

# ***NETSCORE***

*PADEL APP*

Colabora:



**Florida**

Universitaria

## **INDICE DE LA MEMORIA**

### **1- DATOS GENERALES DEL PROYECTO (3)**

- Introducción**
- Diagnóstico de la problemática**
- Objetivos Generales**

### **2.- EJECUCIÓN DEL PROYECTO (4-7)**

- Metodología utilizada**
- Actividades realizadas**
- Recursos utilizados**
- Viabilidad del proyecto**

### **3.-VIABILIDAD DEL PROYECTO (8)**

### **4.-REFORMULACIÓN O AJUSTE DEL PROYECTO (8)**

### **5.-LINEAS DE CONTINUIDAD DEL PROYECTO**

### **6.-DOCUMENTACIÓN Y BIBLIOGRAFÍA (8)**

## 1- DATOS GENERALES DEL PROYECTO

### Introducción:

NetScore es una plataforma de muestra de datos en directo que permite consultar a tiempo real los resultados de los marcadores de distintos partidos de pádel.

A su vez, la plataforma permite crear un entorno de puntuación no compartido con el resto de usuarios con la finalidad de llevar un auto-registro de la puntuación y evitar errores por factor humano.

### Diagnóstico de la problemática

NetScore nace con el objetivo de cubrir la necesidad que muchas organizaciones enfocadas al mundo del deporte de raqueta tienen al tratar de gestionar ciertos aspectos en sus eventos.

En la actualidad los sistemas de marcadores para este tipo de eventos son muy escasos y complejos por su tipo de instalación: pantallas led, marcadores digitales de baja resolución (tablero físico con números digitales)...



### Objetivos Generales

Los principales objetivos de NetScore son:

- Facilitar la gestión de torneos y competiciones con bajo presupuesto o incluso partidos de aficionados al deporte de raqueta proponiendo una solución sencilla a la hora de compartir el transcurso de los partidos. Dando la posibilidad de mantener informados a los seguidores desde cualquier lugar con conexión a internet.
- Ofrecer una herramienta a aquellos jugadores que quieren plasmar los datos de sus partidos o entrenamientos en su dispositivo.

## **2.- EJECUCIÓN DEL PROYECTO**

### **Metodología utilizada**

El proyecto se desarrolla siguiendo una estructura de sprints. Cada sprint se enfoca en una funcionalidad ya sea de forma general o específica.

Las primeras jornadas de desarrollo se centran en la organización y planteamiento del proyecto (investigación y recopilación de herramientas necesarias) lo que da lugar a la decisión de abordar el desarrollo desde tres ámbitos técnicos:

- Interfaz
- Funciones internas
- Interacciones con la base de datos

Cada sprint, a pesar de enfocarse en uno de los tres ámbitos mencionados previamente, se subdivide en tareas de ese ámbito. Es así como se implementa el uso de tableros Kanban como apoyo en la organización del proyecto.

### **Actividades realizadas**

#### **Primer Sprint**

En primer lugar, se estudia la información esencial que se requiere para poder seguir un partido de forma clara sin tener una imagen del juego, simplemente con datos planos.

Una vez recopilados los datos esenciales se crea un caso de la aplicación sobre papel, a mano alzada, con la intención de estimar qué distribución de los elementos puede ser la más clara y, a su vez, la que mejor pueda transmitir al usuario los datos esenciales a simple golpe de vista.

Estando ya definida, a nivel visual, la idea general de la aplicación se da forma a un boceto digital con el software AxureRP el cual permite dar algo más de fidelidad a la idea y tener una previsualización algo más clara de lo que será la primera versión de la interfaz de la aplicación.

#### **Segundo Sprint**

En el segundo sprint se ha creado una interfaz poco elaborada, sin navegación entre vistas ni uso de componentes, simplemente vistas separadas con la intención de ver qué componentes pueden aislarse para ser reutilizados en distintas zonas de la aplicación y más adelante obtener un código más limpio.

### Tercer Sprint

Ya teniendo una interfaz 'útil' se empieza con el diseño de funciones internas de la aplicación.

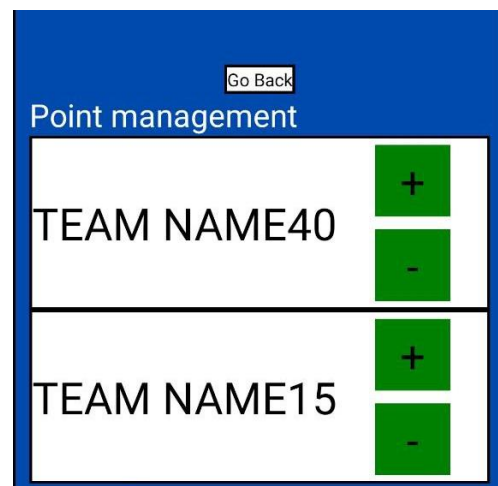
El primer paso es crear un diagrama de flujo con el software de diagramas Dia (<http://dia-installer.de/index.html.es>)

Ya con una estructura de datos más clara se empieza a implementar el uso de navegación entre pantallas para agilizar los procesos de comprobaciones y compartición de datos entre vistas.

### Cuarto Sprint

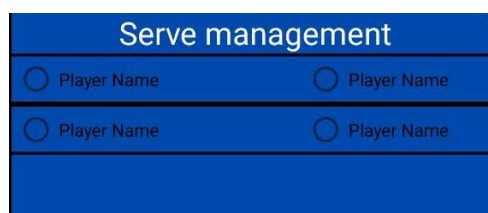
En el cuarto sprint se hace una búsqueda de componentes para cubrir la necesidad de los pulsadores de control de puntuación.

Ninguna de las librerías existentes se adecua a las necesidades de la funcionalidad que se quiere afrontar y por ello se decide crear un componente propio que maneje las puntuaciones y a su vez sea intuitivo para el usuario.



Point management	
TEAM NAME40	<div>+</div> <div>-</div>
TEAM NAME15	<div>+</div> <div>-</div>

Se añade también el componente de jugador para controlar su servicio

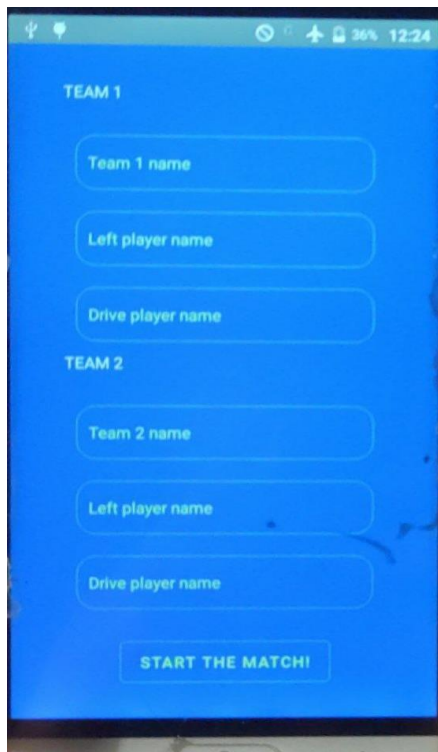


Serve management	
<input type="radio"/> Player Name	<input type="radio"/> Player Name
<input type="radio"/> Player Name	<input type="radio"/> Player Name

## Quinto Sprint

Se crean los primeros formularios para poder empezar a testear las pantallas ya creadas con datos 'reales'.

Los primeros datos de prueba se sustituyen por datos introducidos por pantalla, por lo que se crean las primeras comunicaciones entre componentes



Una vez dado el visto bueno a los componentes implicados en el testing se decide simplificar la introducción de datos por parte del usuario creando una función que genera automáticamente el nombre de los equipos en relación a los jugadores del mismo:

Las tres primeras letras del nombre en mayúscula de cada pareja separadas por una barra

- pepe y marcos = PEP/MAR
- Antonio y SAMUEL = ANT/SAM

Estando ya implementada esta función de automatización se eliminan los campos de nombre de equipo en el formulario.

## Sexto Sprint

Se comienza la configuración de la aplicación con la base de datos. Se crea una API en php en un servidor plesk que conecta una base de datos SQL con la aplicación mediante peticiones con AXIOS en react-native que atacan directamente a los ficheros php almacenados en el servidor.

Durante el proceso de despliegue en la base de datos SQL queda claro que la escalabilidad del proyecto se ve muy limitada por el tipo de base de datos que se está empleando y se llega a la conclusión de que la mejor decisión es buscar un servicio que permita mayor flexibilidad en la gestión de datos e incluso la ampliación del servicio en caso de ser necesario.

La gestión de datos del proyecto se traslada a firebase, el servicio 'auto-hosteado' de Google, que ofrece un servicio de bases de datos no relacional llamado firestore.

Una de las características más llamativas de firestore es la gestión de los datos a tiempo real, lo que permite una mayor sincronía con la aplicación.



Cabe destacar que el servicio es gratuito aunque en caso de necesitar una configuración con mayores prestaciones es posible adquirir una suscripción que escala estas prestaciones sin necesidad de alterar el código del proyecto.

### Séptimo sprint

Expo limita la interacción con algunos servicios de firebase por lo tanto se inicia una nueva aplicación, esta vez desde react native-cli, lo cual permite implementar código específico de firebase para dispositivos móviles y no requerir una configuración como aplicación web del servicio de Google.

Se adaptan ciertos componentes y se reinstalan las librerías empleadas en la aplicación de expo.

### Octavo Sprint

Teniendo ya en funcionamiento un producto mínimo viable se dedica una etapa de pulido a la interfaz para darle una estética más actual y uniforme.

ICA/BRE	
Icardo	Brea
ICA/BRE	0 0 0 0
TRI/SAL	0 0 0 0
Triay	Salazar
TRI/SAL	

### Recursos utilizados

AxureRP: Creación de wireframe y mockup  
Dia: Diseño de diagramas de flujo de las funciones de la aplicación  
Visual Studio Code  
Github  
Firebase

### **3.-VIABILIDAD DEL PROYECTO**

Como se puede apreciar en el proceso del proyecto hasta la fecha, la escalabilidad es uno de los puntos clave en los que se ha centrado el curso del desarrollo. Esto es porque a pesar ser una aplicación muy sencilla puede llegar a generar un gran volumen de datos, lo cual resultaría un inconveniente a la hora de crecer entre los usuarios.

Teniendo claro el factor de crecimiento de la aplicación se estima una viabilidad técnica muy positiva.

### **4.-REFORMULACIÓN O AJUSTE DEL PROYECTO**

Tras varias etapas de pruebas en la aplicación se plantea la posibilidad de realizar pequeños cambios funcionales como implementar un buscador en la pantalla de inicio o también la posibilidad de agrandar el tamaño de las fuentes para aprovechar mejor el espacio de las pantallas.

### **5.-LINEAS DE CONTINUIDAD DEL PROYECTO**

El siguiente paso en la evolución del proyecto es la implementación de un módulo de reconocimiento de voz para monitorear los resultados sin necesidad de manipular los dispositivos en los que se utiliza la aplicación.

### **6.-DOCUMENTACIÓN Y BIBLIOGRAFÍA**

<https://www.reactnative.dev>

<https://firebase.google.com/docs/firestore>

<https://www.github.com>

<https://javascript.tutorialink.com/>

<https://stackoverflow.com/>

<https://iqcode.com/>

<https://www.codegrepper.com/>

<https://www.w3schools.com/>

<https://medium.com/>