

Proyecto Final Técnicas de Programación I

Lisete Tatiana Buitrago Palencia 506222007

Lisetet.buitragop@konradlorenz.edu.co

Resumen—Una compañía líder en Seguros de Latino América está requiriendo construir un prototipo que permita proveer la información necesaria para un estudio del índice de desigualdad GINI en los diferentes países de Latino América, para esto se implementan varios procesos que permiten realizar el trabajo en óptimas condiciones, procesos que irán explicándose a continuación en el presente documento.

Palabras clave—Json, Scopus, GINI, programación, etc.

I. PROCESO CON RESULTADOS INTELLIJ IDEA ENTORNO DE DESARROLLO



Figura 1. Copy of idea.png

Para poder realizar dicha conexión y construir el prototipo que pedía la compañía de Seguros, se utilizó IntelliJ IDEA como entorno de desarrollo y con el lenguaje de programación JAVA.

IntelliJ IDEA es un entorno de desarrollo que tiene varias funcionalidades que ayudan al desarrollo exitoso de cualquier proyecto realizable. Sus ventajas son las siguientes:

- Su inicio es sencillo, tanto así que tiene un asistente que ayuda a especificar los lenguajes que se desean, las herramientas de compilación y el JDK.
- Contiene una información detallada del código con el lenguaje de programación elegido, IntelliJ IDEA proporciona un mapa virtual del proyecto donde se pueden detectar errores que tal vez no se hayan observado anteriormente.
- El entorno cuenta con los diagramas para visualizar y analizar el código y navegar, la ventaja de los diagramas es que puede reflejar las estructuras de clases y métodos.
- Administra bases de datos directamente, puede consultar SQL, conectarse a bases de datos dinámicas y todo esto también permite gestionar una interfaz visual.

II. ESTRUCTURA BASE DEL PROYECTO

El proyecto se entrega con una arquitectura 3 capas que se denominan la lógica, la persistencia y presentación, cada una cumple con un objetivo para lograr que el proyecto sea generado de una forma correcta.

Para desarrollar la estructura base, inicialmente se crearon los paquetes que separan cada una de las capas. En la persistencia se encarga de separar el acceso a datos, en la lógica se encarga de llevar el código, este permite la sincronización del paquete de persistencia con el de presentación, en la presentación se encarga de las interfaces. Se creó una interfaz denominada BuscandoPaíses la interfaz tiene su formulario generado donde se encuentran los datos que se requieren para solucionar el problema inicial. Son archivos .form lo que nos indica que contiene una representación de un formulario de interfaz gráfica de usuario en java.

En el siguiente paso se crearon dos clases, una en el paquete de lógica que se nombra como DatosPaísesDTO, el DTO (Data Transfer Object), el cual se encarga de consolidar los datos de las interfaces, realizando las tareas adicionales para esos datos y así mismo transferirlos a la capa de persistencia.

La siguiente clase creada se llamó DatosPaísesDao, el DAO (Data Access Object) encargándose de todo el acceso a los datos.

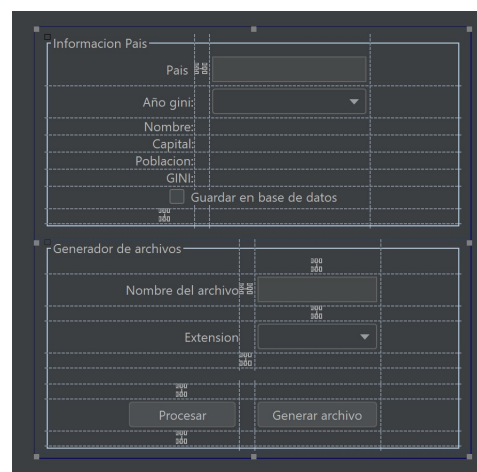


Figura 2. Estructura de la interfaz

III. INTERFACES CARGADAS

Paso A Paso

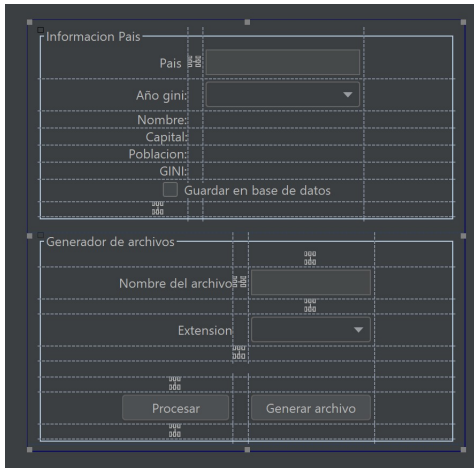


Figura 3. Estructura de la interfaz

Siguiendo del proceso anteriormente mencionado, al definir todo se recomienda utilizar siempre un GridLayout, ya que este define la posición de los elementos en el JFrame, el JFrame es la ventana general de la interfaz que se está realizando.

También se utilizaron los siguientes aspectos para las interfaces:

- JLabel: Este es un objeto que se usa para dibujar en el JFrame que se define a través de una etiqueta como una expresión estática.

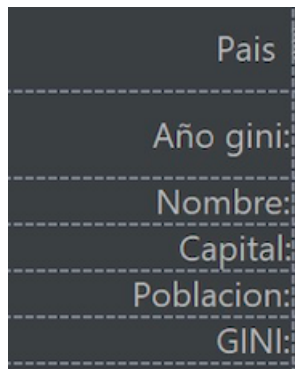


Figura 4. JLabel con datos definidos

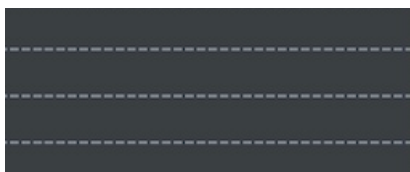


Figura 5. JLabel vacios que se llenan con los resultados generados por la API

- JButton: Este tiene como objetivo que el usuario pueda interactuar con el. Atrapando el clic del objeto con la implementacion de un ActionListener.



Figura 6. JButton

- JTextField: Es un componente de campo de texto que permite suministrar un espacio para ingresar textos, ya sea para obtener datos o mostrar datos.



Figura 7. JTextField



Figura 8. JTextField

- JComboBox: Este permite seleccionar un string de una lista.

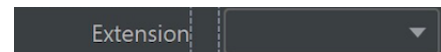


Figura 9. JComboBox



Figura 10. JComboBox

- **CheckBox:** Estos son botones de comprobaciones y se utilizan generalmente como botones de estado. Una de sus características es que se puede seleccionar mas de una si así lo requiere la aplicacion.

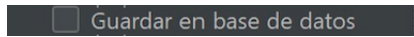


Figura 11. CheckBox

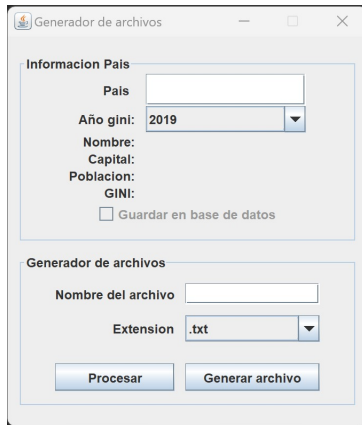


Figura 12. Interfaz Finalizada

IV. DISEÑO UML

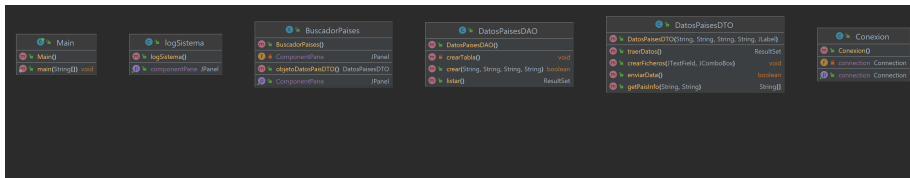


Figura 13. Interfaz Finalizada

V. CONCLUSIONES

Como conclusión se logra realizar la interfaz principal que conecta con la API, y nos logra mostrar los países con su GINI actualizado, algunos países no lograban obtener el GINI porque no han sido actualizados despues del año 2019, para eso se agrego un JComboBox que deja aplicar una opcion por año para así obtener el índice. Se intento realizar la otra interfaz para la fecha y demas dependencias, pero no se logró ya que se trato de hacer metodos Getter que iba a permitir obtener los label y modificar los textos con el metodo setText cuando se le hiciera clic al CheckBox de guardar datos en la base de datos, sin embargo por algun motivo que desconozco cuando iba a establecer el texto para el label no cambiaba.

VI. REFERENCIAS

REFERENCIAS

- [1] 2023 <https://restcountries.com/v3.1/name/name>
- [2] <https://restcountries.com/>