

# Orientación a Objetos 2

## Trabajo Práctico de Frameworks

### Objetivo

En este trabajo práctico, se espera que usted desarrolle una aplicación Web utilizando el framework Seaside. La aplicación Web esperada está basada en el ToDoList, el primer ejercicio, de la práctica 5.

### Tareas

1. Revise su solución del ejercicio del ToDoList con un ayudante y refactorice su solución si hace falta. También repase la guía de buenas prácticas de programación.
  - a. También revise sus tests de unidad SUnit.
2. Si aún no lo hizo, realice el tutorial de Seaside provisto por la cátedra.
3. Implemente una aplicación usando Seaside que contemple los requerimientos definidos a continuación. Tenga en consideración que dados los nuevos requerimientos, su solución original del ToDoList puede requerir también algún nuevo cambio.

### Entrega

Usted debe entregar un archivo .st, que debe ser exportado haciendo un “File Out” del paquete que debe contener todas las clases necesarias para correr y probar su aplicación (clases del modelo, tests de unidad y las clases de aplicación Seaside).

### Fecha de entrega

Lunes 25 de Junio

### Requerimientos

Usted debe desarrollar una aplicación Web para el control de un ToDoList. Las funcionalidades mínimas requeridas están especificadas a partir de los siguientes mockups.

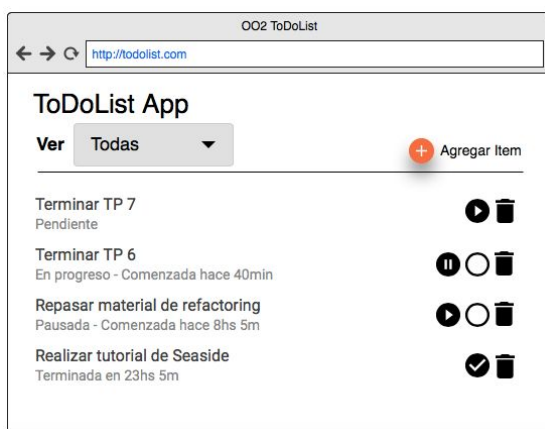
Los mockups son meramente ilustrativos y no es necesario que su aplicación implemente exactamente el estilo de la interfaz de usuario. Solo se requiere que la aplicación muestre los datos y funcionalidades definidas en el enunciado.

El Mockup 1 presenta la vista principal de la aplicación, en donde se debe mostrar al usuario:

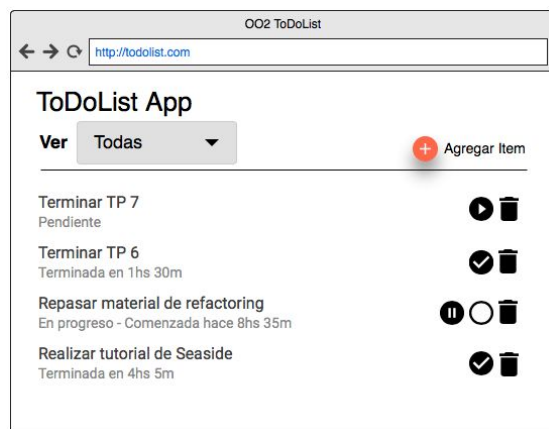
- Un listado de ToDoItems, mostrando su estado y un menú de operaciones según sea dicho estado. El listado de ToDoItems debe estar ordenado descendientemente según la fecha de creación de los mismos.

- Un menú que permita filtrar los ToDoltem visualizados según su estado, en el caso del Mockup 1, el usuario está visualizando todos los ToDoltem
- Un botón para agregar un nuevo ToDoltem.

El menú correspondiente a cada ToDoltem debe mostrar las operaciones correspondientes al estado del ToDoltem, pero además para todos los ToDoltem debe estar disponible la operación para eliminarlo. Cuando el usuario realiza un click sobre alguna de las operaciones, la UI debe actualizarse. Por ejemplo, el Mockup 2 muestra la misma lista de ToDoltems que el Mockup 1, excepto que el ToDoltem “Terminar TP 6” ha sido finalizada, mientras que el ToDoltem “Repasar material de refactoring” ha vuelto a estar en progreso.

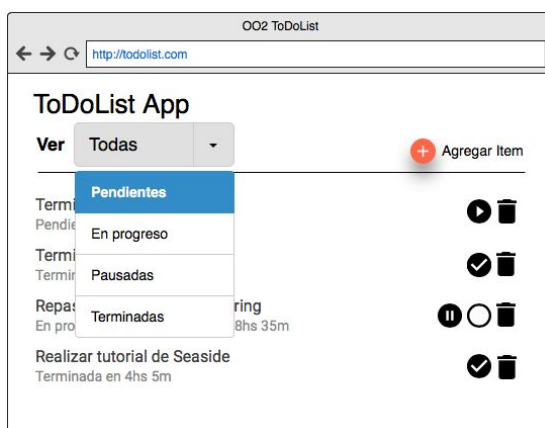


**Mockup 1:** Vista general de la aplicación



**Mockup 2:** Vista general de la aplicación, cambios de estados respecto a Mockup 1

Su aplicación debe permitir filtrar los ToDoltem según un estado en particular, los Mockup 3 y 4 detallan esta funcionalidad.

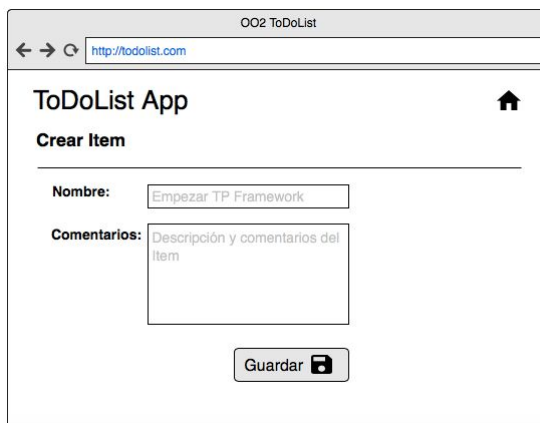


**Mockup 3:** Menú para filtrar ToDoltem por estado

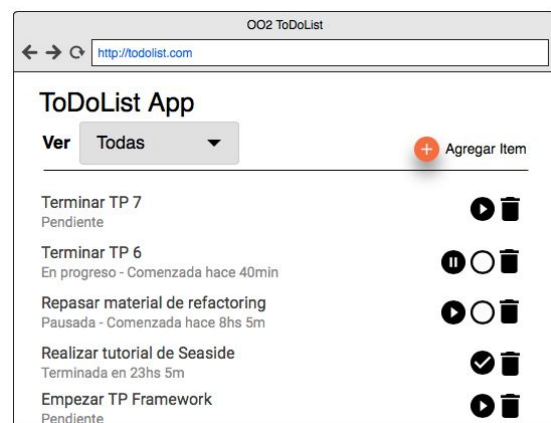


**Mockup 4:** Lista de ToDoltem finalizados

Cuando el usuario hace click en el botón “Agregar ítem”, la aplicación debe mostrar al usuario un formulario para crear un nuevo ToDoltem especificando su nombre y un comentario, como muestra el Mockup 5. Una vez completada la información requerida (nombre y comentario), al hacer click en el botón “Guardar”, debe crearse un nuevo ToDoltem y volver a visualizar la lista con todos los ToDoltem (es decir, sin filtrar los ToDoltem), como muestra el Mockup 6. En este caso, la única validación a tener en cuenta es que el usuario haya ingresado un texto válido como nombre del ToDoltem.

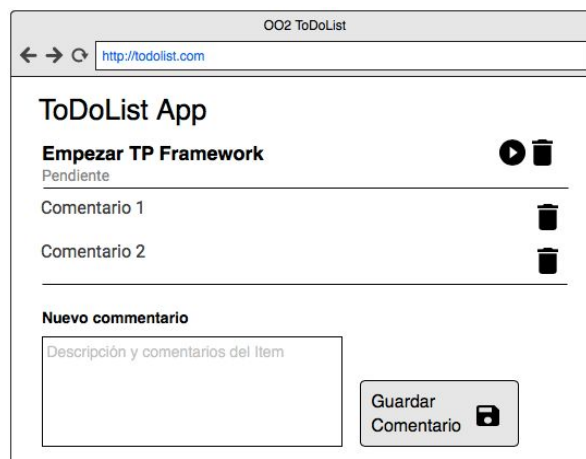


**Mockup 5:** Creando un nuevo ToDoltem



**Mockup 6:** ToDoltem creado

Finalmente, si el usuario hace click en el nombre de un ToDoltem, la aplicación debe mostrar la UI del detalle del ToDoltem, donde se muestra su nombre, su estado, las operaciones posibles, el listado de los comentarios y además un campo para agregar un nuevo comentario para el ToDoltem, como se ilustra en el Mockup 7.



**Mockup 7:** Vista del detalle de un ToDoltem