



(v0.0.1 - 28-05-2024)

Team Lead: Stiven Loaiza

# Prueba de desempeño NestJS

## Proposito:

Como equipo de formación Riwi, queremos evaluar las habilidades de desarrollo backend en NestJS de un aspirante a Junior developer, específicamente en el diseño e implementación de APIs RESTful, manejo de bases de datos relacionales, y aplicación de buenas prácticas de NestJS. Esto nos permitirá determinar si el candidato posee los conocimientos técnicos y la capacidad de resolución de problemas necesarios para contribuir efectivamente en proyectos reales y en entornos laborales.

## Metodología:

La prueba técnica se estructurará como una historia de usuario Scrum con una duración estimada de 8 horas. Esta metodología ágil permite simular un entorno de trabajo realista, donde los desarrolladores reciben tareas con plazos definidos y deben gestionar su tiempo y recursos de manera eficiente.

Se permitirá el uso de documentación oficial, ejemplos de código abierto y herramientas de IA como apoyo, fomentando así la investigación y el aprendizaje autónomo. Sin embargo, se enfatizará la importancia de comprender y explicar el código generado, evitando el plagio y la dependencia excesiva de herramientas externas.



[www.riwi.io](http://www.riwi.io)



301 732 53 27



Cl. 16 # 55 - 129



La evaluación se centrará en la capacidad del candidato para aplicar los conocimientos teóricos en la práctica, resolver problemas de manera creativa y demostrar un dominio sólido de las tecnologías y conceptos fundamentales de NestJS.

Esta metodología no solo evalúa las habilidades técnicas del candidato, sino también su capacidad de adaptación, gestión del tiempo y aprendizaje autónomo, cualidades esenciales para un Junior developer exitoso.

## Rubricas de evaluación:

x

## Historia de usuario a entregar al Coder (Enunciado):

¡Bienvenido Coder a tu prueba de desempeño de framework (NestJS)!

- **Las reglas:**
  - Tienes 8 horas máximo para la culminación de la prueba. (6am a 2pm) con 3 descansos (8:00 am a 8:20 am, 10:00 am a 10:20 am y 12:00 m a 12:20 m)
  - Puedes usar documentación oficial, ejemplos de código abierto y herramientas de IA como apoyo, evitando el plagio.
  - Debes tener la capacidad de explicar tu código.
  - Debes realizar la prueba de forma individual.
  - Se deben cumplir los criterios de aceptación de la HU.
- **Las metodologías:**



[www.riwi.io](http://www.riwi.io)



301 732 53 27



Cl. 16 # 55 - 129



- Trabajarás en una historia de usuario similar a un entorno de trabajo real.
- Demuestra tus conocimientos técnicos y tu capacidad para resolver problemas.
- ¡Buena suerte y manos a la obra!

- **Historia de usuario:**

**Empresa:** Torneos Van Rossum

**Contexto:** Empresa dedicada a la gestión de torneos de videojuegos(esports) a nivel Colombia, tiene la necesidad de implementar una API para la gestión de torneos.

**Tarea asignada:**

Como desarrollador junior de una plataforma de gestión de torneos de esports, quiero implementar las funcionalidades crear, actualizar, eliminar y consultar torneos, jugadores y resultados, para poder administrar eficientemente los eventos y mantener la información actualizada. Esta API me debe proporcionar las capacidades para poder registrar, consultar y actualizar un jugador, crear un torneo, actualizarlo, eliminarlo, agregar participantes a ese torneo y consultar el torneo con sus participantes, también me debe permitir a través de un endpoint crear al azar la asignación de la competencia de un torneo entre los jugadores inscritos, luego también me debe permitir registrar los resultados de dicha competencia con los datos de ganador y perdedor y el puntaje obtenido por cada competidor, debo poder consultar los resultados de los jugadores



[www.riwi.io](http://www.riwi.io)



301 732 53 27



Cl. 16 # 55 - 129



por torneo con mínimo un filtro de puntajes ( $\geq$ ), ordenamiento por puntajes y paginación.

**Tiempo:** 8 horas

### **Criterios de Aceptación:**

- \*Persistencia de Datos:
  - Utilizar TypeORM con PostgreSQL o MySQL para la persistencia de datos.
  - Modelar y crear las tablas requeridas para soportar los torneos, jugadores y resultados.
  - Plus si se implementan scripts de poblado de datos (seeds, helpers o migrations)
  - Para los casos de eliminación utilizar softdelete.
- Manejo de Rutas (Routing)
  - Implementar rutas según RESTful API para las operaciones CRUD de torneos, jugadores y resultados.
  - Asegurar que las rutas sean intuitivas y sigan las convenciones REST.
- \*Data y DTO
  - Manejar correctamente los params, query params, body y headers.
  - Utilizar DTOs para la validación de datos en las rutas de creación y actualización.



[www.riwi.io](http://www.riwi.io)



301 732 53 27



Cl. 16 # 55 - 129



- Códigos de Respuesta:  
Devolver los códigos de respuesta adecuados para cada operación (201, 200, 400, 401, 404, 500, etc.).
- \*Documentación:
  - Incluir un archivo README que explique cómo configurar y ejecutar el proyecto.
  - Utilizar Swagger para documentar la API generada.
- Gitflow:
  - Utilizar Gitflow para la gestión de ramas y versiones del proyecto.
  - Asegurarse de que los commits sean descriptivos y las ramas sigan la convención de Gitflow.
- Paginación, Filtros y Ordenamiento:
  - Implementar paginación en las consultas de torneos, jugadores y resultados.
  - Incluir al menos un filtro básico en la consulta de jugadores por torneo. Plus si se aplican filtros en otras consultas o filtros más avanzados.
  - Incluir al menos un ordenamiento en la consulta de jugadores por torneo. Plus si se aplica ordenamiento en otras consultas.
- Validación de Peticiones:
  - Implementar un x-api-key en los headers para validar que la petición de crear torneo sea confiable.



www.rivi.io



301 732 53 27



Cl. 16 # 55 - 129



- Uso de Herramientas del Framework:
  - Implementar guards para asegurar las rutas que requieran protección.
  - Implementar validaciones y/o pipes para asegurar la integridad de los datos.

### Entregables:

- Código fuente del proyecto en un repositorio GitHub. Adjuntar url del repositorio y el código en ZIP en Moodle. (solo se revisan ramas con commits que máximo tengan la fecha y hora de finalización de la prueba).

### Notas Adicionales:

Se valorará la claridad y organización del código, así como la correcta implementación de las herramientas y prácticas solicitadas.



[www.riwi.io](http://www.riwi.io)



301 732 53 27



Cl. 16 # 55 - 129