



SI730 - APLICACIONES WEB
EXAMEN PARCIAL
2024-1

Sección: SI91, SV51, SW51, SW52, SW53, SW55, SW56,
WS51, WS52, WS53, WX53, WX54, WX55, WX56
Profesores: Bautista Ubillús, Efraín Ricardo
Cáceres Honores, Francisco José
Mori Paiva, Hugo Allan
Reupo-Musayón Gastulo, Naldo
Sánchez Ponce, Alex Humberto
Tinoco Licas, Juan Carlos
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** con proyecto de software en la Actividad Examen parcial.
3. Para el desarrollo del examen se requiere archivos de configuración que se encuentran como archivo adjunto (**upc-pre-202401-si730-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 profesor asignado a su sección, según se indica a continuación:

SI91: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe
SV51: Morales Arévalo, Juan Carlos al correo pcsijumo@upc.edu.pe
SW51: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe
SW52: Prialé De La Peña, Mónica Rosario al correo pcsimpri@upc.edu.pe
SW53: Prialé De La Peña, Mónica Rosario al correo pcsimpri@upc.edu.pe
SW55: Bautista Fuentes, Iván Christian al correo pcsiibau@upc.edu.pe
SW56: Morales Arévalo, Juan Carlos al correo pcsijumo@upc.edu.pe
WS51: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe
WS52: Bautista Fuentes, Iván Christian al correo pcsiibau@upc.edu.pe
WS53: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe
WX53: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe
WX54: Prialé De La Peña, Mónica Rosario al correo pcsimpri@upc.edu.pe
WX55: Bautista Fuentes, Iván Christian al correo pcsiibau@upc.edu.pe
WX56: Morales Arévalo, Juan Carlos al correo pcsijumo@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 analíticos de rendimiento sobre los evaluadores y las evaluaciones en general, sin embargo, desea ir iniciando con el desarrollo del web frontend para la plataforma.

La evaluación tiene un empaquetado adjunto conteniendo un archivo, el cual puede ser usado 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 el archivo de configuración mencionado (debe ubicar el archivo 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 *Mental State Exams* (*id, patientId, examinerId, examDate, orientationScore, registrationScore, attentionAndCalculationScore, recallScore, languageScore*) se encuentra en el endpoint:

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

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>

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

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

- Un Toolbar, donde se muestra a la izquierda el logo de HIGN y a continuación el nombre de la empresa (Hartford Institute for Geriatric Nursing). Al extremo derecho se muestra las opciones “Home”, “Examiner Performance Overview”.
- El toolbar debe ofrecer además a la derecha de las opciones del toolbar, select buttons con “EN” y “ES” para el switching de idioma.
- La vista **Home** muestra como título el texto “Home” y como contenido el texto “Welcome to HIGN”.
- Dentro de la vista **Home** se muestra también una sección que se presenta en un card. La sección tiene como título “Mental State Exam Analytics”, como subtítulo “Score Analytics” y como contenido los siguientes indicadores generales: Exam Count (Cantidad de Evaluaciones), Highest Score (Calificación Total más alta), Lowest Score (Calificación Total más baja) y Average Score (Calificación Total promedio). Al lado de cada etiqueta de indicador se muestra el valor calculado correspondiente, en base a la información de Mental State Exams. Para estos indicadores se considera el valor de *total score* calculado en cada mental state exam.
- El concepto Calificación Total de un examen (*total score*), debe ser calculado como la sumatoria de puntajes obtenidos por el *patient* en la evaluación para *orientationScore, registrationScore, attentionAndCalculationScore, recallScore* y *languageScore*.
- La vista Examiner Performance Overview, muestra una vista con el título “Examiner Performance Overview” y debajo un conjunto de cards, (2 cards por cada fila). En cada card se muestra como header el nombre completo del *Examiner*, como title el texto *NPI* seguido del valor de *nationalProviderIdentifier*, como subtitle el texto *Mental State Exam Performance*. En la sección de contenido de cada card se presenta los indicadores calculados en base a los exámenes asignados a dicho Examiner: *Current Assigned Mental State Exam Count* (Cantidad Actual de Exámenes de Salud Mental Asignados) y *Average Assigned Mental State Exam Total Score* (Promedio de Calificación Total de Exámenes de Salud Mental Asignados).

- 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 **Examiner Performance Overview** es accesible desde la ruta de navegación *nursing/examiner-performance-overview*.
- 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 solo 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 Vue.js, organización y eficiencia del código. Igualmente se tomará en cuenta la aplicación de patrones de diseño.

Restricciones técnicas:

Nombre su proyecto como `upc2401si730eaucode` donde **code** es su código de estudiante en minúsculas. Debe generar el proyecto con Vite y utilizar Vue 3. El equipo requiere que la interfaz de usuario esté basada en Material Design utilizando la biblioteca de componentes de UI **PrimeVue**, junto con la biblioteca de utilidades de CSS **PrimeFlex** y la biblioteca de icons **PrimeIcons**, mientras que para la comunicación con el backend debe apoyarse en **axios** (<https://github.com/axios/axios>). La aplicación debe soportar in-app navigation y utilizar **VueRouter** (<https://router.vuejs.org/>) para el manejo de routing en la aplicación. Debe incluir ARIA attributes en las vistas. La interfaz de usuario debe mostrar los textos en **inglés**. Para el soporte de i18n utilice **Vue-i18n** (<https://vue-i18n.intlify.dev/>). La organización del proyecto debe ser domain-driven, 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 exams, incluyendo lo relacionado a examiners). La programación en JavaScript debe ser object-oriented. Aplique buenas prácticas y convenciones aplicadas en clase para nomenclatura lógica y física de clases y componentes. El proyecto de aplicación debe poder abrirse sin problemas en JetBrains WebStorm. Para la nomenclatura física de componentes, clases e interfaces utilice kebab-case. Aplique en los nombres físicos sufijos que indiquen el tipo de elemento: `.component.vue`, `.service.js`. El sufijo *component* no debe ser parte del nombre del componente cuando se use en templates (Por ejemplo `user-list.component.vue` debería utilizarse en código de template como `<user-list></user-list>`). Para la nomenclatura de componentes de PrimeVue dentro de los templates utilice kebab-case. Asigne el prefijo *pv-* al momento de incorporar su uso en el proyecto (Por ejemplo, `pv-card`). 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.

Para los cálculos analíticos como promedios u otros, construya funciones personalizadas, pudiendo utilizar en su implementación iteraciones, o las funciones `reduce()`, `map()` o `forEach()` de JavaScript según convenga (ver referencias).

Para aplicar filtros a colecciones de elementos puede utilizar la función `filter()` de JavaScript (ver referencias). Para dar formato a números puede utilizar la función de JavaScript *NumberFormat* (ver referencias). Comente los archivos de código fuente en JavaScript elaborados por usted utilizando las

convenciones de **JSDoc** (ver referencias), con block comments incluyendo descripción resumida, 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, features y su información como author. Antes de la generación del archivo **.zip** para el envío, elimine la carpeta *node_modules*.

Referencias:

<https://router.vuejs.org/>
<https://router.vuejs.org/installation.html>
<https://vue-i18n.intlify.dev/>
<https://primevue.org/setup>
<https://primevue.org/flex>
<https://primevue.org/icons/>
<https://primevue.org/toolbar/>
<https://primevue.org/button/>
<https://primevue.org/card>
<https://primevue.org/gridsystem>
<https://primevue.org/image/>
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/NumberFormat
<https://github.com/typicode/json-server>
<https://axios-http.com/docs/intro>
<https://www.geekbits.io/javascript-average-function/>
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter

Rúbrica de calificación

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, idioma y rutas de navegación. Se evidencia que el Toolbar presenta las opciones Home, y Examiner Performance Overview. Se evidencia que la aplicación recupera y presenta elementos en Home según lo solicitado y presenta una vista de tipo page-not-found ante rutas de navegación no soportadas.	La interfaz de usuario no evidencia con claridad que aplica Responsive Web Design, o no cumple con todas las características solicitadas, sin embargo, presenta el Toolbar con las opciones solicitadas, 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. Examiner Performance Overview Feature View and Operation	Se evidencia que la aplicación proporciona en el Toolbar la opción Examiner Performance Overview, llevando a la ruta de navegación solicitada, asociada a la vista que incluye cards para presentar la información de performance indicators de cada examiner recuperado del backend, con el comportamiento esperado, con validaciones, control de errores de interacción o navegación, con comunicación adecuada hacia el usuario. Las vistas cumplen con los requisitos de interfaz de usuario, ruta de navegación, con todas las características y restricciones indicadas.	Se evidencia que la aplicación proporciona en el Toolbar la opción Examiner Performance Overview llevando a la ruta de navegación solicitada, asociada a la vista que incluye cards para presentar la información de performance indicators de cada examiner recuperado 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.	La aplicación presenta las vistas en la ruta de navegación, pero no presenta la información solicitada o incumple la mayoría de características y comportamiento solicitados, o solo implementa una de las dos opciones.	No se implementa ninguna de las dos opciones.	
	7.0 puntos	4.5 puntos	2.0 puntos	0 puntos	
C04. Code Organization	El desarrollador organiza el código y los elementos de frontend de la solución, aplicando buenas prácticas de JavaScript y Vue, 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 todas las características y restricciones técnicas sobre organización.	El desarrollador aplica en frontend un enfoque domain-driven, la mayoría de convenciones, recomendaciones y buenas prácticas de JavaScript o Vue, cumpliendo con la mayoría de las características y restricciones técnicas sobre organización.	El desarrollador aplica en frontend solo algunas de las convenciones, recomendaciones y buenas prácticas de JavaScript o Vue, cumpliendo con algunas de las características y 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 JavaScript, el framework Vue 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 JavaScript y el framework Vue. 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, ó existen 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 solo algunas restricciones técnicas, o 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, aunque la codificación es funcional.	No utiliza el lenguaje de programación JavaScript o no utiliza Vue, 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, cumpliendo con todas las características y restricciones técnicas relacionadas con nomenclatura.	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, cumpliendo con la mayoría de características o restricciones técnicas relacionadas con nomenclatura.	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, cumpliendo con algunas de las características o restricciones técnicas relacionadas con nomenclatura.	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		

Lima, 8 de Mayo del 2024