

UniLiving



Jorge Leris Lacort
Luis Daniel Gómez Sevilla
Ahmed Karafy Mohib
Kamal Bouizy Ghazzal



Tabla de contenidos

01

Fuentes de datos

02

Uso de los datos

03

Back-end y API

04

Frontend

05

Mejoras

06

Validación y pruebas

07

Demo de la aplicación



Fuentes de datos


- Dos fuentes de datos utilizadas
- Fuente de datos pública del ayuntamiento de Zaragoza
- Fuente de datos de Idealista (solicitud de acceso a api vía web)





Uso de los datos - Zaragoza

API Zaragoza → https://www.zaragoza.es/docs-api_sede/#/

- 
- Integración de datos directa con pequeño mapeado.
 - Datos demográficos.

- Integración de datos de equipamientos con algoritmo de filtrado y cálculo de distancias.
- Bibliotecas, Hospitales, Centros de Salud, Transporte Urbano, Educación Infantil, Educación Secundaria, Universitaria, Fitness, Supermercados Medianos, Supermercados Pequeños.



Uso de los datos - Idealista

- Api con acceso a información pisos, habitaciones, oficinas, etc.
- Acceso limitado permitido solo 100 peticiones/mes, solución: solicitar los pisos de manera periódica y almacenarlos en la BD.
- Uso para acceder a pisos y habitaciones disponibles para alquilar en Zaragoza.
- Información extensa pero limitada, disponible datos interesantes (ubicación, metros, amueblado, mascotas, descripción, etc.)
- No disponibles las imágenes (solo la de presentación).



Backend y API



- Uso de Nodejs.
- Uso de MongoDB y mongoose para persistencia de datos.
- Uso de Winston para logging: en desarrollo imprime por consola en producción envía correos a través de una cuenta de google (nodemailer).
- Uso de Passport para autenticación local y con google.
- Uso de ESLint y Prettier para herramientas de análisis y formato automático de código.
- Documentación en swagger.



Frontend



- Tecnologías principales
 - React
 - Bootstrap + React-Bootstrap
- Estructura básica de aplicaciones React
 - Componentes, páginas, assets ...
- Funcionalidades destacadas
 - Mapas interactivos (Leaflet y clustering)
 - Gráficos (Recharts)
 - Zonas con GeoJSON (barrios Zaragoza)
 - Etc



Mejoras implementadas

- Uso de reCAPTCHA v2 en el formulario de registro (prevención de bots) y sistema de control de intentos fallidos, tras un nº de intentos se bloquea temporalmente.
- Integración continua con github actions.
 - Backend: comprobación de código con ESLint, ejecución de tests y carga de los mismo en SonarCloud, despliegue en Render.
 - Frontend: comprobación de código con ESLint y nota mínima de Codacy para despliegue.
- Análisis de código estático, uso de SonarQube en el Backend y Codacy en Frontend.

Mejoras implementadas 2

Coverage	▼
Overview	
New Code	
Coverage	77.0%
Lines to Cover	941
Uncovered Lines	229
Line Coverage	75.7%
Conditions to Cover	330
Uncovered Conditions	63
Condition Coverage	80.9%

Coverage on New Code 77.0%			
New code: since previous version			
	Coverage on New Code	Uncovered Lines on New Code	Uncovered Conditions on New Code
app_server	77.0%	229	63
1 of 1 shown			

controllers	85.1%	79	40
externalData	0.0%	86	0
middleware	100%	0	0
models	93.2%	2	3
routes	73.4%	37	0
tests	–	0	0
utils	72.1%	11	18



Validación y pruebas

- Para la validación del Backend se ha utilizado jest para realizar tests, además de esto también se ha utilizado MongoMemoryServer para tener una base de datos en memoria de prueba para realizar testing sobre los modelos.
- Se han realizado diversas pruebas para cada módulo y también pruebas que utilizaban distintos módulos en conjunto.
- Para poder comprobar que el testeo es correcto y cubre los suficientes casos se ha utilizado el test coverage de SonarCloud (coverage alcanzado de 77%)

Demo de la aplicación



GRACIAS!

¿Alguna pregunta?

845647@unizar.es

819304@unizar.es

850111@unizar.es

838700@unizar.es
