

Práctico 2

Los objetivos del práctico son:

1. avanzar en la comprensión de los conceptos vistos en el teórico a través de la práctica
2. avanzar en el fortalecimiento de capacidades para la selección, utilización e integración de metodologías, tecnologías y herramientas heterogéneas

Los temas que aborda el práctico son:

- Java Message Service (JMS) y Message Driven Beans
- Servicios Web SOAP / REST
- Interacción con Servicios Externos
- Java Persistence API
- Frameworks Javascript
- Aplicaciones Android

Fechas Importantes:

- Publicación del Práctico: 27/08/2025
- Entrega del Práctico (gitlab): Semana del 06/10/2025

Ejercicio 1: JMS y Message Driven Beans

Objetivo: El objetivo de este ejercicio es avanzar en la comprensión de Java Message Service (JMS) y Message Driven Beans (MDBs).

Se quiere implementar una solución de mensajería capaz de paralelizar el alta de la entidad con la que se trabajó en el Práctico 1. Para ello, se decidió implementar un MDB que reciba mensajes provenientes de una cola de mensajes denominada `queue_alta_[entidad]` (p. Ej. `queue_alta_empresa`). El formato del mensaje es un texto plano en el cual se incluyen los atributos para el alta de la entidad separados por un “|” (pipe).

1.1) Implementar el MDB para realizar el alta de la entidad.

1.2) Extender las aplicaciones Web y Java del Ejercicio 1 del Práctico 1, de forma tal que se pueda realizar el alta de la entidad a través de dichas aplicaciones y en base a lo desarrollado en 1.1.

Ejercicio 2: Servicios Web SOAP / REST

Objetivo: El objetivo de este ejercicio es avanzar en la comprensión de la tecnología de Servicios Web y, en particular, de los mecanismos que brinda Jakarta EE para trabajar con Servicios Web SOAP y REST.

Se desea permitir que otras aplicaciones y sistemas interactúen con el Gestor de Entidades a través de Servicios Web.

2.1) Exponer a través de Servicios Web (SOAP y REST) las funcionalidades que brinda la capa de negocio implementada en el Ejercicio 1 del Práctico 1.

2.2) Utilizar la herramienta SoapUI¹ para invocar los Servicios Web desarrollados.

Ejercicio 3: Java Persistence API

Objetivo: El objetivo de este ejercicio es avanzar en la comprensión de Java Persistence API (JPA).

Implementar una nueva capa de datos que utilice JPA para almacenar los datos de entidades en una base de datos PostgreSQL.

Ejercicio 4: Servicios Externos (opcional)

Objetivo: El objetivo de este ejercicio es profundizar en la interacción con servicios externos.

Se quiere extender el Gestor de Entidades para obtener datos de servicios externos.

4.1) Implementar un Servicio Web REST que reciba un identificador de un Servicio Web externo y retorne datos de entidades obtenidas a través de dicho servicio. Se deben utilizar al menos dos servicios externos.

Nota: El Servicio Web REST a implementar puede recibir otros parámetros.

4.2) Invocar el servicio implementado utilizando: i) un explorador Web y ii) la herramienta Curl²

¹ <https://www.soapui.org/>

² <https://curl.haxx.se/>

Ejercicio 5: Aplicaciones Android (opcional)

Objetivo: El objetivo de este ejercicio es avanzar en la comprensión del desarrollo de aplicaciones Android.

Se desea implementar una aplicación mobile Android que permita interactuar con el Gestor de Entidades.

Utilizando los Servicios REST desarrollados en el Ejercicio 2, implementar un cliente Android que liste y permita ver el detalle de las entidades (i.e. valores de sus atributos).

Ejercicio 6: Frameworks Javascript (opcional)

Objetivo: El objetivo de este ejercicio es comprender el funcionamiento de los frameworks Javascript en el desarrollo de aplicaciones Web modernas.

6.1) Desarrollar una aplicación utilizando un framework Javascript (p. ej. Angular, ReactJS, Ember.JS). La aplicación debe permitir seleccionar un atributo por el cual realizar una búsqueda de entidades, especificar un valor para la búsqueda e invocar el Servicio Web REST implementado en el Ejercicio 2 para realizar la búsqueda utilizando el valor. Luego se deben desplegar al usuario las entidades obtenidas, manteniendo un histórico de las búsquedas del lado del cliente (p. ej. una tabla paginada).

6.2) Modificar la aplicación anterior para mostrar en todo momento la cantidad de usuarios de cualquier cliente (p. ej. Web, Android) que hayan utilizado el Servicio Web REST en las últimas 24 horas. Para ello es necesario utilizar el mecanismo de Push Notifications, de forma de lograr el envío de información desde el servidor a los clientes. Se recomienda la utilización de algún servicio como Firebase, Parse, Pusher o similares.