

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

CENTRO UNIVERSITARIO DE OCCIDENTE

DIVISIÓN DE CIENCIAS DE LA INGENIERÍA



**Proyecto 1
Generador de horario**

Modelacion y simulacion 2

PRESENTADO POR:

Hernandez Pérez, Sergio Rolando

201931555

DOCENTE:

Ing. Pedro Domingo

QUETZALTENANGO – QUETZALTENANGO – GUATEMALA

22-05-2023

Descripción General

El Sistema Generador de Horarios es una aplicación esencial para instituciones educativas y organizaciones que necesitan gestionar eficazmente la asignación de recursos y actividades en el tiempo. Proporciona una solución versátil para crear horarios optimizados y adaptados a las necesidades específicas de la entidad. Este proyecto se enfoca en simplificar y automatizar el proceso de programación de horarios, lo que ahorra tiempo, recursos y reduce errores.

Inicio
Parametros
Resultado

Bienvenido al generador de horario

| | SALON 1 | SALON 2 | SALON 3 | SALON 4 | SALON 5 | SALON 6 | SALON X |
|-------------|---------------------------------|-------------------------------------|-------------------------------------|---------|----------------------------------|---------|---------|
| 15:20-16:10 | Matemáticas I (Prof. A. Gómez) | Historia de México (Prof. B. López) | Química General (Prof. C. Pérez) | | Biología Celular (Prof. D. Ruiz) | | |
| 16:10-17:00 | Matemáticas II (Prof. A. Gómez) | Historia de México (Prof. B. López) | Química General (Prof. C. Pérez) | | Biología Celular (Prof. D. Ruiz) | | |
| 17:00-17:50 | | Matemáticas I (Prof. A. Gómez) | Historia de México (Prof. B. López) | | | | |
| 17:50-18:40 | Matemáticas II (Prof. A. Gómez) | | Química General (Prof. C. Pérez) | | | | |
| 18:40-19:30 | Matemáticas I (Prof. A. Gómez) | | Historia de México (Prof. B. López) | | | | |
| 19:30-20:20 | Matemáticas II (Prof. A. Gómez) | | Química General (Prof. C. Pérez) | | | | |
| 20:20-21:10 | | | Historia de México (Prof. B. López) | | | | |
| 21:10-22:00 | | | Química General (Prof. C. Pérez) | | | | |

Inicio
Parametros
Resultado

Parámetros

Horario inicial:
15:20

Horario final:
22:00

Tiempo de duracion de periodos
50

Prioridad:
Materias

Generar horario

Resultados

Eficiencia: 53.85%

- La materia 'Química 1' con seccion 'A' tiene el mismo horario que un curso del mismo semestre
- La materia 'Social Humanística 2' con seccion 'A' tiene el mismo horario que un curso del mismo semestre
- No hay ningun profesor disponible para Social Humanística 2 Seccion 'A'
- La materia 'Orientación Y Liderazgo 2' con seccion 'A' tiene el mismo horario que un curso del mismo semestre
- La materia 'Química 2' con seccion 'A' tiene el mismo horario que un curso del mismo semestre
- No hay ningun profesor disponible para Química 2 Seccion 'A'
- No hay ningun profesor disponible para Matemática Intermedia 2 Sección 'A'
- La materia 'Análisis Probabilístico' con seccion 'A' tiene el mismo horario que un curso del mismo semestre
- No hay salon disponibles para Inteligencia Artificial
- No hay salon disponibles para Seminario De Sistemas 2
- No hay espacio en el horario para la materia Materia Civil
- No hay espacio en el horario para la materia Materia Industrial

| | SALON 1 | SALON 2 | SALON 3 | SALON 4 | SALON 5 | SALON 6 | SALON X |
|-------------|--|--|--|--|--|--|--|
| 15:20-16:10 | Matemática Intermedia 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN | Química 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN |
| 16:10-17:00 | Orientación Y Liderazgo 2 Sección A ÁREA COMÚN | Química 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN | Matemática Intermedia 2 Sección A ÁREA COMÚN |

El proyecto de generador de horario es un proyecto que busca automatizar la creación de horarios para estudiantes universitarios. El objetivo es generar horarios que cumplan con los requisitos de los estudiantes, como la disponibilidad de materias, profesores y salones.

Este proyecto se divide en dos partes principales:

La primera parte se encarga de recopilar la información necesaria para generar el horario. Esta información incluye:

- * Las materias que el estudiante debe cursar.
- * Los profesores que están disponibles para impartir cada materia.
- * Los salones que están disponibles para cada materia.

La segunda parte utiliza la información recopilada para generar el horario. El algoritmo de generación de horarios toma en cuenta los siguientes factores:

La disponibilidad de materias.

El algoritmo debe asegurarse de que el estudiante pueda cursar todas las materias que necesita.

La disponibilidad de profesores.

El algoritmo debe asegurarse de que los profesores estén disponibles para impartir las materias que el estudiante necesita.

La disponibilidad de salones.

El algoritmo debe asegurarse de que los salones estén disponibles para las materias que el estudiante necesita.

Además de estos factores, el algoritmo de generación de horarios también puede tomar en cuenta otros factores, como:

Los requisitos de la carrera.

Algunas carreras pueden tener requisitos específicos para los horarios de los estudiantes. Por ejemplo, una carrera de ingeniería puede requerir que los estudiantes tomen un conjunto específico de materias en un orden específico.

Las preferencias del estudiante.

El algoritmo puede permitir al estudiante indicar sus preferencias para el horario, como el horario de clases o el salón en el que prefiere asistir a las clases.

El proyecto de generador de horario tiene el potencial de ser una herramienta muy útil para los estudiantes universitarios. Un horario bien diseñado puede ayudar a los estudiantes a tener éxito en sus estudios.

Beneficios potenciales

El proyecto de generador de horario tiene el potencial de ofrecer los siguientes beneficios a los estudiantes universitarios:

Ahorro de tiempo y esfuerzo. El proyecto puede ayudar a los estudiantes a ahorrar tiempo y esfuerzo al generar automáticamente el horario.

Mejora de la eficiencia El proyecto puede ayudar a los estudiantes a ser más eficientes en sus estudios al asignarles las materias y los profesores que mejor se adapten a sus necesidades.

Aumento de la satisfacción. El proyecto puede ayudar a los estudiantes a estar más satisfechos con sus horarios al tener en cuenta sus preferencias.

Tecnologías utilizadas

El proyecto de generador de horario utiliza las siguientes tecnologías:

API. El proyecto utiliza una API para recopilar la información necesaria para generar el horario. La API proporciona información sobre las materias, los profesores y los salones disponibles.

React. El proyecto utiliza React para crear una interfaz de usuario para que los estudiantes puedan indicar sus preferencias para el horario.

Python. El proyecto utiliza Python para implementar el algoritmo de generación de horarios.

Parámetros

Horario inicial:

15:20

Horario final:

22:00

Tiempo de duracion de periodos

50

Prioridad:

Materias

Generar horario

Horario generado con éxito!

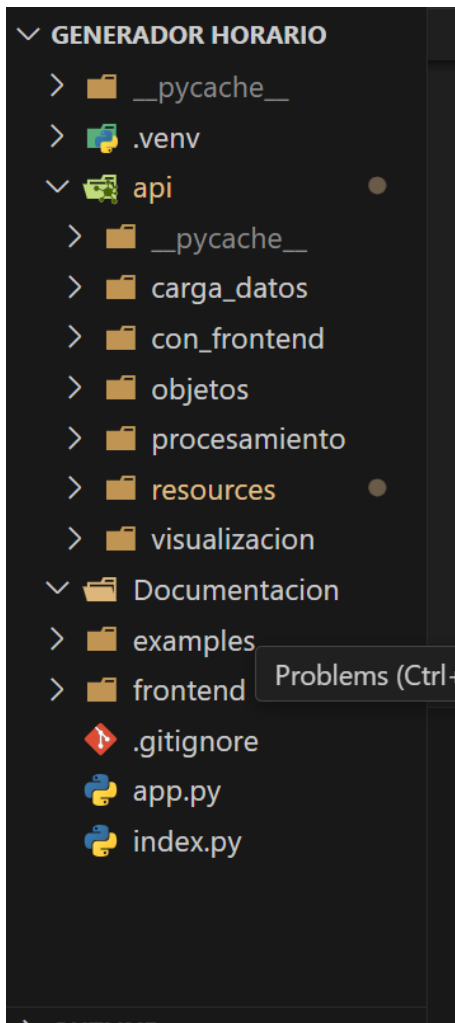
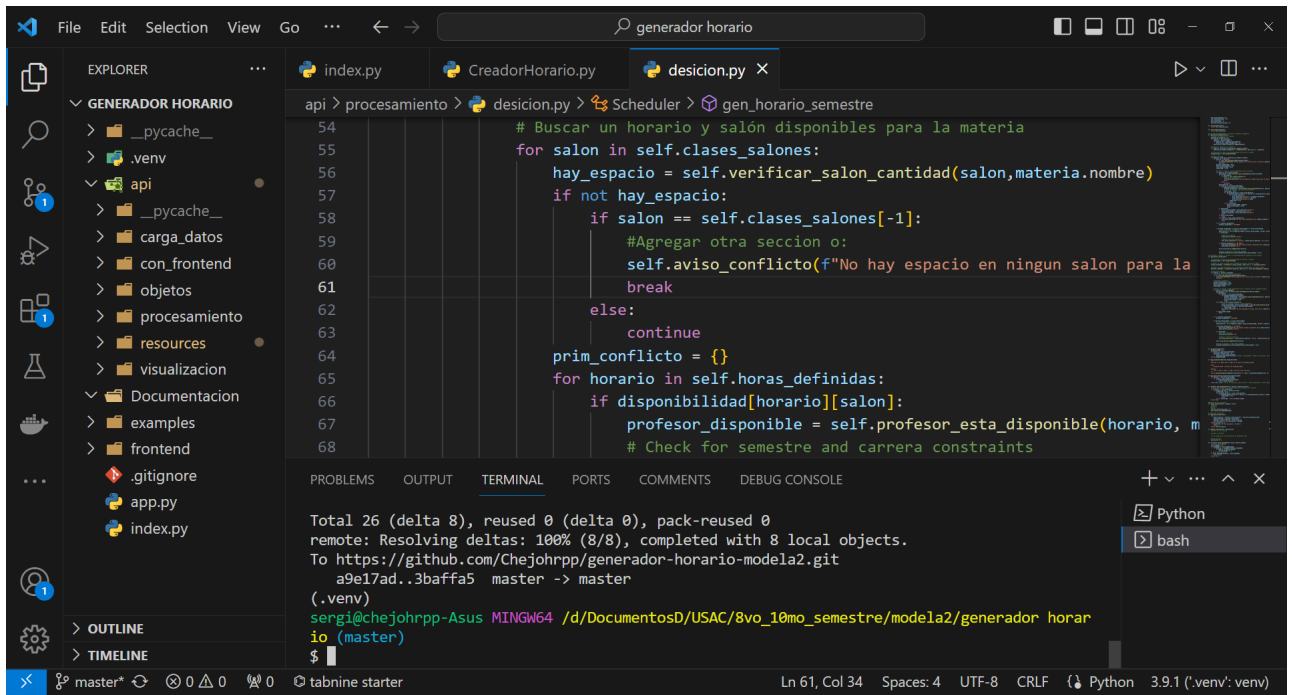
Resultados

Eficiencia: 94.74%

No hay ningun profesor disponible para Orientación Y Liderazgo Seccion 'A'

| | SALON 1 | SALON 2 | SALON 3 | SALON 4 | SALON 5 | SALON 6 | SALON X |
|-------------|---|--|---|---|---------|---|---------|
| 15:20-16:10 | Matemática I Sección A Profesor A Materia A | Matemática I Sección B Profesor B Materia A | Orientación Y Liderazgo Sección A Profesor A Materia A | Matemática I Sección A Profesor A Materia A | | Matemática I Sección B Profesor B Materia A | |
| 16:10-17:00 | Orientación Y Liderazgo Sección A Profesor A Materia A | | Matemática I Sección B Profesor B Materia A | Orientación Y Liderazgo Sección B Profesor B Materia A | | Orientación Y Liderazgo Sección B Profesor B Materia A | |
| 17:00-17:50 | Matemática I Sección A Profesor A Materia A | | Matemática I Sección B Profesor B Materia A | | | | |
| 17:50-18:40 | Matemática I Sección A Profesor A Materia A | | Matemática I Sección B Profesor B Materia A | | | | |
| 18:40-19:30 | Matemática I Sección A Profesor A Materia A | | Matemática I Sección B Profesor B Materia A | | | | |
| 19:30-20:20 | | | Matemática I Sección B Profesor B Materia A | | | | |
| 20:20-21:10 | | | Matemática I Sección B Profesor B Materia A | | | | |
| 21:10-22:00 | | | Matemática I Sección B Profesor B Materia A | | | | |

Desarrollo



La estructura del código del generador de horario parece estar organizada en clases y métodos que se encargan de diferentes aspectos del proceso de generación de horarios. Aquí se presenta una descripción general de las partes principales del código:

Clases Principales:

GeneradorHorario:

Esta parece ser la clase principal que coordina todo el proceso de generación de horarios. Contiene métodos para generar horarios en función de diferentes criterios.

Métodos Principales:

gen_horario_semestre():

Este método genera horarios en función de la prioridad del semestre de las materias. Agrupa las materias por semestre, ordena las materias dentro de cada semestre y busca horarios disponibles que cumplan con las restricciones, como la disponibilidad de profesores y la no repetición de materias del mismo semestre en el mismo horario.

gen_horario_salon():

Este método parece ser parte de la lógica de generación de horarios basada en la prioridad de los salones. No se muestra completamente en el código proporcionado.

aviso_conflicto():

Este método parece ser responsable de emitir avisos cuando se detectan conflictos en la generación de horarios.

`profesor_esta_disponible()`:

Este método verifica si un profesor está disponible en un horario determinado.

Inicialización de Datos:

El código muestra la inicialización de diversas listas y datos que se utilizan en la generación de horarios, como `clases_salones`, `profesores`, `materias`, `asignaciones`, `lista_horarios`, `horas_definidas`, entre otros.

Recopilación de Datos:

No se muestra en el código proporcionado, pero se menciona que hay una etapa de recopilación de datos. Esto podría incluir la obtención de información sobre materias, profesores y salones a través de una API.

Lógica de Generación de Horarios:

Los métodos `gen_horario_semestre()` y `gen_horario_salon()` contienen la lógica principal para generar los horarios. Estos métodos toman en cuenta factores como la disponibilidad de recursos (materias, profesores, salones), las restricciones de semestre y carrera, y la prioridad de salones.

Manejo de Conflictos y Avisos:

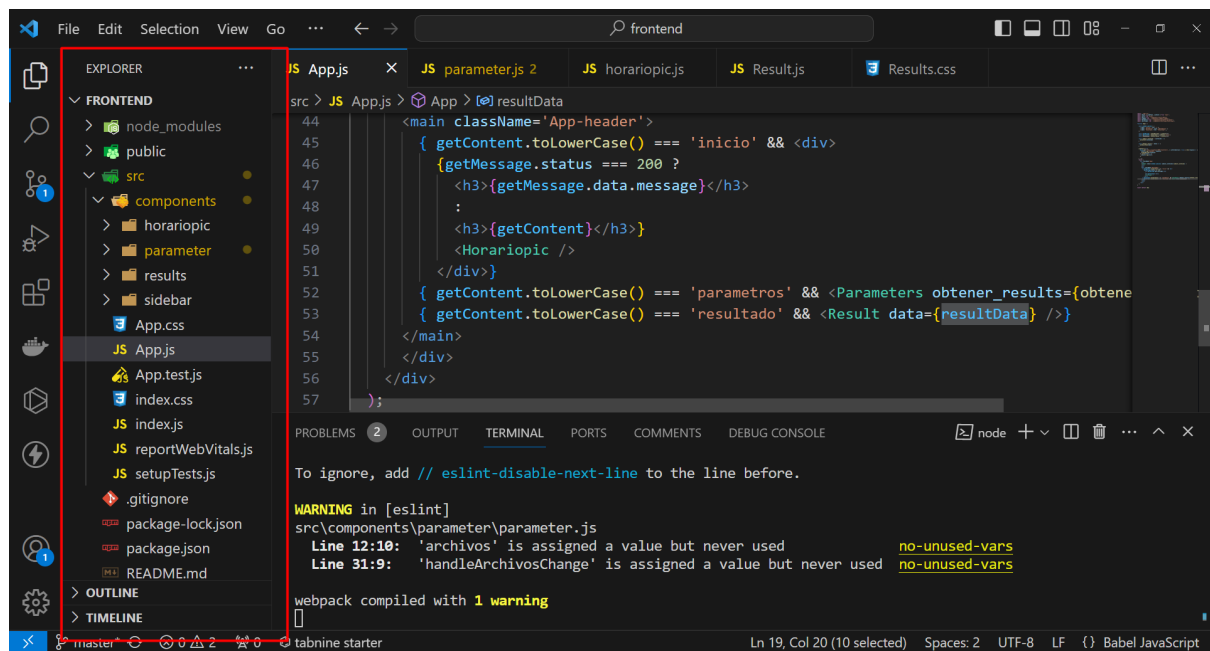
El código parece tener una función para manejar conflictos y emitir avisos cuando se detectan problemas en la generación de horarios.

Optimización de Eficiencia:

Se menciona que el código calcula la eficiencia de los horarios generados en función de la cantidad de conflictos y otros factores. Esta eficiencia se utiliza para ajustar los horarios.

Otras Funciones:

Hay otras funciones y métodos que no se muestran completamente en el código proporcionado, como `verificar_salon_cantidad()`, `get_salon_horario_vacio()`, `cant_asignados()`, entre otros.




El código del frontend del proyecto de generador de horario está organizado en los siguientes archivos:

App.js es el archivo principal del frontend. Es el punto de entrada de la aplicación y contiene la estructura básica de la interfaz de usuario.

Components es un directorio que contiene los componentes de la interfaz de usuario. Cada componente es un archivo JavaScript que representa una parte de la interfaz de usuario, como un formulario, un botón o una lista.


Styles es un directorio que contiene los estilos de la interfaz de usuario. Los estilos se definen utilizando la sintaxis CSS.


▼ FRONTEND


>  node_modules


>  public


▼  src


▼  components

>  horariopic


>  parameter


>  results

>  sidebar

 App.css

JS App.js


 App.test.js


 index.css


JS index.js


JS reportWebVitals.js

JS setupTests.js

 .gitignore

 package-lock.json

 package.json

 README.md

> OUTLINE

> TIMELINE

src > JS

44

45

46

47

48

49

50

51

52

53

54

55

56

57

PROBLE

To ign

WARNI

src\co

Line

Line

webpac



App.js

El archivo App.js es el punto de entrada de la aplicación. Es el primer archivo que se carga cuando se inicia la aplicación. Este archivo contiene la estructura básica de la interfaz de usuario, que consiste en un formulario para que los estudiantes indiquen sus preferencias para el horario.

```
import './App.css';
import React, { useEffect, useState } from 'react';
import axios from 'axios'
import Result from './components/results/Result'
import Sidebar from './components/sidebar/sidebar';
import Parameters from './components/parameter/parameter';
import Horariopic from './components/horariopic/horariopic';

function App() {

  const sidebar_options = [
    { label: "Inicio", href: "/" },
    { label: "Parametros", href: "#parametros" },
    { label: "Resultado", href: "#resultado" },
  ];

  const [getMessage, setGetMessage] = useState({})
  const [getContent, setContent] = useState('inicio')
  const [resultData, setResultData] = useState({})

  const cambiar_contenido = (contenido) => {
    setContent(contenido)
  }

  const obtener_results = (data) => {
    setResultData(data)
  }

  useEffect(()=>{
    axios.get('http://localhost:5000/flask/hello', { withCredentials:
true }).then(response => {
      console.log("SUCCESS", response)
      setGetMessage(response)
    })
  })
}
```

```

    }).catch(error => {
      console.log(error)
    })

  }, [])
  return (
    <div className="App">
      <div>
        <Sidebar items={sidebar_options}
cambiar_contenido={cambiar_contenido} />
      </div>
      <div>
        <main className='App-header'>
          { getContent.toLowerCase() === 'inicio' && <div>
            {getMessage.status === 200 ?
              <h3>{getMessage.data.message}</h3>
              :
              <h3>{getContent}</h3>}
            <Horariopic />
          </div>}
          { getContent.toLowerCase() === 'parametros' && <Parameters
obtener_results={obtener_results} />}
          { getContent.toLowerCase() === 'resultado' && <Result
data={resultData} />}
        </main>
      </div>
    </div>
  );
}

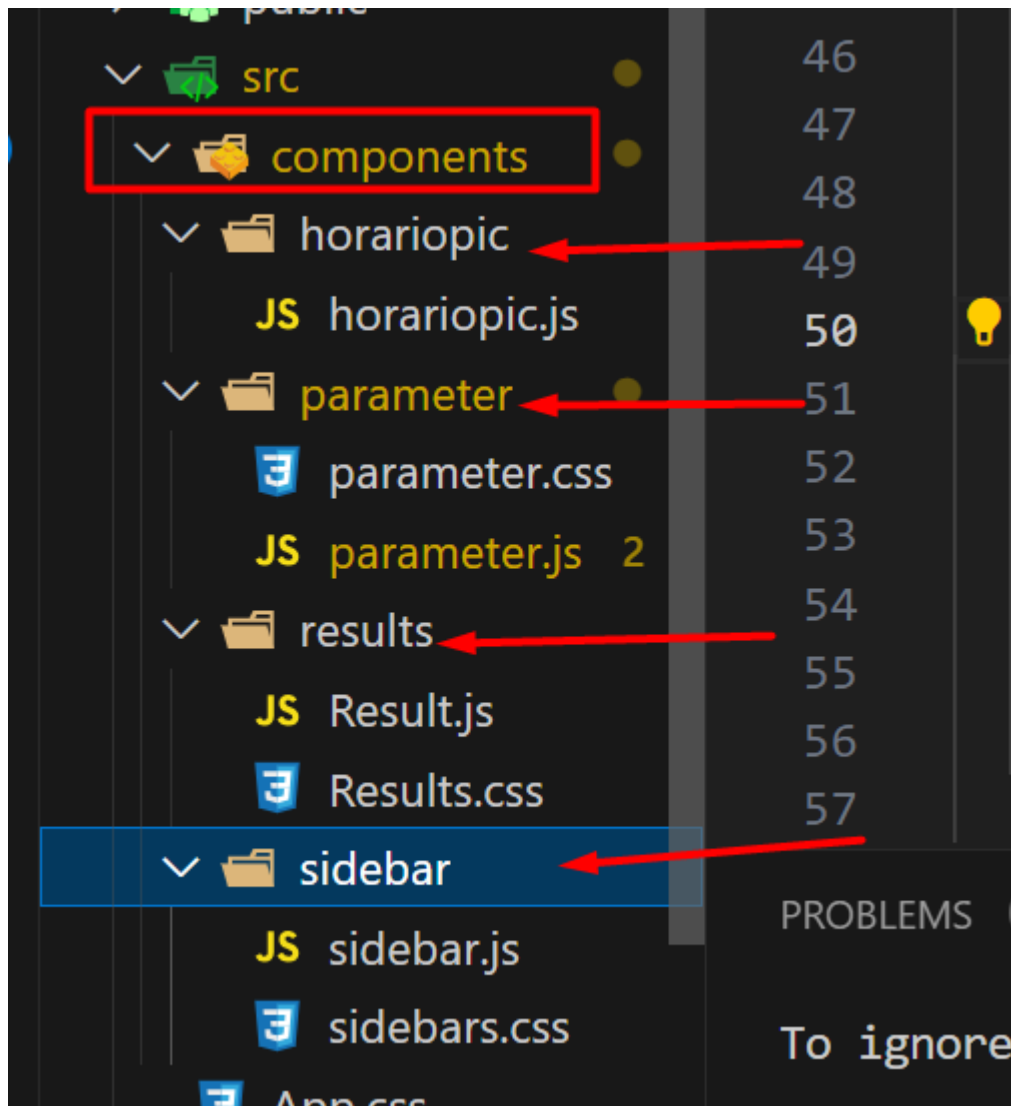
export default App;

```

Components

El directorio Components contiene los componentes de la interfaz de usuario. Cada componente es un archivo JavaScript que representa una parte de la interfaz de usuario, como un formulario, un botón o una lista.

El componente Form es el componente principal de la interfaz de usuario. Este componente contiene los campos para que los estudiantes indiquen sus preferencias para el horario.



El proyecto de generador de horario tiene el potencial de ser una herramienta muy útil para los estudiantes universitarios.