**IES NERVION, Adventure’s Map**

**Introducción**

Aplicación que permite a los usuarios colgar ubicaciones de interés referentes al senderismo, a la acampada controlada o al vivaqueo. Cada ubicación colgada tendrá un tipo (ZAC, vivac, camping, Paraje natural, Hostales…), una información acreditada por el usuario y una puntuación inicial. Los usuarios podrán puntuar las localizaciones de otros usuarios. Cuando a un usuario le interese una ubicación podrá seleccionarla y marcar una ruta (a través de Google maps). La aplicación también tendrá una zona para descargar o abrir mapas, útil en el caso de no tener cobertura. Al abrir un mapa también se generará una brújula en la parte superior derecha de la pantalla.

**Objetivo**

El objetivo de la aplicación se basa en que los usuarios puedan planificar sus próximas escapadas, tomando información de las diferentes ubicaciones mostradas en el mapa, creando sus propias rutas de viaje… y la posibilidad de que los usuarios se orienten gracias al mapa online (u offline en caso de falta de datos o cobertura), filtrando por zonas de senderismo, pernocta, zonas de abastecimiento o restaurantes.

**Justificación del proyecto**

En el mercado hay una gran cantidad de aplicaciones basadas al senderismo, pero no tantas orientadas al camping o al vivaqueo. La aplicación toma como punto fuerte la posibilidad de obtener la localización de las zonas ZAC o vivaqueo. Además de disponer de filtros útiles según las necesidades del usuario (Zonas de abastecimiento, pernocta…).

**EVS1: Establecimiento del alcance del sistema**

Esta aplicación va dirigida a habituales de la acampada y el vivaqueo, donde también se pueden realizar rutas de senderismo o viaje. En la actualidad existen aplicaciones como ViewRanger, Guru Maps… Que permiten obtener ubicaciones de interés relacionadas con el senderismo y con la posibilidad de crear rutas.

**EVS2: Estudio de la situación actual**

**EVS 2.3: descripción de los sistemas de información existentes**

**Wikiloc**

* Permite una mejor búsqueda con los filtros por kilometraje, desnivel acumulado o acotación de zona de interés en el mapa. Además, los usuarios Premium pueden buscar rutas de personas a las que siguen y filtrarlas por fecha de grabación, con la posibilidad de encontrar rutas en temporadas del año específicas.
* Posee un indicador de ruta, brújula y alertas sonoras por si el usuario se aleja de la ruta marcada que funcionan offline.
* Permite grabar la ruta offline y obtener estadísticas en tiempo real como velocidad, distancia recorrida y gráfica de elevación. Se pueden grabar fotos y puntos de interés, que para ser compartidos necesitan conexión a Internet.
* Permite compartir la localización del usuario en la ruta a otras personas mediante su seguimiento en vivo (requiere Wikiloc Premium y conexión a Internet).
* Cuenta con una gran cantidad de mapas offline gratuitos.
* Da la opción de compartir las rutas con los usuarios que le acompañasen en la ruta y en redes sociales como Facebook o Twitter.
* Puede conectarse tanto a un Apple Watch como a un Garmin para descargar las rutas en estos dispositivos.
* Disponible en 9 idiomas.

**Viewranger**

* Cuenta con Skyline, una tecnología de realidad aumentada que permite identificar elementos clave en los alrededores de las rutas.
* Permite descargar mapas topográficos del CNIG (Centro Nacional de Información Geográfica), de carreteras y caminos de OpenStreetMap y OpenCycleMap, de pistas de esquí y cuenta con algunos mapas adicionales de gran calidad para Europa, EE.UU., Canadá y Nueva Zelanda.
* Da la posibilidad tanto de crear rutas como de descargar rutas de senderismo y ciclismo creadas por otros usuarios, agencias de turismo, editoriales de guía, etc.
* Al guardar los mapas en el teléfono, no necesita conexión a Internet ni cobertura.
* Se puede grabar la ruta para compartirla con amigos, familiares y contactos, añadiendo fotos a ésta desde Instagram o Flickr y permite mirar estadísticas del trayecto, como velocidad, distancia, altitud, etc.
* Con BuddyBeacon es posible compartir la ubicación actual con amigos y familia.
* Compatible con Android Wear.

**Guru Maps**

* Mapas offline de todos los países del mundo detallados y que ocupan poco espacio, con datos abiertos basados en el proyecto OpenStreetMap.
* Búsqueda offline rápida, que funciona por nombre, categoría o coordenadas.
* Sincronización con otros dispositivos, entre plataformas Android e iOs y copia de seguridad en servidor.
* Registra los viajes incluso en segundo plano y permite obtener estadísticas en tiempo real como la velocidad, distancia, altitud, etc. Permite compartir las rutas vía email, Facebook o Twitter y exportar en formato KML y GPX.
* Permite guardar localizaciones con distintos iconos.
* Disponible en 11 idiomas.

**EVS3: Definición de requisitos del sistema**

**EVS 3.2: Identificación de requisitos**

-La aplicación debe consumir los menos recursos posibles.

-Debe ocupar lo mínimo posible en memoria.

**EVS 3.3: Catalogación de requisitos**

Tipos

**Rendimiento**

-La aplicación debe consumir los menos recursos posibles.

**Ocupación en memoria**

-Debe ocupar lo mínimo posible en memoria.

**Estudio de alternativas de solución**

Podríamos desarrollar la aplicación para las diferentes plataformas:

* Teléfonos móviles.
* SmartWath’s.

Con los siguientes sistemas de desarrollo:

* Android.
* IOS.
* .Net (FrameWork interesante, pero WindowsPhone fue una causa perdida en el mercado.)

La aplicación solo será válida para móviles o plataformas SmartWatch. Debido al concepto determinado de la aplicación.

La aplicación solo contendrá lo siguiente:

* Almacenamiento de mapas.
* Creación y almacenamiento de rutas, permitiendo que estas se compartan.
* Un mapa con un buscador y un menú de filtrado de localizaciones de interés (Parques naturales, ZAC…).
* Los usuarios podrán validar las localizaciones de otros usuarios.

La aplicación no contendrá lo siguiente:

* Una red social dentro de la misma aplicación, donde se puedan compartir las rutas, chatear o ver el estado de las últimas escapadas de usuarios agregados.
* Chats privados entre diferentes usuarios.

**Valoración de las alternativas (Falta ampliar explicaciones de costes y presupuestos) Aquí también se explica la selección de la solución**

La aplicación será realizada para dispositivos Android. Es la solución más versátil y con más salidas de mercado, debido a la gran popularidad de este sistema. El hecho de que trabajemos con un sistema de software libre nos permite reducir costes.

La idea de adaptar la aplicación para las plataformas SmartWatch queda eliminada, ya que supondría un aumento en el tiempo de desarrollo, además de evitarnos posibles licencias de pago como sucede con Apple.

La aplicación será desarrollada para móviles con sistema Android. También será escrita en código Kotlin.

**Estudio de los riesgos de la aplicación a realizar**

Cada riesgo tendrá un indicador de importancia, cuanto más grave sea dicho riesgo menor será si indicador:

1 🡪 Catastrófico (Puede poner en peligro la realización del proyecto)

2 🡪 Crítica (El rendimiento del sistema se degrada)

3 🡪 Marginal (La aplicación no cumple plenamente un objetivo)

4 🡪 Despreciable (Sufrimos inconvenientes)

|  |  |
| --- | --- |
| Riesgos | Gravedad |
| Adaptación para el nuevo lenguaje (Kotlin) | 1 |
| Complejidad a la hora de controlar el mal uso de la aplicación (Puede haber usuarios que se dediquen constantemente a insertar información errónea y debemos controlarlo) | 2 |
| Inexperiencia con la tecnología de mapeo | 2 |
| Superar el espacio de memoria prestablecido que debe ocupar la aplicación | 3 |
| Interfaz no muy amigable | 4 |

**Estrategia de desarrollo**

Se optará por un ciclo de vida en espiral, realizando una planificación previa y general sobre toda la aplicación y luego tomando una serie de puntos de referencia que se deberán ir cumpliendo según unas fechas establecidas. El objetivo será ir completando el proyecto por fases, determinando los objetivos de cada fase, tomando un análisis más específico sobre los riesgos que pueden ocurrir en dicha fase para luego comenzar su desarrollo y prueba de errores.

**Diagrama de hitos**

|  |  |
| --- | --- |
| Actividad | Finalización |
| Creación del esqueleto completo de la aplicación (Layouts) | Dos días |
| Almacenador y visualizador de mapas sin conexión. (Aquí podrás descargar mapas y visualizarlos) | Medio mes |
| Creador de rutas | Medio mes |
| Creación de la base de datos con todas sus funciones y procedimientos | 2 semanas |
| Creador de punto de localización de interés, junto con la visualización de sus datos. (Cuando cliques en uno aparecerá un fragmento con sus datos, podrás pulsar en un botón del fragmento para cargar una pantalla al completo con sus datos) | Una semana |
| Listado de Rutas | 1 día |
| Listado de puntos de localización propios | 1 día |
| Listado de puntos de localización favoritos | 1 día |
| Mapa principal según la posición actual o por defecto si no tienes conexión (En este solo aparecerán los puntos de localización ya almacenados) | 2 semanas |
| Traspaso de datos de la base de datos online a la local de nuestra aplicación | 2 días |
| Creación de un login screen | 3 días |

**ASI 1: Definición del Sistema**

**ASI 1.1: Determinación del alcance del sistema**

El sistema debe ser capaz de:

-Crear, almacenar y visualizar rutas de senderismo.

-Crear, almacenar y visualizar puntos de interés, con la opción extra de compartirlos a los demás usuarios de la aplicación.

-Descargar, almacenar y visualizar mapas sin conexión.

-Realizar filtrados específicos en el mapa principal.

El sistema se comunicará con lo siguiente:

-La API de Google Maps.

-Una base de datos alojada en un servidor Web.

**ASI 1.4: Identificación de los usuarios participantes y finales**

Existirán usuarios participantes anónimos que probarán la aplicación durante una semana en diferentes localizaciones.

Los usuarios finales serán personas familiarizadas con el senderismo, la acampada, el vivaqueo o que deseen preparar un viaje creando las rutas a seguir del mismo, generando además puntos de localización que vean interesantes.

**ASI 2: Establecimiento de Requisitos**

**ASI 2.1 Obtención de requisitos**

Se adjunta a este documento una serie de formularios rellenados por futuros clientes de la aplicación.

**ASI 2.2 Especificación de Casos de Uso**

*Interfaces de la aplicación (Falta la pantalla de descarga de mapas)*

Casos De Uso

|  |  |
| --- | --- |
| Nombre | Loguearse en la aplicación |
| Descripción | El usuario desea loguearse en la aplicación con su cuenta. |
| Precondición | El usuario debe tener una cuenta en la aplicación. |
| Secuencia principal | |  |  | | --- | --- | | 1 | El usuario inserta su nombre de usuario y contraseña. | | 2 | Si el nombre de usuario y contraseña son correctos el sistema carga la primera pantalla de inicio de la aplicación. | |
| Errores / Alternativas | Si el usuario o contraseña es incorrecto, el sistema mostrará un mensaje de error por pantalla. |
| Postcondiciones | Usuario logueado. |
| Notas | No |

|  |  |
| --- | --- |
| Nombre | Crear una cuenta en la aplicación |
| Descripción | El usuario desea crearse una cuenta en la aplicación. |
| Precondición | Ninguna |
| Secuencia principal | |  |  | | --- | --- | | 1 | El usuario hace click en el botón de “Crear cuenta”. | | 2 | El sistema carga un formulario para el usuario. | | 3 | El usuario rellena todos los datos necesarios del formulario. | | 4 | Si todos los datos son correctos, el sistema almacena los datos, carga la página de login de la aplicación e informa al usuario que su cuenta a sido creada con éxito. | |
| Errores / Alternativas | Si los datos del formulario no son correctos, el sistema le informa de ello al usuario. |
| Postcondiciones | Cuenta de usuario creada. |
| Notas | No |

|  |  |
| --- | --- |
| Nombre | Recuperar contraseña |
| Descripción | El usuario desea recuperar su contraseña. |
| Precondición | Ninguna. |
| Secuencia principal | |  |  | | --- | --- | | 1 | El usuario hace click en “¿Olvido su contraseña?”. | | 2 | El sistema carga una nueva pantalla con un cuadro de texto para el correo electrónico. | | 3 | El usuario escribe su correo en el cuadro de texto y le da al botón enviar. | | 4 | Si el correo es válido y la aplicación tiene conexión a internet, el sistema carga una nueva pantalla con un cuadro de texto, envía un código de clave a ese correo e informa de ello al usuario. | | 5 | Si el usuario es el propietario de ese correo, obtendrá la clave, la insertará en el campo de texto y hará click en el campo de envío. | | 6 | Si la clave es válida y la aplicación tiene conexión a internet, el sistema carga una nueva pantalla con dos cuadros de texto para la nueva contraseña y un botón de envío. | | 7 | El usuario inserta dos veces la nueva contraseña y hace click en el botón de envío. | | 8 | Si las dos contraseñas son iguales y cumplen la seguridad mínima requerida por el sistema, el sistema carga la pantalla de login e informa al usuario que el cambio de contraseña fue un éxito. | |
| Errores / Alternativas | 3 🡪 Si el correo tiene un formato incorrecto o sino se encuentra almacenado en la aplicación, muestra un mensaje de error por pantalla.  3 🡪 Si la aplicación no tiene conexión a internet en ese momento, el sistema informa de ello al usuario impidiendo el cambio de contraseña.  6 🡪 Si la clave no es válida le informa de ello al usuario.  6 🡪 Si la aplicación no tiene conexión a internet en ese momento, el sistema informa de ello al usuario impidiendo la inserción de la clave.  8 🡪 Si las contraseñas no son iguales o sino cumplen los requisitos mínimos de seguridad, el sistema informa de ello al usuario.  8 🡪 Si la aplicación no tiene conexión a internet en ese momento, el sistema informa de ello al usuario impidiendo el cambio de contraseña. |
| Postcondiciones | El usuario consigue cambiar su contraseña. |
| Notas | En el paso número 5, si la aplicación se encontraba sin conexión el usuario puede volver a pedir al sistema que le envíe otra clave. |

|  |  |
| --- | --- |
| Nombre | Moverse entre las pantallas de la aplicación |
| Descripción | El usuario desea moverse a una de las pantallas adyacentes de la aplicación. |
| Precondición | Ninguna |
| Secuencia principal | |  |  | | --- | --- | | 1 | El usuario arrastra el dedo por la pantalla, en dirección a la pantalla que desea moverse. | | 2 | Si existe una pantalla en esa dirección, el sistema carga esa pantalla. | |
| Errores / Alternativas | Si no existe una pantalla en esa dirección, se mantiene la pantalla actual. |
| Postcondiciones | El usuario se mueve entre las pantallas principales de la aplicación. |
| Notas | Sobre las pantallas principales de la aplicación existe un recuadro con los cuatro nombres de cada pantalla, se encontrará iluminado el nombre de la pantalla actual en la que se encuentre el usuario. |