MANUAL TÉCNICO PROYECTO 1 202003654

1. CLASE PRINCIPAL

A. En Principal.java se encuentra la clase principal del proyecto que invoca a App.java, debido a que por conflictos de la versión de java junto con Maven al momento de ser buildeado el proyecto no reconoce una clase principal valida el .jar

2. CLASE APP

```
public class App extends Application {

    3 usages
    private static Scene scene;
    23 usages
    private static Teacher[] teachers = new Teacher[100];
    9 usages
    private static int teacherCount = 0;
    23 usages
    private static Course[] courses = new Course[100];
    9 usages
    private static int courseCount = 0;
    2 usages
    private static Teacher currentTeacherSession;
    3 usages
    private static Course[] teacherSessionCourses = new Course[100];
```

A. La clase App orquesta toda la información necesaria para el funcionamiento correcto durante el ciclo de vida completo del programa,

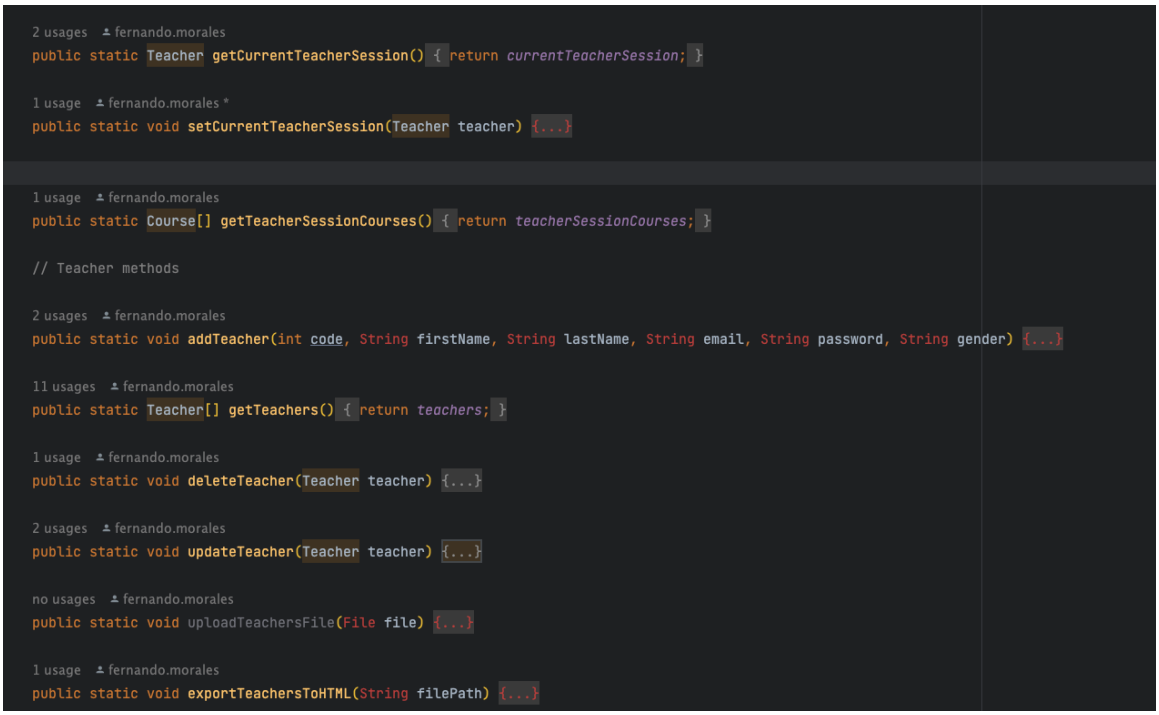
i. Scene proveerá la vista a mostrar.
ii. Teacher almacena la información de todos los maestros agregados.
iii. El contador de maestros y cursos para la generación de los códigos de los maestros basados en cuantos hay.
iv. Courses almacena los cursos agregados.
v. currentTeacherSession guardará al maestro que esté en la sesión activa.
vi. teacherSessionCourses los cursos que el maestro tenga asignados con base a su código al iniciar sesión

```
fernando.morales
@Override
public void start(Stage stage) throws IOException {...}

33 usages  fernando.morales
static void setRoot(String fxml) throws IOException {...}

2 usages  fernando.morales
private static Parent loadFXML(String fxml) throws IOException {...}
```

vii. Start Inicializa la aplicación.
viii. setRoot incará cual será la vista a mostrar.
ix. loadFXML cargará la vista.
x. Parent buscará el archivo de la vista solicitada.



xi. getCurrentTeachersession proporcianará la información del profesor que esté logeado

xii. setCurrentTeacherSession buscará al maestro y lo seteará como el maestro de la sesión actual

xiii. addTeacher agregará un nuevo maestro

xiv. getTeachers retornará los maestros ingresados

xv. deleteTeacher eliminará del listado al maestro seleccionado

xvi. updateTeacher actualizará al maestro enviado

xvii. exportTeachersToHTML exporatrá el listado de maestros en un HTML

xviii. addCourse agregará un nuevo curso al listado

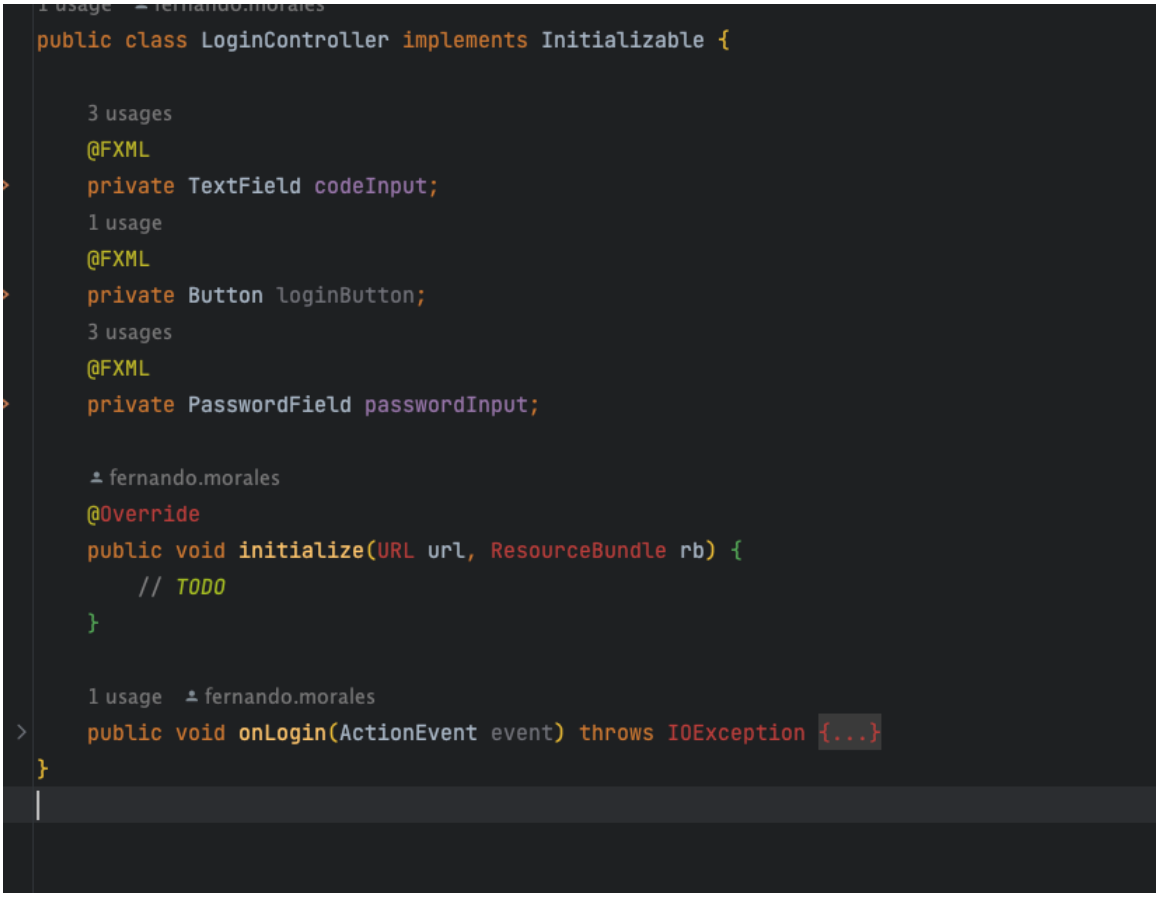
xix. getCourses retornará los cursos

xx. deleteCourse eliminará el curso enviado

xxi. updateCourse actualizará el curso seleccionado

xxii. exportCoursesToHTML exportará el listado de cursos en un html

3. LOGIN CONTROLLER



i. onLogin valida si la sesión es de un admin que la contraseña sea correcta, si no es un admin buscará al profesor y con el código verificará su contraseña y lo guardará como la sesión del profesor actual.

4. ADMIN MENU CONTROLLER

```java
1 usage  fernando.morales
public class AdminMenuController implements Initializable {

    /**
     * Initializes the controller class.
     */
    fernando.morales
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }
    1 usage
    @FXML
    private Button adminCoursesButton;
    1 usage
    @FXML
    private Button adminTeachersButton;
    1 usage
    @FXML
    private Button logoutButton;

    1 usage  fernando.morales
    public void onAdminCourses(ActionEvent event) throws IOException {
        App.setRoot("coursesAdmin");
    }

    1 usage  fernando.morales
    public void onAdminTeachers(ActionEvent event) throws IOException {
        App.setRoot("teachersAdmin");

    }

    fernando.morales
    public void onLogout(ActionEvent event) throws IOException {
        App.setRoot("login");
    }

}
```

A. Administra el menú del administrador

i. onAdminCourses seteará la vista del administrador de cursos

ii. onAdminTeachers seteará la vista del administrador de maestros

iii. onLogout cerrará la sesión del administrador

5. TEACHER MENU CONTORLLER

```
 */
 fernando.morales
@Override
public void initialize(URL url, ResourceBundle rb) {
    // Set up the columns to display the appropriate properties of the Course class
    courseCodeColumn.setCellValueFactory(new PropertyValueFactory<>( s: "code"));
    courseNameColumn.setCellValueFactory(new PropertyValueFactory<>( s: "name"));

    // Get the teacher session courses from the App class
    Course[] teacherSessionCourses = App.getTeacherSessionCourses();

    // Set the items in the TableView
    teacherCoursesTable.getItems().clear(); // Clear existing items

    // Add non-null courses to the TableView
    for (Course course : teacherSessionCourses) {
        if (course != null) {
            teacherCoursesTable.getItems().add(course);
        }
    }

    // Add event handler to the TableView
    teacherCoursesTable.setOnMouseClicked(event -> {
        // Get the selected course
        Course newSelectedCourse = teacherCoursesTable.getSelectionModel().getSelectedItem();

        // Check if a course is selected
        if (newSelectedCourse != null) {
            // Perform your desired action with the selected course
            selectedCourse = newSelectedCourse;
            try {
                App.setRoot("manageCourse");
            } catch (IOException ex) {
                ex.printStackTrace();
            }
        }
    });
}
```

A. Administra el menú del profesor

i. Initialize setea los cursos del profesor y los pinta en la tabla, también espera el evento al seleccionar un curso para entrar al menú del curso y setea el curso seleccionado

```
// Getter method to access the selected course from other controllers
1 usage  fernando.morales
public static Course getSelectedCourse() { return selectedCourse; }

fernando.morales
@FXML
public void onLogout(ActionEvent event) throws IOException {
    App.setRoot("login");
}

1 usage  fernando.morales
@FXML
public void onEditProfile(ActionEvent event) throws IOException {
    App.setRoot("editTeacherProfile");
}

no usages  fernando.morales
```

ii. getSelectedCourse retornará el curso seleccionado

iii. OnLogout cerrará sesión

iv. onEditProfile enviará a la vista para que el profesor pueda editar su perfil