

24 DE NOVIEMBRE DE 2025

MANUAL TECNICO Y DE USUARIO

BASE DE DATOS

NATASHA VERASTEGUI AGUSTIN 174653
ALDO ESTEBAN GARCIA VALENCIA 179862
UNIVERSIDAD POLITECNICA DE SAN LUIS POTOSI

El presente Manual Técnico documenta el diseño, implementación y funcionamiento del Sistema de Optimización de Corte de Materia Prima Laminar, desarrollado sobre el gestor de bases de datos PostgreSQL.

El propósito del sistema es mejorar el aprovechamiento de materiales como madera, metal en lámina, tela u otros sustratos laminares mediante el registro estructurado de piezas, productos y eventos de optimización.

Alcance

El sistema cubre los siguientes aspectos desarrollados en PostgreSQL:

- Gestión de usuarios y roles, incluyendo permisos diferenciados para Administrador y Operador.
- Registro, consulta y administración de materia prima, con parámetros de configuración específicos como dimensiones y distancias mínimas.
- Administración de productos y piezas, incluyendo definición geométrica básica y cantidad de elementos.
- Colocación, rotación y posicionamiento de piezas, mediante procedimientos almacenados y eventos en formato JSON.
- Cálculo de la utilización de materia prima, mediante una función especializada que determina el porcentaje de aprovechamiento según las piezas colocadas.
- Registro y procesamiento de eventos JSON, utilizados para documentar operaciones de optimización.
- Activación automática de validaciones y recálculo mediante triggers, asegurando integridad operacional.
- Seguridad y control de acceso, aplicando privilegios adecuados a cada rol.
- Pruebas unitarias básicas, necesarias para comprobar la correcta ejecución del sistema.

Arquitectura del sistema

La arquitectura del sistema está diseñada bajo un modelo modular aprovechando la gestión de operaciones relacionadas con la optimización del corte de material. En el cual se utiliza el lenguaje de PostgreSQL como base de datos para ejecutar las operaciones mediante procedimientos, funciones y triggers.

La estructura asegura la separación entre:

- Capa de persistencia (tablas, claves, restricciones).
- Capa lógica (funciones, procedimientos y triggers).
- Capa de seguridad (roles PostgreSQL y permisos).

El modelo está compuesto por las entidades centrales del proceso de optimización del corte: usuarios y roles (gestión de seguridad), materia prima, productos, piezas, piezas colocadas, optimización de corte, geometrías y eventos de optimización (almacenamiento de eventos JSON).

Cada tabla incluye claves primarias, claves foráneas, reglas de integridad y restricciones necesarias para asegurar consistencia.

Procedimientos almacenados

Los procedimientos ejecutan operaciones críticas del flujo de trabajo:

- “sp_alta_materia_prima”: Registrar materia prima con sus dimensiones y parámetros de separación.
- “sp_alta_producto”: Dar de alta un producto, su pieza base y su geometría.
- “sp_rotar_posicionar_figuras”: Aplicar transformaciones geométricas a piezas colocadas y registrar eventos JSON.

Estas rutinas encapsulan la lógica del negocio dentro de la base de datos, reduciendo errores y reforzando la seguridad.

Funciones

Destacan:

- fn_calcular_utilizacion: Retorna el porcentaje de aprovechamiento de la materia prima, considerando el área total y el área ocupada por piezas colocadas.
- fn_parsear_evento_json (si forma parte del diseño): Deserializa el contenido JSON para interpretar orientación, coordenadas o parámetros enviados por el sistema de optimización.

Estas funciones permiten cálculos automáticos y consultas más completas.

Triggers

- Los triggers brindan automatización e integridad avanzada.
- Validación automática de distancia mínima entre piezas.
- Recalcular el porcentaje de aprovechamiento después de cada inserción o actualización.
- Registro automático de eventos cuando se detectan cambios relevantes.

Gestión de eventos JSON

Todos los movimientos, rotaciones y configuraciones aplicadas sobre piezas son almacenadas en una tabla llamada “eventos de optimización” en el cual permite a ver una auditoría completa, hacer reversión de cambios o reconstrucción de escenarios, además de comunicación con sistemas externos de optimización.

Seguridad y roles

PostgreSQL implementa un modelo basado en roles que asegura un adecuado control de acceso:

Administrador: permite dar de alta/baja de usuarios, la ejecución completa de procedimientos de modificación de materia y productos.

Operador: Registrar eventos, mover o rotar piezas y ver resultados de corte.

Instalación y configuración

Se utiliza la versión PostgreSQL +14, se requiere un permiso de rol “super usuario” para crear los roles en caso de usar un compilador en línea. Tambien se utiliza Github para clonar repositorios y hacer pruebas, ahí ejecutamos el archivo utilizando Python para validar ciertas funciones y pruebas.

Modelos de datos

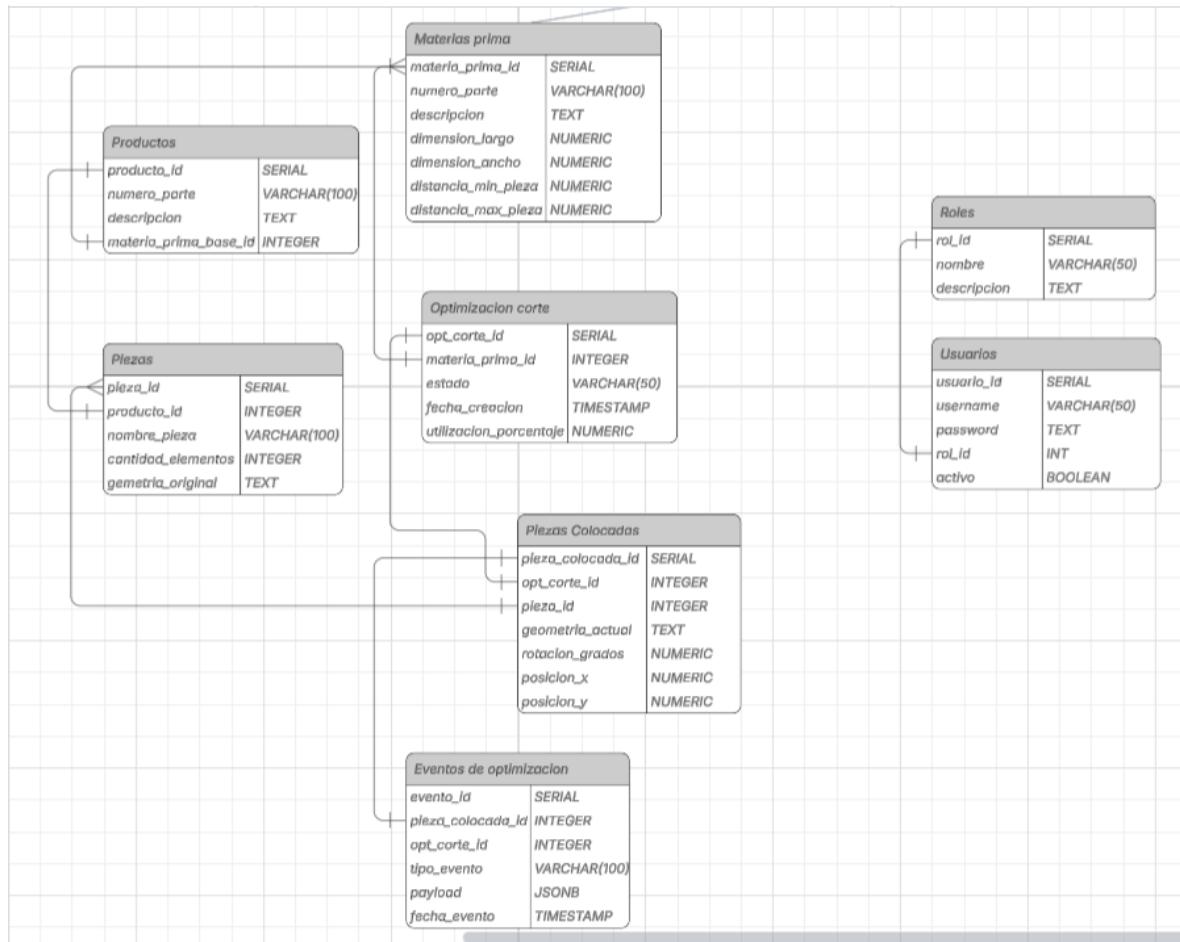


Tabla	Descripción
Roles	Tabla que define los distintos roles del sistema (Administrador, Operador). Controla los permisos y alcances funcionales de cada usuario.
Usuarios	Tabla que almacena a los usuarios del sistema junto con el rol que tienen asignado.
Materias primas	Contiene los registros de láminas, maderas, chapas metálicas u otros materiales disponibles para corte.
Productos	Tabla que representa un producto ensamblable que requiere piezas específicas cortadas de materia prima.
Piezas	Tabla que define cada pieza perteneciente a un producto.
Optimización de corte	Tabla que representa una corrida o sesión de optimización del corte.

Piezas colocadas	Registra cada pieza colocada dentro de una sesión de optimización, con su rotación y posición actual.
Eventos de optimización	Registro histórico de eventos aplicados durante la optimización, en formato JSON.

Procedimientos almacenados

Procedimiento sp_alta_materia_prima: Registra una nueva entrada de materia prima en el catálogo general y permite controlar inventario base para procesos de corte y optimización.

Recibe ID de pieza, ángulo de rotación, coordenadas de posición y evento JSON.

Comportamiento

- Inserta un nuevo registro en la tabla materia_prima.
- Realiza validaciones básicas: número de parte único, dimensiones válidas.
- Puede ser usado por administradores para dar de alta nuevos materiales.

Procedimiento sp_alta_producto: Crea un nuevo producto, definiendo su número de parte, descripción, materia prima asociada y su pieza base.

Recibe de parámetros número de parte, descripción, ide de materia prima, nombre de pieza, cantidad de elementos y geometría original.

Comportamiento

- Inserta el producto.
- Inserta automáticamente su pieza base.
- Garantiza consistencia transaccional.

Procedimiento sp_rotar_posicionar_figuras

Aplica transformaciones de rotación y posicionamiento a una pieza colocada dentro de una optimización de corte. Además, registra el evento JSON asociado.

Recibe ide de pieza colocada, angulo de rotación, posición en x y en y, eventos en json

Comportamiento

- Obtiene la optimización de corte a la que pertenece la pieza.
- Genera la nueva geometría simulada con la rotación/posición aplicada.
- Actualiza la pieza colocada.
- Inserta un registro en eventos optimización.

Procedimiento auxiliar sp_registrar_evento_json

Registra un evento JSON en la tabla eventos_optimizacion.

Recibe el id de optimización de corte, id de piezas colocadas, tipos de evento.

Comportamiento

- Inserta un evento completo y consistente.

Funciones

Función fn_calcular_utilizacion

Calcula el porcentaje de aprovechamiento de la materia prima usada en una optimización.

Recibe el id de optimización de corte.

Comportamiento

- Recupera la materia prima base.
- Calcula el área total de todas las piezas colocadas.
- Calcula la utilización.

Triggers

Validación de Distancia Mínima entre Piezas (trg_validar_distancia_piezas)

Tabla asociada: piezas_colocadas, se ejecuta cuando lee “BEFORE INSERT OR UPDATE”.

Propósito:

- Validar que una pieza colocada no infrinja la distancia mínima respecto a otras piezas ya ubicadas en la superficie de corte.
- Protege contra solapamiento y violaciones de restricciones de diseño.

Actualización automática de aprovechamiento (trg_actualizar_utilizacion)

Tabla asociada: piezas_colocadas, se ejecuta cuando “AFTER INSERT OR UPDATE OR DELETE”.

Propósito:

- Recalcular automáticamente el porcentaje de utilización de la materia prima cada vez que una pieza es colocada, movida o eliminada

Registro Automático de Eventos (trg_registrar_evento)

Tabla asociada: piezas_colocadas, se ejecuta cuando “AFTER UPDATE”.

Propósito:

- Registrar automáticamente un evento JSON indicando cambios en posición, rotación o geometría de una pieza.

Validación de Geometría (trg_validar_geometria)

Tabla asociada: piezas, se ejecuta cuando “BEFORE INSERT OR UPDATE”

Propósito:

- Garantizar que la geometría registrada en la pieza contenga un formato válido (por ejemplo, WKT o estructura JSON definida).

Restricción de Orilla en Materia Prima (trg_validar_orilla)

Tabla asociada: piezas_colocadas se ejecuta cuando “BEFORE INSERT OR UPDATE”

Propósito:

- Asegurar que las piezas colocadas respeten la distancia mínima a los bordes de la materia prima, evitando cortes defectuosos.

Auditoría de CRUD en Usuarios (trg_auditoria_usuarios)

Tabla asociada: usuarios, se ejecuta cuando “AFTER INSERT OR UPDATE OR DELETE”

Propósito:

- Registrar en una bitácora interna cambios hechos sobre usuarios del sistema.

Manejo de eventos

Función para procesar eventos JSON (fn_procesar_evento_json)

Esta función recibe un objeto JSON con instrucciones de transformación (rotación, traslación, ajustes geométricos).

Valida la estructura del JSON, extrae los datos relevantes y devuelve un resultado estandarizado que puede ser usado por procedimientos internos.

Recibe evento tipo JSON.

Comportamiento:

- Verifica si el JSON contiene claves mínimas como: "angulo", "posición en x", "posición en y".
- "motivo"
- Extrae los valores con operadores JSON (->, ->>).
- Devuelve un registro estructurado con la información relevante.
- En caso de estructura inválida, devuelve un mensaje de error controlado.

Registro de eventos de aplicar rotación/configuración (p_rotar_posicionar_figuras).

Cada vez que una pieza es rotada o reposicionada, se almacena un evento en formato JSON dentro de la tabla de auditoría del sistema.

Comportamiento:

- La pieza es rotada o reposicionada.
- El procedimiento genera una nueva geometría.
- Se registra un evento JSON con toda la meta data.
- El evento queda disponible para auditoría o análisis.

Conclusión Técnica

El desarrollo de este proyecto permitió implementar un sistema completo para la optimización del corte de materia prima laminar utilizando PostgreSQL como lenguaje principal. Los resultados fueron favorables y educativas, ya que se logró construir una arquitectura modular que integra tablas normalizadas, procedimientos almacenados, funciones de cálculo, triggers automáticos y manejo de eventos en formato JSON, cumpliendo con el flujo operativo esperado para un sistema de corte y aprovechamiento de materiales.

Sin embargo, se identificaron algunas limitaciones asociadas al entorno de desarrollo:

dado que el proyecto fue ejecutado en un compilador en línea, las operaciones relacionadas con creación de roles y asignación de privilegios del sistema no pudieron ser implementadas directamente desde PostgreSQL. Estas configuraciones tuvieron que documentarse y plantearse de manera externa, lo que restringió la posibilidad de probar completamente el módulo de seguridad dentro del mismo entorno.

Aun con estas limitaciones, la solución final demuestra la viabilidad del modelo relacional propuesto, la correcta integración de funciones y triggers, y su aplicabilidad en escenarios reales de optimización de corte.

Manual de Usuario del Sistema

Este manual está diseñado para guiar al personal en el uso correcto de las funciones de **Seguridad, Catálogos, y Operación** del sistema de optimización, detallando las responsabilidades de cada rol.

1. Perfiles de Usuario y Acceso

El sistema implementa un modelo de seguridad basado en roles, lo que garantiza el control de acceso y la asignación de privilegios.

Rol	Descripción	Tareas Funcionales Clave
Administrador	Posee acceso total al sistema de gestión de usuarios y a la configuración de catálogos.	Dar de alta/baja de usuarios, ejecutar procedimientos de modificación de materia prima y productos, y configurar el sistema.
Operador	Usuario principal responsable de la ejecución de cortes y el registro de la optimización.	Iniciar sesiones de corte, mover/rotar piezas, registrar eventos, y ver resultados de corte.

Proceso de Login y Seguridad

- **Autenticación:** El sistema utiliza una función de seguridad (`fn_autenticar_usuario`) que verifica su identidad. Sus contraseñas son almacenadas mediante **Hashing seguro** (`pgcrypto`) y nunca en texto plano.
- **Gestión de Cuentas (Admin):** Los administradores son responsables de crear usuarios (`sp_crear_usuario`) y asignarles un rol.

2. Módulo de Catálogos (Rol: Administrador)

Este módulo es la base para las operaciones de corte, ya que define los materiales y las piezas a utilizar.

A. Alta de Materia Prima

El Administrador debe utilizar el procedimiento **sp_alta_materia_prima** para registrar cada tipo de material laminado.

Campo	Propósito	Regla Crítica (Aplicada por el Trigger)
Dimensiones	dimension_largo y dimension_ancho del material.	Utilizadas para calcular el Porcentaje de Utilización .
Distancia Mínima a Orilla	Margen de seguridad (ej. 10.0 mm) que el sistema debe respetar entre la pieza cortada y el borde del material.	Este valor es obligatorio y el <i>trigger</i> lo valida automáticamente.

B. Alta de Productos y Piezas

El Administrador usa el procedimiento **sp_alta_producto** para registrar un producto, asociándolo con la materia prima base y su pieza geométrica.

- **Producto:** Se registra el número de parte y la descripción del artículo.
- **Pieza:** Automáticamente se registra la pieza base, su geometría original y la cantidad de elementos requeridos.

3. Módulo de Operación (Rol: Operador)

El Operador interactúa directamente con la superficie de corte para lograr la máxima eficiencia.

A. Posicionamiento, Rotación y Auditoría

La tarea principal del Operador es posicionar las piezas. Esto se realiza llamando al procedimiento **sp_rotar_posicionar_figuras**

Acción del Operador	Lógica de BD Invocada	Propósito
Mover y Rotar Pieza	sp_rotar_posicionar_figuras	Actualiza la posicion_x, posicion_y y la rotacion_grados de la pieza.
Registro de la Acción	Registro de Eventos JSON	Automáticamente inserta un registro en eventos_optimizacion con la metadata de la acción.

Auditoría con JSONB: Cada movimiento genera un registro histórico inmutable. Esto permite:

- Reconstrucción de escenarios.
- Auditoría de quién realizó la modificación (si el payload JSON incluye el usuario).

B. Validación Automática (Triggers)

El sistema garantiza la integridad del proceso de corte a través de *triggers* automáticos que previenen errores y pérdidas de material.

Regla de Negocio	Trigger Aplicado	Consecuencia para el Operador
Restricción de Orilla	tr_validar_colocacion_before (Validación de Orilla)	El sistema RECHAZA la colocación si la pieza está más cerca del borde que la distancia_min_orilla definida. Esto evita cortes defectuosos.
Distancia entre Piezas	trg_validar_distancia_piezas	El sistema RECHAZA la colocación si la nueva pieza se solapa o infringe la distancia mínima respecto a otras piezas ya colocadas.

El Operador nunca podrá guardar una posición que viole estas reglas, reforzando la seguridad y calidad del dato.

C. Monitoreo de Eficiencia

El objetivo del Operador es lograr la máxima eficiencia. La métrica se actualiza en tiempo real:

- **Cálculo:** La función `fn_calcular_utilizacion` calcula el porcentaje de aprovechamiento del material (Área total de piezas colocadas / Área total de la Materia Prima).
- **Actualización:** El *trigger* `tr_actualizar_utilizacion` se dispara automáticamente después de cada inserción, modificación o eliminación de una pieza, manteniendo el porcentaje de utilización siempre actualizado