



**DESARROLLO DE APLICACIONES OPEN SOURCE (SI729)**  
**EXAMEN FINAL**  
**2024-1**

**Sección:** SV54, SV55, SW51, SW53, SW54, SW57, SW58,  
WS51, WS52, WS53, WX52, WX54, WX55, WX56

**Profesores:** Bautista Ubillús, Efraín Ricardo  
Flores Moroco, Juan Antonio  
Mori Paiva, Hugo Allan  
Navarrete Vilca, Elio Jefferrson,  
Sánchez Seña, Alberto Wilmer  
Velásquez Núñez, Ángel Augusto

**Duración:** 170 minutos

**Indicaciones:**

1. El examen consta de 1 pregunta, y tendrá **170 minutos** para resolverlas.
2. La pregunta es de tipo Proyecto de Software y la entrega de su respuesta es a través de envío de archivo empaquetado **.zip** (único formato válido) con nombre *upc-pre-202401-si729-<sección>-eb-u<código-estudiante>.zip*, conteniendo el proyecto de software, en la Actividad para el Examen final.
3. Puede utilizar como referencia los materiales publicados en el aula virtual, los sitios referenciados en el enunciado, así como sitios web de documentación de frameworks o tecnologías referenciadas.
4. Cada examen cuenta con un equipo académico, el cual estará conectado **durante el examen**.
5. El alumno debe dedicar los primeros 15 minutos a revisar las preguntas del examen y de presentarse alguna duda enviar un correo al(los) profesor(es)

Secciones SV54, SW58, WS51, WS52, WX54: Delgado Vite, Jorge Luis al correo pcsjdev@upc.edu.pe

Secciones SW51, SW53, SW54, WX55, WX56: Rojas Malasquez, Royer Edelwer al correo pcisroj@upc.edu.pe

Secciones SV55, SW57, WS53, WX52: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe

6. Los profesores en mención, solo recibirán correos provenientes de las cuentas **UPC**, de ninguna manera se recibirán correos de cuentas públicas.
  7. Ante problemas técnicos, debe de forma obligatoria adjuntar evidencias del mismo, como capturas de pantalla, videos, fotos, etc. Siendo requisito fundamental que, en cada evidencia se pueda apreciar claramente la fecha y hora del sistema operativo del computador donde el alumno está rindiendo el examen.
  8. No se recibirán problemas técnicos una vez culminado el examen.
-

## Enunciado:

### Caso Hartford Institute for Geriatric Nursing.

Desde su inicio en 1996, el objetivo singular del Hartford Institute for Geriatric Nursing o HIGN (<https://hign.org/>) ha sido moldear la calidad de la atención médica de los adultos mayores. El compromiso con este objetivo, exhibido por el dedicado liderazgo, el personal y las organizaciones afiliadas de HIGN, ha hecho de HIGN una presencia geriátrica reconocida mundialmente. HIGN es la rama geriátrica de la Facultad de Enfermería Rory Meyers de la Universidad de Nueva York y se ha convertido, a lo largo de los años, en un faro para todos aquellos que desean avanzar en la atención geriátrica a través del liderazgo de la enfermería y la atención en equipo interprofesional.

El deterioro cognitivo ya no se considera un cambio normal e inevitable del envejecimiento. Aunque los adultos mayores corren mayor riesgo que el resto de la población, los cambios en la función cognitiva a menudo exigen una acción rápida y agresiva. En los pacientes mayores, es especialmente probable que el funcionamiento cognitivo disminuya durante una enfermedad o lesión. La evaluación que realizan las enfermeras del estado cognitivo de un adulto mayor es fundamental para identificar cambios tempranos en el estado fisiológico, la capacidad de aprender y evaluar las respuestas al tratamiento.

El Mini Mental State Examination (Mini Examen del Estado Mental) o MMSE, es una herramienta que se puede utilizar para evaluar de forma sistemática y exhaustiva el estado mental. Es una medida de 11 preguntas que evalúa cinco áreas de *cognitive function* (función cognitiva): *orientation* (orientación), *registration* (registro), *attention and calculation* (atención y cálculo), *recall* (recuerdo) y *language* (lenguaje). La puntuación máxima es 30. Una puntuación de 23 o menos es indicativa de *cognitive impairment* (deterioro cognitivo). El MMSE tarda sólo entre 5 y 10 minutos en administrarse y, por lo tanto, es práctico de usar de forma repetida y rutinaria.

El MMSE es eficaz como herramienta de detección del deterioro cognitivo en adultos mayores, que viven en comunidades, hospitalizados e institucionalizados. La mejor manera de lograr la evaluación de la función cognitiva de un adulto mayor es cuando se realiza de forma rutinaria, sistemática y exhaustiva.

## Pregunta 1 (20 p.).

Usted se integra al backend software developer team, a cargo de la creación de un **RESTful API** que brinde soporte a las operaciones de HIGN. El ecosistema de HIGN Platform requiere que el HIGN Medical Exams Management Web Application cuente con **Endpoints** en el RESTful API, para el manejo de la información de: *Examiners*, conformadas por los atributos `id`, `firstName`, `lastName` y `nationalProviderIdentifier`. *Mental State Exams*, conformadas por `id`, `patientId`, `examinerNationalProviderIdentifier`, `examDate`, `orientationScore`, `registrationScore`, `attentionAndCalculationScore`, `recallScore` y `languageScore`.

Como reglas de negocio, HIGN:

- Establece que la información que se desea preservar de los *Examiners*, incluye **id** (Long, Primary Key, Autogenerado), **firstName** (String, Obligatorio, No vacío), **lastName** (String, Obligatorio, No vacío), **nationalProviderIdentifier** (identificador del negocio, Obligatorio, No vacío, solo puede contener un valor UUID válido).
- Establece que la información de los *Mental State Exams*, incluye **id** (Long, Primary Key, Autogenerado), **patientId** (Long, Obligatorio, No vacío), **examinerNationalProviderIdentifier** (embedded type Obligatorio, No vacío), **examDate** (Date, Obligatorio, no vacío), **orientationScore** (Integer, Obligatorio), **registrationScore** (Integer, Obligatorio), **attentionAndCalculationScore** (Integer, Obligatorio), **recallScore** (Integer, Obligatorio), **languageScore** (Integer, Obligatorio).
- Especifica que el atributo **nationalProviderIdentifier**<sup>1</sup> debe ser embedded y solo puede contener un valor UUID válido, que debe proporcionarse (no autogenerarse) al momento de registrarse.
- Especifica que **nationalProviderIdentifier** se considera un identificador interno del negocio. No se debe registrar en la base de datos dos *examiners* con el mismo valor de **nationalProviderIdentifier**.
- Especifica que el atributo **examinerNationalProviderIdentifier** debe ser embedded y el valor UUID contenido debe corresponder con el valor de *nationalProviderIdentifier* para un *examiner* previamente registrado.
- Requiere que el valor de **examDate** no sea mayor que la fecha actual.
- Requiere que los valores válidos para **orientationScore** solo sean enteros entre 0 y 10.
- Requiere que los valores válidos para **registrationScore** solo sean enteros entre 0 y 3.
- Requiere que los valores válidos para **attentionAndCalculationScore** solo sean enteros entre 0 y 5.
- Requiere que los valores válidos para **recallScore** solo sean enteros entre 0 y 3.
- Requiere que los valores válidos para **languageScore** solo sean enteros entre 0 y 9.
- Especifica que un *Examiner* puede evaluar varios mental state exams, pero un mental state exam en particular, solo puede ser evaluado por un *examiner*.
- Especifica que tanto *Examiner* como *Mental State Exam* pertenecen a diferentes bounded contexts, por lo que se necesita un *Anti-Corruption Layer (ACL)* para que el bounded context que lo requiera, acceda a capacidades que exponga el otro bounded context como parte de la implementación de una característica requerida.
- Especifica que tanto *Examiner* como *Mental State Exam* son Aggregate Roots en sus respectivos bounded contexts, por lo que requiere contar con atributos de auditoría para registrar **createdAt** (fecha y hora de creación de registro) y **updatedAt** (fecha y hora de última actualización de registro), de uso interno y poblados de forma automática por el sistema al momento de las operaciones de creación o actualización.

Durante la etapa de desarrollo, le asignan trabajar en específico sobre dos Endpoints:

/api/v1/examiners  
/api/v1/mental-state-exams

Incluya como parte del desarrollo la implementación de todas las reglas de negocio.

---

<sup>1</sup> **National Provider Identifier** es un identificador único, que la entidad gubernamental *US Centers for Medicare & Medicaid Services* genera para los proveedores de servicios de cuidado de la salud, sirviendo para identificarlos a nivel nacional.

### Examiners Endpoint ( /api/v1/examiners )

Debe implementar **solo una** operación en el RESTful API: agregar (POST) un **examiner**. Los valores de **id** no se solicita, son autogenerados al momento de almacenar la información. Los valores de **nationalProviderIdentifier** que se proporciona como String deben poder convertirse a valores válidos de tipo UUID versión 4<sup>2</sup>. Al agregar se debe retornar en el response el status 201 (created). Incluya en el response un objeto con el **id** generado para el **examiner** y sus demás atributos, incluyendo el **nationalProviderIdentifier** como String. No incluya en el response los atributos de auditoría.

### Mental State Exams Endpoint ( /api/v1/mental-state-exams )

Debe implementar **solo una** operación sobre los **mental state exams** en el RESTful API: agregar (POST) un **mental state exam**. Los valores de **id** no se solicita, son autogenerados al momento de almacenar la información. El valor de **examDate** debe recibirse como String con el formato YYYY-MM-DD (Por ejemplo "2024-04-13"). El valor de **examinerNationalProviderIdentifier** debe aceptarse como String. Al agregar se debe retornar en el response el status 201 (created). Al agregar se debe retornar en el response el objeto para el mental state exam, con el **id** generado para el mismo y sus demás atributos, incluyendo el **examinerNationalProviderIdentifier** como String. No incluya en el response los atributos de auditoría.

### Technical constraints

1. Elabore la solución con Java 22 y Spring Boot Framework 3.X.
2. Cree su proyecto de software con el nombre **si729ebu<código-estudiante>** (por ejemplo, **si729ebu201621873**).
3. La información debe ser persistente en una base de datos relacional (MySQL), en un esquema **hign**.
4. Los packages de su solución deben tener como nombre raíz **org.hign.platform.u<código-estudiante>** (por ejemplo, **org.hign.platform.u201621873**).
5. Considere que el concepto **Examiner** pertenece al bounded context **personnel** y el concepto **Mental State Exam** pertenece al bounded context **assessment**.
6. Considere el bounded context **shared** para elementos base comunes/reutilizables que puedan ser aprovechados o extendidos por elementos en otros bounded contexts.
7. Aplique buenas prácticas de Arquitectura de Software, enfoque de **Domain-Driven Design**, separación en bounded contexts, **layered architecture** (domain, application, interfaces, infrastructure), patrones de strategic y tactical Domain-Driven Design, patrón CQRS, patrón Anti-Corruption Layer (ACL), principios y patrones de diseño de software orientado a objetos, convenciones de nomenclatura en **inglés**, así como buenas prácticas de nomenclatura en Java (entre ellas Upper-Camel-Case para Clases, Lower-Camel-Case para atributos y métodos) y buenas prácticas para nomenclatura de objetos de Base de Datos (entre ellas snake case, tablas en plural, sin mnemónicos).
8. Utilice minúsculas para los nombres de URL y términos compuestos separados por guión medio (-) para todos los endpoints.
9. Utilice la biblioteca Lombok para el manejo de métodos constructores y de acceso en las clases de Java.
10. Utilice records en vez de clases para almacenamiento de valores inmutables.
11. Para **examiner** y **mental state exam**, incluya atributos de auditoría createdAt y updatedAt con valores poblados de forma automática por Spring Boot al momento de la creación.
12. Utilice el patrón Assembler para el Object Mapping la sección **transform** en interfaces layer.
13. Documente su código con **JavaDoc** (ver referencias), colocando en inglés información de propósito para principales objetos de programación, así como propósito, parámetros y valor retornado en clases y métodos relevantes. Incluya como parte de la documentación sus nombres y apellidos como valor para @author.
14. Incluya documentación de Endpoints con **OpenAPI**.
15. Considere la gestión de excepciones en la aplicación.
16. Empaquete su solución como un archivo **.zip**. (**único formato válido**) con el nombre **upc-pre-202401-si729-<sección>-eb-u<código-estudiante>.zip** (por ejemplo, **upc-pre-202401-si729-sw51-eb-u201621873.zip**).
17. Suba su archivo de solución en la Actividad indicada para el Examen final.

### NO forma parte del alcance del proyecto:

1. Soporte de CORS.
2. Security.
3. Testing.

---

<sup>2</sup> **Nota:** Para obtener valores UUID versión 4 válidos, puede utilizar la herramienta en línea **Online UUID Generator** (ver sección de referencias).

## Rúbrica de calificación

Criterio de Calificación	Sobresaliente (S)	Esperado (E)	Necesita Mejorar (M)	Insuficiente (I)	Calificación
<b>C01. Building y ejecución</b>	Al abrir el proyecto y ordenar la ejecución, ésta se inicia sin problemas. El API es accesible en la ruta indicada.	La aplicación no llega a iniciar y ejecutarse, sin embargo el proceso de building llega a concluir.	Al cargar el proyecto el proceso de building presenta errores y no llega a concluir.	No elabora solución.	
	<b>2.0 puntos</b>	<b>1.0 punto</b>	<b>0.5</b>	<b>0 puntos</b>	
<b>C02. Examiners Endpoint</b>	El RESTful API expone el endpoint especificado en el enunciado. Se evidencia la funcionalidad de las operaciones solicitadas, proporcionando en cada caso los valores esperados y respondiendo adecuadamente ante las excepciones. Se evidencia la persistencia de los objetos solicitados, cumpliendo la estructura según enunciado, en una tabla nombrada según especificaciones, en base de datos relacional según enunciado en el esquema indicado. Se incluye documentación del Endpoint con OpenAPI.	El RESTful API expone el endpoint especificado en el enunciado. Se evidencia parcialmente la funcionalidad de las operaciones solicitadas, proporcionando en algunos casos los valores esperados y respondiendo de forma parcialmente adecuada ante las excepciones, o se evidencia parcialmente la persistencia en base de datos relacional, o se evidencia parcialmente la persistencia de los objetos solicitados con estructura según enunciado, en una tabla <i>nombrada según especificaciones</i> en base de datos relacional según enunciado en el esquema indicado.	La aplicación implementa y expone el endpoint especificado en el enunciado, pero no cumple con la ruta especificada o no se puede registrar elementos.	La aplicación no implementa o expone el endpoint solicitado.	
	<b>4.0 puntos</b>	<b>3.0 puntos</b>	<b>1.5 puntos</b>	<b>0 puntos</b>	
<b>C03. Mental State Exams Endpoint</b>	El RESTful API expone el endpoint especificado en el enunciado. Se evidencia la funcionalidad de las operaciones solicitadas, proporcionando en cada caso los valores esperados y respondiendo adecuadamente ante las excepciones. Se evidencia la persistencia de los objetos solicitados, cumpliendo la estructura según enunciado, en una tabla nombrada según especificaciones, en base de datos relacional según enunciado en el esquema indicado. Se incluye documentación del Endpoint con OpenAPI.	El RESTful API expone el endpoint especificado en el enunciado. Se evidencia parcialmente la funcionalidad de las operaciones solicitadas, proporcionando en algunos casos los valores esperados y respondiendo de forma parcialmente adecuada ante las excepciones, o se evidencia parcialmente la persistencia en base de datos relacional, o se evidencia parcialmente la persistencia de los objetos solicitados con estructura según enunciado, en una tabla <i>nombrada según especificaciones</i> en base de datos relacional según enunciado en el esquema indicado.	La aplicación implementa y expone el endpoint especificado en el enunciado, pero no cumple con la ruta especificada o no se puede registrar elementos.	La aplicación no implementa o expone el endpoint solicitado.	
	<b>4.0 puntos</b>	<b>3.0 puntos</b>	<b>1.5 puntos</b>	<b>0 puntos</b>	
<b>C04. Business Rules</b>	El desarrollo incluye la implementación de reglas de negocio, cubriendo de forma completa las condiciones y escenarios establecidos, siendo éstas ejecutables, con adecuado manejo de excepciones, implementando éstas en las capas más adecuadas, aplicando convenciones y buenas prácticas.	El desarrollo incluye la implementación de la mayoría de reglas de negocio, cubriendo de forma parcial las condiciones y escenarios establecidos, siendo éstas ejecutables, implementándolas en las capas adecuadas en la mayoría de casos, ó aplicando parcialmente convenciones y buenas prácticas.	El desarrollo incluye la implementación de algunas de las reglas de negocio, incumpliendo la mayoría de condiciones y escenarios establecidos, siendo éstas ejecutables, implementándolas en las capas adecuadas en muy pocos casos, ó con poca evidencia de aplicar convenciones y buenas prácticas.	No implementa reglas de negocio, o no cubre escenarios más allá de operaciones CRUD básicas, o éstas no son ejecutables.	
	<b>4.0 puntos</b>	<b>3.0 puntos</b>	<b>1.5 puntos</b>	<b>0 puntos</b>	
<b>C05. Code Organization</b>	El desarrollador organiza el código y los elementos de backend de la solución, aplicando buenas prácticas de Java, Spring Boot Framework y Domain-Driven Design, agrupando los elementos de la solución según convenciones, manteniendo organización de paquetes y carpetas recomendadas por el fabricante y buenas prácticas de la industria de software.	El desarrollador aplica en la mayoría de casos para el backend convenciones, recomendaciones y buenas prácticas de Java, Spring Boot Framework y Domain-Driven Design.	El desarrollador aplica en algunos casos para el backend convenciones, recomendaciones y buenas prácticas de Java, Spring Boot Framework y Domain-Driven Design.	No se evidencia un criterio de organización para los elementos de la solución.	
	<b>2.0 punto</b>	<b>1.0 puntos</b>	<b>0.5</b>	<b>0 puntos</b>	
<b>C06. Code Quality</b>	Utiliza para el backend el lenguaje de programación Java. La codificación tiene un estilo claro, indentando los bloques de código según los estándares de programación correspondientes al lenguaje, aplicando una lógica consistente en los métodos, condicionales sin escenarios no contemplados, uso adecuado de reutilización de código para evitar redundancia. Aplica patrones de arquitectura y patrones de diseño. Distribuye el código en los niveles correspondientes, asignando lógica de persistencia, lógica de negocio, lógica de control, lógica de mapping y transferencia a las interfaces y clases que corresponden. Cumple de forma completa con los technical constraints.	Utiliza para el backend el lenguaje de programación Java. La codificación es funcional, aplica en la mayoría de casos los estándares de indentación de bloques de código, ó existen algunas ineficiencias en la codificación: redundancia ó inconsistencias en la lógica de programación. Aplica parcialmente patrones de arquitectura y patrones de diseño, o existe en algunas partes una distribución de la lógica en los niveles incorrectos. Cumple con la mayoría de technical constraints.	Utiliza para el backend el lenguaje de programación Java. La codificación es funcional, pero solo aplica algunos de los estándares de indentación de bloques de código, ó existen muchas ineficiencias en la codificación: redundancia ó inconsistencias en la lógica de programación. Aplica algunos patrones de arquitectura y patrones de diseño, o existe en muchos casos una distribución de la lógica en los niveles incorrectos. Cumple con solo algunos de los technical constraints.	No utiliza el lenguaje de programación Java para el backend, ó la codificación es funcional pero no se evidencia aplicación de estándares ó criterios de eficiencia en la codificación, con ausencia de comentarios, ó no aplica patrones de arquitectura ni patrones de diseño, o la codificación no es funcional.	
	<b>3.0 puntos</b>	<b>2.0 punto</b>	<b>1.0</b>	<b>0 puntos</b>	
<b>C07. Naming Standards</b>	El desarrollador aplica en todos los nombres de objetos de programación y base de datos como paquetes, componentes, interfaces, clases, objetos, variables, constantes, métodos, tablas, columnas la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, tablas, columnas, así como los recursos.	El desarrollador aplica en la mayoría de casos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador aplica en muy pocos casos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador no aplica nomenclatura en inglés para los objetos de programación ó recursos.	
	<b>1.0 puntos</b>	<b>0.5 punto</b>	<b>0.25 puntos</b>	<b>0 puntos</b>	
<b>Total</b>	<b>20 puntos</b>	<b>13.5 puntos</b>	<b>6.5 puntos</b>	<b>0 puntos</b>	

## Anexos

### Anexos A. Referencias

Comprimir y descomprimir archivos:

<https://support.microsoft.com/es-es/windows/comprimir-y-descomprimir-archivos-8d28fa72-f2f9-712f-67df-f80cf89fd4e5>

REST API Tutorial:

<https://restfulapi.net/>

Project Lombok:

<https://projectlombok.org/>

springdoc-openapi:

<https://springdoc.org/>

Spring Data JPA - Reference Documentation:

<https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#reference>

Class UUID:

<https://docs.oracle.com/en/java/javase/22/docs/api/java.base/java/util/UUID.html>

Online UUID Generator:

<https://www.uuidgenerator.net/version4>

How to Write Doc Comments for the Javadoc Tool:

<https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>

Formats for date and dateTime in JSON payloads

<https://www.geeksforgeeks.org/formats-for-date-and-datetime-in-json-payloads/>

Spring Data JPA Auditing:

<https://docs.spring.io/spring-data/jpa/reference/auditing.html#auditing.annotations>