Introducción

Durante el desarrollo de la prueba tecnica, se utilizaron herramientas de Inteligencia Artificial para acelerar y optimizar diferentes etapas del proceso, tanto en el backend (Spring Boot) como en el frontend (Angular). La IA fue clave en tareas como:

- Generación automática de código repetitivo (DTOs, controladores, servicios).
- Sugerencias de buenas prácticas y patrones de diseño.
- Refactorización y mejora de consultas y lógica de negocio.
- Creación de documentación y ejemplos de uso

Optimizaciones logradas con IA

- **Reducción de código repetitivo:** Se generaron plantillas base para controladores, servicios y repositorios, lo que permitió concentrarse en la lógica de negocio.
- **Validaciones automáticas:** Se recibieron sugerencias para implementar validaciones robustas y un manejo de excepciones adecuado.
- **Mejoras en seguridad:** Se aplicaron recomendaciones para la configuración de JWT y filtros de autenticación.
- **Eficiencia en el frontend:** Se generaron componentes y servicios en Angular que cumplen con buenas prácticas de modularidad y reutilización.
- **Documentación clara y precisa:** Se generaron descripciones automáticas para endpoints y modelos, facilitando la integración y el mantenimiento.

Ejemplos específicos de código generado

Backend (Spring Boot)

DTO y Controlador de Autenticación:

```
// AuthRequest.java (generado con ayuda de IA)
public class AuthRequest {
    private String email;
```

```
    private String password;
    // getters y setters
}

// AuthController.java (fragmento generado con ayuda de IA)
@PostMapping("/login")
public ResponseEntity<AuthResponse> login(@RequestBody AuthRequest request)
{
    AuthResponse response = authService.authenticate(request);
    return ResponseEntity.ok(response);
}
```

Manejo de Excepciones:

```
// GlobalExceptionHandler.java (sugerido por IA)
@ExceptionHandler(ResourceNotFoundException.class)
public ResponseEntity<ErrorResponse>
handleNotFound(ResourceNotFoundException ex) {
    ErrorResponse error = new ErrorResponse("NOT_FOUND", ex.getMessage());
    return new ResponseEntity<>(error, HttpStatus.NOT_FOUND);
}
```

Frontend (Angular)

Servicio para tareas:

```
// task.service.ts (generado con ayuda de IA)
getTasks(): Observable<Task[]> {
  return this.http.get<Task[]>(`${this.apiUrl}/tasks`);
}
```

Componente de lista de tareas:

```
<!-- task-list.component.html (estructura sugerida por IA) -->
<ul>
  <li *ngFor="let task of tasks">{{ task.title }} - {{ task.status }}</li>
</ul>
```

Tiempo estimado ahorrado

- **Generación de código repetitivo:** Ahorro estimado del 40% aproximadamente frente al desarrollo manual.

- **Refactorización y optimización:** Ahorro del 20 al 30% en revisiones y mejoras.

- **Documentación técnica:** Reducción del 50% del tiempo requerido para escribir y estructurar documentación.