

# Presentación Técnica

## Previo al desarrollo del proyecto

La aplicación marca como objetivo crear una página que permita a los usuarios, en su mayoría consumidores de contenido de deporte electrónico, crear sus equipos ideales y competir con sus amigos o grupos sociales y comerciales para descubrir quién es capaz de formar el mejor conjunto de jugadores.

La idea principal es crear una aplicación de funcionamiento sencillo que permita a los usuarios que estén interesados en usar la aplicación un manejo fácil, entendible y visualmente satisfactorio.

Desde un punto de vista comercial, nuestra aplicación permite a las marcas interesadas en formar parte de nuestros servicios, crear salas para la participación de grandes números de personas con premios para atraer y llegar a sus seguidores.

Con estas ideas, desarrollamos un pensamiento previo sobre tecnologías y métodos que podemos utilizar para desarrollar la aplicación de manera que fuera ampliable y de fácil manipulación.

El debate se inició al pensar en la base de datos que queríamos utilizar, la cantidad de datos que debíamos ofrecer y la regularidad con la que estos podían ser actualizados o modificados.

Al darnos cuenta de que los registros eran exponenciales, es decir, cada juego tiene obligatoriamente ligas y que a su vez esta tiene equipos y cada uno jugadores, rápidamente nos dimos cuenta que debíamos utilizar alguna aplicación superior. Por ello, descartamos MongoDB y utilizamos finalmente phpmyadmin/mysql.

Una vez aclarado el punto anterior, pensamos en la distribución del contenido y como podría ser modificado de manera sencilla y adaptada a la base de datos de la manera más cómoda posible.

Tras una conversación no muy larga, elegimos usar como herramienta una API que nos habían recomendado los profesores. Esta está desarrollada por “*mevdschee*” y se ha utilizado de manera limitada en nuestra web. Su funcionalidad se basa en mostrar contenido que puede ser maximizado o ampliado. Esto nos permite que, al

introducir nuevo contenido en la base de datos, como por ejemplo, nuevos equipos, jugadores etcétera, el resultado se vea de manera inmediata en la web.

## Sprint 1.

### Objetivos a realizar.

El primer sprint giró en torno a crear una visualización sencilla, dinámica y escalable del contenido que fuera siendo introducido en la base de datos. Los puntos indicados en el Jira, giraron entorno a este primer objetivo.

### Transcurso del sprint.

Con la idea anterior, nos pusimos a investigar métodos eficientes para realizarlo. Nuestra decisión final giró entorno al uso de componentes en Vue. La razón era sencilla, nuestro contenido era escalable y contenía varios elementos iguales en importancia y contenido, por lo que debían seguir una estructura idéntica.

Así pues, combinando Vue con la Api Crud, desarrollamos un primer despliegue del contenido de la base de datos.

Los elementos de dicha base de datos constaban de varios juegos, algunas de las ligas más relevantes y equipos de la más cercana a nosotros con sus jugadores.

Así pues, utilizando ambas tecnologías, el código resultó sencillo, compacto y eficiente.

```
getJuegos: function() {
  document.getElementById("seleccionJugadores").style.display = "none";
  document.getElementById("filtros").style.display = "none";
  axios.get('http://esports-madness.electronica-garcilaso.cat/API/api.php/records/Juegos')
    .then(response => {
      this.juegos = response.data.records,
      console.table(response.data);
    })
    .catch(error => console.error(error));
},
```

Para crear el contenido resultante, es decir, cada Juego mostrará el contenido de si mismo, relacionamos la petición de la Api con las claves foráneas, de esta manera el contenido se creó acorde al objeto al que hacíamos referencia.

```

getLigas: function(ID) {
  this.equipos = null;
  this.jugadores = null;
  this.entrenadores = null;
  axios.get('http://esports-madness.electronica-garcilaso.cat/API/api.php/records/Ligas?filter=Juego,eq,' + ID)
    .then(response => {
      this.ligas = response.data.records,
      console.table(response.data);
    })
    .catch(error => console.error(error));
},

```

Como bien se indica, la petición Axios a la Api contiene un filtro que nos proporciona las ligas del juego al que hemos accedido.

```

/Ligas?filter=Juego,eq,' + ID

```

Tras rellenar el contenido de la base de datos y crear esta primera imagen del contenido que queríamos mostrar en la web de manera que pudiera ser exponencial, habíamos terminado nuestra idea de contenido a realizar durante este sprint a falta de dos días para la finalización del mismo.

Por ello, añadimos las funcionalidades que teníamos pensadas para el próximo, en este caso, sabiendo que no terminaríamos a tiempo y que formarían parte del próximo.

En estas nuevas tareas se incorporó la creación del registro y login en la web y una modificación a la hora de mostrar el contenido de nuestra web.

Estas tareas empezaron su proceso durante este periodo, en el caso de la actualización visual, fue realizada con éxito creando una división entre ligas y contenido seleccionable para guardado.

En el caso del registro y login, el trabajo no pudo ser finalizado, pero no resultó problema puesto que era parte de la idea ya concebida.

Se procedió a la subida al servidor del contenido tras ver su funcionamiento en local y el proceso no tuvo ninguna complicación.

## Resultado y decisiones.

El transcurso de este primer periodo nos mostró que íbamos por buen camino, el contenido deseado estaba ya siendo mostrado y el proceso de registro en el horno.

Por ello, decidimos preparar el próximo sprint en base a ello, poniendo alguna tarea más que inicialmente.

## Sprint 2.

### **Objetivos a realizar.**

Para este sprint, el objetivo era crear un login y registro que funcionara y la posibilidad de elegir jugadores por parte del usuario, además de un perfil para el usuario acorde con los datos que debían ser introducidos durante el registro.

### **Transcurso del sprint.**

El principio de este periodo no fue preocupante, pero según pasaban los días el proceso de creación de la fase de registro y logeo se iba retrasando más de lo esperado. Nuestra idea principal fue utilizar el sistema de registro y login de Google, pero tras investigar y no conseguir las claves de acceso, procedimos a crearlo de manera manual.

Mientras se intentaba encontrar una manera para realizar el apartado anterior, se siguió desarrollando el resto de ideas para este sprint.

El desarrollo de la interfaz de perfil para el usuario fue satisfactoria. Utilizamos un registro directamente creado en la base de datos para crear el contenido de manera dinámica, utilizando la Api ya mencionada anteriormente.

La pantalla perfil, estaba preparada para, una vez funcionara el registro y relacionado el usuario conectado con la aplicación, mostrara contenido específico para cada usuario.

Mientras tanto..

Tras mucha reflexión y proceso de visualización de código, encontramos un método que funcionaría y se puso en proceso de creación.

Situados a un día de la finalización del segundo periodo, el registro y login funcionaba correctamente a nivel local, así pues, procedimos a su subida a servidor, chocando con el primer gran problema.

Al realizar la subida, el código no funcionaba en el servidor.

Por ello, tuvimos que pensar varias razones por las que podía estar causado el error.

En primera instancia, pensamos en la posibilidad de un error relacionado con el servidor web. Al ser un hosting compartido quizás no permitía la realización de ciertas peticiones que nosotros estábamos realizando.

Así pues, nos pusimos en contacto con nuestro gestor web para preguntar si alguna de las peticiones que estábamos intentando lanzar contra el servidor no estaban permitidas, y tras una conversación online con un técnico, llegamos a la conclusión de que aquél no era el error.

Nuestra segunda opción fue comprobar las líneas de código mysql, quizás alguna estaba funcionando de manera diferente a la esperada.

Tras un vistazo lento y comprobando el resultado que ofrecían al ser directamente ejecutadas en el servidor de phpmyadmin, llegamos a la conclusión que estaba teniendo un problema de interpretación del contenido creado con "PrepareStatment". Así pues, cambiamos todo el contenido, todas las peticiones por código sql más simple.

Como resultado, las peticiones funcionaron correctamente en el servidor y así solucionamos el primer problema.

### **Resultado y decisiones.**

El proceso de solucionado de la pestaña de login y registro nos ralentizó en la propuesta de ideas iniciales que habíamos tenido. Así pues, varias de las actividades destinadas a este segundo sprint tuvieron que ser movidas al tercero.

## **Sprint 3**

### **Objetivos a realizar.**

Este sprint tenía como base, gracias al funcionamiento del registro y login, crear partes de código relacionando el usuario actual con el guardado de datos y muestra de información.

Así pues, la idea era crear sesiones de usuario que nos permitieran manipular el contenido acorde a ello.

### **Transcurso del sprint.**

El primer objetivo era crear variables *session* y poder acceder a ellas desde diferentes puntos de la aplicación.

Así, decidimos utilizar la variable `$_SESSION` para guardar el contenido deseado. En este caso, un identificador del usuario registrado para poder acceder a él de manera fácil.

Para acceder a esta variable desde toda la parte PHP no tuvimos ningún problema. Solo debíamos acceder al contenido utilizando una función llamada `session_start()`, el problema surgió al necesitar acceder a este valor desde el javascript.

La solución que se nos ocurrió, fue guardar esta variable en un elemento html para poder acceder a él utilizando las funciones `getElement`. Al ser información no susceptible a mal uso, no crea ningún tipo de problema al usuario y nos facilita su manejo desde toda la web.

```
session_start();
$misesion = $_SESSION['user_id'];
print_r("<h1 id='sess' class='". $misesion . "'></h1>");
```

Con esto funcionando, el desarrollo de los perfiles fue sencillo y el guardado de jugadores seleccionados también se desarrolló de manera eficaz.

Relacionado con el guardado de jugadores, decidimos preparar el código para los diferentes casos posibles.

Para la selección de un nuevo usuario, se debía de realizar una inserción en base de datos. En cambio, para un usuario que ya tiene sus jugadores y que decide cambiar, la opción debía ser una modificación a la ya escogida.

Así pues, antes de proceder a una u otra opción, procedemos siempre a comprobación de si el usuario tiene ya una elección anterior.

Para el perfil, como bien hemos indicado antes, guardar la variable de sesión como contenido de una clase nos permite fácilmente acceder a los datos deseados.

```
getInfoUser: function() {  
    var mises = document.getElementById("sess").getAttribute("class");  
    axios.get('http://esports-madness.electronica-garcilaso.cat/API/api.php/records/Usuarios?filter=ID,eq,' + mises)  
        .then(response => {  
            // You, 19 days ago * Creado Perfil Usuario Con info  
            this.perfilUsuario = response.data.records,  
            console.table(response.data);  
        })  
        .catch(error => console.error(error));  
},
```

## Resultado y decisiones.

Tras finalizar esta penúltima fase, el proceso debía girar entorno a solucionar errores conocidos y permitir algún tipo de relación entre usuarios.

Por suerte, todo el contenido esperado para este tercer periodo fue terminado a tiempo y no hubo ningún problema al realizar su subida al servidor.

## Sprint 4.

### Objetivos a realizar.

Al tratarse del último sprint previo a la presentación, nuestros esfuerzos giraron entorno a mejorar el contenido y el funcionamiento de la página. Nuestros objetivos principales se centraron en arreglar errores que podrían ocurrir, entre ellos, mejorar el sistema de registro poniendo filtros para que varios usuarios no pudieran registrarse con el mismo correo.

Además, queríamos poder crear salas con varios miembros que se pudieran unir mediante un código.

### Transcurso del sprint.

El inicio del periodo ya tenía las bases claras así que desarrollamos los procesos deseados sin problemas.

Por lo que hace referencia al registro, creamos un filtro para no permitir que varios usuarios pudieran registrarse con el mismo correo, ya que es el elemento que vamos a utilizar para hacer el login.

Por otro lado, creamos una codificación para la contraseña, para mejorar la seguridad de los usuarios al registrarse en nuestra web.

```
$cifrado="AES-256-OFB";  
$ivlength=openssl_cipher_iv_length($cifrado);  
$options=0;  
$encryptaiv='1234567891011121';  
$encryptkey="EsportMadness";  
$passencrypt=openssl_encrypt($pass, $cifrado, $encryptkey, $options, $encryptaiv);
```

El proceso de creación de las salas giró entorno a la variable JSON que contendrá los usuarios que forman parte de nuestra sala. Cada sala tiene un jefe de sala, que es el miembro que la formó.

Tenemos un control para que un usuario jefe de sala no pueda unirse a su propia sala.

```
$sqlUpdate ="UPDATE salasJuego SET  
Participantes = ' ' . $jsonArray . ' '  
WHERE Sala = ' ' . $codigo . ' '";  
  
if ($conn->query($sqlUpdate) === TRUE) {  
echo $sqlUpdate;  
} else {  
echo "Error: " . $sqlUpdate . "<br>" . $conn->error;  
}  
}  
}else{  
print_r("<h2 class='msgErrorUnion'>Esa sesión ya es tuya, lamentamos no poder unirte</h2>");  
}
```

Para impedir que un usuario que ya tiene sala pueda crear otra, tenemos un cambio de contenido mostrable en la pestaña de salas, haciendo que cambie de “Crear Sala” o “Mostrar Mi Sala”.

```
$sql = "SELECT JefeSala FROM SalasJuego  
WHERE JefeSala = ' ' . $mision . ' '";  
  
$result = $conn->query($sql);  
  
if($result->num_rows > 0){  
print_r('<button id="crearSala"><a class="nav-link" onclick="mostrarMiSala()">Mostrar Mi Sala</a></button>');  
}else{  
print_r('<button id="crearSala"><a class="nav-link" onclick="crearSala()">Crear Sala</a></button>');  
}
```



## **Resultado y decisiones.**

Durante este sprint, los problemas surgieron en el método de guardado vía JSON, pero no tuvimos grandes problemas al encontrar una solución realizando el encode y decode de la variable.

Llegado a este momento, decidimos también ampliar los contenidos de la base de datos en cuestiones de equipos y jugadores, para permitir a los usuarios una muestra más expansiva del contenido que se puede llegar a enseñar.

## **¿Sprint 5?**

En el caso de haber podido realizar un quinto sprint, el objetivo hubiera sido la creación de partidos, crear una muestra de puntos por enfrentamiento jugado.

La variable “puntuación final” hubiera tenido una serie de factores que habrían afectado en su subida en relación a los acontecimiento ocurridos “in game”.

El acceso a visibilizar estos puntos se realizaría desde las pestañas de salas y desde el propio perfil de jugador.

El desglose de la puntuación se mostrará por jornadas y se podría acceder a una versión más específica de los sumatorios realizados para dicho score final.

Por otro lado, este quinto sprint hubiera resultado en una mejora del aspecto visual de algunos apartados de la página web.

# Errores conocidos.

Actualmente somos conocedores de estos errores:

- El registro no requiere de un “E-Mail” propiamente dicho. Un elemento con ...@.. com / es
- El contenido no es responsive.
  - La solución en muchos apartados pasa por cambiar el grid de cinco elementos a X elementos dependiendo de la pantalla.
- La pestaña “Crear/Unirse a Sala”, cuando creamos una sala, no actualiza la página y, por ende, puede crearse más de una sala por el mismo usuario.
  - La solución debería ser producir una actualización de la pestaña o un cambio del contenido a mostrado.
- Los filtros por rol no diferencian entre jugadores de diferentes ligas.
  - La solución es mejorar el filtrado para que diferencie entre ligas.

## Opinión personal y cambios que realizaría

( Javier Suárez Arcos | [a14javsuaarc@inspedralbes.cat](mailto:a14javsuaarc@inspedralbes.cat) )

El proyecto me ha servido para observar las posibilidades que se pueden crear con los conocimientos adquiridos, a ser capaz de tomar decisiones entorno a la cantidad de datos que se deben manipular.

A día de hoy, cambiaría varios apartados de la web, intentando dividirla más en diferentes aspectos, entre “Qué puedo hacer si es la primera vez que entro” y “Que hago cuando ya tengo mi equipo formado”.

( Oscar Ortega Sola | [a18oscortsol@inspedralbes.cat](mailto:a18oscortsol@inspedralbes.cat) )

En este proyecto he aprovechado para ver como es el trabajo desde casa, como coordinarte con una persona o varias personas sin estar cara a cara para hablar sobre un punto del proyecto o ponerse de acuerdo en como hacer alguna parte.

Algo que cambiaría de este proyecto es ponerle un apartado de noticias actuales del mundo de los videojuegos. Noticias de los equipos, fichajes, fechas de eventos etc. Para que el usuario interesado no tenga que ir a buscarlo a otro lado y en una misma página tener la diversión y la información.