

## Sesión # 20: Componente Práctico

### Proyecto Integrador - Parte II

1. Cree el esqueleto de la vista requerida para la solución (archivo fxml y controlador) y haga su conexión al proyecto. Utilice tabs para cada una, se sugieren las siguientes pestañas: Tripulantes, Instructores, Cursos

Ejemplo:



2. Añada a las pestañas los componentes necesarios para obtener información y cumplir con los requerimientos del cliente (administración MINTIC) recuerde aplicar los métodos get y set para interactuar con el controlador y la vista.
3. Implemente en los modelos correspondientes las funciones/transacciones básicas en bases de datos CRUD (create (crear), read (leer), update (actualizar), delete (eliminar)). Utilice el componente textArea para listar datos y formularios para transacciones de registro, actualización y eliminación.

Para las pestaña de Instructores y Tripulantes implemente los métodos: Crear, Actualizar, Eliminar, Limpiar formulario y Listar información que mostrará en un textarea.

Para la pestaña de cursos: Implemente los métodos Crear, limpiar formulario y Listar información que mostrará en un textarea.

4. Realice pruebas a su proyecto y haga las validaciones pertinentes para garantizar el éxito de las transacciones.

Ejemplos:

### MISIÓN TIC 2021

InstructoresTripulantesCursos

No. Identificacion

Nombre

Apellido

Email

Curso

CREAR

ACTUALIZAR

ELIMINAR

LIMPIAR

### MISIÓN TIC 2021

InstructoresTripulantesCursos

No. Identificacion

Nombre

Apellido

Edad

Curso

CREAR

ACTUALIZAR

ELIMINAR

LIMPIAR

## MISIÓN TIC 2021

Instructores
Tripulantes
Cursos

Nombre del curso

Horario

▼

Instructor

▼

CREAR

LIMPIAR

### Instrucciones

- Cree los métodos para transacciones a la base de datos desde la clase modelo de cada tabla y haga el llamado de ellos donde corresponda en el controlador.
- Utilice choicebox para listar opciones. Ejemplo metodo para listar cursos en componente choicebox:

```
selectCursoTripulante.setItems(Curso.selectCursos());
```

El metodo Curso.selectCursos en el modelo es:

```
public static ObservableList<String> selectCursos() {
    String query = "SELECT * from cursos order by id asc;";
    ObservableList<String> cursos =
    FXCollections.observableArrayList("Seleccione un curso");

    try (Connection conn = Connect.connect(); Statement stmt =
    conn.createStatement()) {
        ResultSet resultSet = stmt.executeQuery(query);

        while (resultSet.next()) {
            cursos.add(resultSet.getInt("id") + "-" +
            (resultSet.getString("nombre")));
        }
        stmt.close();
        resultSet.close();
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
    return cursos;
}
```

- Para eliminar y actualizar registros haga las búsquedas de los registros por medio del campo num de identificación.
- Después de cada transacción retorne un mensaje del resultado de la misma hacia el controlador y envíelo a su vista. Ejemplo: luego de crear un tripulante, retorne el mensaje *'Tripulante creado exitosamente'* y muestre el mismo en un label de su vista.
- Para mostrar en el TextArea los registros puede hacer llamado de un método como el siguiente y posteriormente setear su resultado en el componente:

```
public static String listaCursos() {
    String query = "SELECT * from cursos order by id asc ";
    String cursos = "";
    try (Connection conn = Connect.connect(); Statement stmt = conn.createStatement()) {
        ResultSet resultSet = stmt.executeQuery(query);
        ResultSetMetaData rsmd = resultSet.getMetaData();
        int columnsNumber = rsmd.getColumnCount();

        while (resultSet.next()) {
            for (int i = 1; i <= columnsNumber; i++) {
                String columnValue = resultSet.getString(i);
                cursos = cursos + rsmd.getColumnName(i) + ": " + " " + columnValue + " ";
                if (i == columnsNumber)
                    cursos = cursos + "\n";
            }
        }
        return cursos;
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
    return cursos;
}
```

- Para que pueda visualizar los cambios efectuados en la base de datos en tiempo real, implemente en su controlador un método que actualice su TextArea luego de cada acción (crear, eliminar, actualizar) y llámelo en cada una de ellas.

*Utilice este componente para practicar el siguiente reto.*