

# Java back-end developer test

El objetivo de esta prueba es ayudarnos a valorar las capacidades de análisis además de la calidad y estilo de código.

Los puntos que más se valoraran son los siguientes:

- La arquitectura de la aplicación.
- El diseño del modelo.
- La implementación de la lógica de negocio.
- La calidad de los tests.
- El performance.

## Descripción de la prueba

Para la realización de la prueba se facilitará un repositorio de Git donde se deberán hacer los commits y desde ahí se revisará.

La prueba empezará a ser revisada en un plazo de 10 días a partir del envío de la misma. Pasado ese plazo, se puede seguir trabajando sobre la prueba por lo que no es una fecha límite.

Se valorarán los puntos listados anteriormente, por lo que es mejor centrarse en la calidad que en terminar el 100% de los requisitos técnicos y funcionales, obviando, si fuese necesaria, la persistencia de los datos.

## Requisitos técnicos

Implementar una API REST que exponga la lógica de negocio descrita más adelante.

Se recomienda el uso de Java 8 para la implementación de esta prueba y se anima a hacer uso de sus APIs siempre que sean de utilidad.

No es necesaria la implementación de la persistencia aunque sí que debe tener presencia en el código. Los objetos pueden almacenarse en memoria, instancias estáticas o la implementación que se crea oportuna.

La lógica de negocio deberá ser testeada unitariamente, siendo estos el único tipo de tests requeridos en esta prueba.

# Requisitos funcionales

## Modelo

El **alumno** al realizar una **actividad**, la aplicación le calcula la nota y la envía al servidor, esta nota se considera un **intento**. Un alumno puede hacer un número ilimitado de intentos sobre la misma actividad.

Se deberá implementar un servicio que reciba estos intentos y calcule la **media aritmética** de la actividad.

Tanto la nota de los intentos como la media deberá ser sobre 100 sin decimales.

Este servicio, se tiene que diseñar totalmente desacoplado a otros servicios, como los de usuario o los del libro, por lo que no se deberá modelar ni la estructura del libro ni el usuario. En este sistema solo tendremos referencias a estos objetos mediante su id, en este caso su [UUID](#).

## Funcional a implementar

En la API se tienen que implementar dos endpoints para este funcional.

- Añadir intento.  
En este endpoint se tiene que enviar la nota del intento de un usuario para una actividad. La respuesta de este endpoint será la nota media de la actividad, ya que por requisitos funcionales esta nota debe ser mostrada al usuario como feedback instantáneo.
- Leer medias.  
Dada un UUID de una actividad, apartado o tema. Se tiene que devolver la nota media de ese elemento.