



Forma de Evaluación

## **BASES DE DATOS MULTIMEDIA**

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS  
**LIC. EN MULTIMEDIA Y ANIMACIÓN DIGITAL**

#### 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
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 ( <i>Model View Controller</i> ).
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.

## Descripción del proyecto

Funcionalidad	
Portal para cursos en línea	<p>Construir un portal en el cual se puedan ofrecer cursos en línea. Ejemplos: (Pluralsight, Udemy, Coursera, etc.)</p> <p>Los cursos deben estar diseñados para hacerlos de forma autodidacta y se deben de agrupar en alguna de las diferentes categorías. Ejemplos de categorías: (IT &amp; Software, Marketing, Design, etc.).</p>
Roles de usuario	<p>Se deben manejar dos roles para los usuarios, un rol para los que ofrecen cursos y otro rol de usuario para los estudiantes que solo van a tomar los cursos.</p> <p>Además, se debe contar con un usuario administrador para bloqueo y desbloqueo de usuarios, quitar comentarios ofensivos y obtener algunos reportes.</p>
Usuarios	<p>Los usuarios se pueden registrar en el mismo portal. Cada usuario debe tener una imagen del tipo avatar.</p> <p>Los usuarios registrados del rol “instructor” pueden ofrecer cursos en diferentes fases o niveles, a cada nivel se debe agregar un video con parte del contenido del curso.</p> <p>Los usuarios registrados como “estudiante” pueden comprar cursos y se debe registrar su avance.</p>

	<p>Cada usuario debe de tener los datos usuales de identificación, como lo son nombre completo, género, fecha de nacimiento, foto, email, contraseña y fecha de cuando se registraron o cuando fue el último cambio que se hizo en su información personal.</p> <p>La contraseña debe de incluir 8 caracteres al menos, y debe incluir una mayúscula, un carácter especial, y un número al menos.</p> <p>Si para ingresar al portal un usuario se equivoca tres veces consecutivas al capturar su contraseña, el usuario debe quedar deshabilitado. Se podría volver a activar con un usuario administrador.</p> <p>Los usuarios pueden cambiar la información de su perfil en cualquier momento, tener cuidado con la llave.</p> <p>Carácter especial = cualquier símbolo generado por el teclado que no sea letra ni número, por ejemplo: (¡"#\$%&amp;/'?¡¿;,:.-_+*{} })</p>
Cursos	<p>Se debe establecer la cantidad de niveles que tiene un curso, el costo por el curso completo, o por cada nivel, descripción de lo que trata el curso en general, comentarios de los usuarios y el promedio de la calificación en general del curso.</p> <p>Los usuarios podrán ofrecer algunos cursos completos gratis o solo algunos niveles gratis con la finalidad de atraer a más estudiantes.</p> <p>Se le debe agregar una imagen a cada curso, un título, una descripción de lo que contiene cada curso y el costo de este.</p> <p>El curso puede tener contenido texto, archivos adjuntos PDF, links a páginas externas, imágenes y videos.</p> <p>Al menos cada nivel del curso debe contar con un video de forma obligatoria.</p> <p>Cuando un alumno termine un curso de forma completa, revisando todos los niveles que lo componen, el usuario con el rol "instructor", le va a generar de forma automática un certificado o diploma al usuario con el rol "estudiante", donde avale que ha terminado satisfactoriamente un curso.</p> <p>El diploma debe de contener al menos la fecha de terminación del curso, nombre completo del alumno, nombre del curso, y nombre de quien certifica el curso.</p> <p>El usuario "instructor" que así lo decida podrá dar de baja los cursos de forma lógica. Estos ya no podrán ser consultados en las búsquedas ni</p>

	<p>se mostrarán en la pantalla principal, pero se podrán ver en los reportes de ventas y en el Kardex del alumno.</p> <p>Para los cursos completos con costo o algunos niveles del curso con costo, se debe de elegir una forma de pago.</p>
Categorías	<p>Cada curso debe pertenecer al menos a una categoría, cada usuario puede crear categorías o se pueden agregar cursos a las categorías que otros usuarios agregaron previamente.</p> <p>A las categorías se les puede agregar una descripción donde explique el tipo de cursos que contendrá.</p> <p>En la información de las categorías se debe de almacenar el usuario que la creó, fecha y hora de creación.</p> <p>Solo los usuarios administradores pueden editar categorías.</p>
Mensajes	<p>Debe haber una pantalla para que los usuarios se envíen mensajes de forma privada para hacer preguntas acerca del curso.</p> <p>Los mensajes generalmente serán entre el alumno y el instructor para obtener información acerca del curso.</p> <p>Se debe reflejar la fecha y hora de cada mensaje, nombre del usuario y su imagen.</p>
Comentarios	<p>Una vez que un usuario finalice un curso completo puede hacer sus comentarios a favor o en contra y calificar el curso. La forma de calificación a cada curso la define el programador, ejemplos: me gusta o no me gusta, 5 estrellas, del 1 al 10, etc.</p> <p>El promedio de la calificación de cada curso se debe de mostrar al visualizar la información de cada curso al igual que los comentarios de los usuarios que ya terminaron el curso, deben estar visibles en la descripción del curso.</p> <p>Los comentarios deben de tener fecha y hora de creación, así como el usuario que lo creó.</p>
Página principal	<p>Cualquier persona puede navegar por el portal y ver los títulos y las imágenes de los cursos que se ofrecen, los mejor calificados, los más vendidos o los más recientes.</p> <p>Pero para ver el detalle del curso se tiene que registrar y comprar el curso o registrarse en un curso que sea gratis.</p>

	<p>Solo se deben de mostrar los cursos activos.</p> <p>Se puede navegar por los cursos de diversas categorías que existen en el portal, la lista de categorías debe estar siempre visible.</p>
Kardex (Perfil alumno)	<p>Cada usuario debe tener la forma de ver el total de cursos a los que se ha registrado y el progreso que ha avanzado en cada uno de ellos, se podrá visualizar la fecha en la que se inscribió al curso, última fecha en la que ingresó a algún nivel y en dado caso, la fecha de terminación del curso, si el curso está completo o incompleto.</p> <p>La vista de la información de los cursos debe ser como un Kardex.</p> <p>Establecer filtros para la obtención del Kardex:</p> <ul style="list-style-type: none"> <li>▪ Rango de fechas de inscripción al curso</li> <li>▪ Categoría (una o todas)</li> <li>▪ Solo cursos terminados o todos</li> <li>▪ Solo cursos activos o todos</li> </ul>
Ventas (Perfil instructor)	<p>El programa debe tener la posibilidad de visualizar las ventas de los usuarios que ofrecen cursos de dos formas:</p> <ol style="list-style-type: none"> <li>1) Una lista con cada curso que ofrece, cantidad de alumnos inscritos, nivel promedio que ha cursado cada alumno y total de ingresos que le genera ese curso. Al final el total de ingresos por todos los cursos, desglosado por forma de pago.</li> <li>2) Por cada curso, información con el detalle cada alumno inscrito, nombre del alumno, fecha de inscripción al curso, nivel de avance, precio pagado por ese alumno y forma de pago. Al final el total de ingresos por el curso.</li> </ol> <p>Establecer filtros para la obtención de esos reportes:</p> <ul style="list-style-type: none"> <li>▪ Rango de fechas de creación de los cursos</li> <li>▪ Categoría (una o todas)</li> <li>▪ Solo cursos activos o todos</li> </ul> <p>(Las cantidades monetarias, deben tener el símbolo de moneda, coma para separar los miles y solo dos decimales. El formato de fechas debe ser día, tres letras del mes y 4 dígitos para el año)</p>
Buscador	<p>El portal permitirá hacer la búsqueda de cursos por categoría, por título del curso, por usuario que los ha publicado, o por rango de fechas de publicación del curso.</p>

	<p>La búsqueda por el título del curso puede ser parcial, debe buscar una palabra o todo el título del curso y mostrar todos los cursos que tengan esa palabra.</p> <p>La búsqueda de cursos solo debe mostrar cursos activos, los cursos dados de baja no se deben de mostrar.</p>
Host (opcional)	Es opcional que el proyecto esté alojado en un servicio de Web Hosting para la revisión de la funcionalidad.
<b>Restricciones del proyecto</b>	<p>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 <a href="#">anexo 1. Lista de Cotejo</a>.</p> <p><b>Es obligatorio que el proyecto cumpla todos los puntos sin excepción, de otro modo no será revisado.</b></p>

### Entregas para el semestre

**Primera Entrega Obligatoria:** Se revisará de forma presencial, puede ser en el equipo del alumno. Después de revisado el avance, el alumno realizará las correcciones que el maestro le indique y subirá el proyecto en un archivo .ZIP o .RAR.

El archivo debe tener las siglas del curso, grupo, matrícula y nombre del alumno e indicar el número de avance. Se subirán a una tarea creada en Teams.

- El proyecto en HTML
- Con todos los archivos necesarios: JS, CSS, etc.
- Documento con la primera propuesta del diseño relacional de la base de datos

**Segunda Entrega Obligatoria:** Se revisará de forma presencial, puede ser en el equipo del alumno. Después de revisado el avance, el alumno realizará las correcciones que el maestro le indique y subirá la estructura de la base de datos y el proyecto en un archivo .ZIP o .RAR.

El archivo debe tener las siglas del curso, grupo, matrícula y nombre del alumno e indicar el número de avance. Se subirán a una tarea creada en Teams.

- Un documento en Word con los elementos del diseño de la base de datos.
- SP's en archivos .SQL
- Proyecto Web (HTML, JS, CSS, PHP)
- Videos

**Tercera Entrega Obligatoria:** Se debe de entregar en una memoria USB y en archivos .rar o .zip lo siguiente:

- Todos los archivos de la aplicación web (PHP, HTML, CSS, JavaScript, etc.) y demás archivos necesarios para el funcionamiento de la aplicación.

- Un documento en Word con los elementos del diseño de la base de datos, corregido según la retroalimentación del avance previo.
- 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 de la base de datos:
  - Procedimientos almacenados,
  - Procedimientos disparadores,
  - Vistas,
  - Funciones,
  - Carga de información, (si se requiere)
  - en uno o varios archivos texto .sql
- Documento en Word para la implementación y operación del proyecto.



**Lista de chequeo de características a evaluar para el 1° avance. Valor de: 100pts. 20% cal total**

Matrícula: \_\_\_\_\_ Nombre: \_\_\_\_\_ Calif: \_\_\_\_\_

REQ	Cumple	Puntos	A Evaluar
Si	<input type="checkbox"/>	15	El diseño de la todas las páginas en HTML.
Si	<input type="checkbox"/>	10	Validaciones en JavaScript o JQuery.
Si	<input type="checkbox"/>	10	Incluye todos los elementos que se solicitan en cuanto a funcionalidad.
Si	<input type="checkbox"/>	20	Diseño del modelo relacional de la base de datos
Si	<input type="checkbox"/>	10	Mostrar mensajes o avisos para indicar el correcto funcionamiento de la página web.
Si	<input type="checkbox"/>	10	Incluye hojas de estilo (CSS)
No	<input type="checkbox"/>	5	Incluye Bootstrap
Si	<input type="checkbox"/>	5	Diseño de estilo propio, sin utilizar plantillas.
No	<input type="checkbox"/>	5	Incluye algunos datos de cursos ficticios como ejemplos.
Si	<input type="checkbox"/>	10	Diseño de un diploma o certificado que se deberá entregar a cada alumno que termine satisfactoriamente algún curso.

**Puntos totales recibidos en 1ra entrega: \_\_\_\_\_**

**Lista de chequeo de características a evaluar para el 2° avance. Valor de: 100pts. 20% cal total**

Matrícula: \_\_\_\_\_ Nombre: \_\_\_\_\_ Calif: \_\_\_\_\_

REQ	Cumple	Puntos	A Evaluar
Si	<input type="checkbox"/>	10	Establece la estructura de la base de datos en un modelo relacional definiendo las llaves primarias y llaves foráneas.
Si	<input type="checkbox"/>	10	Script de la creación de tablas, índices, llaves primarias y foráneas en MySQL
Si	<input type="checkbox"/>	10	En el script con la creación de las tablas, sin omitir la relación de alguna tabla. Determina los tipos de datos adecuados para cada campo.
No	<input type="checkbox"/>	15	Página para el Registro, modificación y consulta del perfil del usuario, por rol de usuario.
Si	<input type="checkbox"/>	10	Validación de requisitos de contraseña para el perfil de usuario.
Si	<input type="checkbox"/>	5	Obtiene el diagrama relacional de MySql.
Si	<input type="checkbox"/>	10	Diccionario de Datos (incluyendo dominio para cada atributo y descripción, ambos en la estructura de cada tabla).
Si	<input type="checkbox"/>	10	Script con el o los SP necesarios para todas las gestiones del perfil de usuario.
No	<input type="checkbox"/>	5	Clase de PHP para la conexión a la base de datos.
Si	<input type="checkbox"/>	15	Creación de 3 videos tutoriales con temas de BDM que serán cargados al portal en la entrega final.

**Puntos totales recibidos en 2da entrega: \_\_\_\_\_**

## Lista de chequeo de características a evaluar para la 3ra entrega y final. Valor de: 100pts. 60% cal total

Matrícula: \_\_\_\_\_ Nombre: \_\_\_\_\_ Calif: \_\_\_\_\_

REQ	Cumple	Puntos	A Evaluar
Si	<input type="checkbox"/>	10	Uso avanzado de SQL: mínimo 2 Triggers, 8 Views, 2 Functions <i>(que sean de utilidad al proyecto)</i> .
Si	<input type="checkbox"/>	5	Pantalla principal con diseño óptimo y eficaz y la visualización de los cursos mejor calificados, los más vendidos y los más recientes.
Si	<input type="checkbox"/>	15	Ver detalle del curso (Imágenes, videos, información del curso, vínculos, etc.). Edición del curso.
Si	<input type="checkbox"/>	5	Registro de categorías. Validaciones.
Si	<input type="checkbox"/>	10	Mensajes entre usuarios de forma privada.
Si	<input type="checkbox"/>	5	Elaboración de los comentarios y calificación de cada curso.
Si	<input type="checkbox"/>	10	Mostrar el Kardex del alumno.
Si	<input type="checkbox"/>	10	Vista general de los cursos vendidos.
Si	<input type="checkbox"/>	5	Vista detallada de información por cada curso vendido.
Si	<input type="checkbox"/>	5	Búsqueda por los filtros establecidos.
Si	<input type="checkbox"/>	0	Validaciones en toda la operación del proyecto (se descontará en cada rubro)
Si	<input type="checkbox"/>	5	Archivos con scripts de creación de objetos de la BD.
Si	<input type="checkbox"/>	5	Generación de certificado para el estudiante al finalizar el curso.
No	<input type="checkbox"/>	10	Web Hosting

**Puntos totales recibidos en 3ra entrega:** \_\_\_\_\_

## Ponderación de la CP para el promedio final

Elemento de evaluación	Ponderación
Proyecto	90
Laboratorios	10
<b>Calificación Final</b>	<b>100%</b>

La ponderación de la CP es el 100% de la calificación final, tomando en cuenta el 90% de los avances del proyecto y el proyecto final y un 10% del laboratorio de la Unidad de Aprendizaje. Esto último aplica para los estudiantes que cursan la primera oportunidad, en el caso de cursar la tercera o quinta oportunidad, el 100% de la calificación se tomará de la suma de los porcentajes de los tres entregables, tomando en cuenta que el primero y el segundo valdrían un 15% respectivamente.

## 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 será 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.

## Anexo 1. Lista de cotejo

Antes de entregar el proyecto, éste debe cumplir con todos los siguientes criterios de forma obligatoria, cualquier duda favor de referirla al profesor. Si no cumple con un punto, no será revisado.

Criterios	Escala		Observaciones
	Si cumple	No cumple	
DISEÑO DE BASES DE DATOS			
Cuenta con una estructura de la base de datos con todas las tablas normalizadas hasta la tercera forma normal.			
Utiliza la generación automática de ID's en algunas de las tablas ( <i>Autoincrement</i> ).			
Utilizando un sistema gestor de base de datos, elabora y construye un diagrama con las tablas del modelo relacional.			
Establece un diccionario de datos donde se incluye cada elemento de la base de datos, una descripción, tipo de dato, dominio y restricciones.			
Los campos donde se va a guardar contenido multimedia se deben definir como BLOB de forma obligatoria para las imágenes.			
Los campos donde se va a guardar contenido multimedia se deben definir como BLOB de forma opcional para los videos.			
DESARROLLO DE LA APLICACIÓN			
Desarrolla un proyecto de una aplicación Web (en PHP) siguiendo la metodología de POO (programación orientada a objetos) y MVC ( <i>Model View Controller</i> ), 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 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.			
<b>LENGUAJE ESTRUCTURADO DE CONSULTAS (SQL)</b>			
Todas las consultas a la base de datos (Select, Update, Insert o Delete, etc.) a tablas o a vistas, deberán de realizarse a través de procedimientos almacenados (Stored Procedures), nunca de forma directa.			
No se permite el símbolo * (asterisco) para cualquier tipo de consultas donde se utilice el SELECT, ni para tablas ni para vistas, se deben de especificar siempre los campos a mostrar.			
Se incluye un archivo de respaldo de la base de datos (Opcional).			
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 <i>checklist</i> del proyecto).			