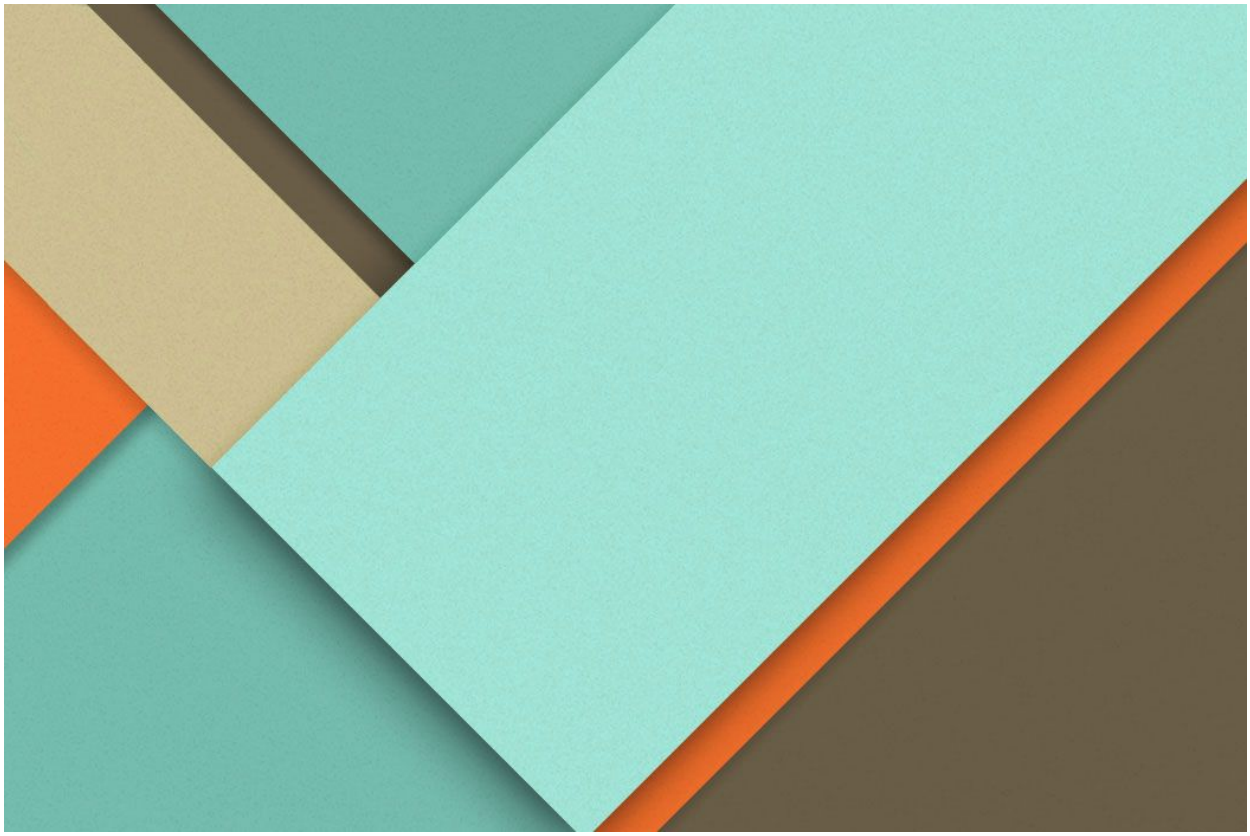


Manual técnico Te Vi Colombia

28.03.2019

Andrés Mauricio Montoya Sánchez
Te Vi Colombia
Girardot, Cundinamarca Colombia
2019

FASES DEL PROCESO DE DESARROLLO DEL SOFTWARE

Un proyecto de software a medida en la mayoría de los casos suele generar un impacto muy positivo en la empresa en donde se lleva a cabo ya que ha sido diseñado y creado según las necesidades exactas de quién lo ha pedido. Pero para que el impacto sea positivo realmente y que se generen beneficios, el proceso debe ser llevado con orden con buenas prácticas y estándares que aseguren los tiempos promedios, los costos y que el sistema cumpla con el papel para el cual fue creado cumpliendo con todos los requerimientos indicados por el cliente. (Neosystems, 2014); Según lo mencionado, estas fases consisten en:

ANALISIS

En la etapa de análisis se realiza la visión previa del desarrollo final del software, teniendo en cuenta el problema encontrado y la solución a aplicar con tecnologías de desarrollo que se van a mencionar, y aplicando métodos para la recolección y análisis de datos. La metodología de desarrollo seleccionada para el software, permite realizar incrementos y aplicando a cada una de las fases. Del resultado del análisis con herramientas para la recolección de datos, se encontraron las siguientes particularidades para el desarrollo del software:

- Visualización de formulación de estrategias para un análisis incentivo en áreas previamente establecidas.
- Estudio de técnicas para el mejoramiento y búsqueda de información requerida por un integrante de la aplicación.
- Eventualidades basadas en funciones de almacenamiento de información, para el consumo y estudio de determinados integrantes de la plataforma.
- Interfaz agradable y amigable para la visualización del usuario.

DISEÑO

Después de la información recolectada de la fase de análisis del desarrollo final, se encuentra que a manera general se utilizará elementos como: base de datos para el almacenamiento de información, interfaz adaptada a cualquier dispositivo para la visualización de datos, determinación de las tecnologías de desarrollo a utilizar para el planteo de información de una empresa o usuario registrado.

DESARROLLO

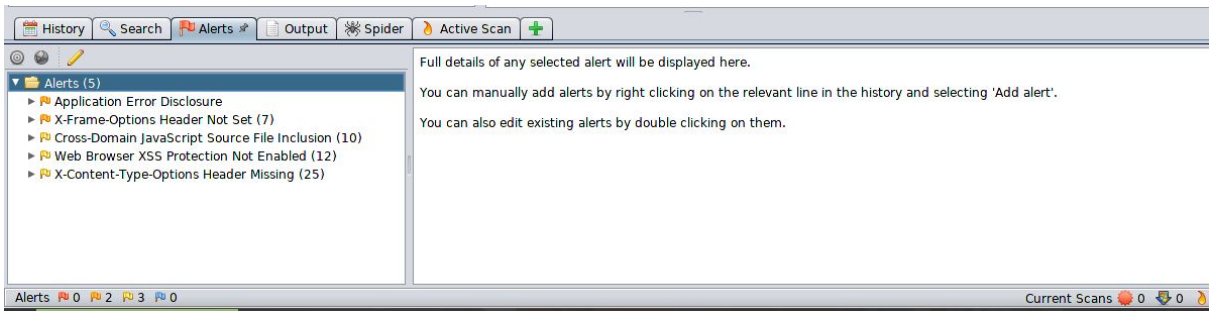
Con las herramientas de desarrollo anteriormente mencionadas y seleccionadas, se realiza el desarrollo de una aplicación bajo la guía de distintos diagramas. Estos se dividen en dos:

- API (Application programming interface). Encargada de recibir y procesar datos para visualizarlos en la aplicación.
- SPA (Single page application). Aplicación web encargada de recibir la información de la API.

PRUEBAS

En el desarrollo de funciones en particulares áreas de la aplicación, se hará sus debidas pruebas, junto a herramientas como Jest qué es una tool para el testeo de aplicaciones escritas en Javascript; Asegurando que el código devuelve lo esperado. También se incluye la herramienta de seguridad como OWASP Zed Attack Proxy (ZAP) siendo la tool de seguridad informática más usada en el mundo.

En este caso, estos son los resultados que dio OWASP Zed Attack Proxy (ZAP):



Fuente: autor.

Como se puede ver, no se encontró ninguna vulnerabilidad grave, lo cual está bien. Pero lo interesante en este análisis es que Te Vi Colombia fue puesto a prueba sin tener algún Firewall, CDN o Proxy reverso detrás protegiéndolo, y a pesar de ello, supo protegerse de una manera positiva.

Obviamente, esta aplicación en producción con el servicio de Now será más segura.

DOCUMENTACIÓN

- Manual de usuario: Consiste en un documento de texto el cual explicará el funcionamiento de la aplicación. Tratando temas como, qué navegadores son compatibles, cuantas pulgadas debe de tener un dispositivo y cómo cada función del software trabaja.
- Manual técnico: Se indica el propósito y breve descripción de cada método/función, con su prototipo indicando argumentos y respuestas.

IMPLEMENTACIÓN Y MANTENIMIENTO

La aplicación requiere de inversión económica para el pago de servidores, DNS, Databases as a service, entre otros, que van a hacer implementados en servicios de la nube como lo pueden ser Now y Google Cloud for Databases and Storage, con relación a otras tecnologías como lo pueden ser Node.js, GraphQL, React.js, MySQL, Redis y demás, estás cuentan con licenciamiento de software libre, lo cual hace que a nivel de mantenimiento no haya ningún problema financiero o legal. Todo esto corresponde a los gastos a la mano de obra de los administradores.

HERRAMIENTAS

En el desarrollo de la aplicación final, se utilizaron las siguientes herramientas para el diseño y desarrollo:

- React.js
- Next.js
- Node.js
- GraphQL
- MySQL
- Redis
- TypeScript

PROPUESTA SOLUCIÓN

El sistema de información para la conexión de tejidos virtuales en Colombia está orientado en un Ambiente Web, el cual se basa en el estudio de acompañamiento y ayuda en Emprendimiento, Práctica profesional y Empleabilidad.

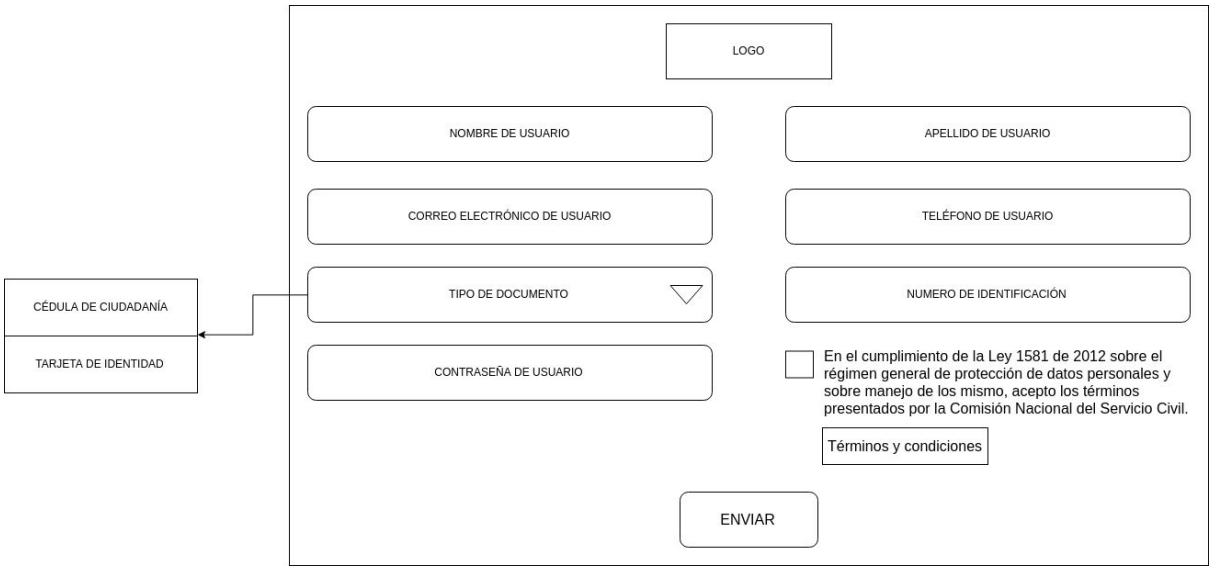
Esta exploración logra su objetivo a través de la información que se encuentra almacenada en una base de datos, las cual son representaciones que sintetizan

algunos de los factores, parámetros o características más relevantes para seleccionar el beneficiario o empresa más apropiada en función de los objetivos perseguidos, las circunstancias del entorno y los recursos y capacidades que este ofrece.

Este software se orienta en los posteriores módulos:

REGISTRO DE USUARIO.

El usuario o empresa puede crear una cuenta, con la cual se podrá identificar en el aplicativo.



Fuente: Autor.

Únicamente se pide información muy básica para lograr diferenciar a cada usuario o empresa.

Los términos y condiciones están inspirados en el registro de la página de empleabilidad de la UNIMINUTO http://empleabilidad.uniminuto.edu/, el cual cuenta con la suficiente información descriptiva para establecer una cláusula segura ante los datos de los usuarios.

Términos y condiciones para usuarios.

Los datos personales suministrados por el Titular serán utilizados por UNIMINUTO para identificar, recopilar y registrar a los prospectos de cualquier persona interesada en los

servicios que ofrece la plataforma *Te vi Colombia*, con el fin de poder prestar sus servicios de generar conexiones de ayuda y alternativas para poder emprender.

Así mismo los datos se recopilan con el fin de mantener actualizada la información del beneficiario, contactarlo, verificar sus datos, identificar las necesidades de formación y capacitación de estudiantes, realizar informes, análisis de información, búsqueda de tendencias en los servicios, enviar información, obtener indicadores, generar información institucional para los procesos de acreditación, generar reportes a entidades externas y requerimientos internos de información.

Los datos serán objeto de recolección, almacenamiento, actualización y copia de seguridad de acuerdo con el tratamiento establecido en la Política de Tratamiento de información de UNIMINUTO.

El responsable y Encargado de Tratamiento de los datos será UNIMINUTO.

Sus datos se mantendrán almacenados por el periodo establecido en el Manual de Protección de Datos Personas de UNIMINUTO.

El Titular tiene derecho a conocer, actualizar, rectificar, renovar, solicitar la supresión, presentar quejas y reclamos y demás derecho contenidos en la Ley 1581 de 2012 y sus decretos reglamentarios, respecto de los datos suministrados.

Declaro bajo la gravedad de juramento que todos los datos aquí contenidos son exactos y veraces y declaro conocer la Política de Tratamiento de información de UNIMINUTO, en el cual me ha informado de manera previa y expresa los derechos que se asisten, la finalidad, tratamiento y vigencia que se le dará a mis datos personales.

En consecuencia de lo anterior, autorizo expresamente de manera libre, previa, voluntaria y debidamente informada, a la Corporación universitaria Minuto de Dios - UNIMINUTO, para que haga el Tratamiento de datos, de acuerdo con las finalidades y condiciones mencionadas en el aviso de privacidad, el cual declaro conocer y aceptar.

Términos y condiciones para empresas.

Los datos personales suministrados por el Titular serán utilizados por *Te vi Colombia* con el fin de canalizar oportunidades de cooperación y/o donaciones, trabajos conjuntos, proyectos, emprendimiento, etc. De igual manera, se utiliza esta formación para conocer los aliados de UNIMINUTO.

Los datos serán objeto de recolección, almacenamiento, actualización y copia de seguridad de acuerdo con el tratamiento establecido en el Manual de Protección de Datos Personales de UNIMINUTO.

El Responsable y Encargado del Tratamiento de los datos será UNIMINUTO.

Sus datos se mantendrán almacenados por el periodo establecido en el Manual de Protección de Datos Personales de UNIMINUTO.

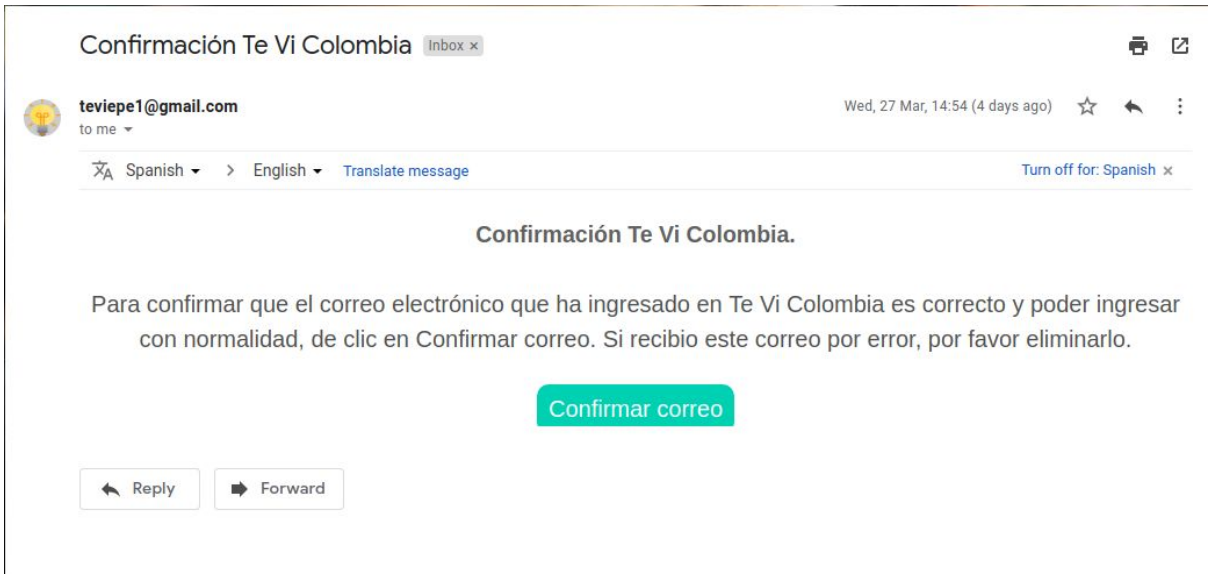
El Titular tiene derecho de conocer, actualizar, rectificar, revocar, solicitar la supresión, presentar quejas y reclamos y demás derecho contenidos en la Ley 1581 de 2012 y sus decretos reglamentarios, respecto de los datos suministrados.

Autorización.

Declaro bajo la gravedad de juramento que todos los datos aquí contenidos son exactos y veraces y declaró conocer el Manual de Protección de Datos personales de UNIMINUTO, en el cual me ha informado de manera previa y expresa los derechos que me asisten, la finalidad, tratamiento y vigencia que se le dará a mis datos personales.

En consecuencia de los anterior, autorizo de manera libre, previa, voluntaria y debidamente informada, a la Corporación universitaria Minuto de Dios - UNIMINUTO, para que haga el Tratamiento de datos, de acuerdo con las finalidades y condiciones mencionadas en el aviso de privacidad, el cual declaro conocer y aceptar.

Una vez los términos y condiciones aceptados, el usuario o empresa, para saber que el correo ingresado no es falso, se verifica a través de un correo electrónico que el usuario debe de revisar para validar su cuenta de usuario.

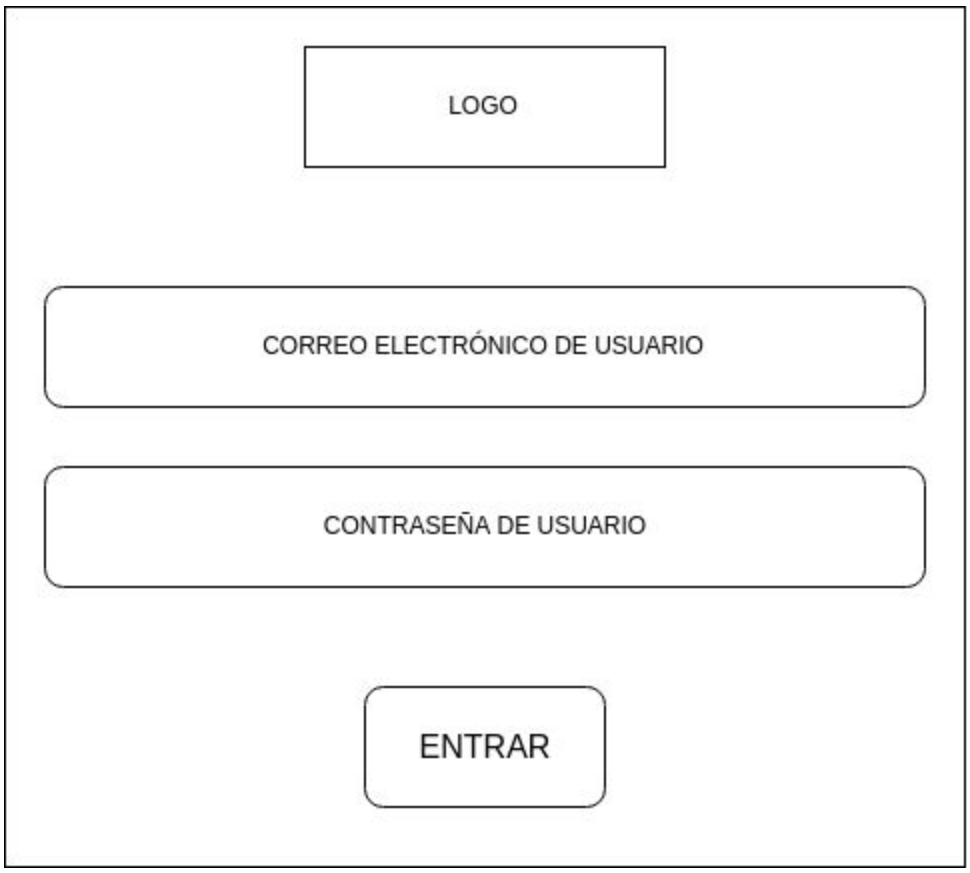


Fuente: Autor.

Dando clic en el texto “Confirmar correo” con subrayado, el sistema comprobará si en realidad la cuenta es de un usuario verdadero.

LOGIN DE USUARIO.

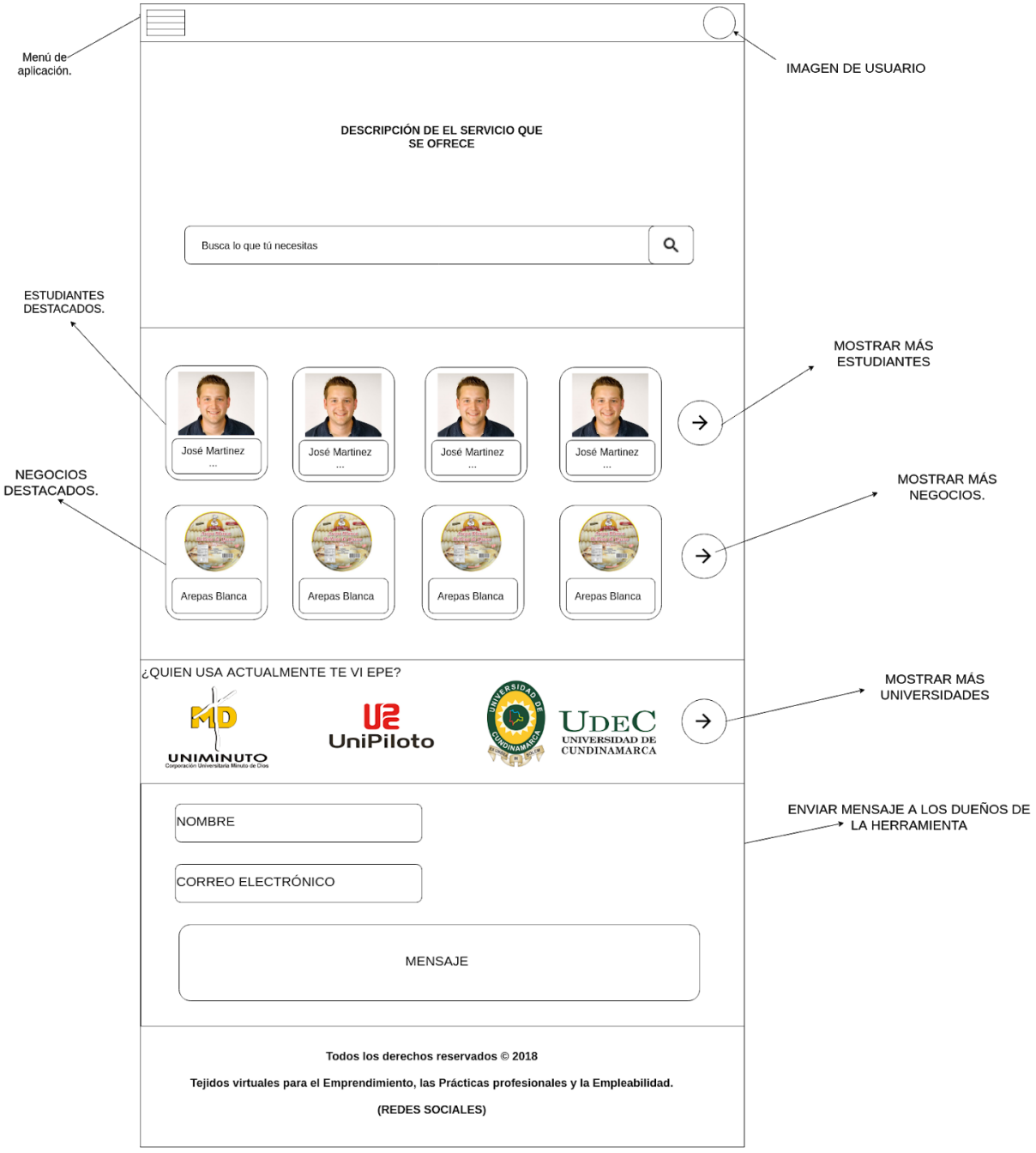
Una vez el usuario creado y correo validado, este procederá con entrar a la aplicación únicamente con su correo y contraseña, dando clic en *ENTRAR*.



Fuente: Autor.

PÁGINA DE INICIO.

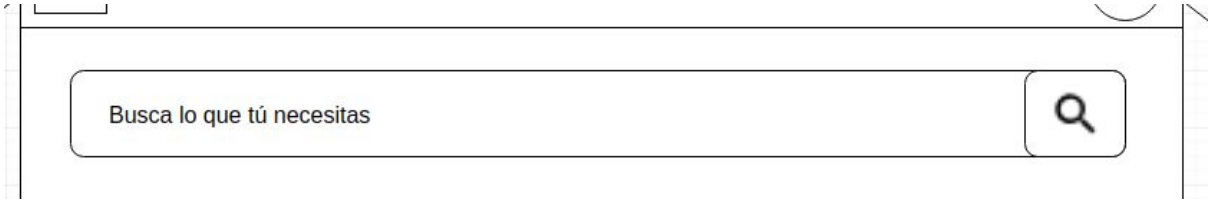
Ya el usuario dentro de la aplicación, lo primero que observará es la página de inicio, la cual contendrá la siguiente vista:



Fuente: Autor.

Esta sección cuenta con cuatro (4) características importantes, las cuales son la lista de usuarios y empresas (negocios) más destacados entre todas las carreras, el buscador que ayuda a filtrar el contenido, las universidades, colegios, compañías, organizaciones más interesadas en el software y el servicio de ayuda.

FUNCIONALIDAD BUSCADOR.



Fuente: Autor.

En el caso del buscador, tiene una manera particular de trabajar, dependiendo de la sentencia ingresada, este logrará identificar lo necesario respetando las siguientes opciones:

En el caso de un usuario (Más información en *Mi perfil*), sería:

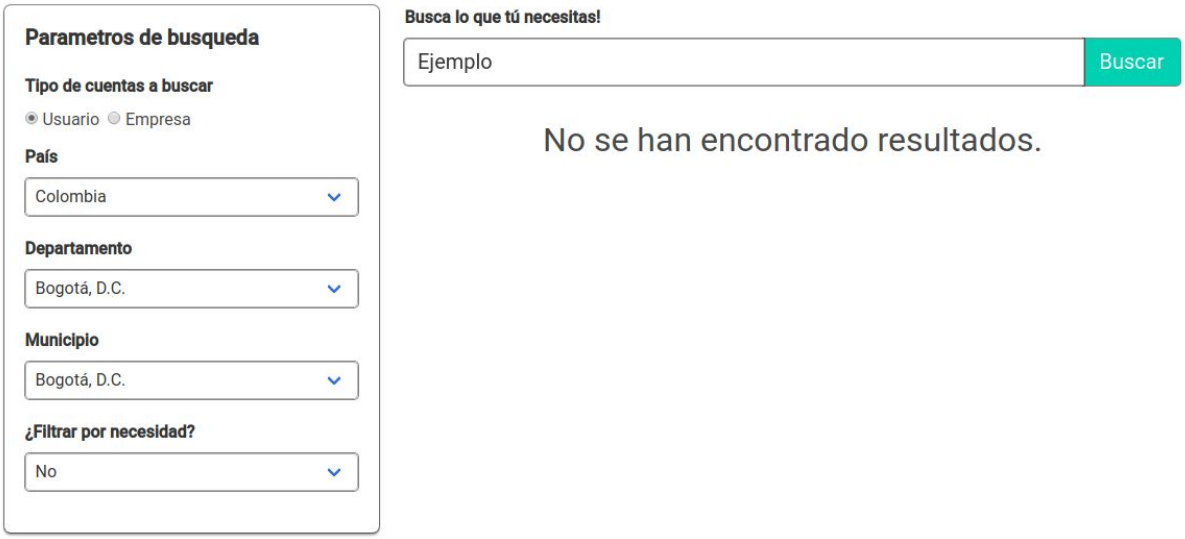
- Nombre.
- Apellido.
- Descripción.
- Habilidades.

En el caso de una empresa (Más información en Empresa), sería:

- Nombre de compañía.
- Descripción.
- Habilidades.
- Puesto de trabajo.
- Habilidades (Habilidades basadas en puesto de trabajo requerido).

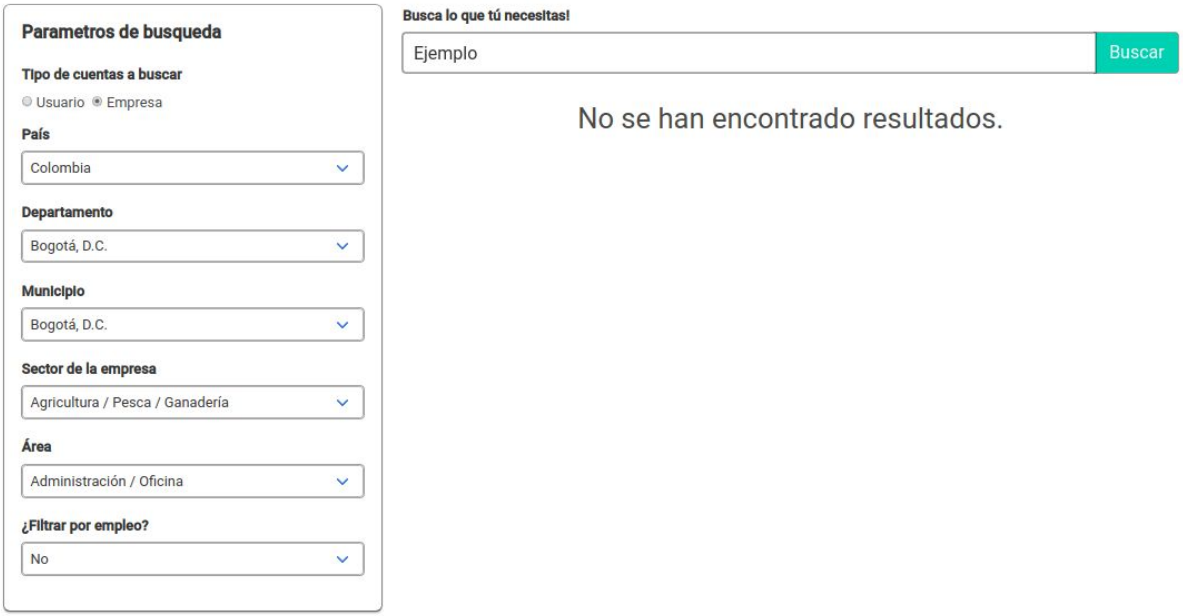
Esté se filtra a través de dos categorías “Usuarios” y “Empresas”, que se dividen en diferentes parámetros que sirven para que la búsqueda sea más específica.

- Usuarios.
    - Departamento.
    - Municipio.
    - País.
    - Necesidad. (Necesidades sin completar)



Fuente: Autor.

- Empresa.
    - Departamento.
    - Municipio.
    - Nacionalidad.
    - Sector.
    - Área.
    - Empleo.

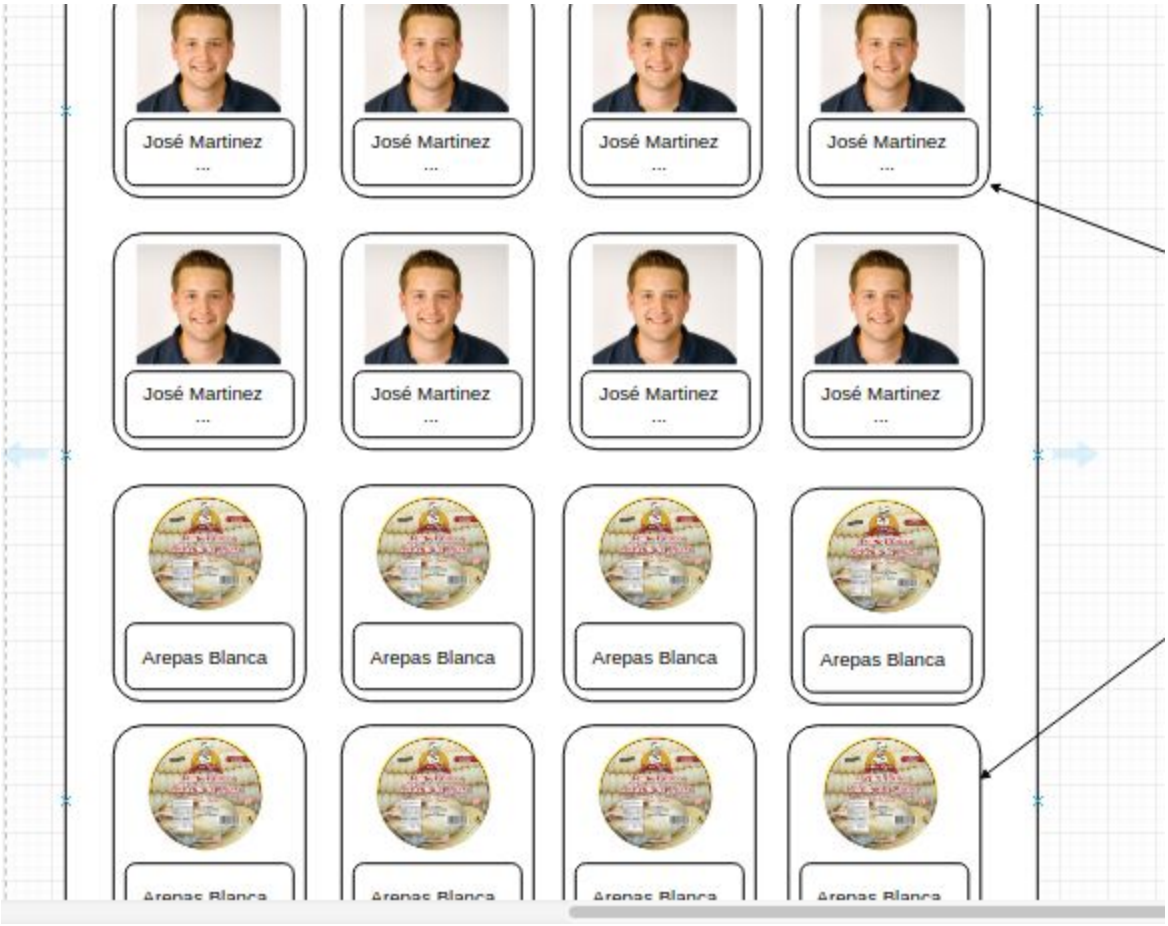


Fuente: Autor.

Estos resultados van a depender mucho de qué tan apoyado sea un usuario o empresa, todo esto basado en la cantidad de estrellas obtenidas por medio de la categoría Feedback, éstas se categorizan por ser la forma por la cual un beneficiario puede aumentar su rango en la aplicación, se obtienen por medio del puntaje dado por algún usuario o empresa de la plataforma.

FUNCIONALIDAD DE LISTADOS DE USUARIOS Y NEGOCIOS.

El listado de usuarios y empresas va ir cambiando cada vez que las estrellas de estos aumenten o disminuyen, haciendo referencia a la cantidad de estrellas obtenidas por el Feedback.



Fuente: Autor.

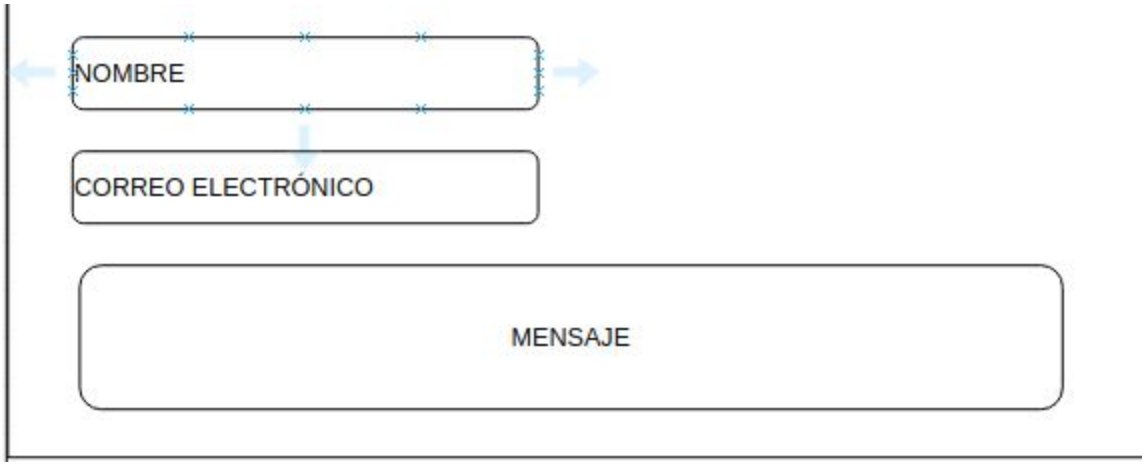
INTERESADOS EN TE VI COLOMBIA.

Fuente: Autor.

Este apartado está más orientado como una pequeña estrategia de marketing y a la vez generar confianza al visitante, para que sepa que universidades, corporaciones o compañías confían en el producto *Te vi Colombia*. Esta funcionalidad se añadirá cuando se tenga una buena cantidad de información recolectada.

SERVICIO DE AYUDA.

Simplemente con ingresar el nombre, correo y mensaje del remitente, esta información se enviará al correo que se esté manejando para atender las inquietudes de los usuarios o visitantes.



Fuente: Autor.

MENU FUNCIONAMIENTO.

Fuente: Autor.

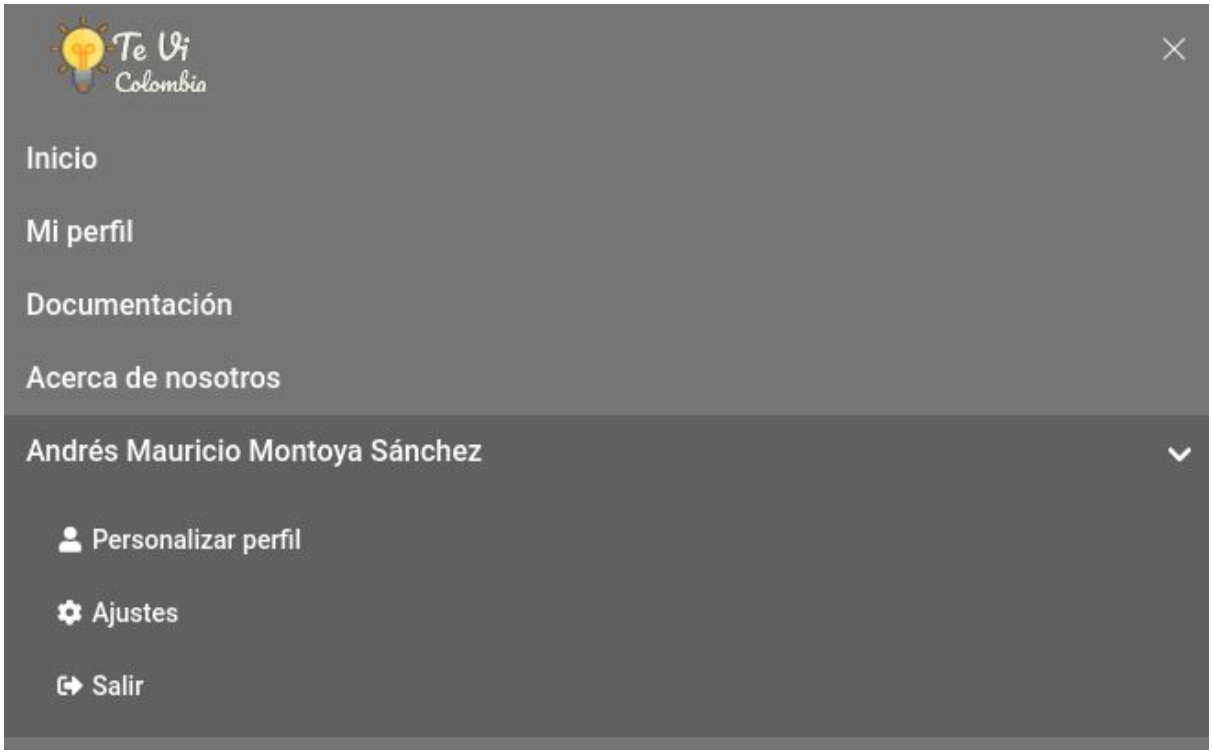
Este cuenta con dos apartados, los cuales son el botón de Menú y configuración de usuario.

En el círculo, al dar clic se muestra un menú.

Fuente: Autor.

En el cual usuario podrá entrar a “personalizar perfil” para cambiar la información de la empresa o usuario, “Ajustes” cambiar correo electrónico principal o contraseña, y en “salir”, simplemente saldrá del sistema.

En el menú, al dar clic, se encuentra una serie de rutas, las cuales tienen documentación de la aplicación y vista y personificación del perfil, tanto usuarios como empresas.



Fuente: Autor.

MI PERFIL.

Esta sección cuenta con mucha información por parte del usuario registrado, ya que en este se deriva su monto de estrellas, comentarios positivos y negativos, información general, qué formación y empleo tuvo y tiene actualmente, si cuenta con alguna empresa, con que necesidad cuenta y descripción de los proyectos que ha hecho o colaborado.

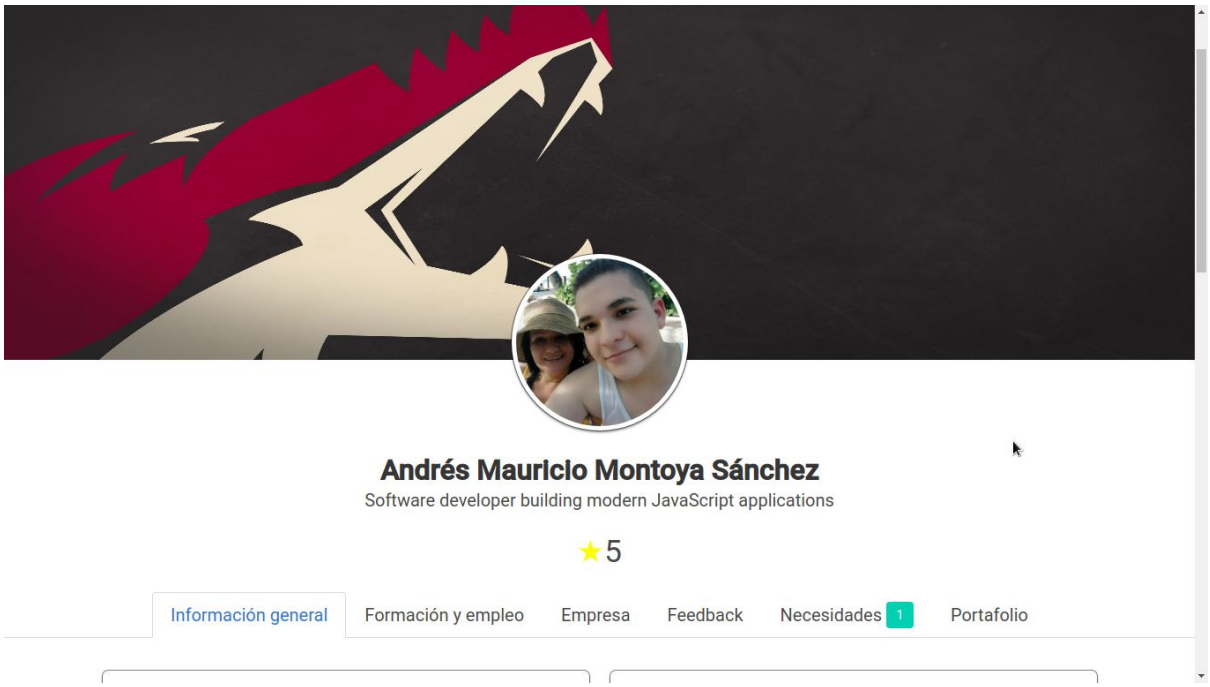
Toda esta información se divide en:

- Información general.
    - Nombre.
    - Apellido.
    - Descripción.
    - Tipo de documento de identidad.
    - Documento de identidad.
    - Dirección.
    - Teléfono.
    - Departamento.
    - Municipio.

    - Nacionalidad.
    - Fecha de nacimiento.
    - Estado civil.
    - Sitio web personal.
    - Genero.
    - Habilidades.
    - Redes sociales.
    - Preferencia de empleo.
    - Idiomas.

- Formación y empleo.
    - Estudio.
        - Lugar.
        - Nivel.
        - Area (Unicamente para grados universitarios).
        - Periodo de tiempo (Inicio o culminación).
    - Trabajo.
        - Nombre de compañía.
        - Puesto de trabajo.
        - Departamento.
        - Sector.
        - Area.
        - Descripción y metas obtenidas.
        - Periodo de tiempo (Inicio o culminación).
    - Anexos.
        - Descargar o subir anexos en word o pdf.
- Feedback.
    - Cantidad de estrellas y comentarios obtenidos por usuarios o empresas.
- Empresa.
    - Lista de empresas asociadas al perfil.
- Necesidad
    - ¿Con qué necesidad cuanta el usuario? (Ejemplos)
        - Dónde hacer la práctica profesional.
        - Necesidad de empleo.
        - Y más necesidades…
- Portafolio
    - ¿Qué proyectos ha hecho un usuario? (Ejemplos)
        - Creación de aplicación móvil. Descripción. Imágenes y vídeos.
        - Ayuda voluntaria a niños pobres en Colombia.

Todo este contenido se divide entre secciones, las cuales dependiendo de las que más interés genere al usuario, se mostrará su respectiva información.



Fuente: Autor.

EMPRESA.

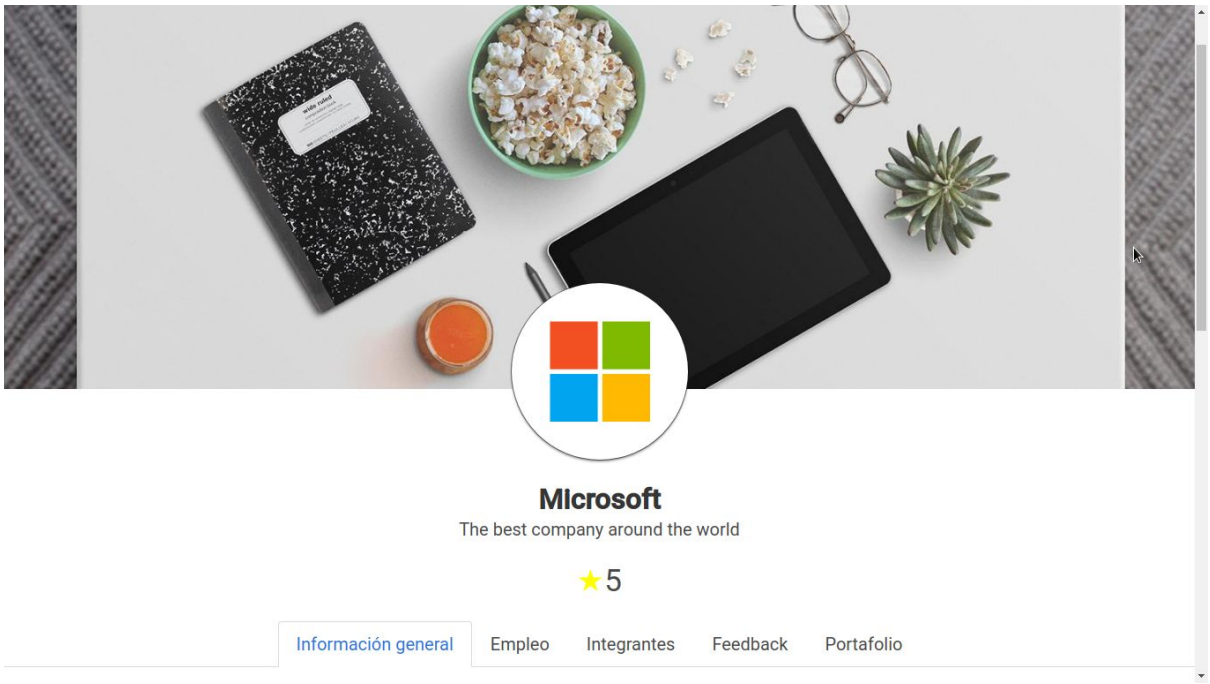
Esta sección cuenta con mucha información por parte del negocio registrado, ya que en este se deriva su monto de estrellas, comentarios positivos y negativos, información general, puestos de trabajo requeridos, localización exacta por Google Maps y con que integrantes cuenta la empresa.

Toda esta información se divide en:

- Información general
  - Nombre.
  - Descripción.
  - Teléfono.
  - Correo electrónico.
  - Nacionalidad.
  - Departamento.
  - Ciudad.

    - Dirección.
    - Redes sociales.
    - Sitio web.
    - Localización Google Maps.
- Feedback.
    - Cantidad de estrellas y comentarios obtenidos por usuarios o empresas.
- Vacantes de empleo.
- Portafolio.
- Integrantes.
    - Lista de usuarios registrados que hacen parte de la empresa.

Todo este contenido se divide entre secciones, las cuales dependiendo de las que más interés genere al usuario, se mostrará su respectiva información.



Fuente: Autor.

ALCANCES FUTUROS

Te vi Colombia tiene la meta de ofrecer el servicio de la herramienta por un año para saber cómo esta influye en la ciudad de Girardot, Cundinamarca. Si esto genera resultados favorables, se puede llegar a la conclusión de expandir esta aplicación en otros lugares de Colombia. Así mismo, se seguirá trabajando en el desarrollo de esta,

agregando la funcionalidad de poder administrar la información de la plataforma o usuarios.

LIMITACIONES

Las limitaciones que se pueden encontrar en el desarrollo e implementación en el sistema de información son:

- Seguridad. Se está trabajando arduamente para que no existan este tipo de errores, para ello lo mejor es frecuentemente actualizar las dependencias de la aplicación.
- Aceptación. Esta puede tomar tiempo, ya que relativamente este software es nuevo y lleva poco tiempo en el mercado.

ESTRUCTURA DE SOFTWARE

BASE DE DATOS

La base de datos principal, donde se guarda la información de cada usuario está basada en el motor de base de datos MySQL.

En el carpeta de `Docker` se puede encontrar un archivo con el nombre de `docker-compose.yml`, en este se encuentra la configuración e instancias de las herramientas que necesita Te Vi Colombia, si lo ejecuta con:

```
docker-compose up --build
```

Se iniciará toda la aplicación en modo `Development`. Este archivo trae consigo que puertos de entrada y salida maneja MySQL, Redis, Node.js y PhpMyAdmin. En este caso el servicio de PhpMyAdmin viene con la configuración para que dentro de Docker corra el puerto 80 y externamente el 8000:

```
phpmyadmin:
    image: phpmyadmin/phpmyadmin
    links:
      - db:db
    ports:
```

```
    - 8000:80
  environment:
    MYSQL_USER: root
    MYSQL_PASSWORD: root
```

Por ello, para ingresar a PhpMyAdmin solo basta con ingresar en algún navegador web http://localhost:8000 y se mostrará la configuración de esta herramienta. Se incluye esta funcionalidad para que pueda cualquier desarrollador ver la aplicación de base de datos de una manera gráfica y amigable.

Hay muchas tablas en esta base de datos las cuales cumplen con tareas completamente diferentes, por ello aquí se mencionara la descripción y funcionalidad de cada una, cada variable contará con variables en paréntesis con su respectivo nombre en la tabla.

Recordemos que a nivel de código, cada tabla cuenta con su propia Entidad (Entity) la cual describe la estructura de la base de datos en código TypeScript con el ORM TypeORM, se recomienda leer estas entidades ya que son más descriptivas y claras si el lector domina el idioma Inglés en un nivel básico.

TABLA USER

Esta se encarga de guardar los usuarios de la aplicación, la primera vez que un usuario va a ingresar a Te Vi Colombia, es necesario que ingresa:

- Nombres (Name)
- Apellidos (Lastname)
- Tipo de documento de identificación (identificationDocumentType)
- Documento de identificación (identificacionDocument)
- Indicativo de teléfono por país (telephoneCountry)
- Teléfono de usuario (telephone)
- Correo electrónico (email)
- Contraseña (password)

Una vez hecho, simplemente se enviará un correo al usuario para saber si este no es un bot o ingreso un correo falso. Los demás datos de la tabla se llenan en la sección “Personalizar perfil”.

user
id
routePhoto
routeCover
name
lastname
description
identificationDocumentType
identificationDocument
address
telephoneCountry
telephone
telephone2Country
telephone2
departament
town
nationality
birth
civilStatus
website
gender
disability
optionalEmail
email
password
skills
socialnetwork
confirmed
forgotPasswordLocked
preferworkId

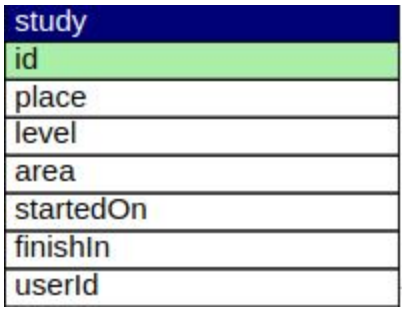
Fuente: Autor.

El campo `confirmed` y `forgotPasswordLocked` son simplemente campos para proteger la cuenta al hacer un cambio de contraseña y confirmar que la cuenta es validada.

Los demás datos son personalizables en la sección anteriormente mencionada, a excepción de `peferworkId`, simplemente es una foreign key con una relación One to One a `PreferWork`.

TABLA STUDY

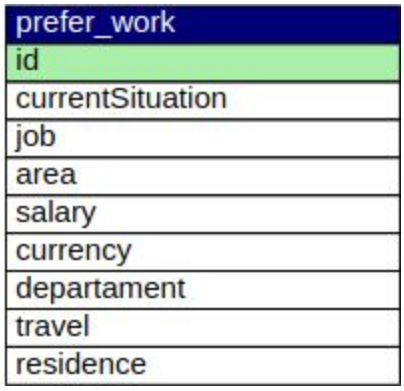
Esta tabla guarda los datos de estudio de cada usuario, por medio de una relación One to Many a User.



Fuente: Autor.

TABLA PREFER WORK

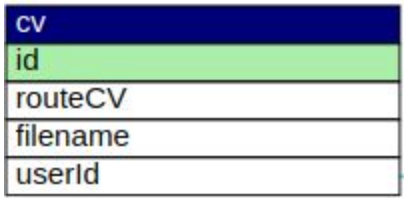
Cada usuario tiene un trabajo que quisiera o no tener, esta tabla guarda estos datos por medio de una relación One To Many a User.



Fuente: Autor.

TABLA CV

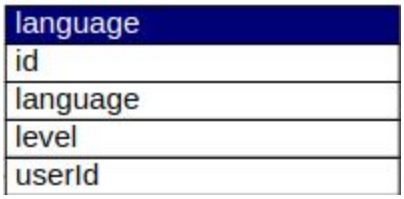
Simplemente es guardar la ruta y nombre de los anexos que tenga un usuario. Es una relación One to Many a User.



Fuente: Autor.

TABLA LANGUAGE

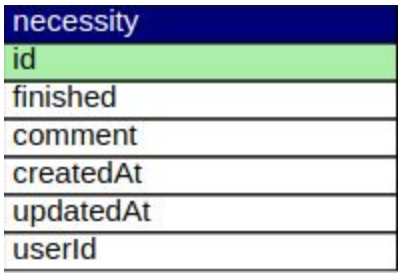
Guardar qué idiomas sabe un usuario, con su nivel y lengua. Es una relación One To Many a User.



Fuente: Autor.

TABLA NECESSITY

Guardar qué necesidades tiene un usuario, aquellas que estén o no finalizadas. Es una relación One To Many a User.



Fuente: Autor.

TABLA WORK

Guardar qué experiencias laborales tiene un usuario. Es una relación One To Many a User.

work
id
company
job
departament
sector
area
goals
startedOn
finishIn
userId

Fuente: Autor.

TABLA BUSINESS

Esta se encarga de guardar los usuarios con empresas en la aplicación, la primera vez que un usuario va a ingresar a Te Vi Colombia, es necesario que ingresa:

- Nombre de compañía (Name)
- Indicativo de teléfono por país (telephoneCountry)
- Teléfono de usuario (telephone)
- Sector de la empresa (sector)
- Correo electrónico (email)
- Contraseña (password)

business
id
routePhoto
routeCover
name
description
address
telephoneCountry
telephone
telephone2Country
telephone2
departament
town
nationality
sector
website
googleMapsLocalization
optionalEmail
email
password
skills
socialnetwork
confirmed
forgotPasswordLocked

Fuente: Autor.

El campo `confirmed` y `forgotPasswordLocked` son simplemente campos para proteger la cuenta al hacer un cambio de contraseña y confirmar que la cuenta es validada.

TABLA EMPLOY

Guardar que emplos requiere una empresa. Es una relación One To Many a Business.

employ
id
position
description
area
skills
minStudy
minExperience
language
travel
residence
country
departament
town
time
contract
minSalary
maxSalary
currency
createdAt
businessId

Fuente: Autor.

TABLA MEMBER

Cuando una empresa quiere añadir quiénes son los dueños de esta, los puede agregar por esta tabla, simplemente es una relación Many To Many a ambas cuentas.

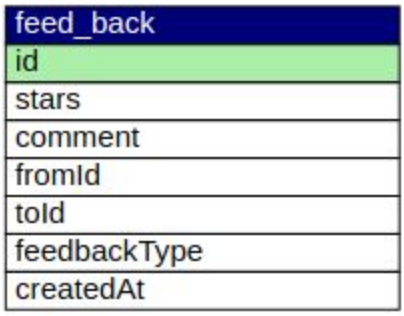
member
businessId
userId

Fuente: Autor.

TABLA FEEDBACK

Esta tabla es especial, ya que cuenta con una polymorphic relation. Esta trata de tener un campo clave el cual ayuda a identificar 2 o más relaciones. En este caso, los usuarios User y Business son dos tipos de cuentas que acepta la aplicación, para saber quien ha recibido feedback se tiene un campo llamado `feedbackType`, en el cual se almacena User / Business como identificadores.

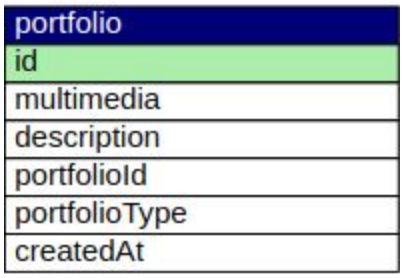
El campo `fromId` y `toId` son para identificar quien ha recibido y dado su comentario y cantidad de estrellas.



Fuente: Autor.

TABLA PORTFOLIO

Al igual que la tabla `Feedback`, esta cuenta con una polymorphic relation. Este campo identificador es `portfolioType` que almacena si el portafolio o publicación es de User o Business.



Fuente: Autor.

El campo `portfolioId` guarda el ID de algún User o Business, para saber quien es el dueño de esa publicación.

REDIS COMO GUARDADO DE SESSIONS.

Cada vez que un usuario entra a la aplicación, se genera una cookie, está cuenta con una sesión para poder diferenciar cada usuario, esta sesión es guardada en Redis.

Esta se puede guardar perfectamente en MySQL a través de una tabla nueva. Esta metodología es muy vieja y a largo plazo una gran mala idea, para evitar problemas y la identificación de usuarios sea eficaz, se usa esta In Memory database. Más adelante se explica su funcionalidad en la aplicación y como la API logra hacer este proceso.

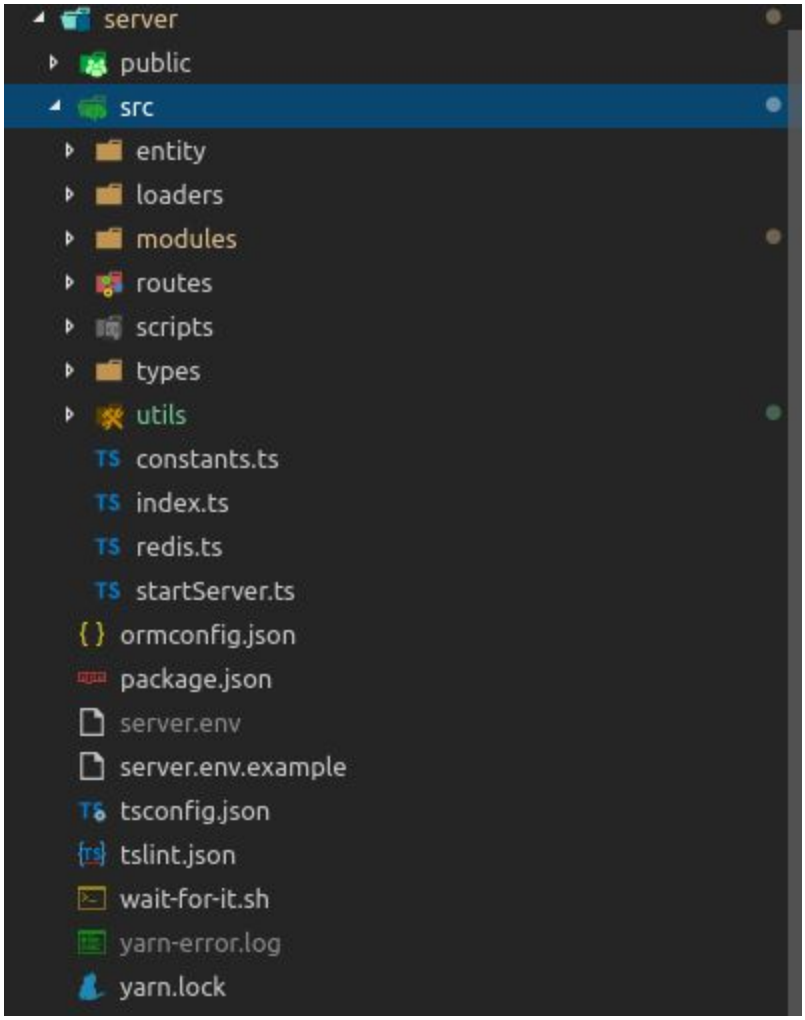
APPLICATION PROGRAMMING INTERFACE (A.P.I)

La A.P.I está basada en una tecnología llamada GraphQL, que es manejado por el runtime engine Node.js, a través del Framework Express.js. Este permite que el cliente consuma los datos que se necesitan, reduciendo el peso del response y ayudando a que no haya un overflow de información innecesaria. Cuenta con un esquema sólido, confiable y bien definido que permite tener un excelente control en la aplicación, así mismo la experiencia de desarrollo es inigualable.

Este servicio, como se mencionó anteriormente, trabaja sobre MySQL, el cual es manejado por el ORM, TypeORM.

La estructura del proyecto se basa en que tenemos una carpeta llamada `src` la cual cuenta con todo el source de la A.P.I, esta se basa en que:

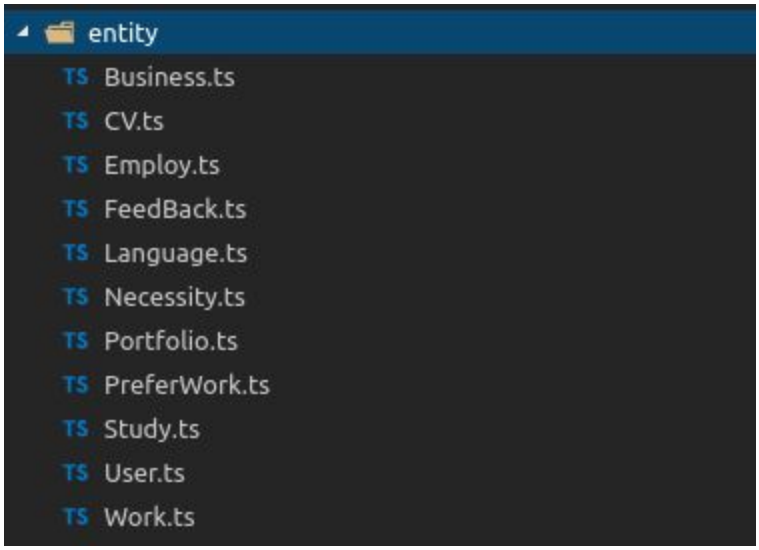
- **Entity:** Es la carpeta en donde se guardan todos los modelos de base de datos que lee TypeORM para interpretar las consultas a MySQL.
- **Loaders:** Es la carpeta en donde se guardan aquellas peticiones que se repiten constantemente en la aplicación, estas peticiones son `cacheadas` sobre `Dataloader`.
- **Modules:** Aquí se guardan los `resolvers` y `schemas` que usa GraphQL internamente para cumplir su funcionalidad.
- **Routes:** Así como en una API Rest típica de Node.js, en la cual cada ruta cuenta con un controlador, esto es lo mismo, son rutas que tienen funcionalidades las cuales no afectan directamente el flujo de trabajo de GraphQL.
- **Scripts:** El proyecto usa TypeScript para tener un tipado de cada schema y ayudar al desarrollador identificar qué parámetros es el que se está usado, para ello esta carpeta cuenta con un script que crea estos `Types` por cada `schema.graphql` y `resolvers.ts` que tiene `Modules`.
- **Types:** En algún lugar tiene que quedar estos `Types` generados, para ello existe esta carpeta, además que se usa para personalizar y generar más `Types` propios.
- **Utils:** Son archivos que se usan globalmente en la aplicación, por lo regular son funciones que se usan en varios `resolvers`.
- **Demás archivos:** Simplemente son instancias de TypeORM o Redis que se tienen a fuerta para organizar su estructura. (Son archivos que hacen una función muy específica, como iniciar el servidor, crear una instancia de Redis, etc).



Hay otros que están a fuera de la carpeta `src`, los cuales son:

- **Ormconfig.json:** Archivo el cual indica la configuración que usa TypeORM.
- **Package.json:** Archivo que indica que dependencias de producción y desarrollo utiliza la aplicación.
- **Server.env:** Es el archivo que se usa para leer las variables de entorno. Este esté inspirado en la configuración de `server.env.example`.

La primera carpeta es `Entity` que simplemente tiene los modelos de la base de datos.



La segunda es `Loaders` el cuenta con las funciones que utiliza `Dataloader` para cachear valores.

```
const batchUserInformation: BatchUserInformation = async ids => {
  const user = await User.find({
    where: { id: In(ids) },
    relations: ["language", "study", "work", "preferwork", "cv", "member"]
  });

  const userMap: { [key: string]: User } = {};
  user.forEach(u => {
    userMap[u.id] = u;
  });

  return ids.map(id => userMap[id]);
};
```

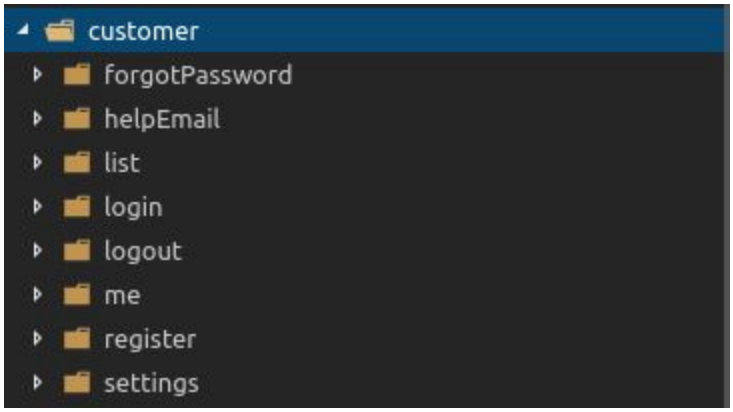
En este caso la función `batchUserInformación`, simplemente hace una `Query` a la entity `User` y estos valores son guardados en memoria, lo mismo pasaría con `batchBusinessInformation`.

Dentro de la carpeta `Modules` está la estructura de folders que tiene un Usuario o Empresa al manejar Te Vi Colombia.

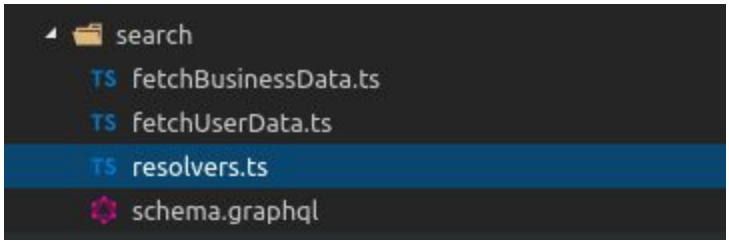
Simplemente son las funcionalidades que trae consigo Business (Empresa) y User (Usuario), cada método que hace en la sección `Mi perfil`, se puede ver ahí. En la carpeta `Shared` (Compartido) son funcionalidades que tanto Business como User tienen, como son que ambos pueden recibir Feedback y tener Portafolios.

En la carpeta `Customer` (Cliente) simplemente se encuentran aquellas funciones que puede hacer cualquier beneficiario que esté usando Te Vi Colombia, como lo podría ser, Registrar una cuenta, Entrar, Salir, Información del perfil actual (me), lista de usuarios y empresas que tienen más estrellas, cambio de correo y contraseña y entre otras funcionalidades que todas las cuentas tienen.

Así mismo, también en esta carpeta se implemente la funcionalidad de Redis, la cual ayuda a que aquellos usuarios que hayan olvidado su contraseña, Redis cree un código especializado para ellos.



En la carpeta `Search` es todo aquello que el buscador de Te Vi Colombia puede hacer.

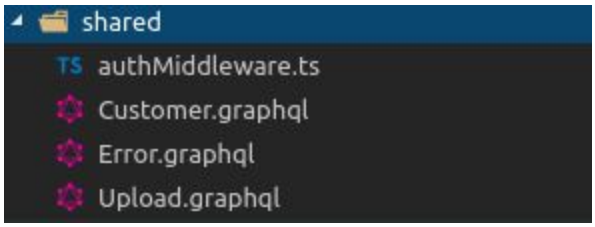


En este caso, aquí se muestra un ejemplo de cómo se puede diferenciar entre los diferentes tipos de usuarios que hay.

```
let response: Array<User | Business> = [];

if (type === "User") {
  response = await FetchUserData(value, user, limit);
}

if (type === "Business") {
  response = await FetchBusinessData(value, business, limit);
}

// Getting ids from `User` or `Business`.
const ids = await response.map(res => res.id);

if (ids.length > 0) {
  // From `ids`, getting the stars.
  const feedbackStars = await getConnection()
    .getRepository(FeedBack)
    .createQueryBuilder("feedback")
    .select("DISTINCT(feedback.toId)", "id")
    .addSelect("SUM(feedback.stars)", "stars")
    .where("feedback.toId IN (:...ids)", { ids })
    .andWhere("feedback.feedbackType = :type", { type })
    .groupBy("feedback.toId")
    .orderBy("stars", "DESC")
    .getRawMany();

  const idsFeedbackStars = feedbackStars.map(item => item.id);

  return mapOrder(response, idsFeedbackStars, "id");
}
```

Las funciones `FetchUserData` y `FetchBusinessData` simplemente devuelven lo que el usuario requiere dependiendo de los parámetros enviados y valor ingresado, por

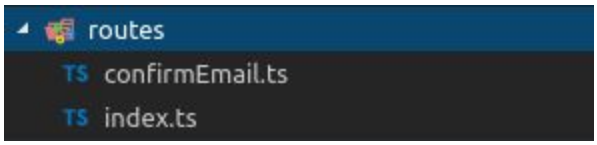
último el método `feedbackStars` obtiene las estrellas de todas aquellas Empresas o Usuarios que la variable `response` arrojo para que `mapOrder` ordene la información como debe de ser.



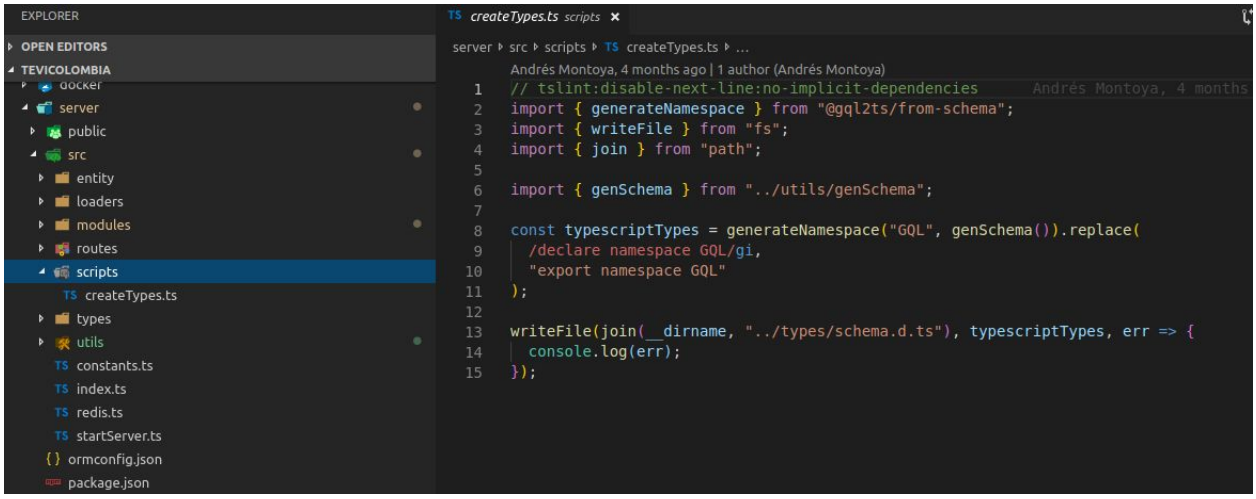
En la carpeta `shared` simplemente son `schemas` globales que se usan típicamente en la A.P.I, como lo puede ser el mostrar los errores de una respuesta, la información de un cliente, que variables necesitan ciertos métodos, y por último el middleware de autenticación.

Y por último la carpeta `Types`, está se cuenta con dos tipos de datos especiales que necesita la aplicación para validar la entrada y salida de datos, que son los `Types` para fechas y números enteros de 64bits.

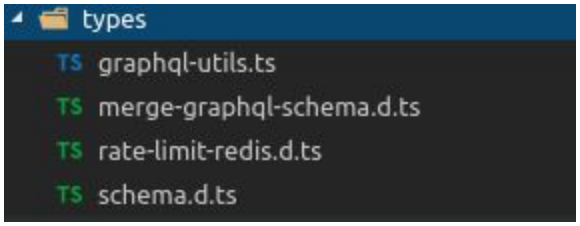
En la carpeta `routes` ubicada en `src` son aquellas peticiones que están afuera de GraphQL y ayudan a que peticiones no frecuentemente utilizadas, como lo es la confirmación de un correo, no afecte el rendimiento de GraphQL.



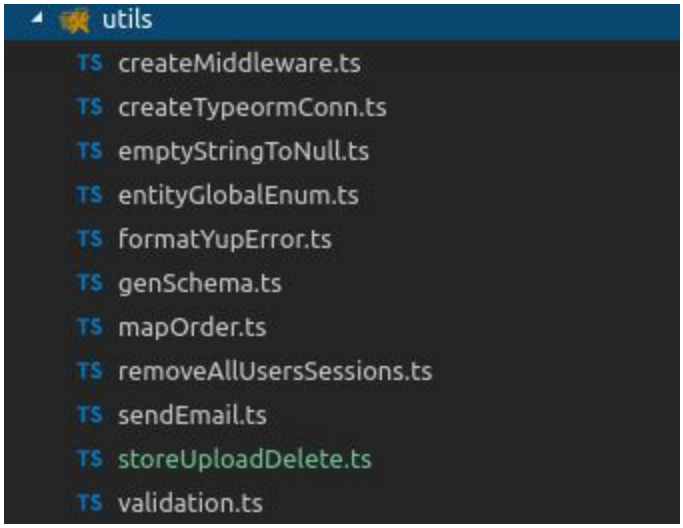
En la carpeta `scripts` simplemente está el archivo que genera los `Types` para la A.P.I.



En la carpeta `Types` están todos los `Types` que necesita TypeScript.



En `Utils` están aquellas funciones que se usan en varias partes de la aplicación, como lo podría ser el ordenamiento de datos, eliminar aquellas sesiones de usuarios no utilizadas, enviar correos, subir o eliminar archivos, organizar las peticiones de error, entre otras funcionalidades.



Aquí solo se explico la estructura de las carpetas de la A.P.I y algunas partes de código, lo demás no se muestra ya que el código de Te Vi Colombia es privado.

SINGLE PAGE APPLICATION

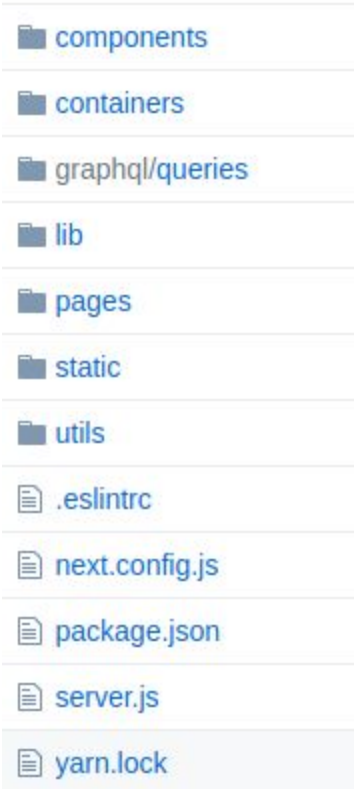
Todo lo que ve el usuario es completamente reactivo, haciendo que la experiencia de usuario sea completamente única, para lograr este cometido se uso la Libreria React.js junto a Next.js.

React.js es una librería escrita en JavaScript desarrollada por Facebook para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuarios. En cambio Next.js es un complemento para esta biblioteca, ya que añade Server Side Rendering (SSR) y SEO, mejorando el uso cotidiano de React, esta herramienta fue hecha por la empresa Zeit, la misma que creo Now.

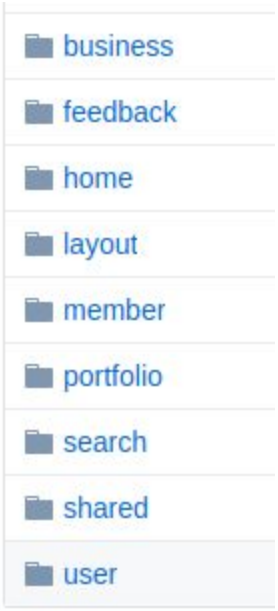
La estructura que sigue este proyecto está en la carpeta `client`, la cual cuenta con las siguientes carpetas:

- **Components:** Como su nombre lo indica, son los componentes que usa la aplicación, estos están divididos de las funcionalidad que cumple una empresa o usuario, así mismo el Layout y componentes compartidos (reutilizables).
- **Containers:** Son aquellos componentes que se utilizan globalmente y variadamente. Con esto se quiere decir que, cada vista que se repita más de una vez en `pages` o en `components`, para reutilizar esta vista, se crea un contenedor.
- **GraphQL:** Son aquellas queries que se utilizan más de una vez en diferentes componentes.
- **Lib:** Los datos que genera Apollo GraphQL tienen que ser devueltos por la vista, de una forma en la que soporte Server Side Rendering y a la vez no genere problemas, lo que se encuentra en esta carpeta son scripts que cumplen esta función, así mismo también son utilizados para la autorización de usuarios.
- **Pages:** Next.js utiliza esta carpeta para identificar qué rutas tendrá la aplicación, y que código tendrá el componente de esta ruta.
- **Static:** Son archivos estáticos que utiliza la aplicación, como archivos CSS, imágenes, y el `manifest.json` que necesita el PWA de Te Vi Colombia.
- **Utils:** Simplemente son los archivos JSON que necesita los input selects, la validación de los diferentes inputs, y un archivo con nombre `normalizeErrors` que ayuda a normalizar los errores que devuelve la A.P.I para que sean compatibles con Formik.

- **Demás archivos:** Dos en especial son necesarios para Next.js, los cuales son `next.config.js` el cual trae la configuración de los paquetes `next-offline` y `@zeit/next-css`, y el archivo `server.js` que tiene las rutas de la aplicación y otras configuración, este fichero es usado comúnmente en modo desarrollo `Development` ya que en producción se usa Now en su versión 2 en el cual en el JSON `now.json` se definen sus rutas, basadas en serverless.



La primera carpeta es Components, esta carpeta trae diferentes carpetas con más componentes que cumplen diferentes funciones.

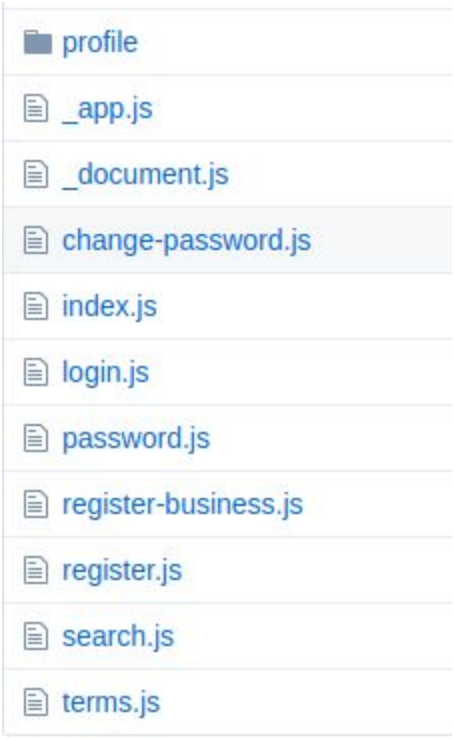


Si ya ha leído el manual de usuario, o ya ha manejado la aplicación Te Vi Colombia y ha usado todas sus funcionalidades, comprender este listado no será difícil, ya que aquellas funcionalidades que se repitan más de una vez entre usuarios y empresas, se encuentran con su propia carpeta, como lo es Feedback, Member, Portfolio. En el caso de Shared son componentes que se comparten entre sus antecesores, un ejemplo podría ser los input que tiene Te Vi Colombia, todos estos son un solo componente que se reutiliza en muchos lados.

Como lo sería el archivo `globalField` que como su nombre indica, son campos, inputs, selects, checkboxes, etc, que se usan en toda la aplicación y su principal libreria que maneja el estado, es Formik.

Los demás archivos, que son Home y Search simplemente son todo lo que se ve en el inicio, el buscador, el formulario de ayuda, la lista de los mejores usuarios o empresas, etc… En el caso de Layout, es el footer, el header, entre otros que se ven en todas las secciones de Te Vi Colombia.

En `Pages` son las rutas que cuenta Te Vi Colombia, con su respectivo estilo y estado:



El archivos `__app.js` se usa para globalizar la configuración de Apollo GraphQL y configurar el Layout. En `__document.js` es simplemente el `<head>` de HTML que toda típica aplicación tiene.

Aquí no se mostró ninguna línea de código, ya que Te Vi Colombia es un proyecto privado.