|  |
| --- |
| **UML LENGUAJE UNIFICADO DE MODELADO** |

|  |
| --- |
|  |

|  |  |
| --- | --- |
| Nombre | Coria Eric |
| Curso | 5° 3ra Computación |
| Video Juego | FootBall Manager 2012 |
| Plataforma | Playstation |

|  |
| --- |
| **HISTORIA** |

El juego simula la gestión de un equipo de fútbol, en el cual el jugador tiene que hacerse manager de este equipo. La versión del 2012 te permite tomar el control de cualquier club de fútbol de más de 50 países de todo el mundo, gestiona donde quieras, cuando quieras, añadir o quitar naciones jugables en tus partidas guardadas todas las veces que quieras, gestionar otro país al comienzo de la próxima temporada, por lo que no tienes por qué decidir todas las naciones jugables desde el principio de tu carrera, entre otras funciones nuevas.

|  |
| --- |
|  |

El juego es de simulación, hay muchas caracteristicas que permiten al jugador controlar a los pases de jugadores, contratar nuevos, etc.

|  |
| --- |
| **DESCRIPCIÓN DEL ÁREA SELECCIONADA** |

*Función del Club/Equipo*, me base en lo que tiene un Club de fútbol, como los Jugadores, Empleados, Directivos, los colores del mismo y la cancha que posee. También en los campeonatos o ligas que juega, etc.

|  |
| --- |
|  |

El área seleccionada consiste en una simple herencia en la cual hay cinco clases, que de alguno de ellos posee clases derivadas a su vez.

|  |
| --- |
| DIAGRAMA DE CLASE |

|  |
| --- |
|  |

|  |
| --- |
| **DESCRIPCIÓN DEL DIAGRAMA** |

Se inicia con la superclase ***Clase Abstracta Club*** del cual heredan varios métodos y atributos las demás clases derivadas.

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS | TIPO DE DATO | ACCESO |
| NombreClub | string | private |
| ApodoClub | string | private |
| ColorClub | Int | private |
| Jugador | int | private |
| Dirigente | Int | private |
| Hinchada | float | private |

+ TotalPuntos(): void   
Método de acceso *public* con procedimiento void cuya función es mostrar los puntos obtenidos con los partidos jugados, gracias a esto, dependiendo que puntaje posea el club, podrá participar de las ligas.  
  
+ ParticiparLiga(): void  
Método de acceso *public* con procedimiento void cuya función es participar en ligas.  
  
+ ParticiparCampeonato(): void  
Método de acceso *public* con procedimiento void cuya función es participar en el campeonato.  
  
  
  
 ***Clase Liga***

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS | TIPO DE DATO | TIPO DE ACCESO |
| NombreEquipo | string | private |
| CantidadEquipo | string | private |
| NombreLiga | string | private |
| NombreCopa | string | private |

+ ComenzarLiga(): void   
Método de acceso *public* con procedimiento void cuya función comenzar la liga. (CantidadEquipo es el atributo encargado de seleccionar los equipos que van a participar en la liga. Ej: Liga de 8 equipos)  
  
***Clase Campeonato***

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS | TIPO DE DATO | TIPO DE ACCESO |
| NombreEquipo | string | private |
| CantidadEquipo | string | private |
| NombreCampeonato | string | private |
| NombreCopa | string | private |

+ ComenzarCampeonato(): void   
Método de acceso *public* con procedimiento void cuya función comenzar la liga. (CantidadEquipo es el atributo encargado de seleccionar los equipos que van a participar en el campeonato. Ej: Liga de 8 equipos)  
  
***Interfaz Detalle***  
Clase interfaz encargada de ***mostrar*** los detalles del Club como:

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS | TIPO DE DATO | TIPO DE ACCESO |
| colorEquipo | string | private |
| banderaEquipo | int | private |
| hinchadaEquipo | float | private |

***Interfaz Estadio***  
Clase interfaz encargada de ***mostrar*** los detalles del Club como:

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS | TIPO DE DATO | TIPO DE ACCESO |
| Capacidad | doublé | private |
| Dirección | string | private |
| PersonalLimpieza | int | private |
| NombreEstadio | string | private |

Es una clase abstracta de la cual DirectorTécnico, Presidente y Jugador heredan sus atributos y métodos.  
  
***Clase Personal***Clase abstracta que funciona como superclase para que DirectorTécnico, Presidente y Jugador hereden sus atributos y métodos.

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS | TIPO DE DATO | TIPO DE ACCESO |
| Nombre | string | private |
| Cargo | string | private |
| Edad | string | private |

+ ObtenerDato(): voidMétodo de acceso *public* con procedimiento void cuya función es obtener atributos mediante este, ya que los atributos están en acceso private. ***Clase Jugador***Objeto jugador que se encarga de entrenar y jugar los partidos.

|  |  |  |
| --- | --- | --- |
| ATRIBUTOS | TIPO DE DATO | TIPO DE ACCESO |
| NumeroCamiseta | int | private |
| ApodoJugador | string | private |
| PosicionJugador | string | private |

+ Defender(): void  
Jugadores que se encargan de defender el arco para que el equipo rival no convierta gol.  
  
+ Atacar(): void  
Jugadores que se encargan de meter gol al equipo rival para ganar el partido.  
  
+ Atajar(): voidJugador que se encarga de atajar los pelotazos que recibe del equipo rival para no perder el partido ***Clase DirectorTécnico***Objeto DirectorTécnico que se encarga de entrenar a los jugadores y crear estrategias para mejorar el equipo.  
  
+ DirigirJugador(): void  
Tiene como función dirigir las posiciones de los jugadores para controlar la pelota, entre otras cosas.   
  
+ CrearEstrategia(): void  
Tiene como función crear estrategias de jugadas para los jugadores atacantes y defensores (Ej: posicionarse para que al otro equipo le cobren offside   
  
  
+ CrearFormacion(): void  
Tiene como función posicionar a los jugadores con formaciones ya hechas para estabilizar correctamente al equipo.  
 ***Clase Presidente***Objeto Presidente que se encarga de administrar todo el club.  
  
+ ComprarJugador(): void  
Tiene como función la compra de jugadores.  
+ VenderJugador(): void  
Tiene como función la venta de jugadores.  
+ HacerNegocio(): voidTiene como función negociar para hacer partidos amistosos, entre otras cosas.

|  |
| --- |
| **RELACION ENTRE CLASES Y OBJETOS** |

|  |
| --- |
| ***1.*** |

|  |  |  |  |
| --- | --- | --- | --- |
| INTERFAZ | SUPERCLASE | RELACIÓN | METODO |
| Detalle | Club | RelaciónInterfaz | Todos |
| Estadio | Club | RelaciónInterfaz | Todos |

|  |
| --- |
| ***2.*** |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASE DERIVADA | SUPERCLASE | RELACIÓN | MÉTODOS |
| Liga | Club | Asociación | Todos |
| Campeonato | Club | Asociación | Todos |
| RayoOptico | Club | Asociación | Todos |

|  |
| --- |
| ***3.*** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLASE DERIVADA | SUPERCLASE | RELACIÓN | MÉTODOS | |
| Personal | Club | Agregacion | Ninguno | |
| ***4.*** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| OBJETOS | SUPERCLASE | RELACIÓN | MÉTODOS |
| Jugador | Club | Herencia | Todos |
| DirectorTécnico | Club | Herencia | Todos |
| Presidente | Club | Herencia | Todos |