

Análisis, diseño, implementación y gestión de un sistema de bases de datos para automatizar los procesos de facturación de la empresa “OKI DOKI”.

Henry Recalde¹; Walter Zambrano²

¹ Universidad Tecnológica Israel, Quito-Ecuador, hrecalde@uisrael.edu.ec, ²Universidad Técnica De Manabí, Manabí-Ecuador, walter.zambrano@utm.edu.ec

Resumen: Las razones por las que se decidió automatizar los requerimientos de esta empresa es porque nos incentiva a desarrollar un poco más nuestra manera de razonar en base a la programación o digitalización de código en MySQL.

Mediante un Sistema de Gestión de Bases de Datos de Desarrollo Se trabajó con la plataforma de Power Designer para el diseño del esquema conceptual, también con la aplicación informática access conjuntamente con MySQL Administrador para la conexión de la Base de Datos.

Palabras clave: Tecnología, Java, Bases de Datos.

Abstract.

Most integrated development environments (IDEs) for Java make use of a project concept. A project in any IDE, is a global "container" where we can include or create all files that in one way or another will be used in our program, classes, interfaces, text files, images, Packages, etc.

A project in a development environment allows us to give order and classification to our work and our programs, thus avoiding that we are developing something with the files of different programs mixed in the same place.

1. INTRODUCCIÓN

La empresa “OKI DOKI” tiene una gran trayectoria de varios años prestando sus servicios en la provincia de Manabí, de tal manera la empresa ha crecido simultáneamente en el campo de compra y venta de productos, el dueño al ver este incremento ha tomado la decisión de automatizar su empresa, debido a la necesidad de obtener un respaldo de lo que se hace en su empresa, tales como facturas, clientes, proveedores, etc., y poder facilitar a sus trabajadores el rendimiento en su área de responsabilidad. De tal manera, el gerente necesita llevar un listado de todos los proveedores existentes y nuevos que lleguen al local, se necesita saber el nombre del proveedor, identificación, número de teléfono, si es que tiene, correo electrónico. También se necesita saber el tipo de producto que se deja en el local, ya que si hay alguna devolución se lo pueda llamar, para su previo cambio. Al momento que el proveedor entrega los productos al encargado de bodega, este se encarga de registrar todos los productos con su nombre, cantidad, precio unitario, código único, marca, descripción de utilización y nombre del proveedor. El gerente también necesita saber sobre los usuarios que llegan a su local, y les pide a los

encargados de venta que los registren para su previa facturación, y ellos necesitan saber el nombre del cliente, cedula de identidad, dirección, número de teléfono y correo electrónico, para que la factura también pueda llegar electrónicamente. Después que los encargados de venta hayan registrado a los usuarios, ellos se encargan de poder ver los productos en la bodega para ayudarles a los clientes sobre la existencia del mismo o no, un vendedor puede vender y solicitar un producto, si el producto no está en bodega el vendedor puede solicitar su compra. En la factura que se le entrega al cliente se necesita constatar el nombre del vendedor que le ha atendido al cliente, ya que pueda a ver un reclamo y el vendedor debe de hacerse cargo del problema, entonces el gerente necesita saber sobre sus empleados, su nombre, cedula de identidad, número de teléfono, correo electrónico, dirección donde residen, y dos referencias personales. Las referencias son obligatorias ya que el gerente es muy exigente con sus empleados, de las referencias solo se necesitan saber nombre, número de teléfono, fecha de la referencia y la dirección de donde vive, un empleado puede ser referenciado de un cliente del local. Después que se haya cumplido

con todas las normas el empleado se necesita crear un código único para su identificación y llevar su control dentro del establecimiento. Delimitación del Trabajo Este sistema se logrará implementar en el área administrativa, tanto para el personal de la empresa y sus clientes con las funciones de compra y venta de un producto.

2. METODOLOGÍA

- 1) En Data Source Name Ponemos nuestro Nombre en la Descripción ponemos Cualquier nombre en nuestro caso pusimos proyecto en Server pondremos localhost y contraseña ponemos la misma que configuramos cuando instalamos el administrador del Mysql en nuestro caso la clave sería mysql luego le damos clic La
- 2) DataBase y seleccionamos la conexión que creamos en nuestro caso sería proyecto

Empleado (<u>código</u> , Nombre, Cedula, Teléfono, Correo, Dirección, <u>cedula</u>)
Cliente (<u>Cedula</u> , Nombre, Apellido, Dirección, Teléfono)
Producto (<u>cod_producto</u> , nombre, producto, stock, precio, venta, descripción, <u>fecha ingreso</u>)
Detalle_Factura (<u>cod_factura</u> , código_articulo, cantidad, <u>código Factura</u>)
Proveedor (<u>id proveedor</u> , nombre_prov, correo, teléfono)
Cargo (<u>cod_cargo</u> , <u>tipo_cargo</u> , <u>cod_producto</u>)
Factura (<u>num factura</u> , fecha_factura, toa_factura, <u>cod_producto</u> , <u>cedula</u>)

Abstract Factory

Permite producir familias de objetos relacionados sin especificar sus clases concretas.

Builder

Permite construir objetos complejos paso a paso. Este patrón nos permite producir distintos tipos y representaciones de un objeto empleando el mismo código de construcción.

2.1 Command

Convierte una solicitud en un objeto independiente que contiene toda la información sobre la solicitud.

Esta transformación te permite parametrizar los métodos con diferentes solicitudes, retrasar o poner en cola la ejecución de una solicitud y soportar operaciones que no se pueden realizar.

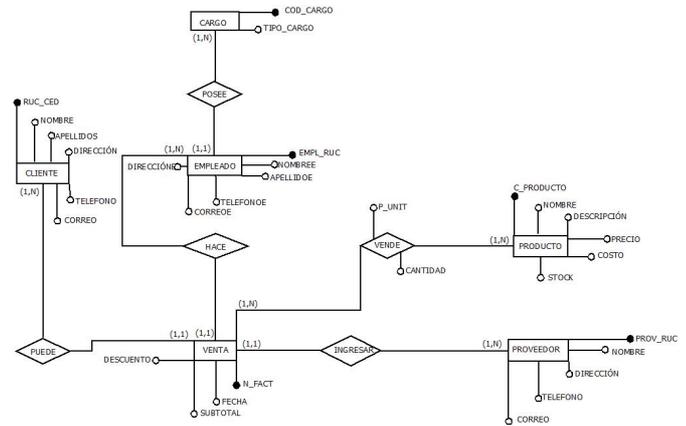


Ilustración 4.

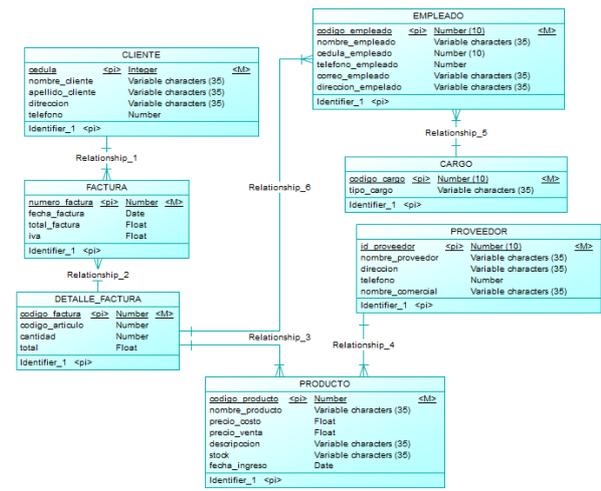


Ilustración 5.

Dentro del desarrollo de la aplicación usamos las herramientas de calendario incluido los datos

```

Pauquete → package paquete;
Clase → public class Clase {
    public static void main(String[] args) {
        //Esto es un comentario → Comentario
        System.out.println("Hello World!");
    }
}
Función Main →
Instrucción →
    
```

Ilustración 6.

Las solicitudes constan de los siguientes elementos:

- Patrones creacionales

- Patrones estructurales
- Builder
- Factory Method
- Decorator
- Flyweight
- Patrones de comportamiento

Los patrones de diseño son soluciones habituales a problemas que ocurren con frecuencia en el diseño de software. Son como planos prefabricados que se pueden personalizar para resolver un problema de diseño recurrente en tu código. No se puede elegir un patrón y copiarlo en el programa como si se tratara de funciones o bibliotecas ya preparadas. El patrón no es una porción específica de código, sino un concepto general para resolver un problema particular.

2.2 Pruebas

Dentro de las pruebas realizadas se escogió una web para su análisis <http://uisrael.edu.ec/> en esta web registramos los parámetros de exclusión de documentos e imágenes para mejorar el rendimiento de la búsqueda en tiempo. Así como el tiempo de espera en una devolución por parte del servidor de 15 segundos.

La velocidad de las consultas dependerá de la conexión de internet, para este análisis se tomó una conexión de internet de 10 Mbps en un canal de compartición 8 a 1.

Internamente existe una restricción de dominio en las URLs encontradas para no extender la búsqueda por un concepto de anidamiento.

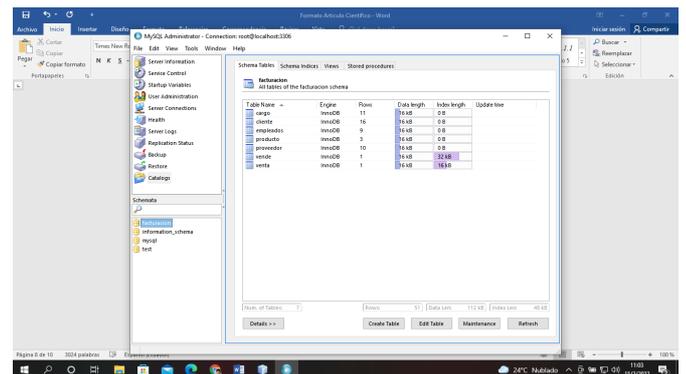
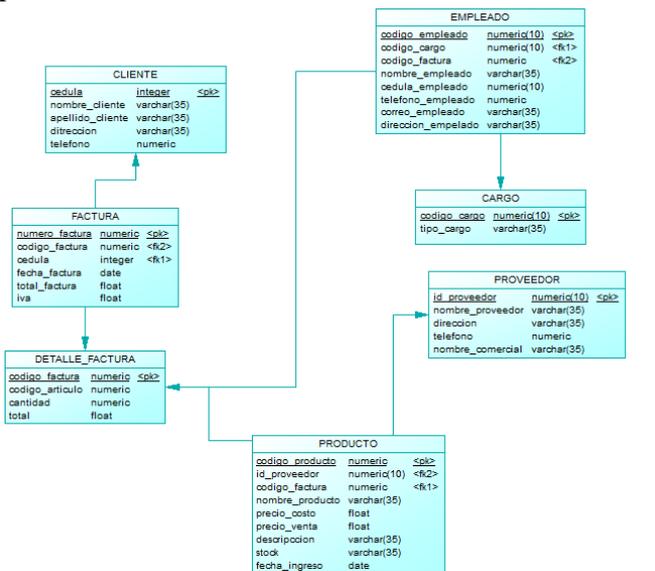


Ilustración 8.

Se inició el proceso y se puede observar las primeras bases de datos y sus tablas obtenidas con las exclusiones previamente indicadas.

Desarrollo

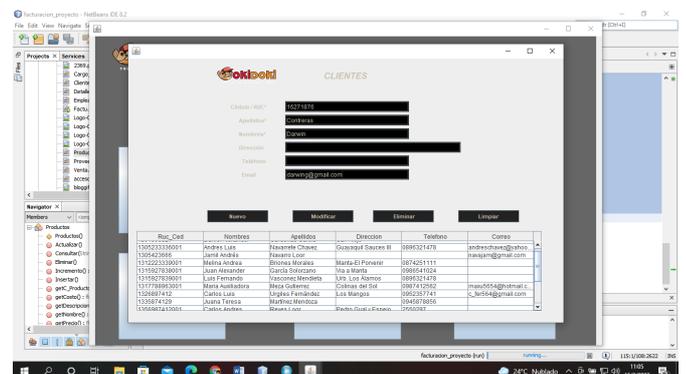
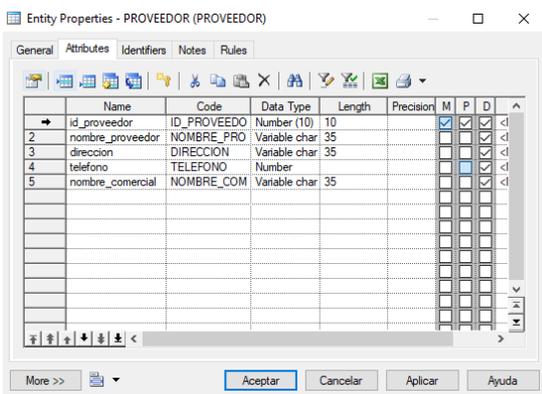


Ilustración 9.

Generamos un DataGridView el cual nos permite observar las URLs obtenidas y un identificador único para procesos posteriores de actualización de información por estado “Encontrado” o “No Encontrado”



Ilustración 10. Implementación.

Una vez finalizado el proceso inmediatamente se puede validar por su estado nos identifiquen que presentan un inconveniente y su posterior notificación para corrección.

3. RESULTADOS Y DISCUSIÓN

a. Identificación de contenido.

Neatbeans se comunica con la base de datos a través de la conexión (JDBC) dependiendo de la base de datos que se emplee en este caso se requiere el manejador (drive) para MYSQL. Un archivo de configuración de interfaz crea la comunicación entre esta herramienta y la bases de datos, es mediante este archivo que interfaz conoce en contacto con el ide netbeans.

Existen varios métodos o patrones de programación que ayudan con la organización y la arquitectura del código fuente de un sistema o aplicación; comúnmente se ven patrones como el mvc o el desarrollo por capas, sin embargo, se considera una mejor practica de programación, emplear el patrón mvc, ya que en este se asigna responsabilidades puntuales a cada componente.

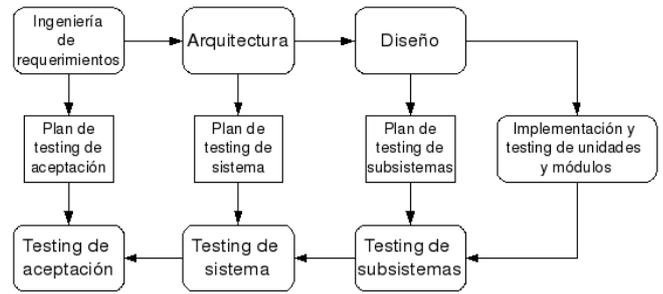


Ilustración 11

El acceso a esta información permite la manipulación del programa puede interactuar con la emulación de una interacción humana. Esto permite extraer la información necesaria para completar el proceso que son:

FASE DE PLANEACION

Para iniciar esta fase se procedió a obtener los requerimientos del usuario, para lo cual se llevó una serie de reuniones con el cliente, para abstraer lo que realmente el cliente.

En la fase de planeación se tuvo como resultado, el perfil del proyecto, en el mismo que se aprecia el modulo a desarrollar, así como también la descripción de cada módulo, con cada una de las iteraciones.

FASE DE DISEÑO

En la fase de diseño se obtuvo los prototipos de la interfaz gráfica con la que contara la aplicación, para lo cual se utilizó la herramienta Balsamiq Mockup, la misma que permite realizar un „dibujo“ de los requerimientos del usuario.

FASE DE CODIFICACION

En esta fase se procedió a la codificación en el lenguaje de programación Java de cada una de las funcionalidades del sistema.

FASE DE PRUEBAS

En esta fase se procedió a realizar las validaciones de cada una de los casos de pruebas de las funcionalidades del sistema, para lo cual se hizo necesario la construcción de un plantilla que contenga la funcionalidades del sistema.

b. Proceso y Búsqueda.

Los procesos se hacen mediante un acceso directo que lo redirección directamente a la aplicación ejecutable, toda la información será almacenada en un sistema de bases de datos SQL que le ayudará al cliente.

c. Resultados.

Las pruebas, se explicó de manera específica el funcionamiento del programa, tomando como referencia las operaciones de mayor relevancia como son : cliente ,factura, inventario, producto de igual forma se presentó el desenvolvimiento de la aplicación en cuanto a concurrencia y duplicidad de datos; se presentó la forma en la que se cargan datos de un distribuido anterior y finalmente se expusieron los reportes que la aplicación es capaz de presentar en diferentes formatos.

4. CONCLUSIONES

En este momento la Facultad de Ingeniería de la Universidad Técnica de Manabí, cuenta con un sistema informático que permite la facturación de la empresa oki doki; permitiendo de esta manera la cobranza y administración.

- Mediante el programa, puede hacer el ingreso y egreso de todos los programas incluido el producto, cliente, realizar la factura, detallar el producto todo se almacenará en una base de datos sql
- Actualmente las empresas están implementando nuevas tecnologías para así llevar un mejor manejo en el ámbito contable y administrativo de todo lo que hay dentro de la empresa
- A través de esta aplicación, es posible visualizar, almacenar e imprimir el distributivo, tanto general de la empresa o de manera individual siempre y cuando el cliente lo solicite.
- En cuanto a los clientes, es factible visualizar, almacenar e imprimir la información del Distributivo individual, filtrando los datos por Cédula, Nombres, Apellidos, Dedicación, Tipo

- Durante el transcurso de desarrollo del presente proyecto, hemos tenido la oportunidad de conocer a profundidad las herramientas orientadas a Desarrollo de aplicaciones en java.

REFERENCIAS

1. bulk. (s.f.). *bulk*. Obtenido de <https://www.bulk.com/es/the-core-es/los-5-mejores-suplementos-para-ganar-musculo/>
2. excelahealth. (s.f.). *excelahealth*. Obtenido de <https://www.excelahealth.org/health-library/article?chunkid=21811&lang=Spanish&db=hls#:~:text=Dosis%20Terap%C3%A9uticas,-La%20melatonina%20se&text=Para%20el%20insomnio%20com%C3%BAn%2C%20la,la%20hora%20deseada%20de%20dormir.>
3. Fernández, Á. (s.f.). *feda*. Obtenido de <https://www.feda.net/dips-fondos-buen-constructor-pectoral/>
4. Fernando. (22 de 10 de 2015). <http://fernand0.github.io/>. Obtenido de <http://fernand0.github.io/Recuperacion-Claves-Preguntas-Personales/>
5. Gutiérrez, P. (5 de 11 de 2013). *genbeta*. Obtenido de <https://www.genbeta.com/desarrollo/fundamento-de-las-bases-de-datos-modelo-entidad-relacion>
6. Izquierdo, J. (4 de 9 de 2014). *iebschool*. Obtenido de <https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum/#:~:text=Agile%20es%20el%20marco%20de,c%C3%B3digo%20desarrollado%20y%20la%20realimentaci%C3%B3n.>
7. Mateos, A. S. (3 de 10 de 2016). *lavanguardia*. Obtenido de <https://www.lavanguardia.com/vivo/ejercicio/20160917/41363183278/que-hacer-antes-despues-ejercicio.html>

8. meg-3. (2010). *meg-3*. Obtenido de https://www.meg-3.com/es_es/omega-3s.html
9. normasapa. (s.f.). *normasapa*. Obtenido de <https://normasapa.com/como-referenciar- imagenes-figuras-segun-las-normas-apa/>
10. Saavedra, E. M. (10 de 5 de 2019). *menshealth*. Obtenido de <https://www.menshealth.com/es/fitness/a27264157/mejores-ejercicios-pecho-pectoral/>