



## Examen Parcial 2023-02 Open Source

Aplicaciones Open Source (Universidad Peruana de Ciencias Aplicadas)



Escanea para abrir en Studocu



**DESARROLLO DE APLICACIONES OPEN SOURCE (SI729)**  
**EXAMEN PARCIAL**  
**2023-2**

**Sección:** SV52, SW53, SW54, SW55, WS51, WX52

**Profesores:** Flores Moroco, Juan Antonio  
Mori Paiva, Hugo Allan  
Navarrete Vilca, Elio Jefferrson  
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-202302-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:

SV52: Delgado Vite, Jorge Luis al correo [pcsijdev@upc.edu.pe](mailto:pcsijdev@upc.edu.pe)

SW53: García Rojas, Fidel Eugenio al correo [pcsifgar@upc.edu.pe](mailto:pcsifgar@upc.edu.pe)

SW54: Rojas Malásquez, Royer Edelwer al correo [pcisrroj@upc.edu.pe](mailto:pcisrroj@upc.edu.pe)

SW55: García Rojas, Fidel Eugenio al correo [pcsifgar@upc.edu.pe](mailto:pcsifgar@upc.edu.pe)

WS51: Delgado Vite, Jorge Luis al correo [pcsijdev@upc.edu.pe](mailto:pcsijdev@upc.edu.pe)

WX52: Rojas Malásquez, Royer Edelwer al correo [pcisrroj@upc.edu.pe](mailto:pcisrroj@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 se recibirán como **máximo 15 minutos culminado el examen**.

## Enunciado:

### Pregunta 1 (20 p.).

#### Caso Ember Technologies, inc.

Clay Alexander es un inventor y empresario en serie estadounidense que posee más de 200 patentes en todo el mundo y es el fundador y director ejecutivo de Ember (<https://ember.com/>), empresa que se creó para resolver un problema simple pero universal para los bebedores de café y té. Utilizando su tecnología patentada de control de temperatura, Ember, una marca mundial de electrónica de consumo que inventó una manera de extender el momento de perfección con su bebida caliente favorita, fabricando de la primera taza con control de temperatura del mundo.

Tras el éxito de Ember en el mercado de consumo, Alexander reconoció la importancia del control de temperatura para otras aplicaciones. En febrero de 2022, Alexander anunció una asociación histórica con Cardinal Health en la que la tecnología de refrigeración termoeléctrica de Ember se utilizará para interrumpir la cadena de frío en la industria de la salud. Alexander y su equipo crearon Ember Cube, la primera caja de envío autorefrigerada basada en la nube del mundo.

La visión de Alexander para Ember se extiende mucho más allá de su posición actual en el mercado de bebidas de consumo. Ember planea aprovechar el poder del control de la temperatura para transformar la forma en que el mundo come, bebe y vive.

Para ello, su equipo se encuentra desarrollando para Ember Technologies, inc, una plataforma de backend para presentar información sobre sus productos, 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://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 Products en general (*id, title, description, photoUrl, price, bundleId, rating*) se encuentra en el endpoint:

`http://localhost:3000/api/v1/products`

La información de los Bundles (*id, title, description, photoUrl, price, rating*) se encuentra en el endpoint:

`http://localhost:3000/api/v1/bundles`

La información de los Products que pertenecen a un Bundle específico se puede encontrar en el endpoint:

http://localhost:3000/api/v1/bundles/:bundleId/products

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 **Ember** y a la derecha se muestra las opciones "Home" y "Bundles".

La vista **Home** muestra una sección con la imagen en el URL:

[https://ember.com/cdn/shop/files/230914\\_launch\\_WebBanner-Desktop\\_2048x.jpg](https://ember.com/cdn/shop/files/230914_launch_WebBanner-Desktop_2048x.jpg)

ocupando el ancho de la pantalla (hero image). Debajo del hero image debe mostrar un texto de título centrado con la frase *The perfect temperature, every time*. Debajo del título debe aparecer un párrafo centrado con el contenido *Ember's patented technology allows you to experience precision temperature control in your everyday life, with the vision to transform how the world eats, drinks, and lives, empowering you to experience your world without limits*.

La vista **Bundles** muestra una colección de cards, donde en cada card se muestra la información de un bundle, incluyendo *photo* (la ruta de la imagen para cada bundle está disponible en *photoUrl*), *title*, *rating*, *price* (en formato \$ 999.99), *You Save* (en formato \$ 99.99), *description*. El valor de *You Save* debe calcularse como la diferencia del precio del bundle menos los precios unitarios de los productos que forman parte del bundle<sup>1</sup>.

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 **Bundles** es accesible desde la ruta de navegación */store/bundles*.

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 `upc2302si729eaucode` 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 **CurrencyPipe** en los templates para los formatos de currencies y **DecimalPipe** para los formatos de números con decimales (ver sección de referencias). Debe incluir

---

<sup>1</sup> Sobre el valor a colocar para *You Save*: Se entiende como el valor que estaría ahorrando el cliente al adquirir el bundle en vez de los productos que lo conforman de manera individual. Por ejemplo, si el precio del bundle es \$ 39.99 y la suma de los productos que lo conforman es \$ 43.99 entonces el valor a colocar para *You Save* sería \$ 4.00.

ARIA attributes in the views. The user interface must show the texts in **inglés**. The project application must be able to open without problems in JetBrains WebStorm.

The organization of the project must be domain-driven applying layered architecture, considering the sub-domains *public* (for elements or components general of the user interface, like home), *shared* (for elements base or of common use in other sub-domains) and *store* (for components or elements related to the catalog of products for sale, like bundles).

Apply good practices for logical and physical naming of classes and components. Distribute adequately the elements inside each sub-domain folder, considering folders like *pages*, *components*, *services*, *model*. In *pages* only are placed components that have direct relation with navigation routes. In *components* are placed components that are included in others and are not directly associated to navigation routes. Apply good practices for logical and physical naming of classes and components. Comment the source code files elaborated by you, including *summary* with the purpose and *author* with your name and surname. Include in the README.md file, the information of the application, description and your information as 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/CurrencyPipe>

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

<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>

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. 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	
	<b>2.0 puntos</b>	<b>1.25 punto</b>	<b>0.5 puntos</b>	<b>0 puntos</b>	
<b>C02. User Interface &amp; Home View</b>	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 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.	
	<b>5.0 puntos</b>	<b>3.5 puntos</b>	<b>1.5 puntos</b>	<b>0 puntos</b>	
<b>C03. Bundles View and Operation</b>	Se evidencia que la aplicación proporciona la opción Bundles, llevando a la ruta de navegación /store/bundles, asociada a la vista que incluye una colección de cards con información de los bundles, 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 Bundles, llevando a la ruta de navegación /store/bundles, asociada a la vista que incluye una colección de cards para presentar la información de bundles, 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 Bundles, 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.	
	<b>7.0 puntos</b>	<b>4.5 puntos</b>	<b>2 puntos</b>	<b>0 puntos</b>	
<b>C04. Code Organization</b>	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.	
	<b>2.0 punto</b>	<b>1.25 puntos</b>	<b>0.5 puntos</b>	<b>0 puntos</b>	
<b>C05. Code Quality</b>	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 ineficiencias 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.	
	<b>3.0 puntos</b>	<b>2.0 punto</b>	<b>1.0 puntos</b>	<b>0 puntos</b>	
<b>C06. Naming Standards</b>	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.	
	<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>13.0 puntos</b>	<b>5.75 puntos</b>	<b>0 puntos</b>	

Lima, 2 de Octubre del 2023