Una caricatura de una persona

El contenido generado por IA puede ser incorrecto.UNIVERSIDAD DE BURGOS ESCUELA POLITÉCNICA SUPERIOR



Máster en Ingeniería en Informática

TFM del Máster en Ingeniería Informática

**Avatar**

**Estudio del Arte**



Presentado por Manuel C. León Rivera

en Universidad de Burgos — 20 de enero de 2021 Tutor: María Belén Vaquerizo García

**Índice General**

[**Plan de Proyecto Software** 2](#_Toc201497008)

[1.1 Introducción 2](#_Toc201497009)

[1.2 Planificación temporal 2](#_Toc201497010)

[1.3 Estudio de viabilidad 11](#_Toc201497011)

[1.4 Beneficios del Proyecto 13](#_Toc201497012)

[**Estudio del Arte** 16](#_Toc201497013)

[1.1 Sistema experto basado en reglas 16](#_Toc201497014)

[1.2 Análisis de Proveedores 18](#_Toc201497015)

[**D-ID** 21](#_Toc201497016)

[1.3 Visión general 21](#_Toc201497017)

[1.4 Arquitectura general 21](#_Toc201497018)

[1.5 Funcionalidades principales 21](#_Toc201497019)

[1.6 Ejemplo de llamada 22](#_Toc201497020)

[1.7 Características técnicas 22](#_Toc201497021)

[**Synthesia** 24](#_Toc201497022)

[1.1 Visión general 24](#_Toc201497023)

[1.2 Arquitectura general 24](#_Toc201497024)

[1.3 Funcionalidades principales 24](#_Toc201497025)

[1.4 Ejemplo de llamada 24](#_Toc201497026)

[1.5 Características técnicas 25](#_Toc201497027)

[**HeyGen** 27](#_Toc201497028)

[1.1 Visión general 27](#_Toc201497029)

[1.2 Arquitectura general 27](#_Toc201497030)

[1.3 Funcionalidades principales 27](#_Toc201497031)

[1.4 Ejemplo de llamada 27](#_Toc201497032)

[1.5 Características técnicas 28](#_Toc201497033)

[**Elay** 29](#_Toc201497034)

[1.1 Visión general 29](#_Toc201497035)

[1.2 Arquitectura general 29](#_Toc201497036)

[1.3 Funcionalidades principales 29](#_Toc201497037)

[1.4 Ejemplo de llamada 30](#_Toc201497038)

[1.5 Características técnicas 30](#_Toc201497039)

[**VEED** 32](#_Toc201497040)

[1.1 Visión general 32](#_Toc201497041)

[1.2 Arquitectura general 32](#_Toc201497042)

[**Colossyan** 33](#_Toc201497043)

[1.1 Visión general 33](#_Toc201497044)

[1.2 Arquitectura general 33](#_Toc201497045)

[1.3 Funcionalidades principales 33](#_Toc201497046)

[1.4 Ejemplo de llamada 33](#_Toc201497047)

[1.5 Características técnicas 34](#_Toc201497048)

[**Speechify** 36](#_Toc201497049)

[1.1 Visión general 36](#_Toc201497050)

[1.2 Arquitectura general 36](#_Toc201497051)

[1.3 Funcionalidades principales 36](#_Toc201497052)

[1.4 Ejemplo de llamada 36](#_Toc201497053)

[1.5 Características técnicas 37](#_Toc201497054)

[**Tavus** 38](#_Toc201497055)

[1.1 Visión general 38](#_Toc201497056)

[1.2 Arquitectura general 38](#_Toc201497057)

[1.3 Funcionalidades principales 38](#_Toc201497058)

[1.4 Ejemplo de llamada 38](#_Toc201497059)

[1.5 Características técnicas 39](#_Toc201497060)

[**Runway** 40](#_Toc201497061)

[1.1 Visión general 40](#_Toc201497062)

[1.2 Arquitectura general 40](#_Toc201497063)

[1.3 Funcionalidades principales 40](#_Toc201497064)

[1.4 Ejemplo de llamada 40](#_Toc201497065)

[1.5 Características técnicas 41](#_Toc201497066)

[**A2E** 42](#_Toc201497067)

[1.1 Visión general 42](#_Toc201497068)

[1.2 Arquitectura general 42](#_Toc201497069)

[1.3 Funcionalidades principales 42](#_Toc201497070)

[1.4 Ejemplo de llamada 43](#_Toc201497071)

[1.5 Características técnicas 44](#_Toc201497072)

[**AI Studios** 45](#_Toc201497073)

[1.1 Visión general 45](#_Toc201497074)

[1.2 Arquitectura general 45](#_Toc201497075)

[1.3 Funcionalidades principales 45](#_Toc201497076)

[1.4 Ejemplo de llamada 45](#_Toc201497077)

[1.5 Características técnicas 46](#_Toc201497078)

[**Akool** 48](#_Toc201497079)

[1.1 Visión general 48](#_Toc201497080)

[1.2 Arquitectura general 48](#_Toc201497081)

[1.3 Funcionalidades principales 48](#_Toc201497082)

[1.4 Ejemplo de llamada 49](#_Toc201497083)

[1.5 Características técnicas 50](#_Toc201497084)

[**Vidnoz** 51](#_Toc201497085)

[1.1 Visión general 51](#_Toc201497086)

[1.2 Arquitectura general 51](#_Toc201497087)

[1.3 Funcionalidades principales 51](#_Toc201497088)

[1.4 Ejemplo de llamada 51](#_Toc201497089)

[1.5 Características técnicas 52](#_Toc201497090)

[**Pipio** 53](#_Toc201497091)

[1.1 Visión general 53](#_Toc201497092)

[1.2 Arquitectura general 53](#_Toc201497093)

[1.3 Funcionalidades principales 53](#_Toc201497094)

[1.4 Ejemplo de llamada 53](#_Toc201497095)

[1.5 Características técnicas 54](#_Toc201497096)

[**Creatify** 55](#_Toc201497097)

[1.1 Visión general 55](#_Toc201497098)

[1.2 Arquitectura general 55](#_Toc201497099)

[1.3 Funcionalidades principales 55](#_Toc201497100)

[1.4 Ejemplo de llamada 55](#_Toc201497101)

[1.5 Características técnicas 56](#_Toc201497102)

[**Normalización de los datos** 57](#_Toc201497103)

[1.1 Normalización de columnas para el sistema experto basado en reglas 57](#_Toc201497104)

[1.2 Sistema de reglas. Cálculo de la Similitud Total en el Sistema CBR 62](#_Toc201497105)

[1.3 Caso de ejemplo 67](#_Toc201497106)

[1.4 Proveedor seleccionado para el proyecto 71](#_Toc201497107)

[**Proveedores fuera de estudio** 72](#_Toc201497108)

[**Especificación de requisitos** 73](#_Toc201497109)

[1.1 Introducción 73](#_Toc201497110)

[1.2 Objetivos generales 73](#_Toc201497111)

[1.3 Catálogo de requisitos 73](#_Toc201497112)

[1.4 ¿Se han logrado los objetivos al acabar el proyecto? 74](#_Toc201497113)

[1.5 Casos de Uso 75](#_Toc201497114)

[**Problemas encontrados** 84](#_Toc201497115)

**Índice de Figuras**

[Figura A0: Diagrama de Gantt 3](#_Toc201497147)

[Figura A1: Milestone Burndown - Sprint 1 4](#_Toc201497148)

[Figura A2: Tareas – Sprint 1 4](#_Toc201497149)

[Figura A3: Milestone Burndown - Sprint 3 5](#_Toc201497150)

[Figura A4: Milestone Burndown - Sprint 3 6](#_Toc201497151)

[Figura A5: Tareas – Sprint 3 6](#_Toc201497152)

[Figura A6: Milestone Burndown - Sprint 4 7](#_Toc201497153)

[Figura A7: Tareas – Sprint 4 7](#_Toc201497154)

[Figura A8: Milestone Burndown - Sprint 5 8](#_Toc201497155)

[Figura A9: Tareas – Sprint 5 8](#_Toc201497156)

[Figura A10: Milestone Burndown - Sprint 6 9](#_Toc201497157)

[Figura A11: Tareas – Sprint 6 9](#_Toc201497158)

[Figura A12: Milestone Burndown - Sprint 7 10](#_Toc201497159)

[Figura A13: Tareas – Sprint 7 10](#_Toc201497160)

[Figura A14: Sistema basado en reglas GoAvatar 17](#_Toc201497161)

[Figura A15: Ejemplo de llamada D-ID 22](#_Toc201497162)

[Figura A16: Ejemplo de llamada Synthesia 25](#_Toc201497163)

[Figura A17: Ejemplo de llamada HeyGen 28](#_Toc201497164)

[Figura A18: Ejemplo de llamada Elay 30](#_Toc201497165)

[Figura A19: Ejemplo de llamada Colossyan 34](#_Toc201497166)

[Figura A20: Ejemplo de llamada Speechify 37](#_Toc201497167)

[Figura A21: Ejemplo de llamada Tavus 39](#_Toc201497168)

[Figura A22: Ejemplo de llamada Runway 41](#_Toc201497169)

[Figura A23: Ejemplo de llamada A2E 43](#_Toc201497170)

[Figura A15: Info precio A2E - https://www.a2e.ai/volume-price-of-a2e-api/ 44](#_Toc201497171)

[Figura A24: Ejemplo de llamada AI Studios 46](#_Toc201497172)

[Figura A25: Ejemplo de llamada Akool 49](#_Toc201497173)

[Figura A25: Ejemplo de llamada Vidnoz 51](#_Toc201497174)

[Figura A26: Ejemplo de llamada Pipio 54](#_Toc201497175)

[Figura A27: Ejemplo de llamada Creatify 56](#_Toc201497176)

[Figura B1: Casos de Uso 75](#_Toc201497177)

**Índice de Tablas**

[Tabla A1: Precios sofware 12](#_Toc201498106)

[Tabla A2: Viabilidad legal – Entorno de desarrolloy herramientas 16](#_Toc201498107)

[Tabla A3: Viabilidad legal – Servicios API utilizados 16](#_Toc201498108)

[Tabla A4: Métodos D-ID 22](#_Toc201498109)

[Tabla A5: Características D-ID 24](#_Toc201498110)

[Tabla A6: Métodos Synthesia 25](#_Toc201498111)

[Tabla A7: Características Synthesia 27](#_Toc201498112)

[Tabla A8: Métodos HeyGen 28](#_Toc201498113)

[Tabla A9: Características HeyGen 29](#_Toc201498114)

[Tabla A10: Métodos Elay 30](#_Toc201498115)

[Tabla A11: Características Elay 32](#_Toc201498116)

[Tabla A12: Métodos Colossyan 34](#_Toc201498117)

[Tabla A13: Características Colossyan 36](#_Toc201498118)

[Tabla A14: Métodos Speechify 37](#_Toc201498119)

[Tabla A15: Características Speechify 38](#_Toc201498120)

[Tabla A16: Métodos Tavus 39](#_Toc201498121)

[Tabla A17: Características Tavus 40](#_Toc201498122)

[Tabla A18: Características Runway 42](#_Toc201498123)

[Tabla A19: Métodos A2E 44](#_Toc201498124)

[Tabla A20: Características A2E 45](#_Toc201498125)

[Tabla A21: Métodos AI Studios 46](#_Toc201498126)

[Tabla A22: Características AI Studios 48](#_Toc201498127)

[Tabla A23: Métodos Akool 49](#_Toc201498128)

[Tabla A24: Características Akool 51](#_Toc201498129)

[Tabla A25: Métodos Vidnoz 52](#_Toc201498130)

[Tabla A26: Características Vidnoz 53](#_Toc201498131)

[Tabla A27: Métodos Pipio 54](#_Toc201498132)

[Tabla A28: Características Pipio 55](#_Toc201498133)

[Tabla A29: Métodos Creatify 56](#_Toc201498134)

[Tabla A30: Características Creatify 57](#_Toc201498135)

[Tabla A31: TABLA GENERAL PROVEEDORES (SIN NORMALIZAR) 61](#_Toc201498136)

[Tabla A32: TABLA GENERAL PROVEEDORES (NORMALIZADA) 62](#_Toc201498137)

[Tabla A33: Reglas y pesos 67](#_Toc201498138)

[Tabla A34: Tabla Proveedores Normalizadas (2025\_V1) 70](#_Toc201498139)

[Tabla A35: Tabla Proveedores Normalizadas (2025\_V1) – Resultados Similitud 71](#_Toc201498140)

[Tabla A36: Tabla Proveedores Normalizadas (2025\_V1) – Entradas al sistema 72](#_Toc201498141)

[Tabla B1: Tabla de objetivos 76](#_Toc201498142)

[Tabla B2: CU-01 Inicio de sesión 77](#_Toc201498143)

[Tabla B3: CU-02 Creación de vídeos 78](#_Toc201498144)

[Tabla B4: CU-03 Gestión de vídeos 79](#_Toc201498145)

[Tabla B5: CU-04 Configurar el comportamiento del Avatar en Tiempo Real 80](#_Toc201498146)

[Tabla B6: CU-05 Conversación con avatar en Tiempo Real 81](#_Toc201498147)

[Tabla B7: CU-06 Buscar proveedor óptimo 82](#_Toc201498148)

[Tabla B8: CU-07 Edición de Proveedores 83](#_Toc201498149)

[Tabla B9: CU-08 Configuración de pesos 84](#_Toc201498150)

# **Plan de Proyecto Software**

1. Introducción

En este primer apartado se detalla cómo se ha planificado el desarrollo del proyecto titulado “Avatar”.

Este proyecto tiene como objetivo realizar un estudio del estado del arte de las principales tecnologías existentes en el mercado que permiten generar avatares virtuales realistas a partir de texto o voz, así como desarrollar una aplicación funcional que, mediante razonamiento basado en casos (CBR) [1], recomiende la API más adecuada según los requisitos del usuario. Aunque, a mitad del desarrollo, se cambió por un sistema experto basado en reglas [2] tal y como se describe en la memoria.

La planificación incluye tanto una fase de investigación tecnológica como otra de desarrollo práctico en un entorno controlado, utilizando contenedores Docker [2] para facilitar la portabilidad del proyecto, y herramientas de control de versiones para el seguimiento del desarrollo. Asimismo, se tendrá en cuenta la viabilidad técnica y económica, estableciendo estimaciones de tiempo, costes y beneficios que podría aportar la solución propuesta.

1. Planificación temporal

Durante las primeras reuniones de orientación, se identificaron los pasos iniciales necesarios para abordar correctamente el desarrollo del proyecto. Como metodología de trabajo, se optó por utilizar SCRUM, de forma que se establecen sprints iterativos y funcionales en los que se planifican tareas concretas con entregables parciales.

Desde el comienzo del proyecto se ha creado un repositorio en GitHub, donde se gestionará todo el código fuente, documentación y versiones del sistema. Para una organización visual y ágil de las tareas, se ha incorporado la extensión ZenHub, que permite definir Issues, agruparlas por Sprints y realizar un seguimiento del avance mediante su sistema de Boards.

Cada sprint se inicia con una reunión de planificación en la que se definen las funcionalidades a implementar. El enfoque busca que en cada iteración se obtenga un prototipo funcional —aunque incompleto—, asegurando así una entrega incremental y continua del producto, susceptible de mejora progresiva.

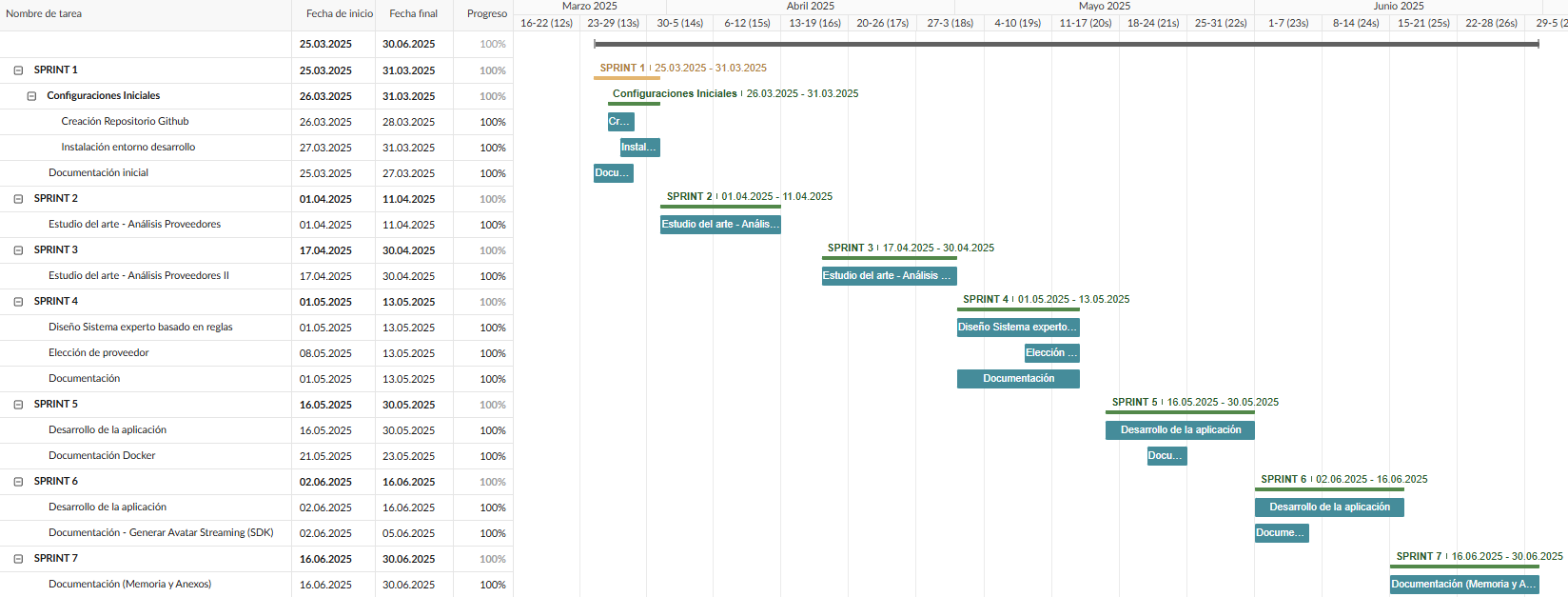
****

Figura A0: Diagrama de Gantt

**SPRINT 1 – 25/03/2025 – 02/04/2025**

Durante esta primera iteración, más corta que el resto, se ha orientado a establecer un entorno de trabajo funcional, con herramientas acordes a las necesidades del proyecto tanto para la gestión como para el desarrollo del mismo.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura A1: Milestone Burndown - Sprint 1

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Figura A2: Tareas – Sprint 1

**SPRINT 2 – 01/04/2025 – 16/04/2025**

En este Sprint se inició el estudio del arte, comenzando con los tres primeros proveedores: D-ID, Synthesia y HeyGen. En este inicio se tardó más porque tuve que ir probando las maneras en las que era más efectiva probar una API. Vi los primeros problemas: que algunas de estas APIs son de pago y no se pueden pagar previo pago. La tarea “Gestión de tareas” se basó en adaptarme a buscar una solución para hacer un estudio que fuera efectivo y pudiera normalizar con el resto de proveedores, como tomar valores de la resolución de vídeo o del número de avatares que ofrecen.

Gráfico

El contenido generado por IA puede ser incorrecto.

Figura A3: Milestone Burndown - Sprint 3

**SPRINT 3 – 17/04/2025 – 30/04/2025**

En este Sprint se estudió las características técnicas de los proveedores: Elay, VEED, Colossyan, Speechify, Tavus, Rinway, A2R, AI Studios, Akool y Vidnoz. Además, de ir realizando los primeros esbozos del anexo de este documento.

Este Sprint supuso horas de estudio sobre lo que ofrecían los distintos proveedores, iba apuntando todo en una hoja Excel para poder cotejar diferencias y preparando la documentación.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura A4: Milestone Burndown - Sprint 3

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura A5: Tareas – Sprint 3

**SPRINT 4 – 01/05/2025 – 16/05/2025**

Este Sprint tiene una tarea muy importante: “Documentación – Configuración de pesos y elección de proveedor”. Inicialmente, al tener todos los proveedores, me di cuenta de que, al no poder acceder a muchas de las APIs, el Razonamiento Basado en Casos (CBR) no tenía sentido, por lo que optamos por cambiarlo a un sistema experto basado en reglas. En esta tarea definí las reglas que harían decidir si un proveedor es mejor que otro dependiendo de unas entradas. Se aportaron unos pesos (importancia) a cada característica técnica que mejorara el valor o lo empeorase. La definición de reglas junto con los cambios en el modelo de reglas supuso el mayor tiempo en esta iteración. “Casos de Uso” y “Especificación de Requisitos” son tareas de documentación inicial.

Se hicieron pruebas sobre el sistema basado en reglas [Tabla Proveedores Normalizada (2025\_V1).xlsx] donde se puede simular el proceso de forma “manual y directa” en excel.

Gráfico

El contenido generado por IA puede ser incorrecto.

Figura A6: Milestone Burndown - Sprint 4

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Figura A7: Tareas – Sprint 4

**SPRINT 5 – 16/05/2025 – 31/05/2025**

Aquí empecé a desarrollar la aplicación. Comencé con la pantalla de Login. Después continué con la parte del usuario “experto”, que es el usuario encargado de ver el mejor proveedor en la aplicación. Las pantallas que hice fueron: Gestión de Proveedores, Configuración de Pesos. Con la lógica de negocio. Aquí inicié la pantalla del usuario “admin”, con sus cuatro botones. Además tuve que documentarme sobre cómo funciona Docker, pues la aplicación iba a ser distribuida y quería saber cómo podía empaquetar toda la configuración en un contenedor cerrado para migrar la aplicación de forma fácil.

Este SPRINT fue también muy importante, aquí tuve que tomar muchas decisiones técnicas como qué partes de la API de D-ID podía ir desarrollando sin problemas y cómo podía hacer la aplicación lo más sencilla posible de cara al gestor que la utilizara.

Aunque no aparezca como una tarea al uso, también dediqué mucho tiempo en mirar documentación de node cada vez que me topaba con algo que no sabía desarrollar.

Gráfico

El contenido generado por IA puede ser incorrecto.

Figura A8: Milestone Burndown - Sprint 5

Texto, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Figura A9: Tareas – Sprint 5

**SPRINT 6 – 01/06/2025 – 16/06/2025**

Aquí continué con la parte del usuario de administrados: pantalla de creación de vídeos, pantalla de Gestión de Vídeos, Gestión de avatar en tiempo real y Gestión de Avatar. Estas tareas no aparecen dentro del marco del Sprint porque olvidé marcarlas en el programa y no salen.

La pantalla de Gestión de Avatar la tuve que dejar después de desarrollarla porque, como digo en la memoria, la API no genera Avatares Express, que son los que me permite el plan, contradiciendo lo que dice la documentación de D-ID.

Este Sprint fue el que más pruebas realicé llamando a la API de D-ID y donde gasté más créditos en vídeo. Dejé el Avatar Streaming para el final, pues requería de un proyecto tutorial que ofrece D-ID para usar su SDK.

Aquí fue donde dediqué más tiempo, en la parte técnica, pero, para mi suerte, node y react no me dieron tantos problemas como pensé en un principio.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Figura A10: Milestone Burndown - Sprint 6

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Figura A11: Tareas – Sprint 6

**SPRINT 7 – 16/06/2025 – 31/06/2025**

Este Sprint ya es el cierre. Las tareas que realizo son de documentación para el anexo y para la memoria.

Interfaz de usuario gráfica, Gráfico

El contenido generado por IA puede ser incorrecto.

Figura A12: Milestone Burndown - Sprint 7

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura A13: Tareas – Sprint 7

1. Estudio de viabilidad

El desarrollo del proyecto GoAvatar se ha planteado como si fuera realizado por un único desarrollador contratado durante 2 meses, trabajando 30 horas semanales, lo que representa un total de 240 horas.

1. **Coste de Personal:**

Se ha tomado como referencia un salario medio de 40 €/hora [3], acorde al mercado actual en España para perfiles de desarrolladores con experiencia en tecnologías como Node.js, React, MongoDB y APIs de inteligencia artificial. Esta tarifa refleja una dedicación profesional con nivel técnico intermedio-alto.

Para estimar el coste real que supondría este trabajador para la empresa, se han aplicado las cotizaciones oficiales vigentes para el Régimen General de la Seguridad Social en 2025 [4], que ascienden al 29,90 % sobre el salario bruto. El cálculo del coste por hora se realiza mediante la siguiente fórmula:

Multiplicado por las 240 horas estimadas de desarrollo:

* Coste total estimado del desarrollador: **13.694,40 €**

1. **Coste del hardware:**

Únicamente se requiere de un ordenador portátil estimado en 1000€ que tenga las capacidades necesarias para realizar un desarrollo como este (procesador y memoria).

La vida útil para amortizarse suele estar en 4 años (48 meses).

Entonces, el coste mensual es:

Como el proyecto dura 2 meses:

* Coste amortizado del hardware: **41,66€**

1. **Coste del sofware:**

En este apartado se describe todo el software que se utiliza en el proyecto durante los 2 meses:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tecnología / Servicio | Descripción | Precio mensual | Meses usados | Coste total (€) |
| Visual Studio Code | Editor de código principal, extensible y gratuito | 0 € | 2 | 0 € |
| Postman | Herramienta para pruebas de APIs REST | 0 € | 2 | 0 € |
| Ngrok (versión gratuita) | Túnel de red para exponer servidores locales, usado con webhooks | 0 € | 2 | 0 € |
| MongoDB Atlas (free tier) | Plataforma de base de datos remota usada en producción | 0 € | 2 | 0 € |
| MongoDB Compass | Interfaz gráfica para gestionar MongoDB | 0 € | 2 | 0 € |
| Git + GitHub | Sistema de control de versiones y plataforma de repositorios remotos | 0 € | 2 | 0 € |
| Docker | Plataforma de contenedores para despliegue de backend y frontend | 0 € | 2 | 0 € |
| Render.com | Hosting de contenedores (frontend + backend) | 14 € | 2 | 28 € |
| D-ID Plan Launch | API de generación de avatares | 30,70 € | 2 | 61,40 € |
| Total |  |  |  | **89,40 €** |

Tabla A1: Precios sofware

* Coste total del sofware: **89,40€**

1. **Costes indirectos:**

Los costes indirectos corresponden a aquellos gastos asociados al entorno de trabajo necesarios para el desarrollo del proyecto, pero que no forman parte directa del software ni del hardware. En este caso, se han considerado dos elementos principales:

Coste proveedor telecomunicaciones en España (media): 25€

Consumo eléctrico (estimación):

Según *Exclusivas Energéticas* y datos del mercado mayorista (OMIE), el precio promedio de la electricidad en España durante los primeros meses de 2025 es de aproximadamente **55 €/MWh** [5]

1. Uso: portátil + router = 75 W
2. Duración: 24 h/día × 30 días = 720 h/mes
3. Consumo mensual: 75 W × 180 h = 54 kWh
4. Coste mensual: 54 kWh × 0,055 €/kWh ≈ **2,97 €**

* Coste total de costes indirectos: 25 + 2,97 x 2 = **30,94 €**
* **RESUMEN:**

1. Coste de personal: **13.694,40 €**
2. Coste de hardware: **41,66€**
3. Coste de software: **89,40€**
4. Costes indirectos: **30,94**
5. **TOTAL = 13.856,4 €**
6. Beneficios del Proyecto

Este apartado describe los beneficios de la aplicación una vez esté en funcionamiento.

1. **Ahorro de tiempo, recursos humanos y equipo de vídeo:**

La generación automatizada de vídeos con avatares puede sustituir tareas de comunicación rutinaria (como FAQs, presentaciones institucionales, respuestas a clientes), reduciendo así la dependencia de personal humano en tareas repetitivas.

Por ejemplo, si una empresa genera 8 vídeos al mes para redes sociales, o atención al cliente y, de forma tradicional, cada vídeo costara 80€ en producirse, el ahorro mensual con GoAvatar sería de:

Si además, contamos con que nos ahorramos a un equipo técnico que atienda por videoconferencia servicio de 24 horas a los usuarios interesados, estaríamos ante un ahorro de:

* + Una persona puede trabajar 40h/semana como jornada completa.
  + Coste estimado mensual:

Y anualmente:

* **RESUMEN DE BENEFICIOS:**
  + Sólo los vídeos al año: 640€ x 12 = 7.680 € / año
  + Avatar online 24 horas: 120.930 € / año
  + TOTAL: 7.680€ + 120.930€ = **128.610 € / año**

Esto representa un **ahorro drástico** frente a un sistema tradicional, y refuerza la viabilidad económica del uso de avatares automatizados en contextos de atención al usuario, formación, o comunicación institucional.

1. **Aplicabilidad en distintos sectores**

El sistema desarrollado en el proyecto GoAvatar presenta una alta capacidad de reutilización y escalabilidad, lo que le permite ser aplicado en diversos contextos profesionales y organizacionales.

En el ámbito de la atención al cliente, GoAvatar puede actuar como un canal de respuesta automatizada mediante vídeos personalizados, ofreciendo soporte inmediato a preguntas frecuentes, guías de uso o consultas recurrentes sin la intervención directa de personal humano.

En el sector educativo, la herramienta puede emplearse para generar vídeos explicativos por materias, facilitando la comprensión de contenidos de forma visual y dinámica. Esto puede aplicarse tanto en educación reglada como en formación profesional o contenidos de e-learning.

Dentro de la administración pública, GoAvatar podría servir como canal de comunicación directa con la ciudadanía. Por ejemplo, para difundir mensajes institucionales, explicar trámites o responder a preguntas habituales, todo ello con una imagen personalizada y representativa del organismo.

En marketing digital y redes sociales, el sistema permitiría generar campañas audiovisuales segmentadas, de forma rápida y económica, adaptando mensajes según públicos, productos o épocas del año y publicando en redes sociales de forma automática.

1. Viabilidad legal

En el desarrollo del proyecto GoAvatar se han utilizado tecnologías y servicios de terceros que, en su mayoría, cuentan con licencias de uso abiertas y compatibles con proyectos académicos y no comerciales. A continuación, se detallan los principales elementos tecnológicos y su situación legal:

Entorno de desarrollo y herramientas:

|  |  |  |
| --- | --- | --- |
| Herramienta / Tecnología | Licencia | Uso en el proyecto |
| Visual Studio Code | MIT | Editor de código |
| Node.js | MIT | Backend y servidor |
| React | MIT | Interfaz frontend |
| MongoDB (local y Atlas) | SSPL (Server Side Public License) | Base de datos documentales |
| Docker | Apache 2.0 | Contenedores y despliegue |
| Git / GitHub | Open Source | Control de versiones y repositorio remoto |
| Ngrok | Freemium | Exposición de puertos para pruebas con webhook |
| Postman | Gratis / uso limitado | Pruebas de API y simulación de peticiones |
| MongoDB Compass | Gratis | Cliente gráfico para MongoDB |

Tabla A2: Viabilidad legal – Entorno de desarrolloy herramientas

Servicios API utilizados:

|  |  |  |
| --- | --- | --- |
| API / Plataforma | Licencia | Uso en el proyecto |
| D-ID (Plan Launch) | Licencia comercial | Generación de vídeos con avatar realista |
| Render.com | Freemium / pago | Hosting de backend y frontend |

Tabla A3: Viabilidad legal – Servicios API utilizados

Los servicios D-ID y Render se han utilizado mediante cuentas de usuario que aceptan sus respectivas condiciones de uso, las cuales permiten el acceso y prueba del servicio a cambio de una suscripción mensual. No se ha redistribuido su tecnología, ni se ha vulnerado el uso permitido.

El proyecto GoAvatar es legalmente viable en el contexto académico en el que ha sido desarrollado. Las herramientas empleadas son de libre uso o utilizadas bajo suscripción legítima, y no se incurre en vulneración de licencias ni en apropiación indebida de tecnología de terceros. En caso de evolución a producto comercial, sería necesario revisar las condiciones de licenciamiento de servicios como D-ID y Render para garantizar la conformidad legal.

# **Estudio del Arte**

1. Sistema experto basado en reglas

Un sistema experto [2] es una arquitectura de inteligencia artificial diseñada para reproducir el razonamiento humano experto a través de una base de conocimientos, un conjunto de hechos, y un motor de inferencia que aplica reglas lógicas para obtener conclusiones o recomendaciones. Cuando estas reglas se expresan en formato IF–THEN, hablamos de un sistema experto basado en reglas.

Estos sistemas constan habitualmente de tres componentes principales:

1. **Base de conocimientos**: un conjunto estructurado de reglas del tipo *si... entonces...* que modelan decisiones o acciones basadas en condiciones observables.
2. **Base de hechos:** el conjunto de datos concretos disponibles en un momento determinado sobre los que se aplican las reglas. No se refiere únicamente a la entrada del usuario ni a toda la base de datos, sino al subconjunto activo de información que se está evaluando en el proceso.
3. **Motor de inferencia**: el mecanismo que compara los hechos con las reglas, aplica aquellas que correspondan, y genera conclusiones o recomendaciones.

En el proyecto GoAvatar, se ha implementado un sistema experto basado en reglas para seleccionar el proveedor de avatares más adecuado en función de los criterios técnicos introducidos por el usuario (por ejemplo: precio máximo, duración mínima de vídeo, número de avatares, etc.).

Cabe señalar que el sistema no implementa reglas encadenadas ni estrategias de inferencia multietapa como encadenamiento hacia adelante o hacia atrás. En su lugar, aplica reglas independientes y evaluaciones directas sobre los hechos de cada proveedor, en un proceso de decisión basado en similitud ponderada.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figura A14: Sistema basado en reglas GoAvatar

Este sistema ha sido desarrollado de forma personalizada dentro de la función *obtenerMejorProveedor* (backend), cuya lógica puede interpretarse como un motor de inferencia procedimental embebido.

* La **base de conocimientos** está formada por las reglas de evaluación codificadas en la estructura condicional switch-case, que determinan cómo debe compararse cada característica técnica entre lo deseado por el usuario y lo ofrecido por el proveedor. Estas reglas varían según el tipo de campo:
* Valores numéricos (se evalúa si el proveedor ofrece más o menos que el umbral requerido)
* Booleanos (se busca coincidencia exacta)
* Categóricos (se evalúa la igualdad de valor)
* La **base de hechos** se construye dinámicamente durante el proceso de evaluación, combinando los datos reales de cada proveedor con los criterios técnicos definidos por el usuario. Es decir, para cada proveedor analizado, el sistema genera un conjunto de hechos particulares sobre los que aplica las reglas previamente definidas.
* Aunque los **pesos** asociados a cada característica no forman parte estricta de la base de conocimientos, sí actúan como modificadores externos que ajustan la importancia relativa de cada regla. Estos pesos están almacenados en una colección JSON en la base de datos, lo que permite configurarlos sin modificar el código fuente.

Durante la ejecución, el sistema (**motor de inferencia**):

1. Recorre todos los proveedores disponibles.
2. Evalúa cada uno frente a los criterios seleccionados.
3. Aplica las reglas correspondientes por campo, calculando una similitud ponderada con base en los pesos configurables.
4. Ordena los proveedores por puntuación final y devuelve el más adecuado o los que igualan la máxima similitud.

Con esta funcionalidad, podemos seleccionar el proveedor que mejor nos conviene según las entradas indicadas por pantalla.

1. Análisis de Proveedores

Durante el desarrollo del proyecto, una parte importante ha sido comparar distintas APIs que permiten generar vídeos con avatares. La idea inicial era usar una metodología basada en ejemplos prácticos, lo que se conoce como razonamiento basado en casos (CBR). En este enfoque, cada proveedor se analiza como si fuera un caso distinto, evaluando siempre los mismos puntos para poder compararlos de forma clara.

A la hora de investigar, me encontré con varias limitaciones. Algunas plataformas publican la información de forma poco clara o incompleta. En ciertos casos promocionan funcionalidades que luego no están bien explicadas o que solo están disponibles en planes de pago sin mucha transparencia. Esto hace que resulte complicado comparar unos servicios con otros de forma objetiva.

Para poder organizar mejor la información, creé una plantilla [Tabla Proveedores Normalizada (2025\_V1).xlsx] con los aspectos más importantes que me interesaba evaluar: calidad del vídeo, velocidad, facilidad de integración, personalización, etc. Esa estructura me ayudó a recopilar los datos de forma ordenada y a tomar decisiones más fundamentadas sobre qué proveedor utilizar en cada situación.

* **Precios (paquete anual, precio mensual)**: análisis del coste base mensual y anual según el plan mínimo disponible para desarrolladores o empresas. En ocasiones vendrán como créditos que podrán contratarse de forma opcional para aumentar los minutos de vídeo.
* **Tiempo de vídeo consumible al mes:** minutos de vídeo generables con avatares dentro del plan contratado.
* **Tiempo de streaming consumible al mes:** minutos disponibles para vídeo en tiempo real (si el proveedor lo permite).
* **Tiene streaming:** indica si la API ofrece generación de vídeo sincronizado en tiempo real.
* **Streaming con SDK:** La API ofrece unas herramientas de integración en el proyecto para Streaming.
* **Tiene traducción de vídeo:** evalúa si es posible traducir y regenerar vídeos manteniendo la sincronización labial.
* **Resolución del vídeo basado en foto:** calidad de salida cuando el avatar es generado a partir de una imagen fija (por ejemplo, rostro frontal JPG/PNG).
* **Resolución del vídeo basado en foto:** calidad de salida cuando el avatar es generado a partir de una imagen fija (por ejemplo, rostro frontal JPG/PNG).
* **Resolución máxima de vídeo:** resolución de exportación permitida en la API (720p, 1080p, 4K, etc.).
* **Tiempo máximo de vídeo:** duración máxima por cada solicitud.
* **Sincronización labial:** grado de precisión con el que el avatar mueve los labios en relación al audio (Baja, Media o Alta).
* **Número de avatares disponibles:** cantidad de presentadores o modelos humanos listos para usar.
* **Avatares personales:** posibilidad de crear avatares personalizados a partir de imagen o grabación.
* **Voces (proveedores):** número de voces disponibles y variedad de proveedores de TTS soportados.
* **Expresiones del avatar:** soporte para emociones visuales como alegría, neutralidad, tristeza o entusiasmo.
* **Clonado de voz:** funcionalidad para generar una voz IA personalizada a partir de muestras reales.
* **Idiomas soportados:** número de idiomas y dialectos disponibles para la generación de voz.
* **Entrada alternativa:** posibilidad de subir un archivo de audio propio en lugar de usar texto.
* **Licencia y uso comercial:** condiciones legales para utilizar los vídeos generados con fines empresariales, de marketing o distribución.
* **Calidad de la API (documentación):** valoración de la documentación oficial, ejemplos, claridad y recursos para desarrolladores.
* **Velocidad de generación de vídeo:** tiempo estimado entre el envío de la solicitud y la recepción del vídeo final.

Para la identificación y selección de las APIs más relevantes de 2025 disponibles en el mercado actual, se ha realizado un proceso de exploración documental y análisis técnico a través de fuentes especializadas [6].

# **D-ID**

1. Visión general

La API de D-ID proporciona una interfaz RESTful orientada a la generación de contenido audiovisual mediante avatares virtuales animados a partir de una imagen estática (fotografía de rostro o torso) y una fuente de voz (texto o audio). Su arquitectura está diseñada para ofrecer servicios escalables de síntesis de vídeo con soporte multilingüe, personalización de voz y animación facial automática.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON)
* **Autenticación**: Basic Auth mediante API Key
* **API Referencia:** <https://docs.d-id.com/reference/get-started>
* **Base URL**: <https://api.d-id.com>
* **API versión**: 4.2.1
* **Formatos admitidos**: JSON en entrada y salida, imágenes JPEG/PNG, audio opcional

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /talks | Genera un video a partir de imagen + texto o audio |
| GET /talks/{id} | Recupera el estado o resultado de un video generado |
| POST /clips | Crea un vídeo de calidad a partir de avatares premium o personalizados. |
| POST /talks/streams | Genera animación en tiempo real (streaming interactivo) tomando como base una imagen. |
| POST /clips/streams | Genera animación en tiempo real (streaming interactivo) de alta calidad para asistentes virtuales (Premium). |
| POST /translations | Traduce un video a otros idiomas |
| GET /voices / GET /presets | Consulta voces TTS y modelos de avatares disponibles |

Tabla A4: Métodos D-ID

1. Ejemplo de llamada

**Texto

El contenido generado por IA puede ser incorrecto.**

Figura A15: Ejemplo de llamada D-ID

1. Características técnicas

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | Build | Launch | Scale |
| Precios (paquete anual, precio mensual) | 12.63 € | 30.71 € | 121.60€ |
| Tiempo de vídeo consumible al mes | 16 minutos | 45 minutos | 200 minutos |
| Tiempo de Streaming consumible al mes | 32 minutos | 90 minutos | 400 minutos |
| Tiene Streaming | Sí | Sí | Sí |
| Streaming con SDK | Sí | Sí | Sí |
| Tiene traducción de video | Sí | Sí | Sí |
| Resolución videos de avatar basado en foto (D-ID Technical Support, 2024) | Depende de la resolución de la imagen subida. | Depende de la resolución de la imagen subida. | Depende de la resolución de la imagen subida. |
| Resolución vídeos avatar | 1080p | 1080p | 1080p |
| Tiempo máximo de video | 5 minutos | 5 minutos | 5 minutos |
| Sincronización labial | Normal | Normal | Normal |
| Número avatares | 0 (sólo avatares de foto) | 111 | 111 |
| Avatares Personales | 1 | 3 | 5 |
| Voces (Proveedores) | 127 voces (+idiomas) | 127 voces (+idiomas) | 127 voces (+idiomas) |
| Expresiones del avatar | Neutral, Feliz, Serio, Sorpresa | Neutral, Feliz, Serio, Sorpresa | Neutral, Feliz, Serio, Sorpresa |
| Clonado de voz | No | 1 voz | 3 voces |
| Idiomas soportados | 119 lenguajes | 119 lenguajes | 119 lenguajes |
| Entrada alternativa | Audio | Audio | Audio |
| Licencia y uso comercial | No comercial | Comercial | Comercial |
| Calidad API (Documentación) | Normal | Normal | Normal |
| Velocidad de generación de vídeo | Baja | Media | Alta |

Tabla A5: Características D-ID

# **Synthesia**

1. Visión general

La API de Synthesia proporciona una interfaz RESTful orientada a la generación de videos mediante avatares virtuales con apariencia humana. Estos videos se crean a partir de texto proporcionado por el usuario, que es interpretado mediante modelos de texto a voz (TTS) y renderizado por presentadores digitales predefinidos. La API está diseñada para integrarse en flujos de trabajo automatizados, permitiendo crear, gestionar y descargar videos de forma programática.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON)
* **Autenticación**: Bearer Token
* **API Referencia:** <https://docs.synthesia.io/reference/introduction>
* **Base URL**: <https://api.synthesia.io>
* **API versión**: V2.
* **Formatos admitidos**: JSON en entrada y salida, vídeo MP4 en salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /v2/videos | Crea un video con un avatar digital a partir de texto y parámetros de configuración |
| GET /v2/videos/{id} | Consulta el estado del video generado (pendiente, procesando, completado, error) |
| POST /v2/webhooks | Lista de Webhooks activos. Pare recibir notificaciones cuando ocurran ciertos eventos en Synthesia. |

Tabla A6: Métodos Synthesia

1. Ejemplo de llamada

Texto

El contenido generado por IA puede ser incorrecto.

Figura A16: Ejemplo de llamada Synthesia

1. Características técnicas

|  |  |  |
| --- | --- | --- |
| Parámetro | Starter\* | Creator |
| Precios (paquete anual, precio mensual) | 16 € | 58 € |
| Tiempo de vídeo consumible al mes | 10 minutos | 30 minutos |
| Tiempo de Streaming consumible al mes | - | - |
| Tiene Streaming | No | No |
| Streaming con SDK | No | No |
| Tiene traducción de video | No | No |
| Resolución videos de avatar basado en foto | - | - |
| Resolución vídeos avatar | 720p | 1080p |
| Tiempo máximo de video | 5 minutos | 5 minutos |
| Sincronización labial | Alta | Alta |
| Número avatares | 125 | 180 |
| Avatares Personales | 3 | 5 |
| Voces (Proveedores) | 127 voces (+idiomas) | 127 voces (+idiomas) |
| Expresiones del avatar | No configurable | No configurable |
| Clonado de voz | No | No |
| Idiomas soportados | 140 | 140 |
| Entrada alternativa | Audio | Audio |
| Licencia y uso comercial | No comercial | No comercial |
| Calidad API (Documentación) | Baja | Baja |
| Velocidad de generación de vídeo | Media | Media |

Tabla A7: Características Synthesia

\*El modo Starter no incluye integración con API

# **HeyGen**

1. Visión general

La API de HeyGen proporciona una interfaz RESTful diseñada para la generación de contenido audiovisual mediante avatares digitales realistas. Permite generar vídeos a partir de texto en múltiples idiomas utilizando avatares predefinidos o personalizados, con capacidades avanzadas de integración en flujos de trabajo web y corporativos. Además, ofrece funcionalidad de streaming con respuesta en tiempo real mediante WebRTC y SDKs.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON) + WebSocket/WebRTC (streaming, SDK)
* **Autenticación**: Bearer Token
* **API Referencia:** <https://docs.heygen.com/docs/quick-start>
* **Base URL**: <https://api.heygen.com>
* **API versión**: V4.0.8
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /v1/video/generate | Crea un avatar digital a partir de texto / audio y parámetros configurables |
| GET /v1/video\_status.get | Consulta el estado de un vídeo |
| POST /v2/videos/{id}/generate | Crea un vídeo a partir de una plantilla |
| POST /v2/video\_translate | Traduce a otro idioma un vídeo. |
| POST /v1/streaming.new | Inicia una sesión Streaming |
| POST /v2/photo\_avatar/photo/generate | Crea un avatar a partir de una foto |

Tabla A8: Métodos HeyGen

1. Ejemplo de llamada

Texto

El contenido generado por IA puede ser incorrecto.

Figura A17: Ejemplo de llamada HeyGen

1. Características técnicas

|  |  |  |
| --- | --- | --- |
| Parámetro | Pro | Scale |
| Precios (paquete anual) | 87.65 € | 292.16 € |
| Tiempo de vídeo consumible al mes | 100 minutos | 660 minutos |
| Tiempo de Streaming consumible al mes | 500 minutos | 3300 minutos |
| Tiene Streaming | Sí | Sí |
| Streaming con SDK | Sí | Sí |
| Tiene traducción de video | No | Sí |
| Resolución videos de avatar basado en foto | Depende de la resolución de la imagen subida. | Depende de la resolución de la imagen subida. |
| Resolución vídeos avatar | 1080p (Streaming: 72op, 480p, 360p) | 4k (Streaming: 72op, 480p, 360p) |
| Tiempo máximo de video | 5 minutos | 30 minutos |
| Sincronización labial | Alta | Alta |
| Número avatares (consulta a través de API) | 1169 + 1699 (avatar foto) | 1169 + 1699 (avatar foto) |
| Avatares Personales (Group generation) | 3 | 10 |
| Voces (consulta a través de API) | 1855 | 1855 |
| Expresiones del avatar | Voz (Excited / Friendly / Serious / Smoothing / Broadcaster), Postura (Stable / Expressive), Expresión (default, happy) | Voz (Excited / Friendly / Serious / Smoothing / Broadcaster), Postura (Stable / Expressive), Expresión (default, happy) |
| Clonado de voz | No definido | No definido |
| Idiomas soportados | 175 | 175 |
| Entrada alternativa | Audio | Audio |
| Licencia y uso comercial | Comercial | Comercial |
| Calidad API (Documentación) | Alta | Alta |
| Velocidad de generación de vídeo | 25 segundos – 1 minuto.(*average time for a video to be created*, no date) | 25 segundos – 1 minuto.(*average time for a video to be created*, no date) |

Tabla A9: Características HeyGen

# **Elay**

1. Visión general

Elai es una plataforma de generación de vídeos mediante avatares digitales humanos que permite transformar texto en contenido audiovisual profesional. Su API está diseñada para automatizar la creación de vídeos, integrando funciones como selección de presentadores, idiomas, plantillas y configuración de música, todo a través de una interfaz RESTful. Está especialmente orientada a soluciones de aprendizaje corporativo para la generación de vídeos de avatares y la formación.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON)
* **Autenticación**: Bearer Token
* **API Referencia:** <https://elai.readme.io/reference/getting-started-with-your-api>
* **Base URL**: <https://api.elai.io>
* **API versión**: V1.0
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /api/v1/videos | Crea un vídeo |
| GET /api/v1/videos/{videoid} | Consulta el estado de un vídeo |
| POST /api/v1/streams | Crea un vídeo stream |
| POST /api/v1/videos/renderTemplate/{videoid} | Crea vídeos personalizados para clientes en base a un vídeo raíz. |
| POST /api/v1/videos/translate/{videoid} | Traduce un vídeo |
| POST /api/v1/avatars | Crea un avatar a partir de una foto |

Tabla A10: Métodos Elay

1. Ejemplo de llamada

Texto

El contenido generado por IA puede ser incorrecto.

Figura A18: Ejemplo de llamada Elay

1. Características técnicas

|  |  |  |
| --- | --- | --- |
| Parámetro | Basic | Advance |
| Precios (paquete mensual) | 20.22 € | 51.88 € |
| Tiempo de vídeo consumible al mes | 15 minutos (ampliable a 25, 40) | 100 minutos |
| Tiempo de Streaming consumible al mes | No se especifica. | No se especifica. |
| Tiene Streaming | No (Fase beta, sólo paquete Premium) | No (Fase beta, sólo paquete Premium) |
| Streaming con SDK | No | No |
| Tiene traducción de video | Sí | Sí |
| Resolución videos de avatar basado en foto | Depende de la resolución de la imagen subida. | Depende de la resolución de la imagen subida. |
| Resolución vídeos avatar | 1080p | 4k |
| Tiempo máximo de video | 5 minutos | 30 minutos |
| Sincronización labial | Normal | Normal |
| Número avatares | +80 | +80 |
| Avatares Personales | No (plugin aparte) | No (plugin aparte) |
| Voces | +450 | +450 |
| Expresiones del avatar | No | No |
| Clonado de voz | No definido | No definido |
| Idiomas soportados | +75 | +75 |
| Entrada alternativa | Audio | Audio |
| Licencia y uso comercial | Comercial | Comercial |
| Calidad API (Documentación) | Alta | Alta |
| Velocidad de generación de vídeo | 25 segundos – 1 minuto.(*average time for a video to be created*, no date) | 25 segundos – 1 minuto.(*average time for a video to be created*, no date) |

Tabla A11: Características Elay

# **VEED**

1. Visión general

VEED.io es una plataforma SaaS centrada en la edición de vídeo directamente desde el navegador, que incluye la funcionalidad de generación de avatares parlantes (Talking Avatars). A diferencia de otros proveedores como D-ID, HeyGen o Elai, **VEED no ofrece una API pública**, lo que impide su integración en pipelines automatizados o en aplicaciones personalizadas mediante programación. Su propuesta de valor reside en la accesibilidad inmediata, una interfaz visual que no requiere conocimientos técnicos y herramientas complementarias como subtitulación automática, edición por texto y transcripción de audio.

1. Arquitectura general

* **Tipo de API**: No tiene. Se utiliza el estudio a través de navegador.
* **Autenticación**: -
* **API Referencia:** -
* **Base URL**: -
* **API versión**: -
* **Formatos admitidos**: Peticiones por navegador.

# **Colossyan**

1. Visión general

Colossyan se distingue por su enfoque especializado en formación corporativa interactiva. A diferencia de otros proveedores como D-ID o HeyGen, Colossyan permite crear conversaciones entre avatares, añadir cuestionarios y ramificaciones en los vídeos, y generar contenido compatible con SCORM para plataformas de e-learning. También destaca por ofrecer clonación de voz gratuita en todos los planes, algo inusual frente a soluciones que lo reservan para niveles Enterprise. Sin embargo, no cuenta con funcionalidades de streaming en tiempo real, y su API solo está disponible en planes superiores, lo que limita su acceso a integradores individuales o pequeños desarrollos.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON)
* **Autenticación**: Bearer Token
* **API Referencia:** <https://docs.colossyan.com/getting-started/quickstart>
* **Base URL**: <https://app.colossyan.com>
* **API versión**: V1.0
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /api/v1/video-generation-jobs/template-jobs | Crea un vídeo a partir de una plantilla |
| POST /api/v1/video-generation-jobs | Crea un vídeo manualmente |
| GET /api/v1/video-generation-jobs/{videoId} | Consulta el estado de un vídeo |
| GET /api/v1/generated-videos/{videoid} | Obtiene la información de un vídeo |

Tabla A12: Métodos Colossyan

1. Ejemplo de llamada

Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

Figura A19: Ejemplo de llamada Colossyan

1. Características técnicas

|  |  |
| --- | --- |
| Parámetro | Business (único paquete con API) |
| Precios (paquete anual) | 61.15 € |
| Tiempo de vídeo consumible al mes | Ilimitado |
| Tiempo de Streaming consumible al mes | - |
| Tiene Streaming | No |
| Streaming con SDK | No |
| Tiene traducción de video | Sí |
| Resolución videos de avatar basado en foto | Depende de la resolución de la imagen subida. |
| Resolución vídeos avatar | 1080p |
| Tiempo máximo de video | 10 minutos |
| Sincronización labial | Normal |
| Número avatares | +170 |
| Avatares Personales | 10 |
| Voces | 70 |
| Expresiones del avatar | No |
| Clonado de voz | 2 |
| Idiomas soportados | 70 |
| Entrada alternativa | Audio |
| Licencia y uso comercial | Comercial |
| Calidad API (Documentación) | Bajo |
| Velocidad de generación de vídeo | No definido |

Tabla A13: Características Colossyan

# **Speechify**

1. Visión general

Speechify es una plataforma centrada en la conversión de texto a voz (TTS), que ofrece una API RESTful diseñada para integrarse en aplicaciones, servicios y flujos automatizados. Su enfoque principal es la generación de voz realista y personalizada, con soporte para más de 60 idiomas, control por SSML y clonación de voz. A diferencia de proveedores como Elai, D-ID o HeyGen, Speechify no genera vídeos ni avatares, sino que se especializa en producción de audio con alta calidad y personalización, siendo ideal para aplicaciones centradas en accesibilidad, lectura de contenido o asistentes virtuales.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON)
* **Autenticación**: Bearer Token
* **API Referencia:** <https://docs.sws.speechify.com/v1/api-reference/api-reference/introduction>
* **Base URL**: <https://api.sws.speechify.com>
* **API versión**: V1
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /v1/audio/speech | Crea un audio a partir de texto |
| POST /v1/voices | Clona una voz |

Tabla A14: Métodos Speechify

1. Ejemplo de llamada

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Figura A20: Ejemplo de llamada Speechify

1. Características técnicas

|  |  |
| --- | --- |
| Parámetro | Plan Genérico para API |
| Precios | No especificado |
| Tiempo de vídeo consumible al mes | - |
| Tiempo de Streaming consumible al mes | - |
| Tiene Streaming | No |
| Streaming con SDK |  |
| Tiene traducción de video | No |
| Resolución videos de avatar basado en foto | - |
| Resolución vídeos avatar | - |
| Tiempo máximo de video | - |
| Sincronización labial | - |
| Número avatares | - |
| Avatares Personales | - |
| Voces | +1000 |
| Expresiones del avatar | No |
| Clonado de voz | Sí |
| Idiomas soportados | +50 |
| Entrada alternativa | - |
| Licencia y uso comercial | Comercial |
| Calidad API (Documentación) | Bajo |
| Velocidad de generación de audio | 300ms latencia |

Tabla A15: Características Speechify

# **Tavus**

1. Visión general

Tavus es una plataforma especializada en la generación automatizada y personalizada de vídeos mediante inteligencia artificial. Su propuesta se centra en la creación de vídeos personalizados a escala, donde el avatar del usuario puede ser generado a partir de una grabación original y replicar múltiples mensajes de forma dinámica. Permite generar un avatar personalizado del usuario para crear miles de vídeos con nombres, productos o segmentos variables, lo que la hace ideal para marketing automatizado y ventas personalizadas B2B.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON) + WebSocket/WebRTC (streaming)
* **Autenticación**: Bearer Token
* **API Referencia:** <https://www.postman.com/tavus-api/tavus/collection/393k3hs/tavus-platform-api-developer-beta>
* **Base URL**: <https://tavusapi.com>
* **API versión**: V2
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /v2/avatars/train | Crea un avatar a mediante un vídeo |
| POST /v2/requests | Crea un vídeo con un Avatar |
| POST /v1/requests | Crea un vídeo con el input dinámico |

Tabla A16: Métodos Tavus

1. Ejemplo de llamada

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura A21: Ejemplo de llamada Tavus

1. Características técnicas

|  |  |  |
| --- | --- | --- |
| Parámetro | Starter | Growth |
| Precios | 51.75 € | 348.25 € |
| Tiempo de vídeo consumible al mes | 10 minutos | 100 minutos |
| Tiempo de Streaming consumible al mes | 100 minutos | 500 minutos |
| Tiene Streaming | Sí | Sí |
| Streaming con SDK | No | No |
| Tiene traducción de video | No | No |
| Resolución videos de avatar basado en foto | - | - |
| Resolución vídeos avatar | 1080p | 1080p |
| Tiempo máximo de video | 20 minutos | 60 minutos |
| Sincronización labial | Alta | Alta |
| Número avatares (consulta a través de API) | 25 | +100 |
| Avatares Personales (Group generation) | 3 | 7 |
| Voces (consulta a través de API) | 25 | +100 |
| Expresiones del avatar | No | No |
| Clonado de voz | Sí | Sí |
| Idiomas soportados | +30 | +30 |
| Entrada alternativa | No | No |
| Licencia y uso comercial | Comercial | Comercial |
| Calidad API (Documentación) | Normal | Normal |
| Velocidad de generación de vídeo | Rápida (1 segundo latencia) | Rápida (1 segundo latencia) |

Tabla A17: Características Tavus

# **Runway**

1. Visión general

Runway se diferencia de otras soluciones al centrarse en tareas avanzadas de edición y manipulación visual mediante inteligencia artificial, más allá de la simple generación de avatares parlantes. Su API permite crear vídeos completos a partir de texto (modelos Gen-3), eliminar fondos en tiempo real, aplicar estilos visuales y realizar inpainting inteligente en vídeo. A diferencia de APIs como D-ID o Tavus, que están orientadas a la personalización facial o avatar, Runway ofrece herramientas de postproducción creativa basadas en IA que permiten construir contenido desde cero o transformar material ya existente con un alto grado de control artístico y visual.

1. Arquitectura general

* **Tipo de API**: Integración
* **Autenticación**: Bearer Token
* **API Referencia:** <https://docs.dev.runwayml.com/>
* **Base URL**: <https://tavusapi.com>
* **API versión**: V2024-11-06
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

Esta API no está orientada a la generación de avatares reales, sino a la modificación de videos con avatares. No entra dentro del ámbito de las necesidades del proyecto.

1. Ejemplo de llamada

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura A22: Ejemplo de llamada Runway

1. Características técnicas

|  |  |
| --- | --- |
| Parámetro | API Usage |
| Precios | 0.22 € / 5 segundos |
| Tiempo de vídeo consumible al mes | Sin límite |
| Tiempo de Streaming consumible al mes | - |
| Tiene Streaming | No |
| Streaming con SDK | Sí |
| Tiene traducción de video | No |
| Resolución videos de avatar basado en foto | - |
| Resolución vídeos avatar | 720p |
| Tiempo máximo de video | 5 segundos |
| Sincronización labial | - |
| Número avatares (consulta a través de API) | No |
| Avatares Personales (Group generation) | No |
| Voces (consulta a través de API) | No |
| Expresiones del avatar | No |
| Clonado de voz | No |
| Idiomas soportados | No |
| Entrada alternativa | Imagen, vídeo |
| Licencia y uso comercial | Comercial |
| Calidad API (Documentación) | Normal |
| Velocidad de generación de vídeo | No especifica |

Tabla A18: Características Runway

# **A2E**

1. Visión general

A2E (Avatar to Emotion) es una plataforma que combina generación de vídeo con control emocional avanzado sobre avatares digitales. A diferencia de otras soluciones como D-ID o Tavus que se centran en la síntesis facial o voz, A2E destaca por permitir al usuario elegir un avatar y aplicar emociones realistas aumentando el tiempo de procesado de vídeo, obteniendo como salida un vídeo renderizado directamente desde su API. Esto lo hace especialmente útil para contenidos donde la expresividad facial del avatar es un factor determinante. Es una plataforma que está más orientada a avatares personales que los que pueda ofrecer el servicio.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON)
* **Autenticación**: Basic Auth mediante API Key
* **API Referencia:** <https://api.a2e.ai/>
* **Base URL**: <https://video.a2e.ai>
* **API versión**: V1
* **Formatos admitidos**: JSON en entrada y salida

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /api/v1/video/generate | Genera un video a partir de un avatar seleccionado. Si se utiliza el parámetro **isSkipRs** el procesado tarda más, pero la sincronización labial es mejor. |
| POST /api/v1/userVideoTwin/startTraining | Crea un avatar personal a partir de un vídeo o una imagen. |
| POST /api/v1/userVideoTwin/continueTraining | Continúa el entrenamiento de un avatar para la sincronización labial. |
| POST /api/v1/userImage2Video/start | A partir de una imagen crea un vídeo de 5 segundos con una calidad de 720p. Normalmente tarda 10 minutos en procesar. |
| POST /api/v1/streaming-avatar/set-qa-context | Crea un contexto para un avatar que tomará como base de conocimiento en un stream. |
| POST /api/v1/streaming-avatar/agora-token | Devuelve la información de un Avatar en Stream para conectarse a su canal |

Tabla A19: Métodos A2E

1. Ejemplo de llamada

Escala de tiempo

El contenido generado por IA puede ser incorrecto.

Figura A23: Ejemplo de llamada A2E

1. Características técnicas

|  |  |  |
| --- | --- | --- |
| Parámetro | Pay as You Go | Dedicated Line |
| Precios (precio mensual) | 8.77 € | 526.05 € |
| Tiempo de vídeo consumible al mes | 10 minutos | 694 minutos |
| Tiempo de Streaming consumible al mes | 40 minutos | 2777 minutos |
| Tiene Streaming | Sí | Sí |
| Streaming con SDK | No | No |
| Tiene traducción de video | No | No |
| Resolución videos de avatar basado en foto | Depende de la resolución de la imagen subida (recomendado 1080p). | Depende de la resolución de la imagen subida (recomendado 1080p). |
| Resolución vídeos avatar | 1080p | 1080p |
| Tiempo máximo de video | 5 minutos | 5 minutos |
| Sincronización labial | Alta | Alta |
| Número avatares | No especificado | No especificado |
| Avatares Personales | 1 | 3 |
| Voces (Proveedores) | No especificado | No especificado |
| Expresiones del avatar | Alta | Alta |
| Clonado de voz | Sí | Sí |
| Idiomas soportados | No especificado | No especificado |
| Entrada alternativa | Audio | Audio |
| Licencia y uso comercial | No comercial | Comercial |
| Calidad API (Documentación) | Buena | Buena |
| Velocidad de generación de vídeo | 1 minuto video = 5 minutos procesado | Media |

Tabla A20: Características A2E



Figura A15: Info precio A2E - https://www.a2e.ai/volume-price-of-a2e-api/

**Plan Pay as You Go**: 8.77€ = 600 coins = 600 /60 = 10 minutos de vídeo = 600 coins/15 -> 40 minutos de streaming.

**Dedicated Line**: 5026.05€ = 41666 coins /mes = 41666/60 -> 694 minutos apróx de vídeo = 41666/15 -> 2777 minutos apróx de streaming.

# **AI Studios**

1. Visión general

AI Studios destaca por ofrecer avatares de alta fidelidad que simulan presentadores humanos reales, con enfoque en formación empresarial, anuncios corporativos y producción de contenido institucional. Permite crear vídeos a partir de texto con sincronización labial precisa y control total sobre idioma, voz, formato de salida y fondo.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON), integrada en el proyecto.
* **Autenticación**: Bearer Token
* **API Referencia:**  <https://v2.aistudios.com>
* **Base URL**: <https://aistudios.com>
* **Planes:** <https://www.aistudios.com/pricing>
* **API versión**: V3
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /api/odin/v3/simple/video | Crea un vídeo con un avatar y unas ropas definidas |
| GET /api/odin/v3/editor/progress/{projectKey} | Consulta un vídeo |
| POST /api/odin/v3/editor/Project/{projectKey} | Edita un vídeo |

Tabla A21: Métodos AI Studios

Para Streaming, se usa un SDK integrado en la aplicación.

1. Ejemplo de llamada

Imagen que contiene Tabla

El contenido generado por IA puede ser incorrecto.

Figura A24: Ejemplo de llamada AI Studios

1. Características técnicas

|  |  |  |
| --- | --- | --- |
| Parámetro | Personal | Equipo |
| Precios (paquete anual, pago mensual) | 21.13 € | 158.49 € |
| Tiempo de vídeo consumible al mes | Ilimitado | Ilimitado |
| Tiempo de Streaming consumible al mes | No se especifica. | No se especifica. |
| Tiene Streaming | Sí (con SDK) | Sí (con SDK) |
| Streaming con SDK | Sí | Sí |
| Tiene traducción de video | No | Sí |
| Resolución videos de avatar basado en foto | Depende de la resolución de la imagen subida. | Depende de la resolución de la imagen subida. |
| Resolución vídeos avatar | 1080p | 1080p |
| Tiempo máximo de video | 10 minutos | 30 minutos |
| Sincronización labial | Normal | Normal |
| Número avatares | +80 | +80 |
| Avatares Personales | 3 | 5 |
| Voces | +100 | +100 |
| Expresiones del avatar | No | No |
| Clonado de voz | Sí | Sí |
| Idiomas soportados | +80 | +80 |
| Entrada alternativa | Audio | Audio |
| Licencia y uso comercial | Comercial | Comercial |
| Calidad API (Documentación) | Compleja (Baja) | Compleja (Baja) |
| Velocidad de generación de vídeo | No se especifica | No se especifica |

Tabla A22: Características AI Studios

# **Akool**

1. Visión general

Akool se distingue por ofrecer funcionalidades como reenactment facial (lip sync de vídeos originales), face swap, traducción labial sincronizada y edición por prompt sobre imágenes y vídeos reales. Su fortaleza radica en crear contenido hiperpersonalizado con rostros reales, incluyendo traducción automática manteniendo la sincronización labial para un doblaje localizado y contenidos personalizados en múltiples idiomas. Sin embargo, no dispone de un sistema completo de generación de vídeo desde cero mediante texto, y su documentación y acceso API están más limitados que en otras plataformas más desarrolladas.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON), integrada en el proyecto.
* **Autenticación**: Bearer Token
* **API Referencia:**  <https://docs.akool.com/authentication/usage>
* **Base URL**: <https://openai.akool.com>
* **Planes:** <https://akool.com/pricing>
* **API versión**: V3
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /api/open/v3/faceswap/highquality/specifyvideo | Crea un vídeo suplantando las caras por otra imagen |
| POST /api/open/v3/content/video/createbytranslate | Traduce un vídeo |
| POST /api/open/v3/content/video/createbytalkingphoto | Crea un vídeo de un avatar mediante una foto como parámetro y un audio |
| POST /api/open/v3/content/video/lipsync | Hace sincronizar los labios con el audio. Recomendable vídeos <25fps |
| POST /api/open/v3/talkingavatar/create | Crea un vídeo de un avatar mediante un audio |
| POST /api/open/v4/liveAvatar/session/create | Crea una sesión streaming |

Tabla A23: Métodos Akool

1. Ejemplo de llamada

Texto

El contenido generado por IA puede ser incorrecto.

Figura A25: Ejemplo de llamada Akool

1. Características técnicas

|  |  |  |  |
| --- | --- | --- | --- |
| Parámetro | PRO | PRO MAX | STUDIO |
| Precios (paquete anual, pago mensual) | 18.47 € | 69.48 € | 307.83 |
| Tiempo de vídeo consumible al mes | 10 mins | 40 mins | 200 mins |
| Tiempo de Streaming consumible al mes | 60 mins | 340 mins | 2000 mins |
| Tiene Streaming | Sí | Sí | Sí |
| Streaming con SDK | Sí | Sí | Sí |
| Tiene traducción de video | Sí | Sí | Sí |
| Resolución videos de avatar basado en foto | Depende de la resolución de la imagen subida. | Depende de la resolución de la imagen subida. | Depende de la resolución de la imagen subida. |
| Resolución vídeos avatar | 1080p / 4K | 1080p / 4K | 1080p / 4K |
| Tiempo máximo de video | No especificado | No especificado | No especificado |
| Sincronización labial | Alto\* | Alto\* | Alto\* |
| Número avatares | +60 (60 fotos) | +60 (100 fotos) | +60 (500 fotos) |
| Avatares Personales | 3 | 5 | 10 |
| Voces | No se especifica | No se especifica | No se especifica |
| Expresiones del avatar | No | No | No |
| Clonado de voz | Sí | Sí | Sí |
| Idiomas soportados | +155 | +155 | +155 |
| Entrada alternativa | No | No | No |
| Licencia y uso comercial | Comercial | Comercial | Comercial |
| Calidad API (Documentación) | Normal | Normal | Normal |
| Velocidad de generación de vídeo | No se especifica | No se especifica | No se especifica |

Tabla A24: Características Akool

\*Con un procesado especial, pagando (POST /api/open/v3/content/video/lipsync)

# **Vidnoz**

1. Visión general

Vidnoz Su principal fortaleza es ofrecer funciones profesionales a precios asequibles, con un plan gratuito limitado, aunque su API es básica y no está orientada a grandes integraciones técnicas.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON)
* **Autenticación**: Bearer Token
* **API Referencia:**  <https://es.vidnoz.com/docs/authentication.html>
* **Base URL**: <https://devapi.vidnoz.com>
* **Planes:** <https://www.vidnoz.com/api-pricing.html>
* **API versión**: V2
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /v2/task/ai-headshots | Crea una imagen para currículum a partir de una imagen genérica |
| POST /v2/task/generate-taking-head | Crea un vídeo de un avatar a partir de una imagen |

Tabla A25: Métodos Vidnoz

1. Ejemplo de llamada

Texto

El contenido generado por IA puede ser incorrecto.

Figura A25: Ejemplo de llamada Vidnoz

1. Características técnicas

|  |  |  |
| --- | --- | --- |
| Parámetro | PRO | ESCALA |
| Precios (se paga por mes) | 69.27€ (1000 créditos) | 262.16€ (6000 créditos) |
| Tiempo de vídeo consumible al mes | 250 mins | 1500 mins |
| Tiempo de Streaming consumible al mes | - | - |
| Tiene Streaming | No | No |
| Streaming con SDK | No | No |
| Tiene traducción de video | No | No |
| Resolución videos de avatar basado en foto | Depende de la resolución de la imagen subida. | Depende de la resolución de la imagen subida. |
| Resolución vídeos avatar | No especificado | No especificado |
| Tiempo máximo de video | No especificado | No especificado |
| Sincronización labial | Normal | Normal |
| Número avatares | +50 | +50 |
| Avatares Personales | No. Sólo fotos. | No. Sólo fotos |
| Voces | No se especifica | No se especifica |
| Expresiones del avatar | No | No |
| Clonado de voz | Sí | Sí |
| Idiomas soportados | +100 | +100 |
| Entrada alternativa | Audio | Audio |
| Licencia y uso comercial | Comercial | Comercial |
| Calidad API (Documentación) | Baja | Baja |
| Velocidad de generación de vídeo | No se especifica | No se especifica |

Tabla A26: Características Vidnoz

# **Pipio**

1. Visión general

Vidnoz Su principal fortaleza es ofrecer funciones profesionales a precios asequibles, con un plan gratuito limitado, aunque su API es básica y no está orientada a grandes integraciones técnicas.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON)
* **Autenticación**: Bearer Token
* **API Referencia:**  <https://docs.pipio.ai/>
* **Base URL**: <https://generate.pipio.ai>
* **Planes:**
  + <https://www.pipio.ai/pricing>
  + <https://www.g2.com/products/pipio/pricing>
* **API versión**: -
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /sigle-clip | Crea un vídeo con un avatar |
| POST /project/generate/dubbing | Traduce un vídeo |
| POST /Project/{PROJECT\_ID}/template | Modifica el template de un video |

Tabla A27: Métodos Pipio

1. Ejemplo de llamada

Texto

El contenido generado por IA puede ser incorrecto.

Figura A26: Ejemplo de llamada Pipio

1. Características técnicas

|  |  |  |
| --- | --- | --- |
| Parámetro | PRO | TEAM |
| Precios (se paga por año, precio por mes) | 14.06€ (10 créditos) | 505.11€ (500 créditos) |
| Tiempo de vídeo consumible al mes\* | 10 mins | 500 mins |
| Tiempo de Streaming consumible al mes | - | - |
| Tiene Streaming | No | No |
| Streaming con SDK | No | No |
| Tiene traducción de video | Sí | Sí |
| Resolución videos de avatar basado en foto | No | No |
| Resolución vídeos avatar | 1080p | 4k |
| Tiempo máximo de video | No especificado | No especificado |
| Sincronización labial | Normal | Normal |
| Número avatares | +60 | +60 |
| Avatares Personales | 2 | 2 |
| Voces | +140 | +140 |
| Expresiones del avatar | No | No |
| Clonado de voz | Sí | Sí |
| Idiomas soportados | +40 | +40 |
| Entrada alternativa | Audio | Audio |
| Licencia y uso comercial | Comercial | Comercial |
| Calidad API (Documentación) | Baja | Baja |
| Velocidad de generación de vídeo | Normal | Alta |

Tabla A28: Características Pipio

\* Tiempo de vídeo en créditos [7]

# **Creatify**

1. Visión general

Creatify genera vídeos de avatares, su fortaleza radica en automatizar la creación de anuncios, no en ofrecer control detallado sobre el avatar o permitir integraciones profundas en sistemas conversacionales dinámicos.

Sólo el plan PRO tiene acceso a la API.

1. Arquitectura general

* **Tipo de API**: RESTful (HTTP sobre JSON)
* **Autenticación**: Bearer Token
* **API Referencia:**  <https://creatify.mintlify.app/api-reference/introduction>
* **Base URL**: <https://api.creatify.ai>
* **Planes:** <https://creatify.ai/pricing>
* **API versión**: -
* **Formatos admitidos**: JSON en entrada y salida.

1. Funcionalidades principales

|  |  |
| --- | --- |
| Endpoint | Descripción técnica |
| POST /api/link\_to\_videos | Crea un vídeo para un anuncio mediante un enlace a una url |
| POST /api/lipsyncs\_v2 | Convierte texto o audio reproducido por una persona hablando |
| POST /api/api\_scripts | Genera un vídeo con IA con una combinación entre la URL introducida y un texto descriptivo |
| POST /api/ai\_shorts | Crea un vídeo viral a través de un texto |
| POST /api/text\_to\_speech | Genera un vídeo con un avatar |
| POST /api/personas\_v2 | Crea un avatar. Tarda entre 2 y 5 días |

Tabla A29: Métodos Creatify

1. Ejemplo de llamada

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura A27: Ejemplo de llamada Creatify

1. Características técnicas

|  |  |
| --- | --- |
| Parámetro | PRO |
| Precios (se paga por año, precio por mes) | 42.97€ (2400 créditos) |
| Tiempo de vídeo consumible al mes | 20 mins |
| Tiempo de Streaming consumible al mes | - |
| Tiene Streaming | No |
| Streaming con SDK | No |
| Tiene traducción de video | No |
| Resolución videos de avatar basado en foto | No |
| Resolución vídeos avatar | 4k |
| Tiempo máximo de video | 10 mins |
| Sincronización labial | Normal |
| Número avatares | +700 |
| Avatares Personales | 3 |
| Voces | +170 |
| Expresiones del avatar | No |
| Clonado de voz | NO |
| Idiomas soportados | +29 |
| Entrada alternativa | Audio |
| Licencia y uso comercial | Comercial |
| Calidad API (Documentación) | Alta |
| Velocidad de generación de vídeo | Alta |

Tabla A30: Características Creatify

# **Normalización de los datos**

1. Normalización de columnas para el sistema experto basado en reglas

Antes de poder trabajar con los datos obtenidos de los proveedores, se tienen que normalizar ya que muchos de ellos ofrecen la información según sus criterios o escalas. Para ellos, se decide normalizar los resultados de los proveedores de la siguiente manera para después poder trabajar con una lógica para las reglas del sistema experto.

**1. Columna: “Precio (€/mes)”**

* **Formato**: Decimal.

**2. Columna: “Vídeo mensual (min)”**

* **Formato**: Decimal
* **Criterio**:
  + Valores como “Ilimitado” se transforman en un valor alto estandarizado (por ejemplo, 10000).
  + Vacíos o no definidos se convierten en 0.

**3. Columna: “Streaming mensual (min)”**

* **Formato**: Decimal
* **Criterio**:
  + Valores como “Ilimitado” se transforman en un valor alto estandarizado (por ejemplo, 10000).
  + Vacíos o no definidos se convierten en 0.

**4. Columna: “Tiene Streaming”**

* **Formato**: 0 (No), 1 (Sí)

**5. Columna: “Streaming con SDK”**

* **Formato**: 0 (No), 1 (Sí)

**6. Columna: “Traducción”**

* **Formato**: 0 (No), 1 (Sí)

**7. Columna: “Resolución”**

* **Formato**:
  + 0 = 720p
  + 1 = 1080p
  + 2 = 4K
* **Criterio**: No especificado = “0”

**8. Columna: “Duración máxima vídeo”**

* **Formato**: Decimal
* **Criterio**:
  + Valores como “5 segundos” se transforman a 0.08 min
  + “Ilimitado” → 10000
  + Sin definir → 5 minutos

**9. Columna: “Sincronización Labial”**

* **Formato**:
  + 0 = BAJA
  + 1 = NORMAL
  + 2 = ALTA

**10. Columna: "Clonado de Voz"**

* **Formato**: 0 (No), 1 (Sí)
* **Criterio**: Se asigna 1 si el proveedor ofrece clonación de una o más voces. Se asigna 0 si no ofrece o si está indicado como "No definido".

**11. Columna: "Expresiones del avatar"**

* **Formato**: 0 (No), 1 (Sí)
* **Criterio**: Se asigna "1" si existe configuración de expresiones, aunque no se detalle cuáles. "0" si el campo es "No", "No configurable" o está vacío.

**12. Columnas numéricas con símbolo "+" (ej. "+170")**

* **Formato**: Decimal
* **Criterio**: Se elimina el signo y se toma el valor numérico conocido. Si el campo es "No definido" o vacío, se asigna 0.
* **Columnas afectadas**:
  + "Número avatares"
  + "Avatares Personales"
  + "Voces"
  + "Idiomas soportados"

**13. Columna: "Entrada alternativa"**

* **Formato**:
  + 0 = Solo texto
  + 1 = Audio
  + 2 = Audio + imagen/vídeo
* **Criterio**: Se evalúa el contenido textual y se asigna el valor según la combinación de entradas.

**14. Columna: "Licencia y uso comercial"**

* **Formato**: 0 (No comercial), 1 (Comercial)
* **Criterio**: Se asigna 1 si el proveedor permite uso comercial. Caso contrario, se asigna 0.

**15. Columna: "Calidad API (Documentación)"**

* **Formato**:
  + 0 = BAJA
  + 1 = NORMAL
  + 2 = ALTA

**16. Columna: "Velocidad de generación de vídeo"**

* **Formato**:
  + 0 = BAJA
  + 1 = NORMAL
  + 2 = ALTA
* **Criterios específicos**:
  + "25s-1min", "Normal", "Media", "1 minuto", "No especifica" → 1
  + "300ms latencia", "1 segundo latencia", "Alta" → 2
  + "5 min por 1 min procesado", "Baja" → 0



Tabla A31: TABLA GENERAL PROVEEDORES (SIN NORMALIZAR)



Tabla A32: TABLA GENERAL PROVEEDORES (NORMALIZADA)

1. Sistema de reglas. Cálculo de la Similitud Total en el Sistema CBR

Este apartado define las reglas del sistema experto basado en reglas.

Se define el concepto regla “Similitud” a la afinidad de una característica técnica con la decisión tomada por el usuario. Por ejemplo, si un usuario selecciona un precio de 30€ y el proveedor analizado tiene un precio de 20€, esta característica tendrá una similitud baja (30€ > 20€ → Similitud baja).

La similitud debe estar normalizada en un rango entre [0, 1], donde 0 es una similitud deficiente y 1 es una similitud idónea.

A continuación se muestra las reglas que se utilizarán en este sistema experto de reglas para cada una de las especificaciones técnicas.

**1. Columna: “Precio (€/mes)”**

* *Si Proveedor* ***≤*** *Elección → 1*
* *Si Proveedor* ***>*** *Elección → MAX (0, 1 – ((Proveedor – Elección)/Proveedor))*

**2. Columna: “Vídeo mensual (min)”**

* *Si Proveedor ≥ Elección → 1*
* *Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))*

**3. Columna: “Streaming mensual (min)”**

* *Si Proveedor ≥ Elección → 1*
* *Si Proveedor <**Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))*

**4. Columna: “Tiene Streaming”**

* *Si Elección == Proveedor → 1*
* *Si Elección <****>*** *Proveedor → 0*

**5. Columna: “Streaming con SDK”**

* *Si Elección == Proveedor → 1*
* *Si Elección <****>*** *Proveedor → 0*

**6. Columna: “Traducción”**

* *Si Elección == Proveedor → 1*
* *Si Elección <****>*** *Proveedor → 0*

**7. Columna: “Resolución”**

* *Si (Elección > Proveedor) → 0*
* *Si (Elección <= Proveedor) → 1*

*En este caso, por ejemplo, si Selección = 1080p:*

*Proveedores que ofrezcan 720p → 0*

*Proveedores que ofrezcan 1080p → 1*

*Proveedores que ofrezcan 4K → 1*

**8. Columna: “Duración máxima vídeo”**

* *Si Proveedor ≥**Elección → 1*
* *Si Proveedor <**Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))*

**9. Columna: “Sincronización Labial”**

* *Si (Elección > Proveedor) → 0*
* *Si (Elección <= Proveedor) → 1*

*En este caso, por ejemplo, si Selección = NORMAL:*

*Proveedores que ofrezcan BAJA → 0*

*Proveedores que ofrezcan NORMAL → 1*

*Proveedores que ofrezcan ALTA → 1*

**10. Columna: "Clonado de Voz"**

* *Si Elección == Proveedor → 1*
* *Si Elección <> Proveedor → 0*

**11. Columna: "Expresiones del avatar"**

* *Si Elección == Proveedor → 1*
* *Si Elección <> Proveedor → 0*

**12. Columnas numéricas con símbolo "+" (ej. "+170")**

* *Si Proveedor ≥ Elección → 1*
* *Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))*
* **Columnas afectadas**:
  + "Número avatares"
  + "Avatares Personales"
  + "Voces"
  + "Idiomas soportados"

**13. Columna: "Entrada alternativa"**

* *Si Elección == Proveedor → 1*
* *Si Elección <> Proveedor → 0*

**14. Columna: "Licencia y uso comercial"**

* *Si Elección == Proveedor → 1*
* *Si Elección <> Proveedor → 0*

**15. Columna: "Calidad API (Documentación)"**

* *Si (Elección > Proveedor) → 0*
* *Si (Elección <= Proveedor) → 1*

*En este caso, por ejemplo, si Selección = NORMAL:*

*Proveedores que ofrezcan BAJA → 0*

*Proveedores que ofrezcan NORMAL → 1*

*Proveedores que ofrezcan ALTA → 1*

**16. Columna: "Velocidad de generación de vídeo"**

* *Si (Elección > Proveedor) → 0*
* *Si (Elección <= Proveedor) → 1*

*En este caso, por ejemplo, si Selección = NORMAL:*

*Proveedores que ofrezcan BAJA → 0*

*Proveedores que ofrezcan NORMAL → 1*

*Proveedores que ofrezcan ALTA → 1*

Cada característica técnica irá asociada a un peso según su importancia en el sistema. Por ejemplo, el precio del proveedor puede ser determinante en muchos casos y tendrá un peso mayor que el tiempo máximo de vídeo, por ejemplo. Estas características técnicas deben ir ajustadas con su peso.

Los pesos tomarán estos valores:

**IMPORTANTÍSIMO**: 1

**ALTO**: 0.8

**MEDIO**: 0.5

**BAJO**: 0.3

**MUY BAJO**: 0.2

**DESPRECIABLE**: 0.1

De modo que, finalmente, obtendremos la fórmula general de la Similitud Total, es decir, la similitud obtenida multiplicada por el peso de la característica analizada.

**Fórmula General:**

***SimilitudTotal = ∑ (Similitud\_i x Peso\_i)***

Donde:

* *Similitud\_i* es la similitud individual del atributo *i*, normalizada entre 0 y 1.
* *Peso\_i* es el peso asignado a ese atributo según su importancia.

El **máximo de similitud** que se podrá obtener en cada prueba dependerá de las características técnicas elegidas y su peso. Si sólo se elige la característica técnica “Precio” y su peso asociado es 1, su máxima similitud será 1\*1 = 1, nuca se podrá superar este valor.

**Ponderación por Característica Técnica**

|  |  |  |
| --- | --- | --- |
| Atributo | Peso | Fórmula de Similitud Individual |
| Licencia comercial | IMPORTANTÍSIMO | * *Si Elección == Proveedor → 1* * *Si Elección <> Proveedor → 0* |
| Precio | IMPORTANTÍSIMO | * *Si Proveedor ≤ Elección → 1* * *Si Proveedor > Elección → MAX (0, 1 – ((Proveedor – Elección)/Proveedor))* |
| Tiene streaming | MUY ALTO | * *Si Elección == Proveedor → 1* * *Si Elección <> Proveedor → 0* |
| Calidad API | MUY ALTO | 1 - abs(valor\_usuario - v\_proveedor)/2 |
| Clonado de voz | MUY ALTO | * *Si Elección == Proveedor → 1* * *Si Elección <> Proveedor → 0* |
| Resolución del video | MEDIO | * *Si (Elección > Proveedor) → 0* * *Si (Elección <= Proveedor) → 1* |
| Minutos de video/mes | MEDIO | * *Si Proveedor ≥ Elección → 1* * *Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))* |
| Minutos de streaming/mes | MEDIO | * *Si Proveedor ≥ Elección → 1* * *Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))* |
| Traducción automática | MEDIO | * *Si Elección == Proveedor → 1* * *Si Elección <> Proveedor → 0* |
| Sincronización labial | MEDIO | * *Si (Elección > Proveedor) → 0* * *Si (Elección <= Proveedor) → 1*   *En este caso, por ejemplo, si Selección = NORMAL:*  *Proveedores que ofrezcan BAJA → 0*  *Proveedores que ofrezcan NORMAL → 1*  *Proveedores que ofrezcan ALTA → 1* |
| Nº Avatares | MEDIO | * *Si Proveedor ≥ Elección → 1* * *Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))* |
| Expresiones del avatar | BAJO | * *Si Elección == Proveedor → 1* * *Si Elección <> Proveedor → 0* |
| Avatares personales | MUY BAJO | * *Si Proveedor ≥ Elección → 1* * *Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))* |
| Número de voces | DESPRECIABLE | * *Si Proveedor ≥ Elección → 1* * *Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))* |
| Idiomas soportados | DESPRECIABLE | * *Si Proveedor ≥ Elección → 1* * *Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))* |
| Duración máxima de video | DESPRECIABLE | * *Si Proveedor ≥**Elección → 1* * *Si Proveedor <**Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))* |
| Entrada alternativa | DESPRECIABLE | * *Si Elección == Proveedor → 1* * *Si Elección <> Proveedor → 0* |
| SDK de streaming | DESPRECIABLE | * *Si Elección == Proveedor → 1* * *Si Elección <> Proveedor → 0* |
| Velocidad de generación | DESPRECIABLE | * *Si (Elección > Proveedor) → 0* * *Si (Elección <= Proveedor) → 1*   *En este caso, por ejemplo, si Selección = NORMAL:*  *Proveedores que ofrezcan BAJA → 0*  *Proveedores que ofrezcan NORMAL → 1*  *Proveedores que ofrezcan ALTA → 1* |

Tabla A33: Reglas y pesos

1. Caso de ejemplo

Para poder probar las distintas casuísticas, se recomienda utilizar el documento “**Tabla Proveedores Normalizada (2025\_V1).xls**” para visualizar mejor los resultados y probar nuevas configuraciones sobre los distintos proveedores.

**Analizamos el Proveedor D-ID, Plan LAUNCH**

Imaginemos que un usuario quiere obtener el mejor proveedor según estas características:

1. Licencia Comercial → Sí.
2. Precio ≤ 50€ / mes
3. Tiene Streaming → Sí
4. Tiempo de generación de vídeo al mes ≥ 50 minutos
5. Tiempo de generación de vídeo streaming al mes ≥ 100 minutos
6. Clonado de voz → Sí
7. Números de avatares ofrecidos ≥ 100

Vamos a aplicar la ponderación según los pesos asignados en la tabla <PONER TABLA> para el caso de **D-ID** y su Plan LAUNCH.

1. **Licencia comercial** (Elección → Sí || Proveedor → Sí):
   1. Peso = ***IMPORTANTÍSIMO (1)***
   2. Fórmula de similitud:

* *Si Elección == Proveedor → 1* (CUMPLE CRITERIO)
* *Si Elección <> Proveedor → 0*
  1. **Total =** 1 \* 1 **= 1**

1. **Precio** (Elección → 50€ || Proveedor → 30.71€):
   1. Peso = ***IMPORTANTÍSIMO (1)***
   2. Fórmula de similitud:

* *Si Proveedor ≤ Elección → 1* (CUMPLE CRITERIO)
* *Si Proveedor > Elección → MAX (0, 1 – ((Proveedor – Elección)/Proveedor))*
  1. **Total =** 1 \* 1 **= 1**

1. **Tiene Streaming** (Elección → Sí || Proveedor → Sí):
   1. Peso = ***ALTO (0.8)***
   2. Fórmula de similitud:
      * + *Si Elección == Proveedor → 1* (CUMPLE CRITERIO)
        + *Si Elección <> Proveedor → 0*
   3. **Total** = 0.8 \* 1 = **0.8**
2. **Tiempo de generación de vídeo al mes** (Elección → 50 minutos || Proveedor → 45 minutos)
   1. Peso = ***MEDIO (0.5)***
   2. Fórmula de similitud:

* Si Proveedor ≥ Elección → 1
* Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección)) (CUMPLE CRITERIO)
  1. **Total** = 0.5 \* MAX(0, 1-((50 – 45) / 50)) = 0.5 \* MAX(0, 0.9) = 0.5 \* 0.9 = **0.45**

1. **Tiempo de generación de vídeo streaming al mes** (Elección → 100 minutos || Proveedor → 90 minutos)
2. Peso = ***MEDIO (0.5)***
3. Fórmula de similitud:

* Si Proveedor ≥ Elección → 1
* Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección)) (CUMPLE CRITERIO)

1. **Total** = 0.5 \* MAX(0, 1 – ((100-90) / 100)) = 0.05 \* MAX(0, 0.9) = 0.5 \* 0.9 = **0.45**
2. **Clonado de voz** (Elección → SÍ || Proveedor → SÍ)
3. Peso = ***ALTO (0.8)***
4. Fórmula de similitud:

* Si Elección == Proveedor → 1 (CUMPLE CRITERIO)
* Si Elección <> Proveedor → 0

1. **Total** = 0.8 \* 1 = **0.8**
2. **Nº Avatares Ofrecidos** (Elección → 100 || Proveedor → 111)
3. Peso = ***MEDIO (0.5)***
4. Fórmula de similitud:

* Si Proveedor ≥ Elección → 1 (CUMPLE CRITERIO)
* Si Proveedor < Elección → MAX (0, 1 – ((Elección – Proveedor)/Elección))

1. **Total** = 0.5 \* 1 = **0.5**

***SimilitudTotal*** = 1 + 1 + 0.8 + 0.45 + 0.45 + 0.8 + 0.5 = **5**

A continuación se muestra la tabla obtenida en “Tabla Proveedores Normalizada (2025\_V1).xlsx” con los inputs de este ejercicio:

Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Tabla A34: Tabla Proveedores Normalizadas (2025\_V1)

Se puede observar, que el proveedor D-ID con el plan Launch es el que mejor nota a obtenido en Similitud Total (5, rotulado en azul) sobre los demás.



Tabla A35: Tabla Proveedores Normalizadas (2025\_V1) – Resultados Similitud

1. Proveedor seleccionado para el proyecto

El proveedor seleccionado para el proyecto mediante el sistema experto basado en reglas ha sido D-ID [8], en su modalidad Launch, al obtener una similitud total de 0.59 respecto a los requisitos establecidos en el caso de entrada.

Las entradas al sistema fueron estas:

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla A36: Tabla Proveedores Normalizadas (2025\_V1) – Entradas al sistema

# **Proveedores fuera de estudio**

Este apartado identifica a los proveedores que no entran dentro del estudio del arte; los motivos son dos: no tienen API o es necesaria invitación para poder utilizar la API.

* Fliki.ai: <https://fliki.ai/enterprise/text-to-speech-api>
  + Requiere autenticación privada para usar la API.
* Zebracat.ai: <https://www.zebracat.ai/>
  + No tiene API
* InVideo.io: <https://invideo.io/ai/ai-image-generator/>
  + No tiene API.
* Picsart.com: <https://picsart.com/blog/api-program/>
  + Requiere invitación para usar la API
* SoulsMachines: <https://soulsmachines.com>
  + No tiene API.

# **Especificación de requisitos**

* 1. Introducción

Este apartado recoge los requisitos funcionales y particulares que definen el comportamiento esperado del sistema a desarrollar. Estos requisitos sirven como base para el diseño, implementación y validación de la aplicación.

* 1. Objetivos generales

El objetivo principal del proyecto es desarrollar una aplicación que permita la creación de vídeos utilizando avatares realistas basados en inteligencia artificial, excluyendo avatares animados o modelos 3D.

La aplicación permitirá difundir estos vídeos en redes sociales o ser descargados a disco duro.

El usuario externo a la aplicación podrá interactuar con un avatar en tiempo real que le asesorará sobre un tema específico.

La aplicación deberá permitir la creación de avatares personales con la imagen y voz que le indiquemos.

Existirá un usuario “experto” que podrá añadir o eliminar proveedores registrados en el sistema, del mismo modo que podrá introducir los pesos (la importancia) de cada característica técnica (como duración de vídeo o precio) y podrá analizar qué proveedor es el óptimo según las entradas indicadas al sistema.

* 1. Catálogo de requisitos

**REQUISITOS FUNCIONALES**

* **R.F.-1 Inicio de sesión (Login):** Tanto el usuario “admin” como el “experto” deben ingresar a la aplicación.
* **R**.**F.-2 Creación de vídeos:** El usuario “admin” tiene acceso a esta pantalla. Aquí se podrá crear los vídeos eligiendo un avatar, una voz, el texto o audio que el avatar ejecutará y un fondo. Una vez creado, el vídeo deberá aparecer en la pantalla de Gestión de Vídeos.
* **R.F.-3 Gestión de Vídeos**: El usuario “admin” podrá gestionar los vídeos creados con anterioridad en la pantalla “Gestión de vídeos”, permitiendo ver si el vídeo está listo, descargarlo o publicado en la red social X (Twitter). También puede eliminar el vídeo del sistema.
* **R.F.-4 Configurar el comportamiento del Avatar en Tiempo Real:** El usuario “admin” puede configurar el aspecto, la voz y el comportamiento que tendrá el avatar en tiempo real.
* **R.F.-5 Conversación con Avatar en Tiempo Real**: El usuario externo puede mantener sesiones de conversación en tiempo real con el avatar, que responderá según sus parámetros configurados.
* **R.F.-6 Buscar Proveedor Óptimo:** El usuario “experto” usa el sistema experto basado en reglas para obtener el mejor proveedor según los parámetros introducidos en esta pantalla.
* **R.F.-7 Edición de Proveedores:** El usuario “experto” puede añadir, eliminar o editar proveedores en esta pantalla.
* **R.F.8 Configuración de pesos:** El usuario “experto” puede configurar los pesos de las características técnicas en esta pantalla, que tendrá impacto a la hora de obtener el proveedor óptimo.

**REQUISITOS PARTICULARES**

* **R.P.-1**: Para una mayor capacidad de control, se priorizará el uso del SDK del agente de streaming (avatar en tiempo real) en lugar de realizar llamadas HTTP (POST).
* **R.P.-2:** Utilizar únicamente la API del proveedor D-ID para la generación de contenido.
* **R.P.-3**: La red social utilizada para publicar contenido de este proyecto es X (Twitter).
  1. ¿Se han logrado los objetivos al acabar el proyecto?

Sí, salvo uno de los apartados, que no se ha podido por errores técnicos de la API del proveedor.

|  |  |  |  |
| --- | --- | --- | --- |
| Objetivo | Descripción técnica | ¿Logrado? | Comentario |
| Creación de vídeos | La aplicación permite generar vídeos hechos por avatares. | ✔️ |  |
| Publicación en redes sociales | La aplicación permite gestionar los vídeos y publicarlos en redes sociales (X – Twitter) | ✔️ |  |
| Creación de avatares personales | Creación de avatar personal con la imagen y voz que configuremos. | ❌ | La API devuelve “unauthorized” al intentar crear el avatar |
| Avatar en tiempo real | El usuario externo a la aplicación podrá interactuar con un avatar virtual en tiempo real para informarse de un tema específico. | ✔️ |  |
| Elección de Proveedor | La aplicación tiene el usuario “experto” que puede acceder a las pantallas de elección de proveedores donde puede añadir, eliminar y configurar los pesos de las características técnicas de los proveedores. Es en este apartado donde se desarrolla el sistema experto basado en reglas para la elección del proveedor óptimo. | ✔️ |  |

Tabla B1: Tabla de objetivos

* 1. Casos de Uso

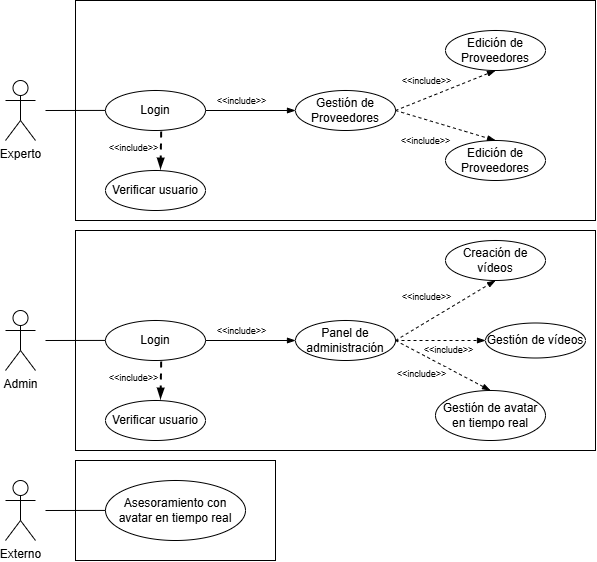


Figura B1: Casos de Uso

Actores

En la aplicación existen 3 actores:

* **Experto**: Es el usuario que se encarga de gestionar los proveedores y los pesos de las características técnicas. Además puede obtener el proveedor óptimo según los criterios de selección.
* **Admin**: Es el usuario que se encarga de la gestión de vídeos y configuración del avatar en tiempo real. Accede al panel de configuración.
* **Externo**: Este usuario no entra en el sistema, sólo puede asesorarse con el avatar virtual en tiempo real.

|  |  |
| --- | --- |
| CU-01 | Inicio de sesión |
| Requisitos relacionados | R.F.1 |
| Requisitos de los que depende | - |
| Actores | Experto, Admin |
| Descripción | El usuario se loguea |
| Precondiciones | El usuario no debe estar logueado. Debe estar en base de datos registrado. |
| Acciones | 1. Mostrar formulario de inicio de sesión con campos: email y contraseña  2. Esperar a que el usuario pulse el botón “Entrar”  3. Si el email y la contraseña están en la base de datos:  → Iniciar sesión  → Redirigir a la pantalla principal (Panel de administración o Gestión de proveedores)  4. Si las credenciales son incorrectas:  → Mostrar mensaje: “Usuario o contraseña incorrectos” |
| Postcondiciones | Si las credenciales son correctas, el administrador accede a la aplicación |
| Excepciones | Si las credenciales no son correctas, se le informa por pantalla “Usuario o contraseña incorrectos” |
| Importancia | Alta |

Tabla B2: CU-01 Inicio de sesión

|  |  |
| --- | --- |
| CU-02 | Creación de vídeos |
| Requisitos relacionados | R.F.2 |
| Requisitos de los que depende | R.F.1 |
| Actores | Admin |
| Descripción | El usuario admin accede a la funcionalidad de creación de vídeos, selecciona las opciones necesarias (avatar, texto, voz, fondo, etc.) y lanza la generación automática del vídeo. |
| Precondiciones | El usuario debe haber iniciado sesión como admin y acceder a esta pantalla (Creación de Vídeo) desde Panel de Administración. |
| Acciones | 1. Mostrar formulario de creación de vídeo con campos:  - Nombre  - Descripción  - Avatar  - Voz o audio  - Texto  - Fondo  2. Esperar a que el usuario pulse el botón “Generar vídeo”  3. Enviar la información al backend mediante petición HTTPS  4. Si el envío es correcto:  → Mostrar mensaje verde indicando que el vídeo está en proceso  5. Si ocurre un error:  → Mostrar mensaje rojo con el detalle del fallo |
| Postcondiciones | Si el vídeo se ha enviado al proveedor, saldrá un mensaje verde indicando que vaya a la pantalla de Gestión de Vídeos para ver si el vídeo se ha terminado de procesar en el Proveedor. |
| Excepciones | Si el vídeo no se ha conseguido procesar, aparecerá un mensaje en rojo indicando el error. |
| Importancia | Alta |

Tabla B3: CU-02 Creación de vídeos

|  |  |
| --- | --- |
| CU-03 | Gestión de vídeos |
| Requisitos relacionados | R.F.3 |
| Requisitos de los que depende | R.F.1 |
| Actores | Admin |
| Descripción | El usuario admin accede a la pantalla de gestión de vídeos, donde puede consultar el estado de los vídeos generados previamente, así como realizar acciones como visualizar, descargar, publicar en redes sociales o eliminar. |
| Precondiciones | El usuario debe haber iniciado sesión como admin y acceder a esta pantalla (Gestión de vídeos) desde Panel de Administración. |
| Acciones | 1. Mostrar al usuario la lista de vídeos disponibles  2. Esperar a que el usuario seleccione un vídeo  3. Mostrar las acciones disponibles:  - Visualizar vídeo  - Descargar vídeo  - Publicar en red social (X)  - Eliminar vídeo  4. Ejecutar la acción seleccionada  5. Mostrar mensaje confirmando la acción realizada o indicando si ha habido algún error |
| Postcondiciones | El usuario ha gestionado uno o varios vídeos: puede haberlos visualizado, descargado, publicado o eliminado. |
| Excepciones | Si en algún momento se produce un error, aparecerá un mensaje en rojo indicándolo. |
| Importancia | Alta |

Tabla B4: CU-03 Gestión de vídeos

|  |  |
| --- | --- |
| CU-04 | Configurar el comportamiento del Avatar en Tiempo Real |
| Requisitos relacionados | R.F.4 |
| Requisitos de los que depende | R.F.1 |
| Actores | Admin |
| Descripción | El usuario admin accede a la pantalla de configuración de avatar en tiempo real, donde puede seleccionar la apariencia del avatar, la voz y el comportamiento del agente en sesiones interactivas. |
| Precondiciones | El usuario debe haber iniciado sesión como admin y acceder a esta pantalla (Gestión de Avatar en Tiempo Real) desde Panel de Administración. |
| Acciones | 1. Mostrar el formulario de configuración del avatar en tiempo real  2. Esperar a que el usuario seleccione:  - Voz del agente  - Apariencia del avatar (imagen o vídeo)  - Comportamiento del agente  3. Esperar a que el usuario pulse el botón “Guardar cambios”  4. Enviar los datos al backend mediante petición HTTPS  5. Si se guardan correctamente:  → Mostrar mensaje verde de confirmación  6. Si ocurre un error:  → Mostrar mensaje rojo informando del problema |
| Postcondiciones | El sistema guarda los cambios. Indicándolo con un mensaje con fondo verde. |
| Excepciones | Si hay algún error, éste se mostrará en un mensaje de fondo rojo. |
| Importancia | Alta |

Tabla B5: CU-04 Configurar el comportamiento del Avatar en Tiempo Real

|  |  |
| --- | --- |
| CU-05 | Conversación con avatar en Tiempo Real |
| Requisitos relacionados | R.F.5 |
| Requisitos de los que depende | - |
| Actores | Externo |
| Descripción | El usuario externo accede a la funcionalidad de asesoramiento en tiempo real, donde puede interactuar directamente con un avatar configurado previamente. |
| Precondiciones | El avatar debe haber sido configurado previamente por el administrador. El usuario externo accede a través de una URL pública. |
| Acciones | 1. Mostrar botón “¿Necesita asesoramiento?” en la pantalla inicial  2. Cuando el usuario externo pulsa el botón:  → Cargar el agente de conversación (avatar)  3. Esperar a que el avatar se muestre correctamente  4. El usuario escribe su consulta en el campo de entrada de texto  5. El usuario pulsa el botón “Enviar”  6. El agente procesa la entrada usando parámetros configurados y un LLM (modelo de lenguaje)  7. Mostrar la respuesta del avatar en tiempo real  8. Si ocurre un error (conexión o proveedor):  → Mostrar mensaje de error al usuario |
| Postcondiciones | El usuario externo recibe la información del agente en tiempo real. |
| Excepciones | Si hay problemas con la conexión con el proveedor, el agente no se cargará y aparecerá un mensaje de error. |
| Importancia | Alta |

Tabla B6: CU-05 Conversación con avatar en Tiempo Real

|  |  |
| --- | --- |
| CU-06 | Buscar proveedor óptimo |
| Requisitos relacionados | R.F.6 |
| Requisitos de los que depende | R.F.1 |
| Actores | Experto |
| Descripción | Permite al administrador crear o modificar un avatar personalizado que podrá ser utilizado posteriormente en la generación de vídeos. |
| Precondiciones | |  | | --- | |  |  |  | | --- | | El usuario debe haber iniciado sesión como experto. Debe haber al menos un proveedor registrado y una configuración de pesos activa. | |
| Acciones | 1. Mostrar la pantalla con los campos de características técnicas disponibles  2. El usuario selecciona las características que desea utilizar como criterios de búsqueda  3. Por cada característica seleccionada, el usuario introduce un valor deseado (ej. precio, duración, etc.)  4. Pulsar el botón “Buscar”  5. Enviar la consulta al backend para que el sistema experto evalúe los proveedores registrados  6. Mostrar en pantalla el proveedor o proveedores con mayor similitud, según los pesos configurados  7. Si ocurre algún error (sin datos, error de cálculo, etc.):  → Mostrar mensaje en pantalla |
| Postcondiciones | Se muestra al usuario el proveedor más adecuado según los parámetros definidos y la lógica del sistema experto. |
| Excepciones | Si sucede algún error, será informado por pantalla. |
| Importancia | Alta |

Tabla B7: CU-06 Buscar proveedor óptimo

|  |  |
| --- | --- |
| CU-07 | Edición de proveedores |
| Requisitos relacionados | R.F.7 |
| Requisitos de los que depende | R.F.1, R.F.6 |
| Actores | Experto |
| Descripción | El usuario experto accede a la pantalla de gestión de proveedores, donde puede añadir nuevos proveedores, modificar sus datos o eliminar aquellos que ya no se deseen mantener en el sistema. Esta funcionalidad permite mantener actualizada la base de datos utilizada por el sistema experto. |
| Precondiciones | |  | | --- | |  |  |  | | --- | | El usuario debe haber iniciado sesión como experto y acceder a través de la pantalla “Edición de proveedores”. | |
| Acciones | 1. Mostrar la pantalla de proveedores registrados  2. El usuario elige una de las siguientes opciones:  - Pulsar “Nuevo proveedor” para añadir uno nuevo  - Seleccionar un proveedor existente para editarlo  - Seleccionar un proveedor para eliminarlo  3. Si se trata de una creación o edición:  - Mostrar formulario con campos técnicos (precio, minutos, resolución, etc.)  - Esperar a que el usuario complete y pulse “Guardar”  4. Si se trata de una eliminación:  - Confirmar con el usuario y pulsar “Eliminar proveedor”  5. Si la operación se ha realizado correctamente:  → Mostrar mensaje verde de confirmación  6. Si ocurre un error:  → Mostrar mensaje rojo informando del fallo |
| Postcondiciones | Si todo ha ido bien, aparecerá un mensaje con fondo verde con el éxito de la operación. |
| Excepciones | Si sucede algún error, será informado por pantalla con un mensaje de fondo rojo. |
| Importancia | Baja |

Tabla B8: CU-07 Edición de Proveedores

|  |  |
| --- | --- |
| CU-08 | Configuración de pesos |
| Requisitos relacionados | R.F.7 |
| Requisitos de los que depende | R.F.1, R.F.6 |
| Actores | Experto |
| Descripción | El usuario experto accede a la pantalla de configuración de pesos, donde puede ajustar la importancia relativa de cada característica técnica (precio, duración, funcionalidades, etc.). Estos valores afectan directamente a la lógica de selección del proveedor óptimo implementada en el sistema experto basado en reglas. |
| Precondiciones | |  | | --- | |  |  |  | | --- | | El usuario debe haber iniciado sesión como experto y acceder a través de la pantalla “Edición de proveedores”. | |
| Acciones | 1. Mostrar la pantalla con las características técnicas listadas (precio, duración, etc.)  2. El usuario introduce un valor numérico (peso) para cada característica según su importancia  3. Pulsar el botón “Guardar”  4. Enviar los datos al backend para actualizar la configuración de pesos  5. Si la operación es exitosa:  → Mostrar mensaje de confirmación en fondo verde  6. Si ocurre algún error:  → Mostrar mensaje informativo en fondo rojo |
| Postcondiciones | El sistema guarda los nuevos pesos, que se utilizarán en futuras búsquedas de proveedor óptimo. |
| Excepciones | Si sucede algún error, será informado por pantalla con un mensaje de fondo rojo. |
| Importancia | Media |

Tabla B9: CU-08 Configuración de pesos

# **Especificación de Diseño**

* 1. Introducción

Este anexo detalla los datos, la arquitectura y el diseño procedimental del proyecto.

* 1. Diseño de los datos

Para el manejo de los datos en la aplicación GoAvatar, se ha optado por el uso del formato JSON (JavaScript Object Notation) como estructura de representación, tanto para el almacenamiento como para las comunicaciones entre componentes frontend y backend. Esta elección permite trabajar con los datos de forma directa, sin necesidad de realizar procesos de transformación o mapeo, lo que simplifica el flujo de desarrollo e integración.

Con el objetivo de mantener esta estructura de datos de forma nativa, se ha seleccionado MongoDB [9] como sistema gestor de base de datos. MongoDB es una base de datos NoSQL orientada a documentos, en la que los registros se almacenan como documentos BSON (una extensión binaria de JSON). Al no ser relacional, MongoDB no requiere la definición previa de esquemas rígidos ni el establecimiento explícito de relaciones entre colecciones. Como es nuestro caso, que sólo queremos guardar contenido no relacionado.

A continuación, se identifican las colecciones:

Diagrama

El contenido generado por IA puede ser incorrecto.

Figura C1: Colecciones

* **avatar\_streams:** Esta colección guarda la configuración del agente que asesora en tiempo real.
* **características\_tecnicas:** Almacena las distintas características técnicas que se pueden utilizar para elegir el proveedor.
* **pesos:** Indica la relevancia que puede tener una característica técnica, sus valores pueden ser: IMPORTANTISIMO, ALTO, MEDIO, BAJO, MUY BAJO y DESPRECIABLE.
* **proveedores:** Esta colección identifica a cada proveedor y sus atributos.
* **videos:** Esta colección guarda la información relevante de los vídeos generados.
* **usuarios:** Sólo habrá 2 registrados: “experto” y “admin” que son aquellos que utilizarán la aplicación.
* **avatars:** No se usa actualmente. Esta colección estaba pensada para guardar la información de los avatares personales.

Un ejemplo de datos guardados en la tabla de videos:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Figura C2: Ejemplo JSON de video

Para gestionar las colecciones de datos en el backend desarrollado con Node.js [10], se ha utilizado la biblioteca Mongoose [11], que permite conectar con MongoDB de forma sencilla. Mongoose actúa como un intermediario entre la base de datos y la aplicación, permitiendo definir modelos de datos que representan las colecciones utilizadas en el sistema, como usuarios, proveedores o vídeos.

Una vez definidos estos modelos, el sistema puede realizar operaciones como guardar nuevos registros, consultar datos existentes, modificarlos o eliminarlos, utilizando funciones proporcionadas por esta librería. Esta aproximación facilita el acceso a los datos, asegura consistencia en la estructura de los documentos y mejora la mantenibilidad del código.

* 1. Diseño procedimental

En este apartado se describe el diseño procedimental del flujo más representativo del sistema GoAvatar: la generación de un vídeo a través del panel de administración. Se detalla el proceso completo desde que el administrador introduce los parámetros del vídeo en la interfaz, hasta que la petición llega al proveedor externo, se procesa, y el resultado es devuelto al sistema para su visualización y almacenamiento.

Este flujo constituye el núcleo funcional del sistema, al integrar tanto la lógica interna como la interacción con servicios de terceros. Acciones complementarias como el almacenamiento en base de datos o la actualización del estado del vídeo se consideran implícitas en este modelo, por lo que no se desarrollan de forma redundante en otros diagramas.

Además, se incluye el proceso de la conexión con el avatar virtual en tiempo real, que prescinde del backend y trabaja directamente con el SDK del proveedor.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figura C2: Proceso de Creación de vídeo (admin)

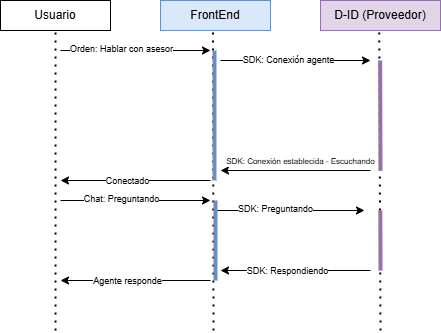


Figura C3: Proceso de asesoramiento avatar en tiempo real (externo)

* 1. Diseño arquitectónico

El diseño arquitectónico de la aplicación GoAvatar responde a un modelo cliente-servidor desacoplado, donde el frontend y el backend funcionan como componentes independientes que se comunican mediante solicitudes HTTPS a través de una API REST. Esta estructura favorece la escalabilidad, el mantenimiento y la reutilización de componentes, que es lo que se conoce actualmente como sistema distribuido.

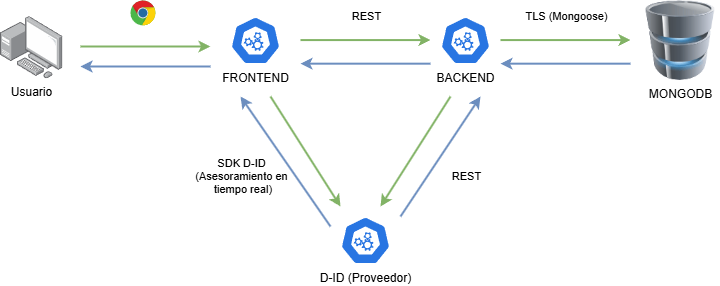


Figura C4: Arquitectura

Arquitectura general

La aplicación se divide en tres capas principales:

1. **Frontend (Cliente)**  
   Desarrollado con React [12], este módulo proporciona la interfaz gráfica que utilizan tanto el administrador como el experto y el usuario externo.
   * Gestiona la navegación con React Router.
   * Permite a los usuarios interactuar con formularios, consultar información y visualizar vídeos o sesiones en tiempo real.
   * Se comunica con el backend mediante peticiones HTTPS usando axios.
2. **Backend (Servidor)**  
   Desarrollado con Node.js y Express [13], actúa como API REST que centraliza la lógica del sistema.
   * Expone endpoints protegidos mediante autenticación JWT.
   * Gestiona la lógica de negocio, como la generación de vídeos o la selección de proveedores óptimos.
   * Se conecta a una base de datos MongoDB para almacenamiento persistente.
   * Incluye controladores para modularizar la lógica, modelos Mongoose y rutas bien estructuradas.
   * Controladores más relevantes:
     + **bestProviderController.js**

Contiene el método obtenerMejorProveedor, que implementa el sistema experto basado en reglas para determinar qué proveedor se ajusta mejor a los criterios técnicos introducidos por el usuario. Aplica pesos personalizados a cada característica y devuelve un ranking de similitud. Es el núcleo de la lógica de recomendación del sistema.

* + - **didController.js**

Gestiona la interacción con la API de D-ID. Incluye métodos como generarVideo (envío de datos multimedia para generar un avatar con voz e imagen), recibirWebhook (respuesta del proveedor una vez procesado el vídeo), guardarAvatarStream y crearAgente (configuración del avatar en tiempo real).

* + - **uploadController.js**

Encargado del tratamiento previo de archivos multimedia. Convierte audios y vídeos al formato adecuado, gestionando temporalmente los ficheros antes de ser enviados al proveedor externo. Ejecuta procesos como la conversión a MP3 mediante herramientas como FFmpeg.

* + - **xController.js**

Implementa la funcionalidad de publicación automática de vídeos en la red social X (Twitter), a partir del contenido generado por la plataforma. Permite ampliar la proyección del sistema y automatizar su difusión.

* + - **authController.js**

Responsable de la autenticación mediante tokens JWT. Incluye el método login y protege el acceso a las funcionalidades restringidas según el rol del usuario (admin o experto).

* + - **caracteristicaController.js y pesoController.js**

Estos dos controladores permiten leer, modificar y estructurar las características técnicas y sus pesos asociados. Aunque su funcionalidad es auxiliar, son fundamentales para el funcionamiento del sistema experto.

1. **Base de datos (MongoDB)**  
   Se utiliza MongoDB en la nube, con colecciones orientadas a documentos JSON.
   * Permite almacenar usuarios, proveedores, configuraciones de peso, vídeos generados, etc.
   * La conexión entre backend y MongoDB se establece mediante TLS cifrado.

Patrón Modelo-Vista-Controlador (MVC)

En el backend, se ha seguido el patrón MVC:

* **Modelo**: Se definen esquemas de datos con Mongoose, que representan colecciones como Proveedor, Video, CaracteristicasTecnicas, Usuario, etc.
* **Controlador**: Contiene la lógica que gestiona cada endpoint (ej. obtenerMejorProveedor, generarVideo, recibirWebhook, etc.).
* **Vista**: Aunque la vista como tal está en React, en esta arquitectura el backend sólo devuelve datos (JSON), por lo que no se generan vistas desde el servidor.

Comunicación con servicios externos

El sistema se integra con la API del proveedor D-ID mediante solicitudes HTTPS:

* Envío de datos para crear vídeos con avatar.
* Gestión del webhook de retorno para conocer cuándo se ha generado un vídeo.
* Visualización en tiempo real mediante el SDK de D-ID, integrado directamente en el frontend.

Despliegue

La arquitectura del sistema está preparada para ejecutarse en entornos cloud mediante el uso de contenedores, lo que permite una mayor portabilidad, escalabilidad y simplicidad en los despliegues. Para ello, se ha empleado Docker [14] como tecnología principal de empaquetado y despliegue.

Docker permite encapsular cada componente del sistema (frontend y backend) junto con todas sus dependencias en contenedores independientes. Esto garantiza que la aplicación se comporte de la misma forma en cualquier entorno, ya sea de desarrollo, pruebas o producción, sin necesidad de realizar configuraciones manuales complejas.

El proceso de despliegue se ha llevado a cabo de la siguiente manera:

* El frontend (React) y el backend (Node.js + Express) se ejecutan en Render.com [15], cada uno dentro de su propio contenedor Docker.
* La base de datos se aloja de forma externa en MongoDB Atlas [16], un servicio gestionado que permite escalar y administrar la base de datos sin preocuparse por la infraestructura subyacente.
* La comunicación entre estos servicios se realiza mediante protocolos seguros(HTTPS para la API y TLS para MongoDB).

El uso de Docker no sólo ha simplificado la configuración de los entornos de ejecución, sino que también ha facilitado tareas como la replicación del entorno en diferentes máquinas, la automatización del despliegue o la reducción de errores derivados de diferencias entre entornos locales y producción.

# **Documentación técnica de programación**

* 1. Introducción

Este apartado describe la información relacionada con el código, la configuración de ficheros y las rutas.

* 1. Estructura de directorios

Se parte del repositorio del proyecto en Github:

* <https://github.com/Maeroth/TFM_AVATAR>

La estructura contiene los siguientes directorios:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Figura D1: Directorios en Github

* **README.md**: Contiene las instrucciones para comenzar a utilizar el proyecto.
* **Docs**: Contiene toda la documentación asociada al proyecto, incluyendo tablas complementarias del sistema experto basado en reglas, imágenes y las colecciones en formato JSON necesarias para el proyecto.
* **ProjectAvatar**: Contiene tanto los ficheros fuente del BackEnd como del FrontEnd. A continuación se informa sobre las carpetas más importantes:
  + **frontend/src**:
    - components: Contiene módulos reutilizables en react de la aplicación.
    - pages: Páginas de la aplicación.
  + **backend/src:**
    - config: aquí se encuentran los ficheros de configuración de base de datos, principalmente.
    - controllers: todos los controladores, el motor de la lógica de negocio del backend.
    - models: Esquemas de Mongoose para MongoDB
    - routes: Definición de las rutas
    - uploads: En esta ruta, se guardan los ficheros que genera la aplicación para enviarlos a la API de D-ID. Después se borran. Por ejemplo, al generar un vídeo, D-ID puede necesitar la imagen de fondo elegida o el audio con la voz de una persona, estos se tienen que ubicar en una ruta visible para D-ID, de modo que usamos la propia aplicación donde alojarlos. Una vez que llega la petición resuelta por el Webhook, los borramos.
  1. Manual del programador

El proyecto se ha pensado para facilidad de despliegue, es decir, que toda la configuración inicial como librerías y dependencias ya están embebidas en un contenedor Docker, que es un entorno cerrado que se puede migrar sin tener que pasar otra vez por esa configuración inicial tan pesada.

No obstante, hay ficheros de configuración que conviene conocer para poder ajustar los parámetros de la aplicación .

Requisitos previos.

1. Hay que instalar Docker Desktop (<https://www.docker.com/products/docker-desktop>) , compatible con Windows, macOS y WS 2. Incluye Docker Engine y Docker Compose.
2. Tener acceso a un servidor con MongoDB instalado. Por ejemplo Atlas [16], que tiene un plan gratuito para clústeres de poca carga:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura D2: Clústeres en MongoDB Atlas

1. Cuenta y claves para D-ID o X (Twitter).
2. Tener cuenta en Docker-Hub [17]

Clonación del repositorio

Desde consola, clonar el repositorio:

>>git clone https://github.com/tu-usuario/TFM\_AVATAR.git

>>cd TFM\_AVATAR

Configuración del entorno

Una vez descargado el proyecto completo (FrontEnd y BackEnd), hay que crear dos ficheros de configuración, uno en cada proyecto.

1. En el directorio <RUTA\_PROYECTO>\frontend crear el fichero .env y añadir los siguientes parámetros:

#PARA PRODUCCIÓN

VITE\_API\_URL=

#TOKENS PARA VIDEO STREAMING

VITE\_DID\_KEY=

VITE\_AGENT\_ID=

Éstos parámetros son:

* + VITE\_API\_URL: Ruta del Backend
  + VITE\_DID\_KEY: Token del Avatar Streaming (D-ID) que identifica la aplicación.
  + VITE\_AGENT\_ID: Identificador del agente de (D-ID).

1. En el directorio <RUTA\_PROYECTO>\backend crear el fichero .env y añadir los siguientes parámetros:

PORT=5000

#MONGODB

MONGO\_URI=mongodb+srv://<USUARIO>:<CONTRASEÑA>@cluster0.dxkqsfr.mongodb.net/AvatarDB

JWT\_SECRET=secreto

#TOKENS D-ID

DID\_API\_KEY=

DID\_API\_URL=https://api.d-id.com

#TOKENS DE X (TWITTER)

TWITTER\_API\_KEY=

TWITTER\_API\_SECRET=

TWITTER\_ACCESS\_TOKEN=

TWITTER\_ACCESS\_SECRET=

#URL DE BACKEND ABIERTO

PUBLIC\_URL=

Éstos parámetros son:

* + MONGO\_URI: La ruta del clúster de MongoDB en Atlas, con el usuario y contraseña a configurar.
  + DID\_API\_KEY: Token generado en la plataforma D-ID para creación de vídeos.
  + VITE\_AGENT\_ID: Identificador del agente de (D-ID).
  + TWITTER\_API\_KEY: Token de Twitter.
  + TWITTER\_API\_SECRET: Token secreto de Twitter.
  + TWITTER\_ACCESS\_TOKEN: Token de acceso a Twitter.
  + TWITTER\_ACCESS\_SECRET: Token secreto de acceso a Twitter.
  + PUBLIC\_URL: URL del backend.

1. Por ahora, estos parámetros se pueden dejar vacíos hasta que tengamos los tokens o las urls requeridas (ver apartado **Obtención de Tokens**).

Trabajar en modo desarrollo

Para poder trabajar con todas las dependencias gracias a Docker, el desarrollador debe hacer unos ajustes en el siguiente fichero:

<RUTA\_PROYECTO>\docker-compose.yml

Reemplazar el contenido por este:

version: "3.9"

services:

frontend:

build:

context: ./frontend

ports:

- "3000:3000"

volumes:

- ./frontend:/usr/src/app

command: npm start

depends\_on:

- backend

backend:

build:

context: ./backend

ports:

- "5000:5000"

volumes:

- ./backend:/usr/src/app

environment:

- MONGO\_URI=mongodb+srv://<usuario>:<password>@<cluster>.mongodb.net/AvatarDB?retryWrites=true&w=majority

command: npm run dev

Revisar el **usuario,** **password** y **cluster** de MONGO\_URI. Este cambio es necesario para el modo desarrollo. El anterior era orientado a producción, donde las “tags” hacían referencia a las etiquetas de Docker-Hub donde se suben los volúmenes que luego se cargan en contenedores como Render.

Para que los cambios en el código local se reflejen directamente en los contenedores sin necesidad de reconstruir, introducir el siguiente comando (dentro de la raíz del proyecto).

>>docker-compose -f docker-compose.dev.yml up build

Subir a producción

Este salto es para añadir un contenedor a un repositorio Docker Hub, donde las plataformas de alojamiento web podrán leer la imagen y cargarlas. Para ello, se realizan los siguientes comandos.

>>docker login

>>docker build -t shirohara/avatar:frontend ./frontend

>>docker build -t shirohara/avatar:backend ./backend

>>docker push shirohara/avatar:frontend

>>docker push shirohara/avatar:backend

Nota: cambiar shirohara/avatar:frontend por una ruta válida generada en Docker-Hub. La actual es para pruebas.

Esto creará las imágenes de lo que se haya hecho en el proyecto y las subirá a Docker Hub, que, como digo anteriormente, lo utilizarán las webs de alojamiento (las que tengan Docker) para cargar directamente el contenido en el servidor.

Obtención de Tokens

# **Problemas encontrados**

-Se ha cambiado el CBR por un sistema experto basado en reglas porque la metodología CBR no se ajusta al proyecto (no se han podido probar todas las APIS porque algunas necesitan previo pago para usarlas).

-Zotero se deja de usar.

-Zapier no se termina de usar, mejor desde la aplicación (ZAPIER sale mejor interaccionando directamente con el proveedor, mejor lo hacemos nosotros desde la aplicación). Además, Zapier no está completamente integrado con D-ID.

¿Por qué REACT? No trabajad directamente con el DOM, sino con los componentes. Cada vez que un componente cambia de estado lo muestra por pantalla. Con Jquery o Javascript había que reescribir el DOM.

Se optó por node, ya que no se requiere hacer mucho backend. Pero a mitad de mayo, decidimos que podíamos aplicar la elección del proveedor en la aplicación en lugar de documentar las reglas. Esto supuso un sobreesfuerzo para jugar con node y realizar procesos en lógica de negocio.

Se iba a crear una pantalla de gestión de parámetros, donde se indicaría el token y la ruta donde se iban a colocar los ficheros, pero por temas de seguridad, se han hecho interno en la apliación, en el fichero .env. Se intentó hacer la pantalla para la configuración de ruta de descarga de videos pero al ser dos aplicaciones distintas (front de back) esto acarrea un peligro de seguridad.

Para ver los avatares en el navegador, hay que habilitar las cookies de terceros:

Captura de pantalla de un celular con la imagen de un hombre

El contenido generado por IA puede ser incorrecto.

Trabajaremos con las voces que ofrece Microsoft para mantener un estándar (amazon, Google… tienen su propio patrón). Las voces no se pueden probar (<https://docs.d-id.com/discuss/65eea7c7cae820002b702db7>)

Usamos ngrok para usarlo como puente y ser visibles desde local.

De cara al futuro: Mandar el vídeo vía correo según los clientes registrados.

Ha habido un problema al grabar el audio. El front guarda de forma nativa webm, pero D-ID requiere mp3, así que ha habido que reconvertir en el back.

25/05/2025 – Problema para generar subtítulos: https://docs.d-id.com/discuss/671f3725c3d7c70050ea4de3

Para configurar X:

Ir a Developer Portal y crearse una cuenta: <https://developer.x.com/en>

Una vez en el dashboard: <https://developer.x.com/en/portal/dashboard>

Lo primero es ir a App Settings

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Vamos a User atuthentication settings y pulsamos el botón Edit

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Marcamos en App persmissions: Read and write and Direct Message.

En Type of App marcamos Native App.

En App info ponemos el callback cuando se tenga la ruta del servidor [TODO] y la website.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Finalmente pinchamos save.

Ahora vamos a la pestaña Keys and tokens.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Y generamos las siguientes claves:

API Key and Sectret.

Access Token and Secret.

Las claves las colocamos en el fichero .env de backend para que todo funcione.

Patrón de fondo

El contenido generado por IA puede ser incorrecto.

STREAMS:

https://github.com/de-id/Agents-SDK-Demo

Key para el avatar streaming:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

La API de D-ID, en específico el LLM, te devuelve información en inglés si no se especifica el idioma. Hay que forzarlo a que devuelva en un idioma específico. Esto debería hacerse desde la creación del agente, pero no existe el campo “language”.

La generación de Avatar HD no está disponible en API pese a lo contrario. Los Avatares Express (que son los que ofrece le plan actual) no se catalogan como Premium y no aparecen en la lista de Avatares seleccionables actualmente. Debería implementarse una lista con Avatares Estándar y Express para la generación de clips.

Referencias Bibliográficas

1. Aamodt, A., & Plaza, E. (1994). *Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications, 7(1), 39–59*. <https://www.iiia.csic.es/~enric/papers/AICom.pdf>
2. GeeksforGeeks. (2024, August 27). *Rule‑Based System in AI*. Retrieved from <https://www.geeksforgeeks.org/rule-based-system-in-ai/>
3. ILERNA. (2024, 21 febrero). *¿Cuánto cobra un desarrollador web en España?* ILERNA Blog. <https://www.ilerna.es/blog/sueldo-desarrollador-web>
4. Seguridad Social Portal (2025). *Bases y tipos de cotización*. <https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537?changeLanguage=es>
5. Cadena SER (2025). *Estas son las horas en las que este domingo la luz tendrá un coste cero o negativo*. <https://cadenaser.com/nacional/2025/06/14/estas-son-las-horas-en-las-que-este-domingo-la-luz-tendra-un-coste-cero-o-negativo-cadena-ser>
6. Fuentes comparativas y artículos web consultados sobre generadores de avatares con IA (2025):
   1. HeyGen. (2025). *13 best AI avatar generators to explore in 2025*. <https://www.heygen.com/blog/ai-avatar-generator>
   2. D-ID. (2025). *The 15 best AI Avatar Generators of 2025*. <https://www.d-id.com/blog/best-ai-avatar-generators/>
   3. Synthesia. (2025). *Best AI Avatar Generator: Top 19 picks of 2025*. <https://www.synthesia.io/post/best-ai-avatar-generator>
   4. Synthesia. (2025). *7 Best AI Avatar Generators of 2025*. <https://www.synthesia.io/post/best-ai-avatar-generator>
   5. Vozo. (2025). *Best AI Avatar Generators in 2025*. <https://www.vozo.ai/blogs/best-ai-avatar-generators>
   6. Reddit user: u/AIToolTesting. (2025). *5 Generadores de vídeo con IA gratuitos para probar en 2025 – experiencia honesta*. <https://www.reddit.com/r/AIToolTesting/comments/1hm5gpr>
   7. Outmost Studio. (2025). *Best AI Avatars for Video Marketing Compared in 2025*. <https://www.outmost.studio/post/best-ai-avatars-for-video-marketing-reviewed-in-2025>
   8. Product Hunt. (2025). *The best avatar generators in 2025*. <https://www.producthunt.com/categories/avatar-generators?order=best_rated>
   9. MagicShot. (2025). *Top AI Avatar Creators You Need to Try in 2025*. <https://magicshot.ai/blog/top-ai-avatar-creators/>
   10. Synthesia. (2025). *¿Cuál es el mejor generador de avatares con IA? Nuestras 19 propuestas*. <https://www.synthesia.io/es/post/mejores-generadores-de-avatares-con-ia>
   11. Unite.AI. (2025). *10 Best AI Avatar Generators (April 2025)*. <https://www.unite.ai/best-ai-avatar-generators/>
   12. Synthesia. (2025). *The 18 Best AI Video Generators (Free & Paid) to Try in 2025*. <https://www.synthesia.io/post/best-ai-video-generators>
   13. ZebraCat. (2025). *11 Best AI Video Generators in 2025*. <https://www.zebracat.ai/post/11-best-ai-video-generators-in-2025>
7. PPIO (2025). *Tiempo de vídeo en créditos.* <https://help.pipio.ai/en/articles/9886186-what-are-credits>
8. D-ID (s.f.). *Portal de D-ID*. <https://www.d-id.com/>
9. MongoDB. (s. f.). *MongoDB Documentation*. <https://www.mongodb.com/docs>
10. Node.js. (s. f.). *Node.js*. <https://nodejs.org>
11. Mongoose. (s. f.). *Mongoose ODM*. <https://mongoosejs.com>
12. React. (s. f.). *React – A JavaScript library for building user interfaces*. <https://reactjs.org>
13. Express.js. (s. f.). *Express - Node.js web application framework*. <https://expressjs.com>
14. Docker Inc. (2024). *What is Docker?* Recuperado de <https://www.docker.com/resources/what-container/>
15. Alojamiento web Render (s.f.). *Despliega los contenedores Dockers*. <https://render.com/>
16. Cloud MongoDB Atlas (s.f.). *Clústers de MongoDB.* <https://cloud.mongodb.com/>
17. Docker Hub (s.f.). *Repositorio de imágenes Docker*. <https://hub.docker.com/>