



# **UNIVERSIDAD TECNOLÓGICA DE HERMOSILLO**

**INGENIERÍA EN DESARROLLO Y GESTIÓN DE  
SOFTWARE**

**GRUPO: IDGS10-1**

**ASIGNATURA: CREACIÓN DE VIDEOJUEGOS**

**DOCUMENTACIÓN DEL VIDEOJUEGO  
“100 DEVS DIJERON”**

**FECHA: 13/11/2025**

# 1. INFORMACIÓN GENERAL

## 1.1. Título del Proyecto

- **Nombre tentativo:** *100 Programadores Dijeron*
- **Nombre final:** *1100100 Devs Dijeron*
- **Nombre del estudio / desarrollador:** *IDGS 10-1 Studios*
- **Versión del documento:** v1.0.0 (Versionado semántico inicial)
- **Fecha de creación / actualización:** 13 / Noviembre / 2025
- **Responsables:**
  - **Director de juego:** Julio Cesar Vasquez Alvarez
  - **Programador líder:** Julio Cesar Vasquez Alvarez
  - **Programadores:** Julio Cesar Vasquez Alvarez, Andrik Gerardo Flores Murrieta, Ivan Andrey Garcia Saavedra
  - **Diseñadores de niveles:** Julio Cesar Vasquez Alvarez
  - **Responsables de arte y sonido:** Andrik Gerardo Flores Murrieta (UI/UX y diseño visual), Ivan Andrey Garcia Saavedra (música y efectos de sonido)
  - **Responsable(s) de pruebas:** Equipo de Testers de IDGS 101 Studios (Yordi Monreal, Victor Rivera, Juan Buzane, Juan Ruacho, Roberto Barrios, Manuel Miranda)
  - **Documentación:** Yordi Azael Monreal Zazueta, Victor Francisco Rivera Soto, Juan Carlos Buzane Leon
  - **Manual de usuario:** Roberto Barrios, Manuel Miranda.

## 1.2. Resumen del Juego

“Un juego de trivia competitivo donde los desarrolladores deben adivinar las respuestas más populares a preguntas sobre programación, cultura del software y/o cultura general.”

## 1.3. Género y Subgéneros

- **Género principal:** *Trivia / Party Game*
- **Subgéneros:** *Multijugador local, Educativo / Temático (Desarrollo de software)*

## 1.4. Plataforma

- **Plataforma objetivo:** *Web (navegador)*
- Compatible con PC, laptop, y pantallas táctiles (tablets).
- Tecnologías: *Laravel, MySQL*

## 1.5. Público Objetivo

- **Edad recomendada:** +12 (contenido sin violencia, lenguaje técnico y humor para adultos jóvenes)
- **Tipo de jugador:** Casual y competitivo — ideal para entornos de oficina, clases o eventos de tecnología.
- **Perfil psicológico / demográfico:**

Jugadores con interés en la programación, desarrollo web o cultura geek, que disfrutan de retos rápidos, humor técnico y dinámicas de grupo tipo “quiz show”.

## 1.6. Elevator Pitch

“Un divertido juego de trivia donde los devs compiten por adivinar las respuestas más populares sobre programación, bugs y la vida en el código. ¡Al estilo ‘100 mexicanos dijeron’, pero con bits y café!”

## 2. CONCEPTO Y VISIÓN CREATIVA

### 2.1. Tema Central

**“Colaboración, humor y conocimiento compartido.”**

El juego busca transmitir el espíritu de comunidad entre desarrolladores, destacando cómo la programación puede ser divertida, impredecible y llena de momentos sociales.

### 2.2. Historia Breve

En “*1100100 Devs Dijeron*”, los jugadores asumen el papel de participantes en un programa de trivia digital donde se enfrentan para descubrir qué opinó la comunidad de desarrolladores sobre diversos temas. A través de rondas rápidas y respuestas ingeniosas, el jugador busca acumular puntos y demostrar que conoce tanto la lógica del código como la mente de los devs.

### 2.3. Sinopsis Detallada

En un mundo donde los programadores han alcanzado fama global, un nuevo tipo de competencia se ha vuelto viral: “**1100100 Devs Dijeron**”, el concurso que pone a prueba el conocimiento técnico y cultural de los desarrolladores.

Cada partida comienza con una serie de preguntas que fueron respondidas previamente por una comunidad de “devs promedio”.

Los jugadores deben adivinar las respuestas más populares según esa comunidad, no necesariamente las más correctas.

A medida que avanzan las rondas, las preguntas se vuelven más específicas y temáticas: frameworks, bugs icónicos, frases típicas de los programadores o situaciones de oficina.

En la ronda final, los mejores jugadores se enfrentan en un “Juego rápido”, donde deben responder con rapidez para coronarse como el *Dev Supremo del código*.

El tono del juego es humorístico y autocrítico, con referencias a la vida del programador, memes y anécdotas que sólo un verdadero dev entendería.

## 2.4. Ambientación y Mundo

- **Época:** Presente, en un entorno digital contemporáneo.
- **Lugar:** Mundo virtual basado en un estudio de desarrollo de software gamificado. Las interfaces y escenarios simulan un estilo tecnológico.
- **Estilo visual:** Minimalista con inspiración *retro-futurista*. Uso de colores oscuros con acentos neón (azules, verdes, púrpuras). Tipografía tipo consola.
- **Cultura e influencias:** Cultura geek y del desarrollo de software, con referencias a memes, frameworks, lenguaje de programación y clichés del entorno laboral tech. Influencias visuales: *100 mexicanos dijeron*, *Hacknet*, *Tron Legacy*.

## 2.5. Personajes

Campo	Descripción
Rol / Arquetipo	Anfitrión / Persona que maneja el controlador del tablero
Motivación	Mantener el orden del concurso
Habilidades	Controla el flujo de juego, válida respuestas y muestra estadísticas
Relación con otros personajes	Con tono humorístico hacia los jugadores

Campo	Descripción
Rol / Arquetipo	Competidores / Personas que responden a las preguntas del tablero
Motivación	Demostrar su conocimiento sobre el mundo del desarrollo y ganar reconocimiento
Habilidades	Razonamiento rápido y conocimiento técnico
Relación con otros personajes	Equipo con otros jugadores y rivalidad con otros equipos

### **3. MECÁNICAS DE JUEGO**

#### **3.1. Loop de Juego Principal**

El ciclo principal del jugador es el siguiente:

**Recibir Pregunta → Responder / Adivinar → Evaluación del Anfitrión →  
Acumulación de puntos → Avanzar a la siguiente ronda.**

Durante cada partida, los jugadores deben adivinar las respuestas más populares dadas por una comunidad de desarrolladores. El ciclo se repite con nuevas preguntas hasta llegar a la ronda final, donde se define el ganador con base en la puntuación total.

**Loop resumido:**

- Leer pregunta
- Dar respuesta verbal o escrita (según modo)
- HostBot evalúa (correcta/incorrecta)
- Recibir retroalimentación visual y sonora
- Sumar puntos y continuar con la siguiente pregunta.

#### **3.2. Controles**

El sistema de juego se maneja **completamente a través de botones e íconos interactivos en pantalla**, sin necesidad de atajos de teclado ni combinaciones especiales.

Todas las acciones —desde iniciar una ronda hasta validar una respuesta o mostrar resultados— se ejecutan mediante **interacción directa con la interfaz gráfica**.

#### **3.3. Sistema de Progresión**

**Puntuación:**

Cada respuesta correcta suma puntos proporcionales a la popularidad de la respuesta (más votos = más puntos).

**Rondas:**

El juego está dividido en **rondas normales y ronda final (Batalla de Bits)**, donde los puntos se duplican.

**Recompensas (opcional / futuro):**

EL reconocimiento de ganar la competición.

### 3.4. Interacción

#### Interacción:

- El *Controlador* (Game Master) selecciona si la respuesta del jugador es válida o no, basándose en criterio y coincidencia con las respuestas populares.
- Los jugadores pueden debatir, bromear o justificar sus respuestas, fomentando el componente social del juego.

#### Decisiones:

Cada ronda presenta la posibilidad de aceptar una respuesta parcialmente correcta, lo que añade dinamismo y discusión.

#### Penalización:

Tres errores consecutivos eliminan al jugador de la ronda, pero pueden reincorporarse en la siguiente.

## 3.7. HUD / Interfaz de Usuario

#### Componentes principales del HUD:

- **Zona central:** Pregunta actual.
- **Paneles laterales:** Respuestas de cada equipo.
- **Zona superior:** Puntaje general de la ronda.

#### Estilo visual:

- Tipografía tipo consola
- Colores oscuros con colores llamativos para botones o títulos.

#### Animaciones de UI:

- Transiciones entre rondas.
- Mostrar una 'x' cuando la respuesta sea incorrecta o el tiempo acabe.
- Animación al mostrar una respuesta correcta en el tablero de respuestas con la respuesta proporcionada.

## 4. DISEÑO DE NIVELES Y ENTORNOS

### 4.1. Filosofía de Diseño

El diseño del juego se basa en ofrecer una experiencia interactiva, accesible y divertida para todo tipo de público, en especial estudiantes de la carrera. La filosofía del proyecto fue mantener la simplicidad visual y funcional, priorizando la fluidez del juego y la facilidad de comprensión de las reglas.

El objetivo principal es recrear el ambiente competitivo y dinámico del programa “100 mexicanos Dijeron”, pero adaptado al entorno web, permitiendo que varios usuarios participen desde sus dispositivos.

La interfaz se diseñó con colores vivos, íconos representativos y animaciones ligeras que aportan dinamismo sin distraer del juego principal.

### 4.2. Estructura del Mundo

El “mundo” del juego está dividido en dos secciones principales, que representan las etapas del concurso:

#### 1. Rondas de Preguntas (Fase 1):

Consta de tres rondas donde los equipos compiten respondiendo preguntas con múltiples posibles respuestas. Cada ronda tiene un nivel de dificultad incremental y un límite de tiempo.

#### 2. Dinero Rápido (Fase 2):

Es la fase final del juego, donde el equipo ganador responde cinco preguntas rápidas con un límite de tiempo más estricto. El puntaje acumulado define si logran el “ premio virtual”.

Nivel / Ronda	Descripción	Objetivo del Jugador	Elementos Clave
Ronda 1 – Calentamiento	Preguntas básicas de cultura general o programación sencilla.	Responder rápido para acumular puntos iniciales.	Preguntas con 3–4 respuestas posibles, temporizador de 20 s.
Ronda 2 – Intermedia	Preguntas con mayor dificultad o que	Aumentar puntaje y ganar ventaja	Efectos sonoros al acertar o fallar,

	requieren pensamiento lógico.	sobre el otro equipo.	transiciones dinámicas.
Ronda 3 – Reto Final	Preguntas más elaboradas o de doble puntaje.	Asegurar la victoria antes de pasar a Dinero Rápido.	Música más intensa, interfaz resaltada.
Fase Final – Dinero Rápido	Cinco preguntas consecutivas con límite total de tiempo.	Acumular la mayor cantidad de puntos para ganar.	Contador visible, fondo animado, música de tensión.

El juego se desarrolla en un entorno web con transiciones entre pantallas que simulan la progresión del concurso, y mantiene coherencia visual en todos sus elementos.

## 5. DISEÑO ARTÍSTICO (ADD – Art Design Document)

## **5.1. Dirección de Arte**

El estilo visual del juego 1100100 Devs Dijeron está inspirado en una estética cibernetica y minimalista, con predominio de tonos oscuros y acentos de luz neón en colores azul turquesa y naranja.

El objetivo fue crear una atmósfera moderna, que evocara la temática tecnológica y al mismo tiempo mantuviera la sensación de estar en un concurso digital.

La dirección de arte buscó equilibrar claridad visual con dinamismo, utilizando tipografías claras, transiciones suaves y efectos brillantes que refuerzan la sensación de estar dentro de un sistema informático o consola interactiva.

## **5.2. Personajes**

El juego no cuenta con personajes animados tradicionales, sino con representaciones abstractas de los equipos, que aparecen como cuadros digitales con nombres personalizables (por ejemplo, *Equipo Dinamita*, *Equipo Aleatorio*).

El “presentador” se representa mediante mensajes en pantalla y efectos visuales, manteniendo la atención en el tablero y en las preguntas.

Este enfoque refuerza la idea de una competencia automatizada y visualmente centrada en la acción del juego, no en avatares.

## **5.3. Entornos**

El entorno principal es una interfaz tipo dashboard futurista, con una composición en dos vistas:

- Vista del controlador: donde el moderador o administrador selecciona preguntas, define rondas y gestiona los puntos.
- Vista del tablero principal: donde los jugadores visualizan la pregunta activa, las respuestas y el marcador de ambos equipos.

El fondo mantiene un degradado oscuro con brillos suaves, que simula una pantalla holográfica. Los contenedores, botones y cuadros de texto tienen bordes luminosos con resplandor, destacando sobre el fondo sin saturar la pantalla.

## **5.4. UI / UX**

La interfaz se diseñó bajo los principios de claridad, inmediatez y control total del flujo de juego.

- La UI del moderador prioriza la funcionalidad, con botones grandes (“Enviar al tablero”, “Reset”, “Nueva partida”) y colores diferenciados para evitar errores durante la partida.

- La UI del jugador muestra la información esencial: puntos, ronda, multiplicador (x1, x2, x3), equipos y respuestas visibles.
- Los estados del juego (en juego, finalizado, etc.) se indican con texto y color, permitiendo identificar rápidamente el progreso.
- La tipografía monoespaciada y el uso de tonos cian neón evocan el ambiente de un sistema digital o “terminal de programación”, reforzando la identidad del proyecto.

La experiencia se diseñó para ser usable tanto en eventos presenciales (proyectado en pantalla) como en navegadores individuales, manteniendo la coherencia entre ambas vistas.

## 5.5. Animaciones

Las animaciones son sutiles y funcionales, buscando dar fluidez al juego sin distraer:

- Efecto de resplandor en botones al pasar el cursor.
- Transiciones suaves entre rondas o fases.
- Animaciones de puntos y multiplicadores con destellos cuando se activan.
- Efecto de “error” (X roja) al fallar una respuesta, acompañado de un parpadeo visual.

El estilo se mantiene coherente con la temática digital, imitando pulsos de luz o señales electrónicas.

## 5.6. Cinemáticas

No se emplearon cinemáticas tradicionales; sin embargo, el juego incluye secuencias visuales automáticas que funcionan como introducciones o cierres de fase:

- Inicio de partida: encendido progresivo de los elementos de la interfaz.
- Transición al modo Dinero Rápido: cambio de iluminación y activación de un fondo animado con tonos más intensos.
- Final de partida: efectos de luz y sonido que simulan una “celebración digital” (confeti animado, puntuaciones parpadeantes).

# 6. SONIDO Y MÚSICA

## 6.1. Dirección Sonora

El sonido se diseñó para reforzar la emoción del concurso, acompañando cada acción del jugador.

La idea fue mantener un equilibrio entre lo dinámico y lo divertido, sin sobrecargar el entorno.

Se utilizaron efectos cortos y música de fondo en bucle, simulando un show televisivo.

## **6.2. Efectos de Sonido**

- Sonido de campana al acertar una respuesta.
- Bocina o zumbido al fallar.
- Transición tipo “quién quiere ser millonario” al pasar de ronda.
- Efecto de conteo en la fase de Dinero Rápido.
- Confeti sonoro al ganar.

## **6.3. Música**

La banda sonora principal utiliza música instrumental rítmica con sintetizadores, inspirada en programas de concurso.

Cada fase tiene una variante:

- Rondas normales: Música alegre y ligera.
- Dinero Rápido: Tonos más tensos y acelerados.
- Pantalla de victoria: Tema triunfal corto, con percusión y fanfarria.

## **6.4. Voz**

No se grabaron voces reales, pero se incluyen efectos de sonido para acompañar momentos.

# **7. DISEÑO TÉCNICO (TDD – Technical Design Document)**

## **7.1. Motor del Juego**

El juego está desarrollado sobre el **framework Laravel (PHP)**, que actúa como el motor principal de la aplicación. Laravel gestiona la lógica de juego, las rondas, el cálculo de puntajes y la comunicación entre el servidor y el cliente.

El motor del juego se encarga de:

- Controlar las rondas y fases del juego (pregunta, respuestas, puntaje, siguiente ronda).
- Administrar el modo clásico y el modo rápido.
- Validar respuestas y calcular el puntaje automáticamente.
- Controlar los tiempos y respuestas del modo rápido.

## 7.2. Lenguajes y Frameworks

- **Backend:** PHP (Laravel 10).
- **Frontend:** Blade (HTML, CSS, JavaScript con Bootstrap)
- **Base de datos:** MySQL / MariaDB.
- **Servidor local:** WAMP

## 7.3. Arquitectura del Proyecto

El proyecto sigue el patrón **MVC (Modelo–Vista–Controlador)** típico de Laravel:

- **Modelos:** Representan las entidades principales como Question, Answer, Game, Player y Score.
- **Controladores:** Gestionan la lógica del juego, la validación de respuestas y la comunicación entre el usuario y el sistema.
- **Vistas:** Implementadas con Blade, renderizan la interfaz del juego y muestran los puntajes, preguntas y resultados.
- **Rutas:** Definidas en web.php para manejar las solicitudes del jugador y los modos de juego.

## 7.4. Herramientas de Desarrollo

- **IDE:** Visual Studio Code.
- **Control de versiones:** Git y GitHub.
- **Servidor local:** WAMP.
- **Diseño visual:** Figma / Canva para prototipos de interfaz.
- **Pruebas:** Laravel Artisan (php artisan serve, php artisan test).

## 7.5. Requerimientos Técnicos

### Hardware mínimo:

- Procesador dual-core o superior.
- 4 GB de RAM.
- 500 MB de espacio libre.

### Software:

- PHP 8.1 o superior.
- Composer instalado.
- MySQL 5.7 o superior.
- Navegador actualizado (Chrome, Edge, Firefox).

### Dependencias principales:

- Laravel Framework.
- Bootstrap.

## 7.6. Sistema de Guardado

El sistema de guardado utiliza una base de datos **MySQL**, donde se registran:

- Preguntas y respuestas (tabla questions y answers).
- Partidas jugadas y puntajes (games, scores).
- Estadísticas del modo rápido y clásico.  
Los datos se almacenan mediante modelos Eloquent, lo que permite una gestión sencilla y relacional.

## 7.7. Red / Online (si aplica)

El juego puede ejecutarse en modo local o modo en línea mediante un servidor HTTP (Apache).

En el modo online, varios jugadores pueden conectarse a la misma sesión mediante enlaces compartidos o un sistema de turnos gestionado por el backend.

Laravel gestiona las peticiones vía HTTP y, opcionalmente, puede integrarse con Laravel WebSockets o Pusher para comunicación en tiempo real.

## 8. PRODUCCIÓN Y GESTIÓN (PMD – Project Management Document)

### 8.1. Cronograma General

Fase	Descripción	Duración	Estado
Análisis	Diseño de reglas y lógica del juego	1 semana	Completado
Desarrollo Backend	Implementación de Laravel y base de datos	2 semanas	Completado
Desarrollo Frontend	Interfaz del juego con Blade	1 semana	Completado
Modo rápido	Implementación y prueba del modo rápido	1 semana	Completado
Pruebas y correcciones	Ajustes y mejora de rendimiento	1 semana	En curso

### 8.2. Hitos (Milestones)

- **M1:** Configuración del entorno Laravel.
- **M2:** Base de datos funcional con preguntas y respuestas.
- **M3:** Lógica del juego clásico implementada.
- **M4:** Lógica del modo rápido completada.
- **M5:** Interfaz final y pruebas completas.
- **M6:** Publicación del proyecto.

### 8.3. Presupuesto

Concepto	Descripción	Costo estimado
Hosting y dominio	Para la versión en línea	\$20 USD/año
Software	Gratis (Laravel, VS Code, WAMP)	\$0
Material gráfico y diseño	Iconos, fondos, tipografía	\$10
Total estimado		<b>\$30 USD</b>

### 8.4. Riesgos

Riesgo	Probabilidad	Impacto	Mitigación
Fallos en la base de datos	Media	Alta	Backups automáticos
Lentitud del servidor	Media	Media	Optimizar consultas y caché
Errores en lógica del modo rápido	Alta	Media	Pruebas unitarias y de integración

### 8.5. Marketing y Publicación

El juego se promoverá principalmente de manera digital:

- **Redes sociales:** Facebook, Instagram y TikTok.
- **Sitio web:** Publicación en hosting gratuito (por ejemplo, 000webhost o Render).
- **Promoción:** Mostrar clips del juego, trivias y retos para atraer usuarios.
- **Objetivo:** Atraer público casual interesado en juegos de preguntas tipo “Family Feud” con temática mexicana.

## 9. ANEXOS

### 9.1. Glosario de Términos

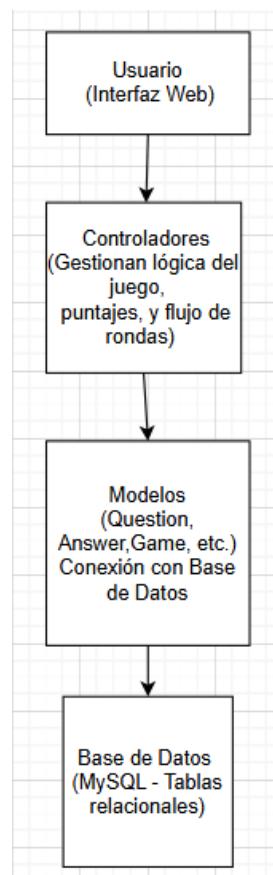
Término	Definición
<b>Laravel</b>	Framework de desarrollo web en PHP que facilita la creación de aplicaciones estructuradas con el patrón MVC.
<b>MVC (Modelo–Vista–Controlador)</b>	Patrón de arquitectura de software que separa la lógica de negocio, la interfaz y el control de flujo.
<b>Backend</b>	Parte del sistema que gestiona la lógica, datos y comunicación con la base de datos.
<b>Frontend</b>	Parte visual e interactiva del sistema que ve el usuario final.
<b>Blade</b>	Motor de plantillas nativo de Laravel utilizado para generar vistas dinámicas en HTML.
<b>Base de datos relacional</b>	Sistema que organiza los datos en tablas con relaciones entre ellas (por ejemplo, MySQL).
<b>Eloquent ORM</b>	Herramienta de Laravel que facilita la interacción con la base de datos mediante modelos.
<b>Modo rápido</b>	Modo de juego adicional en el que los jugadores deben responder más preguntas en menos tiempo.
<b>API</b>	Conjunto de funciones que permiten la comunicación entre distintas partes del sistema o con servicios externos.
<b>WebSocket</b>	Protocolo que permite comunicación en tiempo real entre el cliente y el servidor.
<b>Puntaje</b>	Valor obtenido por el jugador según la cantidad de respuestas correctas.
<b>Ronda</b>	Fase o etapa del juego que contiene una pregunta y sus posibles respuestas.

## 9.2. Referencias Externas

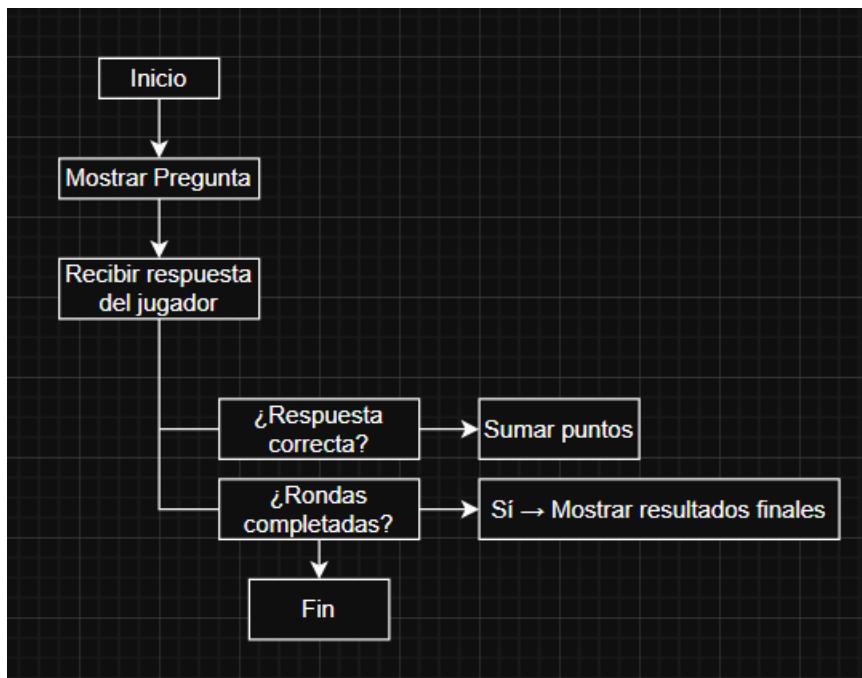
- **Documentación oficial de Laravel:**  
<https://laravel.com/docs>
- **Bootstrap (Framework CSS):**  
<https://getbootstrap.com>
- **MySQL Reference Manual:**  
<https://dev.mysql.com/doc>
- **Guía de diseño de interfaces en Figma:**  
<https://www.figma.com/resources/learn-design/>
- **Inspiración del formato del juego:**  
Basado en el programa de televisión “100 Mexicanos Dijeron”, producido por Televisa (referencia cultural para la mecánica de preguntas y respuestas).

## 9.3. Diagramas y Esquemas

### a) Diagrama de Arquitectura General del Proyecto (MVC)



## b) Diagrama de Flujo del Juego (Modo Clásico)



## 9.4. Documentos Relacionados

Documento	Descripción	Versión / Fecha
<b>TDD – Technical Design Document</b>	Documento que detalla el diseño técnico, herramientas, arquitectura y sistema de guardado del juego.	v1.0 / Nov 2025
<b>PMD – Project Management Document</b>	Documento que describe la gestión del proyecto, cronograma, riesgos y presupuesto.	v1.0 / Nov 2025
<b>Manual de Usuario (opcional)</b>	Documento guía para jugadores y administradores sobre el uso del juego y sus modos.	En desarrollo
<b>Base de datos (exportación .sql)</b>	Archivo con la estructura de tablas y relaciones del proyecto.	Última versión disponible

## **10. REFERENCIAS BIBLIOGRÁFICAS**

### **10.1. Libros**

- Fowler, M. (2003). *Patterns of enterprise application architecture*. Addison-Wesley Professional.
- Grinberg, M. (2018). *Flask web development: Developing web applications with Python (2nd ed.)*. O'Reilly Media.
- Lerdorf, R., Tatroe, K., & MacIntyre, P. (2013). *Programming PHP (3rd ed.)*. O'Reilly Media.
- Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico (7<sup>a</sup> ed.)*. McGraw-Hill.

### **10.2. Artículos académicos**

- González, L., & Pérez, J. (2020). Diseño de sistemas interactivos educativos basados en mecánicas de juego. *Universidad Autónoma de México*.
- Ramírez, C., & Torres, A. (2019). Aplicaciones web interactivas como herramienta de aprendizaje colaborativo. *Revista Iberoamericana de Tecnología Educativa*, 15(2), 45–58.

### **10.3. Sitios web y documentación técnica**

- Bootstrap. (2025). *Bootstrap documentation*. Recuperado de <https://getbootstrap.com/docs/>
- GitHub. (2025). *Documentación oficial de GitHub*. Recuperado de <https://docs.github.com/>
- Laravel. (2025). *Laravel documentation*. Recuperado de <https://laravel.com/docs>
- MySQL. (2025). *MySQL documentation*. Recuperado de <https://dev.mysql.com/doc/>
- PHP. (2025). *Manual de PHP*. Recuperado de <https://www.php.net/manual/es/>
- Tailwind CSS. (2025). *Documentación oficial*. Recuperado de <https://tailwindcss.com/docs>

#### **10.4. Videos / Entrevistas**

IEEE Software Channel. (2022). *Entrevista a Steve McConnell sobre gestión de proyectos de software* [Video]. YouTube.

Traversy Media. (2024). *Laravel from scratch* [Video]. YouTube.

<https://www.youtube.com/watch?v=ImtZ5yENzgE>

Dev en Español. (2023). *Cómo hacer un juego tipo “100 Mexicanos Dijeron” en la web* [Video]. YouTube.

#### **10.5. Imágenes y concept art**

Canva. (2025). *Diseño del logo del proyecto*. Recuperado de

<https://www.canva.com/>

Flaticon. (2025). *Iconos e ilustraciones libres*. Recuperado de

<https://www.flaticon.com/>

Freepik. (2025). *Fondos e imágenes libres para uso educativo*. Recuperado de

<https://www.freepik.com/>

Capturas de pantalla y esquemas generados por el autor. (2025). *Juego “100 Mexicanos Dijeron” – Proyecto en Laravel*.