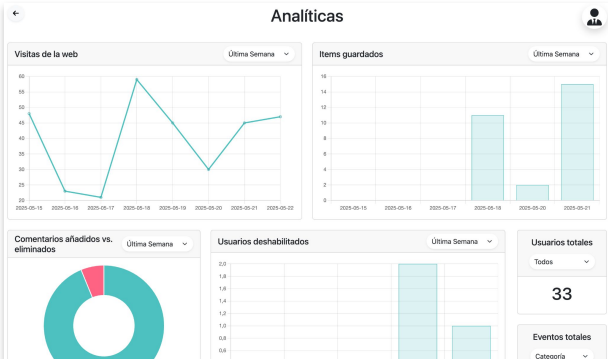
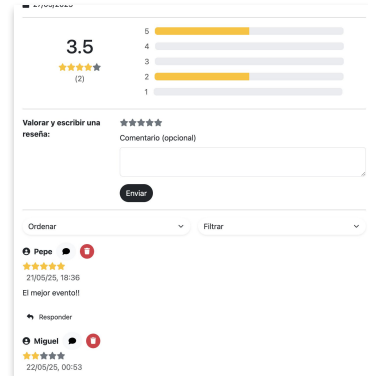
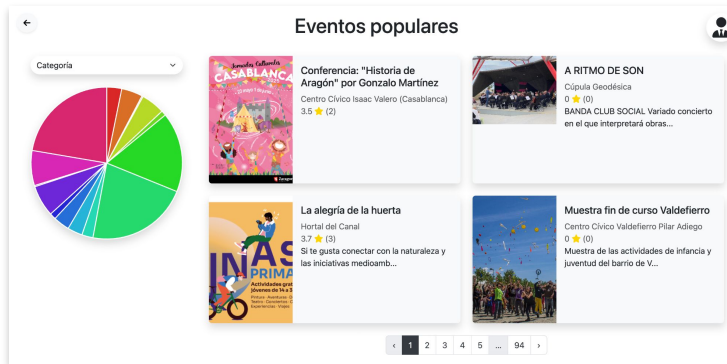
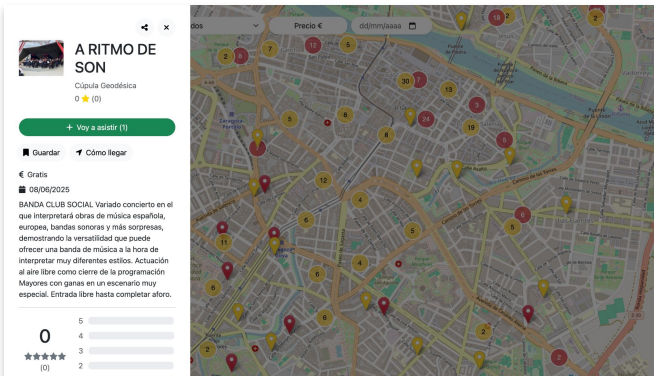


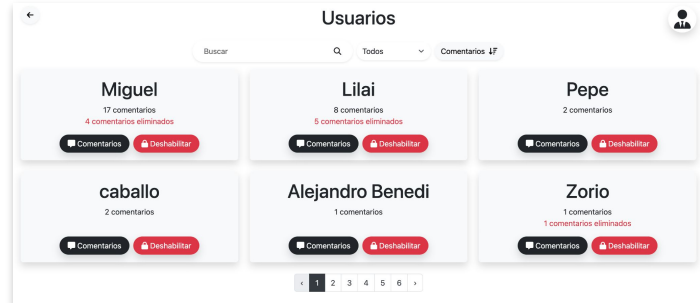
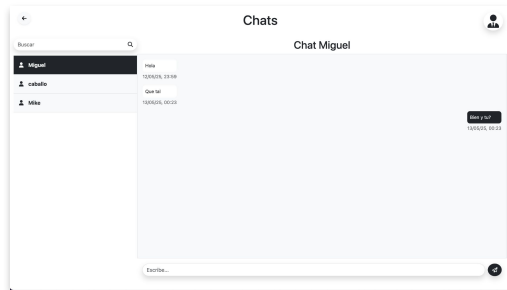


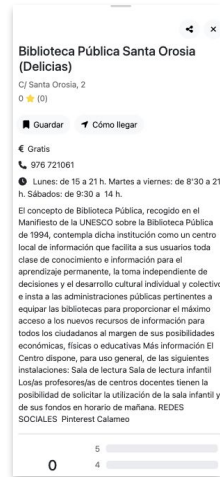
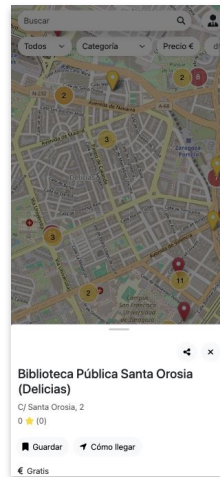
CulturaViva

Miguel Granel 798913
Juan Antonio Almodóvar 839953
Alejandro Benedí 843826
Elizabeth Lilai Naranjo 840091



¿Qué es?





Usuarios

Buscar

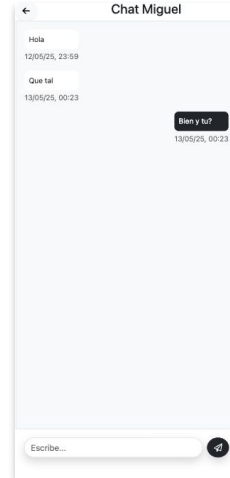
Todos Comentarios 17

Miguel 17 comentarios 4 comentarios eliminados	Comentarios Desahabilitar
Lilai 8 comentarios 5 comentarios eliminados	Comentarios Desahabilitar
Pepe 2 comentarios	Comentarios Desahabilitar
caballo 2 comentarios	Comentarios Desahabilitar
Alejandro Benedi	Comentarios

Chats

Buscar

Miguel
caballo
Mike



Fuentes de datos

Datos de Eventos:

API REST

- Catálogo de datos Agenda Zaragoza
(<https://www.zaragoza.es/sede/portal/datos-abiertos/servicio/catalogo/282>)

Datos de Lugares:

- Catálogos de datos de infraestructuras culturales: Salas de música, museos, centros de tiempo libre, zonas jóvenes, centros cívicos, galerías de arte...
(<https://www.zaragoza.es/sede/portal/datos-abiertos/servicio/catalogo/?query=Cultura>)

JSON



Zaragoza
AYUNTAMIENTO

Zaragoza OpenData

Fuentes de datos

Carga de datos utilizando *node-cron*

Basado en *cron* de GNU/Linux

- Agenda Zaragoza -> actualización diaria
-> tarea cron todos los días a las 00:00
- Infraestructuras -> actualización anual
-> última vez en diciembre -> tarea cron
el 1 de enero de cada año

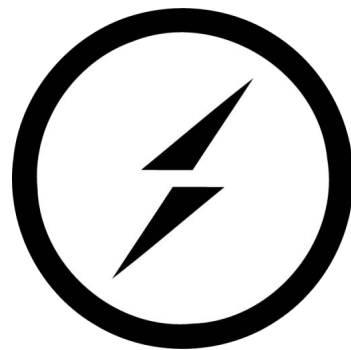
```
# |----- second (optional)
# |----- minute
# |----- hour
# |----- day of month
# |----- month
# |----- day of week
# |
# * * * * *
```



Back-end

Desarrollado en node con el *framework* express

- **Autenticación:** Passport.
 - Login social → GitHub y Google
 - Login tradicional (usuario y contraseña)
 - JWT
- **Base de datos:** MongoDB en Atlas.
 - Mongoose para definir los modelos y mapeo OO→ documentos
- **Comunicación tiempo real:** Socket.io
 - WebSockets con fallback a *long polling*.



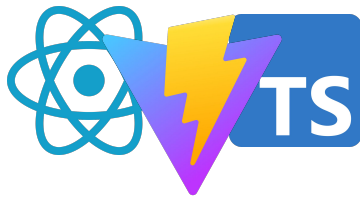
Back-end

- Documentación API: Swagger
- Envío de emails: nodemailer
- Logging: Winston con rotación de logs
- Validación de Schemas: AJV
- Encriptación: Bcrypt y crypto



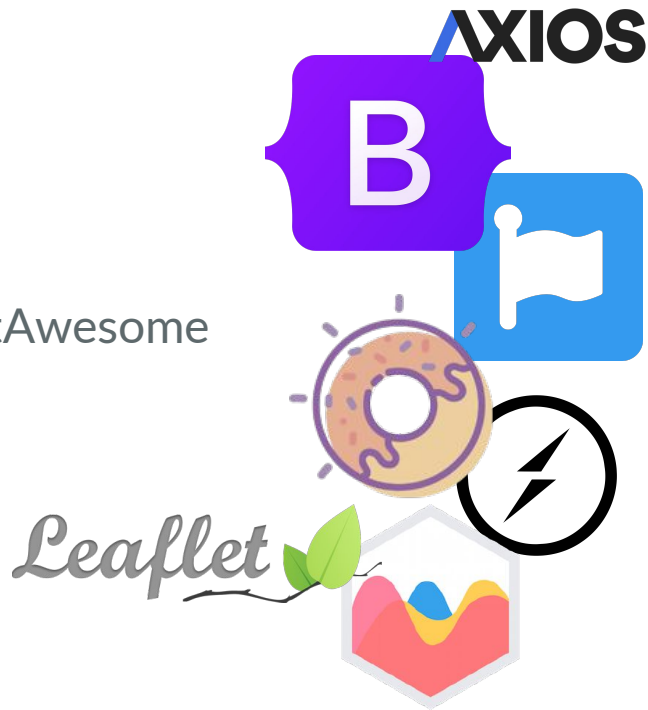
Nodemailer

Frontend



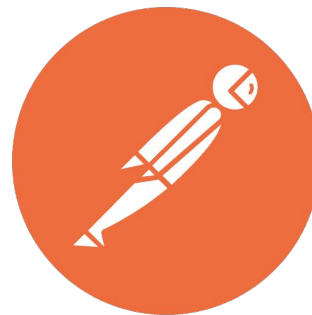
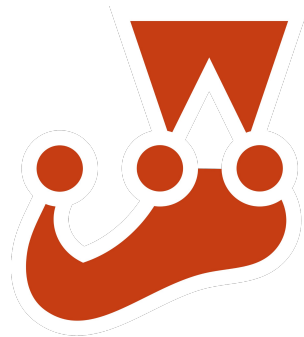
Desarrollado con el *framework* React junto a vite y empleando typescript

- Comunicación HTTP: Axios
- Queries, caché, auth: React Query (+Auth)
- Routing: React Router
- Formularios y validación: React Hook Form y Zod
- Estilos, componentes e iconos: Bootstrap 5 y FontAwesome
- Gráficas: Chart.js
- Mapas: Leaflet
- Notificaciones & Alertas: SweetAlert2
- WebSockets: socket.io-client



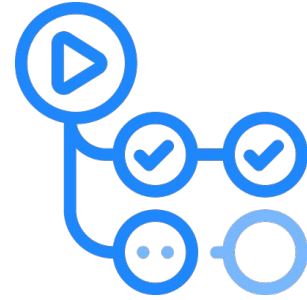
Validación y pruebas

- Tests unitarios: Jest
- Validación de Schemas: AJV
- Test e2e: Postman



Mejoras implementadas

- DevOps: GitHub Actions
- Login con al menos 2 sistemas externos: Github y Google
- Cobertura de tests: Entre 70 y el 80% de coverage
- Analizadores de código: SonarQube Cloud



Demo

<https://culturaviva-frontend.onrender.com/>