



INSTITUTO SUPERIOR TECNOLÓGICO
SUDAMERICANO
QUITO - ECUADOR

ESCUELA DE
DESARROLLO DE SOFTWARE

PROGRAMACIÓN VISUAL 2

**DESARROLLAR E IMPLEMENTAR UN CRUD CON POSTGRESQL
BASADO EN PROGRAMACIÓN JAVA PARA LA EMPRESA
TECNINOVA**

ALUMNO(S): Requena Tonony Emmanuel Alejandro
 Vinueza Ayala Erick Thomas
 Sandoval Pilca Javier Mauricio

DOCENTE TUTOR: Ing. Moncayo Mosquera Guido Armando

MARZO 2023
QUITO – ECUADOR

TABLA DE CONTENIDO

1.	Introducción	1
2.	Justificación.....	2
3.	Antecedentes	3
4.	Objetivos	4
4.1.	Objetivo General	4
4.2.	Objetivos Específicos	4
5.	Metodología	5
6.	Marco teórico	6
6.1.	Programación	6
6.1.1	Algoritmos	6
6.1.2	Compilación.....	7
6.1.3	Lenguaje de programación	7
6.1.4	Algunos lenguajes de programación.....	8
6.1.5	Librerías en programación.....	10
6.2	Bases de datos	10
6.2.1	Que son bases de datos	10
6.2.2	Gestores de bases de datos.....	11
6.2.3	PostgreSQL.....	11
6.2.4	MySQL	12
6.2.5	JDBC	12
6.3	JAVA.....	13
6.3.1	JDK.....	13
6.3.2	IDE's y Editores de texto.....	14
6.3.3	Eclipse	14
6.3.4	WindowsBuilder.....	14
7.	Desarrollo del Trabajo.....	16
7.1	Diseño de la base de datos prototipo en PostgreSQL.....	16
7.2	Diseño de la interfaz en JAVA, con Eclipse y WindowsBuilder.....	16

7.2.1 Instalación de WindowsBuilder.	16
7.2.2 Creación de las ventanas.	18
8. Conclusiones y Recomendaciones	25
8.1. Conclusiones	25
8.2. Recomendaciones.....	25
9. Referencias	27

ÍNDICE DE FIGURAS

Imagen I: Instalación WindowsBuilder.....	16
Imagen II: Instalación WindowsBuilder	16
Imagen III: Instalación WindowsBuilder	17
Imagen IV: Instalación WindowsBuilder.....	17
Imagen V: Instalación WindowsBuilder	17
Imagen VI: Función Design	18
Imagen VII: VentanaCaratula	18
Imagen VIII: Ventana1.....	19
Imagen IX: Ventana2	19
Imagen X: VentanaClientes	20
Imagen XI: VentanaDirecciones	20
Imagen XII: VentanaProductos	21
Imagen XIII: VentanaServicios.....	21
Imagen XIV: VisorClientes.....	22
Imagen XV: VisorDirecciones	22
Imagen XVI: VisorProductos.....	23
Imagen XVII: VisorServicios.....	23
Imagen XVIII: VentanaResumen.....	24

1. Introducción

En la ciudad de Quito existe un pequeño negocio de reparación y mantenimiento de equipos de línea blanca, que está iniciando su posicionamiento en el mercado con el nombre de TECNINOVA.

En este negocio se maneja una cantidad de clientes nuevos y también frecuentes que buscan atención con sus electrodomésticos. Para TECNINOVA es importante mantener un registro ordenado de estos clientes y sus datos correspondientes.

Los sistemas de bases de datos son una herramienta fundamental para los negocios que, como casi todos, planifican su crecimiento y sientan sus bases muy claras. Actualmente hay empresas de software dedicadas a crear bases de datos para otras empresas y también para este tipo de negocios que se encuentran en sus etapas iniciales. El mantener una base de datos de todo lo que se realiza, permite que el dueño o gerente tenga una visión muy clara de cómo avanza el negocio para, según eso, poder tomar decisiones a corto, mediano y largo plazo. Las empresas más grandes no lo habrían llegado a ser sin una base de datos en la que registrasen todas sus actividades, clientes, etc. Para obtener estadísticas y proceder con la toma de decisiones.

Existen varias técnicas y herramientas para crear una base de datos. En este proyecto tomaremos una de ellas para tener una interfaz gráfica amigable con el usuario y fácil de administrar.

2. Justificación

Aplicar las correctas técnicas de programación visual, desarrollando el algoritmo para las ventanas en Java, aprovechando la facilidad que permite Windows Builder con el IDE Eclipse, creando una interfaz entendible y de fácil manejo, para satisfacer las necesidades del usuario.

Utilizar como base lo aprendido en la materia de Programación Visual2, para construir las ventanas y realizar la conexión con las bases de datos creadas en MySQL y PostgreSQL.

Todo formulado para llevar una organización correcta de los datos de TECNINOVA para que pueda consolidarse cada vez más en el mercado y que pueda brindar un mejor servicio a sus clientes en base a las decisiones que vaya tomando.

La importancia principal de este proyecto es que TECNINOVA está sentando sus bases en este momento y tener una correcta organización de los datos como; clientes, sus direcciones de domicilio, productos del cliente y servicios realizados, es importante para que logre tener los resultados esperados según su planificación. Este sistema le permitirá tener esa organización y a un costo mínimo.

3. Antecedentes

TECNINOVA, como nombre comercial, solo tenía a su propietario como único trabajador, pero en este momento busca consolidarse en el mercado, tener más trabajadores a disposición y se plantea una visión a futuro como miniempresa y posteriormente como empresa, para esto debe mantener una correcta organización de los datos necesarios en su área de negocio, lo cual hasta el momento no se ha tenido.

Actualmente existen varias empresas líderes en el campo de las bases de datos como ORACLE o SAP, que han ayudado a gestionar los datos de varias empresas reconocidas a nivel mundial. También existen otras empresas más pequeñas que ofertan sistemas de bases de datos para empresas medianas o microempresas y para negocios pequeños, es importante decidir cuál es la mejor opción o buscar otras alternativas.

Aunque estas empresas están bien establecidas, tienen una utilidad exagerada para TECNINOVA que se encuentra iniciando su negocio, sobre todo por los costos elevados que tendría el contratar un servicio de gestión de bases de datos a una empresa como las mencionadas anteriormente. Por ello el presente proyecto es el ideal debido a costos reducidos y centrado en la única problemática presentada de crear un sistema completo de base de datos para este negocio.

4. Objetivos

4.1. Objetivo General

Desarrollar e implementar un CRUD con PostgreSQL basado en programación java para la empresa TECNINOVA.

4.2. Objetivos Específicos

1. Desarrollar el Marco teórico.
2. Diseñar la base de datos prototipo en PostgreSQL.
3. Diseñar la GUI en JAVA, utilizando el IDE Eclipse, con ayuda de la herramienta WindowsBuilder.
4. Implementar el JDBC, que permitirá la conexión de JAVA con las bases de datos.
5. Probar el funcionamiento del JDBC
6. Probar el correcto funcionamiento del CRUD, desde las ventanas creadas en JAVA.
7. Corregir los errores producidos en el proyecto.
8. Realizar las pruebas de funcionamiento finales.

5. Metodología

La metodología permite operacionalizar las variables y recolectar los datos, para comprobar las hipótesis, obtener y analizar la información, así como permitirá determinar el logro de los objetivos. Para desarrollar los trabajos académicos y los proyectos integradores, se ha considerado que los métodos más sustanciales son:

- Método Descriptivo.
- Método Deductivo.

6. Marco teórico

6.1. Programación

La programación se refiere a ordenar un conjunto de acciones a realizar en secuencia para cumplir con un objetivo. En informática, la programación es escribir paso a paso las instrucciones que debe seguir un ordenador para cumplir una función. Este escrito se lo realiza mediante el teclado en un lenguaje específico que permitirá la comunicación entre el programador y la máquina.

En otras palabras, es una especie de arte de traducir los deseos de una persona al lenguaje de la máquina. [1]

La programación está presente en todo lo que tenga que ver con tecnología, pues es gracias a esta que las personas se pueden comunicar con el mundo tecnológico. Permite la creación de aplicaciones, programas, softwares, interfaces, etc., que se desarrollan desde un código fuente, el cual lo escribe el programador en un lenguaje entendible para él y que, al ejecutarlo, el ordenador se encarga de convertirlo a lenguaje de máquina, es decir un lenguaje binario utilizando 1 y 0, de esa manera el ordenador entiende las instrucciones escritas por el programador y realiza las funciones especificadas.

6.1.1 Algoritmos

Un algoritmo es un conjunto de instrucciones que se deben seguir secuencialmente para cumplir un objetivo o función específica. Por ejemplo, para preparar una taza de té su algoritmo sería el siguiente:

1. Poner agua en una olla
2. Colocar la olla sobre la cocina
3. Encender la cocina.
4. Esperar a que hierva el agua
5. Cuando ya hierva el agua apagar la cocina
6. Retirar la olla de la cocina
7. Poner el agua en una taza
8. Agregar las cucharadas de café deseadas
9. Agregar azúcar si se desea
10. Menear el contenido con una cuchara hasta disolver por completo

En informática un algoritmo es una secuencia de instrucciones, gracias a la cual pueden llevarse a cabo ciertos procesos y darse una respuesta a determinadas necesidades o decisiones. Se trata de conjuntos ordenados y finitos de pasos, que nos permiten resolver un problema o tomar una decisión. [2]

El algoritmo de un programa es su código fuente, son todas las líneas de código desarrolladas por el programador, con las instrucciones bien estructuradas y definidas.

6.1.2 Compilación

La programación inicia con un programador desarrollando un algoritmo en un lenguaje de programación, el cual es entendible para él. Pero si la máquina debe seguir las instrucciones de este algoritmo directamente desde el código escrito por el programador no lo va a poder hacer, ya que el ordenador solo entiende su lenguaje de máquina, un lenguaje binario que comprende miles y millones de “1” y “0”, el cual sería complicado y tardío de comprender para el programador.

Para que exista la comunicación, entre el programador y el ordenador, existe un compilador, que es otro programa el cual se encarga de realizar la compilación. La compilación es el proceso de convertir las líneas de texto escritas por el programador a lenguaje de máquina de 1s y 0s.

Básicamente, la tarea de compilar se refiere al proceso de traducción del código fuente, entendiéndose por código fuente las líneas de código que se han escrito en un lenguaje de programación. [3]

6.1.3 Lenguaje de programación

El lenguaje es esencial entre las personas pues es lo que permite la comunicación y la interacción entre humanos. En el mundo existen varias lenguas en las que se puede comunicar entre personas, cada lengua depende de su ubicación geográfica o de la cultura a la que se pertenece.

Así mismo existen varios lenguajes con los que un programador puede comunicarse con un ordenador para darle las instrucciones a seguir, estos son los lenguajes de programación.

Un lenguaje de programación nos permite comunicarnos con las computadoras a través de algoritmos e instrucciones escritas en una sintaxis que la computadora entiende e interpreta en lenguaje de máquina. [4]

Los lenguajes de programación permiten que el programador pueda dar grandes y complejas instrucciones a un ordenador y este las realice de forma rápida y eficiente. La información que

una persona procesaría en varias horas o días, un ordenador la puede procesar en segundos, con las instrucciones adecuadas escritas en el código fuente que después será compilado para que el ordenador pueda entender y ejecutar dichas instrucciones.

6.1.4 Algunos lenguajes de programación

En la actualidad existen decenas de distintos lenguajes en los que se pueden programar, la diferencia está en el uso que se le vaya a dar, ya sea para sistemas financieros, creación de videojuegos, creación de páginas web, etc., cada uno tiene sus ventajas y desventajas.

Según Lucía González [5], programadora especialista en páginas web y administradora del blog Dinahosting, los 10 lenguajes de programación más utilizados en la actualidad son los siguientes:

1. **PHP**

Es un lenguaje utilizado en el desarrollo de páginas y aplicaciones web ya que se ejecuta en un servidor y se compagina con HTML. Es muy confiable ya que el código fuente está oculto para el usuario. Es necesario un servidor para ejecutarlo y se debe tener conocimientos básicos sobre HTML, que es un lenguaje de marcado para estructurar las páginas web.

2. **PERL**

Un lenguaje muy sencillo y de alto rendimiento, muy popular para la creación de aplicaciones web, también se puede aplicar para la extracción de archivos de texto y generación de informes. De mucha utilidad Para la administración de sistemas, muchos sitios web con alto tráfico como por ejemplo Amazon, utilizan este lenguaje

3. **JAVA**

Es uno de los lenguajes de programación más populares, es un software libre, ejecutable en la mayoría de sistemas operativos. Desde su creación en 1995, ha impulsado gran parte del mundo digital y aún ahora, varios productos y servicios digitales siguen basándose en Java. Aunque la sintaxis de Java es compleja de aprender, es tan poderoso este lenguaje que es el que impulsa Android.

4. **C++**

Es un lenguaje muy versátil y potente, muy utilizado en la creación de sistemas operativos y bibliotecas, permite la compilación de su código en múltiples plataformas. Es un lenguaje que permite la programación orientada a objetos y a través del cual se puede aprender otros como son Java, C# o PHP. No se utiliza

mucho para páginas web o base de datos ya que es muy pesado comparado con otros para el mismo propósito.

5. C#

Es un lenguaje creado por Microsoft, permite la programación orientada a objetos y se usa para la creación de videojuegos, servicios web, aplicaciones cliente-servidor, etc. Su sintaxis es muy similar a C++ o Java por lo que su curva de aprendizaje es muy elevada, pero es muy potente y flexible.

6. PYTHON

Es un lenguaje de programación muy versátil y flexible, muy popular en la actualidad por su aplicación en muchos aspectos digitales. Se emplea para crear aplicaciones, juegos, en la informática científica y en big data que actualmente está muy de moda. Funciona en múltiples plataformas, soporta la programación de GUI y se conecta fácilmente con aplicaciones de bases de datos.

7. JAVASCRIPT

Es un lenguaje de programación muy robusto y popular en la implementación de páginas web, mejora la experiencia del usuario generando interacción dinámica entre diferentes sitios web. Funciona en múltiples plataformas y permite la programación orientada a objetos, compatible con la mayoría de navegadores. Puede ser vulnerable ya que su código está visible para cualquier usuario.

8. SWIFT

Es un lenguaje de programación reciente y de código abierto, lanzado por Apple para diseñar aplicaciones en sistemas IOS y macOS, también se lo puede utilizar en Linux. Es muy seguro, ligero y fácil de aprender, aunque tiene muchas reglas en su sintaxis y cada que haya una actualización de su versión se tendrá que reprogramar toda la aplicación.

9. R

Es un lenguaje súper potente para realizar cálculos científicos, y estadísticos, así como para crear gráficos de gran calidad. Actualmente se ha vuelto muy famoso por su uso en el big data, que es el análisis de grandes cantidades de datos. El complejo de usar, es de código abierto y multiplataforma.

10. RUST

Un lenguaje que está aumentando su popularidad, fundado por Mozilla, permite desarrollar software de sistemas, creación de GUI y muy utilizado en el campo de

las criptomonedas, es muy rápido y seguro, tiene sus propias reglas lo que lo hace algo complejo de aprender.

6.1.5 Librerías en programación

Una librería o biblioteca es un conjunto de archivos que se utiliza para desarrollar software. Suele estar compuesta de códigos y datos, y su fin es ser utilizada por otros programadores de forma totalmente autónoma. Simple y llanamente es un archivo importable [6].

Proporcionan funcionalidades comunes, que ya han sido resueltas previamente por otros programadores y evitan la duplicidad del código. Además, reducen el tiempo de desarrollo y aumentan la calidad del mismo [7].

6.2 Bases de datos

6.2.1 Que son bases de datos

Una base de datos es un conjunto de datos interrelacionados que se organizan de manera estructurada en un sistema informático, con el fin de permitir su almacenamiento y recuperación eficiente. Según Elmasri y Navathe [8], una base de datos se define como "un conjunto de datos relacionados entre sí, representados de manera lógica y almacenados en un soporte físico controlado por un sistema de gestión de bases de datos (SGBD)". Las bases de datos son una herramienta esencial en muchos campos, como la gestión empresarial, la investigación científica y la administración de sistemas de información, entre otros.

Las bases de datos se dividen comúnmente en dos categorías principales: las bases de datos relacionales y las no relacionales. Las bases de datos relacionales utilizan tablas para almacenar los datos y relaciones, mientras que las no relacionales emplean otros modelos, como documentos, grafos y claves-valor. Cada tipo de base de datos tiene sus ventajas y desventajas, y la elección de un modelo dependerá de las necesidades específicas de cada aplicación. En cualquier caso, las bases de datos permiten una gestión eficiente de grandes cantidades de datos y son esenciales para la toma de decisiones en entornos empresariales y científicos [8].

6.2.2 Gestores de bases de datos

Un gestor de bases de datos (SGBD) es un software diseñado para gestionar y manipular bases de datos, proporcionando una interfaz entre el usuario y los datos almacenados en el sistema. Según Liu y Özsu [9], un SGBD se define como "un software que controla el acceso, la organización, el almacenamiento y la recuperación de datos en una base de datos". Los SGBD son una parte integral de cualquier sistema de gestión de bases de datos y proporcionan herramientas para la gestión de los datos, la realización de consultas y la generación de informes.

Los SGBD se utilizan en muchos entornos empresariales y científicos, y existen varios tipos de SGBD disponibles en el mercado, cada uno diseñado para satisfacer necesidades específicas. Algunos ejemplos de SGBD incluyen MySQL, Oracle, SQL Server y MongoDB. Los SGBD son esenciales para la gestión eficiente de grandes cantidades de datos y permiten a los usuarios acceder y manipular la información almacenada de manera eficiente y segura [9].

6.2.3 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto y gratuito que se utiliza ampliamente en aplicaciones empresariales y de investigación científica. Según Stonebraker y Rowe [10], PostgreSQL es "un sistema de gestión de bases de datos relacional de objeto que proporciona un alto nivel de concurrencia para cargas de trabajo de escritura y lectura simultáneas". PostgreSQL se basa en el lenguaje SQL y ofrece una amplia gama de características avanzadas, como soporte para transacciones ACID, replicación, particionamiento y manejo de objetos complejos.

PostgreSQL es una de las bases de datos de código abierto más populares y ha ganado una gran aceptación en la comunidad empresarial debido a su capacidad para manejar grandes cantidades de datos y soportar una alta concurrencia de usuarios. Además, PostgreSQL es altamente personalizable y se puede extender a través de módulos adicionales para satisfacer necesidades específicas de aplicación. PostgreSQL es ampliamente utilizado en aplicaciones empresariales, científicas y gubernamentales en todo el mundo [10].

6.2.4 MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto que se utiliza ampliamente en aplicaciones web y empresariales. Según Widenius y Axmark [11], MySQL se define como "un sistema de gestión de bases de datos relacional que utiliza el lenguaje SQL para gestionar el acceso a los datos". MySQL es altamente escalable y puede manejar grandes cantidades de datos y un alto volumen de tráfico simultáneo.

MySQL es ampliamente utilizado en aplicaciones web, como WordPress y Drupal, así como en aplicaciones empresariales, científicas y gubernamentales. MySQL es de código abierto y gratuito, lo que lo hace accesible para una amplia gama de usuarios y desarrolladores. Además, MySQL es altamente personalizable y se puede extender mediante el uso de módulos adicionales, como los motores de almacenamiento InnoDB y MyISAM [11].

6.2.5 JDBC

JDBC (Java Data Base Connectivity) es una API (Interfaz de Programación de Aplicaciones) de Java que permite a los desarrolladores interactuar con bases de datos a través del lenguaje Java. Según Balci [12], "JDBC es una API de bajo nivel que proporciona métodos para conectarse a una base de datos, enviar consultas y recibir resultados". JDBC se utiliza en aplicaciones Java para permitir la comunicación entre una aplicación y una base de datos.

JDBC es ampliamente utilizado en el desarrollo de aplicaciones Java empresariales y de software de escritorio. La API proporciona una forma eficiente de interactuar con bases de datos, lo que permite a los desarrolladores realizar consultas y manipular datos de manera efectiva. Además, JDBC es compatible con una amplia gama de bases de datos, lo que lo hace muy versátil y útil para los desarrolladores de Java [12].

6.3 JAVA

Java es un lenguaje de programación orientado a objetos ampliamente utilizado en el desarrollo de aplicaciones empresariales y de software de escritorio. Según Gosling, Joy y Steele [13], "Java es un lenguaje de programación de alto nivel que se caracteriza por ser portátil, seguro y robusto". Java fue desarrollado por Sun Microsystems en la década de 1990 y se ha convertido en uno de los lenguajes de programación más populares del mundo.

Java se utiliza en una amplia variedad de aplicaciones, desde aplicaciones móviles y web hasta aplicaciones empresariales de alta complejidad. Java es conocido por su capacidad de ejecución en múltiples plataformas y su alto nivel de seguridad, lo que lo hace popular en el desarrollo de aplicaciones web y móviles. Además, Java es altamente modular y escalable, lo que permite a los desarrolladores de software crear aplicaciones complejas y de gran escala [13].

6.3.1 JDK

JDK (Java Development Kit) es un conjunto de herramientas de software que se utilizan para desarrollar aplicaciones en el lenguaje de programación Java. Según Oracle [14], "JDK incluye el compilador de Java, herramientas de depuración, bibliotecas de clases y otras herramientas necesarias para desarrollar y ejecutar aplicaciones Java". JDK se utiliza ampliamente en el desarrollo de aplicaciones empresariales, de software de escritorio y móviles.

Una de las características más importantes de JDK es su capacidad para compilar código Java en código ejecutable, lo que permite a los desarrolladores crear aplicaciones que pueden ser ejecutadas en cualquier sistema operativo que tenga una máquina virtual de Java instalada. Además, JDK incluye herramientas de depuración que permiten a los desarrolladores identificar y corregir errores en sus aplicaciones de manera eficiente [14].

6.3.2 IDE's y Editores de texto

Las IDEs (Integrated Development Environments) y los editores de texto son herramientas de software utilizadas en el desarrollo de aplicaciones informáticas. Según Koskela [15], una IDE es "un entorno completo para el desarrollo de software que incluye un editor de código, herramientas de depuración, compiladores, asistentes de programación, administración de proyectos y herramientas de pruebas". Las IDEs son una solución completa para el desarrollo de software, ya que ofrecen una amplia gama de herramientas y recursos para el desarrollo de aplicaciones.

Por otro lado, los editores de texto son herramientas más simples que se utilizan para editar y escribir código fuente. Estos editores no tienen las mismas características que las IDEs, pero son muy populares entre los desarrolladores debido a su simplicidad y flexibilidad. Según Charland y Orsini [16], los editores de texto "son programas diseñados para editar texto sin formato, incluyendo el código fuente de los programas". Los editores de texto son una opción popular para los desarrolladores que prefieren una herramienta simple para escribir código sin la complejidad de una IDE.

6.3.3 Eclipse

Eclipse es una IDE de código abierto muy popular para el desarrollo de aplicaciones en diferentes lenguajes de programación, como Java, C++, Python, entre otros. Según Hunt y Thomas [17], "Eclipse es una plataforma extensible que proporciona una amplia variedad de herramientas y plugins para el desarrollo de software". Eclipse es conocido por su alto nivel de personalización y flexibilidad, lo que lo hace ideal para el desarrollo de aplicaciones complejas.

Eclipse cuenta con una amplia gama de características que lo hacen muy popular entre los desarrolladores. Entre ellas se encuentran el autocompletado de código, la navegación de código, la depuración, la integración con herramientas de control de versiones y la capacidad de desarrollar aplicaciones web y móviles. Además, Eclipse tiene una gran comunidad de usuarios y desarrolladores que contribuyen con plugins y herramientas para hacer que la IDE sea aún más poderosa [17].

6.3.4 WindowsBuilder

WindowsBuilder es una herramienta de código abierto que se utiliza para el diseño visual de interfaces gráficas de usuario (GUI) en Java. Esta herramienta permite a los desarrolladores crear y editar interfaces de usuario de forma gráfica, sin tener que escribir código manualmente.

Según la documentación oficial de Eclipse [18], "WindowsBuilder proporciona una amplia gama de características para el diseño de interfaces de usuario, incluyendo herramientas de diseño de formularios, paletas de componentes y un editor de propiedades". Además, WindowsBuilder es compatible con diferentes frameworks y bibliotecas de GUI en Java, como Swing, SWT y GWT.

WindowsBuilder es una herramienta muy popular entre los desarrolladores de Java debido a su facilidad de uso y eficiencia en la creación de interfaces de usuario. Además, es una herramienta integrada con la plataforma Eclipse, lo que hace que su integración con otras herramientas de desarrollo sea muy sencilla. Según Martin [19], "WindowsBuilder es una herramienta muy valiosa para aquellos desarrolladores que desean crear interfaces de usuario de alta calidad de forma rápida y eficiente".

7. Desarrollo del Trabajo

7.1 Diseño de la base de datos prototipo en PostgreSQL.

7.2 Diseño de la interfaz en JAVA, con Eclipse y WindowsBuilder.

Las ventanas para la interfaz se las realizaron todas en Java con el IDE Eclipse, aquí se puede utilizar una herramienta llamada WindowsBuilder, que facilita la construcción de las ventanas.

7.2.1 Instalación de WindowsBuilder.

1. Eclipse 2022-12; Help -> Install new software

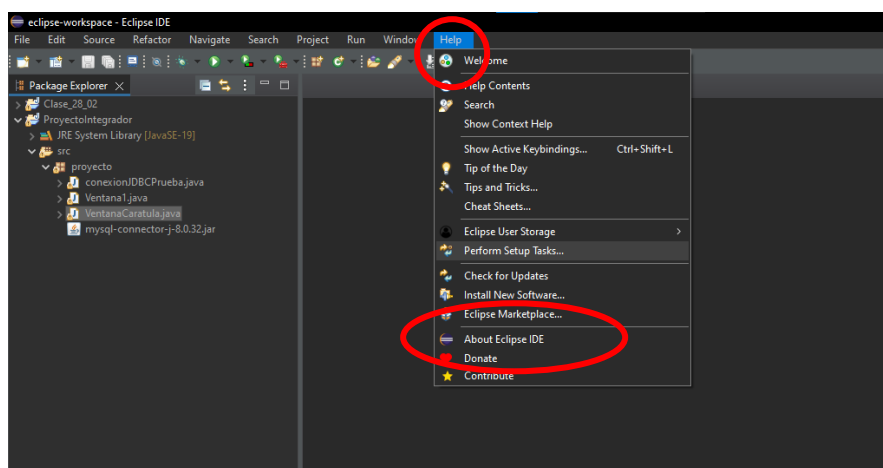


Imagen I: Instalación WindowsBuilder

2. All available sites -> windows builder -> Next

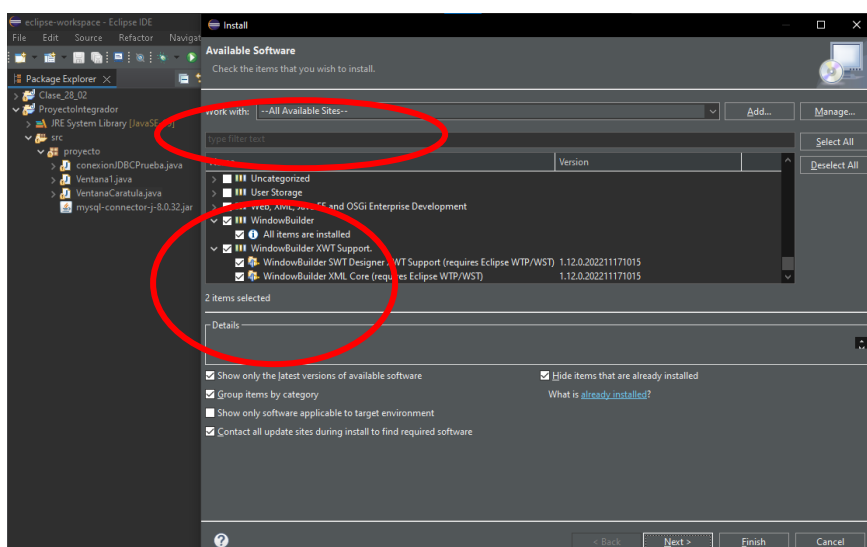


Imagen II: Instalación WindowsBuilder



3. Finish

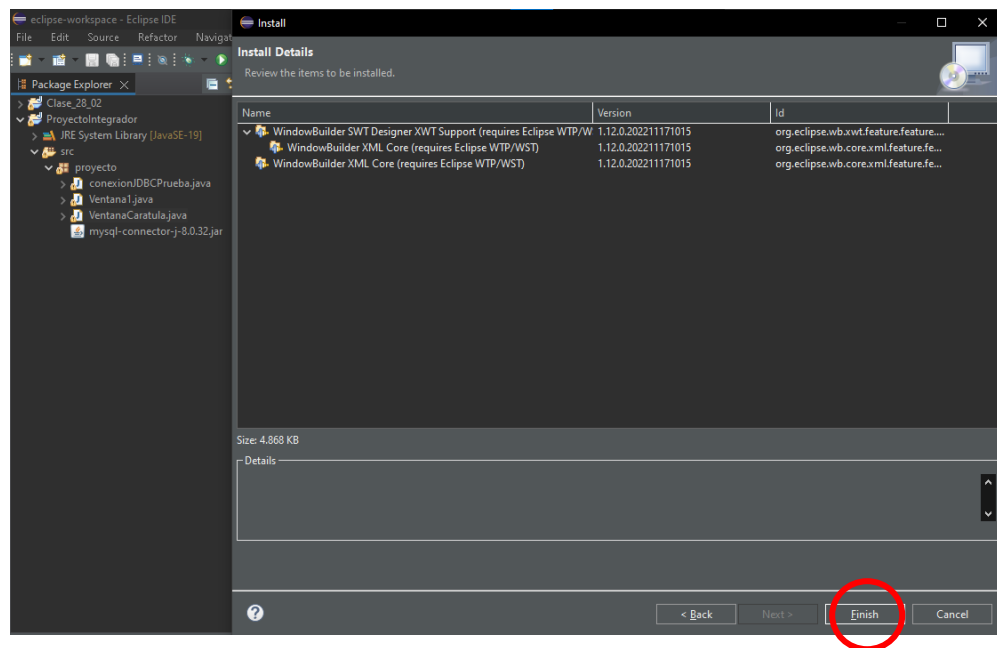


Imagen III: Instalación WindowsBuilder

4. Esperar la descarga

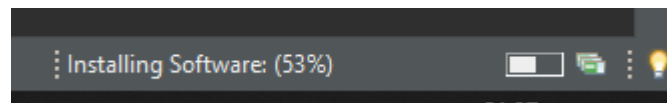


Imagen IV: Instalación WindowsBuilder

5. Reiniciar eclipse

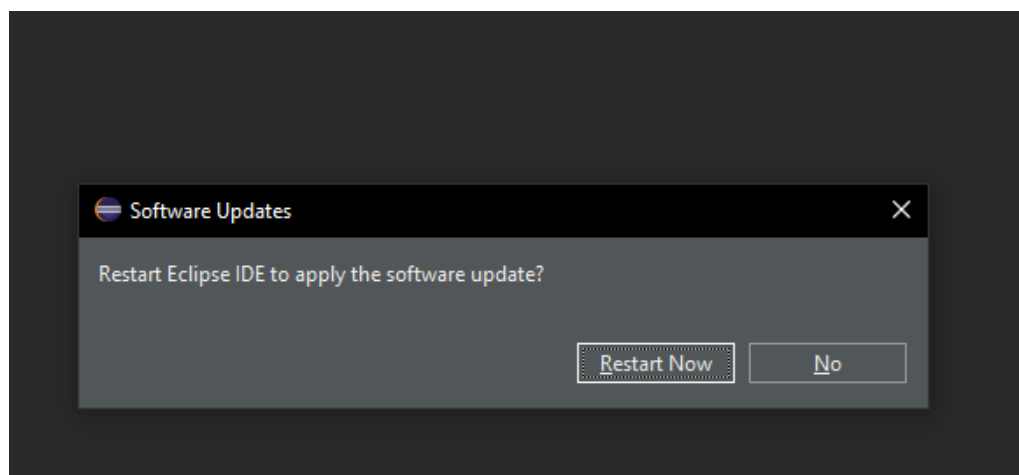


Imagen V: Instalación WindowsBuilder

7.2.2 Creación de las ventanas.

1. La aplicación ahora tiene la función Design, que permite ver la creación de la ventana de forma gráfica.

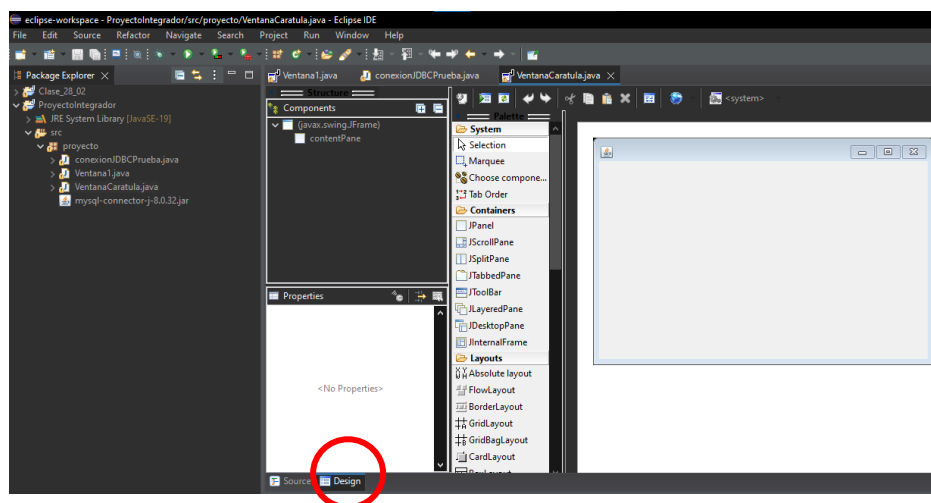


Imagen VI: Función Design

2. Se inició creando la VentanaCaratula. Cada línea de texto es una etiqueta (JLabel) con su respectiva elección de fuente, y tamaño de letra. El logo de la parte superior izquierda también es una etiqueta, con un ícono agregado. Por último, se colocó un botón en la parte inferior que permitirá pasar a la siguiente ventana.

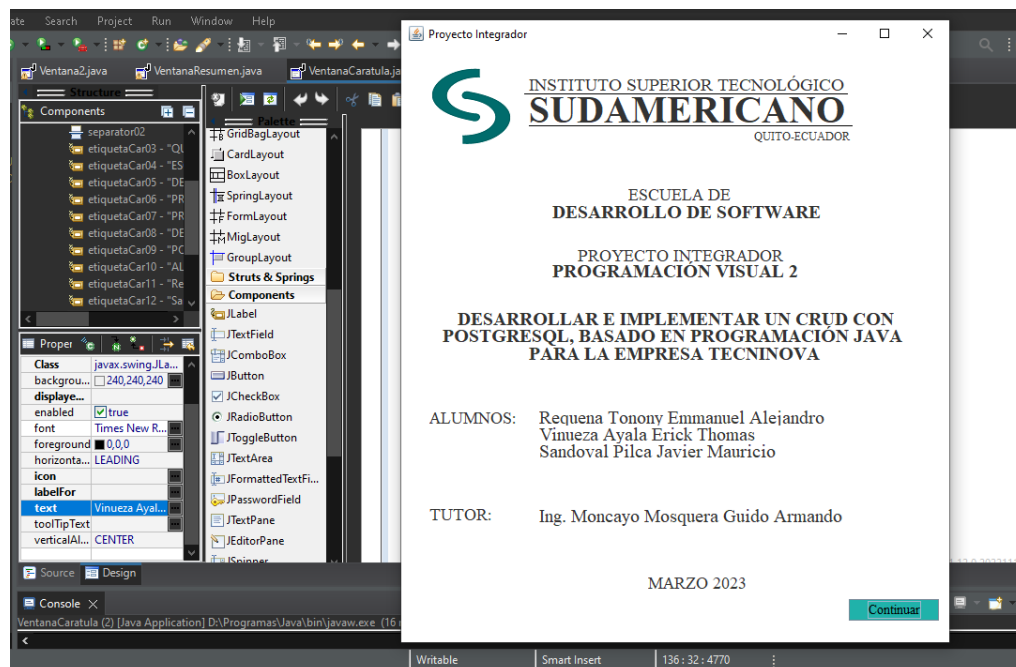


Imagen VII: VentanaCaratula

3. La siguiente es la Ventana1, que permite colocar una seguridad para el acceso, con un usuario y una contraseña. El cajón donde se colocará el usuario (JTextField) permitirá validar el nombre de usuario, y el cajón para la contraseña (JPasswordField) permitirá validar la contraseña, la cual se verá solo en símbolos para seguridad al escribir la contraseña. Los botones al final permiten “Ingresar” en caso de que la validación de usuario y contraseña sea correcta y “Salir” en caso de que el usuario desee salir del programa.

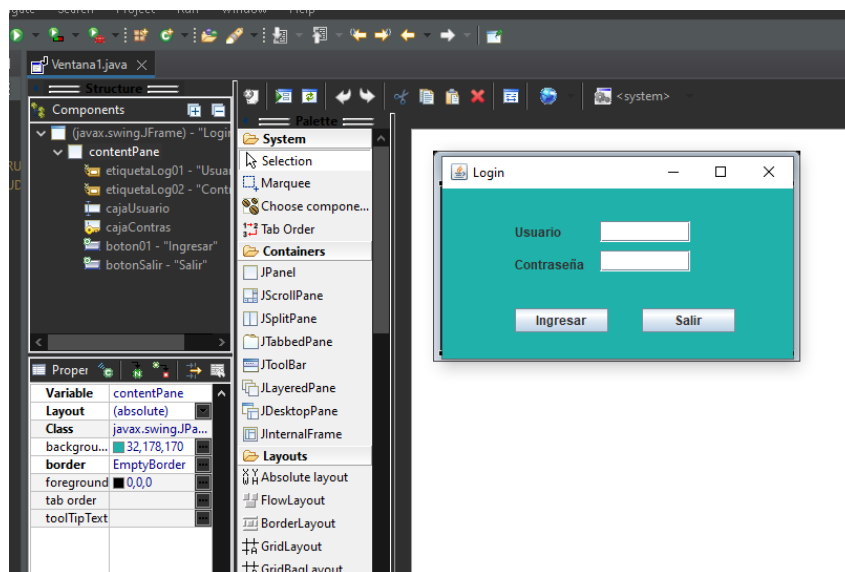


Imagen VIII: Ventana1

4. A continuación, se creó la Ventana2, la cual contiene los botones (JButton) que llevarán a las siguientes ventanas, ya sea para ingresar un nuevo registro, o abrir las tablas disponibles. El título es el logo de la empresa TECNINOVA, el cual está dentro de una etiqueta (JLabel).

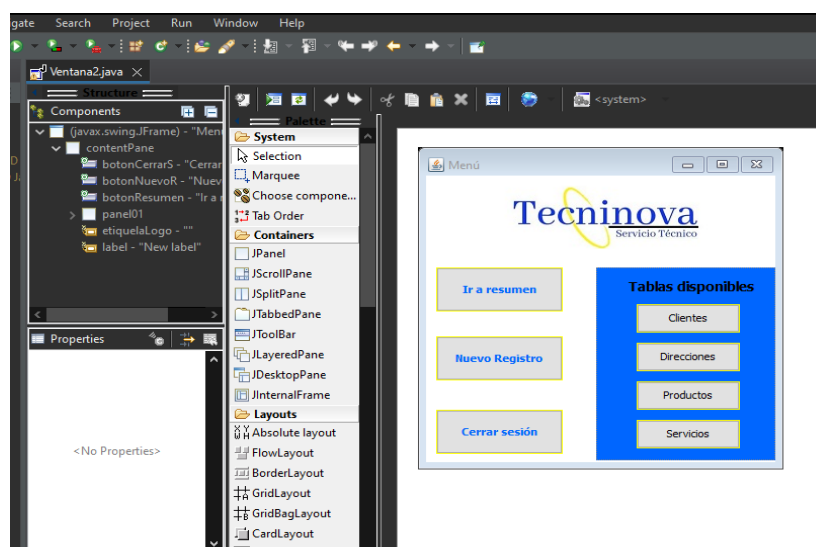


Imagen IX: Ventana2

5. La siguiente creación es la VentanaClientes, la cual contiene etiquetas (JLabel) para los textos, cajones (JTextField) para que el usuario pueda ingresar los datos y tres botones (JButton) para retroceder (“<”), avanzar a la siguiente ventana (“>”) y guardar (“Guardar”) los datos anotados.

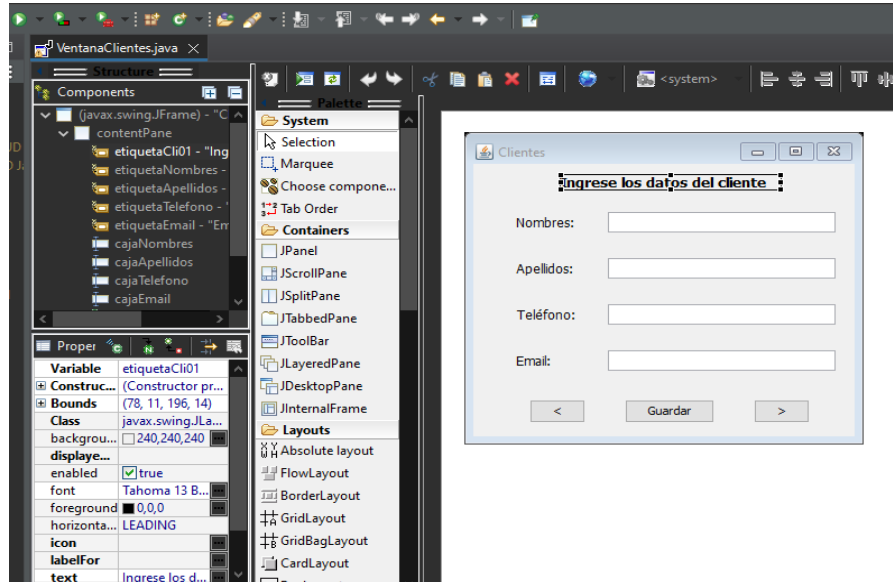


Imagen X: VentanaClientes

6. La próxima es la VentanaDirecciones que, al igual que la anterior, contiene etiquetas, cajones de texto y botones. En esta ventana se registrarán los datos para la tabla de direcciones.

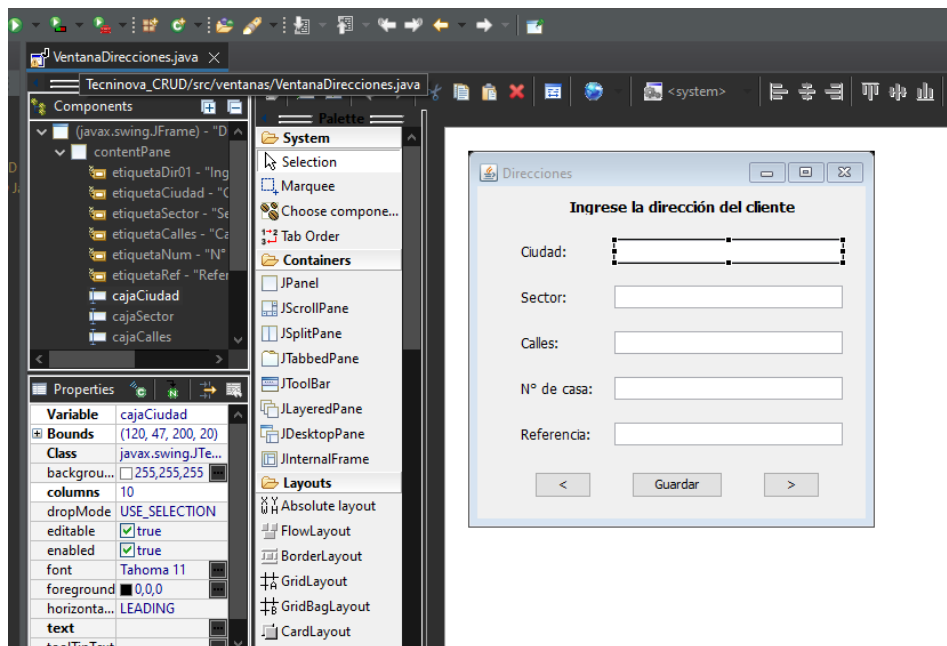


Imagen XI: VentanaDirecciones

7. Continuamos con la VentanaProductos, de igual forma q contiene los widgets para registrar los datos en la tabla de productos, que sería la tercera.

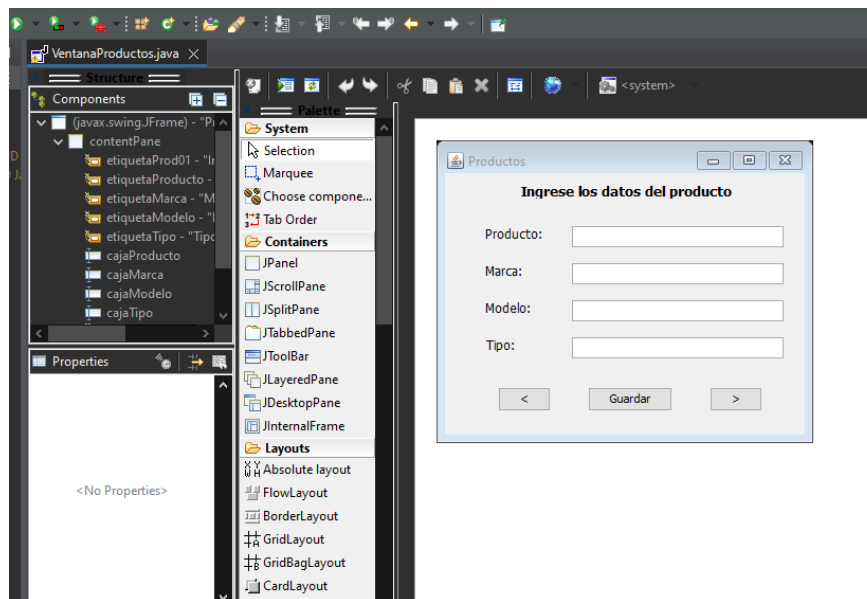


Imagen XII: VentanaProductos

8. La VentanaServicios es la última para ingresar un nuevo registro. Su contenido es similar a las tres anteriores, pero se eliminan los botones (“Guardar”) y avanzar (“>”), y se agrega un botón para finalizar (“Finalizar”) el registro y el cual lleva de regreso a la Ventana2.

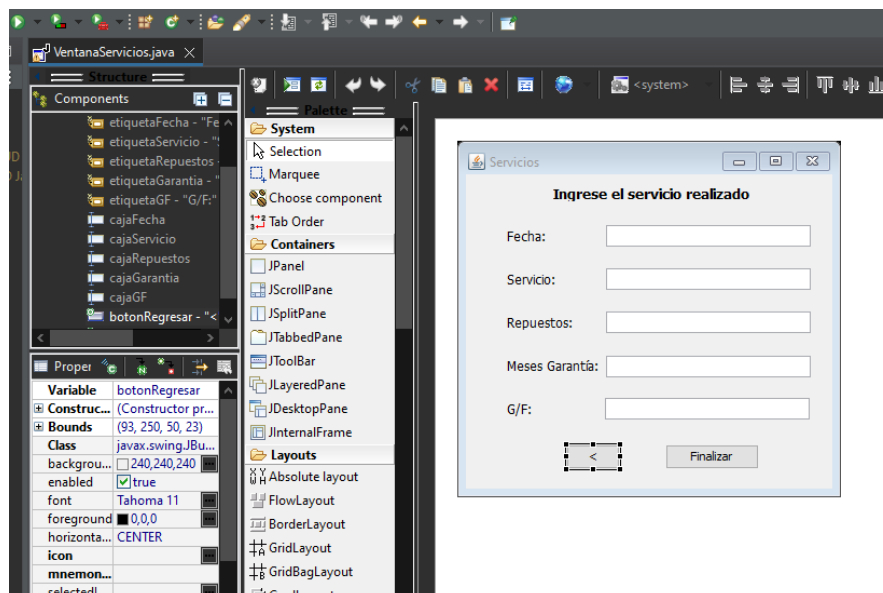


Imagen XIII: VentanaServicios

9. El botón “Clientes” de la Ventana2 dirige al usuario a la ventana VisorClientes, aquí se colocó un contenedor con scroll (JScrollPane) y dentro contiene la tabla de 5 columnas para mostrar todos los clientes de la base de datos. En el lado derecho contiene etiquetas (JLabel), cajones de texto (JTextField), y botones para “Seleccionar” y en función de lo seleccionado, se podrá “Actualizar” o “Eliminar” los datos al dar click en los respectivos botones. Y un botón “Atrás” para volver a la Ventana”.

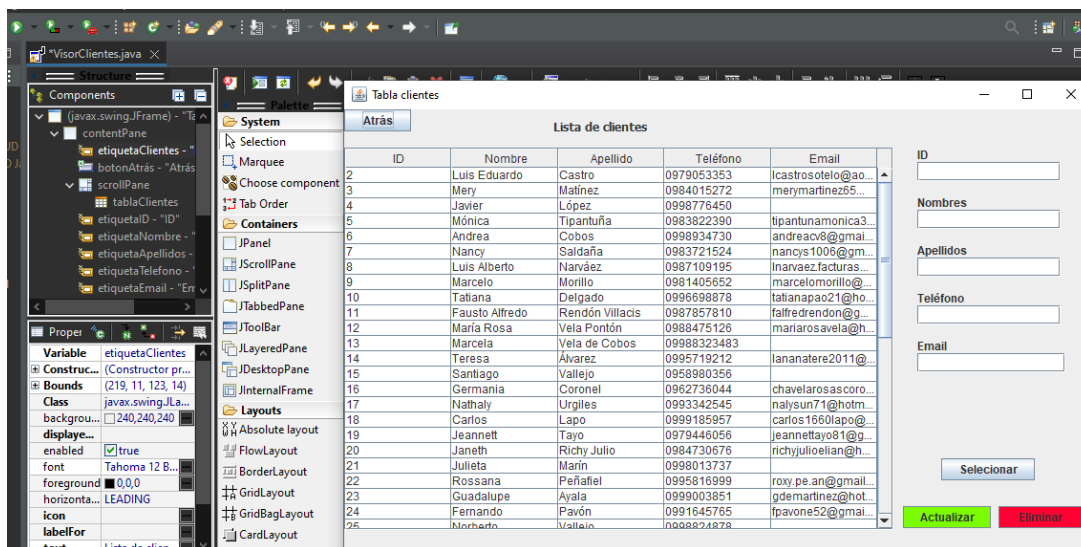


Imagen XIV: VisorClientes

10. El botón “Direcciones” de la Ventana2 dirige al usuario a la ventana VisorDirecciones. Así mismo, en esta ventana se podrán ver todas las direcciones de los clientes y se podrán “Seleccionar” para “Actualizar” o “Eliminar”.

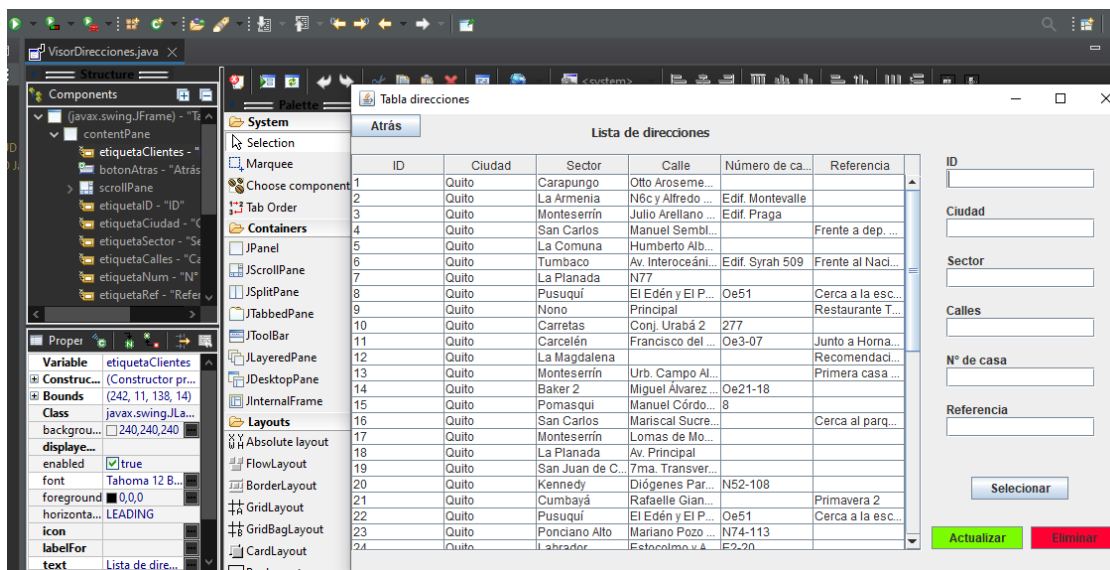


Imagen XV: VisorDirecciones

11. El botón “Productos” de la Ventana2 dirige al usuario a la ventana VisorProductos. De igual forma aquí se podrán visualizar todos los datos de los productos de los clientes en una tabla con seis columnas, con las mismas opciones de actualización y eliminación, y el botón “Atrás” que regresa a la Ventana2.

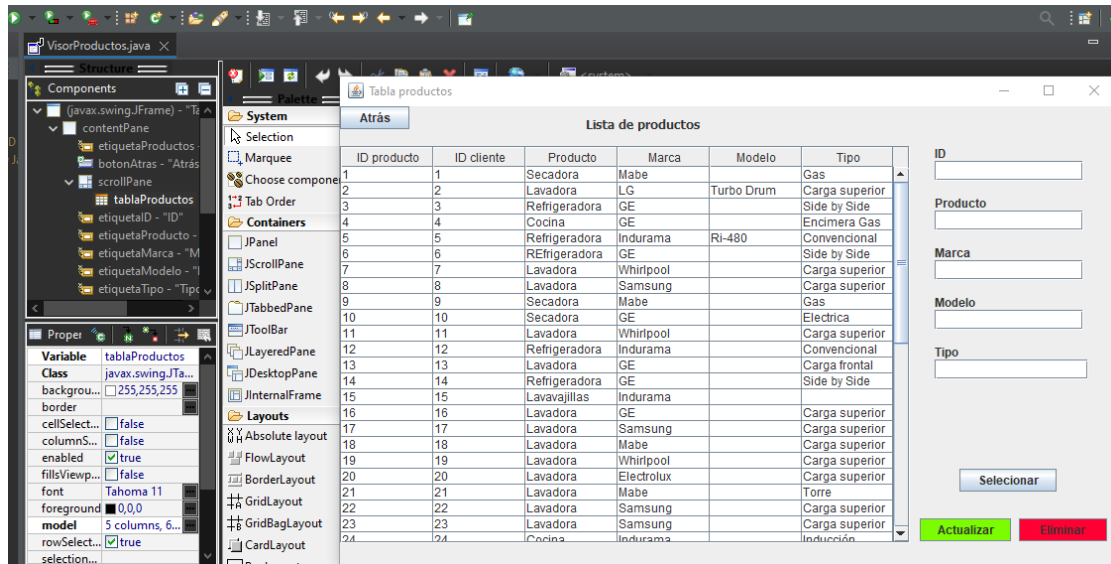


Imagen XVI: VisorProductos

12. El botón “Servicios” de la Ventana2 dirige al usuario a la ventana VisorServicios. También es una ventana para poder visualizar todos los servicios realizados, con las mismas opciones de los botones que las tres ventanas anteriores.

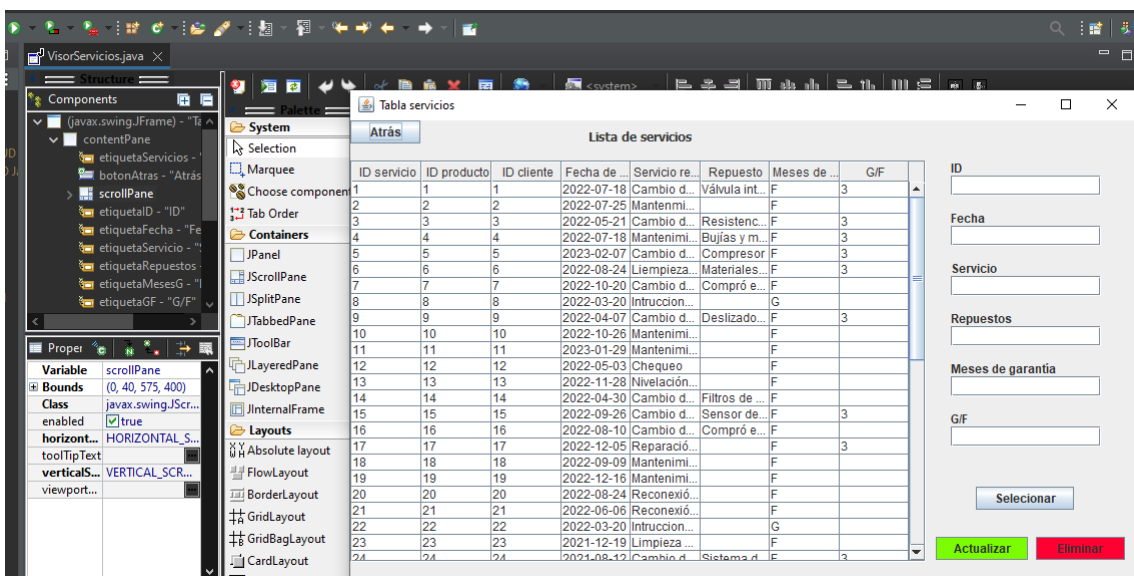


Imagen XVII: VisorServicios

13. Por último, se creó la VentanaResumen la cual se abrirá al presionar el botón “Resumen” de la Ventana2. Aquí se visualizarán los datos del cliente en cajones de texto y en la tabla de la ventana inferior se verán los datos principales de los productos y servicios del cliente mostrado. Los botones servirán para cambiar entre clientes y sus datos de uno en uno hacia atrás o hacia adelante, y también para ir al primer y al último registro.

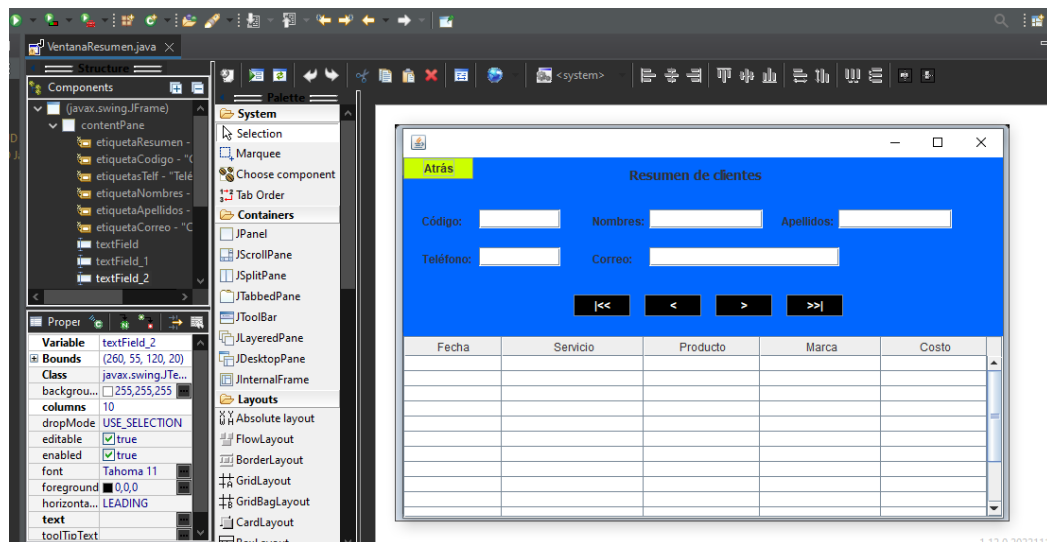


Imagen XVIII: VentanaResumen

8. Conclusiones y Recomendaciones

8.1. Conclusiones

- Podemos concluir que la realización del marco teórico nos dio la capacidad de poder verificar y comprobar la mejor forma de implementar el proyecto acorde a nuestros propios conocimientos.
- El uso de la herramienta WindowsBuidier facilita por mucho la creación de las ventanas en JAVA, además que nos dio la opción de ser un portal entre el usuario final y una base de datos lo cual a veces es complejo para otros usuarios que no hagan servicio a una base de datos.
- La base de datos creada permite la organización de los datos de manera profesional y ampliamente utilizado en la actualidad.
- El JDBC fue nuestra clave para la realización de todo, ya que gracias a esta API pudimos fácilmente conectar y ejecutar sentencias SQL a una base de datos, sin necesidad de reescribir un código debido a que este ya estaba escrito.
- Como conclusión del proyecto en general podemos mencionar la finalización exitosa del proyecto, con diversas funcionalidades y cambios efectuados en el mismo, acorde a lo solicitado por el profesor y adecuado a la situación actual de la empresa. Con adaptabilidad a cualquier cambio requerido o aplicación a otra empresa.

8.2. Recomendaciones

- Implementar un formato para los nombres de todas las etiquetas, cajas de texto, botones, etc., ayuda a que el código se vea organizado y facilita el reconocer a que parte pertenece cada widget.
- Es mejor tener las imágenes que se utilizarán, en la misma carpeta para no tener problemas con las rutas de almacenamiento.

- Cualquier edición del código es buena identificar para futuros programadores vean con claridad que se ha realizado, además de tratar de mantener la estructura de codificación.
- El proyecto final sufrió de muchos cambios significativos, muchos de ellos estéticos, así que para más detalle de los cambios efectuados puede visitar el link de GitHub para descargar el proyecto. [20]
- Se anexó también un código de conexión de base de datos para MySQL, pero el mismo no fue tratado a profundidad por motivos de tiempo.

9. Referencias

- [1] Centro Europeo de Postgrados, «CEUPE,» CEUPE, 1 julio 2020. [En línea]. Available: <https://www.ceupe.com/blog/que-es-la-programacion.html?dt=1663687973814>. [Último acceso: 20 septiembre 2022].
- [2] EDITORIAL ETECÉ, «CONCEPTO,» 5 agosto 2021. [En línea]. Available: <https://concepto.de/algoritmo-en-informatica/>. [Último acceso: 21 septiembre 2022].
- [3] TECNOLOGÍA FÁCIL, «TECNOLOGÍA FÁCIL,» [En línea]. Available: <https://tecnologia-facil.com/que-es/que-es-compile/>. [Último acceso: 21 septiembre 2022].
- [4] M. López Mendoza, «OPEN WEBINARS,» 16 julio 2020. [En línea]. Available: <https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/>. [Último acceso: 2022 septiembre 2022].
- [5] L. González, «DINAHOSTING,» 17 agosto 2018. [En línea]. Available: <https://dinahosting.com/blog/los-10-lenguajes-de-programacion-mas-usados/>. [Último acceso: 22 septiembre 2022].
- [6] devCamp, «devCamp,» [En línea]. Available: <https://devcamp.es/que-es-libreria-programacion/>. [Último acceso: 25 septiembre 2022].
- [7] tiThink, «tiThink,» 29 agosto 2018. [En línea]. Available: <https://www.tithink.com/es/2018/08/29/framework-o-librerias-ventajas-y-desventajas/>. [Último acceso: 25 septiembre 2022].
- [8] E. y. Navethe, Fundamentos de los sistemas de bases de datos, Pearson, 2016.

- [9] T. O. Ling Liu, «Enciclopedia de los sistemas de bases de datos,» Springer Link, 2020. [En línea]. Available: <https://link.springer.com/referencework/10.1007/978-0-387-39940-9>. [Último acceso: 13 03 2023].
- [10] S. y. rowe, «ACM digital library,» 2018. [En línea]. Available: <https://dl.acm.org/doi/abs/10.1145/3299869.3314045>. [Último acceso: 13 marzo 2023].
- [11] A. Widenius, «ACM digital library,» 2016. [En línea]. Available: <https://dl.acm.org/doi/abs/10.1145/3299869.3314058>. [Último acceso: 13 marzo 2023].
- [12] Balci, «Springer Link,» 2018. [En línea]. Available: https://link.springer.com/referenceworkentry/10.1007/978-1-4614-8265-9_223-2. [Último acceso: 13 marzo 2023].
- [13] S. Gosling, «Springer Link,» 2014. [En línea]. Available: https://link.springer.com/chapter/10.1007/978-3-319-05350-0_1. [Último acceso: 13 marzo 2023].
- [14] Oracle, «JAVA SE Development Kit,» 2021. [En línea]. Available: <https://www.oracle.com/java/technologies/downloads/#java16>. [Último acceso: 13 marzo 2023].
- [15] Koskela, Desarrollo de Java basado en pruebas con Spring y JUnit, Pack Publishing Ltda., 2018.
- [16] O. Charland, Desarrollo de aplicaciones móviles Xamarin para Android., Pack Publishing Ltda, 2017.
- [17] T. Hunt, El programador pragmático, Addison-Wesley Professional, 2017.
- [18] Eclipse, «Eclipse,» 2021. [En línea]. Available: <https://www.eclipse.org/windowbuilder/>. [Último acceso: 14 marzo 2023].
- [19] M. R. C., Arquitectura limpia, Prentice Hall, 2015.

- [20] R. Emmanuel, V. Erick y S. Javier, «Repositorio CRUD - GitHub,» 27 03 2023. [En línea]. Available: <https://github.com/erequena0999123/Repositorio-CRUD>.