Diccionario de clase, métodos y variables

Busqueda\_una\_tablaController: Controlador de la interfaz gráfica definida en el archivo FXML.

Métodos:

• initialize(URL url, ResourceBundle rb): Método llamado al inicializar el controlador.

• choicebox\_action(): Método para manejar la acción de la ChoiceBox.

• click\_continuar(ActionEvent event): Método para manejar el clic en el botón "Continuar".

• click\_volver(ActionEvent event): Método para manejar el clic en el botón "Volver".

• setConnection(JDBC connection): Método para establecer la conexión.

• getConection(): Método estático para obtener la conexión.

Variables:

• connection: Variable estática que almacena la conexión a la base de datos.

• btn\_continuar, btn\_volver, desp\_tabla: Elementos gráficos de la interfaz.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BusquedasController: Controlador de la interfaz gráfica definida en el archivo FXML.

Métodos:

• initialize(URL url, ResourceBundle rb): Método llamado al inicializar el controlador.

• click\_una\_tabla(ActionEvent event): Método para manejar el clic en el botón "Buscar una tabla".

• click\_dos\_tablas(ActionEvent event): Método para manejar el clic en el botón "Buscar dos tablas".

• click\_volver(ActionEvent event): Método para manejar el clic en el botón "Volver".

Variables:

• btn\_una\_tabla, btn\_dos\_tablas, btn\_volver: Elementos gráficos de la interfaz.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Busquedas\_dos\_tablasController: Controlador de la interfaz gráfica definida en el archivo FXML.

Métodos:

• initialize(URL url, ResourceBundle rb): Método llamado al inicializar el controlador.

• choicebox1\_action(): Método para manejar la acción de la ChoiceBox 1.

• choicebox2\_action(): Método para manejar la acción de la ChoiceBox 2.

• click\_continuar(ActionEvent event): Método para manejar el clic en el botón "Continuar".

• click\_volver(ActionEvent event): Método para manejar el clic en el botón "Volver".

• setConnection(JDBC connection): Método para establecer la conexión.

• getConection(): Método estático para obtener la conexión.

Variables:

• connection: Variable estática que almacena la conexión a la base de datos.

• btn\_continuar, btn\_volver, desp\_tabla2, desp\_tabla1: Elementos gráficos de la interfaz.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Condiciones\_busquedaController: Controlador de la interfaz gráfica definida en el archivo FXML.

Métodos:

• initialize(URL url, ResourceBundle rb): Método llamado al inicializar el controlador.

• esNoNumerico(String valor): Método para verificar si un valor es numérico o no.

• queryUnaTabla(): Método para construir la consulta SQL para una tabla.

• showAlert(String message): Método para mostrar una alerta.

• click\_continuar(ActionEvent event): Método para manejar el clic en el botón "Continuar".

• click\_volver(ActionEvent event): Método para manejar el clic en el botón "Volver".

• click\_mas1(ActionEvent event): Método para manejar el clic en el botón "+".

• click\_menus1(ActionEvent event): Método para manejar el clic en el botón "-".

Variables:

• connection: Variable estática que almacena la conexión a la base de datos.

• tablaSelected: Variable estática que almacena el nombre de la tabla seleccionada.

• choiceBoxAtributos: ArrayList que almacena las ChoiceBox dinámicas creadas durante la ejecución.

• Otras variables para gestionar la interfaz gráfica y el estado del formulario.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Condiciones\_busqueda\_cruzadaController: Controlador de la interfaz gráfica definida en el archivo FXML.

Métodos:

• initialize(URL url, ResourceBundle rb): Método llamado al inicializar el controlador.

• Métodos con anotaciones @FXML para manejar acciones específicas en la interfaz gráfica.

• click\_masTabla1(ActionEvent event): Maneja el clic en el botón "+" para agregar atributos de la Tabla 1.

• click\_menusTabla1(ActionEvent event): Maneja el clic en el botón "-" para eliminar atributos de la Tabla 1.

• click\_masTabla2(ActionEvent event): Maneja el clic en el botón "+" para agregar atributos de la Tabla 2.

• click\_menusTabla2(ActionEvent event): Maneja el clic en el botón "-" para eliminar atributos de la Tabla 2.

• AndOr2(), operador\_condicion1(), operador\_condicion2(): Configuran opciones en ChoiceBox.

• condicion1TablasCruzadas(), condicion2TablasCruzadas(): Establecen opciones en ChoiceBox para condiciones de las Tablas 1 y 2.

• operadorRelacion(), AndOr1(): Configuran opciones en ChoiceBox para la relación entre tablas.

• atributoTabla1Relacion(), atributoTabla2Relacion(): Establecen opciones en ChoiceBox para los atributos de las Tablas 1 y 2.

• click\_volver(ActionEvent event): Maneja el clic en el botón "Volver".

• click\_continuar(ActionEvent event): Maneja el clic en el botón "Continuar".

Variables:

• connection: Variable estática que almacena la conexión a la base de datos.

• tablaSelected1, AcronimoTabla1, tablaSelected2, AcronimoTabla2: Variables estáticas que almacenan información sobre las tablas seleccionadas.

• choiceBoxAtributosTabla1, choiceBoxAtributosTabla2: ArrayLists que almacenan las ChoiceBox dinámicas creadas durante la ejecución.

• Otras variables para gestionar la interfaz gráfica y el estado del formulario.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Diccionario de la Clase Crear\_tablaController:

Variables:

• connection: Objeto de la clase JDBC para la conexión a la base de datos.

• tablaSelected: Nombre de la tabla seleccionada.

• choiceBoxAtributos: ArrayList de ChoiceBox para almacenar tipos de datos de atributos.

• textFieldAtributosName: ArrayList de TextField para almacenar nombres de atributos.

• txt\_nombre: TextField para ingresar el nombre de la nueva tabla.

• btn\_crear, btn\_volver, btn\_mas1, btn\_menus1: Botones de la interfaz gráfica.

• scrollPane: Panel de desplazamiento para la interfaz gráfica.

• vBoxAddAtributos: VBox para agregar dinámicamente atributos a la interfaz.

• contadorAtributos: Contador para el número de atributos agregados dinámicamente.

Métodos:

• initialize: Método de inicialización de la interfaz.

• queryCrear: Construye y devuelve la consulta SQL para crear la nueva tabla.

• showAlert: Muestra una alerta en caso de que falten campos.

• click\_crear: Maneja el evento de creación de la tabla.

• click\_volver: Maneja el evento de volver a la ventana anterior.

• click\_mas1: Maneja el evento de agregar un nuevo atributo a la tabla.

• click\_menus1: Maneja el evento de eliminar el último atributo agregado.

Descripción General: La clase Crear\_tablaController representa el controlador para la creación de nuevas tablas en la base de datos. Permite al usuario ingresar el nombre de la nueva tabla y agregar dinámicamente atributos con sus respectivos tipos de datos. Utiliza un conjunto de ChoiceBox para seleccionar los tipos de datos y TextField para ingresar los nombres de los atributos. Proporciona métodos para construir y ejecutar la consulta SQL de creación de la tabla. Implementa funciones para manejar eventos de botones y controlar la interfaz gráfica.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Diccionario de la Clase Crear\_vistaController:

Variables:

• btn\_volver, btn\_crear: Botones de la interfaz gráfica.

Métodos:

• initialize: Método de inicialización de la interfaz.

• click\_volver: Maneja el evento de volver a la ventana de resultados de búsqueda.

• click\_crear: Maneja el evento de crear una vista (no está implementado correctamente).

Descripción General: La clase Crear\_vistaController representa el controlador para la creación de vistas en la interfaz de usuario. Proporciona métodos para manejar eventos de botones y controlar la navegación en la interfaz gráfica. No se ha implementado correctamente el método click\_crear, ya que intenta cargar una escena que no existe ("/proyecto\_bases\_datos/FXMLResultado\_busquedas1.fxml"). Puede ser un error tipográfico o falta de archivo.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Diccionario de la Clase Insertar\_registroController:

Variables:

• btn\_insertar, btn\_refresh, btn\_volver: Botones de la interfaz gráfica.

• txt\_titulo: Etiqueta de título en la interfaz gráfica.

• vBoxRegistros: Contenedor vertical para los elementos de la interfaz.

• atributos: Lista de nombres de atributos.

• tipos: Lista de tipos de atributos.

• listaTextField: Lista de campos de texto para ingresar valores.

• tablaSelected: Nombre de la tabla seleccionada.

Métodos:

• initialize: Método de inicialización de la interfaz.

• refrescar: Método para actualizar la interfaz según las listas de atributos y tipos.

• queryInsertar: Método para construir la consulta de inserción basada en los valores ingresados.

• click\_insertar: Maneja el evento de inserción de un nuevo registro.

• click\_volver: Maneja el evento de volver a la vista de tablas.

Descripción General: La clase Insertar\_registroController representa el controlador para la inserción de registros en la interfaz de usuario. Proporciona métodos para manejar eventos de botones, construir consultas SQL y controlar la navegación en la interfaz gráfica. La interfaz permite al usuario ingresar valores para cada atributo de la tabla seleccionada y realizar la inserción en la base de datos. Maneja posibles excepciones de SQL y muestra un mensaje de error en caso de problemas con la base de datos.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Diccionario de la Clase InterfazController:

Variables:

• T: Pane utilizado como contenedor principal en la interfaz gráfica.

• root: Nodo raíz de la interfaz gráfica.

• txt\_usr, txt\_puerto, txt\_clave: Campos de texto para ingresar usuario, puerto y contraseña.

• btn\_acceder: Botón para acceder o realizar alguna acción en la interfaz.

• desp\_maquina: ChoiceBox (cuadro de selección) utilizado en la interfaz.

• connection: Objeto de la clase JDBC para gestionar la conexión a la base de datos.

Métodos:

• initialize: Método de inicialización de la interfaz.

• acceder\_action: Maneja el evento de acceder. Realiza la validación de campos, crea una conexión JDBC y redirige a la siguiente interfaz si la conexión es exitosa.

Descripción General: La clase InterfazController representa el controlador para la interfaz de inicio, donde se ingresan los datos de conexión a la base de datos. Proporciona métodos para manejar eventos de botones, validar campos de entrada y gestionar la transición a la siguiente interfaz. Realiza la creación de una conexión JDBC y, en caso de éxito, redirige a la interfaz del menú principal (Menu.fxml). Muestra un mensaje de error si algún campo está vacío o si hay problemas con la conexión a la base de datos.

Diccionario de la Clase KeyController:

Variables:

• conexion: Objeto de la clase JDBC para gestionar la conexión a la base de datos.

• btn\_volver: Botón para regresar a la interfaz anterior.

• desp\_tipo\_dato: ComboBox para seleccionar el tipo de dato.

• txt\_nombre: Campo de texto para ingresar el nombre.

• btn\_crear: Botón para realizar la acción de creación.

• CheckPrimaryKey, CheckForeignKey: CheckBox para seleccionar si es clave primaria o foránea.

• tableSelected: Nombre de la tabla seleccionada.

Métodos:

• initialize: Método de inicialización de la interfaz. Llena los tipos de datos y establece listeners para CheckBox.

• setConexion: Establece la conexión JDBC.

• setTableSelected: Establece la tabla seleccionada.

• click\_volver: Maneja el evento de regresar a la interfaz anterior.

• llenarTiposDatos: Llena el ComboBox con los tipos de datos disponibles.

• click\_crear: Maneja el evento de creación. Realiza la validación de campos y ejecuta la acción correspondiente en la base de datos.

Descripción General: La clase KeyController representa el controlador para la interfaz de creación de claves (primarias o foráneas) en una tabla de la base de datos. Proporciona métodos para inicializar la interfaz, establecer la conexión JDBC, manejar eventos de botones y realizar la acción de creación en la base de datos. Incluye lógica para la creación de claves primarias y foráneas, con validaciones y mensajes de error adecuados. Utiliza objetos de la clase JDBC para interactuar con la base de datos.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Diccionario de la Clase MenuController:

Variables:

• conexion: Objeto de la clase JDBC para gestionar la conexión a la base de datos.

• desp\_bases: ComboBox para seleccionar la base de datos.

• btn\_acceder, btn\_eliminar, btn\_crear, btn\_volver, btn\_exit: Botones para realizar diversas acciones.

• raiz: Pane de la interfaz.

Métodos:

• initialize: Método de inicialización de la interfaz. Llama al método choiceboxAction si hay una conexión.

• setConexion: Establece la conexión JDBC.

• getConexion: Devuelve la conexión actual.

• choiceboxAction: Maneja el evento de la ComboBox para cargar las bases de datos disponibles.

• click\_acceder: Maneja el evento para acceder a la interfaz de Tablas.

• click\_eliminar: Maneja el evento para eliminar la base de datos seleccionada.

• click\_crear: Maneja el evento para acceder a la interfaz de creación de nueva base de datos.

• click\_volver: Maneja el evento para volver a la interfaz de inicio.

• click\_exit: Maneja el evento para cerrar la aplicación.

Descripción General: La clase MenuController representa el controlador para la interfaz principal que muestra las bases de datos disponibles. Proporciona métodos para inicializar la interfaz, establecer la conexión JDBC, manejar eventos de botones y realizar diversas acciones en la base de datos. Incluye lógica para cargar las bases de datos, acceder a la interfaz de Tablas, eliminar bases de datos, crear nuevas bases de datos y gestionar el cierre de la aplicación. Utiliza objetos de la clase JDBC para interactuar con la base de datos.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Diccionario de la Clase Modificar\_registroController:

Variables:

• btn\_modificar, btn\_refrescar, btn\_volver: Botones para realizar diversas acciones.

• scrollPane: Panel de desplazamiento para manejar un conjunto de nodos hijos que pueden ser más grandes que el área de visualización.

• atributosActualizar: VBox para contener los atributos que se van a modificar.

• primaryKey: ChoiceBox para seleccionar la clave primaria.

• operadorPrimaryKey: ChoiceBox para seleccionar el operador de la clave primaria.

• valorCondicionPrimaria: TextField para ingresar el valor de la condición de la clave primaria.

• query: Variable para almacenar la consulta SQL.

• tableName: Nombre de la tabla.

• conexion: Objeto de la clase JDBC para gestionar la conexión a la base de datos.

• informacionFila: Lista observable que contiene la información de la fila seleccionada.

• columnasSeleccionadas: Lista que contiene las columnas seleccionadas.

• valores: Lista de TextField para almacenar los valores de las columnas a modificar.

Métodos:

• initialize: Método de inicialización de la interfaz.

• refrescar: Método para refrescar la interfaz y mostrar los atributos seleccionados para modificar.

• click\_modificar: Método que maneja el evento de modificación de los registros en la base de datos.

• click\_volver: Método que maneja el evento de volver a la interfaz de resultados de búsquedas.

• Métodos setters para establecer la conexión, la consulta SQL, el nombre de la tabla, la información de la fila y las columnas seleccionadas.

Diccionario de la Clase Modificar\_tablaController:

Variables:

• conexion: Objeto de la clase JDBC para gestionar la conexión a la base de datos.

• tableSelected: Nombre de la tabla seleccionada.

• desp\_campos\_modificables: ComboBox para seleccionar los campos que se van a modificar.

• desp\_campos\_key: ComboBox para seleccionar el tipo de clave que se va a agregar.

• lbl\_nombretabla: Label para mostrar el nombre de la tabla que se está modificando.

• txt\_nuevo\_nombre: TextField para ingresar el nuevo nombre de la columna.

• btn\_modificar, btn\_añadir\_campo, btn\_borrar, btn\_añadir\_key, btn\_volver: Botones para realizar diversas acciones.

• desp\_tipo\_dato: ComboBox para seleccionar el nuevo tipo de dato de la columna.

Métodos:

• initialize: Método de inicialización de la interfaz.

• setConnection: Método para establecer la conexión a la base de datos.

• cambioLBL: Método para cambiar el texto del label (lbl\_nombretabla) mostrando la tabla que se está modificando.

• llenarTiposDatos: Método para llenar el ComboBox (desp\_tipo\_dato) con los tipos de datos disponibles.

• choicebox\_action: Método para llenar el ComboBox (desp\_campos\_modificables) con los campos de la tabla seleccionada.

• choiceboxkey\_action: Método para llenar el ComboBox (desp\_campos\_key) con los tipos de clave disponibles.

• setTableSelected: Método para establecer el nombre de la tabla seleccionada.

• click\_modificar: Método que maneja el evento de modificar una columna de la tabla.

• click\_campo, click\_atributo, click\_key, click\_volver: Métodos que manejan diferentes eventos de botones.

Descripción General: La clase Modificar\_tablaController representa el controlador para la interfaz de modificación de la estructura de una tabla. Proporciona métodos para inicializar la interfaz, establecer la conexión JDBC, manejar eventos de botones y realizar acciones como modificar campos, añadir claves, entre otras. Utiliza objetos de la clase JDBC para interactuar con la base de datos.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Diccionario de la Clase Nueva\_baseDController:

Variables:

• conexion: Objeto de la clase JDBC para gestionar la conexión a la base de datos.

• txt\_nombre\_baseD: TextField para ingresar el nombre de la nueva base de datos.

• btn\_crear, btn\_volver: Botones para realizar la creación de la nueva base de datos y volver al menú principal, respectivamente.

Métodos:

• initialize: Método de inicialización de la interfaz.

• setConnection: Método para establecer la conexión a la base de datos.

• click\_crear: Método que maneja el evento de crear una nueva base de datos.

• click\_volver: Método que maneja el evento de volver al menú principal.

Descripción General: La clase Nueva\_baseDController representa el controlador para la interfaz de creación de una nueva base de datos. Proporciona métodos para inicializar la interfaz, establecer la conexión JDBC, y manejar eventos de botones para la creación y retorno al menú principal. Utiliza objetos de la clase JDBC para interactuar con la base de datos.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Resultado\_busquedas1Controller:

Métodos:

• initialize(URL url, ResourceBundle rb): Método de inicialización del controlador.

• setConnection(JDBC connection): Establece la conexión JDBC.

• setQuerie(String querie): Establece la consulta SQL.

• setNombreTabla(String nombre): Establece el nombre de la tabla.

• click\_volver(ActionEvent event): Maneja el evento de clic en el botón "Volver".

• click\_crear\_vista(ActionEvent event): Maneja el evento de clic en el botón "Crear Vista".

• click\_borrar(ActionEvent event): Maneja el evento de clic en el botón "Borrar".

• isNumeric(String str): Verifica si una cadena es un número.

• click\_izquierdo(): Maneja el evento de clic izquierdo en la tabla.

• click\_modificar(ActionEvent event): Maneja el evento de clic en el botón "Modificar".

• usarInformacion(): Ejecuta la consulta SQL y carga los datos en la tabla.

Variables:

• btn\_volver, btn\_crear\_vista, btn\_borrar, btn\_modificar: Botones de la interfaz.

• tabPane\_Tablaresultado: TableView que muestra los resultados de la consulta.

• queriee: Consulta SQL.

• TablaName: Nombre de la tabla.

• conexion: Objeto JDBC para la conexión a la base de datos.

• columnasSeleccionadas: Lista de nombres de columnas seleccionadas.

• informacionFila: Lista de información de la fila seleccionada.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TablasController:

Métodos:

• initialize(URL url, ResourceBundle rb): Método de inicialización del controlador.

• usarInformacion(): Crea pestañas en el TabPane para cada tabla en la base de datos y muestra la estructura de la tabla.

• setConnection(JDBC connection): Establece la conexión JDBC.

• getConection(): Obtiene la conexión JDBC.

• Métodos asociados a los botones:

• click\_modificar(ActionEvent event): Maneja el evento de clic en el botón "Modificar".

• click\_crear(ActionEvent event): Maneja el evento de clic en el botón "Crear".

• click\_borrar(ActionEvent event): Maneja el evento de clic en el botón "Borrar".

• click\_volver(ActionEvent event): Maneja el evento de clic en el botón "Volver".

• click\_borrar\_registro(ActionEvent event): Maneja el evento de clic en el botón "Borrar Registro".

• click\_nuevo\_dato(ActionEvent event): Maneja el evento de clic en el botón "Nuevo Dato".

• click\_buscar(ActionEvent event): Maneja el evento de clic en el botón "Buscar".

Variables:

• conexion: Objeto JDBC para la conexión a la base de datos.

• dataBaseSelected: Nombre de la base de datos seleccionada.

• TabPane\_Tablas: Pane que contiene las pestañas de las tablas.

• btn\_buscar, btn\_modificar, btn\_crear, btn\_borrar, btn\_volver, btn\_modificar\_registro, btn\_borrar\_registro, btn\_crear\_datos: Botones de la interfaz.

• box\_SeleccionTabla: ChoiceBox para seleccionar una tabla.

• GETTABLES: Consulta SQL para obtener la lista de tablas.