



Upc-pre-202301-si729-examen-parcial v1

Aplicaciones Open Source (Universidad Peruana de Ciencias Aplicadas)



Escanea para abrir en Studocu



DESARROLLO DE APLICACIONES OPEN SOURCE (SI729)
EXAMEN PARCIAL
2023-1

Sección: SW51, SW53, WS51, WS52, WX51, WX52

Profesores: Flores Moroco, Juan Antonio

Mori Paiva, Hugo Allan

Navarrete Vilca, Elio Jefferrson

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-202301-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:

SW51: Prialé De La Peña, Mónica Rosario al correo pcsimpri@upc.edu.pe

SW53: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe

WS51: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe

WS52: Prialé De La Peña, Mónica Rosario al correo pcsimpri@upc.edu.pe

WX51: Rojas Malásquez, Royer Edelwer al correo pcisrroj@upc.edu.pe

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. Los problemas técnicos se recibirán como **máximo 15 minutos culminado el examen**.

Enunciado:

Pregunta 1 (20 p.).

Caso TechnoGym, inc.

Su cliente, TechnoGym, inc. (<https://www.technogym.com/>) fue fundada en 1983 por Neri Alessandri, a la edad de 22 años. Desde esa época la empresa ha venido creciendo y adquiriendo mayor prestigio, cosechando grandes logros empresariales, como convertirse en el Official Supplier para entrenamiento atlético en las Olimpiadas de Sidney en el 2000, donde más de 2000 atletas utilizaron equipos Technogyms. En el año 2012, se lanzó mywellness (<https://www.mywellness.com/>), el primer cloud platform en el sector de fitness y wellness, ofreciendo la posibilidad de acceder a programas personales desde cualquier lugar y en cualquier momento, a la vez que se permite a los operadores conectarse con sus clientes utilizando web o mobile systems, dentro o fuera del gimnasio. Para ello ofrece un website donde los developers pueden registrarse para obtener acceso al mywellness cloud API o al Unity SDK (el SDK permite crear aplicaciones para los equipos de ejercicio con display, como los Treadmills, Bikes o Ellipticals).

Como parte de su estrategia de negocio, ellos desean implementar el evento *Indoor Marathon*, una competencia en la que pueden participar individuos o equipos, donde un porcentaje de los ingresos se destinaría a causas benéficas.

El proyecto de *Indoor Marathon* contempla la implementación de 6 centros de competición ubicados cada uno en una ciudad (Los Angeles, New York, Sidney, Montreal, París y Madrid). Cada *Indoor Marathon Center*, contará con 42 Treadmills *Technogym Run* (https://www.technogym.com/en-US/product/technogym-run_DGK0.html), tres pantallas gigantes en alta definición en tres lados del ambiente donde se realizará la maratón. Los participantes contarán con una aplicación disponible para Móvil Android o iOS, o wearable la cual, utilizando Bluetooth o NFC, permitirá que se autentiquen en la máquina en que decidan realizar la maratón. En caso que exista algún inconveniente y decida cambiar de Treadmill, se enlazará con el otro de los 42 equipos disponibles en el centro y continuará en la Maratón.

Cada *Technogym Run* cuenta con conectividad WiFi (ver especificaciones en https://www.technogym.com/en-US/product/technogym-run_DGK0.html#specifications-table), lo cual permitirá el envío de información a un Edge Gateway que contendrá una Edge Application elaborada con Unity SDK, la cual estará enlazada con las tres pantallas gigantes para la presentación del ranking local de participantes en tiempo real y gráficos analíticos de desempeño. La información de ranking de los 7 ganadores (7 mejores records) de cada Indoor Marathon Center, será enviada por la Edge Application hacia una base de datos Cloud FireStore de Google, la cual servirá de fuente de datos para el Website Oficial del concurso, elaborado con Angular y publicado en Firebase Hosting, donde se publicarán los rankings generales a nivel de todos los Marathon Centers. Adicionalmente, TechnoGym desea que el Edge Gateway centralice la información periódica (cada 20 segundos) de *health check* de cada uno de los 42 Treadmills (incluyendo en cada envío individual el id del equipo, timestamp, voltaje en voltios, potencia de consumo en Watts, Horse Power, temperatura en grados Centígrados, tiempo de uso acumulado desde el inicio del evento). Al cierre del evento, estos datos serán consolidados en mywellness para ser analizados posteriormente por el equipo técnico de TechnoGym. Para tal fin, cada equipo *Technogym Run* contará con una aplicación elaborada en Unity SDK que recolectará esa información y la enviará al Edge Gateway.

Como beneficio adicional, los deportistas que deseen, podrán conectar su dispositivo compatible con *Apple GymKit* para que se sincronice los datos (<https://support.apple.com/guide/watch/use->

gym-equipment-with-apple-watch-apdf855bb3c0/watchos) que recolecta el *Technogym Run* que esté utilizando. La información de los manuales de producto de los equipos de TechnoGym se encuentran en <https://support.technogym.com/product-manuals/all>¹.

Para ello, su equipo se encuentra desarrollando para TechnoGym, una plataforma de backend para presentar a ganadores y records del Indoor Marathon, sin embargo, desea ir iniciando con el desarrollo del website de presentación de resultados.

La evaluación tiene un empaquetado adjunto contenido dos archivos: db.json y routes.json, los cuales pueden ser usados por json-server (<https://www.npmjs.com/package/json-server>) 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 --routes routes.json
```

Esto inicia el Fake API.

La información de los Participants en general (*id, firstName, lastName, photoUrl, centerId, ranking, recordTime*) se encuentra en el endpoint:

<http://localhost:3000/api/v1/participants>

La información de los Marathon Centers (*id, name*) se encuentra en el endpoint:

<http://localhost:3000/api/v1/centers>

La información de los Participants con mejores tiempos en un Marathon Center específico se puede encontrar en el endpoint:

<http://localhost:3000/api/v1/centers/:centerId/participants>

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

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

- Un toolbar, donde a la izquierda se muestra el branding **Technogym Indoor Marathon** y a la derecha se muestra las opciones “Home” y “Records”.
- La vista **Home** muestra un card con título “Marathon Winner”, incluyendo *photo* (la ruta de la imagen para cada participant está disponible en *photoUrl*), *first name*, *last name*, *record time* (en formato hh:mm:ss), *marathon center name*, para el participant con mejor tiempo de entre todos los Marathon Centers (el cual tiene valor de *ranking* 1).
- La vista **Records** presenta una vista de tabla de **solo lectura** (es decir que no se requiere soporte de adición, modificación o eliminación de registros) con la información de los participantes con mejor record time en cada Marathon Center. En cada fila debe apreciarse el *participant id, first name, last name, marathon center name, ranking y record time* (en

¹ Para efectos de la evaluación, de requerirse tome como referencia la información técnica en el manual del equipo *Technogym MyRun*.

formato hh:mm:ss). Debe presentarse **solo una tabla** con toda la información solicitada, **no** una tabla por cada center. La tabla debe ofrecer pagination, sorting y filtering.

- 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 **Records** es accesible desde la ruta de navegación */marathon/records*.
- La vista **raíz** (accesible desde la ruta de navegación */*) debe redirigir al usuario a la vista */home*.

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 `upc2301si729eau 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. Debe incluir ARIA attributes en las vistas. La interfaz de usuario debe mostrar los textos en **inglés**. El proyecto de aplicación debe poder aperturarse sin problemas en JetBrains WebStorm. 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 *marathon* (para componentes o elementos relacionados con el marathon, como *participants* o *centers*). 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.

Referencias:

<https://angular.io/guide/router>
<https://angular.io/guide/http>
<https://material.angular.io/components/card/overview>
<https://material.angular.io/components/card/overview#accessibility>
<https://material.angular.io/components/table/overview>
<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>

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 y rutas de navegación. Se evidencia que la aplicación recupera y presenta en Home el card con lo solicitado en base a la información del backend.	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 del backend.	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. Records View and Operation	Se evidencia que la aplicación proporciona la opción Records, llevando a la ruta de navegación /marathon/records, asociada a la vista que incluye una tabla para presentar la información de los participantes, 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 Records, llevando a la ruta de navegación /marathon/records, asociada a la vista que incluye una tabla para presentar la información de participantes, 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 Records, llevando a la vista que incluye una tabla de navegación, /marathon/records, asociada a la vista que incluye una tabla para presentar la información de participantes, 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.	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, ó existen inefficiencias 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 TypeScript o no utiliza Angular, ó 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, 9 de Mayo del 2023