UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Desarrollo de una aplicación web con React para la creación de avatares personalizados para O-CITY

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Valls Martinez, Alberto

Tutor/a: Marín-Roig Ramón, José

Cotutor/a externo: Burruchaga Sola, Daniel

CURSO ACADÉMICO: 2023/2024

Resumen

O-CITY 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.

Es por ello por lo que este trabajo se ha centrado en la generación de avatares personalizados mediante IA. Estos avatares guiarán al usuario a través de la plataforma de O-CITY. Para su desarrollo se ha utilizado JavaScript, express, OpenAI[2], React[3] y CSS.

Palabras clave

React, OpenAI, avatares, IA, personalizados.

Abstract

O-CITY is a digital atlas of natural and cultural heritage aimed at promoting 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 driven the project's evolution towards new ways of visualizing information on the map.

Therefore, this work has focused on the generation of personalized avatars through AI. These avatars will guide the user through the O-CITY platform. JavaScript, Express, OpenAI, React, and CSS have been used for its development.

Keywords

React, OpenAI, avatars, AI, customized.

ÍNDICE

ÍNDICE DE IMÁGENES

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.

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 compatibles con esta filosofía y que aporten viabilidad al futuro del proyecto. Estas alternativas incluyen la creación de páginas de pago dentro de la aplicación con una experiencia de navegación mejorada para los usuarios, utilizando, por ejemplo, la herramienta desarrollada a continuación, es decir avatares generados mediante Inteligencia Artificial (IA).

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/.

El TFG se centra en crear una nueva herramienta para mejorar la experiencia del usuario a través de los mapas de O-CITY y brindarle una mejor aplicación educativa. Esta herramienta añadiría un valor muy atractivo a los clientes. Esta herramienta consiste en la creación de avatares por parte de los usuarios administradores tanto por texto como por imagen. El usuario podrá escribir una breve descripción del avatar que desea generar y seleccionar entre varios estilos disponibles. También será capaz de subir una imagen con el rostro de una persona y convertirla en avatar.

En los siguientes capítulos se hablará de O-CITY y se explicará su evolución tanto a nivel de organización como de software. Y lo que permite esta nueva herramienta en este entorno.

Objetivos

El principal objetivo de este proyecto es la creación de una aplicación capaz de generar avatares personalizados gracias a una prompt donde los usuarios administradores introducirán un texto con las características, detalles y estilos que desean en su avatar personalizado.

Como objetivos secundarios encontramos:

- Generación de avatares a través de una imagen.
- Interfaz sencilla y agradable para el usuario.
- Descargar los avatares generados.
- Elección de APIS eficientes.
- Documentación del proyecto.

Metodología de trabajo

Para asegurar el mejor resultado en el desarrollo del proyecto, se utilizó la metodología ágil SCRUM. Esta metodología divide el proyecto en ciclos cortos, generalmente de dos semanas, conocidos como sprints. Al final de cada sprint, se debe entregar un resultado completo de las tareas asignadas por el Product Owner (PO). Durante el proyecto, se realizaron sprints de 2 o 3 semanas, y al finalizar cada uno, se evaluó el trabajo realizado con el PO, quien actuaba como tutor del proyecto. Además, se llevó a cabo una revisión del sprint y una planificación del siguiente sprint para mantener un proceso dinámico y eficiente.

Ilustración 1. Metodología Scrum

Estructura de la memoria

La estructura del presente documento es la siguiente:

Capítulo 1: Introducción.

En el primer capítulo se realiza una breve introducción al proyecto, cuáles son sus objetivos, sus fases de realización y una descripción de la estructura del documento.

Capítulo 2: Marco teórico.

En este segundo capítulo se pretende dar a conocer la base teórica del proyecto, haciendo hincapié en los diferentes tipos de API para la generación de avatares personalizados, así como las herramientas actuales disponibles para llevar a cabo un desarrollo frontend que cumpla con los requisitos que se piden.

Capítulo 3: Diseño.

En este capítulo se realiza un estudio de los requisitos de la aplicación, donde se comparan las diferentes APIS de generación de avatares por texto y por imagen.

Capítulo 4. Implementación.

En este cuarto capítulo se muestra todo el proceso de desarrollo e implementación de la aplicación web, tanto la generación de avatares por texto como por imagen. Se detalla las herramientas utilizadas para su correcto desarrollo. Posteriormente se describe la estructura de la aplicación y los problemas de implementación resueltos.

Capítulo 5. Testeo.

En este capítulo se realiza una evaluación final del sistema para comprobar el recorrido virtual de los avatares generados, tanto por texto como por imagen, y para ello se utiliza la herramienta Selenium IDE.

Capítulo 6. Conclusiones y futuras líneas.

En este último capítulo se realiza una valoración final del proyecto teniendo en cuenta como la IA puede ayudar en el turismo y mejorar la experiencia del usuario. Se expondrán los logros conseguidos y se hablará de futuras líneas de mejora.

Anexo 1. ODS

En el Anexo 1, al final del documento, vemos la relación de este trabajo respecto a los Objetivos de Desarrollo Sostenible pero aquí vemos algo más en detalle por que se ha decidido que estos ODS están relacionados directamente con el trabajo realizado.

Fases de realización

Para la realización de este proyecto se siguieron un total de 7 etapas que se detallan a continuación:

Tabla 1. Horas requeridas para el desarrollo de la aplicación

Fase	Tarea	Acciones realizadas	Tiempo
1.	Recopilación de información.	Búsqueda de las diferentes herramientas y APIS para la generación de avatares personalizados.	45h.
2.	Desarrollo de la aplicación en local.	Desarrollo de la aplicación que generará los avatares personalizados.	75h.
3.	Implementación de la conversión de foto en avatar	Convertir una imagen subida en un avatar	30h.
4.	Implementación del frontend.	Implementación del diseño de la web.	15h.
5.	Comprobación de errores y fluidez	Comprobación de la fluidez y eficiencia a la hora de generar avatares y seleccionarlos.	15h.
6.	Redacción del documento final	Redacción del desarrollo y los puntos anteriores.	20h.

Capítulo 2. Marco teórico

¿Qué es el proyecto O-City?

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) [1].

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 a los profesionales.

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 [2]:

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 12 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.

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 lineas 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, Riba-roja,

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) 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 Tesauro de clasificación de patrimonio de más de 150 palabras clave.

Importación de datos a través de la aplicación de turismo profesional: 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. [3]

¿Qué es la Inteligencia Artificial?

Como se menciona en [3] la inteligencia artificial (AI, por sus siglas en inglés) es un campo de la informática que se dedica al desarrollo de sistemas y tecnologías capaces de realizar tareas que normalmente requieren inteligencia humana.

La IA se divide en varias subdisciplinas, entre las cuales se destacan:

1. **Aprendizaje automático (Machine Learning):** Una rama de la IA que permite a las máquinas aprender y mejorar a partir de la experiencia sin ser programadas explícitamente para ello. Utiliza algoritmos que pueden identificar patrones en datos y hacer predicciones o decisiones basadas en esos datos.

2. **Procesamiento del lenguaje natural (NLP):** Se centra en la interacción entre las computadoras y el lenguaje humano. Permite que las máquinas comprendan, interpreten y respondan al lenguaje hablado y escrito de manera natural.

3. **Visión por computadora:** Permite a las máquinas interpretar y comprender el mundo visual, analizando imágenes y videos para identificar objetos, personas, escenas y actividades.

4. **Robótica:** Combina la IA con la ingeniería para diseñar y construir robots que pueden realizar tareas físicas, desde las más simples hasta las más complejas, en diversos entornos.

5. **Sistemas expertos:** Programas que imitan la toma de decisiones de un experto humano en un campo específico. Utilizan una base de conocimientos y reglas para resolver problemas específicos.

Tecnologías Subyacentes

Redes Neuronales Convolucionales (CNN):

Las CNN son una clase de redes neuronales profundas que han demostrado ser particularmente efectivas en tareas de procesamiento de imágenes. Estas redes son capaces de aprender características visuales a partir de datos de entrenamiento, lo que les permite reconocer patrones complejos y realizar transformaciones visuales avanzadas.

Generative Adversarial Networks (GAN):

Las GAN consisten en dos redes neuronales que se entrenan juntas: una red generadora que crea imágenes y una red discriminadora que evalúa su autenticidad. Este enfoque ha revolucionado la generación de imágenes sintéticas, permitiendo la creación de avatares que no solo son realistas sino también estilísticamente coherentes con las entradas originales.

Transferencia de Estilo (Style Transfer):

La transferencia de estilo es una técnica que utiliza redes neuronales para aplicar el estilo visual de una imagen (como una obra de arte) a otra imagen (como una foto). Esta técnica es fundamental en la creación de avatares estilizados, donde se desea mantener las características faciales de la foto original mientras se aplica un estilo artístico particular.

¿Qué es OpenAI?

OpenAI es una organización de investigación y desarrollo en inteligencia artificial fundada en diciembre de 2015. Es una de las mayores compañías de inteligencia artificial del mundo.

Principales logros de OpenAI:

- GPT (Generative Pre-trained Transformer): Uno de los logros más conocidos de OpenAI es la serie de modelos GPT, con GPT-3 y GPT-4 siendo los más recientes. Estos modelos de lenguaje natural pueden generar texto coherente y relevante en una variedad de contextos, lo que ha abierto múltiples aplicaciones en automatización, asistencia virtual, creación de contenido, y más.

- Codex: Un modelo que puede entender y generar código, utilizado en productos como GitHub Copilot, que ayuda a los desarrolladores a escribir código de manera más eficiente.

- DALL-E[6]: Un modelo que puede generar imágenes a partir de descripciones textuales, demostrando la capacidad de la IA para entender y crear contenido visual.

- OpenAI Gym: Una plataforma para desarrollar y comparar algoritmos de aprendizaje por refuerzo. Proporciona una variedad de entornos que los investigadores pueden usar para entrenar y evaluar sus algoritmos.

Capítulo 3. Diseño

3.1. Requisitos de la aplicación

El requisito para este proyecto es que se generen unos avatares que se correspondan con la descripción introducida por el usuario, que tengan una buena calidad de imagen y que se puedan descargar dichos avatares para seleccionarlos como guías virtuales por el mapa de O-CITY.

Los requisitos de la aplicación son los siguientes:

- Crear un generador de avatares por texto donde se genere un avatar fiel al texto introducido.
- Crear un generador de avatares por imagen donde se convierta la foto de una persona en un avatar.
- Que existan varias alternativas de estilos para seleccionar en los avatares.
- Que sea integrable con el código actual (React).
- Que se puedan descargar los avatares generados.

Aquí tenemos un ejemplo de visualización de cómo quedarían los avatares cuando el usuario introduce una descripción en la prompt proporcionada.

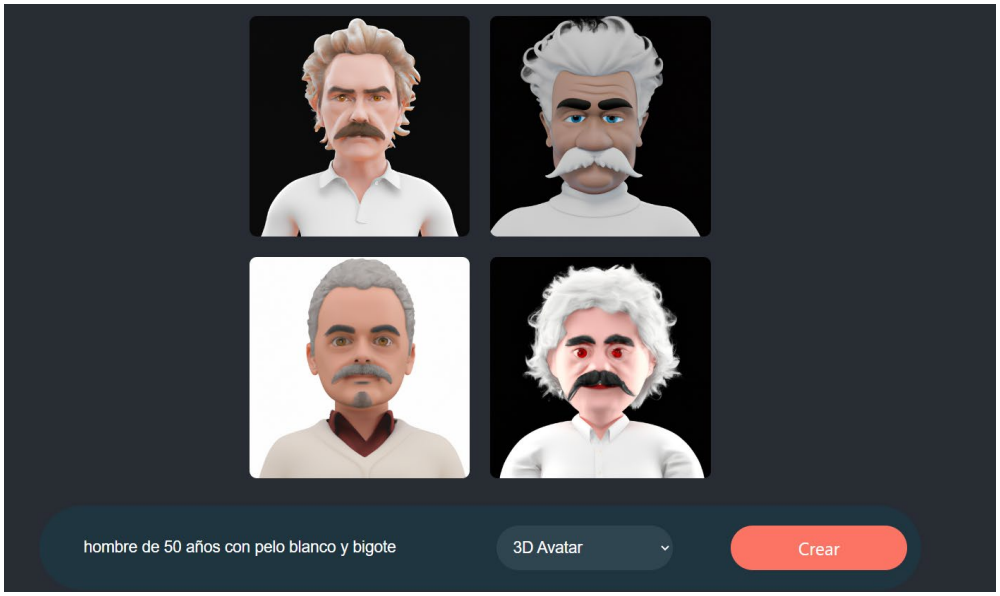

Ilustración 2. Prueba generación de avatares

3.2. Avatares generados por texto

Herramientas para el desarrollo:

Según define Wikipedia [4], React es una biblioteca de JavaScript desarrollada por Facebook que se utiliza para construir interfaces de usuario (UI) interactivas y eficientes. Es una de las herramientas más populares y ampliamente utilizadas para el desarrollo frontend en la actualidad. A continuación, se destacan algunos puntos clave sobre React:

Componentes: React se basa en el concepto de componentes reutilizables. Un componente en React es una pieza autónoma de la interfaz de usuario que puede contener tanto lógica como estructura visual. Estos componentes pueden ser simples, como un botón o un cuadro de texto, o complejos, como una barra de navegación o una página completa.

Virtual DOM: React utiliza un modelo de Document Object Model (DOM) virtual para optimizar el rendimiento. En lugar de manipular directamente el DOM del navegador, React crea una representación virtual de los elementos DOM en memoria y solo actualiza los elementos que han cambiado. Esto mejora significativamente la eficiencia y la velocidad de las actualizaciones de la interfaz de usuario.

JSX: JSX es una extensión de sintaxis JavaScript utilizada en React que permite escribir HTML directamente dentro del código JavaScript. Esto facilita la creación de componentes y el mantenimiento del flujo de datos entre los componentes de React.

Unidireccionalidad de los Datos: React promueve el flujo unidireccional de datos, lo que significa que los datos fluyen en una sola dirección: desde los componentes principales hacia los componentes secundarios. Esto facilita la depuración, el mantenimiento y la gestión del estado de la aplicación.

React Hooks: Los Hooks son funciones que permiten utilizar el estado y otras características de React sin necesidad de escribir clases. Esto simplifica el desarrollo y la reutilización del código, especialmente en componentes funcionales.

Librería, no Framework: A diferencia de frameworks como Angular, React se describe a menudo como una librería debido a su enfoque en la gestión de la interfaz de usuario, mientras que otras partes del desarrollo, como el enrutamiento o la gestión del estado, pueden ser manejadas por otras bibliotecas o soluciones.

Ecosistema y Comunidad Activa: React cuenta con una amplia comunidad de desarrolladores y tiene un ecosistema robusto de herramientas, bibliotecas y extensiones (como Redux para la gestión del estado, React Router para el

enrutamiento, entre otros) que complementan sus funcionalidades y facilitan el desarrollo de aplicaciones complejas.

APIS para generación de imágenes:

OpenAI DALL-E

DALL-E es un modelo de inteligencia artificial desarrollado por OpenAI que genera imágenes realistas a partir de descripciones textuales. Puede utilizarse para crear avatares personalizados y otras imágenes sintéticas basadas en texto. Genera imágenes de alta calidad a partir de descripciones textuales específicas. Permite personalización y creatividad en la generación de avatares. Es Ideal para aplicaciones que requieren avatares únicos y personalizados basados en descripciones detalladas.

Avatoon

Avatoon es una aplicación y API que permite a los usuarios crear avatares personalizados mediante herramientas de edición intuitivas y opciones de personalización. Ofrece una variedad de características faciales y estilísticas para crear avatares únicos. Incluye opciones de ajuste de peinados, ropa, accesorios y fondos. Es ampliamente utilizado en aplicaciones de redes sociales, juegos, y plataformas de mensajería para personalizar perfiles de usuarios.

Hugging Face

Hugging Face es una plataforma que ofrece modelos de inteligencia artificial pre-entrenados y herramientas para el procesamiento del lenguaje natural (NLP). Su API puede utilizarse para generar texto y, en algunos casos, imágenes a partir de descripciones. Aunque Hugging Face se centra en modelos de NLP, su comunidad ha desarrollado modelos y técnicas para generar imágenes a partir de texto, aunque no es tan específica para avatares como otras opciones. Principalmente utilizado en aplicaciones de NLP y generación de contenido textual, con capacidades emergentes en generación de imágenes a través de modelos como CLIP.

DoppelMe API

DoppelMe proporciona una API para la creación de avatares personalizados basados en texto. Permite a los desarrolladores integrar fácilmente la funcionalidad de creación de avatares en sus aplicaciones.

Creación de avatares basada en descripciones textuales.
Personalización de características faciales, peinados, ropa, etc.
Opciones de integración flexible. Es utilizado en juegos, redes sociales, plataformas de mensajería y aplicaciones educativas para personalizar perfiles de usuario.

Tabla comparativa

A continuación, se muestra una tabla comparativa entre varias herramientas diferentes para la creación de imágenes personalizadas. La elección de la API de DALL-E responde a que tiene las mejores características con respecto a sus competidores.

De esta tabla también se ha probado la API de Hugging Face y la de DoppleMe, con resultandos menos exitosos que con DALL-E.

Tabla 2. Comparativa APIS generación de avatares por texto

	DoppelMe API	**Hugging Face**	**Avatoon**	**OpenAI DALL-E**
Propósito	Crear avatares personalizados	Plataforma de modelos de IA	Crear avatares y caricaturas	Generación de imágenes a partir de texto
Funcionalidad	Generación de avatares	Modelos de NLP, visión, y más	Diseño y personalización de avatares	Generación de imágenes mediante descripciones textuales
Tipo de Imágenes	Avatares simples en 2D	Variedad de imágenes y datos	Avatares caricaturescos y personalizados	Imágenes de alta calidad basadas en texto
Uso Comercial	Permitido, con restricciones	Sí, depende del modelo	Sí	Sí, con licencia
Accesibilidad	API con suscripción	Acceso gratuito y de pago	Aplicación móvil y web	API, acceso con suscripción
Facilidad de Uso	Moderado	Moderado a avanzado	Fácil	Moderado

Escalabilidad	Limitada	Alta	Limitada	Alta
Personalización	Limitada a características básicas	Amplia (según el modelo)	Amplia (colores, estilos, accesorios)	Amplia (basada en descripciones textuales)
Calidad Visual	Básica	Variable, alta calidad en general	Alta	Alta
Documentación	Moderada	Extensa y detallada	Básica	Completa y detallada
Soporte Técnico	Limitado	Amplio, comunidad activa	Limitado	Amplio

Costes, limitaciones y uso API DALL-E:

El coste de generar los avatares depende del modelo que se esté utilizando y la resolución requerida, a mayor resolución, mayor es el precio por imagen.

El límite máximo que permite gastar OpenAI por mes es de 120 dólares.

Model	Quality	Resolution	Price
DALL·E 3	Standard	1024×1024	$0.040 / image
	Standard	1024×1792, 1792×1024	$0.080 / image
DALL·E 3	HD	1024×1024	$0.080 / image
	HD	1024×1792, 1792×1024	$0.120 / image
DALL·E 2		1024×1024	$0.020 / image
		512×512	$0.018 / image
		256×256	$0.016 / image

Ilustración 3. Precios API Dall-E

Tanto dall-e 2 como 3 permiten generar 5 imágenes personalizadas por minuto.

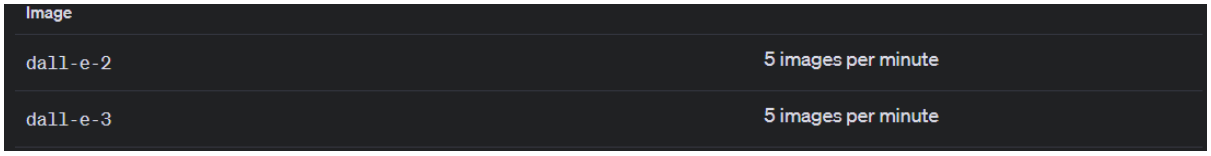

Image	
dall-e-2	5 images per minute
dall-e-3	5 images per minute

Ilustración 4. Límites API Dall-E

3.3 Avatares generados por imagen

Herramientas y Tecnologías Utilizadas:

Se ha utilizado React.js para la construcción de las interfaces de usuario, Node.js como entorno de ejecución, Express.js como framework web para Node.js, Axios como cliente HTTP para realizar solicitudes HTTP, y Sharp que es una biblioteca de Node.js para el procesamiento de imágenes.

Como API se ha utilizado Cutout.pro para el procesamiento de imágenes y la generación de los avatares.

Cutout.Pro es una plataforma de inteligencia artificial que ofrece una variedad de herramientas para el procesamiento de imágenes. Estas herramientas están diseñadas para ayudar a los usuarios a realizar tareas complejas de edición de imágenes de manera rápida y eficiente. La plataforma utiliza algoritmos avanzados de IA para proporcionar servicios como eliminación de fondos, mejora de fotos, generación de avatares, y más. Estas son algunas de sus características:

- Convierte fotos de retratos en avatares de estilo cartoon utilizando IA.
- Ofrece múltiples estilos de avatar como Magic Mirror, Fairy, Dreamland, Manga, K-POP, Pixel, y más.
- Soporta tanto la conversión de la cara principal en la imagen como la inclusión de parte de la ropa en el avatar final.

La API de Cutout.Pro permite a los desarrolladores integrar las capacidades de procesamiento de imágenes en sus propias aplicaciones. Aquí se describen algunas características y funcionalidades clave de la API.

Características Principales de la API

- La API está diseñada para ser fácil de integrar con ejemplos de código claros y una documentación detallada.
- Permite la subida de archivos en varios formatos de imagen.
- Soporta tanto la devolución de imágenes procesadas como la entrega de resultados en formato base64.
- Ofrece múltiples parámetros de configuración para ajustar el procesamiento de imágenes según las necesidades del usuario.
- Incluye opciones para seleccionar diferentes estilos de avatar en la herramienta de Cartoon Selfie.

URL: `https://www.cutout.pro/api/v1/cartoonSelfie`
Método: POST
Parámetros:

`file`: Archivo de imagen a procesar.
`cartoonType`: Tipo de avatar a generar (valores entre 0 y 13).
`outputFormat`: Formato de la imagen de salida (png, jpg, etc.).

Tabla comparativa de alternativas a Cutout.Pro

Tabla 3. Comparativa APIS avatares generados por imagen

Característica	Cutout.Pro	DeepAI	Avatarify	Cartoonify
URL de la API	https://www.cutout.pro/api/v1/cartoonSelfie	https://api.deepai.org/api/toonify	https://github.com/alievk/avatarify	https://cartoonify.de/api/v1/convert
Método de Solicitud	POST	POST	Instalación local	POST
Formatos de Entrada	PNG, JPG, JPEG, BMP, WEBP	PNG, JPG	PNG, JPG	PNG, JPG, JPEG
Formatos de Salida	PNG, JPG, JPEG	PNG, JPG	PNG, JPG	PNG, JPG
Estilos Disponibles	0-13 (Magic Mirror, Fairy, Manga, etc.)	Toonify	Basado en GANs	Cartoon, Sketch

Límite de Tamaño de Archivo	15 MB	No especificado	Dependiente de hardware local	5 MB
Límite de Resolución	4096x4096 píxeles	No especificado	Dependiente de hardware local	2048x2048 píxeles
Autenticación	API Key	API Key	No requiere (open source)	API Key
Costo	Planes gratuitos y de pago disponibles	Planes gratuitos y de pago disponibles	Gratuito (open source)	Gratuito con límites, planes de pago disponibles
Tiempo de Respuesta	Rápido (dependiente de la red)	Rápido (dependiente de la red)	Inmediato (procesamiento local)	Rápido (dependiente de la red)
Documentación	Completa, ejemplos claros	Completa, ejemplos claros	Documentación y comunidad activa en GitHub	Documentación completa
Soporte Técnico	Email y soporte en línea	Email y soporte en línea	Comunidad y GitHub Issues	Email y soporte en línea

Alternativas:

Cutout.Pro:

Cutout.Pro ofrece una gran variedad de estilos de avatares que se pueden personalizar en detalle, permitiendo a los usuarios ajustar aspectos específicos de sus avatares según sus preferencias.

Este servicio soporta avatares en alta resolución y archivos de gran tamaño, asegurando que los avatares generados sean de alta calidad y adecuados para diversas aplicaciones. Cutout.Pro proporciona una API que está bien documentada, lo que facilita su integración en diferentes plataformas y aplicaciones, permitiendo a los desarrolladores aprovechar sus funcionalidades sin complicaciones.

DeepAI[9]:

DeepAI se centra en ofrecer servicios de generación de avatares que son fáciles de usar, lo que lo hace accesible incluso para aquellos con poca experiencia técnica.

Comparado con Cutout.Pro, DeepAI ofrece menos opciones de personalización, lo que puede ser una limitación para usuarios que buscan un control más detallado sobre el diseño de sus avatares. Este servicio es ideal para aplicaciones que requieren una solución de generación de avatares rápida y sencilla, sin necesidad de configuraciones complejas.

Avatarify[10]:

Avatarify es una solución de código abierto que utiliza redes generativas antagónicas (GANs) para crear avatares, ofreciendo a los usuarios una tecnología avanzada y flexible.

A diferencia de otras opciones, Avatarify necesita ser instalado y configurado localmente, lo que puede ser un desafío para algunos usuarios debido a su complejidad.

Gracias a su naturaleza de código abierto, Avatarify proporciona una gran flexibilidad y posibilidades de personalización, permitiendo a los usuarios modificar y adaptar el software según sus necesidades específicas.

Cartoonify[11]:

Cartoonify se especializa en convertir fotos en dibujos animados o bocetos, ofreciendo una transformación visual creativa y divertida.

El servicio es gratuito para un uso básico, con opciones de pago disponibles para usuarios que requieren funciones más avanzadas o un uso más intensivo.

Cartoonify es una buena opción para usuarios que necesitan un servicio rápido y accesible para convertir fotos en ilustraciones estilo cartoon, sin necesidad de configuraciones complicadas.

Capítulo 4. Implementación

4.1 Tecnologías

Se han utilizado las siguientes tecnologías para el desarrollo de la aplicación del proyecto:

- **React.js**: Biblioteca de JavaScript para la construcción de interfaces de usuario.
- **Node.js[5]**: Entorno de ejecución para JavaScript del lado del servidor.
- **Express.js[4]**: Framework web para Node.js.
- **Axios**: Cliente HTTP basado en promesas para realizar solicitudes HTTP.
- **Sharp**: Biblioteca de Node.js para el procesamiento de imágenes.
- **Cutout.Pro API**: Servicio que ofrece herramientas de procesamiento de imágenes basadas en IA.
- **OpenAI API**: Servicio que ofrece la generación de avatares personalizados.

Durante el desarrollo del proyecto se utilizó Git como herramienta de control de versiones. Se han ido subiendo los cambios que se han ido realizando en un repositorio, tanto de la parte del frontend como en la del servidor que gestiona las peticiones de la API.

Después se introducen estos cambios en el dominio de test de O-CITY.

La aplicación de generación de avatares está diseñada utilizando una arquitectura basada en componentes con React para el frontend y Express para el backend. Esta estructura modular facilita el mantenimiento y la escalabilidad del proyecto. La aplicación permite a los usuarios generar avatares personalizados utilizando la API de OpenAI y ofrece opciones para descargar los avatares generados con fondos transparentes. Y con la API de Cutout.pro es posible convertir una imagen de una persona en un avatar.

Frontend

El frontend de la aplicación está construido con React. Utiliza componentes funcionales y hooks para gestionar el estado y los efectos secundarios. Los principales componentes son `Intro`, `AvatarGenerator`, `Footer` y `PhotoToAvatar`.

Se utiliza CSS para el diseño de estas páginas.

Páginas comunes:

Intro.jsx

- Página de introducción que explica el funcionamiento de la aplicación.
- Permite a los usuarios iniciar el proceso de generación de avatares o importar un avatar existente desde su dispositivo.
- Contiene un botón para abrir el explorador de archivos y seleccionar una imagen.

Footer.jsx

- Componente que muestra el pie de página en todas las páginas.
- Incluye información sobre la política de privacidad, derechos de autor, y el logo de la UE.

Navegación

La navegación entre componentes se gestiona mediante `react-router-dom`, permitiendo a los usuarios moverse entre la página de introducción, la página de generación de avatares por texto y la página de generación de avatares por imagen.

4.2 Implementación avatares por texto

Hooks Utilizados

- useState: Para gestionar el estado de los avatares generados, la carga y los errores.
- useRef: Para referenciar el input de tipo `file` y manejar el diálogo de selección de archivos.
- useEffect: Para gestionar efectos secundarios como la actualización de errores cuando la carga está activa.

```
const imageGenerator = async () => {
    const prompt = inputRef.current.value.trim();
    if (prompt === "") {
        setError("Escribe algo en el cuadro de texto");
        return;
    }
    setLoading(true);

    try {
        const response = await fetch(
            "http://localhost:3001/generate-avatar",
            {
                method: "POST",
                headers: {
                    "Content-Type": "application/json"
                },
                body: JSON.stringify({ prompt, style: selectedStyle }),
            }
        );

        if (!response.ok) {
            const errorText = await response.text();
            if (response.status === 429) {
                setError("Has alcanzado el límite para generar avatares. Espera unos segundos.");
            } else {
                setError(`API call failed with status: ${response.status} ${response.statusText}: ${errorText}`);
            }
            setLoading(false);
            return;
        }

        const data = await response.json();
        console.log("API Response:", data);
        if (!data || !data.data || data.data.length === 0) {
            setError("No data returned from API or data is empty");
            setLoading(false);
            return;
        }

        setImageUrls(data.data.map(img => img.url));
        setSelectedImage(null);
        setLoading(false);
    } catch (error) {
        setError('Failed to fetch');
        setLoading(false);
    }
}

const throttledImageGenerator = throttle(imageGenerator, 60000);  // 60 segundos de espera entre solicitudes
```

Ilustración 5. Código para enviar la solicitud OpenAI

Backend

El backend de la aplicación está construido con Express y se encarga de manejar las solicitudes de generación y descarga de avatares utilizando la API de OpenAI.

Servidor Express

server.js

- Configura el servidor Express.
- Define las rutas para la generación (`/generate-avatar`) y descarga (`/download-avatar`) de avatares.
- Maneja la sanitización de entradas y la gestión de errores.

Rutas Principales

- POST /generate-avatar
    - Recibe un prompt y un estilo.
    - Llama a la API de OpenAI para generar avatares basados en la descripción proporcionada.
    - Devuelve las URLs de las imágenes generadas.

- POST /download-avatar
    - Recibe la URL de una imagen.
    - Descarga la imagen y la envía al cliente.

Integración con OpenAI

La aplicación utiliza la API de OpenAI para generar imágenes basadas en descripciones textuales. Las claves API se almacenan de manera segura en un archivo `.env` y se utilizan para autenticar las solicitudes.

Manejo de Claves API

Archivo .env

- Almacena la clave API de OpenAI y el puerto del servidor.
- Se asegura de que las claves API no se expongan en el código fuente público.

```
app.post('/generate-avatar', async (req, res) => {
    const { prompt, style } = req.body;

    if (!prompt) {
        return res.status(400).send("Prompt is required");
    }

    const sanitizedPrompt = prompt.replace(/[^a-zA-Z0-9 ]/g, '');
    const apiKey = process.env.OPENAI_API_KEY;
    if (!apiKey) {
        console.error("OpenAI API key is missing");
        return res.status(500).send("Server configuration error: API key is missing");
    }

    console.log(`Generating avatar with prompt: "${sanitizedPrompt}" and style: "${style}"`);

    try {
        const response = await fetch("https://api.openai.com/v1/images/generations", {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
                Authorization: `Bearer ${apiKey}`,
            },
            body: JSON.stringify({
                prompt: `${style} avatar of ${sanitizedPrompt}`,
                n: 4,
                size: "512x512",
            }),
        });

        if (!response.ok) {
            const errorText = await response.text();
            console.error(`OpenAI API response error: ${response.status} ${response.statusText}: ${errorText}`);
            return res.status(response.status).send(errorText);
        }

        const data = await response.json();
        console.log("Generated avatars:", data);
        return res.json(data);
    } catch (error) {
        console.error("Error mientras se generaba el avatar:", error);
        return res.status(500).send(error.message);
    }
```

Ilustración 6. Código servidor local OpenAI

```
app.post('/download-avatar', async (req, res) => {
    const { imageUrl } = req.body;

    if (!imageUrl) {
        return res.status(400).send("Imagen URL es requerida");
    }

    try {
        const response = await fetch(imageUrl);
        if (!response.ok) {
            throw new Error('Failed to fetch image');
        }
        const buffer = await response.buffer();

        const processedImage = await sharp(buffer)
            .resize(256, 256, { fit: 'contain', background: { r: 0, g: 0, b: 0, alpha: 0 } })
            .toBuffer();

        res.writeHead(200, {
            'Content-Type': 'image/png',
            'Content-Disposition': 'attachment; filename=avatar.png',
            'Content-Length': processedImage.length
        });
        return res.end(processedImage);
    } catch (error) {
        console.error("Error descargando el avatar:", error);
        res.status(500).send(error.message);
    }
});

app.use(express.static(path.join(__dirname, 'build')));

app.get('*', (req, res) => {
    res.sendFile(path.join(__dirname, 'build', 'index.html'));
});

app.listen(port, () => {
    console.log(`Server is running on http://localhost:${port}`);
});
```

Ilustración 7. Código para descargar los Avatares generados

Proceso de generación de avatares

Inicio en la Página de Introducción

- El usuario lee las instrucciones y puede empezar a generar un avatar o importar uno existente.

Ilustración 8. Inicio Web Avatares

Generación de Avatares por texto:

El usuario ingresa una descripción y selecciona un estilo, se envía una solicitud al backend para generar avatares, el backend llama a la API de OpenAI y devuelve las imágenes generadas, las imágenes se muestran en la interfaz y el usuario puede seleccionar y descargar su avatar preferido.

Ilustración 9. Prompt para generar los avatares

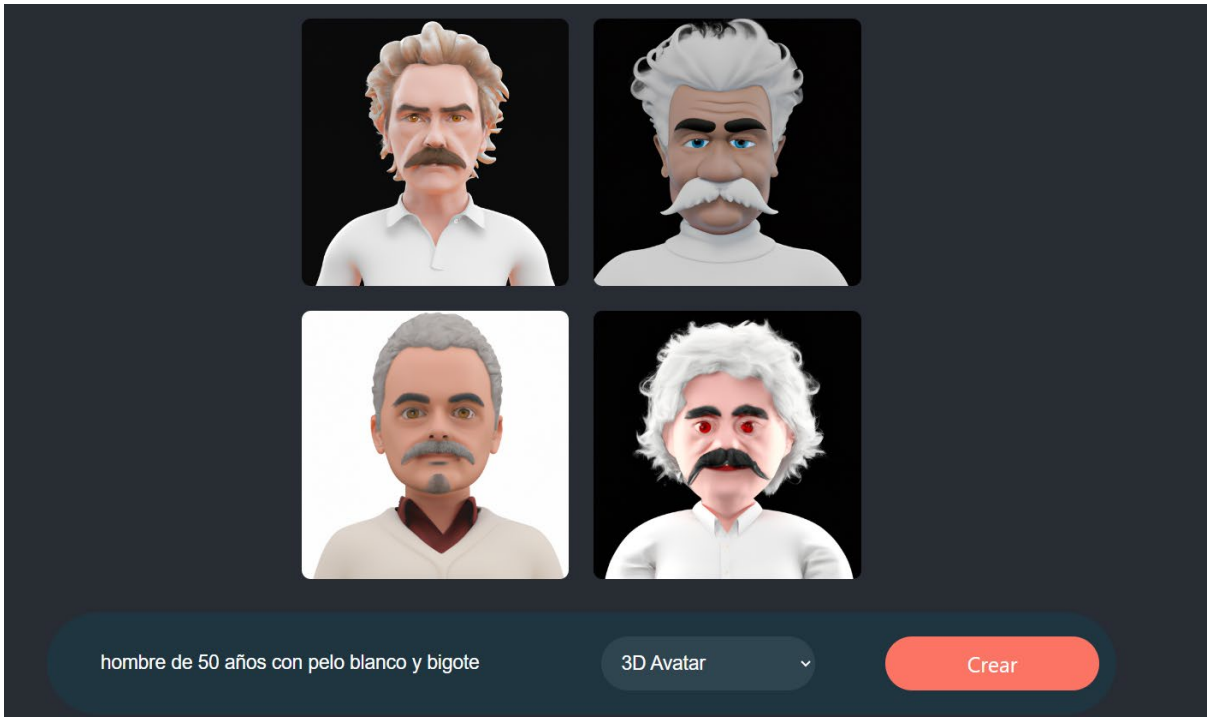

Ilustración 10. Prueba real generada de avatares con un texto.


```
Server is running on http://localhost:3001
Generating avatar with prompt: "hombre de 50 años con pelo blanco y bigote" and style: "3D Avatar"
Generated avatars: {
  created: 1720452589,
  data: [
    {
      url: 'https://oaidalleapiprodscus.blob.core.windows.net/private/org-JvuyAH2JtsTaNcSJvsWqFRLQ/user-wWM2njJ7xDaE6l0S
ggE97YFs/img-pt9ND857w0z3cifYe0HtJpgL.png?st=2024-07-08T14%3A29%3A49Z&se=2024-07-08T16%3A29%3A49Z&sp=r&sv=2023-11-03&sr=
b&rscd=inline&rsct=image/png&skoid=6aaadede-4fb3-4698-a8f6-684d7786b067&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2
024-07-07T19%3A19%3A29Z&ske=2024-07-08T19%3A19%3A29Z&sks=b&skv=2023-11-03&sig=BDPWmQJnfWwUrACtnC2Ip1E2XgYmGKcrlpDhWbB%2B
K84%3D'
    },
    {
      url: 'https://oaidalleapiprodscus.blob.core.windows.net/private/org-JvuyAH2JtsTaNcSJvsWqFRLQ/user-wWM2njJ7xDaE6l0S
ggE97YFs/img-grjrOP0WP5S36S6dmVznoI7c.png?st=2024-07-08T14%3A29%3A49Z&se=2024-07-08T16%3A29%3A49Z&sp=r&sv=2023-11-03&sr=
b&rscd=inline&rsct=image/png&skoid=6aaadede-4fb3-4698-a8f6-684d7786b067&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2
024-07-07T19%3A19%3A29Z&ske=2024-07-08T19%3A19%3A29Z&sks=b&skv=2023-11-03&sig=FkwAexeQpFomKjyjLhVlFGsTI9UxRcf9O9QZW6L8l3
E%3D'
    },
    {
      url: 'https://oaidalleapiprodscus.blob.core.windows.net/private/org-JvuyAH2JtsTaNcSJvsWqFRLQ/user-wWM2njJ7xDaE6l0S
ggE97YFs/img-Kjlyt63og0UKUGr2k4ITcwfe.png?st=2024-07-08T14%3A29%3A48Z&se=2024-07-08T16%3A29%3A48Z&sp=r&sv=2023-11-03&sr=
b&rscd=inline&rsct=image/png&skoid=6aaadede-4fb3-4698-a8f6-684d7786b067&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2
024-07-07T19%3A19%3A29Z&ske=2024-07-08T19%3A19%3A29Z&sks=b&skv=2023-11-03&sig=5kr33QivSmelIWw1tKHWS7EUkPtpaWiZ21ulVu/SQn
```


Ilustración 11. Respuesta del servidor

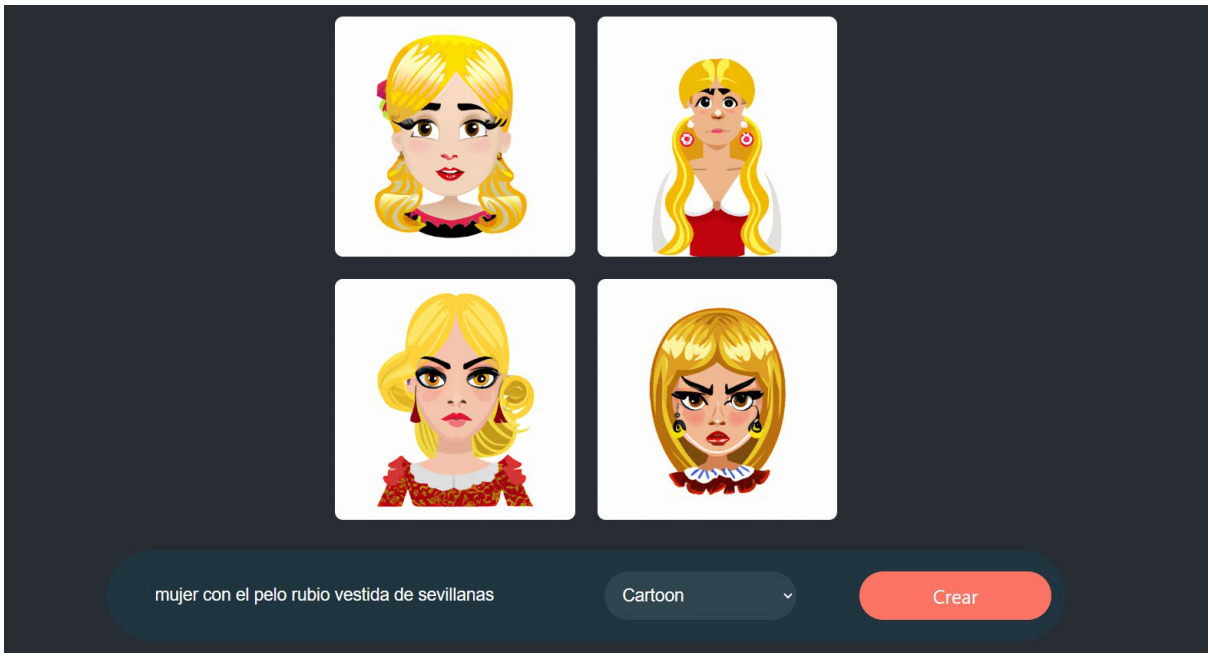

Ilustración 12. Prueba generando otro Avatar con diferente estilo

4.3 Implementación avatares por imagen

1. Configuración del Entorno

Para comenzar a usar la API de Cutout.Pro, es necesario obtener una clave API desde su plataforma. Esta clave se utilizará para autenticar las solicitudes API. Se puede obtener una simplemente registrándose en la plataforma y creando un perfil.

2. Ejemplo de Solicitud API

El siguiente ejemplo muestra cómo realizar una solicitud para convertir una imagen en un avatar usando la herramienta Cartoon Selfie:

```
import axios from 'axios';
import FormData from 'form-data';
import fs from 'fs';

const apiKey = 'your_api_key';
const imagePath = 'path_to_image.jpg';
const cartoonType = '10';

const formData = new FormData();
formData.append('file', fs.createReadStream(imagePath));

axios.post(`https://www.cutout.pro/api/v1/cartoonSelfie?cartoonType=${cartoonType}`, formDa
    headers: {
        'APIKEY': apiKey,
        ...formData.getHeaders()
    },
    responseType: 'arraybuffer'
})
.then(response => {
    fs.writeFileSync('avatar.png', response.data);
    console.log('Avatar saved as avatar.png');
})
.catch(error => {
    console.error('Error converting photo:', error);
});
```

Ilustración 13. Código para enviar la solicitud en Cut.pro

Límites de Uso y Restricciones

Cutout.Pro establece ciertos límites y restricciones para el uso de su API:

- La API permite un máximo de 5 solicitudes por segundo.
- Los usuarios pueden aumentar este límite solicitando un plan de suscripción superior.
- La resolución máxima soportada es de 4096x4096 píxeles.
- El tamaño máximo del archivo de imagen es de 15 MB.
- La API soporta los formatos de imagen más comunes, incluyendo PNG, JPG, JPEG, BMP, y WEBP.

Cutout.Pro proporciona una poderosa API para el procesamiento de imágenes que puede ser utilizada para una amplia variedad de aplicaciones. Desde la creación de avatares personalizados hasta la mejora y restauración de fotos, las herramientas de Cutout.Pro facilitan la realización de tareas complejas de edición de imágenes con una alta calidad y precisión. Integrar esta API en aplicaciones puede ofrecer a los usuarios finales funcionalidades avanzadas y una experiencia de usuario mejorada.

Implementación de la herramienta

En las siguientes capturas se muestra la generación de un avatar con diferentes estilos a partir de subir dos imágenes diferentes.

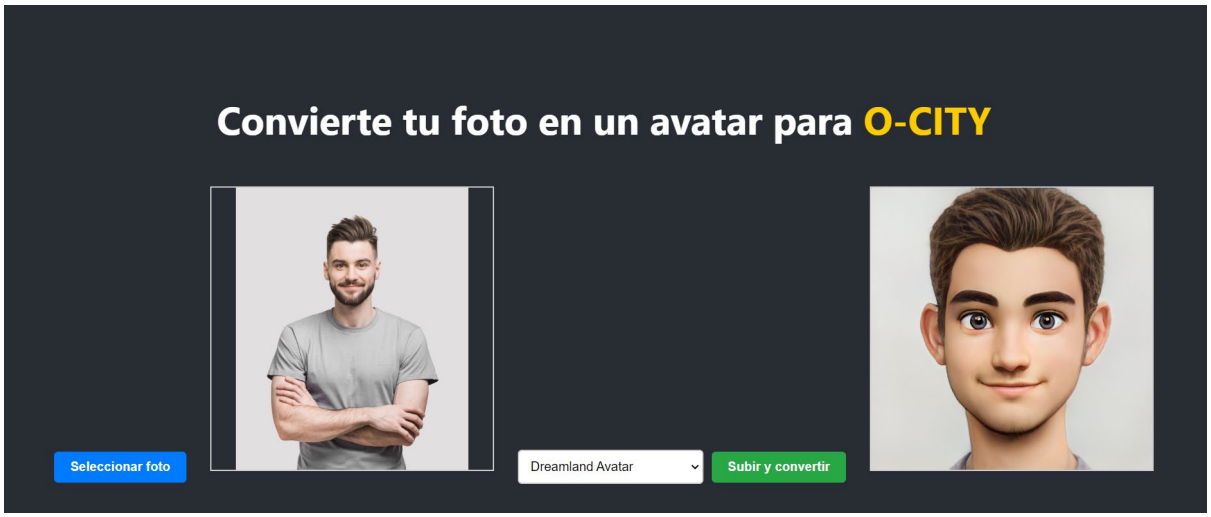

Ilustración 14. Prueba con una imagen y estilo de avatar dreamland

Ilustración 15. Prueba con estilo de avatar comic

Ilustración 16. Conversión de otra foto a avatar estilo hada

Ilustración 17. Conversión de foto a avatar estilo pixar

Ilustración 18. Conversión de foto a un Avatar estilo dreamland

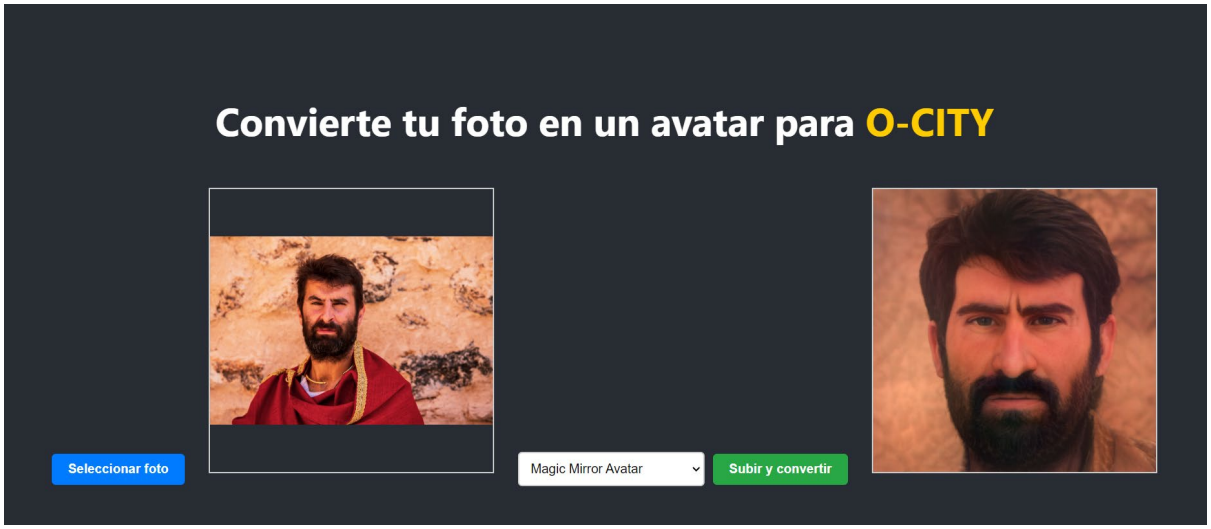

Ilustración 19. Conversión de foto a un Avatar estilo mágico

Contamos con un botón para seleccionar una imagen con un tipo de formato de imagen válido (si no es una imagen no se podrá seleccionar el archivo) y un selector de los diferentes tipos de estilos para aplicar al avatar. Una vez seleccionado, lo convertimos en avatar y descargamos el avatar que más nos guste.

Obtenemos esta respuesta del servidor:

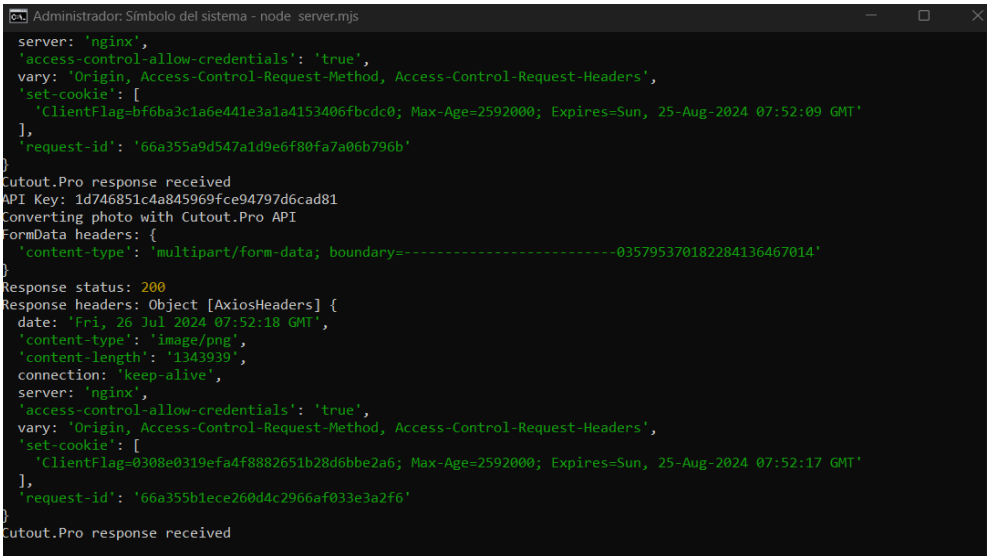

Ilustración 20. Respuesta del servidor Cutout.pro

Problemas de implementación resueltos:

Durante el desarrollo de la aplicación de generación de avatares por IA, surgieron varios desafíos técnicos y de diseño. A continuación, se detallan algunos de los problemas más significativos y cómo se resolvieron:

Problema 1: Manejo de Límites de Peticiones de la API de OpenAI

Descripción**:** La API de OpenAI impone límites estrictos en la cantidad de solicitudes que se pueden hacer por minuto y por día. Esto generaba errores de "Rate Limit Exceeded" cuando los usuarios intentaban generar múltiples avatares en un corto período.

Solución**:** Se implementó una función de throttling para limitar la cantidad de solicitudes que se pueden enviar en un período de tiempo determinado.

Problema 2: Validación de Nombres de Archivos al Importar Avatares

Descripción**:** Se requería que los usuarios solo pudieran importar archivos de imágenes que contengan la palabra "avatar" en el nombre del archivo. No es posible limitar los nombres de archivo directamente en el input de tipo `file`.

Solución**:** Se manejó la validación del nombre del archivo una vez seleccionado mediante JavaScript. Si el archivo no cumple con el criterio, se muestra una alerta y se restablece el valor del input.

Problema 3: Descarga de Imágenes con Fondo Transparente

Descripción**:** Se necesitaba que los avatares generados pudieran descargarse con un fondo transparente para mejorar la integración en diversas aplicaciones y plataformas.

Solución**:** Se configuró la API de OpenAI para generar imágenes con fondos transparentes y se manejó correctamente la descarga de las imágenes en formato PNG.

Problema 4: Consistencia de Capitalización en Rutas de Importación

Descripción**:** Hubo problemas con la inconsistencia en la capitalización de los nombres de archivos y rutas de importación, lo que causaba errores en sistemas sensibles a mayúsculas y minúsculas.

Solución**:** Se aseguró de que todas las rutas de importación y los nombres de archivos usaran la misma capitalización de manera consistente. Esto incluyó renombrar archivos y actualizar todas las importaciones correspondientes.

Capítulo 5. Testeo de la aplicación

Se ha utilizado Selenium para el testeo de la aplicación, en la siguiente imagen se muestran los resultados obtenidos al hacer un testeo:

	Command	Target	Value
1	✓ *open*	/	
2	✓ *set window size*	1326x732	
3	✓ *click*	css=.start-button	
4	✓ *click*	css=.search-input	
5	✓ *type*	css=.search-input	hombre de 50 años con pelo blanco y barba
6	✓ *click*	css=.generate-btn	
7	✓ *click*	css=img:nth-child(4)	
8	✓ *click*	css=img:nth-child(3)	
9	✓ *click*	css=.download-btn	

Ilustración 21. Testeo Selenium avatares generados por texto

	Command	Target
1	✓ *open*	/
2	✓ *set window size*	1552x832
3	✓ *click*	css=.photo-to-avatar-button
4	✓ *click*	css=.select-button
5	✓ *click*	css=.upload-button

Ilustración 22. Testeo Selenium avatares generados por imagen

Capítulo 6. Conclusiones

La aplicación de generación de avatares por IA desarrollada en este proyecto demuestra la capacidad y versatilidad de las tecnologías actuales para crear soluciones interactivas y personalizadas. A través del uso de React para el frontend, Express para el backend y la API de OpenAI para la generación de imágenes, se ha construido una plataforma robusta que permite a los usuarios generar, seleccionar y descargar avatares personalizados. Además, también cuenta con una herramienta para convertir fotos con la cara de una persona en avatares con diferentes estilos gracias a la API de Cutout.pro.

Ventajas y Logros

Se han conseguido los siguientes hitos en la aplicación:

Personalización y Flexibilidad: La aplicación permite una amplia gama de personalizaciones, desde la descripción del avatar hasta la selección de estilos, lo que ofrece a los usuarios una experiencia única y personalizada.

Uso Eficiente de API: La implementación de técnicas de throttling y manejo de errores asegura que las limitaciones de la API de OpenAI se manejen de manera efectiva, proporcionando una experiencia fluida al usuario.

Interfaz de Usuario Intuitiva: La interfaz de usuario es simple e intuitiva, facilitando la interacción con la aplicación, incluso para usuarios sin experiencia técnica.

Descarga de Imágenes de Alta Calidad: La capacidad de descargar avatares con fondo transparente mejora la usabilidad y versatilidad de los avatares en diversas plataformas.

Conversión de imágenes en Avatares: La aplicación incluye una herramienta que convierte con precisión una foto donde se encuentre una cara a un avatar eligiendo entre varios estilos y si se desea cuerpo entero o sólo la cabeza.

La aplicación de generación de avatares por IA ha demostrado ser una herramienta poderosa y flexible, capaz de proporcionar una experiencia de usuario personalizada y de alta calidad. Sin embargo, siempre hay espacio para mejorar y expandir las funcionalidades. Al abordar las áreas de mejora y explorar nuevas oportunidades de desarrollo, esta aplicación puede seguir evolucionando para satisfacer mejor las necesidades de los usuarios y mantenerse relevante en un entorno tecnológico en constante cambio.

6.1. Trabajo futuro

-Ampliación de Estilos de Avatares:

Más Opciones de Estilo: Incorporar más estilos de avatares, como estilos de cómic, ilustraciones y estilos específicos de diferentes culturas.

Personalización Detallada: Permitir personalizaciones más detalladas, como la selección de accesorios, ropa y expresiones faciales.

-Optimización del Rendimiento:

Caching: Implementar técnicas de caching para reducir la cantidad de solicitudes a la API y mejorar los tiempos de respuesta.

Optimización de Carga: Mejorar la eficiencia de carga de imágenes y recursos para ofrecer una experiencia más rápida y fluida.

-Animación de Avatares:

Animaciones de voz: Hacer que la boca de los avatares se mueva cuando digan algo.

Pequeñas animaciones: Para que gesticulen los brazos y la cabeza.

Referencias

[1] 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/

[2] 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

[3] 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/

[4] OpenAI. (2023). OpenAI API Documentation. OpenAI. https://platform.openai.com/docs (acceso Mar. 14 2023).

[5] Meta. (2023). React Documentation. React. https://reactjs.org/docs/getting-started.html (acceso Mar. 14 2023).

[6] Express.js. (2023). Express Documentation. Express.js. https://expressjs.com/ (acceso Mar. 14 2023).

[7] OpenJS Foundation. (2023). Node.js Documentation. Node.js. https://nodejs.org/en/docs/ (acceso Mar. 14 2023).

[8] Mlot S. (2022). Developers Can Now Integrate DALL-E 2 Image Generator Into Their Apps. PCMag. https://www.pcmag.com/news/developers-can-now-integrate-dall-e-2-image-generator-into-their-apps (acceso Mar. 20 2024).

[9] AI Hungry. (2024). DALL-E Pricing Plans and Cost Breakdown - Updated 2024. AI Hungry. https://aihungry.com/tools/dalle/pricing (acceso Mar. 21 2024).

[10] Cutout.Pro. API Documentation. https://www.cutout.pro/api-document/ (acceso Jul. 8 2024).

[11] DeepAI. (2024). API Reference. Retrieved from https://deepai.org/machine-learning-model/toonify (acceso Abr. 18 2024).

[12] Avatarify. (2024). GitHub Repository. Retrieved from https://github.com/alievk/avatarify (acceso Jul. 8 2024).

[13] Cartoonify. (2024). API Documentation. Retrieved from https://cartoonify.de/api-document/ (acceso Jul. 8 2024).

[14] Goodfellow I. Pouget-Abadie J. Mirza M. Xu B. Warde-Farley D. Ozair S. ... & Bengio Y. (2014). Generative adversarial nets. Advances in neural information processing systems 27. https://arxiv.org/abs/1406.2661 (acceso Jul. 11 2024).

[15] He K. Zhang X. Ren S. & Sun J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778). https://arxiv.org/abs/1512.03385 (acceso Jul. 12 2024).

[16] Ronneberger O. Fischer P. & Brox T. (2015). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer Cham. https://arxiv.org/abs/1505.04597 (acceso Jul. 14 2024).

[17] Kingma D. P. & Welling M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. https://arxiv.org/abs/1312.6114 (acceso Jul. 17 2024).

[18] Zalando Research. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. https://github.com/zalandoresearch/fashion-mnist (acceso Jul. 18 2024).

[19] NVIDIA. (2024). NVIDIA Deep Learning AI. https://www.nvidia.com/en-us/deep-learning-ai/ (acceso Jul. 18 2024).

[20] TensorFlow. (2024). TensorFlow Documentation. https://www.tensorflow.org/ (acceso Jul. 18 2024).

[21] PyTorch. (2024). PyTorch Documentation. https://pytorch.org/ (acceso Jul. 18 2024).

[22] Microsoft Azure. (2024). Azure Cognitive Services. https://azure.microsoft.com/en-us/services/cognitive-services/ (acceso Jul. 18 2024).

[23] Google Cloud. (2024). Google Cloud Vision API. https://cloud.google.com/vision (acceso Jul. 20 2024).

[24] Adobe. (2024). Adobe Photoshop and Sensei AI. https://www.adobe.com/sensei.html (acceso Jul. 20 2024).

[25] IBM. (2024). IBM Watson Visual Recognition. https://www.ibm.com/cloud/watson-visual-recognition (acceso Jul. 21 2024).

[26]Amazon Web Services. (2024). AWS Rekognition. https://aws.amazon.com/rekognition/ (acceso Jul. 21 2024).

[27] Ng A. (2016). Machine Learning Yearning.

[28] LeCun Y. Bengio Y. & Hinton G. (2015). Deep learning. Nature 521(7553) 436-444.

[29] Russell S. & Norvig P. (2016). Artificial Intelligence: A Modern Approach.

[30] Dean J. Corrado G. & Monga R. (2012). Large Scale Distributed Deep Networks.

[31] Schmidhuber J. (2015). Deep learning in neural networks: An overview. Neural Networks 61 85-117.

[32] Vaswani A. Shazeer N. Parmar N. Uszkoreit J. Jones L. Gomez A. N. Kaiser Ł. & Polosukhin I. (2017). Attention is all you need.

[33] Bojarski M. Testa D. D. Dworakowski D. Firner B. Flepp B. Goyal P. ... & Zhang X. (2016). End to end learning for self-driving cars.

[34] Sun C. Shrivastava A. Singh S. & Gupta A. (2017). Revisiting unreasonable effectiveness of data in deep learning era.

[35] Karpathy A. & Fei-Fei L. (2015). Deep visual-semantic alignments for generating image descriptions.

[36] Joulin A. Grave E. Bojanowski P. & Mikolov T. (2017). Bag of tricks for efficient text classification.

[37] Hochreiter S. & Schmidhuber J. (1997). Long short-term memory. Neural computation 9(8) 1735-1780.

[38] Radford A. Narasimhan K. Salimans T. & Sutskever I. (2018). Improving language understanding by generative pre-training.

[39] Devlin J. Chang M. W. Lee K. & Toutanova K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding.

[40] Zhang X. Zhao J. & LeCun Y. (2015). Character-level convolutional networks for text classification.

[41] Dosovitskiy A. Beyer L. Kolesnikov A. Weissenborn D. Zhai X. Unterthiner T. ... & Houlsby N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale.

[42] Brown T. Mann B. Ryder N. Subbiah M. Kaplan J. D. Dhariwal P. ... & Amodei D. (2020). Language models are few-shot learners.

[43] Krizhevsky A. Sutskever I. & Hinton G. E. (2012). ImageNet classification with deep convolutional neural networks.

[44] Silver D. Schrittwieser J. Simonyan K. Antonoglou I. Huang A. Guez A. ... & Hassabis D. (2017). Mastering the game of Go without human knowledge.

[45] Hinton G. Deng L. Yu D. Dahl G. E. Mohamed A. r. Jaitly N. ... & Kingsbury B. (2012). Deep neural networks for acoustic modeling in speech recognition.

Link repositorio Github: https://github.com/Gausek17/proyecto

Anexo 1. Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030.

Grado de relación del trabajo con los 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 especifico 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.