

Universidad del Valle de Guatemala

Facultad de Ingeniería

Departamento de Ciencia de la Computación

Bases de Datos I

1966

UNIVERSIDAD

GUATEMALA



DESIGN BETTER

PLATAFORMA DE DISEÑO Y PERSONALIZACIÓN DE MODA

Pablo José Méndez Alvarado – 23975

Luis Fernando Palacios López – 239333

Roberto Samuel Nájera Marroquín – 23781

André Emilio Pivaral López – 23574

Walter Alexander Cruz Coronado – 20673

DEL VALLE DE

Catedrático: Erick Francisco Marroquín Rodríguez

Sección: 10

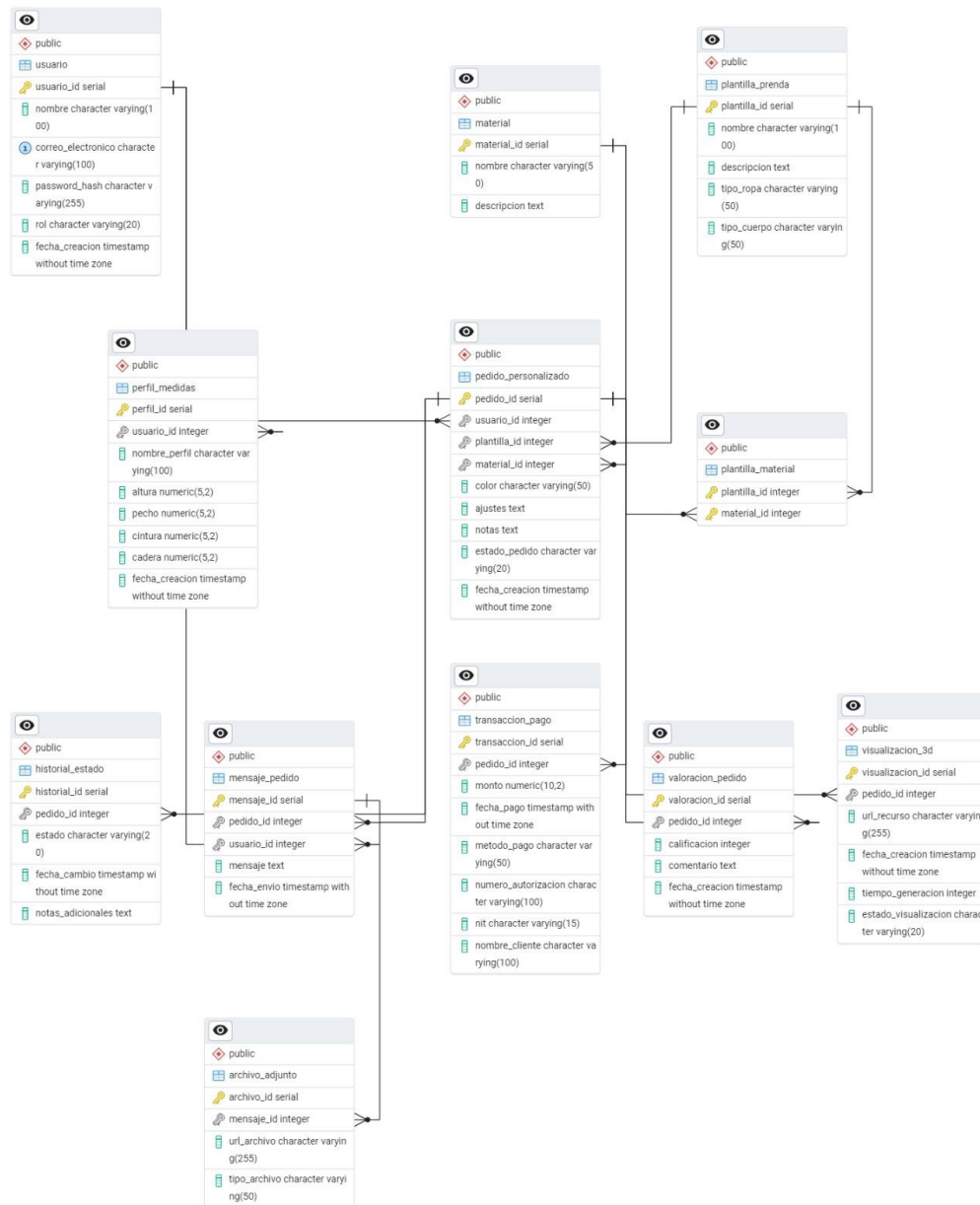
Nueva Guatemala de la Asunción, 24 de marzo de 2024

Excelencia que trasciende

Índice

Índice.....	1
Modelo ER	2
Justificación del Diseño	2
Documentación de Reglas de Negocio y Restricciones.....	3
Enlace al versionamiento con los scripts	7
Enlace con las reflexiones individuales	7
Documentación de los cambios	7

Modelo ER



Justificación del Diseño

La base de datos diseñada para el proyecto sigue un modelo relacional, garantizando integridad, eficiencia y escalabilidad. Se emplearon claves primarias autoincrementales (IDs seriales) para asegurar la unicidad y facilitar relaciones claras mediante claves foráneas entre entidades principales como Usuarios, Pedidos y Plantillas de Prendas.

Se utilizó una tabla intermedia (plantilla_material) para representar la relación muchos a muchos entre Plantillas y Materiales, permitiendo flexibilidad. Además, el historial del seguimiento de pedidos se gestiona mediante una tabla dedicada (historial_estado), que proporciona una trazabilidad clara desde el inicio hasta la entrega del pedido.

Para gestionar eficazmente la comunicación cliente-diseñador, se incluyeron las tablas mensaje_pedido y archivo_adjunto, facilitando el intercambio ordenado de mensajes y archivos multimedia. La generación de visualizaciones en 3D se almacena en una tabla independiente (visualizacion_3d), admitiendo múltiples iteraciones y modificaciones.

Se añadieron tablas especializadas para almacenar valoraciones de los clientes (valoracion_pedido) y transacciones financieras (transaccion_pago), estas últimas cumpliendo con las normativas fiscales locales. Finalmente, la estructura permite gestionar múltiples perfiles de medidas corporales (perfil_medidas) asociados a un mismo usuario, favoreciendo una personalización más precisa.

Documentación de Reglas de Negocio y Restricciones

Categoría 1: Usuarios

RN-1: Un usuario puede registrarse como Diseñador o Cliente, pero no ambos con la misma cuenta.

Implementación DB: Tabla usuario con una columna rol que permita valores específicos como 'diseñador' o 'cliente'. Se usa una restricción CHECK para validar roles únicos.

RN-2: Cada usuario debe proporcionar nombre, correo electrónico único, contraseña y fecha de creación.

Implementación DB: Tabla usuario con campos como usuario_id (PK), nombre, correo_electronico (UNIQUE), password_hash, y fecha_creacion.

Categoría 2: Gestión de medidas corporales

RN-3: Los clientes podrán registrar múltiples perfiles de medidas corporales (p.ej.: para diferentes personas).

Implementación DB: Tabla perfil_medidas vinculada mediante FK con la tabla usuario. Campos esenciales como altura, pecho, cintura, cadera, etc.

RN-4: Cada conjunto de medidas corporales pertenece exactamente a un cliente.

Implementación DB: Tabla perfil_medidas con una FK obligatoria al usuario propietario (usuario_id).

Categoría 3: Gestión de Plantillas

RN-5: Existen plantillas base de prendas clasificadas por tipo de ropa (vestidos, pantalones, etc.) y tipo de cuerpo (pequeño, mediano, grande).

Implementación DB: Tabla plantilla_prenda con campos como plantilla_id (PK), nombre, descripcion, tipo_ropa, tipo_cuerpo, entre otros atributos relevantes.

RN-6: Cada plantilla base podrá asociarse con múltiples tejidos o materiales recomendados.

Implementación DB: Relación muchos a muchos entre tabla plantilla_prenda y tabla material, mediante tabla intermedia plantilla_material.

Categoría 4: Personalización de prendas

RN-7: Un cliente puede crear múltiples pedidos personalizados a partir de plantillas existentes.

Implementación DB: Tabla pedido_personalizado con FK hacia usuario (cliente) y plantilla_prenda.

RN-8: Cada pedido personalizado puede especificar personalizaciones específicas (color, materiales, ajustes menores, notas adicionales).

Implementación DB: Tabla pedido_personalizado con columnas adicionales como color, material_id (FK), ajustes, y notas.

Categoría 5: Visualización y Prototipado

RN-9: Cada pedido personalizado genera una visualización 3D que debe almacenarse como referencia para futuras consultas.

Implementación DB: Tabla visualizacion_3d con FK a la tabla pedido_personalizado. Almacenar ruta o URL del recurso generado (imagen/renderizado 3D).

RN-10: Las visualizaciones 3D deben almacenar información sobre fecha de creación, tiempo de generación, y estado (generado, pendiente, error).

Implementación DB: Tabla visualizacion_3d con campos: fecha_creacion, tiempo_generacion, estado_visualizacion (usando CHECK).

Categoría 6: Seguimiento del Pedido

RN-11: Cada pedido personalizado tendrá un seguimiento con etapas claras (pendiente, diseño, producción, entrega, completado, cancelado).

Implementación DB: Tabla pedido_personalizado con columna estado_pedido restringida por CHECK ('pendiente', 'diseño', 'produccion', 'entrega', 'completado', 'cancelado').

RN-12: El cliente puede ver el historial de cambios de estado en su pedido.

Implementación DB: Tabla historial_estado con FK hacia pedido_personalizado, estado, fecha_cambio, y notas_adicionales.

Categoría 7: Comunicación

RN-13: Los clientes y diseñadores pueden intercambiar mensajes relacionados a un pedido específico.

Implementación DB: Tabla mensaje_pedido con FK hacia pedido_personalizado, usuario_id, mensaje, fecha_envio.

RN-14: Se puede agregar archivos multimedia adjuntos en los mensajes (imágenes, PDFs).

Implementación DB: Tabla adicional_archivo_adjunto con FK hacia mensaje_pedido, url_archivo, tipo_archivo.

Categoría 8: Comentarios y Valoraciones

RN-15: El cliente puede valorar y comentar sobre su pedido personalizado una vez esté marcado como “completado”.

Implementación DB: Tabla valoracion_pedido con FK a pedido_personalizado, campos: calificacion (1-5), comentario, y fecha_creacion.

Categoría 9: Administración del sistema

RN-16: Un administrador puede gestionar usuarios, pedidos y plantillas (crear, actualizar, eliminar).

Implementación DB: Tabla administrador que extiende la tabla usuario o rol específico mediante columna rol con valor 'administrador'.

Categoría 10: Cumplimiento Legal

RN-17: Todas las transacciones (pagos) deben registrar información básica para cumplir normativas fiscales (SAT de Guatemala).

Implementación DB: Tabla transaccion_pago con campos requeridos como pedido_personalizado_id (FK), monto, fecha_pago, metodo_pago, numero_autorizacion, y datos de facturación (nit, nombre_cliente, etc).

Categoría 11: Seguridad

RN-18: Las contraseñas deberán almacenarse de forma cifrada y segura.

Implementación DB: Campo password_hash (bcrypt o similar) en tabla usuario.

RN-19: No almacenar datos sensibles de tarjetas bancarias.

Implementación DB: Asegurar mediante restricciones de aplicación (no almacenamiento explícito).

Categoría 12: Rendimiento y Escalabilidad

RN-20: La base de datos debe permitir realizar búsquedas rápidas (filtros por cliente, estado de pedido, fechas, etc.).

Implementación DB: Índices adecuados sobre campos como estado_pedido, fecha_creacion, y FK claves como usuario_id.

Enlace al versionamiento con los scripts

<https://github.com/WalterCruz1600/DesignBetter>

Enlace con las reflexiones individuales

<https://github.com/WalterCruz1600/DesignBetter/tree/main/reflexiones>

Documentación de los cambios

Fecha de Implementación de Cambios:

19/03/2024

Motivo de los Cambios:

Tras la implementación inicial de la base de datos, se identificó la necesidad de mejorar el rendimiento de las consultas, especialmente en operaciones relacionadas con la recuperación de pedidos, historial de estados y mensajes. Para abordar esto, se decidió agregar índices en columnas clave que se utilizan con frecuencia en filtros y relaciones.

Resumen de Cambios Realizados:

Creación de Índices para Optimización de Consultas

Se añadieron los siguientes índices para mejorar la eficiencia de las consultas en la base de datos:

- Índice en usuario(correo_electronico)
- Objetivo: Acelerar la búsqueda de usuarios por correo electrónico, ya que es un campo único y de uso frecuente.
- Sentencia SQL:

```
CREATE INDEX idx_usuario_correo ON usuario(correo_electronico);
```

- Índice en pedido_personalizado(usuario_id)
- Objetivo: Mejorar la velocidad en la recuperación de pedidos de un usuario específico.
- Sentencia SQL:

```
CREATE INDEX idx_pedido_usuario ON pedido_personalizado(usuario_id);
```

- Índice en pedido_personalizado(estado_pedido)
- Objetivo: Agilizar las consultas que filtran pedidos según su estado (pendiente, diseño, producción, etc.).

- Sentencia SQL:

```
CREATE INDEX idx_pedido_estado ON pedido_personalizado(estado_pedido);
```

- Índice en historial_estado(pedido_id)
- Objetivo: Optimizar la recuperación del historial de estados de un pedido específico
- Sentencia SQL:

```
CREATE INDEX idx_historial_pedido ON historial_estado(pedido_id);
```

- Índice en mensaje_pedido(pedido_id)
- Objetivo: Agilizar la obtención de mensajes relacionados con un pedido.
- Sentencia SQL:

```
CREATE INDEX idx_mensaje_pedido ON mensaje_pedido(pedido_id);
```

Impacto en el Sistema

- Mejoras en el rendimiento: Las consultas que utilizan filtros en los campos indexados ahora se ejecutan más rápido, reduciendo el tiempo de respuesta en operaciones clave.
- Menor carga en el servidor: Se optimizó el acceso a los datos sin afectar la estructura de las tablas.
- Espacio en disco: El uso de índices puede aumentar el consumo de almacenamiento, pero el beneficio en rendimiento justifica su implementación.

Pruebas Realizadas

Se realizaron pruebas comparativas en las siguientes consultas antes y después de agregar los índices:

- Búsqueda de usuario por correo electrónico.
- Recuperación de pedidos por usuario y estado.
- Consulta del historial de estados de un pedido.
- Obtención de mensajes asociados a un pedido.

Los tiempos de respuesta mostraron una mejora significativa, especialmente en tablas con una mayor cantidad de registros.

Recomendaciones Futuras

- Monitorear el rendimiento de los índices periódicamente para asegurarse de que siguen siendo efectivos.
- Evaluar la necesidad de nuevos índices si el volumen de datos sigue creciendo.
- Optimizar consultas adicionales en base a estadísticas de uso de la base de datos.