

Situar un campo imagen en una celda de un TableView

Supongamos que queremos visualizar en un tableView los atributos de una clase Persona que tiene una propiedad de tipo String y un atributo de tipo Image. En la inicialización del controlador definiremos el `setCellValueFactory` y `setCellFactory` para las dos columnas de modo que se pueda ver tanto la imagen como el nombre.

```
package modelo;

import javafx.beans.property.SimpleStringProperty;
import javafx.beans.property.StringProperty;
import javafx.scene.image.Image;

/**
 *
 * @author Juan
 */
public class Persona {
    private StringProperty nombre = new SimpleStringProperty();
    private Image avatar; // representación de la imagen persona
    public Persona(StringProperty nombre, Image avatar) {
        this.nombre = nombre;
        this.avatar = avatar;
    }

    public Persona() {
    }

    public final void setNombre(String value) {
        nombre.set(value);
    }
    // resto de setters y getters
}
```

Queremos incluir el primer atributo (nombre) en la primera columna y la imagen (avatar) en la segunda.

En la inicialización del controlador debemos situar el siguiente código:

```
public class FXMLDocumentController implements Initializable {

    @FXML
    private TableView<Persona> tableView;
    @FXML
    private TableColumn<Persona, String> columnaNombre;
    @FXML
    private TableColumn<Persona, Image> columnaAvatar;

    private ObservableList<Persona> misPersonas = FXCollections.observableArrayList();

    private Connect4 connect4 = null;

    @Override
    public void initialize(URL url, ResourceBundle rb) {

        columnaNombre.setCellValueFactory( celda -> celda.getValue().nombreProperty());
        columnaAvatar.setCellValueFactory( celda2 -> new SimpleObjectProperty<Image>
                                                (celda2.getValue().getAvatar()));
        // de forma alternativa
        //columnaAvatar.setCellValueFactory( new PropertyValueFactory<Persona, Image>("avatar"));

        columnaAvatar.setCellFactory(new Callback<TableColumn<Persona, Image>, TableCell<Persona,
                                                Image>>() {

            @Override
            public TableCell<Persona, Image> call(TableColumn<Persona, Image> columna) {
                return new TableCell<Persona,Image>() {

```

```

        private ImageView view = new ImageView();
        @Override
        protected void updateItem(Image item, boolean empty) {
            super.updateItem(item, empty);
            if (item == null || empty) setGraphic(null);
            else {
                view.setFitWidth(50); // fijar anchura
                view.setFitHeight(60); // fijar altura
                view.setImage(item);
                setGraphic(view);
            }
        }
    };
}
});

// se obvia el código de connexion con la bd

Player player2 = connect4.getPlayer("nickName9");

String Nombre = player2.getNickName();
Image imagen = player2.getAvatar();

misPersonas = FXCollections.observableArrayList();
Persona p = new Persona();
p.setAvatar(imagen); p.setNombre(Nombre);
misPersonas.add(p);

tableView.setItems(misPersonas);

}

```

Lo que se muestra en ejecución es una tabla con un único elemento

