

# Documento de especificación

## Aplicación web

Esta aplicación web consiste en el despliegue de información de contactos, haciendo la consulta a un endpoint (API rest full). La aplicación fue desarrollada con la tecnología de ReactJs usando como lenguaje de programación base JavaScript con la convención ES6 que nos brinda una sintaxis más sencilla y nuevas funcionalidades.

### Estructura

El proyecto fue creado usando la plantilla create-react-app que nos proporciona npx, haciendo que el proyecto contenga las configuraciones y componente básicos para el despliegue. Como parte adicional a dicha plantilla, se crearon las carpetas *src/* *src/* *pages* y *src/components* que será donde el proyecto se irá desarrollando.

La estructura de archivos y directorios es la siguiente:

```

users-web
├── README.md
├── package-lock.json
├── package.json
├── public
│   ├── favicon.ico
│   ├── index.html
│   ├── logo192.png
│   ├── logo512.png
│   ├── manifest.json
│   ├── normalize.css
│   └── robots.txt
├── src
│   ├── componets
│   │   ├── ContactBio
│   │   ├── ContactRow
│   │   ├── Footer
│   │   └── layout.jsx
│   ├── pages
│   │   ├── index.jsx
│   │   ├── notFound.jsx
│   │   └── styles.jsx
│   ├── reportWebVitals.js
│   ├── index.js
│   └── router.js
└── yarn.lock
  
```

Dentro de la estructura, podemos identificar dos directorio principales *public/* y *src/*. En el directorio *public/* podemos encontrar los assets que el navegador usará para personalizar el sitio web en los navegadores, así como *normalize.css* que es una dependencia utilizada para forralear las etiquetas HTML que los navegadores establecen por defecto, para que esto no interfiera en la visualización de la aplicación entre navegadores.

El directorio *src/* es donde se almacenan todos los scripts necesarios para construir la aplicación y dentro de éste directorio se encuentran dos archivos vitales el *index.js* y *router.js* que en conjunto es la forma de desplegar las páginas y el flujo establecido.

*Nota: Identificamos los componentes visuales de la aplicación usando la extensión .jsx y los componentes únicamente funcionales con .js ya que es un tipo de convención pero que no afecta al rendimiento o despliegue del sitio.*

Dentro del directorio *pages/* podemos encontrar los componentes que serán renderizados como pantallas independientes, identificando a la principal como *index.jsx*. Además dentro del directorio *components/* encontraremos aquellos componentes independientes que ayudarán a construir cada pantalla del sitio web, separando cada uno en subdirectorios individuales que agruparán toda la lógica de ellos, así si un mismo componente requieres de diversos scripts y para mantener el bajo acoplamiento del código se deberían de crear archivos independientes pero en el mismo directorio identificando así funcionalidades únicas pero extensas del componente.

## Pre-requisitos

Debemos tener instalado NodeJs así como el gestor de paquetes npm.

Tener al menos un navegador web.

Conexión a internet para la descarga de las dependencias del proyecto.

## Configuración y despliegue

Para la configuración general del proyecto fue necesario instalar la dependencia *react-router-dom* que nos proporciona el mecanismo para el enrutamiento y acceso a las diferentes paginas del sitio. Dicha dependencia fue usada para generar el archivo *src/router.js* que maneja el enrutamiento del sitio.

En la constitución del proyecto, el router se encuentra encapsulado dentro de un componente *Layout* que nos ayudará a configurar en todas la paginas por defecto en este caso un *Footer*, de esta forma no repetiremos código y será un componente mantenible.

La forma de despliegue consiste en los scripts que están preparados en el archivo *package.json*, los cuales pueden ser ejecutados con el comando *npm <comando>* y dependiendo el comando ejecutado sería el resultado deseado.

Una vez que el proyecto a sido descargado y guardado en un directorio local, procederemos a ejecutar los siguientes comando para ejecutarlo y trabajar con él.

```
$ cd users-web
```

```
$ npm install
```

```
$ npm start
```

Con esto tendremos el proyecto corriendo de forma local en el puerto 3000, por lo que podemos visualizarlo en el navegador colocando la siguiente URL: *localhost:3000*

## Aplicación móvil

Esta aplicación móvil consiste en el despliegue de información de contactos, haciendo la consulta a un endpoint (API rest full). La aplicación fue desarrollada con la tecnología de React Native usando como lenguaje de programación base JavaScript con la convención ES6 que nos brinda una sintaxis más sencilla y nuevas funcionalidades.

### Estructura

El proyecto fue creado usando la plantilla *react-native init* que nos proporciona npx, haciendo que el proyecto contenga las configuraciones y componente básicos para el despliegue. Como parte adicional a dicha plantilla, se crearon las carpetas *src/* *src/* *screens* y *src/components* que será donde el proyecto se irá desarrollando.

La estructura de archivos y directorios es la siguiente:

```
users-mobile
├── __tests__
│   └── App-test.js
├── android
├── ios
├── app.json
├── babel.config.js
├── index.js
├── metro.config.js
├── package-lock.json
├── package.json
├── src
│   ├── navigatorRoot.js
│   ├── components
│   │   └── ContactRow
│   └── screens
│       ├── Home
│       ├── InfoProfile
│       └── SplashScreen
└── yarn.lock
```

Dentro de la estructura podemos identificar dos directorios importantes, el de *adroid/* y *ios/* que son los que contienen las configuraciones necesarias para que al compilarse la aplicación

genere los archivos nativos correspondientes a la plataforma. De igual forma tenemos el archivo principal *index.js* que es el que lanzará la plataforma. Por último tenemos el directorio *src/* que contiene todos los archivos que estructuran la aplicación, comenzando con el archivo *src/navigatorRoot.js* que es el responsable de organizar el flujo de la aplicación, gestionando las pantallas y el Stack de navegación.

Dentro de la carpeta *src/screens* se encuentran los directorios correspondientes a cada pantalla que que contienen el archivo que contienen el código de la pantalla así como sus estilos y de ser necesario podría contener scripts para la lógica de procesamiento y de esta forma tener desacoplado el código y hacerlo más mantenible.

## Pre-requisitos

Debemos tener instalado NodeJs así como el gestor de paquetes npm.

Dependiendo de el Sistema Operativo móvil donde se desee probar o para el cual se desee compilar, es necesario tener instalador:

Xcode -> compilar para iOS

Android estudio -> compilar para android

Además si se desea desplegar la aplicación en un simulador, éste debe de descargarse y configurarse.

Conexión a internet para la descarga de las dependencias del proyecto.

## Configuración y despliegue

Para la configuración general del proyecto fue necesario instalar las dependencias :

*react-native-gesture-handler*

*@react-navigation/native*

*@react-navigation/stock*

que nos proporcionan el mecanismo para la navegación y acceso a las diferentes pantallas de la aplicación. Dichas dependencia fueron usadas para generar el archivo *src/navigatorRoot.js* que maneja el estaca de navegación de la aplicación.

La forma de despliegue consiste en los scripts que están preparados en el archivo *package.json*, los cuales pueden ser ejecutados con el comando *npm <comando>* y dependiendo el comando ejecutado sería el resultado deseado.

Una vez que el proyecto a sido descargado y guardado en un directorio local, procederemos a ejecutar los siguientes comando para ejecutarlo y trabajar con él.

*\$ cd users-mobile*

*\$ npm install*

*\$ cd ios && pod install && cd ..*

*\$ npm run ios*            ó            *\$npm run android*