

Universidad Nacional

Sede Regional Brunca



Programación II

Proyecto #1

“PREGUNTADOS”

Estudiante:

Kendall Fonseca Hidalgo

Cédula: 119200599

Profesor:

Carlos Carranza Blanco

I Ciclo, 2018

Tabla de contenidos

Introducción.....	3
Objetivos:.....	3
Descripción General del Sistema.....	4
Planteamiento del Problema.....	4
Solución Adaptada.....	5
ERS (Especificación de Requerimientos).....	5
Estándares Utilizados.....	7
Modelo Relacional.....	7
Diccionario de Datos.....	8
Conclusiones.....	10
Recomendaciones.....	10

Introducción

El sistema descrito aquí es una variación del conocido juego de mesa "Preguntados", adaptado para la competencia digital entre múltiples jugadores. A lo largo de este documento, se explorará la estructura del sistema, los objetivos del proyecto, la descripción general y los requerimientos específicos que guían su desarrollo.

Objetivos:

Objetivo General:

Diseñar y desarrollar un sistema interactivo basado en el juego de mesa "Preguntados" que permita la competencia entre hasta seis jugadores, enfocándose en la usabilidad, el manejo eficiente de bases de datos para el registro y seguimiento de jugadores y sus estadísticas, así como la persistencia y recuperación de información de partidas.

Objetivos Específicos:

- Desarrollar un módulo de registro de jugadores que permita a los usuarios ingresar al sistema y guardar sus datos (nombre, correo generado automáticamente...)
- Optimizar la gestión de preguntas y respuestas para garantizar la diversidad del contenido, permitiendo una fácil actualización y expansión de la cantidad de preguntas dentro del sistema.
- Implementar un entorno de juego para el desarrollo de la partida, donde cada jugador tenga la opción de seleccionar su ficha y una posición en el tablero, además de trabajar con distintas modalidades de juego que alteran el desarrollo de la partida según la elección del jugador (fácil, intermedio, difícil).
- Facilitar el seguimiento del rendimiento de los jugadores mediante un sistema de estadísticas, que permita visualizar la cantidad de preguntas respondidas y acertadas por cada categoría.
- Crear una interfaz gráfica amigable y atractiva que facilite la interacción del usuario y enriquezca su experiencia de juego, además de contar con una diversidad que permita la obtención de la información en la partida por cada jugador (personajes obtenidos, corona...)

Descripción General del Sistema

El sistema desarrollado es una adaptación digital del popular juego de mesa "Preguntados", diseñado para permitir la competencia entre hasta seis jugadores de manera local. El juego abarca seis categorías de conocimiento: Geografía, Historia, Arte, Ciencia, Entretenimiento y Deportes, y el objetivo es acumular personajes de cada categoría respondiendo correctamente a las preguntas. Esta versión digital busca replicar la experiencia del juego de mesa.

La base de datos se emplea para manejar toda la información relacionada con las partidas, incluyendo el registro y seguimiento de cada jugador, sus respuestas, y el progreso general en el juego. Una parte crucial de esta arquitectura es la entidad "PartidasJugadores", que vincula a los jugadores con sus respectivas partidas activas y completadas, facilitando operaciones como la reanudación de partidas y la consulta de estadísticas en partidas anteriores.

Desde el punto de vista técnico, el sistema centra su desarrollo en la interfaz de usuario, la cual está diseñada para ser intuitiva, garantizando que jugadores de todas las edades y habilidades puedan manejarla sin dificultades. La navegación por el sistema se ha simplificado para asegurar un acceso fluido a todas las funcionalidades esenciales, desde iniciar una nueva partida hasta revisar el historial de juegos. Se ha puesto un especial énfasis en la experiencia visual y la facilidad de uso, con el fin de hacer del juego una actividad accesible y agradable para todos.

Planteamiento del Problema y Solución adoptada

Planteamiento del Problema

El principal desafío de este proyecto es la creación de un sistema interactivo que emule la experiencia completa de un entorno de persistencia enfocado a los videojuegos, requiriendo de funcionalidades como el guardar, cargar y eliminar partidas además de un control de estadísticas, y de las propias preguntas dentro del juego, también considerando que el sistema debe permitir la participación de hasta seis jugadores. La complejidad radica en la necesidad de desarrollar un sistema de persistencia fluido y eficiente, asegurando que las interacciones con los datos de los jugadores y partidas se mantengan consistentes y seguros. Además, se debe garantizar una interfaz gráfica intuitiva y accesible que enriquezca la experiencia de juego sin comprometer la funcionalidad.

Solución Adoptada

Para resolver estos desafíos, se adoptará un enfoque modular en el desarrollo del software, utilizando JavaFX en conjunto con CSS para la interfaz de usuario, lo que facilita la creación de una experiencia de usuario visualmente atractiva y responsiva. La lógica del juego se gestionará a través de un modelo de clases bien definido que asegure el manejo eficiente de las reglas y el flujo del juego.

En cuanto a la gestión de datos, se implementará una base de datos Oracle para almacenar información relativa a los jugadores, partidas y el progreso de cada juego. Utilizando la entidad "PartidasJugadores", se optimizará la relación entre jugadores y partidas, permitiendo un manejo de datos eficiente y flexible, facilitando operaciones como la reanudación de juegos, y el seguimiento detallado del progreso del jugador.

ERS (Especificación de Requerimientos)

A continuación, se indicarán los requerimientos que se desarrollarán en la aplicación.

3.1 Levantamiento de Requerimientos

3.2.1 Requerimientos Funcionales

Módulo de Jugadores

REQ1. El sistema debe permitir la gestión integral de jugadores, incluyendo la capacidad para registrar nuevos participantes. Estos registros incluirán datos básicos como nombre y correo electrónico.

REQ2. Es esencial que el sistema permita a los jugadores ingresar y recuperar sus partidas guardadas, haciendo uso de su identificación única dentro del sistema.

REQ3. El sistema deberá proporcionar un historial detallado de las partidas de cada jugador, incluyendo estadísticas de victorias, preguntas respondidas y categorías en las que más destacan.

3.2 Requerimientos No Funcionales

REQ4. El sistema debe ser capaz de soportar hasta seis jugadores concurrentes sin degradación del rendimiento o la experiencia del usuario.

REQ5. Todos los aspectos de la interfaz de usuario y las interacciones del jugador deben ser intuitivas y accesibles para usuarios de todas las edades.

3.3 Requerimientos de Software

REQ6. El software debe funcionar en cualquier sistema operativo que soporte la última versión de Java y JavaFX, además de tener una integración para el levantamiento de la base de datos tanto Docker o OracleXE instalados en el sistema .

3.4 Requerimientos de Hardware

REQ7. Sistemas de 64 bits: equipo con procesador Intel o compatible a 2 GHz o superior, tanto para la base de datos como para el sistema.

3.5 Requerimientos de Desempeño

REQ8. El sistema debe garantizar que el tiempo de respuesta al devolver información desde la base de datos o durante la interacción con la interfaz no exceda los 5 segundos en condiciones normales de uso. Esto asegura una experiencia de usuario eficiente.

REQ9. El sistema debe ser capaz de manejar hasta 6 jugadores sin degradación significativa en el rendimiento, lo que incluye la carga y ejecución de preguntas, la actualización de estadísticas en tiempo real.

REQ10. Se requiere que el sistema realice regularmente copias de seguridad de los datos del juego, incluyendo estadísticas de jugadores y estados de partida, con el objetivo de no perder información.

Requerimiento	REQ01	Estado	No aplicable
Solicitante	Kendall Fonseca	Puesto	Estudiante
Descripción	Módulo	Jugadores	
El sistema fue diseñado para funcionar sin la necesidad de implementar un módulo que diferencie entre permisos de administrador y usuario común. Esto se debe a la naturaleza educativa y de demostración del proyecto, donde la funcionalidad principal se centra en la interacción con el juego y no en la gestión de usuarios diferenciados.			
Especificación			
Este requerimiento inicial contemplaba la posibilidad de implementar un sistema de control de accesos que permitiera una administración diferenciada de accesos para administradores y usuarios. Sin embargo, tras un análisis de las necesidades reales del proyecto, se determinó que la implementación de tal sistema no era necesaria. En su lugar, se priorizó el desarrollo de funcionalidades directamente relacionadas con la lógica del juego y la experiencia del usuario.			
Justificación			
La decisión de no implementar un sistema de separación de roles se			

justifica por el alcance y los objetivos del proyecto, que están orientados a demostrar habilidades de programación y diseño de interfaz, más que en la gestión de usuarios. Esto simplifica el desarrollo y reduce la complejidad del sistema, permitiendo un enfoque más directo en la funcionalidad del juego y mejorando la accesibilidad para todos los usuarios sin distinción.

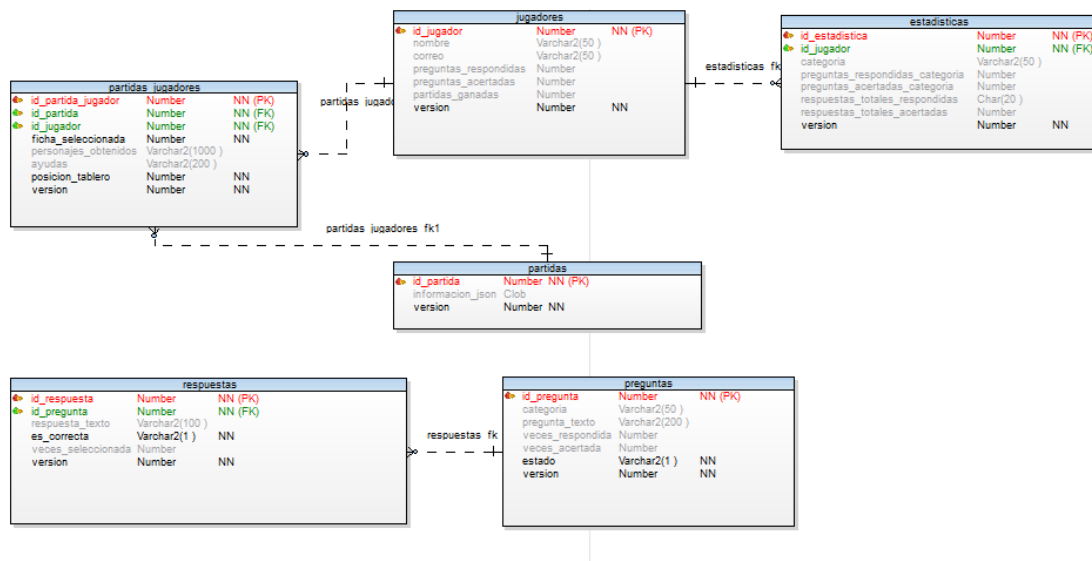
Estándares Utilizados

En el desarrollo de este proyecto, se optó por no adherirse estrictamente a un conjunto específico de estándares formales. La decisión de no utilizar un estándar predeterminado permitió una mayor flexibilidad en el diseño y la implementación del sistema, adaptando las soluciones a las necesidades específicas del proyecto y fomentando la innovación en el proceso de desarrollo.

Modelo Relacional

El modelo relacional del sistema está diseñado para soportar la dinámica del juego de preguntas y respuestas 'Preguntados'. El diseño incluye las siguientes entidades principales:

1. **Jugadores:** Almacena información sobre cada jugador, incluyendo su identificación, nombre, correo, y otras estadísticas relacionadas con su desempeño en el juego.
2. **Partidas:** Registra cada sesión de juego, asociando detalles específicos como la información del juego en formato JSON y la versión de la partida.
3. **PartidasJugadores:** Relaciona las partidas con los jugadores, capturando datos como la selección de fichas, los personajes obtenidos por cada jugador y su posición en el tablero.
4. **Preguntas y Respuestas:** Contienen los detalles de las preguntas formuladas durante el juego y las respuestas correspondientes, incluyendo cuántas veces ha sido seleccionada cada respuesta.
5. **Estadísticas:** Contiene detalles sobre el desempeño de los jugadores en distintas categorías, como preguntas respondidas, acertadas y totales por categoría.



Diccionario de datos

Jugadores:

- id_jugador: Número (Primary Key) - Identificador único para cada jugador.
- nombre: Varchar2(50) - Nombre completo del jugador.
- correo: Varchar2(50) - Dirección de correo electrónico del jugador.
- version: Number - Número de versión para control de concurrencia.

Partidas:

- id_partida: Number (Primary Key) - Identificador único para cada partida.
- informacion_json: Clob - Información detallada de la partida en formato JSON.
- version: Number - Número de versión para el control de concurrencia.

PartidasJugadores:

- id_partida_jugador: Number (Primary Key) - Identificador único para cada relación partida-jugador.
- id_partida: Number (Foreign Key) - Clave foránea que enlaza con la tabla Partidas.
- id_jugador: Number (Foreign Key) - Clave foránea que enlaza con la tabla Jugadores.

- `ficha_seleccionada`: Number - Identificador de la ficha seleccionada por el jugador.
- `personajes_obtenidos`: Varchar2(1000) - Registro de personajes obtenidos durante la partida.
- `Ayudas`: Varchar2(250) -Comodines disponibles para el uso del jugador,
- `posicion_tablero`: Number - Posición actual del jugador en el tablero.
- `version`: Number - Número de versión para el control de concurrencia.

Preguntas:

- `id_pregunta`: Number (Primary Key) - Identificador único de la pregunta.
- `pregunta_texto`: Varchar2(1000) - Texto de la pregunta.
- `categoria`: Varchar2(250) - Categoría de la pregunta.
- `veces_respondida`: Number - Número de veces que ha sido respondida la pregunta.
- `estado`: Varchar2(1) - Estado de la pregunta (activa o inactiva).
- `version`: Number - Número de versión para control de concurrencia.

Respuestas:

- `id_respuesta`: Number (Primary Key) - Identificador único de la respuesta.
- `id_pregunta`: Number (Foreign Key) - Clave foránea que enlaza con la tabla Preguntas.
- `respuesta_texto`: Varchar2(200) - Texto de la respuesta.
- `veces_seleccionada`: Number - Número de veces que ha sido seleccionada la respuesta.
- `version`: Number - Número de versión para el control de concurrencia.

Estadísticas:

- `id_estadistica`: Number (Primary Key) - Identificador único de la estadística.
- `id_jugador`: Number (Foreign Key) - Clave foránea que enlaza con la tabla Jugadores.
- `preguntas_respondidas`: Number - Número total de preguntas respondidas por el jugador.
- `preguntas_acertadas`: Number - Número total de preguntas acertadas por el jugador.

- respuestas_totales_categoría: Number - Total de respuestas por categoría.
- version: Number - Número de versión para control de concurrencia.

Conclusiones

Se deben indicar de forma clara a que conclusiones llegaron una vez finalizado el proyecto.

Una vez finalizado el proyecto se llegaron a las siguientes conclusiones:

- El sistema y la lógica del juego "Preguntados" no fueron implementados completamente. Este estado ha limitado la capacidad de realizar pruebas y la evaluación completa del funcionamiento del sistema.
- A pesar de que el diseño inicial prometía un sistema dinámico, la falta de implementación completa ha impedido que se materialice todo el potencial del proyecto. Como resultado, no se ha podido demostrar plenamente la efectividad de las funciones planeadas y su impacto en la experiencia del usuario.
- Aunque el desarrollo del sistema no se completó en su totalidad, el proyecto destaca por su interfaz gráfica altamente eficiente y estéticamente atractiva, la cual ha sido desarrollada con gran atención al detalle. Esta interfaz no solo mejora la usabilidad, sino que también enriquece la experiencia visual del usuario, estableciendo una base sólida para futuras mejoras y desarrollos del juego.
- El diseño del sistema ha probado ser escalable, proporcionando una base sólida para futuras mejoras y la incorporación de nuevas funcionalidades.

Recomendaciones

- Completar el desarrollo: Es esencial priorizar la finalización de los requerimientos pendientes y la lógica de juego para asegurar que el sistema sea completamente funcional.
- Pruebas intensivas: Se recomienda realizar pruebas intensivas para asegurar que todos los componentes del sistema funcionen de manera coherente y eficiente. Las pruebas deben incluir escenarios de juego múltiples para verificar la estabilidad y la respuesta del sistema bajo diferentes condiciones de uso.