



**SI730 - APLICACIONES WEB
EXAMEN PARCIAL
2023-1**

Sección: SW51, SW52, SW53, SV54, WS51, WS52
Profesores: Reupo-Musayón Gastulo, Naldo
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-202301-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:

SW51: Morales Arévalo, Juan Carlos al correo pcsijumo@upc.edu.pe

SW52: Morales Arévalo, Juan Carlos al correo pcsijumo@upc.edu.pe

SW53: García Rojas, Fidel Eugenio al correo pcsifgar@upc.edu.pe

SV54: Bautista Fuentes, Iván Christian al correo pcsiibau@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

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 Nerio 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 autenticuen 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 Vue 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.

En paralelo al equipo de desarrollo del proyecto del website de ganadores, Technogym desea encargarle el desarrollo de una aplicación dirigida al equipo técnico de investigación y desarrollo de Technogym. Para ello, un equipo se encuentra desarrollando para TechnoGym, una plataforma de backend para presentar la información de estado de los Treadmills *Technogym Run*. Mientras tanto, su equipo se encargará del desarrollo de la web application para analytics sobre los datos de health check de los equipos *Technogym Run* instalados en cada Marathon Center. Esto permitirá al equipo de investigación y desarrollo, identificar oportunidades de mejora en el diseño y proceso de fabricación de los equipos.

La evaluación tiene un empaquetado adjunto conteniendo 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 Treadmills en general (`id`, `serialNumber`, `centerId`) se encuentra en el endpoint:

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

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

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

La información de los Health Check Records (`id`, `treadmillId`, `year`, `month`, `day`, `hour`, `minutes`, `seconds`, `volts`, `watts`, `hp`) se puede encontrar en el endpoint:

<http://localhost:3000/api/v1/health-checks>

La información de los Health Check Records de un Treadmill en particular se encuentran en el endpoint:

<http://localhost:3000/api/v1/treadmills/:treadmillId/health-checks>

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

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 a la izquierda se muestra el branding **TechnoGym TechnoRun Analytics** y a la derecha se muestra las opciones “Home” y “Health Checks”.
- La vista **Home** (accesible desde la ruta de navegación /home) presenta un card con el título *Average Performance*, conteniendo la información de los valores promedio en general (sin distinguir equipos en particular) para los siguientes indicadores: volts, watts, hp. **No** considere en el cálculo de promedios aquellos *health check records* capturados en el momento que el equipo correspondiente no estuvo operando en ese instante en la Marathon, es decir que tenga **cero** (0) en su valor de *hp* en el *health check record*. Tome en cuenta que cada equipo Treadmill puede tener asociados varios health check records y que puede darse el caso que en un momento un equipo haya estado siendo usado y en otro no.
- La vista **Health Checks**, muestra una tabla de **solo lectura** (es decir que no se requiere soporte de adición, modificación o eliminación de registros) donde se muestra la información de health check records, presentando como columnas: *Record Id*, *Treadmill Id*, *Serial Number*, *Center Name*, *Date* (en yyyy-mm-dd), *Time* (en hh:mm:ss), *Volts*, *Watts*, *HP*. La tabla debe ofrecer pagination, sorting y filtering. Considere todos los registros.
- La vista **Home** es accesible desde la ruta de navegación /home.
- La vista **Health Checks** es accesible desde la ruta de navegación /analytics/health-checks.
- La vista **raíz** (accesible desde la ruta de navegación /) debe redirigir al usuario a la vista /home.
- 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*.

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 `upc2301si730eaucode` 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**, 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** 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**. 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 *analytics* (para componentes o elementos relacionados como treadmills o health check records). 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, construya funciones personalizadas, pudiendo utilizar en su implementación iteraciones, o las funciones `reduce()` o `forEach()` de JavaScript (ver referencias). Para aplicar filtros a colecciones de elementos puede utilizar la función `filter()` de JavaScript (ver referencias).

Referencias:

<https://router.vuejs.org/>
<https://router.vuejs.org/installation.html>
<https://primevue.org/setup>
<https://primevue.org/card>
<https://primevue.org/button>
<https://primevue.org/toolbar>
<https://primevue.org/setup>
<https://primevue.org/gridsystem>
<https://primevue.org/flex>
<https://primevue.org/datatable/>
<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 y rutas de navegación. Se evidencia que la aplicación recupera y presenta en Home según lo solicitado en base a la información del backend 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, 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. Health Checks View and Operation	Se evidencia que la aplicación proporciona la opción Health Checks, llevando a la ruta de navegación /analytics/health-checks, asociada a la vista que incluye una tabla para presentar la información de los health check records, 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 Health Checks, llevando a la ruta de navegación /analytics/health-checks, asociada a la vista que incluye una tabla para presentar la información de health check records, 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.	La aplicación presenta la vista en la ruta de navegación, pero no presenta la información solicitada o incumple la mayoría de características y comportamiento solicitados.	No se implementa la opción.	
	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, 12 de Mayo del 2023