IES Nervión

Football

Training Manager

1. **Estudio de viabilidad del Sistema (EVS)**
   1. Descripción general del sistema

Se pretende realizar una aplicación que ayude a un entrenador de fútbol a gestionar el entrenamiento y la táctica principal de un equipo. El usuario que quiera usar la aplicación deberá registrarse previamente. Una vez registrado, podrá acceder a su panel de gestión del entrenamiento. Tendrá la opción de gestionar la formación del equipo, el entrenamiento colectivo por semana, así como añadir las notas que quiera, relativas a los partidos, las tácticas, etc. Además, también dispondrá de una “pizarra” donde podrá pintar movimientos para enseñar o corregir acciones tácticas a sus jugadores.

La pantalla principal mostrará un formulario con los datos de perfil del usuario, foto, nombre, correo, etc. Estos datos se podrán editar en cualquier momento.

A la derecha se podrá desplegar un menú *hamburger* con las distintas secciones de la aplicación.

En la sección de ‘*formación*’ podrá gestionar el sistema táctico, los roles asignados por posición en dicha formación, la mentalidad principal (ofensiva, contraataque, etc.). El usuario también podrá definir las instrucciones que quiera, en el apartado de instrucciones dentro de formación.

En ‘*entrenamiento de equipo’* decidirá el programa de entrenamiento semanal a desarrollar, especificando la característica principal en la que se enfocará cada semana y el nivel de carga de trabajo.

En el apartado ‘*notas*’, el entrenador escribirá cualquier tipo de nota que desee, sobre los entrenamientos, los partidos, etc.

Y, por último, en la sección de ‘*pizarra*’, podrá dibujar los movimientos que considere, así como guardar como imagen el movimiento dibujado.

1.1.2 Diagrama de contexto del sistema



1.2 Descripción de los sistemas de información actuales

Actualmente existen sistemas desarrollados, pero su uso no está muy extendido.

Los sistemas actuales están orientados a una parte más concreta, es decir, hay aplicaciones por ejemplo para usar una pizarra táctica, otras para la gestión de estadísticas, etc. Pero no es común que abarque varias secciones a la vez.

Esta aplicación abarca gestión táctica, la planificación de entrenamiento, una pizarra táctica y también un apartado personal de notas.

Es decir, abarca varias secciones, pero de un modo más general y menos al detalle. Por ejemplo, una aplicación que se centre exclusivamente en una pizarra táctica será más especifica o completa que la sección dedicada en Football Manager Training.

Por lo tanto, buscamos una gestión más general pero que abarque varias secciones distintas en el entrenamiento y táctica de un equipo de fútbol. Para así poder gestionar de una manera sencilla aspectos generales del equipo.

1.3 Catálogo de requisitos previo

1.3.1 Requisitos funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| Requisitos funcionales | | | |
| **Código** | **Descripción** | **Prioridad** | **Fecha** |
| RQF1 | Debe registrar usuarios y validar el acceso de estos | 1 | 02/04/2019 |
| RQF2 | Debe permitir recuperar la contraseña a partir del correo del usuario previamente registrado | 2 | 02/04/2019 |
| RQF3 | Debe permitir editar los datos del perfil del usuario | 2 | 02/04/2019 |
| RQF4 | Debe permitir cambiar la contraseña | 3 | 02/04/2019 |
| RQF5 | Debe permitir modificar la imagen del perfil del usuario. | 3 | 02/04/2019 |
| RQF6 | Debe permitir navegar entre las distintas secciones a través del menú | 1 | 02/04/2019 |
| RQF7 | Debe permitir crear y editar una formación táctica personalizada | 1 | 02/04/2019 |
| RQF8 | Debe permitir crear y eliminar instrucciones tácticas en la sección de formación | 2 | 02/04/2019 |
| RQF9 | Debe permitir dibujar sobre la pizarra táctica | 2 | 02/04/2019 |
| RQF10 | Debe permitir guardar como imagen cada pizarra táctica editada | 3 | 02/04/2019 |
| RQF11 | Debe permitir editar y guardar la configuración del entrenamiento | 1 | 02/04/2019 |
| RQF12 | Debe permitir crear, leer, modificar y eliminar notas. | 2 | 02/04/2019 |
| RQF13 | Debe permitir desloguearse y volver a la pantalla inicial en cualquier sección | 2 | 02/04/2019 |

1.3.2 Requisitos de datos

|  |  |  |  |
| --- | --- | --- | --- |
| Requisitos de Datos | | | |
| **Código** | **Descripción** | **Prioridad** | **Fecha** |
| RQD1 | Deben guardarse los datos de perfil de cada usuario | 1 | 02/04/2019 |
| RQD2 | Deben guardarse los datos de la formación táctica de cada usuario | 1 | 02/04/2019 |
| RQD3 | Debe guardarse la configuración del entrenamiento de cada usuario | 1 | 02/04/2019 |
| RQD4 | Deben guardarse las notas creadas de cada usuario | 2 | 02/04/2019 |

1.3.3 Requisitos de Interfaz

|  |  |  |  |
| --- | --- | --- | --- |
| Requisitos de Interfaz | | | |
| **Código** | **Descripción** | **Prioridad** | **Fecha** |
| RQI1 | Debe mostrar los datos de perfil del usuario | 1 | 02/04/2019 |
| RQI2 | Debe mostrar la imagen que represente el sistema táctico escogido | 2 | 02/04/2019 |
| RQI3 | Debe mostrar todos los datos de la formación, así como un botón de guardado en caso de edición | 1 | 02/04/2019 |
| RQI4 | Debe mostrar una pizarra táctica sobre la que dibujar y la opción de guardar como imagen | 1 | 02/04/2019 |
| RQI5 | Debe mostrar las notas creadas, así como también botones para editarlas, eliminarlas o crear más. | 2 | 02/04/2019 |
| RQI6 | Debe mostrar la planificación del entrenamiento editada por el usuario y un botón para guardar cambios. | 1 | 02/04/2019 |

1.3.3 Requisitos no funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| Requisitos de no funcionales | | | |
| **Código** | **Descripción** | **Prioridad** | **Fecha** |
| RQNF1 | Debe funcionar en cualquier ordenador con Windows 10, independientemente de sus recursos hardware | 2 | 02/04/2019 |

1.4 Alternativas de solución

1.4.1 Alternativa I

La primera alternativa consiste en desarrollar una UWP. Tendría la opción de desplegarse para ordenador, tablet y móvil. Siempre y cuando use el sistema operativo de Windows.

1.4.1.1 Hardware

Se necesita un servidor para almacenar los datos. Pudiendo perfectamente utilizar los recursos que nos ofrece Azure, por ejemplo.

No se requieren de requisitos hardware en el usuario que use esta aplicación.

1.4.1.1 Requisitos

La aplicación solo puede ser desplegada en dispositivos con Windows.

1.4.1.1 Tecnologías

Motor de BBDD de SQL Server, programación en C# de UWP, programación de API en C# para mejorar y facilitar el acceso a los datos. Front en XAML.

1.4.1.1 Estudio de riesgo

|  |  |  |
| --- | --- | --- |
| Riesgo | Porcentaje | Importancia |
| **Falta de Experiencia de los desarrolladores** | 10% | 1 |
| **Incumplimiento de plazos** | 15% | 1 |
| **Poca estabilidad** | 15% | 3 |
| **Problemas para soportar determinado número de usuarios** | 15% | 3 |
| **Poco aprovechamiento de la tecnología** | 10% | 2 |
| **Incompatible con distintos SO** | 80% | 2 |
| **Compatibilidad entre dispositivos** | 10% | 1 |

1.4.2 Alternativa II

La segunda alternativa consiste en desarrollar una aplicación web.

1.4.1.1 Hardware

Se necesita de un servidor donde desplegar además de nuestros datos, nuestra aplicación web para poder acceder a ella.

El usuario no necesita de recursos hardware potentes de ningún tipo.

1.4.1.1 Requisitos

La aplicación debe ser desplegada en un servidor web.

1.4.1.1 Tecnologías

Motor de BBDD de SQL Server, programación en JavaScript, junto con HTML y CSS.

1.4.1.1 Estudio de riesgo

|  |  |  |
| --- | --- | --- |
| Riesgo | Porcentaje | Importancia |
| **Falta de Experiencia de los desarrolladores** | 15% | 1 |
| **Incumplimiento de plazos** | 35% | 1 |
| **Poca estabilidad** | 15% | 3 |
| **Problemas para soportar determinado número de usuarios** | 15% | 3 |
| **Poco aprovechamiento de la tecnología** | 10% | 2 |
| **Incompatible con distintos SO** | 10% | 2 |
| **Compatibilidad entre dispositivos** | 15% | 1 |

1.4.3 Alternativa III

La tercera alternativa consiste en desarrollar una aplicación Android.

1.4.1.1 Hardware

El usuario no necesita de recursos hardware potentes en su dispositivo móvil.

1.4.1.1 Requisitos

La aplicación debe ser desplegada en dispositivos Android.

1.4.1.1 Tecnologías

Motor de BBDD de SQL Server, programación en Java de la aplicación, Front en XML.

1.4.1.1 Estudio de riesgo

|  |  |  |
| --- | --- | --- |
| Riesgo | Porcentaje | Importancia |
| **Falta de Experiencia de los desarrolladores** | 10% | 1 |
| **Incumplimiento de plazos** | 35% | 1 |
| **Poca estabilidad** | 10% | 3 |
| **Problemas para soportar determinado número de usuarios** | 10% | 3 |
| **Poco aprovechamiento de la tecnología** | 15% | 2 |
| **Incompatible con distintos SO** | 80% | 2 |
| **Compatibilidad entre dispositivos** | 30% | 1 |

1.5 Justificación de alternativa seleccionada

Escogemos la primera alternativa por varios motivos, los desarrolladores tienen amplia experiencia en desarrollo de UWP, no se requieren de requisitos hardware especiales, y además tiene la opción de desplegarse en dispositivos de cualquier tipo siempre y cuando usen el sistema operativo Windows en una versión actual.

Ofrece una buena estabilidad, y dado el corto plazo disponible, el conocimiento de la tecnología a desarrollar ofrece una mayor posibilidad a cumplir los plazos establecidos.

En contra de la segunda alternativa que implicaría más tiempo dada la experiencia menor en comparación, además de la necesidad de desplegar la aplicación en un servidor web.

La tercera alternativa dada la naturaleza de la aplicación se antoja poco compatible para mostrar la interfaz de la manera concreta y limita su uso a dispositivos móviles y tablets. También requiere de algo más de tiempo el desarrollo en Android.

**2. Gestión del proyecto**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre Tarea | Inicio | Fin | Duración (días) |
| EVS | 01/04/2019 | 03/04/2019 | 3 |
| GPI | 03/04/2019 | - | - |
| ASI | 03/04/2019 | - | - |
|  |  |  |  |

**3. Análisis de sistemas de información**

3.1 Descripción general del entorno tecnológico del sistema

Para el desarrollo del sistema se va a crear una UWP, implementada en C# y XAML para las vistas. Estará dividida en capas, DAL (acceso a datos), Entidades y UI (interfaz de usuario). Se aplicará el patrón MVVM (model-view-viewmodel), separando así la lógica de la interfaz.

El motor de la base de datos será SQL Server. La API de acceso a estos datos será también desarrollada en C#.

Para el desarrollo de la UWP y la API se utilizará Visual Studio, para la base de datos Microsoft SQL Server Management Studio.

3.2 Catálogo de usuarios

|  |  |  |
| --- | --- | --- |
| Código | Nombre | Descripción |
| **Usuario registrado** | Manager | Gestiona todas las funcionalidades a su disposición en la aplicación |
| **Usuario no registrado** | Usuario | Puede registrarse, para pasar a ser Manager |

3.3 Casos de uso

