



Proyecto Fin de Grado

Desarrollo de Aplicaciones Multiplataforma Dual

IES Santiago Hernández

Jorge Pascual Cases

02/09/2019

ÍNDICE

1.	Análisis previo	1
1.1.	Descripción del producto	1
1.2.	Descripción de la empresa	1
1.3.	Justificación del proyecto	1
1.4.	Planificación del proyecto	3
2.	Modelado: Análisis y diseño	5
2.1.	Diagrama de Casos de Uso	5
2.2.	Diagrama de Entidad Relación y Relacional	6
2.3.	Diagrama de Clases	17
2.4.	Diseño de interfaces	18
2.5.	Implementación y pruebas	22
3.	Bibliografía	29

1. Análisis previo

1.1. Descripción del producto

FedesApp es un nuevo producto multiplataforma que nace con la finalidad de facilitar tareas cotidianas en federaciones deportivas. Está planteada para ser utilizada en sus smartphones por los árbitros y en ordenador por los administradores de la federación.

La Federación Aragonesa de Baloncesto, en concreto, demanda una aplicación con las siguientes funciones:

- Designar los partidos de la semana
 - o La aplicación deberá tener en cuenta la disponibilidad del árbitro
 - o En el futuro se puede implementar una funcionalidad que controle que cuando se pongan partidos seguidos en sitios diferentes, el árbitro tenga tiempo de llegar al pabellón
- Obtener la disponibilidad semanal de los árbitros
 - o Desde el móvil los árbitros podrán enviar las franjas del fin de semana o entre semana que tienen libres
- Recibir los resultados de los partidos
 - o Para partidos que no utilicen acta digital, los árbitros seguirán siendo los encargados de enviar los resultados de los partidos y para ello podrán usar un servicio de la aplicación.

El programa guardará los partidos realizados en base de datos y a partir de ellos generará recibos de las nóminas mensualmente, los cuales podrán comprobar los árbitros desde su móvil.

1.2. Descripción de la empresa

Este proyecto va a ser llevado a cabo por un programador freelance, que trabaja en un espacio de coworking. Está empezando a dedicarse a la programación de aplicaciones multiplataforma y quiere introducirse en un mercado bastante competitivo y cambiante, lo que supone un verdadero reto.

1.3. Justificación del proyecto

Ya existen algunas posibilidades en el mercado a la hora de gestionar una federación, entre ellas:

1. Federatio: <https://dclinformatica.com/>



Ofrece una solución completa que incluye servicio de página web, árbitros e incluso acceso a seleccionadores.

2. GesDeportiva: <https://www.gesdeportiva.es/>



Esta aplicación ofrece un servicio más parecido a lo que demanda la Federación Aragonesa de Baloncesto, ya que además GesDeportiva está centrada principalmente en este deporte.

Sin embargo, FedesApp se ajusta mejor a las necesidades específicas del cliente porque está diseñada a medida según las demandas y las peticiones de éste y no va a requerir que contrate un módulo que no va a usar.

En cuanto a las plataformas escogidas para su desarrollo, se han escogido para adaptar la experiencia a cada tipo de usuario. Los administradores van a llevar a cabo tareas más complejas y que requieren más tiempo, por lo que hacerlas desde un móvil podía ser un inconveniente. Por el contrario, para los árbitros es más útil poder utilizar las funcionalidades de la aplicación desde cualquier parte (ej: Para mandar el resultado de un partido nada más terminarlo) por lo que la necesitan en su smartphone.

Para gestionar el desarrollo multiplataforma de forma eficiente se usará la tecnología Xamarin, que con el mismo código realiza la portabilidad de forma nativa a teléfonos móviles Android e iOS (aunque estos últimos no tienen soporte todavía).

La meta que se persigue con este proyecto es mejorar las condiciones de trabajo de los integrantes de una federación deportiva, para que ésta pueda funcionar de forma más eficiente y cómoda repercutiendo en la calidad de los servicios que ofrecen a los deportistas.

1.4. Planificación del proyecto

Este proyecto se va a dividir fundamentalmente en tres fases:

- Diseño y documentación (28 días):

Esta primera fase es esencial para el éxito del proyecto. En ella se definen los requisitos y se planea cómo va a ir el desarrollo. Se preparan bocetos de las interfaces y se realizan diagramas de casos de uso y de entidad relación entre otros.

- Preparación para el desarrollo (7 días):

Se comienza esta fase creando el servidor de la base de datos (algo imprescindible para continuar con el desarrollo) y la base de datos, con todas sus tablas e insertando algún dato. Después se empieza a escribir código XAML únicamente para las pantallas.

- Desarrollo / Programación (35 – 40 días):

En esta última fase es en la que se escribe la mayor parte del código del proyecto y donde se realizan las pruebas de que todo funciona correctamente.

A cargo del proyecto va a estar un único programador. Debido a esto, la metodología para el desarrollo va a ser una metodología en cascada. Las metodologías ágiles son atractivas pero se les saca más partido cuando el equipo de desarrollo está compuesto por más de una persona. Sin embargo, para

organizarse las ideas y las tareas pendientes, el programador utiliza un tablero “Kanban” con la aplicación Trello:

Tablero Trello:

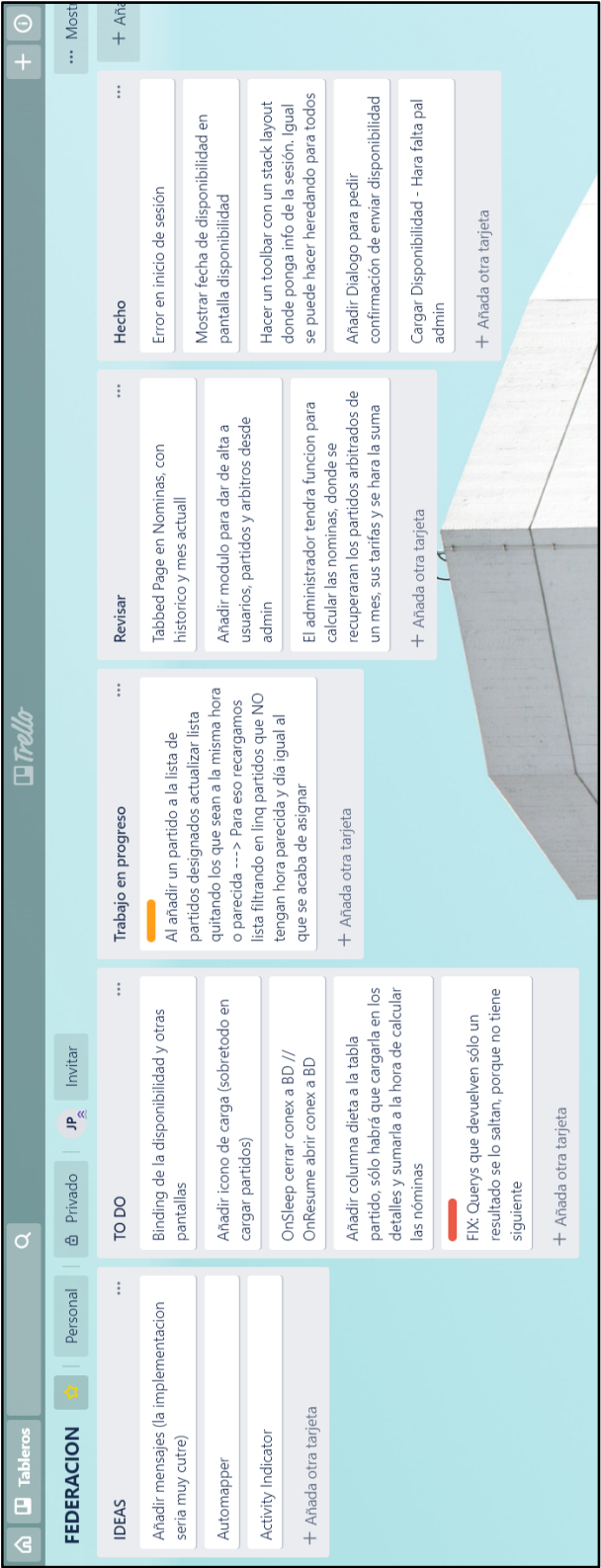
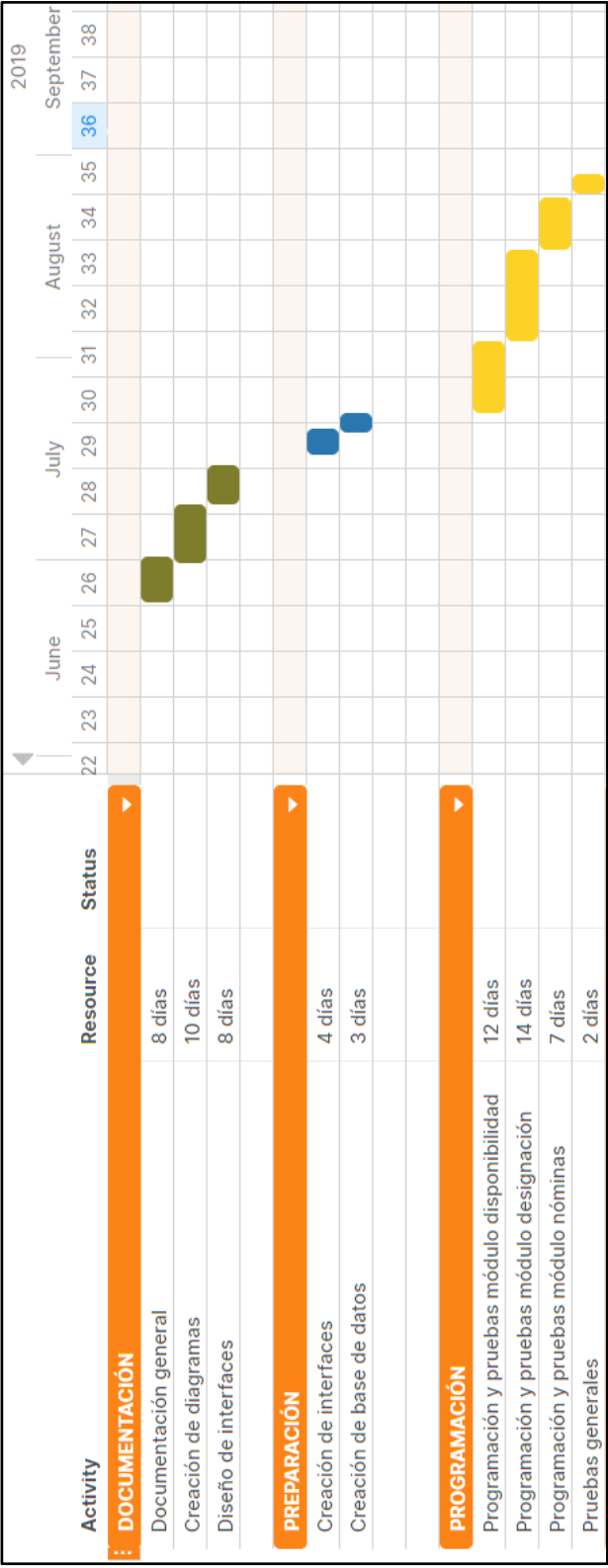
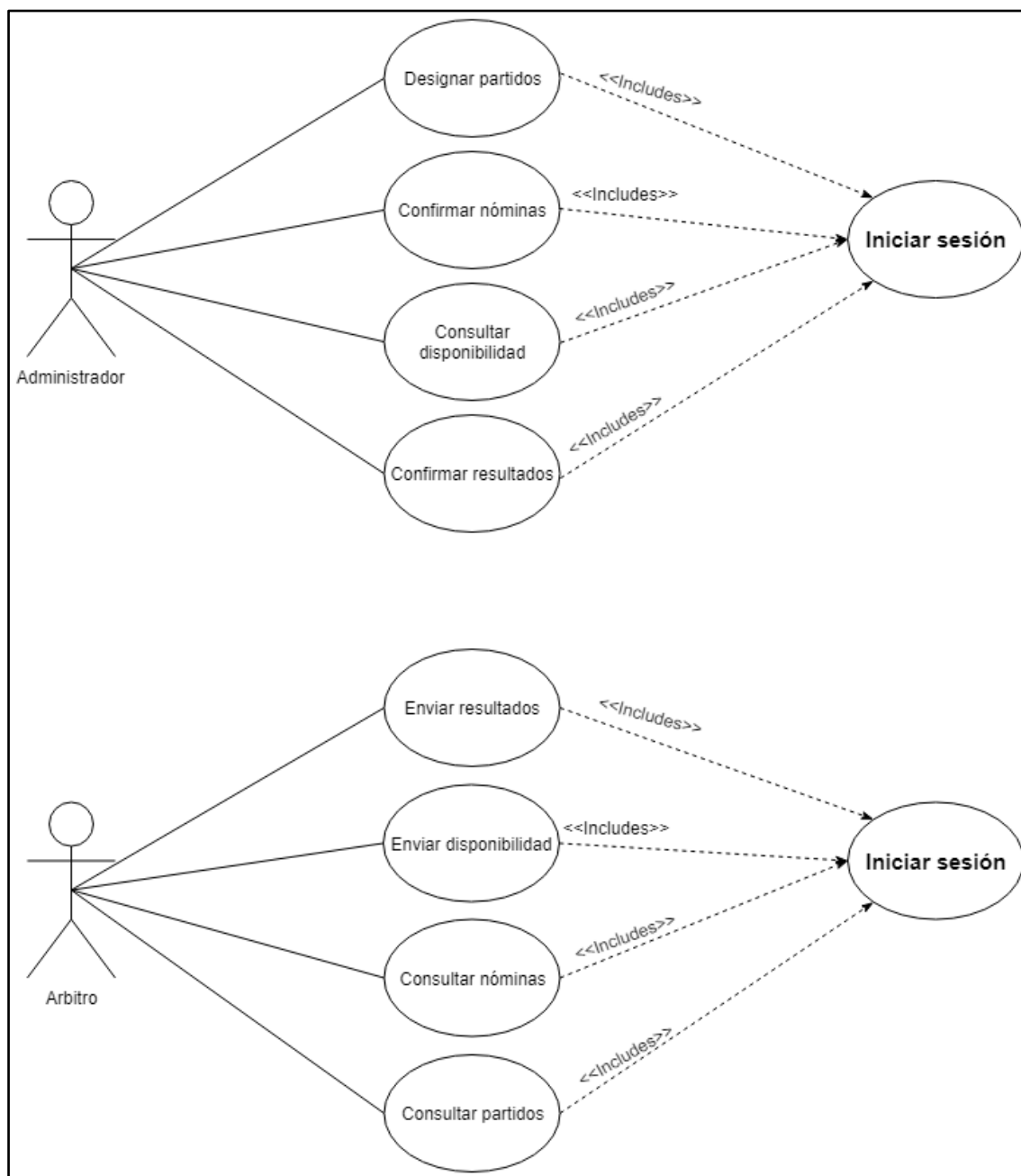


Diagrama de Gantt:

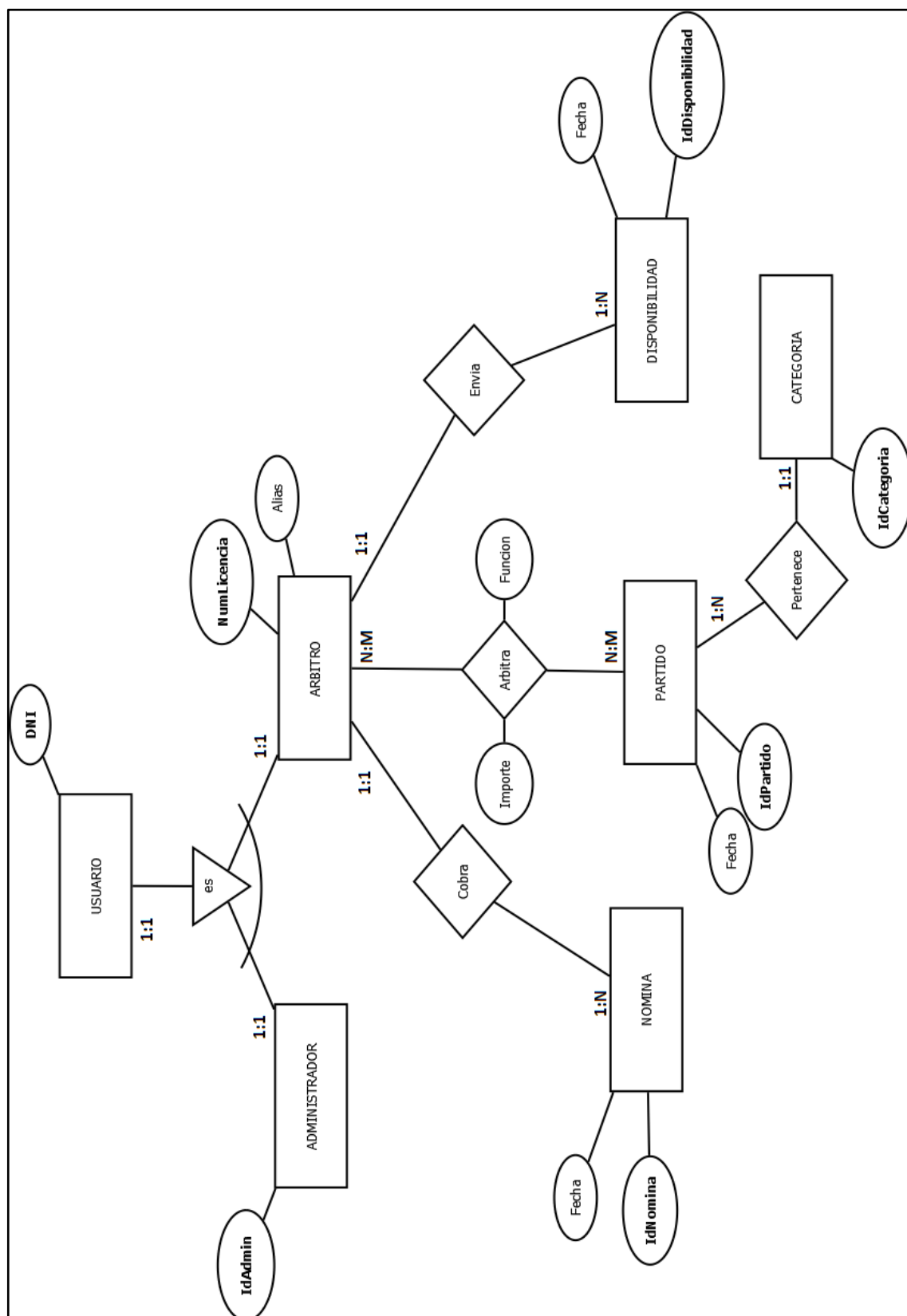


2. Modelado: Análisis y diseño

2.1. Diagrama de Casos de Uso



2.2. Diagrama de Entidad Relación y Relacional

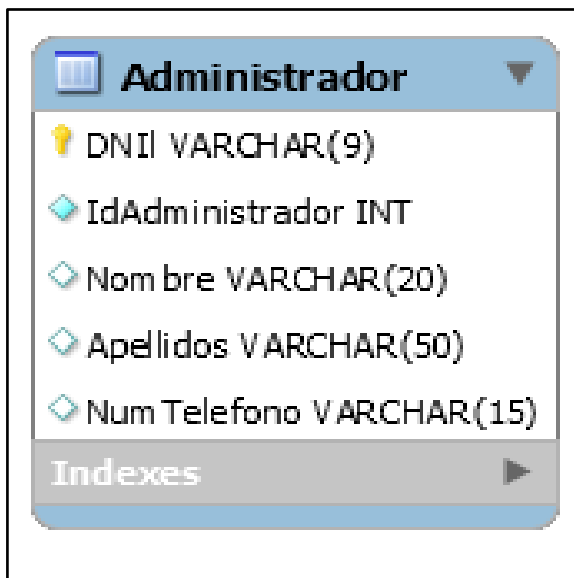


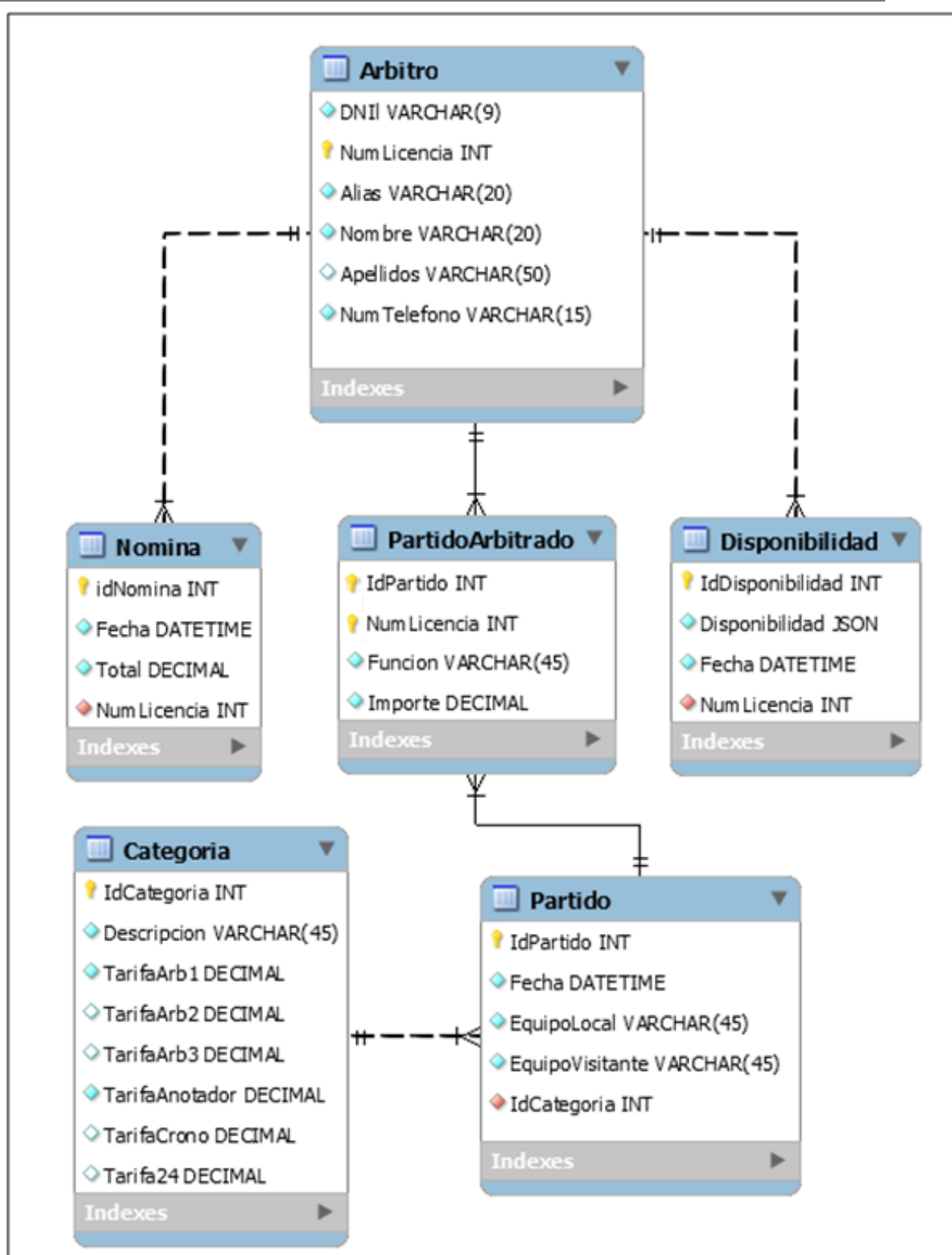
La tabla **disponibilidad** posee un campo XML / JSON (*no relacional*) que contiene en detalle las franjas horarias que tiene libres el árbitro ese fin de semana. El documento tiene la siguiente estructura:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<disponibilidad fecha="YYYY/MM/DD">
  <viernesTarde>NO</viernesTarde>
  <sabadoMañana>SI</sabadoMañana>
  <sabadoTarde>NO</sabadoTarde>
  <domingoMañana>SI</domingoMañana>
  <domingoTarde>SI</domingoTarde>
  <comentarios></comentarios>
</disponibilidad>
```

En el campo comentarios puede poner el árbitro alguna nota u observación para que interprete el administrador.

- Modelo relacional:





Scripts creación BBDD:

```
-----  
-----CREATE TABLES-----  
-----  
-- Table Administrador  
-----  
CREATE TABLE Administrador (  
  IdAdministrador INT NOT NULL UNIQUE IDENTITY(1, 1),  
  DNI VARCHAR(9) NOT NULL,  
  Nombre VARCHAR(20) NULL,  
  Apellidos VARCHAR(50) NULL,  
  NumTelefono VARCHAR(15) NULL,  
  Alias VARCHAR(20) NOT NULL UNIQUE,  
  PRIMARY KEY (IdAdministrador)  
)  
-----  
-- Table Arbitro  
-----  
CREATE TABLE Arbitro (  
  IdArbitro INT NOT NULL IDENTITY(1, 1),  
  Alias VARCHAR(20) NOT NULL UNIQUE,  
  DNI VARCHAR(9) NOT NULL,  
  Nombre VARCHAR(20) NOT NULL,  
  Apellidos VARCHAR(50) NOT NULL,  
  NumTelefono VARCHAR(15) NOT NULL,  
  PRIMARY KEY (IdArbitro)  
)  
-----  
-- Table Funcion  
-----  
CREATE TABLE Funcion(  
  IdFuncion INT NOT NULL IDENTITY(1, 1),  
  Descripcion VARCHAR(50) NOT NULL  
  PRIMARY KEY (IdFuncion)  
)  
-----  
-- Table Nomina  
-----  
CREATE TABLE Nomina (  
  IdNomina INT NOT NULL IDENTITY(1, 1),  
  Fecha DATETIME NOT NULL,  
  Total DECIMAL(5,2) NOT NULL,  
  IdArbitro INT NOT NULL,  
  PRIMARY KEY (IdNomina),  
  CONSTRAINT fk_Nomina_Arbitro  
  FOREIGN KEY (IdArbitro)  
  REFERENCES Arbitro (IdArbitro)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION  
)  
-----  
-- Table Categoria  
-----  
CREATE TABLE Categoria (  
  IdCategoria INT NOT NULL IDENTITY(1, 1),  
  Descripcion VARCHAR(45) NOT NULL,  
  Abreviatura VARCHAR(40)  
  PRIMARY KEY (IdCategoria)  
)  
-----  
-- Table Partido  
-----  
CREATE TABLE Partido (  
  IdPartido INT NOT NULL IDENTITY(1, 1),
```

```
Fecha DATETIME NOT NULL,  
EquipoLocal VARCHAR(45) NOT NULL,  
EquipoVisitante VARCHAR(45) NOT NULL,  
IdCategoria INT NOT NULL,  
Ubicacion VARCHAR(200) NOT NULL,  
Observaciones VARCHAR(200),  
Resultado VARCHAR(20),  
PRIMARY KEY (IdPartido),  
CONSTRAINT fk_Partido_Categoria  
FOREIGN KEY (IdCategoria)  
REFERENCES Categoria (IdCategoria)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
)
```

```
CREATE UNIQUE NONCLUSTERED INDEX IX_Partido_Fecha_Ubicacion  
ON Partido (Fecha, Ubicacion)
```

```
CREATE UNIQUE NONCLUSTERED INDEX IX_Partido_Equipos_Fecha_Categoria  
ON Partido (Fecha, EquipoLocal, EquipoVisitante, IdCategoria)
```

```
-----  
-- Table PartidoArbitrado  
-----
```

```
CREATE TABLE PartidoArbitrado (  
IdPartidoArbitrado INT NOT NULL IDENTITY(1, 1),  
IdPartido INT NOT NULL,  
IdArbitro INT NOT NULL,  
IdFuncion INT NOT NULL,  
Importe DECIMAL(5,2),  
PRIMARY KEY (IdPartidoArbitrado),  
CONSTRAINT fk_PartidoArbitrado_Partido  
FOREIGN KEY (IdPartido)  
REFERENCES Partido (IdPartido)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT fk_PartidoArbitrado_Arbitro  
FOREIGN KEY (IdArbitro)  
REFERENCES Arbitro (IdArbitro),  
CONSTRAINT fk_PartidoArbitrado_Funcion  
FOREIGN KEY (IdFuncion)  
REFERENCES Funcion (IdFuncion)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
)
```

```
CREATE UNIQUE NONCLUSTERED INDEX IX_PartidoArbitrado  
ON PartidoArbitrado (IdPartido, IdArbitro)
```

```
CREATE UNIQUE NONCLUSTERED INDEX IX_PartidoArbitrado_Funcion  
ON PartidoArbitrado (IdPartido, IdFuncion)
```

```
-----  
-- Table Disponibilidad  
-----
```

```
CREATE TABLE Disponibilidad (  
IdDisponibilidad INT NOT NULL IDENTITY(1, 1),  
Disponibilidad XML NOT NULL,  
Fecha DATETIME NOT NULL,  
IdArbitro INT NOT NULL,  
PRIMARY KEY (IdDisponibilidad),  
CONSTRAINT fk_Disponibilidad_Arbitro  
FOREIGN KEY (IdArbitro)  
REFERENCES Arbitro (IdArbitro)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
)
```

```
-----  
-- Table Usuario  
-----
```

```

CREATE TABLE Usuario (
    IdUsuario INT NOT NULL IDENTITY(1, 1),
    IdArbitro INT,
    IdAdministrador INT,
    Usuario VARCHAR(9) NOT NULL UNIQUE,
    Password VARCHAR(20) NOT NULL,
    IsAdmin bit
    PRIMARY KEY (IdUsuario),
    CONSTRAINT FK_Usuario_Arbitro
    FOREIGN KEY (IdArbitro)
    REFERENCES Arbitro(IdArbitro),
    CONSTRAINT FK_Usuario_Administrador
    FOREIGN KEY (IdAdministrador)
    REFERENCES Administrador(IdAdministrador)
)

-----
-- Table CategoriaLinea
-----

CREATE TABLE CategoriaLinea (
    IdCategoriaLinea INT NOT NULL IDENTITY(1, 1),
    IdCategoria INT NOT NULL,
    IdFuncion INT NOT NULL,
    Importe DECIMAL(5,2)
    PRIMARY KEY (IdCategoriaLinea),
    CONSTRAINT FK_CategoriaLinea_Categoria
    FOREIGN KEY (IdCategoria)
    REFERENCES Categoria(IdCategoria),
    CONSTRAINT FK_CategoriaLinea_Funcion
    FOREIGN KEY (IdFuncion)
    REFERENCES Funcion(IdFuncion)
)

CREATE UNIQUE NONCLUSTERED INDEX IX_CategoriaLinea
ON CategoriaLinea (IdCategoria, IdFuncion)

-----
-- Table Informe
-----

CREATE Table Informe (
    IdInforme INT NOT NULL IDENTITY(1, 1),
    IdArbitro INT NOT NULL,
    FechaInforme DATETIME NOT NULL,
    TextoInforme VARCHAR(MAX),
    IsFavorable BIT
    PRIMARY KEY (IdInforme),
    CONSTRAINT FK_Informe_Arbitro
    FOREIGN KEY (IdArbitro)
    REFERENCES Arbitro(IdArbitro)
)

-----
-----INSERTS-----
-----
insert into administrador values ('11223344a', 'luis', 'admin', '976976976', 'LUIS')
insert into arbitro values ('jorge', '12345678a', 'jorge', 'pascual cases', '123456789')
insert into arbitro values ('pascual', '12345678b', 'pepe', 'pascual', '111222333')
insert into arbitro values ('perea', '12345678c', 'fran', 'perea', '666777888')
insert into arbitro values ('blasco', '12345678c', 'nuria', 'blasco', '686868686')
insert into arbitro values ('sanchez', '12345678e', 'luis', 'sanchez', '655555556')
insert into arbitro values ('mar', '12345678e', 'mar', 'rojo', '612121212')
insert into usuario values (1, null, 'jorge', 'pass', 0)
insert into usuario values (2, null, 'pepe', '123', 0)
insert into usuario values (3, null, 'perea', '1', 0)
insert into usuario values (4, null, 'blasco', '1', 0)
insert into usuario values (5, null, 'sanchez', '1', 0)
insert into usuario values (null, 1, 'admin', 'admin', 1)
insert into categoria values ('social oro', 'SOC. ORO')
insert into categoria values ('social plata', 'SOC. PLATA')
insert into categoria values ('social bronce', 'SOC. BRONCE')

```

```

insert into partido values ('20190818 12:00:00', 'doctor azua', 'boscas', 1, 'pabellón cesaraugusto. calle asín y palacios, 5', null, null)
insert into partido values ('20190818 09:15:00', 'helios', 'cbz', 1, 'pabellón helios. avda. ranillas, 2', 'hora oficial de comienzo del partido: 09:25', null)
insert into partido values ('20190821 18:00:00', 'solteros', 'casados', 3, 'pabellón dominicos, pza. san francisco', null, null)
select * from Partido
INSERT INTO PartidoArbitrado values(1, 1, 1, null)
INSERT INTO PartidoArbitrado values(1, 2, 2, null)
INSERT INTO PartidoArbitrado values(1, 3, 3, null)
INSERT INTO PartidoArbitrado values(1, 4, 4, null)
INSERT INTO PartidoArbitrado values(1, 5, 5, null)
INSERT INTO PartidoArbitrado values(1, 6, 6, null)
INSERT INTO PartidoArbitrado values(3, 2, 1, null)
INSERT INTO PartidoArbitrado values(3, 1, 4, null)
INSERT INTO Funcion VALUES ('Arbitro Principal')
INSERT INTO Funcion VALUES ('Arbitro Auxiliar')
INSERT INTO Funcion VALUES ('Tercer Arbitro')
INSERT INTO Funcion VALUES ('Anotador')
INSERT INTO Funcion VALUES ('Cronometrador')
INSERT INTO Funcion VALUES ('Operador 24')
INSERT INTO CategoriaLinea VALUES (1, 1, 35)
INSERT INTO CategoriaLinea VALUES (1, 2, 27.50)
INSERT INTO CategoriaLinea VALUES (1, 3, NULL)
INSERT INTO CategoriaLinea VALUES (1, 4, 14.50)
INSERT INTO CategoriaLinea VALUES (1, 5, 17.00)
INSERT INTO CategoriaLinea VALUES (1, 6, 17.00)
INSERT INTO CategoriaLinea VALUES (2, 1, 32)
INSERT INTO CategoriaLinea VALUES (2, 2, 25.00)
INSERT INTO CategoriaLinea VALUES (2, 3, NULL)
INSERT INTO CategoriaLinea VALUES (2, 4, 13.50)
INSERT INTO CategoriaLinea VALUES (2, 5, 15.00)
INSERT INTO CategoriaLinea VALUES (2, 6, 15.00)
INSERT INTO CategoriaLinea VALUES (3, 1, 30.00)
INSERT INTO CategoriaLinea VALUES (3, 2, NULL)
INSERT INTO CategoriaLinea VALUES (3, 3, NULL)
INSERT INTO CategoriaLinea VALUES (3, 4, 15.00)
INSERT INTO CategoriaLinea VALUES (3, 5, NULL)
INSERT INTO CategoriaLinea VALUES (3, 6, NULL)
INSERT INTO Nomina VALUES('20190705', 123.50, 1)
INSERT INTO Nomina VALUES('20180705', 123.50, 1)
insert into Nomina values ('20190514', 140, 1)
INSERT INTO Informe VALUES(1, '20190816', 'Evoluciona favorablemente.
Debe pulir algunos aspectos todavia')
-----
--STORED PROCEDURES--
-----
--=====
-- Author: Jorge Pascual
-- Create Date: 14/08/2019
-- Description: Obtiene todos los arbitros que participan en un partido
--=====
CREATE PROCEDURE [dbo].[GetArbitrosPartido]
(
    @IdPartido int
)
AS
BEGIN

    SELECT Arbitro.IdArbitro, Alias, Nombre, Apellidos, NumTelefono, Descripcion FROM Arbitro
    JOIN PartidoArbitrado ON PartidoArbitrado.IdArbitro = Arbitro.IdArbitro
    JOIN Funcion ON Funcion.IdFuncion = PartidoArbitrado.IdFuncion
    where IdPartido = @IdPartido

    order by Funcion.IdFuncion

END

--=====
-- Author: Jorge Pascual
-- Create Date: 13/08/2019
-- Description: Obtiene toda la informacion necesaria para mostrar el partido
--=====
CREATE PROCEDURE [dbo].[GetDatosPartido]

```

```

(
    @IdArbitro int,
    @FechaDesde Date,
    @FechaHasta Date
)
AS
BEGIN

    SELECT Partido.IdPartido, Fecha, EquipoLocal, EquipoVisitante, Partido.IdCategoria, Categoria.Descripcion,
    ISNULL(Abreviatura, '') AS Abreviatura,
    Alias, ISNULL(NumTelefono, '') AS NumTelefono, ISNULL(Resultado, '') as Resultado
    FROM Partido JOIN Categoria ON Categoria.IdCategoria = Partido.IdCategoria
    JOIN PartidoArbitrado ON PartidoArbitrado.IdPartido = Partido.IdPartido
    JOIN Arbitro ON Arbitro.IdArbitro = PartidoArbitrado.IdArbitro
    JOIN Funcion ON Funcion.IdFuncion = PartidoArbitrado.IdFuncion

    WHERE Arbitro.IdArbitro = @IdArbitro
    AND cast(Fecha as date) >= ISNULL(@FechaDesde, '21000101') and cast(Fecha as Date) <= ISNULL(@FechaHasta,
    '20000101')

    ORDER BY Fecha

END

-- =====
-- Author: Jorge Pascual
-- Create Date: 18/08/2019
-- Description: Obtiene datos de árbitros + el numero de partidos que ya tienen asignados ese finde
-- =====
CREATE PROCEDURE [dbo].[GetNumPartidosArbitro]
(
    @FechaDesde DATE,
    @FechaHasta DATE
)
AS
BEGIN
    WITH NumPartidos AS(
        SELECT IdArbitro, Count(IdPartidoArbitrado) as NumPartidosDesignados
        FROM PartidoArbitrado
        WHERE IdPartido in
        (
            SELECT IdPartido
            FROM Partido
            WHERE CAST(Fecha AS date) >= @FechaDesde
            AND CAST(Fecha AS DATE) <= @FechaHasta
        )
        GROUP BY IdArbitro
    ),
    DisponibilidadSemana AS (
        SELECT IdDisponibilidad, Fecha, Disponibilidad, IdArbitro
        FROM Disponibilidad D
        WHERE cast(D.Fecha as date) >= @FechaDesde and cast(D.Fecha as date) < @FechaHasta
    )
    select Arbitro.IdArbitro, Alias, ISNULL(Nombre, '') AS Nombre, ISNULL(Apellidos, ''), ISNULL(NumPartidosDesignados, 0)
    FROM Arbitro
    LEFT JOIN NumPartidos ON NumPartidos.IdArbitro = Arbitro.IdArbitro
    LEFT JOIN DisponibilidadSemana D ON D.IdArbitro = Arbitro.IdArbitro
    WHERE
    (D.Disponibilidad.value('(DisponibilidadXMLDTO/Viernes)[1]', 'bit') = 1
    OR D.Disponibilidad.value('(DisponibilidadXMLDTO/Sabado1)[1]', 'bit') = 1
    OR D.Disponibilidad.value('(DisponibilidadXMLDTO/Sabado2)[1]', 'bit') = 1
    OR D.Disponibilidad.value('(DisponibilidadXMLDTO/Domingo1)[1]', 'bit') = 1
    OR D.Disponibilidad.value('(DisponibilidadXMLDTO/Domingo2)[1]', 'bit') = 1)
    OR D.Disponibilidad IS NULL

END

-- =====
-- Author: Jorge
-- Create Date: 17/08/2019
-- Description: Obtiene los partidos por asignar que cumplen la disponibilidad de un árbitro

```

```

--=====
CREATE PROCEDURE [dbo].[GetPartidosDisponibles]
(
    @IdArbitro int,
        @FechaDesde datetime,
        @FechaHasta datetime
)
AS
BEGIN

    CREATE TABLE #PartidosDisponibles
    (
        IdPartido INT NOT NULL,
        Fecha DATETIME NOT NULL,
        EquipoLocal VARCHAR(45) NOT NULL,
        EquipoVisitante VARCHAR(45) NOT NULL,
        IdCategoria INT NOT NULL,
        Ubicacion VARCHAR(200) NOT NULL,
        Observaciones VARCHAR(200),
        --ComentariosDisponibilidad VARCHAR(MAX)
    )

    --Obtengo la fecha del sábado y del domingo porque harán falta después
    DECLARE @FechaViernes DATE
    DECLARE @FechaSabado DATE
    DECLARE @FechaDomingo DATE

    --Suponemos que el FechaHasta vendrá bien desde código, SIEMPRE tendrá que ser el lunes de la semana siguiente.
    --Es decir, los dos días anteriores son los días del fin de semana que nos interesan
    SET @FechaViernes = DATEADD(DAY, -3, @FechaHasta)
    SET @FechaSabado = DATEADD(DAY, -2, @FechaHasta)
    SET @FechaDomingo = DATEADD(DAY, -1, @FechaHasta)

    --Separamos en variables la disponibilidad del XML para hacer más cómodamente las comparaciones
    DECLARE @DisponibilidadXML XML
    DECLARE @Viernes BIT
    DECLARE @Sabado1 BIT
    DECLARE @Sabado2 BIT
    DECLARE @Domingo1 BIT
    DECLARE @Domingo2 BIT
    --DECLARE @Comentarios nvarchar(max)

    SET @DisponibilidadXML = (SELECT Disponibilidad FROM Disponibilidad where IdArbitro = @IdArbitro AND Fecha >
    @FechaDesde AND Fecha < @FechaHasta)

    SET @Viernes = @DisponibilidadXML.value('(DisponibilidadXMLDTO/Viernes)[1]', 'bit')
    SET @Sabado1 = @DisponibilidadXML.value('(DisponibilidadXMLDTO/Sabado1)[1]', 'bit')
    SET @Sabado2 = @DisponibilidadXML.value('(DisponibilidadXMLDTO/Sabado2)[1]', 'bit')
    SET @Domingo1 = @DisponibilidadXML.value('(DisponibilidadXMLDTO/Domingo1)[1]', 'bit')
    SET @Domingo2 = @DisponibilidadXML.value('(DisponibilidadXMLDTO/Domingo2)[1]', 'bit')
    --SET @Comentarios = @DisponibilidadXML.value('(DisponibilidadXMLDTO/Comentarios)[1]', 'nvarchar(max)')

    --Insertamos en la tabla temporal los partidos que cumplan la disponibilidad y no estén designados ya a este árbitro
    --Si el arbitro no ha enviado disponibilidad se sobreentiende que puede hacer partidos en todas las franjas
    IF @DisponibilidadXML IS NULL
    BEGIN
        INSERT INTO #PartidosDisponibles
        SELECT IdPartido, Fecha, EquipoLocal, EquipoVisitante, IdCategoria, Ubicacion, Observaciones--,
        @Comentarios
        FROM Partido
        where cast(Fecha as date) between @FechaViernes AND @FechaDomingo
        and IdPartido not in (SELECT IdPartido FROM PartidoArbitrado WHERE IdArbitro = @IdArbitro)
    END
    ELSE
    BEGIN
        IF @Viernes = 1

            INSERT INTO #PartidosDisponibles
            SELECT IdPartido, Fecha, EquipoLocal, EquipoVisitante, IdCategoria, Ubicacion, Observaciones--,
            @Comentarios
    
```



```

FROM Partido
where cast(Fecha as date) = @FechaViernes
and IdPartido not in (SELECT IdPartido FROM PartidoArbitrado WHERE IdArbitro = @IdArbitro)

IF @Sabado1 = 1

INSERT INTO #PartidosDisponibles
SELECT IdPartido, Fecha, EquipoLocal, EquipoVisitante, IdCategoria, Ubicacion, Observaciones--,
@Comentarios

FROM Partido
where cast(Fecha as date) = @FechaSabado
and (DATEPART(HOUR, Fecha) >= 9 and DATEPART(HOUR, Fecha) <= 14)
and IdPartido not in (SELECT IdPartido FROM PartidoArbitrado WHERE IdArbitro = @IdArbitro)

IF @Sabado2 = 1

INSERT INTO #PartidosDisponibles

SELECT IdPartido, Fecha, EquipoLocal, EquipoVisitante, IdCategoria, Ubicacion, Observaciones--,
@Comentarios

FROM Partido
where cast(Fecha as date) = @FechaSabado
and (DATEPART(HOUR, Fecha) >= 16 and DATEPART(HOUR, Fecha) <= 21)
and IdPartido not in (SELECT IdPartido FROM PartidoArbitrado WHERE IdArbitro = @IdArbitro)

IF @Domingo1 = 1

INSERT INTO #PartidosDisponibles

SELECT IdPartido, Fecha, EquipoLocal, EquipoVisitante, IdCategoria, Ubicacion, Observaciones--,
@Comentarios

FROM Partido
where cast(Fecha as date) = @FechaDomingo
and (DATEPART(HOUR, Fecha) >= 9 and DATEPART(HOUR, Fecha) <= 14)
and IdPartido not in (SELECT IdPartido FROM PartidoArbitrado WHERE IdArbitro = @IdArbitro)

IF @Domingo2 = 1

INSERT INTO #PartidosDisponibles

SELECT IdPartido, Fecha, EquipoLocal, EquipoVisitante, IdCategoria, Ubicacion, Observaciones--,
@Comentarios

FROM Partido
where cast(Fecha as date) = @FechaDomingo
and (DATEPART(HOUR, Fecha) >= 16 and DATEPART(HOUR, Fecha) <= 21)
and IdPartido not in (SELECT IdPartido FROM PartidoArbitrado WHERE IdArbitro = @IdArbitro)

END;

--Devolvemos los partidos disponibles
SELECT #PartidosDisponibles.IdPartido, #PartidosDisponibles.Fecha, EquipoLocal, EquipoVisitante,
ISNULL(#PartidosDisponibles.IdCategoria, 0) as IdCategoria, --ComentariosDisponibilidad,
Ubicacion, ISNULL(Observaciones, ''), Descripcion, ISNULL(Abreviatura, '')
FROM #PartidosDisponibles
JOIN Categoria ON Categoria.IdCategoria = #PartidosDisponibles.IdCategoria
order by #PartidosDisponibles.Fecha

END

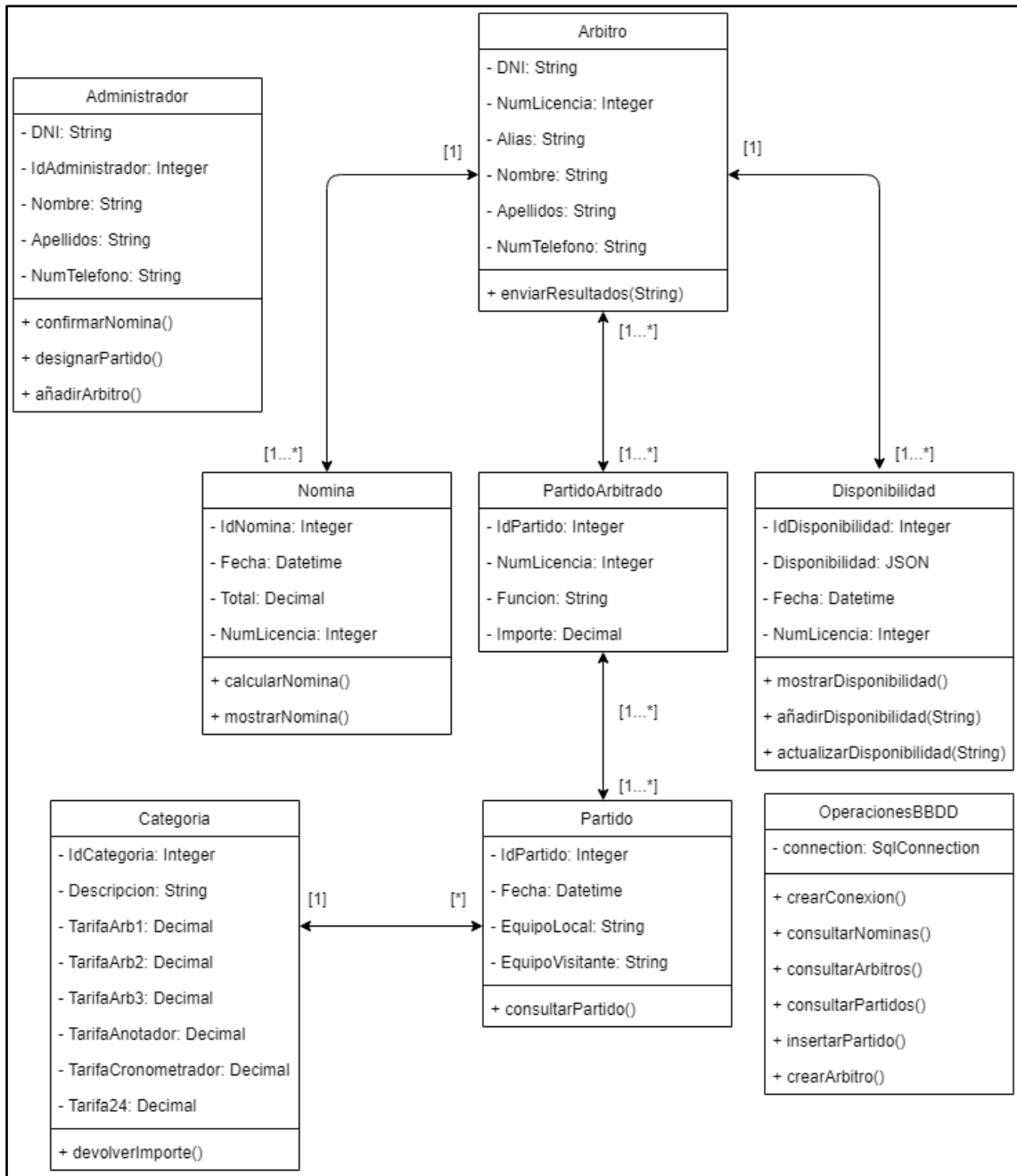
=====
-- Author: Jorge
-- Create Date: 21/08/2019
-- Description: Obtiene todos los partidos disputados una semana (independientemente del arbitro) con su resultado
=====
CREATE PROCEDURE [dbo].[GetResultadosPartidos]
(
    @FechaDesde as DATETIME,
    @FechaHasta as DATETIME
)
AS
BEGIN

```

```
SELECT Partido.IdPartido, Fecha, EquipoLocal, EquipoVisitante, Partido.IdCategoria, Categoria.Descripcion, ISNULL(Abreviatura, "")
AS Abreviatura,
    Ubicacion, ISNULL(Resultado, "") as Resultado
FROM Partido JOIN Categoria ON Categoria.IdCategoria = Partido.IdCategoria
WHERE cast(Fecha as date) >= ISNULL(@FechaDesde, '21000101') and cast(Fecha as Date) <= ISNULL(@FechaHasta,
'20000101')
AND IdPartido IN (SELECT IdPartido FROM PartidoArbitrado)

ORDER BY Fecha
END
```

2.3. Diagrama de Clases



2.4. Diseño de interfaces

Administrador:

Desde el inicio podrá navegar a todas las pantallas que salen en el menú y desde cada una de las pantallas volver al inicio. La única pantalla que tiene otro funcionamiento es la de “Designación de partidos” de la cual pasas a “Designación árbitro” al seleccionar un árbitro del listado.



DESIGNACIÓN ÁRBITRO

ÁRBITRO ALIAS N° NUM

FECHA FECHA

PARTIDOS SIN ASIGNAR

☐ Partido 1

☐ Partido 2

☐ Partido 3

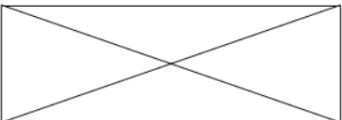
☐ Partido 4

☐ Partido 5

☐ Partido 6

☐ Partido 7

DISPONIBILIDAD



RESULTADOS FECHA

FILTROS:

FILTER

PARTIDO 1	LOCAL 15 - 30 VISITANTE
PARTIDO 2	LOCAL 34 - 42 VISITANTE
PARTIDO 3	LOCAL 68 - 27 VISITANTE

NÓMINAS FECHA		
FILTROS:	Selecciona fecha	FILTER
		BUSCAR
ÁRBITRO 1	Total = 0.00€	↑ ↓
ÁRBITRO 2	Total = 0.00€	
ÁRBITRO 3	Total = 0.00€	

Árbitro:

Al igual que en el administrador, desde el inicio se puede navegar al resto de pantallas y desde éstas volver al inicio.



← Colegiado: Nombre N° 111 Fecha: DD/MM/YY

VIERNES TARDE ☒

SABADO MAÑANA ☐

SABADO TARDE ☐

DOMINGO MAÑANA ☒

DOMINGO TARDE ☒

Introduce información extra aquí

REINICIAR ENVIAR

← Colegiado: Nombre N° 111 Fecha: DD/MM/YY

PERSONA O EQUIPO	Local	Visitante
PERSONA O EQUIPO	Local	Visitante
PERSONA O EQUIPO	Local	Visitante
PERSONA O EQUIPO	Local	Visitante
PERSONA O EQUIPO	Local	Visitante

REINICIAR ENVIAR

Enviar disponibilidad

Enviar resultados

← Colegiado: Nombre N° 111 Fecha: DD/MM/YY

INFORMACIÓN PARTIDO

INFORMACIÓN PARTIDO

INFORMACIÓN PARTIDO

INFORMACIÓN PARTIDO

INFORMACIÓN PARTIDO

← Colegiado: Nombre N° 111 Fecha: DD/MM/YY

INFORMACIÓN NÓMINA

INFORMACIÓN NÓMINA

INFORMACIÓN NÓMINA

INFORMACIÓN NÓMINA

INFORMACIÓN NÓMINA

Consultar Partidos

Consultar Nóminas

2.5. Implementación y pruebas

El primer paso para implementar la aplicación era disponer de un servidor de base de datos para comunicar las acciones que realizaba el administrador con el árbitro y viceversa. Tras investigar las posibles opciones, se escogió utilizar un servidor de Azure, que ofrece un mes de prueba de un servidor de base de datos con recursos suficientes para este proyecto.

Para conectar desde código a la base de datos se recomienda utilizar una API o Web Service como intermediaria entre el código y el servidor de base de datos, para evitar tener la cadena de conexión (con todos los datos como usuario y contraseña) explícitamente en el código, debido a que se podría acceder a ellos si se decompilase la aplicación. Pese a todo, la aplicación no cuenta actualmente con ningún servicio intermediario, aunque se plantea en el futuro implementar uno. Como sustitución de éste método, en nuestro proyecto se utiliza la librería .NET System.Data.SqlClient, que contiene métodos para realizar la conexión a una base de datos de SQL Server con la clase SqlConnection y realizar comandos de inserción, actualización, eliminación y de consultas con la clase SqlCommand.

El desarrollo en Xamarin se compone de dos partes: Las pantallas que se diseñan con XAML y el código que está escrito en C#. En las pantallas es conveniente utilizar absolute layouts y grids al hacer el diseño para que se adapten bien a dispositivos de todos los tamaños.

En cuanto al código se planteó utilizar la arquitectura MVVM (Model – View - ViewModel) pero se acabó desechando la idea por la complejidad que planteaba frente a los escasos beneficios para este proyecto en concreto. Finalmente, el código de las pantallas (eventos de botones, cargar listas etc.) se realiza en la clase asociada al XAML que crea Xamarin y se usa una clase auxiliar para las conexiones con la base de datos.

Hay un grupo amplio de pantallas de la aplicación que tiene un funcionamiento similar. Al iniciarse la pantalla, se realiza una consulta de base de datos para rellenar una lista y que se visualice en pantalla en un ListView. Siguen esta descripción en la aplicación del árbitro: la consulta de partidos, la consulta de nóminas y la consulta de informes. En la aplicación del administrador también funcionan igual la visualización de resultados y la de nóminas.

Las consultas más sencillas se pueden hacer desde código, pero para otros casos (como obtener los partidos de una semana que cumplen la disponibilidad del árbitro) era necesario un procedimiento almacenado creado desde SQL Server, que es el programa que se ha utilizado para gestionar la base de datos.

El otro grupo de pantallas es más diverso. Desde la aplicación del administrador se pueden encontrar tres formularios para crear nuevos árbitros, usuarios y partidos, pero la pantalla principal de la aplicación es la de designar partidos. Ésta es ligeramente más complicada y la comentaremos después. Por otro lado, desde

Android se puede acceder a la pantalla de enviar disponibilidad que también tiene un funcionamiento más especial.

Respecto a las funcionalidades planteadas en los casos de uso:

- Consulta de partidos (Árbitro):

Se muestran los partidos de la semana actual o un mensaje si no hay partidos designados. Otra funcionalidad que se deseaba para el árbitro era la de enviar los resultados de los partidos que había arbitrado, y en desarrollo se decidió implementar esto desde el módulo de consultar partidos, al entrar a cada partido. Sin embargo, el botón de enviar resultado se bloquea siempre que el partido no se haya disputado todavía o que el árbitro que ha iniciado sesión no tenga la función de árbitro principal en ese partido.

Partidos semana 26/08/19

Lista de partidos:

Monzon - Fleta	SOC. PLATA
Arbitro Principal	>
Pabellon Monzon	30/08/19 11:00

Sergio - Jorge	SOC. BRONCE
Arbitro Principal	>
Casa	31/08/19 12:00

Consultar Partidos

Detalles Partido

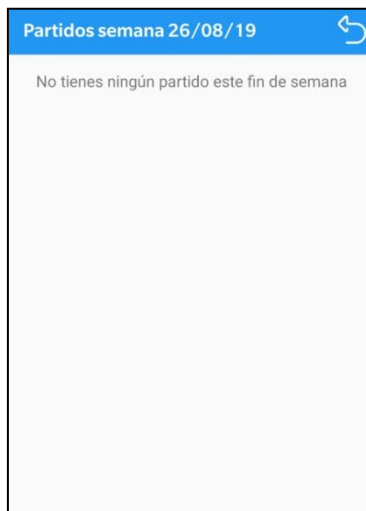
DETALLES PARTIDO

Equipos: Monzon - Fleta
Categoría: Social Plata
Fecha y hora: 30/08/19 11:00
Función: Arbitro Principal
Lugar: Pabellon Monzon
Observaciones:

Árbitros: Jose - Arbitro Principal
Pepe Pascual 111222333
LEON - Cronometrador
Miguel Leon Bello 67776667

15 - 76 **ENVIAR RESULTADO**

Detalles Partidos (Resultado)



[Consultar Partidos sin resultados](#)

- Consulta de nóminas (Árbitro):

Para la visualización de nóminas se diseñó una pantalla con dos pestañas para visualizar el desglose mensual y el desglose anual (histórico) con un simple gesto.

Nóminas mensuales

Nóminas anuales

Nóminas agosto	Nóminas agosto
MENSUAL	HISTORICO
<p>Solteros3 - Casados3 SOC. BRONCE</p> <p>Anotador 15,00€</p> <p>Pabellón Dominicos, Pza. San Francisco 16/08/19 19:00</p> <p>Helios - CBZ SOC. ORO</p> <p>Árbitro Auxiliar 27,50€</p> <p>Pabellón Helios. Avda. Ranillas, 2 18/08/19 09:15</p> <p>Monzon - Fleta SOC. PLATA</p> <p>Árbitro Principal 32,00€</p> <p>TOTAL: 74,50€</p>	<p>Del 1 al 31 de marzo de 2019 30,00€</p> <p>Del 1 al 31 de mayo de 2019 140,00€</p> <p>Del 1 al 31 de diciembre de 2019 78,00€</p> <p>TOTAL: 248,00€</p>
Nóminas agosto	Nóminas agosto
MENSUAL	HISTORICO
<p>No hay nóminas de este mes</p> <p>TOTAL: 0€</p>	<p>No hay nóminas de este año</p> <p>TOTAL: 0€</p>

- Enviar disponibilidad (Árbitro):

Disponibilidad semana 02/09/19	
Disponibilidad:	
Viernes tarde	<input checked="" type="checkbox"/>
Sabado mañana	<input checked="" type="checkbox"/>
Sabado tarde	<input checked="" type="checkbox"/>
Domingo mañana	<input type="checkbox"/>
Domingo tarde	<input checked="" type="checkbox"/>
<input type="text" value="Introduce información extra"/>	
<input type="button" value="REINICIAR"/>	<input type="button" value="ENVIAR"/>

Disponibilidad semana 02/09/19	
Disponibilidad:	
Viernes tarde	<input type="checkbox"/>
Sabado mañana	<input type="checkbox"/>
Sabado tarde	<input type="checkbox"/>
Domingo mañana	<input type="checkbox"/>
Domingo tarde	<input type="checkbox"/>
<input type="text" value="Introduce información extra"/>	
<input type="button" value="REINICIAR"/>	<input type="button" value="ENVIAR"/>

Disponibilidad

Disponibilidad ya enviada

Resultó ser un módulo complicado porque se tenía que traducir de los switches de la pantalla a un xml para insertarlo en base de datos (y que posteriormente lo leyese el administrador), y tener especial cuidado con las fechas. Es importante para el usuario darse cuenta de que, pese a que en la mayoría de pantallas se ven datos actuales, en este módulo se envía la disponibilidad de la semana siguiente. Por eso se pone en el título la fecha a la que corresponde la disponibilidad enviada.

Una vez es enviada la disponibilidad de una semana no se puede volver a enviar hasta la semana siguiente. Para ello, cada vez que se inicia la pantalla se busca si el árbitro ya ha enviado la disponibilidad de esta semana con una consulta en base de datos. Si es así, se bloquean todos los elementos de la pantalla. Al principio del desarrollo de esta pantalla, se diseñó para que no se pudiese entrar en la pantalla más que una vez por semana pero se modificó más tarde para que los árbitros pudiesen comprobar la disponibilidad que habían enviado.

- Confirmar nóminas y resultados (administrador):

Se escogió el verbo confirmar en el diagrama de casos de uso porque la idea inicial era visualizar las nóminas y resultados enviados y que no se hiciesen oficiales hasta que no los supervisase un administrador, pero finalmente solo se les da la opción de visualizar el resultado (y en el caso de las nóminas de lanzar el cálculo de éstas).

[Consultar resultados](#)

Resultados semana 28/10/19		↶ ...
No se ha jugado ningún partido este fin de semana		

Resultados semana 26/08/19		↶ ...
Lista de partidos:		
Filtrar por categoría		
Monzon 15 - 76 Fleta	SOC. PLATA	
Pabellon Monzon	30/08/19 11:00	
Olivar 14 - 25 Casablanca	SOC. BRONCE	
Casa	31/08/19 12:00	

[Consultar nóminas](#)

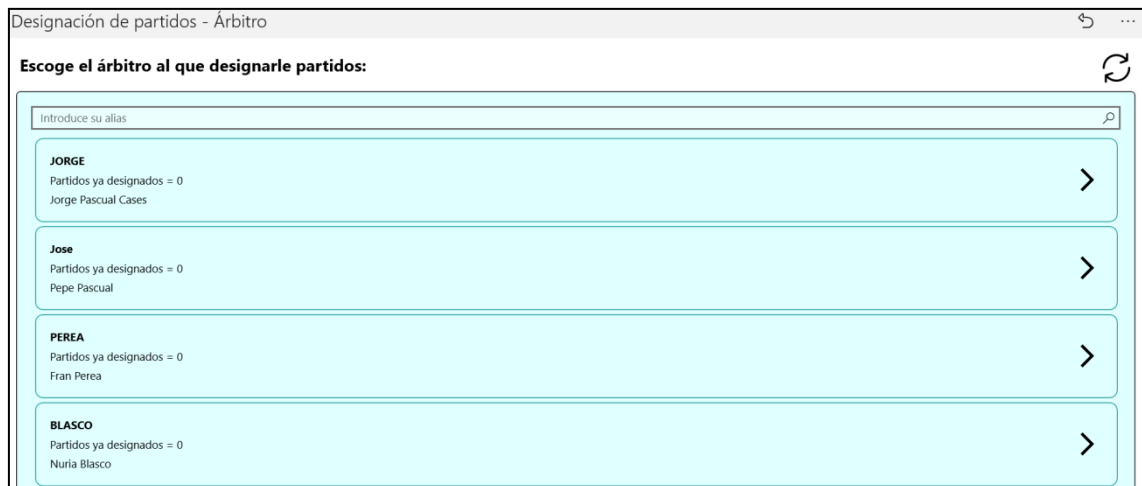
Nóminas agosto		↶ ...
No hay nóminas para visualizar este mes		

Nóminas junio		↶ ...
Lista de nóminas:		
Filtrar por alias		
PEREA	Del 1 al 30 de junio de 2019	15.00 €
BLASCO	Del 1 al 30 de junio de 2019	32.00 €
SANCHEZ	Del 1 al 30 de junio de 2019	25.00 €
		TOTAL: 72.00 €

- Designar partidos(Administrador):

La designación de partidos se organizó de esta manera:

Primero se elige el árbitro al que le quieres designar algún partido entre una lista de árbitros (con disponibilidad).



Y te lleva a la siguiente pantalla donde se puede consultar la disponibilidad (última función de los casos de uso) y se ve una lista con los partidos ya designados y otra con los que se le podrían designar.



En estas pantallas se realizaron numerosas pruebas asignando y desasignando partidos con los botones de arriba (el de guardar y la cruz) y hubo que cambiar el procedimiento almacenado varias veces para que tuviese en cuenta bien la disponibilidad. Como mejora para el futuro se sugiere la modificación del procedimiento para que en cuanto asignes un partido se borren de la lista de posibles los que sean dos horas antes o después y no estén en la misma ubicación.

En el resto de pruebas han ayudado mucho dos elementos del desarrollo. El primero fue un fichero de log en el que se escribe cualquier excepción que suceda en la aplicación. Y el segundo es el uso del lanzamiento de excepciones hasta la clase asociada a la pantalla, desde la que se puede lanzar una alerta visual con el error, para que lo vea el usuario. Sin embargo, esto último funciona en todos los casos excepto en los que el error se genera cuando se está cargando la pantalla y es un fallo que habrá que investigar en el futuro para solventarlo.

```
524 8/26/2019 9:34:02 PM: Error al insertar arbitro en BD:String or binary data would be truncated.
525 The statement has been terminated.
526 at System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction)
527 at System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction)
528 at System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose)
529 at System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj, Boolean& dataReady)
530 at System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString)
531 at System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, Boolean async, Int32 timeout, Task& task, Boolean asyncWrite, SqlDataReader ds)
532 at System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, TaskCompletionSource`1 completion, Int32 timeout, Task& task, Boolean asyncWrite, String method)
533 at System.Data.SqlClient.SqlCommand.InternalExecuteNonQuery(TaskCompletionSource`1 completion, Boolean sendToPipe, Int32 timeout, Boolean asyncWrite, String methodName)
534 at System.Data.SqlClient.SqlCommand.ExecuteNonQuery()
535 at Federacion.Helpers.DatabaseHelper.CrearArbitro(ArbitroDTO arbitro)
536
537
538 8/26/2019 9:38:02 PM: Error al cargar los admins: Data is Null. This method or property cannot be called on Null values.
539 at System.Data.SqlClient.SqlBuffer.get_String()
540 at System.Data.SqlClient.SqlDataReader.GetString(Int32 i)
541 at Federacion.Helpers.DatabaseHelper.CargarTodosArbitros()
542 8/26/2019 9:43:15 PM: Error al cargar los admins: Data is Null. This method or property cannot be called on Null values.
543 at System.Data.SqlClient.SqlBuffer.get_String()
544 at System.Data.SqlClient.SqlDataReader.GetString(Int32 i)
545 at Federacion.Helpers.DatabaseHelper.CargarTodosArbitros()
```

Ejemplo Fichero Log

3. Bibliografía

Librería SqlClient:

<https://docs.microsoft.com/es-es/dotnet/api/system.data.sqlclient?view=netframework-4.8>

Documentación Xamarin:

<https://docs.microsoft.com/es-es/xamarin/>

Documentación Azure:

<https://docs.microsoft.com/es-es/azure/>

Ejemplo servidor Azure:

<https://github.com/brminnick/XamList>

Ejemplos aplicaciones Xamarin:

<https://developer.xamarin.com/samples>

Logos utilizados en la aplicación:

<https://www.flaticon.es/>

<https://www.flaticon.com/authors/freepik>