



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Desarrollo de una aplicación web con React para la  
creación automática de páginas HTML para O-CITY

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Marín-Roig Zozaya, Jorge

Tutor/a: Marín-Roig Ramón, José

Cotutor/a externo: Burruchaga Sola, Daniel

CURSO ACADÉMICO: 2023/2024

# Resumen

O-CITY (<https://o-city.org/>) es un atlas digital del patrimonio natural y cultural para promover el turismo responsable, la educación para el desarrollo sostenible y la economía naranja en ciudades de todo el mundo. Los constantes avances tanto de la tecnología, como del modelo de negocio de O-CITY han motivado la evolución del proyecto hacia nuevas formas de visualizar la información en el mapa.

Este TFG ofrece una solución de navegación clásica, de tipo mosaico, para la visualización de páginas que representan los elementos de patrimonio georreferenciados en el mapa. A través de estas nuevas herramientas de desarrollo, como son el framework React-JavaScript y AntDesign, para realizar el frontend se busca una experiencia más accesible para los usuarios, y una herramienta de marketing alternativa para las ciudades, para la promoción de su cultura, patrimonio y naturaleza.

Palabras clave: O-CITY, React, JavaScript, AntDesign, navegación, mosaico.

## **Abstract**

O-CITY (<https://o-city.org/>) is a digital atlas of natural and cultural heritage to promote responsible tourism, education for sustainable development and the orange economy in cities around the world. The constant advances in both technology and the O-CITY business model have motivated the evolution of the project towards new ways of visualizing the information on the map.

This TFG offers a classic navigation solution, mosaic type, for the visualization of pages that represent the heritage elements georeferenced on the map. Through these new development tools, such as the React-JavaScript framework and AntDesign, to create the frontend we are looking for a more accessible experience for users, and an alternative marketing tool for cities to promote their culture, heritage and nature.

Keywords: O-CITY, React, JavaScript, AntDesign, navigation, mosaic.

# Agradecimientos

Primero de todo, especialmente quería darle las gracias a mi tutor de este trabajo, mi padre. Por toda la ayuda y apoyo tanto en las aulas como en casa desde el inicio de la carrera hasta el final. También a toda mi familia mi madre

Mayte por siempre estar ahí para todo, mi hermana Carla y por supuesto mi abuela Delfina. A mi pareja Mireia por apoyarme desde primer momento y a mi amigo Toledano. Por último, a Asun por toda la ayuda prestada siempre que la he necesitado. A todas estas personas muchas gracias.

# Índice

|  |           |
|--|-----------|
| <b>Capítulo 1. Introducción .....</b>                        | <b>6</b>  |
| Objetivos .....  | 7         |
| Metodología.....   | 8         |
| Estructura del documento.....                                | 9         |
| <b>Capítulo 2. Marco teórico. Contexto tecnológico .....</b> | <b>10</b> |
| O-CITY. Concepto .....                                       | 10        |
| O-CITY. Navegación .....                                     | 15        |
| Tecnologías de desarrollo web .....                          | 17        |
| Evolución tecnológica de O-CITY.....                         | 18        |
| <b>Capítulo 3. Análisis del problema .....</b>               | <b>21</b> |
| Solución propuesta. Especificaciones.....                    | 22        |
| <b>Capítulo 4. Diseño de la solución .....</b>               | <b>24</b> |
| Arquitectura del sistema .....                               | 24        |
| Diseño del frontend .....                                    | 25        |
| Tecnología utilizada.....                                    | 26        |
| <b>Capítulo 5. Desarrollo de solución propuesta.....</b>     | <b>28</b> |
| Backend.....   | 28        |
| Frontend.....  | 30        |
| <b>Capítulo 6. Testeo .....</b>                              | <b>38</b> |
| <b>Capítulo 7. Conclusiones .....</b>                        | <b>39</b> |
| Objetivos de desarrollo sostenible (ODS).....                | 40        |
| Futuras líneas de trabajo.....                               | 41        |
| <b>Capítulo 8. Bibliografía .....</b>                        | <b>43</b> |
| <b>Anexo A. Estructura APIs .....</b>                        | <b>44</b> |
| <b>Anexo B. PDF Patrimonio.....</b>                          | <b>49</b> |

# Índice de tablas

|   |    |
|---|----|
| Tabla 1. Tareas Trello.....                       | 8  |
| Tabla 2. Evolución de tecnologías en O-CITY ..... | 19 |
| Tabla 3. API de países .....                      | 44 |
| Tabla 4. API de estados .....                     | 45 |
| Tabla 5. API de ciudades .....                    | 45 |
| Tabla 6. API de los elementos patrimoniales.....  | 47 |

# Índice de figuras

|   |    |
|---|----|
| Figura 1. Logo O-CITY. Fuente: <a href="https://o-city.webs.upv.es/">https://o-city.webs.upv.es/</a> .....            | 10 |
| Figura 2. Planeta O-CITY. Fuente: <a href="https://doi.org/10.1145/3631121">https://doi.org/10.1145/3631121</a> ..... | 12 |
| Figura 3. Página inicio O-CITY. Fuente: <a href="https://o-city.org/">https://o-city.org/</a> .....                   | 15 |
| Figura 4. Inicio O-CITY Zoom1. Fuente: <a href="https://o-city.org/">https://o-city.org/</a> .....                    | 15 |
| Figura 5. Visión ciudades. Fuente: <a href="https://o-city.org/">https://o-city.org/</a> .....                        | 15 |
| Figura 6. Visión patrimonio. Fuente: <a href="https://o-city.org/">https://o-city.org/</a> .....                      | 15 |
| Figura 7. Play Contents patrimonio. Fuente: <a href="https://o-city.org/">https://o-city.org/</a> .....               | 15 |
| Figura 8. Arquitectura versión primera. Fuente: Recurso propio .....  | 24 |
| Figura 9. Arquitectura actual. Fuente: Recurso propio .....   | 25 |
| Figura 10. Propuesta de diseño frontend. Fuente: Recurso propio .....   | 26 |
| Figura 11. Arquitectura Backend. Fuente: recurso propio.....  | 28 |
| Figura 12. Ejemplo de lectura del API. Fuente: Recurso propio .....   | 29 |
| Figura 13. Ejemplo filtrado de id. Fuente: recurso propio .....   | 29 |
| Figura 14. Ejemplo del uso de datos entre componentes. Fuente: recurso propio .....                                   | 30 |
| Figura 15. Estructura de la navegación de la página. Fuente: recurso propio .....                                     | 31 |
| Figura 16. Mosaico Países. Fuente: recurso propio.....  | 31 |
| Figura 17. Inicio de la ruta del proyecto. Fuente: recurso propio .....   | 32 |
| Figura 18. Selección de elementos en el mosaico. Fuente: recurso propio.....  | 32 |
| Figura 19. Muestra de datos de los mosaicos. Fuente: recurso propio .....   | 33 |
| Figura 20. Ruta del proyecto entero codificada. Fuente: recurso propio .....  | 33 |
| Figura 21. Mosaico de elementos patrimoniales de Alzira. Fuente: recurso propio.....                                  | 34 |
| Figura 22. Mosaico de ciudades en Valencia. Fuente: recurso propio .....  | 35 |
| Figura 23. Mosaico de estados en España. Fuente: recurso propio .....   | 35 |
| Figura 24. Página de los elementos patrimoniales. Fuente: recurso propio.....   | 36 |
| Figura 25. Testeo con Selenium IDE. Fuente: recurso propio .....  | 38 |

# Capítulo 1. Introducción

En un mundo cada vez más globalizado y digitalizado, la preservación y promoción del patrimonio natural y cultural de nuestras ciudades se ha convertido en un desafío y una oportunidad única. En este contexto, el proyecto O-CITY (Orange: Creativity Innovation & TechnologY) surge como una innovadora iniciativa europea financiada por el programa Erasmus+ (Knowledge Alliance), con el objetivo de crear un atlas digital del patrimonio mundial. Este proyecto no solo busca fomentar el turismo responsable y la educación para el desarrollo sostenible, sino también impulsar la economía naranja mediante la digitalización y promoción de activos culturales y naturales.

Este proyecto incorpora dos modos de visualización: en primer lugar, a través de la utilización de un mapa donde los elementos patrimoniales de cada una de las ciudades registradas aparecen geo-referenciados; y en segundo lugar, posteriormente, se añadió un estilo de navegación clásica, tipo mosaico, motivado esto fundamentalmente para incorporar una estrategia SEO en el proyecto. La navegación por mosaicos distribuye en una retícula rectangular los elementos a visualizar en cada página. En el caso de O-CITY los mosaicos corresponden a países, estados, ciudades y elementos de patrimonio.

Por otra parte, la organización mantiene la filosofía de “sin ánimo de lucro”, heredada del origen del proyecto bajo el paraguas del programa Erasmus+, pero ya explora alternativas de monetización que sean compatibles con esta filosofía y que aporten viabilidad al futuro del proyecto. Estas alternativas pasan por la creación de páginas de pago dentro de la aplicación con una experiencia de navegación aumentada para los usuarios, a través, por ejemplo, de la utilización de avatares alimentados con Inteligencia Artificial (IA)

La evolución de las tecnologías, que conlleva la aparición de nuevas herramientas de desarrollo, y la propia evolución/crecimiento del proyecto O-CITY y de su modelo de negocio, han motivado el desarrollo de una nueva versión del software y es en este marco de trabajo donde se plantea la realización de este trabajo de fin de grado (TFG).

El TFG se centra en crear una nueva estructura dinámica de páginas web generadas con React y tecnologías adyacentes. Unas páginas de navegación en estilo mosaico, donde se puede navegar entre países, regiones, y ciudades, para obtener los elementos patrimoniales de éstas últimas a través de la información obtenida de APIs (Application Programming Interface), utilizadas como fuentes de datos del desarrollo.

En los siguientes capítulos se hablará de O-CITY y se explicará su evolución, tanto a nivel de organización, como a nivel de software. De hecho, yo anteriormente ya trabajé en este proyecto, en unas prácticas de empresa, generando la primera versión de navegación por mosaicos, y al ver el gran cambio en el software que se estaba realizando, le planteé a la organización una refactorización del trabajo realizado hace dos años, utilizando ahora tecnologías actuales. Por lo tanto, se ha realizado la remodelación, utilizando páginas dinámicas y con un código que permite una mayor escalabilidad para futuras modificaciones.

## **Objetivos**

El objetivo de este proyecto es el desarrollo de una aplicación web que ofrezca un modelo de navegación de cinco niveles, en modo mosaico, para mostrar las páginas con la información de cada uno de los elementos patrimoniales.

Dentro de este objetivo principal se deben concretar otros sub-objetivos:

1. Comparar diversas tecnologías de desarrollo web con la finalidad de seleccionar la más adecuada para el objetivo planteado.
2. Analizar la primera versión y los cambios que motivan la generación de una segunda versión.
3. Diseñar un prototipo de navegación por mosaico para O-CITY.
4. Desarrollar el software de navegación.
5. Realizar pruebas con datos reales obtenidos de O-CITY.
6. Documentar el trabajo.

## Metodología

Se ha utilizado una metodología Scrum [1], tal y como se nos ha enseñado en el grado.

Se iban realizando cierto número de tareas que posteriormente iban a ser revisados por el “Scrum Master” y el “Product Owner” para poder seguir con el avance del proyecto. Se tenían reuniones cada cierto tiempo medida que se finalizaban un número de tareas para así corregirlas inmediatamente. También cabe recalcar que se ha trabajado con el Trello y se han generado una serie de tareas que se muestran en la tabla 1:

Tabla 1. Tareas Trello.

|   |   |
|---|---|
| T101 - Diseño de página                         | Se desarrollará el diseño de una página en Figma.<br>Se desarrollan 2 diseños<br>Debe de contener toda la info de una ruta.<br>Generalizada para poder “templatizarla”.   |
| T201 - Inicio de proyecto React                 | Se debe obtener acceso a la organización de GitHub y crear un inicio de proyecto de React en JavaScript.<br>Al entrar a esta ruta debe de renderizar una page, que tendrá, dos “containers”, cada uno con un componente “Hello World”               |
| T202 - Mocks de datos                           | Se debe de realizar un archivo JSON. El cual contenga los datos, para poder “mockear”.  |
| T203 - Maquetar componentes                     | Se debe crear secciones del diseño las cuales llamaremos “containers”, y estos tendrán componentes.<br>Se recomiendo crearlos por separado e ir añadiéndolos al proyecto así luego será como montar un puzle = separar problemas grandes en pequeño |
| T301 - Llamada a una API                        | Llamaremos a un API para leer los datos que anteriormente obteníamos mockeados.   |
| T302 - Representación de datos obtenidos de API | Una vez obtenidos los datos de la API los representaremos en su sitio.  |

## Estructura del documento

A continuación, se muestra la estructura del documento, indicando brevemente el contenido de cada capítulo.

**Capítulo 1: Introducción** En el primer capítulo se realiza una breve descripción de la motivación, de los objetivos del trabajo, la metodología empleada y la estructura del documento.

**Capítulo 2: Marco teórico. Contexto tecnológico** En el segundo capítulo se expone la actualidad de los frameworks web y se explica detalladamente el proyecto O-CITY y su evolución a lo largo del tiempo desde diferentes perspectivas.

**Capítulo 3: Análisis del problema** En el tercer capítulo se expone el problema a resolver, así como la solución que se aplicará.

**Capítulo 4: Diseño de la solución** En el cuarto capítulo se expone el diseño de la arquitectura del sistema y las tecnologías empleadas en aplicar la solución al problema planteado.

**Capítulo 5: Desarrollo de la solución propuesta** En el quinto capítulo se expone la implementación tanto del backend como del frontend del proyecto con explicaciones detalladas.

**Capítulo 6: Testeo** En este capítulo se realizará un testeo de la aplicación usando Selenium IDE para comprobar el correcto funcionamiento.

**Capítulo 7: Conclusiones** Capítulo en que se realiza la valoración final del proyecto, del impacto con las ODS y las futuras líneas de trabajo.

**Capítulo 8: Bibliografía** Para finalizar, aparecen las referencias utilizadas en el trabajo.

# Capítulo 2. Marco teórico. Contexto tecnológico

## O-CITY. Concepto



Figura 1. Logo O-CITY. Fuente: <https://o-city.webs.upv.es/>

El proyecto O-CITY (Orange: Creativity, Innovation & TechnologY) ha sido desarrollado en el marco del programa europeo Erasmus+ (Knowledge Alliance). Ha recibido una financiación de 992.472€ y su periodo de implementación se extiende desde el 1 de enero de 2019, hasta el 31 de diciembre de 2021. 13 socios de 6 países diferentes (España, Italia, Grecia, Serbia, Eslovenia y Colombia) han trabajado durante este periodo liderados por la UPV (Universitat Politècnica de València – España) [2]. En la figura 1 se observa una imagen del logo del proyecto.

### **High Concept:**

O-CITY es un atlas digital de patrimonio natural y cultural para promocionar en las ciudades del mundo el turismo responsable, la educación para el desarrollo sostenible, y la economía naranja.

### **Retos del Proyecto:**

El proyecto persigue fundamentalmente tres objetivos.

1. La transformación digital de los sectores económicos tradicionales a través de la formación de profesionales y la promoción económica de las ciudades mediante el turismo cultural y natural.
2. La introducción de herramientas educativas innovadoras a través de

planes de formación en competencias tanto profesionales como personales diseñados para facilitar el trabajo de los profesores en sus aulas.

3. El descubrimiento y la promoción de la cultura, el patrimonio, las tradiciones y el entorno natural de las ciudades del mundo.

En resumen, O-CITY es un proyecto de participación ciudadana, donde los destinos turísticos inteligentes a través de su sistema educativo, ofrecen a sus propios ciudadanos la posibilidad de aportar su punto de vista en formato multimedia, sobre el patrimonio natural y cultural de la ciudad, sin renunciar al rigor que exige la información pública (a disposición de los turistas) sobre el destino.

Además, O-CITY ofrece como valor añadido la posibilidad de exportar sus datos de patrimonio a otras aplicaciones, a través de una API. De esta manera, hay ciudades con aplicaciones para turismo que usan los datos de O-CITY, junto con los de restaurantes, hoteles, eventos, diversión, etc., para ofrecer a sus turistas una experiencia integral en el destino.

### **Herramientas Desarrolladas:**

Según se observa en la figura 2, para cumplir los tres retos descritos del proyecto, se han desarrollado dos herramientas [3]:

1. El mundo de las ciudades O-CITY (<https://o-city.org/>), una aplicación web responsive para visualizar el patrimonio natural y cultural de las ciudades. En esta aplicación (de gestión distribuida) participan diferentes actores como gestores públicos municipales, responsables universitarios, profesores, etc.
2. La plataforma educativa O-CITY, destinada a los profesores, que ofrece cursos (MOOC) distribuidos en 4 módulos de formación en materia de negocios, técnica (multimedia), cultural y habilidades blandas.

El proyecto O-CITY no podría entenderse sin estas dos herramientas operando juntas: por una parte a través de los planes formativos, estudiantes de diferentes ciclos pueden desarrollar proyectos multimedia en sus aulas, dirigidos por sus

profesores, sobre elementos de patrimonio natural y/o cultural de sus propias ciudades; y por otra parte el premio por el trabajo bien hecho es poder exhibir estos trabajos en la plataforma del mundo de las ciudades, donde todo el mundo puede verlos y aprender con ellos. Y esto último, es un elemento de motivación muy poderoso para los profesores en el aula.



Figura 2. Planeta O-CITY. Fuente: <https://doi.org/10.1145/3631121>

### Programa de Formación O-CITY:

Se ha abordado la formación de los futuros profesionales de la economía naranja desde un punto de vista integral, es decir combinando enseñanzas y aprendizajes en diferentes áreas de conocimiento y trabajando en equipo; ya que un profesional de este sector del siglo XXI, para integrarse en el mercado laboral o para crear su propia empresa, va a necesitar tanto conocimientos técnicos como conocimientos en materia de negocios, culturales y por supuesto habilidades blandas.

### Integración e interoperabilidad de datos:

En el contexto de la gestión de datos cabe decir que O-City, que es una herramienta sin ánimo de lucro, que se mueve en el ámbito de lo público (actualmente custodiada y mantenida por la UPV), ha nacido con la vocación de intercambiar datos, es decir se ofrece como un servicio público (*Open Data*) para generar valor añadido a los destinos, a las empresas de turismo profesional y las instituciones tanto en el nivel de la educación como de la investigación.

O-City en su corta vida ya tiene importantes referencias en el ámbito de la integración e interoperabilidad de datos. Sirvan los siguientes casos de aplicación en dos líneas de acción:

- **Exportación de datos** a través de una API desarrollada según norma UNE 178503 (Destinos turísticos inteligentes, Semántica aplicada al turismo).
- **Mejora de la experiencia en destino:** Proyecto experimental “Plataforma Inteligente Rs3 Ruta de la Seda CV” desarrollado por el Instituto Seda España con la colaboración de Turisme Generalitat Valenciana. Con el objetivo de mejorar la experiencia de los turistas en los destinos, este proyecto realizó la integración de datos de diferentes fuentes en una única aplicación (Base de datos Sentilo, Diputación de Valencia). Con este proyecto responsables de las oficinas de turismo y gestores públicos de ciudades como Buñol, Requena, Ribarroja, Alzira, Algemesí y Carcaixent demostraron la facilidad y usabilidad de O-City a la hora de registrar nuevos elementos patrimoniales, con sus videos promocionales adjuntos.
- **I+D+i en turismo:** Utilización de los datos de los destinos Benidorm, Gandia y Valencia para entrenamiento de algoritmos de IA con aplicación de recomendaciones turísticas. El Instituto Universitario Valenciano de Investigación en Inteligencia Artificial ([vrAIn.upv.es](http://vrAIn.upv.es)) de la UPV, utiliza los datos de O-City, en primer lugar porque son datos verificados, y en segundo lugar porque O-City dispone de un Tesoro de clasificación de patrimonio de más de 150 palabras clave.
- **Importación de datos** a través de la aplicación de turismo profesional [Infotourist.es](http://Infotourist.es), empresa que ofrece soluciones para destinos turísticos inteligentes. InfoTourist ofrece a sus clientes la posibilidad de que sus puntos de interés aparezcan en O-City para ello ha desarrollado una API que permite esta importación, de esta manera los destinos se encuentran con el valor añadido de que sus atractivos turísticos pueden ser visualizados desde cualquier parte del mundo través de O-City, haciendo uso de las herramientas de búsqueda avanzada que posee la aplicación.

En la actualidad O-City ha recibido financiación de la diputación de Valencia para la puesta en valor del patrimonio en los 266 municipios de la provincia y para la

creación y difusión de los atractivos turísticos (rutas culturales y naturales) de sus 17 comarcas haciendo uso de las herramientas de marketing de aplicación. En este proyecto han digitalizado entre 2 y 5 elementos de patrimonio natural y cultural de cada municipio, poniendo en el mapa pueblos y comarcas con graves problemas de despoblación. [4]

# O-CITY. Navegación



Figura 3. Página inicio O-CITY. Fuente: <https://o-city.org/>



Figura 4. Inicio O-CITY Zoom1. Fuente: <https://o-city.org/>

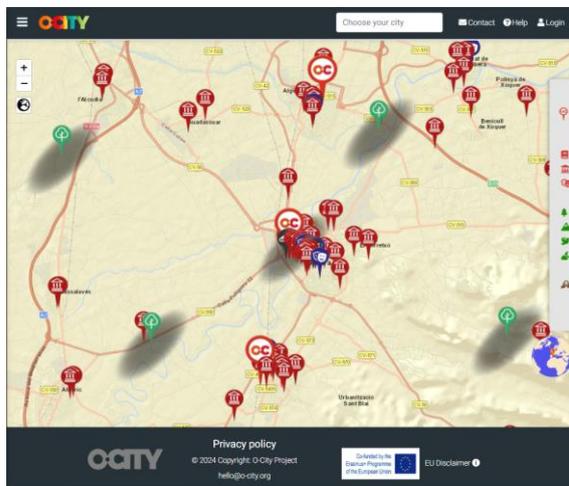


Figura 5. Visión ciudades. Fuente: <https://o-city.org/>

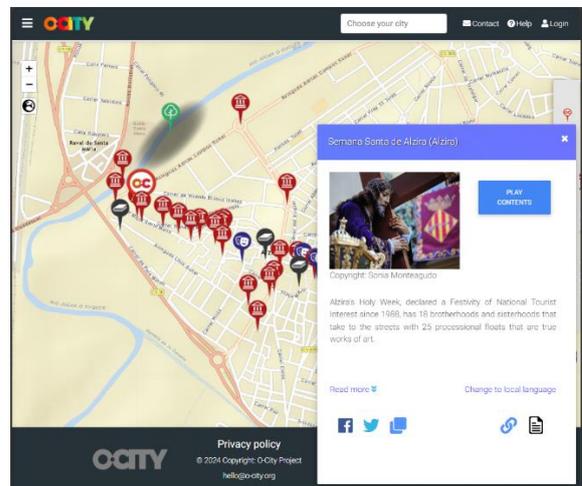


Figura 6. Visión patrimonio. Fuente: <https://o-city.org/>

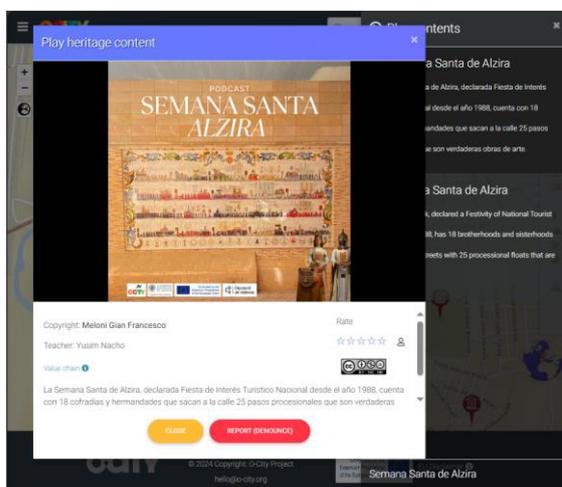


Figura 7. Play Contents patrimonio. Fuente: <https://o-city.org/>

A través de las imágenes anteriores se muestra la navegación por el mundo O-CITY, desde que se inicia hasta poder visualizar los elementos patrimoniales.

En la figura 3 se observa el mapa del mundo con todas las ciudades indicadas que han añadido elementos patrimoniales. Aparecen con números, pero al pulsarlo iremos a la figura 4. En esta se observan claramente las mismas ciudades, pero con señalizadores en cada una. En este caso se ha seleccionado España, ya que es el mejor ejemplo de todo el proyecto, dado su gran producción patrimonial.

Al pulsar la ciudad seleccionada ya podremos ver todos los elementos patrimoniales que se han añadido, pero también se observarán otros en los alrededores, lo que permite ver la gran variedad que hay y que al usuario le entren ganas y curiosidad de observar lo que le rodea. Esto último es un arma de doble filo, ya que, si bien es un motivador de la curiosidad, también complica mucho la navegación y ensombrece, en general la experiencia de usuario. Por lo que en la nueva versión se ha decidido cambiar el modelo de navegación y que sólo se visualice en el mapa el patrimonio de la ciudad seleccionada, rodeado de las ciudades más próximas.

Hay una alternativa para ir a una ciudad en concreto si ya se tiene el destino claro. En el “*header*” aparece un recuadro blanco que permite escribir su nombre y directamente poder acceder a ella, lo que mejora mucho la velocidad a la hora de buscar una ciudad.

Por último, tenemos la página del patrimonio, aquí se pueden observar varios elementos; como el de “*Play Contents*” que es para visualizar los elementos multimedia (videos, podcasts, animaciones, infografías) relacionados, la descripción corta con su respectiva traducción al idioma local, y la posibilidad de poder ver la descripción extensa. Existen además otros iconos, como el de las redes sociales para compartir el elemento en Twitter (X) y Facebook, así como un enlace directo a la navegación tipo mosaico donde se muestra la página del propio elemento patrimonial en este formato. Finalmente se puede visualizar una ficha del patrimonio en formato PDF, generado de forma dinámica, según se muestra en el anexo B.

O-CITY plantea una navegación sencilla para todo tipo de públicos donde se busca poder visitar una ciudad sin necesidad de estar allí y poder conocer toda su cultura, tradiciones e historia a través de los elementos patrimoniales. [5]

## Tecnologías de desarrollo web

Para abordar el contexto tecnológico en desarrollo web en la actualidad, es esencial considerar varios aspectos clave que incluyen tecnologías *frontend*, *backend*, herramientas de desarrollo, metodologías y tendencias emergentes.

En primer lugar, con respecto a las **tecnologías frontend** cabe decir lo siguiente:

HTML5 es la última versión de HTML que ofrece nuevas funcionalidades como etiquetas semánticas, formularios mejorados y soporte multimedia. CSS3 incluye características avanzadas como animaciones, transiciones y flexbox/grid para un diseño “responsive”. JavaScript es el lenguaje de programación esencial para la web, utilizado para crear aplicaciones interactivas.

Dentro de los **frameworks** y librerías de JavaScript, React.js, desarrollado por Facebook, es popular por su enfoque de componentes y su virtual DOM. Angular, mantenido por Google, es conocido por su arquitectura MVC (Modelo Vista Controlador) y su robustez. Vue.js es un framework progresivo que se puede integrar fácilmente en proyectos existentes, conocido por su flexibilidad y simplicidad.

En cuanto a los **preprocesadores y postprocesadores** de CSS, Sass y Less son herramientas que extienden CSS con variables, anidación y funciones, haciendo el desarrollo más eficiente. PostCSS utiliza plugins para transformar el CSS con PolyFills y otras funciones.

Una vez abordadas las tecnologías más comúnmente usadas en el frontend, es relevante mencionar las que encontramos en el **backend**:

JavaScript/Node.js permite ejecutar JavaScript en el servidor, facilitando el desarrollo full-stack con un solo lenguaje. Python, utilizado en frameworks como

Django y Flask, es conocido por su simplicidad y legibilidad. Ruby, con el popular framework Ruby on Rails, sigue el principio de “Convención sobre configuración”.

En cuanto a los **frameworks y plataformas**, Express.js es un framework minimalista para Node.js que facilita la creación de aplicaciones web y API. Django y Flask son frameworks de Python, donde Django es más completo y Flask es más ligero y flexible. Spring Boot es un framework de Java que simplifica el desarrollo de aplicaciones empresariales robustas.

Este panorama muestra que el desarrollo web es un campo dinámico y en constante evolución, impulsado por la innovación en tecnologías y metodologías que buscan mejorar la eficiencia, la escalabilidad y la experiencia de usuario. [6]

## **Evolución tecnológica de O-CITY**

La evolución tecnológica de O-CITY muestra un claro avance en la adopción de nuevas herramientas y enfoques de desarrollo. A continuación, se detallan las tecnologías utilizadas en la primera versión y la versión en desarrollo, junto con una explicación de las diferencias entre ambas:

En la primera versión de O-CITY, se utilizó el framework Angular para la construcción del frontend, empleando HTML5 y CSS para estructurar y diseñar las páginas web. MDBootstrap se integró para facilitar el desarrollo de una interfaz de usuario estilizada y responsive. Los mapas se implementaron con Leaflet y las APIs de Google Maps para geocodificación y búsqueda de lugares. La base de datos utilizada fue MySQL, y las interfaces se diseñaron para ser responsives. Los servicios web se desarrollaron en PHP, y se creó una API en PHP para el acceso externo.

En la versión en desarrollo de O-CITY, se han realizado cambios significativos en la tecnología utilizada. El framework React ha reemplazado a Angular para la construcción del frontend, utilizando HTML JSX y CSS para estructurar y diseñar las páginas web. La librería Ant Design se ha integrado para asegurar una interfaz de usuario consistente y profesional. Los mapas siguen utilizando Leaflet, pero ahora con una base de datos propia para países, estados y ciudades. La base de datos ha cambiado a PostgreSQL, y las interfaces

continúan siendo responsivos. La API para el acceso externo se ha desarrollado en TypeScript.

Las principales diferencias entre las tecnologías de la primera versión y la versión en desarrollo son notables. En la primera versión, Angular proporcionaba la estructura del frontend, mientras que en la versión en desarrollo, React ofrece una mayor flexibilidad y rendimiento. MDBootstrap en la primera versión ha sido reemplazado por Ant Design en la versión en desarrollo, mejorando la consistencia y profesionalidad de la interfaz de usuario. La implementación de mapas ha evolucionado de usar Leaflet y APIs de Google Maps a utilizar Leaflet con una base de datos propia para países, estados y ciudades. MySQL ha sido reemplazado por PostgreSQL como la base de datos, ofreciendo ventajas en la gestión y escalabilidad. Finalmente, los servicios web y la API, que inicialmente se desarrollaron en PHP, han sido sustituidos por una API desarrollada en TypeScript, lo que mejora la robustez y el mantenimiento del código.

Estas diferencias reflejan una evolución hacia tecnologías más modernas y eficientes, alineándose con las mejores prácticas del desarrollo web actual, y asegurando un sistema más robusto, escalable y de alta calidad. [7]

*Tabla 2. Evolución de tecnologías en O-CITY*

| <b>Primera versión</b>  | <b>Versión en desarrollo</b>                        |
|---|---|
| Framework: Angular  | Framework: React                                    |
| HTML5 & CSS   | HTML JSX & CSS                                      |
| MDBootstrap   | Antd library  |
| Mapas Leaflet & APIs de Google Maps (geocoding & search places) | Mapa leaflet & db propia de ccountry, sttates, city |
| BBDD MySQL  | BBDD PostgreSQL                                     |
| Interfaces Responsive   | Interfaces Responsive                               |
| Webservices en PHP  | Api para acceso externo TS                          |
| API para acceso externo PHP                                     |   |

El cambio de React a Angular es algo importante a destacar en este apartado ya que React busca una mayor eficiencia gracias a sus componentes y su virtual

DOM para el renderizado de interfaces superando las limitaciones del angular. También la librería AntDesign ofrece componentes para su uso en el propio React de manera mucho más abierta que Bootstrap ya que permite una integración mucho más personalizada y sencilla.

# Capítulo 3. Análisis del problema

Durante los últimos años, O-CITY está recibiendo un gran número de cambios que plantea modificar muchas de las ideas que se tenían en un inicio. Lo primero que cabe resaltar es el cambio de las tecnologías: anteriormente el software se realizaba de una manera y ahora se ha buscado la remodelación con nuevas tecnologías, más avanzadas para dotar de flexibilidad al proyecto. Básicamente, con el software original el proyecto estaba estancado, ya que no permitía el crecimiento adaptado a las demandas tecnológicas actuales (Inteligencia Artificial entre ellas).

Haciendo un poco de historia, el planteamiento original de O-CITY implementaba un solo modo de visualización: la navegación por el mapa. El uso de Angular como framework impedía el posicionamiento SEO de los ítems del mapa de O-CITY, tanto ciudades como elementos de patrimonio, lo que resultaba un hándicap a la hora de crecer en los destinos turísticos. Este problema, sobrevenido por la tecnología seleccionada, fue solventado a través de la creación de un segundo sistema de navegación: el modo mosaico. Este sistema creaba páginas HTML de forma estática y permitía la indexación en Google a través de archivos sitemaps.xml [8], es decir una estrategia de posicionamiento SEO. La idea original de crear estas páginas, una vez al día si se detectaban cambios en el sistema, nunca llegó a implementarse.

El cambio de tecnologías de desarrollo, de Angular a React, coincidió con un cambio de estrategia en el proceso de monetización de O-CITY. Se decidió incorporar en el proyecto la posibilidad de crear páginas específicas monetizables: “O-CITY Pages” e historias relacionadas con la cultura y el patrimonio dentro de esas páginas: “O-CITY Stories”. Estas páginas sí serían indexables en Google y la flexibilidad de React orientado a componentes permite su fácil implementación. Por ejemplo, podría existir la página de pago <https://o-city.org/Alzira>, pero manteniendo el formato inicial de O-CITY sin ánimo de lucro en <https://o-city.org/>. Obviamente la página de pago ofrecerá un valor añadido lo suficientemente alto como para ser atractivo a los posibles clientes;

introduciendo, por ejemplo, avatares animados con IA y una experiencia de usuario superior.

Llegado a este punto se planteó si mantener o no la navegación por mosaicos, ya que la indexación ahora no era motivo para crear este sistema. Sin embargo se ha decidido mantenerla, debido fundamentalmente a que este modo de visualización es más accesible para personas con ciertas limitaciones físicas, mentales o neurodiversidades, y su generación dinámica y la navegación entre mosaicos evita el uso del mapa lo que mejora de forma colateral la accesibilidad tecnológica.

## **Solución propuesta. Especificaciones**

Básicamente el objetivo del proyecto es conseguir la misma estructura de navegación por mosaicos que se tenía en la anterior versión. Se buscará un frontend similar, pero a través de un backend diferente.

La solución que se ha propuesto para resolver el problema detallado anteriormente se hará de la siguiente manera. Para realizar este producto dentro del proyecto O-CITY se busca una estructura determinada donde las primeras páginas van a ser mosaicos navegables hasta llegar a un elemento patrimonial como destino final. En este camino vamos a encontrar cuatro mosaicos diferentes:

- Países: aparecerán todos los países que se encuentran en la base de datos de O-CITY para poder seleccionar el que nosotros queramos e investigar alrededor del mundo.
- Estados: cada país tiene unas divisiones, en España por ejemplo podemos encontrar la división en provincias: Castellón, Valencia, Alicante, etc.
- Ciudades: dentro de cada estado aparecen todas las ciudades dadas de alta en O-CITY.
- Elementos de patrimonio: esta página es algo diferente al resto ya que el mosaico de los patrimonios presenta novedades con respecto a las anteriores. En la cabecera aparece la descripción que se ha proporcionado de la ciudad, así como un botón para cambiar del inglés al idioma local. Evidentemente, aquí se muestran todos los elementos

patrimoniales de la ciudad, con sus respectivas imágenes para que sea mucho más visual, y el usuario pueda acceder de manera rápida y sencilla.

Una vez se ha navegado por todos estos mosaicos nos encontramos en la página del patrimonio. En esta se muestra el nombre y la ciudad a la que pertenece. Además de una fotografía y su descripción larga donde se explica toda la historia de ese elemento patrimonial, inicialmente se encontrará en inglés, pero con un botón, como en el último mosaico, es posible cambiar de idioma a su idioma local. Es decir, si buscamos un patrimonio ubicado en España, el texto se podrá leer en inglés y en español. Pero si la ubicación está en Francia se leerá en inglés y francés. Se busca con esto promover los idiomas locales de todos los países ya que es un elemento cultural indispensable en cada estado.

Se proponen también como solución una serie de funcionalidades importantes para la mejorar la experiencia de usuario:

1. **“View on O-CITY Map”**, este botón hace que se pueda ubicar el patrimonio en el mapa para ver su localización exacta.
2. **Las redes sociales**, estos permiten compartir con todas las amistades del usuario si la información le ha resultado interesante.
3. **Los links de interés**: para que el usuario pueda obtener información adicional accediendo a enlaces de YouTube o a páginas del ayuntamiento con explicaciones detalladas.

Para finalizar, en el pie de página se incorporan elementos de diseño de marca y contacto, como el logo de O-CITY y su correo corporativo; y en la cabecera de la página, un elemento imprescindible para la mejora de la experiencia de usuario en la navegación a través del sistema de mosaicos: **las migas de pan**. Esto se utiliza para ver el rastro seguido y así volver directamente al mosaico que queremos. Por último y no menos importante en el pie de página se define que tipo de patrimonio es, si es tangible, intangible, natural o cultural, etc.

# Capítulo 4. Diseño de la solución

Una vez fijadas las especificaciones, en este capítulo se resumen los puntos fundamentales del diseño de la solución. En primer lugar, se expone la arquitectura del sistema y se realiza una comparativa con la arquitectura de la primera versión del software. En segundo lugar, se expone el trabajo realizado para el diseño del frontend y finalmente se realiza un breve resumen de las tecnologías utilizadas.

Hay que hacer notar que, aunque el frontend de ambas versiones sea prácticamente igual por especificación del “Product Owner”, la arquitectura del backend cambia radicalmente, como ya se ha comentado en el punto anterior.

## Arquitectura del sistema

La figura 8 representa la primera versión de la navegación por mosaicos, donde se observa que la información recogida de la página de O-CITY se almacena en la base de datos. Esta información de la base de datos es utilizada en un fichero PHP que realiza la conexión y genera todas las carpetas, de manera estática, de todos los mosaicos, y de todas las páginas de los elementos patrimoniales.

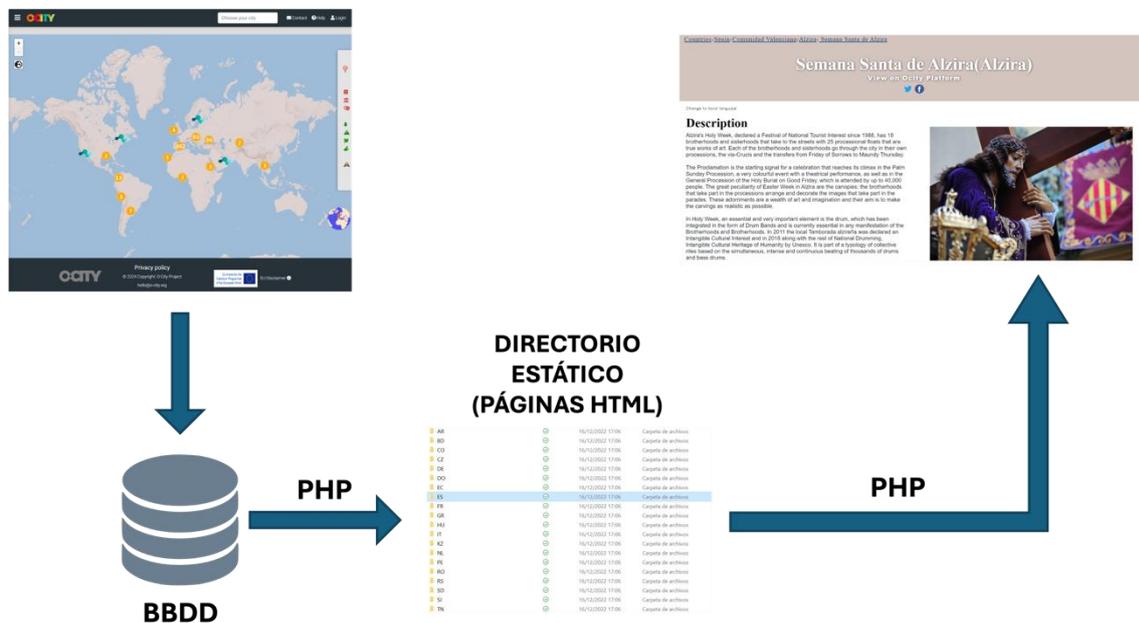


Figura 8. Arquitectura versión primera. Fuente: Recurso propio

A continuación, se muestran en la figura 9 los cambios introducidos en la arquitectura actual. Igual que en la primera versión, los usuarios introducen toda la información sobre cualquier patrimonio en la aplicación de O-CITY en la que se requerirá una información básica. Después de este proceso esta información se almacenará en la base de datos general del proyecto.

Posteriormente se han generado una serie de APIs para poder trabajar con ellas sin necesidad de acceder a la base de datos directamente. Por lo tanto, la información se modificará en esas APIs para que en el proceso final se pueda mostrar de forma dinámica las páginas generadas para este proyecto.

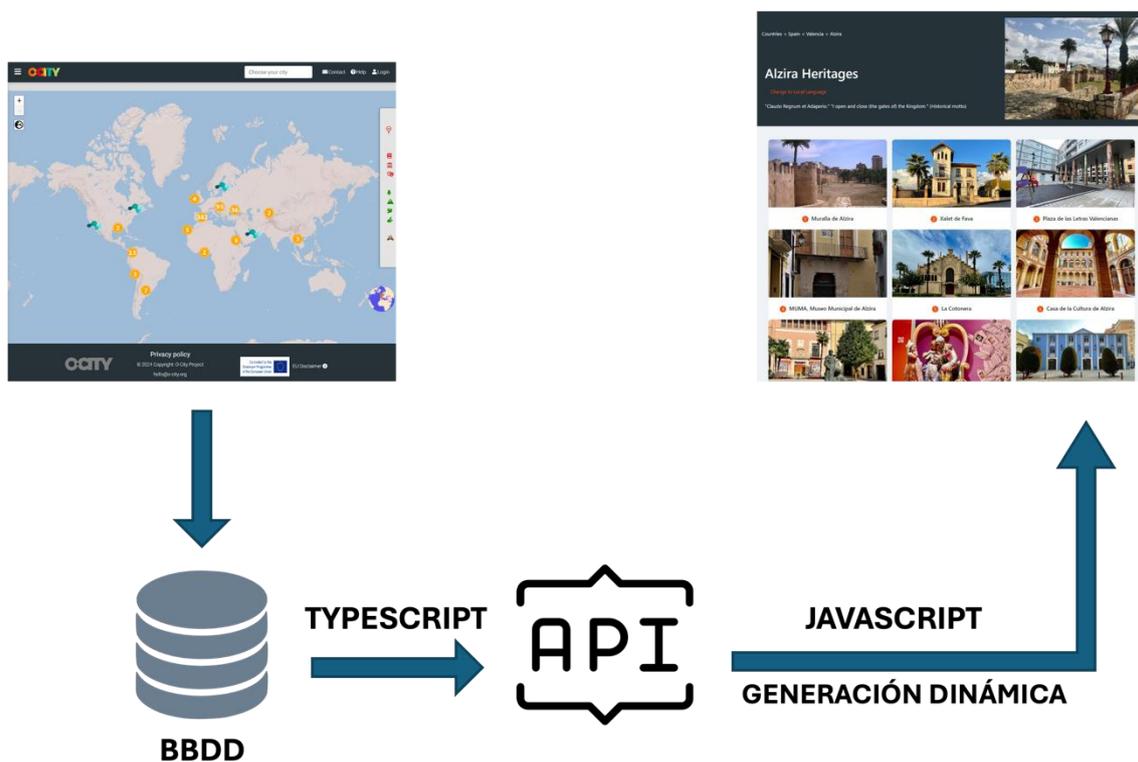


Figura 9. Arquitectura actual. Fuente: Recurso propio

## Diseño del frontend

Al inicio del proyecto se planteó un diseño del frontend de la página del elemento patrimonial, en ella aparecía básicamente los mismos elementos que los del diseño realizados en la primera versión. No se plantearon diseños alternativos

para las páginas de mosaicos por prescripción de la organización que optó por mantener directamente los diseños de la primera versión.

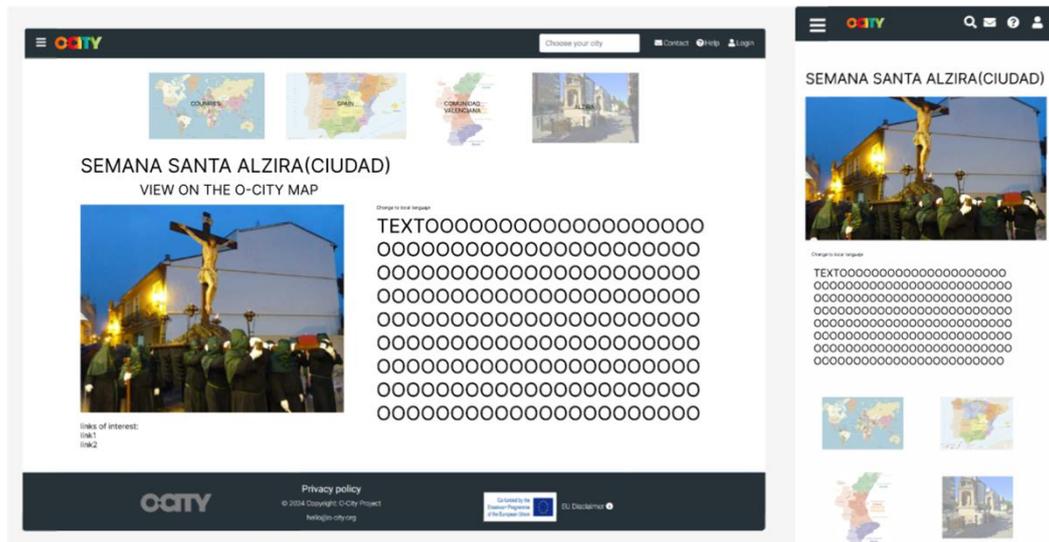


Figura 10. Propuesta de diseño frontend. Fuente: Recurso propio

La figura 10 muestra el primer planteamiento para las páginas de patrimonio. Se buscaba una manera más dinámica y visual de mostrar las migas de pan a través las imágenes propias de cada uno de sus elementos en vez del método tradicional. También se realizó el diseño para móvil que se muestra en la misma figura, donde se observa que las migas de pan aparecen al final, tras la información para así ampliar las posibilidades de navegación del usuario.

La organización finalmente descartó este diseño y optó por continuar con el diseño de la primera versión, ya que las migas de pan clásicas son mucho más intuitivas para el usuario y descargan a los administradores de labores de gestión de imágenes.

## Tecnología utilizada

En este proyecto, se ha decidido utilizar una arquitectura moderna y eficiente para el desarrollo del sistema, seleccionando herramientas y tecnologías que se complementan y optimizan el proceso de desarrollo. El frontend se desarrollará en JavaScript, incorporando la librería de AntDesign para asegurar una interfaz de usuario consistente y profesional.

React es el framework determinado para el desarrollo del frontend, ya que facilita la construcción de interfaces de usuario interactivas y dinámicas. Algunas razones para esta elección son su capacidad de uso de componentes, que permite crear componentes reutilizables facilitando el mantenimiento y la escalabilidad del código; el uso del DOM virtual, que mejora el rendimiento y la eficiencia; y una robusta comunidad de desarrolladores junto con un extenso ecosistema de paquetes y herramientas complementarias como Redux o React Router, que simplifican el desarrollo de aplicaciones.

El backend empleará ciertas APIs para obtener los datos de la base de datos. La librería Axios se ha utilizado para realizar las llamadas a estas APIs, facilitando la lectura y gestión de datos.

Ant Design ha sido incorporado para la interfaz de usuario, proporcionando una amplia gama de componentes con diferentes estilos y listos para usar. Las razones para esta elección incluyen la consistencia que aporta al diseño, mejorando la experiencia del usuario; la productividad, ya que la disponibilidad de componentes preconstruidos acelera el desarrollo permitiendo enfocarse más en la lógica de negocio; y la extensa documentación y comunidad activa, que hacen que sea fácil de aprender e implementar.

La integración de estas tecnologías permite un desarrollo ágil y eficiente, aprovechando las ventajas de cada una: React en el frontend facilita la creación de interfaces de usuario dinámicas y reutilizables; las APIs proporcionan una manera eficaz de gestionar y obtener datos; y Ant Design asegura una UI coherente y profesional con menor esfuerzo de diseño. En conjunto, estas tecnologías permiten desarrollar un sistema robusto, escalable y de alta calidad, alineado con las mejores prácticas del desarrollo web moderno.

# Capítulo 5. Desarrollo de solución propuesta

## Backend

Este apartado se centrará en hablar de las APIs utilizadas, así como de la lectura de estas. [9]

Para realizar este proyecto se han utilizado cuatro APIs diferentes, cada una con una estructura determinada y con unos datos en concreto.

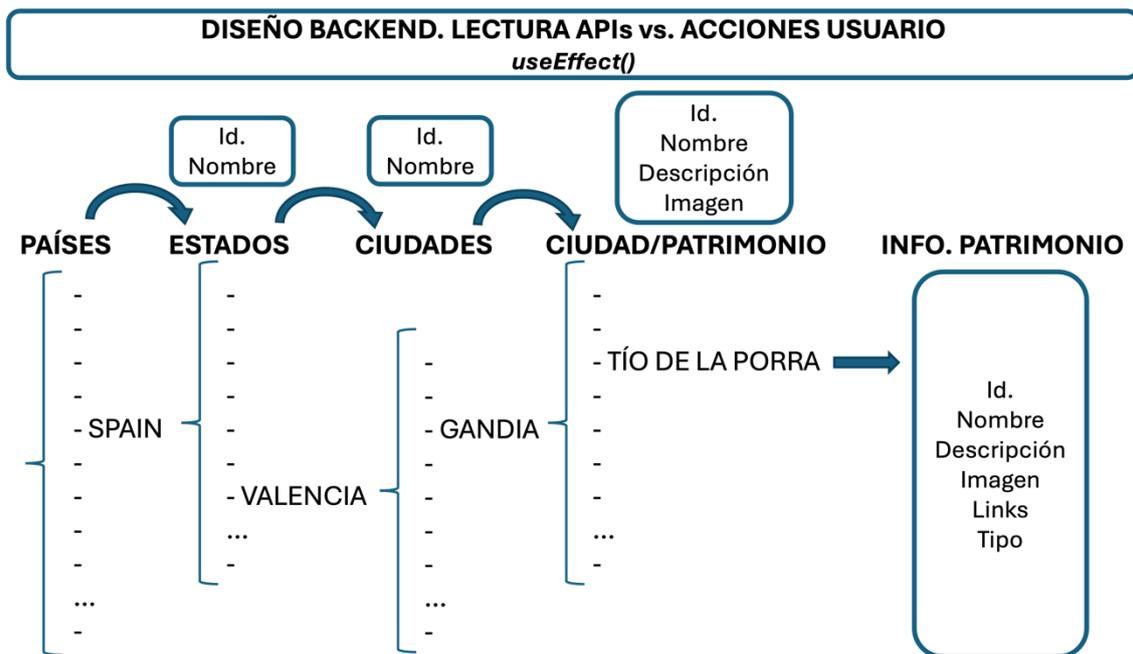


Figura 11. Arquitectura Backend. Fuente: recurso propio.

La figura anterior muestra la arquitectura que se ha diseñado en el backend de la aplicación. En este diseño se muestra el proceso de lectura de las cuatro APIs involucradas, en respuesta a las acciones del usuario, es decir las selecciones realizadas a la vista de los mosaicos ofrecidos. Estas selecciones son requeridas en todos los casos salvo en el mosaico de países que aparecen todos de manera predeterminada como punto de inicio de la cadena de selecciones.

En primer lugar, se utiliza el **API de países** cuya estructura se muestra en la primera sección del anexo A.

De este API podremos extraer todos los países en el mosaico inicial. Esta petición la haremos utilizando la función de efecto “useEffect” (figura 12), que asegura que al montar el componente, se haga la solicitud a la API a través de un HTTP GET. En la API aparecen una gran cantidad de datos interesantes del país como bien puede ser de la moneda, código telefónico, las coordenadas, etc. Pero para esta aplicación se ha utilizado exclusivamente el id de cada país y su nombre para poder mostrarlo en pantalla.

```
useEffect(() => {  
  axios.get(`https://api.test-ocity.icu/api/country`)  
    .then((response) => setCountries(response.data));  
}, []);
```

Figura 12. Ejemplo de lectura del API. Fuente: Recurso propio

Este mismo proceso se realiza de manera consecutiva en el resto de las APIs: estados, ciudades y elementos patrimoniales, descritas en el anexo A. La única diferencia respecto al de países es que la llamada a la función useEffect se realiza con el “id” del elemento seleccionado por el usuario según se muestra en la figura 13.

```
useEffect(() => {  
  axios.get(`https://api.test-ocity.icu/api/State`)  
    .then((response) => {  
      const filteredStates = response.data.filter(state => state.country_id === parseInt(idCountry));  
      setStates(filteredStates);  
    });  
}, [idCountry]);
```

Figura 13. Ejemplo filtrado de id. Fuente: recurso propio

Por último, ya obtenemos la página de los patrimonios donde aquí se volverá a leer el API 4 del anexo A. De esta última lectura se extraerán el conjunto de datos necesarios para la descripción del elemento patrimonial. Se extraerán datos como:

- el nombre del patrimonio,
- la ciudad,
- el estado,
- el país,

- las descripciones en dos idiomas disponibles,
- los links de interés si se han incorporado,
- el tipo de patrimonio que es (heritageField.name) y
- la imagen.

Todo esto genera una página con toda la información necesaria sobre el elemento patrimonial seleccionado por el usuario.

Para reducir el número de llamadas al API se realiza una primera llamada desde la página del index.js del patrimonio, se almacenan los datos y posteriormente se transfieren de un componente a otro. Un ejemplo de cómo hacerlo sería: una vez hecha la primera llamada al API, se añade el componente y se le envían los datos, es decir, si un componente es header sería: `<Header data={heritage}/>` siendo “heritage” los datos del API que posteriormente se utilizarán como en las anteriores ocasiones.

```
return (  
  <div className="App">  
    <Header data={heritage} />  
    <Links data={heritage} />  
  </div>  
)
```

*Figura 14. Ejemplo del uso de datos entre componentes. Fuente: recurso propio*

Se puede observar que obtenemos y filtramos los datos de una manera muy metódica ya que esta permite tener un código sencillo y funcional. Gracias a las APIs se ha podido trabajar de manera cómoda y escalable en el caso de que hubiera grandes modificaciones en la base de datos general del proyecto.

## Frontend

Para realizar la estructura del frontend del proyecto se han creado las rutas de navegación con la función “react-router-dom”. Esta es una librería muy popular en React que se utiliza para manejar el enrutamiento en aplicaciones web. Proporciona componentes y funciones que estas facilitan la implementación y gestión de las rutas de navegación. En este caso como se ha explicado

anteriormente la ruta está definida a través de la estructura visual que se muestra en la siguiente figura:



Figura 15. Estructura de la navegación de la página. Fuente: recurso propio

En este diagrama de bloques se puede observar la navegación de la página. Hay que tener en cuenta que las migas de pan permiten que se pueda acceder a todos los mosaicos de la ruta. Por lo tanto, queda así una estructura jerárquica donde desde la página del patrimonio se puede acceder a cualquier mosaico previo.

Lo primero que se visualizará al ejecutar el proyecto de React es lo siguiente.

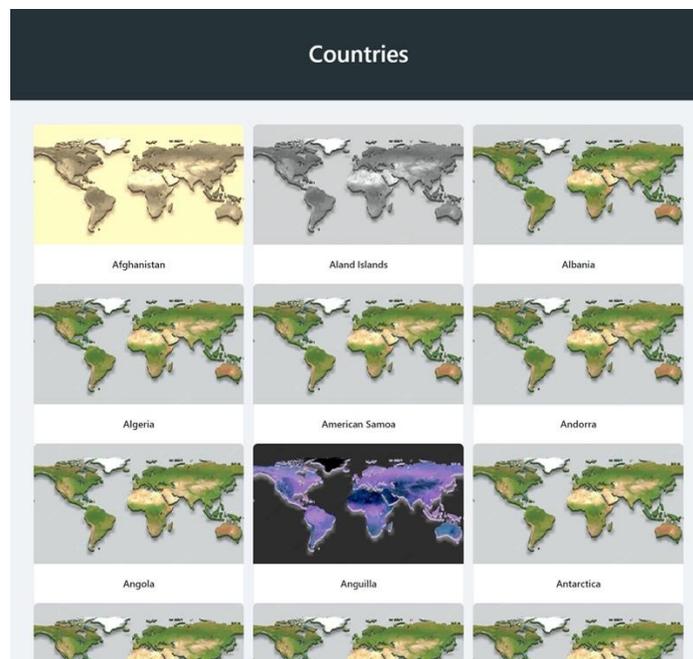


Figura 16. Mosaico Países. Fuente: recurso propio

Las imágenes que aparecen tanto en los mosaicos de países como de regiones se han generado internamente ya que no aparecen en la base de datos. En el API aparece el código de la bandera del país, pero estas no se utilizan ya que pueden ser un motivo de enfrentamiento político. Por este mismo motivo en el

mapa que aparece en la página principal de [O-CITY](#) no hay ningún tipo de fronteras.

Lo primero que se ha de configurar en el proyecto son las rutas (“path”). Éstas permitirán que todo el proyecto tenga una estructura como una gestión de carpetas, pero en lugar de crear carpetas estáticas, como en la primera versión del software, se generará todo de manera dinámica. Por ejemplo, la ruta `country/207/state/1175` dirigirá al usuario a la visualización del mosaico de las ciudades de la provincia de valencia,

Estas rutas quedarán configuradas en un archivo `index.jsx`, este tipo de archivos son propios de React. Es una librería de JavaScript que permite escribir código similar al HTML. La primera ruta configurada es la que muestra el mosaico Countries que aparece el ejemplo de cómo realizar el código en la figura 17.

```
function DashboardRoutes() {  
  return (  
    <Routes>  
      <Route exact path="" element={ <MosaicoCountries/> } />  
    )  
  }  
}
```

Figura 17. Inicio de la ruta del proyecto. Fuente: recurso propio

Una vez mostrado el mosaico, cada uno de los elementos que aparecen en él (en este caso son todos los países) tendrán asociada una función para que al hacer click sobre ellos nos lleven a sus respectivos estados. Se utilizará la función `handleCardClick` de la figura 18, y todos los mosaicos la utilizarán con su respectiva navegación. La función utiliza la función `navigate` de la librería de `react-router-dom`. Esta al ser ejecutada dirige al usuario a la ruta que está especificada.

```
const handleCardClick = (id, name) => {  
  navigate(`/country/${id}`, { state: { countryName: name } });  
};
```

Figura 18. Selección de elementos en el mosaico. Fuente: recurso propio

Para mostrar cada uno de los elementos se realizará un mapeo de los datos extraídos del API. Se recogerá el nombre y el id de cada uno y se mostrarán en diferentes tarjetas usando la librería de `antDesign`. Es un proceso metódico

realizado en cada uno de los mosaicos, se puede observar un ejemplo en la figura 19.

```
{countries.map((item) => (  
  <Col span={8} key={item.id}>  
    <Card  
      hoverable  
      onClick={() => handleCardClick(item.id, item.name)}  
      cover={<div className='image-container'><img onError={handleErrorImage} alt={item.name} src={`https://o-city.org/manifestations_me  
    >  
    <Meta title={item.name} className='meta-title' />  
  </Col>  
)
```

Figura 19. Muestra de datos de los mosaicos. Fuente: recurso propio

Una vez introducido estos procedimientos, se modificará el archivo index.jsx para establecer todas las rutas de inicio al final y que componente se ha de mostrar en cada una de ellas. Se puede observar en la figura 20 este archivo al completo.

```
{ <Route  
  path="/country/:idCountry"  
  element={  
    <MosaicoStates/>  
  }  
}</Route>  
{ <Route  
  path="/country/:idCountry/state/:idState"  
  element={<MosaicoCities/> }  
}</Route>  
{ <Route  
  path="/country/:idCountry/state/:idState/city/:idCity"  
  element={<MosaicoHeritage/> }  
}</Route>  
<Route  
  path="/country/:idCountry/state/:idState/city/:idCity/heritage/:idHerritage"  
  element={<HeritageIndividual/> }  
</Route>  
</Routes>
```

Figura 20. Ruta del proyecto entero codificada. Fuente: recurso propio

Se ha buscado una visualización sencilla de las páginas de los mosaicos. Únicamente se muestra una cabecera y sus respectivos elementos. La organización planteó una modificación en uno de los mosaicos. Se decidió que fuera el mosaico de cada ciudad con sus patrimonios ya que permite que cada ciudad sea única y tenga tanto la foto de esta como una breve descripción que añada el usuario. Como novedad respecto a la primera versión se ha añadido la enumeración de los elementos patrimoniales, para en un futuro proyecto dentro de O-CITY, crear los mosaicos de las rutas turísticas (históricas, naturales, culturales, etc.); como podrían ser la ruta de la seda, el camino del Santo Grial o

la ruta del cacao. En este caso sí es importante el orden aparición de los elementos patrimoniales, por esta razón se ha decidido implementar esta funcionalidad en este mosaico.

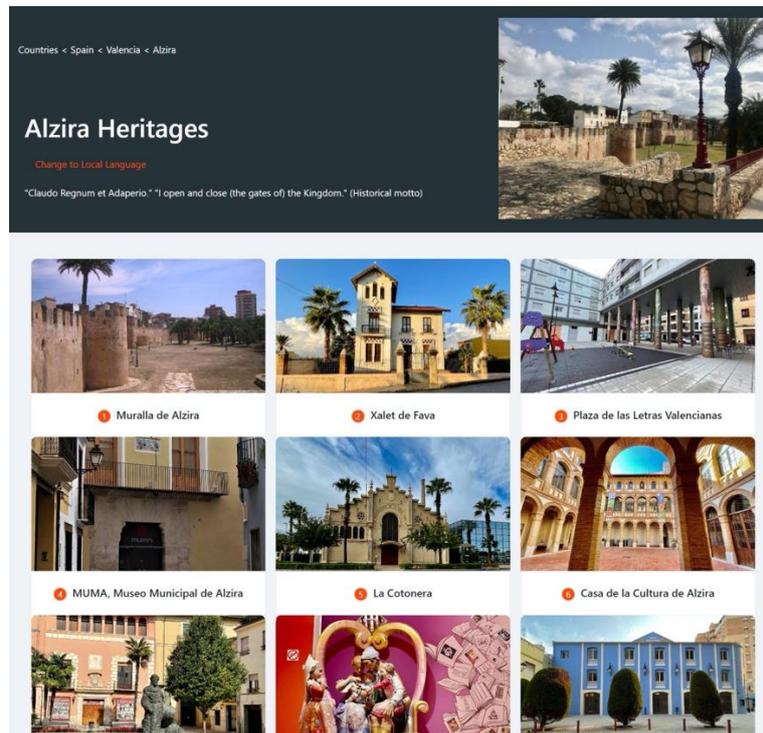


Figura 21. Mosaico de elementos patrimoniales de Alzira. Fuente: recurso propio

Lo último que cabe destacar de esta sección de mosaicos es la generación de migas de pan en todos ellos. Ya que permitirá la navegación entre ellos siguiendo el método tradicional de separar estos elementos con el símbolo "<".

Como ejemplo se van a añadir los mosaicos de las ciudades y los estados para observar el correcto funcionamiento de las migas de pan:

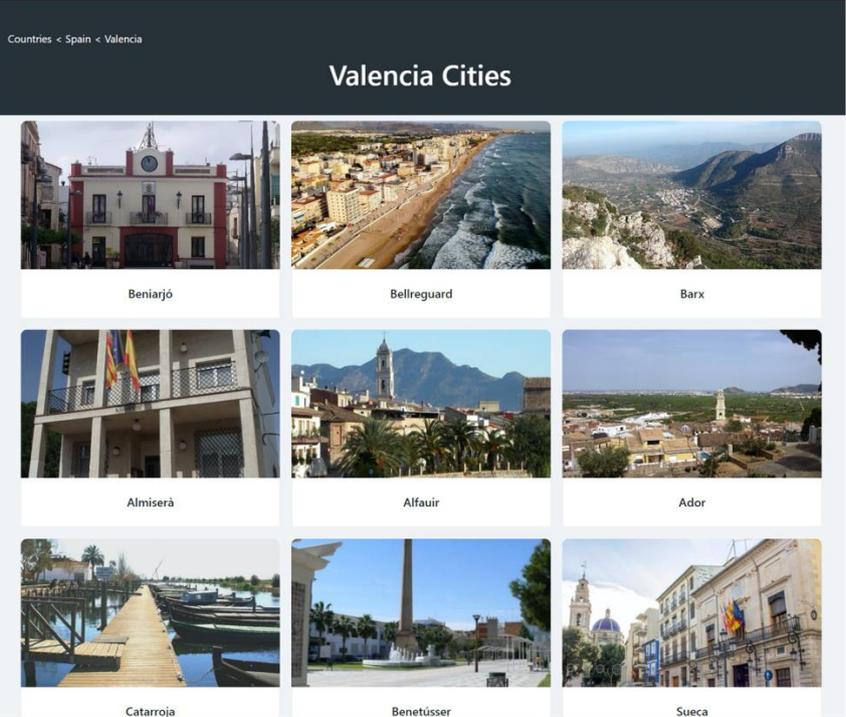


Figura 22. Mosaico de ciudades en Valencia. Fuente: recurso propio

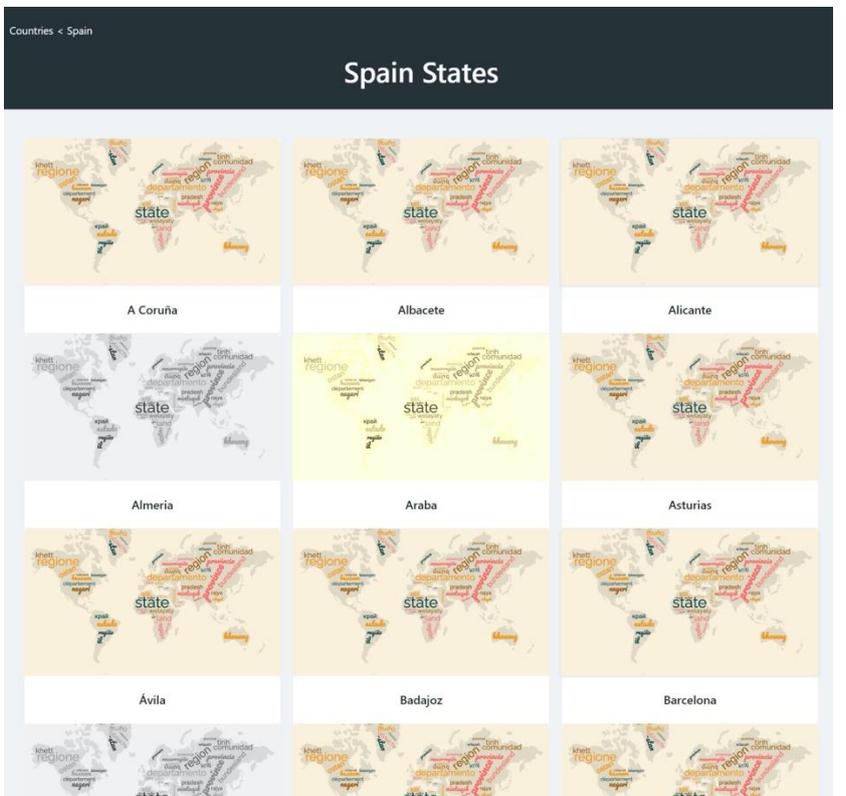


Figura 23. Mosaico de estados en España. Fuente: recurso propio

Para finalizar esta sección, una vez descritos los mosaicos, la página de patrimonio (“Heritage”) cierra la cadena de navegación. Para realizar esta página se han creado varios componentes para que permita un código más limpio y estructurado.

Aparecen 3 componentes en la página: header, links y description. Como bien indican sus nombres cada uno tiene una funcionalidad para mejorar el código y que sea mucho más limpio. Los componentes son:

1. El componente header corresponde a la parte superior de la pantalla, en este caso se mostraría el nombre del patrimonio, la ciudad, el ‘View on OCITY map’, las redes sociales y por último las migas de pan.
2. El componente description corresponde a la parte central de la pantalla, muestra la imagen del elemento patrimonial y su descripción. Todo ello junto al botón donde permite cambiar el idioma al local.
3. El componente Links corresponde a la parte posterior de la pantalla, aparecen los links de interés e información adicional como el logo de OCITY y su correo corporativo y finalmente que tipo de patrimonio es.

La llamada en el código a estos componentes aparece en la [Figura 14. Ejemplo del uso de datos entre componentes. Fuente: recurso propio.](#)

Aquí se puede observar un ejemplo de cómo se vería la página del patrimonio completa uniendo todos los componentes:



Figura 24. Página de los elementos patrimoniales. Fuente: recurso propio

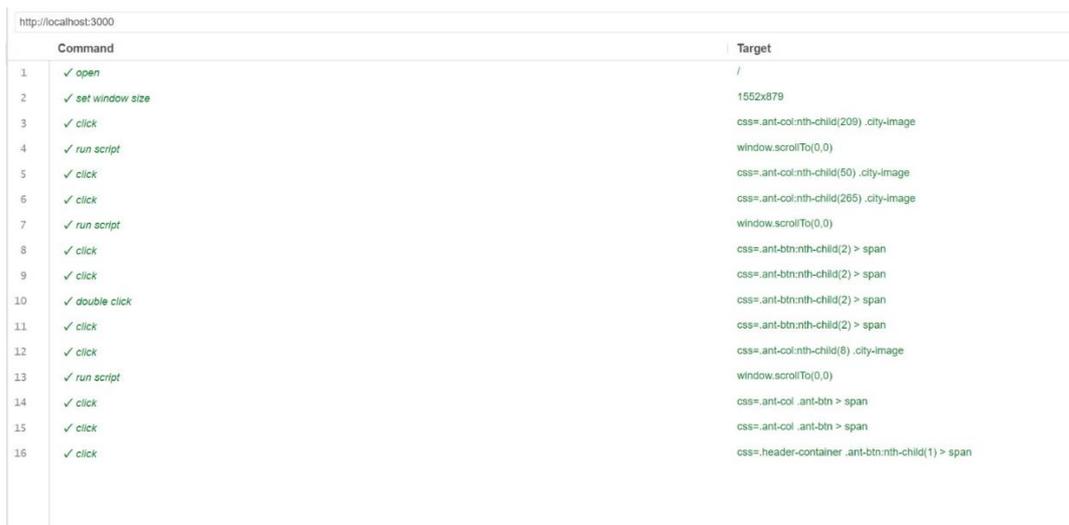
El botón “*Change to local language*” cambiaría el idioma del texto en este caso al castellano ya que es el idioma local del patrimonio donde se encuentra. Se ha optado por un diseño sencillo y funcional para que cualquier tipo de público sin importar la edad.

Podemos observar que el frontend se ha desarrollado todo utilizando la herramienta AntDesign [10] y todos sus componentes de librería son utilizados como el Flex, botones, filas, columnas, enlaces, etc. El diseño está totalmente adaptado para todas las plataformas para que, en formato móvil, tablet y ordenadores se pueda visualizar correctamente sin ningún y bien ordenado.

En este capítulo se ha explicado toda la navegación de las páginas, así como el método utilizado para realizarla correctamente utilizando las librerías que ofrece React. También se ha hablado sobre los componentes y la información que contiene cada uno y la importancia de estos para un código mucho más limpio.

# Capítulo 6. Testeo

Se ha realizado un testeo utilizando la extensión de Google Selenium IDE. Esta permite realizar una demostración de la página web para posteriormente comprobar que todo funciona correctamente. En la figura 25 se puede comprobar como todas las funcionalidades desde los cambios de idioma hasta las migas de pan han sido verificadas, por lo tanto, el proyecto cumple los requisitos de funcionalidad especificados.



| Command             | Target   |
|---------------------|--|
| 1 ✓ open            | /  |
| 2 ✓ set window size | 1552x879   |
| 3 ✓ click           | css=.ant-col:nth-child(209) .city-image            |
| 4 ✓ run script      | window.scrollTo(0,0)                               |
| 5 ✓ click           | css=.ant-col:nth-child(50) .city-image             |
| 6 ✓ click           | css=.ant-col:nth-child(265) .city-image            |
| 7 ✓ run script      | window.scrollTo(0,0)                               |
| 8 ✓ click           | css=.ant-btn:nth-child(2) > span                   |
| 9 ✓ click           | css=.ant-btn:nth-child(2) > span                   |
| 10 ✓ double click   | css=.ant-btn:nth-child(2) > span                   |
| 11 ✓ click          | css=.ant-btn:nth-child(2) > span                   |
| 12 ✓ click          | css=.ant-col:nth-child(6) .city-image              |
| 13 ✓ run script     | window.scrollTo(0,0)                               |
| 14 ✓ click          | css=.ant-col .ant-btn > span                       |
| 15 ✓ click          | css=.ant-col .ant-btn > span                       |
| 16 ✓ click          | css=.header-container .ant-btn:nth-child(1) > span |

Figura 25. Testeo con Selenium IDE. Fuente: recurso propio

# Capítulo 7. Conclusiones

En este proyecto, realizado en el marco de O-CITY, se ha desarrollado una aplicación web para visualizar el patrimonio, con toda su información, y navegar a través del mundo O-CITY en formato mosaico. Se han alcanzado todas las metas inicialmente planteadas. El proyecto ha sido testeado con Selenium IDE y revisado por el “Product Owner” hasta conseguir el resultado esperado, sobre todo respecto a la generación de las páginas de manera dinámica sin necesidad de crear archivos HTML. El proyecto aún no ha sido incluido dentro de la plataforma O-CITY, pero se espera que en las próximas semanas se pueda ver todo este resultado para el uso público. Se ha publicado el código en un repositorio de github en el siguiente enlace: [Jorgius99/proyecto \(github.com\)](https://github.com/Jorgius99/proyecto)

Hay que hacer notar que este proyecto incorpora de forma implícita dos versiones de software para conseguir el mismo objetivo: la navegación por mosaicos. La primera versión realizada por mí hace dos años no se expone de forma explícita, pero sí se muestra su arquitectura y se utiliza en las comparativas. A nivel personal, la refactorización del software ha supuesto una experiencia de aprendizaje muy útil, gratificante y conectada con el mundo real.

El proyecto, como inicialmente se había planteado, se ha realizado en React utilizando JavaScript; el frontend ha sido programado usando la librería AntDesign, que permite perfectamente incluirlo en el proyecto actual de O-CITY sin ningún problema ya que está todo estructurado en componentes.

Todo el proyecto tanto desde los inicios en la investigación hasta en la codificación de este han sido un gran impulso a todo lo enseñado en el grado, ha mejorado las capacidades de aprendizaje y se ha planteado un reto complicado el cual he tenido que hacer varias versiones para aprovechar al máximo todas las herramientas de React.

Estoy seguro de que todos los conocimientos adquiridos y la capacidad de aprendizaje las aprovecharé al máximo en mi desarrollo y futuro profesional.

## Objetivos de desarrollo sostenible (ODS)

**Objetivo O-CITY 1:** La transformación digital de los sectores económicos tradicionales a través de la formación de profesionales y la promoción económica de las ciudades mediante el turismo cultural y natural.

Vinculación con **ODS 8:** Trabajo decente y crecimiento económico

El ODS 8 tiene como meta promover el crecimiento económico inclusivo y sostenible, el empleo pleno y productivo, y el trabajo decente para todos. La transformación digital de los sectores económicos tradicionales implica modernizar y optimizar procesos mediante la tecnología, lo que puede aumentar la productividad y la competitividad de las ciudades. Además, la promoción económica a través del turismo cultural y natural crea nuevas oportunidades de empleo, fomenta el emprendimiento y apoya el desarrollo de las comunidades locales. Esto contribuye a un crecimiento económico sostenible y a la creación de trabajos dignos.

**Objetivo O-CITY 2:** La introducción de herramientas educativas innovadoras a través de planes de formación en competencias tanto profesionales como personales diseñados para facilitar el trabajo de los profesores en sus aulas.

Vinculación con **ODS 4:** Educación de calidad

El ODS 4 busca garantizar una educación inclusiva y equitativa de calidad y promover oportunidades de aprendizaje durante toda la vida para todos. La introducción de herramientas educativas innovadoras y la formación en competencias profesionales y personales alinean directamente con este objetivo, ya que mejoran la calidad de la educación y facilitan el trabajo de los profesores. Al proporcionar recursos y programas de formación, se empodera a los docentes y se mejora la experiencia educativa de los estudiantes, fomentando así un aprendizaje más efectivo y accesible.

**Objetivo O-CITY 3:** El descubrimiento y la promoción de la cultura, el patrimonio, las tradiciones y el entorno natural de las ciudades del mundo.

Vinculación con **ODS 11:** Ciudades y comunidades sostenibles

El ODS 11 pretende hacer que las ciudades y los asentamientos humanos sean inclusivos, seguros, resilientes y sostenibles. Promover la cultura, el patrimonio, las tradiciones y el entorno natural contribuye a la sostenibilidad de las ciudades al valorar y preservar su identidad y herencia. Esto no solo atrae el turismo cultural y natural, sino que también fomenta un sentido de pertenencia y orgullo entre los residentes. La conservación y promoción de estos elementos ayudan a mantener el equilibrio entre el desarrollo urbano y la preservación de recursos naturales y culturales, lo cual es esencial para la sostenibilidad a largo plazo de las comunidades.

Finalmente, cabe destacar sobre el desarrollo específico realizado en este proyecto que impacta especialmente en el objetivo 3 de O-CITY aunque de alguna manera de forma colateral influye en los objetivos 1 y 2. [11]

## **Futuras líneas de trabajo**

En primer lugar resulta evidente que el diseño gráfico de la página es claramente mejorable. Mi carencia de aptitudes en esta área ha supuesto la realización de un esfuerzo extra para alcanzar los resultados expuestos, pero el uso profesional de las páginas requiere la intervención de personas con capacidades. Este trabajo requeriría modificar el CSS y retocar los componentes.

Otro aspecto que resulta evidente es la generación de imágenes en las páginas de que muestran los mosaicos de países y estados. Actualmente se ha optado por una imagen estática modificada gráficamente, pero se requiere un cambio, otra solución que no pase por el uso de banderas para evitar enfrentamientos. La idea de la organización es apoyarse en generación de imágenes con enseñas culturales o naturales, a través de IA, supervisadas por los administradores de la plataforma, embajadores de cada país.

Las rutas de O-CITY es claramente otra línea de acción que se puede incorporar al proyecto de mosaicos. En la versión primera del software no se realizó este tipo de visualización de las rutas de O-CITY porque era un proyecto incipiente entonces, se estaba depurando, y ahora ha sucedido lo mismo, aún no se ha realizado el trasvase de datos a la nueva base de datos. Su incorporación sería inmediata emulando una página de ciudad (figura 19). Simplemente habría que

leer de una nueva API creada para ad-hoc las rutas. La numeración de los elementos patrimoniales se ha incorporado en esta versión pensando en esta futura implementación.

Por último y no menos importante, dentro de la nueva estrategia de monetización de O-CITY: “Pages y Stories” es posible también realizar páginas de mosaicos, en este caso añadiendo un avatar que vaya explicando cada uno de los patrimonios para que el usuario tenga una visita guiada por un experto. Estos avatares serán utilizados en la versión del mapa y se podrían transferir a la visualización por mosaicos para mejorar la accesibilidad de O-CITY.

# Capítulo 8. Bibliografía

[1] Schwaber, Ken, and Jeff Sutherland. "The scrum guide." Scrum Alliance 21.1 (2011): 1-38

[2] José Marín-Roig Ramón. Tienes una carta de O-CITY. (2024) [en línea] Disponible en: <https://ocity.webs.upv.es/en/2021/07/12/tienes-una-carta-de-o-city/>

[3] Asun Pérez-Pascual, Jose Luis Giménez-López, Daniel Palacio, and José Marín-Roig. 2024. O-City: Implementation of an Innovative Multimedia Platform for Promoting Orange Economy. J. Comput. Cult. Herit. 17, 1, Article 1 (March 2024), 15 pages. <https://doi.org/10.1145/3631121>

[4] José Marín-Roig Ramón. El documento "R2.1 Creative cities\_ culture for development". (2024) [en línea] Disponible en: <https://o-city.webs.upv.es/en/reports/>

[5] O-CITY. (2024) [en línea] Disponible en: <https://o-city.org/>

[6] OpenAI (2024). Chat GPT (May 24 version) [Estoy realizando mi trabajo de final de grado y me gustaría obtener información sobre las tecnologías actuales del desarrollo web] <https://chatgpt.com/>

[7] O-CITY nueva versión. (2024) [en línea] Disponible en: <https://test-ocity.icu/>

[8] Crear y enviar un sitemap (2024) [en línea] Disponible en: <https://developers.google.com/search/docs/crawling-indexing/sitemaps/build-sitemap?hl=es#xml>

[9] React. (2024) [en línea] Disponible en: <https://es.react.dev/>

[10] Ant Design. (2024) [en línea] Disponible en: <https://ant.design/>

[11] Objetivos de desarrollo sostenible. (2024) [en línea] Disponible en: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

**ENLACE PROYECTO GITHUB:** <https://github.com/Jorgius99/proyecto>

# Anexo A. Estructura APIs

## API 1: Países

'https://api.test-ocity.icu/api/country'

Tabla 3. API de países

| nombre del dato | Ejemplo  |
|-----------------|--|
| name            | Spain  |
| iso3            | ESP  |
| numeric_code    | 724  |
| iso2            | ES   |
| phonecode       | 34   |
| capital         | Madrid   |
| currency        | EUR  |
| currency_name   | Euro   |
| currency_simbol | €  |
| tld             | .es  |
| native          | España   |
| región          | Europe   |
| región_id       | 4  |
| subregión       | Southern europe  |
| subregión_id    | 16   |
| nationality     | Spanish  |
| timezones       | [{"zoneName":"Africa/Ceuta","gmtOffset":3600,"gmtOffsetName":"UTC+01:00","abbreviation":"CET","tzName":"Central European Time"}] |
| translations    | {"kr":"스페인","pt-BR":"Espanha","pt":"Espanha","nl":"Spanje","hr":"Španjolska"}  |
| latitude        | 40.00000000  |
| longitude       | -4.00000000  |

|            |                          |
|------------|--------------------------|
| emoji      | ES                       |
| emojiiu    | U+1F1EA U+1F1F8          |
| created_at | 2018-07-21T09:11:03.000Z |
| updated_at | 2023-08-09T21:23:19.000Z |
| Flag       | true                     |
| wikidataid | Q29                      |

## API 2: Estados

<https://api.test-ocity.icu/api/State>

Tabla 4. API de estados

|              |                          |
|--------------|--------------------------|
| id           | 1175                     |
| name         | Valencia                 |
| country_id   | 207                      |
| country_code | ES                       |
| flips_code   | 60                       |
| iso2         | V                        |
| type         | Province                 |
| latitude     | 39.48401080              |
| longitude    | -0.75328090              |
| crated_at    | 2019-10-05T23:18:40.000Z |
| updated_at   | 2022-08-28T12:54:14.000Z |
| flag         | True                     |
| wikidataid   | Q5720                    |

## API 3: Ciudades

[https://api.test-ocity.icu/api/city\\_ocity?limit=0&offset=0](https://api.test-ocity.icu/api/city_ocity?limit=0&offset=0)

Tabla 5. API de ciudades

|           |     |
|-----------|-----|
| id        | 55  |
| contry_id | 207 |

|                   |   |
|-------------------|---|
| city_id           | 32344   |
| state_id          | 1175  |
| ambassador_id     | 3   |
| image             | b6a619e1e19cfc3f9789c6bb62bb4fd32336504b1621421408.jpg  |
| enrollment_letter | a3312e6957efe0f1c717eb89b3889011015c727d1601909056.pdf  |
| hall_name         | 46  |
| hall_email        | sonia.monteagudo@alzira.es  |
| description       | "Claudo Regnum et Adaperio." (Historical motto)   |
| description_local | "Abro y cierro (las puertas del) el Reino." (Lema histórico)  |
| latitude          | 39.15304906126962   |
| longitude         | -0.4446939508789116   |
| user_id           | 46  |
| created_at        | 2024-06-18T16:02:38.366Z  |
| upgrated_at       | 2024-06-18T16:02:38.366Z  |
| deleted_at        | null  |
| ambassador        | <pre> "ambassador": {   "id": 3,   "name": "Universitat Politècnica de València (UPV - Gandia)",   "website": "http://www.upv.es/contenidos/CGANDIA/",   "country_id": 207,   "state_id": 1,   "latitude": 38.996657283704,   "longitude": -0.16629695892334,   "image": "ca95e0ce5041ea23fa794dfc3bbac6319a8ac1f31625672796.jpg",   "created_at": "2024-06-18T14:07:07.761Z",   "updated_at": "2024-06-18T14:07:07.761Z",   "deleted_at": null }, </pre> |
| validators        | <pre> "validators": [   {     "id": 827,     "city_id": 55,     "user_id": "245"   },   {     "id": 826,     "city_id": 55,     "user_id": "46"   } ], </pre>   |
| country           | Toda la información del primer api  |
| state             | Toda la información del segundo api   |

|       |  |
|-------|--|
| city  | <pre>"city": {   "id": 32344,   "name": "Alzira",   "state_id": 1175,   "state_code": "V",   "country_id": 207,   "country_code": "ES",   "latitude": "39.15000000",   "longitude": "-0.43333000",   "created_at": "2019-10-06T00:15:34.000Z",   "updated_at": "2022-08-29T13:35:40.000Z",   "flag": true,   "wikidataid": "Q931780" },</pre>                            |
| links | <pre>"links": [   {     "id": 2541,     "city_id": 55,     "name": "Red de Ciudades Seda Creativa",     "english_name": "Creative Silk Cities Network",     "url": "https://amigosdelaseda.com"   },   {     "id": 2545,     "city_id": 55,     "name": "Turismo Valencia",     "english_name": "Valencia Tourism",     "url": "https://turisme.dival.es/#"   } ],</pre> |

#### API 4: Elementos patrimoniales

[https://api.test-ocity.icu/api/heritage/lists/byCityId/\\${idCity}](https://api.test-ocity.icu/api/heritage/lists/byCityId/${idCity})

Tabla 6. API de los elementos patrimoniales

|                 |  |
|-----------------|--|
| id              | 601  |
| name            | Semana santa de Alzira                                     |
| date            | 2021-07-12T00:00:00.000Z                                   |
| country_id      | 207  |
| state_id        | 1  |
| city_id         | 32344  |
| organization    | Ayuntamiento de Alzira (Turismo)                           |
| ownership       |  |
| periodicity     |  |
| email           | sonia.monteagudo@alzira.es                                 |
| image_copyright | Sonia Monteagudo   |
| image           | 3ffe00415c6cfa4309d15ca10b2983515107bc96<br>1626088801.JPG |

|                                     |  |
|-------------------------------------|--|
| short_heritage_description          | Descripción corta  |
| extended_heritage_description       | Descripción larga ingles   |
| short_local_heritage_description    | Descripción corta  |
| extended_local_heritage_description | Descripción larga idioma local   |
| tags                                |  |
| subtype                             | 108  |
| documentation                       |  |
| is_museum                           | False  |
| is_protected                        | False  |
| is_verified                         | False  |
| protected_values                    |  |
| latitude                            | 39.15135180033445  |
| longitude                           | -0.439715770947271   |
| heritage_field_id                   | 1  |
| created_at                          | 2024-05-24T00:20:01.430Z   |
| updated_at                          | 2024-05-24T00:20:01.430Z   |
| deleted_at                          | null   |
| heritagefield                       | <pre>"heritageField": {   "id": 1,   "name": "Immovable Tangible Cultural",   "category_id": 1,   "category": {     "id": 1,     "name": "Cultural"   } },</pre> |
| city                                | Toda la información del tercer api   |
| country                             | Toda la información del primer api   |
| links                               | <pre>"links": [   {     "id": 36094,     "heritage_list_id": 601,     "name": "",     "url": ""   } ]</pre>  |

# Anexo B. PDF Patrimonio



## Mirador del Mediterráneo (Tavernes de la Vallidigna)

Creation date: 2024-06-26 Country: Spain

Organization or Personal Information: José Marín-Roig Ramón

Email: jomara@eln.upv.es

Location (lat,long): 39.08487017422064 , -0.24853563658447264



Author: Pedro Monteros Valdivieso

## Description

| Local language (short)   | English (short)   |
|--|---|
| Desde el llamado Mirador del Mediterráneo o de la Vallidigna se divisa el paisaje agrario actual, ribeteado por el mar. Los arrozales de antaño dieron paso a los huertos de naranjos a mediados del siglo XX. | From the so-called Mirador del Mediterráneo or Mirador de la Vallidigna you can see the current agricultural landscape, bordered by the sea. The rice fields of yesteryear gave way to orange groves in the middle of the 20th century. |

| Local language  | English   |
|---|---|
| <p>Desde el llamado Mirador del Mediterráneo o de la Vallidigna se divisa el paisaje agrario actual, ribeteado por el mar. Los arrozales de antaño dieron paso a los huertos de naranjos a mediados del siglo XX. Y hoy conviven el azul anacrónico del mar, con el pardo de la marjal y el verde de los árboles en el mismo espacio natural.</p> <p>Este paisaje forma parte de la conocida "Ruta del Racó de Joana", situada en la pintoresca región de Tavernes de la Vallidigna, es un tesoro de belleza natural y biodiversidad. Este sendero serpentea a través de un paisaje exuberante, ofreciendo a los excursionistas una experiencia inmersiva en la esplendorosa naturaleza.</p> <p>La ruta es conocida por su rica y diversa flora, que incluye especies mediterráneas nativas como encinas, algarrobos y arbustos aromáticos como el romero y el tomillo. En primavera, la zona se llena de color con flores silvestres que adornan el camino.</p> <p>La fauna es igualmente abundante, con avistamientos de diversas especies de aves como el águila perdicera, cernícalos y abubillas. También son comunes pequeños mamíferos como conejos y zorros, junto con una variedad de reptiles y anfibios.</p> <p>El paisaje en sí es impresionante, caracterizado por colinas ondulantes, manantiales cristalinos y vistas panorámicas del valle circundante. Los puntos de referencia notables incluyen el área de Sant Llorenç, Ullals Gran y Penyetes, cada uno ofreciendo características naturales y miradores únicos.</p> <p>Esta ruta es ideal para aquellos que buscan conectarse con la naturaleza, ofreciendo tranquilidad y un profundo sentido del patrimonio natural de Tavernes de la Vallidigna.</p> | <p>From the so-called Mirador del Mediterráneo or Mirador de la Vallidigna you can see the current agricultural landscape, bordered by the sea. The rice fields of yesteryear gave way to orange groves in the middle of the 20th century. And today the anachronistic blue of the sea coexists with the brown of the marshland and the green of the trees in the same natural space.</p> <p>This landscape forms part of the well-known "Ruta del Racó de Joana", situated in the picturesque region of Tavernes de la Vallidigna, is a treasure trove of natural beauty and biodiversity. This trail winds through the lush landscape, offering hikers an immersive experience in nature's splendor.</p> <p>The route is renowned for its rich and diverse flora, including native Mediterranean species such as holm oaks, carob trees, and aromatic shrubs like rosemary and thyme. In spring, the area bursts into color with wildflowers adorning the path.</p> <p>Fauna is equally abundant, with sightings of various bird species like the Bonelli's eagle, kestrels, and hoopoes. Small mammals such as rabbits and foxes are also common, along with a variety of reptiles and amphibians.</p> <p>The landscape itself is breathtaking, characterized by rolling hills, crystal-clear springs, and panoramic views of the surrounding valley. Notable landmarks include the Sant Llorenç area, Ullals Gran, and Penyetes, each offering unique natural features and viewpoints.</p> <p>This route is ideal for those seeking to connect with nature, offering tranquility and a deep sense of the natural heritage of Tavernes de la Vallidigna.</p> |

## Fields of the heritage

Natural: Natural site

Tags: LANDSCAPE

Related documentation: photos, videos, maps

## Links of Bibliography

[Valencia Bonita](#)

## Links of interest

[Valencia Bonita](#)

[Wikiloc](#)

Tue, 02 Jul 2024 09:52:17 GMT