**INSTITUTO DE EDUCACIÓN SUPERIOR CIBERTEC**

**DIRECCIÓN ACADÉMICA**

**CARRERAS PROFESIONALES**

**CURSO** : Desarrollo de Aplicaciones Web II

**NOTA**

**PROFESOR** : Victor Fernández Quintana

**SEMESTRE** : 2025 - 3

**CICLO** : Sexto

**SECCION** : T6BN

**GRUPO** : 01

**FECHA** : 15/10/2025

|  |
| --- |
| **ALUMNO (A) :** |

**PRIMERA EVALUACIÓN DE LABORATORIO**

**Consideraciones generales:**

* Se considerará el orden, la claridad de las respuestas y las buenas prácticas.
* Las preguntas deben resolverse de acuerdo con los conceptos discutidos en clase.
* Para el desarrollo del proyecto cree la estructura necesaria con Gradle o Maven y nombre el proyecto: EduConnect\_T2\_APELLIDO\_NOMBRE.

|  |
| --- |
| **LOGRO DE LA EVALUACION:**  Al término de la evaluación, el alumno implementa una solución de microservicios distribuida en Java y Spring Boot, incluyendo comunicación entre servicios mediante Feign Client y aplicando conocimientos con Spring Security e implementando OAuth2.0 como resource server para la validación de token firmado, aplicando principios SOLID. |

**Consolidado**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pregunta** | **Puntaje** | | **Llenar solo en caso de Recalificación justificada** | |
| **Máximo** | **Obtenido** | **Sustento** | **Puntaje** |
| **1** | **20** |  |  |  |
| **Nota Recalificada** | | | |  |

**Caso práctico:**

La empresa EduConnect, especializada en soluciones digitales para instituciones educativas,

ha identificado la necesidad de modernizar su sistema de gestión de cursos y profesores.

Actualmente, el proceso es manual y propenso a errores.

El área de TI propone una arquitectura basada en microservicios para mejorar el control y

la escalabilidad del sistema. Como desarrollador backend, debes implementar esta solución

utilizando buenas prácticas y tecnologías modernas.

**Arquitectura solicitada:**

• Microservicio de profesores (protegido con Spring Security como resource server)

• Microservicio de cursos (protegido con Spring Security como resource server)

• Microservicio auth-service (Authorization Server)

• Comunicación vía Feign Client y autorización JWT entre servicios

• auth-service como proveedor de tokens JWT para la comunicación segura entre microservicios mediante OAuth2, aplicando principios SOLID.

**Modelo de datos:**

### Microservicio: profesores-service

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descripción** |
| id | Long | Identificador único del profesor |
| nombre | String | Nombre completo del profesor |
| especialidad | String | Especialidad docente |
| estado | Enum (ACTIVO, INACTIVO) | Estado del profesor |

### Microservicio: cursos-service

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descripción** |
| id | Long | Identificador único del curso |
| nombre | String | Nombre del curso |
| creditos | Integer | Cantidad de créditos |
| profesorId | Long | ID del profesor asignado |
| estado | Enum (PENDIENTE, ACTIVO, CERRADO) | Estado del curso |

**Pregunta 01**

Implemente los microservicios solicitados utilizando Java y Spring Boot. Asegúrese de:

• Implementar el servicio de Authorization Server.

Cuando un microservicio (por ejemplo, `cursos-service`) realice una solicitud POST al endpoint `/oauth2/token` enviando su `clientId` y `clientSecret`, `auth-service` debe entregar un token firmado.

• Construcción de los microservicios presentados, estos tendrán la característica de ser resource servers, clientes registrados y manteniendo la configuración de roles mediante los scopes de cada cliente.

• Cada microservicio debe proteger los endpoints correspondientes con sus scopes personalizados, por ejemplo, cursos-service: cursos.scope y profesores-service: profesores.scope.

• Cada microservicio debe tener la configuración de Spring Security para el acceso de sus recursos.

• Se debe considerar la propagación del Token entre los microservicios.

• Utilizar Feign Client para que el microservicio de cursos consulte datos del microservicio de profesores.

• El microservicio `cursos-service` debe solicitar el token y luego enviarlo como Authorization Bearer

al `profesores-service`, el cual debe estar configurado como Resource Server y solo permitir accesos con JWT válidos.

• Probar los endpoints usando Postman.

**RÚBRICA DE LA PRIMERA EVALUACIÓN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rúbrica** | | | | | |
| **Pregunta** | **puntaje** | **20 puntos** | **13 puntos** | **5 puntos** | **0 puntos** |
| 1 | 20 puntos | Implementa correctamente el auth-service, emitiendo JWT desde /oauth2/token. El microservicio cliente consume este token vía Feign y accede correctamente al resource server protegido. | Implementa los microservicios correctamente pero no se comunican entre sí. | Implementa solo uno de los microservicios solicitados. | No existen microservicios registrados. |