



**SI730 - APLICACIONES WEB**  
**PRÁCTICA CALIFICADA 1**  
**2024-1**

**Sección:** SW55  
**Profesor:** Cáceres Honores, Francisco José  
**Duración:** 100 minutos  
**Indicaciones:**

1. El examen consta de 1 pregunta, y tendrá 100 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 PC1 (Práctica Calificada 1).
-

## Enunciado:

### Pregunta 1 (20 p.).

#### Online Gaming List

Formas parte del equipo de desarrollo de BITZ IT S.A.C, una empresa de software en pleno crecimiento que compite en un mercado altamente competitivo por proyectos de licitación.

Se te ha solicitado desarrollar un sitio web con la información de los videojuegos mas destacados del momento, para ello se te ha entregado un endpoint en donde encontrarás toda la información.

Para implementar la solución deberás tener en cuenta lo siguiente:

- Obtener datos sobre los videojuegos, accediendo al endpoint:  
<https://www.freetogame.com/api/games>
- Para el desarrollo web de lado web frontend, se ha seleccionado JavaScript como lenguaje de programación y Vue como Frontend Framework.
- **Implemente las siguientes características:**
  - La aplicación debe presentar en la vista home un Toolbar en la parte superior, con el título **"Online Gaming List"**.
  - La aplicación debe presentar debajo una vista "Gaming List" con cards para los videojuegos.
  - Para cada **card**, la información del producto a mostrar incluye:
    - **Código:** Que consiste en el ID del videojuego acompañado con tantos ceros se necesitan para que complete un código de 8 dígitos, por ejemplo, para el ID 12 debe mostrar 00000012.
    - **Título:** La información está en la propiedad *title*.
    - **Descripción:** La información está en la propiedad *short\_description*.
    - **Imagen:** El enlace de la imagen están en la propiedad *thumbnail* del objeto.
    - **Plataforma:** El valor de la plataforma está en la propiedad *platform* del objeto.
    - **Editor:** El valor del editor está en la propiedad *publisher* del objeto.
    - **Desarrollador:** El valor del desarrollador está en la propiedad *developer* del objeto.
    - **Fecha de lanzamiento:** El valor de la fecha de lanzamiento está en la propiedad *release\_date* del objeto.
    - **Botón para ingresar al juego:** El valor del enlace de redireccionamiento está en la propiedad *freetogame\_profile\_url* y debe abrirse en una pestaña nueva.
- La aplicación debe presentar en la parte inferior de la vista "Gaming List", una sección de footer con la información "Copyright © 2024 Gaming List, inc. All rights reserved." en la primera línea. Además debe incluir en la segunda línea del footer la información "Developed by" y los datos del Developer (considérese a usted como autor, indicando código, nombre y apellido.)

El CTO de la compañía 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:

- 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>).
- Debe incluir ARIA attributes en las vistas.
- La interfaz de usuario debe mostrar los textos en **inglés y en español utilizando i18n** (<https://vue-i18n.intlify.dev/>).
- La organización del proyecto debe ser domain-driven y con layered architecture, considerando los sub-dominios *public* (para elementos o componentes generales de la interfaz de usuario, como home) y *gaming* (para componentes o elementos relacionados con el site como *games*).
- 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*, *.entity.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).
- Comente los archivos de código fuente elaborados por usted, incluyendo *summary* con el propósito y *author* con su nombre y apellido.

### Fuera del alcance:

- Sidebar
- Internationalization
- Routing

### Referencias:

- <https://fakestoreapi.com/>
- <https://router.vuejs.org/>
- <https://router.vuejs.org/installation.html>
- <https://primevue.org/installation>
- <https://primevue.org/button>
- <https://primevue.org/toolbar>
- <https://primevue.org/toast>
- <https://primeflex.org/installation>
- <https://primeflex.org/gridsystem>
- <https://primeflex.org/flex>
- <https://axios-http.com/docs/intro>

## Rúbrica de calificación

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>1.0 punto</b>	<b>0.5 puntos</b>	<b>0 puntos</b>	
<b>C02. User Interface</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.	La interfaz de usuario no evidencia con claridad que aplica Responsive Web Design, o no cumple con todas las características solicitadas.	La aplicación no implementa una interfaz de usuario con las características solicitadas.	No se evidencia que presenta la información indicada.	
	<b>3.0 puntos</b>	<b>2.0 puntos</b>	<b>1.0 puntos</b>	<b>0 puntos</b>	
<b>C03. Features</b>	La vista cumple de forma completa con los requisitos de interfaz de usuario, con todas las características y restricciones indicadas en el enunciado. Se cumple que la información es recuperada desde el endpoint solicitado.	La vista cumple de forma parcial con los requisitos de interfaz de usuario, o con la mayoría de las características y restricciones indicadas en el enunciado. Se cumple que la información es recuperada desde el endpoint solicitado.	La vista cumple de forma parcial con los requisitos de interfaz de usuario, con solo algunas de características y restricciones indicadas en el enunciado, o se cumple que la información es recuperada desde el endpoint solicitado, sin embargo, esta no se visualiza.	No se implementa el feature.	
	<b>7.0 puntos</b>	<b>4.5 puntos</b>	<b>2.0 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 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 las 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, 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 JavaScript o Vue, 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>3.0 punto</b>	<b>2.0 puntos</b>	<b>1.0 puntos</b>	<b>0 puntos</b>	
<b>C05. Code Quality</b>	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, 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 JavaScript o no utiliza Vue, 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 o recursos.	
	<b>2.0 puntos</b>	<b>1.0 punto</b>	<b>0.5 puntos</b>	<b>0 puntos</b>	
<b>Total</b>	<b>20 puntos</b>	<b>12.5 puntos</b>	<b>6.00 puntos</b>		

Lima, 19 de Abril del 2024