

# Evaluación Parcial N° 2

**Implementando frontend y pruebas unitarias a mi tienda online**  
**Estudiante**

| Sigla   | Nombre Asignatura       | Tiempo Asignado | % Ponderación |
|---------|-------------------------|-----------------|---------------|
| DSY1104 | DESARROLLO FULLSTACK II | 1 SEMANA        | 30%           |

## 1. Situación evaluativa 1

|  |                       |
|--|-----------------------|
|  | Ejecución<br>práctica |
|--|-----------------------|

|   |                       |
|---|-----------------------|
| X | Entrega de<br>encargo |
|---|-----------------------|

|  |              |
|--|--------------|
|  | Presentación |
|--|--------------|

## 2. Situación evaluativa 2

|  |                       |
|--|-----------------------|
|  | Ejecución<br>práctica |
|--|-----------------------|

|  |                       |
|--|-----------------------|
|  | Entrega de<br>encargo |
|--|-----------------------|

|   |              |
|---|--------------|
| X | Presentación |
|---|--------------|

### 3. Instrucciones generales para el/la estudiante

- **Descripción general de la evaluación:** Cada evaluación parcial será una parte del examen transversal, dividiendo el caso en tres partes para las tres evaluaciones parciales. En cada evaluación parcial se evaluará el avance del caso, centrado en el logro de los indicadores de logro relacionados con el RA2. La evaluación parcial 2 consiste en una entrega de encargo (proyecto e informes) con presentación. Después de completar la Parte 1 en la Evaluación Parcial 1, el equipo continuará trabajando en el desarrollo del caso semestral. Se proporcionará una plantilla de tienda en HTML, CSS, JavaScript y Bootstrap, con un carrito de compras ya implementado en JavaScript. El equipo deberá actualizar el proyecto integrando un nuevo framework de frontend, creando y estructurando componentes, implementando navegación y formularios, y aplicando la lógica de validación del nuevo framework. Además, deberán analizar, documentar y desarrollar pruebas para los componentes frontend para asegurar su correcto funcionamiento.
- **Tiempo** asignado para esta evaluación es de **1 semana** y se realiza en **equipos** (máximo 3 estudiantes por equipo) en **taller de alto cómputo**. Sin embargo, **la evaluación del encargo y la presentación será individual**.
- La **distribución de los porcentajes de las situaciones evaluativas que componen esta evaluación** es la siguiente:

| Evaluación              | Porcentaje dentro de la asignatura | Tipo de situación evaluativa | Distribución de porcentajes |
|-------------------------|------------------------------------|------------------------------|-----------------------------|
| Evaluación Parcial N° 2 | 30%                                | Entrega de encargo           | 40%                         |
|                         |                                    | Presentación                 | 60%                         |

- **Orientaciones para la implementación de la evaluación**
  - Las instrucciones se entregarán en la semana 10 y la entrega en la semana 11.
  - El trabajo debe ser desarrollado en Taller de alto cómputo.
  - El informe debe ser cargado en AVA en la fecha correspondiente.
  - La situación evaluativa 1, denominada Entrega de encargo, representa la culminación del trabajo colaborativo realizado por el equipo. En esta fase, es fundamental que la evaluación sea equitativa entre todos los integrantes del grupo.
  - La situación evaluativa 2, la Presentación, constituye una oportunidad para que los estudiantes defiendan el proyecto desarrollado por sus equipos. Durante esta etapa, es esencial evaluar individualmente a cada estudiante, ya sea mediante su participación en la

*exposición del proyecto o a través de las respuestas dadas a las preguntas formuladas por el docente. Es crucial que tanto la justificación como la argumentación presentada sean consistentes con el proyecto y pertinentes a la asignatura en cuestión.*

## **Instrucciones específicas de la Evaluación:**

### **Situación evaluativa 1: Entrega por encargo**

En la propuesta de actualización del proyecto, los estudiantes deben:

1. Incorpora frontend basado en tecnología JS para la construcción de una aplicación web, integrando elementos de responsividad para el desarrollo de páginas web simples.
  - Comprueba que el proyecto use un framework moderno de JavaScript, como React y que esté bien estructurado.
  - Revisa que los componentes React estén correctamente gestionados en cuanto a propiedades y estados, asegurando su funcionalidad y personalización.
  - Asegúrate de que el diseño del proyecto sea responsivo, adaptándose a diferentes tamaños de pantalla mediante el uso de Bootstrap.
  - Explica cómo utilizaste un framework de JavaScript moderno para diseñar el proyecto frontend.
  - Muestra el proceso de desarrollo de componentes React en tu proyecto, detallando cómo gestiono las propiedades y estados para asegurar un funcionamiento correcto y personalizado de la ampliación.
  - Explica cómo implemento el diseño responsivo utilizando Bootstrap y proporciona ejemplos de su aplicación en diferentes dispositivos.
2. Crea pruebas unitarias para los componentes FRONT-END, abarcando diferentes casos de uso, verificando la lógica y el comportamiento del componente, así como la manipulación del DOM, utilizando Jasmine y Karma.
  - Comprueba que existan pruebas unitarias que validen las funciones clave y comportamientos de los componentes.
  - a. Demuestra la creación de pruebas unitarias para los componentes frontend utilizando Jasmine y Karma. Explicando cómo verificaste la lógica y comportamientos para asegurar la calidad del código.
3. Implementa un proceso de testeo en el desarrollo de software, a través de pruebas unitarias para el FRONT-END, abordando los conceptos claves del proceso de testeo.
  - a. Asegúrate de que se hayan configurado las herramientas de testeo adecuadas y que las pruebas cubran las funcionalidades principales.
  - b. Explica cómo implementaste un proceso de testeo unitario en el proyecto frontend. Detalla los conceptos clave aplicados, como la configuración del entorno de pruebas, escritura de pruebas unitarias, uso de mocks y análisis de resultados.

Documentos requeridos del proyecto por el grupo:

- El documento ERS (Especificación de Requisitos del Software): entregar documento añadiendo la propuesta personalizada, última versión.
- Documento de cobertura de testing: Debe validar el alcance de los elementos cubiertos en el testing.

*Entregables:*

- *Enlace GitHub público del proyecto frontend actualizado.*
- *Proyecto frontend actualizado comprimido.*
- *Documento ERS (propuesta terminada)*
- *Documento de cobertura de testing.*

### **Situación evaluativa 2: Presentación**

Durante la presentación, cada estudiante debe:

1. Expone cómo se diseñó el proyecto utilizando un framework de JavaScript para frontend, demostrando habilidades en la construcción de aplicaciones frontend modernas.
2. Muestra el proceso de desarrollo de componentes React, gestionando propiedades y estados para un funcionamiento correcto y personalizado, e implementando diseño responsivo mediante Bootstrap.
3. Demuestra la creación de pruebas unitarias para componentes frontend, verificando la lógica y comportamientos con Jasmine y Karma para asegurar la calidad del código.
4. Explica cómo se implementó un proceso de testeo unitario aplicando conceptos clave como la configuración del entorno de pruebas, escritura de pruebas unitarias, uso de mocks, y análisis de resultados.

## 4. Pauta de Evaluación

Tipo de Pauta: Rúbrica

| Categoría            | % logro | Descripción niveles de logro  |
|----------------------|---------|---|
| Muy buen desempeño   | 100%    | Demuestra un desempeño destacado, evidenciando el logro de todos los aspectos evaluados en el indicador.  |
| Buen desempeño       | 80%     | Demuestra un alto desempeño del indicador, presentando pequeñas omisiones, dificultades y/o errores.  |
| Desempeño aceptable  | 60%     | Demuestra un desempeño competente, evidenciando el logro de los elementos básicos del indicador, pero con omisiones, dificultades o errores.  |
| Desempeño incipiente | 30%     | Presenta importantes omisiones, dificultades o errores en el desempeño, que no permiten evidenciar los elementos básicos del logro del indicador, por lo que no puede ser considerado competente. |
| Desempeño no logrado | 0%      | Presenta ausencia o incorrecto desempeño.   |

| Indicador de Evaluación   | Categorías de Respuesta   |  |  |   |  | Ponderación<br>Indicador de<br>Evaluación |
|---|---|--|--|---|--|---|
|   | Muy buen<br>desempeño<br>100%   | Buen desempeño<br>80%  | Desempeño<br>aceptable<br>60%  | Desempeño<br>incipiente<br>30%  | Desempeño no<br>logrado<br>0%  |   |
| SITUACIÓN EVALUATIVA 1: ENCARGO. INFORME CON EVALUACIÓN GRUPAL.   |   |  |  |   |  |   |
| <b>IE2.1.1 Diseña un proyecto utilizando un framework de JavaScript para frontend, demostrando habilidades en la construcción de aplicaciones fontend.</b>  | Diseña un proyecto utilizando un framework de JavaScript con una estructura sólida y funcionalidades completas.                     | NO APLICA  | Diseña un proyecto utilizando un framework de JavaScript con algunas funcionalidades o partes estructurales incompletas. | NO APLICA   | No diseña un proyecto funcional utilizando un framework de JavaScript. | 10%                                       |
| <b>IE2.1.2 Desarrolla componentes React, gestionando las propiedades y estados del componente para su funcionamiento correcto y personalizado e implementa diseño responsivo al proyecto, mediante el framework de CSS Bootstrap.</b> | Desarrolla componentes React gestionando propiedades y estados de forma correcta, con un diseño responsivo perfectamente integrado. | Desarrolla componentes React pero con pequeñas imperfecciones en la gestión de propiedades, estados o diseño responsivo. | Desarrolla componentes React pero con varios errores en la gestión de propiedades o en el diseño responsivo.             | Desarrollo componentes deficiente con problemas significativos en propiedades, estados o diseño responsivo. | No desarrolla componentes funcionales utilizando React.                | 10%                                       |

|  |   |  |  |  |  |            |
|--|---|--|--|--|--|------------|
| <b>IE2.2.1 Crea pruebas unitarias a los componentes frontend, verificando las logicas y comportamientos de los componentes mediante Jasmine y Karma.</b>   | Crea todas las pruebas (10/10) bien estructuradas, verifican la lógica y los comportamientos esperados de los componentes sin errores.    | Crea la mayoría de las pruebas (8/10) bien estructuradas y verifican la lógica con algunos errores menores.                    | Crea la mitad de las pruebas (5/10), pero con errores notables o faltantes en la verificación de lógica y comportamientos. | Crea pocas pruebas (3/10) están presentes, con serios errores y faltas en la estructura y verificación de la lógica. | No se han creado pruebas unitarias o todas las pruebas creadas son incorrectas.        | 12%        |
| <b>IE2.3.1 Implementa un proceso de testeo unitario en el desarrollo de una aplicación web del front-end, aplicando conceptos clave como configuración del entorno de pruebas, escritura de pruebas unitarias, uso de mocks, análisis de resultados y cobertura de código.</b> | Proceso completo con entorno de pruebas bien configurado, uso efectivo de mocks, análisis detallado de resultados y cobertura exhaustiva. | Implementación sólida con buena configuración del entorno y uso de mocks, pero con algunos errores en el análisis y cobertura. | Implementación básica con entorno de pruebas y uso de mocks presentes, pero faltan análisis y cobertura adecuados.         | Configuración del entorno y uso de mocks presentes pero incompletos, con graves errores en análisis y cobertura.     | No se ha implementado el proceso de testeo unitario o la implementación es incorrecta. | 8%         |
| <b>Total Situación evaluativa 1</b>  |   |  |  |  |  | <b>40%</b> |

| SITUACIÓN EVALUATIVA 2: PRESENTACIÓN. DESEMPEÑO INDIVIDUAL.  |   |  |  |  |   |     |
|--|---|--|--|--|---|-----|
| <b>IE2.1.3 Expone cómo se diseñó un proyecto utilizando un framework de JavaScript para frontend, demostrando habilidades en la construcción de aplicaciones frontend modernas.</b>                            | Expone correctamente el diseño de un proyecto en JavaScript, mostrando habilidades avanzadas en la construcción de aplicaciones frontend modernas.      | Expone bien, pero con algunos detalles menores omitidos en la construcción de aplicaciones.                              | Expone adecuadamente pero con omisiones evidentes en técnicas o métodos de construcción. | Exposición insuficiente con errores notables en la construcción de aplicaciones frontend.                              | No expone de manera adecuada el diseño o la construcción de aplicaciones frontend modernas. | 15% |
| <b>IE2.1.4 Muestra el proceso de desarrollo de componentes React, gestionando propiedades y estados para un funcionamiento correcto y personalizado, e implementando diseño responsivo mediante Bootstrap.</b> | Muestra un proceso ejemplar de desarrollo de componentes React, con una gestión perfecta de propiedades y un diseño responsivo completamente integrado. | Muestra un proceso de desarrollo competente con pequeños errores en la gestión de propiedades o en el diseño responsivo. | Muestra un proceso con varios errores o deficiencias en la gestión o el diseño.          | Muestra un proceso de desarrollo deficiente con errores graves en la gestión de propiedades o en el diseño responsivo. | No muestra un proceso de desarrollo de componentes React efectivo o adecuado.               | 20% |



|   |  |   |   |   |  |      |
|---|--|---|---|---|--|------|
| <b>IE2.2.2 Demuestra la creación de pruebas unitarias para componentes frontend, verificando la lógica y comportamientos con Jasmine y Karma para asegurar la calidad del código.</b>                                   | El estudiante demuestra 10 pruebas unitarias, verificando claramente la lógica y comportamientos con Jasmine y Karma. La presentación es clara, organizada y muestra cómo las pruebas aseguran la calidad del código sin errores.                          | El estudiante demuestra 8 pruebas unitarias efectivas. La presentación es buena, aunque podría mejorar en algunos aspectos menores. Las pruebas contribuyen significativamente a la calidad del código. | El estudiante demuestra 6 pruebas unitarias, cubriendo parcialmente la lógica y comportamientos esenciales. La presentación tiene omisiones o errores notables, pero se entiende el concepto general. | El estudiante demuestra 3 pruebas unitarias, con grandes problemas de cobertura y precisión. La presentación es confusa, con deficiencias significativas en la explicación de cómo las pruebas mejoran el código. | El estudiante no demuestra pruebas unitarias o la demostración es incorrecta.          | 15%  |
| <b>IE2.3.2 Explica cómo se implementó un proceso de testeo unitario aplicando conceptos clave como la configuración del entorno de pruebas, escritura de pruebas unitarias, uso de mocks, y análisis de resultados.</b> | El estudiante explica de manera exhaustiva cómo se implementó el proceso de testeo unitario, detallando la configuración del entorno, escritura de pruebas, uso de mocks y análisis de resultados. La explicación es clara, detallada y bien estructurada. | El estudiante explica bien el proceso de testeo unitario, cubriendo la mayoría de los aspectos, pero con algunos detalles menores que podrían mejorarse.  | El estudiante explica el proceso de manera básica, cubriendo algunos aspectos clave, pero omite detalles importantes o presenta errores conceptuales.   | El estudiante proporciona una explicación limitada, con serias deficiencias en la comprensión de la configuración del entorno, uso de mocks y análisis de resultados. La explicación es incompleta o incorrecta.  | El estudiante no explica el proceso de testeo unitario o la explicación es incorrecta. | 10%  |
| Total Situación evaluativa 2  |  |   |   |   |  | 60%  |
| Total   |  |   |   |   |  | 100% |