

# Práctica 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/ppaassoftware> (subdirectorío asociado: Practica 1)

💬 Canal de consultas (Discord): <https://discord.gg/y7R48Crr> (servidor PPAA de Software)

✉ Contacto (email): [lbibbo@lifa.info.unlp.edu.ar](mailto:lbibbo@lifa.info.unlp.edu.ar) y [jsuarez@lifa.info.unlp.edu.ar](mailto: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 (CRUDs) mediante 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.

## 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.