

Desarrollador de Software

Farmacia CRUD

Problemática

Para mejorar la experiencia de un afiliado y/o paciente, la farmacia default que se ofrece para adquirir un medicamento debe ser la más cercana a su ubicación. Para esto, una de las funcionalidades que se necesita es conocer la más cercana a un punto dado (latitud y longitud), tomando en cuenta que este punto es la ubicación del paciente.

Requerimientos

1. Diseñar e implementar un servicio que exponga en su API las operaciones CRUD (únicamente creación y lectura por id) de la entidad **Farmacia** y la consulta de la farmacia más cercana a un punto dado.

```
GET http://localhost:<port>/api/farmacia?lat=<number>&lon=<number>
```

2. Diseñar e implementar (Test Unitario) al menos un caso de prueba exitoso y uno no exitoso de dominio para la operación de creación de una farmacia.
3. Diseñar e implementar al menos un test unitario exitoso del cálculo de distancias.

Datos Relevantes

Los campos de dicha entidad deben ser **id**, **nombre**, **dirección**, **latitud** y **longitud**.

Evaluación

Se tendrá en cuenta:

- El correcto funcionamiento del servicio.
- Correcta ejecución de los Test Unitarios.
- Diseño de la solución.
- Arquitectura del servicio (clean architecture).
- Diseño de la API.
- Manejo de errores.
- Claridad del código (clean coding, estándar de programación, uso de logs, etc.).
- Claridad en la documentación (ver Consideraciones).

Consideraciones

- Las farmacias creadas se deben persistir en una base de datos.
- La documentación de compilación y ejecución debe estar en el README del proyecto (dependencias y comandos a ejecutar).
- Documente el API suponiendo que no va a tener interacción con los programadores que la vayan a consumir.
- La entrega es el repo de github/gitlab/Bitbucket (a elección) que debe ser **privado** y se va a compartir el acceso con los usuarios indicados en el correo.
- Se valorará el agregado de **login** y **tracing**.
- **Fecha límite de entrega 1 semana** luego de recibido el mensaje.

Extras Deseables

- Usar Docker para build y run del servicio.
- Usar docker-compose para build y run del stack (servicio y base de datos)
- API documentada con alguna herramienta.
- Uso de Linters (ESLint, CodeSniffer, Otros).