



Batec de Festa

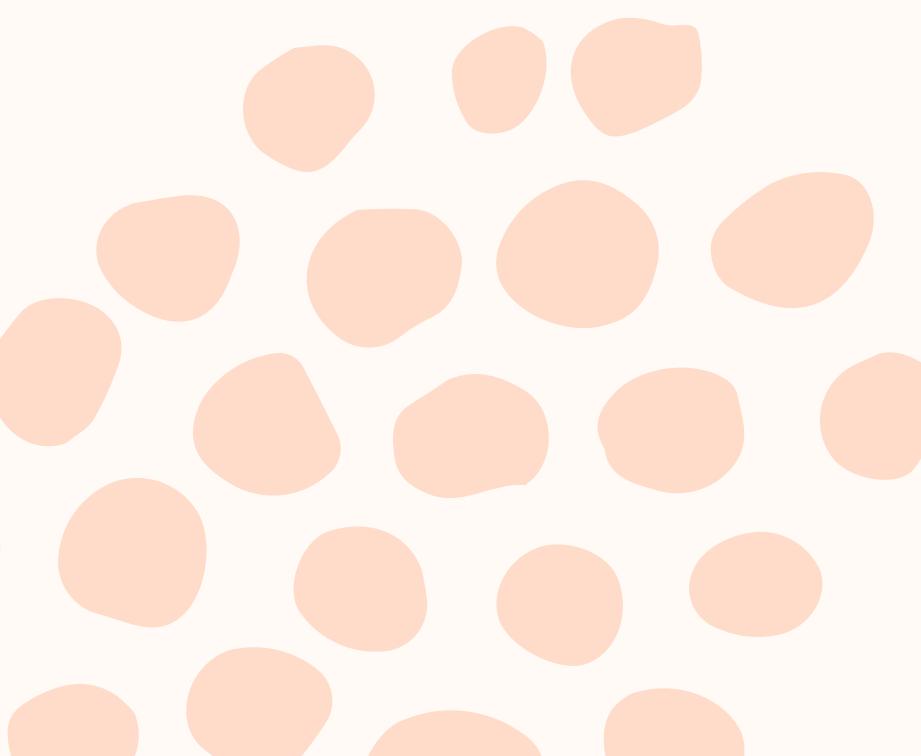
- Cristina Mateo -

Contenidos

1. Introducción
2. Desarrollo
3. Frontend
4. BBDD
5. Backend
6. Demo



Introducción



Introducción

Inici
Registre

Batec de festa

Filtrar per ciutat:
Sense filtre

Prova per eliminar

Ctrl Alt Del

Ciutat: Barcelona
Data: 2023-12-19

Web de organización
de eventos
Sin usuario loggeado

Vista detalle del evento

Benvingut. Hola

Detalls de l'esdeveniment

Nadal a casa
Barcelona

On?
Dones a casa, on ha de ser

Quan?
2023-12-25, 14:06:00

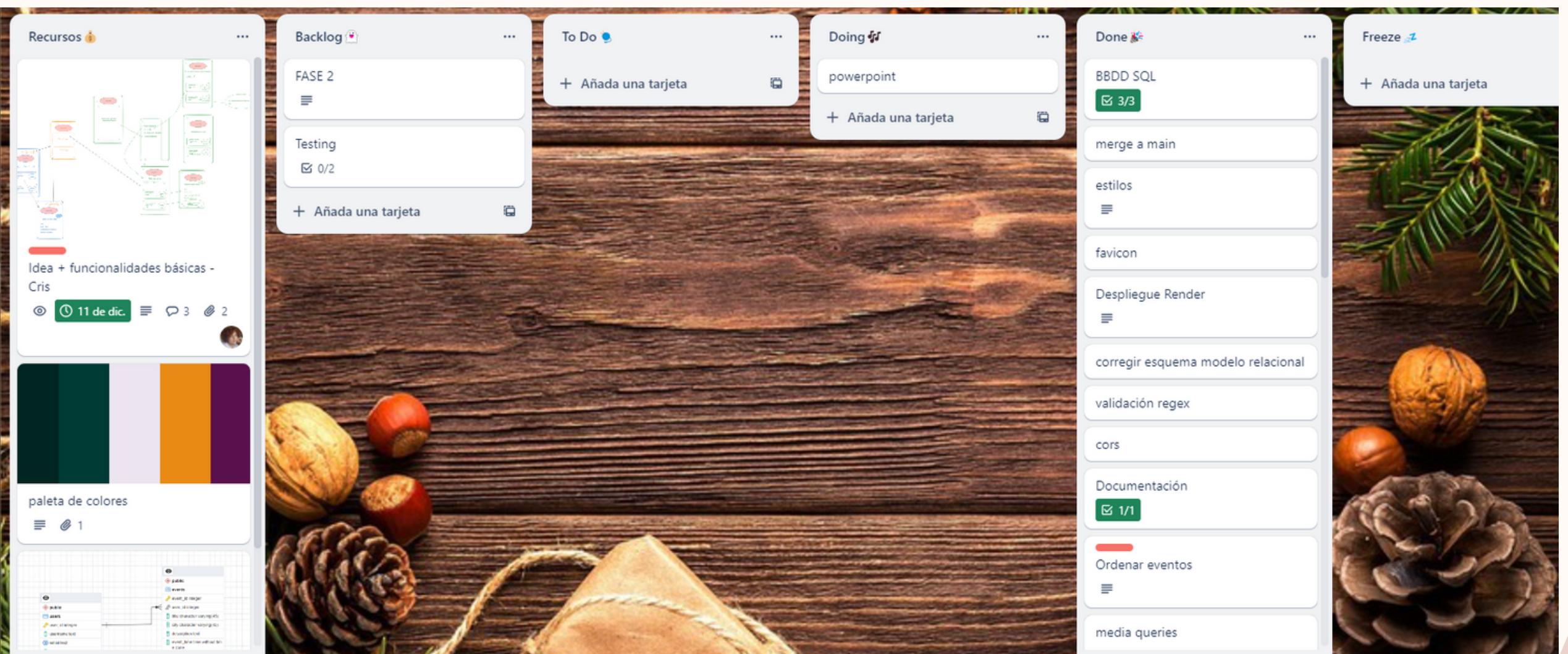
Celebrarem el nadal i ens posarem de corrons fins amunt

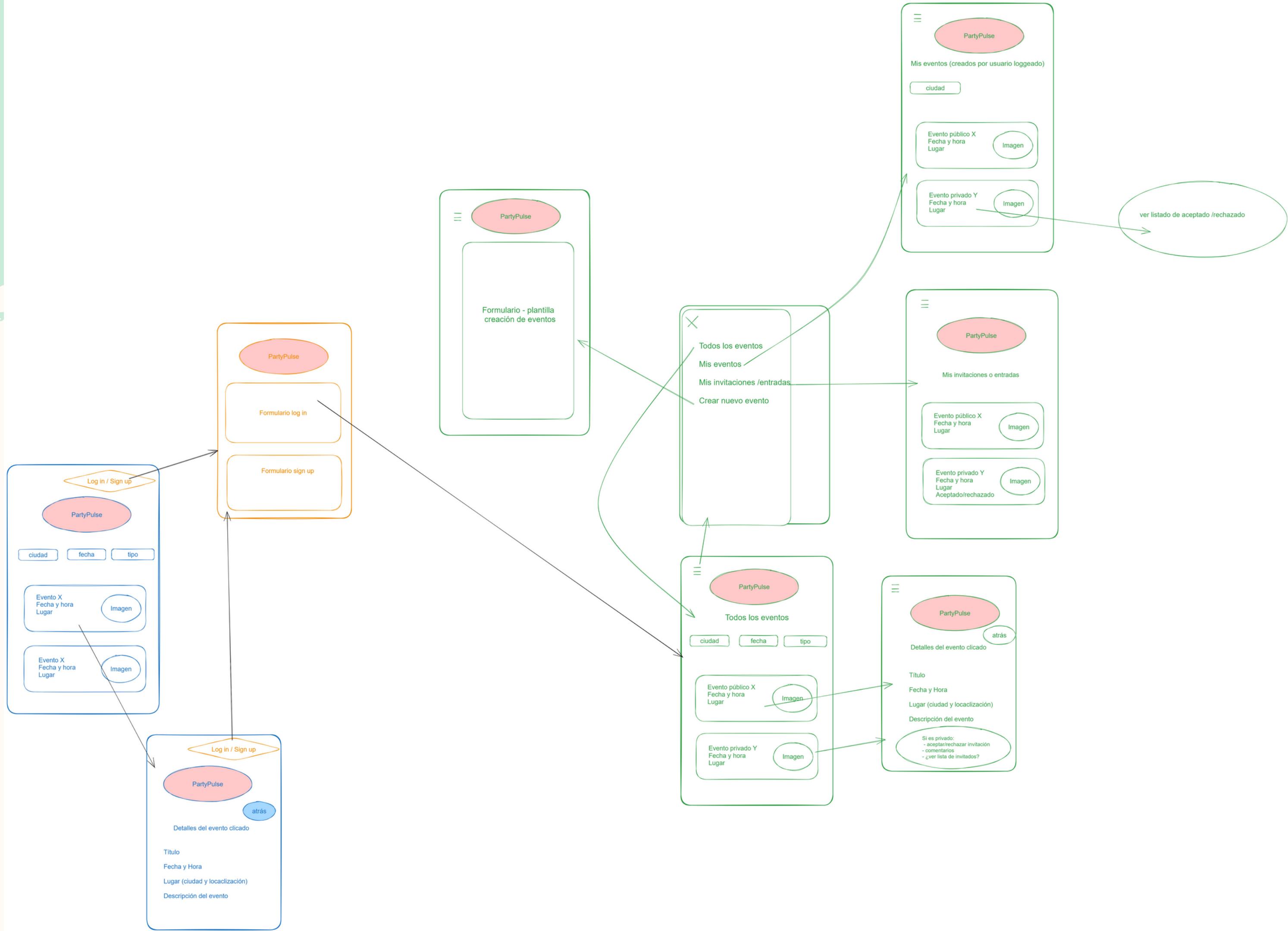
Desarrollo



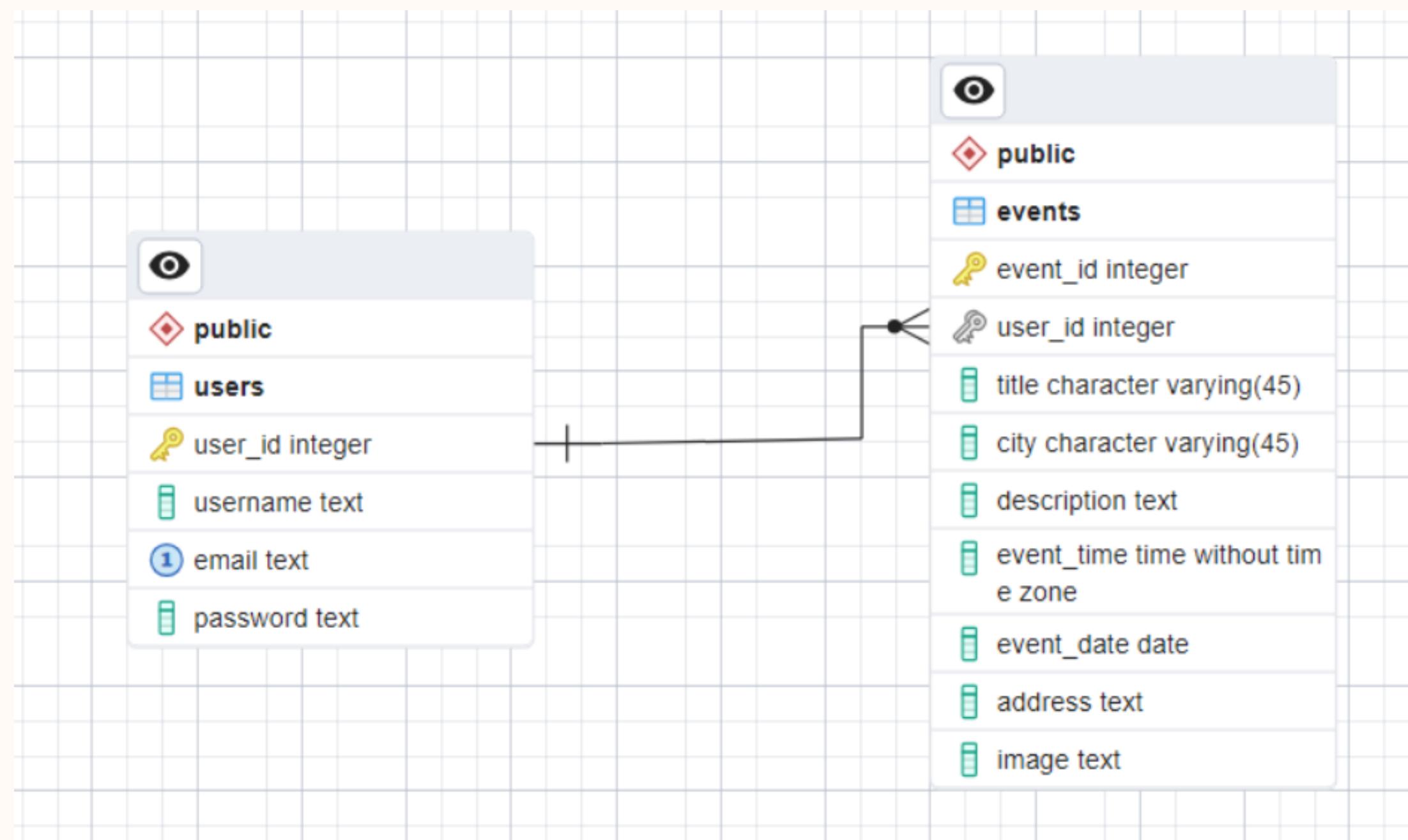
Desarrollo

- Node.js
- Express
- JWT
- PostgreSQL
- ElephantSQL
- React.js
 - Sass
- Docker
- Render





Desarrollo





Frontend

Frontend

Listar todos

Registro

CRUD de eventos propios

Filtrar por ciudad

Vista detalle

Mis eventos

```
const Home = () => {  
  
  const [allEvents, setAllEvents] = useState([]);  
  const [filteredEvents, setfilteredEvents] = useState([]);  
  const [cities, setCities] = useState([]);  
  const [selectedCity, setselectedCity] = useState("");  
  const [isLoading, setIsLoading] = useState(false);  
  
  const fetchAllEvents = async () => {  
    try {  
      setIsLoading(true)  
      const response = await axios.get('/api/events');  
      const eventData = response.data;  
      setAllEvents(eventData);  
      setfilteredEvents(eventData)  
      setIsLoading(false)  
    } catch (error) {  
      console.error("Error fetching events", error);  
    }  
  };  
  
  const fetchCities = async () => {  
    try {  
      const response = await axios.get('/api/cities')  
      const cityData = await response.data.map(obj => obj.city);  
      setCities(cityData)  
    } catch (error) {  
      console.error("Error fetching cities", error);  
    }  
  };  
  
  useEffect(() => {  
    fetchAllEvents();  
    fetchCities();  
  }, []);  
  
  const filter = (event) => {  
    setselectedCity(event.target.value)  
  }  
};
```

Inici

Crear esdeveniment

Els meus esdeveniments

Batec de festa

Benvingut, "Hola"

Filtrar per ciutat:

Sense filtre

Frontend

```
title: datos.title,
img: datos.image,
city: datos.city,
time: datos.event_time,
date: datos.event_date,
address: datos.address,
description: datos.description,
email: datos.email
}
setEventDet(detailedData);
} catch (error) {
  console.log("Lo sentimos, no hemos podido acceder a los detalles")
}
setIsLoading(false)
}
fetchEventsDet();
}, []);

const deleteEvent = async (e) => {
  e.preventDefault()
  await axios.delete(`/api/myevents/${id}/${loggedemail}`)
  navigate('/myevents')
}
return (
  <section>
    <h2>Details de l'esdeveniment</h2>
    {isLoading && <Loader />}
    {!isLoading && eventDet &&
      <article id="eventDet">
        <h3>{eventDet.title}</h3>
        <h4>{eventDet.city}</h4>
        <p>On? <br />{eventDet.address}</p>
        <p> Quan? <br />{eventDet.date.slice(0, 10)} , {eventDet.time}</p>
        <p>{eventDet.description}</p>
        <img className="eventDetImg" src={eventDet.img} alt="imagen del evento" />
        <br />
        <span>
          {loggedUser && loggedemail === eventDet.email &&
            <button className="delete-button" onClick={deleteEvent}>Eliminar aquest esdeveniment</button>}
        </span>
        <br />
        <span>
          {loggedUser && loggedemail === eventDet.email &&
            <Link className={'update-link'} to='/update'>Punxa aquí per editar un esdeveniment</Link>
          </span>
        </span>
      </article>
    }
  </section>
)
```



Backend

Backend

Log in & sign up

```
usersSqlRouter.post("/addUser", usersSqlController.createUser)  
usersSqlRouter.post("/login", usersSqlController.login)
```

CRUD

```
apiRouter.get("/events", apiController.getAllEvents)//todos los eventos de la bbdd  
apiRouter.get("/events/:id", apiController.getOneEvent)//un evento, detalles  
apiRouter.get("/cities", apiController.getCities)  
apiRouter.get("/myevents/:email", apiController.getMyEvents)//eventos propios  
apiRouter.post("/myevents/:email", apiController.createEvent)//crear evento  
apiRouter.put("/myevents", apiController.updateEvents)//modificar evento  
apiRouter.delete("/myevents/:id/:email", apiController.deleteEvent)//eliminar evento
```

Backend

Queries

```
--obtener todos los eventos (GET)
SELECT title, city, image, event_time, event_date, event_id,address,description
| | FROM events;
--obtener eventos propios (GET)
SELECT title, city, e.image, event_time, event_date, event_id,address,description
| | FROM events AS e
| | INNER JOIN users AS u
| | ON e.user_id=u.user_id
| | WHERE u.email = $1;
--Obtener un solo evento(GET)
SELECT title, city, e.image, event_time, event_date, event_id, address, description, u.email
| | FROM events AS e
| | INNER JOIN users AS u
| | ON e.user_id=u.user_id
| | WHERE event_id=$1;
--Crear nuevo evento (POST)
INSERT INTO events(title, image, city, address, description, event_time, event_date, user_id)
| | VALUES ($1, $2, $3, $4, $5, $6, $7,
| | (SELECT user_id FROM users WHERE email=$8));
--Modificar datos del evento propio (PUT)
UPDATE events AS e
| | SET title = $1,
| | image = $2,
| | city =$3,
| | address = $4,
| | description = $5,
| | event_time = $6,
| | event_date = $7
| | FROM users AS u
| | WHERE e.user_id = u.user_id
| | AND e.title = $8 -- Título actual
| | AND u.email = $9; --mail del usuario
--Eliminar evento propio por título(DELETE)
DELETE FROM events AS e
| | USING users AS u
| | WHERE e.user_id = u.user_id
| | AND e.event_id = $1
| | AND u.email = $2;
--Crear nuevo usuario (POST)
INSERT INTO users(username,email,image,password)
| | VALUES
| | ($1, $2, $3, $4);
--get info de usuario por email para proceso de login
SELECT username, email, image, password
| | FROM users
```

Backend

```
 Dockerfile X
 Dockerfile
 1 FROM node:18-alpine
 2 WORKDIR /app
 3 + COPY . .
 4 RUN npm install
 5 WORKDIR /app/client
 6 RUN npm install && npm run build
 7 WORKDIR /app
 8 CMD ["npm", "run", "deploy"]
 9 EXPOSE 8080
```

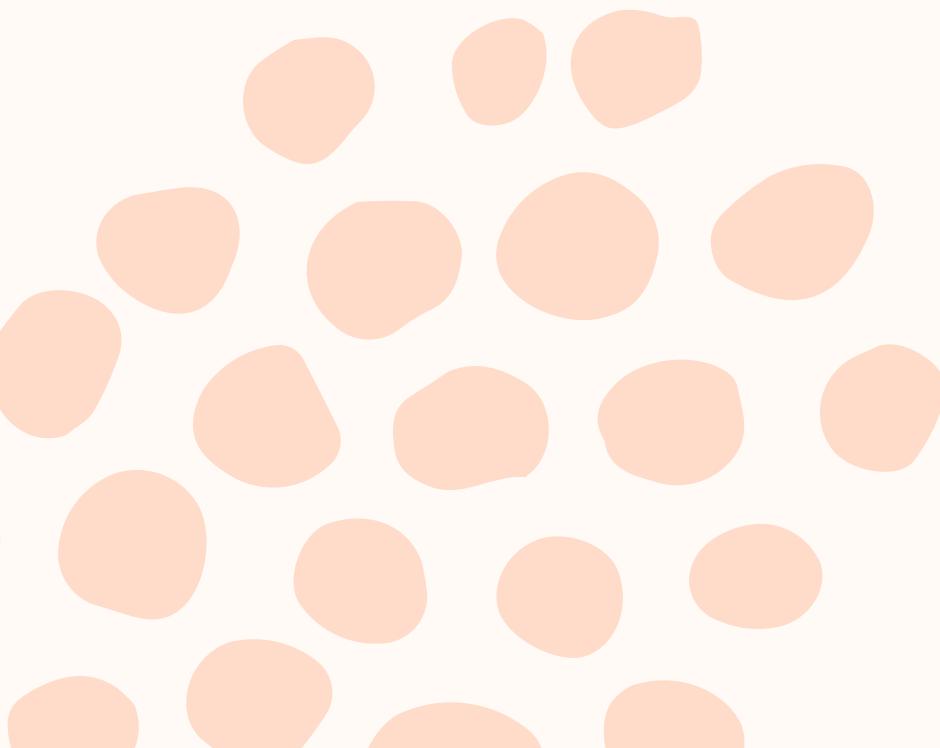
```
* Serve static assets in production, must be at this location of this file
if (process.env.NODE_ENV === 'production') {
    /*Set static folder
    app.use(express.static('client/dist'));

    app.get('*', (req,res) => res.sendFile(path.resolve(__dirname, 'client', 'dist','index.html')));
}

//para rutas no existentes
app.use('*',error404)
```



Demo





Muchas
Gracias