



# Certified Tech Developer

The Ultimate Degree

## Back End I

# Clínica dental

## Objetivo

El objetivo de esta actividad es poder integrar los conocimientos vistos hasta el momento. Te proponemos realizarla en forma individual, pero tené en cuenta que podés consultar con tus compañeros en las mesas de trabajo. Luego, podrán resolver las dudas junto al docente. A tener en cuenta:

- Ejercicio individual
- Nivel de complejidad: alto 🔥🔥🔥

## Desafío

Una clínica dental necesita informatizar su operatoria. Por lo cual, te solicitan un sistema para la gestión de sus pacientes, donde puedan darlos de alta, buscarlos, eliminarlos y modificarlos. Al registrar un paciente los datos que se le solicitan son:

- Apellido
- Nombre
- DNI
- Fecha de ingreso
- Domicilio
  - Calle
  - Número



- Localidad
- Provincia

Además, le agregaremos un ID ~~autoincrementable~~ tanto a los pacientes como a los domicilios.

Se pide utilizar H2 como base de datos, aplicar el patrón DAO y testear con JUnit. Tener en cuenta que el modelado de clases debe contar con al menos dos clases: **PACIENTE** y **DOMICILIO**, con la consideración de que los pacientes podrán tener solo un domicilio.

Crear solo una clase de servicio, PacienteService, y crear por cada entidad un DAO, es decir, DomicilioDAOH2 y PacienteDAOH2. Al guardar y buscar un paciente en PacienteDAOH2 deberás invocar el guardar y buscar de DomicilioDAOH2.

## Tests de aceptación

Guardar en la base de datos dos o más pacientes con sus respectivos domicilios antes de ejecutar los tests.

Utilizando la capa de servicio:

1. Agregar un paciente con domicilio y buscarlo en la base de datos por ID.
2. Eliminar el paciente anteriormente creado y luego buscarlo, debe tener también el domicilio.
3. Buscar e imprimir en consola todos los pacientes con sus domicilios, serán los pacientes que agregamos antes de ejecutar el primer test.