

**PROGRAMACIÓN WEB DE CAPA INTERMEDIA**

**Programación Web de Capa Intermedia**



# Diego Alan Adame García

AGOSTO – DICIEMBRE 2024

**Descripción de la rúbrica**

**En esta Unidad de Aprendizaje, cada una de las rúbricas tiene una ponderación interna que define la calificación parcial del estudiante.**

La ponderación global de cada una de las rúbricas define la calificación final del estudiante.

**Ponderación de cada rúbrica**

**Competencia Práctica (CP): Para acreditar la CP deberá obtener una calificación igual o mayor a 70 puntos de promedio final, asimismo tiene que cumplir con todos los puntos descritos como requisitos en la Lista de Chequeo, los cuales definen la ponderación del mismo.**

**Las actividades opcionales no representan faltas en el cumplimiento de la CP, solo descuentan la calificación correspondiente al proyecto en la cantidad de puntos especificada.**

**Las actividades marcadas como requisito que no se hayan realizado de la manera especificada determinan que el proyecto está incompleto y no puede ser acreditado para tomar la CP como aprobatoria. Los ejercicios y tareas realizados a lo largo del curso serán solo para poder otorgar retroalimentación al alumno y ayudarlo con su desempeño; y no entrarán en la ponderación de la Forma de Evaluación elegida.**



**Definición del Proyecto Integrador de Aprendizaje**

|  |  |  |
| --- | --- | --- |
| **Características Generales** | | |
| **Integrantes** | **2 (del mismo grupo)** | |
| **Roles** | | |
| Programador web | Diseño de la página web, estilos, funcionalidad, validaciones. Desarrollo en HTML, JavaScript, CSS y PHP con POO (programación orientada a objetos) y MVC (Model View  Controller). | Nombre |
| Base de datos. | Diseño de la base de datos, su estructura, relaciones, creación de tablas. Desarrolla la programación de procedimientos  almacenados, vistas, funciones y disparadores. En MySQL. | Nombre |

**Descripción del proyecto**

|  |  |
| --- | --- |
| **Funcionalidad** | |
| Aplicación de compras en línea | Una aplicación web que simule una tienda en línea que permita a los usuarios comprar, vender y solicitar cotizaciones de productos publicados que son autorizados previamente por un administrador. |
| Registro de usuarios | Crear una página para el registro de usuarios.  Deberá desarrollar las llamadas a una API necesarias para crear, eliminar y modificar un usuario:   * Correo electrónico (único). * Nombre de usuario con mínimo 3 caracteres (único) * Contraseña con un mínimo de 8 caracteres, una mayúscula, una minúscula, un número y un carácter especial. * Rol de usuario. * Imagen tipo avatar. * Nombre Completo * Fecha de nacimiento * Sexo * Fecha de ingreso al portal   Carácter especial = cualquier símbolo generado por el teclado que no sea letra ni número, por ejemplo: (¡”#$%&amp;/=’?¡¿:;,.-\_+\*{][})  Se deberán hacer las validaciones necesarias para que el usuario se registre a la aplicación web. |



|  |  |
| --- | --- |
|  | Los usuarios podrán seleccionar si quieren ser públicos (para quienes quieren vender o comprar, o si son administradores) o privados (para quienes únicamente quieren comprar).  Se deben manejar tres roles para los usuarios, uno para los vendedores, otro rol de usuario para los clientes y otro para los administradores, este  último creado por un único superadministrador que es quien tiene control completo de los aspectos de la aplicación. |
| Inicio de sesión | Se deberá generar las llamadas a una API necesarias para iniciar y terminar una sesión. Se podrá acceder a la aplicación mediante nombre de usuario o correo, y contraseña.  Además, el usuario podrá elegir la opción de ser recordado tras cerrar su navegador. |
| Creación de listas | Cada usuario podrá crear listas, por ello deberemos generar las llamadas necesarias para crear, eliminar y modificar listas.   * Nombre de lista * Descripción * Imágenes (opcional)   Pública o Privada (es decir, si es una lista de compras personal o es una lista de deseos para que alguien pueda regalar el producto a esa persona). |
| Agregar objetos a la lista | Las listas podrán ser llenadas con productos requeridos según los usuarios necesiten, estos podrán ser de distintos ámbitos tal como videojuegos, música, arte, ropa, figuras, etc. Deberá contener lo siguiente:   * Nombre de objeto * Descripción * Precio (si lo tiene publicado) |
| Perfil de usuario | Cada usuario registrado deberá contar con un perfil de usuario mostrando cada uno su foto de perfil y nombre de usuario, pero el resto de su contenido dependerá de su rol y características:   * Si es privado, no mostrar contenido y mostrar únicamente su foto de perfil y nombre de usuario, así mismo, mostrar explícitamente que el perfil es privado con un mensaje en pantalla. * Si es público, mostrar las listas públicas que tengan con el contenido de los productos que desean. * Si es vendedor, mostrar los productos publicados que previamente fueron autorizados por un administrador.   Si es administrador, mostrar los productos que fueron autorizados por este mismo. |
| Categorías | Cada producto debe pertenecer al menos a una categoría, el vendedor puede crear categorías o se pueden agregar productos a las categorías que otros agregaron previamente.  A las categorías se les puede agregar una descripción donde explique el tipo de productos que contendrá.  Es importante guardar qué usuario creó qué categoría. |



|  |  |
| --- | --- |
| Productos | Los productos publicados por vendedores deberán estar previamente autorizados por un administrador, estos deberán contener los siguientes elementos:   * Nombre. * Descripción. * Imágenes (mínimo: 3) * Video (mínimo: 1) * Categoría (al menos una) * Si es para cotizar o para vender. * Precio (siempre y cuando sea para vender). * Cantidad disponible. (si no hay disponibilidad, mostrar un mensaje) * Valoración (sistema de 5 estrellas, puntuación del 1 al 10, etc.). * Comentarios.   Busca la forma de siempre tener productos Relacionados del mismo vendedor para ofrecer en algún momento de la navegación o de la  compra. |
| *Landing page* | Habrá una página principal que da la bienvenida de manera atractiva y acorde a la temática de la aplicación web. Esta pantalla permitirá el registro o acceso a la aplicación para los usuarios que quieran comprar o  solicitar cotizaciones (recordar que solo los usuarios registrados pueden hacer uso de la aplicación). |
| Página de inicio | Una vez iniciada sesión la página de inicio deberá mostrar en pantalla los productos más vendidos, recomendados o populares, todo dividido en secciones para que los clientes puedan interesarse en los diferentes  productos existentes. |
| Barra lateral, superior u  otro elemento de navegación | Se incluirá una barra lateral, superior u otro elemento que permita la  navegación dentro de las secciones de la página. Este elemento estará presente en la mayoría de las pantallas de la aplicación. |
| Buscador | Agregar un buscador de contenidos o usuarios con filtro de búsqueda (menor/mayor precio, mejor calificados, más/menos vendidos, etc.).  Se debe buscar por parte del nombre o nombre completo, ya sea del producto o del usuario, y haciendo las búsquedas por las combinaciones  posibles en cualquier filtro. |
| Middleware | La API podrá usar algún middleware a su preferencia y según el lenguaje  de desarrollo lo permita. |
| Métodos de pago | La página deberá contar con un sistema de compra/venta de productos que se tenga en existencia y con un precio fijado, con 2 métodos de pago  como mínimo, se puede implementar un sistema de compra y pago por medio de PayPal (simulado con el modo *sandbox*), tarjeta o similares. |
| Carrito de Compras | Al momento de estar navegando consultado los productos, en cualquier momento se pueden agregar a un carrito de compras, incluyendo la cantidad del producto que se desea comprar, esa cantidad no debe ser mayor a la cantidad disponible.  Los productos en el carrito de compras se pueden modificar en cualquier momento. |



|  |  |
| --- | --- |
|  | Una vez que se desea realizar la compra, se debe proceder al pago y posterior a esto el usuario comprador debe calificar y hacer un comentario acerca de cada producto comprado.  Luego de la compra el carrito debe quedar vacío para volverlo a llenar en futuras compras.  El carrito de compras siempre debe estar disponible hasta que el mismo usuario lo borre o se realice la compra.  Se debe guardar siempre la fecha y hora en la que se realiza la compra. |
| Sistemas de cotización | El vendedor podrá publicar sus productos sin precio (previamente autorizados por un administrador), para que estos puedan ser vistos por los compradores interesados y solicitar una cotización de este o ser negociados, todo esto se hará mediante un sistema de mensajería o chat  según sea conveniente. |
| Consulta de Ventas | Para los usuarios con rol de Vendedor, se deberá agregar una página donde se les permita consultar las ventas realizadas, las ventas se podrán consultar por un rango de fechas y/o una categoría (una o todas). Los resultados de la consulta se deberán de mostrar de dos formas:   1. Consulta detallada:    * Fecha y hora de la venta    * Categoría    * Producto    * Calificación    * Precio    * Existencia actual (al momento de realizar la consulta). 2. Consulta agrupada:    * Mes-Año de la venta    * Categoría    * Ventas   Además, se debe contar con una pantalla donde se muestre un listado de los productos actuales y la existencia disponible, se hará el filtro por categoría o sin filtro mostraría todos los productos que el vendedor ha  agregado. |
| Consulta de Pedidos realizados | Los usuarios con rol de Comprador podrán consultar todas las compras que hayan realizado, el filtro para la consulta debe ser un rango de fechas y/o Categoría (una o todas). Los datos para mostrar son:   * Fecha y hora de compra * Categoría * Producto * Calificación * Precio |
| Host  (Opcional) | Es opcional que el proyecto esté alojado en un servicio de Web Hosting  para la revisión de la funcionalidad. |



|  |  |
| --- | --- |
| Entrega final del proyecto | Se debe de entregar en un archivo .**rar o .zip** y subir a *Teams* lo siguiente:   * Todos los archivos de la aplicación web (PHP, HTML, CSS, JavaScript, etc.) y demás archivos necesarios para el funcionamiento de la aplicación. * Los scripts en MySQL de la creación de BD y la creación de tablas en un solo archivo texto .sql * Los scripts en MySQL de toda la programación (procedimientos almacenados, disparadores, vistas, funciones, carga de información, etc.) en uno o varios archivos texto .sql   Documento en Word para la implementación y operación del proyecto. |
| **Restricciones del proyecto** | Para el desarrollo del proyecto en PHP y en MySQL se deben tomar en cuenta de forma obligatoria los requisitos que se muestran en el Anexo 1. Lista de Cotejo.  **(OBLIGATORIA)** |
| **Revisión del proyecto** | Estar atentos a la fecha y hora de la revisión presencial y tener listo el proyecto terminado y probado para mostrarlo al maestro capturando y ejecutando lo que se le indique, además de responder todas las preguntas que se le realicen con respecto a cualquier tema del desarrollo y la construcción del proyecto y de la base de datos.  El proyecto debe cumplir los elementos establecidos en la lista de cotejo y apegarse a las restricciones definidas en este mismo documento, ya que al no incluirlas en el proyecto será causa de no aprobación.   * ​ |

|  |  |
| --- | --- |
| Entregas para el semestre | |
| Entregas | **Primera Entrega Obligatoria:** *Se subirá a la plataforma definida por el maestro un archivo .zip o .rar con el nombre y matrícula del alumno,el cual contendría el proyecto completo con sus siguientesfuncionalidades:*   * Ventanas desarrolladas * Validaciones en JS o JQuery * Incluye todos los elementos que se solicitan en cuanto a funcionalidad. * Diseño del modelo relacional de la base de datos. * Propuesta para las API creadas desde 0 * Incluye hojas de estilo (CSS) * Incluye Bootstrap o algún otro framework de estilo |



|  |  |
| --- | --- |
|  | * Diseño de estilo propio, sin utilizar plantillas. * Incluye datos dummy.   **Segunda Entrega Obligatoria**, *Se subirá a la plataforma definida por el maestro un archivo .zip o .rar con el nombre y matrícula del alumno,el cual contendría el proyecto completo con sus siguientesfuncionalidades:*   * Propuesta para al menos 1 middleware * Al menos 1 API propia aplicada al avance * Landing page atractiva y acorde a la temática. * Página para el Registro, modificación y consulta del perfil del usuario, por rol de usuario. * Validación de requisitos de contraseña para el perfil de usuario. * Propuesta de API de terceros. * Propuesta para funcionamiento de al menos 1 función AJAX en el proyecto. * Clase para la conexión a la base de datos en PHP.   **Tercera Entrega Obligatoria:** *Se subirá a la plataforma definida por el maestro un archivo .zip o .rar con el nombre y matrícula del alumno,el cual contendría el proyecto completo con sus siguientes funcionalidades correspondientes al proyecto totalmente finalizado incluyendo las funcionalidades pendientes en la última entrega.* |



**Lista de chequeo de características a evaluar para el 1° avance. Valor de: 100pts. 30% cal total**

Matrícula: Nombre: Calif:

|  |  |  |  |
| --- | --- | --- | --- |
| **REQ** | **Cumple** | **Puntos** | **A Evaluar** |
| Si |  | 20 | El diseño de la todas las páginas en HTML. |
| Si |  | 10 | Validaciones en JavaScript. |
| Si |  | 10 | Incluye todos los elementos que se solicitan en cuanto a funcionalidad. |
| Si |  | 25 | Páginas responsivas. |
| Si |  | 10 | Propuesta para las API creadas desde 0 |
| Si |  | 10 | Incluye hojas de estilo (CSS) |
| No |  | 5 | Incluye Bootstrap o algún otro *framework* de estilo |
| Si |  | 5 | Diseño de estilo propio, sin utilizar plantillas. |
| No |  | 5 | Incluye datos *dummy*. |

# Puntos totales recibidos en 1ra entrega:



**Lista de chequeo de características a evaluar para el 2° avance. Valor de: 100pts. 30% cal total**

Matrícula: Nombre: Calif:

|  |  |  |  |
| --- | --- | --- | --- |
| **REQ** | **Cumple** | **Puntos** | **A Evaluar** |
| Si |  | 10 | Propuesta para al menos 1 middleware |
| Si |  | 25 | Al menos 1 API propia aplicada al avance |
| Si |  | 10 | Landing page atractiva y acorde a la temática. |
| No |  | 15 | Página para el Registro, modificación y consulta del perfil del usuario, por rol de usuario. |
| Si |  | 10 | Validación de requisitos de contraseña para el perfil de usuario. |
| Si |  | 10 | Propuesta de API de terceros. |
| No |  | 10 | Propuesta para funcionamiento de al menos 1 función AJAX en el proyecto. |
| No |  | 10 | Clase para la conexión a la base de datos en PHP. |

# Puntos totales recibidos en 2da entrega:



**Lista de chequeo de características a evaluar para la 3ra entrega y final. Valor de: 100pts. 40% cal total**

Matrícula: Nombre: Calif:

|  |  |  |  |
| --- | --- | --- | --- |
| **REQ** | **Cumple** | **Puntos** | **A Evaluar** |
| Si |  | 10 | APIs propias y de terceros aplicadas al proyecto |
| Si |  | 10 | Gestión de Usuarios. Inicio de Sesión. |
| Si |  | 10 | Gestión de Productos. |
| Si |  | 10 | Gestión de Categorías.  Creación de Listas, agregar objetos. |
| Si |  | 5 | Página de inicio, barra de navegación. |
| Si |  | 10 | Carrito de Compras |
| Si |  | 10 | Manejo de cotizaciones |
| Si |  | 5 | Buscador |
| Si |  | 10 | Consulta de Ventas.  Consulta de Pedidos realizados. |
| Si |  | 0 | Validaciones en toda la operación del proyecto (se descontará en cada  rubro) |
| No |  | 5 | Al menos 1 método de pago usando PayPal en modo sandbox o parecidos. |
| No |  | 10 | Web Hosting. |
| No |  | 5 | 1 función AJAX aplicada al proyecto. |

# Puntos totales recibidos en 3ra entrega:



**Ponderación de la CP para el promedio final**

|  |  |
| --- | --- |
| **Elemento de evaluación** | **Ponderación** |
| Proyecto | 100 |
| **Calificación Final** | **100%** |

**La ponderación de la CP es el 100% de la calificación final.**

**Reglamento**

# De la conducta:

Se tomará asistencia al inicio de cada clase a criterio del profesor. La buena asistencia no provee puntos a favor ni la inasistencia genera reprobación solo es un registro para control.

Se debe tratar con respeto a maestros y compañeros independientemente de la plataforma que se utilice en el grupo.

El chat del grupo será usado exclusivamente para tratar temas del curso.

Durante las sesiones se les pide silenciar micrófono y cámara. Si el alumno tiene una pregunta deberá notificarlo en el chat de la sesión. El maestro definirá un horario para responder dichas preguntas y procederán de forma en la que hayan sido enviadas.

La conducta inapropiada será́ reportada a la Coordinación de la carrera.

Se deberá firmar de enterado en el respaldo de este documento, los estudiantes que no asistan el día de la mención de estos puntos a clases se dan por enterados del compromiso.

El profesor tiene el derecho a pedirle al estudiante que salga del grupo en caso de provocar distracción, incumplimiento a cualquier punto de arriba o desorden en general. **Según el Artículo 141 Fracción VII, XIII y XIV del Estatuto General de la UANL.**

# De las obligaciones:



El estudiante deberá ser puntual en la sesión de clase los días de las entregas y revisiones de proyectos señalados en el Calendario LMAD.

Los profesores y alumnos deben de seguir las fechas de establecidas por el Calendario LMAD.

Todo proyecto entregado para evaluación debe ser de la propia autoría. En caso de que el profesor indique que está permitido el uso de referencias, modelos o contenidos de un tercero, se deberá indicar en el proyecto la fuente referenciada.



|  |  |  |  |
| --- | --- | --- | --- |
| **Criterios** | **Escala** | | **Observaciones** |
| **Si cumple** | **No cumple** |
| DESARROLLO DE LA APLICACIÓN | | | |
| Desarrolla un proyecto de una aplicación Web (en PHP) siguiendo la metodología de POO (programación orientada a objetos) y MVC (*Model View Controller*), estableciendo clases para la conexión a una base de datos y métodos que utilicen llamadas a procedimientos almacenados en la base de datos para gestionar toda la información  del proyecto. |  |  |  |
| El proyecto incluye todas las validaciones para que el programa no genere errores durante su ejecución. |  |  |  |
| Se deben de mostrar mensajes o avisos para indicar la correcta operación del programa, por ejemplo, al gestionar registros, que muestre mensajes que los cambios se hayan realizado con éxito, o cuando falte un campo obligatorio por capturar, que confirme antes de eliminar un registro, etc. |  |  |  |
| Limpiando los campos de captura, teniendo un diseño limpio en cada pantalla.  Y objetos de captura de información de acuerdo con su tipo de dato. Por ejemplo, si se va a capturar una fecha, que te muestre un calendario. Si se va a capturar una contraseña, que por defecto los caracteres sean \*, etc. |  |  |  |
| Utiliza etiquetas para indicar todos los elementos en cada pantalla, incorpora variedad de controles en cada pantalla. |  |  |  |
| No permite la captura de un ID, para la consulta o modificación de algún elemento de entidades, a |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| menos que así se indique en la definición del proyecto.  Si usa tablas en las páginas, utiliza letreros apropiados en cada columna, no usando los campos de las tablas de la BD. |  |  |  |
| El proyecto incluye todas las pantallas para la gestión de las entidades establecidas en el diseño, donde se permita agregar información, modificar la información ya existente por medio de las llaves primarias y eliminar información no dependiente  ya sea física o lógicamente. |  |  |  |
| Para acceder al programa se debe contar con una clave de usuario y contraseña, debidamente guardados en una tabla de la base de datos y en la pantalla de acceso incluir las validaciones necesarias. |  |  |  |
| El programa debe resolver los reportes que se soliciten en la definición del proyecto y contar con pantallas para la elección de los filtros que serán utilizados como parámetros en los procedimientos almacenados que generan la información. La pantalla de filtros debe incluir las validaciones necesarias. |  |  |  |
| La aplicación debe ser de uso fácil, intuitivo y amigable al usuario final; deberá estar bien estructurado y con un diseño limpio y atractivo. |  |  |  |
| Elabora un video con el funcionamiento del proyecto de 10 minutos de duración donde se vea el funcionamiento del programa, su explicación y se visualice el código y la información de la base de datos. El proyecto debe ser publicado en YouTube. |  |  |  |
| LENGUAJE ESTRUCTURADO DE CONSULTAS (SQL) | | | |
| Se incluye un archivo de respaldo de la base de datos. |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| Incluye los scripts con la creación de los objetos de la base de datos en archivos texto, para ser ejecutados de forma autónoma. |  |  |  |
| Hace uso de programación avanzada de SQL, incluyendo un mínimo de procedimientos disparadores, vistas y funciones que son de utilidad para el proyecto (según se especifique en el *checklist* del proyecto). |  |  |  |