

Viade - Rutas

Arquitectura del Software - EII

Viade_es5b

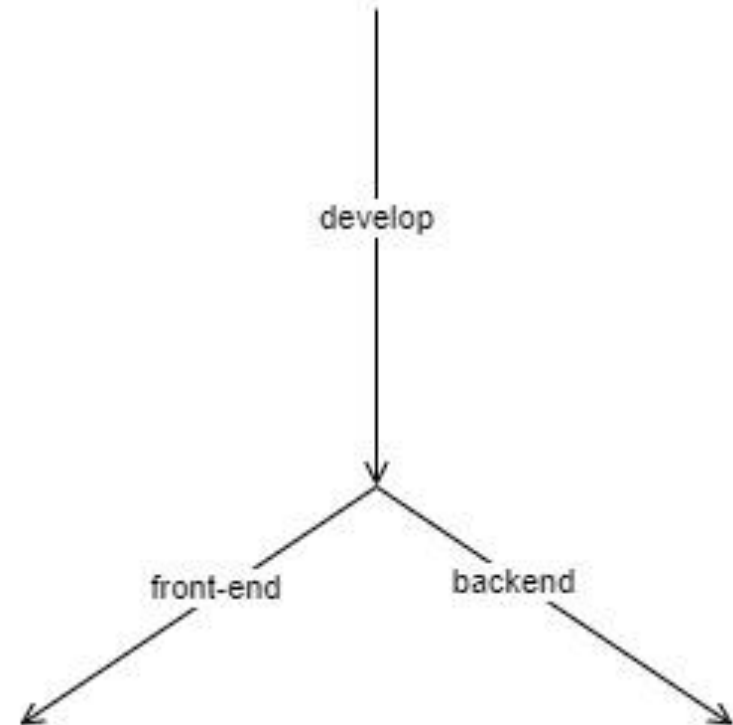
*Alejandro Flórez Muñoz
Diego Marcos Pérez
Pedro José Fernández Franco
Lucía Prado García
Adnane Mouloud Larbi
Alejandro Iglesias Calvo
Francisco Barriocanal García*

Tabla de Contenidos

- 1. Gestión del Proyecto
- 2. Arquitectura (Visión global)
 - FrontEnd
 - BackEnd
- 3. Testing
 - TDD
 - BDD
- 4. Entornos y tecnologías auxiliares
- 5. *Demo* de la aplicación.

Gestión del proyecto - División del trabajo

- Dos ramas principales: ***front-end, backend.***
- Una rama por funcionalidad.
- Ramas auxiliares: *bugs, errores, refactoring...*



Gestión del proyecto - Issues

- Uso de las *issues* para comunicar *bugs*, errores...

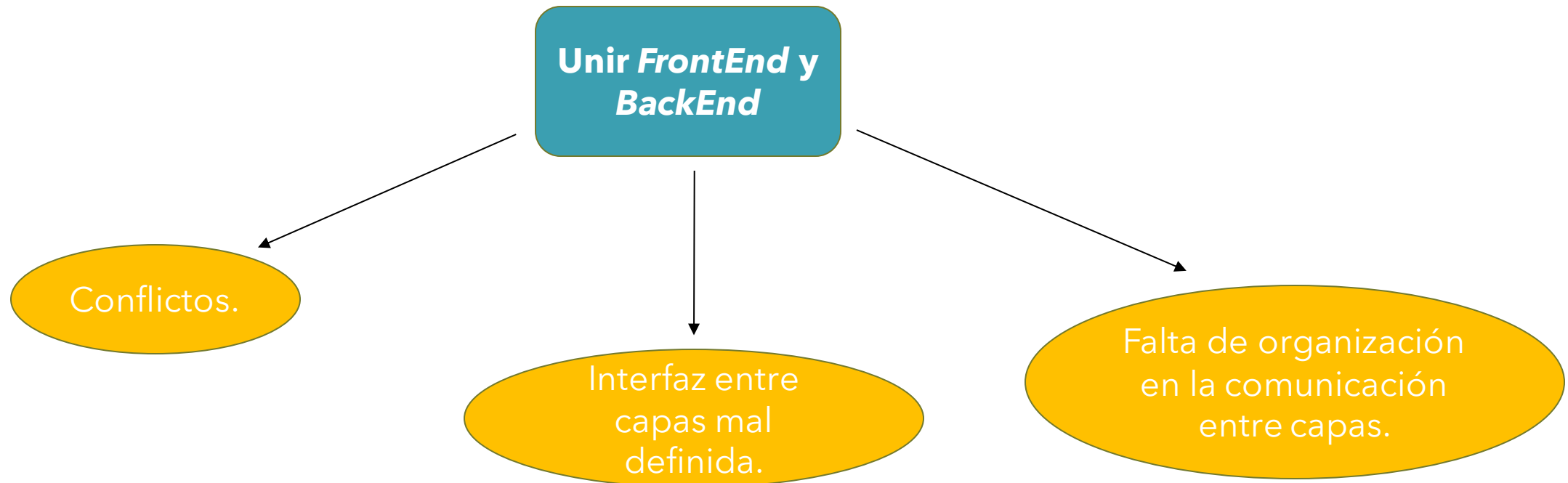
| | | |
|--------------------------|--|--|
| <input type="checkbox"/> | 3 Open ✓ 9 Closed | Author ▾ |
| <input type="checkbox"/> | Report contributions | #34 opened 7 days ago by labra |
| <input type="checkbox"/> | La documentación ha dejado de funcionar bug | #33 opened 12 days ago by pedrojofernandezfranco |
| <input type="checkbox"/> | Bug al eliminar una ruta | #32 opened 21 days ago by alex-florez |

Parche temporal

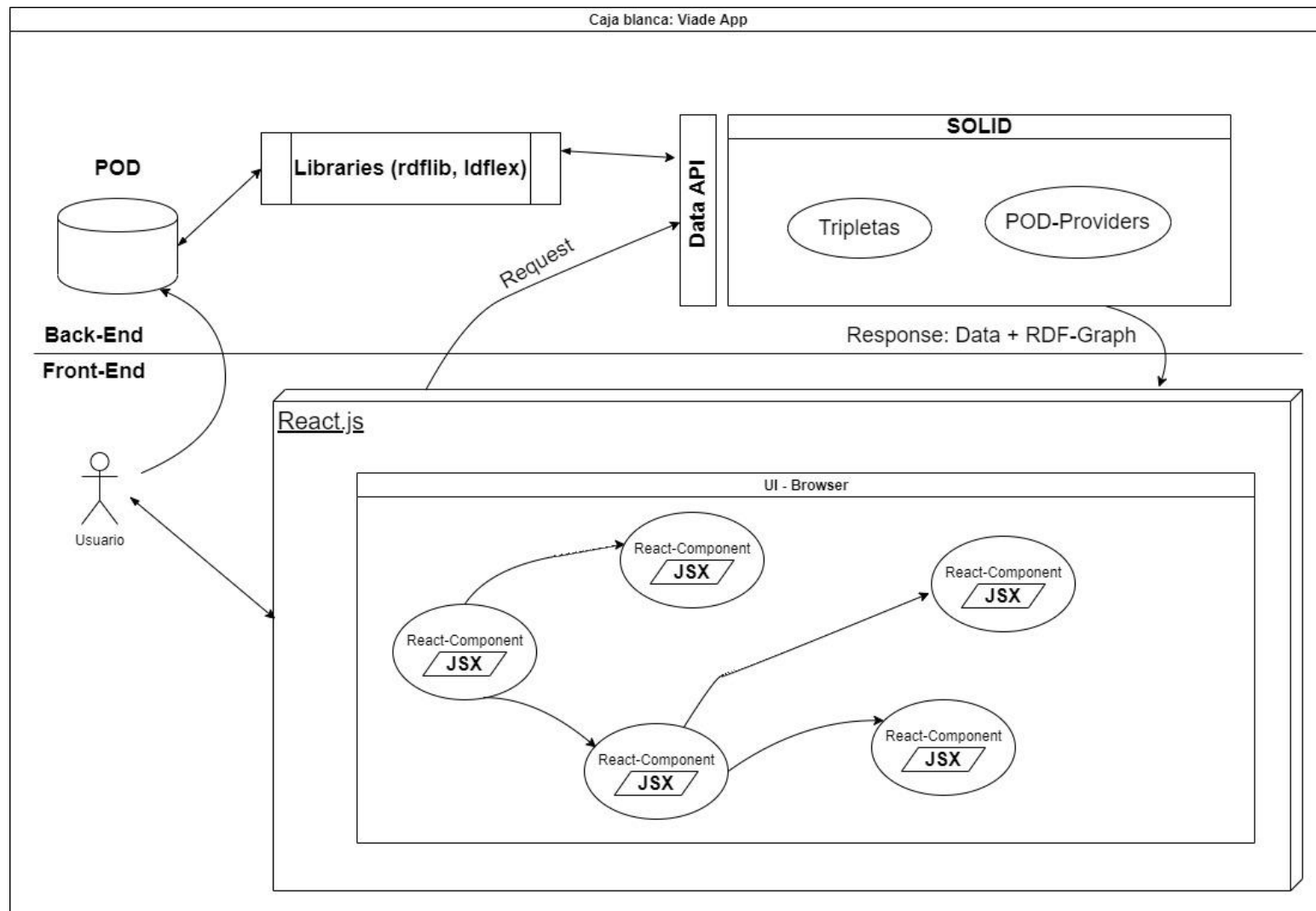
| | | | | | | | |
|--------------------------|---|--|---------------|------------------|--------------|------------|--------|
| <input type="checkbox"/> | 3 Open ✓ 9 Closed | Author ▾ | Label ▾ | Projects ▾ | Milestones ▾ | Assignee ▾ | Sort ▾ |
| <input type="checkbox"/> | Revisión prototipo | #31 by cidinene was closed 1 hour ago | | | | | 1 |
| <input type="checkbox"/> | Unificar codificación para mejorar la calidad del código | #28 by UO247346 was closed 1 hour ago | documentation | good first issue | | | |
| <input type="checkbox"/> | Al eliminar una ruta sigue en la lista de rutas compartidas | #27 by pedrojofernandezfranco was closed 28 days ago | | | | | 2 |
| <input type="checkbox"/> | Corregir Agregar amigo | #26 by alex-florez was closed on 27 Mar | | | | | 1 |
| <input type="checkbox"/> | Corregir UI EliminarRuta | #22 by alex-florez was closed on 29 Mar | | | | | 1 |
| <input type="checkbox"/> | Revisión prototipo | #21 by cidinene was closed 27 days ago | | | | | |
| <input type="checkbox"/> | Corregir la url para acceder al Server de PODs | #20 by UO247346 was closed on 27 Feb | | | | | 1 |
| <input type="checkbox"/> | Revisión Arquitectura | #19 by cidinene was closed 27 days ago | | | | | 2 |
| <input type="checkbox"/> | Ejecutar npm run docs para añadir el technical diagram | #9 by UO245503 was closed on 19 Feb | | | | | |

Gestión del proyecto - Dificultades

- Dificultades durante la primera fase del desarrollo.
- Dos equipos separados: *frontend*, *backend*. **Mayor productividad, menor comunicación.**



Arquitectura de la aplicación



FrontEnd



FrontEnd - Implementación

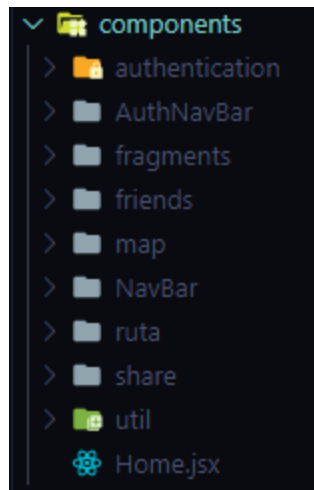
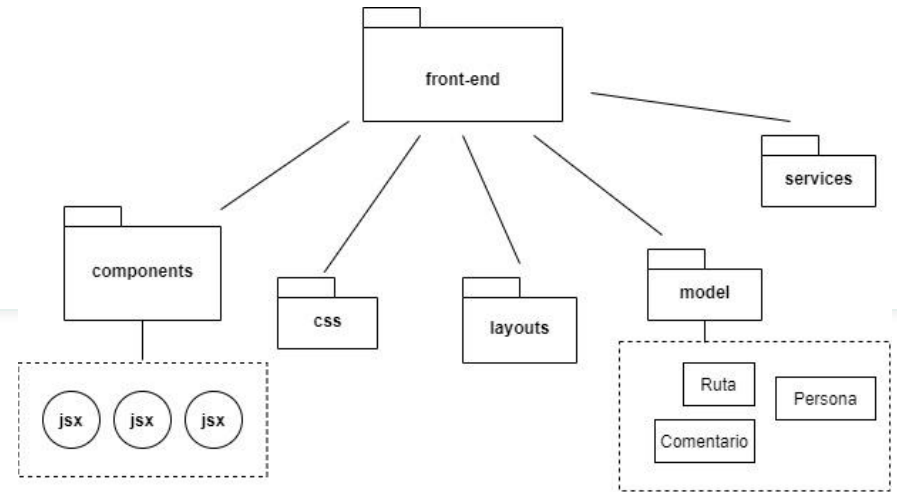
- No se ha utilizado ninguna plantilla, implementación desde cero utilizando la biblioteca *React*.
- Componentes *jsx* para modelar la UI.
- *Lifting State Up*.
- Uso de *React Refs* para identificar a los componentes.
- Funciones predefinidas de la biblioteca, como ***componentDidMount()*** o ***componentDidUpdate()***, para inicializar los componentes o recuperar datos del *backend*.
- Muchos componentes reciben en sus ***props*** las referencias a las funciones que necesitan.

FrontEnd - Vistas

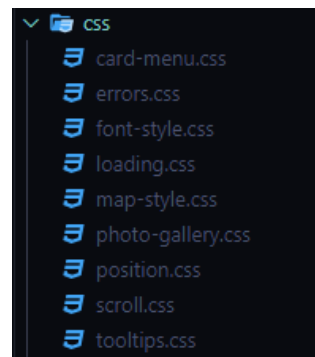
- **Home**
- **Ver rutas:** listado de todas las rutas almacenadas en el POD del usuario.
- **Submenú para crear rutas**
- **Crear ruta con formulario:** añadir una ruta introduciendo las coordenadas mediante un formulario.
- **Crear una ruta con mapa:** muestra un mapa *clickable* para ir dibujando la ruta sobre el mismo.
- **Añadir y listar amigos:** añadir un amigo mediante su *WebID* y lista de amigos actuales.
- **Panel para compartir ruta:** escoger varios amigos con los que compartir la ruta seleccionada.
- **Ver rutas compartidas conmigo:** información de las rutas que me han compartido, incluyendo el autor de las mismas.

FrontEnd - Arquitectura

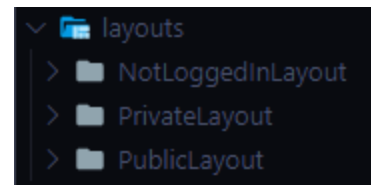
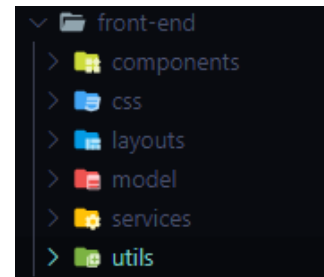
- Estructura de paquetes: paquete **front-end**.



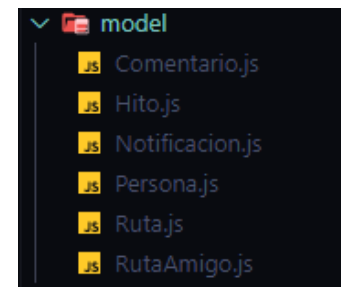
Componentes jsx organizados por funcionalidad.



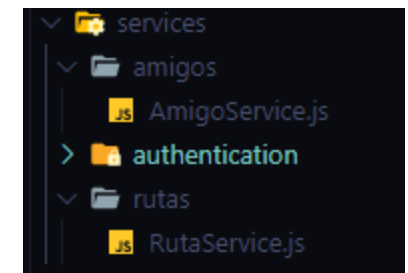
Hojas de estilo CSS.



Layouts



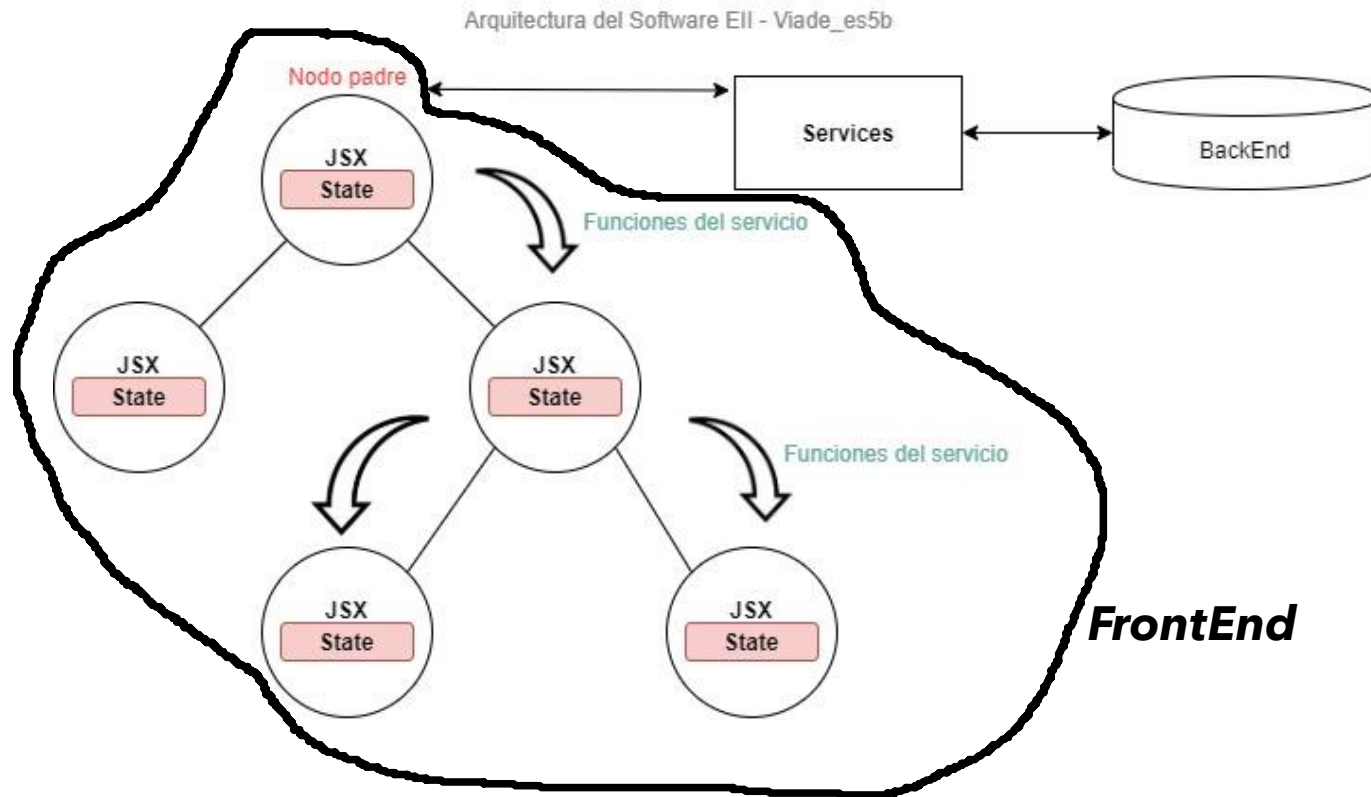
Entidades del dominio de alto nivel.



Capa de servicios.

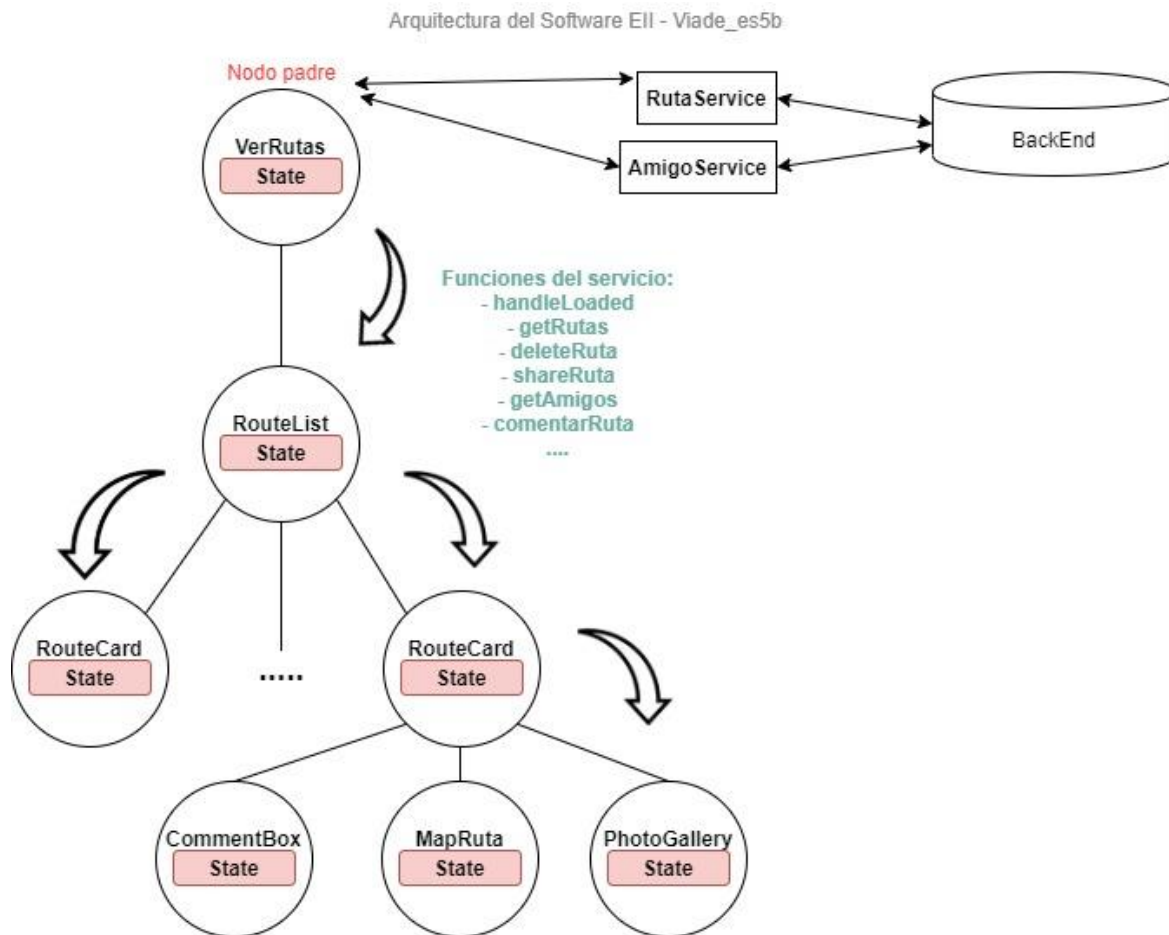
FrontEnd - Arquitectura

- Diagrama de componentes `jsx` de la aplicación.



- Jerarquía **padre-hijos**.
- Los padres instancian las clases de la capa de servicios.
- Los hijos reciben las referencias a las funciones de sus padres, solo de **aquellas que necesitan**.

FrontEnd - Ejemplo de arquitectura



- Listado de rutas del usuario.
- El nodo padre **VerRutas** instancia los servicios **RutaService** y **AmigoService**.
- Los hijos reciben de sus padres **solo** aquellas funciones que necesiten, no se introduce acoplamiento.
- Componentes más fáciles de *testear* (*mock functions*).

FrontEnd - Tecnologías

- Módulos y tecnologías empleadas:
 - **CSS / React Bootstrap.**
 - **React framework:** componentes *jsx* para modelar UI.
 - **React Leaflet / Leaflet:** representar rutas en un mapa, utilidades, marcadores y polilíneas.
 - **React-router-dom:** navegación mediante elementos del menú, uso de *Router* y *Switch*.
 - **React-grid-gallery:** galería para almacenar fotos de la ruta y mostrarlas en un carrusel.
 - **React-notifications:** componentes que modelan notificaciones de compartir rutas, comentarios, fotos...
 - **Solid-react-components:** múltiples usos, como obtener información del *profile* del usuario (nombre, *WebID*...).
LoggedIn y **LoggedOut** para vistas privadas o públicas.
 - **React-spinners** y **react-loading-overlay:** modelan iconos y pantallas de carga para mejorar la experiencia del usuario en operaciones cuyo tiempo de respuesta sea notable.
 - **API File:** para la carga de imágenes desde el cliente.

FrontEnd – Problemas y dificultades

- **Llamadas asíncronas al *BackEnd*:**

- Llamar a funciones de los servicios implica que se debe esperar hasta que el *BackEnd* recupere los datos del *POD* o los modifique.
- Problemas a la hora de renderizar componentes de *React* cuyo estado depende de datos del *POD*.

```
async componentDidMount() {  
  let rutas = await this.props.getRutas();  
  let permisosValidos = await this.props.permisosValidos;  
  console.log("permisos validos :" + permisosValidos);  
  this.setState({  
    rutas: rutas,  
    permisosValidos: permisosValidos,  
    emptyList: rutas.length === 0,  
  });  
  this.props.handleLoaded(); // Indicamos al padre que ya  
}
```

Solución: cláusula *async/await*. Se obliga a esperar a que lleguen los datos del *BackEnd* antes de continuar.

Se usan las funciones de *React* como *componentDidMount()*, una vez montado el componente se obtienen los datos que definen su estado.

FrontEnd – Problemas y dificultades

- Despliegue de la aplicación en *GitHub Pages*.
 - Al desplegar la aplicación no se podía navegar entre las opciones de menú, no se encontraban los recursos.
 - Causa: conflictos con el componente *BrowserRouter* de *ReactRouter*.

```
import { HashRouter as Router, Switch } from "react-router-dom";
```

```
const AuthNavBar = (props) => {  
  const logout = () => {  
    auth.logout();  
    window.location = "#/login";  
  };  
  return (  
    <Navbar bg="primary" variant="dark">  
      <Navbar.Brand href="#/">Viade-Rutas</Navbar.Brand>  
      <Nav>  
        <Nav.Link href="#/ver-rutas">Mis rutas</Nav.Link>  
        <Nav.Link href="#/add-menu">Crear rutas</Nav.Link>  
        <Nav.Link href="#/friends">Amigos</Nav.Link>  
        <Nav.Link href="#/shared">Compartido conmigo</Nav.Link>  
      </Nav>  
    </Navbar>  
  );  
};
```

Solución: sustituir *BrowserRouter* por *HashRouter*. Se antepone '#' en la url.

Implica modificar atributo **href** de los *links*.

FrontEnd – Problemas y dificultades

- *Problemas con el mapa de ReactLeaflet y componentes de React Bootstrap.*
 - Envolver mapa dentro de un componente de *bootstrap* (*Accordion* , panel *Modal...*), no se renderizan bien sus *layers*.
 - Zonas del mapa que quedaban sin cargar, con un tono gris.

```
<div
  id={`mapa-${this.props.ruta.getNombre()}`}
  className="m1-3 mb-3"
>
  {this.props.showMap && (
    <MapRuta
      className="map"
      ruta={this.props.ruta}
      data-testid="mapa"
      refMapa={this.refMapa}
    />
  )}
</div>
```

Solución: no envolver el componente del mapa en componentes de *bootstrap* como *Accordion* o *Modal* y utilizar *divs* con estilo en su lugar.

FrontEnd - Opinión sobre *React*.

- **Ventajas:**

- Potente gracias a la actualización automática de los elementos del *DOM*.
- Gran cantidad de librerías y componentes disponibles que encapsulan funcionalidad (galería de fotos, componentes *SOLID*...)

- **Inconvenientes:**

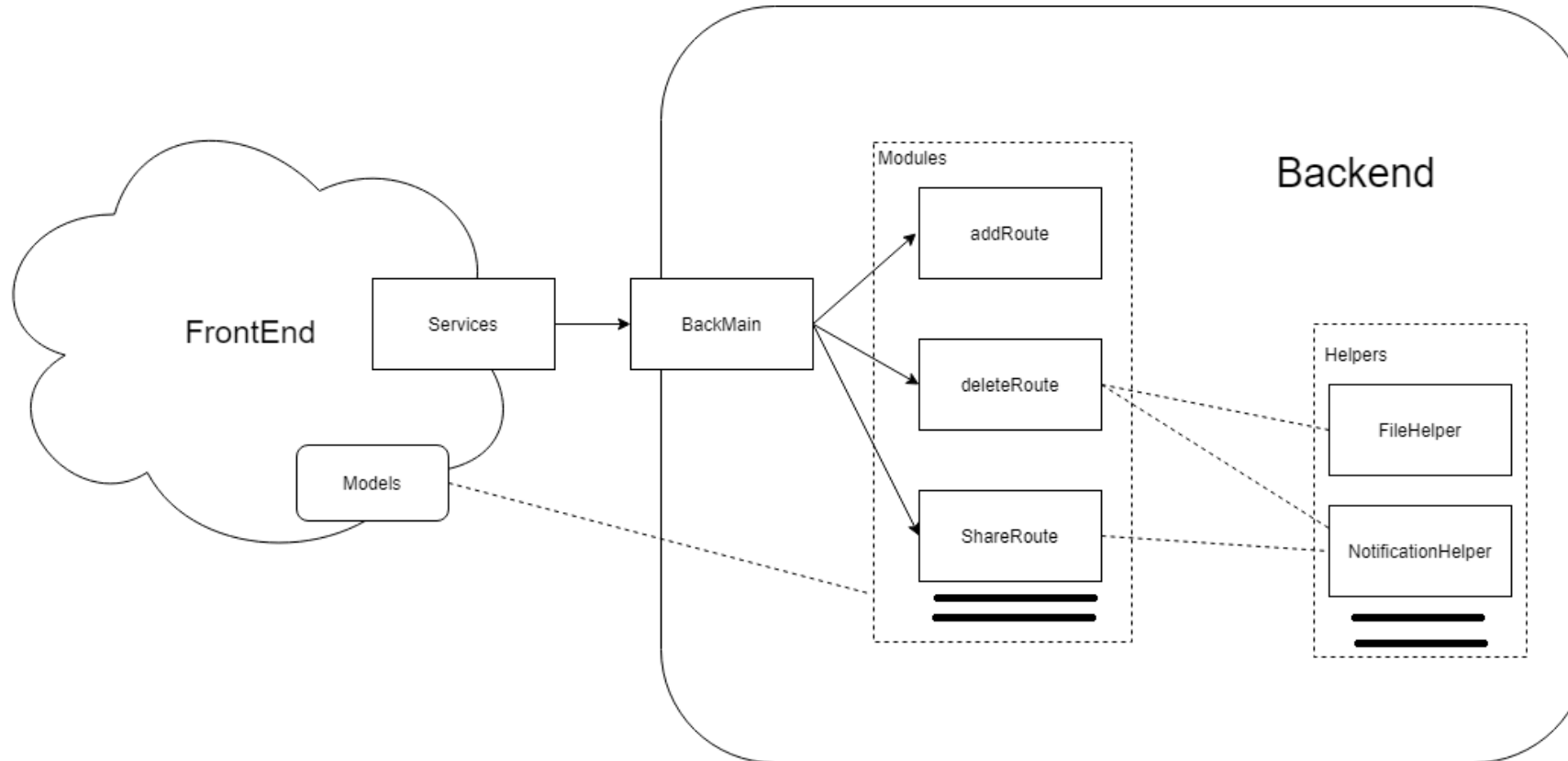
- Dificultad para adaptarse a su uso. Manejo del **estado** interno de los componentes, jerarquía de padre-hijos...
- Implementación compleja de algunos aspectos de la interfaz: listado de rutas, panel para compartir con amigos...

BackEnd

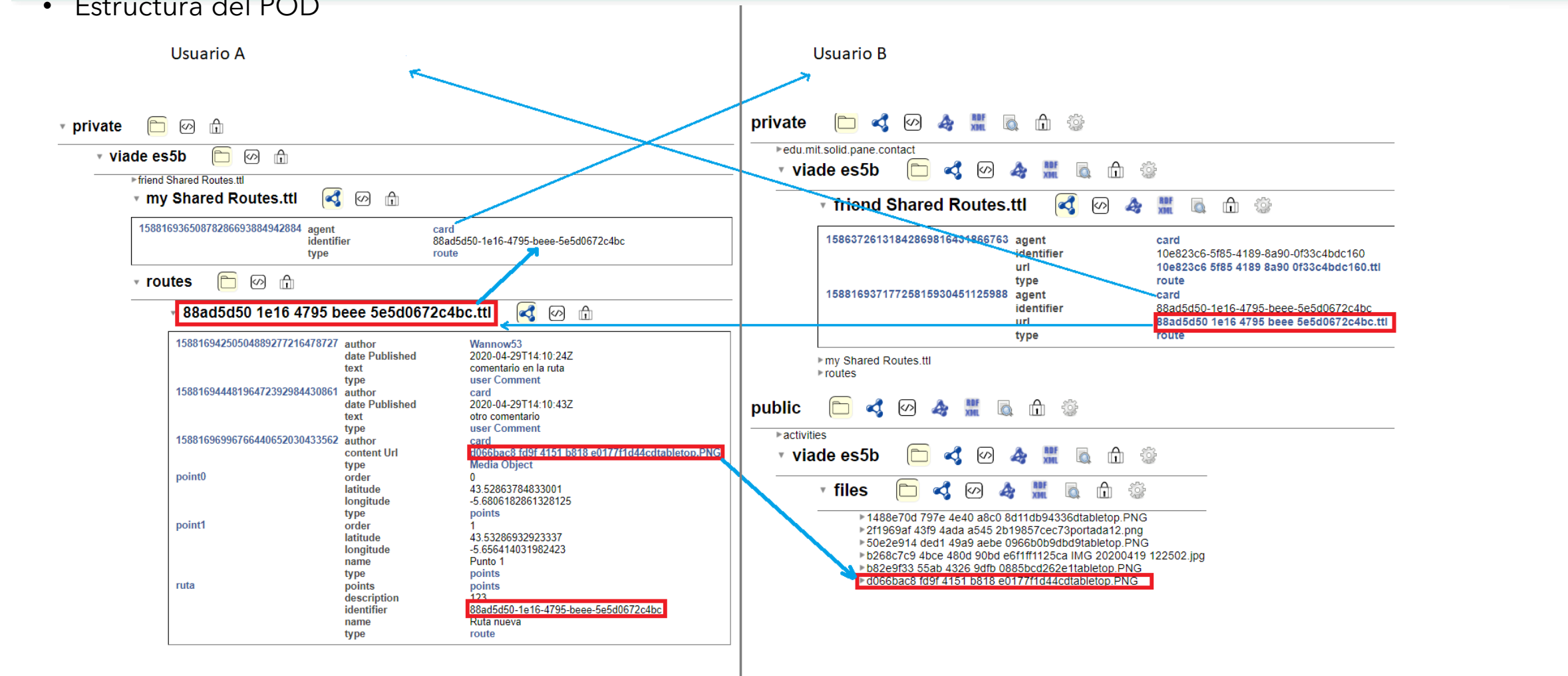


BackEnd- Arquitectura

- Estructura de paquetes: paquete backend.



100



BackEnd- Funciones Principales

- ¿Cómo se comparte una ruta?
 - Se establece como compartida
 - Se le dan permisos al otro usuario
 - Se manda una notificación
 - Se agrega a rutas que me han compartido
- ¿Cómo se comenta o se sube un fichero a una ruta?
 - Se busca la ruta en el pod del dueño
 - Se guarda la direccion del fichero o el comentario

BackEnd- Problemas y dificultades

- ¿Cómo hacer saber a un amigo que tiene rutas nuevas?
 - *Bandeja de entrada en el POD*
 - *Procesado en segundo plano de mensajes*
- ¿Cómo guardar datos de rutas en un POD en el que no tienes permisos?
 - *Modificar fichero de permisos*

BackEnd - Tecnologías

- Módulos y tecnologías empleadas:
 - **TripleDoc:** librería de manejo de documentos SOLID sencilla
 - **Solid-File-Client:** librería de control de ficheros en el pod
 - **Solid-Auth-Client:** librería de autenticación



Tests

Tests TDD

- Tecnologías utilizadas
 - Librería react-testing
 - Librería testing-library/jest-dom
 - Librería react-test-renderer
 - Librería react-dom/test-utils

Test TDD – Dificultades

- **Dificultad para *testear* los componentes.** Pruebas unitarias fallidas debido a que los componentes hacían uso de funciones concretas de los servicios.

```
let rutaService = new RutaService();

test("Se renderizan bien los componentes de SharedWithMe", () => {
  afterAll(cleanup);
  const { getByTestId } = render(<SharedWithMe
    handleLoaded={rutaService.handleLoaded}
    getRutasCompartidasConnmigo={rutaService.getRutasCompartidasConnmigo}
    subirFicheroARuta={rutaService.subirFicheroARuta}
    obtenerFicherosRuta={rutaService.obtenerFicherosRuta}
    comentarRuta={rutaService.comentarRuta}
    obtenerComentariosRuta={rutaService.obtenerComentariosRuta}></SharedWithMe>);
  expect(getByTestId("title")).toHaveTextContent("Compartido connmigo");
  expect(getByTestId("textoCompartidoConnmigo")).toHaveTextContent("En esta sección puedes v
});
```

Solución:

Gracias al desacoplamiento del frontend pudimos simular las llamadas a funciones del backend con datos de prueba en servicios creados por nosotros.

Test TDD – Dificultades

- **Dificultad para testear los componentes.** Pruebas unitarias fallidas al probar componentes que usaban dependencias externas como un mapa o conexión a la cuenta del *POD*.

```
test("RouteCard contiene el botón de eliminar.", () => {  
  const mock = jest.fn();  
  const { getByTestId } = render(  
    <RouteCard  
      ruta={ruta}  
      obtenerFicherosRuta={rutaService.obtenerFicherosRuta}  
      flyTo={mock}  
    ></RouteCard>  
  );  
  expect(getByTestId("rb-eliminar")).toHaveTextContent("Eliminar");  
});
```

Solución:

Utilizar los mocks proporcionados por jest para crear nuestras propias funciones que simulen el comportamiento y pasárselas a los componentes en sus props.

Hacer pruebas en base a los argumentos recibidos por los mocks, nº de veces que son llamados...

Tests BDD

- Módulos utilizados:
 - Jest-cucumber
 - Jest-puppeteer
- Organización de las carpetas:

```
✓ e2e
  > features
  > step-definitions
  JS jest.config.js
```

Problemas encontrados y soluciones

- Problema: Timeout demasiado bajo.
- Solución: Aumento del timeout de los componentes.
- Problema: Launchtime bajo.
- Solución: Modificación del archivo de configuración "jest-puppeteer.config.js".

Entornos y Tecnologías auxiliares

Docker

Dockerfile

Dockerfile > FROM

```
1 FROM node:10.13-alpine
2 ENV NODE_ENV production
3 WORKDIR /usr/src/app
4 COPY ["package.json", "package-lock.json*", "npm-shrinkwrap.json*", "./"]
5 RUN npm install
6 COPY . .
7 EXPOSE 3000
8 CMD ["npm", "start"]
```

docker-compose.yml

docker-compose.yml

```
1 version: '3.4'
2
3 services:
4   viade_es5b:
5     image: viade_es5b
6     build: .
7     environment:
8       NODE_ENV: production
9     ports:
10      - "3000:3000"
11   solidserver:
12     image: nodesolidserver/node-solid-server
13     build: ./node-solid-server/
14     volumes:
15       - ./volumes/soliddata:/usr/src/app/data
16       - ./volumes/soliddb:/usr/src/app/.db
17     ports:
18       - "8443:8443"
19   volumes:
20     soliddata:
21       external: false
22     soliddb:
23       external: false
```

Tests Gatling

> gatling-charts-highcharts-bundle-3.3.1

✓ GatlingTest

PruebaAñadirAmigo.scala

PruebaAñadirRuta.scala

PRuebaCompartirRuta.scala

PruebaCrearCuenta.scala

PruebaCrearRuta.scala

PruebaIniciarSesion.scala



Entorno

- Codacy
- Codecov

Viade_es5b - Demo

- [Añadir una ruta, visualizarla y eliminarla.](#)
- [Compartir una ruta, adjuntarle fotos y comentarios.](#)
- [Comentar en rutas compartidas y adjuntarles fotos.](#)

FIN

