

Diego Oscar Romero DNI:31.295.941

Se desea desarrollar un sistema que permita registrar la cantidad de Goles que han convertido Jugadores de Futbol en los distintos equipos de futbol que han Jugado, con el fin de poder analizar la cantidad de Goles que han convertido o realizado a lo largo de su trayectoria con información adicional del equipo de Futbol. Para esto, se deberá permitir registrar distintos Jugadores y posteriormente su respectiva Cantidad de Goles informando su Equipo. Es decir, un Jugador registrará muchos Goles en distintos equipos de futbol en toda su trayectoria.

Las entidades relevantes candidatas a tener en cuenta son:

- Jugador (id (int), Nombre (string))
- GolesPorJugadorEquipo (id int, CantidadGoles int, Equipo (string), IdJugador int).

- 1) Crear las tablas y el modelo conceptual correcto para la relación uno a muchos anterior. (Un Jugador de Futbol ha realizado Uno o Muchos goles en equipos de Futbol y un registro en la tabla GolesPorJugadorEquipo pertenece a un único Jugador) OJO-Ayuda: **NO Olvidar FK sobre la tabla GolesPorJugadorEquipo.**
- 2) La aplicación por defecto debe iniciar con una vista de Inicio de Bienvenida: “Bienvenido” que muestre un mensaje, “Bienvenido/a a la evaluación de aprendizaje Nro 2”. Debajo de este mensaje se debe mostrar dos links de acción.

- Administrar de Jugadores de Futbol. [- Administrar de Jugadores de Futbol.](#)

- Administrar Goles

[- Administrar Goles](#)

Considerar el desarrollo de un Layout principal para todo el sitio, que tenga un footer con los datos del alumno (Nombre y Apellido – DNI) y un link que permita regresar al inicio: “Bienvenido”. Todas las vistas deberán utilizar ese Layout.

[Ir a Administrar de Jugadores de Futbol](#) [Administrar Jugadores de Futbol \(JugadoresController\):](#)

3) [Administrar Jugadores de Futbol \(JugadoresController\):](#)

- a. Cuando el usuario haga click en “[Ir a Administrar de Jugadores de Futbol](#)” se deberá mostrar una vista con la lista de Jugadores existentes:

Crear Nuevo Jugador

“ListaJugadores”. Esta vista deberá tener un link o botón: “[Crear Nuevo Jugador](#)”.

Crear Nuevo Jugador

- b. Una vez que se elija “[Crear Nuevo Jugador](#)” se debe redirigir a una funcionalidad página o vista “AltaJugador” que permita registrar nuevos Jugadores. El nombre del Jugador no debe tener más de 80 caracteres y debe ser de ingreso obligatorio. Una vez registrado el nuevo Jugador, se deberá redireccionar a “ListaJugadores”.

Administrar Goles (GolesController):

4) [Administrar Goles \(GolesController\):](#)

- a. Cuando el usuario haga click [Administrar Goles](#) se deberá mostrar una vista con la lista de Goles existentes : “ListaGolesJugador”.

Información a Mostrar en la Vista:

Equipo | Cantidad Goles | Jugador (Nombre)

Esta vista deberá tener un link o botón: “[Crear Nuevo Registro de Goles](#)”.

- b. Una vez que se elija “[Crear Nuevo Registro de Goles](#)” se debe redirigir a una funcionalidad página o vista “AltaGolesJugador” que permita: Registrar la Cantidad de Goles de un Jugador en un Equipo informativo:

Jugador (Combo),
Equipo (Text),
Cantidad Goles (Text) en ese equipo informado.

Se debe validar que todos los campos sean de ingreso obligatorio. Cantidad de Goles debe ser mayor a 1 y menor a 1000. Luego de persistir el registro de Goles, se deberá redirigir a “**ListaGolesJugador**”.

5) En el registro anterior se debe dar tratamiento o control a la situación de Registro duplicado para un mismo Jugador y Equipo. Es decir, en el caso que ya exista registro de la cantidad de Goles de un Jugador para un equipo en particular, se deberá actualizar el registro existente por la nueva cantidad de Goles para ese mismo equipo y Jugador.

6) (Deseable) Generar un servicio Web Api (Controller Web Api: “CantidadTotalGolesEquipoApiController”), que tenga un EndPoint tipo “Get” que calcule la cantidad total de Goles para un equipo recibido por parámetro, y devuelva un string con la cantidad calculada.
/CantidadTotalGolesEquipoApi/barcelona

NOTAS:

- Usar ASP.NET MVC. Es obligatorio el uso de Entity Framework (DataBaseFirst).
- Considerar usar validaciones DataAnnotations.
- Se valorará buenas prácticas de programación: (Prolijidad en Código, Controllers “Flacos”, separación en capas lógicas o físicas, etc) (Servicios-DAL), - No Usar Scaffolding. Usar LINQ To Entities.

COMPLETADO