

Trabajo Práctico 1

Objetivo

🎯 Generar un proyecto backend (un servicio web sencillo) que permita la gestión de personas.

📁 Como guía práctica más detallada recomendamos clonar el repositorio:

<https://github.com/jose-manuel-suarez/ppaasoftware> (subdirectorío asociado: Practica 1)

💬 Canal de consultas (Discord): <https://discord.gg/XAdRPKVZCT> (servidor PPAA de Software)

✉ Contacto (email): lbibbo@lifa.info.unlp.edu.ar y jsuarez@lifa.info.unlp.edu.ar

Ejercicio 1

Creación de un proyecto backend, que representa nuestro servicio web para la creación de personas y su gestión (mediante CRUDs) utilizando una API REST.

Ejercicio 2

Configuración de la conexión a una base de datos, que nos servirá para probar las operaciones realizadas por cada endpoint expuesto por la API REST. Utilizaremos una BBDD en memoria del servidor, que tendrá datos inicializados al lanzar la aplicación, únicamente a fines de pruebas.

Ejercicio 3

Creación de un servicio web sencillo respetando el modelo de capas sobre una arquitectura monolítica, que realizará un CRUD sobre una entidad Persona.

- Ejercicio 3.a)
 - Crear la entidad Persona (modelo de datos), con al menos los siguientes atributos:
 - Id
 - Apellidos
 - Nombres

- Fecha de nacimiento
- Estos atributos deben ser públicamente accesibles (getters y setters).
- Deben poder configurarse al instanciar un modelo de Persona.
- Crear un DTO (objeto de transferencia de datos) para la Persona, con al menos los siguientes atributos:
 - Id
 - Nombre completo (surge de la concatenación de los apellidos y nombres siguiendo el patrón: [apellidos], [nombres])
 - Edad
- Definir un repositorio para las Personas.
- Ejercicio 3.b)
 - Crear la capa de servicios de la aplicación.
- Ejercicio 3.c)
 - Crear la capa de controladores de la aplicación.

Ejercicio 4

Inicialización de los datos de pruebas.

Si nuestra BBDD es mantenida en memoria del servidor web, es importante que permitamos que tenga ciertos datos por defecto al iniciar.

Ejercicio 5

Gestionar el avance y seguimiento colaborativo del proyecto mediante un DCVS (recomendamos GitHub o GitLab). Debe entregarse el acceso a dicho repositorio que permita el análisis del código fuente.

Dockerizar el proyecto y generar una imagen del mismo. Debe entregarse dicha imagen del proyecto.

Mantener una batería de invocaciones de pruebas y entregarlas para permitir comprobar las pruebas efectuadas.

 Sitio: [GitHub](https://github.com)

 Sitio: [GitLab](https://gitlab.com)

Ejercicio 6

Documentar la arquitectura de la aplicación utilizando el modelo C4.

Al menos en esta etapa inicial del proyecto, será suficiente con la documentación de los 3 niveles superiores de la aplicación, pudiendo prescindir de la documentación asociada al código de la aplicación (4º nivel), que consideraremos de realización y entrega opcional.

 Sitio: [C4 model](#)

Ejercicio 7

Utilizar los principios de 12-Factor y considerarlos ante cualquier decisión arquitectónica sobre el proyecto.

 Sitio: [The Twelve-Factor App](#)

Entregables

- Acceso al código en GitLab.
- Acceso a la imagen Docker generada.
- Colección de escenarios de pruebas (PostMan, ThunderClient u otros).
- Acceso a la documentación de la arquitectura (esquemas C4).