JUSTIFICACIÓN

INTRODUCCIÓN

A lo largo de nuestra historia, **los humanos hemos vivido en sociedad**. **Somos seres sociables**, y, por lo tanto, **dichas interacciones representan una parte importante de nuestra vida**, tanto en la parte privada como en la parte pública.

Estas mismas, a pesar de su aparente sencillez, **pueden llegar a ser extremadamente complicadas para algunas personas,** sobre todo si eres nuevo en una ciudad o trabajo haciendo que conocer gente nueva puede ser una tarea muy complicada.

**La finalidad de este proyecto es precisamente facilitar esta tarea a todo tipo de personas que lo necesiten**, por medio del desarrollo de una red social que permita a los usuarios registrados en esta tanto **crear eventos sociales de cualquier tipo** y tópico como poder unirse a eventos de otros usuarios.

**Estos eventos los verá el usuario dependiendo de su ubicación** y de un radio en km delimitado por el mismo. Así mismo podrá buscar eventos en ubicaciones especificas así como eventos con nombres especificos.

Con ello, **conseguimos crear una herramienta que facilita a las personas el conocer a gente que comparten sus gusto y aficiones**, además de permitir realizar actividades en compañía. Así, **cualquier persona que tenga dificultades para socializar o encontrar nuevos amigos**, puede hacerlo, asegurandose por el camino que tienen gustos y aficiones en común, consiguiendo así evitar los típicos sentimientos de soledad acompañados a estos.

**Para la elaboración de dicho proyecto, usaremos distintas tecnologías**. Primero de todo, la plataforma para la que estará disponible serán los teléfonos Android. Usaremos **como lenguaje de programación Java**, acompañado de ciertas tareas que serán realizadas **usando PHP**. Contaremos con una base de datos, siendo esta el **servicio MySQL que proporciona el paquete de aplicaciones XAMPP**. Dicha base de datos se conectará al WorkBench de MySQL para poder permitir las conexiones remotas a la base de datos usando el teléfono móvil.

**El IDE utilizado será Android Studio**, pues tiene las funcionalidades necesarias para poder realizar aplicaciones móviles.

OBJETIVOS

El desarrollo de este TFG tiene varias finalidades:

1. **Ayudar a personas con dificultades a la hora de conocer gente**, a que puedan realizar dicha tarea en un entorno seguro y cómodo para ellos.
2. **Comprender y estudiar el corrector funcionamiento de distintas tecnologías** como PHP o XAMPP para la realización de tareas de desarrollo.
3. **Analizar el funcionamineto interno de las redes sociales** y de la relación que guardan con las bases de datos.
4. **Crear una herramienta con una utilidad real y un funcionamiento útil** que pueda ayudar a las personas en su día a día.

Al final de este documento, y del desarrollo de la aplicación aquí presentada, esperamos el haber cumplido dichos objetivos de forma efectiva, aportando soluciones útiles a los problemas planteados anteriormete.

Nuestro mayor deseo es poder aportar un cambio real a un problema al que muchas personas se tiene que enfrentar sin saber como lidiar exactamente con ellos.

DESARROLLO DEL PROYECTO

TECNOLOGIA USADA

Antes de explicar este apartado, **hemos de entender cuales son las funcionalidades** que contendrá nuestra aplicación:

1. **Creación de cuentas** dentro del servicio ofrecido, enviando un correo de confirmación de cuenta.
2. **Creación de eventos sociales** en distintas ciudades, con distintas plazas y, así como nombre y descripción.
3. **Sistema de busqueda de eventos según la ubicación del usuario**. Posibildiad de bscar eventos según ubicación, usuario creador del evento así como nombre del evento.
4. **Links asociados a cada evento** a los que te puedes apuntar por medio de entrar a estos mismos.
5. **Creación de QR especficos para caada evento creado,** a los que te puedes unir por medio de escanear dichos QR

Así pues, teniendo en cuenta esta información, podemos entender y justificar el por que de los distintos elementos que utilizaremos para el desarrollo de este trabajo.

Lo primero de todo y lo más importante es conocer cuales son los entornos de desarrollo que utilizaremos. En este caso, contaremos con 2 IDE distintos:

* Android Studio:

Es el IDE principal y donde estará desarrollada la mayoría de los servicios de nuestra aplicación. La utilización de este IDE se debe a la comodidad que nos proporcioda y a la familiriadad que tenemos con este.

* Visual Studio:

Se usa principalmente para poder desarrolar las 3 páginas web usando PHP que necesitamos para poder enviar un correo de confirmación así como asociar una página web especifica a cada evento. Estas 3 páginas desarrolladas son:

* + Evento:

Es la página web que tiene asociada un evento para poder unirte a este automaticamente al acceder a dicha página.

* + Inicio de sesión:

Permite al usuario logearse para poder confirmar asistencia a los eventos de la página anterior.

* + Confirmación de cuenta:

Cuando el usuario accede a esta, la cuenta creada por el usuario es confirmada automaticamente y por tanto el usuario es capaz de acceder a ella.

Otro aspecto importante es considerar cuales son los lenguajes de programación que vamos a utilizar para el desarrollo. En este aspecto, contamos con 2 lenguajes de programación distintos:

* Java:

Es el lenguaje de programación principal y en el que estará desarrollada la mayoría de la app. Su uso se debe a las facilidades que nos proporciona y que es uno de los lenguajes principales utilizados para este tipo de desarrollos. Así mismo, la gran comunidad de usuarios distintos así como a un compilado más rápido y el poder crear aplicaciones modulares, la hace una opción más interesante que la alternativa de usar Kotlin. A tener en cuenta que Android Studio nos permite transformar el código de Java a Kotlin en caso de ser necesario.

* PHP:

Lo utilizaremos como un lenguaje de apoyo para poder desarrollar las páginas web que necesitaremos descritos anteriormente. Su uso frente a otros lenguajes backend como podria ser Node.js se resume a dos hechos principales: lo primero de todo que es el lenguaje que hay que usar para poder usar el servicio MySQL del paquete de XAMPP, y segundo, debido a la sencillez de la sintaxis utilizada para realizar la conexión a la base de datos. Al ser realmente 3 páginas web que no son el elemento principa de la aplicación y cuya unica tarea es realizar conexiones a la base de datos, nos podemos permitir no utilizar otros lenguajes con mejores o más funcionalidades.

En la aplicación utilizaremos distintas librerias para realizar distintas tareas. Estos JAR serán:

1. Java Mail:

Lo usaremos para poder enviar mensajes desde un correo especificado por medio de nuestra aplicación Java. Tenemos que crear una contraseña en nuestra cuenta de google para poder permitir que aplicaciones de terceros puedan acceder al correo para realizar tareas (en este caso, enviar mails con un link de confirmación de cuenta)

1. Zxing:

Es la librería encargada de poder crear codigos QR. Lo usaremos para poder guardar información en una imagen que estará asociada a cada evento creado. En este caso, guardara un link que servira para poder confirmar un evento a través de este.

1. JDBC:

Necesaria para poder realizar la conexión a la base de datos desde el teléfono móvil. Así, podemos realizar distintas consultas en la base de datos que estarán relacionadas con distintas tareas que se pueden hacer en la base de datos.

Así mismo, tenemos que tener en cuenta otras consideraciones:

1. XAMPP:

Para realizar ciertas tareas de la aplicación, necesitamos utilizar páginas webs y urls, como es el caso del envió del correo de confirmación, el cual envia al email del usuario una página web en PHP para poder confirmar las creación de la cuenta. Es debido a esto que la base de datos que utilizamos esta desarrollada con phpMyAdmin, pues así podemos acceder a ella por medio de PHP para realizar dichas funciones.

1. MySQL Workbecnh:

Evidentemente, necesitaremos conectarnos a la base de datos que tenemos con el dispositivo móvil. Desafortunadamente, no hay ninguna librería que nos permita realizar la conexión entre Java y phpMyAdmin.

Para solucionar dicho problema, hemos conectado dicha base de datos a el Workbench de MySQL por medio de crear una conexión utilizando el puerto 3306 (que es el que tenemos configurado para el servicio de mySQL del paquete de XAMPP)

1. Endpoints/API’S:

Dentro de nuestra aplicación, tambien utilizamos APIS que nos proporcionan distinta información útil. Utilizamos la API de Nominatum, que nos proporciona dos servicios disintos:

* Comprobación de si una calle especificada realmente existe:

Necesaria para comprobar si la ubicación del evento introducida por el usuario es una ubicación real.

* Pasar a latitud y longitud una ubicación:

Usada para poder obtener las coordenadas geográficas de una localidad. Esta inforamción será utilizada para calcular la distancia entre la ubicación del usuario (por medio de su GPS) y la localidad de los distintos eventos guardados en la base datos. Si la distancia es mayor que la delimitada por el usuario, dicho evento no se muesta al usuario.

Evidentemente, necesitaremos permitir las conexiones remota para que los usuarios puedan acceder a la información almacenada en la base de datos. Para ello, usaremos 2 servicios distintos:

* Amazon EC2:

Contaremos con una instancia de EC2 donde alojaremos la base de datos que usaremos para guardar la inforamción de las cuentas de los usuarios. En este servidor, instalaremos XAMPP, y crearemos las regals de entrada necesarias para poder acceder remotamente a la base de datos y a las páginas diseñas con XAMPP. El puerto que abriremos para este caso será el 3306.

* Ngrok:

Es una aplicación que usaremos para realizar pruebas y comprobar el correcto funcionamiento de nuestra pásginas creadas con PHP. Adeás, un aspecto importante es que te asigna automatica una URL. Funciona como un tunel, que nos permite redireccionar el tráficoel puerto que especifiquemos para permitir accesos remotoros a loss recursos guardas en la carpeta htcdocs. El puerto a abrir en cuestión sera el puerto 80.

Todas estas tecnologías descritas son las que tendremos que utilizar para poder poner en marcha los servicios ofrecidos en la aplicación. Importante recalcar que todas y cada una de ellas son necesarias para permitir un uso óptimo y correcto de la aplicación.

PANTALLAS

La aplicación consta de 11 pantallas distintas:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Logotipo, nombre de la empresa

Descripción generada automáticamente

REGISTRO CORREO

PANTALLA INICIAL

INICIO DE SESIÓN

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



VER EVENTOS

REGISTRO USUARIO

REGISTRO USUARIO

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

ASISTENCIAS

BUSCAR EVENTOS

CREAR EVENTOS

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente

PERFIL SEGUIDOR

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente

PERFIL

SEGUIDORES

Las pantallas son las siguientes:

* Pantalla Inicial:

Es la pantalla inicial de nuestra aplicación. Muestra el logo de la aplicación. Por debajo, busca en la base de datos de SQLlite la existencia de algún nombre de usuario o contraseña introducidas con anterioridad. Si existe, salta a la pantalla de Ver Eventos. En caso contrario, salta a la de inicio de sesión.

* Inicio de sesión:

Es la pantalla de inicio de sesión de nuestra aplicación. Loguea al usuario en la aplicación y, en caso de ser necesario, introduce el nombre y la contraseña introducidas en la base de datos de SQLite para poder loguearse automaticamente.

* Registro de Correo:

Es la pantalla encargada de registrar un correo en la aplicación. Lo utilizamos para poder enviar un correo de confirmación de creación de cuenta para la aplicación. Comprueba que el correo introducido no exista con anterioridad en la base de datos.

* Registro de Usuario:

Es la pantalla encargada de registrar un nombre de un usuario en nuestra aplicación. Comprueba que el nombre del usuario no exista con anterioridad.

* Registro de Contraseña:

Es la pantalla encarga de registrar la contraseña del usuario en la cuenta. Así mismo, desde aquí se envia un email de confirmación al correo del usuario desde el correo de la aplicación.

* Ver evento:

Permite ver los eventos cercanos en nuestra aplicación. Podemos escoger tanto el radio de lejania de los eventos, como buscar eventos por nombre o ubicación asi como encontrar usuarios.

* Crear Eventos:

Permite crear eventos. Los usuarios se podrán apuntar a este evento una vez haya sido creada.

* Asistencias:

Permite comprobar a que eventos y a cuantos eventos hemos confirmado la asistencia. Estan ordenados según la fecha de confirmación de asistencia. Desde aquí, podemos cancelar la asistencia del evento.

* Perfil:

Es la pantalla de perfil del usuario de la aplicación. Permite ver tanto los seguidores como a quien seguimos así como comprobar los eventos que hemos creado, y si han finalizado o no. Tambien desde aquí podemos editar el perfil.

* Perfil persona:

Es la pantalla de perfil de los usuarios. Sirve para poder comprobar la información del perfil de un usuario que nos siga. Podemos ver eventos creados y confirmar asistencias.

* Seguidores:

Es la pantalla para ver los seguidores que tiene un usuario especifico. Podemos acceder al perfil de estos seguidores pulsando sobre ellos, saltando así a la pantalla de seguidores.

Hay algunas pantallas que están contenidos dentro de otras, pues a estas se acceden por medio de distintos fragmentos. Esto se hace para dinamizar la interacción entre usuario, aplicación e interfaz, pues limitamos así la cantidad de pantallas de carga con las que hay que interactuar para poder realizar distintas operaciones.

MAPA NAVEGACIÓN

ASISTENCIAS

CREAR

PERFIL

REGISTRO CORREO

REGISTRO NOMBRE.U

REGISTRO CONTRASEÑA

INICIO DE SESIÓN

PANTALLA INICIAL

SEGUIDORES

VER

BUSCAR

BASE DE DATOS

Primero de todo. La base de datos que vamos a utilizar es una base de datos relacional. Hay 4 razones principales para ello:

1. Estructura de datos coherentes:

Con las bases de datos relacionales, nos podemos asegurar de que todos los datos guardados están debidamente organizados además de que son fácilmente accesibles por los distintos usuarios. Esto es un aspecto importante, pues nos podemos cerciorar de que nuestros clientes puedan visualizar información detallada de los dinstintos eventos creados.

1. Complejidad de las consultas:

Podemos elaborar consultas realmente complejas, juntando distintas tablas entre sí, así como utilizando distintas funciones, que serán utilizadas por los usuarios para poder encontrar eventos, ya sea delimitados por la distancia respecto a su ubicación o bien por el nombre, ubicación o creador del evento.

1. Escalabilidad:

Este punto se centra en que es capaz de manejar unas grandes volúmenes de datos y un gran volumen de usuarios. Es realmente importante pues es una aplicación que puede crecer rápidamente con el paso del tiempo-

1. Restricciones:

Podemos establecer restricciones de integridad para asegurarnos de que los datos introducidos en la base de datos cumplen con unos límites que hayamos constituido con anterioridad.

Todas estas ventajas hacen que la opción correcta para el desarrollo, consistan en pasar por medio de un modelo relacional en vez de un modelo NoSQL.

Así pues, la base de datos está construida usando el servicio de MySQL, por medio de phpMyAdmin, proporcionada por el paquete de XAMPP. Esto debido a que usamos PHP para realizar 2 operaciones distintas:

* Enviar un correo de confirmación de creación de cuenta al momento de crear una cuenta en la aplicación.
* Permitir a un usuario confirmar asistencia a un evento al momento de escanear el código QR de un evento por medio de la aplicación de Google Lens

Para poder realizar la conexión a la base de datos por medio de Java, hemos usado el Driver JDBC de MySQL. Hemos conectado la base de datos que esta guardada en phpMyAdmin a la aplicación de MySQL Worckbench por medio del puerto 3306. Así, hemos podido acceder a la base de datos usando Java sin afectar el acceso usando PHP.

Así mismo, para poder permitir conexiones remotas, hemos añadido la línea *bind-addres: 0.0.0.0* para poder hacer que dispositivos que se encuentras en nuestra misma red se puedan conectar a la base de datos y puedan realizar cambios por medio de un usuario creado previamente.

El modelo Entidad/Relación de la base de datos es el siguiente:

Diagrama

Descripción generada automáticamente

Podemos encontrar 4 tablas distintas:

Contienen toda la información referente a la cuenta del usuario. Guarda datos tales como el nombre del usuario, la contraseña, el email, la descripción de la cuenta del usuario y un booleano para conocer si la persona a validado su cuenta. Un usuario puede crear y asistir a muchos eventos.

Así mismo, un usuario puede seguir a otros muchos usuarios.

USUARIOS

Contiene toda la información necesaria para crear eventos. Contiene el identificador del evento, la información de la ubicación del evento (calle, ciudad, localidad), la hora y día del evento, el nombre junto con la descripción y las plazas, así como el nombre del creador del evento, el código QR del evento, la latitud y la longitud.

Un evento puede ser creado por un único y a un evento acuden varios.

EVENTO

SEGUIDORES

Es la tabla que se encarga de relacionar a los usuarios entre sí. Un usuario puede seguir a otro, para poder ver los eventos de este o bien poder ver la información de perfil de la otra persona, ya sea él número de seguidores o a los usuarios a los que sigue.

ASISTIR EVENTO

Es la tabla encargada de guardar los asistentes de un evento. Contiene el id del evento y el nombre de usuario del asistente. Realmente es una tabla intermedia resultante de una relación N:M entre la tabla de Evento y Usuario.

La base de datos contará con triggers y eventos para realizar distintas tareas de mantenimiento como podrían ser el borrado de eventos ya finalizados, así como de cuentas que no hayan confirmado su creación.

aplicación