PARADIGMAS DE PROGRAMACIÓN



Grupo: PP02-10

Fecha inicio: 24 de junio de 2023

Fecha finalización: 24 de junio de 2023

Objetivo: Código fuente del proyecto

SCRUM TEAM

|  |  |  |
| --- | --- | --- |
| Apellidos y Nombres | Rol | Correo |
| Andrés Valbuena | Scrum Master | [avalbuena@poligran.edu.co](mailto:avalbuena@poligran.edu.co) |
| Andrés Valbuena | Product Owner | [avalbuena@poligran.edu.co](mailto:avalbuena@poligran.edu.co) |
| Andrés Valbuena | Analista | [avalbuena@poligran.edu.co](mailto:avalbuena@poligran.edu.co) |
| Andrés Valbuena | Administrador BD | [avalbuena@poligran.edu.co](mailto:avalbuena@poligran.edu.co) |
| Andrés Valbuena | Desarrollador | [avalbuena@poligran.edu.co](mailto:avalbuena@poligran.edu.co) |

ScrumMaster/ Stakeholder: Víctor Fabián Castro Pérez

DESARROLLO DEL SPRINT

Enlace a Github: <https://github.com/VALBUENA54/ProyectoUno.git>

PARADIGMA DE PROGRAMACION

ENTREGA FINAL

TUTOR: VICTOR FABIEN CASTRO PEREZ

GRUPO PP02 - 10

Andrés Valbuena ID 100197795

Universidad Politécnico Gran Colombiano Paradigma de Programación

Junio 27 del 2023

# Nombre del juego: ElectroDesafío

**Porqué su elección**

El nombre ElectroDesafío se escoge para resaltar el enfoque del juego en la electrónica y las preguntas relacionadas con este tema. De igual manera, el nombre sugiere una experiencia interactiva y desafiante para los jugadores. Está modalidad de juego ha demostrado ser exitoso en diferentes plataformas y ha capturado la atención de una amplia audiencia. El juego ofrece un desafío intelectual al tiempo que brinda diversión y emoción al intentar responder preguntas y acumular puntuación donde traerá jugadores interesados en electrónica y tecnología.

# Descripción detallada del juego

ElectroDesafío es un juego de preguntas pedagógico sobre electrónica desarrollado en lenguaje Java. Está diseñado para que los jugadores pongan a prueba y amplíen sus conocimientos en el campo de la electrónica a través de una serie de preguntas interactivas y educativas.

# Tipo de juego

ElectroDesafío un juego de preguntas y respuestas. Esto sería un juego de modo trivia clasificado en conocimientos electrónicos y tecnológicos

# Número de jugadores

El juego es para una sola persona para que sus jugadores para que sus jugadores puedan ser una única puntuación.

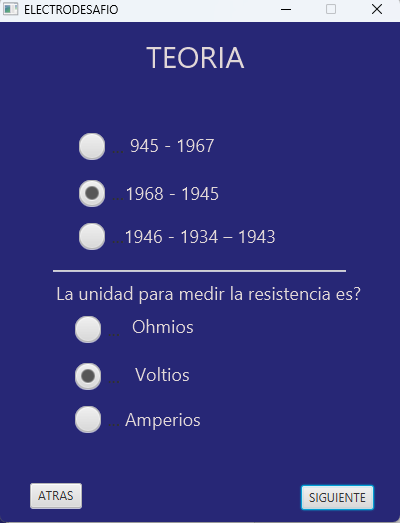
Es la cantidad de usuarios que quiera medir sus conocimientos y reforzarlos, permitiendo al jugador identificar sus conocimientos sobres sobre los conceptos de la electrónica y llevar su nivel de conocimiento mas alto.

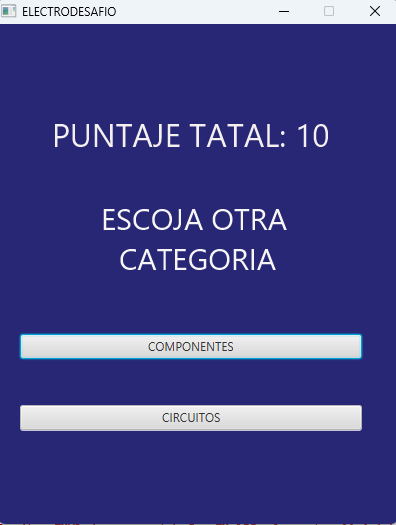
# Reglas del juego

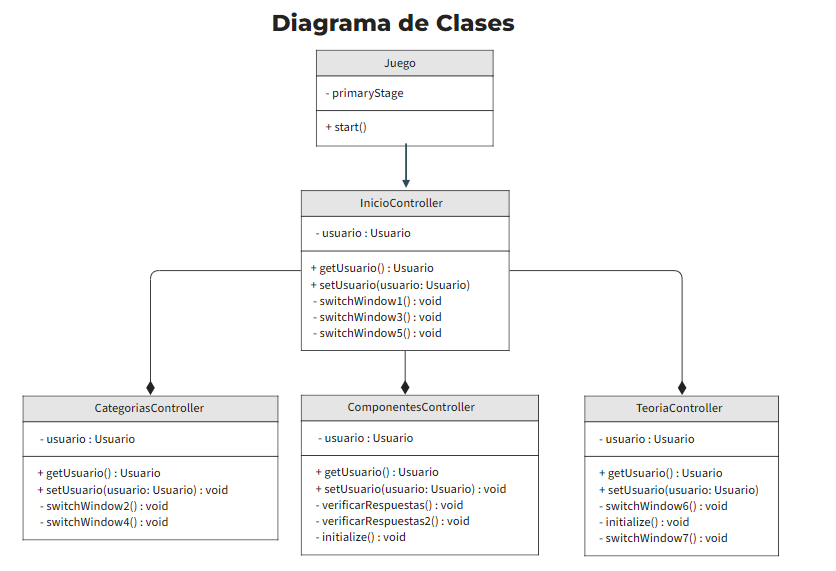
1. El jugador debe elegir una categoría de preguntas relacionadas con la electrónica, como circuitos, componentes y teoría básica.
2. Se muestra una pregunta con múltiples opciones de respuesta.
3. El jugador debe seleccionar la respuesta correcta entre las opciones proporcionadas.
4. Si la respuesta es correcta, el jugador suma puntos y pasa a la siguiente pregunta. Si la respuesta es incorrecta, el jugador pierde puntos y se le proporciona la respuesta correcta.
5. El juego continúa con una serie de preguntas hasta que se alcance un límite de tiempo predefinido o se hayan respondido todas las preguntas de la categoría seleccionada.
6. Al final del juego, se muestra la puntuación obtenida por el jugador.

# Imágenes

****



**** 



El diagrama de clases UML que representa la estructura y las relaciones entre las diferentes clases

y controladores de un juego. Aquí está una explicación de cada elemento del diagrama:

**Clase Juego**

Esta clase representa el juego en sí y tiene una referencia a la ventana principal (primaryStage) que se utiliza para mostrar la interfaz gráfica del juego.Tiene un método start() que se encarga de iniciar el juego.

**Clase InicioController**

Este controlador está asociado a la pantalla de inicio del juego,Tiene un atributo usuario de tipo Usuario que representa al usuario actual del juego,tiene un método getUsuario() para obtener el usuario actual, tiene un método setUsuario(usuario: Usuario) para establecer el usuario actual, tiene métodos switchWindow1(), switchWindow3(), y switchWindow5() que se utilizan para cambiar entre diferentes ventanas del juego.

**Clase CategoriasController**

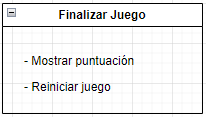
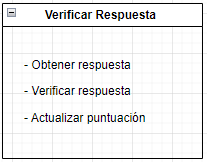
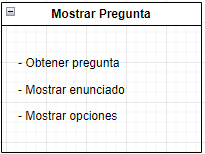
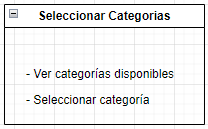
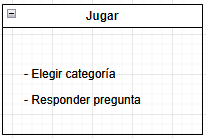
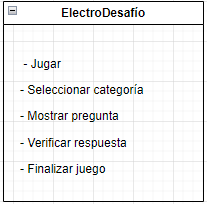
Este controlador está asociado a la pantalla de selección de categorías del juego, también tiene un atributo usuario de tipo Usuario que representa al usuario actual del juego, tiene un método getUsuario() para obtener el usuario actual, tiene un método setUsuario(usuario: Usuario) para establecer el usuario actual, tiene un método switchWindow2() que se utiliza para cambiar a otra ventana del juego.

**Clase TeoriaController**

Este controlador está asociado a la pantalla de teoría del juego, al igual que los controladores anteriores, tiene un atributo usuario de tipo Usuario para representar al usuario actual, tiene métodos getUsuario() y setUsuario(usuario: Usuario) para obtener y establecer el usuario actual, tiene métodos switchWindow6() y switchWindow7() que se utilizan para cambiar entre diferentes ventanas del juego, también tiene un método initialize() que se ejecuta al inicializar el controlador.

El diagrama muestra las relaciones entre las clases y los controladores utilizando líneas de conexión. Las flechas indican la dirección de la relación. Por ejemplo, la clase "InicioController" tiene una asociación con la clase "Usuario" porque tiene un atributo de tipo "Usuario". Del mismo modo, las clases "InicioController" y "TeoriaController" tienen asociaciones con la clase "Juego" porque se utilizan en el contexto del juego.

Diagrama de Casos de Uso



En el diagrama de casos de uso, el bloque principal es ElectroDesafío, que representa el sistema del juego. Tiene varios casos de uso, como Jugar, Seleccionar categoría, Mostrar pregunta, Verificar respuesta y Finalizar juego.

El caso de uso Jugar incluye subcasos de uso como Elegir categoría y Responder pregunta. El subcaso de uso Elegir categoría permite al jugador seleccionar una categoría específica de preguntas para jugar, mientras que el subcaso de uso Responder pregunta representa el proceso de responder una pregunta del juego.

El caso de uso Seleccionar categoría incluye subcasos de uso para Ver categorías disponibles y Seleccionar categoría. Estos subcasos de uso permiten al jugador ver las categorías disponibles y seleccionar una de ellas para jugar.

El caso de uso Mostrar pregunta tiene subcasos de uso para Obtener pregunta, Mostrar enunciado y Mostrar opciones. Estos subcasos de uso representan el proceso de obtener una pregunta aleatoria, mostrar el enunciado de la pregunta y mostrar las opciones de respuesta al jugador.

El caso de uso Verificar respuesta incluye subcasos de uso para Obtener respuesta, Verificar respuesta y Actualizar puntuación. Estos subcasos de uso permiten al sistema obtener la respuesta proporcionada por el jugador, verificar si es correcta y actualizar la puntuación del jugador.

El caso de uso Finalizar juego incluye subcasos de uso para Mostrar puntuación y

Reiniciar juego. Estos subcasos de uso representan el proceso de mostrar la puntuación final obtenida por el jugador y la opción de reiniciar el juego para jugar nuevamente.