

**Curso: 2020/21**

**PMDM03 Tarea evaluativa.**

**José Mari Tomé M*ejías***

**Desarrollo de aplicaciones Multiplataforma**

**Programación multimedia y dispositivos móviles**

**ÍNDICE**

1. INTRODUCCIÓN 1

2. DIFERENTES APARTADOS DE LA TAREA 1

3. AUTOEVALUACIÓN 1

4. BIBLIOGRAFÍA 1

# INTRODUCCIÓN

A lo largo de esta unidad has aprendido a trabajar con alguna de las posibilidades que te ofrecen los **dispositivos móviles** en lo que respecta a su **conectividad**: conexiones **HTTP** y **HTTPS**. Así mismo, también has visto algunos otros aspectos como los relacionados con la **persistencia** (**bases de datos basadas en registros** o el acceso a **sistemas de archivos** de los elementos memoria externos), **extensiones** y **paquetes opcionales**.

Una vez finalizada esta unidad se puede decir que dispones de una visión bastante amplia del tipo de aplicaciones que puedes desarrollar para un dispositivo móvil. Además de crear aplicaciones con **interfaces gráficas de usuario**, ahora puedes también dotarlas de mecanismos de **persistencia** y **conectividad**.

Se trata de dotar de persistencia y conectividad a la app que desarrollaste en la UD anterior. Debes tener en cuenta que para poder gestionar las tareas, estas deberán tener un identificador.

Por último, deberás hacer un pequeño resumen con las dificultades que te has encontrado a lo largo del desarrollo y un manual en el que me indiques cómo debo realizar las pruebas de la aplicación a partir del código fuente (no se debe incluir la carpeta que incluye los módulos). El documento debe cumplir con los siguientes requisitos para ser corregido:

* Fichero PDF
* Portada con nombre y titulo
* Tabla de contenidos
* Formato adecuado (sin colores, texto justificado, páginas numeradas (la portada no debe estarlo)...)

# DIFERENTES APARTADOS DE LA TAREA

**Dificultades del desarrollo**

* Al comprobar el tipo de variables boleanas he tenido que cambiar la manera de comprobar y comprobar el tipo de dato, porque me cargaba el valor.
* Al comprobar con la función fromJson del modelo task, el map de service-tak que carga las tareas, devuelve las mismas aunque falten campos, el return del fromJson entra y crea una nueva.

 datos.map((task) => Task.fromJson(task));

Solucionado con un filtro en los objetos que recibe del método map.

* Al crear un nuevo elemento y no recargar la página el id del elemento es el que se le da su valor en la clase Task id= -1, al hacer cualquier modificación si recargar y no recibir los datos del servidor da el siguiente error al crear el servidor el id:

PUT http://localhost:3000/todos//-1.

* Incidencia en el home.module a la hora de cambiar el estado de tarea terminada a no terminada, si se usa la misma función para los dos checked, se crea un bucle al cambiar de estado continuamente.

**Manual de arranque del programa**

Para iniciar el servidor colocarse en la carpeta del servidor y abrirlo con Visual Studio Code, arrancar el servidor con el comando:

npm start –watch db.json

Tener en cuenta que los campos del modelo tarea deben tener los mismos nombres que la aplicación que lo consume

   {

      "id": ,

      "description": ,

      "isImportant": ,

      "finished":

    }

Para iniciar la aplicación de ionic lo primero es instalar todas las dependencias que usa el proyecto con el comando npm install

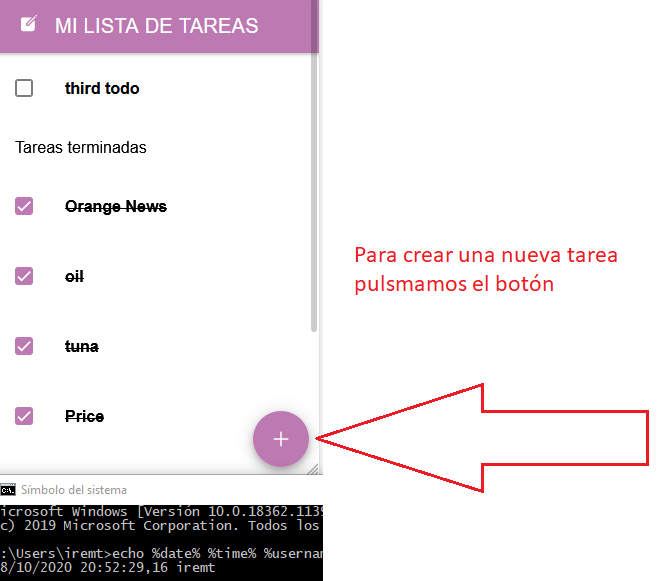
posicionado dentro de la carpeta del proyecto.

Una vez cargadas todas las dependencias lanzamos la aplicación con el comando

ionic s

**Manual de pruebas**

1. Crear nueva tarea

****

# AUTOEVALUACIÓN

# BIBLIOGRAFÍA