PLAZZY

Adrián Primo Bernat

2º CFSW 23-24

|  |  |
| --- | --- |
|  |  |
|  |  |

Contenido

[1. Justificación del Proyecto 5](#_Toc167301199)

[2. Resumen 6](#_Toc167301200)

[3. Estudio de viabilidad 7](#_Toc167301201)

[Descripción del servicio 7](#_Toc167301202)

[Mercado 7](#_Toc167301203)

[Macroentorno 7](#_Toc167301204)

[Microentorno 9](#_Toc167301205)

[Tipo de empresa y sectores 10](#_Toc167301206)

[Estudio de proveedores 10](#_Toc167301207)

[Futuros usuarios 10](#_Toc167301208)

[Competencia 10](#_Toc167301209)

[Recursos materiales y humanos 11](#_Toc167301210)

[Personal necesario y perfil profesional 11](#_Toc167301211)

[DAFO 12](#_Toc167301212)

[Factores 12](#_Toc167301213)

[Estrategias 13](#_Toc167301214)

[Viabilidad económica 15](#_Toc167301215)

[Costes fijos y variables 15](#_Toc167301216)

[Evaluación riesgos 15](#_Toc167301217)

[4. Diseño 17](#_Toc167301218)

[Diseño de personajes: 17](#_Toc167301219)

[5. Diseño BBDD 19](#_Toc167301220)

[Diagrama Entidad-Relación 20](#_Toc167301221)

[Modelo Lógico Relacional 20](#_Toc167301222)

[Diseño Fisico 22](#_Toc167301223)

[Tabla juegos 22](#_Toc167301224)

[Tabla prompts 22](#_Toc167301225)

[Tabla jugadores 23](#_Toc167301226)

[Tabla salas 24](#_Toc167301227)

[Tabla respuestas 24](#_Toc167301228)

[Tabla respuestas\_emergencia 25](#_Toc167301229)

[Estructura Base de Datos 28](#_Toc167301230)

[6. Codificación 28](#_Toc167301231)

[Tecnologías elegidas 28](#_Toc167301232)

[Entorno servidor 29](#_Toc167301233)

[ Descripción general 29](#_Toc167301234)

[ Seguridad 29](#_Toc167301235)

[ Evitar o capturar errores y warnings 29](#_Toc167301236)

[Entorno cliente 29](#_Toc167301237)

[ Descripción general 29](#_Toc167301238)

[Documentación interna 30](#_Toc167301239)

[ Descripción de ficheros 30](#_Toc167301240)

[ Descripción de funciones 31](#_Toc167301241)

[7. Despliegue 37](#_Toc167301242)

[8. Conclusiones 40](#_Toc167301243)

[Conclusiones sobre el trabajo realizado 40](#_Toc167301244)

[Conclusiones personales 40](#_Toc167301245)

[Posibles ampliaciones y mejoras 40](#_Toc167301246)

[9. Bibliografía 41](#_Toc167301247)

1. Justificación del Proyecto

He escogido este proyecto por la razón de tener conexiones a tiempo real entre distintos clientes a un servidor mediante el uso del protocolo WebSocket y llevo un tiempo queriendo intentar hacer un proyecto de este estilo. Además de poder juntar a los usuarios para entretenerse y crear momentos divertidos para recordar.

1. Resumen

Castellano

***Plazzy*** es un portal web de juegos online sociales donde los usuarios pueden jugar a una variedad de juegos con sus amigos o seres queridos. La plataforma está diseñada para poder agregar nuevos modos de juego a lo largo del tiempo con nuevas maneras revolucionarias para entretener a nuestros usuarios.

El único modo de juego disponible en lanzamiento es:

* JokeBattle

***JokeBattle*** es un juego donde un par de jugadores responden o completan una misma frase de la manera más entretenida para ganar los votos del resto de jugadores, tiene una duración de tres rondas. El jugador con más puntos al final de la partida gana.

El próximo modo de juego planeado es el siguiente:

* Chatbot

En un mundo ante la inminente dominación de las maquinas, un grupo de amigos intentan encontrar a la inteligencia artificial infiltrada en ***Chatbot***. Los jugadores reciben un tema del que hablar por el grupo de chat mientras que la “IA” infiltrada recibe un tema relacionado, pero a la vez distinto. Los jugadores votaran en cada ronda quien creen que es el infiltrado y este será “expulsado” del grupo (seguirá jugando, pero no podrá ser votado en rondas posteriores) hasta encontrar al infiltrado. Si los jugadores fallan se muestra una animación de una ciudad arrasada de fondo.

English

***Plazzy*** is a social online gaming portal where users can play a variety of games with their friends or loved ones. The platform is designed to allow the addition of new game modes over time, introducing revolutionary ways to entertain our users.

The only available game mode on launch is:

* JokeBattle

***JokeBattle*** is a game where pairs of players respond to or complete the same phrase in the most entertaining way to win votes from the other players. It consists of three rounds. The player with the most points at the end of the game wins.

The next planned game mode is the following:

* Chatbot

In a world on the brink of machine domination, a group of friends tries to uncover the infiltrated artificial intelligence within ***Chatbot***. Players receive a topic to discuss in the chat group, while the “infiltrated AI” receives a related but distinct topic. In each round, players vote on who they believe is the infiltrator and he will be “banned” from the group (they can continue playing but cannot be voted on in the following rounds) until the true infiltrator is found. If the players fail to identify the infiltrator, an animation of a devastated city background is displayed.

1. Estudio de viabilidad

Descripción del servicio

Plazzy es una plataforma web que agrupa distintas clases de juegos donde se conectan entre 3 a 8 jugadores (puede variar por juego) y permite exportar los datos de la partida para poder compartirlo entre amigos y siempre tener los recuerdos que crearon gracias a nuestra plataforma.

El elemento diferenciador de nuestro servicio online es la API que hemos creado, en un futuro no muy lejano los usuarios podrán introducir sus propios “*prompts*” y respuestas de emergencia a sus juegos favoritos, siendo siempre monitorizados y filtrados por el sistema usando algoritmos de prevención de profanaciones.

Mercado

Macroentorno

* Factores sociales
  + Cultura y hábitos de consumo: Plazzy se beneficia de la creciente tendencia hacia el entretenimiento digital y la socialización en línea. Los consumidores buscan cada vez más experiencias interactivas y compartidas, lo que se alinea perfectamente con la propuesta de Plazzy.
  + Estilo de vida: Con el aumento del teletrabajo y el tiempo libre en casa, la demanda de plataformas de entretenimiento online ha crecido. Plazzy ofrece una manera accesible y divertida de conectar con amigos y familiares, adaptándose a estos cambios en el estilo de vida.
* Factores tecnológicos
  + Innovaciones tecnológicas: La implementación de tecnologías como WebSocket para conexiones en tiempo real y la posibilidad de agregar nuevos modos de juego continuamente posiciona a Plazzy en la vanguardia de la innovación en el entretenimiento digital.
  + Inversión en I+D: Plazzy se beneficia de la inversión en desarrollo de nuevas funcionalidades y modos de juego, lo que puede aumentar significativamente su competitividad en el mercado.
  + Digitalización: La plataforma aprovecha tecnologías digitales avanzadas como inteligencia artificial y big data para mejorar la experiencia del usuario y la personalización del contenido.
* Factores económicos
  + Crecimiento económico: La demanda de entretenimiento en línea está en auge, impulsada por un crecimiento económico que permite a más personas acceder a internet y dispositivos inteligentes.
  + Tasas de interés e inflación: Las políticas económicas que afectan el costo del crédito y los precios pueden influir en la capacidad de Plazzy para expandirse y mejorar sus servicios.
* Factores políticos y legales
  + Estabilidad política: Un entorno político estable favorece las inversiones en tecnología y el crecimiento de startups como Plazzy.
  + Regulaciones: Plazzy debe cumplir con diversas normativas legales, incluyendo leyes de protección de datos y regulaciones sobre contenidos online, lo cual puede impactar sus operaciones.
  + Política fiscal y comercial: Las políticas que facilitan el comercio electrónico y reducen las barreras de entrada pueden beneficiar la expansión internacional de Plazzy.
* Factores demográficos
  + Edad y estructura de la población: La plataforma se dirige a jóvenes adultos y familias, segmentos con alta participación en juegos en línea.
  + Migración: La diversidad cultural y el acceso global a internet permiten que Plazzy tenga un alcance potencialmente internacional.

Microentorno

* Competencia
  + Competidores directos: Plazzy compite con plataformas como *Jackbox* y *Gartic Phone*, que también ofrecen juegos sociales en línea. La ventaja competitiva de Plazzy radica en su accesibilidad gratuita y la capacidad de agregar nuevos modos de juego.
  + Competencia indirecta: Plataformas como redes sociales y otros servicios de entretenimiento online pueden ser competidores indirectos, ofreciendo alternativas a los juegos de Plazzy.
  + Estrategias competitivas: Plazzy debe analizar las estrategias de precios, promociones y diferenciación de sus competidores para mantenerse relevante y atraer a más usuarios.
* Proveedores
  + Relación con los proveedores: La calidad y fiabilidad de los proveedores de servicios de hosting y tecnología son cruciales para mantener un servicio eficiente y de alta calidad.
  + Dependencia de proveedores: Minimizar la dependencia de pocos proveedores es importante para reducir riesgos y asegurar una operación continua.
* *Clientes*
  + Segmentación del mercado: Plazzy identifica a jóvenes adultos, grupos de amigos y familias como sus principales segmentos de clientes, adaptando sus juegos y servicios a sus necesidades específicas.
  + Comportamiento del consumidor: Comprender las motivaciones y preferencias de los usuarios ayuda a Plazzy a mejorar la experiencia de juego y aumentar la fidelización.
  + Fidelización del cliente: Ofrecer actualizaciones continuas y nuevos modos de juego es clave para mantener a los usuarios interesados y comprometidos con la plataforma.
* Prescriptores
  + Influenciadores: Colaborar con youtubers, streamers y otros influenciadores puede ser una estrategia efectiva para aumentar la visibilidad y credibilidad de Plazzy.
  + Recomendaciones: Las recomendaciones positivas de usuarios satisfechos son fundamentales para atraer nuevos usuarios y fortalecer la reputación de la plataforma.

Tipo de empresa y sectores

* Empresa: Plazzy es una startup tecnológica centrada en el desarrollo de software y plataformas de entretenimiento en línea.
* Sectores: Entretenimiento, tecnología, software de juegos y redes sociales.

Estudio de proveedores

* Hardware: Servidores web y hosting (*Amazon Web Services* o *Microsoft Azure*).
* Software: Tecnologías utilizadas incluyen *MySQL*, *React*, *Node.js*, *Blender* y *Three.js*.
* Dominio web: Servicios de IONOS para el registro y gestión de dominios.

Futuros usuarios

La plataforma está dirigida a jóvenes adultos, grupos de amigos, familias y cualquier persona interesada en juegos sociales en línea, ofreciendo una experiencia de juego accesible y divertida.

Competencia

* Jackbox: Es un competidor bastante similar a nosotros en el sentido de que es un paquete de distintos juegos en uno, pero lo que nos diferencia es que nuestra plataforma es accesible gratuitamente y no dependemos de que un cliente comparta su pantalla de juego mediante videoconferencia o estando en la misma habitación.
* Pinturillo y Gartic Phone: Ambos competidores se enfocan en juegos de dibujo. Plazzy, al no tener aún un juego de este tipo, puede explorar oportunidades en este nicho.

Recursos materiales y humanos

Personal necesario y perfil profesional

* *Desarrolladores de Software:*
  + Perfil Profesional: Especializados en Node.js, React, y Three.js.
  + Sueldo Aproximado: 60.000 - 100.000 EUR anuales dependiendo de la experiencia y ubicación.
* *Diseñadores Gráficos:*
  + Perfil Profesional: Experiencia en Blender, diseño 3D e interfaces de usuario.
  + Sueldo Aproximado: 45.000 - 70.000 EUR anuales dependiendo de la experiencia y ubicación.
* *Especialistas en DevOps:*
  + Perfil Profesional: Gestión y mantenimiento de servidores.
  + Sueldo Aproximado: 70.000 - 110.000 EUR anuales dependiendo de la experiencia y ubicación.
* Especialistas en Marketing y Redes Sociales:
  + Perfil Profesional: Promoción y crecimiento de la plataforma.
  + Sueldo Aproximado: 45.000 - 80.000 EUR anuales dependiendo de la experiencia y ubicación.

Recursos materiales

* *Servidores: Hosting web y bases de datos*
  + Coste Aproximado:
    - Servidor Dedicado o VPS (Virtual Private Server): 45 - 450 EUR mensuales dependiendo de la capacidad y características.
    - Servicios en la Nube (AWS, Google Cloud, Azure): 90 - 900 EUR mensuales dependiendo del uso y recursos necesarios.
* Herramientas de Desarrollo
  + Ordenadores:
    - PCs de alto rendimiento: 1.300 - 2.600 EUR por unidad.
  + Entornos de Desarrollo (IDEs):
    - Licencia de DataGrip (JetBrains): 80 EUR por usuario al año.
    - Licencia de WebStorm (JetBrains): 115 EUR por usuario al año.
* Infraestructura de Red
  + Equipos de Red (Routers, Switches, Firewalls): 900 - 4.500 EUR dependiendo de la complejidad y necesidades.
  + Coste Mensual de Conexión a Internet (ISP): 90 - 450 EUR mensuales dependiendo de la velocidad y proveedor.

Resumen de Costes Anuales Aproximados

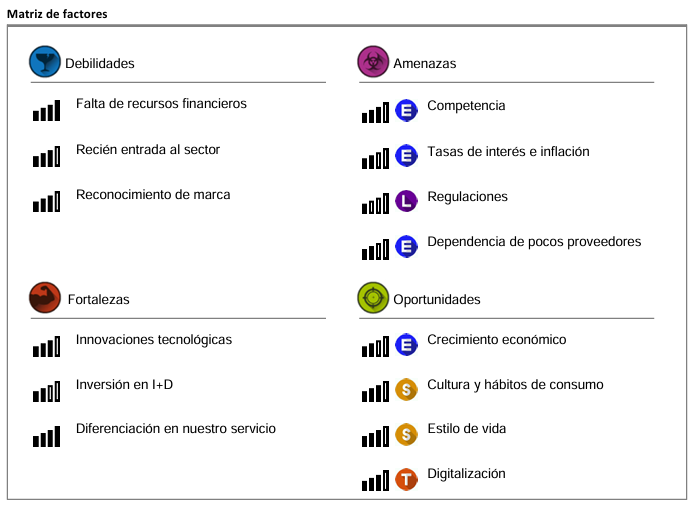
|  |  |
| --- | --- |
| **Concepto** | **Coste Anual** |
| |  |  | | --- | --- | | Desarrolladores de *Software* (3) |  | | 180.000 - 300.000 EUR |
| |  |  | | --- | --- | | Diseñadores Gráficos (2) |  | | 90.000 - 140.000 EUR |
| |  |  | | --- | --- | | Especialistas en *DevOps* (2) |  | | 140.000 - 220.000 EUR |
| |  |  | | --- | --- | | Especialistas en *Marketing* y Redes Sociales (2) |  | | 90.000 - 160.000 EUR |
| |  | | --- | | Servidores | | |  | | --- | | 1.080 - 10.800 EUR | |
| |  |  | | --- | --- | | Ordenadores (8) |  | | 10.400 - 20.800 EUR |
| |  |  | | --- | --- | | Licencias *IDEs* (8) |  | | 1.560 EUR |
| |  | | --- | | Equipos de Red | | |  | | --- | | 900 - 4.500 EUR | |
| |  |  | | --- | --- | | Conexión a Internet |  | | 1.080 - 5.400 EUR |

Total Aproximado Anual:

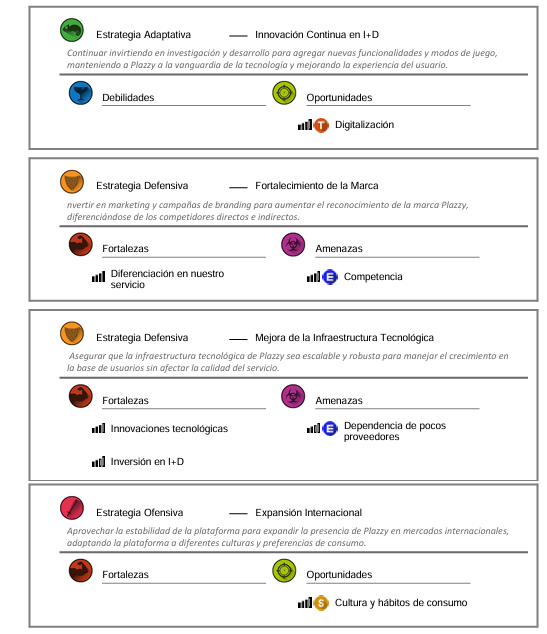
514.020 - 863.860 EUR

Análisis DAFO

Factores



Estrategias







Viabilidad económica

Costes fijos y variables

Tabla 1: Costes fijos y variables

|  |  |
| --- | --- |
| **Costes Fijos** | **Costes Variables** |
| Dominio Web | Actualización y mantenimiento de software. |
| Hosting y servidores | Marketing y promoción |
| Salarios del personal | Servicios adicionales basados en la demanda de usuarios |
| Licencias de software |  |

Calculo umbral rentabilidad

Para poder calcular el umbral de rentabilidad es necesario saber los costes fijos, el coste variable unitario y el precio de venta unitario.

El total de Costes Fijos asciende a 7.000€. El importe de Coste Variable unitario asciende a 50€. Y, por último, el Precio de Venta unitario es de 75€.

Por tanto, el cálculo de umbral de rentabilidad = 7.000€ / (75€ - 50€) = 280€.

Marketing

Estrategias de Marketing en Redes Sociales

1. Facebook e Instagram

* Contenido Regular: Publicar contenido atractivo y relevante diariamente, incluyendo:
  + Noticias sobre nuevos juegos y actualizaciones.
  + Tutoriales y guías de juegos.
  + Historias de éxito y testimonios de usuarios.
  + Imágenes y videos de alta calidad de los juegos.
* Publicidad Pagada:
  + Anuncios Segmentados: Utilizar anuncios de Facebook e Instagram para llegar a audiencias específicas basadas en intereses, comportamientos y datos demográficos.
  + *Retargeting*: Implementar campañas de *retargeting* para convertir visitantes del sitio web en usuarios registrados.
* Concursos y Sorteos:
  + Organizar concursos y sorteos mensuales para incentivar la participación y atraer nuevos usuarios.
* Colaboraciones:
  + Colaborar con páginas y grupos relevantes de juegos para aumentar la visibilidad.

2. X (Previamente Twitter)

* Actualizaciones en Tiempo Real: Compartir noticias y actualizaciones en tiempo real sobre la plataforma.
* Tendencias y Hashtags:
  + Utilizar hashtags populares y crear hashtags de marca (#*PlazzyGames*, #*PlayWithPlazzy*) para aumentar la visibilidad.
* Interacción con la Comunidad:
  + Responder a los comentarios y preguntas de los usuarios de manera oportuna.
  + Realizar encuestas y preguntas para fomentar la participación.

Estrategias de Marketing en YouTube

1. Canal Oficial de Plazzy

* Videos de Introducción:
  + Crear videos que presenten la plataforma y sus características únicas.
* Tutoriales y Guías:
  + Publicar videos tutoriales que expliquen cómo jugar y sacar el máximo provecho de los juegos en *Plazzy*.
* Videos de Actualización:
  + Publicar videos sobre nuevas actualizaciones, modos de juego y características.
* Contenido de Entretenimiento:
  + Crear contenido divertido y atractivo, como desafíos y partidas en vivo.
* Optimización SEO:
  + Optimizar los títulos, descripciones y etiquetas de los videos para mejorar la visibilidad en las búsquedas.

2. Publicidad en YouTube

* Anuncios *Pre-roll* y *Mid-roll*:
  + Crear anuncios cortos y atractivos que se muestren antes y durante videos relevantes.
* Colaboraciones con *YouTubers*:
  + Trabajar con *YouTubers* populares en la industria de los juegos para promocionar Plazzy a sus audiencias.

Estrategias con Influencers

1. Identificación de Influencers

* Micro-Influencers y Macro-Influencers:
  + Identificar influencers en el nicho de los juegos y el entretenimiento que tengan una audiencia comprometida.
* Plataformas:
  + Priorizar influencers en plataformas como Instagram, YouTube y Twitch.

2. Colaboraciones y Patrocinios

* *Streams* y Videos en Vivo:
  + Patrocinar *streams* y videos en vivo donde los *influencers* jueguen a los juegos de Plazzy y hablen sobre sus características.
* Reseñas y Tutoriales:
  + Colaborar con *influencers* para crear reseñas y tutoriales detallados de los juegos en Plazzy.
* Desafíos y Competencias:
  + Organizar desafíos y competencias con *influencers* para atraer a sus seguidores a la plataforma.
* Código de Referencia:
  + Proporcionar a los *influencers* códigos de referencia únicos que sus seguidores puedan usar para registrarse, ofreciendo beneficios tanto a los nuevos usuarios como a los *influencers*.

Calendario de Implementación

Primer Mes: Preparación y Lanzamiento

* Configuración de cuentas de redes sociales y creación de contenido inicial.
* Producción de los primeros videos de *YouTube* y lanzamiento del canal.
* Identificación y contacto con *influencers* relevantes.

Segundo Mes: Campañas de Lanzamiento

* Inicio de campañas de publicidad pagada en Facebook, Instagram y YouTube.
* Publicación de los primeros videos colaborativos con *influencers*.
* Inicio de concursos y sorteos en redes sociales.

Tercer a Sexto Mes: Expansión y Optimización

* Análisis de resultados de las campañas iniciales y ajustes según el rendimiento.
* Creación de contenido adicional y regular en todas las plataformas.
* Continuación de colaboraciones con *influencers* y expansión a nuevos *influencers*.
* Implementación de nuevas características y modos de juego en la plataforma, promocionados a través de todas las estrategias mencionadas.

Sexto Mes en Adelante: Consolidación y Crecimiento Continuo

* Evaluación continua de estrategias y ajuste de tácticas para maximizar el crecimiento.
* Innovación constante en el contenido y las formas de interacción con la comunidad.
* Fomento de una comunidad activa y comprometida a través de interacciones constantes y contenido de alta calidad.

Plan de prevención de riesgos laborales

Evaluación riesgos

Ahora hablaremos de algunos de los riesgos laborales más comunes en el desarrollo web.

* Riesgos ergonómicos

Las largas horas frente ordenador son capaces de causar problemas de postura, fatiga visual y lesiones musculoesqueléticas.

* Riesgos visuales

La carga de trabajo constante frente a pantallas puede causar fatiga visual, sequedad ocular, visión borrosa y otros problemas oculares

* Riesgos psicosociales

El desarrollo web puede ser un trabajo estresante, especialmente cuando se enfrenta a plazos ajustados, presión constante y demandas de clientes. Esto puede llevar a problemas de estrés, ansiedad y agotamiento.

* Riesgos eléctricos

Los equipos electrónicos con los que se trabajan pueden causar descargas eléctricas en los trabajadores.

* Riesgos de seguridad informática

Los desarrolladores web están expuestos a riesgos de seguridad informática, como ataques de hackers, robo de datos y virus informáticos.

* Riesgos de sedentarismo

Esta clase de trabajo suele ser sedentario, lo que puede llevar a problemas de salud como el sobrepeso, la obesidad y enfermedades cardiovasculares.

Medidas de prevención y protección

Ahora que hemos valorado los posibles riesgos debemos hablar de las medidas de prevención y protección:

* Prevención
  + Proporcionar equipos ergonómicos, asegurándose que los trabajadores tienen muebles ajustables y periféricos ergonómicos para evitar lesiones musculoesqueléticas.
  + Establecer pausas regulares: Fomentar que los desarrolladores web tomen descansos regulares para evitar la fatiga visual y el estrés físico.
  + Establecer pausas regulares: Fomentar que los desarrolladores web tomen descansos regulares para evitar la fatiga visual y el estrés físico.
  + Realizar copias de seguridad regulares: Asegurarse de que los datos y proyectos de los desarrolladores web estén respaldados regularmente para evitar la pérdida de información importante.

1. Diseño

Diseño de personajes:

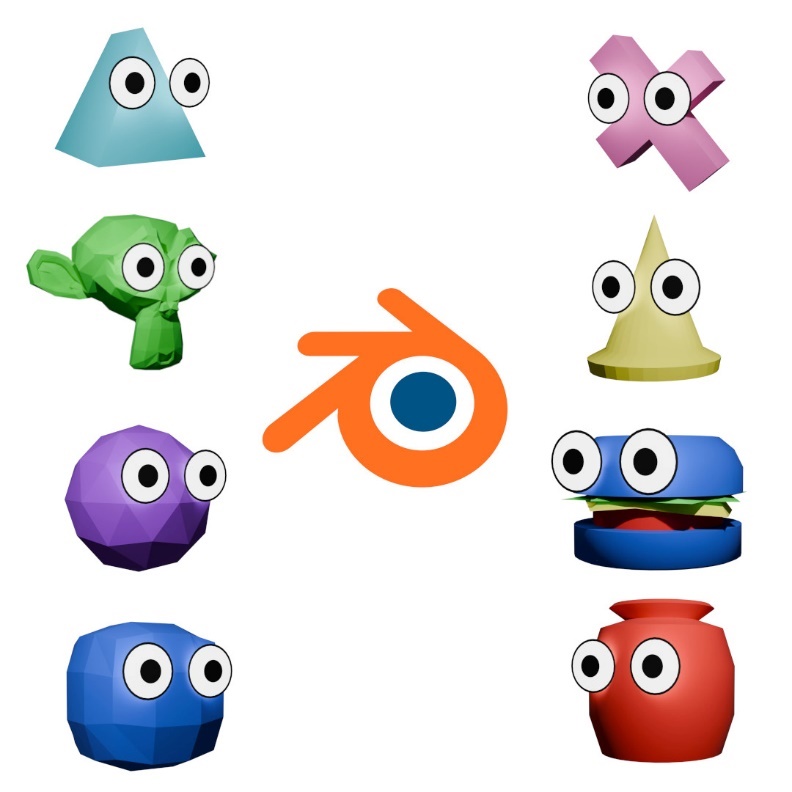
Para los personajes del juego he empleado la herramienta *Blender,* he modelado y exportado a imágenes para poder ser usadas en el juego.

Figura 1: Diseño de personajes creados en Blender

El motor de renderizado utilizado en el software es *Cycles*, ya que genera exportaciones de mayor calidad que su alternativa, *Eevee*.

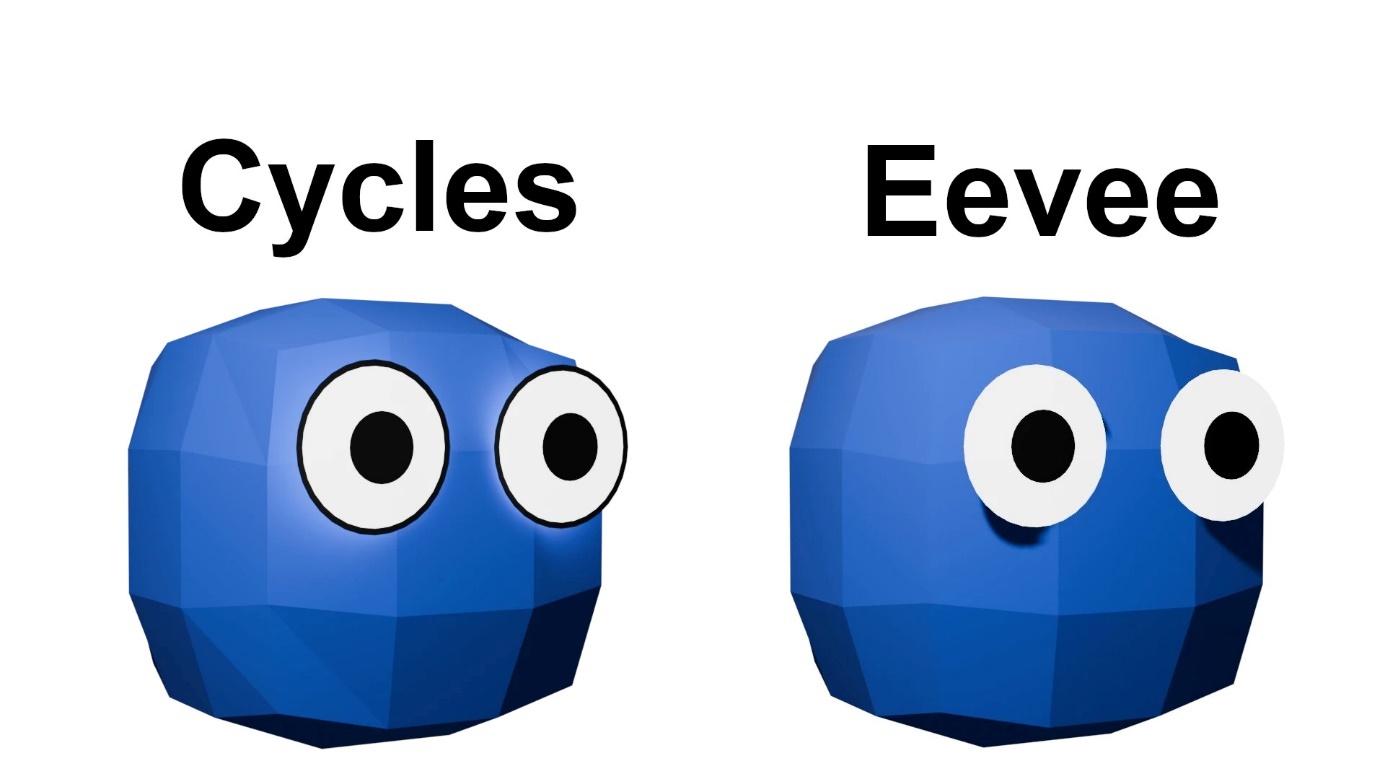


Figura 2: Comparación renderizades entre Cycles y Eevee

Los dos motores tienen diferencias a la hora de crear materiales, como por ejemplo en *Cycles* puedes hacer un contorno alrededor de un objeto a través de un material.



Las imágenes son exportadas con formato *WebP*, el cual optimiza el tamaño de almacenamiento que ocupan en el servidor.

Estos modelos pueden ser utilizados en la web usando la librería *ThreeJS* y exportándolos con un formato *glTF* (*GL Transmission Format*).

1. Diseño BBDD

Nuestra base de datos es relativamente simple, almacenamos la información de las salas para que nuestros usuarios puedan volver a visualizarla y recordar los momentos que pasaron juntos. Los datos prioritarios porque son necesarios para poder volver a visualizar una partida Plazzy son los usuarios, las frases (denominados “*prompts*” en el sistema) y las respuestas.

Diagrama Entidad-Relación

En nuestro diagrama entidad-relación se puede visualizar las diferentes tablas para poder revivir una partida *Plazzy*.

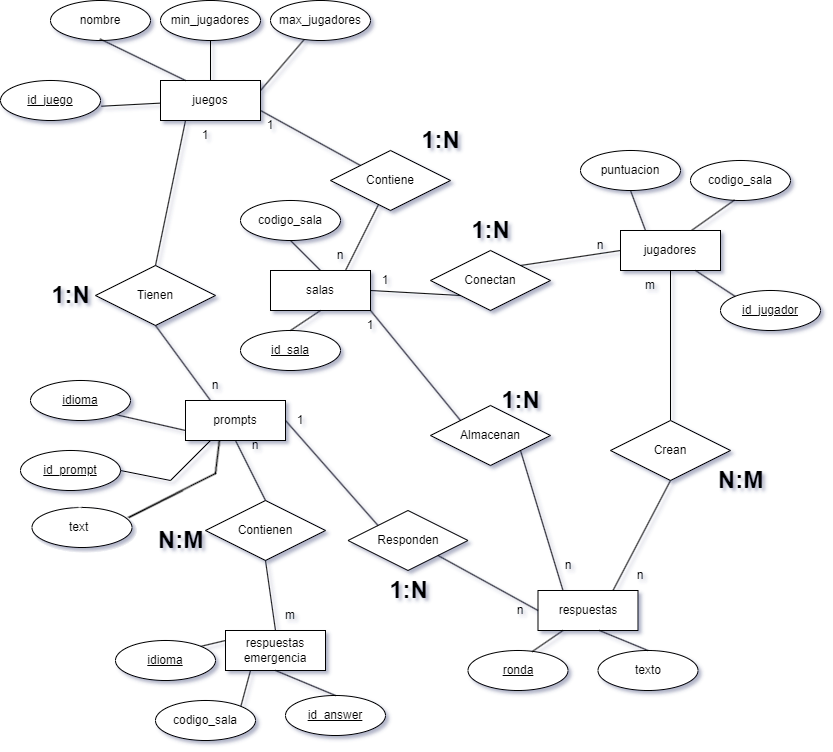


Figura 3: Diagrama Entidad-Relación Plazzy

Modelo Lógico Relacional

Clave Primaria: PK

Clave Foránea: ^FK

juegos(id\_juego, nombre, min\_jugadores, max\_jugadores)

* id\_juego: Valor entero
* nombre: Cadena variable de 20 caracteres
* min\_jugadores: Valor entero
* max\_jugadores: Valor entero

Clave primaria: id\_juego

salas(id\_sala, id\_modoJuego, codigo\_sala)

* id\_sala: Cadena variable de 50 caracteres
* ^id\_modoJuego (Referenciando a id\_juego en juegos): Valor entero
* codigo\_sala: Cadena de 4 caracteres fijos

Clave Primaria: id\_sala

prompts(id\_prompt, text, id\_juego, idioma)

* id\_prompt: Valor entero
* text: Cadena variable de 100 caracteres
* ^id\_juego (Referenciando a id\_juego en juegos): Valor entero
* idioma: Enum

Clave primaria: id\_prompt, id\_juego, idioma

jugadores(id\_jugador, nombre\_jugador, puntuacion, id\_sala)

* id\_jugador: Cadena variable de 50 caracteres
* nombre\_jugador: Cadena variable de 50 caracteres
* puntuacion: Valor entero
* ^id\_sala (Referenciando a id\_sala en salas): Cadena variable de 50 caracteres

Clave primaria: id\_jugador, id\_sala

respuestas(texto, id\_jugador, id\_sala, id\_prompt, ronda)

* texto: Cadena variable de 30 caracteres
* ^id\_jugador (Referenciando a id\_jugador en jugadores): Cadena variable de 50 caracteres
* ^id\_sala (Referenciando a id\_sala en salas): Cadena variable de 50 caracteres
* ^id\_prompt (Referenciando a id\_prompt en prompts): Valor entero
* ronda: Valor entero

Clave primaria: id\_jugador, id\_sala, id\_prompt, ronda

respuestas\_emergencia(text, id\_answer, id\_prompt, idioma)

* text: Cadena variable de 50 caracteres
* id\_answer: Valor entero
* ^id\_prompt (Referenciando a id\_prompt en prompts): Valor entero
* idioma: Enum

Clave priimaria: id\_answer, id\_prompt, idioma

Diseño Fisico

Tabla juegos

create table juegos  
(  
 id\_juego int auto\_increment  
 primary key,  
 nombre varchar(20) not null,  
 min\_jugadores int default 3 not null,  
 max\_jugadores int default 8 not null  
);  
  
INSERT INTO plazzy.juegos (id\_juego, nombre, min\_jugadores, max\_jugadores) VALUES (0, 'JOKEBATTLE', 3, 8);  
INSERT INTO plazzy.juegos (id\_juego, nombre, min\_jugadores, max\_jugadores) VALUES (1, 'CHATBOT', 4, 8);

Tabla prompts

create table prompts  
(  
 id\_prompt int not null,  
 id\_juego int not null,  
 text varchar(100) null,  
 idioma enum ('ES', 'EN') default 'ES' not null,  
 primary key (id\_prompt, id\_juego, idioma),  
 constraint fk\_juego\_prompts  
 foreign key (id\_juego) references juegos (id\_juego)  
);  
  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (0, 0, 'El mayor amigo del perro es ...', 'ES');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (0, 0, 'Dog\'s best friend is ...', 'EN');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (1, 0, 'Estás en 1945, Berlín. Lo primero que haces es ...', 'ES');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (1, 0, 'You\'re in 1945, Berlin. First thing you do is ...', 'EN');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (2, 0, 'Después de ganar la lotería, decides...', 'ES');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (2, 0, 'After winning the lottery, you decide to ...', 'EN');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (3, 0, 'En un mundo habitado por unicornios y dragones, tu misión es...', 'ES');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (3, 0, 'In a world inhabited by unicorns and dragons, your mission is ...', 'EN');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (4, 0, 'Tu superpoder recién descubierto es...', 'ES');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (4, 0, 'Your newly discovered superpower is ...', 'EN');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (5, 0, 'El hashtag que usarías en un golpe de estado', 'ES');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (5, 0, 'The hashtag you\'d use in a coup d\'etat', 'EN');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (6, 0, 'Si pudieras viajar en el tiempo, irías a...', 'ES');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (6, 0, 'If you could travel in time, you\'d go to ...', 'EN');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (7, 0, 'En un mundo sin tecnología, tu habilidad más valiosa sería...', 'ES');  
INSERT INTO plazzy.prompts (id\_prompt, id\_juego, text, idioma) VALUES (7, 0, 'In a world without technology, your most important skill would be ...', 'EN');

Tabla jugadores

create table jugadores  
(  
 id\_jugador varchar(50) not null,  
 id\_sala varchar(50) not null,  
 nombre\_jugador varchar(10) not null,  
 puntuacion int not null,  
 primary key (id\_jugador, id\_sala),  
 constraint fk\_JugadorSala  
 foreign key (id\_sala) references salas (id\_sala)  
);

Tabla salas

create table salas  
(  
 id\_sala varchar(50) not null  
 primary key,  
 id\_modoJuego int not null,  
 codigo\_sala char(4) not null,  
 constraint fk\_JuegoSala  
 foreign key (id\_modoJuego) references juegos (id\_juego)  
);

Tabla respuestas

create table respuestas  
(  
 id\_jugador varchar(50) not null,  
 id\_sala varchar(50) not null,  
 id\_prompt int not null,  
 ronda int not null,  
 texto varchar(30) not null comment 'El límite de caracteres de las respuestas de jugadores es de 30',  
 primary key (id\_jugador, id\_sala, id\_prompt, ronda),  
 constraint fk\_RespuestaJugador  
 foreign key (id\_jugador) references jugadores (id\_jugador),  
 constraint fk\_RespuestaPrompt  
 foreign key (id\_prompt) references prompts (id\_prompt),  
 constraint fk\_RespuestaSala  
 foreign key (id\_sala) references salas (id\_sala)  
);

Tabla respuestas\_emergencia

create table respuestas\_emergencia  
(  
 id\_answer int not null,  
 id\_prompt int not null,  
 text varchar(30) null,  
 idioma enum ('ES', 'EN') default 'ES' not null,  
 primary key (id\_answer, id\_prompt, idioma),  
 constraint safety\_answers\_prompts\_id\_prompt\_fk  
 foreign key (id\_prompt) references prompts (id\_prompt)  
 on update cascade on delete cascade  
);  
  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 0, 'su pelota', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 0, 'run away', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 1, 'huir', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 1, 'ask for a loan', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 2, 'pedir un préstamo', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 2, 'create a dragon farm', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 3, 'crear una granja de dragones', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 3, 'attract wealth', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 4, 'atraer riquezas', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 4, '#LikeTheOldDays', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 5, '#ComoLosViejosTiempos', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 5, 'avoid doing the FDP', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 6, 'evitar hacer el TFG', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 6, 'do the worm', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (0, 7, 'hacer el gusano', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 0, 'su hueso', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 0, 'shave your moustache', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 1, 'cortarte el bigote', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 1, 'share the prize', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 2, 'compartir el premio', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 2, 'rescue the unicorn king', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 3, 'rescatar al rey unicornio', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 3, 'swim in lava', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 4, 'nadar en lava', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 4, '#TomorrowJuanOustsHim', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 5, '#MañanaLoDestituyeJuan', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 5, 'avoid validating business', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 6, 'evitar convalidarme empresa', 'ES');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 6, 'teach tricks to pigs', 'EN');  
INSERT INTO plazzy.respuestas\_emergencia (id\_answer, id\_prompt, text, idioma) VALUES (1, 7, 'enseñar trucos a cerdos', 'ES');

Estructura Base de Datos

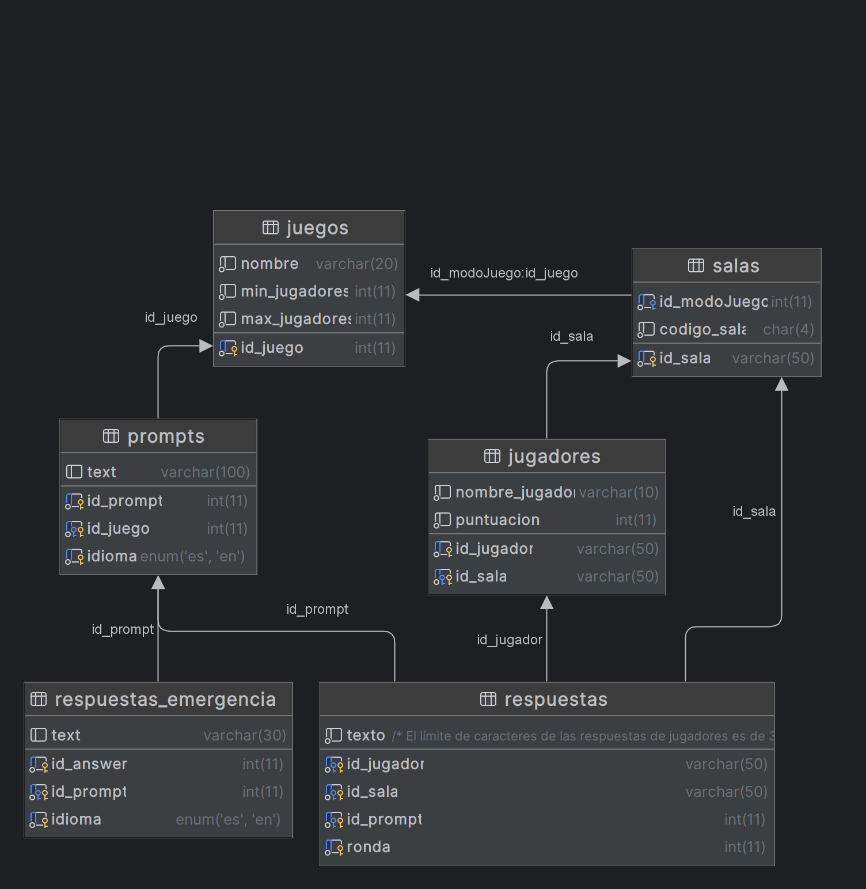


Figura 4: Diagrama BBDD generado con JetBrains DataGrip

1. Codificación

Tecnologías elegidas

* **NodeJS**: Es el entorno que tiene la total responsabilidad de lógica lado servidor, hemos decidido que es una gran elección ya que quita complejidad poder usar el mismo lenguaje para lado cliente y servidor, porque utiliza el lenguaje *Javascript*.
* **React**: Es el *framework* utilizado para el lado cliente, sus componentes y funcionalidades permiten fácilmente crear elementos reutilizables y *SPAs* (*Single-Page Applications*).
* **MySQL:** Es el sistema gestor de base de datos usado para almacenar datos de partidas creadas y jugadas, la razón por la que se eligió por la facilidad de uso y familiaridad gracias a la experiencia que tenemos usándolo.
* **Socket.IO**: Es una biblioteca para *NodeJS* que mantiene una comunicación bidireccional a tiempo real, es utilizada por la facilidad de enviar mensajes entre servidor y clientes para poder hacer operaciones típicas de juegos en línea, como puede ser votar o enviar mensajes de texto entre jugadores.
* **Three.js**: Es una biblioteca de Javascript que permite mostrar gráficos 3D en la web mediante la *API* de *WebGL*, la razón principal de su implementación es la posibilidad de crear fondos 3D animados y crear otros elementos interesantes y llamativos atados al juego.

Entorno servidor

* Descripción general

El entorno *Back-End* de la aplicación tiene tres capas, la base de datos que almacena información importante sobre los juegos disponibles y partidas jugadas anteriormente, el servidor de juego que mantiene todas las salas de juego a tiempo real y proporciona servicio a los jugadores, y finalmente la API, que permite al servidor de juego comunicarse con la base de datos para obtener *prompts* o guardar las respuestas de los usuarios, por ejemplo.

* Seguridad

Los datos se envían y almacenan seguramente gracias a las rutas creadas en la *API*. Las razones por las que las consultas que se hacen son seguras son las siguientes:

1. Consultas Parametrizadas: Utilizan ‘?’ como marcadores de posición para los valores que se van a insertar en las consultas SQL. Los valores reales se pasan como un arreglo en la llamada a *db.query*.
2. Manejo de Errores: Cada consulta maneja los errores apropiadamente, devolviendo los códigos de respuesta relacionados al estado del resultado.
3. Validación de Entrada: Aunque no se muestra validación de entrada explícita (como verificar el tipo o el formato de los datos recibidos), el uso de consultas parametrizadas es un gran paso para asegurar la integridad de la base de datos.

* Evitar o capturar errores y warnings

Todas las funciones que reciben un resultado de una consulta mediante la API comprueban primero si el resultado es correcto y actúan acuerdo a los datos recibidos.

Entorno cliente

* Descripción general

El entorno cliente actúa a base de eventos emitidos por el servidor de juego, como por ejemplo cambiando la pantalla o recibiendo la información de los jugadores de la sala.

Documentación interna

* Descripción de ficheros

Hablaremos solamente sobre los ficheros que crean funcionalidad en la aplicación (no se incluyen hojas de estilo CSS).

|  |  |
| --- | --- |
| **Nombre** | **Funcionalidad** |
| /src/scripts/db.js | Proporciona la conexión a la base de datos. |
| /src/scripts/api.js | Proporciona funciones mediante rutas para recibir o enviar datos. |
| /src/scripts/server.js | Administra las salas activas de juegos y proporciona la gran mayoría de lógica de juego. |
| /src/scripts/cliente.js | Proporciona la conexión al servidor de juego y el evento de error de conexión a la sala. |
| /src/componentes/ComponentesComunes.js | Alberga varios componentes de React que pueden ser reutilizados en distintos juegos. |
| /src/componentes/ModeloJugador.js | Contiene un único componente de React que se encarga de devolver un modelo 3D usando Three.js |
| /src/componentes/UnirseAPartida.js | El componente que se muestra cuando accedes a la web mediante un QR o la ruta /game/:codigoPartida |
| /src/componentes/MainMenu.js | Es el menú principal que permite crear o unirse a una partida. |
| /src/componentes/juegos/Juego.js | Es una clase React plantilla para poder crear juegos siguiendo una misma estructura. |
| /src/componentes/juegos/JokeBattle.js | Es una clase React que ejecuta la lógica cliente del juego *JokeBattle* |
| /src/componentes/juegos/Chatbot.js | Es una clase React que ejecuta la lógica cliente del juego en desarrollo *Chatbot* |

* Descripción de funciones

|  |  |  |
| --- | --- | --- |
| Fichero | Función | Explicación |
| /src/scripts/server.js | closeLobby(lobbyCode) | Cierra la sala en el servidor y desconecta a los usuarios pertenecientes. |
| generateRandomCode() | Genera un código aleatorio de la sala |
| generateUUID() | Genera un identificador aleatorio para la sala |
| getLobby(lobbyCode) | Devuelve una referencia a una sala usando su código de sala |
| randInt(min, max) | Devuelve un entero aleatorio entre *min* y *max* inclusivamente |
| generatePromptsForPlayers(lobbyCode, language) | Obtiene prompts desde el servidor y los asigna a los jugadores de la sala en el servidor de juego |
| getPromptID(prompt, language) | Obtiene el id de un prompt pasándole el texto |
| crearLobbyBBDD(lobby) | Crea una referencia de la sala en la base de datos usando los datos de la sala |
| crearJugadoresBBDD(lobbyId, player) | Crea la referencia de un jugador en la base de datos |
| clearLobbyData(lobby) | Limpia los datos del lobby (prompts y respuestas) en el servidor de juego |
| devolverJugadoresPorPuntuacion(jugdores) | Devuelve una copia del array de jugadores de una sala ordenados por puntuación |
| getRandomSafetyAnswer(id,language) | Devuelve una respuesta de emergencia aleatoria |
| playerUseSafetyAnswer(lobbyCode, playerID) | Almacena en respuestas de un jugador una respuesta de emergencia obtenida aleatoriamente |
| comprobarNumeroDeRespuestas(lobbyCode) | Comprueba si todos los jugadores conectados a una sala han enviado todas sus respuestas restantes |
| getPromptsUnicos(lobby) | Devuelve una copia del array de prompts de la sala donde no se repiten |
| getAnswersDePrompts(lobby, prompt) | Obtiene las respuestas creadas por jugadores de un prompt |
| associateAnswersToUniquePrompt(lobby) | Sobreescribe en el array de respuestas de jugadores las respuestas ordenadas por prompts unicos |
| /src/componentes/ComponentesComunes.js | IconoJugador({nombreClase, nombre, rutaImagen, style}) | Componente de React que genera un Icono de Jugador 2D |
| IconoLoby({gameCode}) | Componente de React que genera un cuadrado mostrando el codigo de la partida y dominio Web |
| CodigoPartida({gameCode}) | Componente que devuelve un encabezado con el codigo de la sala |
| Contador({className, tiempoInicial, onTiempoTerminado }) | Componente temporizador que muestra en pantalla el tiempo restante y ejecuta una función pasada por referencia |
| InputRespuestaLimitado({socket, playerID, gameCode, onHandleSubmitRef ,styles, maxLength = 30}) | Componente de entrada de texto que muestra cuantos caracteres restantes te permite introducir y envía un evento de respuesta al servidor de juego |
| /src/componentes/FondoColoresRandom.js | randomColor() | Función que devuelve un string de color hexadecimal |
| FondoColoresRandom() | Componente de React que muestra un fondo de colores aleatorios en una gradiente lineal y cambia siguiendo un intervalo |
| /src/componentes/FondoColoresRandom.js | GamePage() | Componente de React que mira el parámetro de ruta de /game/ para unirse a una partida con solo introducir el nombre de usuario |
| /src/componentes/ModeloJugador.js | ModeloJugador({modeloPath, animationName, bloquearRespuestas}) | Componente de React que devuelve una escena 3D para ser utilizada en un Canvas usando un modelo y un parámetro para la animación a ejecutar |
| /src/componentes/MainMenu.js | MenuCrear({volverAlMenu, crearPartida}) | Componente de React que permite al usuario crear una sala |
| MenuUnirse({menuCrear, unirsePartida}) | Componente de React que permite al usuario unirse a una sala |
| MenuPrincipal({onCreate, onJoin}) | Componente que permite intercambiar entre MenuUnirse y MenuCrear |
| Index({gameCodeRef = null, playerRef = null}) | Componente que se conecta al servidor de juego para manejar ciertos eventos relevantes al cliente |
| /src/componentes/juegos/Juego.js | Juego() | Clase de React que se usa como plantilla para el desarrollo de juegos |
| /src/componentes/juegos/Chatbot.js | Chatbot() | Clase de React que hereda de Juego para jugar a *Chatbot* |
| /src/componentes/juegos/JokeBattle.js | JokeBattle () | Clase de React que hereda de Juego para jugar a *JokeBattle* |
| Prompt({texto}) | Componente que devuelve un prompt animado |
| Noria({jugadores, offset = 0}) | Componente de React que devuelve una noria con iconos de jugadores rotando sobre ella |
| RespuestaPrompt({  texto, propietario,  desdeIzquierda,  senalMostrarRespuestas, senalMostrarPropietarios,onClick }) | Componente que devuelve una respuesta de jugador animada |
| IntroduccionJokeBattle() | Componente que devuelve como jugar a *JokeBattle* |
| AnimatedPropietario({propietario, senalMostrarPropietarios, senalMostrarRespuestas}) | Componente que muestra el propietario de una respuesta |
|  |  |  |

1. Despliegue

Para desplegarlo en la web hemos utilizado como hosting *Microsoft Azure*, y hemos comprado un dominio en *IONOS*.

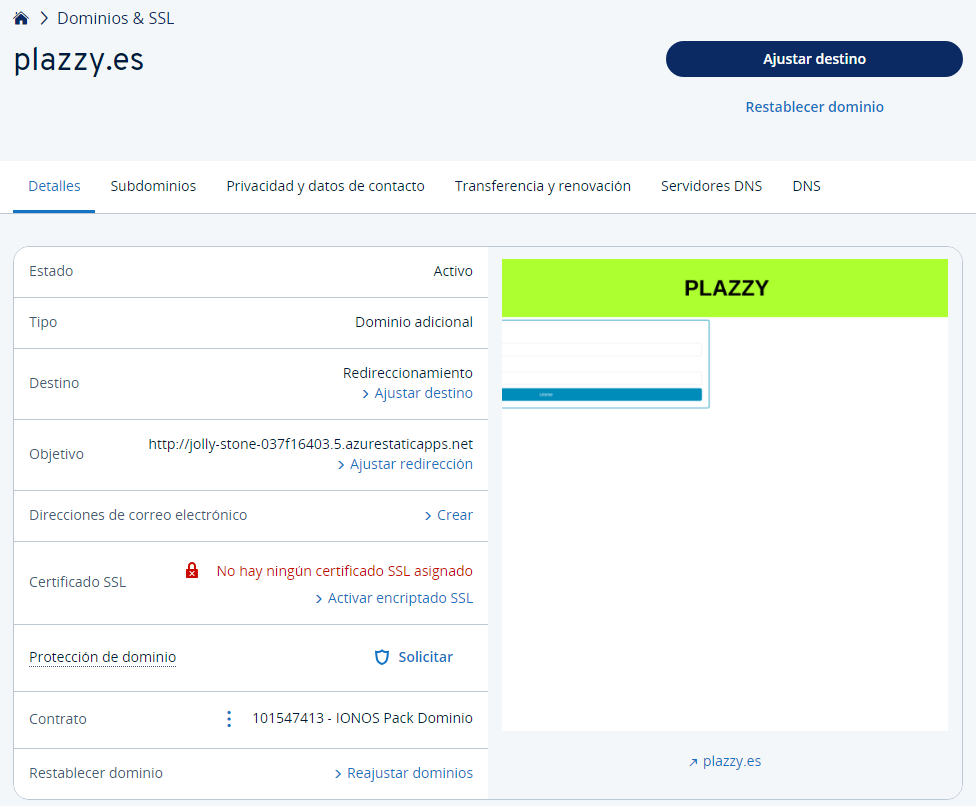


Figura 5: Configuración del dominio comprado en IONOS.es

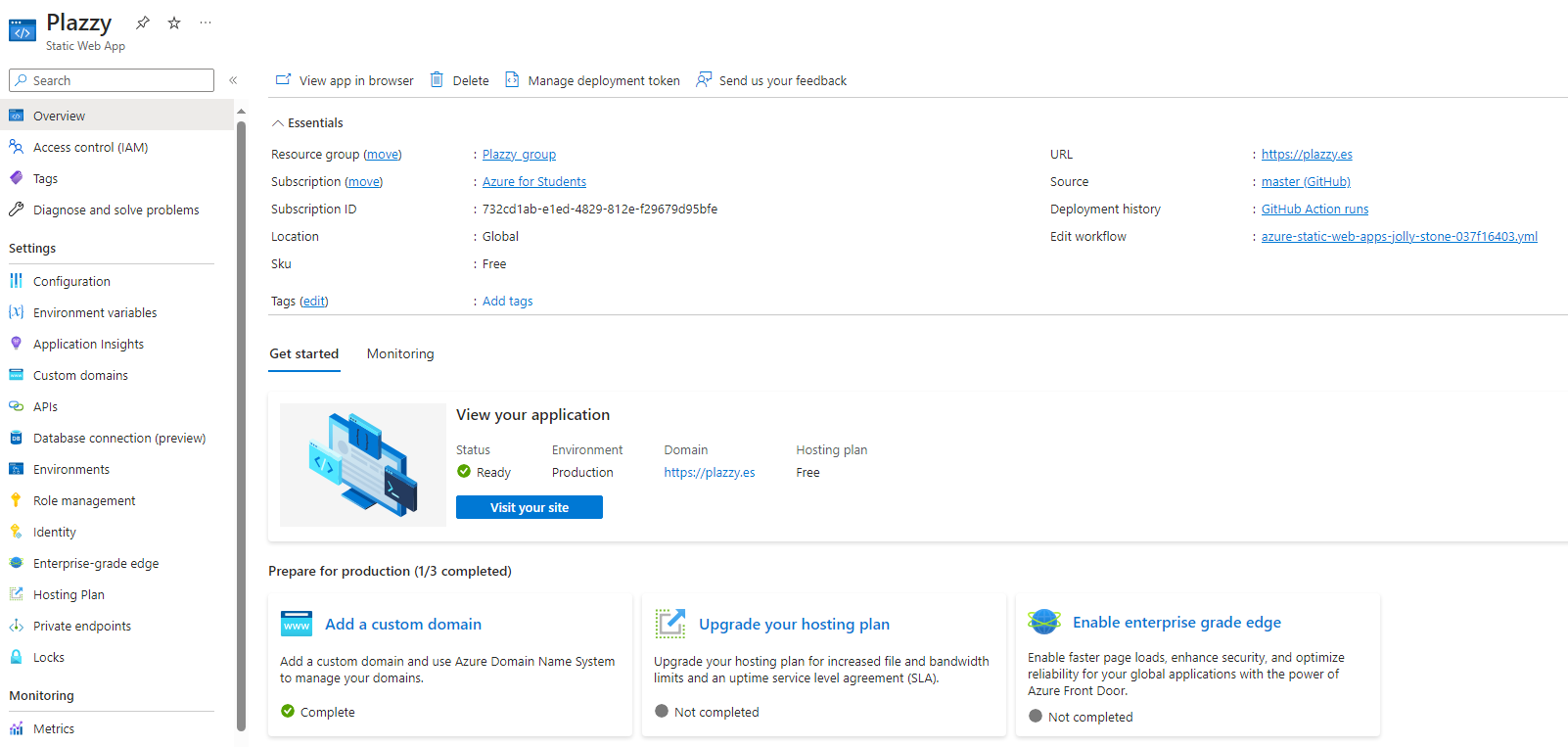


Figura 6: Configuración Aplicación Web Estática de Azure

Como opción de hosting en *Azure*, hemos utilizado *Static Web App*. Ya que permite generar una aplicación web a partir de un repositorio de *GitHub*.

El proceso que usa es detectar con que tecnología se emplea, en nuestro caso *React*, y crea una máquina virtual configurando los entornos y paquetes necesarios para que nuestro programa se ejecute en la web.

En el entorno de desarrollo usaremos una máquina virtual con el servicio *XAMPP* para que nos suministre una base de datos para nuestra aplicación

Para el servidor de base de datos de producción también emplearemos el portal *Azure*, que nos proporciona un servicio de bases de datos con *MySQL* vinculada con nuestra *Web App* y se configura fácilmente.

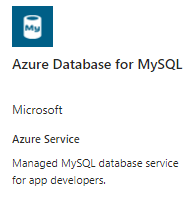


Figura 7: Recurso de Azure Database for MySQL

El proceso de configuración para la conexión desde la aplicación es relativamente fácil gracias al enfoque que hemos creado.

Tan solo hay que editar el fichero *.env* los campos relevantes a la base de datos.

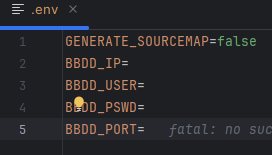


Figura 8: Plantilla fichero .env

1. Conclusiones

Conclusiones sobre el trabajo realizado

El desarrollo de Plazzy ha sido un proyecto integral que abarca desde la conceptualización de una plataforma de juegos sociales hasta su implementación y despliegue. Durante el proceso, se han implementado tecnologías modernas y eficientes, como *NodeJS* para el *backend*, React para el *frontend*, y *MySQL* para la gestión de bases de datos, garantizando una experiencia de usuario fluida y segura. La utilización de *Socket.IO* y *Three.js* añade un nivel avanzado de interactividad y gráficos 3D, enriqueciendo la experiencia del usuario. El despliegue en *Azure* proporciona una infraestructura escalable y confiable, asegurando la disponibilidad y rendimiento del portal. En resumen, *Plazzy* se ha desarrollado como una plataforma robusta y versátil, lista para expandirse con nuevos modos de juego en el futuro.

Conclusiones personales

Este proyecto era algo que queríamos intentar desde hacer un tiempo, hemos aprendido mucho usando las tecnologías de *ThreeJS* y *Socket.IO*. Además de poder implementar una *API* simple usando *NodeJS* con una conexión a un servidor de *MySQL*.

En cuanto al tiempo de desarrollo esperábamos poder tener dos juegos listos para el plazo, lo cual no ha podido ser por causas relacionadas a tener que cambiar de tecnologías o *API* y errores de código e integridad del proyecto.

Otra pega que hemos tenido es a la hora de alojar nuestra página web usando Azure. Cierto que usando aplicaciones web estáticas el despliegue es bastante simple, solo hay que conectar un repositorio de *GitHub* y *Azure* detecta que tecnología utiliza para construir el servidor web, carece de personalización en el servidor (como ha sido el caso de no poder conectarse mediante SSH y no poder integrar correctamente el dominio comprado desde *IONOS* como no poder usar el certificado *SSL* proporcionado por el gestor de dominios).

Posibles ampliaciones y mejoras

La primera ampliación obvia es agregar más modos de juego, también deberíamos mejorar la personalización de los usuarios, permitiendo elegir el icono que quieran además de incluir una mayor elección de iconos para no forzar a un usuario a usar el ultimo restante.

Otra ampliación interesante seria implementarlo con redes sociales, como *Twitch*, *X* (previamente conocido como *Twitter*) e *Instagram*.

1. Bibliografía

<https://dafo.ipyme.org/dafos#&&q=mis-dafos>

<https://prevencion-riesgoslaborales.com/riesgos-laborales-de-un-desarrollador-web#Riesgos_laborales_de_un_desarrollador_web>

<https://threejs.org/manual/>

<https://www.react-spring.dev/docs/getting-started>