

Universidad Mariano Galvez
Facultad de Ingeniería, Matemáticas y Ciencia Físicas
Ing. Michael Rodolfo Asturias L
Programación III



Proyecto Final

Documentación

Integrantes:

0900-22-9686 | Guillermo Antonio Ortiz López
0900-22-149 | Amanda Lorena Garcia Castellanos

Guatemala, 4 de Mayo del 2024

Indice

Documentación del Proyecto	2
Descripción General.....	2
Archivos Principales	2
Funcionalidad General	3
Interacción entre Componentes.....	3
Conclusión	3

Documentación del Proyecto

Descripción General

El proyecto "ProyectoFinal" es una aplicación de gestión de datos relacionados con partidos de fútbol y jugadores. Utiliza una combinación de estructuras de datos como listas enlazadas, tablas hash y árboles AVL para almacenar y organizar la información de manera eficiente.

Archivos Principales

1. Partidos.cs

- **Descripción:** Define la clase `Partidos`, que representa un partido de fútbol con atributos como la fecha, la asistencia, los nombres de los equipos, los goles marcados, la posesión y otros detalles relacionados con el partido.
- **Funcionalidad:** Permite crear instancias de partidos con información específica y acceder a sus atributos.

2. MiListaEnlazada.cs

- **Descripción:** Define la clase `MiListaEnlazada<T>`, que implementa una lista enlazada genérica.
- **Funcionalidad:** Proporciona métodos para agregar elementos al final de la lista, eliminar elementos y recorrer la lista.

3. HashTable.cs

- **Descripción:** Contiene la implementación de una tabla hash para almacenar objetos de tipo `Partidos`.
- **Funcionalidad:** Permite agregar, buscar, modificar y eliminar partidos utilizando una función hash para indexarlos de manera eficiente.

4. NodoLista.cs

- **Descripción:** Define la clase `NodoLista`, que representa un nodo en una lista enlazada.
- **Funcionalidad:** Proporciona métodos para acceder y modificar los nodos de la lista enlazada.

5. ListaJugador.cs

- **Descripción:** Define la clase `ListaJugador`, que representa una lista enlazada de jugadores.
- **Funcionalidad:** Permite almacenar y manipular objetos de tipo `DataJugadores` utilizando nodos enlazados.

6. DataJugadores.cs

- **Descripción:** Define la clase `DataJugadores`, que representa los datos de un jugador de fútbol.
- **Funcionalidad:** Permite crear instancias de jugadores con información como el nombre completo, la edad, el club actual, etc.

7. Team_Registration.cs

- **Descripción:** Contiene la implementación de la interfaz `Comparador` y la clase `Team_Registration`, que representa el registro de un equipo.
- **Funcionalidad:** Permite comparar equipos y proporciona métodos para acceder y modificar la información del equipo.

8. NodoAvl.cs

- **Descripción:** Define la clase `NodoAvl`, que extiende la clase `Nodo` e implementa nodos para un árbol AVL.

- **Funcionalidad:** Proporciona nodos específicos para un árbol AVL con un factor de equilibrio.

Funcionalidad General

El proyecto permite:

- Registrar partidos de fútbol con información detallada.
- Almacenar y gestionar datos de jugadores, incluyendo su nombre, edad y club actual.
- Utilizar estructuras de datos eficientes como listas enlazadas y tablas hash para organizar y acceder a la información.
- Comparar equipos registrados utilizando la interfaz `Comparador`.
- Modificar, buscar y eliminar partidos utilizando una tabla hash.

Interacción entre Componentes

1. La clase `Partidos` se utiliza para representar los detalles de un partido de fútbol.
2. La clase `MiListaEnlazada` proporciona una lista enlazada genérica para almacenar datos de manera ordenada.
3. La clase `HashTable` utiliza una tabla hash para almacenar y gestionar partidos de fútbol de manera eficiente.
4. Las clases `ListaJugador` y `DataJugadores` se utilizan para gestionar datos de jugadores de fútbol.
5. La clase `Team_Registration` y la interfaz `Comparador` permiten comparar equipos registrados.

Conclusión

El proyecto "ProyectoFinal" proporciona una solución completa para gestionar datos relacionados con partidos de fútbol y jugadores, utilizando una variedad de estructuras de datos y técnicas de programación eficientes.