



FrancisGol

FrancisGol

Memoria

Presentado por:

Francisco Rodríguez Belmonte



índice

1 - Análisis software	1
1.1 - Requisitos funcionales	1
1.2 - Requisitos no funcionales	3
1.3 - Casos de uso	4
2 - Diseño lógico	9
2.1 - Arquitectura MVC	9
2.2 - Modelo Entidad-Relación	12
2.3 - Diagrama de clases	13
3 - Diseños Web	14
3.1 Guía de estilos	14
3.2 Colores	14
3.3 Tipografía	17
3.4 Iconos	19
3.5 Navegación	20
3.6 Diseños de alta fidelidad	21



índice

4 - Implementación	22
4.1 - Aspectos destacables	22
4.1.1 - API	22
4.1.2 - Casos de uso	26



Análisis software

Requisitos funcionales

RF-1. Visualización

RF-1.1. Visualizar datos de partidos.

RF-1.1.1. Visualizar estadísticas del partido.

RF-1.1.2. Visualizar resultados.

RF-1.1.3. Visualizar alineaciones.

RF-1.1.4. Visualizar eventos del partido.

RF-1.2 Visualizar competiciones.

RF-1.2.1. Visualizar la clasificación de la competición.

RF-1.2.2. Visualizar las jornadas de la competición.

RF-1.2.3. Visualizar los equipos de la competición.

RF-1.2.4. Visualizar datos de competiciones por año.

RF-1.3. Visualizar fichajes.

RF-1.3.1. Visualizar fichajes de equipos.

RF-1.4. Visualización de favoritos.

RF-1.4.1. Visualizar datos de la competición o equipo favorito.

RF-1.4.2. Visualizar eventos del partido.

RF-1.4.3. Visualizar fichajes de equipos favoritos.



Requisitos funcionales

RF-1.5. Visualizar equipos.

RF-1.5.1. Visualizar la clasificación del equipo por competición.

RF-1.5.2. Visualizar estadísticas del equipo por competición.

RF-1.5.3. Visualizar plantilla del equipo.

RF-1.5.4. Visualizar los últimos fichajes del equipo.

RF-1.5.5. Visualizar datos de del equipo por año.

RF-1.6. Visualización de jugadores.

RF-1.6.1. Visualización de datos del futbolista.

RF-1.6.2. Visualización de estadísticas del futbolista.

RF-1.6.3. Visualizar datos de del futbolista por año.

RF-2. Gestión de favoritos.

RF-2.1. Añadir equipos/competiciones en favoritos.

RF-2.2. Eliminar equipos/competiciones de favoritos.

RF-3. Gestión de plantillas.

RF-3.1. Crear alineación de un equipo.

RF-3.2. Borrar alineaciones propias.

RF-3.3. Editar alineaciones propias.

RF-3.4. Visualizar alineaciones de otros usuarios.

RF-4. Gestión de usuario.

RF-4.1. Crear usuario.

RF-4.2. Editar usuario.

RF-4.3. Borrar usuario.

RF-4.4. Iniciar sesión.

RF-4.5. Cerrar sesión



Requisitos no funcionales

RF-1. Usabilidad

RF-1.1. Mostrará mensajes de error

RF-1.2. Intuitiva y fácil de usar

RF-1.3. Buena legibilidad

RF-1.4. Fácil de navegar entre las distintas páginas

RF-1.5. Capaz de llegar a la misma información desde diferentes puntos

RF-2. Funcionalidad.

RF-2.1. Las plantillas solo se podrán crear con equipos existentes

RF-2.2. Para acceder a un jugador hay que acceder a su equipo

RF-2.3. Para crear plantillas hay que estar registrado

RF-3. Seguridad

RF-3.1. contraseñas de usuarios cifrada

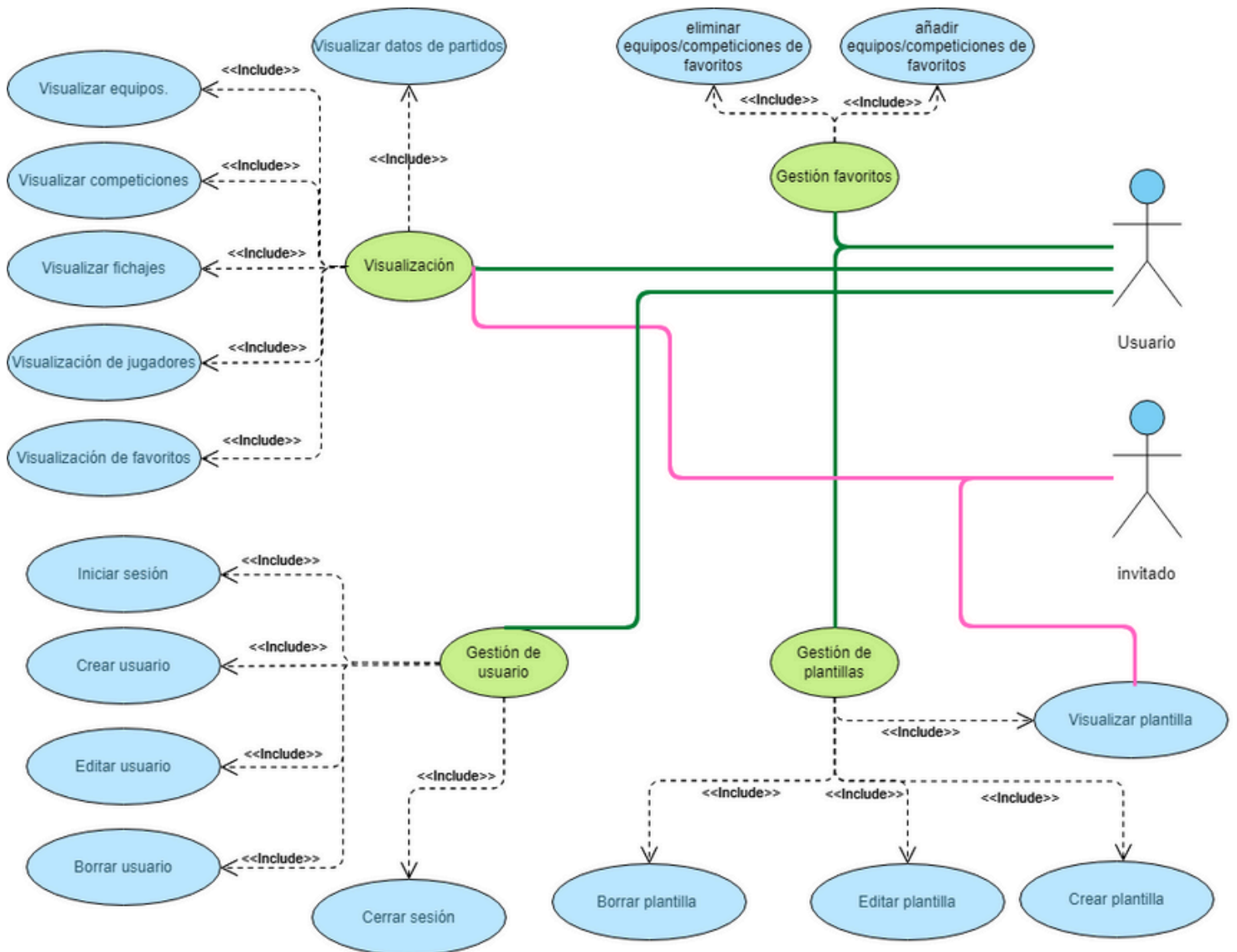
RF-3.1. La página será segura con certificado SSL

RF-4. Disponibilidad

RF-4.1. Se podrá acceder a la página a través de francisgol.com

Casos de uso

Diagrama de casos de uso





Casos de uso

Descripción de los casos de uso

CU-1	Visualización de información de la página
Descripción	Permite visualizar tanto a usuarios como a invitados las competiciones, equipos, plantillas, jugadores y partidos.
Actores	Usuarios e invitados
Precondiciones	Se debe acceder a las secciones de de la página correspondiente a lo que se quiere visualizar
Postcondiciones	Al acceder el usuario o invitado puede ver la información
Escenario principal	<ol style="list-style-type: none">1. Los usuarios o invitados acceden a la página2. Viendo los partidos del momento3. En la parte superior habrá secciones4. Al pulsar en una sección le llevará a la página correspondiente5. Dentro de las secciones habrán más apartados que les permitirá navegar por la página.



Casos de uso

Descripción de los casos de uso

CU-2	Gestión de favoritos
Descripción	El usuario puede guardar en favoritos equipos y competiciones para poder acceder más fácil.
Actor	Usuario
Precondiciones	Se debe registrar e iniciar sesión para poder dar favoritos.
Postcondiciones	Al dar favoritos las secciones serán personalizadas en base a los equipos y competiciones favoritas.
Escenario principal	<ol style="list-style-type: none">1. El usuario se registra.2. Durante el registro selecciona equipos y competiciones favoritas.3. Visita las secciones de partidos/competiciones/fichajes que estarán personalizadas4. Puede seguir añadiendo y quitando equipos y competiciones en favoritos.



Casos de uso

Descripción de los casos de uso

CU-3	Gestión de plantillas
Descripción	El usuario puede crear borrar y editar plantillas de equipos.
Actor	Usuario
Precondiciones	Se debe registrar e iniciar sesión para poder crear, editar y borrar plantillas
Postcondiciones	Al crear una plantilla se vuelve pública para que todo el mundo la vea.
Escenario principal	<ol style="list-style-type: none">1. El usuario se registra.2. Inicia sesión.3. Se dirige a la sección principal de plantillas4. Puede visualizar o crear plantillas.5. Crea una plantilla6. Se publica y ya puede editarla o borrarla

Casos de uso

Descripción de los casos de uso

CU-4	Gestión de usuario
Descripción	El invitado puede crearse una cuenta para obtener privilegios en la aplicación y el usuario puede editar o borrar la cuenta creada
Actor	Usuario e invitado
Precondiciones	Se debe acceder a la página de registro de FrancisGol
Postcondiciones	Al crearse una cuenta se le desbloquea el uso de creación de plantillas y la personalización de información
Escenario principal	<ol style="list-style-type: none"> 1. El invitado accede a la página de registro. 2. Se registra e inicia sesión 3. Como usuario puede acceder al apartado de la cuenta y modificar los datos del usuario o borrar la cuenta

Diseño lógico

Arquitectura MVC

La Arquitectura MVC se basa en 3 partes

- Los controladores: Se encargan de hacer de intermediario entre la vista y el modelo, pide datos a los modelos y devuelve los resultados en las vistas.
- Los modelos: Llevan toda la lógica del programa. es el que hace los cálculos, bucles... Da información al controlador
- Las vistas: Llevan el código HTML y se encarga de mostrar el contenido de la página.

En FrancisGol se va a utilizar este modelo ya que permite tener el código más estructurado.

Lo ideal es ponerle el mismo nombre a la vista, al modelo y al controlador para así saber que archivo pertenece a cada página.

En la próxima página se verá que archivos voy a utilizar para hacer el modelo vista controlador.

Diseño lógico

Arquitectura MVC

Los archivos serán uno por cada página:

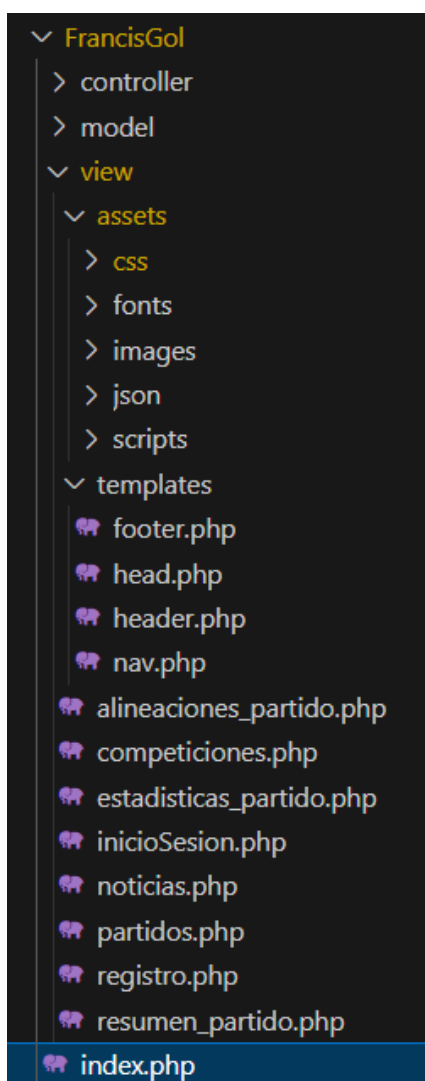
partidos.php
resumen_partido.php
estadísticas_partido.php
alineaciones_partido.php
competiciones.php
clasificacion_competicion.php
jornadas_competicion.php
equipos_competicion.php
fichajes.php
mis_plantillas.php
plantillas_usuarios.php
crear_plantillas.php
inicio_sesion.php
registro.php
competiciones_equipo.php
estadísticas_equipo.php
plantilla_equipo.php
fichajes_equipo.php
datos_jugador.php
estadísticas_jugador.php
cuenta_favoritos.php
cuenta_editar.php

Diseño lógico

Arquitectura MVC

Eso serían las páginas disponibles en la web pero siempre habrán más archivos, como modelos para recoger solicitudes de las APIs o controladores para cerrar sesión o realizar alguna operación como editar o borrar algo.

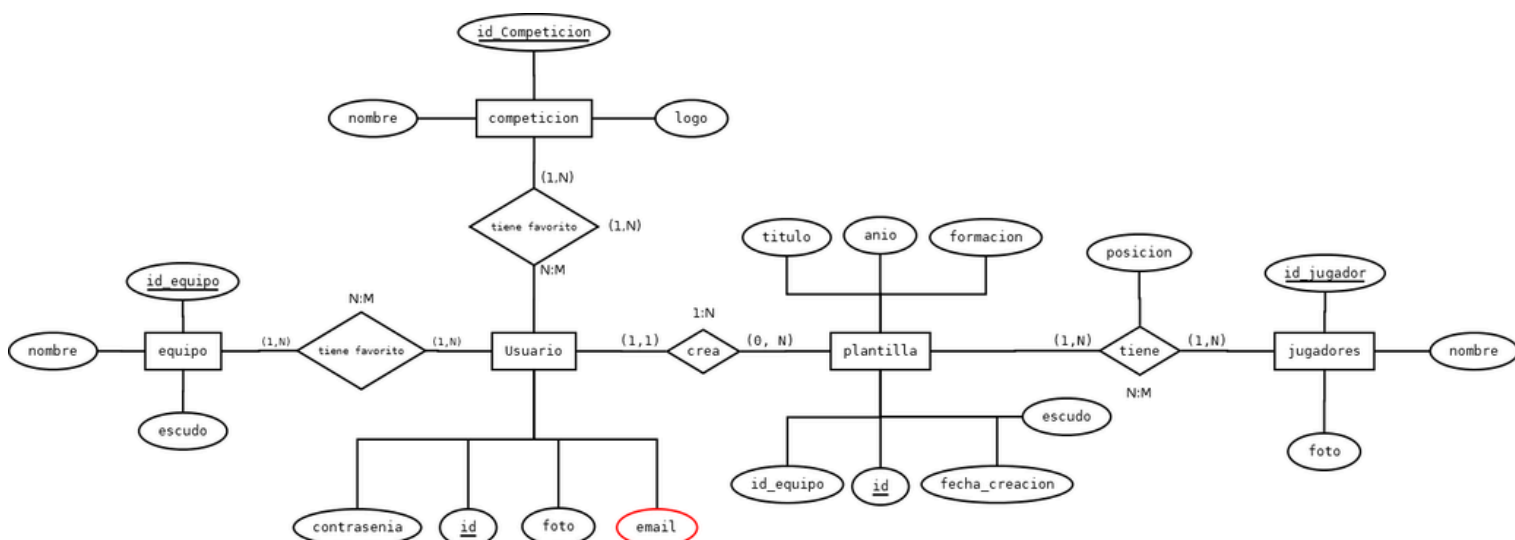
Al terminar quedaría algo así:



Diseño lógico

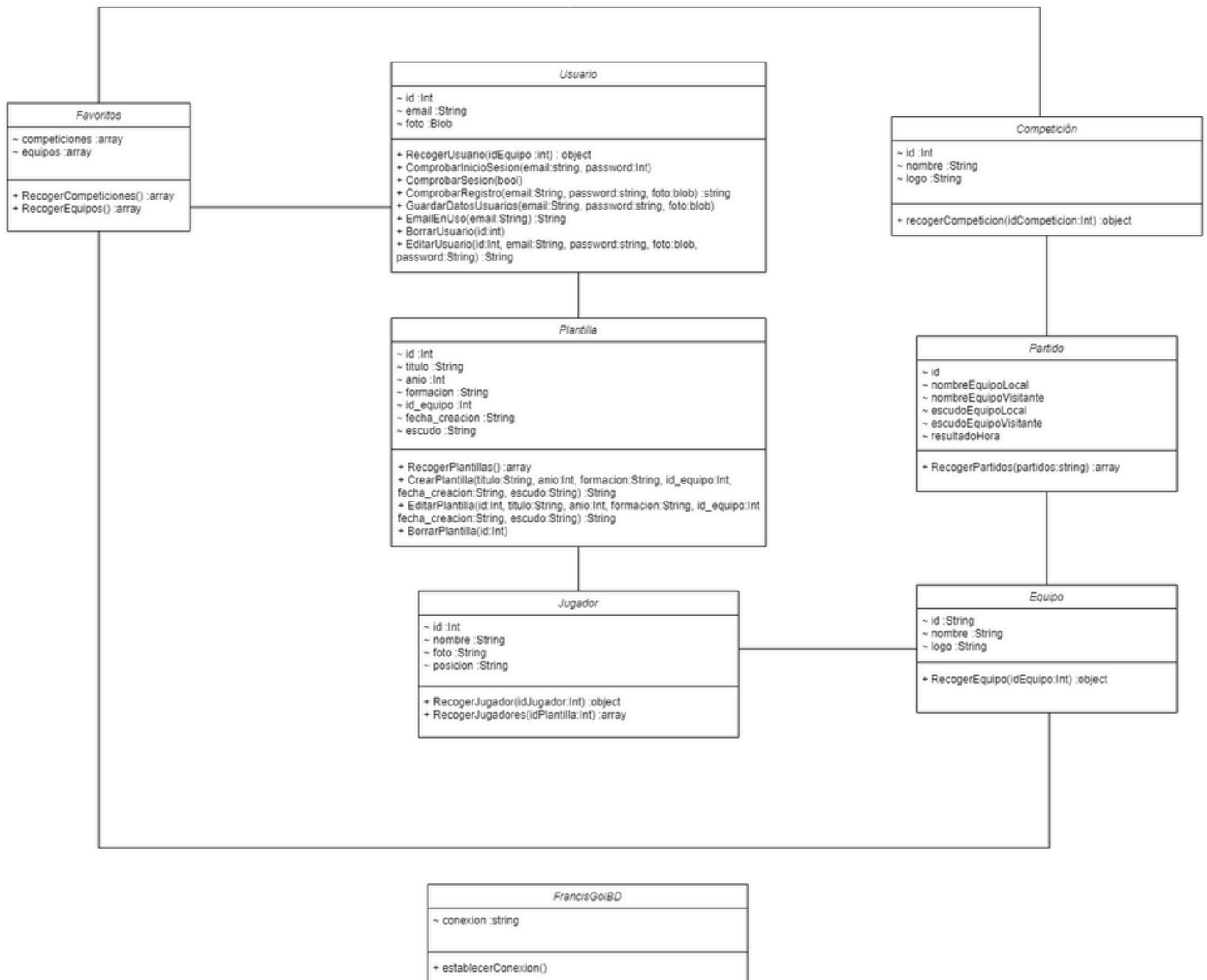
Modelo Entidad-Relación

En la BBDD se va a almacenar lo necesario para ahorrar solicitudes a la API ya que toda la información que llega no se va a almacenar ya que es información que va continuamente cambiando, en el caso de las plantillas la información es estática y en el caso de competiciones o equipos favoritos guardo lo necesario para mostrarlos sin tener que hacer una solicitud a la API.



Diseño lógico

Diagrama de clases





Diseño Web

Guía de estilos

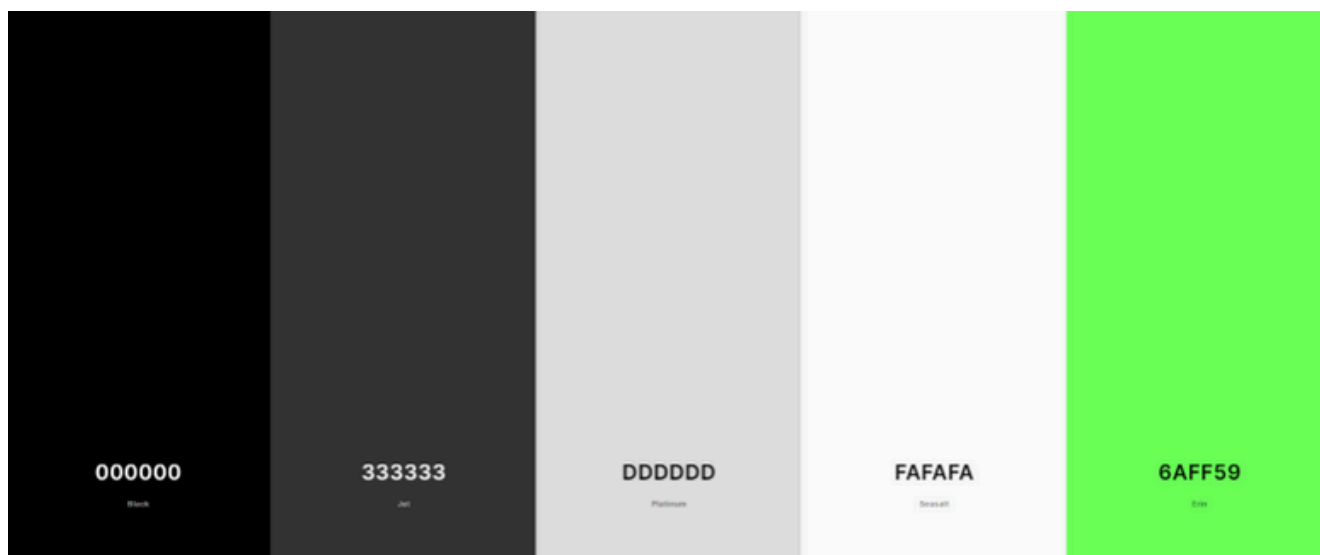
En la guía de estilos se podrá ver todo lo visualmente relacionado con la página FrancisGol, ahí se pueden ver los colores, las fuentes, los diseños...

Para verla correctamente puede ir al siguiente enlace en el que tiene la guía de estilos completa con las explicaciones:

[Enlace](#)

Colores

- Colores principales





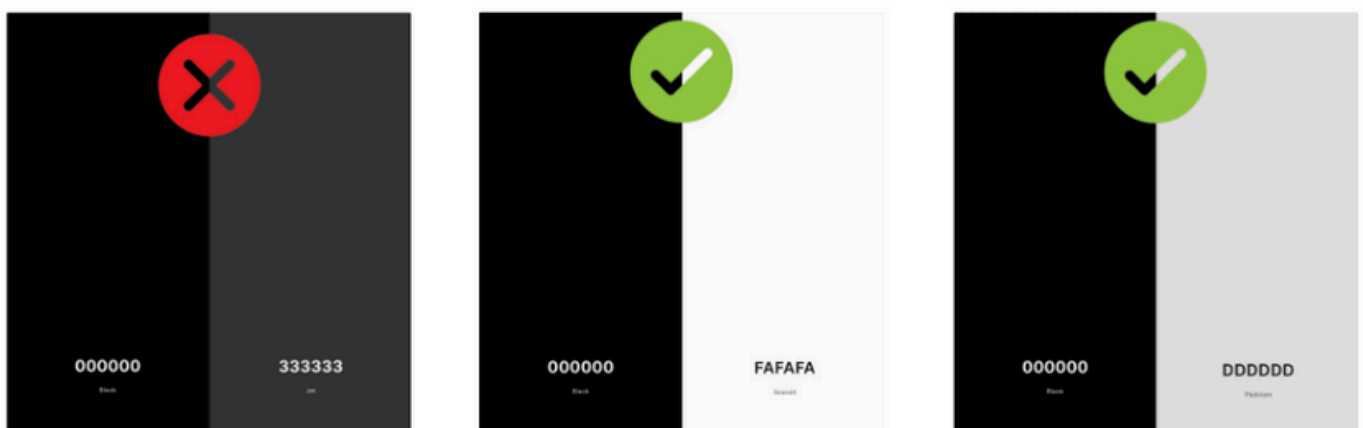
Diseño Web

colores

- Colores secundarios



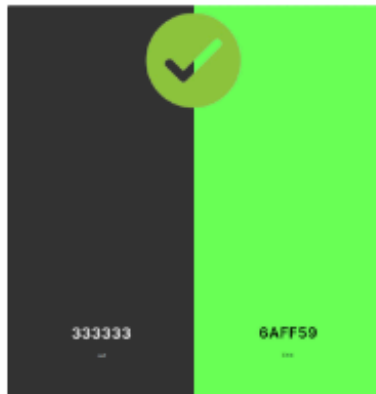
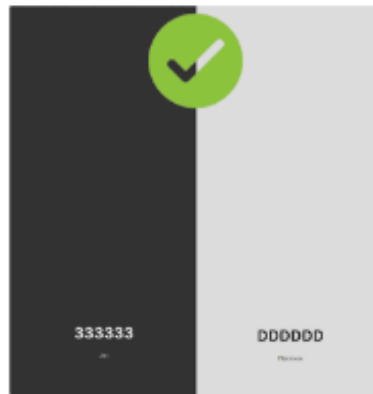
- Unión de colores



Diseño Web

colores

- Unión de colores



Fondo gris con color de letra blanca





Diseño Web

Tipografía

- Bebas Neue

Utilizada para los títulos de las secciones en las que se encuentra en la página, he seleccionado este tipo de letra en mayúsculas y sin serifa, este tipo de letra es normalmente utilizado en páginas de deporte. Este tipo de letra da un buen contraste con los tipos de letra Sans Serif.

Bebas Neue

Designed by Ryochi Tsunekawa

WHEREAS DISREGARD AND CONTEMPT FOR HUMAN RIGHTS HAVE RESULTED

ESTADÍSTICAS DEL PARTIDO
NOTICIAS
PLANTILLAS

Diseño Web

Tipografía

- Manrope

Utilizada para textos, botones, secciones y cualquier otro elemento que no sean títulos. El estilo de este tipo de letra es Sans Serif lo que facilita su legibilidad lo que permite leer correctamente noticias, marcadores, estadísticas y todo lo relacionado con mostrar información.

Manrope

Designed by Mikhail Sharanda

Whereas disregard and contempt for human rights
have resulted

Messi da la victoria a su equipo. Atlético de Madrid



Diseño Web

Iconos



Logo de FrancisGol - Representa la F de Francisco/Fútbol y el punto hace referencia al balón de fútbol.

Iconos de los resúmenes de los partidos



Gol



Sustitución



Tarjeta
amarilla



Tarjeta
roja



Partido en
directo

Otros iconos



En favoritos



Sin favoritos



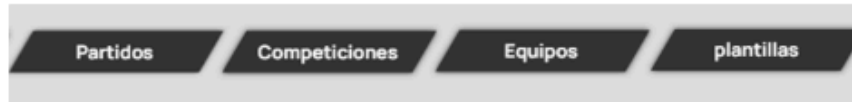
Menú
hamburguesa



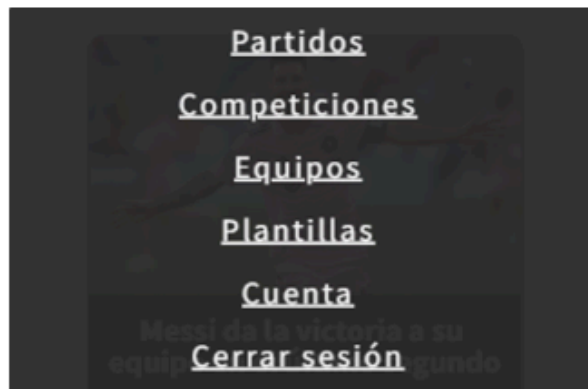
Menú
hamburguesa
desplegado

Diseño Web

Navegación



Sección principal con los apartados que permite la Web



Menú hamburguesa de sección principal



Inicio de sesión y registro



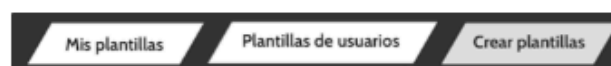
Opciones de los partidos



Opciones de un club



Opciones de un club



Opciones de plantillas

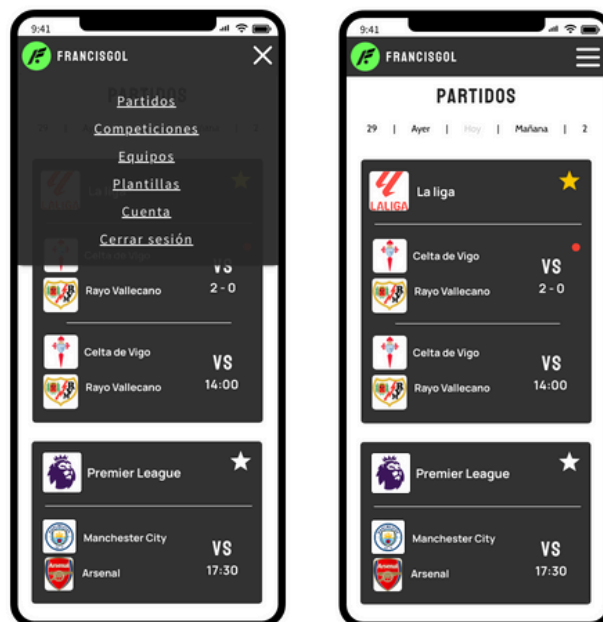
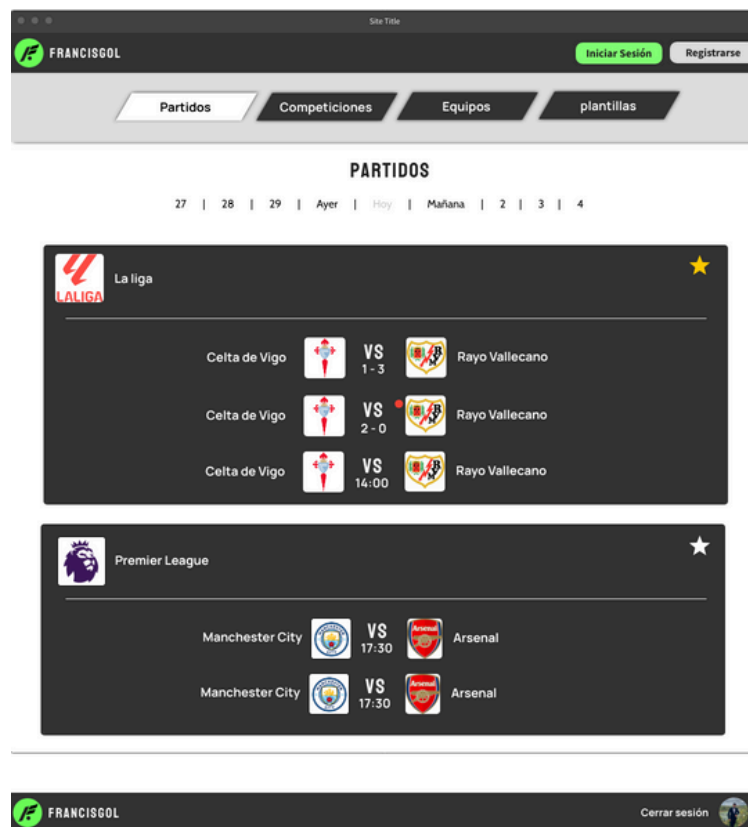


Opciones de jugador

Diseño Web

Diseños de alta fidelidad

Para ver todos los diseños al completo diríjase al siguiente [Enlace](#).



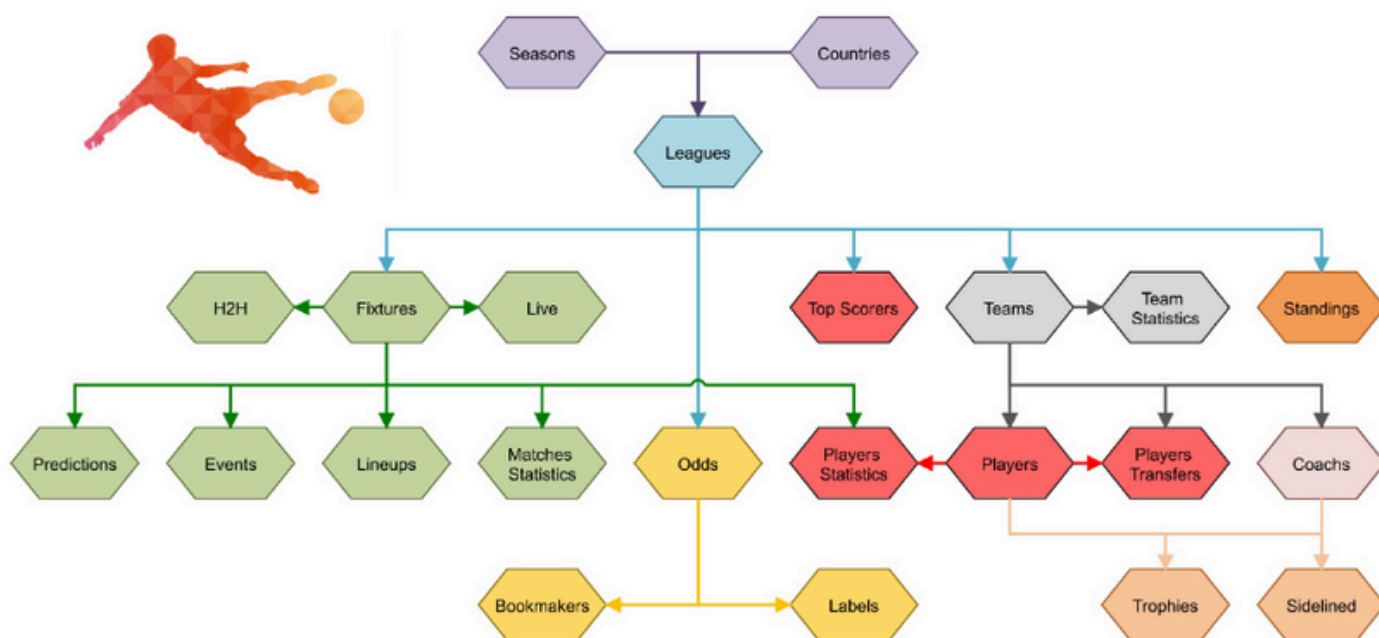
Implementación

Aspectos destacables

En este punto se verán todos los problemas que han surgido y recursos que se han utilizado para crear FrancisGol

- API

Uno de los mayores condicionante en la aplicación a la hora de crear la página es la API de api-football ya que es la que me da toda la información mostrada en la APP.



Implementación

Aspectos destacables

- API

En la estructura de datos de la página anterior se puede ver como vienen los datos de la API y como están estructurados, el problema principal al crear la aplicación es que si por ejemplo quiero acceder a los jugadores primero tengo que saber el equipo al que pertenece y para saber el equipo al que pertenece necesito tener la liga en la que está, lo que provoca no poder acceder directamente a los equipos y jugadores.

El siguiente problema que se genera por la API es el número de peticiones por día, en este caso son 100 peticiones al día, lo que provoca que si se accede a muchas páginas se pierdan las peticiones y no se pueda acceder a los datos, para evitar el gasto masivo de la API, para solucionarlo se guardan los equipos, las competiciones y los jugadores en la base de datos permitiendo que una vez que se consulten ya siempre los recoja de la base de datos, pero eso solo si se quiere los datos básicos(id, nombre, logo) si se quieren datos más específicos como estadísticas y demás se guardan en JSON que se generan a medida que se hacen las consultas y se actualizan en el tiempo indicado, un día por lo general.

En la siguientes páginas se puede ver una foto del código que guarda en la BBDD los datos de las competiciones y el archivo que realiza las consultas a la API y genera los JSON.

Implementación

Aspectos destacables

- API

Comprueba si el fichero JSON existe, si existe y fue creado hace menos de un día recoge la información del JSON, si no hace la petición en la API y luego genera o guarda los datos en el fichero JSON

```
function realizarConsulta($nombreJson, $rutaApi, $tiempoGuardado) {

    $archivo = "../view/assets/json/$nombreJson.json"; // Nombre del archivo

    // Verifica si existe un archivo y si fue modificado en el último día
    if (file_exists($archivo) && (time() - filemtime($archivo) < $tiempoGuardado)) {

        $datos = file_get_contents($archivo);
        $resultado = json_decode($datos);

    } else {

        $curl = curl_init();

        curl_setopt_array($curl, array(
            CURLOPT_URL => "https://v3.football.api-sports.io/$rutaApi",
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_ENCODING => '',
            CURLOPT_MAXREDIRS => 10,
            CURLOPT_TIMEOUT => 0,
            CURLOPT_FOLLOWLOCATION => true,
            CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
            CURLOPT_CUSTOMREQUEST => 'GET',
            CURLOPT_HTTPHEADER => array(
                'x-rapidapi-key: be6e260f0828d5854c973280d67305cd',
                'x-rapidapi-host: v3.football.api-sports.io'
            ),
        ));

        $response = curl_exec($curl);
        curl_close($curl);

        file_put_contents($archivo, $response);

        $resultado = json_decode($response);

    }

    return $resultado;
}
```

Implementación

Aspectos destacables

- API

Hace una consulta a la BBDD y comprueba si está la competición insertada, si lo está genera el objeto con los datos de la competición recogidos de la BBDD, si no está en la BBDD, hace la petición genera el objeto y llama a la función que inserta los datos de la competición en la BBDD

```
public static function recogerCompeticion($idCompeticion) {  
  
    $conexion = FrancisGolBD::establecerConexion();  
    $consulta = "SELECT * FROM competicion WHERE idCompeticion = $idCompeticion";  
    $resultado = $conexion->query($consulta);  
  
    if (mysqli_num_rows($resultado) == 1) {  
        while($row = mysqli_fetch_assoc($resultado)) {  
            $competicion = new Competicion($row['idCompeticion'], $row['nombre'], $row['logo']);  
        }  
    } else {  
        $competicion = realizarConsulta("competicion_$idCompeticion", "leagues?id=$idCompeticion", 86400);  
        $competicion = $competicion->response[0]->league;  
        $competicion = new Competicion($competicion->id, $competicion->name, $competicion->logo);  
        $competicion->insertarCompeticion();  
    }  
  
    return $competicion;  
}
```

Implementación

Aspectos destacables

- API

Para insertarlo en la BBDD, lo que hace es que lo intenta introducir y si se genera un error porque ya está introducido lo obvia.

```
public function insertarCompeticion() {
    $conexion = FrancisGolBD::establecerConexion();

    $idCompeticion = $this->__get("id");
    $nombreCompeticion = $this->__get("nombre");
    $logoCompeticion = $this->__get("logo");

    $consulta = $conexion->prepare("INSERT INTO competicion (idCompeticion, nombre, logo) VALUES (?, ?, ?)");

    try {
        $consulta->bind_param("iss", $idCompeticion, $nombreCompeticion, $logoCompeticion);
        $consulta->execute();
    } catch (mysqli_sql_exception) {
        // Si la competición ya está insertado no hace nada
    }
}
```

- Plantillas

Uno de los mayores problemas al crear la sección de plantillas es la forma en la que se va a mostrar y guardar los jugadores, ya que si un usuario mueve a jugadores por la plantilla se tiene que guardar la posición exacta en la que lo ha dejado, para ello se usa un sistema de clases que guarda la posición del jugador y además permite mediante GRID por CSS colocar a los jugadores en el once inicial en una posición concreta, para ello en las siguientes páginas se verá el resultado y el código de como funciona.

Implementación

Aspectos destacables

- Plantillas

Resultado



Implementación

Aspectos destacables

- Plantillas

Explicación

Los jugadores del once inicial llevan las clases generadas automáticamente a través de la formación, cada formación tiene sus clases siguiendo la siguiente estructura: 4-4-3

1- jugador_1_1

4 - jugador_2_1, jugador_2_2, jugador_2_3, jugador_2_4

3 - jugador_3_1, jugador_3_2, jugador_3_3

3 - jugador_4_1, jugador_4_2, jugador_4_3

Esta estructura lo que permite es a través de GRID colocar a los jugadores en su posición.

Los jugadores reservas llevan todos las clases desde suplentes_11 hasta suplentes_22 y los no convocados llevan las clases desde la no_convocado_23 hasta que se caben los jugadores del equipo.

Los jugadores se pueden intercambiar arrastrando o seleccionando 2, para ello cuando se cambian hay que cambiar las clases de los dos jugadores y también el jugador completo ya que si uno se encuentra dentro de la plantilla y otro fuera no se colocaría dentro del campo.



Fin

