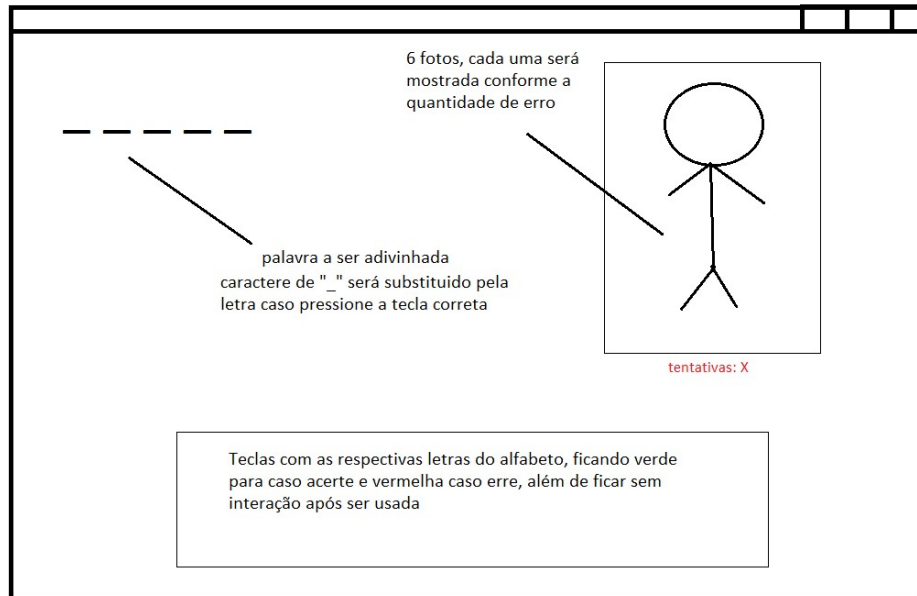
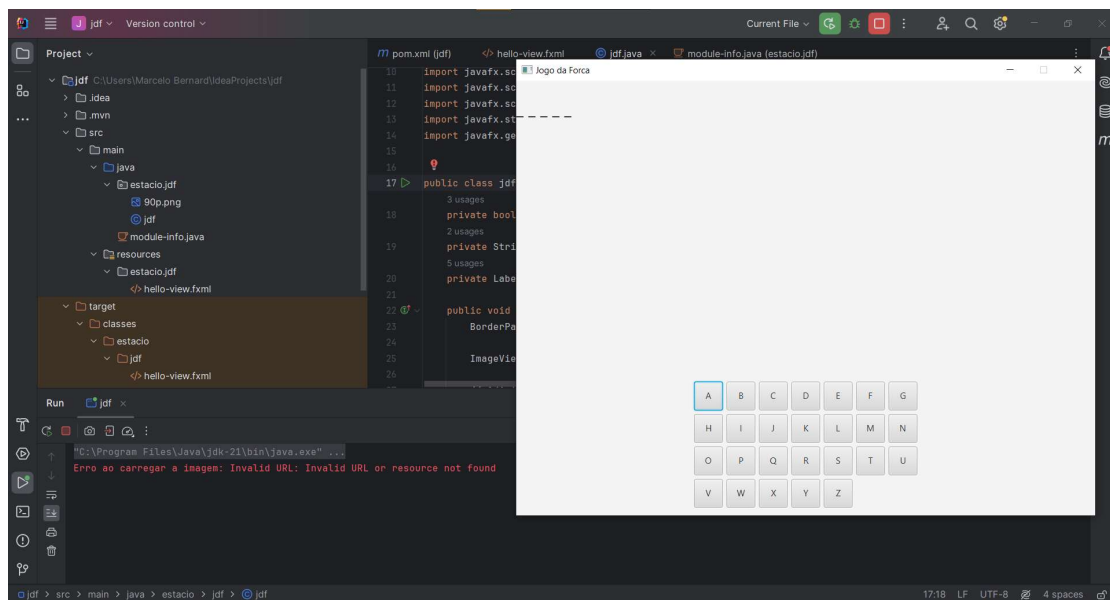


**Alunos:** Marcelo Bernard Vieira da Silva e Philipe Martins Soares

**Planejamento:** Criar um programa com temática de jogo, leve e simples, semelhante a jogos como o Campo Minado



**Código em desenvolvimento:**



Código em execução, com o atual problema de direcionamento do arquivo de imagem sendo capturado pelo try-catch

package estacio.jdf;

```

import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;
import javafx.geometry.Pos;

public class jdf extends Application {
    private boolean respostaCorreta = false;
    private String palavra = "carro";
    private Label palavraLabel;

    public void start(Stage primaryStage) {
        BorderPane root = new BorderPane();

        ImageView imageView = new ImageView();

        // Tenta achar a imagem, estou tendo problemas aqui então criei um try-catch para não
        // atrapalhar o progresso
        try {
            Image image = new Image("90p.png");
            imageView.setImage(image);
            imageView.setFitWidth(300);
            imageView.setFitHeight(400);
        } catch (IllegalArgumentException e) {
            System.err.println("Erro ao carregar a imagem: " + e.getMessage());
        }

        HBox topRightBox = new HBox(imageView);
        root.setTop(topRightBox);
        BorderPane.setAlignment(topRightBox, Pos.TOP_RIGHT);
        BorderPane.setMargin(topRightBox, new Insets(10));

        GridPane buttonPane = new GridPane();
        buttonPane.setPadding(new Insets(10));
        buttonPane.setHgap(5);
        buttonPane.setVgap(5);

        // Cria os botões com o alfabeto
        int col = 0;
        int row = 0;
        for (char c = 'A'; c <= 'Z'; c++) {
            Button button = new Button(String.valueOf(c));

```

```

        button.setPrefSize(40, 40);
        button.setOnAction(e -> {
            if (verificarResposta()) {
                button.setStyle("-fx-background-color: green;");
                button.setDisable(true);
                String letrasCorretas = palavraLabel.getText().replaceAll("\\s", "");
                palavraLabel.setText(getPalavraEscondida(letrasCorretas));
            } else {
                button.setStyle("-fx-background-color: red;");
            }
        });
        buttonPane.add(button, col, row);
        col++;
        if (col > 6) {
            col = 0;
            row++;
        }
    }
}

// Botões na parte inferior e central
HBox bottomButtonBox = new HBox(buttonPane);
bottomButtonBox.setAlignment(Pos.CENTER);
root.setBottom(bottomButtonBox);

// Palavra na parte superior e direita
palavraLabel = new Label(getPalavraEscondida(""));
palavraLabel.setStyle("-fx-font-size: 24px;"); // CSS
HBox bottomRightBox = new HBox(palavraLabel);
root.setRight(bottomRightBox);
BorderPane.setMargin(bottomRightBox, new Insets(0, 10, 10, 0));

Scene scene = new Scene(root, 800, 600);

primaryStage.setScene(scene);
primaryStage.setTitle("Jogo da Forca");
primaryStage.setResizable(false);

primaryStage.show();
}

private boolean verificarResposta() {
    respostaCorreta = !respostaCorreta;
    return respostaCorreta;
}

private String getPalavraEscondida(String letrasCorretas) {
    StringBuilder palavraEscondida = new StringBuilder();
    for (int i = 0; i < palavra.length(); i++) {
        char letra = palavra.charAt(i);
        if (letrasCorretas.contains(String.valueOf(letra))) {

```

```
        palavraEscondida.append(letra).append(" ");
    } else {
        palavraEscondida.append("_ ");
    }
}
return palavraEscondida.toString();
}

public static void main(String[] args) {
    launch(args);
}
}
```