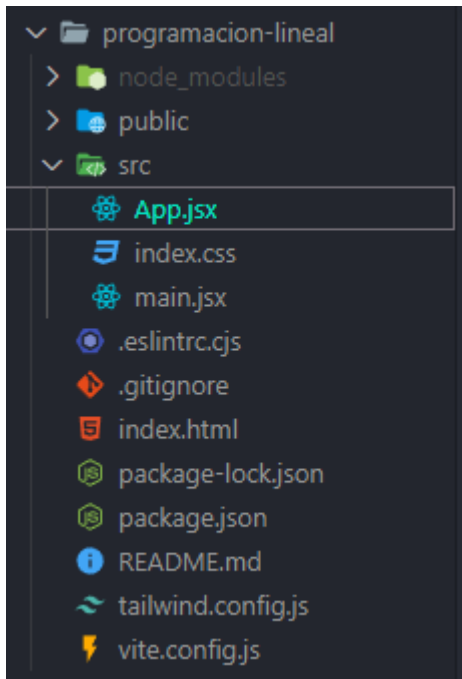


Documentación “Programación Lineal”

El presente documento representa la documentación del simulador de cartas, se eligió javascript como lenguaje de programación pues este fue es el lenguaje el cual todos los integrantes estamos familiarizados

El proyecto consiste en una aplicación de react, la misma cuenta con un único componente el cual contiene todo lo referente al problema, para ejecutar la aplicación se utiliza el comando “npm install” y posteriormente “npm run dev” en la terminal.

En el directorio podemos encontrar la siguiente estructura de archivos:



La parte más fundamental de la aplicación se encuentra en el componente App.jsx.

```

1  import solver from "javascript-lp-solver";
2
3  const App = () => {
4      const model = {};
5
6      // Agrega propiedades al objeto model
7      model.optimize = "price";
8      model.opType = "min";
9      model.constraints = {};
10     model.variables = {};
11
12     // agrega restricciones al objeto model
13     model.constraints.screenSize = { min: 6 };
14     model.constraints.memory = { min: 128 };
15
16     console.log(model.constraints); x2 { screenSize

```

En estas primeras líneas del código encontramos el import de la librería “javascript-lp-solver” la cual fue la escogida principalmente por la sencillez de su uso, luego encontramos también el inicio del componente principal, la creación de un objeto model al cual se le agrega la variable a optimizar en el problema escogido, el tipo de optimización, en este caso es minimización, se crean objetos vacíos para las restricciones y las variables y por último se establecen las restricciones, en este caso son un tamaño de pantalla de mínimo 6 pulgadas y una memoria de mínimo 128GB, también encontramos un console.log el cual muestra las restricciones establecidas.

```

// agrega variables al objeto model
model.variables.product = {
    name: "Samsung Galaxy S22",
    price: 799,
    screenSize: 6.1,
    memory: 128,
};

// agrega un nuevo telefono al objeto model
model.variables.product2 = {
    name: "Google Pixel 6 Pro",
    price: 999,
    screenSize: 6.2,
    memory: 256,
};

// resuelve el modelo
const results = solver.Solve(model);

```

Por último, tenemos la declaración de los 2 productos agregados los cuales son dos teléfonos inteligentes y de los cuales queremos saber cuál sería la mejor compra teniendo en cuenta las restricciones previamente establecidas. Luego de añadidas las variables al objeto model creamos una constante results en la cual se guarda el resultado del problema para posteriormente este ser mostrado en el front.