

INFORME FINAL PROYECTO

Índice

Autor del proyecto.....	2
Nombre del proyecto.....	2
Plataformas de destino.....	2
Descripción funcional.....	2
Descripción técnica y elementos usados.....	3
Problemas surgidos.....	4
Tutorial de instalación.....	4

Autor del proyecto

Alexandre Valverde Comesaña

Nombre del proyecto

Organizador de tareas

Plataformas de destino

Xamarin para la app cliente de IOS, Windows Forms .NET para el cliente de Windows y XAMPP para el servidor y la base de datos en Windows 10.

Descripción funcional

El proyecto consta de 4 partes: La app cliente para IOS, el programa de Windows, el servidor (API Rest) y la base de datos Mysql.

La aplicación para IOS permite crear y visualizar tareas. Tiene la opción de marcarlas como completadas. Permite ordenar las tareas de varias formas. Las tareas están compuestas por los datos del cliente, localización y descripción de la tarea.

El programa de Windows permite crear usuarios, distinguiendo entre usuario normal o administrador.

El servidor se comunica con la app y la base de datos para el login del usuario y obtener y crear tareas y con el programa de Windows para crear usuarios.

La base de datos Mysql almacena los usuarios, tareas, clientes y localizaciones.

Descripción técnica y elementos usados

App cliente IOS:

Para desarrollarla he utilizado Visual Studio 2019 con Xamarin en OSX y Xcode 11 para la interfaz de usuario.

Para la conexión con el servidor utilizo las clases nativas de IOS adaptadas con mi propio código. Para realizar las conexiones utilizo hilos para esperar a obtener los datos y mostrar el texto cargando en movimiento para que no se bloquee la app. También he creado un botón que muestre un indicador de actividad cuando se pulsa mientras se obtienen los datos del servidor.

Utilizo SQLite para guardar las tareas obtenidas y poder ordenarlas.

Para guardar los datos de inicio de sesión del usuario utilizo “NSUserDefaults” que es la forma nativa de IOS para almacenar pares clave-valor de forma persistente.

Utilizo el GPS para obtener la localización del usuario y mostrar a que distancia se encuentran las tareas y mostrar la región en el mapa donde se encuentra el usuario para facilitar encontrar la ubicación al crear una nueva tarea.

Para leer los datos JSON obtenidos del servidor utilizo la clase nativa de IOS y para crearlos he creado mi propio código.

Cliente Windows

Para desarrollarlo he utilizado Visual Studio 2019 en Windows.

Servidor (API Rest)

Para desarrollar el servidor he utilizado Eclipse 2019 con Maven.

Para la autenticación de los usuarios utilizo JSON Web Tokens.

Devuelvo y leo datos en JSON o como texto simple.

Para las conexiones con la base de datos he creado un Pool de conexiones.

Para guardar encriptada la contraseña de los usuarios utilizo AES Encryption. La clave AES la creo desde el servidor y la guardo en un fichero.

Base de datos MYSQL

Para la conexión del servidor con la base de datos he creado un usuario con solo permisos para llamar a los procedimientos almacenados y que no se pueda acceder a todas las tablas.

Puntos no realizados

Por falta de tiempo el cliente de Windows solo permite crear usuarios.

Problemas surgidos

En la app de IOS, al obtener las tareas y mostrarlas en la tabla, según el tipo de tarea la muestro de un color u otro. El problema es que aunque obtiene bien los datos del servidor no siempre las muestra del color del tipo que tienen asignado.

El problema creo que viene de la reutilización de las celdas en la tabla, porque las primeras tareas las muestra del color correcto.

En la app de IOS, en muy pocas ocasiones y sin motivo aparente, en el momento de crear la tarea y validar los datos muestra que algún campo está vacío cuando no lo está, por lo que no permite crear la tarea.

He intentado comprobar que campo es el que da el error de que no tiene datos con un punto de interrupción pero de todas las veces que lo he probado no me da el error.

Tutorial de instalación

1. Instalar y lanzar XAMPP en Windows 10.
2. Crear base de datos Mysql con tablas y procedimientos.
3. Crear usuario Mysql en la base de datos y dar permiso para llamar a los procedimientos almacenados.
4. Modificar en el código del servidor el path donde se va a guardar la clave AES. Está en el package 'aes' y en la clase 'Encryptor'.
5. Modificar en el código del servidor la url de la base de datos en el package 'mysql' y la clase 'MysqlPool'. También cambiar el nombre y contraseña del usuario de Mysql (en caso de que fuera necesario).
6. Lanzar servidor con Tomcat.

7. Crear clave AES con plugin RESTer para Chrome:

URL → `http://urlserver:nº puerto/tareasvivas/app/aes/creakey`

Method → POST

Accept → text/plain

8. Reiniciar servidor.

9. Ejecutar cliente Windows y crear usuario.

10. Ejecutar app cliente IOS desde Visual Studio 2019 con Xamarin en OSX y loguearse con el usuario creado para visualizar y crear tareas.