

EduKids

¡En busca de la excelencia!

1º. Desarrollo de Aplicaciones Web

Equipo D

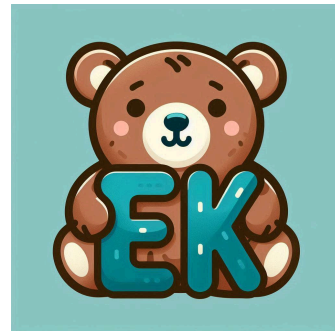
Bryan Arana Herrera

Erick Guzmán Cruz

Imanol Pérez

Julen López Baldeon

17/05/2024 San Sebastián-Donostia



Índice

Introducción.....	2
EduKids.....	3
Entornos de desarrollo.....	6
Identificación de recursos (Planificación).....	6
Diagrama UML.....	18
Bases de datos.....	21
Lenguaje de marcas.....	24
Sistemas informáticos.....	30
Programación.....	32
Formación y orientación laboral (FOL).....	35
Implementaciones futuras.....	36
Problemas encontrados y soluciones para ellas.....	37
Conclusión.....	38
Bibliografía.....	39

Introducción

El presente informe detalla el proceso de desarrollo sobre nuestro proyecto orientado a la creación de un videojuego, desde su concepción inicial hasta su implementación final. A lo largo del informe, se explorarán las diferentes fases del proyecto, incluido el análisis de requisitos, el diseño, el desarrollo, las pruebas y el lanzamiento.

Se explicará cada una de las materias a desarrollar para que su creación sea de manera eficiente y óptima, también, la estructura, la mecánica del juego, las decisiones clave de diseño y la tecnología utilizada para implementar el juego. Además, su desarrollo paso a paso, se presentan en detalle los desafíos enfrentados durante el desarrollo y las soluciones adoptadas para superarlos.

El objetivo principal de este informe es proporcionar una descripción general sobre el desarrollo de nuestro proyecto, y proporcionar una valiosa herramienta de documentación para proyectos futuros. Sin más preámbulos, aquí le presentamos una introducción a la fundación de la empresa, nuestros objetivos y otros aspectos importantes del desarrollo del proyecto.

EduKids

Introducción

Nos complace anunciar el lanzamiento de EduKids, una nueva plataforma en línea diseñada para mejorar el aprendizaje y el rendimiento infantil. EduKids es una fuente de conocimiento, ofreciendo una amplia variedad de preguntas de diferentes categorías.

Esperamos que EduKids pueda aportar una visión cada vez más clara a nuestra comunidad educativa y al desarrollo de aprendizaje de los niños. Pronto, exploremos el mundo de EduKids y veamos cómo mejora el rendimiento y aprendizaje infantil en nuestros hijos.

¿Qué es EduKids?, ¿Cuándo nace la idea?, ¿Para qué sirve?

Es una aplicación de juegos infantiles para fomentar o mejorar el aprendizaje de los niños de una manera entretenida. Especialmente está hecha para los niños, pero también diferentes personas la podrán utilizar tranquilamente, ya que, tendrá diferentes categorías de aprendizaje, desde matemáticas hasta deportes.

Nace en la necesidad de brindar una educación para niños de una manera divertida y llamativa, para que así, los niños no se aburran y les guste aprender.

Para qué los niños jueguen y a la vez aprendan, ya que, es un método de aprendizaje muy efectivo. Los padres y los centros educativos podrán monitorear el aprendizaje de los niños, por lo tanto, los centros educativos tendrán esta aplicación como una opción de aprendizaje.

¿Cuáles son los objetivos principales de EduKids?

Educación Infantil: proporcionar acceso universal a la educación infantil. Queremos eliminar las barreras geográficas y económicas para que cualquier persona, en cualquier parte del mundo, pueda acceder a recursos educativos.

Innovación: nos mantendremos firmes en el desarrollo de la innovación educativa en el ámbito infantil. Esto implica ofrecer contenido actualizado y relevante, así como utilizar tecnologías emergentes para mejorar la experiencia de aprendizaje de nuestros usuarios.

Promover el aprendizaje: valoramos el aprendizaje de los niños ya que son el futuro de nuestro mundo y todos deben tener una educación adecuada. Por lo tanto, nos esforzamos por ofrecer contenido que abarque una amplia variedad de materias educativas y formas de aprendizaje.

¿Qué tiene nuestra plataforma en línea?

EduKids, ofrece una amplia gama de características y herramientas diseñadas para brindar una experiencia educativa completa y enriquecedora. Estos son los campos que se muestran en nuestra plataforma web.

Inicio: es nuestra página principal donde se mostrará y se explicara lo que es nuestra plataforma y en qué consiste.

Contenido del juego: en este apartado está la información del contenido de la plataforma donde se mostrará y se explicará las categorías de las preguntas que habrá en ella, geografía, deportes, animales, matemática e historia.

Juego: en este apartado se podrá descargar nuestro videojuego de manera gratuita, en Windows o en Linux, tanto en ordenadores, Tablet y móviles.

Acerca de nosotros: Es el apartado donde está la información de nuestra empresa y de cada integrante de la plataforma, también, habrá un formulario para que las personas que tengan dudas puedan preguntar tranquilamente.

¿Qué tendrá nuestro juego?

Nuestro juego tendrá diferentes apartados, es un juego de preguntas y respuestas, pero se diferencia de otros juegos ya que tendrá categorías tipo: **historia, deporte, matemáticas, geografía entre otras**. Además, las categorías también estarán divididas por nivel de dificultad por lo tanto tendremos nivel: **fácil, medio, alto**. Por otro lado, la estructura de nuestro juego tendrá diferentes interacciones, desde elegir opciones para la respuesta correcta, cómo escribir la respuesta correcta y también mostrar imágenes en las que van a tener que buscar objetos y señalar la respuesta correcta.

Para empezar a todos nuestros usuarios al entrar al juego les aparecerá una pantalla donde tendrán que identificarse poniendo su usuario y contraseña. Después de identificarse, a los usuarios les aparecerá para seleccionar las categorías y dificultad que quieren para jugar, tienen que seleccionar y darle al botón de aceptar. (Implementaremos un programa de elección aleatoria de las categorías y dificultades, esto para que sea más divertido al no saber qué va a tocar).

Cuando seleccionen las categoría y dificultad podrán empezar a jugar, les aparecerá la pantalla en la que sale la pregunta con sus respuestas. Por un lado, si la pregunta es de seleccionar la respuesta correcta. Aparecen los botones con las opciones y la pregunta. Por

otro lado, si la pregunta es de escribir la respuesta, los botones quedan deshabilitados y aparece un *TextField* en el que pueden escribir la respuesta y al lado aparece un botón para verificar si es correcta o no.

¿Hasta dónde queréis llegar?

Alcance en línea: queremos llegar a los alumnos de todo el mundo, ofreciendo nuestra plataforma en línea en múltiples idiomas para garantizar que sea accesible para una audiencia global.

Alcance físico: aspiramos a establecer asociaciones con escuelas de todo el mundo, todo tipo de instituciones educativas en todo el mundo para ofrecer programas de educación en persona complementados por nuestra plataforma en línea.

Entornos de desarrollo

Introducción

Este apartado proporciona una descripción general sobre la planificación del proyecto y, además, los recursos utilizados para la creación de este, centrándose en la identificación de lenguajes de programación, *scripts* y herramientas especiales, una breve explicación sobre las bases de datos, tareas a realizar y un plan anti-riesgos, Además, proporciona una descripción general del diseño de *WordPress* y la gestión de equipos.

Identificación de recursos (Planificación)

Servidores, Plantilla Web

¿Dónde está alojado nuestro sitio web?

La base de datos está alojada en una instancia Amazon RDS (*Amazon Relational Database Service*) en la web de AWS (*Amazon Web Services*) en un servidor totalmente independiente y diferente del sitio web.

Para esta página web utilizamos una **plantilla** por defecto de las opciones que ofrece *WordPress*. Por lo tanto, nos centramos enteramente en modificar y optimizar la página ya seleccionada.

FrontEnd

En el proceso de modificación de nuestra página web, utilizamos los lenguajes de: HTML (Lenguaje de marcado de hipertexto o *HyperText Markup Language* por sus siglas en inglés), CSS (Abreviado en sus siglas en inglés, *Cascading Style Sheets*, que significa hojas de estilo en cascada) y *JavaScript* de una manera un poco sutil ya que *WordPress* nos ofrece de manera automática la utilización de estos lenguajes. Mejoraremos la plantilla para crear una experiencia fluida y atractiva para nuestros usuarios.

HTML: es la base de nuestra estructura. Se utiliza un poco de HTML para definir la estructura de nuestra página web, incluyendo encabezados, párrafos, listas, enlaces y otros elementos fundamentales. Esto nos permite organizar el contenido de manera semántica y accesible para nuestros visitantes y los motores de búsqueda.

CSS: juega un papel crucial en la apariencia visual de nuestra página web. Con la facilidad de diseño en *WordPress* y la utilización de su interfaz amigable, definimos el estilo, el diseño y la presentación de nuestros elementos HTML. Controlamos aspectos como colores, fuentes,

márgenes, alineaciones y mucho más, lo que nos permite crear una experiencia de usuario atractiva y coherente en diferentes dispositivos y tamaños de pantalla.

En resumen, *WordPress* nos brinda la estructura HTML, CSS de manera automática, esto nos ayuda a la interactividad. Juntos, estos lenguajes nos permitieron crear una página web atractiva, funcional y fácil de usar para nuestros usuarios.

Estructura básica de WordPress

Temas: los temas son la base visual de un sitio web de *WordPress*. Determinan cómo se ve y se siente el sitio, incluyendo el diseño, la tipografía, los colores y otros aspectos visuales [1].

Plantillas de página: las plantillas de página son archivos que determinan cómo se muestra el contenido en una página específica del sitio web [1].

Tipos de contenido personalizados: además de las páginas y las entradas predeterminadas, *WordPress* permite a los usuarios crear sus propios tipos de contenido personalizados. Esto significa que se puede definir y gestionar diferentes tipos de contenido [1].

Taxonomías: las taxonomías son una forma de clasificar y organizar el contenido en *WordPress* [1].

Menús: los menús en *WordPress* permiten a los usuarios crear y personalizar la navegación del sitio web [1].

Widgets: los *widgets* son pequeñas áreas de contenido que se pueden agregar y personalizar en diferentes ubicaciones del sitio web, como barras laterales o áreas de pie de página [1].

Plugins: los *plugins* son extensiones que se pueden añadir a *WordPress* para agregar nuevas funcionalidades o modificar las existentes [1].

BackEnd

PHP (*Hypertext Preprocessor*), JS (*JavaScript*): como se mencionó anteriormente, *WordPress* es fácil de usar y no requiere conocimientos muy avanzados en lenguajes de programación. *WordPress* utiliza principalmente PHP y *JavaScript* en el *backend*, pero esto se hace de forma automatizada y no requiere una intervención muy profunda.

Algunas Herramientas utilizadas en el *backend* de *WordPress*:

- **All-in-One WordPress Migration:** esta herramienta la utilizamos para la migración de todos los datos de nuestro sitio web. Importa o exporta el contenido de nuestro sitio creando a su vez copias de seguridad.

- ***Autooptimize***: esta herramienta la utilizamos para la optimización de todas las páginas en nuestro sitio web.
- ***Contact Form 7***: esta herramienta la hemos utilizado para la creación del formulario de contacto en el apartado de “acerca de nosotros” en nuestro sitio web.
- ***CookieYes | GDPR (Reglamento general de protección de datos)***: esta herramienta se utilizó para la implementación de *Cookies* en nuestro sitio web.
- ***Creative Commons***: herramienta utilizada para la asignación del *creative commons* de nuestro sitio web.
- ***hCaptcha for WordPress***: *hCaptcha* nos ayuda a implementar un sistema de identificación para las personas y para evitar el *spam* como el uso de *bots* dentro de nuestro sitio web.
- ***User Registration***: esta herramienta la utilizamos para la creación de nuestro formulario de registro e inicio de sesión dentro de nuestro sitio web.

Bases de datos

MySQL (*Structured Query Language* o Lenguaje de Consulta Estructurada).

Utilizaremos *SQL Server*. Ya que se construye principalmente en torno a una estructura de tablas basada en filas que conecta los elementos de datos relacionados en diferentes tablas entre sí.

Para nuestra base de datos orientada a la aplicación Java:

La base de datos de nuestro juego es muy sencilla ya que solo almacena los datos de las preguntas con sus respuestas y quienes han iniciado sesión, por lo tanto, son dos tablas independientes de cada una.

La tabla “preguntasJuego” tiene los datos de las preguntas, tales como: respuesta correcta, las respuestas incorrectas, su categoría, dificultad, etc. La categoría y la dificultad serán apartados importantes, ya que, nuestros usuarios cuando jueguen podrán elegir la categoría y su dificultad. La columna de tipo será utilizada para ver si la pregunta es de seleccionar la respuesta o es de escribir la respuesta, esto facilitará a la hora de programar el código del juego.

Para nuestra base de datos orientada a él *WordPress*:

Con la creación y registro de una cuenta en *WordPress* se crea automáticamente la base de datos del sitio web, pero nosotros hemos decidido tomar esa base de datos e importarla al

servidor de Amazon RDS. *WordPress* utiliza el motor MySQL para la gestión de bases de datos en el que se crean tablas que se utilizarán para el desarrollo de sitios web.

Estas son algunas de las tablas que se crean y una breve explicación de ellas:

- ***wp_posts***: esta tabla almacena todos los contenidos del sitio, como entradas de *blog* y páginas estáticas. Cada fila en esta tabla contiene información sobre un contenido específico, como el título, el contenido y la fecha de publicación [2].
- ***wp_comments***: esta tabla almacena todos los comentarios que se han realizado en el sitio. Cada fila en esta tabla contiene información sobre un comentario específico, como el contenido del comentario, el autor y la fecha de publicación [2].
- ***wp_users***: esta tabla almacena información sobre los usuarios del sitio, como nombres de usuario y contraseñas [2].
- ***wp_options***: esta tabla almacena configuraciones generales del sitio, como el título y la descripción. También se utiliza para almacenar opciones de *plugin* y tema [2].

Además de estas tablas básicas, *WordPress* también tiene varias otras tablas que se utilizan para almacenar información específica, como enlaces externos, categorías y etiquetas.

- ***wp_usermeta***: almacena metadatos adicionales sobre los usuarios del sitio [2].
- ***wp_commentmeta***: esta tabla almacena metadatos adicionales sobre los comentarios en el sitio. Por ejemplo, se puede utilizar para almacenar información sobre el correo electrónico del autor del comentario ó el estado de moderación [2].
- ***wp_links***: esta tabla se utiliza para almacenar enlaces externos que se hayan añadido a través del administrador de enlaces de *WordPress*. Cada fila en esta tabla contiene información sobre un enlace específico, como el título y la URL (*Uniform Resource Locator*) [2].
- ***wp_postmeta***: esta tabla almacena metadatos adicionales sobre los contenidos del sitio. Por ejemplo, se puede utilizar para almacenar información sobre la fecha de modificación de una entrada de *blog* o el autor de una página [2].
- ***wp_terms***: esta tabla almacena información sobre términos utilizados en el sitio, como categorías y etiquetas. Cada fila en esta tabla contiene información sobre un término específico, como el nombre y la descripción [2].
- ***wp_term_relationships***: esta tabla establece relaciones entre términos y contenidos del sitio. Por ejemplo, si una entrada de blog está asignada a la categoría «*noticias*», se crearía una fila en esta tabla que enlazará esa entrada con el término «*noticias*» en la tabla ***wp_terms*** [2].

- ***wp_term_taxonomy***: esta tabla se utiliza para clasificar a los términos en diferentes categorías taxonómicas, como categorías o etiquetas [2].

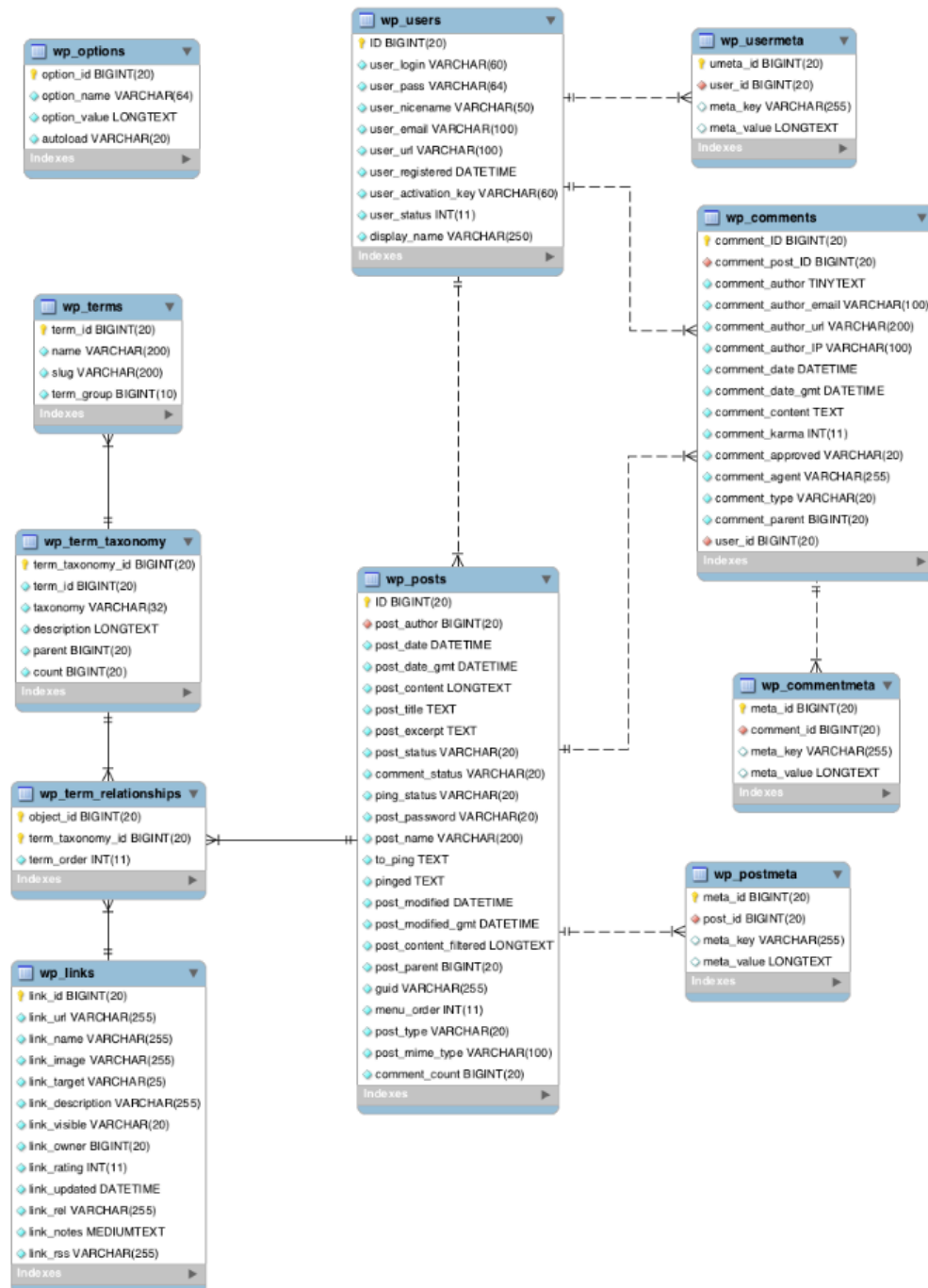


Imagen 1: Tablas de WordPress, breve explicación a continuación. Esta imagen se encuentra en la carpeta general Reto 3 Subcarpeta Bases de datos nombrada como “*DB_wordpress*”, para una mejor visualización de ella.

Relación entre tablas:

1. ***wp_users*** – ***wp_usersmeta***: Juntas, estas tablas se utilizan para administrar y almacenar información del usuario en WordPress. Por ejemplo, cuando un usuario inicia sesión en un sitio web, se crea una fila en la tabla ***wp_users*** para almacenar el nombre de usuario y la contraseña, y luego se pueden agregar metadatos sobre el usuario a través de la tabla ***wp_usermeta***. El diseño de la tabla de enlaces permite a *WordPress* administrar y mostrar bien la información.
2. ***wp_post*** – ***wp_postmeta***: Juntas, estas tablas se utilizan para administrar y almacenar contenido en WordPress. Por ejemplo, cuando se publica una nueva publicación de *blog*, se crea una fila en la tabla ***wp_posts*** para almacenar el título, el contenido y la fecha de publicación, y se pueden agregar metadatos sobre la publicación a través de la tabla ***wp_postmeta***. Esta estructura de tabla relacional permite a *WordPress* administrar y mostrar información correctamente.
3. ***wp_term*** – ***wp_term_taxonomy*** – ***wp_term_relationships***: Juntas, estas tres tablas se utilizan para administrar y almacenar información sobre temas y grupos de *WordPress*. Por ejemplo, cuando se crea una nueva categoría denominada Noticias, se crea una fila en la tabla ***wp_terms*** para almacenar el nombre y la descripción del término, y luego se crea una fila en la tabla ***wp_term_taxonomy*** para indicar que el término es una categoría. Cuando el término se especifica en una publicación de blog, se crea una fila en la tabla ***wp_term_relationships*** para verificar la relación entre el término y el contenido.
4. ***wp_users*** – ***wp_comments*** – ***wp_posts***: Estas tres tablas se utilizan juntas para gestionar y almacenar la información sobre comentarios en el sitio web. Por ejemplo, si existe una página llamada «sugerencias» y en esta se pueden hacer comentarios sobre ella, en el momento que se realice un comentario se crearía una fila en la tabla ***wp_comments*** con el id del usuario que lo realizó sumando también el id del post de la página donde se realizó el comentario y el comentario de ese usuario se crearía en la tabla ***wp_commentmeta*** relacionándolo con el id del comentario.

Habilidades

WordPress: Hemos utilizado *WordPress* para el desarrollo de nuestro sitio web ya que mediante experiencias previas y trabajos anteriores el uso de este se nos facilita porque este es muy manejable y fácil de usar. Además, también conocemos algunos *plugins* que nos van a ayudar con el desarrollo del sitio web. Por otro lado, *Wordpress* nos da la posibilidad de la gestión de base de datos y la creación automática de ella.

Eclipse: Hemos utilizado este lenguaje (*Java*), ya que, tenemos mucho conocimiento sobre este tipo de programa. Anteriormente hemos realizado proyectos y trabajos utilizando *Java*, tanto para buscar y extraer datos en una base de datos, hasta para hacer algunos tipos de juegos de adivinar la respuesta correcta.

Workbench: Hemos utilizado *Workbench* para la creación de las bases de datos de nuestro programa creado por en java teniendo en cuenta nuestra experiencia anterior sobre el gestor (*workbench*), lo utilizaremos para guardar la información necesaria.

Sitio de trabajo

Trabajamos de lunes a viernes en el centro de estudios de Nazaret. El horario de trabajo es desde las 8 y 10 am hasta las 2 pm. también tuvimos unos descansos asignados que son, primero: de 10:00 am hasta las 10:10am, segundo: 10:55 am hasta las 11:15am y el tercero: 01:00 pm hasta las 01:10 pm.

Organización

Los métodos de organización principales que hemos implementado para nuestro proyecto son:

- **GitHub** para el control de versiones y también para facilitarnos la práctica en todo el desarrollo del videojuego.
- **Google Drive** a manera de refuerzo en caso de que Git falle o no funcione de manera correcta.
- También, nos comunicamos mediante **WhatsApp y el correo electrónico** para mejorar el desarrollo de todo el proyecto y en caso de que alguna persona tenga algún imprevisto tenga la facilidad de avisar con tiempo y así poder cubrir esa baja momentánea o permanente.

División de tareas

Tareas

- Planificación
 1. Introducción, objetivos y organización, Identificar recursos
(Del 26-04-2024 al 30-04-2024).
 2. Habilidades, desglose de tareas, reuniones, *calendar*
(Del 26-04-2024 al 30-04-2024).
 3. *Gantt*, Riesgos y Prevenciones, Problemas o retrasos
(Del 26-04-2024 al 30-04-2024).
- *WordPress*
 4. Cambios en el diseño de la plantilla seleccionada
(Del 30-04-2024 al 02-05-2024).
 5. Cambios en la estructura y creación de la base de datos para la página web
(Del 30-04-2024 al 02-05-2024).
 6. Añadir los *plugins* necesarios
(Del 30-04-2024 al 02-05-2024)
- Bases de datos
 7. Diagramas entidad y relación
(El 03-05-2024)
 8. Creación de la base de datos para el juego
(El 03-05-2024).
- Lenguaje de marcas
 9. Hacer el JSON y el Schema del JSON. (1 pattern)
(El 03-05-2024).
- Programación
 10. Planificar la estructura del programa.
(Del 26-04-2024 al 29-05-2024)
 11. Programar código
(Del 29-04-2024 al 16-05-2024).
- Entornos de desarrollo
 12. Diagrama UML(*Java*).
(Del 02-05-2024 al 06-05-2024).
 13. Documentación
(Del 26-04-2024 al 16-05-2024).

- Inglés
14. Creación del guión, grabación y edición del video
(El 14-05-2024).
- Formación y orientación laboral
15. Contratos
(El 07-05-2024).
- General
16. Presentación
(Del 15-05-2024 al 16-05-2024).
- Sistemas informáticos
17. Conexión al servidor
(Del 10-05-2024 al 16-05-2024).

Encargados de las tareas

Encargado	Tareas
Bryan Arana	5,7,8,11,17,16
Imanol Pérez	1,2,9,14,16
Erick Guzmán	4,6,10,13,16,17
Julen López	3,12,15,16

Tabla 1, Asignación de tareas, en esta tabla se muestra el personal encargado del proyecto y cada tarea asignada a esa persona.

Diagramas de Gantt

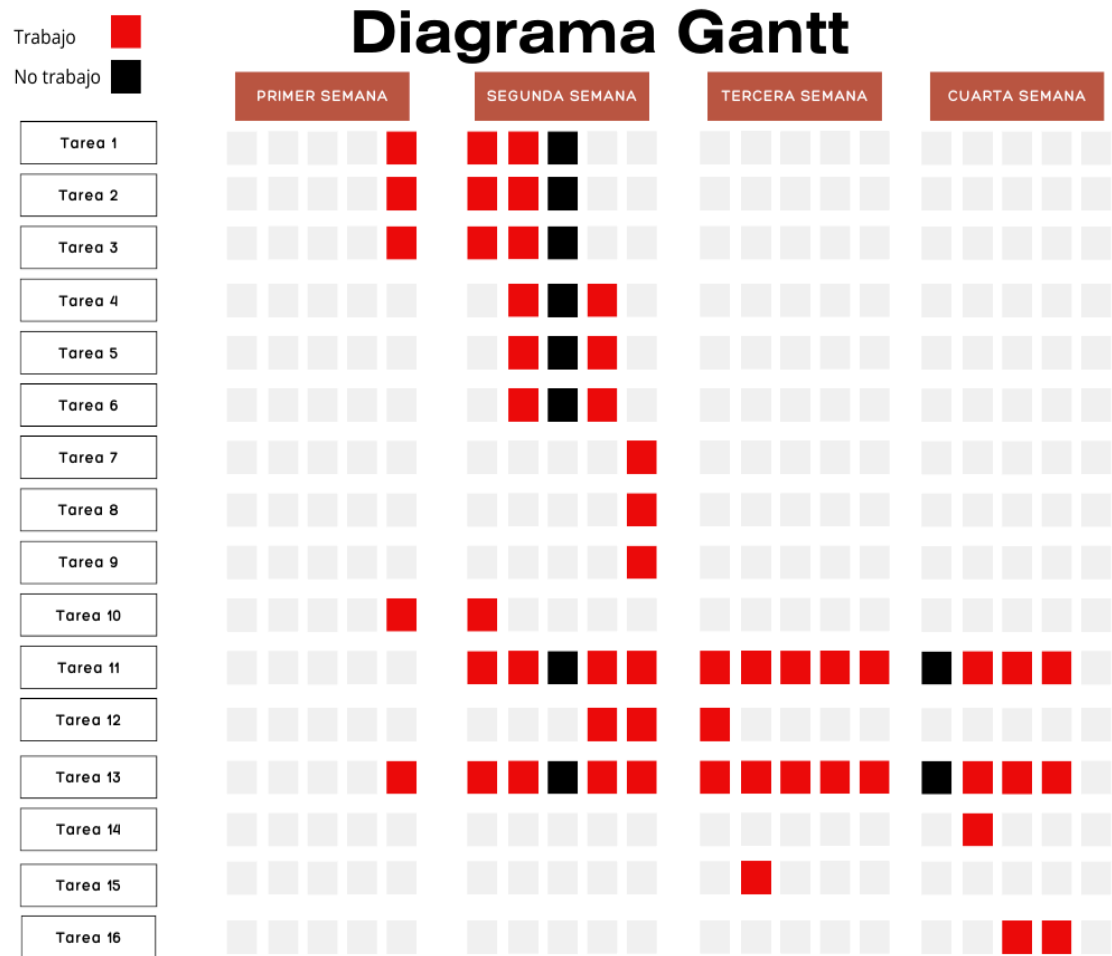


Imagen 2: Diagrama de Gantt 1: La imagen muestra el cronograma de actividades que planificamos inicialmente cuando comenzó el proyecto.

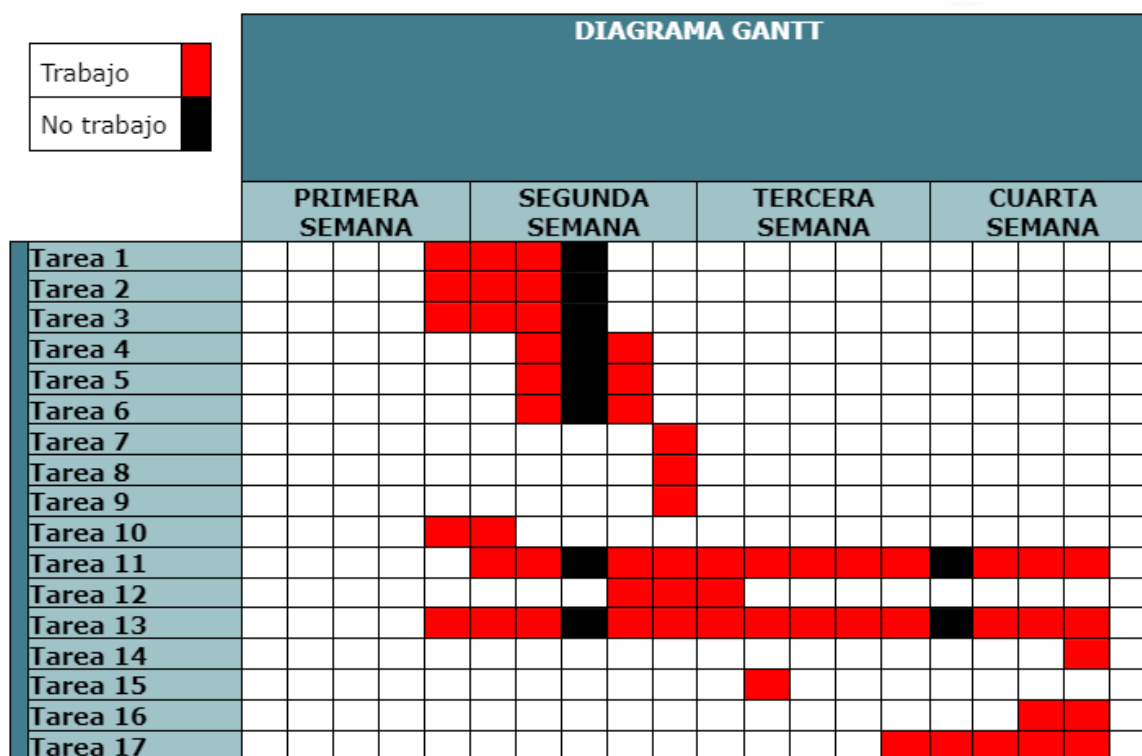


Imagen 3: Diagrama de Gantt 2: Esta imagen muestra el desarrollo real del proyecto y cómo evolucionó durante el tiempo dedicado al mismo. Además, verá tareas adicionales que no agregó antes.

Riesgos y plan de respuesta

Riesgos	Plan de respuesta
Retraso en la entrega del contenido por parte de los colaboradores.	Establecer comunicación clara y frecuentemente con los empleados para garantizar la entrega a tiempo. Implementar planes de contingencia para permitir que el contenido existente se actualice para cumplir con los plazos de entrega. Considerar contratar personal adicional para ayudar en las operaciones cuando sea necesario .
Cambios en los requisitos del cliente durante el desarrollo.	Establecer un proceso claro para la gestión de cambios, incluyendo la evaluación del impacto en el cronograma y el presupuesto. Mantener una comunicación abierta y transparente con el cliente para comprender sus necesidades y expectativas en todo momento. Documentar cualquier cambio en los requisitos del proyecto y obtener la aprobación del cliente antes de implementarlos.
Incumplimiento de los plazos de entrega debido a problemas imprevistos.	Establecer un margen de tiempo adicional en el cronograma para hacer frente a posibles retrasos . Implementar una comunicación proactiva con el cliente sobre cualquier desafío o problema que pueda afectar los plazos de entrega. Priorizar las tareas críticas y asignar recursos adicionales si es necesario para cumplir con los plazos acordados .

Tabla 2, Plan de respuesta: La tabla describe los riesgos que pueden surgir en el proyecto y proporciona un plan específico para abordar cada riesgo. En esta tabla solo se muestran algunos de los riesgos, los demás están en la planificación, que está en la carpeta Reto 3 Grupo D, en la subcarpeta Entornos de desarrollo.

Diagrama UML

Hemos hecho un diagrama UML basándonos en las diferentes clases y funciones con las que cuenta nuestro juego.

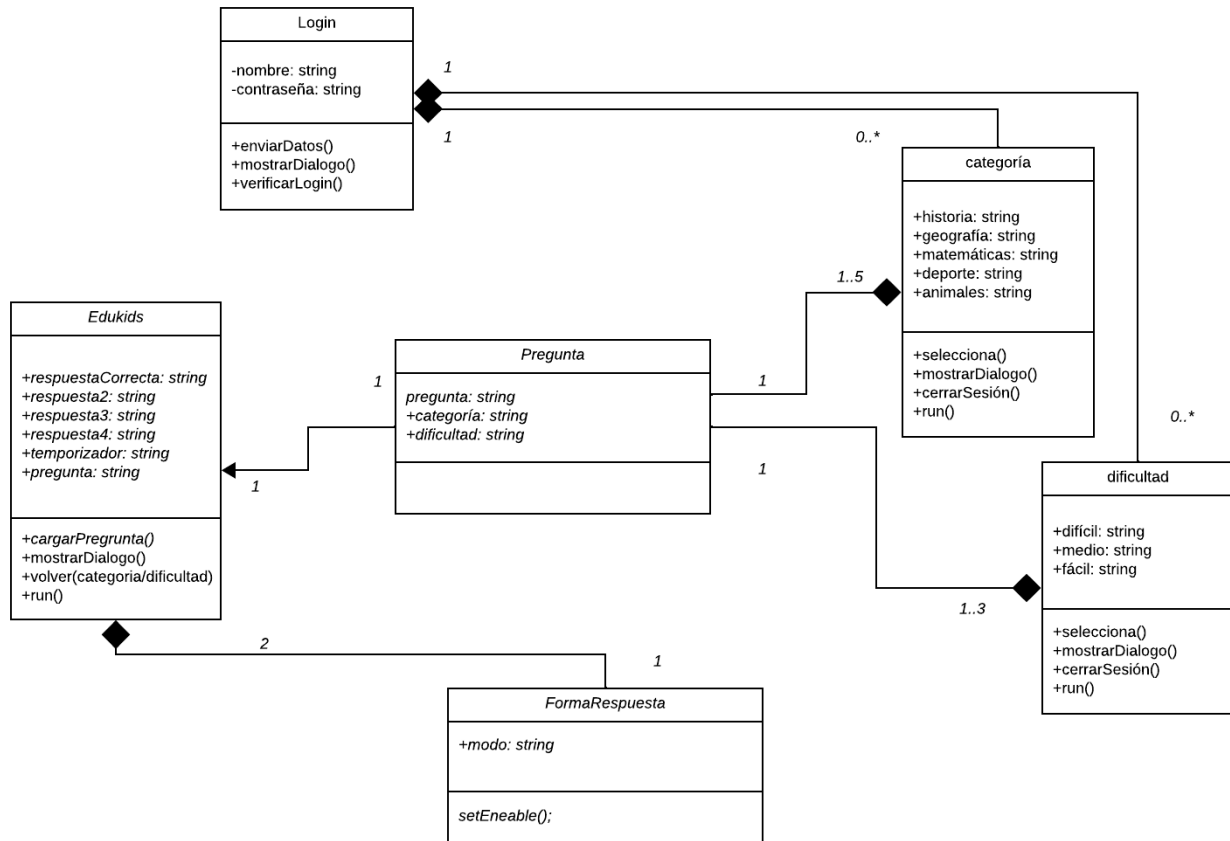


Imagen 4: En esta imagen podemos visualizar las diferentes clases y funciones que tiene el programa o el juego de *java*.

Para empezar, vamos a ver la clase *Login*. Esta es la Superclase de nuestro *Login* ya que de ahí derivan todas las demás clases, la clase *Login* cuenta con dos atributos que son nombre y contraseña, estos atributos se ocupan para que los usuarios escriban y puedan registrarse o iniciar sesión. Por otro lado, esta clase también cuenta con tres métodos los cuales son: en primer lugar, `enviarDatos()` este método se utiliza para cuando un usuario quiere registrarse y la función que realiza es enviar los datos a la base de datos. En segundo lugar, tenemos `mostrarDialogo()` este método tiene como función mostrar mensajes en pantalla. Por último, `verificarLogin()` este método se utiliza cuando una persona quiere iniciar sesión, por lo tanto, el método verifica si ya estás registrado o no.

Ahora pasamos a una clase llamada *EdukidsCategDificultad*, pero la hemos dividido en dos partes ya que *categoria* y *dificultad* son dos grupos distintos, por lo tanto, en la imagen 4

podemos ver que tenemos estas dos clases. Estas dos clases son una composición de la clase Login ya que es necesario que exista la clase *login* para poder utilizar las clases de categoría y dificultad. La clase categoría cuenta con 5 atributos que son: historia, matemáticas, geografía, animales y deporte, mientras que la clase dificultad cuenta con tres atributos que son fácil, medio y difícil. Para explicar los métodos hay que recordar que en realidad estas dos clases son una sola (EdukidsCategDificultad), por lo tanto, tienen los mismos métodos, estos con los siguientes: en primer lugar el método selecciona() este método tiene como función que los usuarios seleccionen una de las distintas categorías y una de las distintas dificultades, en segundo lugar mostrarDialogo() este método se utiliza para mostrar mensajes en pantalla, en tercer lugar cerrarSesión() este método tiene como función cerrar función y darte la posibilidad de abrir sesión otra vez o cambiar de usuario, por último el método run() este método se utilizan para que el contador funcione al mismo tiempo que los usuarios juegan.

Como podemos visualizar en la imagen 4 la siguiente clase sería pregunta. Está no es una clase aparte de nuestro juego, pero la hemos puesto como una clase aparte ya que dependiendo de la categoría y dificultad escogida por los usuarios salen unas u otras preguntas distintas. Esto también nos quiere decir que esta clase es una composición de las clases categoría y dificultad, ya que, dependen de estas. Esta cuenta con los atributos de categoría, dificultad y también con la información de la pregunta. Por último, la clase pregunta es herencia de la clase Edukids ya que esta pregunta se utiliza en la clase Edukids como podemos ver en la imagen 4.

Ahora vamos a ver la clase de Edukids, esta clase es donde los usuarios juegan a contestar las preguntas, por lo tanto, esta clase cuenta con los siguientes atributos: respuestaCorrecta, respuesta2, respuesta3, respuesta4, temporizador y pregunta. por otra parte en la imagen 4 podemos visualizar un total de 4 métodos y estos son los siguientes: en primer lugar cargarPregunta() este método tiene como función cargar las preguntas de la base de datos según la categoría y dificultad seleccionada, en segundo lugar mostrarDialogo() este método se utiliza para mostrar mensajes en la pantalla, en tercer lugar volver() este método es utilizado para que los usuarios puedan cambiar de categoría y de dificultad cuando ellos quieran, por último el método run() este método sirve para que el temporizador se reinicie cada vez que pasa una pregunta.

La clase formaRespuesta, tiene un único atributo que es modo y tiene un solo método que es *setEneable()* este método se utiliza para seleccionar uno de los dos tipos de respuestas que

puede tener el usuario. Por otra parte, es una composición de la clase Edukids ya que depende de éste para poder existir. Si la clase edukids no existe, la clase de FormaRespuesta no puede existir ya que depende de la información de la clase de edukids.

Ahora vamos a explicar la multiplicidad de cada una de las tablas. Para empezar entre las tablas de Login, categoría y dificultad sería una multiplicidad de 1 ya que los usuarios que se logueen tendrán la opción de escoger una única categoría, lo mismo pasa con la dificultad, del otro lado tendríamos una multiplicidad de 0..* ya que se puede loguear 0 o muchos usuarios. Entre las clases de categoría, dificultad y pregunta tendríamos distintas multiplicidades, empezando por pregunta, pregunta puede ser de entre 1.. 5 categorías y la pregunta puede ser de entre 1.. 3 dificultades esto debido a que solo hay esas opciones, por otra parte, la dificultad sólo puede ser una ya que no puedes seleccionar dos al mismo tiempo y pasa lo mismo con la categoría.

Entre las clases de pregunta y Edukids tenemos las siguientes multiplicidades: la clase Edukids puede tener una sola pregunta y la clase pregunta solo puede estar 1 vez en la clase de edukids. Por último, entre las clases de Edukids y formaRespuesta tenemos la siguiente multiplicidad: La clase edukids solo puede contener 1 de los dos tipos de formaRespuesta, mientras que la clase formaRespuesta contiene dos opciones para responder.

Para una mejor visualización del diagrama UML hemos adjuntado el archivo en la carpeta de “Entornos de desarrollo” en la carpeta general del reto 3 o en el enlace marcado en la referencia [3].

Conclusión

En conclusión, este capítulo resume el desarrollo de nuestra planificación entregada anteriormente mostrando lo más importante de nuestro proyecto, también, muestra los cambios que se hicieron a medida que transcurría el tiempo en proyecto.

Bases de datos

Introducción

En este capítulo se explicará detalladamente cada una de las metodologías y partes a desarrollar para la creación y optimización en las bases de datos, también, la estructura, la mecánica de ellas detalladamente, las decisiones clave y el porqué de ellas, además, la tecnología utilizada para su implementación.

Creación de las bases de datos

Inicialmente creamos una base de datos con dos tablas independientes entre sí que contienen la información relativa a las preguntas de nuestro videojuego y el inicio de sesión de nuestros clientes llamada *EduKids*, pero, después hemos decidido optar por agregar la tabla preguntas a la base de datos que nos proporciona *WordPress* para mayor facilidad de uso. Ya que, *WordPress* nos proporciona una tabla de usuarios (*wp_users*) decidimos descartar y borrar la tabla de “*Login*” creada inicialmente en *EduKids*.

La base de datos *EduKids* la hemos creado inicialmente en *PhpMyAdmin*, y hemos usado XAMPP para conectarnos a ella en localhost, para después mudarse a un servidor en línea.

Modelo entidad relación Edukids inicial

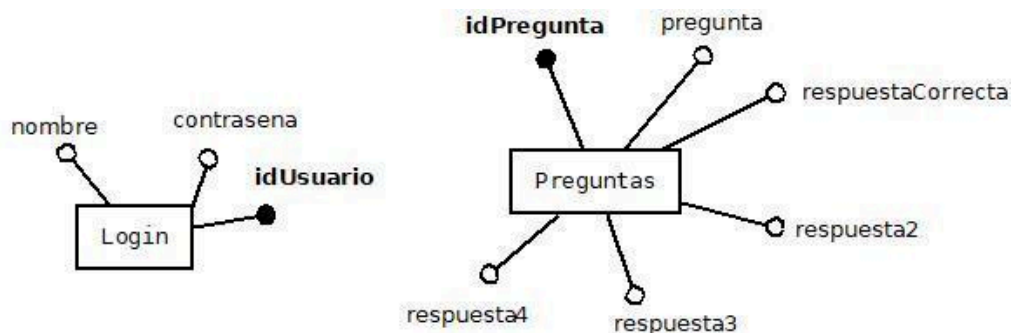


Imagen 5: Modelo entidad relación, esta imagen muestra el diagrama inicial de nuestras tablas en la base de datos “*EduKids*”.

Modelo lógico Edukids inicial:

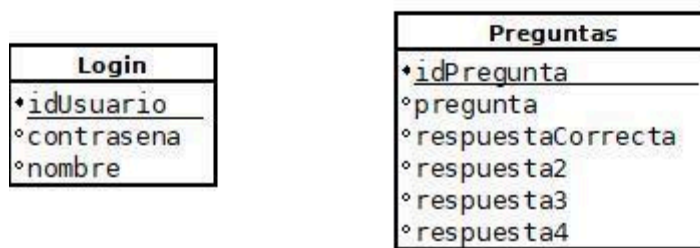


Imagen 6: Modelo lógico, esta imagen muestra el diagrama inicial de nuestras tablas en la base de datos “*EduKids*”.

Modelo entidad relación Edukids final

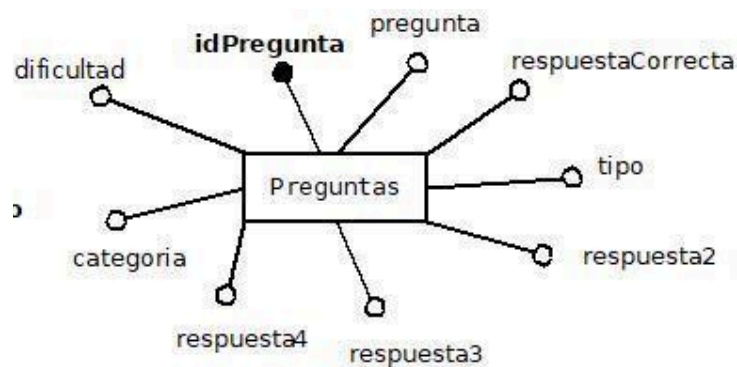


Imagen 7: Modelo entidad relación, esta imagen muestra el diagrama final de nuestra tabla en la base de datos “*EduKids*”, en esta se eliminó la tabla “**Login**” para después adaptar la tabla “**Preguntas**” a la base de datos de *WordPress*.

Modelo lógico Edukids final

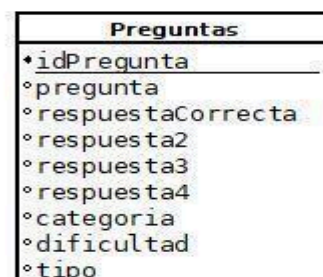


Imagen 8: Modelo lógico, esta imagen muestra el diagrama final de nuestra tabla en la base de datos “*EduKids*”, en esta se eliminó la tabla “**Login**” para después adaptar la tabla “**Preguntas**” a la base de datos de *WordPress*.

¿Por qué la razón de estos cambios?

Inicialmente decidimos tener dos bases de datos independientes para cada apartado (Sitio web, Juego) pero repentinamente nos surgió el problema que nos hizo cambiar de decisión, este problema nos obligaba a manualmente ingresar los usuarios registrados en la base de datos de *WordPress* a la base de datos del juego, eso nos haría perder demasiado tiempo y recursos, entonces, decidimos optar por adaptar la tabla “**Preguntas**” a la base de datos de *WordPress* y utilizar la tabla ya generada en esta llamada “*wp_users*” para, de manera más óptima utilizarla de manera paralela en nuestro sitio web y en nuestro juego, así, los usuarios de cada apartado se podrían registrar ya fuese en el juego o en el sitio web y de esta manera también podrían iniciar sesión en esta.

Conclusión

En conclusión, este capítulo resume los procesos y grupos involucrados en la construcción y optimización de una base de datos, incluido el diseño y las decisiones clave. Se crearon dos bases de datos separadas para el videojuego EduKids y el sitio web de WordPress, desarrolladas y migradas utilizando herramientas como *PhpMyAdmin* y XAMPP.

Lenguaje de marcas

Introducción

En este capítulo se explicará detalladamente cada una de las metodologías y partes a desarrollar para la creación y optimización en el sitio web de Edukids, también, la estructura, la mecánica de esta detalladamente, la creación de un JSON basado en una de las tablas existentes en la base de datos de “Edukids”, la creación de una expresión regular y su implementación en el JSON, además de una breve muestra de nuestro sitio web.

Estructura del sitio web

En el proceso de modificación de nuestra página web, hemos utilizado *WordPress*, que es una plataforma de gestión de contenidos, y fuimos capaces de mejorar la plantilla para crear una experiencia fluida y atractiva para nuestros usuarios. *WordPress* nos ha brindado el control necesario sobre el diseño, la interactividad y las funciones de nuestra plataforma en línea.

Nuestra página web está dividida en cuatro campos, donde a cada campo le pertenece una página distinta. Estos son los campos:

1. **Inicio:** es nuestra página principal donde se mostrará y se explicara lo que es nuestra plataforma y en qué consiste.



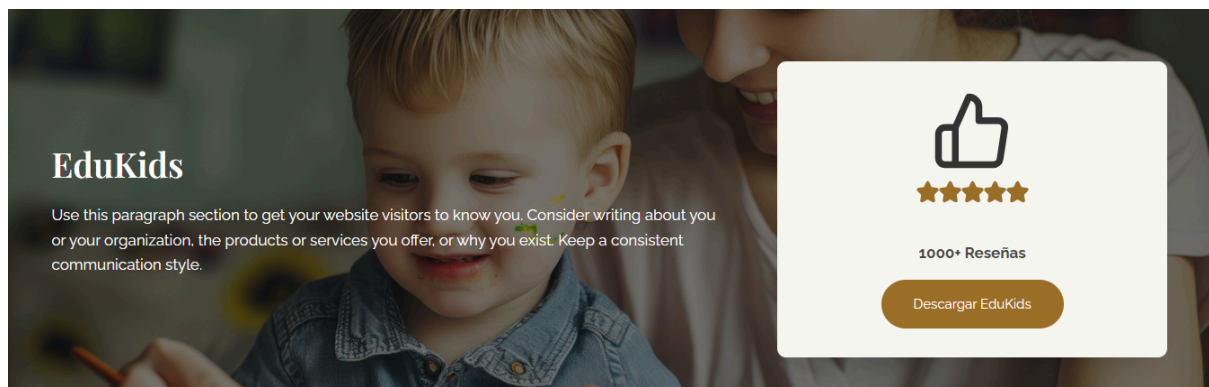
Imagen 9: Imagen de sitio web, muestra el apartado de inicio (*Home*) de nuestro sitio web mostrando información sobre nuestra empresa.

2. **Contenido del juego:** en este apartado está la información del contenido de la plataforma donde se mostrará y se explicará las categorías de las preguntas que habrá en ella, geografía, deportes, animales, matemática e historia.



Imagen 10: Imagen de sitio web, muestra el apartado “Contenido del juego” de nuestro sitio web mostrando información sobre el juego desarrollado por nuestros programadores.

3. **Descarga del juego:** es la sección donde se podrá descargar nuestra plataforma, en Windows o en Linux, tanto en ordenadores, tabletas y móviles, podrán visualizar las valoraciones de nuestro juego.



Características del videojuego



Imagen 11: Imagen de sitio web, muestra el apartado de Descarga del videojuego y sus requisitos y características para la instalación.

4. **Acerca de nosotros:** Es el apartado donde estará la información de nuestra empresa y de cada integrante de la plataforma, también, habrá un formulario para que las personas que tengan dudas tengan la opción de preguntar tranquilamente.

Descripción de nuestra empresa.

En Edukids, contamos con un equipo de profesionales altamente capacitados en educación infantil, con una amplia experiencia en el diseño de contenidos educativos y en el uso de herramientas tecnológicas para facilitar el proceso de enseñanza-aprendizaje.

En resumen, en Edukids nos comprometemos a proporcionar educación de calidad, estimulante y adaptada a las necesidades individuales de cada niño, a través de una plataforma web innovadora y segura. ¡Únete a la comunidad Edukids y comienza a descubrir un mundo de aprendizaje y diversión para los más pequeños!

Erick es un desarrollador web con una amplia experiencia en la creación de sitios interactivos y funcionales. Su pasión por la tecnología y su creatividad lo convierten en un profesional altamente calificado en el campo del desarrollo web.



Erick Alexander
Desarrollador De Web

Brayan es un programador talentoso y apasionado por la tecnología, con una habilidad excepcional para resolver problemas y crear soluciones innovadoras en el mundo de la programación. Su dedicación y experiencia lo convierten en un profesional altamente capacitado y confiable en su campo.



Brayan Alexander
Programador

Inti es un administrador de redes altamente capacitado, con habilidades sólidas en la configuración y mantenimiento de



Imagen 12: Imagen de sitio web, muestra el apartado de “acerca de nosotros” (*about*), también muestra información sobre todos nuestros empleados y una breve descripción de ellos.

Aparte, nos hemos asegurado de utilizar herramientas que son ampliamente compatibles, flexibles y bien conocidas por nuestro equipo de desarrollo. Esto nos ha permitido trabajar de manera eficiente y efectiva para ofrecer una experiencia web de alta calidad a nuestros usuarios.

En resumen, WordPress nos ha brindado una estructura atractiva, funcional y fácil de usar para nuestros usuarios.

JSON (*JavaScript Object Notation*), JSON schema y expresión regular

Hemos desarrollado un JSON en base a nuestra tabla de preguntas mostrando en él algunos de los ejemplos que existen dentro de nuestra tabla, también, seguidamente desarrollamos el *schema* del mismo dándole los parámetros y restricciones más convenientes para cada sección, además, elaboramos una expresión regular para este mismo basándose enteramente sobre el apartado “Pregunta”.

Para una mejor visualización de estos JSON hemos adjuntado cada uno de ellos en la carpeta de “Lenguaje de marcas” en la carpeta general del reto 3.

```
{
  "$schema": "../JSONreto3tabla1.schema.json",
  "preguntasjuego": [
    {
      "Pregunta": "¿Si juegas a la NFL, qué deporte practicas?",
      "respuestaCorrecta": "Fútbol americano",
      "respuesta2": "Baloncesto",
      "respuesta3": "Rugby",
      "respuesta4": "Fútbol",
      "categoria": "Deportes",
      "dificultad": "Fácil",
      "tipo": "escrito"
    },
  ],
}
```

Imagen 13: JSON, Esta imagen muestra la estructura de nuestro JSON creado a partir de la tabla “Preguntas”.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "preguntasjuego": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "Pregunta": {
            "type": "string",
            "pattern": "¿[a-zA-ZÀ-ŸA-Z0-9\\, ]+( [a-zA-ZÀ-ŸA-Z0-9])*\\?"
          },
          "respuestaCorrecta": {
            "type": "string"
          },
          "respuesta2": {
            "type": "string"
          },
          "respuesta3": {
            "type": "string"
          },
          "respuesta4": {
            "type": "string"
          },
          "categoria": {
            "type": "string",
            "enum": ["Deportes", "Matemáticas", "Historia", "Animales", "Geografía"]
          },
          "dificultad": {
            "type": "string",
            "enum": ["Fácil", "Medio", "Difícil"]
          },
          "tipo": {
            "type": "string",
            "enum": ["seleccionar", "escrito"]
          }
        },
        "required": ["Pregunta", "respuestaCorrecta", "respuesta2", "respuesta3", "respuesta4", "categoria", "dificultad", "tipo"]
      }
    }
  }
}
```

Imagen 14: JSON, Esta imagen muestra la estructura de nuestro JSON *schema* creado a partir de la tabla “Preguntas” agregando a él la expresión regular creada por nuestro equipo.

Pattern

Para el JSON Schema hemos implementado la siguiente expresión regular en el apartado de pregunta para mejorar el sistema de filtrado de preguntas y evitar errores en ellas.

"pattern": "^[a-zA-ÿA-Z0-9\\,]+([a-zA-ÿA-Z0-9])*\\|?"

Estos son algunos ejemplos de su utilización

La expresión regular determina que:

```
{
  "Pregunta": "¿Si juegas a la NFL, qué deporte practicas",
  "respuestaCorrecta": "Cada 4 años",
  "respuesta2": "Cada 6 años",
  "respuesta3": "Cada 2 años",
  "respuesta4": "Cada año",
  "categoria": "Deportes",
  "dificultad": "Fácil",
  "tipo": "escrito"
}
```

Imagen 15: JSON, Esta imagen muestra que la pregunta debe de llevar siempre signos de interrogación al inicio y al final en esta, sino automáticamente nos mostrará error en ella.

```
{
  "Pregunta": "¿?",
  "respuestaCorrecta": "Cada 4 años",
  "respuesta2": "Cada 6 años",
  "respuesta3": "Cada 2 años",
  "respuesta4": "Cada año",
  "categoria": "Deportes",
  "dificultad": "Fácil",
  "tipo": "escrito"
}
```

Imagen 16: JSON, Esta imagen muestra que la pregunta debe llevar dentro de los signos de interrogación como mínimo un *string* de letras y números sino automáticamente nos mostrará error en ella.

```
{
  "$schema": "./JSONreto3tabla1.schema.json",
  "preguntasjuego": [
    {
      "Pregunta": "¿+45345sdfsdf?",
      "respuestaCorrecta": "Fútbol americano",
      "respuesta2": "Baloncesto",
      "respuesta3": "Rugby",
      "respuesta4": "Fútbol",
      "categoria": "Deportes",
      "dificultad": "Fácil",
      "tipo": "escrito"
    },
  ],
}
```

Imagen 17: JSON, Esta imagen muestra que la pregunta no puede empezar por caracteres especiales., sino automáticamente nos mostrará error en ella.

Conclusión:

En conclusión, este capítulo resume el diseño y desarrollo de nuestro sitio web, explica sistemáticamente cada parte y describe brevemente cada parte. También, describe la implementación de los formatos JSON y JSON *Schema* estos basados en la tabla “Preguntas” de la base de datos, además, la explicación del *pattern* creado para este mismo.

Sistemas informáticos

Introducción:

En este capítulo se explica cómo se realizó la instalación de la base de datos a un servidor independiente del sitio web, a su vez se explican los problemas que se tuvieron y las soluciones para este.

Instalación del servidor

Inicialmente hemos tomado la decisión de montar la base de datos y el sitio web en dos servidores distintos e independientes de cada uno, pero a medida que transcurre el tiempo son cada vez más los problemas que nos surgen al intentar hacer eso.

Creamos las bases de datos del sitio web y del juego, después, montamos la base de datos para el juego y que en ella se pudieran recoger los datos de los jugadores, a parte, el sitio web tenía su base de datos independiente a la del juego montada en *DinaHosting* y aquí empezaba el primer problema.

Ya que cada una no se relacionaba con la otra existía el problema que, en el juego las personas que se registraban en el sitio web no podían ingresar ya que eran bases de datos diferentes y así también viceversa, las personas que se registraban en el sitio web no podían en el juego. Para ello hemos decidido adaptar la tabla “**PreguntasJuego**” de la base de datos del juego a la base de datos del sitio web para que los datos recopilados en el juego se pudiesen utilizar en la base de datos general del sitio web y viceversa, luego, exportamos la base de datos que nos proporciona *WordPress* para luego utilizarla más adelante.

Luego de exportar la base de datos de *WordPress*, la importamos en la instancia Amazon RDS para utilizarla en el juego dándole las credenciales a el archivo PHP de *WordPress* para que los datos ingresados en el sitio web se guardarán en la base de datos montada en RDS, Seguidamente nos surgió el segundo problema, la optimización del sitio web se veía afectada de manera muy extraña y exagerada a tal punto que cada página de nuestro sitio tardaba alrededor de veinte segundos en cargar, optimizamos utilizando *plugins* y cambiando el tamaño de las imágenes pero aun así el sitio web no funcionaba correctamente, entonces, decidimos montar la base de datos para el juego en un solo servidor junto con el sitio web (*DinaHosting*) y así conseguimos que el sitio web funcionará de manera óptima y correctamente.

Y es así como hemos montado la base de datos para nuestro sitio web junto con la base de datos para nuestro juego.

Conclusión

En resumen, la implementación de la base de datos y el sitio web en servidores separados creó problemas de integración e impidió el acceso entre usuarios. Las bases de datos se fusionaron y exportaron a Amazon RDS, pero esto ralentizó el sitio web. Finalmente, implementar ambas bases de datos en un único servidor en *DinaHosting* resolvió los problemas de rendimiento e hizo que el sitio web y el juego funcionaran de manera óptima. Esta experiencia resalta la importancia de considerar la integración y el desempeño desde el inicio de la infraestructura del proyecto.

Programación

Introducción

En este apartado explicaremos qué es y en qué consiste nuestro juego educativo. Explicaremos todos los detalles y también las funcionalidades que tiene el programa.

Nuestro juego consiste en un conjunto de preguntas en el cual los usuarios pueden elegir las categorías e incluso las dificultades. El juego también tiene dos opciones de respuesta, una es escribiendo y la segunda es seleccionando la respuesta.

Inicio de Sesión

Para comenzar explicaremos lo que tiene el diseño del *Login* inicio de sesión y registro. Este apartado pide al usuario un su nombre y contraseña, por lo que, son dos campos para rellenar, pero la contraseña tiene que cumplir con unos requisitos para que esté correcta. Los requisitos están escritos en la parte inferior de los botones de registro y inicio de sesión, cada vez que la contraseña cumpla con uno de los requisitos ese requisito se pondrá de color verde, en caso contrario, los requisitos estarán en rojo para indicar que aún no se cumplen.

Los usuarios tienen que iniciar sesión en el juego, y si no están registrados tienen que registrarse. Los datos se enviarán a la base de datos para almacenarlos y utilizarlos para el inicio de sesión. Se pide a los usuarios el nombre y la contraseña. Para poder iniciar sesión o registrarse la contraseña tienen que cumplir un conjunto de requisitos, por ejemplo: la contraseña tiene que contener 8 o más caracteres, la contraseña deberá tener al menos una mayúscula.

Si haces clic en el botón de regístrate o inicio de sesión y no está rellenado alguna de las dos columnas nombre y contraseña, saldrá un mensaje de que aún no lo has rellenado, por lo tanto, son campos obligatorios para poder continuar en una nueva ventana en la que los usuarios tienen que elegir entre las distintas dificultades y categorías que hay para empezar a jugar.

El botón de inicio de sesión verifica si ya has iniciado sesión anteriormente, mientras que el botón de "Regístrate" envía los datos a la base de datos para registrarte. Cuando inicias sesión se realiza una búsqueda del usuario en la base de datos para ver si ya está registrado o no. Si ya está registrado se inicia sesión, si no está registrado pone un mensaje en pantalla. Cuando te registras se realiza una búsqueda en la base de datos para verificar que no hay otro usuario idéntico al que se quiere registrar, por lo tanto, todos son diferentes.

Categoría/Dificultad

Esta parte es importante ya que se extraerán de la base de datos las preguntas y respuestas, según la categoría y dificultad que elijan, por lo tanto, cada categoría tendrá un conjunto de preguntas distintas según la dificultad elegida.

Por un lado, hay 5 categorías distintas que son: Historia, Matemáticas, Deporte, Geografía y Animales, por otro lado, hay 3 tipos de dificultades que son: Fácil, Medio y Difícil.

Este apartado está programado con dos grupos distintos. El grupo de categorías y el grupo de dificultades. Estos grupos están hechos con *ButtonGroup*, por lo tanto, cada uno de los grupos está formado por un conjunto de *RadioButton*. Esto hace que solo puedan elegir una de las opciones de cada grupo.

Si no has seleccionado categoría o dificultad y haces clic en el botón de enviar, mostrará un aviso en pantalla de que hay que seleccionar una categoría y una dificultad obligatoriamente. El botón de aceptar enviará las opciones al apartado de pregunta y respuesta para hacer la búsqueda de las preguntas en la base de datos.

Por último, hay un botón de cerrar sesión. Si haces clic en este botón se cerrará la sesión, por lo tanto, te enviará al apartado de *login* para que inicies sesión o te registres.

Pregunta y respuesta

En esta parte los usuarios juegan a adivinar las respuestas según la pregunta que les ha tocado, cada pregunta tiene 4 opciones y solo una es la respuesta correcta. Por otro lado, cada pregunta puede ser de dos tipos. Puede ser escribir la respuesta o seleccionar la respuesta correcta. Hay un tiempo límite para cada pregunta y se reinicia cada vez que cambias la pregunta.

El diseño se basa en un *JLabel* que es donde va la pregunta, este *label* está en la parte superior de la pantalla, después tenemos las cuatro opciones que son botones que los usuarios pueden hacer clic, estos botones se utilizan cuando la pregunta es de tipo seleccionar la respuesta. Por otro lado, en la parte inferior podemos encontrar un *JTextField* y un botón de verificar, estos se utilizan cuando la pregunta es tipo escrito, cuando la pregunta es de tipo seleccionado el *JTextField* y el botón de verificar de la parte inferior están deshabilitados.

Hay un botón de volver, este botón es necesario para regresar a la parte donde eliges la categoría y la dificultad. Esto está hecho así por si quieres cambiar las opciones que estás

jugando, por lo que, podrás volver a elegir la dificultad y la categoría para volver a jugar con distintas opciones.

Las preguntas y respuestas que se mostrarán en pantalla dependen de lo que has seleccionado en el apartado de categoría y dificultad. También dependiendo de la pregunta tendrás que escribir o seleccionar la respuesta.

Hay un temporizador en la parte superior izquierda, si este temporizador llega al tiempo establecido mostrará un mensaje en pantalla diciendo que se ha acabado el tiempo y cambiará automáticamente a la pregunta siguiente. Cuando el temporizador está en el segundo 25s cambiará a color rojo en señal de que se está acabando el tiempo.

Dependiendo de si has acertado la pregunta ya sea seleccionando o escribiendo se mostrará un mensaje en pantalla en el cual dice si has acertado o te has equivocado. Después se cambia a la siguiente pregunta automáticamente.

Para finalizar si haces clic en el botón de volver, antes de ir al apartado de categoría y dificultad, se muestra un mensaje en pantalla en el que aparece el total de preguntas que has hecho y también la cantidad de preguntas que has hecho bien y que has hecho mal.

Conclusión

En conclusión, el proceso de inicio de sesión y registro requiere que el usuario proporcione un nombre de usuario y una contraseña que cumplan ciertos requisitos. Si la contraseña está incompleta o no es válida, aparecerá un mensaje de error. Los usuarios deben iniciar sesión o registrarse para jugar, y su información será verificada por la base de datos. Este sistema te permite elegir entre cinco niveles y tres niveles de dificultad utilizando únicamente el botón de selección. El juego cuenta con diferentes preguntas que pueden ser escritas y respondidas o de opción múltiple, con un cronómetro que muestra cuánto tiempo queda. Finalmente, se muestra un resumen de las acciones del jugador.

Formación y orientación laboral (FOL)

Introducción

En este capítulo se analizan las cuestiones más importantes en la clase de formación y orientación laboral, es necesario prevenir los accidentes laborales para garantizar la seguridad y la salud de los trabajadores, y se utilizan planes de gestión para analizar las prácticas de gestión empresarial. También se examina el proceso de planificación de costes, las diferencias en los contratos laborales y la importancia de los informes empresariales como herramienta de comunicación. Esta guía describe los aspectos más importantes de la formación y el desarrollo profesional, proporcionando a los lectores conocimientos y habilidades prácticas.

Organigrama

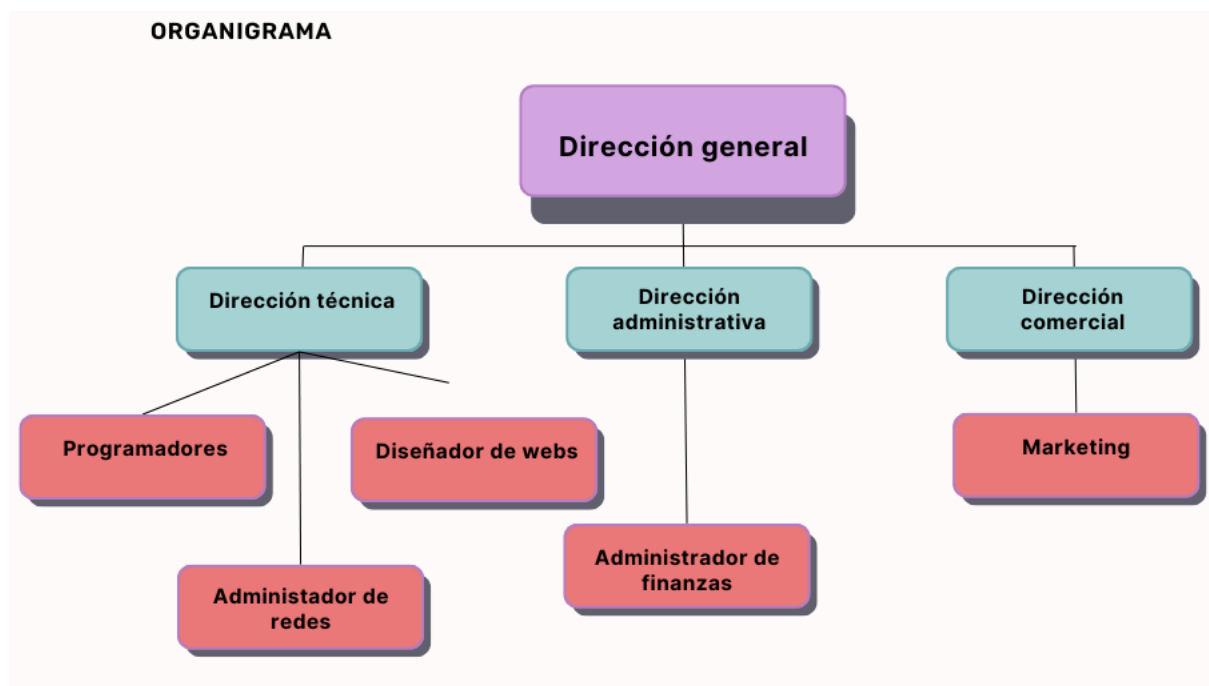


Imagen 18: Organigrama, muestra la composición de nuestra empresa y la clasificación de los empleados.

Nóminas y contratos.

En el apartado de nóminas hemos creado las nóminas de nuestros trabajadores, entre ellos están: Administrador de finanzas, Administrador de redes, Diseñador web, Especialista en Marketing y Programador cada uno con su correspondiente contrato laboral.

Estos se adjuntan en la carpeta de FOL en la carpeta general del Grupo-D-retro 3.

Implementaciones futuras

Lógica

Sistemas informáticos

Instalar el servidor web y el servidor de bases de datos MySQL en una dirección diferente a la ya establecida (*Dinahosting*) para realizar pruebas de instalaciones propias fuera del ámbito escolar.

Bases de datos

Realizar mejoras en la configuración, manejo y optimización de la base de datos para futuras actualizaciones en el sitio web como implementación de sistemas de pagos, ventas en el sitio web y otros.

Lenguaje de marcas

Establecer una conexión Java-JSON para la lectura automática de datos en la base de datos del servidor web.

Entornos de desarrollo

Buscar sistemas de coordinación para optimizar el trabajo en equipo y mejoras en la planificación para una mejor optimización del proyecto o de los proyectos a futuro.

Programación

Insertar nuevas funciones como podría ser más categorías de dificultad, otros modos de juego, sistema de puntaje, etc. Y optimizar el código para posibles nuevas aplicaciones o juegos.

Optimizar y mejorar la interfaz gráfica para una experiencia más amigable y atractiva para el público.

Gráfica

Página web

Mejorar el estilo general de la página e incluir nuevos juegos.

Problemas encontrados y soluciones para ellas

Tema	Problemas	Soluciones
Servidores	A la hora de poner la base de datos en el servidor AWS y la página web en Dinahosting surge un problema de optimización y velocidad. La página web tarda aproximadamente 20 segundos en cambiar de página o apartado, por lo tanto, no es nada óptimo para los usuarios.	Para solucionar el problema de los servidores hemos colocado la base de datos y la página web en un mismo servidor, en Dinahosting. Esto ha hecho que la página web cargue con mayor velocidad y que sea más óptima para los usuarios, ahora tarda 1 segundo en cargar cada página.
Integrantes	A lo largo del trabajo hemos perdido a uno de los integrantes del equipo. Esto supone mucho más trabajo para los demás integrantes.	Para solucionar esto, hemos repartido las tareas de este integrante y hemos trabajado aún más duro para poder llegar al objetivo.
Salidas inesperadas	A lo largo del trabajo hemos tenido algunas situaciones inesperadas en las cuales los integrantes del grupo han faltado a algunas horas de trabajo, he incluso han faltado días enteros. Esto debido a enfermedades o problemas externos urgentes.	Para solucionar esto, algunos días se ha trabajado desde casa o los integrantes que faltaron han tenido que trabajar más para retomar el tiempo perdido.
Bases de datos	Teníamos creadas dos bases de datos al inicio de nuestro proyecto, lo cual no era nada óptimo ya que los usuarios tendrían dos cuentas distintas para registrarse.	Solucionamos este problema uniendo las dos bases de datos y así tener todos los datos en una sola.
Fallos en la red	Un día en el horario de trabajo falló la conexión en algunos dispositivos, por lo tanto no podíamos seguir trabajando.	Como algunos dispositivos si tenían conexión a internet, seguimos trabajando con ellos, además utilizamos algunos programas que no necesitan conexión como eclipse, por lo tanto, podíamos seguir trabajando.
Login	A la hora de iniciar sesión tenemos problemas ya que los métodos de encriptación de wordpress y de eclipse son diferentes por lo que a la hora de iniciar sesión hay problemas	La solución es que en un futuro implementemos otro método de encriptación para wordpress o modifiquemos algunas cosas del programa de encriptar de wordpress .

Tabla 3, Problemas y soluciones: En esta tabla podemos observar algunos problemas que hemos tenido a lo largo del trabajo y también cómo los hemos solucionado.

Conclusión

En este proyecto, hemos diseñado y desarrollado un videojuego para agilizar la mente y practicar mientras te diviertes. Nuestra aplicación ofrece funciones básicas de programación pero escalables para que en un futuro no muy lejano desarrollar el videojuego de manera más avanzada. Los resultados de nuestra evaluación de usabilidad mostraron que a los usuarios les gustan las funciones de la aplicación. Además, los usuarios informaron un aumento significativo en agilidad mental. Finalmente, nuestro proyecto demuestra el valor de los videojuegos no solo como una distracción en la realidad de los adolescentes sino también como una herramienta eficaz para mejorar la capacidad mental de cada persona. Para futuras mejoras, implementaremos funciones de colaboración y ampliaremos la disponibilidad de la aplicación para diferentes plataformas.

Bibliografía

[1]

<https://www.byronvargas.com/web/como-es-la-estructura-de-wordpress/>

[2]

https://www.optimbyte.com/base-de-datos-wordpress/#:~:text=La%20tabla%20wp_users%20almacena%20informaci%C3%B3n,sobre%20los%20usuarios%20del%20sitio.

[3]

https://lucid.app/lucidchart/1540053f-b6bf-455c-9994-38c4f21e749a/edit?utm_source=youtu.be&utm_medium=video&utm_campaign=uml_class_es&invitationId=inv_d65904cf-cdab-41b8-af9f-6e6005b57b9f&page=0_0#

SITIO WEB

<https://kiolariburuz.eus/>