - Ejemplo de consulta en pgAdmin	14
Ilustración 12 - Alcance de aplicaciones de Auth0.....	15
Ilustración 13 - Comparación entre JavaScript y TypeScript.....	17
Ilustración 14 - Inicio de Sesión en Peeplo	19
Ilustración 15 - Vista de Perfil de Empleado.....	20
Ilustración 16 - Vista de Registro de horas en Peeplo	20
Ilustración 17 - Aplicación utilizando React con Redux	21
Ilustración 18 - Actividades realizadas durante una semana de trabajo	22
Ilustración 19 - Registro de Hora en Peeplo.....	24
Ilustración 20 - Creación de proyectos duplicados en Peeplo	25
Ilustración 21 - Modificaciones de los campos de fechas.....	26
Ilustración 22 - Error de permisos de empleado.....	27

Ilustración 23 - Envío de muchas solicitudes para guardar un registro de tiempo.....	28
Ilustración 24 - Botón deshabilitado	29
Ilustración 25 - Registro con hora de finalización igual a la hora actual.....	30
Ilustración 26 - Actividades con duración de 0 minutos.....	31
Ilustración 27 - Valor booleano en la columna isActive.....	31
Ilustración 28 - Comportamiento del estado corregido	32
Ilustración 29 - Filtro de empleados mostrando primer nombre.....	32
Ilustración 30 - Filtro de empleados mostrando nombre completo	33
Ilustración 31 - Contador de caracteres en información de pago	34
Ilustración 32 - Ordenamiento erróneo de la tabla de empleados	34
Ilustración 33 - Ordenamiento correcto de la tabla de empleados.....	35
Ilustración 34 - Plataforma caída al ver la información de un empleado.....	36
Ilustración 35 - Vista de cargado	37
Ilustración 36 - Registro editable contra registro no editable.....	38
Ilustración 37 - Registros que se pueden archivar y registros que no se pueden archivar	39
Ilustración 38 - Botón para archivar una organización.....	40
Ilustración 39 - Ventana de confirmación para archivar una organización	40
Ilustración 40 - Resultado final al archivar una organización	41
Ilustración 41 - Desarchivar una organización.....	42
Ilustración 42 - Resultado final al desarchivar una organización	42
Ilustración 43 - Asignación de permisos en una organización.....	43
Ilustración 44 - Vista de horas trabajadas con permisos de Administrador.....	44
Ilustración 45 - Vista de horas trabajadas con permisos de Gerente.....	44

Ilustración 46 - Error al crear organizaciones con nombres repetidos.....	45
Ilustración 47 - Estructura original y modificada de la tabla Employee.....	46
Ilustración 48 - Estructura original y modificada de la tabla User.....	46
Ilustración 49 - Enlace de invitación en perfil de empleado.....	47
Ilustración 50 - Enlace de invitación a Peeplo en navegador.....	48
Ilustración 51 - Vista de validación de invitación	48
Ilustración 52 - Vista de invitación	49
Ilustración 53 - Vista de invitación expirada.....	50
Ilustración 54 - CSV de horas trabajadas con la columna "Description"	50
Ilustración 55 - Campo de Contacto de Emergencia.....	51
Ilustración 56 - Límite de 48 horas para registrar horas	51
Ilustración 57 - Organización creada sin empleados	52
Ilustración 58 - Estructura original y modificada de la tabla History	53
Ilustración 59 - Historial de acciones.....	54
Ilustración 60 - Diagrama de Gantt.....	55

LISTA DE SIGLAS Y GLOSARIO

AA	Acklen Avenue
API	Interfaz de Programación de Aplicaciones (Application Programming Interface)
CRAI	Centro de Recursos para el Aprendizaje y la Investigación
CSV	Valores Separados por Coma (Comma Separated Values)
DOM	Modelo de Objetos del Documento (Document Object Model)
DTO	Objeto de Transferencia de Datos (Data Transfer Object)
ECDSA	Algoritmo de Firma Digital de Curva Elíptica
FI	Facultad de Ingeniería
FRP	Programación Funcional Reactiva (Functional Reactive Programming)
GM	Guild Master
HMAC	Código de Autenticación de Mensajes en Clave-Hash
HU	Hero Unit
JWT	JSON Web Token
JSON	Notación de Objetos JavaScript (JavaScript Object Notation)
MVCC	Control de Concurrencias Multiversión (Multiversion Concurrency Control)
OE	Objetivo Específico
OG	Objetivo General
ORM	Mapeo Objeto-Relacional (Object-Relational Mapping)
PF	Programación Funcional
POO	Programación Orientada a Objetos
PP	Práctica Profesional
QA	Aseguramiento de Calidad (Quality Assurance)
RDBMS	Sistema de Gestión de Bases de Datos Relacionales (Relational Database Management System)
RSA	Sistema criptográfico Rivest, Shamir y Adleman

SDK	Kit de Desarrollo de Software (Software Development Kit)
SQL	Lenguaje de Consulta Estructurada (Structured Query Language)
Ágil	Es un proceso que permite al equipo dar respuestas rápidas e impredecibles a las valoraciones que reciben sobre su proyecto. (DevOps - ¿Qué es metodología ágil?, s.f.)
Back-end	Parte del desarrollo de una aplicación que se encarga de que toda la lógica de ella funcione. Es la parte de la aplicación que el usuario final no ve. (Chapaval, 2017)
Badgr	Plataforma en línea para emitir insignias digitales y micro credenciales. (Badgr, s.f.)
Decorador	Etiquetas de TypeScript que permiten añadir anotaciones y metadatos o cambiar el comportamiento de clases, propiedades y métodos (Boscan, 2018)
DevOps	Un modo de abordar la cultura, la automatización y el diseño de plataformas para aportar el mayor valor empresarial y capacidad de respuesta mediante la prestación ágil de servicios de alta calidad. (Red Hat, s.f.)
Endpoint	Es el punto de entrada de una interfaz de programación de aplicaciones en un canal de comunicación cuando dos sistemas interactúan. (Endpoint - What is an API Endpoint?, s.f.)
Framework	Una estructura conceptual y tecnológica de soporte definida en base a la cual otro proyecto de software puede ser organizado y desarrollado. (Ecured, s.f.)
Front-end	Parte del desarrollo de una aplicación que tiene interacción directa con los usuarios. Es todo lo que el usuario ve en la pantalla. (Chapaval, 2017)

Full Stack	Desarrollador capaz de crear software de cliente y de servidor. Es decir, maneja tecnologías front-end y back-end, como también bases de datos. (W3Schools, s.f.)
GitLab	Plataforma DevOps completa que crea un flujo de trabajo de software optimizado, librando de las limitaciones de una cadena de herramientas enlazadas. (GitLab Inc., s.f.)
GSuite	Es un paquete de Google que ofrece herramientas para mensajería y colaboración que satisfacen las necesidades de la empresa, incrementan la productividad y reducen los costos. (Google, s.f.)
NestJS	Framework progresivo de Node.js para crear aplicaciones del lado del servidor eficientes, confiables y escalables. (NestJS, s.f.)
Node.js	Entorno de ejecución de JavaScript orientado a eventos asíncronos, diseñado para crear aplicaciones network escalables. (Node.js, s.f.)
Peeplo	Plataforma en línea para administración de empleados. (Peeplo, s.f.)
PgAdmin	Herramienta oficial para administrar una base de datos desarrollada en PostgreSQL. (PostgreSQL Tools, s.f.)
PostgreSQL	Sistema de bases de datos objeto relacional de código abierto. (PostgreSQL, s.f.)
React	Una biblioteca de JavaScript para construir interfaces de usuario. (React, s.f.)
Scrum	Es un framework simple que promueve la colaboración en los equipos para lograr desarrollar productos complejos. (Francia, s.f.)

I. INTRODUCCIÓN

El desarrollo de software ha hecho que la calidad de vida de los seres humanos mejore de numerosas formas. Actualmente, existen softwares que permiten controlar todo desde el celular, sin necesidad de moverse de lugar. Muchas compañías están optando a cambiar sus herramientas de trabajo, dejando atrás lo tradicional e implementando software que les permita agilizar sus procesos y desarrollar sus tareas de manera eficaz y rápida.

La profesión de desarrollador de software ha evolucionado desde sus inicios en la década de 1940; el que ejerce esta profesión no solamente busca crear un sistema, también debe maximizar su calidad empleando diferentes técnicas, como patrones de diseño, estructuras de datos que maximicen la eficiencia de este y aplicar principios para escribir código para que su programa se vuelva entendible y mantenible.

Con la reciente pandemia de COVID-19 y la cuarentena que trajo con ella, muchas empresas se vieron forzadas a cambiar su modalidad de trabajo presencial a modalidad remota, capacitando a sus empleados en tecnologías nuevas que les permitan desarrollar sus labores desde su casa. Sin embargo, uno de los retos que presentan las empresas es saber con exactitud las horas de entrada y salida de sus empleados, las tareas que han desarrollado durante un periodo de tiempo, cuánto tiempo les han dedicado a sus labores y en qué parte de los días son más activos.

Acklen Avenue, en su división Hero Unit, presenta una solución al problema previamente planteado a través de la plataforma Peeplo. Peeplo es una plataforma en línea que, mediante autenticación de Google, permite la administración de organizaciones y empleados. Para los administradores, permite crear organizaciones, empleados y proyectos, asignar empleados a proyectos, ver una lista detallada de las horas trabajadas por cada uno de los empleados y registrar qué horas han sido pagadas. Para los empleados, permite registrar las actividades que han realizado en el día, cuánto tiempo le dedicaron a cada actividad, a qué proyecto pertenece la actividad realizada y ver qué horas han sido pagadas.

En este informe se presenta una descripción general de la empresa Acklen Avenue, se describe el puesto de trabajo ocupado durante el desarrollo de la práctica profesional, se mencionan los objetivos generales y específicos los cuales se desean llevar a cabo.

En el marco teórico se mencionan las metodologías utilizadas, las tecnologías empleadas para el desarrollo de la plataforma.

La sección de desarrollo describe las actividades realizadas durante la práctica profesional y se ejemplifica mediante un diagrama de Gantt.

II. GENERALIDADES DE LA EMPRESA

2.1 DESCRIPCIÓN DE LA EMPRESA

Creamos aplicaciones web y móviles personalizadas a la velocidad de la oportunidad, para los innovadores que quieren hacer cosas. En un mundo donde el desarrollo de software es a menudo lento, costoso, arriesgado y doloroso, se necesita un sistema simple, confiable, consistente y de alta calidad (Acklen Avenue, s.f.).

Nuestro equipo de profesionales de desarrollo de software ágil se apasiona por el código limpio y las experiencias de usuario simples. Gracias a los esfuerzos heroicos de nuestro equipo y las ideas asombrosas de nuestros clientes, Acklen Avenue entrega amorosamente software diariamente desde Nashville, Tennessee y Honduras a socios de todo el mundo (Acklen Avenue, s.f.).

En cada compromiso, entendemos el producto y los objetivos del cliente tan bien que cada miembro del equipo podría dar una presentación en nombre del cliente. Somos dueños del éxito de nuestros clientes (Acklen Avenue, s.f.).

Acklen Avenue se rige por cinco valores fundamentales y se aseguran de que cada empleado comprenda lo que realmente significan cada uno de ellos (Mezquita, 2018):

Integridad (Integrity): haremos lo correcto, incluso si duele, al no participar o apoyar la deshonestidad, la corrupción o la injusticia hacia cualquier grupo de personas. Haremos lo mejor para nuestros clientes, incluso si eso significa perder dinero o decir adiós. Seremos honestos en todo lo que hagamos.

Dominio (Mastery): seremos mejores que ayer encontrando formas de mejorar cada día y dando la bienvenida a mejores formas de hacer las cosas. Vamos a capacitar a nuestros compañeros de equipo para que lo hagan mejor, facilitando el aprendizaje para aquellos dentro y fuera de nuestra organización. Agradeceremos los comentarios desde todos los ángulos y siempre exigiremos soluciones elegantes.

Calidad (Quality): crearemos productos de alta calidad, por dentro y fuera, asumiendo la propiedad del éxito de nuestros clientes. Dedicaremos el tiempo necesario para hacer un gran

trabajo para ayudar a nuestros clientes a obtener lo que realmente necesitan, no solo lo que dijeron que necesitaban hace 6 meses. Entregaremos un producto que haga feliz a nuestro cliente y sea digno de repetir negocios.

Agilidad (Agility): entregaremos productos valiosos, flexibles y mantenibles tan a menudo como sea posible al asociarnos con nuestros clientes para maximizar el aprendizaje y el valor del producto mediante el uso de un enfoque ágil para el desarrollo de software. Permitiremos alentarnos para acelerar.

Amor (Love): nos preocuparemos por nuestros colegas, tanto profesional como personalmente, al mismo tiempo que cuidamos a nuestros clientes y su éxito.

2.2 DESCRIPCIÓN DEL DEPARTAMENTO

El departamento de Desarrolladores de Software es el encargado del desarrollo y mantenimiento de programas solicitados por los clientes de la empresa empleando diferentes tecnologías front-end y back-end. La mayoría de los integrantes están ubicados en Honduras, y trabajan en conjunto con integrantes en Estados Unidos, Perú y España.

2.3 OBJETIVOS DEL PUESTO

2.3.1 OBJETIVO GENERAL

Continuar con el desarrollo de la plataforma Peeplo, implementando nuevas características al sistema, mejorando el rendimiento y la experiencia de usuario de esta y corrigiendo errores detectados por los usuarios.

2.3.2 OBJETIVOS ESPECÍFICOS

- Desarrollar las características solicitadas por el dueño del producto.
- Corregir errores reportados por los ingenieros de aseguramiento de calidad (QA) y por los usuarios de la plataforma en la etapa de producción.

III. MARCO TEÓRICO

Para obtener un mejor entendimiento del trabajo desarrollado en Acklen Avenue (AA), a continuación, se describen términos, herramientas y metodologías utilizadas durante el tiempo de práctica.

Peeplo es una plataforma orientada al área de recursos humanos donde los dueños y administradores de la empresa pueden llevar un mejor control sobre sus empleados, el registro de sus horas y el pago de estas. (Peeplo, s.f.)

Para el desarrollo de los productos de la empresa AA se utiliza la metodología ágil Scrum. Scrum es un marco de trabajo ágil para desarrollar productos y servicios innovadores. En una metodología ágil, se crea un backlog del producto, una lista de características necesarias para desarrollar un producto exitoso, de acuerdo con las especificaciones del cliente. A cada elemento de esta lista se le debe asignar una prioridad, y se trabajarán las tareas de prioridad más alta primero (Rubin, 2012).

El trabajo se realiza en iteraciones de tiempo cortas, variando entre una semana a un mes, llamadas sprints. Durante todos los días del sprint, el equipo debe tener daily scrums, reuniones diarias no mayores de quince minutos en donde el equipo responde tres preguntas:

1. ¿Qué hice ayer?
2. ¿Qué haré hoy?
3. ¿Qué obstáculos se me presentaron?

Usualmente, el equipo multifuncional no puede llevar a cabo todas las tareas registradas en el backlog durante este corto periodo de tiempo, es por eso por lo que antes de iniciar un sprint, Scrum realiza una reunión en donde se planea qué tareas se van a llevar a cabo durante ese sprint (Rubin, 2012).

Al final de cada sprint, se debe llevar a cabo una reunión en donde el equipo se reúne con las partes interesadas (clientes, patrocinadores, entre otros) para mostrar el trabajo realizado y recibir retroalimentación de este. Una vez culminada esta reunión, el proceso se repite nuevamente, planeando el siguiente sprint (Rubin, 2012).

La Ilustración 1 muestra las etapas de un sprint, desde su planeación hasta la retroalimentación.

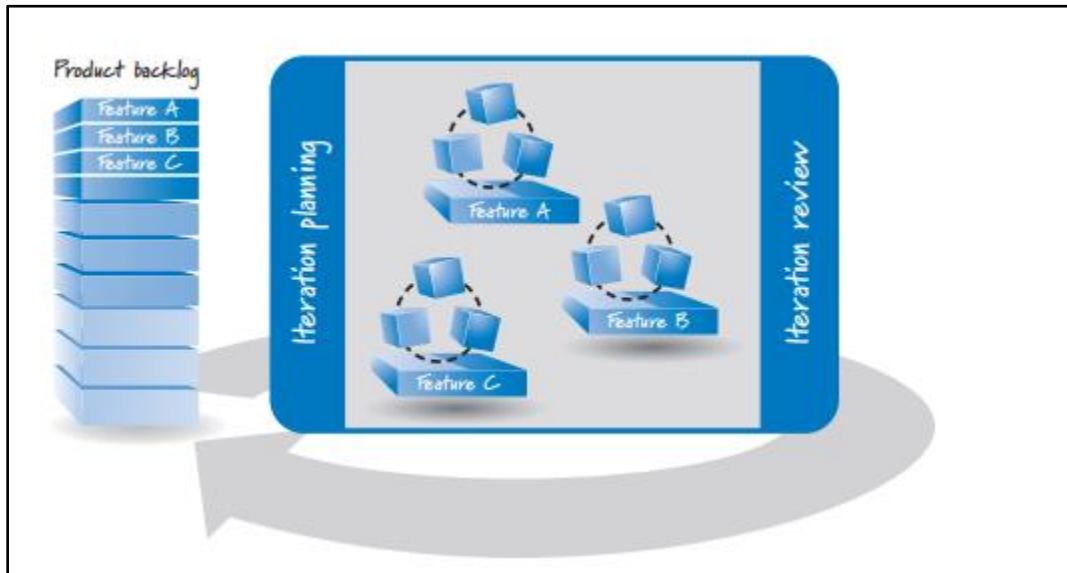


Ilustración 1 - Etapas de un Sprint

Fuente: (Rubin, 2012).

El marco de trabajo Scrum presenta tres roles principales. Rubin hace la aclaración de que pueden existir más roles diferentes a estos, pero el marco de trabajo Scrum requiere estos tres.

1. Propietario del Producto: es el punto central empoderado del liderazgo del producto. Él es la única autoridad responsable de decidir qué características y funcionalidad construir y ordena en qué construirlas.
2. Scrum Master: se asegura que se esté siguiendo la metodología Scrum (valores, prácticas y principios). Es el encargado de ayudar al equipo a resolver problemas y quitar obstáculos que afecten la productividad del equipo.
3. Equipo de desarrollo: es un equipo multifuncional y multidisciplinario responsable de diseñar, construir y probar el producto por desarrollar. Este equipo está comprendido por diferentes especialidades, como arquitectos de software, programadores, QAs, administradores de bases de datos, diseñadores, entre otros.

La Ilustración 2 muestra los tres principales roles en Scrum.

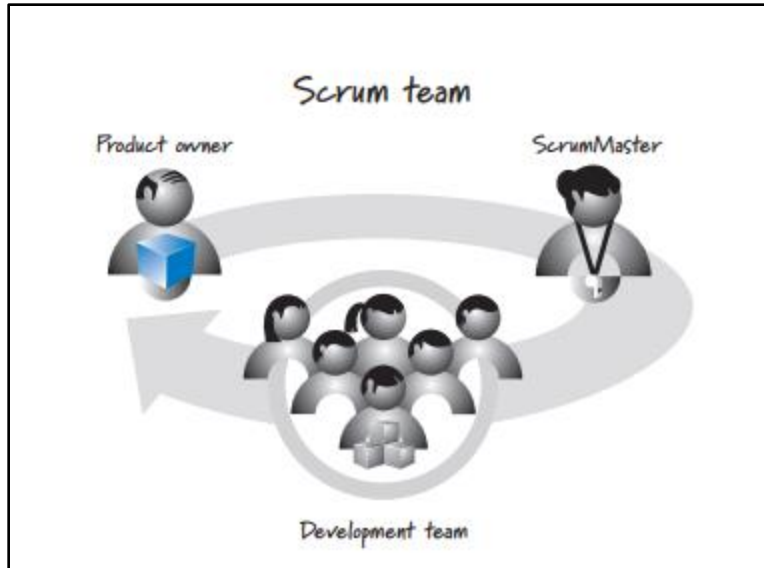


Ilustración 2 - Equipo Scrum

Fuente: (Rubin, 2012).

La Ilustración 3 presenta más a detalle las diferentes etapas de la metodología Scrum.

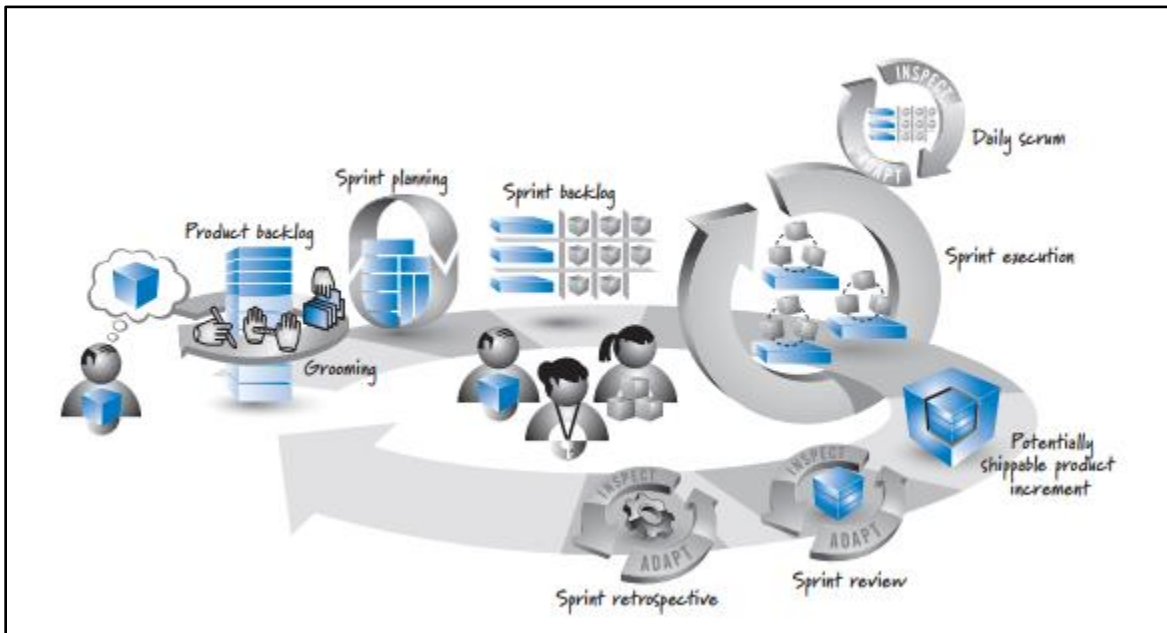


Ilustración 3 - Metodología Scrum

Fuente: (Rubin, 2012).

Peeplo es una aplicación full-stack desarrollada en React con TypeScript para el front-end, y para el back-end se utilizó NestJS con base de datos en PostgreSQL. Para la autenticación de usuarios se utilizó el servicio de Auth0.

React (también conocido como ReactJS) es una librería JavaScript de código abierto desarrollada por Facebook enfocada a la visualización. Esta tecnología permite el desarrollo de interfaces de usuario de forma sencilla mediante componentes interactivos y reutilizables (Campillo, 2020). React permite diseñar vistas simples para cada estado en una aplicación, encargándose de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien. Una de las principales características de React es que, gracias a las vistas declarativas, permite que el código se vuelva predecible y, por ende, fácil de depurar (React, s.f.).

React se basa principalmente en la creación de componentes encapsulados que manejen su propio estado que, al ser unidos, se convierten en una interfaz de usuario compleja. De igual manera, permite el paso de datos de forma sencilla a través de los componentes de la aplicación, manteniendo el estado fuera del Modelo de Objetos del Documento (DOM).

La Ilustración 4 muestra el código para la creación de un componente en React.

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hola {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="Taylor" />,
  document.getElementById('hello-example')
);
```

Ilustración 4 - Componente en React

Fuente: (React, s.f.)

Una de las principales ventajas de utilizarlo en lugar de otros frameworks de desarrollo web es el uso del DOM virtual que React ofrece. Es capaz de generar el DOM de forma dinámica, realizando los cambios en una copia en memoria y después los compara con la versión actual del DOM, de esta forma evita renderizar toda la página cada vez que haya cambios, simplemente se aplica dicho cambio al componente que haya sido actualizado. Esto proporciona una mejor experiencia de usuario, además de un alto rendimiento y fluidez. (Campillo, 2020)

La Ilustración 5 ejemplifica cómo React actualiza únicamente los componentes que hayan sido actualizados. Nótese que la actualización se hace en forma de cascada; es decir, un componente que haya sido actualizado actualiza a todos los componentes que dependen de él.

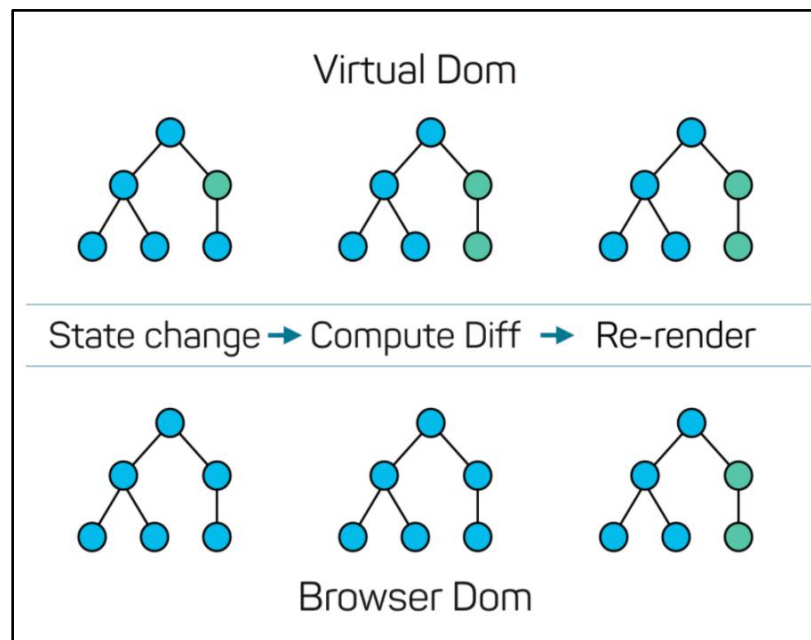


Ilustración 5 - Actualización del DOM y DOM Virtual en React

Fuente: (React, s.f.)

NestJS es un framework de código abierto progresivo de Node.js para crear aplicaciones del lado del servidor eficientes, confiables y escalables. Utiliza JavaScript progresivo, pero está desarrollado con y soporta complementemente TypeScript. De igual manera, combina elementos de Programación Orientada a Objetos (POO), Programación Funcional (PF) y Programación Funcional Reactiva (FRP) (NestJS, s.f.)

NestJS provee un nivel de abstracción superior en comparación a otros frameworks de Node.js, como Express y Fastify, pero de igual manera expone sus interfaces de programación de aplicaciones (APIs) directamente al desarrollador, dándole a este la libertad de usar la gran cantidad de módulos de terceros que están disponibles para la plataforma subyacente. (Node.js, s.f.)

NestJS se basa en tres conceptos básicos, definidos a continuación. (NestJS, s.f.)

- Extensibilidad: brinda una amplia flexibilidad permitiendo el uso de otras librerías gracias a su arquitectura modular.
- Versatilidad: ofrece un ecosistema adaptable que funciona como el fundamento para todo tipo de aplicaciones del lado del servidor.
- Progresividad: implementa las características recientes de JavaScript, introduciendo patrones de diseño y soluciones maduras al mundo de Node.js.

NestJS se apoya en TypeORM, el cual se define a continuación, para la comunicación e interacción con la base de datos. Es importante mencionar que la creación del esquema debe hacerse manualmente en la base de datos para que la aplicación de NestJS pueda detectarla al ser inicializada. (Ruiz Santiago, 2020)

TypeORM es un Mapeo Objeto-Relacional que puede ser ejecutado en diferentes frameworks como Node.js, Cordova, Ionic, React Native, entre otros y puede ser utilizado con TypeScript o JavaScript como soporte para el desarrollo de aplicaciones que utilicen bases de datos de lenguaje de consulta estructurado (SQL) y NoSQL. (TypeORM, s.f.)

ORM es una herramienta que provee un proceso de desarrollo simplificado automatizando las conversiones objeto a tabla y tabla a objeto, haciendo que el código sea fácil de actualizar, mantener y reutilizar. (Tutorialspoint, s.f.)

Las siglas ORM definen lo siguiente: (Tutorialspoint, s.f.)

- El Objeto se refiere al dominio o modelo de la aplicación.
- Lo Relacional se refiere a la relación entre las tablas en un Sistema de Gestión de Base de Datos Relacionales (RDBMS) como Oracle, MySQL, PostgreSQL, entre otros.
- El Mapeo se refiere a la conexión entre el modelo y los esquemas de las bases de datos.

Para poder conectar TypeORM con la base de datos, se debe crear un archivo de configuración que especifique, entre otras cosas, el gestor de base de datos que se utilizará, usuario y contraseña del RDBMS, nombre de la base de datos y directorio en donde se encuentran los archivos de entidades que servirán para el mapeo objeto a tabla.

La Ilustración 6 muestra un archivo de configuración para la conexión entre la base de datos y TypeORM.

```
import { TypeOrmModuleOptions } from '@nestjs/typeorm'

export const typeOrmConfig: TypeOrmModuleOptions = {
  type: 'mysql',
  host: 'localhost',
  port: 3306,
  username: 'root',
  password: '',
  database: 'tasks_db',
  entities: [__dirname + '/../**/*entity.{js,ts}'],
  synchronize: true
}
```

Ilustración 6 - Archivo de configuración TypeORM

Fuente: (Ruiz Santiago, 2020)

La Ilustración 7 muestra cómo debe modelarse una entidad con TypeORM. Se debe definir la entidad mediante el uso de decoradores de la misma manera en que está definida la tabla que se desea modelar en la base de datos.

```
import { BaseEntity, Entity, PrimaryGeneratedColumn, Column, Index } from 'typeorm';

@Entity()
@Index(['title'])
export class Task extends BaseEntity {
  @PrimaryGeneratedColumn({ type: 'int' })
  id: number;

  @Column({ type: 'varchar', nullable: false })
  title: string;

  @Column({ type: 'varchar', nullable: true })
  description: string;
}
```

Ilustración 7 - Modelo de una entidad en TypeORM

Fuente: (Ruiz Santiago, 2020)

Como se mencionó previamente, TypeORM automatiza las conversiones de objeto a tabla y tabla a objeto. Para lograr esto, se utilizan los Objetos de Transferencia de Datos (DTO). Un DTO es un esquema que define cómo se deben recibir los datos para realizar una acción sobre la base de datos y permite añadir funcionalidad para validar los datos recibidos.

La Ilustración 8 muestra una definición de un objeto para realizar la acción de inserción en la base de datos. Note que, mediante los decoradores, se define el tipo de dato que se espera y se realizan validaciones sobre un campo si este es opcional u obligatorio.

```
import { IsNotEmpty, IsOptional, IsString } from 'class-validator'

export class CreateTaskDto {
  @IsNotEmpty()
  @IsString()
  title: string;

  @IsOptional()
  @IsString()
  description: string;
}
```

Ilustración 8 - Definición de un DTO en TypeORM

Fuente: (Ruiz Santiago, 2020)

PostgreSQL es un sistema de base de datos objeto relacional de código abierto que utiliza y, al mismo tiempo, amplía el lenguaje de consulta estructurada (SQL). Este sistema, combinado con otras características, es capaz de almacenar y escalar de manera segura, las cargas de trabajo de datos más complicadas. (PostgreSQL, s.f.)

El término base de datos objeto relacional se usa para describir una base de datos que ha evolucionado desde el modelo relacional hasta una base de datos híbrida, que contiene ambas tecnologías (relacional y de objetos). (Universidad de Valencia, 2015)

En otras palabras, una base de datos objeto relacional es una base de datos relacional que implementa los conceptos de POO.

La Ilustración 9 muestra un comparativo entre el modelo de bases de datos orientado a objeto y el modelo relacional.

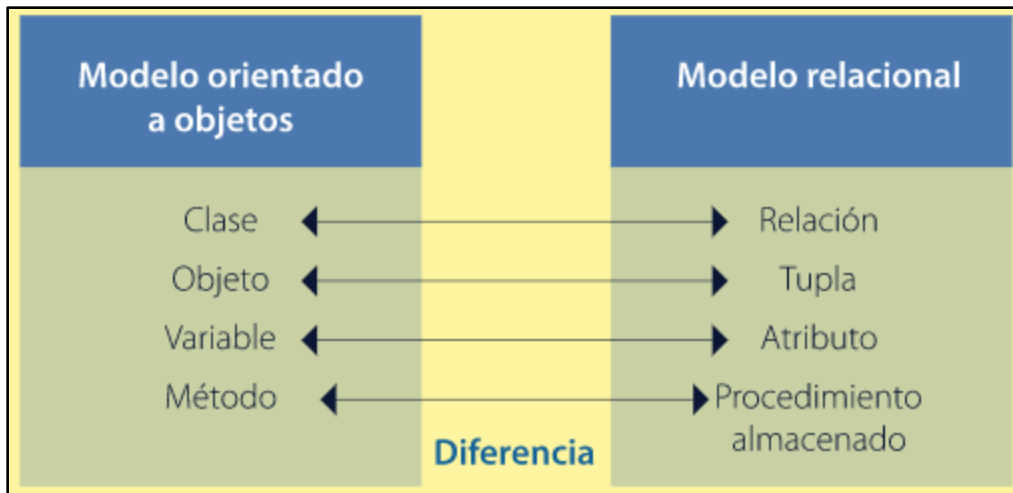


Ilustración 9 - Comparación entre modelo orientado a objetos y modelo relacional

Fuente: (Universidad Nacional Autónoma de México, s.f.)

Una de las principales características de PostgreSQL es el control de concurrencias multiversión (MVCC), un método que agrega una imagen o captura del estado actual de la base de datos a cada transacción, permitiendo hacer transacciones eventualmente consistentes, ofreciendo grandes ventajas en el rendimiento. (Dorantes, 2015)

PostgreSQL ofrece a los desarrolladores un mayor nivel de escalabilidad al no aplicar bloqueos de lectura al momento de realizar una transacción y también implementando "Hot-Standby". La documentación oficial de PostgreSQL define "Hot-Standby" como la capacidad de conectarse al servidor y ejecutar consultas de solo lectura mientras el servidor está en modo de recuperación de archivos o en modo de espera, lo cual es útil para fines de replicación y restauración de copias de seguridad al estado deseado. (PostgreSQL, s.f.)

Para visualizar y manipular una base de datos en PostgreSQL es necesario utilizar la herramienta oficial PgAdmin. Esta herramienta permite la ejecución de comandos SQL para consultas en la base de datos, así como la creación de total de una base de datos mediante una interfaz gráfica fácil e intuitiva. (Dorantes, 2015)

La Ilustración 10 muestra el dashboard de PgAdmin.

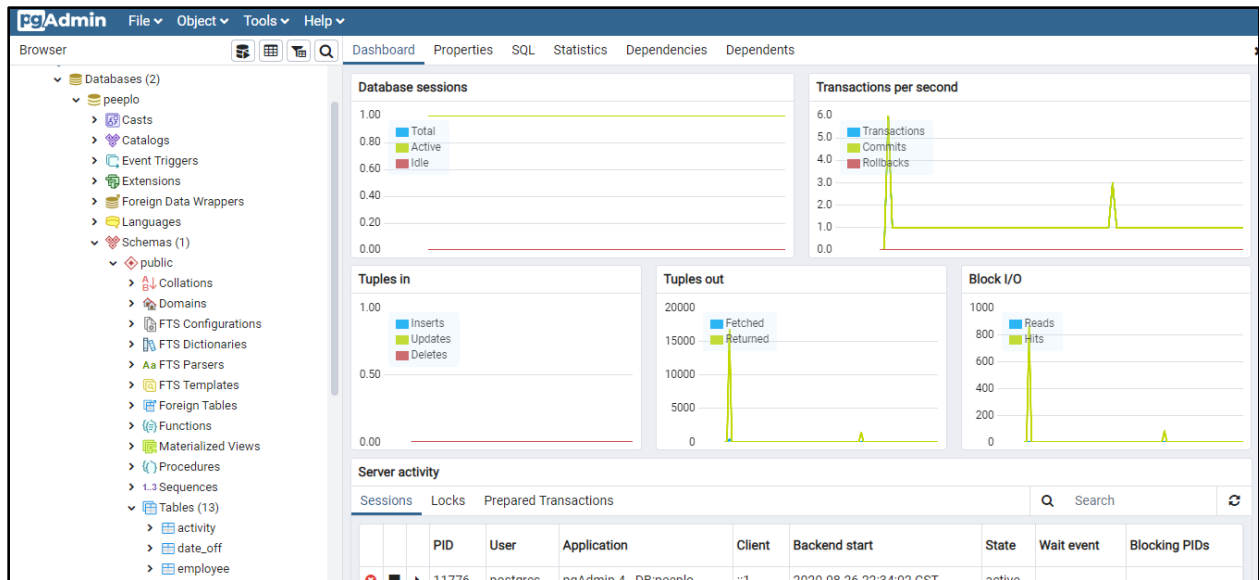


Ilustración 10 - PgAdmin Dashboard

Fuente: (PostgreSQL Tools, s.f.)

La Ilustración 11 muestra la realización de una consulta mediante la “Query Tool” de PgAdmin y el resultado de ésta.

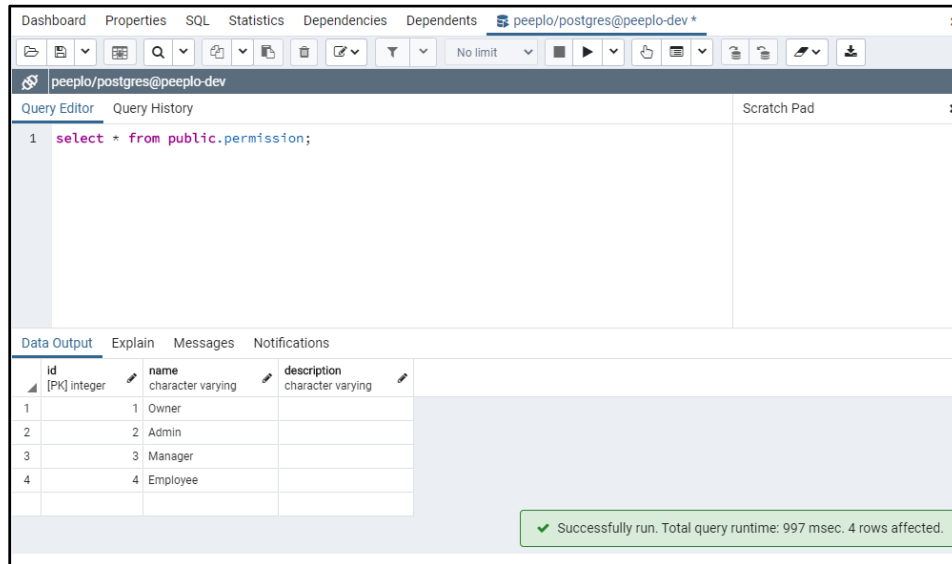


Ilustración 11 - Ejemplo de consulta en pgAdmin

Fuente: (PostgreSQL Tools, s.f.)

Auth0 es una herramienta flexible y sencilla de integrar para agregar servicios de autenticación y autorización a aplicaciones, reduciendo los costos, tiempo de implementación y riesgos de

brechas de seguridad que conlleva la creación de una API propia para autenticación y autorización. (Auth0, s.f.)

La Agencia Española de Protección de Datos (2020) define una brecha de seguridad como un incidente que afecta a datos de carácter personal. Este incidente puede tener un origen accidental o intencionado y además puede afectar a datos tratados digitalmente o en formato papel. Se trata de un suceso que ocasione destrucción, pérdida, alteración, comunicación o acceso no autorizado a datos personales.

Auth0 puede conectarse con todo tipo de aplicación, escrita en cualquier lenguaje de programación y utilizando cualquier pila tecnológica (ver la Ilustración 12 para ver el alcance que tiene Auth0). También, se pueden definir los proveedores de identidad con los que desea que los usuarios se autenticuen en la aplicación. Una vez definidos los proveedores, se debe conectar la aplicación con el kit de desarrollo de software (SDK) o con la API de Auth0 y así, cada vez que un usuario intente autenticarse, Auth0 verificará su identidad y enviará la información requerida a la aplicación. (Auth0, s.f.)

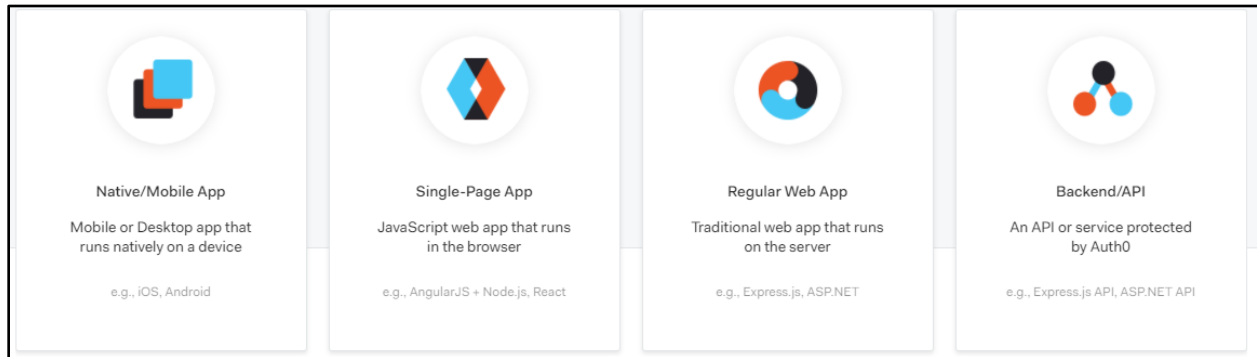


Ilustración 12 - Alcance de aplicaciones de Auth0

Fuente: (Auth0, s.f.)

Entre los métodos de autenticación que ofrece Auth0 se encuentra la autenticación mediante Google, convirtiéndose en una facilidad para las empresas que utilizan GSuite para sus asuntos internos.

GSuite es un paquete de Google que ofrece herramientas para mensajería y colaboración que satisfacen las necesidades de la empresa, incrementan la productividad y reducen los costos. Las

herramientas que ofrece incluyen Gmail, Documentos de Google, Google Drive, Calendario y Google Meet. (Google, s.f.)

Auth0 le asigna a cada aplicación un identificador, una cadena alfanumérica única para cada aplicación y también le asigna una llave secreta, una contraseña que debe mantenerse confidencial en todo momento para evitar la filtración de información sensible de los usuarios. (Auth0, s.f.)

Auth0 utiliza JSON Web Tokens (JWT) para brindar seguridad a todas las transacciones que se realicen dentro de una plataforma. Las siglas JSON significan Objeto en Notación JavaScript. JWT es un estándar abierto que define una forma compacta y autónoma de transmitir información de forma segura entre las partes involucradas como un objeto JSON. Esta información se puede verificar y confiar porque está firmada digitalmente. Los JWT se pueden firmar utilizando una llave secreta utilizando un código de autenticación de mensajes en clave-hash (HMAC) o un par de llaves públicas/privadas usando el sistema criptográfico Rivest, Shamir y Adleman (RSA) o el Algoritmo de Firma Digital de Curva Elíptica (ECDSA). (JWT, s.f.)

JWT es utilizado mayormente para autorización e intercambio de información:

- Autorización: cuando el usuario inicia sesión en una aplicación, cada solicitud de información subsecuente incluirá un JWT, permitiéndole al usuario acceder a rutas, servicios y recursos asociados con ese token.
- Intercambio de información: al implementar algoritmos de cifrado y llaves privadas, se puede confirmar la integridad de las partes involucradas en el intercambio.

Un JWT se compone de tres partes: el encabezado, la carga útil y la firma. (JWT, s.f.)

- El encabezado contiene el tipo de token que se está enviando, en este caso un JWT, y el algoritmo que se está utilizando para el cifrado.
- La carga útil contiene las afirmaciones, declaraciones sobre la entidad (el usuario) e información adicional. Existen tres tipos de afirmaciones: registrados, públicos y privados.
- La firma se genera aplicando un algoritmo de cifrado especificado en el encabezado y aplicándolo a las dos secciones anteriores utilizando una llave secreta.

Las herramientas previamente mencionadas (React, NestJS y TypeORM) utilizan el lenguaje de programación TypeScript. TypeScript es un lenguaje de programación de código abierto desarrollado y mantenido por Microsoft basado en JavaScript con la implementación de definiciones de tipo estático. Los tipos proporcionan una forma de describir la estructura de un objeto, proporcionando una mejor documentación y permitiendo que TypeScript valide que el código funcione correctamente. La definición de tipos puede ser opcional porque la inferencia de tipos permite obtener los mismos resultados sin escribir código adicional. (TypeScript, s.f.)

La Ilustración 13 muestra un bloque de código haciendo comparación entre la declaración de una variable en JavaScript y su equivalente en TypeScript.

```
let helloWorld = "Hello World";  
// ^ = let helloWorld: string
```

Ilustración 13 - Comparación entre JavaScript y TypeScript

Fuente: (TypeScript, s.f.)

IV. DESARROLLO

4.1 DESCRIPCIÓN DEL TRABAJO REALIZADO

4.1.1 CURSO INTRODUCTORIO A LA EMPRESA

Durante la primera semana de práctica en AA, el practicante desarrolló el curso introductorio a la empresa, cubriendo una diversidad de temas tanto técnicos como no técnicos. Se utilizó la plataforma Badgr para llevar registro de las diferentes etapas del curso. Christian García, Coordinador de Carrera en AA, afirmó que culminar este curso es de suma importancia por una variedad de razones, pero la más importante es porque en AA todos somos una familia, y debemos cuidarnos y apreciarnos como tal.

A continuación, se presenta una lista de los tópicos cubiertos en el curso introductorio:

- Introducción al trabajo remoto.
- Manejo de relaciones con el cliente.

- Herramientas que se utilizan en AA.
 - Badgr
 - GitHub
 - Gmail
 - Google Calendar
 - Google Meet
 - Zoom
- Conocer al equipo de AA/ Hero Unit (HU)
 - Conocer a los Guild Masters (GM) de cada área (Diseño, Desarrollo, HU).
- Visión y Valores de la empresa.

4.1.2 GENERALIDADES DE LA PLATAFORMA PEEPLO Y DE LAS REUNIONES

Al concluir el curso introductorio a la empresa, el practicante fue asignado al proyecto Peeplo. Se debe mencionar que Peeplo es una plataforma que, basado en el historial de GitLab, comenzó su desarrollo en 2018 y hoy en día, ya está siendo utilizada por usuarios reales.

La Ilustración 14 muestra la pantalla de inicio de sesión de Peeplo. Es necesario que la organización que desee implementar Peeplo utilice GSuite para sus asuntos internos, dado que por este método se proveerá el acceso a la plataforma.

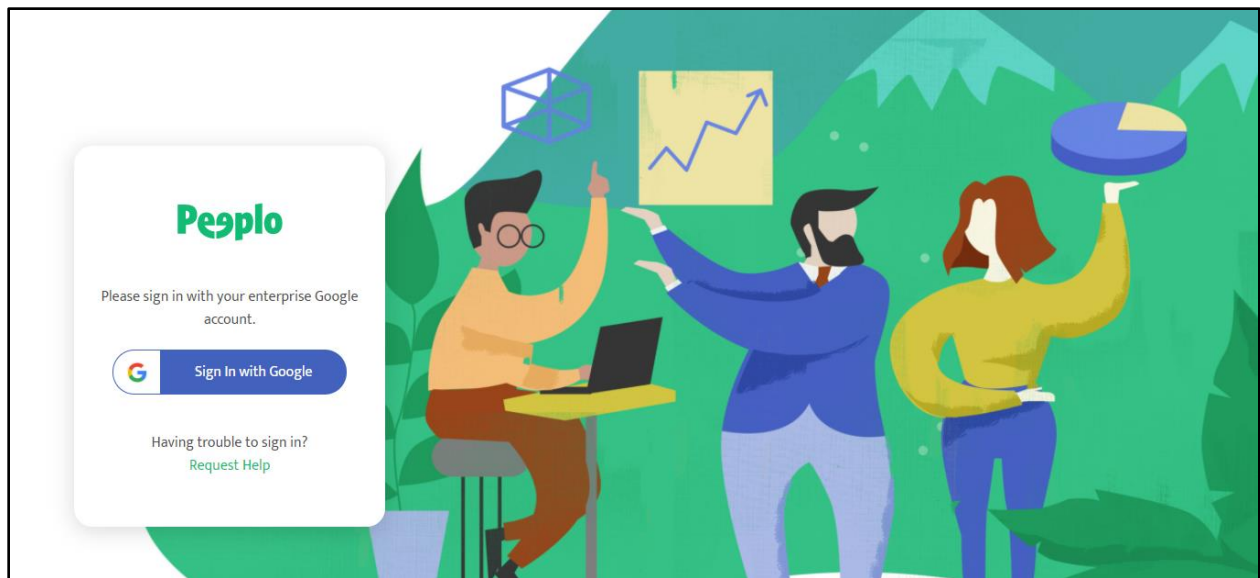


Ilustración 14 - Inicio de Sesión en Peeplo

Fuente: (Peeplo, s.f.)

La Ilustración 15 muestra la vista del perfil del empleado, que, a su vez, es la primera vista que un usuario empleado ve al entrar a la plataforma.

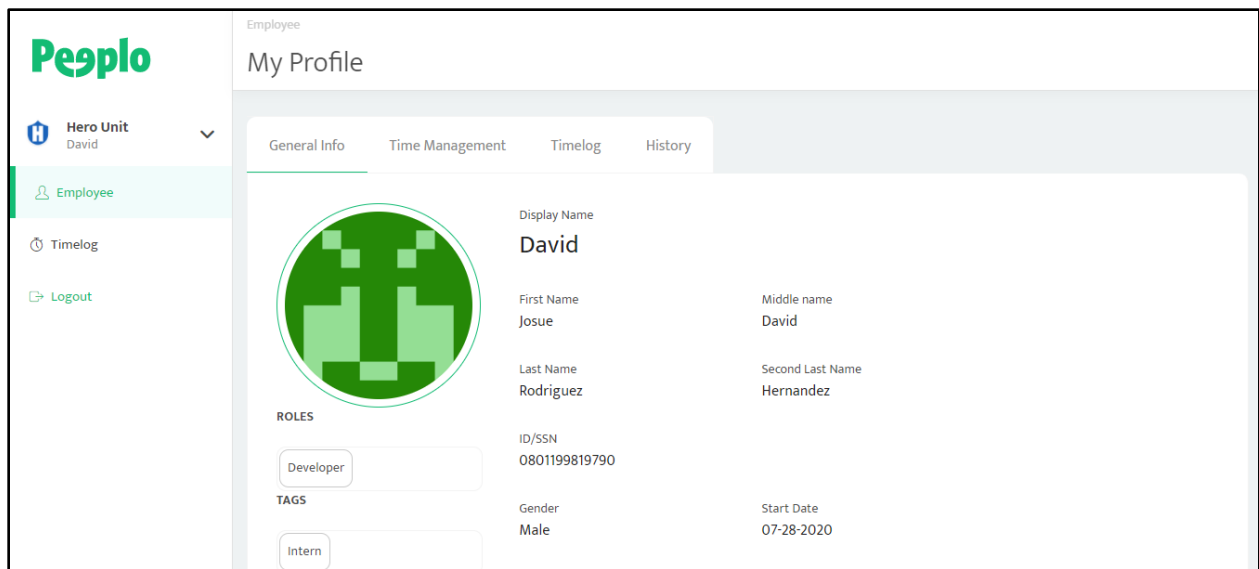


Ilustración 15 - Vista de Perfil de Empleado

Fuente: (Peeplo, s.f.)

La Ilustración 16 muestra la vista de registro de horas de un empleado.

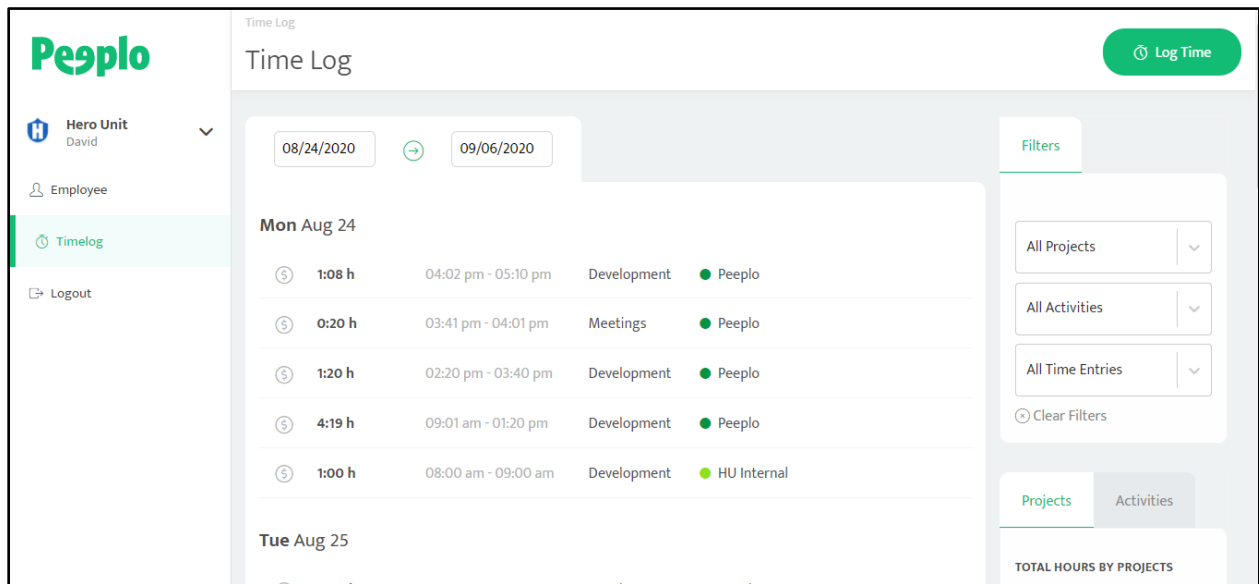


Ilustración 16 - Vista de Registro de horas en Peeplo

Fuente: (Peeplo, s.f.)

Antes de comenzar a desarrollar funcionalidades en la plataforma, se le indicó al practicante que debía desarrollar un curso para fortalecer el manejo de React Redux, dado de que esta es una funcionalidad fundamental de la plataforma. La Ilustración 17 muestra la aplicación desarrollada en este curso. Consiste en un administrador de cursos donde se guarda nombre del curso, autor y la categoría a la que pertenece.

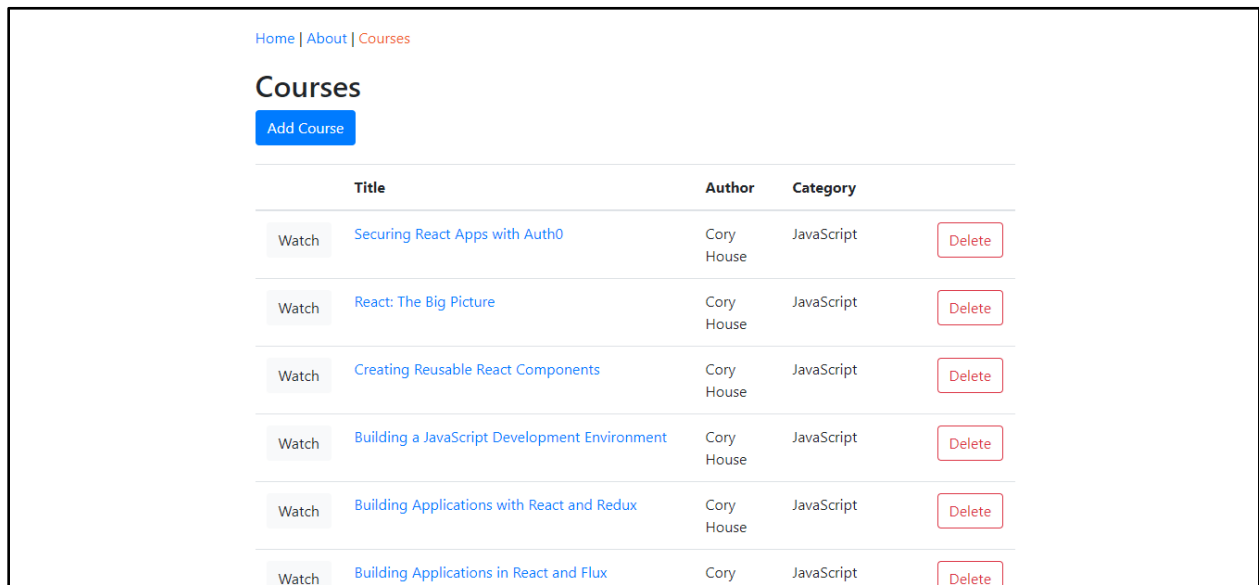


Ilustración 17 - Aplicación utilizando React con Redux

Fuente: Elaboración propia

Una vez completado el curso, se comenzó a incluir al practicante en las actividades de desarrollo de la plataforma. La Ilustración 18 muestra la agenda de una semana normal en el proyecto Peeplo.

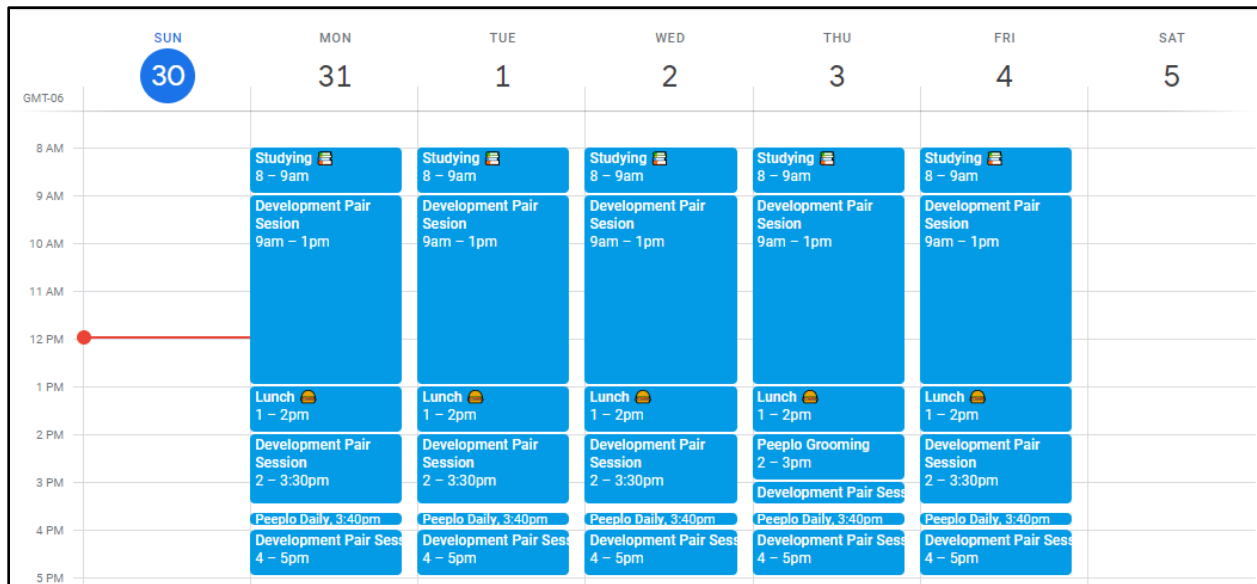


Ilustración 18 - Actividades realizadas durante una semana de trabajo

Fuente: Elaboración propia

AA/HU le da mucha importancia al desarrollo personal y profesional de cada uno de sus empleados. Es por esto por lo que la empresa otorga una hora diaria, incluidas en las horas de su jornada laboral, para que los empleados puedan aprender cosas nuevas, ya sea para apoyar el desarrollo del proyecto al que pertenecen, o bien, aprender nuevas cosas que enriquezcan su currículum.

Una jornada de trabajo de un desarrollador de software se basa fuertemente en las sesiones de desarrollo en pareja, que son sesiones donde los desarrolladores del proyecto, en este caso Peeplo (de ahora en adelante referido únicamente como proyecto), trabajan en conjunto para implementar las nuevas características propuestas por el cliente, resolver errores encontrados por los usuarios de la plataforma, y mejorar el rendimiento y experiencia de usuario de la plataforma.

La manera en que este tipo de sesiones funcionan es que uno de los desarrolladores lidere la sesión compartiendo pantalla y todos participen brindando ideas para resolver el problema en cuestión. Es muy importante la comunicación por parte de todos los desarrolladores para que la sesión sea exitosa.

Durante las reuniones Peeplo Grooming el equipo del proyecto se reunía durante una hora para discutir las tareas pendientes del proyecto y madurarlas, discutiendo las actividades que los desarrolladores, DevOps, QA y diseñadores deben llevar a cabo para culminar exitosamente la tarea. También se agregan nuevas tareas de acuerdo con las reuniones que el Scrum Master tenga con el dueño del producto.

Durante las reuniones Peeplo Daily el equipo se reunía durante quince minutos para discutir los avances realizados durante el día y los hallazgos encontrados durante el desarrollo de las tareas. En estas reuniones, los desarrolladores presentaban avances de las tareas y, en caso de terminar alguna tarea, se realizaba una presentación interna de avances, en donde el Scrum Master hacía una prueba frente a los desarrolladores y el QA de la tarea culminada. De encontrar errores o de no cumplir los requerimientos de la tarea, el Scrum Master hacía las observaciones correspondientes y devolvía la tarea a desarrollo nuevamente. De manera similar, si el QA encontraba errores o inconsistencias en alguna tarea que logró pasar la presentación interna, se reportaba durante esta reunión y se devolvía a desarrollo. Una característica importante de estas reuniones es reportar si se han tenido problemas con alguna de las tareas y de ser así, poder recibir la ayuda del equipo o buscar la ayuda en fuentes externas para poder resolver la dificultad.

4.1.3 CORRECCIÓN DE ERRORES EN LA PLATAFORMA PEEPLO

Como se mencionó anteriormente, la aplicación se encuentra en una etapa madura, y hace poco se comenzó a utilizar con usuarios reales. Es por esto por lo que, durante el mes de agosto, las tareas consistían en reparación de errores encontrados, los cuales son descritos en los siguientes párrafos.

El primer error que al practicante se le asignó resolver consistía en que, al momento de registrar una hora trabajada en donde la hora de finalización era un minuto menos que la hora actual, a esta se le restaba un minuto de manera automática. La Ilustración 19 muestra cómo se registra una hora en la plataforma.

The screenshot shows a 'Log Time' modal window. At the top right is a close button (X). The title 'Log Time' is centered. Below it is a green circular profile picture and the name 'David Rodriguez'. A large green '1:00 h' is displayed, with 'Your Total Time' underneath. The form contains three input fields for 'Date' (08/30/2020), 'Start' (9:40 PM), and 'End' (10:40 PM). Below these are two dropdown menus for 'Project' (Peeplo) and 'Type' (Development). A text area for 'Description' contains the text 'Fixed the bug that subtracted one minute from logged time.' At the bottom are two buttons: an orange 'Save' button and a white 'Cancel' button.

Ilustración 19 - Registro de Hora en Peeplo

Fuente: (Peeplo, s.f.)

Al ingresarse la hora como lo muestra la Ilustración 19, se hubiera guardado el tiempo de finalización como 10:39 PM, restándole un minuto al tiempo que el usuario ingresó. Este error se producía por la manera en que se estaban manejando los formatos de las horas en el front-end y como se estaban mandando al back-end. Este error se solucionó asegurándose que ambos, front-end y back-end, utilicen el mismo formato para las horas y así, no correr la validación para restarle un minuto.

El siguiente error que la plataforma presentaba era la creación de proyectos duplicados para los empleados. Este error fue corregido agregando una validación al momento de crear un nuevo proyecto, en donde se convierte todo el nombre del proyecto a minúsculas y se eliminan los espacios extras que contenga la cadena. Una vez hecha esta conversión, se compara con todos los proyectos previamente creados.

La Ilustración 20 ejemplifica el error en cuestión.

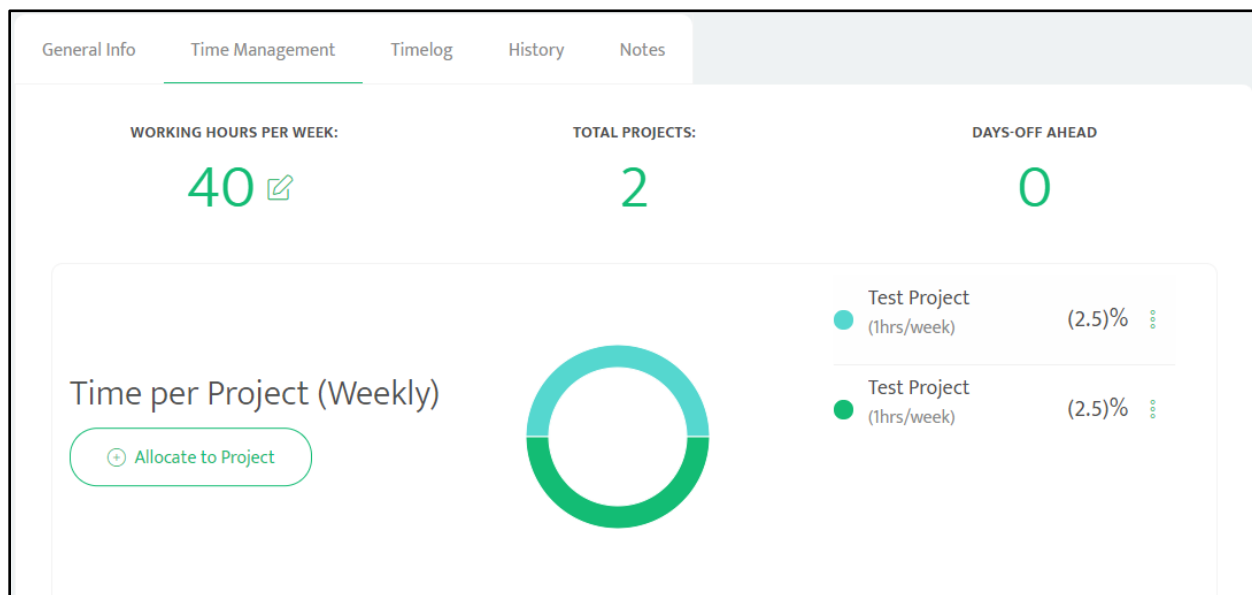


Ilustración 20 - Creación de proyectos duplicados en Peeplo

Fuente: (Peeplo, s.f.)

Otro error encontrado en la plataforma era que, al momento de editar a un empleado, sin importar el o los campos que se editaran, siempre se registraba que todos los campos de fecha habían sido modificados cuando no era así. Este error se producía por la manera en que se estaban manejando las fechas en el front-end y en el back-end. En el front-end se estaba mandando la fecha con la hora 23:59:59, que significa el final del día, y en el back-end se comparaba con las fechas almacenadas en la base de datos. El back-end detectaba que no eran igual y lo tomaba como que si hubiera sido modificada. Este comportamiento se corrigió manipulando las fechas almacenadas en la base de datos al tener la hora final del día, tomando en cuenta que lo que interesaba comparar era que las fechas fueran iguales.

La Ilustración 21 muestra este error en el historial de modificaciones del empleado.

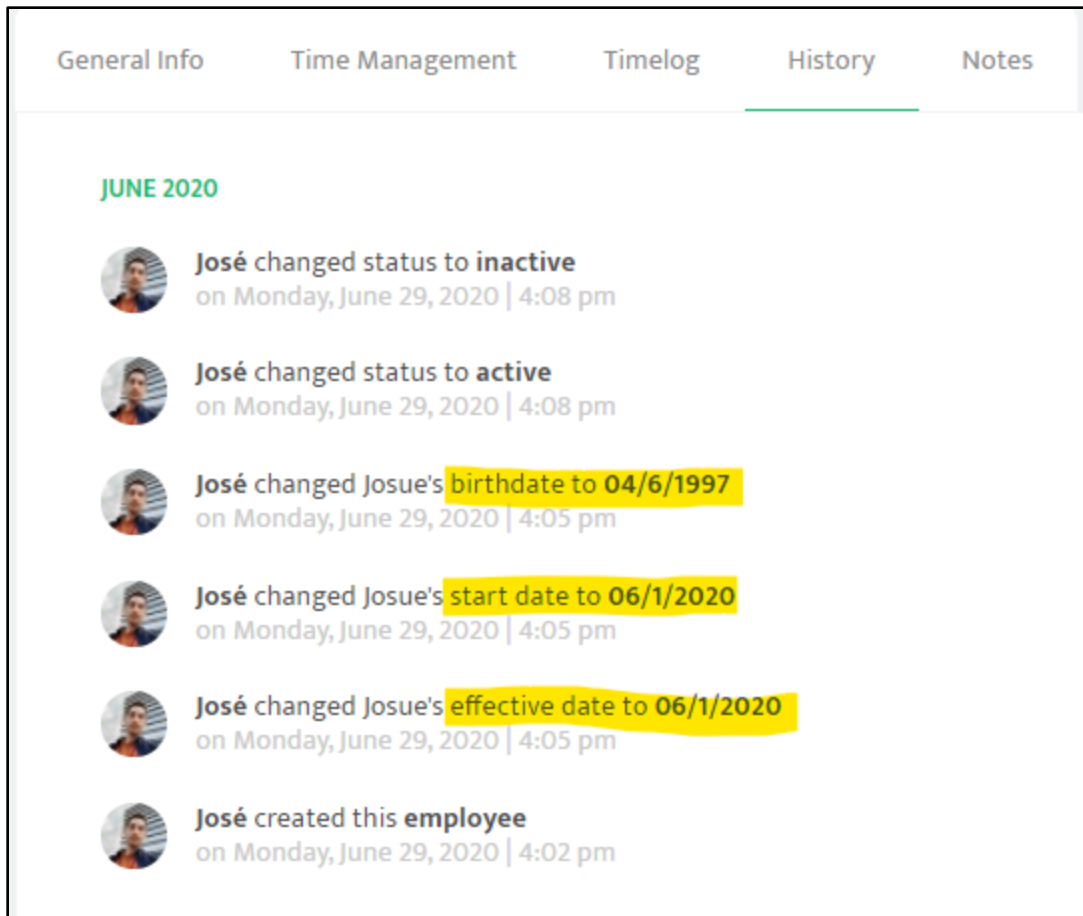


Ilustración 21 - Modificaciones de los campos de fechas

Fuente: (Peeplo, s.f.)

Otro error encontrado que el practicante resolvió consistía en que un empleado, al ser copiado de una organización a otra, podía ver los registros de tiempo de otros empleados. Este error se producía porque al momento de ejecutarse la acción de guardar al empleado copiado a la nueva organización en la base de datos, no se le estaban asignando permisos de ningún tipo.

Este error se corrigió mandando a llamar a la función ya existente para asignar los permisos a los empleados al momento de copiar un empleado a otra organización.

La Ilustración 22 ejemplifica este error desde la cuenta del usuario Rony Vidaur, quien fue copiado de la organización Acklen Avenue a la organización Hero Unit. Nótese en la esquina inferior derecha la cantidad enorme de horas trabajadas producidas por este error.

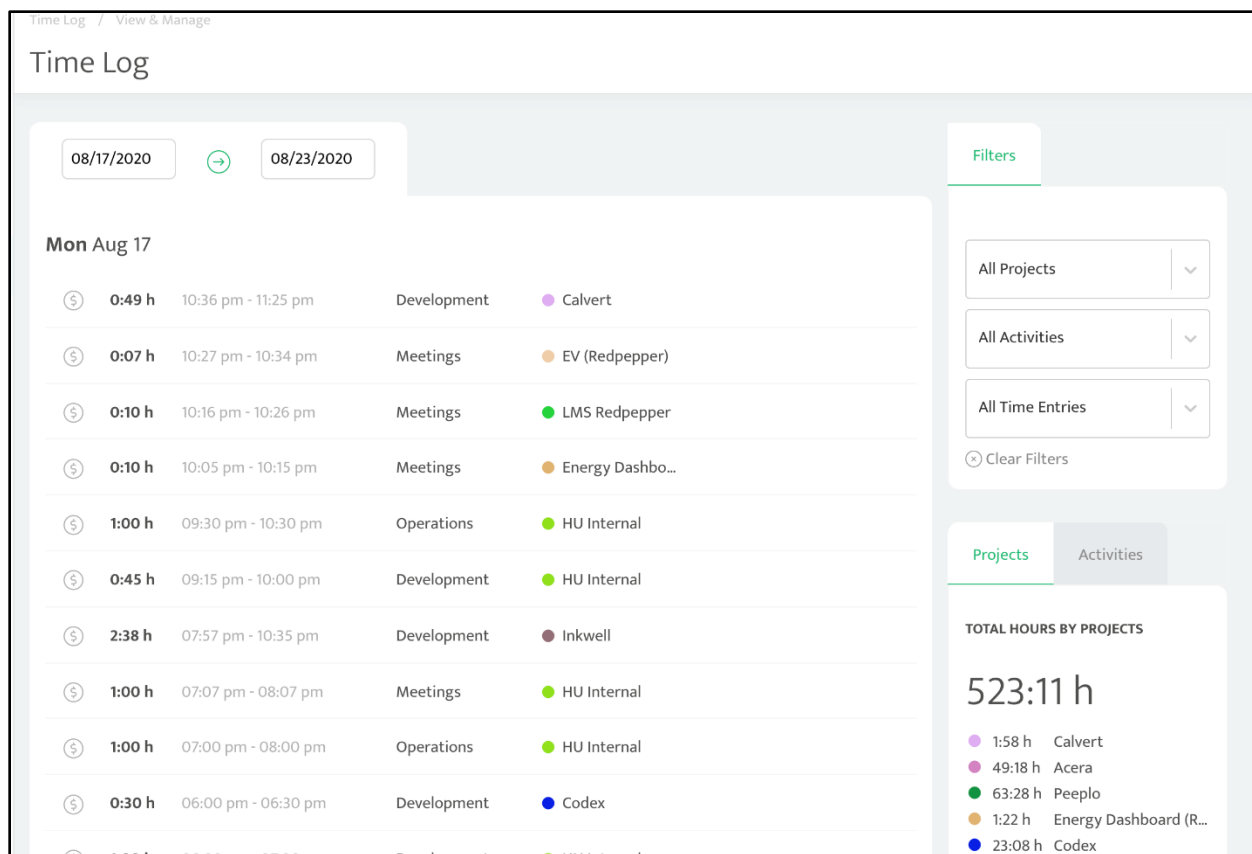


Ilustración 22 - Error de permisos de empleado

Fuente: (Peeplo, s.f.)

Antes, en la plataforma, no existía una validación para evitar que el usuario enviara muchas solicitudes para guardar un registro de hora con los mismos datos. La Ilustración 23 ejemplifica este caso, en donde se guarda en la base de datos el primer registro y luego le indica al usuario que existe una superposición en los tiempos.

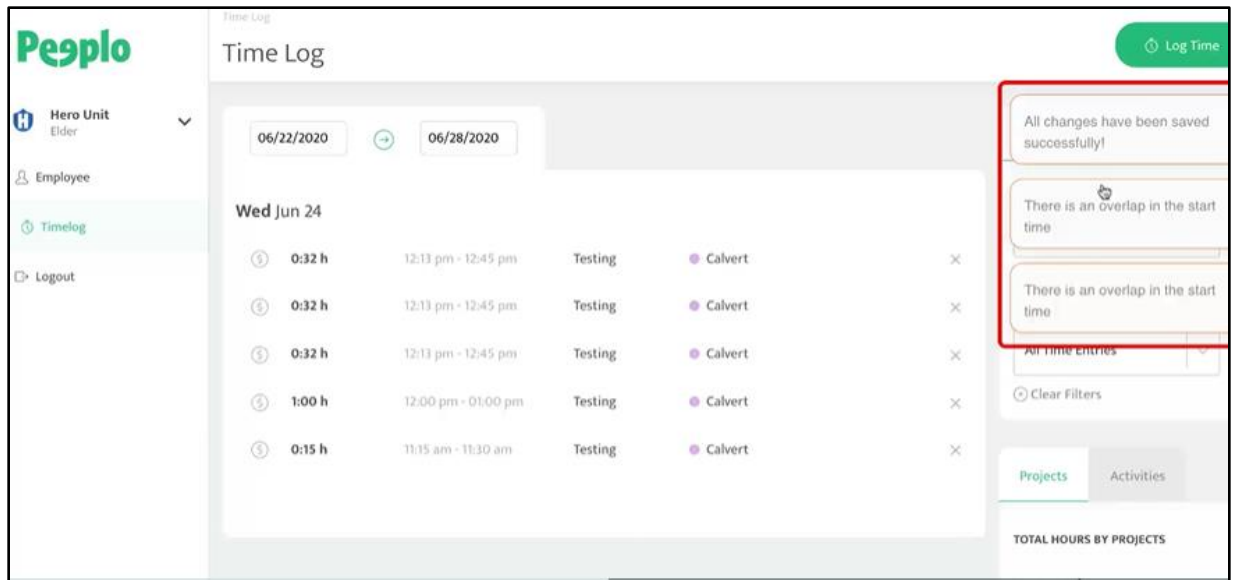


Ilustración 23 - Envío de muchas solicitudes para guardar un registro de tiempo

Fuente: (Peeplo, s.f.)

Este error fue solucionado deshabilitando el botón de guardado de un registro de tiempo luego de hacer una solicitud y, de igual manera, agregando validaciones en el back-end. La Ilustración 24 muestra el botón deshabilitado.

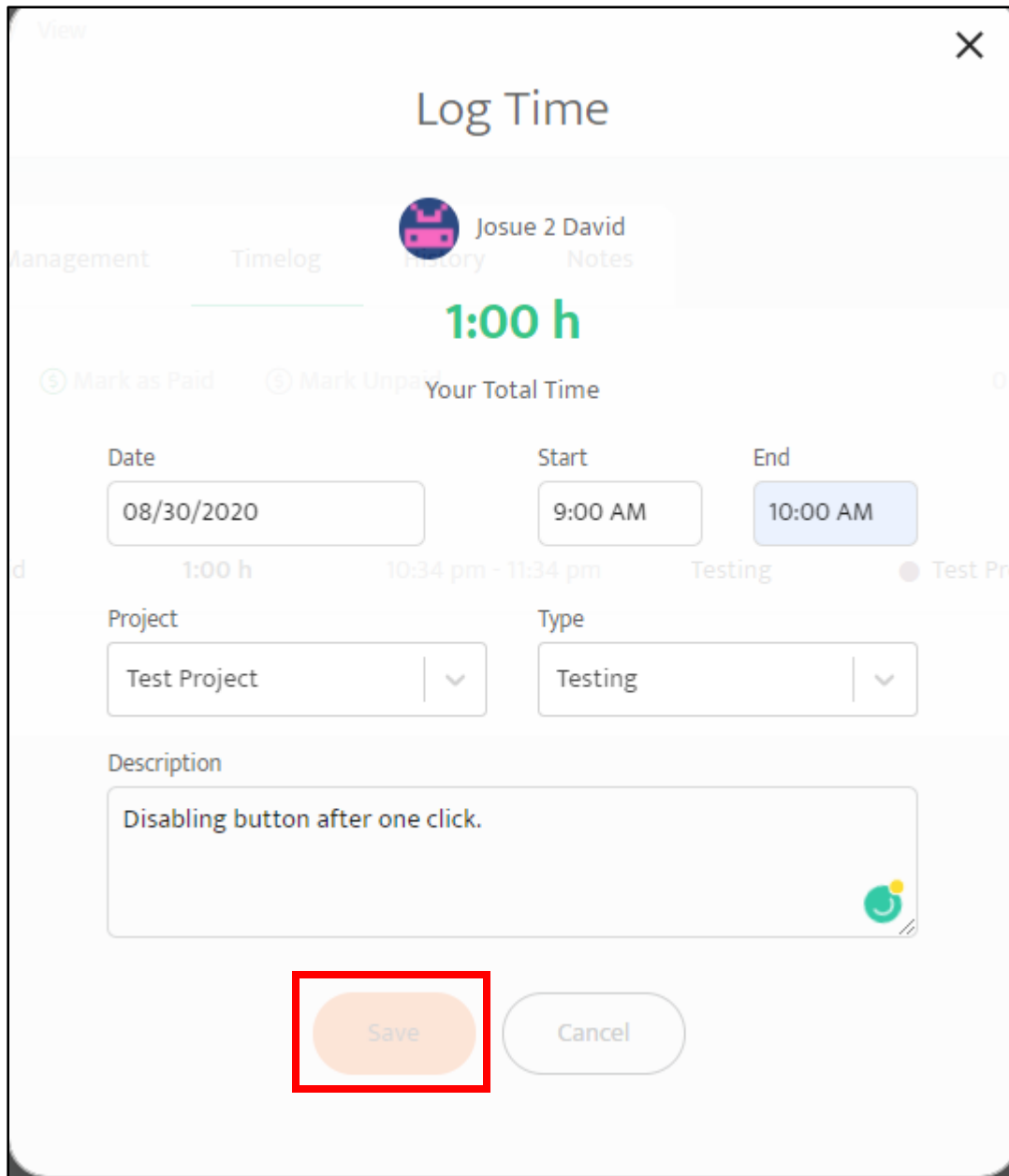


Ilustración 24 - Botón deshabilitado

Fuente: (Peeplo, s.f.)

El siguiente error que se le asignó resolver al practicante consistía en corregir un comportamiento que no le permitía al usuario registrar horas cuya hora de finalización era la hora actual. La Ilustración 25 ejemplifica este error. Este error se producía por la manera en que manejaban las horas. Al agarrar la hora del campo "End" a este se le redondeaba al final del minuto (segundo 59) y se comparaba con la hora actual de la computadora, que podría estar en cualquier segundo

entre 0 y 59. Al hacer la comparación, la plataforma mostraba un error diciendo que no se pueden registrar horas adelantadas.

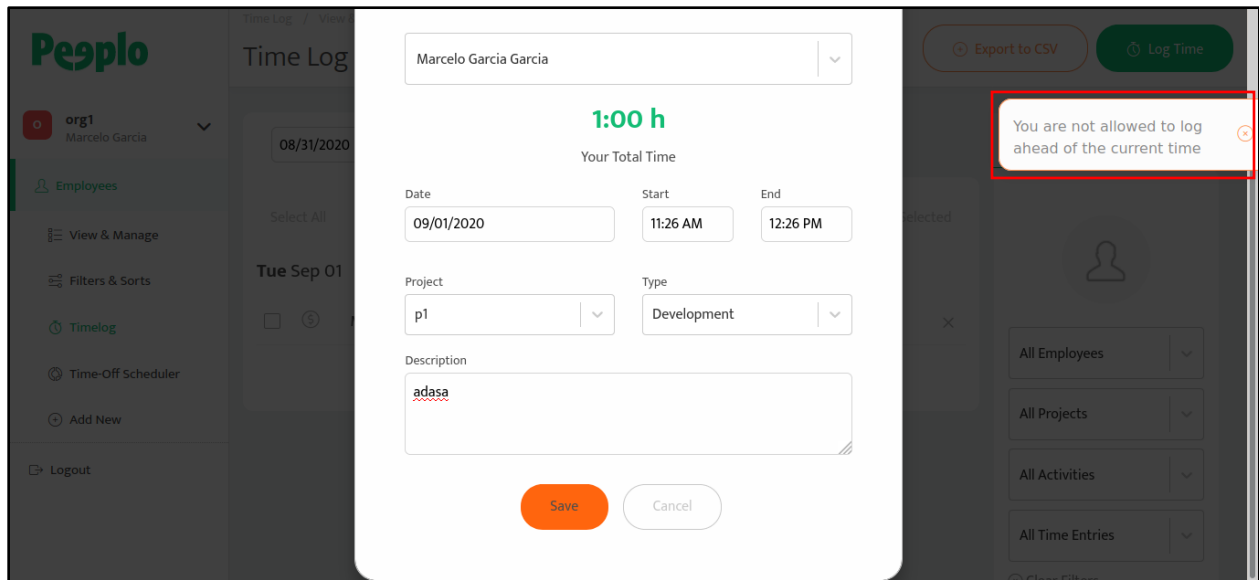


Ilustración 25 - Registro con hora de finalización igual a la hora actual

Fuente: (Peeplo, s.f.)

Este error se corrigió haciendo que ambas horas, tanto la del campo "End" como la que se recupera de la computadora, sigan el mismo formato, haciendo que ambas se redondearán al final del minuto (segundo 59).

Otro error que estaba presente en la plataforma era que el usuario podía registrar horas cuya hora de inicio era la misma que la hora de finalización y, como consecuencia, la duración de la actividad era igual a 0 horas con 0 minutos. La Ilustración 26, dentro del cuadro rojo, muestra dos registros con este error.

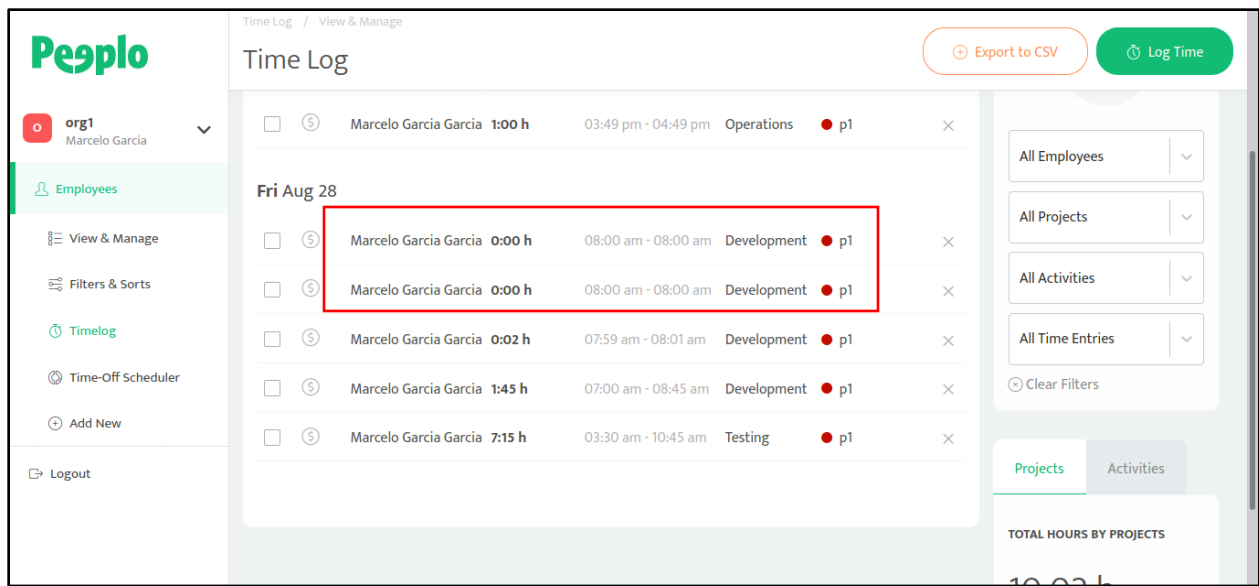


Ilustración 26 - Actividades con duración de 0 minutos

Fuente: (Peeplo, s.f.)

Este comportamiento fue corregido mediante una validación en el back-end evaluando la duración de la tarea. Si tenía una duración menor de 60 segundos, no permitía la creación del registro de hora y el back-end devolvía un error.

El siguiente error que el practicante solucionó se producía al momento de exportar la lista de empleados a formato de valores separados por coma (CSV). En la columna "isActive" se mostraba un valor booleano en lugar del estado actual del empleado (activo o inactivo). La Ilustración 27 muestra este comportamiento.

	A	B	C	D	E	F	G
1	displayName	lastName	roles	salary	city	email	isActive
2	John Doe	Doe		0		josue.rodriguez1551@gmail.com	TRUE
3	Josue David Rodriguez	Rodriguez		0		david.rodriguez@herounit.io	FALSE
4							

Ilustración 27 - Valor booleano en la columna isActive

Fuente: (Peeplo, s.f.)

Para corregir este comportamiento, se colocó una validación antes de crear el archivo CSV evaluando el estado del empleado y devolviendo una cadena con un texto más descriptivo para el estado. La Ilustración 28 muestra el comportamiento corregido.

	A	B	C	D	E	F	G
1	displayName	lastName	roles	salary	city	email	isActive
2	John Doe	Doe		0		josue.rodriguez1551@gmail.com	Active
3	Josue David Rodriguez	Rodriguez		0		david.rodriguez@herounit.io	Inactive
4							

Ilustración 28 - Comportamiento del estado corregido

Fuente: (Peeplo, s.f.)

El siguiente error reportado y que el practicante solucionó consistía en que, al momento de filtrar los registros de tiempo por empleados, solo se mostraba el primer nombre de cada empleado en el filtro, como lo muestra la Ilustración 29. Esto causaba confusión en el administrador de la organización debido a que no se sabía con exactitud a qué empleado estaba viendo.

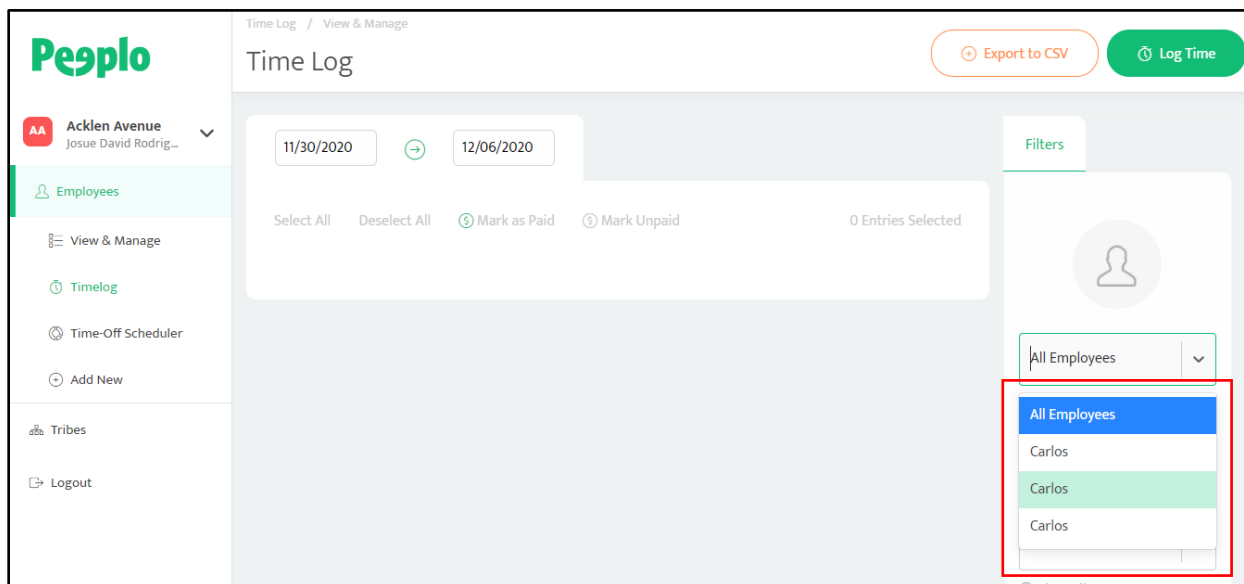


Ilustración 29 - Filtro de empleados mostrando primer nombre

Fuente: (Peeplo, n.d.)

Para corregir este comportamiento, fue necesario que se recuperara la información del apellido del empleado de la tabla "User" de la base de datos, y finalmente, mostrar la información recuperada en la vista correspondiente. La Ilustración 30 muestra el comportamiento corregido.

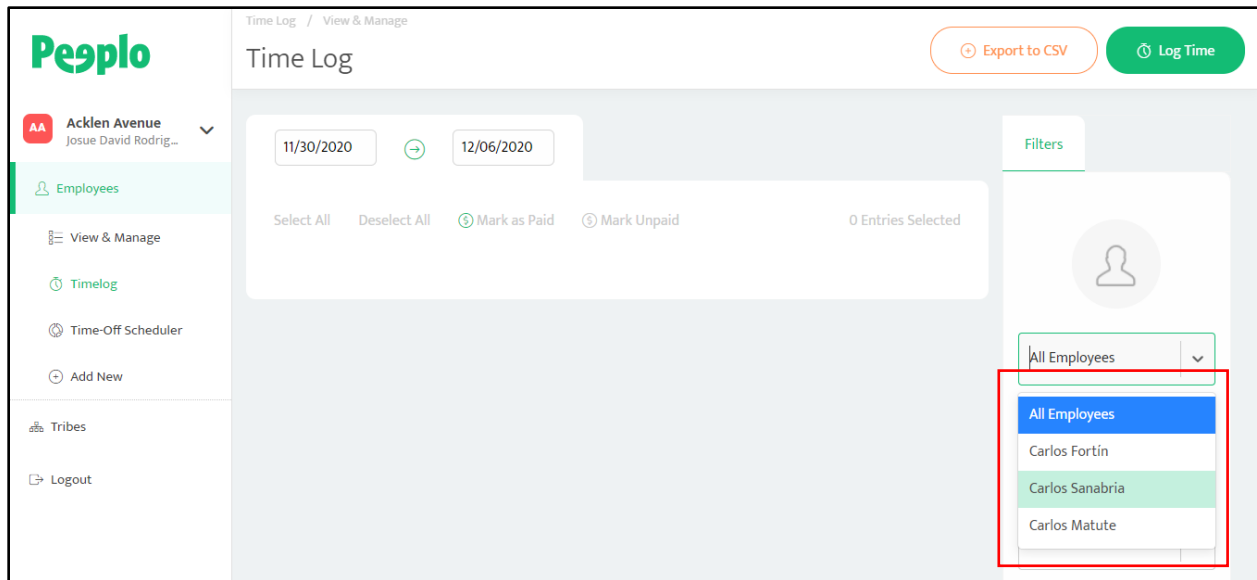


Ilustración 30 - Filtro de empleados mostrando nombre completo

Fuente: (Peeplo, n.d.)

Al momento de crear un empleado nuevo en una organización, uno de los campos que se le solicita llenar al administrador es la "información de pago" del empleado.

Previamente, el límite de caracteres para este campo era de 200 caracteres, pero no se manejaba de manera correcta el error que devolvía la base de datos cuando se excedía el límite, la plataforma simplemente quedaba congelada hasta que el usuario refrescara la página. De igual manera, no había una manera visual de indicarle al usuario que existe un límite de caracteres.

Se le solicitó al practicante incrementar el límite de 200 a 500 caracteres, implementar todas las validaciones necesarias para hacer que la plataforma sea más robusta y también desarrollar una alerta visual que le haga saber al usuario sobre el límite de caracteres.

Para completar lo solicitado, el practicante modificó la columna de la información de pago de la tabla "Employee" en la base de datos, aumentando el límite de caracteres permitido por ésta. Seguidamente, se implementaron validaciones en el back-end para asegurar que la base de datos reciba la información con los formatos correctos. Finalmente, el practicante implementó un contador de caracteres como alerta visual para el usuario. La Ilustración 31 muestra el contador. Una vez se llega al límite, el formulario no permite que se siga escribiendo en el campo de información de pago.

Employees / View & Manage / Add New

< Add a New Employee Cancel Add Employee

Birthday
11/30/2020

PAYMENT INFO

200/500

ADDRESS
Address line

Ilustración 31 - Contador de caracteres en información de pago

Fuente: (Peeplo, n.d.)

El siguiente error que el practicante solucionó en la plataforma se producía en la vista de administración de empleados. Cuando se deseaba ordenar a los empleados por su fecha de inicio, fecha de terminación o fecha de cumpleaños, el ordenamiento de las filas de la tabla se hacía de manera errónea. La Ilustración 32 muestra el comportamiento erróneo de la tabla al ser ordenada por la columna "Start Date" de manera descendente.

Employees

View & Manage Export to CSV Display Columns Add New Employee

Display Na...	Last Name	Second La...	Start Date	Terminatio...	Salary	Email	Birthday	City	Status
<input type="text" value="Enter nam"/>	<input type="text" value="Enter nam"/>	<input type="text" value="Enter nam"/>	<input type="text" value="Select a da"/>	<input type="text" value="Select a da"/>	<input type="text" value="Mir to Ma"/>	<input type="text" value="Enter emai"/>	<input type="text" value="Select a da"/>	<input type="text" value="All ci."/>	<input type="text" value="Active"/>
	Nicolas	Account	11/19/2019	12/09/2017	0	test.acc.jdr...	11/19/2020		Active
	David	Rodríguez	11/18/2018	10/05/2018	0	josue.rodri...	01/31/1999		Active
	Josue David...	Rodríguez	11/17/2020	08/23/2019	0	david.rodri...	04/09/1998		Active

Ilustración 32 - Ordenamiento erróneo de la tabla de empleados

Fuente: (Peeplo, n.d.)

Para corregir este comportamiento, el practicante implementó un algoritmo de ordenamiento de fechas y se sobrescribió el algoritmo de ordenamiento por defecto de la tabla. La Ilustración 33 muestra el comportamiento corregido. El algoritmo fue implementado para las columnas "Start Date", "Termination Date" y "Birthday".

The screenshot shows a web interface for managing employees. At the top, there are buttons for 'Export to CSV', 'Display Columns', and 'Add New Employee'. Below these is a table with the following columns: Display Name, Last Name, Second Last Name, Start Date, Termination Date, Salary, Email, Birthday, City, and Status. The 'Start Date' column is highlighted with a red box. The data in the table is as follows:

Display Name	Last Name	Second Last Name	Start Date	Termination Date	Salary	Email	Birthday	City	Status
Enter name	Enter name	Enter name	Select a date	Select a date	Min to Max	Enter email	Select a date	All cities	Active
Josue David...	Rodriguez		11/17/2020	08/23/2019	0	david.rodri...	04/09/1998		Active
Nicolas	Account		11/19/2019	12/09/2017	0	test.acc.jdr...	11/19/2020		Active
David	Rodriguez		11/18/2018	10/05/2018	0	josue.rodri...	01/31/1999		Active

Ilustración 33 - Ordenamiento correcto de la tabla de empleados

Fuente: (Peeplo, n.d.)

El siguiente error que el practicante resolvió hacía que la plataforma dejara de responder debido a que intentaba acceder a información que no había sido recuperada aún del back-end. Este error se producía cuando un administrador entraba al perfil de uno de los empleados de la organización y era causado, en la mayoría de los casos, por una conexión lenta o inestable de internet. La Ilustración 34 muestra la plataforma caída y el error en consola.

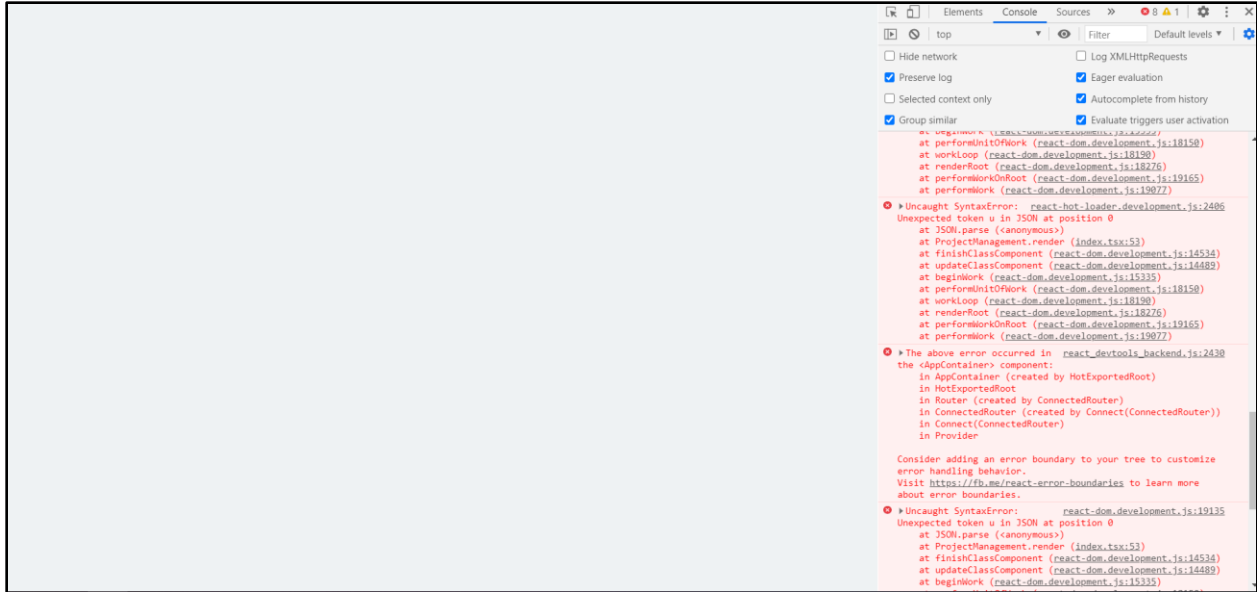


Ilustración 34 - Plataforma caída al ver la información de un empleado

Fuente: (Peeplo, n.d.)

Para corregir este comportamiento, se le indicó al practicante que implementara una vista de cargado mientras se hacían las solicitudes al back-end y, una vez éste haya respondido, mostrar la información correspondiente. La Ilustración 35 muestra la pantalla de cargado que fue implementada.

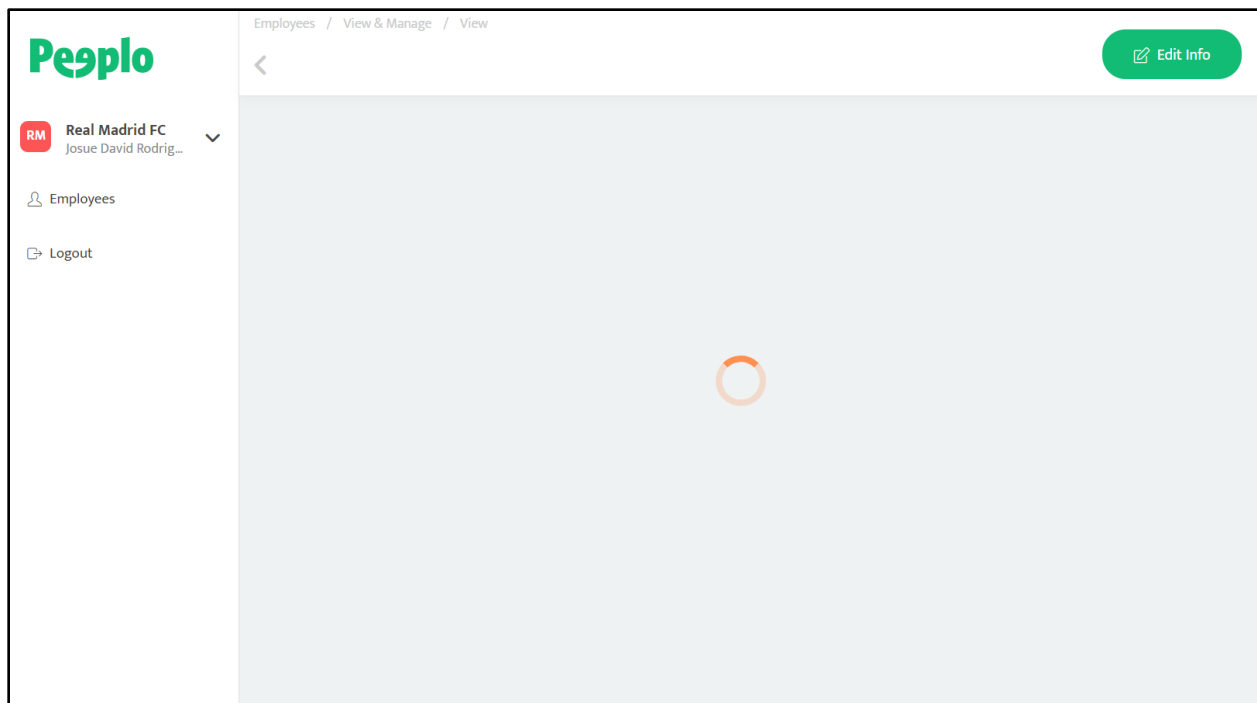


Ilustración 35 - Vista de cargado

Fuente: (Peeplo, n.d.)

4.1.4 IMPLEMENTACIÓN DE NUEVAS CARACTERÍSTICAS EN LA PLATAFORMA PEEPLO

La primera nueva característica que el practicante implementó en la plataforma fue deshabilitar los campos para editar un registro de hora y quitar el botón para archivar un registro luego de que hayan pasado 10 minutos de la creación del registro. La plataforma cuenta con una política ya implementada que pasados 10 minutos no se pueden editar ni archivar registros, pero no había una manera de representar esta política de manera visual para el usuario.

Algo en lo que el cliente hizo mucho hincapié fue en que esta implementación debía ser a prueba de manipulaciones del usuario. Es decir, si el usuario cambia la hora de su computadora para poder archivar o editar un registro pasados los 10 minutos, la plataforma no debe permitirlo. Este requerimiento se logró utilizando únicamente el servidor donde está alojado el back-end para hacer las validaciones de tiempo, dado que es más difícil, casi imposible, que la hora del servidor sea modificada por un usuario.

Luego de realizar las validaciones, se mandaba la carga útil de los registros desde el back-end al front-end con una bandera que indicaba si un registro en específico era editable o no.

La Ilustración 36 muestra un comparativo entre un registro de hora que aún es editable (izquierda) contra un registro de hora que no es editable (derecha). Nótese que en el registro de la derecha los campos toman un color gris oscuro, al igual que el botón de guardado, indicando que los campos no son editables y que el botón está deshabilitado.

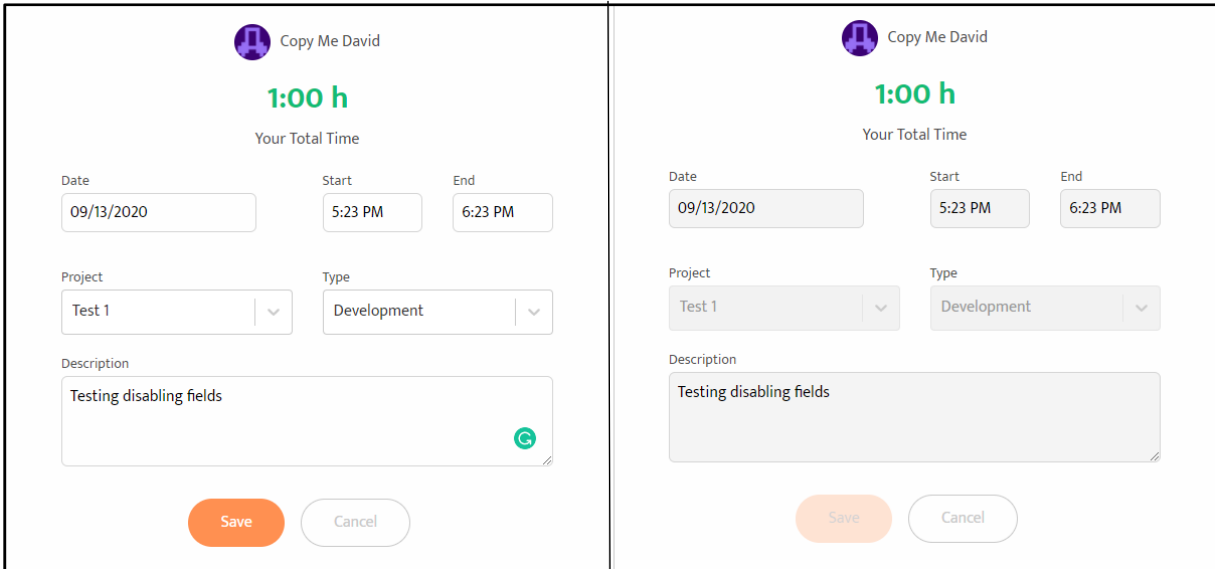


Ilustración 36 - Registro editable contra registro no editable

Fuente: (Peeplo, s.f.)

La Ilustración 37 muestra el requerimiento de no permitir que el usuario archive una hora pasados 10 minutos de su registro. La hora registrada para el día 12 de septiembre cuenta con el botón dado que, para el momento de la prueba, no habían pasado 10 minutos. Sin embargo, los registros del domingo 13 de septiembre carecen de este botón porque ya no se pueden archivar.

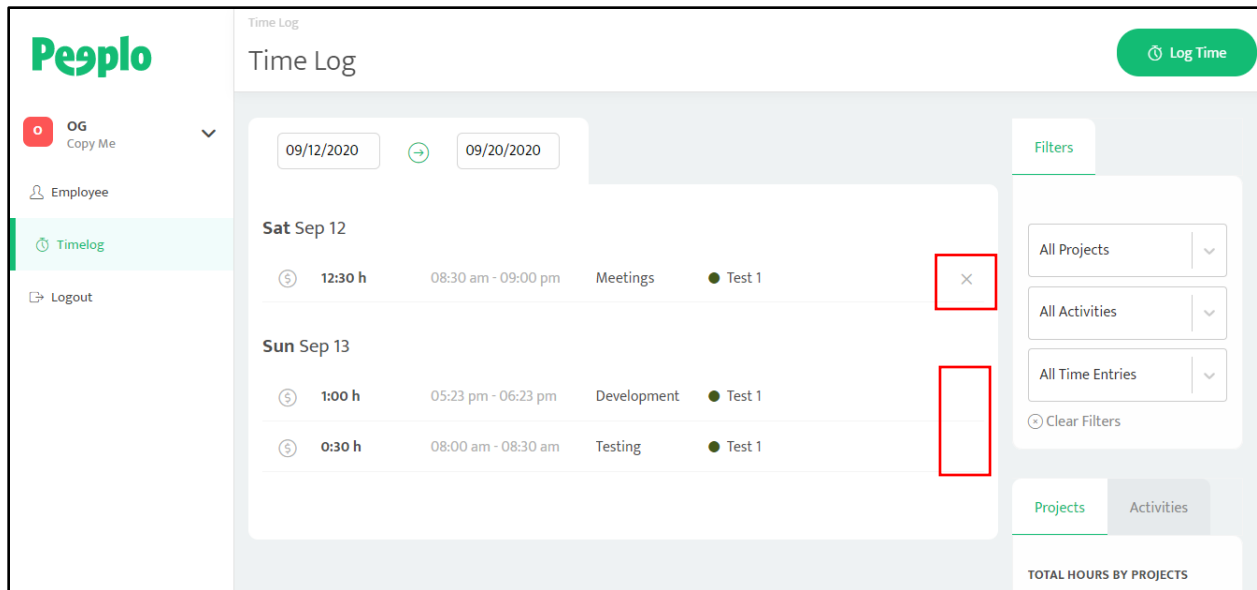


Ilustración 37 - Registros que se pueden archivar y registros que no se pueden archivar

Fuente: (Peeplo, s.f.)

La siguiente nueva característica que el practicante implementó en la plataforma fue la de darle a un administrador la capacidad de archivar o desarchivar una de sus organizaciones.

Para desarrollar esta característica, se creó en el back-end una sentencia SQL para actualizar el estado de una organización en la base de datos. También se creó un endpoint para permitir la comunicación con el front-end del proyecto y devolver el resultado de la acción.

En el front-end del proyecto se implementaron nuevos componentes visuales para el proceso de archivar y desarchivar una organización. Primero, se implementó un nuevo botón en el menú de acciones de la vista de administración de organizaciones como lo muestra la Ilustración 38.

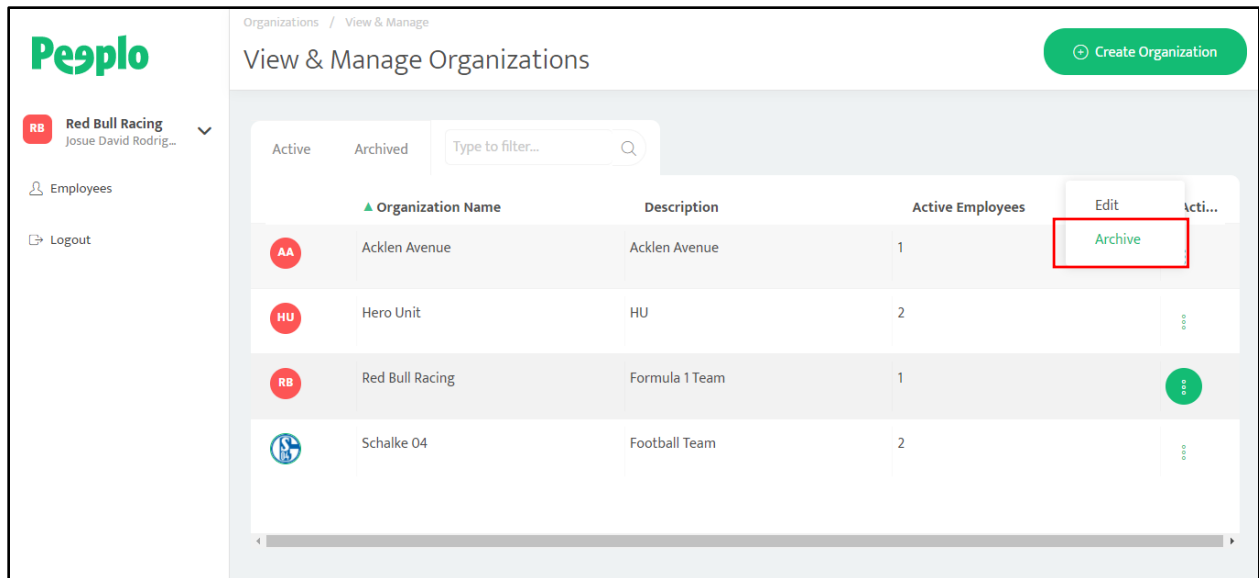


Ilustración 38 - Botón para archivar una organización

Fuente: (Peeplo, s.f.)

Al presionar el botón con la etiqueta "Archive", que en español se traduce a Archivar, se muestra una ventana emergente para que el administrador confirme si desea archivar la organización seleccionada. La Ilustración 39 muestra la ventana con el nombre de la organización y su logo. Esta ventana fue creada por el practicante.

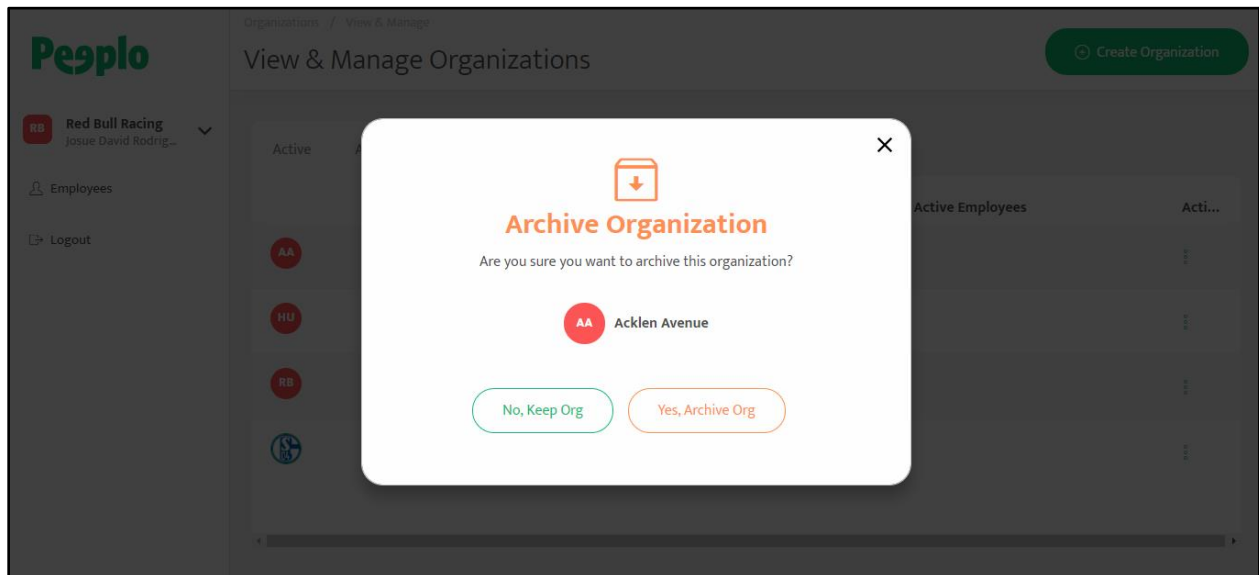


Ilustración 39 - Ventana de confirmación para archivar una organización

Fuente: (Peeplo, s.f.)

Al presionar el botón de confirmación, se hace la solicitud al back-end y posteriormente, se muestra una notificación indicando la culminación exitosa de la acción. La Ilustración 40 muestra la notificación mencionada. Nótese que la organización Acklen Avenue ya no aparece en la pestaña de organizaciones activas.

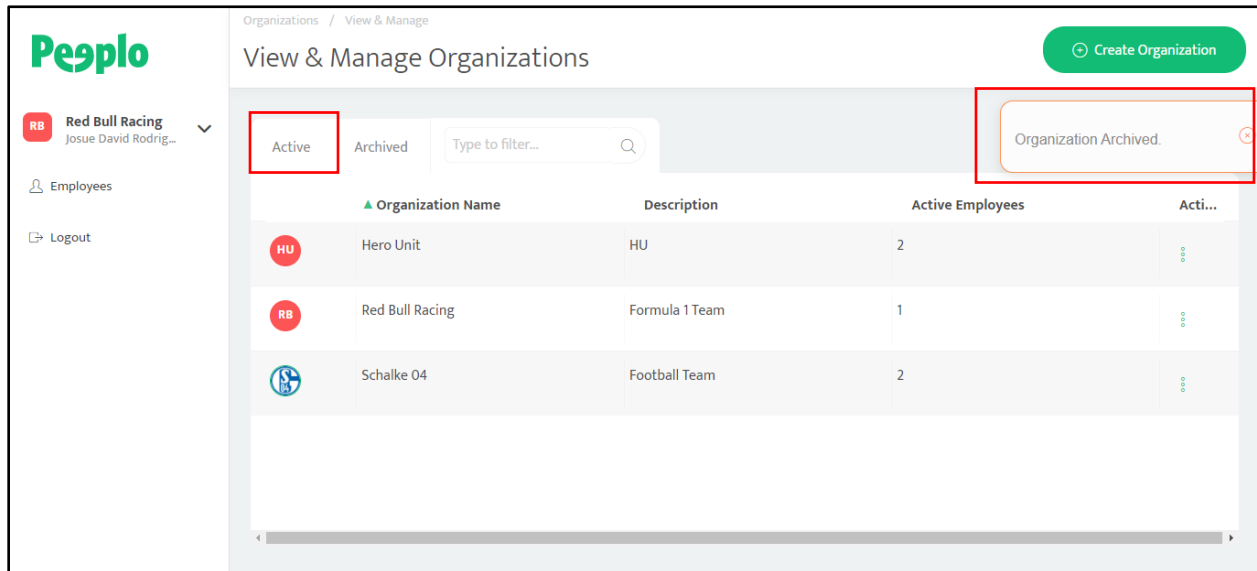


Ilustración 40 - Resultado final al archivar una organización

Fuente: (Peeplo, s.f.)

Al cambiar a la pestaña "Archived", que en español se traduce a Archivados, el administrador puede ver todas las organizaciones que ha archivado. En esta pestaña, se implementó un botón en el menú contextual para desarchivar una organización. A diferencia del proceso de archivado, desarchivar una organización no necesita confirmación.

La Ilustración 41 muestra la pestaña de organizaciones archivadas y el menú contextual con el botón con etiqueta "Unarchive", que en español se traduce a Desarchivar.

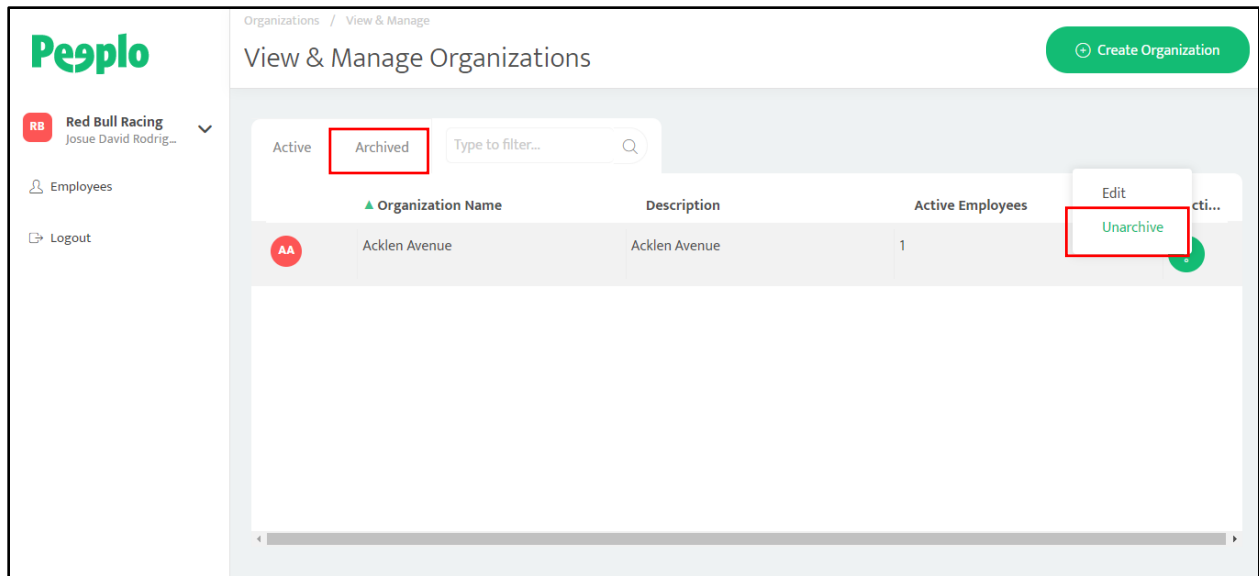


Ilustración 41 - Desarchivar una organización

Fuente: (Peeplo, s.f.)

Al presionar el botón previamente mencionado, se hace la solicitud al back-end y se muestra una notificación indicando que la organización fue desarchivada exitosamente. La Ilustración 42 muestra la notificación.

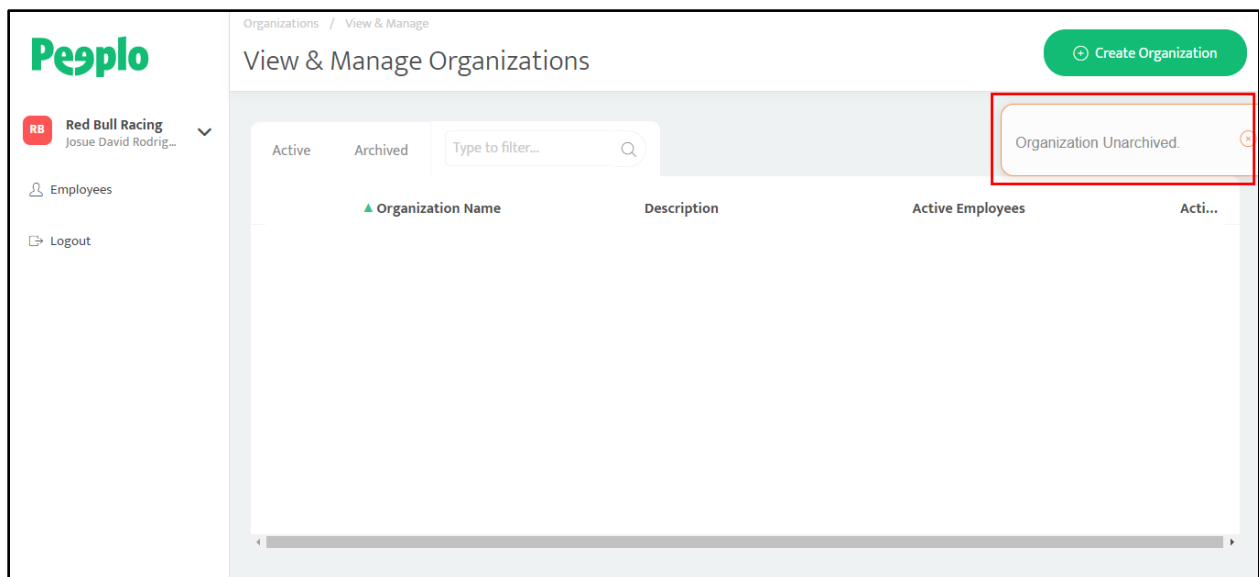


Ilustración 42 - Resultado final al desarchivar una organización

Fuente: (Peeplo, s.f.)

Otra nueva característica que se le asignó al practicante fue la de implementar los permisos de "Manager" o Gerente en una organización. La plataforma Peeplo cuenta con cuatro tipos de permisos: Dueño de la Organización (Owner), Administrador (Administrator), Gerente (Manager) y Empleado (Employee). La Ilustración 43 muestra cómo asignar permisos de Gerente a un empleado de una organización. Para asignar los permisos, primero se debe seleccionar el empleado deseado y luego, el tipo de permiso que se desea asignar.

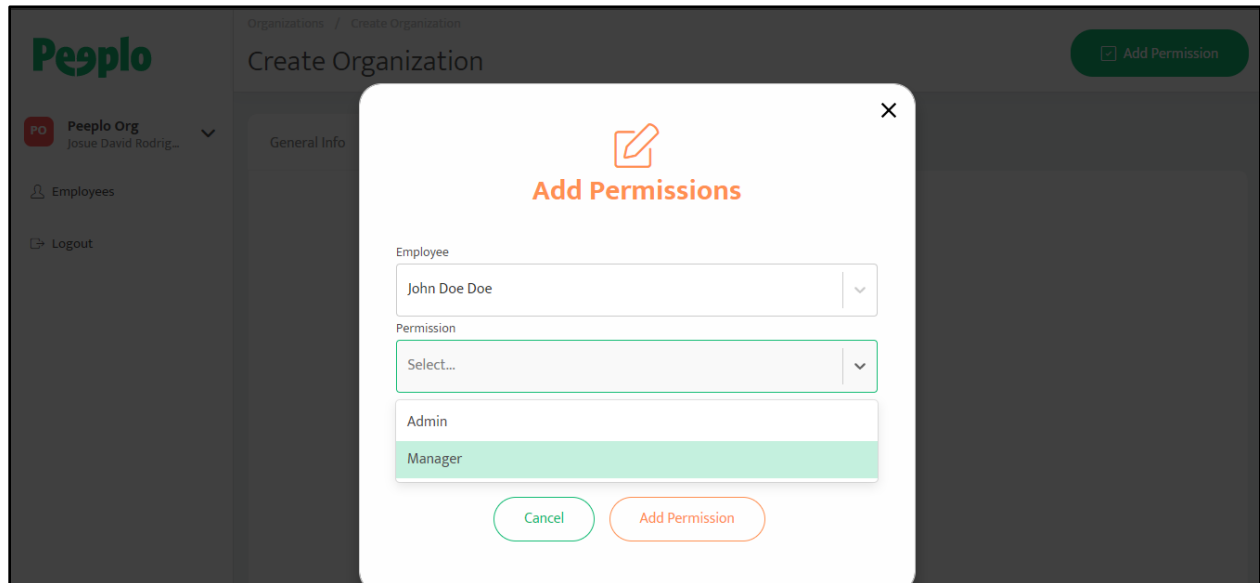


Ilustración 43 - Asignación de permisos en una organización

Fuente: (Peeplo, s.f.)

El alcance de un gerente es similar al alcance de un administrador de organización. Es decir, el gerente tiene acceso a las mismas vistas que un administrador puede ver, con la diferencia que un gerente no puede marcar como pagadas las horas trabajadas de los empleados. La Ilustración 44 muestra la vista de horas trabajadas con permisos de administrador.

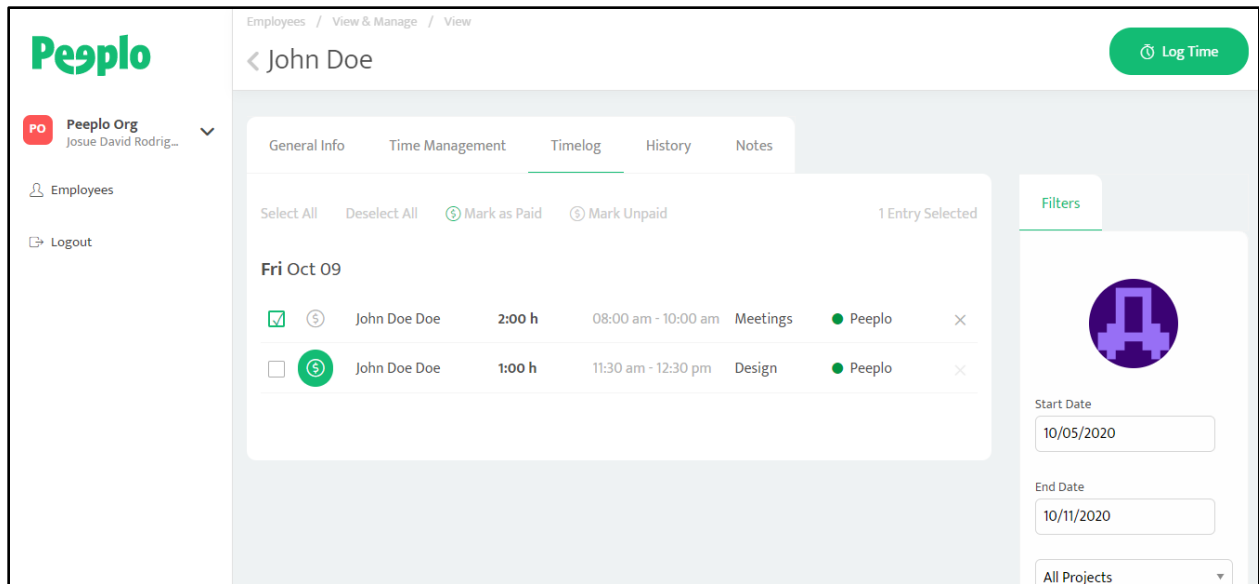


Ilustración 44 - Vista de horas trabajadas con permisos de Administrador

Fuente: (Peeplo, s.f.)

La Ilustración 45 muestra la vista con permisos de gerente. Nótese que en la parte izquierda de la tabla no aparecen los botones para seleccionar la fila y para pagar la hora que aparecen en el rol de Administrador. De igual manera, no aparece la barra de acciones ubicada en la parte superior de la tabla.

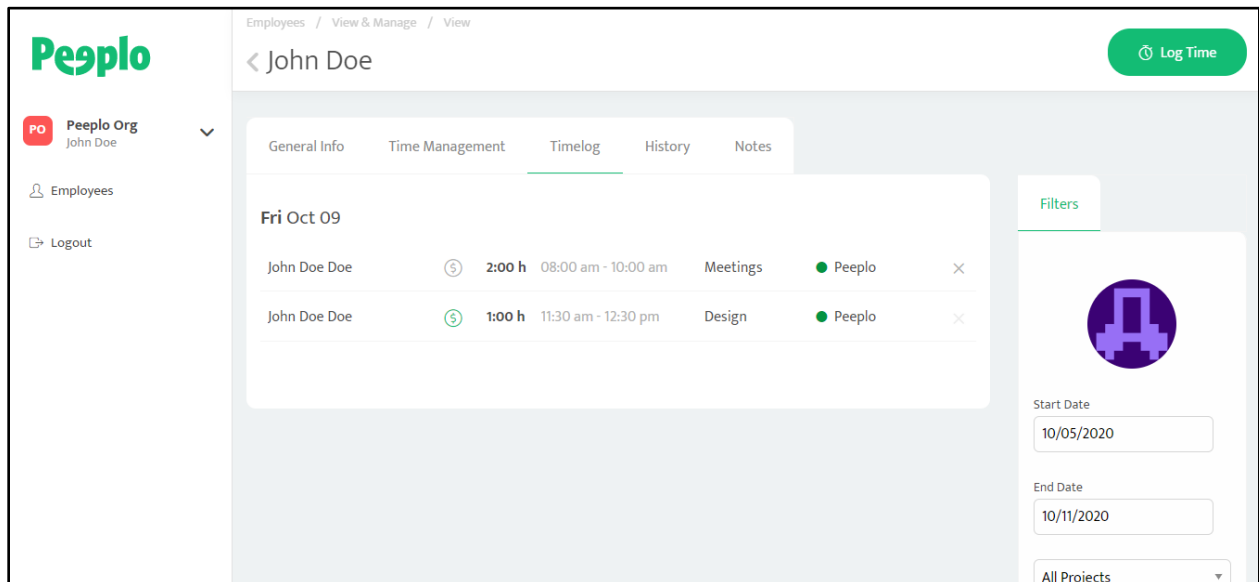


Ilustración 45 - Vista de horas trabajadas con permisos de Gerente

Fuente: (Peeplo, s.f.)

La siguiente nueva característica que el practicante desarrolló fue la de no permitir nombres duplicados al momento de crear una organización. Se solicitó que esta característica debía ser insensible a mayúsculas y minúsculas.

Para desarrollar esta característica, se creó una validación en el back-end para verificar si el nombre ingresado ya existe en la base de datos. De existir, la validación retorna un error al front-end. De no existir, se crea la organización. La Ilustración 46 muestra el error que retorna al crear una organización con nombre repetido. En la ilustración, se intentó crear una organización con un nombre ya existente, cambiando únicamente algunas mayúsculas y minúsculas en el nombre.

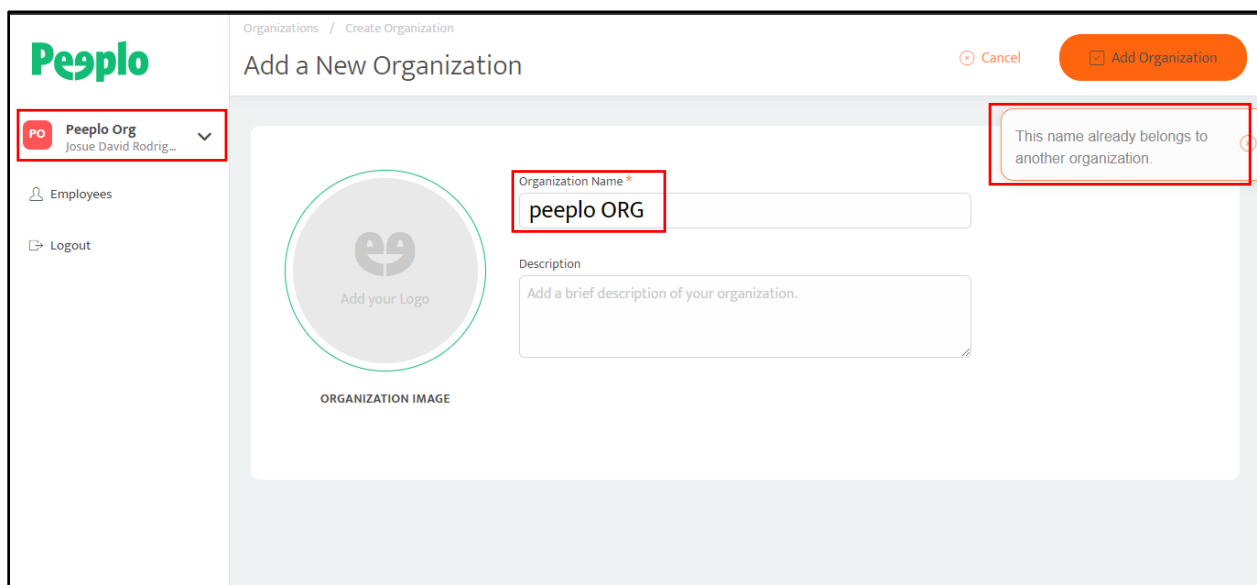


Ilustración 46 - Error al crear organizaciones con nombres repetidos

Fuente: (Peeplo, s.f.)

La siguiente característica que el practicante implementó fue la modificación de la estructura de la tabla "Employee" y la tabla "User" en la base de datos. Se le solicitó al practicante migrar los campos "firstName" y "lastName" de la tabla "Employee" a la tabla "User". La Ilustración 47 muestra del lado izquierdo la estructura original de la tabla "Employee" en la base de datos y del lado derecho la modificación realizada.

Employee		Employee	
PK	<u>id</u>	PK	<u>id</u>
	firstName		middleName
	middleName		secondLastName
	lastName		displayName
	secondLastName		email
	displayName		personalEmail
	email		startDate
	personalEmail		terminationDate
	startDate	FK	organizationId
	terminationDate	FK	userId
FK	organizationId		
FK	userId		

Ilustración 47 - Estructura original y modificada de la tabla Employee

Fuente: Elaboración propia

La Ilustración 48 muestra del lado izquierdo la estructura original de la tabla "User" en la base de datos y del lado derecho la modificación realizada.

User		User	
PK	<u>id</u>	PK	<u>id</u>
	email		email
			firstName
			lastName

Ilustración 48 - Estructura original y modificada de la tabla User

Fuente: Elaboración propia.

El desarrollo de esta característica implicó hacer modificaciones en las sentencias SQL del back-end donde se recuperaba la información de los empleados. Se le agregó a las sentencias relaciones y uniones por la izquierda para recuperar correctamente la información de ambas tablas.

Se le solicitó al practicante desarrollar el módulo para permitir a un Administrador o un Gerente invitar a un empleado a una organización. Para lograr lo mencionado, se generaron enlaces de invitación con un código único para cada invitación. Fue necesario que el practicante desarrollara las diferentes vistas para manejar el proceso de invitación.

Cuando un Administrador o Gerente crea un empleado, se genera automáticamente un enlace de invitación para ese empleado. El Administrador o Gerente puede copiar el enlace existente o generar uno nuevo, como lo muestra la Ilustración 49.

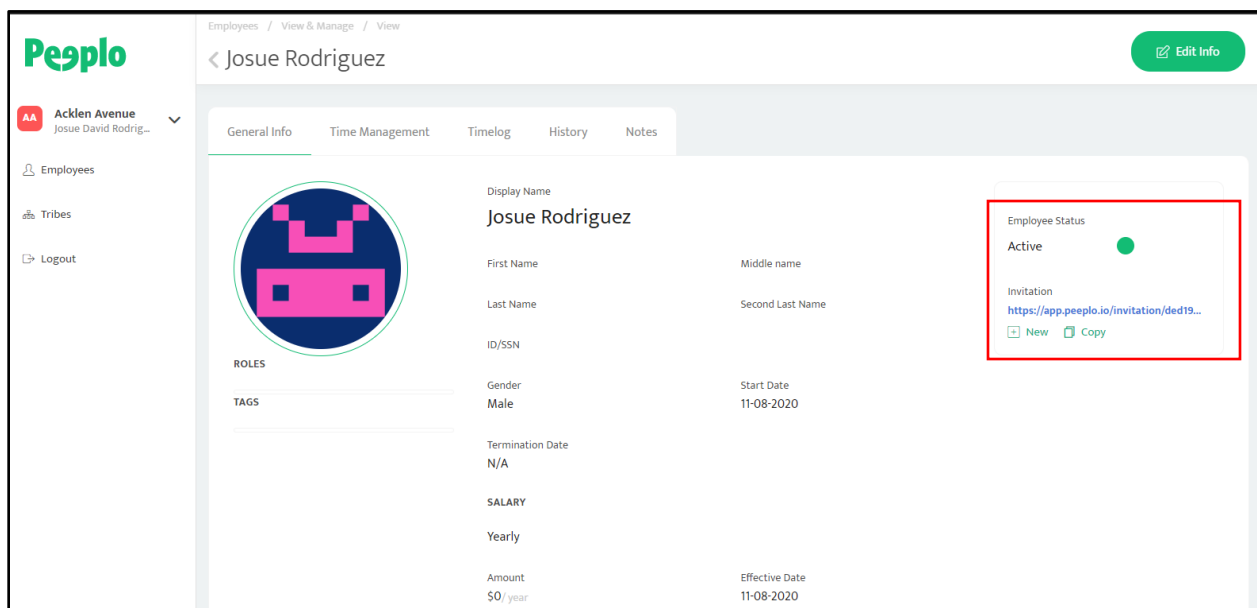


Ilustración 49 - Enlace de invitación en perfil de empleado

Fuente: (Peeplo, s.f.)

La Ilustración 50 muestra el enlace de invitación al ingresarlo en un navegador de internet. Lo que se encierra en el cuadrado es el código único para cada invitación, el cual está ligado al empleado que se desea invitar a la organización como se mencionó anteriormente.

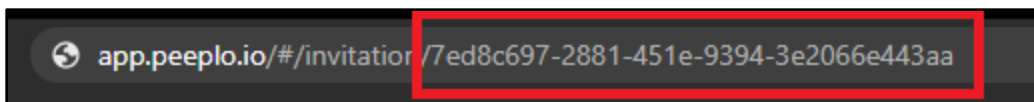


Ilustración 50 - Enlace de invitación a Peeplo en navegador

Fuente: (Peeplo, s.f.)

Al ingresar el enlace de invitación en el navegador de internet, se muestra una vista de cargado en donde se verifica que la invitación sea válida. La Ilustración 51 presenta esta vista.

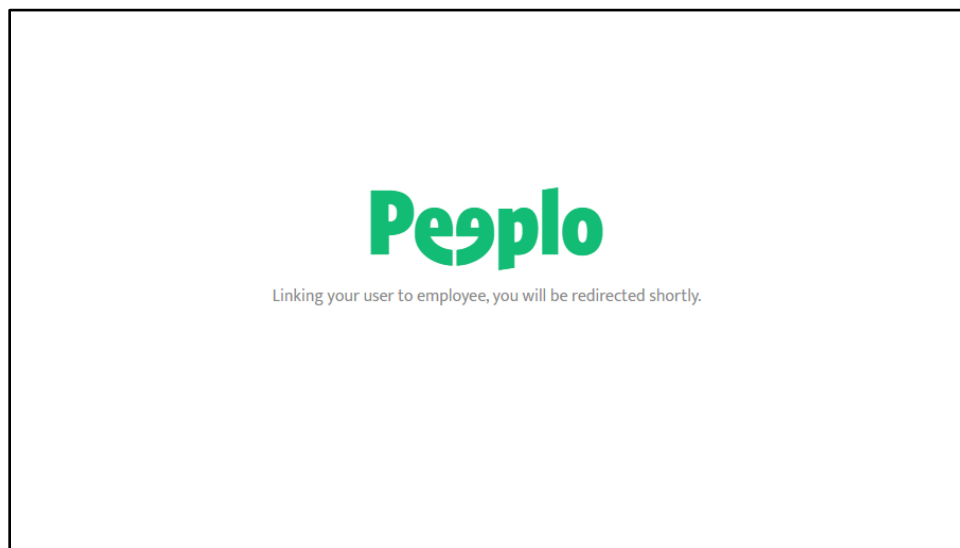


Ilustración 51 - Vista de validación de invitación

Fuente: (Peeplo, s.f.)

Si la invitación es válida y el usuario no ha iniciado sesión, lo redirige a la vista de inicio de sesión. Si el usuario tiene una sesión activa, muestra la invitación, en donde se despliega la información de la cuenta con la cual el usuario inició sesión, la información del perfil del empleado y la información de la organización a la cual se le está invitando. La Ilustración 52 exhibe la invitación con la información mencionada.

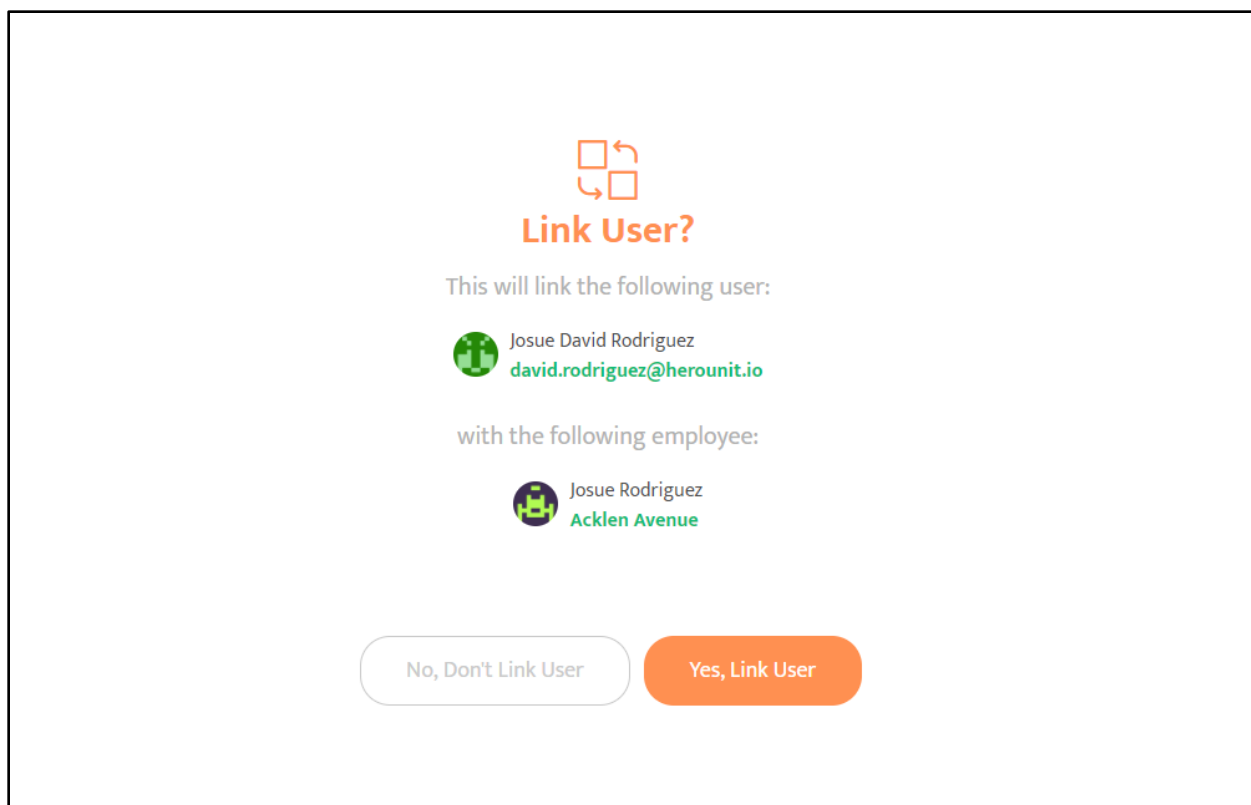


Ilustración 52 - Vista de invitación

Fuente: (Peeplo, s.f.)

Al aceptar la invitación, el usuario tendrá acceso a la organización con permisos de empleado, el enlace de invitación expirará y el usuario será redirigido a la página principal de Peeplo. Al negar la invitación, el usuario será redirigido a la página principal de Peeplo y el enlace seguirá siendo válido.

Si el usuario utiliza un enlace de invitación expirado, la plataforma muestra un mensaje diciendo que el enlace ya no es válido, como se presenta en la Ilustración 53.

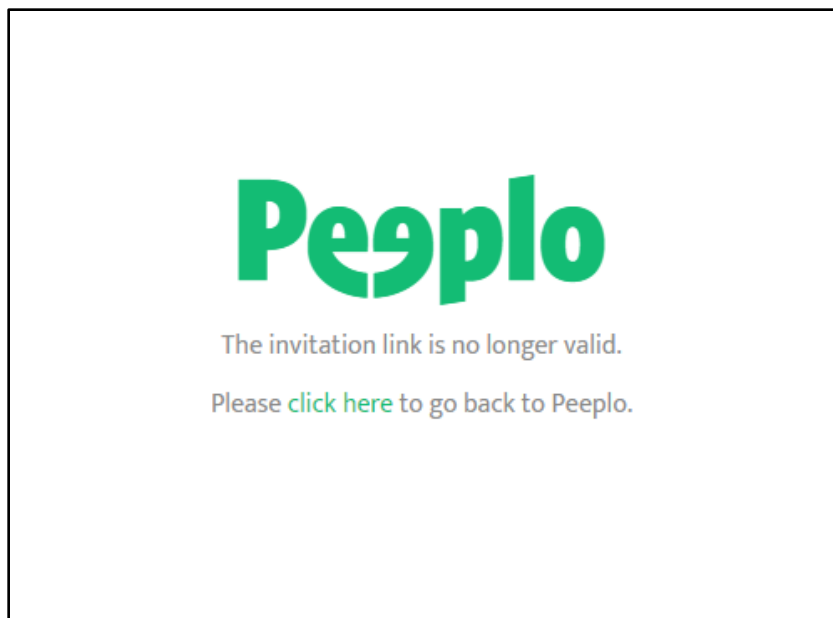


Ilustración 53 - Vista de invitación expirada

Fuente: (Peeplo, s.f.)

Para poder brindar un mejor desglose de las horas trabajadas de los empleados de una organización, se le solicitó al practicante agregar el campo "Description" al documento CSV que exporta la plataforma. Previamente se exportaban todos los campos de los registros de horas a excepción del mencionado. La Ilustración 54 muestra el documento CSV exportado con la columna "Description".

	A	B	C	D	E	F	G	H	I	J
1	date	name	paidOrUnpaid	duration	startTime	endTime	activity	project	description	
2	12/1/2020	Josue David Rodriguez Rodriguez	Unpaid	1:00 h	10:00 AM	11:00 AM	Meetings	Peeplo	Client meeting	
3	12/1/2020	Josue David Rodriguez Rodriguez	Paid	0:30 h	9:00 AM	9:30 AM	Testing	Peeplo	Running some regression tests on new features	
4	12/1/2020	Josue David Rodriguez Rodriguez	Unpaid	1:00 h	8:00 AM	9:00 AM	Development	Peeplo	Developing new features	
5										

Ilustración 54 - CSV de horas trabajadas con la columna "Description"

Fuente: (Peeplo, n.d.)

Con los acontecimientos ocurridos durante el paso de los huracanes Eta e Iota, el dueño del producto vio la necesidad de implementar un campo en la información del empleado donde se pueda ingresar contactos de emergencia para contactar o ubicar a los empleados.

Se especificó que este campo tuviera un límite de 500 caracteres debido a que dentro de él se generalizaría toda la información del contacto (nombre, parentesco, teléfono, correo, entre otros). La Ilustración 55 muestra el campo al momento de crear un empleado.

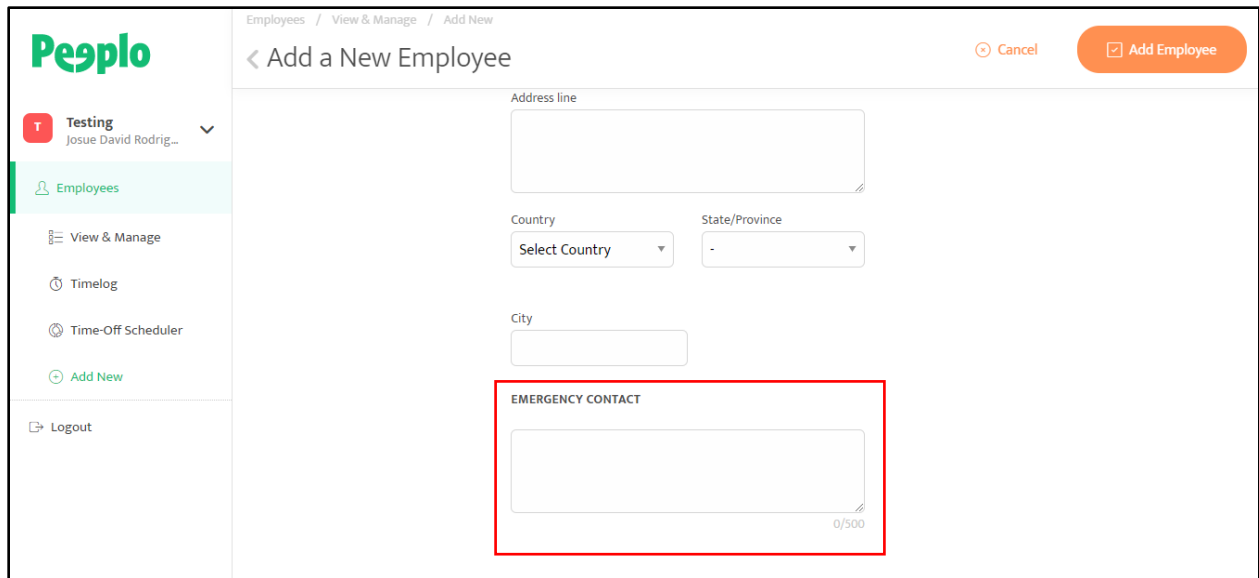


Ilustración 55 - Campo de Contacto de Emergencia

Fuente: (Peeplo, n.d.)

La siguiente nueva característica que el practicante implementó fue limitar el tiempo en que un empleado puede registrar sus horas trabajadas. Anteriormente, el empleado tenía hasta dos semanas para registrar sus horas. Ahora, el empleado cuenta únicamente con dos días para registrar las horas trabajadas. La Ilustración 56 muestra la validación creada.

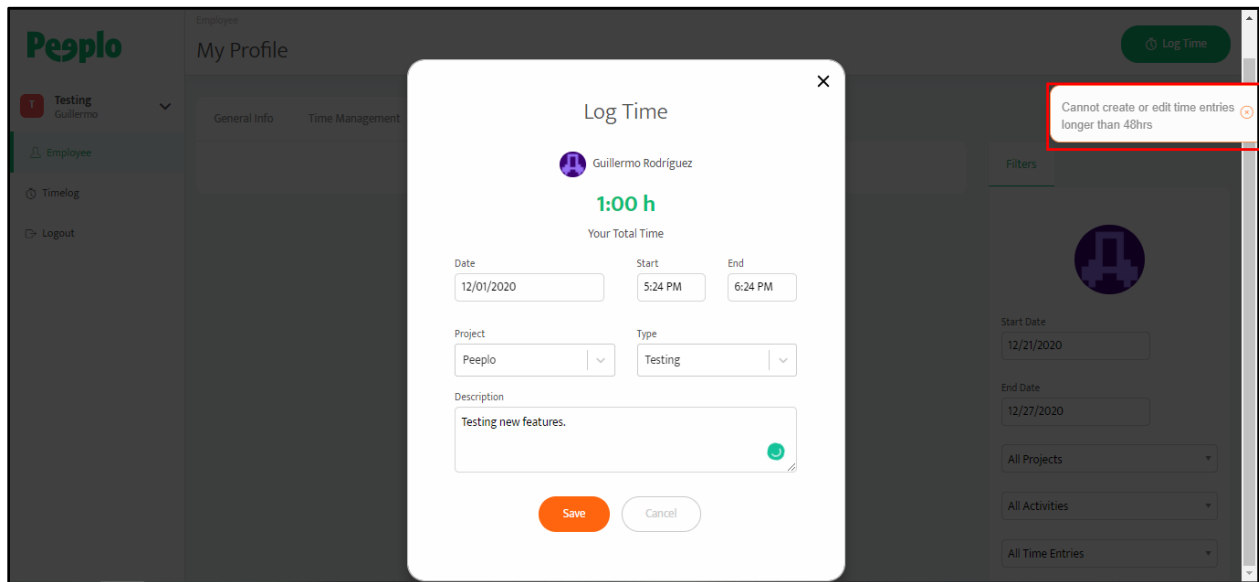


Ilustración 56 - Límite de 48 horas para registrar horas

Fuente: (Peeplo, n.d.)

Se le solicitó al practicante evitar que, al crear una organización nueva, se le creara un empleado por defecto al administrador. Una organización nueva debe contar con 0 empleados hasta que el administrador comience a crearlos y, si el administrador desea tener un empleado, crearse uno para sí mismo y utilizar el enlace de invitación.

El primer paso que se llevó a cabo para completar esta tarea fue eliminar el código que creaba automáticamente un empleado al administrador cuando se crea una nueva organización. La Ilustración 57 muestra una organización nueva sin empleados.

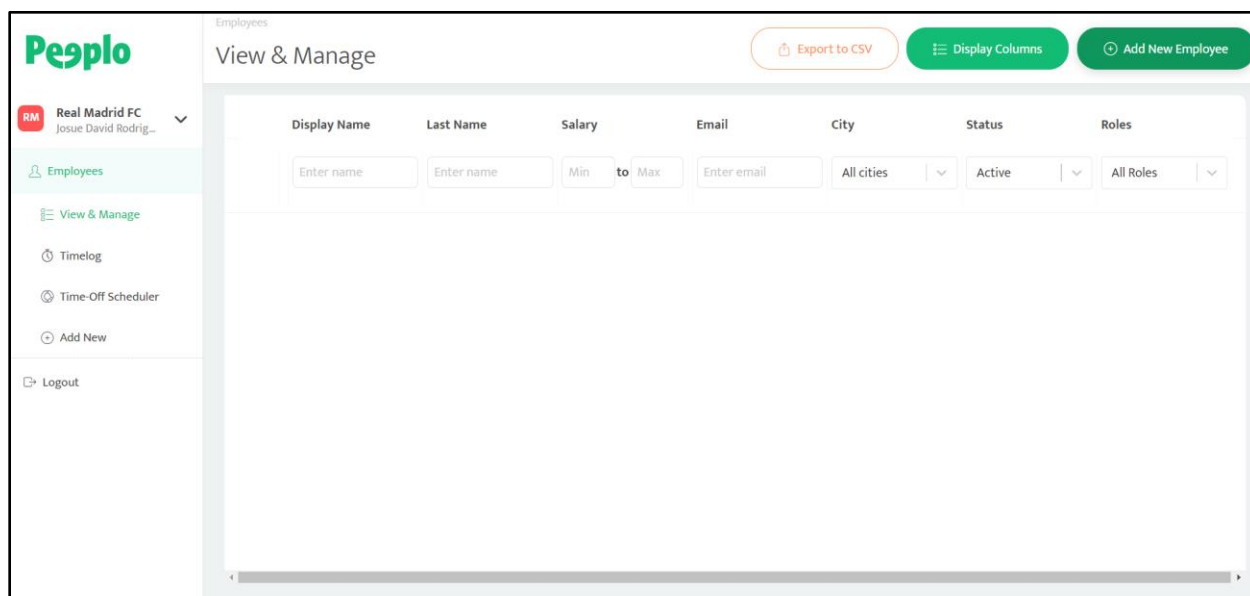


Ilustración 57 - Organización creada sin empleados

Fuente: (Peeplo, n.d.)

El siguiente paso consistió en modificar el historial de acciones realizadas. La plataforma Peeplo cuenta con un historial en donde se registran acciones como empleado creado, empleado borrado, empleado actualizado, hora registrada, hora actualizada, hora pagada, entre otros, y todas estas acciones estaban ligadas al ID del empleado que las realizó. Como de ahora en adelante el administrador no cuenta con un empleado por defecto, pero aún debe poder realizar las acciones previamente mencionadas, se necesitó alterar la estructura del historial y utilizar el ID del usuario y no del empleado.

Para lograr esto, se modificó la estructura de la tabla History en la base de datos como lo muestra la Ilustración 58. La parte izquierda muestra la estructura original de la tabla y la parte derecha

muestra la estructura modificada. Nótese que se decidió mantener el ID del empleado para el historial antiguo, dado que este maneja información sensible como la información de pago de los empleados.

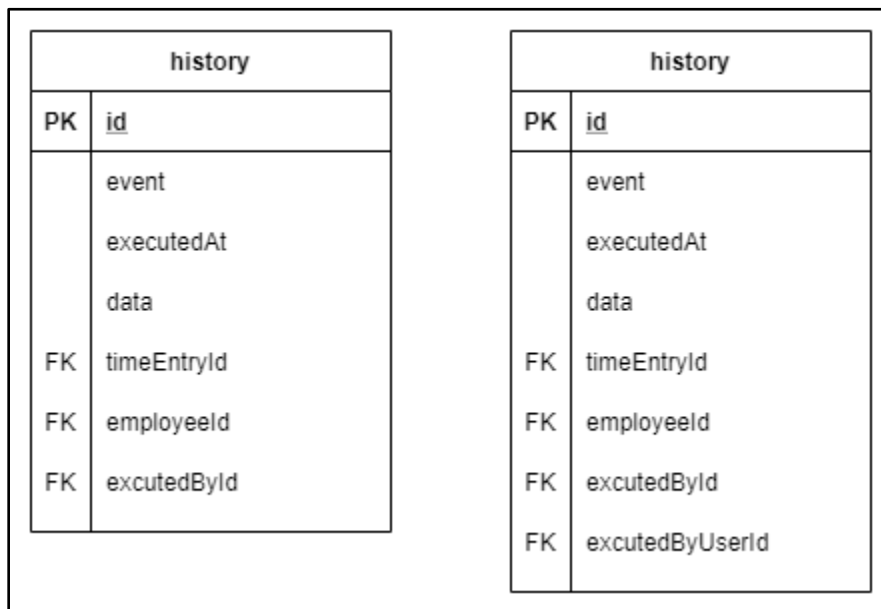


Ilustración 58 - Estructura original y modificada de la tabla History

Fuente: (Peeplo, n.d.)

Seguidamente, se creó una migración para poblar la nueva columna en la tabla "History" de las tuplas ya existentes, empleando uniones de tablas para poder recuperar el ID del usuario de la tabla "User" e insertarlo en la tabla "History", en la columna "executedByUserId".

Finalmente, se actualizaron los endpoints del back-end para que hicieran referencia a la nueva columna, se actualizaron los modelos de objetos del back-end y front-end y se actualizó la información que se muestra en el historial en el front-end. La Ilustración 59 muestra el historial de acciones.

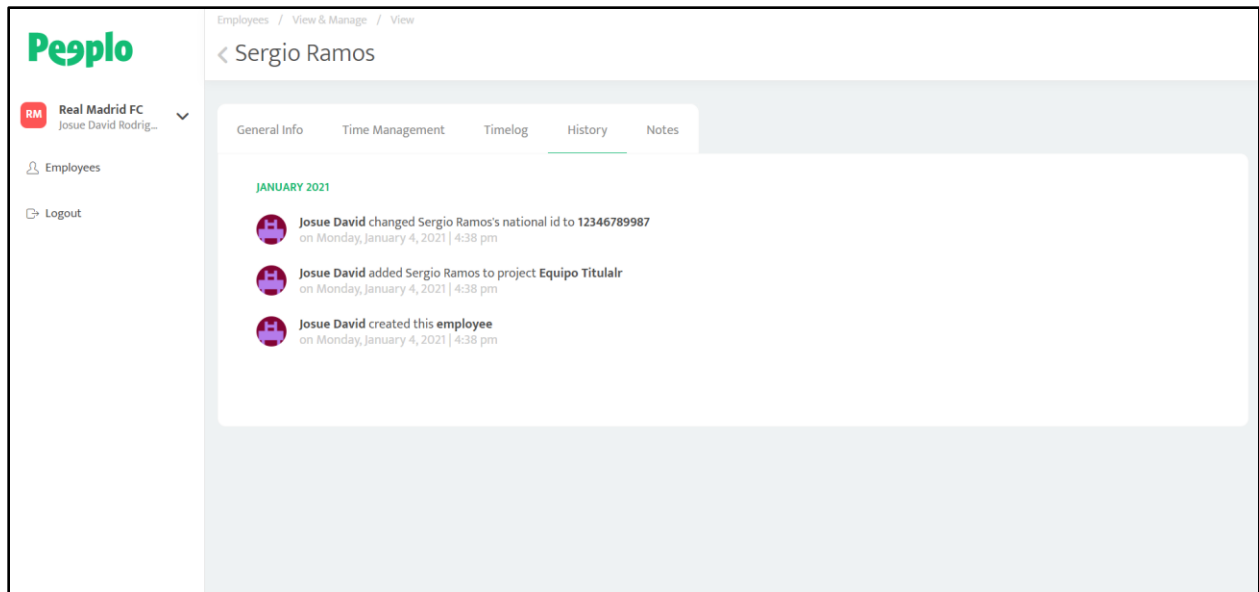


Ilustración 59 - Historial de acciones

Fuente: (Peeplo, n.d.)

4.2 CRONOGRAMA DE ACTIVIDADES

La Ilustración 60 muestra un diagrama de Gantt con las actividades realizadas durante el desarrollo de la práctica profesional.

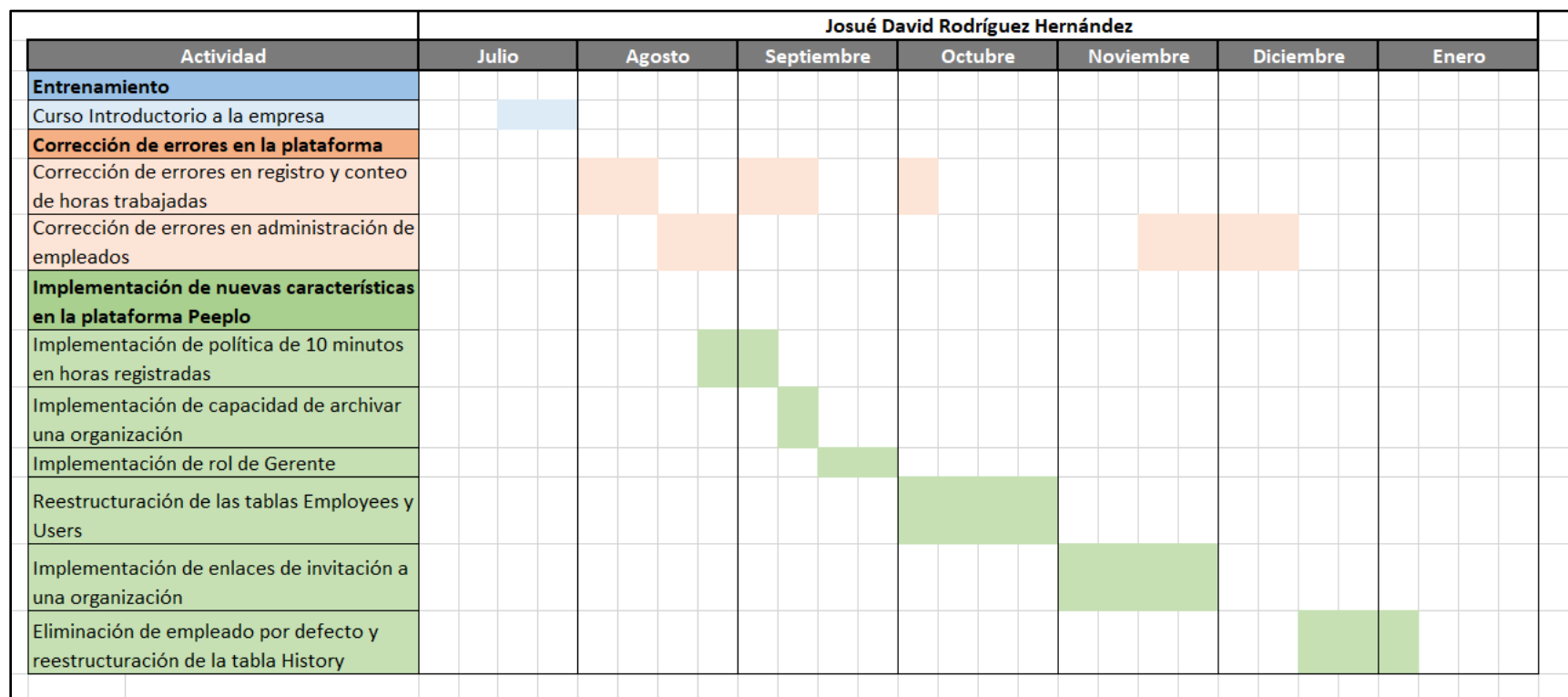


Ilustración 60 - Diagrama de Gantt

Fuente: Elaboración propia.

V. CONCLUSIONES

- Se continuó con el desarrollo de la plataforma Peeplo, llevándola a un nivel de estabilidad y funcionamiento listo para ser lanzada al mercado.
- Se implementaron nuevas características en la plataforma con el objetivo de brindar una propuesta de valor superior a la de plataformas similares.
- Se incrementó la estabilidad, la seguridad y la experiencia de usuario de la plataforma corrigiendo errores que impedían el funcionamiento correcto de la misma.

VI. RECOMENDACIONES

- Se deben implementar estándares de código y de formatos (formatos de fecha y hora, de declaración de clases, entre otros) para generar un programa más limpio y fácil de entender para futuros desarrolladores.
- Se debe establecer de manera más clara y estricta la documentación por parte de QA cuando encuentra un error en la plataforma. Dentro de la documentación se debería de incluir qué sucedió, cómo sucedió (pasos a reproducir), dónde sucedió (en qué ambiente, vista, con qué permisos, entre otros) y cualquier observación extra que el QA considere necesaria para la resolución del problema.

BIBLIOGRAFÍA

1. Acklen Avenue. (n.d.). *Acklen Avenue*. Recuperado agosto 1, 2020, de Acklen Avenue: <https://acklenavenue.com/>
2. Agencia Española de Protección de Datos. (2020, agosto 6). *Brechas de Seguridad*. Recuperado agosto 28, 2020, de Agencia Española de Protección de Datos: <https://www.aepd.es/es/derechos-y-deberes/cumple-tus-deberes/medidas-de-cumplimiento/brechas-de-seguridad>
3. Auth0. (n.d.). *Create a Tenant and Learn the Basics*. Recuperado agosto 28, 2020, de Auth0 Docs: <https://auth0.com/docs/get-started/learn-the-basics>
4. Auth0. (n.d.). *Get Started*. Recuperado agosto 28, 2020, de Auth0 Docs: <https://auth0.com/docs/get-started>
5. Auth0. (n.d.). *Start Building*. Recuperado agosto 28, 2020, de Auth0 Docs: <https://auth0.com/docs/>
6. Badgr. (n.d.). *Badgr*. Recuperado agosto 23, 2020, de Badgr: <https://info.badgr.com/>
7. Boscan, D. (2018, noviembre 14). *Decoradores en TypeScript*. Recuperado agosto 25, 2020, de Diego Boscan: <http://diegoboscan.com/decoradores-en-typescript/>
8. Campillo, A. (2020, enero 27). *¿Qué es React y para que sirve?* Recuperado agosto 24, 2020, de Drauta: <https://www.drauta.com/que-es-react-y-para-que-sirve>
9. Chapaval, N. (2017). *Qué es Frontend y Backend*. Recuperado agosto 23, 2020, de Platzi: <https://platzi.com/blog/que-es-frontend-y-backend/>
10. *DevOps - ¿Qué es metodología ágil?* (n.d.). Recuperado agosto 23, 2020, de Red Hat: <https://www.redhat.com/es/devops/what-is-agile-methodology>
11. Dorantes, C. A. (2015). *Qué es PostgreSQL y cuáles son sus ventajas*. Recuperado agosto 26, 2020, de Platzi: <https://platzi.com/blog/que-es-postgresql/>
12. Ecured. (n.d.). *Framework*. Recuperado agosto 23, 2020, de Ecured: <https://www.ecured.cu/Framework>

13. *Endpoint - What is an API Endpoint?* (n.d.). Recuperado septiembre 14, 2020, de Last Call The RapidAPI Blog: <https://rapidapi.com/blog/api-glossary/endpoint/>
14. Francia, J. (n.d.). *¿Qué es Scrum?* Recuperado agosto 28, 2020, de Scrum.org - The Home of Scrum: <https://www.scrum.org/resources/blog/que-es-scrum>
15. GitLab Inc. (n.d.). *What is GitLab?* Recuperado septiembre 13, 2020, de GitLab: <https://about.gitlab.com/what-is-gitlab/>
16. Google. (n.d.). *G Suite*. Recuperado septiembre 13, 2020, de G Suite: <https://gsuite.google.es/intl/es/>
17. JWT. (n.d.). *Introduction to JSON Web Tokens*. Recuperado agosto 28, 2020, de JWT: <https://jwt.io/introduction/>
18. Mezquita, F. (2018, Marzo 26). *WHAT DEFINES US - ACKLEN AVENUE, A SOFTWARE DEVELOPMENT COMPANY*. Recuperado agosto 01, 2020, de Acklen Avenue: <https://acklenavenue.com/blog/defining-acklen-avenue>
19. NestJS. (n.d.). *Documentación*. Recuperado agosto 25, 2020, de NestJS: <https://docs.nestjs.com/>
20. NestJS. (n.d.). *NestJS*. Recuperado agosto 25, 2020, de NestJS: <https://nestjs.com/>
21. Node.js. (n.d.). *Acerca de Node.js*. Recuperado agosto 25, 2020, de Node.js: <https://nodejs.org/es/about/>
22. Peeplo. (n.d.). *Landing Page*. Recuperado agosto 28, 2020, de Peeplo: <https://peeplo.io/>
23. Peeplo. (n.d.). *Peeplo*. Recuperado agosto 30, 2020, de Peeplo: app.peeplo.io
24. PostgreSQL. (n.d.). *About*. Recuperado agosto 26, 2020, de PostgreSQL: <https://www.postgresql.org/about/>
25. PostgreSQL. (n.d.). *Hot Standby*. Recuperado agosto 26, 2020, de PostgreSQL: <https://www.postgresql.org/docs/9.0/hot-standby.html>
26. PostgreSQL Tools. (n.d.). *pgAdmin*. Recuperado septiembre 12, 2020, de pgAdmin: <https://www.pgadmin.org/>

27. React. (n.d.). *React*. Recuperado agosto 28, 2020, de React: <https://es.reactjs.org/>
28. Red Hat. (n.d.). *El concepto de DevOps*. Recuperado septiembre 12, 2020, de Red Hat: <https://www.redhat.com/es/topics/devops>
29. Rubin, K. S. (2012). *Essential Scrum - A Practical Guide to the Most Popular Agile Process*. Michigan: Addison-Wesley.
30. Ruiz Santiago, J. A. (2020, abril 06). *Apificación de modelos en NestJS (I)*. Recuperado agosto 25, 2020, de Digital55: <https://www.digital55.com/desarrollo-tecnologia/apificacion-modelos-nestjs>
31. Tutorialspoint. (n.d.). *TypeORM - Quick Guide*. Recuperado agosto 25, 2020, de Tutorialspoint: https://www.tutorialspoint.com/typeorm/typeorm_quick_guide.htm
32. TypeORM. (n.d.). *TypeORM*. Recuperado agosto 25, 2020, de TypeORM: <https://typeorm.io/#/>
33. TypeScript. (n.d.). *TypeScript for JavaScript Programmers*. Recuperado agosto 28, 2020, de TypeScript: <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>
34. TypeScript. (n.d.). *What is TypeScript?* Recuperado agosto 28, 2020, de TypeScript: <https://www.typescriptlang.org/>
35. Universidad de Valencia. (2015). Bases de datos objeto-relacionales. *Diseño de Bases de Datos*. Valencia, España. Recuperado agosto 26, 2020, de http://www.patrimoniocultural.gob.cl/614/articles-33613_archivo_01.pdf
36. Universidad Nacional Autónoma de México. (n.d.). *Modelo Orientado a Objetos*. Recuperado agosto 26, 2020, de Unidad de Apoyo al Aprendizaje: https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html
37. W3Schools. (n.d.). *What is Full Stack?* Recuperado agosto 28, 2020, de W3Schools: https://www.w3schools.com/whatis/whatis_fullstack.asp