



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

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

Profesores: Flores Moroco, Juan Antonio
Mori Paiva, Hugo Allan
Navarrete Vilca, Elio Jefferrson,
Sánchez Seña, Alberto Enrique
Velásquez Núñez, Ángel Augusto

Duración: 170 minutos

Indicaciones:

8. El examen consta de 1 pregunta, y tendrá 170 minutos para resolverlas.
8. La pregunta es de tipo Proyecto de Software y la entrega de su respuesta es a través de envío de archivo empaquetado **.zip** con proyecto de software en la Actividad Examen parcial.
8. Para el desarrollo del examen se requiere archivos de configuración que se encuentran como archivo adjunto (**upc-pre-202401-si729-examen-parcial_v1-files.zip**).
4. Cada examen cuenta con un equipo académico, el cual estará conectado durante el examen.
5. De presentarse alguna duda enviar un correo al docente de apoyo **asignado a su sección** según se indica a continuación:

SV54: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe

SV55: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe

SW51: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe

SW53: Delgado Vite, Jorge Luis al correo pcsijdev@upc.edu.pe

SW54: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe

SW57: Delgado Vite, Jorge Luis al correo pcsijdev@upc.edu.pe

SW58: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe

WS51: Delgado Vite, Jorge Luis al correo pcsijdev@upc.edu.pe

WS52: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe

WS53: Delgado Vite, Jorge Luis al correo pcsijdev@upc.edu.pe

WX52: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe

WX54: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe

WX55: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe

WX56: Delgado Vite, Jorge Luis al correo pcsijdev@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. Debe crear su proyecto de solución y evidenciar la autoría.
8. Antes de la generación del archivo **.zip** para el envío, elimine la carpeta **node_modules**.
9. Puede utilizar como referencia los materiales publicados en el aula virtual, los sitios web de documentación de frameworks, lenguaje de programación utilizados, así como ejemplos de clase (solo como referencia, no como fuente de duplicado o copia).

10. 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.
 11. Los problemas técnicos **no se recibirán una vez culminado el examen.**
-

Enunciado:

Pregunta 1 (20 p.).

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 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.

Para ello, su equipo se encuentra desarrollando para HIGN, una plataforma de backend para presentar información sobre sus evaluaciones, sin embargo, desea ir iniciando con el desarrollo del website para su catálogo de productos.

La evaluación tiene un empaquetado adjunto conteniendo dos archivos: `db.json` y `routes.json`, los cuales pueden ser usados por `json-server` (<https://github.com/typicode/json-server/tree/v0>) stable version 0.17.4 para simular un backend con las características indicadas.

Al ejecutar `json-server` en el terminal, utilizando los archivos de configuración mencionados (debe ubicar los archivos en la carpeta `server` de su proyecto), con los comandos:

```
cd server
json-server --watch db.json
```

Esto inicia el Fake API.

La información de los *Examiners* en general (*id, firstName, lastName, nationalProviderIdentifier*) se encuentra en el endpoint:

<http://localhost:3000/examiners>

La información de los *Patients* en general (*id, firstName, lastName, photoUrl, birthDate*) se encuentra en el endpoint:

<http://localhost:3000/patients>

La información de los *Mental State Exams* (*id, patientId, examinerId, examDate, orientationScore, registrationScore, attentionAndCalculationScore, recallScore, languageScore*) se encuentra en el endpoint:

<http://localhost:3000/mental-state-exams>

Para el desarrollo web de lado web frontend, se ha seleccionado TypeScript como lenguaje de programación y Angular como Frontend Framework.

Le proporcionan un conjunto de assets para utilizar en la aplicación.

Image Assets

HIGN Logo: <https://hign.org/sites/default/files/2022-01/HIGN-logo-1200.png>

Hero section image: <https://hign.org/sites/default/files/carousel-images/2022-02/iStock-945439872.jpg>

Text Assets

Hero section phrase (English): « HIGN advances equitable health care for older adults by educating the workforce and future professionals, informing policy, advancing research, and providing community programs. »

Hero section phrase (Spanish): « HIGN promueve la atención médica equitativa para los adultos mayores al educar a la fuerza laboral y a los futuros profesionales, informar las políticas, promover la investigación y brindar programas comunitarios. »

Se le encarga el desarrollo de una aplicación web que implemente las siguientes características:

- Un toolbar, donde a la izquierda se muestra el branding logo de **HIGN** y a la derecha se muestra las opciones “Home” (Inicio) y “Mental State Exams” (Evaluaciones de Salud Mental).
- El toolbar debe ofrecer además toggle buttons con “EN” y “ES” para el switching de idioma.
- La vista **Home** muestra un hero section, con la imagen indicada (*Hero section image*) y debajo la frase indicada (*Hero section phrase*).
- La vista **Mental State Exams** muestra en la parte superior el título “Mental State Exams” (Evaluaciones de Salud Mental), teniendo como contenido una colección de cards, donde en cada card se muestra la información de un exam, incluyendo *photo* (la ruta de la imagen para cada *patient photo* está disponible en *photoUrl*), *patient name*, *born date*, *exam date*, *examiner name*, *examiner national provider identifier*, total score.
- En el caso de *patient name* y *examiner name* el valor correspondiente debe ser conformado por *firstName* y *lastName* (del patient o examiner según corresponda) concatenados y separados por un espacio en blanco.
- En el caso de *total score*, debe ser calculado como la sumatoria de puntajes obtenidos por el *patient* en *orientationScore*, *registrationScore*, *attentionAndCalculationScore*, *recallScore* y *languageScore*

- Considere una vista de tipo *page-not-found* para el caso de rutas de navegación no soportadas. Dicha vista debe mostrar un mensaje incluyendo la ruta especificada que no se encontró y debe ofrecer un botón para retornar a *Home*.
- La vista **Home** es accesible desde la ruta de navegación */home*.
- La vista **Mental State Exams** es accesible desde la ruta de navegación */nursing/mental-state-exams*.
- La vista **raíz** (accesible desde la ruta de navegación */*) debe redirigir al usuario a la vista */home*.
- La interfaz de usuario debe estar disponible en inglés y español, siendo **inglés** el idioma por defecto.

El equipo de IT de su cliente tomará en cuenta no sólo el cumplimiento de las características funcionales, sino el diseño de interfaz de usuario, así como la estructura del proyecto, aplicación de convenciones de nomenclatura de objetos de programación en inglés, convenciones de nomenclatura de Angular, organización y eficiencia del código. Igualmente se tomará en cuenta la aplicación de patrones de diseño. Se toma en cuenta accessibility (uso de ARIA attributes) y usability.

Restricciones técnicas:

Nombre su proyecto como `upc2401si729eaucode` donde **code** es su código de estudiante en minúsculas.

El equipo requiere que la interfaz de usuario esté basada en **Angular Material**, mientras que para la comunicación con el backend debe apoyarse en **HttpClient** (incluido en `@angular/common/http`). La aplicación debe soportar in-app navigation y utilizar **@angular/router** para el manejo de routing en la aplicación. De requerirse, utilice **DatePipe** en los templates para los formatos de date (ver sección de referencias). Debe incluir ARIA attributes en las vistas. La interfaz de usuario debe mostrar los textos en **inglés** por defecto. Debe utilizar **@ngx-translate/core** para la implementación de i18n. El proyecto de aplicación debe poder abrirse sin problemas en JetBrains WebStorm.

La organización del proyecto debe ser domain-driven aplicando layered architecture, considerando los sub-dominios *public* (para elementos o componentes generales de la interfaz de usuario, como *home*), *shared* (para elementos base o de uso común en otros sub-dominios) y *nursing* (para componentes o elementos relacionados con mental state, como *mental state exams*). Aplique buenas prácticas para nomenclatura lógica y física de clases y componentes. Distribuya adecuadamente los elementos dentro de cada carpeta de sub-dominio, considerando carpetas como *pages*, *components*, *services*, *model*. En *pages* solo se ubican componentes que tienen relación directa con rutas de navegación. En *components* se ubican componentes que son incluidos en otros y no están asociados de forma directa a rutas de navegación. Aplique buenas prácticas para nomenclatura lógica y física de clases y componentes. Comente los archivos de código fuente en TypeScript elaborados por usted utilizando las convenciones de **JSDoc** (ver referencias), con block comments incluyendo un texto de *summary* con el propósito y *@author* con su nombre y apellido. Incluya en el archivo README.md, la información de la aplicación, descripción y su información como author.

Referencias:

Materiales del curso

<https://angular.io/guide/router>

<https://angular.io/guide/http>

<https://angular.io/guide/pipes>

<https://angular.io/api/common/DatePipe>

<https://material.angular.io/components/card/overview>

<https://material.angular.io/components/card/overview#accessibility>

<https://material.angular.io/components/toolbar/overview>

<https://material.angular.io/components/toolbar/overview#accessibility>

<https://github.com/typicode/json-server>

<https://www.w3.org/TR/wai-aria/#usage>

<https://github.com/ngx-translate/core>

<https://jsdoc.app/>

Criterio de Calificación	Sobresaliente (S)	Esperado (E)	Necesita Mejorar (M)	Insuficiente (I)	Calificación
C01. Building y ejecución	Al abrir el proyecto y ordenar la ejecución, ésta se inicia sin problemas. La aplicación 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	
	2.0 puntos	1.25 punto	0.5 puntos	0 puntos	
C02. User Interface & Home View	Se evidencia que la interfaz de usuario aplica Responsive Web Design. La interfaz cumple con las características solicitadas para la estructura, elementos de la interfaz de usuario, rutas de navegación, idioma e internacionalización. Se evidencia que la aplicación recupera y presenta en Home la información solicitada según especificaciones.	La interfaz de usuario no evidencia con claridad que aplica Responsive Web Design, o no cumple con todas las características solicitadas, sin embargo, recupera y presenta en Home la información solicitada.	La aplicación no implementa una interfaz de usuario con las características solicitadas que aplique Responsive Web Design aunque presenta parte de la información indicada.	La aplicación no presenta la información indicada.	
	5.0 puntos	3.5 puntos	1.5 puntos	0 puntos	
C03. Mental State Exams View and Operation	Se evidencia que la aplicación proporciona la opción Mental State Exams, llevando a la ruta de navegación /nursing/mental-state-exams, asociada a la vista que incluye una colección de cards con información de los exams, recuperada del backend, con el comportamiento esperado, con validaciones, control de errores de interacción o navegación, con comunicación adecuada hacia el usuario. La vista cumple con los requisitos de interfaz de usuario, ruta de navegación, con todas las características y restricciones indicadas.	Se evidencia que la aplicación proporciona la opción Mental State Exams, llevando a la ruta de navegación /nursing/mental-state-exams, asociada a la vista que incluye una colección de cards para presentar la información de exams, recuperada del backend, pero aplica de forma parcial validaciones, o controla de forma parcial los errores de interacción o navegación, o no comunica de forma adecuada al usuario, o cumple de forma parcial con los requisitos de interfaz de usuario, ruta de navegación, o cumple con parte de las características y restricciones indicadas.	Se evidencia que la aplicación proporciona la opción Mental State Exams, llevando a la vista que incluye una colección de cards, pero no cumple con la ruta de navegación, o no cumple con la mayoría de las características, o no se puede visualizar la información.	No se implementa la opción.	
	7.0 puntos	4.5 puntos	2 puntos	0 puntos	
C04. Code Organization	El desarrollador organiza el código y los elementos de frontend de la solución, aplicando buenas prácticas de TypeScript y Angular, 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, bajo un enfoque domain-driven, cumpliendo con las restricciones técnicas sobre organización.	El desarrollador aplica parcialmente en frontend convenciones, recomendaciones y buenas prácticas de TypeScript o Angular, o no cumple con todas las restricciones técnicas sobre organización.	El desarrollador aplica en frontend solo algunas de las convenciones, recomendaciones y buenas prácticas de TypeScript o Angular, o no cumple con las restricciones técnicas sobre organización.	No se evidencia un criterio de organización para los elementos de la solución, limitándose a la estructura por defecto.	
	2.0 punto	1.25 puntos	0.5 puntos	0 puntos	
C05. Code Quality	Utiliza para el frontend el lenguaje de programación TypeScript, el framework Angular y cumple con las restricciones técnicas indicadas. 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, y transferencia a las interfaces y clases que corresponden.	Utiliza para el frontend el lenguaje de programación TypeScript y el framework Angular. La codificación es funcional, pero cumple de forma parcial con las restricciones técnicas, o sólo aplica parcialmente los estándares de indentación de bloques de código, o existen inconsistencias en la codificación: redundancia o 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 solo algunas restricciones técnicas, o no se evidencia aplicación de estándares o criterios de eficiencia en la codificación, con ausencia de comentarios, o no aplica patrones de arquitectura ni patrones de diseño, aunque la codificación es funcional.	No utiliza el lenguaje de programación TypeScript o no utiliza Angular, o no cumple con la mayoría de restricciones técnicas o el código no es funcional.	
	3.0 puntos	2.0 punto	1.0 puntos	0 puntos	
C06. Naming Standards	El desarrollador aplica en todos los nombres de objetos de programación como paquetes, componentes, interfaces, clases, objetos, variables, constantes y métodos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, 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 sólo en algunos 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.	
	1.0 puntos	0.5 punto	0.25 puntos	0 puntos	
Total	20 puntos	13.0 puntos	5.75 puntos	0 puntos	

Lima, 6 de Mayo del 2024