

**Trabajo Práctico - Informe Final**  
**Guía para pruebas mediante Postman**  
2do cuatrimestre 2020

---

**Implementación de persistencia en un modelo relacional**

**aplicado a un sistema de Seguimiento de Items**

**Guía para pruebas mediante Postman**

<b>GUÍA PARA TESTEAR LOS SERVICIOS IMPLEMENTADOS.....</b>	<b>2</b>
INTRODUCCIÓN:.....	2
SERVICIOS RELACIONADOS CON TIPO DE ESTADO:.....	2
SERVICIOS RELACIONADOS CON TIPO DE ITEMS:.....	2
SERVICIOS RELACIONADOS CON PROYECTOS:.....	3
SERVICIOS RELACIONADOS CON ITEMS:.....	3
SERVICIOS RELACIONADOS CON ESTADOS:.....	4

**Trabajo Práctico - Informe Final**  
**Guía para pruebas mediante Postman**  
2do cuatrimestre 2020

---

## Guía para testear los servicios implementados

### ***Introducción:***

Para testear el funcionamiento de la API REST de la prueba de concepto se utilizó Postman. Se creó una colección denominada POC organizada por carpetas, las que reúnen servicios según temática: Proyectos, Items, Estados, Tipos de Items, Tipos de Estados.

El archivo postman se denomina *POC.postman\_collection.json*, el cual se adjunta al informe.

### ***Servicios relacionados con tipo de estado:***

Los servicios implementados en la carpeta Tipo Estado de Postman, son:

- **Listar los tipos de estado:** en Postman **List**, es un request de tipo GET. Invoca al servicio **ListaTipoEstados()** de la API REST, el que retorna una lista de DTOs de los tipos de estado que existentes.
- **Crear un tipo de estado:** en Postman **Create**, es un request de tipo POST. Invoca al servicio **createTipoEstado** de la API REST que solicita como parámetros el nombre del nuevo tipo de estado, si es inicial y si es final. Controla unicidad del nombre. Retorna el DTO del nuevo tipo de estado creado.

### ***Servicios relacionados con tipo de items:***

Los servicios implementados en la carpeta Tipo Item de Postman, son:

- **Listar los tipos de estado:** en Postman **List**, es un request de tipo GET. Invoca al servicio **ListaTipoItem()** de la API REST, el que retorna una lista de DTOs de los tipos de item que existentes.
- **Crear un tipo de item:** en Postman **Create**, es un request de tipo POST. Invoca al servicio **createTipoItem** de la API REST, que solicita como parámetro el nombre del nuevo tipo de item. Controla unicidad del nombre.

**Trabajo Práctico - Informe Final**  
**Guía para pruebas mediante Postman**  
2do cuatrimestre 2020

---

Retorna el DTO del nuevo tipo de item creado.

### ***Servicios relacionados con proyectos:***

Los servicios implementados en la carpeta Proyecto de Postman, son:

- **Listar los proyectos sin mostrar sus items, ordenados por fecha de inicio descendente:** en Postman **ListSinItemsOrbyFinicioDesc**, es un request de tipo GET. Invoca al servicio **ListProyectos()** de la API REST, el que retorna una lista de DTOs de proyectos ordenándolos por fecha de inicio descendente. Esta funcionalidad no muestra los items de cada proyecto.
- **Crear un proyecto:** en Postman **Create**, es un request de tipo POST. Invoca al servicio **createProyecto** de la API REST que recibe dos parámetros: uno para el nombre del proyecto y otro para la fecha de inicio del mismo. Retorna el DTO del nuevo estado creado.
- **Modificar el nombre de un proyecto:** en Postman **ModifyNombreProyecto**, es un request de tipo POST. Invoca al servicio **modifyNombreProyecto** de la API REST que recibe dos parámetros: uno para el id del proyecto y otro para su nuevo nombre. Controla que el id sea de un proyecto existente. Retorna el DTO del proyecto con su nuevo nombre.
- **Crear un item de proyecto:** en Postman **AddItemAProyecto**, es un request de tipo POST. Invoca al servicio **addItemAProyecto** de la API REST que recibe como parámetros el id del proyecto, el nombre del item, su prioridad y el id del tipo de item. Controla que tanto el id del proyecto como el id del tipo de item, existan. Retorna el DTO del nuevo item creado para el proyecto.
- **Listar los items de un proyecto:** en Postman **ListItemsProyecto**, es un request de tipo GET. Invoca al servicio **ListItemsProyecto** de la API REST que recibe como parámetro el id del proyecto. Controla que el id del proyecto exista. Retorna una lista de DTOs de los items del proyecto, sin sus estados.

### ***Servicios relacionados con items:***

Los servicios implementados en la carpeta Item de Postman, son:

**Trabajo Práctico - Informe Final**  
**Guía para pruebas mediante Postman**  
2do cuatrimestre 2020

---

- **Mostrar el último estado (o estado actual) de un ítem:** en Postman denominado **LastEstadoItem**, es un request de tipo POST. Invoca al servicio ***lastEstadoItem*** de la API REST, que recibe como parámetro el id del ítem. Controla que el id corresponda a un ítem existente y que el ítem tenga estados asociados. Retorna el DTO del último estado del ítem, si existiese.
- **Agregar un estado a un ítem:** en Postman es un request de tipo POST, denominado **AddEstadoItem**. Invoca al servicio ***addEstadoItem*** de la API REST, que recibe como parámetros el id del ítem y el id del tipo de estado. Controla que ambos ids correspondan a un ítem y a un tipo de estado existentes. También controla la correctitud del nuevo estado. Si el ítem no tiene estados, debe ser del tipo inicial. Si el ítem tiene estados, controla que si el último no es del tipo final. Si el estado a agregar es correcto, crea el estado y lo agrega al ítem. Retorna el DTO del estado creado.
- **Eliminar el último estado de un ítem:** en Postman denominado **DeleteUltimoEstadoItem**, es un request de tipo POST. Invoca al servicio ***deleteLastEstadoItem*** de la API REST, que recibe como parámetro el id del ítem. Controla que el id corresponda a un ítem y que tenga estados. Si ambos controles son correctos, elimina el último estado del ítem.
- **Listar los estados de un ítem:** en Postman **ListEstadosItem**, es un request de tipo GET. Invoca al servicio ***ListEstadosItem*** de la API REST que recibe como parámetro el id del ítem. Controla que el id del ítem exista. Retorna una lista de DTOs de los estados del ítem, si existiesen.

### ***Servicios relacionados con estados:***

El servicio implementado es:

- **Mostrar un estado:** en Postman denominado **ShowEstado**, es un request de tipo GET. Invoca al servicio ***showEstado*** de la API REST que solicita como parámetro el id de un estado. Controla que el id ingresado sea un id de estado existente. Retorna el DTO del estado.