

Trabajo Practico 3

Alumno: Octavio Bill Zito

División: 2°E

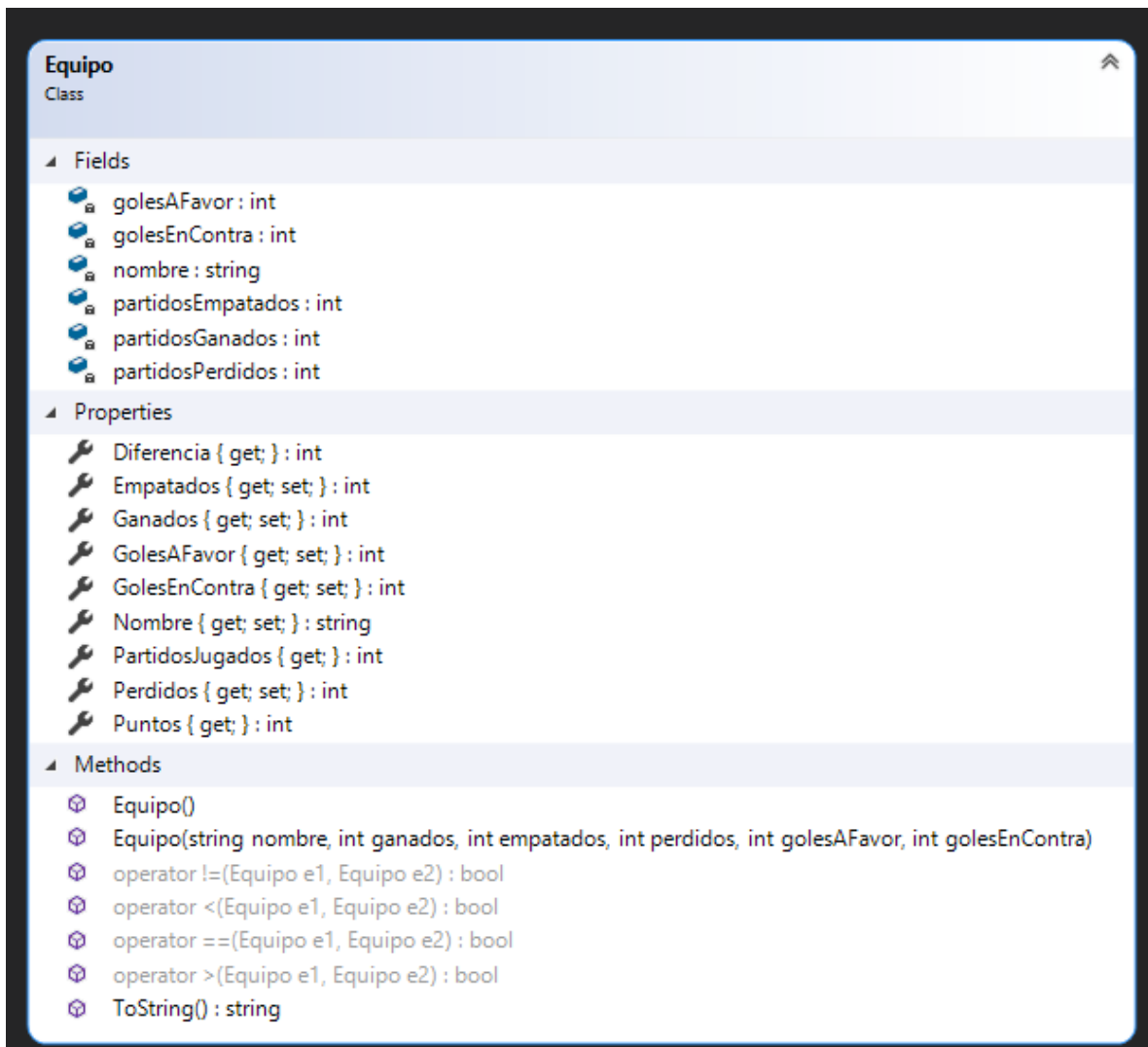
Proyecto: Torneo de Futbol

Breve explicación sobre el funcionamiento de la aplicación

La solución cuenta con 1 proyecto de consola llamado "Test" en la cual se prueban funciones básicas de la clase equipo y torneo. 1 proyecto de clases llamado "Entidades" que cuenta con todas las clases necesarias para el funcionamiento de la aplicación. 1 proyecto de unit test en la cual se testean la inicialización de un equipo, de un torneo, poder agregar y quitar equipos a un torneo, abrir y guardar el torneo en formatos como xml, json y txt. Por ultimo cuenta con 1 proyecto de Windows form que contiene 3 formularios, el principal donde se mostrara la tabla de equipos, la fecha actual, la capacidad máxima de equipos, el nombre del torneo y una tool strip que cuenta con 3 menús desplegables, el menú de archivo en el cual podemos abrir, guardar y guardar como un archivo, el menú de equipos que nos permite agregar, modificar y eliminar un equipo y por último el menú de torneo que nos permite avanzar de fecha, comparar dos equipos, resetear el torneo y ver información del torneo. El formulario equipos nos permite agregar o modificar un equipo dependiendo la opción seleccionada en el formulario principal, no permite cargar datos inválidos como goles en contra o a favor con un total de 0 partidos, tampoco cargar más partidos que fechas actuales ni cargar letras en vez de numero o dejar algún text box vacío. Y por último el formulario torneo en el cual accedemos al momento de ir a "info torneo" en el formulario principal, este nos permite modificar el nombre, la cantidad de fechas máximas y de equipos máximos, no se puede ingresar una cantidad de fechas menor a la fecha actual, ni tampoco una cantidad de equipos menor a la actual, no acepta text box vacíos y excepto el nombre, los demás text box no permiten letras.

La idea de la aplicación es poder crear un torneo agregando la cantidad de equipos que uno desee siempre y cuando no supere la cantidad máxima de equipos, y poder simular fechas en la cual de una manera aleatoria se generan goles a favor y en contra entre 0 a 5 por partido si los goles a favor en esa fecha fueron mayor que los goles en contra se le cuenta como una victoria sumando 3 puntos, en caso de igualdad se cuenta como un empate sumando 1 punto y por ultimo si los goles en contra son mayor que los goles a favor se le cuenta como una derrota, el data grid se actualiza solo ordenando los equipos por mayor puntaje. La aplicación nos permite guardar estos datos en 3 formatos distintos (xml, json y txt) y también nos permite cargar estos datos manteniendo los datos de los equipos.

Diagrama de la clase Equipo:



La propiedad “Diferencia” retorna la resta entre los goles a favor con los goles en contra.

La propiedad “PartidosJugados” retorna la suma de partidos ganados, empatados y perdidos.

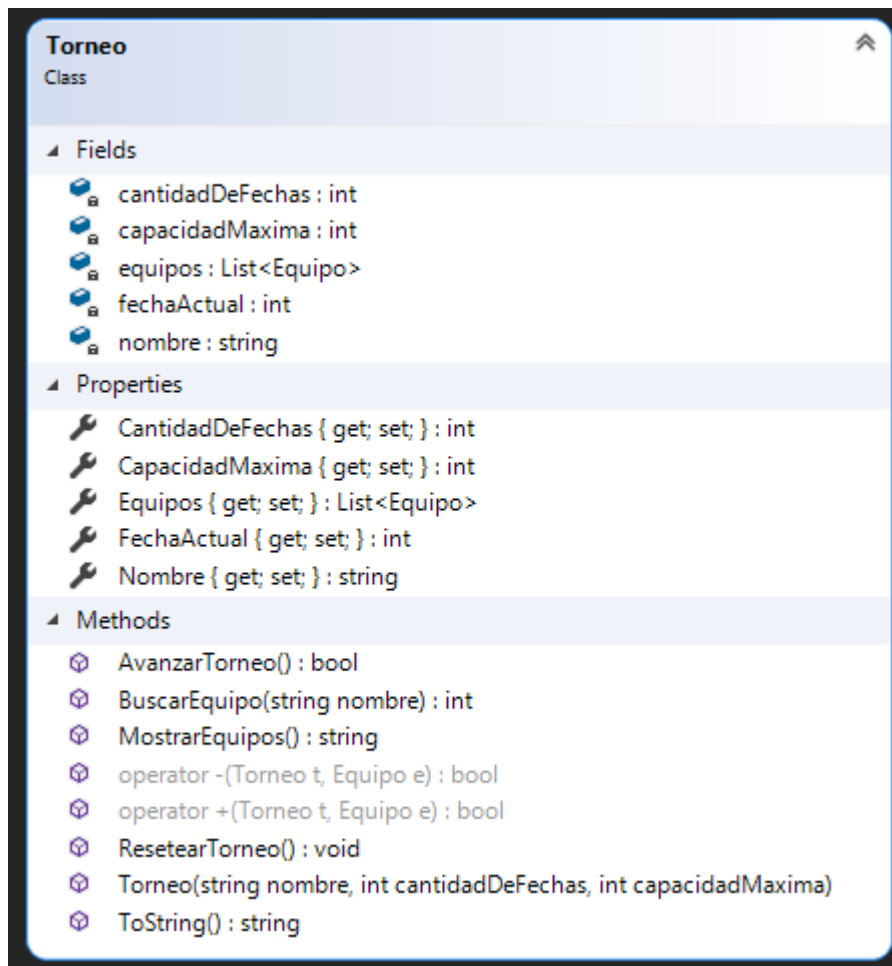
La propiedad “Puntos” retorna la multiplicación de partidos ganados * 3, más la suma de partidos empatados.

Las demás propiedades retornan y permiten setear los atributos con su mismo nombre.

El operador == compara dos equipos por su nombre, y el operador > verifica si el e1 pasado por parámetro tiene más puntos que el e2, en caso de igualdad compara con la propiedad “Diferencia”.

La sobrecarga del método “ToString” retorna el nombre, partidos jugados, ganados, empatados, perdidos, puntos, goles a favor, goles en contra y diferencia.

Diagrama de la clase Torneo:



Las propiedades retornan los atributos que su nombre indica.

El operador + permite agregar un equipo a la lista equipos siempre y cuando no se encuentre otro equipo con el mismo nombre, el operador – quita un equipo de la lista, si es que este equipo se encuentra.

El método “BuscarEquipo” recibe un string, este método busca a través del nombre un equipo en la lista y retorna su posición en la lista.

El método “MostrarEquipos” recorre toda la lista de equipos y con la sobrecarga del operador ToString de los equipos, los muestra, retorna toda la lista de los equipos en un string

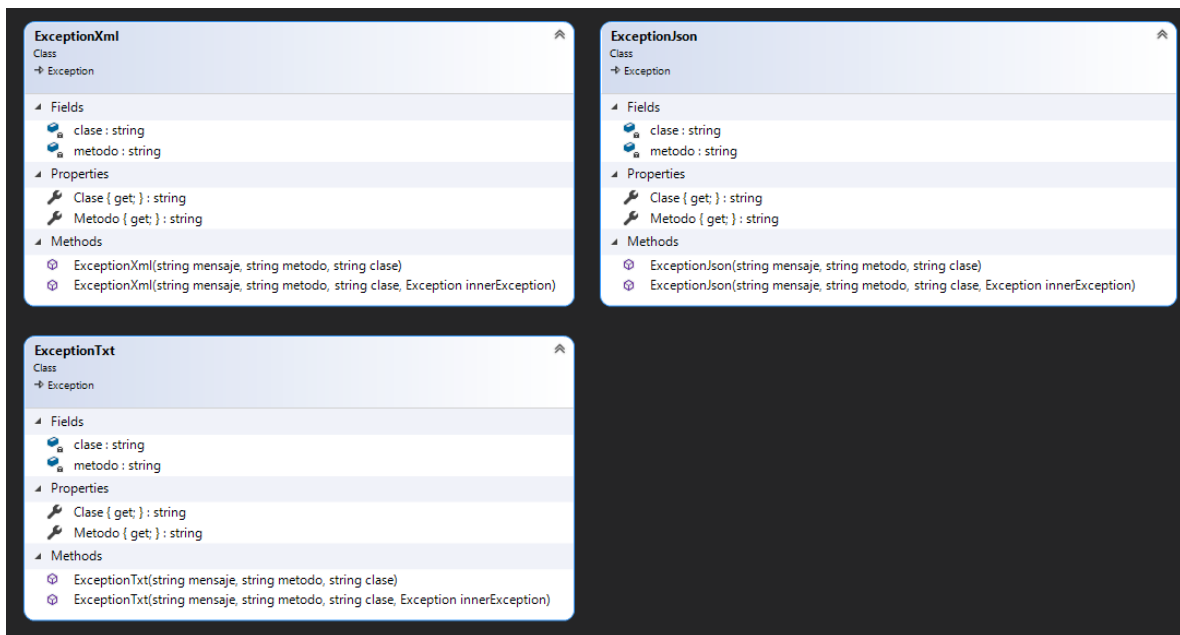
El método “AvanzarTorneo” verifica si la fecha actual es menor a la fecha máxima y que la cantidad de equipos sea mayor a 1, este método genera de manera aleatoria con el uso de la clase Random, golesAFavor y golesEnContra para cada equipo de la lista y a su vez verifica que si los golesAFavor generados son mayor a los golesEnContra, se le suma una victoria al equipo, en caso de igualdad, se le suma un empate y en caso que los golesEnContra sean mayores a los golesAFavor se le suma una derrota, por ultimo agrega estos goles a las estadísticas del equipo. Al

terminar el loop suma 1 la fecha actual. Retorna true si las primeras validaciones fueron correctas o false en caso contrario.

El método “Resetear Torneo” setea la fecha actual en 0 y recorre la lista de equipos y le setea todos sus atributos en 0, solo mantiene el nombre.

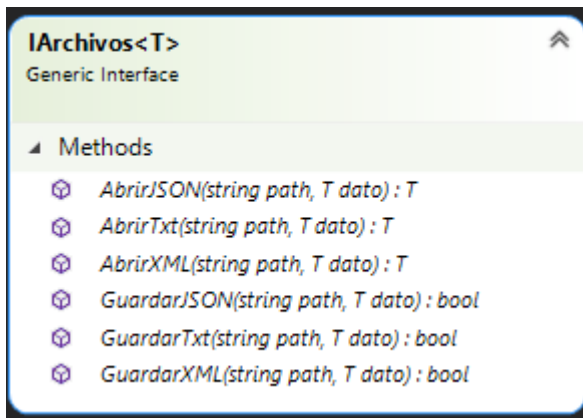
La sobrecarga del método “ToString” muestra el nombre del torneo, la cantidad de equipos máximos, la fecha actual, la cantidad de fechas y todos los equipos de la lista. Esto lo retorna en un string.

Diagrama de clases de las excepciones:



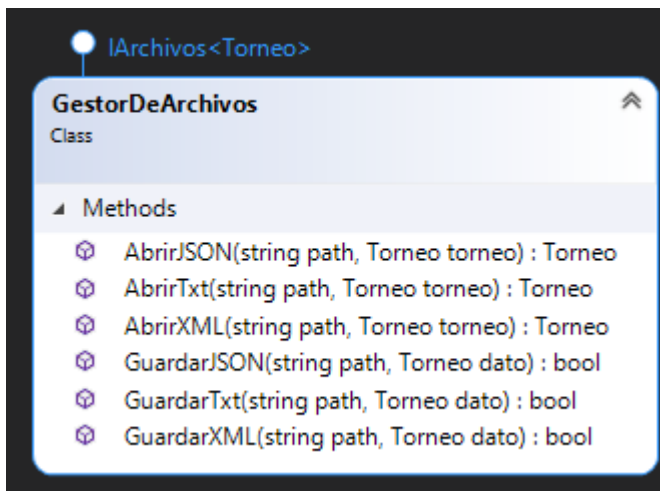
Las 3 clases comparten la misma estructura, las propiedades retornan los atributos a las que su nombre corresponde

Diagrama de clase de IArchivos<T>:



Posee un genérico “T” que tiene como restricción que sea de la clase Torneo.

Diagrama de clase de GestorDeArchivos:



Incluye la interfaz IArchivos con el genérico como clase Torneo.

El método “AbrirJSON”, recibe un string llamado path, la cual es la ruta del archivo json a abrir, y recibe un torneo (el actualmente) para pasar las restricciones (nombre, cantidad de fechas y equipos) al nuevo torneo con los equipos cargados, retorna el nuevo torneo.

El método “AbrirTxt”, recibe un string llamado path, la cual es la ruta del archivo txt a abrir, y recibe un torneo (el actualmente) para pasar las restricciones (nombre, cantidad de fechas y equipos) al nuevo torneo con los equipos cargados, retorna el nuevo torneo. La forma de abrir el archivo será mediante el un slip con el carácter ‘,’.

El método “AbrirXML”, recibe un string llamado path, la cual es la ruta del archivo xml a abrir, y recibe un torneo (el actualmente) para pasar las restricciones (nombre, cantidad de fechas y equipos) al nuevo torneo con los equipos cargados, retorna el nuevo torneo.

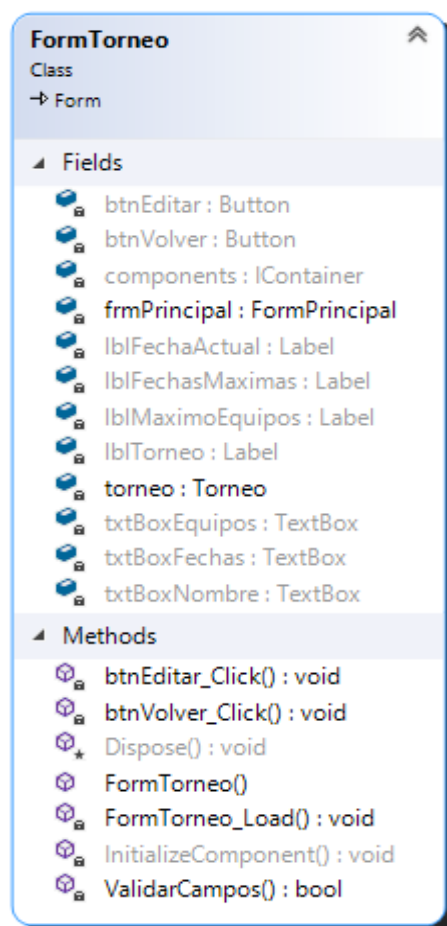
El método “GuardarJSON” recibe un string llamado path, la cual es la ruta del archivo a guardar en formato json, y recibe el Torneo a guardar, si todo salió bien retorna true.

El método “GuardarTxt” recibe un string llamado path, la cual es la ruta del archivo a guardar en formato txt, y recibe el Torneo a guardar, si todo salió bien retorna true. La forma de guardar será

"Nombre PJ G E P GF GC DG Pts" como primera línea que sería el encabezado del archivo. Luego guardara los datos del equipo en ese orden.

El método “GuardarXML” recibe un string llamado path, la cual es la ruta del archivo a guardar en formato xml, y recibe el Torneo a guardar, si todo salió bien retorna true.

Diagrama de clase FormTorneo



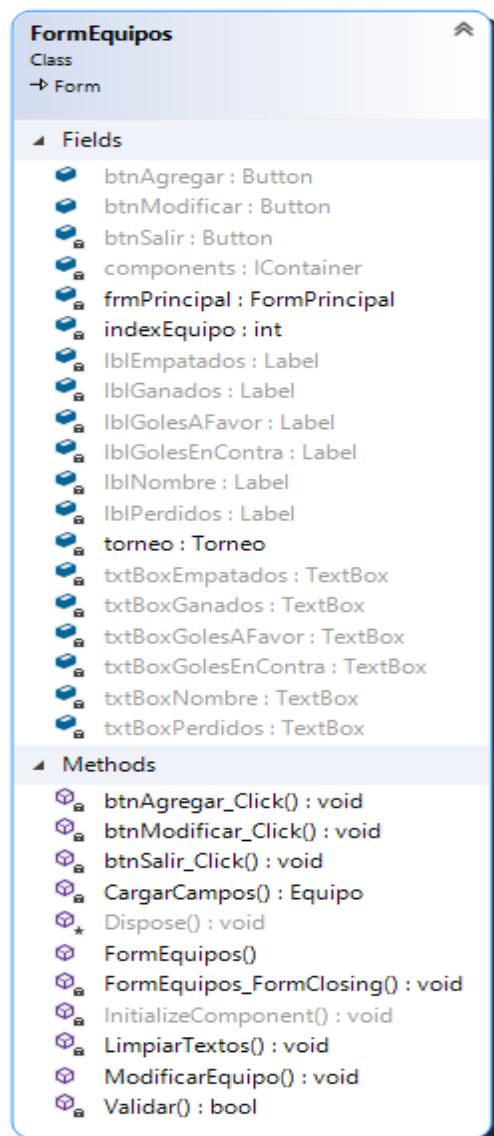
El método “ValidarCampos” retorna un bool false si algún campo este vacío o true en caso contrario.

El método “btnEditar_Click” modifica el nombre, la cantidad de fechas y la capacidad máxima del torneo al hacer click sobre el botón editar. Valida que los campos con el método “ValidarCampos” e intenta parsear el texto de los textBox de cantidad de fechas y cantidad máxima de equipos a int, a su vez valida que la cantidad de fechas ingresada sea mayor o igual a la fecha actual y que la capacidad máxima sea mayor o igual al count de lista de equipos del torneo.

El método “btnVolver_Click” cierra el formulario al hacer click en el botón volver

El método “FormTorneo_Load” carga todos los textBox con los datos del torneo al cargar el formulario.

Diagrama de la clase FormEquipos:



El método “Validar” retorna un bool true en caso que todos los textBox no estén vacios ni sean null caso contrario retorna false

El método "CargarCampos" retorna un equipo nuevo en caso de poder cargar todos los valores de los textBox a un equipo caso contrario retorna un equipo con valor null.

El método "btnAgregar_Click" utiliza el método "CargarCampos" y verifica que ese equipo nuevo no sea null, en caso de ser así intenta agregar el equipo con la sobrecarga + de la clase Torneo, en caso de poder agregarlo muestra un mensaje y actualiza el datagrid con el método "ActualizarForm" del form principal.

El método "BtnSalir_Click" crea un mensaje en pantalla con dos botones para verificar si deseamos salir, en caso de que el resultado sea si, cierra el formulario.

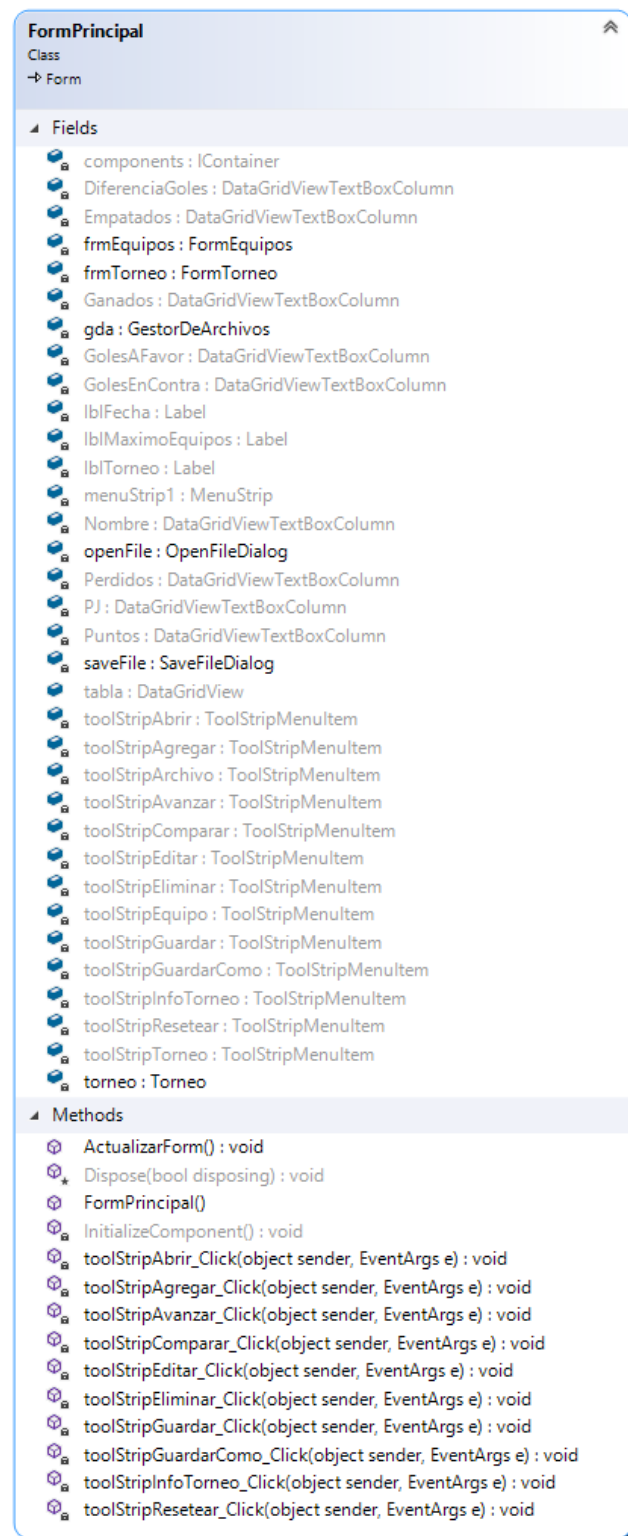
El método "LimpiarTextos" pone todos los textBox en vacíos.

El método "ModificarEquipo" recibe un int que luego es seteado en el atributo indexEquipo. Para poder obtener todos los datos del equipo a través de su posición en la lista. Setea todos los textBox con la información del equipo.

El método "btnModificar_Click" utiliza el método "CargarCampos" y si el nuevo equipo es distinto de null lo carga en la posición que tenga el atributo indexEquipo en la lista de equipos del atributo torneo.

El método "FormEquipos_FormClosing" pone btnAgregar y btnModificar su enable en false y llama al método "LimpiarTextos"

Diagrama de clase del FormPrincipal:



El método "ActualizarForm" limpia las filas de la tabla, y carga los valores nuevamente con la lista de equipos de torneo, luego se utiliza el método sort de tabla, ordenándolos de forma descendiente por la columna 8 que es la de puntos. Por último, se cargan los label de fecha, torneo y máximo equipos con los atributos que tienen torneo.

El método "toolStripGuardarComo_Click" llama al método "ShowDialog" del atributo saveFile y verifica que el FileName de saveFile sea distinto a vacío, luego verifica que cual es la extensión del archivo y llama al método correspondiente de guardar del atributo gda dependiendo su extensión. Por último actualiza el formulario y setea el filename de openFile con el de saveFile.

El método "toolStripGuardar_Click" es igual al método "toolStripGuardarComo_Click" con la diferencia de verificar la extensión con el saveFile lo verifique con openFile, y el parámetro a pasar al momento de guardar también es el de openFile su FileName.

El método "toolStripAbrir_Click" es igual al método "toolStripGuardarComo_Click" pero en vez de llamar a los métodos Guardar de gda, llama al método Abrir de gda dependiendo su extensión.

El método "toolStripAgregar_Click" ajusta el btnAgregar del atributo frmEquipos el enable en true y luego llama al método "ShowDialog" de frmEquipos.

El método "toolStripEditar_Click" verifica que de la tabla el currentCell sea distinto a null en caso contrario muestra un mensaje, si es distinto a null se llama al método "BuscarEquipo" de torneo pasándole como valor el nombre del equipo de la fila seleccionada de la tabla. Por último del atributo frmEquipos el btnModificar su enable en true, y se llama al método "ModificarEquipo" de frmEquipos pasándole el index del equipo que retorna "BuscarEquipo" y también llamar a su "ShowDialog".

El método "toolStripEliminar_Click" este método hace lo mismo que el método "toolStripEditar_Click" pero una vez conseguido el equipo lo elimina del torneo en caso de que lo pueda eliminar muestra un mensaje y utiliza "ActualizarForm".

El método "toolStripAvanzar_Click" llama al método "AvanzarTorneo" de torneo si es retorna true ese método se llama al método "ActualizarForm" en caso contrario se muestra un mensaje de error.

El método "toolStripResetear_Click" llama al método "ResetearTorneo" de torneo y luego llama al método "ActualizarForm".

El método "toolStripComparar_Click" valida que de la tabla la propiedad SelectedRows sea mayor a 0 y menor a 3 en caso contrario muestra un error, luego guarda la fila selecciona a través del atributo SelectedRows[0] en un DataGridViewRow y otra pero con SelectedRows[1]. Llama al método "BuscarEquipo" de torneo y le pasa como parametro las filas obtenidas anteriormente la celda 0 trae el valor y utiliza el método "ToString" para pasar a string el valor recibido, luego dentro de un if compara con el operador > el index del equipo 1 con el index del equipo 2 de la lista de equipos del atributo torneo. Si el valor es true muestra un mensaje en caso de ser false muestra otro mensaje.

El método "toolStripInfoTorneo_Click" llama al método "ShowDialog" del atributo frmTorneo.

