

**INFORME**  
**PROYECTO MANDE**

**Integrantes**

Juan José Viafara Carabalí, 2040751-3743  
Juan Esteban Mazuera Yunda, 2043008-3743  
Sheilly Daylin Ortega Granobles, 2040051-3743

**Docente**

Jefferson Amado Peña  
Andrés Mauricio Castillo

**Universidad del Valle**  
**Facultad de Ingeniería**  
**Bases de datos**  
**Valle del Cauca - Santiago de Cali**  
**Agosto, 2022**

## Resumen

Se realizó una aplicación web, llamada Mande, cuyo objetivo es encontrar personas dispuestas a realizar alguna labor que los usuarios requieran, y brindar una alternativa para personas que buscan ganar dinero extra realizando dichos servicios.

Para el frontend se utilizó react.js y node.js, para el backend: postgresql y express y para su despliegue se usó Docker.

## Introducción

Mande es un emprendimiento que permite, desde una aplicación web, conseguir personas trabajadoras, expertas y honestas para realizar a domicilio labores del día a día en el hogar. En la página principal de la aplicación, los trabajadores asociados deben registrarse. Este registro comprende la información personal del trabajador, con una foto de perfil obligatoria y una imagen de documento de identidad y su dirección de residencia que se convertirá en una coordenada de GPS; además de la selección de una lista predefinida de labores que puede desempeñar, como, por ejemplo: plomero, cerrajero, profesor de inglés, paseador de perros, etc. Para cada labor debe poner su precio por hora o unidad de labor.

Los usuarios de Mande, quienes solicitan los servicios, también deben registrarse en la aplicación. Para estos usuarios se pide la información personal, junto con su dirección de residencia, que se convertirá en una coordenada de GPS para la aplicación, y un recibo de servicio público que se usará internamente para validar el lugar de residencia. Adicionalmente se solicitará un email y número de celular, que será la llave del usuario en la aplicación, ya que todos los servicios deberán pedirse desde el celular registrado. El usuario también debe registrar un medio de pago (tarjeta crédito o débito), y dado que esta información es privada, deberá

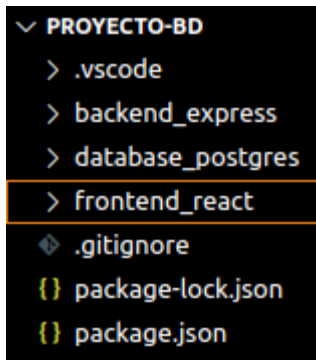
guardarse encriptada en la base de datos. Cuando un usuario necesita un servicio, entra en la aplicación y lo solicita, buscando en la lista de labores ofrecidas. Sólo deben mostrarse las labores para las cuales hay trabajadores inscritos. Con esta selección, se realiza una búsqueda de todos los trabajadores en la zona, y se produce una lista de los trabajadores disponibles, ordenados de acuerdo con el siguiente criterio:

- Número de estrellas obtenidas por el trabajador.
- Distancia del trabajador al lugar requerido.
- Precio por hora o unidad de labor La persona puede agregar una descripción de la labor a realizar.

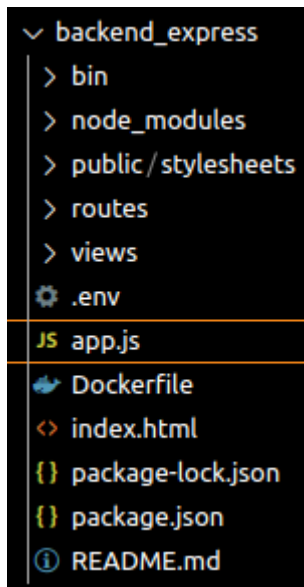
Cuando un trabajador es seleccionado, se le envía una notificación al trabajador con la dirección de la persona que lo necesita y la labor a realizar. Inmediatamente este trabajador se marca como ocupado y continuará así, hasta que él, notifique que ya terminó el trabajo. Solo en ese momento puede aparecer disponible para otros trabajos. La persona que lo contrató debe calificar el servicio (de 1 a 5 estrellas) y realizar el pago al trabajador a través de la aplicación. Se debe guardar un registro de todos los pagos realizados en la aplicación. La calificación realizada por la persona se usará para ranquear al trabajador en futuros trabajos, promediando todas las calificaciones recibidas.

## Discusión

Para llevar a cabo este proyecto se usaron las herramientas postgresql, express, react, node y Docker, obteniendo como resultado un proyecto con la siguiente estructura:



- **backend\_express:** su objetivo es conectar la interfaz (react, node) con la base de datos (postgresql). De esta carpeta hay 2 elementos principales: routes y app.js:



**routes** es primordial para poder conectar la base de datos con la interfaz. Esto permite la interacción entre las dos, como permitir el registro de los nuevos usuarios, el inicio de sesión, guardar las labores, entre otros. También aquí es donde se crea la base de datos para la aplicación.

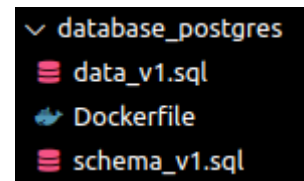
**app.js** se encarga de definir en qué puerto va a estar escuchando el api.

- **database\_postgres:** en esta carpeta se tienen tres archivos:

**data\_v1.sql:** desde ella se pueden insertar datos en la base de datos sin tener que recurrir a la interfaz.

**Dockerfile:** Contiene las sentencias sql de la base de datos.

**schema\_v1.sql:** tiene el esquema de nuestra base de datos, es decir, el conjunto de reglas que rige la base de datos.



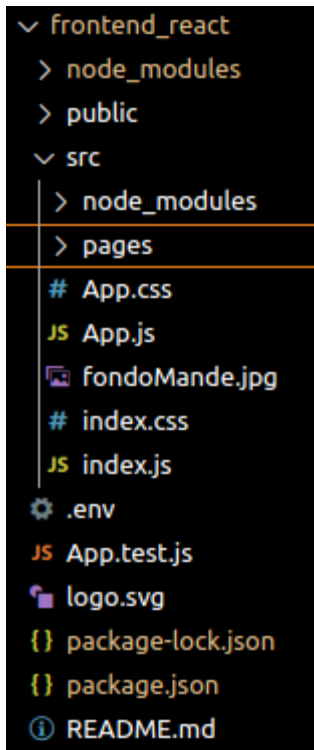
- **frontend\_react:** aquí se desarrolla toda la interfaz de la aplicación, desde ella se crearon todos los formularios de registro, el inicio de sesión, la página donde se ven las labores, botones, en general todo lo relacionado con la interacción entre el usuario y la app.

Los elementos más importantes son pages, app.js e index.js:

**pages** contiene todas las vistas que tiene la app: registro del trabajador y del usuario y el inicio de sesión de los mismos. Hay unos datos que son comunes para el trabajador y para el usuario, por ello al inicio se debe llenar el formulario de registro, luego, dependiendo del rol que se escoja, se llena el formulario de registro correspondiente al rol. Hay dos inicios de sesión: una para el usuario y el otro para el trabajador. Lo siguiente será la interfaz de inicio, registro de las labores y el listado de labores.

**App.js:** permite que la aplicación pueda ser vista desde el navegador y de ahí se crean las rutas que se van a necesitar para poder navegar entre una página y otra.

**index.js:** tiene el “react router dom”, el cual permite hacer las rutas para acceder a cada página de la interfaz.



## Recomendaciones

Para correr la aplicación se necesitan seguir estos pasos:

Levantar el backend y la base de datos:

1. USER\_NAME=juan
2. cd database\_postgres
3. docker build -t \${USER\_NAME}/mande\_db .
4. docker run --name mande\_db -p 5432:5432 -e POSTGRES\_PASSWORD=mysecretpas sword -d \${USER\_NAME}/mande\_db
5. cd ..
6. cd backend\_express
7. docker build -t \${USER\_NAME}/mande\_backend .
8. npm install
9. docker run -it --rm -p 3000:3000 -v \$(pwd):/usr/src/app --link mande\_db:postgres --name mande\_app \${USER\_NAME}/mande\_backend

En otra terminal, en la carpeta raíz del proyecto se levanta el frontend:

```
cd frontend_react && npm start
```

Para realizar consultas de la base de datos:

Debes tener en cuenta que las consultas de usuario y trabajador se hacen con el número de cedula. Tienes las siguientes consultas disponibles:

- Para ver todas las personas de la base de datos: <http://localhost:3000/persona/>
- Para ver todas las direcciones de la base de datos: <http://localhost:3000/persona/direccion>
- Para ver todos los trabajadores: <http://localhost:3000/trabajador/>
- Para ver todos los usuarios: <http://localhost:3000/usuario/>

A cualquiera de esas direcciones se le pone el número de cédula como parámetro en la dirección y devuelve la información de esa persona. Ejemplo:

<http://localhost:3000/persona/11111> devuelve la información de la persona con numero de cedula 11111.

## Conclusión

La unión de todos los elementos previamente mencionados nos da como resultado la aplicación que nos planteamos realizar.

## Referencias

Jonmirchar. Curso React: 1. Introducción - jonmircha [video]. Youtube. (2021). 42:37. [Consultado: 7 de Junio de 2022]. Disponible en: <https://youtu.be/MPLN1ahXgcs>

Jonmirchar. Curso React: 2. Create React App - jonmircha [video]. Youtube. (2021). 41:45. [Consultado: 25 de Junio de 2022]. Disponible en: <https://youtu.be/6rqT8OP-yn0>

Mattweb. III. [BACKEND] Sign in & sign up [video]. Youtube. (2020). 43:59. [Consultado: 08

de Agosto de 2022]. Disponible en:  
<https://www.youtube.com/watch?v=dMiRH0VMa8a8>

Mattweb. IV. [BACKEND] CRUD PostgreSQL, NodeJS & Express [video]. Youtube. (2020). 37:07. [Consultado: 08 de Agosto de 2022]. Disponible en: [https://youtu.be/6K\\_EQ7aiXcw](https://youtu.be/6K_EQ7aiXcw)

NODEJS. Node.js v18.7.0 documentation. [Página web]. [21 de Julio de 2022]. Disponible en: <https://nodejs.org/api/all.html>

POSTGRESQL. PostgreSQL 14.4 Documentation. [Página web]. 19 de Julio de 2022. [Consultado: 30 de Junio de 2022]. Disponible en: <https://www.postgresql.org/docs/current/index.html>

REACTJS. Empezando. [Página web]: Meta Platforms, Inc.. 2022. [Consultado: 11 de Julio de 2022]. Disponible en: <https://es.reactjs.org>

The key elements of a report. University of Otago. [Consultado: 06 de Agosto de 2022]. Disponible en:  
<https://www.otago.ac.nz/accountancyfinance/student-resources/otago01515115.html>