

# DESARROLLO DE APLICACIONES WEB 2025-10

Lenin Javier Serrano Gil MSc.<sup>1</sup>

Facultad de Ingeniería de Sistemas e Informática  
Universidad Pontificia Bolivariana Sección Bucaramanga  
`lenin.serrano@upb.edu.co`  
20 de enero de 2025

## 1. Descripción del curso

Este curso brinda elementos de programación de últimas tecnologías necesarios para desarrollar sistemas de información web siguiendo frameworks, bases de datos relacionales y no relacionales, elementos de seguridad y de control de versiones [1].

## 2. Propósito del curso

El propósito del curso es la formación del estudiante en herramientas de desarrollo de software web con un enfoque teórico-práctico para el fortalecimiento de las habilidades y destrezas acorde a tecnologías actuales. El curso requiere de buen sentido de adaptación y responsabilidad técnica y ética [1].

## 3. Número de créditos del curso

El curso tiene un total de tres (3) créditos, equivalentes a cuarenta y ocho horas (48h) por crédito de trabajo del estudiante (decreto 1330 del 2019). Distribuidas en treinta y dos horas (32h) teóricas, treinta y dos horas (32h) de laboratorio y ochenta horas (80h) de trabajo independiente [1].

## 4. Capacidades humanas previas

- Pensamiento Crítico.
- Empatía.
- Pensamiento Sistémico.
- Autonomía.

## 5. Competencias previas

- Asume una posición crítica frente a sus ideales y convicciones, apoyado en los postulados y teorías del conocimiento de las ciencias disciplinares [1].
- Promueve la sustentabilidad en los productos que desarrolla [1].
- Propone soluciones eficaces a los problemas humanos, actuando de manera sistémica como profesional, articulándose a equipos de trabajo [1].
- Actúa con libertad, independencia, responsabilidad y sentido crítico y propositivo en la toma de decisiones, y lo evidencia en sus formas de aprender [1].

## 6. Competencias

### 6.1. Competencia 1

Propone y ejecuta proyectos de desarrollo de software en todas sus etapas del ciclo de vida para la solución de problemas organizacionales, científicos y sociales, desde los roles operativos, tácticos y estratégicos considerando la aplicación y/o formulación de criterios de viabilidad, calidad, seguridad y buenas prácticas [1].

## 6.2. Competencia 2

Aplica técnicas relacionadas con el aseguramiento de los activos de información de una organización, para gestionar los riesgos de estos a partir del estudio de modelos y estándares aplicados a la seguridad informática contemplando niveles aceptables de confidencialidad, integridad y disponibilidad [1].

## 7. Criterios de competencia

- Planteo soluciones a problemas organizacionales, científico y sociales, cumpliendo las etapas del ciclo de vida del software [1].
- Diseño soluciones computacionales con criterios de viabilidad, calidad, seguridad y buenas prácticas, que respondan a requerimientos funcionales y no funcionales [1].
- Desarrollo proyectos de software, aplicando metodologías, enfoques y herramientas de productividad [1].
- Analizo los recursos informáticos requeridos en un proyecto de desarrollo de software [1].
- Empleo técnicas relativas al aseguramiento de la información que cumplan los criterios básicos de la seguridad informática para la protección de los activos de información [1].
- Utilizo los modelos y estándares para la gestión de incidentes en seguridad informática que conlleven a la mitigación de los riesgos asociados a los activos de información [1].

## 8. Asignaturas del plan de estudios requeridas

- Diseño de Experiencia de Usuario
- Lógica de Programación
- Paradigmas de Programación
- Estructuras de Datos
- Bases de Datos
- Análisis y Diseño de Sistemas
- Ingeniería de Software
- Análisis y Diseño de Algoritmos
- Sistemas Operativos
- Redes de Datos

## 9. Metodología

El curso utiliza una dinámica participativa, a través de procesos de aprendizaje donde el desarrollo de competencias se potencia mediante el conocimiento adquirido a partir de clases magistrales y laboratorios donde se analizan situaciones en diferentes áreas. Se realiza un seguimiento del proceso de comprensión y aprehensión del conocimiento mediante actividades como evaluaciones escritas, mapas conceptuales, proyectos de aula, talleres, estudios de caso y documentos escritos (artículos, informes, o similares).

## 10. Contenido programático

Se explora un amplio espectro de tecnologías y conceptos fundamentales para el desarrollo de aplicaciones web modernas. Desde la ingeniería [2] y arquitectura [3] de software, los pilares tecnológicos HTML [4], CSS [5], JavaScript moderno [6], entornos web [7], patrones de aplicaciones web [8], Re-uso de componentes web [8], Re-factorización de aplicaciones web [9], patrones de autenticación de aplicaciones [10], patrones de persistencia [11], paradigmas de desarrollo frontend y backend [12]. Asimismo, se introduce arquitectura de aplicaciones web [13], modelo multi-capas en aplicaciones web [14], configuración de servidores de aplicaciones [15], servidores de archivos [16] y servidores de bases de datos [17]. Además, se aborda el diseño de interfaces [18], procesadores de plantillas en aplicaciones web [19] y bibliotecas para diseño de interfaces [20]. Los tópicos anteriores se apoyan en conjunto con herramientas para el desarrollo [21], ensambladores de módulos para aplicaciones web [22], gestores de dependencias y pre-procesadores [23]. Así también se exploran temas como la seguridad de las aplicaciones [24] y la implementación de buenas prácticas de desarrollo ágil [25].

## 11. Configuración y/o Framework

El curso utiliza la siguiente configuración Full Stack: **HTML/CSS/Javascript moderno + Node JS + Express.js**.

## 12. Desarrollo del trabajo académico

A continuación se presenta el desarrollo del trabajo académico en el Cuadro 1. Ajustado para el ciclo 2025-10.

**Cuadro 1.** Desarrollo del trabajo académico 2025-10

Semanas	Actividades de aprendizaje	Entregables y/o evaluación
1	Introducción al curso: Javascript Moderno.	Ejercicio práctico y/o de laboratorio.
2	Preprocesadores: Typescript.	Ejercicio práctico y/o de laboratorio.
3 y 4	Backend: Model View Controller (MVC).	Ejercicio práctico y/o de laboratorio.
4	– <b>Prueba: Seguimiento Primer Corte</b> –	
5	Backend: Clean Architecture.	Ejercicio práctico y/o de laboratorio.
6	Backend: Arquitectura de Puertos y Adaptadores.	Ejercicio práctico y/o de laboratorio.
7	Backend: Pruebas Unitarias y Pruebas de End To End.	Ejercicio práctico y/o de laboratorio.
8	– <b>Parcial: Primer Corte</b> –	
9	HyperText Markup Language (HTML).	Ejercicio práctico y/o de laboratorio.
10	Cascading Style Sheets (CSS).	Ejercicio práctico y/o de laboratorio.
11	CSS Frameworks.	Ejercicio práctico y/o de laboratorio.
12 y 13	Frontend: Model View Controller (MVC).	Ejercicio práctico y/o de laboratorio.
13	– <b>Prueba: Seguimiento Segundo Corte</b> –	
14	Backend: Model View Template (MVT).	Ejercicio práctico y/o de laboratorio.
15	Frontend: Model View Template (MVT).	Ejercicio práctico y/o de laboratorio.
16	– <b>Parcial: Segundo Corte</b> –	

## 13. Régimen de calificaciones

El sistema de calificación se define en el Cuadro 2 y se describe a continuación.

**Cuadro 2.** Régimen de calificaciones

Calificaciones	Porcentajes (%)
Seguimiento primer corte	20
Parcial primer corte	25
Seguimiento segundo corte	20
Parcial segundo corte	25
Laboratorio	10

- Todas las asignaciones de trabajo independiente se integran en la calificación de seguimiento.
- El laboratorio corresponde a las actividades prácticas desarrolladas en clase.
- Cualquier actividad de aprendizaje que no se entregue en el plazo asignado tendrá una calificación de 0.0.
- El docente tendrá la potestad de determinar si existe plagio utilizando las herramientas autorizadas de la universidad y, de ser así, la calificación por defecto será de 0,0.
- La inteligencia artificial (IA) está revolucionando la educación al ofrecer herramientas innovadoras que personalizan el aprendizaje, brindan retroalimentación instantánea y adaptan el ritmo de estudio a las necesidades de cada estudiante. Sin embargo, es fundamental usar la IA de forma responsable y ética. Recuerde que cualquier producción generada con IA debe ser citada adecuadamente y no puede ser presentada como propia, ya que esto constituye plagio.
- Por tratarse de un curso que implica evaluar competencias base en diseño, codificación y pruebas. El uso de IA estará restringido en las evaluaciones.

### Importante

La asistencia a clase es obligatoria, y será verificada en cada clase. Un índice de fallas superior al 20 % conduce a la pérdida de la materia por fallas, conllevando a una calificación de 0,0.

## 14. Parcial

Los exámenes parciales se basarán en un caso práctico que pondrá a prueba su capacidad para integrar los conocimientos adquiridos durante el curso. Para el desarrollo del caso, deberán diseñar y desarrollar una solución con una arquitectura limpia para el backend, y Modelo Vista Controlador basado en Componentes en el frontend. La evaluación incluirá:

1. Diseño: Diseño de la estructura, comportamiento, vistas, datos, arquitectura y pruebas.
2. Código fuente: Código fuente completo y documentado.
3. Base de datos: Diseño e implementación de la base de datos (libre elección).
4. Patrones: Uso de patrones y buenas practicas de desarrollo de software.

### Importante

Con el fin de evaluar la comprensión y aplicación de los conceptos fundamentales, no se permite el uso de bibliotecas externas, frameworks, código de terceros o herramientas similares que no se hayan abordado en el contenido del curso.

## 15. Caso práctico

### 15.1. Tienda Ecológica «Buena Vida»

Buena Vida es una tienda de productos naturales ubicada en un barrio tranquilo de la ciudad. En la tienda, los clientes son recibidos por un aroma a hierbas frescas y aceites esenciales. Sus estantes de madera exhiben una amplia variedad de productos:

- Alimentos orgánicos: Frutas y verduras frescas, cereales integrales, semillas, frutos secos, leches vegetales y productos sin gluten.
- Suplementos naturales: Vitaminas, minerales, extractos herbales, probióticos y superalimentos en diferentes presentaciones.
- Cosmética natural: Jabones artesanales, cremas hidratantes, aceites esenciales, champús y acondicionadores elaborados con ingredientes naturales.
- Productos para el hogar: Detergentes ecológicos, ambientadores naturales y productos de limpieza biodegradables.
- Libros y recursos: Guías de alimentación saludable, libros de recetas vegetarianas y veganas, y publicaciones sobre bienestar natural.

La tienda busca expandirse al mundo digital y llegar a un público amplio. Su objetivo es claro, diseñar y desarrollar una plataforma online que les permita exhibir y vender todos estos productos de forma atractiva, manteniendo la esencia y calidez de su tienda física. Después de varias reuniones, lo directivos han definido los siguientes requerimientos generales:

- Catálogo de productos: Se debe crear una estructura organizada para mostrar la amplia gama de productos de "Buena Vida", con imágenes de calidad, descripciones detalladas que resalten sus beneficios y precios actualizados junto con sus descuentos.
- Carrito de compras: se debe implementar un sistema que permita a los usuarios agregar productos, modificar cantidades y eliminar artículos de su selección.
- Gestión de usuarios: Se debe tener un modulo de usuarios, que permita crear cuentas, guardar sus datos de envío y realizar un seguimiento de sus pedidos.
- Buscador: En todas la vistas de la aplicación se debe tener una búsqueda de productos específicos a través de palabras clave o filtros, considerando las diferentes categorías y necesidades de los clientes.
- Diseño responsive: La página debe adaptarse fluidamente a cualquier dispositivo, garantizando una experiencia de usuario óptima en todo momento. Además, la tipografía debe ser legible. Así como, la navegación debe adaptarse a pantallas pequeñas y los menús debe reducirse a tipo hamburguesa.

- Pasarela de pago (Opcional): Integraremos un sistema seguro para procesar pagos online, con opciones como tarjetas de crédito, transferencias bancarias o plataformas de pago digital.

Por otra parte, producto de talleres con el equipo de diseño y el cliente, se ha definido los otros requerimientos en base a la vista.

## 15.2. IU-00 Logo

El logo de la tienda, ver imagen 1. Debe mantenerse en la parte superior izquierda de la web, preferiblemente en la barra superior. En resoluciones menores de 400px debe centrarse y ocupar todo el espacio superior.



Figura 1. Logo: Buena Vida.

## 15.3. IU-01 Vitrina de productos

Se solicita una vista web para para la vitrina con distribución en malla, donde se presenten los productos como muestra la imagen 2.

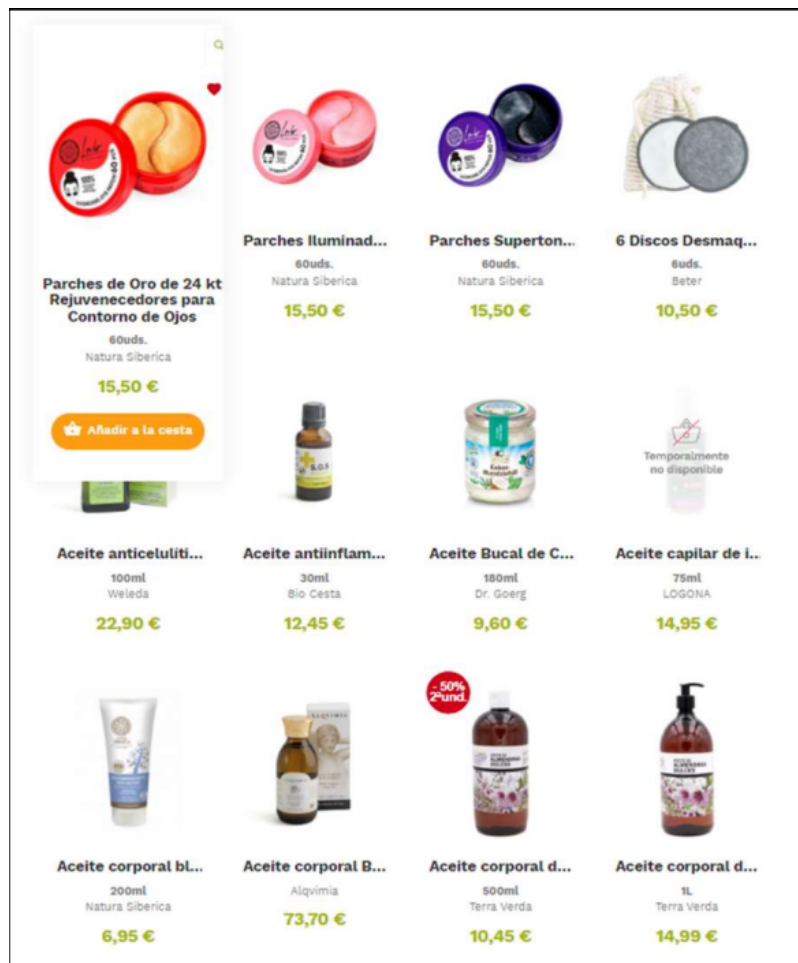


Figura 2. Vitrina de productos.

Cada página debe mostrar un total de 12 productos alineados en cuatro columnas y tres filas. Si los productos superan la vitrina deberán generarse tantas paginas como se necesiten.

#### 15.4. IU-02 Barra de paginación

Las páginas deben ser navegables a través de un control de paginación. Los productos se envían desde el servidor. La barra de navegación debe aparecer al final alineada a la derecha ver imagen 3. Si hay una página, aparece la casilla 1, si hay dos, aparecen las casillas 1 y 2, y así sucesivamente hasta llegar a la 5. Si hay más páginas a la izquierda, aparece una flecha a la izquierda para mostrar las opciones ocultas. Del mismo modo a la derecha.



**Figura 3.** Barra de paginación.

#### 15.5. IU-03 Marca distintiva de promociones

La información por producto es, nombre del producto, contenido o cantidad, y precio. Los productos en promoción deben tener un marcador circular en rojo como muestra la imagen 4.



**Figura 4.** Marca distintiva de promociones.

#### 15.6. IU-04 Relieve del producto

Si el usuario ubica el ratón sobre el producto este debe sobresaltar como se ve en la imagen 5.

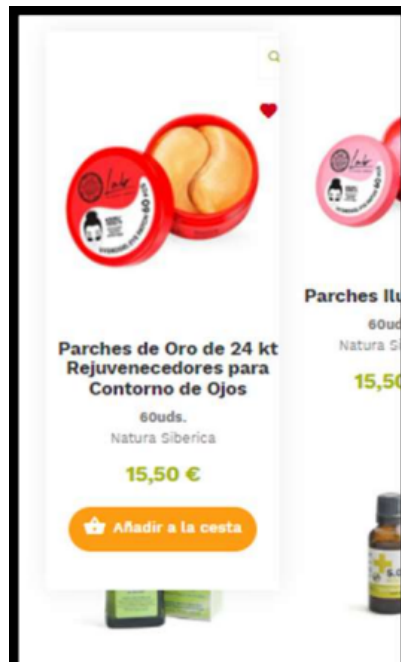


Figura 5. Relieve del producto.

## 16. IU-05 Botón para añadir productos al carro de compras y favoritos

En la vitrina si el producto se muestra en relieve la vista debe presentar un botón para añadir el producto al carrito de compras. También agregar un botón en forma de corazón en la parte superior si el usuario quiere marcar el producto como favorito, ver la imagen 6.

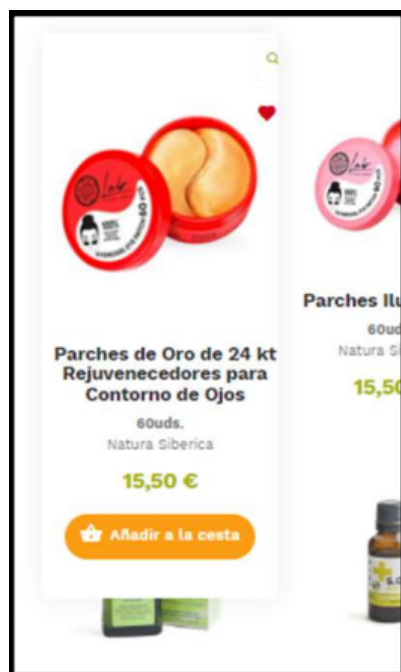


Figura 6. Botón para añadir productos al carro de compras y favoritos.

### 16.1. IU-06 Filtros de productos

La vista debe permitir filtrar los productos por precio, se debe manejar un rango y debe desplegarse como muestra la imagen 7. Estos controles deben aparecer alineados a la izquierda de

la página en paralelo a la vitrina de productos y el valor máximo de búsqueda corresponde con mayor precio disponible.



Figura 7. Filtros de productos.

## 16.2. IU-07 Evento clic y estilo en botones

Todos los botones deben resaltar cuando el ratón pase sobre ellos y cambiar de color de fondo (amarillo - verde) y texto (blanco - negro). Ver paleta de colores.

## 16.3. IU-08 Modal – detalle del producto

Si un usuario hace clic sobre un producto este debe desplegar en una ventada modal el detalle de este, la ventana debe incluir los controles para cantidad, el botón para agregar el producto al carrito de compras y el botón para cerrar la ventana, ver imagen 8. Además, incluir en la parte superior derecha el icono (corazón) para agregar el producto a favoritos.



Figura 8. Modal – detalle del producto.

## 16.4. IU-09 Barra de búsqueda y navegación

Se debe tener una barra de navegación en la parte superior alienada a la derecha donde se pueden buscar los productos, ir a la página de “Mis Favoritos”, ir al menú de “Mi Cuenta” (IU-11), y un icono para ver un resumen flotante (IU-10) de “Mi carrito”, ver imagen 9. Todos los iconos deben ser incluidos. Note que el icono del carrito debe mostrar la cantidad de productos que se ha agregado al carrito de compras (IU-11). “Mis Favoritos” muestra los productos favoritos en la misma distribución de la vitrina. “Entrar” despliega un menú que incluye como acceder con un usuario registrado con email y contraseña.



Figura 9. Barra de búsqueda y navegación.



### 16.5. IU-10 Resumen flotante “Mi Carrito”

El evento clic debe desplegar una ventana flotante que contenga la información distribuida como lo muestra la imagen 10, incluyendo los controles de cantidad, los botones ir al carrito y realizar el pedido.

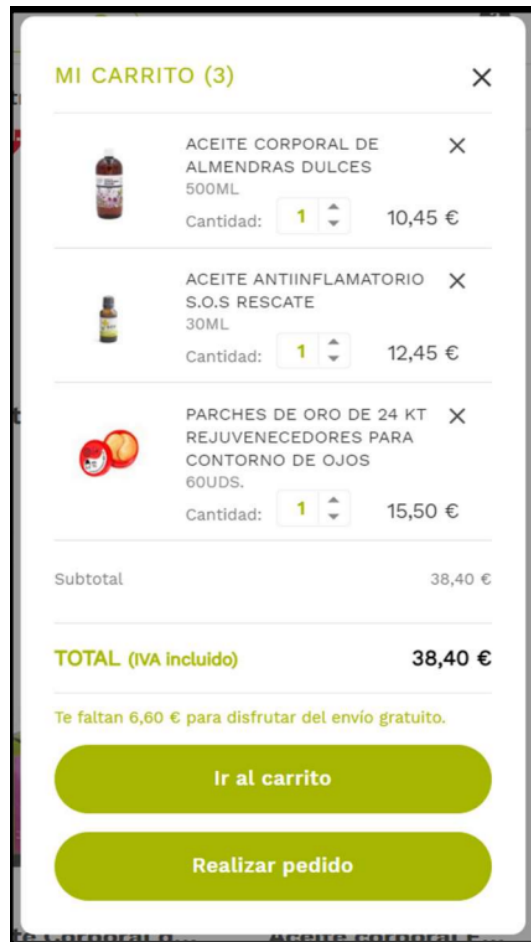


Figura 10. Resumen flotante “Mi Carrito”.

### 16.6. IU-11 Menú de “Mi Cuenta”

El botón “Mi Cuenta” debe desplegar el siguiente menú descrito por la imagen 11. Todos los iconos desben ser incluidos.

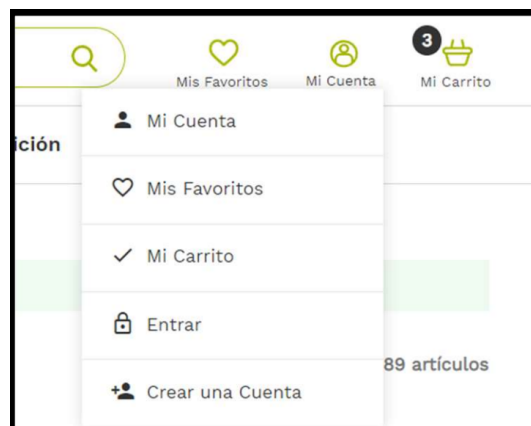


Figura 11. Menú de “Mi Cuenta”.

### 16.7. IU-12 Entrar (inicio de sesión)

El usuario registrado podrá entrar en sesión usando nombre de usuario y contraseña como lo muestra la imagen 12. El formulario debe validar los campos e implementar sistemas de seguridad. También podrá entrar al sistema utilizando un enlace para acceder con un tercero (Opcional). Por ejemplo, Facebook mediante un botón azul como el presentado en la ilustración.

**Figura 12.** Entrar (inicio de sesión).

#### Requerimiento

El control de acceso estará basado en roles. Y deberá implementarse como se describe en las siguientes especificaciones:

- Role-Based Access Control. [Clic aquí.](#)
- Proposed NIST standard for role-based access control. [Clic aquí.](#)

### 16.8. IU-13 Ir a registro de usuario

Si el usuario no está registrado, una opción de registro o botón en la ventana de “Entrar” lo lleva a un formulario de registro, ver imagen 11.

### 16.9. IU-14 Registro de usuario

El nuevo usuario se registra con los campos nombres, apellidos, email y contraseña. Los campos deben validar texto, formato del email y más de 8 caracteres entre alfabéticos, mayúsculas, minúsculas, números y símbolos en la contraseña, como se ve en la imagen 13.



Crear nueva cuenta

Nombre

Apellido

Información de inicio de sesión

Email

Contraseña

Fuerza de la contraseña: 5/5 contraseña

Repetir contraseña

La contraseña debe contener al menos una mayúscula y un número

☐ No soy un robot

Crear cuenta

Accede con Facebook

Figura 13. Registro de usuario.


## 16.10. Carrito de compras

El carrito de compras es una vista que se recupera desde el botón "Ir al carrito" de la UI-10. Muestra los productos y la información. También, controles para gestionar la cantidad (selector) y eliminar productos (ícono "basura"). Además, el botón realizar pedido o el enlace para volver a la tienda "¿Quieres añadir más productos?". ver imagen 14.



Esta es tu cesta de la compra

3 Artículos:

Producto	Cantidad	Precio	Eliminar
 <p>PARCHES SUPERTONIFICANTES PARA CONTORNO DE OJOS Talla: 60uds. Marca: Natura Siberica</p>	1	15,50 €	
 <p>ACEITE CORPORAL DE ALMENDRAS DULCES Talla: 500ml Marca: Terra Verda</p>	1	10,45 €	
 <p>ACEITE CORPORAL DE ROSA MOSQUETA Talla: 100ml Marca: Weleda</p>	1	22,90 €	

Resumen de tu pedido:

SUBTOTAL 48,85 €

TOTAL: 48,85 €  
(Envío no incluido)

Realizar pedido

[¿Quieres añadir más productos?](#)

Figura 14. Carrito de compras.

### 16.11. IU-16 Paleta de colores

La paleta base de colores ha cambiado con respecto al diseño presentado. En la imagen 15 se ilustra las nueve versiones de colores para la implementación final.



**Figura 15.** Paleta de colores.

#### Nota

Las imágenes se tomaron de recursos reales en internet y están sujetas a derechos de autor, sin embargo, para efectos del presente ejercicio no se citan las fuentes.

#### Importante

El estudiante debe realizar todo el ciclo de vida para un producto software. Esto implica que debe analizar los requerimientos, eliminar ambigüedades y concretar en un documento de especificación. Por otra parte, este trabajo es un ejercicio diseñado para demostrar sus competencias. Debe desarrollarse en paralelo con los contenidos del curso. Además, hay que precisar que no todos los elementos de un sistema de e-commerce se han incluido.

## 17. Parcial: Primer Corte - Backend

Esta evaluación práctica se centra en el desarrollo e implementación del backend del caso práctico. La evaluación constará de cuatro componentes:

1. Entrega del backend (**0.5 puntos**): Deberán entregar el código fuente completo del backend de la aplicación, incluyendo la lógica de negocio, la gestión de datos y la API (Interfaz de Programación de Aplicaciones) que permitirá la comunicación con el frontend. El código debe estar bien estructurado, documentado y cumplir con los estándares de calidad definidos en el curso. Se evaluará en la fecha del parcial, la correcta implementación de las funcionalidades, la estructura del código y el manejo adecuado de los datos.
2. Test del producto (**1.0 puntos**): Se realizarán pruebas exhaustivas al backend de la aplicación para verificar su correcto funcionamiento. Se evaluará la capacidad del sistema para procesar solicitudes, gestionar datos, responder a diferentes escenarios y manejar errores. Se utilizarán casos de prueba específicos para comprobar el cumplimiento de los requisitos funcionales y no funcionales.

3. Revisión del diseño (**1.0 puntos**): Se revisará la documentación del diseño del backend, incluyendo la arquitectura, la estructura de datos, los diagramas y las decisiones de diseño. Se evaluará la claridad, la coherencia y la completitud de la documentación, así como la justificación de las decisiones de diseño tomadas.
4. Sustentación (**2.5 puntos**): Se evaluará los aspectos técnicos del proyecto, la comprensión del problema y la solución implementada. Se responderán preguntas sobre el diseño, la implementación y el funcionamiento del backend.

## 18. Parcial: Segundo Corte - Frontend

Esta evaluación práctica se centra en el desarrollo e implementación del frontend del caso práctico. La evaluación constará de cuatro componentes:

1. Entrega del frontend (**0.5 puntos**): Deberán entregar el código fuente completo del frontend de la aplicación, incluyendo el HTML, CSS y JavaScript que definen la estructura, el estilo y el comportamiento de la interfaz de usuario. El código debe estar bien organizado, documentado y cumplir con los estándares de calidad definidos en el curso, priorizando la accesibilidad y la usabilidad. Se evaluará en la fecha del parcial, la correcta implementación de las funcionalidades, la calidad del diseño visual, la resiliencia del código y la experiencia de usuario.
2. Test del producto (**1.0 puntos**): Se realizarán pruebas exhaustivas al frontend de la aplicación para verificar su correcto funcionamiento en diferentes navegadores y dispositivos. Se evaluará la capacidad de la interfaz para mostrar la información correctamente, responder a las interacciones del usuario, manejar eventos y gestionar errores. Se utilizarán casos de prueba específicos para comprobar el cumplimiento de los requisitos funcionales y no funcionales, incluyendo la usabilidad, la accesibilidad y la responsividad.
3. Revisión del diseño (**1.0 puntos**): Se revisará la documentación del diseño del frontend, incluyendo la arquitectura de la información, los prototipos y las decisiones de diseño. Se evaluará la claridad, la coherencia y la completitud de la documentación, así como la justificación de las decisiones de diseño tomadas en términos de usabilidad y experiencia de usuario.
4. Sustentación (**2.5 puntos**): Se evaluará los aspectos técnicos y de diseño del proyecto, la comprensión del problema y la solución implementada. Se responderán preguntas sobre el diseño, la implementación, la usabilidad y el funcionamiento del frontend.

## 19. Pruebas/Talleres

Para evaluar el aprendizaje de forma integral, utilizaremos Pruebas/Talleres que combinarán preguntas teóricas y prácticas. A través de diferentes tipos de preguntas, como opción múltiple, verdadero/falso y ejercicios de aplicación. Se evaluará la comprensión de los conceptos y la capacidad para aplicarlos en situaciones reales (**3.5 puntos**). Además, parte de la evaluación incluye un video de máximo 7 minutos acerca de un tema relacionado con las lecturas recomendadas para cada corte (**1.5 puntos**).

### Requerimiento

El estudiante debe grabar sus videos y publicarlos en la plataforma disponible por la universidad. Deben entregarse en las fechas correspondientes a las pruebas. El trabajo debe contener una introducción, el desarrollo del tema y conclusiones. Se debe ver en el recurso el estudiante. Por otra parte, el estudiante debe citar los textos o imágenes generados con de inteligencia artificial, que no podrá presentar como propios.

### 19.1. Prueba: Seguimiento Primer Corte

El examen incluirá un ejercicio práctico que aborda los temas desarrollados en las semanas anteriores. Además, de una presentación acerca de los temas asociados con las lecturas propuestas.

#### Design Patterns. Erich Gamma. et al. [8]

- Abstract Factory.
- Builder.
- Factory Method.

- Prototype.
- Singleton.
- Adapter.
- Bridge.
- Composite.
- Decorator.
- Facade.
- Flyweight.
- Proxy.
- Chain of Responsibility.
- Command.
- Interpreter.
- Iterator.
- Mediator.
- Memento.
- Observer.
- State.
- Strategy.
- Template Method.
- Visitor.

### **Clean Code. Robert Martin [26]**

- Chapter 1: Clean Code.
- Chapter 2: Meaningful Names.
- Chapter 3: Functions.
- Chapter 4: Comments.
- Chapter 5: Formatting.
- Chapter 6: Objects and Data Structures.
- Chapter 7: Error Handling.
- Chapter 8: Boundaries.
- Chapter 9: Unit Tests.
- Chapter 10: Classes.
- Chapter 11: Systems.
- Chapter 12: Emergence.
- Chapter 13: Concurrency.
- Chapter 14: Successive Refinement.
- Chapter 15: JUnit Internals.
- Chapter 16: Refactoring SerialDate.
- Chapter 17: Smells and Heuristics.

### **19.2. Prueba: Seguimiento Segundo Corte**

El examen incluirá un ejercicio práctico que aborda los temas desarrollados en las semanas anteriores. El estudio del Desarrollo de Aplicaciones Web es un camino forjado por conceptos apilados, no se puede dejar atrás los temas del primer corte. Además, la prueba evalúa una presentación acerca de los temas asociados con las lecturas propuestas.

### **Test-Driven Development By Example. Kent Beck [27]**

- Chapter 1. Multi-Currency Money.
- Chapter 2. Degenerate Objects.
- Chapter 3. Equality for All.
- Chapter 4. Privacy.
- Chapter 5. Franc-ly Speaking.
- Chapter 6. Equality for All, Redux.
- Chapter 7. Apples and Oranges.
- Chapter 8. Makin' Objects.
- Chapter 9. Times We're Livin' In.
- Chapter 10. Interesting Times.
- Chapter 11. The Root of All Evil.

- Chapter 12. Addition, Finally.
- Chapter 13. Make It.
- Chapter 14. Change.
- Chapter 15. Mixed Currencies.
- Chapter 16. Abstraction, Finally.
- Chapter 17. Money Retrospective.
- Chapter 18. First Steps to xUnit.
- Chapter 19. Set the Table.
- Chapter 20. Cleaning Up After.
- Chapter 21. Counting.
- Chapter 22. Dealing with Failure.
- Chapter 23. How Suite It Is.
- Chapter 24. xUnit Retrospective.
- Chapter 25. Test-Driven Development Patterns.
- Chapter 26. Red Bar Patterns.
- Chapter 27. Testing Patterns.
- Chapter 28. Green Bar Patterns.
- Chapter 29. xUnit Patterns.
- Chapter 30. Design Patterns.
- Chapter 31. Refactoring.
- Chapter 32. Mastering TDD.

## Refactoring Improving the Design of Existing Code. Martin Fowler and Kent Beck [9]

- Chapter 1 Refactoring: A First Example.
- Chapter 2 Principles in Refactoring.
- Chapter 3 Bad Smells in Code.
- Chapter 4 Building Tests.
- Chapter 5 Introducing the Catalog.
- Chapter 6 A First Set of Refactorings.
- Chapter 7 Encapsulation.
- Chapter 8 Moving Features.
- Chapter 9 Organizing Data.
- Chapter 10 Simplifying Conditional Logic.
- Chapter 11 Refactoring APIs.
- Chapter 12 Dealing with Inheritance.

## 20. Enlaces de interés

### 20.1. HTML

- W3C - Standard <https://html.spec.whatwg.org/multipage/>
- MDN - Mozilla <https://developer.mozilla.org/en-US/docs/Web/HTML>
- W3schools - HTML <https://www.w3schools.com/html/default.asp>

### 20.2. CSS

- W3C - Standard <https://www.w3.org/Style/CSS/>
- MDN - Mozilla <https://developer.mozilla.org/en-US/docs/Web/CSS>
- W3schools - CSS <https://www.w3schools.com/css/default.asp>
- CSS Tricks <https://css-tricks.com/>
- Flexbox <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- Grid <https://css-tricks.com/snippets/css/complete-guide-grid/>
- Sass <https://sass-lang.com/guide>

### 20.3. Frameworks CSS

- Bootstrap <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- Bootswatch - Templates <https://bootswatch.com/>
- Tailwind <https://tailwindcss.com/docs>

## 20.4. JavaScript

- ECMAScript - Standard <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>
- MDN - Mozilla <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- W3schools - JavaScript <https://www.w3schools.com/js/default.asp>

## 20.5. TypeScript

- TypeScript <https://www.typescriptlang.org/docs/>
- W3schools - TypeScript [https://www.w3schools.com/typescript/typescript\\_intro.php](https://www.w3schools.com/typescript/typescript_intro.php)

## 20.6. Design patterns

- Design patterns <https://refactoring.guru/design-patterns>
- Design patterns - java [https://www.tutorialspoint.com/design\\_pattern/index.htm](https://www.tutorialspoint.com/design_pattern/index.htm)

## 20.7. Documentación en general

- DevDocs <https://devdocs.io/>.

## Referencias

1. UPB, "Carta descriptiva del curso: Desarrollo de aplicaciones web. código da-fo-085n. version 3," 2020.
2. I. Sommerville, M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts, "Edition: Software engineering," *Instructor*, 2019.
3. D. Vieira, "Designing hexagonal architecture with java: An architect's guide to building maintainable and change-tolerant applications with java and quarkus," 2022.
4. C. Musciano and B. Kennedy, *HTML & xhtml: The definitive guide: The definitive guide*. O'Reilly Media, Inc., 2002.
5. E. A. Meyer, *CSS: The Definitive Guide: The Definitive Guide*. O'Reilly Media, Inc., 2006.
6. M. Delamater, *Modern JavaScript: Beginner to Pro*. Murach's, 2023.
7. J. L. Berenguel Gómez, *Desarrollo de aplicaciones web en el entorno servidor*. Ediciones Paraninfo, SA, 2024.
8. K. Beck, "Design patterns: Elements of reusable object-oriented software," *IBM Systems Journal*, vol. 34, no. 3, p. 544, 1995.
9. M. Fowler, *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 2018.
10. S. K. Dash, *Ultimate Web Authentication Handbook: Strengthen Web Security by Leveraging Cryptography and Authentication Protocols such as OAuth, SAML and FIDO*. Orange Education Pvt Ltd, 2024.
11. M. Casciaro and L. Mammino, *Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques*. Packt Publishing Ltd, 2020.
12. E. Sarrion, *JavaScript from Frontend to Backend: Learn full stack JavaScript development using the MEVN stack with quick and easy steps*. Packt Publishing Ltd, 2022.
13. M. Polo Usaola, *Arquitectura, diseño y programación de Aplicaciones Web*. Universidad de Castilla-La Mancha, 2024.
14. F. Sierra, J. Acosta, J. Ariza, and M. Salas, "Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web," *Investigación y desarrollo en TIC*, vol. 4, no. 2, pp. 14–26, 2013.
15. D. DeJonghe, *Nginx cookbook*. O'Reilly Media, 2020.
16. D. J. Harkness, "Apache essentials," *Friendsof ED*, 2004.
17. E. Kenler and F. Razzoli, *MariaDB Essentials*. Packt Publishing Ltd, 2015.
18. J. Tidwell, *Designing interfaces: Patterns for effective interaction design*. O'Reilly Media, Inc., 2010.
19. D. Flanagan, "Javascript: the definitive guide: master the world's most-used programming language," (*No Title*), 2023.
20. B. Frain, *Responsive Web Design with HTML5 and CSS: Build future-proof responsive websites using the latest HTML5 and CSS techniques*. Packt Publishing Ltd, 2022.
21. B. Johnson, *Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers*. John Wiley & Sons, 2019.
22. R. C. Martin, "Clean architecture: A craftsman's guide to," *Available online [link]*, 2017.
23. F. Rappl, *Modern Frontend Development with Node.js: A compendium for modern JavaScript web development within the Node.js ecosystem*. Packt Publishing Ltd, 2022.
24. M. McDonald, *Web security for developers: real threats, practical defense*. No Starch Press, 2020.
25. K. S. Rubin, *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley, 2012.
26. R. C. Martin, *Clean code: a handbook of agile software craftsmanship*. Pearson Education, 2009.
27. K. Beck, *Test driven development: By example*. Addison-Wesley Professional, 2022.