

DESARROLLO DE LA BASE DE DATOS PARA UNA FERRETERIA INTERNACIONAL

1 Enunciado

La empresa Ferrecons & Contratistas Generales Zeol S.A.C. basa su actividad económica en la venta y distribución a nivel internacional de productos de construcción, generando ingresos rentables debido a la gran demanda que experimenta. Cuenta con una serie de procesos que le permiten estar en el mercado y tener una cartera de clientes que paulatinamente va en crecimiento. Todas sus actividades son realizadas de forma tradicional, llevando sus registros de ventas, stock, inventario, clientes, ingresos, egresos, etc. de forma rudimentaria. Esto genera una gran acumulación de papeles, la no disposición inmediata de la información para la toma de decisiones, el consumo de tiempo en cálculos de las ventas diarias, el inadecuado control de stock e inventario que se lleva y los problemas que surgen para dar cumplimiento a los pedidos hechos por los clientes ya que en ciertos casos tienden a retrasarse o a no proveer el material solicitado a tiempo, etc. Todo esto implica grandes pérdidas, deserción de clientes y un trabajo sumamente desordenado e inadecuado. Esto constituye el problema en general que afecta a la empresa.

A continuación, se muestran las áreas donde se ha logrado identificar problemas y posteriormente su solución a ellos:

Área de ventas, es el centro de la actividad económica de la empresa Ferrecons & Contratistas Generales Zeol S.A.C. donde se trata de satisfacer las necesidades de los clientes en cuanto a productos que estos requieran. Para ello, la empresa cuenta con un trabajador que, en ciertas ocasiones, no logra abastecerse debido a la gran demanda que se produce.

Área de pedidos, es un área muy importante ya que es a través de él que los clientes pueden hacer pedidos reservando sus productos para ser entregados en una fecha determinada. Para esto se hace uso de un cuaderno de registros de pedidos, donde se anotan los datos del cliente, los productos pedidos y la fecha que se le harán entrega. Se ha identificado a este proceso como un problema grande, ya que, en ciertas ocasiones se pasa la fecha de entrega de productos, y esto únicamente debido a no revisar el cuaderno de registros de pedidos por los pendientes que se tengan.

Área de stock o inventarios, es un área donde se ha podido identificar grandes problemas, empezando por la no disponibilidad de productos al realizar una venta o por la compra de productos repetidos a los proveedores, esto último hace que el dinero invertido esté estancado por mucho tiempo o al menos hasta que se consuma el producto duplicado.

Área de personal, permite tener detalladamente los datos de los actores que intervienen en el desarrollo de todos estos procesos (nombre, DNI, dirección, teléfono, sueldo, adelanto, deuda, etc.). Este es un problema claro en la empresa, porque no existe un registro alguno, ni siquiera en papel, de estos datos necesarios e indispensables para la empresa. Tampoco existe un registro que determine qué trabajador realiza ciertas ventas o anota pedidos. Esto debido a que la empresa se proyecta a tener más empleados para tratar de suplir la demanda de clientes.

2 IDENTIFICACION DE LAS REGLAS DE NEGOCIO

2.1. GESTIONAR DE SISTEMAS

- En primer lugar, se debe definir los parámetros base para los funcionamientos del sistema.
- Cada empleado tendrá asignado una cuenta de usuario del sistema según sus funciones y cargos.
- Habrá una persona encargada del mantenimiento de este modulo
- La contraseña de los usuarios por seguridad tendrá que ser formado por letras y números.
- El administrador definirá la información que es la correcta y velara por integridad.

2.2. CONTROLAR VENTAS

- El cliente podrá realizar pago solo con efectivo
- Se registran los clientes que se crean necesario según disposición de la administración.
- Los reportes de ventas se deben generar al final de cada turno.

2.3. CONTROLAR ALMACEN

- Se debe controlar el llenado del kardex.
- Se deben controlar las fechas de entradas y salida de bienes a almacén.
- Para sacar un producto de almacén debe haber un pedido de bienes.
- Se debe alertar la ausencia de stock de producto y realizar el requerimiento de compra para posteriormente hagan un pedido a proveedores.
- Todo producto debe estar estrictamente controlado en cuanto a fecha de vencimiento, registró de sanidad y estimar por prioridades.

2.4. CONTROLAR PEDIDOS

- Se debe controlar los pedidos realizados.
- Se deben controlar las fechas de entradas y salida de bienes a almacén.
- Para sacar un producto de almacén debe haber un pedido de bienes.
- Se debe alertar la ausencia de stock de producto y realizar el requerimiento

3. MODELO DE NEGOCIO

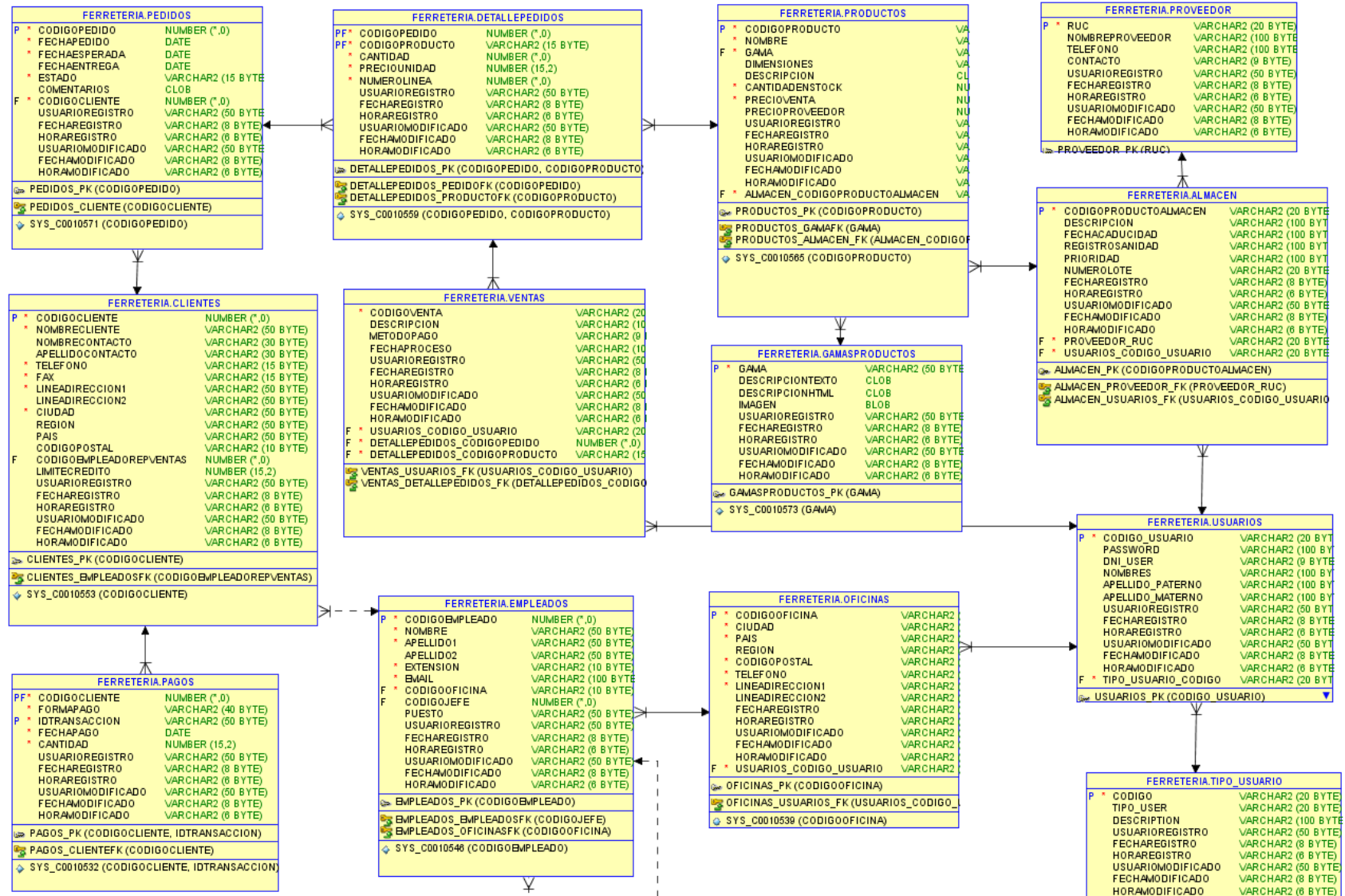
3.1.1. IDENTIFICACION DE ACTORES

- Clientes
- Proveedores
- Administrador
- Vendedor
- Almacenero

3.1.2. IDENTIFICACION DE CASOS DE USO DEL NEGOCIO

- Gestión de Sistemas
- Control de Ventas
- Controlar Almacén
- Controlar Pedidos

4. MODELO ENTIDAD RELACION DE LA BASE DE DATOS



5. CONSULTAS SIMPLES

1. Obtener el nombre y apellidos del jefe de la empresa.

```
select nombre, apellido1, apellido2
from empleados
where codigojefe is null;
```

2. Mostrar el nombre y cargo de los empleados que no sean directores de oficina

```
select nombre, puesto
from empleados
where lower(puesto) <> 'director oficina';
```

3. Muestra el número de clientes norteamericanos.

```
select count(*) as "Num clientes"
from clientes
where upper(pais) = 'USA';
```

4. Número de clientes de cada país.

```
select pais, count(*) as "Num clientes"
from clientes
group by pais;
```

5. Muestra el nombre del cliente y el nombre de su representante de ventas (si lo tiene).

```
select c.nombrecliente as "Nombre cliente", e.nombre as "Nombre representante"
from clientes c join empleados e
on c.codigoempleadorepventas=e.CODIGOEMPLEADO;
```

6. Nombre de los clientes que hayan hecho un pago en 2007

```
select distinct c.nombrecliente, to_char(p.FECHAPAGO,'YYYY') as 'Anio'
from clientes c join pagos p
on c.CODIGOCLIENTE=p.CODIGOCLIENTE
```

```
and to_char(p.FECHAPAGO,'YYYY') = '2007';
```

7. Los posibles estados de un pedido.

```
select distinct lower(estado)
from pedidos;
```

8. Muestra el número de pedido, el nombre del cliente, la fecha de entrega y la fecha requerida de los pedidos que no han sido entregados a tiempo.

```
select p.codigopedido, c.nombrecliente, p.fechaentrega, p.fechaesperada
from clientes c join pedidos p on c.codigocliente = p.CODIGOCLIENTE
and p.FECHAESPERADA < p.fechaentrega;
```

9. Muestra la ciudad y el código postal de las oficinas de España.

```
select ciudad, codigopostal
from OFICINAS
where lower(pais) = 'españa';
```

10. Muestra el número de empleados que hay en la empresa.

```
select count(*) as "Num empleados"
from empleados;
```

6. CONSULTAS COMPLEJAS

11. Muestra el código, nombre y gama de los productos que nunca se han pedido (detalle pedidos).

```
select p.codigoproducto, p.nombre, p.gama
```

```
from productos p JOIN GAMASPRODUCTOS g on p.gama=g.gama
and not exists (select CODIGOPRODUCTO from detallepedidos
where codigoproducto=p.CODIGOPRODUCTO);
```

12. Muestra el nombre y apellidos de los empleados que trabajan en Barcelona.

```
select e.nombre, e.apellido1 || ' ' || e.apellido2
from empleados e join oficinas o on e.codigooficina = o.codigooficina
and trim(lower(o.ciudad)) = 'barcelona';
```

13. Muestra el código y la cantidad de veces que se ha pedido un producto al menos una vez.

```
select p.codigoproducto, sum(dp.cantidad) as "cantidad pedida"
from productos p join detallepedidos dp on p.codigoproducto = dp.codigoproducto
group by p.codigoproducto;
```

14. Mostrar el código de los pedidos donde se haya vendido el producto de la gama ‘Aromáticas’ más caro.

```
from pedidos pe join detallepedidos dp on pe.codigopedido = dp.codigopedido
and dp.codigoproducto in (select codigoproducto
                        from productos
                        where precioventa = (
select max(precioventa)
from productos p, gamasproductos g
where p.gama = g.gama and lower(g.gama) = 'aromáticas'
));
```

15. Mostrar el código de los pedidos donde el precio del pedido sea superior a la media de todos los pedidos.

```
select pe.codigopedido
from pedidos pe
where
(select sum(dp.cantidad * dp.PRECIOUNIDAD) as total
 from pedidos p, detallepedidos dp
```

```

where p.codigopedido = dp.codigopedido and pe.codigopedido = p.codigopedido
group by p.codigopedido)
>
(select avg(t.total)
from (select p.codigopedido, sum(dp.cantidad * dp.PRECIOUNIDAD) as total
from pedidos p, detallepedidos dp
where p.codigopedido = dp.codigopedido
group by p.codigopedido) t);

```

16. Realiza una vista que muestre los datos de un empleado (nombre, apellidos, ciudad de la oficina) y lo mismo para su jefe (en la misma fila).

```

create or replace view empleados_jefes as
select e_subor.nombre as nombre_subor,
       e_subor.apellido1 || ' ' || e_subor.apellido2 as apellidos_subor,
       o_subor.ciudad as ciudad_subor,
       e_jefe.nombre as nombre_jefe,
       e_jefe.apellido1 || ' ' || e_jefe.apellido2 as apellidos_jefe,
       o_subor.ciudad as ciudad_jefe
from empleados e_subor, empleados e_jefe, oficinas o_subor, oficinas o_jefe
where e_subor.codigojefe = e_jefe.codigoempleado
and o_subor.codigooficina = e_subor.codigooficina
and e_jefe.codigooficina = o_jefe.CODIGOOFICINA;

```

17. Realiza una vista que muestre el código de pedido y su total en euros.

```

create or replace view pedidos_total as
select p.codigopedido, sum(dp.cantidad * dp.PRECIOUNIDAD) as total
from pedidos p, detallepedidos dp
where p.codigopedido = dp.codigopedido
group by p.codigopedido;

```

18. Realiza una vista con la información del pedido (código, fecha pedido, fecha esperada, fecha entrega, nombre cliente y total en euros) ordenado por total de forma descendente.

```

create or replace view info_pedido as
select p.codigopedido,
       p.fechapedido,
       p.fechaesperada,
       p.fechaentrega,
       c.nombrecliente,

```



```
pt.total
from pedidos p, clientes c, PEDIDOS_TOTAL pt
where p.CODIGOCLIENTE = c.CODIGOCLIENTE
and pt.CODIGOPEDIDO = p.CODIGOPEDIDO
order by pt.total desc;
```

19. Devolver la gama de productos más vendida. Usando vistas

```
create or replace view gamas_vendidas as
select p.gama, sum(dp.cantidad) as cantidad
from detallepedidos dp, productos p
where p.codigoproducto = dp.codigoproducto
group by p.gama;
```

```
select gv.gama, gv.cantidad
from gamas_vendidas gv
where gv.CANTIDAD = (select max(gv.cantidad)
from gamas_vendidas gv);
```

20. Muestra el país(cliente) donde menos pedidos se hacen.

```
create or replace view pedidos_paises as
select c.pais, count(*) as num_pedidos
from clientes c, info_pedido ip
where c.nombrecliente = ip.nombrecliente
group by c.pais;

select pp.pais, pp.num_pedidos
from PEDIDOS_PAISES pp
where pp.num_pedidos = (select min(num_pedidos) from PEDIDOS_PAISES);
```