Examen Backend Cash

Deberá implementar una app que exponga una API REST de usuarios. Dado un ID de usuario devuelva información del mismo, con la siguiente estructura de respuesta:

**GET /users/{id}**

{

“id”: 1,

“email”: “[test@app.com.ar](mailto:test@app.com.ar)”,

“firstName”: “Pepe”

“lastName”: “Argento”

“loans”: [

{

“id”: 1,

“total”: 2.500

“userId”: 1

}

]

}

Se deberán implementar análogamente los métodos de **POST** para alta y **DELETE** para baja de usuarios. El alta de usuarios no incluye loans. El borrado también borra los loans.

Además, armar otra API REST (dentro de la misma app) que devuelva el listado de préstamos (loans) pudiendo filtrar por ***user\_id*** (opcional) y que se pueda paginar con los parámetros de ***page*** y ***size*** (obligatorios).

**GET /loans?page=1&size=50**

{

"items":[

{

“id”: 1,

“total”: 2500.00

“userId”: 1

},

{

“id”: 2,

“total”: 65120.75

“userId”: 1

}

],

"paging":{

"page": 1,

"size": 50,

"total": 1500

}

}

**GET /loans?page=1&size=50&*user\_id=2*** (La respuesta es igual a la anterior pero filtrando por user\_id).

La carga inicial de datos en la base de datos también forma parte de la implementación y la entrega.

Se evaluará el diseño de la solución, las relaciones entre las entidades, el manejo de errores, el uso de logs, claridad de código, y obviamente que el sistema haga lo que tiene que hacer.

Consideraciones para el desarrollo:

* Java
* SpringBoot (MVC y JPA)
* ORM
* Base de Datos SQL (MySQL, Postgres, H2, etc.).
* Repositorio Git para la entrega del código (se recomienda Bitbucket o Github)
* Tests
* README explicando como levantar la app

Postman Collection para asistir en las pruebas:

<https://www.getpostman.com/collections/d63a32fdc2a1205d05a5>