

# ENUNCIADO PEC 2

**Programación reactiva:**

**RxJS + NgRX**

**Desarrollo front-end avanzado**

**Máster Universitario en Desarrollo de sitios y aplicaciones web**

UOC

Universitat Oberta  
de Catalunya

## Contenido

- Introducción
- Objetivos
- Formato y fecha de entrega
- Enunciado
- Puntuación



# Introducción

En esta segunda práctica nos centraremos en utilizar el patrón de flujo de datos **Redux** estudiado en la teoría de este segundo tema. Por tanto, utilizaremos el paquete **NgRx** en nuestra aplicación **Angular** y para ello deberemos construir el **store**, acciones, selectores, **reducers** y efectos apropiados. Toda la gestión de asincronismos del patrón **Redux** se gestionará utilizando Observables con la implementación de la biblioteca **RxJS**.

## Objetivos

Los objetivos que se deben conseguir con esta práctica son:

- Sentar las bases de una aplicación asíncrona y reactiva utilizando la biblioteca **RxJS** y el patrón Observable e Iterator.
- Utilizar el patrón de flujo de datos **Redux** usando la biblioteca **NgRx**.
- Navegar en una SPA de varias pantallas en las que la modificación del flujo de datos se lleve a cabo sin que provoque una pérdida de control del desarrollo. Es decir, controlar el flujo de datos entre las diferentes pantallas sin que se des controle cuando la aplicación web no es un conjunto de scripts, sino una aplicación de gran envergadura.

## Formato y fecha de entrega

Se entregará un comprimido en formato **.zip** dentro del cual tendremos:

- Una carpeta **Ejercicio\_1** con un **.zip** del proyecto de la aplicación TODO App junto con un pequeño documento con las impresiones al desarrollar la aplicación, problemas encontrados, aportaciones al foro, ...
- Una carpeta **Ejercicio\_2** con el proyecto comprimido en formato **.zip** que resuelve la práctica junto con un pequeño documento explicando las decisiones de diseño más relevantes. Para desarrollar el proyecto del ejercicio 2 partiremos de la solución propuesta de la práctica 1.

Hay que recordar que los proyectos no deben incluir el directorio **"node\_modules"**.

La fecha de entrega se mostrará en el aula de la asignatura.

# Enunciado

Dividiremos el desarrollo de esta práctica en dos apartados.

El primer ejercicio trata de replicar la aplicación TODO App que se explica al documento **Teoria\_REDUX\_parte\_2\_es.pdf** e implementar alguna funcionalidad extra.

Por otra parte, en el segundo ejercicio, deberemos aplicar todo lo aprendido a nuestro propio proyecto. Partiremos de la solución de primera práctica y deberemos aplicar el patrón **Redux** en los apartados que posteriormente explicaremos.

## Ejercicio 1 – Primer contacto con NgRx

La primera tarea que desarrollaremos en esta práctica será trabajar con el documento **Teoria\_REDUX\_parte\_2\_es.pdf**. Deberemos seguir todos los pasos que se explican en este documento para construir poco a poco el proyecto. Tenemos que desarrollar cada parte del documento ya que al final tendremos que entregar el proyecto resultante.

Una vez implementados todos los pasos de este documento, deberemos de hacer dos implementaciones extra:

- Implementar la acción de **completar todas las tareas**
- Implementar la acción de **eliminar todas las tareas que estén completadas**

Posteriormente, se deberá ir al hilo del foro a consultar y aclarar entre todos los conceptos o implementaciones que no se vean claros, o que puedan parecer interesantes para el proceso de aprendizaje. Como mínimo todos los alumnos deben pasar por el hilo para comentar sus impresiones de esta implementación.

Al ser el primer contacto con **NgRx** puede resultar complicado asimilar todos los conceptos y, por tanto, el ejercicio consiste justamente en eso, en analizar una solución para comenzar a plantear la de nuestro proyecto.

Por tanto, de este primer ejercicio debemos entregar un **.zip** con el proyecto de la aplicación TODO App, igual que tenemos al documento **Teoria\_REDUX\_parte\_2\_es.pdf** más las dos implementaciones extra, junto con un pequeño documento donde expliquemos brevemente las impresiones al implementar dicha aplicación, problemas encontrados y cómo los hemos solucionado, las aportaciones que hemos hecho al foro del aula, ... Este documento no debería ocupar más de una página.

## Ejercicio 2 – Aplicar NgRx

Debemos retomar nuestro proyecto de la primera práctica en el que estábamos construyendo una aplicación que consistía en gestionar actividades turísticas.

Antes de nada, piensa en estructurar la aplicación en diferentes módulos (heredado o refactoring de la práctica anterior). Deberemos tener los siguientes módulos (al menos a nivel de aplicación):

- Login
- Profile (información del usuario, ya sea turista o empresa)
- Activities

Además, ahora habrá que pensar qué información necesitamos almacenar en nuestro **Store**, es decir, la que será compartida por toda la aplicación, es aquí donde cobra sentido el patrón **Redux**, cuando tenemos muchas pantallas que comparten datos y se interrelacionan entre sí. Nuestro **Store** inicialmente podrá tener los siguientes campos/secciones:

- Login. Información de autenticación.
- User. Información sobre el usuario (teniendo en cuenta que este almacén será diferente si es un turista o una empresa).
  - Profile. Información personal.
  - Education. Información académica.
- Activities. Información sobre las actividades turísticas. (habría que hacer el mismo ejercicio de pensar cómo se podría confeccionar el almacén).

**Por lo tanto, hay que conectar todos los formularios de la primera práctica con sus correspondientes flujos de datos del patrón Redux usando NgRx. Por supuesto, todos los effects pueden/deben estar simulados, puesto que el servidor no debería interactuar ni ser una preocupación en este punto.**

# Puntuación

A continuación, mostramos cuánto puntúan cada uno de los apartados de la práctica para obtener la nota final de la misma.

- Ejercicio 1 [ **4 puntos** ]
- Ejercicio 2 [ **6 puntos** ]
  - Pantalla login [ **1 punto** ]
  - Pantalla profile (turista) [ **2 puntos** ]
    - Sección profile
    - Sección education
  - Pantalla activities (home) [ **1 punto** ]
  - Pantalla my activities [ **1 punto** ]
  - Pantalla admin [ **1 punto** ]