



UNIVERSIDAD DE ESPECIALIDADES ESPÍRITU SANTO

ASIGNATURA
SISTEMAS DISTRIBUIDOS

PROFESOR
GUILLERMO OMAR PIZARRO VASQUEZ

PROYECTO
REQUERIMIENTOS

ESTUDIANTES

ANTONIO JAVIER ADRIÁN CAJAS
JORGE ASDRUBAL FLORES CHANCAY
ANA CRISTINA PALACIOS SURATI
CRISTHIAN ANDRÉS ZEAS BAQUE

1. Introducción	3
1.1 Propósito del software	3
1.2 Alcance	3
1.3 Definiciones, Acrónimos y Abreviaturas	3
1.4 Referencias	4
1.5 Visión General del Documento	4
2. Descripción General	4
2.1 Perspectiva del Producto	4
2.2 Funcionalidades del Producto	4
2.3 Características del Usuario	5
2.4 Restricciones	5
2.5 Suposiciones y Dependencias	5
3. Requisitos Funcionales	5
3.1 Selección de Palabras	5
3.2 Pistas	5
3.3 Arrastrar y Soltar Letras	6
3.4 Modos de Juego	6
3.4.1 Modo Solo	6
3.4.2 Modo Team	6
3.4.3 Modo Versus	6
3.5 Sistema de Vidas	6
3.6 Personalización de Avatares y Nombres	7
3.7 Selección de Dificultad y Temas Visuales	7
3.8 Retroalimentación y Aprendizaje de Errores	7
4. Requisitos No Funcionales	7
4.1 Rendimiento	7
4.2 Seguridad	7
4.3 Usabilidad	7
4.4 Confiabilidad	8
4.5 Compatibilidad	8
4.6 Mantenimiento	8
4.7 Otros Requisitos No Funcionales	8
5. Atributos de Diseño del Sistema	8
5.1 Arquitectura del Sistema	8
5.2 Interfaces del Sistema	8
5.3 Restricciones de Diseño	8
6. Prototipado	8

1. Introducción

1.1 Propósito del software

El juego que se va a desarrollar con el fin de combatir el "problema silencioso" de la dependencia de los autocorrectores en el aprendizaje de la ortografía, especialmente en niños de primaria, quienes se encuentran en una etapa crucial para desarrollar sus habilidades lingüísticas. Para esto hemos desarrollado una solución por medio de un juego interactivo de ahorcado, el cual brindará una experiencia divertida y atractiva que motive a los niños a aprender jugando. Con la ayuda del personaje dentro del juego el usuario podrá descifrar las palabras colocadas según el nivel de dificultad deseado y en el modo de juego que prefiera, lo que le ayudará a aprender la escritura correcta de diversas palabras, ampliar el vocabulario por medio de nuevas palabras y fortalecer la comprensión y el uso de lenguajes.

1.2 Alcance

El sistema responde de manera exclusiva a la necesidad de mejorar la ortografía y el vocabulario de los niños de primaria a través de un medio interactivo y educativo, por lo que la primera implementación solo incluye las funcionalidades mencionadas como requerimientos. Este proyecto se limita a las características necesarias para crear un juego de El Ahorcado que sea entretenido y pedagógico, dirigido a niños en edad escolar.

Es de destacar que las funcionalidades del juego se limitarán a todo lo acordado en las especificaciones del proyecto, por lo que este proyecto no tiene como objetivo abarcar todas las funcionalidades existentes de los juegos de El Ahorcado en el mercado. El desarrollo se centrará en:

- Proveer una interfaz gráfica amigable y adaptada para niños.
- Implementar modos de juego distintos (Solo, Team, y Versus) que promuevan el aprendizaje colaborativo y competitivo.
- Permitir la personalización de avatares y nombres de los jugadores, así como la elección de temas visuales y niveles de dificultad.
- Incluir pistas interactivas y retroalimentación constructiva para ayudar a los jugadores a aprender de sus errores sin penalizaciones severas.
- Asegurar que el juego sea compatible con sistemas operativos Windows, macOS y Linux, y que funcione con Java Runtime Environment (JRE).

1.3 Definiciones, Acrónimos y Abreviaturas

- **SRS:** Software Requirements Specification

- **UI:** Interfaz de Usuario
- **JFrame:** Contenedor principal de Swing
- **JPanel:** Panel de Swing para organizar componentes
- **JButton:** Botón de Swing
- **JLabel:** Etiqueta de Swing
- **TextField:** Campo de texto de Swing
- **JLayeredPane:** Panel de capas de Swing
- **TransferHandler:** Manejador de transferencia de Swing
- **JDialog:** Diálogo de Swing

1.4 Referencias

- González, G. R. (2012). La ortografía en el aula. Revista Káñina, 36(2), 181-190.
- Pascual-Gómez, I., & Carril-Martínez, I. (2017). Relación entre la comprensión lectora, la ortografía y el rendimiento: un estudio en Educación Primaria. Ocnos: Revista de estudios sobre lectura, 16(1), 7-17.
- [Guía de desarrollo en Java con Swing](#)
- [Patrones de diseño para juegos](#)

1.5 Visión General del Documento

Este documento está estructurado para proporcionar una descripción detallada y comprensiva del desarrollo del juego interactivo de El Ahorcado. Se divide en varias secciones clave que abarcan todos los aspectos esenciales del proyecto, incluyendo la descripción general del juego, requisitos funcionales y no funcionales, y los atributos de diseño del sistema.

2. Descripción General

2.1 Perspectiva del Producto

El juego de El Ahorcado es una aplicación independiente que ayuda a los niños a mejorar su ortografía mediante un enfoque lúdico y educativo. El juego se ejecuta en un entorno de escritorio utilizando la biblioteca Swing de Java, proporcionando una experiencia visualmente atractiva e interactiva que es adecuada para niños en edad escolar.

2.2 Funcionalidades del Producto

- Juego de adivinanza de palabras con pistas.
- Modos de juego: Solo, Team y Versus.

- Sistema de vidas y puntuación.
- Personalización de avatares y nombres de jugadores.
- Selección de dificultad y temas visuales.
- Retroalimentación constructiva y aprendizaje de errores.

2.3 Características del Usuario

- Niños de primaria (6-12 años).
- Padres y maestros interesados en herramientas educativas.
- Usuarios con conocimientos básicos de manejo de aplicaciones de escritorio.

2.4 Restricciones

- El juego debe ejecutarse en sistemas con Java Runtime Environment (JRE) instalado.
- Debe ser compatible con Windows, macOS y Linux.
- Interfaz y recursos gráficos limitados por la capacidad de la biblioteca Swing.

2.5 Suposiciones y Dependencias

- Los usuarios tendrán acceso a una computadora con JRE.
- Los usuarios podrán comprender instrucciones básicas en la interfaz del juego.
- El juego se distribuirá de forma gratuita en plataformas educativas.

3. Requisitos Funcionales

3.1 Selección de Palabras

Descripción: El juego debe permitir la selección de palabras de diferentes categorías (animales, objetos, lugares, etc.). **Entrada:** Nivel de dificultad seleccionado por el jugador. **Proceso:** Selección aleatoria de una palabra dentro de la categoría y dificultad elegida. **Salida:** Palabra oculta representada por guiones bajos en la interfaz del juego. **Criterios de aceptación:** La palabra seleccionada debe corresponder al nivel de dificultad y categoría especificada.

3.2 Pistas

Descripción: El juego debe proporcionar pistas a los jugadores cuando soliciten ayuda. **Entrada:** Solicitud de pista mediante clic en el botón de pistas. **Proceso:** Mostrar una pista relacionada con la palabra oculta. **Salida:** Pista visible en la interfaz del juego. **Criterios de aceptación:** La pista debe ser relevante y ayudar al jugador a adivinar la palabra.

3.3 Arrastrar y Soltar Letras

Descripción: Implementar la funcionalidad de arrastrar y soltar letras. **Entrada:** Acción de arrastrar una letra desde la caja de elementos. **Proceso:** Transferencia de la letra a la casilla correspondiente. **Salida:** Letra colocada en la casilla seleccionada. **Criterios de aceptación:** La letra debe posicionarse correctamente en la casilla y registrarse en el sistema.

3.4 Modos de Juego

3.4.1 Modo Solo

Descripción: El jugador debe adivinar la palabra en el nivel de dificultad que haya escogido. **Entrada:** Selección del modo Solo. **Proceso:** Presentar palabras y permitir al jugador adivinarlas con ayuda de pistas hasta que se quede sin vidas. **Salida:** Interfaz de juego con la palabra oculta y opciones de pista. **Criterios de aceptación:** El juego debe finalizar cuando el jugador adivine la palabra o pierda todas sus vidas.

3.4.2 Modo Team

Descripción: Hasta 4 jugadores pueden conectarse y colaborar para adivinar la palabra o frase correcta. **Entrada:** Selección del modo Team. **Proceso:** Los jugadores toman turnos para adivinar letras y pueden solicitar pistas. **Salida:** Interfaz de juego compartida con indicaciones de turnos y opciones de pista. **Criterios de aceptación:** El juego debe finalizar cuando se adivine la palabra o se pierdan todas las vidas. El puntaje se basa en letras adivinadas correctamente.

3.4.3 Modo Versus

Descripción: De 2 a 4 jugadores compiten para ver quién puede adivinar más palabras antes de quedarse sin vidas. **Entrada:** Selección del modo Versus. **Proceso:** Los jugadores adivinan palabras turno por turno; cada acierto suma puntos, cada error resta una vida. **Salida:** Interfaz de juego con indicaciones de turnos y puntuación. **Criterios de aceptación:** El juego debe finalizar cuando uno de los jugadores pierda todas sus vidas. El puntaje se basa en palabras adivinadas correctamente.

3.5 Sistema de Vidas

Descripción: El juego debe tener un sistema de vidas que se reduzcan con cada error. **Entrada:** Adivinanza incorrecta de una letra. **Proceso:** Restar una vida del total del jugador. **Salida:** Actualización del contador de vidas en la interfaz del juego. **Criterios de aceptación:** El juego debe terminar cuando se agoten las vidas del jugador.

3.6 Personalización de Avatares y Nombres

Descripción: Permitir que los jugadores seleccionen avatares y personalicen sus nombres. **Entrada:** Selección de avatar e ingreso del nombre del jugador. **Proceso:** Guardar y mostrar el avatar y nombre seleccionados en la interfaz del juego. **Salida:** Interfaz de juego con el avatar y nombre del jugador. **Criterios de aceptación:** El avatar y nombre del jugador deben mostrarse correctamente durante el juego.

3.7 Selección de Dificultad y Temas Visuales

Descripción: Ofrecer opciones para seleccionar el nivel de dificultad y el tema visual del juego. **Entrada:** Selección de nivel de dificultad y tema visual por el jugador. **Proceso:** Configuración del juego según las opciones seleccionadas. **Salida:** Interfaz de juego adaptada al nivel de dificultad y tema visual seleccionados. **Criterios de aceptación:** Las palabras, pistas y apariencia del juego deben ajustarse a las opciones seleccionadas.

3.8 Retroalimentación y Aprendizaje de Errores

Descripción: Proporcionar retroalimentación constructiva cuando los jugadores cometan errores. **Entrada:** Adivinanza incorrecta de una letra. **Proceso:** Mostrar un mensaje de retroalimentación constructiva. **Salida:** Mensaje de retroalimentación visible en la interfaz del juego. **Criterios de aceptación:** La retroalimentación debe ser clara y ayudar al jugador a mejorar su ortografía.

4. Requisitos No Funcionales

4.1 Rendimiento

- El juego debe cargarse en menos de 5 segundos.
- Las acciones del jugador deben tener una respuesta en menos de 200 milisegundos.

4.2 Seguridad

- Los datos del jugador deben manejarse de manera segura.
- El juego no debe permitir inyecciones de código ni accesos no autorizados.

4.3 Usabilidad

- La interfaz debe ser intuitiva y fácil de navegar.
- Incluir tutoriales y ayudas contextuales para nuevos usuarios.

4.4 Confiabilidad

- El juego debe tener un tiempo de actividad del 99%.
- Implementar mecanismos de recuperación en caso de fallos.

4.5 Compatibilidad

- Compatible con Windows, macOS y Linux.
- Requiere Java Runtime Environment (JRE) 8 o superior.

4.6 Mantenimiento

- El código debe ser modular y documentado.
- Facilitar actualizaciones y correcciones de errores.

4.7 Otros Requisitos No Funcionales

- Incluir soporte para múltiples idiomas (español, inglés).

5. Atributos de Diseño del Sistema

5.1 Arquitectura del Sistema

- Utilizar una arquitectura MVC (Modelo-Vista-Controlador) para separar la lógica del juego, la interfaz de usuario y el control de flujo.

5.2 Interfaces del Sistema

- Implementar interfaces claras entre los componentes del modelo, vista y controlador.

5.3 Restricciones de Diseño

- Limitar el uso de recursos gráficos y animaciones complejas debido a las capacidades de Swing.

6. Prototipado





