

# Implementación de un Sistema de control de bodega inteligente usando Oracle DataBase

ARMANDO MARTÍNEZ, CESAR ESQUIVEL ESCAMILLA, JOSÉ EDUARDO MÉNDEZ

**Resumen**— El proyecto que se presenta está basado en una empresa de pedidos de diferentes productos, ya que se puede adaptar a las necesidades del cliente, en donde se pueden hacer pedidos dando un tiempo estimado de envío, lo que se busca con esta aplicación es que se pueda tener un excelente control de los envíos y así evitar productos caducos controlando los envíos y las fechas de caducidad, dando prioridad a los envíos con más tiempo de pedido y productos con más tiempo en almacén. Todo esto se logra con la implementación de una base de datos en Oracle, la cual se conecta a una interfaz agradable y de fácil manejo para el cliente.

## I. INTRODUCCIÓN

Una de las motivaciones para éste trabajo radica en la necesidad de tener un mejor control en las fábricas, almacenes y envíos de diferentes productos para las empresas que manejan éstos rubros, ya que se desperdicia mucha materia prima en las fábricas por el mal control de ella, así como, productos por el lapso de almacenamiento por la falta de control de tiempo en almacén y por lo tanto se presenta la pérdida en el tiempo y logística en los envíos por la misma falta de control, por lo cual, en éste proyecto se busca resolver u optimizar cada uno de estos problemas para lograr un sistema de control de bodegas inteligente obteniendo como principales ventajas:

- Disminución de los costos de operación
- Disminución de errores en los procesos
- Optimización de recursos.
- Agilizar las tareas para cada área
- Mejora en el servicio al cliente

Cada una de estas ventajas se desarrollan y optimizan a lo largo del proyecto buscando como meta un sistema bastante robusto que mejore el control de otras áreas como son: la parte administrativa y el control de clientes principalmente.

## II. PLANTEAMIENTO DEL PROBLEMA

En este trabajo se desarrolla un software que permita apoyar la operación y realizar el control de la logística y distribución de los productos terminados de las materias

Armando Martínez, Cesar, José Eduardo Méndez pertenece a la carrera Ingeniería Cibernética y Sistemas Computacionales de la Facultad de Ingeniería y realizaron el proyecto dentro del curso Base de Datos y Modelado de Sistemas Dinámicos.

El proyecto fue asesorado por Juan Damián Silva y Zizilia Zamudio Beltran.

primas de las fábricas, para esto se han realizado diferentes implementaciones las cuales serán explicadas a continuación.

Se realizó una base de datos para poder llevar el control de los pedidos y de los clientes con el fin de una respuesta pronta, eficaz y una atención adecuada a los clientes. En la base datos desarrollada se incluye información de todos los componentes que relaciona el Sistema de Control de bodega inteligente como: la fábrica, materias primas, productos, envíos, transporte, pedidos, bitácoras, máquinas de fábrica, almacén, empleados, proveedores, clientes, etcétera realizando una tabla con sus respectivas relaciones para cada necesidad que se detectaron y las cuales pudieran llegar a ser útiles.

La parte de las relaciones y explicación de las tablas se detallan más adelante para poder entender el porqué de estas y la relación entre cada una de ellas.

Además, se realizó una interfaz gráfica la cual estará conectada con la base datos que permite dar de alta a un cliente, eliminar, así como, le permite al cliente realizar pedidos, consultar dichos pedidos o en dado caso eliminar un pedido ya hecho que no esté en trayecto. Más adelante se demuestra con imágenes como se diseñó la interfaz gráfica pensando en que fuera agradable para el cliente, buscando que sea fácil y sencilla de usar.

## III. VALOR AGREGADO

Para innovar el proyecto se decidió agregar el generar un PDF en el momento que se genera un pedido, el cual despliega una pantalla para agregar la ruta y el nombre del PDF, también, el mismo pedido se envía por correo al cliente solicitando la dirección de email al cual se enviará dicho correo.

## IV. MARCO TEÓRICO

A continuación, se desarrollan los conceptos básicos para la propuesta presentada en este trabajo.

Oracle Database es un sistema de gestión de base de datos de tipo objeto, desarrollado por Oracle Corporation [1], [2] y [3].

Se hizo uso de una base de datos en Oracle DataBase la cual está compuesta por 4 partes, **Utilerías, Usuarios, DBMS y Datos**, teniendo como prioridad de la base de datos evitar la contradicción de datos, así como duplicación de estos:

- **Utilerías.** - Acceso a la base de datos visualizar gráficamente
- **Usuarios.** - Accedan a los datos de la base de datos, Programadores de APP, Usuarios finales, DBA.

- **DBMS.** - Manejador de la base de datos
- **Datos.** - Consultar, recopilar datos organizados protegidos por el manejador deben de ser integrados y compartidos

Para poder utilizar la base datos se hace uso del **lenguaje SQL** para:

• **DDL (Data Definition Language):** se utiliza para crear y definir los campos de cada tabla.

• **DML (Data Manipulation Language):** se utiliza para agregar o modificar datos de las tablas ya creadas.

Una tabla contiene la llave primaria que es una identificación individual única de las ocurrencias de una entidad, los campos que no sean llaves deberán de ser hechos acerca de la ocurrencia y cada hecho deberá ser representado solo una vez en la Base de Datos para evitar duplicación

Todas las tablas deben estar **normalizadas** en las siguientes 3 formas:

• **Primera Forma Normal:** todos los campos son atómicos

• **Segunda Forma Normal:** todos los campos deben llevar llave primaria.

• **Tercera Forma Normal:** No dependencia transitiva entre datos.

El diagrama entidad-relación es la representación grafica de las tablas y sus respectivas relaciones (Diseño Lógico y Físico).

**Transacciones** es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica, los que se ocupan son:

- **BEGIN:** Especifica que va a empezar una transacción.
- **COMMIT:** Le indica al motor que puede considerar la transacción completada con éxito.
- **ROLLBACK:** Indica que se ha alcanzado un fallo y que debe restablecer la base al punto de integridad.

**Trigger** es un bloque de código que se ejecuta automáticamente cuando ocurre algún evento (como inserción, actualización o borrado) sobre una determinada tabla (o vista), es decir, cuando se intenta modificar los datos de una tabla asociada al disparador. Si se intenta modificar (agregar, actualizar o eliminar) datos de una tabla asociada a un disparador, el disparador se ejecuta en forma automática.

---

**Algoritmo 1.** Sintaxis general para crear un disparador

---

```
create or replace trigger NOMBREDISPARADOR
MOMENTO-- before, after o instead of
EVENTO-- insert, update o delete
of CAMPOS-- solo para update
on NOMBRETABLA
NIVEL--puede ser a nivel de sentencia (statement) o de fila (for each
row)
when CONDICION--opcional
begin
CUERPO DEL DISPARADOR--sentencias
end NOMBREDISPARADOR;
```

---

Para la interfaz gráfica se utilizó el lenguaje JAVA donde se pusieron en práctica los conocimientos obtenidos en semestres pasados para realizar el proyecto.

**Java** es la base para prácticamente todo tipo de aplicaciones en red y es el estándar mundial para desarrollar y entregar software empresarial, contenido web, juegos y aplicaciones móviles. Java disfruta de un ecosistema grande y maduro con un fuerte soporte de herramientas. Java ofrece portabilidad de aplicaciones y un rendimiento robusto en muchos entornos informáticos [4] y [5].

**La interfaz de programa de aplicación (API)** es un conjunto de rutinas, protocolos y herramientas para crear aplicaciones de software. Además, las API se utilizan al programar componentes de interfaz gráfica de usuario (GUI). Un programador entonces pone los bloques juntos.

**GUI** es un tipo de interfaz de usuario que permite a los usuarios interactuar con dispositivos electrónicos a través de iconos gráficos e indicadores visuales. Las acciones en una interfaz gráfica de usuario generalmente se realizan mediante la manipulación directa de los elementos gráficos.

**Las clases**, son básicamente una plantilla que sirve para crear un objeto. Una clase es el molde de una galleta y la galleta como tal es el objeto. Cada clase tiene una respetiva función asignada por el programador, para crear una clase primero se establece que tan privada (public, private, protected) será para otros clases o programas

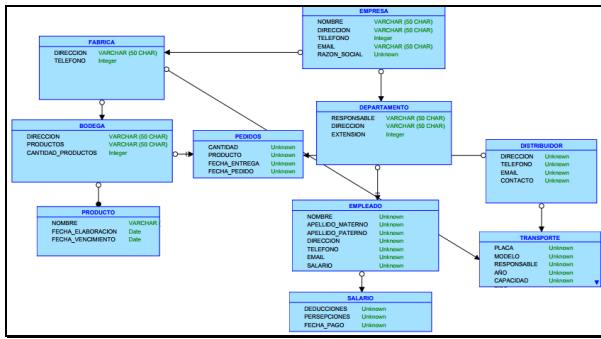
**La herencia**, es una forma de reutilización de software en la que se crea una nueva clase absorbiendo los miembros de una clase existente, y se mejoran con nuevos métodos, o con modificaciones en los métodos ya existentes.

**Polimorfismo**, nos permite escribir programas que procesen objetos que compartan la misma superclase en una jerarquía de clases, como si todos fueran objetos de la superclase; esto puede simplificar la programación.

## V. DESARROLLO DE LA BASE DE DATOS

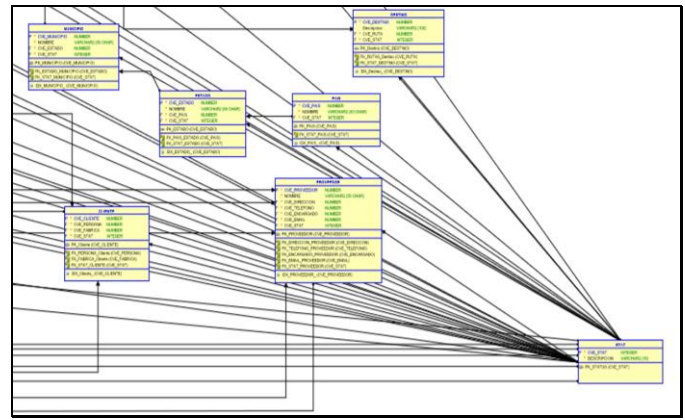
Para hacer posible un mejor control en la fábrica, almacenes y envíos de diferentes productos se creó una base de datos la cual contiene 38 tablas que están relacionadas entre sí para lograr nuestro objetivo para una buena organización y rendimiento.

Para poder comenzar primero se creó un diagrama lógico, el cual contiene las tablas básicas, que nos dan la pauta para la creación del total de las tablas (Imagen 1).



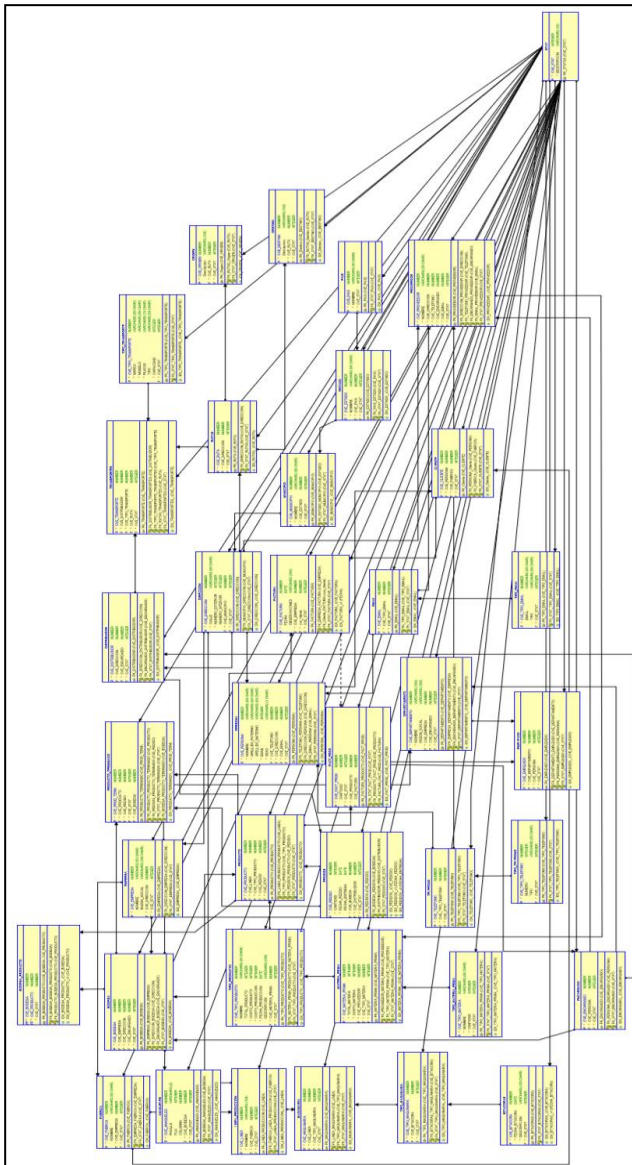
**Imagen 1 | Diagrama Lógico**

Una vez terminado el diagrama lógico se realizó el diagrama físico (Imagen 2 y 2.2), el cual es más robusto y completo ya que incluye el total de las tablas se usan en la base de datos y las relaciones con cada tabla.



**Imagen 3.2 | Diagrama Físico**

Todas las tablas en la base de datos contienen clave primaria y relación con la tabla status, estas tablas son las que contiene el diagrama físico son:



**Imagen 2.1 | Diagrama Físico**

- Anaqueles tiene pasillo y fila donde se encuentra el anaquel.
- Bitácoras tiene la fecha y descripción de las bitácoras.
- Bodega tiene de que empresa es, dirección de la bodega, encargado y productos que almacena.
- Cliente tiene los datos del cliente que se encuentran en persona y dirección.
- Departamento tiene nombre y razón social y el encargado.
- Destino tiene descripción.
- Dirección que contiene Calle, número exterior, número interior y a su vez dividido en:
  - Municipio contiene nombre.
  - Estado contiene nombre.
  - País contiene nombre.
- Distribuidor tiene la tabla dirección y encargado del distribuidor.
- Email que contiene una la relación con la tabla:
  - Tipo Email contiene email.
- Empleado tiene los datos del empleado que se encuentran en persona y dirección.
- Empresa nombre, razón social de la empresa y dirección
- Encargado tiene los datos del encargado que se encuentran en persona y dirección.
- Fabrica tiene el nombre de las fábricas y su empresa
- Factura tiene fecha, observaciones y una relación con la tabla:
  - Factura producto que contiene la cantidad
- Línea Producción tiene nombre y la fabrica
- Maquinaria tiene relación con línea de producción y una tabla:
  - Tipo de maquinaria que tiene la relación con bitácora.
- Materia prima tiene tal de materia, costo con materia, una relación con proveedor y una tabla: y tiene nombre, total, costo de la materia prima.
  - Tipo de materia que contiene nombre y cantidad.
- Origen tiene descripción y relación con rutas.

- Pedidos tiene cantidad de pedidos, fecha de pedido y fecha de entrega, así como sus relaciones.
- Persona que contiene nombre, apellido paterno, apellido materno, edad y sexo y relación con email y teléfono.
- Producto tiene relación con línea de producción, pedido y una tabla:
  - Tipo producto que tiene nombre, total producto, costo producto, costo producción, fecha de producción y descripción.
- Producto terminado tiene sus respectivas relaciones con producto, pedido y bodega.
- Proveedor tiene nombre y relación con la tabla dirección, email y teléfono.
- Rutas tiene relación con dirección
- Teléfono que contiene una relación con la tabla:
  - Tipo Teléfono tiene el teléfono y tipo
- Transporte tiene las relaciones con distribuidor rutas y una tabla:
  - tipo de transporte que contiene marca, modelo, placas, tipo y capacidad.
- Status tiene descripción.

El diseño de las tablas históricas estas relacionadas con los triggers y ayudan a tener documentado las modificaciones que se le hacen a dichas tablas, y cada una de ellas tiene su tabla histórica.

VI. RESULTADOS. INTERFAZ GRAFICA

Para la parte de la interfaz se utilizó en el lenguaje JAVA en el programa de Netbeans, se implementó de tal manera que fuera una interfaz agradable y sencilla para quien la ocupe, pero no dejando de lado la seguridad que se necesita por el tipo de información que se maneja.

Como pantalla inicial se pide un usuario y contraseña que estén dadas de alta previamente en la base de datos, ver Imagen 3.

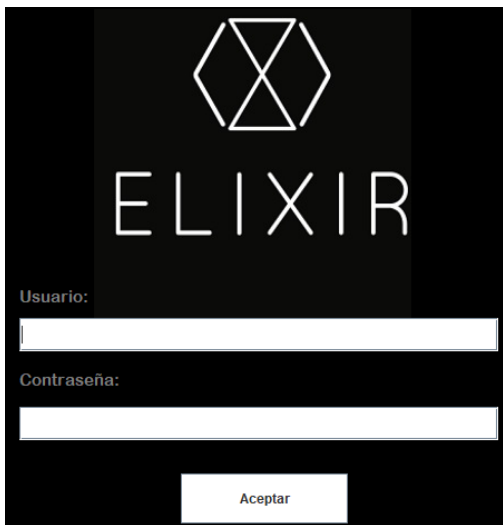


Imagen 4 | Login

El uso de una contraseña se implementó con el fin de tener un control riguroso con la información guardada en la base de datos, si el usuario no existe se desplegará un mensaje de error, de lo contrario le aparecerá la pantalla inicial donde tenemos dos opciones que son *Cientes* y *Pedidos*, como se muestra en la Imagen 4.

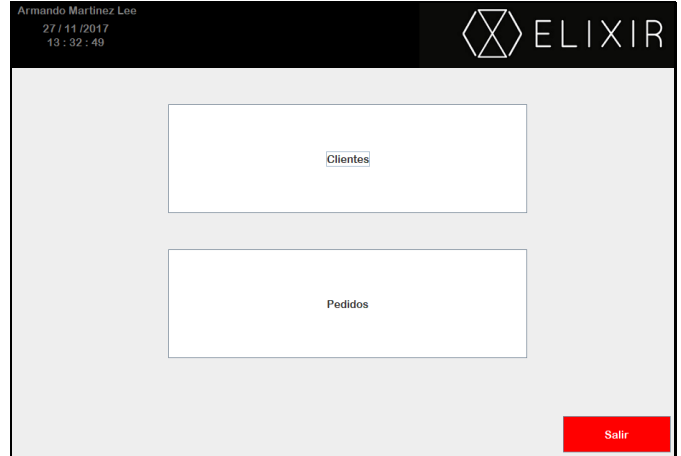


Imagen 5 | Pantalla Inicial

Una vez en la pantalla si se elige “Clientes” se despliegan tres opciones: *Alta de cliente*, *Baja de cliente* y *pedidos*, tal como se muestra en la Imagen 5.

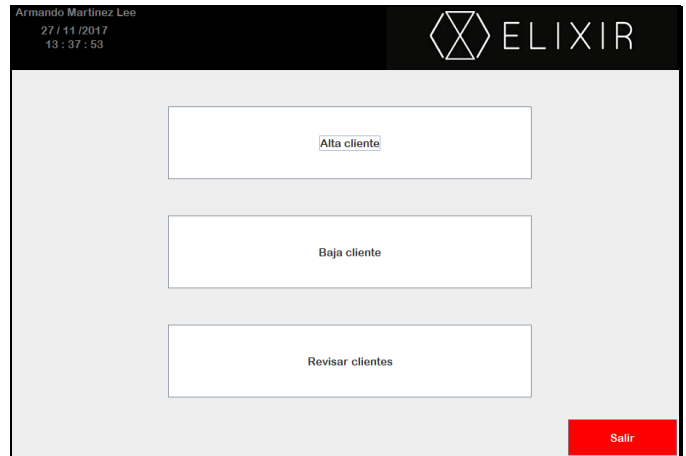
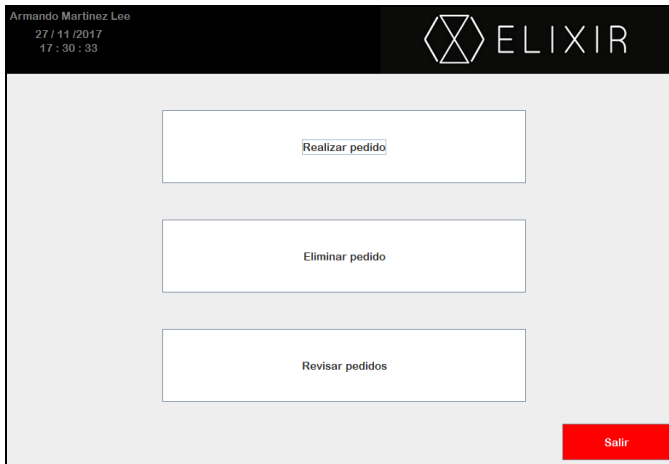


Imagen 6 | Menú clientes

Para estos tres casos se tiene lo siguiente:

- En *alta de cliente* se despliega una pantalla para poder generar un nuevo usuario con la información que la base de datos necesita.
- En *baja de cliente* se tiene la capacidad de eliminar a un usuario que este previamente registrado, con la clave que cada cliente tiene.
- En *revisar al cliente* se logra hacer la revisión de la lista de clientes que se tiene en la base de datos.

De la pantalla inicial si se elige en pedidos se despliega el menú de pedidos, el cual también consta de tres elecciones como se muestra en la Imagen 6.



**Imagen 6 | Menú Pedido**

Para estos tres casos se tiene lo siguiente:

- En *realizar pedido* aparece una pantalla que permite generar un nuevo pedido con la información que la base de datos necesita.
- En *eliminar pedido* permite eliminar un pedido que este previamente cargado en la base de datos con la clave que cada pedido tiene.
- En *revisar pedido* se puede ver el estatus de la lista de pedidos que está registrada en la base de datos.

## VII. CONCLUSIONES

Se elaboró una base de datos e interfaz completa con los requerimientos necesarios y valores agregados descritos con anterioridad, dicha base de datos se implementó en Oracle DataBase.

En el comienzo de este proyecto se tuvo el reto de diseñar la propuesta de diseño a gran escala para poder obtener una gran cantidad de tablas y así tener un proyecto robusto, una vez con las tablas surgió la problemática de las relaciones, ya que era necesario evitar relaciones no útiles o que faltaran para el buen desarrollo del proyecto los cuales se resolvieron exitosamente.

Para la parte de los triggers se creaba una secuencia y las relaciones a esa tabla ya lo tenían, entonces no se tenía que crear una secuencia en esas tablas.

En los insterts, la complejidad se presentó en la información que se iba a registra, ya que, si se colocaba la clave 1 a todas las tablas, éstas se relacionaban entre todas y ese no era el fin, por lo que se tuvo que revisar las tablas como persona, dirección, etcétera.

Para conectar la base de datos se presentó el problema con el puerto 1521 ya que no estaba activado, debido a que se debía estar conectado al servidor de la Salle para la tabla de usuarios, donde se encontraba la contraseña de cada usuario encriptado y este problema se resolvió comparando la contraseña encriptada.

## REFERENCIAS

- [1] Deitel H. M & Deitel P. J. (2008). *Como programar Java* (7ma ed.). México: Pearson.
- [2] Gustavo Guillermo Pérez. (2008). *Aprendiendo Java y Programación Orientada a Objetos*. Cd Mexico, Aragon #28: Claren.
- [3] Oracle. (2008, 2017). *SQL Developer Data Modelar User's Guide*. Noviembre 15, 2017, de Oracle Sitio web: <https://docs.oracle.com/database/sql-developer-17.3/DMDUG/toc.htm>
- [4] Oracle. (2008,2017). *Statements*. Noviembre 23 de 2017, de Oracle Sitio web: [http://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_7004.htm#sthref7885](http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#sthref7885)
- [5] Oracle. (2008,2017). *SQL Developer User's Guide*. Noviembre 11 de 2017, de Oracle Sitio web: <https://docs.oracle.com/database/sql-developer-17.3/RPTUG/toc.htm>