



# Prueba de desempeño

## Ruta avanzada: Java.

**Team Leader:** Antony Martinez Ruiz.

**Clan:** Jeff Bezos.

**Fecha:** 05/08/2024.

**Hora de Inicio:** 02:00 pm.

**Hora de finalización:** 10:00 pm.

### Reglas del espacio de prueba:

- **Comunicación:** Está prohibido hablar o comunicarse de cualquier forma con otros coders durante la prueba.
- **Integridad Académica:** Cualquier forma de fraude, incluido el plagio, copia o uso de material no autorizado, resultará en una calificación de **cero** en la prueba y puede llevar a sanciones adicionales según las políticas de RIWI.
- **Material Permitido:** Solo se permite ver material de apoyo como lo son diapositivas o ejercicios realizados durante el entrenamiento. Si se requiere ver documentación oficial, se debe avisar al team leader y hacerlo bajo su respectiva supervisión.
- **Permanencia en el Aula:** Una vez iniciada la prueba, no se permite salir del aula hasta haber entregado el examen y, de preferencia, hasta que haya transcurrido al menos la mitad del tiempo asignado, con excepción de idas al baño y los respectivos descansos.
- **Recesos:** Se darán 3 descansos de 20 minutos. Como una medida para evitar el **'burnout'** y mantener un buen rendimiento durante el transcurso de la prueba (4:00 - 4:20 pm / 6:00 - 6:20 pm / 8:00 - 8:20 pm).



### Instrucciones Generales:

- La prueba debe estar desarrollada en Java con los paquetes JDBC, SWING (JOptionPane) y una base de datos en MySQL(local o en línea) aplicando el encapetado explicado en las sesiones de entrenamiento (MVC).
- Se debe subir el código a un repositorio de GitHub público. Tenga en cuenta que se revisará el último commit realizado antes de la hora y fecha límite (Los commits posteriores a eso, no serán tomados en cuenta).
- Se debe añadir el script SQL de las creaciones de las tablas y sentencias realizadas durante la prueba dentro de la carpeta "persistence" o "model" (Exportar la base de datos).
- Una vez finalizada la prueba se debe cargar el .zip de la aplicación y .txt que contenga el link del repositorio a la plataforma Moodle en el módulo **R. avanz: Java & Spring Boot** del curso de **desarrollo** dentro de la tarea llamada **Entrega prueba de desempeño**.
- La prueba deberá ser sustentada presencialmente el día **08 de agosto del 2024** de manera individual y el orden será aleatorio con previa cita.

### Recuerda:

*"El éxito es la suma de pequeños esfuerzos repetidos día tras día."*

**- Robert Collier**

**¡Éxitos!**



## Riwi Academy

**Épica:** Sistema de gestión de cursos en línea.

### Descripción:

Riwi está desarrollando una aplicación de cursos en línea llamada "Riwi Academy". La aplicación debe tener un sistema para almacenar información sobre estudiantes, cursos, inscripciones y calificaciones. El objetivo es diseñar y desarrollar una base de datos relacional y crear una aplicación para interactuar con la base de datos (crear, editar, eliminar, buscar y realizar consultas avanzadas).

### Notas adicionales:

1. Se recomienda leer todas las historias de usuario antes de empezar a realizar el modelo entidad-relación para se tenga claridad sobre todos los datos requeridos.

### Historia de Usuario - 01:

**Título:** Creación de la Base de Datos

**Como** administrador,

**Quiero** crear una base de datos llamada "RiwiAcademyDB",

**Para** que pueda almacenar toda la información relevante sobre estudiantes, cursos, inscripciones y calificaciones.

### Criterios de Aceptación:

1. Implementar la restricción ON DELETE CASCADE en las claves foráneas adecuadas.
2. Implementar validaciones requeridas como email único por cada estudiante y los campos que no puedan ser nulos.



3. La base de datos es en MySQL, puede ser local o remota (añadir en un .txt las credenciales).

### Historia de Usuario - 02:

**Título:** Sistema de gestión de estudiantes

**Como** administrador,

**Quiero** un sistema de gestión de estudiantes,

**Para** que pueda crear, editar, ver la información sobre los estudiantes guardada en la base de datos.

### Criterios de Aceptación:

1. Se debe poder crear estudiantes nuevos, en donde el estado por defecto inicialmente será "Activo=true".
2. Se debe poder editar la información de los estudiantes, ya sea parcial o total.
3. Se debe poder listar a todos los estudiantes activos.
4. Se debe poder buscar por ID o por correo electrónico con toda la información, incluyendo los cursos en donde se encuentra inscrito.

### Historia de Usuario - 03:

**Título:** Sistema de gestión de cursos

**Como** administrador,

**Quiero** un sistema de gestión de cursos,

**Para** que se pueda crear, editar, buscar y eliminar cursos almacenados en la base de datos.



### Criterios de Aceptación:

1. Se debe poder crear cursos nuevos en donde se garantice que los nombres de cada curso son únicos, ignorando mayúsculas y minúsculas.
2. Se debe poder eliminar cursos, solo si no tiene ningún estudiante inscrito.
3. Se debe poder listar todos los cursos.

### Historia de Usuario - 04:

**Título:** Sistema de gestión de inscripciones

**Como** administrador,

**Quiero** un sistema de gestión de estudiantes,

**Para** que pueda crear, editar, ver la información sobre los estudiantes guardada en la base de datos.

### Criterios de Aceptación:

1. Se debe poder generar una inscripción siempre y cuando exista el curso y el estudiante, verificar que el estudiante no se encuentre inscrito previamente.
2. Se debe poder eliminar dándole una advertencia de que perderá el historial de calificaciones del estudiante en el curso correspondiente.
3. Cada estudiante solo puede tener un máximo de 3 inscripciones.

### Historia de Usuario - 05:

**Título:** Sistema de gestión de calificaciones

**Como** administrador,

**Quiero** un sistema de gestión de estudiantes,



**Para** que pueda crear, editar, ver la información sobre los estudiantes guardada en la base de datos.

### Criterios de Aceptación:

1. Se debe poder crear una calificación asociada a un curso con una breve descripción, en donde se describa el tipo de nota tomada.
2. Se debe poder editar una calificación tomada.
3. El rango de calificaciones será entre 0 y 100.

### ENTREGABLES:

1. Se debe montar la solución a un repositorio público en Github con una única rama llamada "main".
2. Los commits realizados deben ser coherentes con los cambios realizados, se recomienda realizarlos frecuentemente y preferiblemente en Inglés (Solo se tendrán en cuenta los commits realizados hasta las 10pm).
3. El repositorio en GitHub deberá tener un readme.md en donde pongas el Modelo entidad-relación en el formato que prefieras y el link del repositorio.
4. Solo debe cargar en moodle el repositorio en formato zip y con el siguiente nombre **tu\_nombre\_PruebaJava.zip** (<https://moodle.rwi.io/mod/assign/view.php?id=1099>).
5. Dentro del .zip la entrega deberá tener el script de .sql de la creación de la base de datos.

### CRITERIOS DE EVALUACIÓN:

En el siguiente link encontrarás la rúbrica de evaluación para la prueba realizada. Esta rúbrica detalla los criterios, porcentaje de peso y escala de desempeño que se utilizarán para calificar tu prueba. Revisa la rúbrica para entender cómo se evaluarán tus respuestas y para asegurarte de que cumples con todos los requisitos necesarios:

<https://moodle.rwi.io/mod/book/view.php?id=1129&chapterid=57>