

	<p align="center"> <b>INSTITUTO TÉCNICO RICALDONE</b>  <b>ESPECIALIDAD DESARROLLO DE SOFTWARE</b>  <b>TERCER AÑO DE BACHILLERATO</b>  <b>BTVDS 3.3: ADMINISTRACIÓN DE BASES DE DATOS</b>  <b>FASE DE EJECUCIÓN</b>  <b>DOCENTE: WILFREDO GRANADOS</b>  <a href="mailto:daniel_granados@ricaldone.edu.sv">daniel_granados@ricaldone.edu.sv</a> </p>	
INTEGRANTES:	<ol style="list-style-type: none"> <li>Morales Sandoval, Daniel Alessandro</li> <li>Gómez Martínez, Fernando José</li> <li>Valdés Jacobo, Josué Emiliano</li> <li>Orellana Aguilar, Adriana Elizabeth</li> <li>Sánchez Henríquez, José Alejandro</li> </ol>	<p>GRUPO/SECCIÓN:          ____2A____ FECHA:          ____11/04/24____</p>

### **INSTRUMENTO DE HETEROEVALUACIÓN** **(ESCALA ESTIMATIVA) OBJETIVO:**

Crear una documentación completa y precisa que permita obtener una comprensión clara de la estructura de la base de datos de la tienda en línea, facilitando así la comprensión global de su diseño.

#### CONSIDERACIONES:

- **La tarea inicia el 3 de abril al 12 de abril** y debe ser entregada por el coordinador del equipo en el espacio asignado por el docente a más tardar el 12 de abril a las 4:00 PM para su calificación, de lo contrario la nota ira disminuyendo un punto por entrega tardía y 1 punto cada día que se atrase la entrega.
- La actividad tiene una ponderación del 15% de la fase de valoración del módulo, siendo revisado cada elemento según las indicaciones correspondientes.
- Para realizar la calificación se asigna el puntaje correspondiente a cada criterio de acuerdo con la siguiente escala de valoración: **1 = Deficiente, 2 = Regular, 3 = Bueno, 4 = Muy bueno y 5 = Excelente. 0 si no se cumple el criterio.**
- La sumatoria se calcula sumando todos los puntajes. La nota se obtiene mediante el siguiente procedimiento: Promedio de **SUMATORIA / 14 + Nota individual**

#	CRITERIOS	PUNTAJE
1	<b>Repositorio de GitHub.</b> Se crea el repositorio de GitHub y se agregan como colaboradores a los demás integrantes del equipo y se agrega la información solicitada en el archivo REAMDE.md del repositorio. Se proporciona el enlace para acceder al repositorio de GitHub. El repositorio debe de ser público de no poder observarse se anularán todos los criterios a calificar por no seguir indicaciones.	
2	<b>Correcciones de los Diagramas.</b> Se realizan las correcciones necesarias en los diagramas ER, Modelo Relacional y en el Diccionario de la base de datos de haber obtenido observaciones.	
3	<b>SCRIPT SQL.</b> Se presenta el SCRIPT SQL de la base de datos, para crear la estructura de la base de datos (Solo tablas) de forma que se pueda ejecutar de forma completa. SCRIPT 100% funcional. Incluye al inicio una sentencia <b>DROP DATABASE IF EXISTS.</b> <b>(Si el Script no se puede ejecutar completo, se anula este punto)</b>	

4	<b>Restricciones (Constraint SQL).</b> Las tablas de la base de datos cuentan con restricciones en los campos donde se considere necesario, cada restricción creada cuenta con un nombre utilizando la sentencia CONSTRAINT previo a la creación de cada restricción. Campos como DUI, Correo, teléfono, entre otros campos deben contar con restricciones UNIQUE, campos numéricos que no puedan ser negativos también deben de contar con una restricción.	
5	<b>Relaciones:</b> Las relaciones entre tablas están creadas de forma correcta utilizando las llaves foráneas, cada relación de llave foránea tiene nombre utilizando la sentencia CONSTRAINT fk_relacion.	
6	<b>Nombre de la base de datos:</b> El nombre de la base de datos es significativo y acorde a	

	su propósito. Está escrito todo en minúsculas	
7	<b>Nombres de tablas y atributos.</b> Se respeta el uso de nombres en plural y singular en la base de datos, se define plural o singular para tablas y atributos. Puede utilizarse plurales tablas y singulares atributos. Los nombres de los atributos de cada tabla son únicos y describen correctamente al atributo de la tabla.	
8	<p><b>Uso de nomenclatura snake_case y uso de sentencias SQL:</b> Todos los nombres ya sea de base de datos, tablas, atributos, funciones, triggers, procedimientos, vistas utilizan la nomenclatura snake_case para su escritura y están todas en minúsculas. Todas las sentencias SQL se escriben en Mayúsculas sean sentencias DDL, DML, DQL, DCL. Ejemplo:</p> <pre> 1  DROP DATABASE if EXISTS db_nombre_base_datos; 2 3  CREATE DATABASE db_nombre_base_datos; 4 5  USE db_nombre_base_datos; 6 7  CREATE TABLE tb_estudiantes( 8      id_estudiante INT PRIMARY KEY, 9      nombre_estudiante VARCHAR(50), 10     edad_estudiante INT, 11     carnet_estudiante VARCHAR(20) 12 ); 13 14 -- agregando restricciones a las tablas 15 ALTER TABLE tb_estudiantes ADD 16 CONSTRAINT check_edad 17 CHECK (edad_estudiante &gt;=12); 18 19 ALTER TABLE tb_estudiantes ADD 20 CONSTRAINT unique_carnet 21 UNIQUE (carnet_estudiante); </pre>	
9	<b>Normalización:</b> El diseño de la base de datos refleja un modelado eficiente de acuerdo a la Tercer Forma Normal, sin redundancias ni ambigüedades. No existen campos que puedan generar datos duplicados de tablas que se relacionen entre sí. Se pueden tener campos como el Género, Estados, entre otros dentro de la misma tabla solamente si ya se conocen los valores que se almacenarán en ese campo y estos deberán de ser un tipo de dato ENUM.	

10	<b>Tablas:</b> La base de datos cuenta con un mínimo de 8 tablas y un máximo de 16 tablas, dependiendo de la lógica del negocio del proyecto PTC. (de tener más de 16 tablas, el docente debe de estar al tanto del porqué de cada tabla a utilizar en el proyecto)	
11	<b>Usuarios de la base de datos:</b> Se crea 1 usuario para utilizar dentro de la conexión de la base de datos según las siguientes características: <ul style="list-style-type: none"> <li>• El nombre debe de ser: <b>nombre_base_datos_desarrollador</b></li> <li>• Permisos DML para insertar, actualizar, eliminar y seleccionar datos.</li> <li>• Permisos para poder ejecutar y crear funciones, procedimientos, triggers vistas.</li> <li>• El código deberá de estar presente en un Query aparte del Query principal de la base de datos.</li> </ul>	
12	<b>Inserción de datos.</b> Se debe de tener un script aparte que sirva para insertar datos sobre las tablas de la base de datos del PTC, teniendo en las tablas principales como pueden ser la de productos, categorías si tuviese, clientes, pedidos, entre otras un total de 10 datos como mínimo dentro de cada tabla. (Tablas en donde no puedan existir 10 datos, colocar solamente los datos necesarios)	

13	<b>Triggers, Funciones, Procedimientos Almacenados:</b> Se utiliza al menos un trigger, una función o un procedimiento almacenado dentro de la base de datos del PTC. El código SQL de estos debe de estar en un Script Aparte y se debe de explicar/comentar el uso del código SQL.	
<b>Notas Individuales: Como equipo PTC, se calificarán en un nivel de logro del 1 al 5, de forma honesta y objetiva para reflejar su aporte en el trabajo de la base de datos del PTC.</b>		
14	<b>Estudiante 1:</b> Escribir un nivel de logro del 1 al 5.	
15	<b>Estudiante 2:</b> Escribir un nivel de logro del 1 al 5.	
16	<b>Estudiante 3:</b> Escribir un nivel de logro del 1 al 5.	
17	<b>Estudiante 4:</b> Escribir un nivel de logro del 1 al 5.	
18	<b>Estudiante 5:</b> Escribir un nivel de logro del 1 al 5.	
<b>ASPECTOS GENERALES</b>		
19	<b>Puntualidad.</b> La tarea es entregada dentro del plazo asignado para la entrega.	
SUMATORIA		
NOTA		

OBSERVACIONES:

---



---



---



---



---



---



---



---



---



---