

PROYECTO: PROCESO AUTOPARTES PLASTICAS

1. OBJETO

El presente proyecto tiene como objetivo el diseño de una base de datos relacional orientada a la gestión integral de los proyectos de producción de piezas plásticas dentro de una empresa proveedora de autopartes. Se identificaron los principales procesos involucrados en la fabricación, tales como el desarrollo de proyectos, el moldeo por inyección, los procesos de pintura y armado, y el control de materiales y componentes.

2. OBJETIVOS

- Centralizar la información del proyecto para consolidar una única fuente de consulta
- Permitir el seguimiento de las piezas desde su diseño hasta su producción final.
- Controlar los recursos involucrados: máquinas inyectoras, moldes, materiales, pintura y armado.
- Facilitar la trazabilidad de las piezas producidas para cada cliente.
- Generar una estructura flexible para posteriores consultas, reportes y análisis de desempeño.

3. PROBLEMÁTICA ACTUAL

Frente a la realización de un nuevo proyecto, la información se gesta a partir de lo comunicado por el área comercial y se distribuye a otras áreas relacionadas, como por ejemplo ingeniería de producto quien es el área que se dedica a la realización técnica del proyecto. A lo largo de cotización y negociación comercial, pueden existir variaciones de las piezas a desarrollar por lo cual, al no contar con una base de datos que consolide toda la información, las distintas áreas relacionadas pueden no recibir la revisión final del detalle de piezas por omisiones comunicacionales.

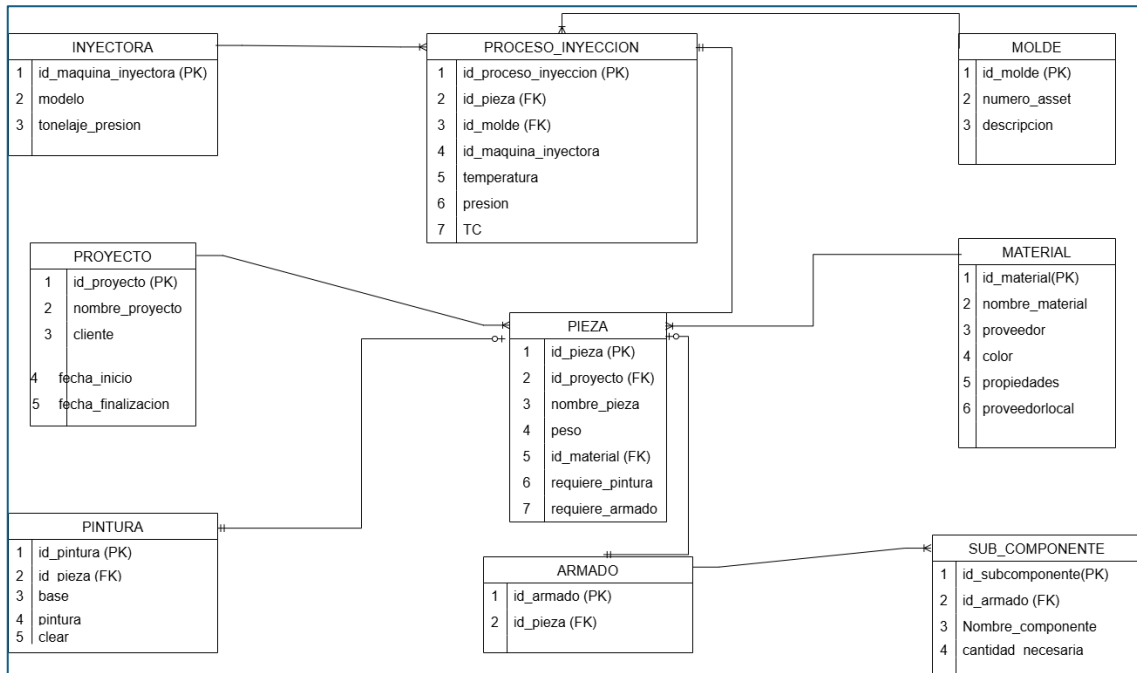
Todo esto se traduce en:

- Dificultad para planificar capacidad productiva.
- Limitada trazabilidad de los procesos de inyección, pintura y armado.

- Complejidad al gestionar proyectos que involucran múltiples áreas y proveedores.

La implementación de esta base de datos busca unificar la información, permitiendo un control eficiente de cada proyecto, pieza y sus procesos asociados.

4. DIAGRAMA DE ENTIDAD DE RELACIONES



5. ENTIDADES Y TIPOS DE DATOS (*corrección entrega 1*)

Cantidad de entidades: 9.

A continuación, se hará mención del nombre de la entidad, seguido por su script, la visualización de su configuración en Workbench y un archivo .xls con el detalle de las entidades con sus respectivos campos (se entrega como ANEXO por separado).

- **ARMADO**

CREATE TABLE armado (

id_armado INT PRIMARY KEY AUTO_INCREMENT,

id_pieza INT,

componente VARCHAR(100),

cantidad INT,

FOREIGN KEY (id_pieza) REFERENCES pieza(id_pieza)

);





Table Name:





Schema: **fabrica_autopartes**

Charset/Collation:

Engine:

Comments:



| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|--|--------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------|
|  id_armado | INT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
|  id_pieza | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  componente | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  cantidad | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

- **INYECTORA**

CREATE TABLE inyectora (

id_maquina_inyectora INT PRIMARY KEY AUTO_INCREMENT,

modelo VARCHAR(100),

tonelaje_presion VARCHAR(100)

);

[illegible]

- **SUB-COMPONENTES**

```
CREATE TABLE sub_componentes (
```

```
id_subcomponente INT PRIMARY KEY AUTO_INCREMENT,
```

id_armado INT,

nombre_componente TEXT,

cantidad_necesaria INT,

```
FOREIGN KEY (id_armado) REFERENCES armado(id_armado)
```

$$);$$
[illegible]

- MATERIAL

```
CREATE TABLE material (
```

```
id_material INT PRIMARY KEY AUTO_INCREMENT,
```

nombre_material VARCHAR(100) NOT NULL,

proveedor VARCHAR(100),

color VARCHAR(50),

propiedades VARCHAR(100),

proveedor_local TINYINT(1)

$$);$$
[illegible]

- MOLDE

```
CREATE TABLE molde (
  id_molde INT PRIMARY KEY AUTO_INCREMENT,
  numero_asset VARCHAR(100),
  descripcion TEXT
);
```

[illegible]






- **PIEZA**

```
CREATE TABLE pieza (  
    id_pieza INT PRIMARY KEY AUTO_INCREMENT,  
    id_proyecto INT,  
    nombre_pieza VARCHAR(100) NOT NULL,  
    peso DECIMAL(10,2),  
    id_material INT,  
    requiere_pintura TINYINT(1),  
    requiere_armado TINYINT(1),  
    FOREIGN KEY (id_proyecto) REFERENCES proyecto(id_proyecto),  
    FOREIGN KEY (id_material) REFERENCES material(id_material)  
);
```

[illegible]

- **PINTURA**

```
CREATE TABLE pintura (
    id_pintura INT PRIMARY KEY AUTO_INCREMENT,
    id_pieza INT,
    base VARCHAR(100),
    color VARCHAR(100),
    clear TINYINT,
    FOREIGN KEY (id_pieza) REFERENCES pieza(id_pieza)
);
```

| Table Name: <input type="text" value="pintura"/> | | Schema: fabrica_autopartes | | | | | | | | |
|--|--------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------|
| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|  id_pintura | INT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
|  id_pieza | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  base | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  color | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  clear | TINYINT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

- **PROCESO_INYECCION**

```
CREATE TABLE proceso_inyeccion (
    id_proceso_inyeccion INT PRIMARY KEY AUTO_INCREMENT,
    id_pieza INT,
    id_molde INT,
    id_maquina_inyectora INT,
    temperatura DECIMAL(10,2),
    presion DECIMAL(10,2),
    tiempo_ciclo DECIMAL(10,2),
    FOREIGN KEY (id_pieza) REFERENCES pieza(id_pieza),
    FOREIGN KEY (id_molde) REFERENCES molde(id_molde),
    FOREIGN KEY (id_maquina_inyectora) REFERENCES
    inyectora(id_maquina_inyectora)
```

);












Table Name:

Schema: **fabrica_autopartes**



| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|--|---------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------|
|  id_proceso_inyeccion | INT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
|  id_pieza | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  id_molde | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  id_maquina_inyectora | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  temperatura | DECIMAL(10,2) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  presion | DECIMAL(10,2) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  Tiempo_ciclo | DECIMAL(10,2) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

- **PROYECTO**

CREATE TABLE proyecto (

id_proyecto INT PRIMARY KEY AUTO_INCREMENT,

nombre_proyecto VARCHAR(100) NOT NULL,

cliente VARCHAR(100),

fecha_inicio DATE,

fecha_finalizacion DATE

);










Table Name:

proyecto

Schema:

fabrica_autopartes



| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|--|--------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------|
|  id_proyecto | INT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
|  nombre_proyecto | VARCHAR(100) | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
|  cliente | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  fecha_inicio | DATE | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
|  fecha_finalizacion | DATE | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

6. ENTREGA N°2

7. VISTAS

1. vista pieza material:

Tiene la funcionalidad de ver cuales son los distintos materiales de las piezas de la bd como así también si lleva color, cual es su proveedor y si es local o no.

SCRIPT:

create view vista_pieza_material as

select

p.nombre_pieza,

m.nombre_material,

```
m.proveedor,  
m.color,  
m.proveedor_local  
from pieza p  
join material m on p.id_material=m.id_material;
```

2. vista_Proyecto_piezas:

Su objetivo es poder ver de manera rápida que piezas pertenecen a cada proyecto junto con el número de activo (asset), que es el número de molde utilizado en el proceso de inyección.

SCRIPT :

```
create view vista_Proyecto_piezas as  
select  
p.id_proyecto,  
pi.nombre_pieza,  
piny.id_molde,  
m.numero_asset  
from proyecto p  
join pieza pi ON p.id_proyecto=pi.id_proyecto  
join proceso_inyeccion piny ON pi.id_pieza=piny.id_pieza  
join molde m ON piny.id_molde=m.id_molde;
```

3. vista_qpiezas_color_proyecto :

La idea es poder tener una vista donde se pueda ver la cantidad de piezas por color que lleva cada proyecto. Clave para empezar a analizar volúmenes necesarios por color.

SCRIPT :

```
create view vista_qpiezas_color_proyecto as  
select  
p.id_proyecto,  
pt.color,  
count(*) as cantidad_piezas  
from pieza p  
join pintura pt on p.id_pieza = pt.id_pieza  
where p.requiere_pintura = 1  
group by p.id_proyecto, pt.color;
```


4. vista_info_tecnica_pieza :

De gran utilidad para que se haga una bajada de informacion clara a los jefes de planta de como tiene que ser el seteo de cada inyectora en función de la pieza a inyectar para evitar retrabajos y piezas defectuosas.

SCRIPT:

```
create view vista_info_tecnica_pieza as

select

p.nombre_pieza,

m.nombre_material,

i.modelo,

i.tonelaje_presion,

py.temperatura,

py.presion,

py.tiempo_ciclo

from pieza p

join material m on p.id_material=m.id_material

join proceso_inyeccion py on p.id_pieza=py.id_pieza

join inyectora i on py.id_maquina_inyectora=i.id_maquina_inyectora;
```

5. vista_armado_completo :

Vista pensada para los jefes de planta de armado, donde se podrá ver los insumos sus cantidades necesaria por pieza de proyecto.

SCRIPT:

```
create view vista_armado_completo as

select

p.id_proyecto,

p.nombre_pieza,

a.componente as componente_principal,

s.nombre_componente as subcomponente,

s.cantidad_necesaria
```

```
from armado a
join pieza p on a.id_pieza = p.id_pieza
left join sub_componentes s on a.id_armado = s.id_armado;
```

8. FUNCIONES

- fn_cantidad_piezas_por_proyecto

SCRIPT :

```
delimiter $$
```

```
create function fn_cantidad_piezas_por_proyecto(id_proj int)
```

```
returns int
```

```
deterministic
```

```
begin
```

```
    declare total int;
```

```
    select count(*) into total
```

```
    from pieza
```

```
    where id_proyecto = id_proj;
```

```
    return total;
```

```
end$$
```

```
delimiter ;
```

```
DELIMITER ;
```

Esta función devuelve la cantidad total de piezas asociadas a un proyecto específico. Es útil para realizar reportes rápidos sobre el avance o la dimensión de cada proyecto en términos de desarrollo de componentes. Se la puede utilizar dentro de consultas más grandes o como métrica individual.

- fn_requiere_armado

SCRIPT :

delimiter \$\$

```
create function fn_requiere_armado(id_pieza_input int)
```

```
returns tinyint
```

```
deterministic
```

```
begin
```

```
    declare resultado tinyint;
```

```
    select requiere_armado into resultado
```

```
    from pieza
```

```
    where id_pieza = id_pieza_input;
```

```
    return resultado;
```

```
end$$
```

delimiter ;

DELIMITER ;

Esta función devuelve un valor booleano según si una pieza requiere o no proceso de armado. Útil para visualizaciones en vistas o reportes permitiendo separar piezas simples de piezas con aquellas que requieran una estructura ensamblada.

9. STORED PROCEDURE

- SP_DETALLE_PROYECTO

La idea de este procedimiento es que muestre una vista general de todas las piezas de un proyecto, incluyendo su peso y material. Tiene como finalidad el

seguimiento técnico y de proyecto. Sirve tanto para un Project manager, áreas de planificación y planta.

SCRIPT :

delimiter \$\$

```
create procedure sp_detalle_proyecto(in id_proyecto_input int)
```

```
begin
```

```
select
```

```
pr.nombre_proyecto,
```

```
p.nombre_pieza,
```

```
p.peso,
```

```
m.nombre_material
```

```
from proyecto pr
```

```
join pieza p on pr.id_proyecto = p.id_proyecto
```

```
join material m on p.id_material = m.id_material
```

```
where pr.id_proyecto = id_proyecto_input;
```

```
end$$
```

delimiter ;

- SP_PIEZAS_POR_MATERIAL

Devuelve todas las piezas que usan un material determinado. Ideal para proyectar volúmenes de material para planificación, compras o ingeniera.

SCRIPT:

delimiter \$\$

```
create procedure sp_piezas_por_material(in id_material_input int)
```

```
begin
```

```
select
```

```
p.id_pieza,  
p.nombre_pieza,  
m.nombre_material,  
m.color  
from pieza p  
join material m on p.id_material = m.id_material  
where m.id_material = id_material_input;  
end$$
```

delimiter ;

10. TRIGGERS

- trg_log_insert_pieza

Este trigger se activa automáticamente cada vez que se inserta una nueva pieza. Lo que hace es guardar en una tabla de log el nombre de la pieza y la fecha en que se insertó. Es útil para llevar un registro de cambios y saber cuándo se agregó una pieza nueva sin tener que cargar esa información a mano.

SCRIPT :

- TABLA LOG

```
create table log_insert_pieza (  
    id_log int auto_increment primary key,  
    id_pieza int,  
    nombre_pieza varchar(100),  
    fecha_insert datetime  
);
```

- Trigger

delimiter \$\$

```
create trigger trg_log_insert_pieza  
after insert on pieza
```

```
for each row  
  
begin  
  
    insert into log_insert_pieza (id_pieza, nombre_pieza, fecha_insert)  
  
    values (new.id_pieza, new.nombre_pieza, now());  
  
end$$
```

delimiter ;

- Trg_valida_peso_pieza

Este trigger se ejecuta justo antes de insertar una pieza, y verifica que el campo de peso no sea menor o igual a cero. En caso de que lo sea, la base de datos muestra un error y no permite cargar el dato. De esta forma se evita que ingresen valores que no tienen sentido o que después generen problemas en otros procesos.

SCRIPT:

```
delimiter $$
```

```
create trigger trg_valida_peso_pieza  
  
before insert on pieza  
  
for each row  
  
begin  
  
    if new.peso <= 0 then  
  
        signal sqlstate '45000'  
  
        set message_text = 'El peso de la pieza debe ser mayor a cero';  
  
    end if;  
  
end$$
```

```
delimiter ;
```

11. Link archivo .SQL :

https://github.com/Javiffa/Proceso_autopartes-Gonzalez-Fassi