



**1ASI0730 - APLICACIONES WEB
EXAMEN PARCIAL
202510**

NRC: 4380

Profesores: Mori Paiva, Hugo Allan
Reupo-Musayón Gastulo, Naldo
Sánchez Ponce, Alex Humberto
Sánchez Seña, Alberto Wilmer
Velásquez Núñez, Ángel Augusto
Villafuerte Bazán, Óscar Iván

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** (único formato válido) con nombre `ea<nrc>u<código-estudiante>.zip` (por ejemplo, `ea4380u201821873.zip`), conteniendo el proyecto de software, en la Actividad para el Examen parcial.
3. Para el desarrollo del examen se requiere archivos de configuración que se encuentran como archivo adjunto (**upc-pre-202510-1asi0730-examen-parcial-4380_v1-files.zip**).
4. Debe crear su proyecto de solución y evidenciar la autoría.
5. Antes de la generación del archivo **.zip** para el envío, elimine la carpeta `node_modules`.
6. 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).

Enunciado:

Pregunta 1 (20 p.).

Caso Paypal

Paypal (<https://developer.paypal.com>) es un servicio de pagos en línea que permite a las personas y empresas enviar, recibir y almacenar dinero electrónicamente. Es una plataforma global que funciona como una alternativa digital a los métodos de pago tradicionales, como cheques o giros postales

La plataforma está diseñada para simplificar el intercambio de dinero de forma rápida y segura, lo que le permite a los usuarios tener una alternativa al sistema bancario tradicional.

Debido al crecimiento exponencial de transacciones a través de la plataforma, el cual está teniendo impacto en el mundo financiero, la empresa ha decidido exponer información consolidada de sus transacciones, a modo de transparencia; para ello, su equipo de desarrollo de TI se encuentra

desarrollando, una plataforma de backend para presentar información sobre el uso de la plataforma, sin embargo, desea ir iniciando con el desarrollo del web frontend para la plataforma informativa.

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 *Countries* en general (*id*, *country*, *description*) se encuentra en el endpoint:

<http://localhost:3000/countries>

La información de las *Categories* (*id*, *name*, *description*, *income*, *expenses*, *countryId*, *registeredInAt*) se encuentra en el endpoint:

<http://localhost:3000/categories>

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 Paypal (para ello utilice el servicio de Clearbit Logo API) y a continuación el texto “paypal.com”.
- En el Toolbar, al centro se muestra las opciones “**Home**”, “**Category**”.
- 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 Paypal”.
- Dentro de la vista **Home** se muestra también una sección con el título “Country” donde se presenta un grid de dos componentes *Country Summary* por columna. Para cada *Country* para el movimiento de dinero por país debe presentarse un componente *Country Summary* conteniendo un card.
- En el componente *Country Summary*, el card de Country debe tener en la sección header el *Country name*. En la sección content debe estar el *Country description*. En la sección footer debe mostrarse un indicador estadístico: Amount.
- El indicador *Amount* muestra el monto total de dinero que se mueve por país para Paypal, el cual se obtiene sumando *income* y *expenses* de todas las categorías de un país.
- La vista **Category**, muestra una vista con el título “**Transaction Category**” y debajo una sección **Category-Validation**.
- La sección **Category-Validation** dentro de la vista **Transaction Category**, presenta un form conteniendo un input text field con el label *Category Name*, junto con un button *Category*

Validation. Cuando el usuario ingresa la categoría y hace click en el botón *Category-Validation*, el sistema valida si la categoría concuerda con alguna ya existente en *Categories*. Si encuentra una coincidencia y el valor de *registeredInAt* no tiene valor, actualiza en el endpoint */categories* el valor de *registeredInAt* con la fecha y hora actual y muestra debajo los datos asociados a la categoría y el país (*Country Id*, *Country Name*, *Category Name* y *registeredInAt*). Si no encuentra coincidencia, debe mostrar debajo el texto *"Invalid Category"*. Si se encuentra coincidencia, pero ya existe un valor de fecha y hora en *registeredInAt*, entonces debe mostrar debajo el texto *"Already Category-Validation"*.

- 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 **Category** es accesible desde la ruta de navegación *countries/transaction-category-details/new*
- 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.
- La interfaz de usuario debe utilizar uno de los tres presets de PrimeVue: *Aura*, *Lara*, *Material* o *Nora*.

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 *eanrcucode* donde *nrc* es su número de NRC y *code* es su código de estudiante en minúsculas (por ejemplo *ea4380u201621873*).

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 aplicando layered y component-based architecture, object-oriented programming y design patterns, 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 **transaction** (para componentes o elementos relacionados con country, incluyendo lo relacionado a transaction-detail). 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. Aplique patrones de diseño, incluyendo *Request/Response*, *Resource* y *Assembler*.

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 en inglés de la aplicación, descripción, features y su información como author.

Antes de la generación del archivo **.zip** (único formato válido) para el envío, elimine la carpeta *node_modules*. El nombre del archivo .zip debe seguir la estructura *eanrcu**code**.zip* (por ejemplo, *ea4380u201821873.zip*).

Referencias:

<https://clearbit.com/blog/logo>
<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/tree/v0>
<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 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. Adicionalmente presenta soporte completo para i18n.	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. Features View and Operation	Se evidencia que la aplicación proporciona los features solicitados, 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. Adicionalmente presenta soporte completo para i18n.	Se evidencia que la aplicación proporciona los features solicitados, llevando a la ruta de navegación solicitada, con el comportamiento esperado, 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.	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, 19 de Mayo del 2025