

IFTorrent



Compartilhando Arquivos e Ideias



Problema e Objetivos

- Divulgação para a comunidade;
- Moodle instável, limitado; Tamanho máximo para novos arquivos: 50Mb,
- Ampliar e facilitar a divulgação institucional, assim como a sua manutenção;
- Melhorar o compartilhamento de materiais das aulas;
- Para resolver e atingir os objetivos citados, uma aplicação será desenvolvida: IFTorrent.



Equipe

- 2 bolsistas: Leonardo Bortolini e Gabriel Müller;
- 4 voluntários: Eduardo Toffolo, Guilherme Giordani, Garrenlus de Souza e Otávio Farinon;
- 2 técnicos administrativos: Gustavo Tausendfreund e Eduardo Balbinot;
- 1 docente: Rafael Vieira Coelho;
- 2 ex-membros: Sérgio Brunetta Jr. e Kléber Macieski.



Oportunidades oferecidas

Os discentes envolvidos têm a oportunidade de trabalhar com tecnologias não oferecidas no curso, além de aprenderem metodologias de desenvolvimento em grupo.

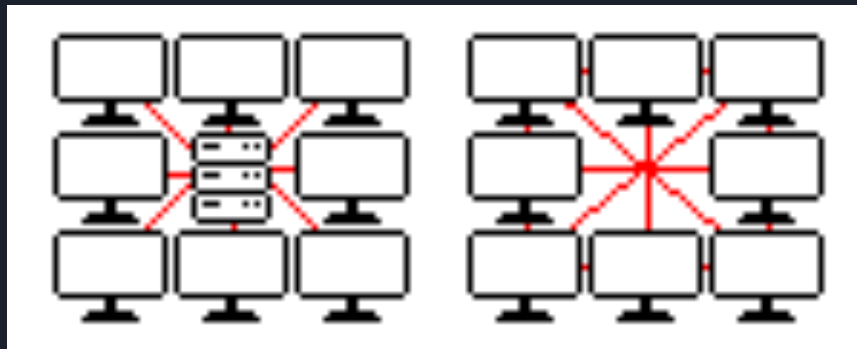


Estudos Realizados

- Java e JavaFX;
- SQL;
- Redes P2P;
- CSS;
- HTML, PHP e CodeIgniter;
- FXML.

P2P (*Peer-to-Peer*)

P2P é uma arquitetura de comunicação de computadores, que foi selecionada por trabalhar de forma desfragmentada, ou seja, sem um servidor principal. Nela, cada membro da rede atua como servidor, baixando e fornecendo arquivos, e não apenas baixando de um local fixo.






Arquivo Torrent

Arquivo criptografado e dividido em pequenos pedaços.

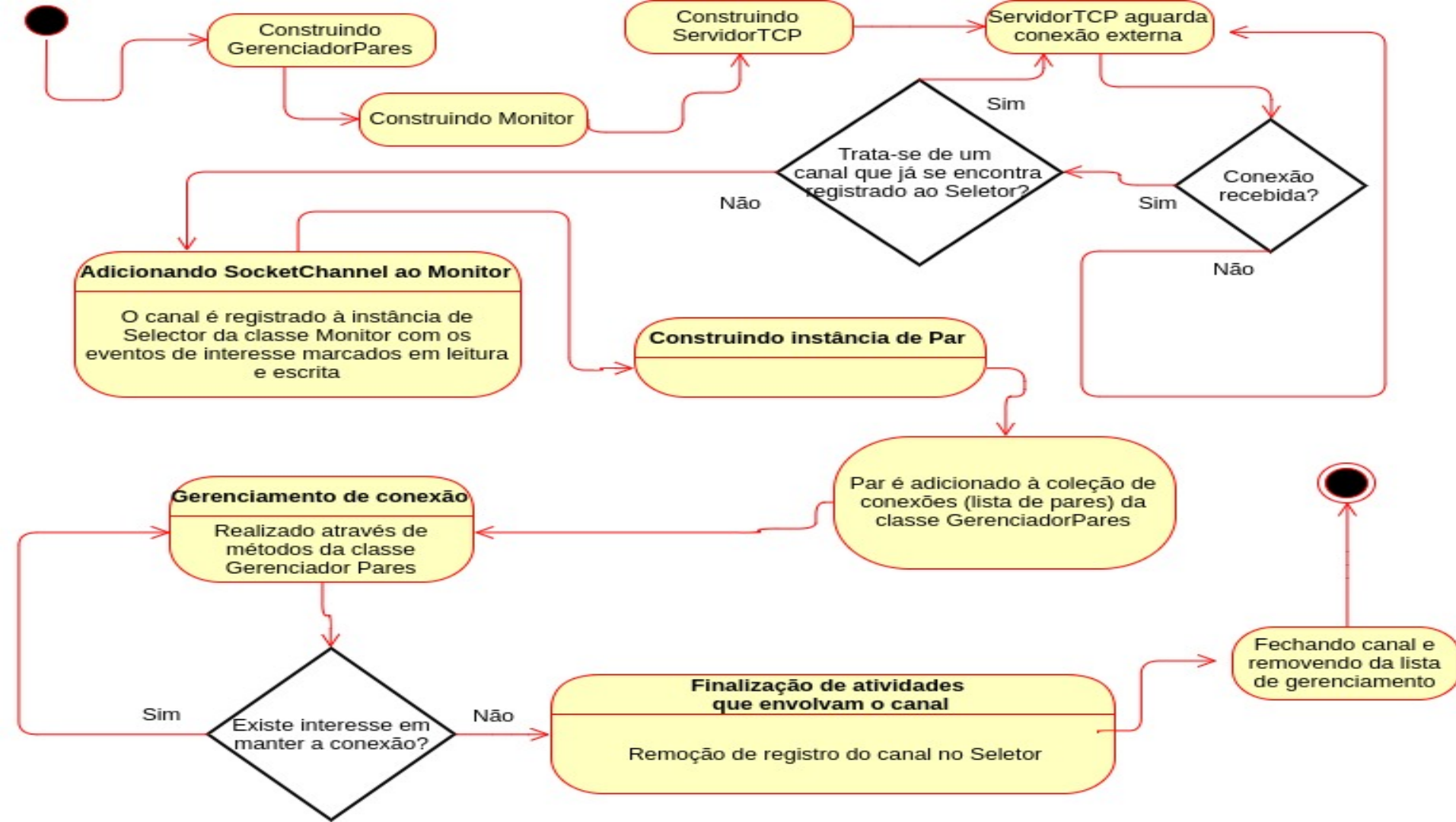
Cada pedaço é enviado de um computador para outro, respeitando o protocolo BitTorrent.

A partir desses conceitos, um mesmo arquivo pode estar sendo baixado de modo em que se receba um pedaço de cada computador.



Elementos do Protocolo BitTorrent

- **Tracker (Rastreador)** -> Responsável por procurar na rede quais computadores possuem o arquivo desejado.
- **Seeder (Semeador)** -> Computador que possui o arquivo desejado o está compartilhando.
- **Leecher (Sanguessuga)** -> Computador que está baixando o arquivo (também pode significar computador que baixou o arquivo mas não o compartilha posteriormente).
- **Peer (Par)** -> Computador que está executando o cliente torrent.



Metodologias Ágeis

1	Responsável	Tarefa	Tipo	Nível	Tempo	Status	Observação
12	Garren	Estudar o uso de DHT	Enhancement	Leve	3	Não iniciada	
13	Garren	Implementação conexao com par (in\out)	Feature	Moderado		Não iniciada	
14	Garren	Implementação ConectorTracker	Feature	Moderado		Não iniciada	
15	Garren	Implementação Fila de Requisições	Feature	Moderado		Não iniciada	
16	Garren	Implementação GerenciadorPedacos	Feature	Moderado		Não iniciada	
17	Leonardo	Azureus (PeerManager)	Planning	Moderado	4	Não iniciada	
18	Leonardo	Fazer janela da interface simples	Feature	Moderado	3	Em andamento	
19	Leonardo	Fazer janela de configurações gerais	Feature	Critico	4	Em andamento	
20	Leonardo	Dao PalavraChave	Feature	Moderado	2	Não iniciada	
21	Leonardo	Wiki: JavaFX	Planning	Moderado	1	Não iniciada	
22	Leonardo	Logotipo do Wiki	Feature	Leve	2	Não iniciada	
23	Leonardo	Slides para o Pense (IFTorrent)	Planning	Critico	2	Não iniciada	
24	Leonardo	Estudo e apresentação sobre o Framework JRebirth (http://www.jrebirth.org)	Planning	Critico	4	Não iniciada	
25	Leonardo	Relatar um erro (menu Ajuda)	Feature	Moderado	2	Não iniciada	
26	Otávio	Estudar biblioteca para usar arquivos Torrents	Planning	Moderado	3	Não iniciada	

Padronização

Para padronizar os códigos criados um documento foi feito com as normas estabelecidas.

1- Para criação de métodos, sempre colocar chaves na linha inicial do método, e não na próxima.

Exemplo correto:

```
17 public static void main(String[] args) {  
18  
19 }
```

Exemplo incorreto:

```
17 public static void main(String[] args)  
18 {  
19  
20 }
```


2- Sempre utilizar nomes significativos para declaração de variáveis, métodos, classes e pacotes.

Exemplo correto:

```
17 public static void main(String[] args) {  
18     int idade do usuario = 18;  
19     String nome do usuario = "Rafael Vieira Lebre";  
20 }
```

Exemplo incorreto:

```
17 public static void main(String[] args) {  
18     int x = 18;  
19     String y = "Exemplo Errado."  
20 }
```



Revisões de Código

A cada semana uma dupla de integrantes revisa os códigos submetidos ao SVN.

Caso a quantidade de novos conteúdos seja pequena, a revisão é adiada.

Semana	Revisor 1	Revisor 2
1	Coelho	Cléber
2	Sergio	Gabriel
3	Guilherme	Leonardo
4	Guilherme	Toffolo
5	Garren	Gabriel
6	Garren	Sergio
7	Coelho	Toffolo
8	Sérgio	Leonardo
9	Guilherme	Gabriel
10	Toffolo	Otávio
11	Otávio	Garren
12	Leonardo	Toffolo
13	Coelho	Gabriel
14	Sérgio	Leonardo
15	Gabriel	Toffolo
16	Garren	Guilherme
17	Coelho	Leonardo
18	Sérgio	Toffolo
19	Gabriel	Sérgio
20	Gabriel	Guilherme
21	Leonardo	Garren
22	Coelho	Sérgio
23	Leonardo	Toffolo
24	Gabriel	Guilherme
25	Coelho	Toffolo
26	Garren	Sérgio
27	Gabriel	Leonardo

Controle de Erros

Para controlar os bugs encontrados na revisão de código, um software chamado *Bugzilla* é utilizado.

2	TestProd	TestComp	nejjoleo	VERI	FIXE	Erros na classe Lista	2017-06-04
4	TestProd	TestComp	Guilherme_Giordani	VERI	FIXE	Erros na classe Mapa	2017-06-08

Rafael Coelho 2017-05-30 09:44:19 -03

[Description](#)

Created [attachment 2 \[details\]](#)

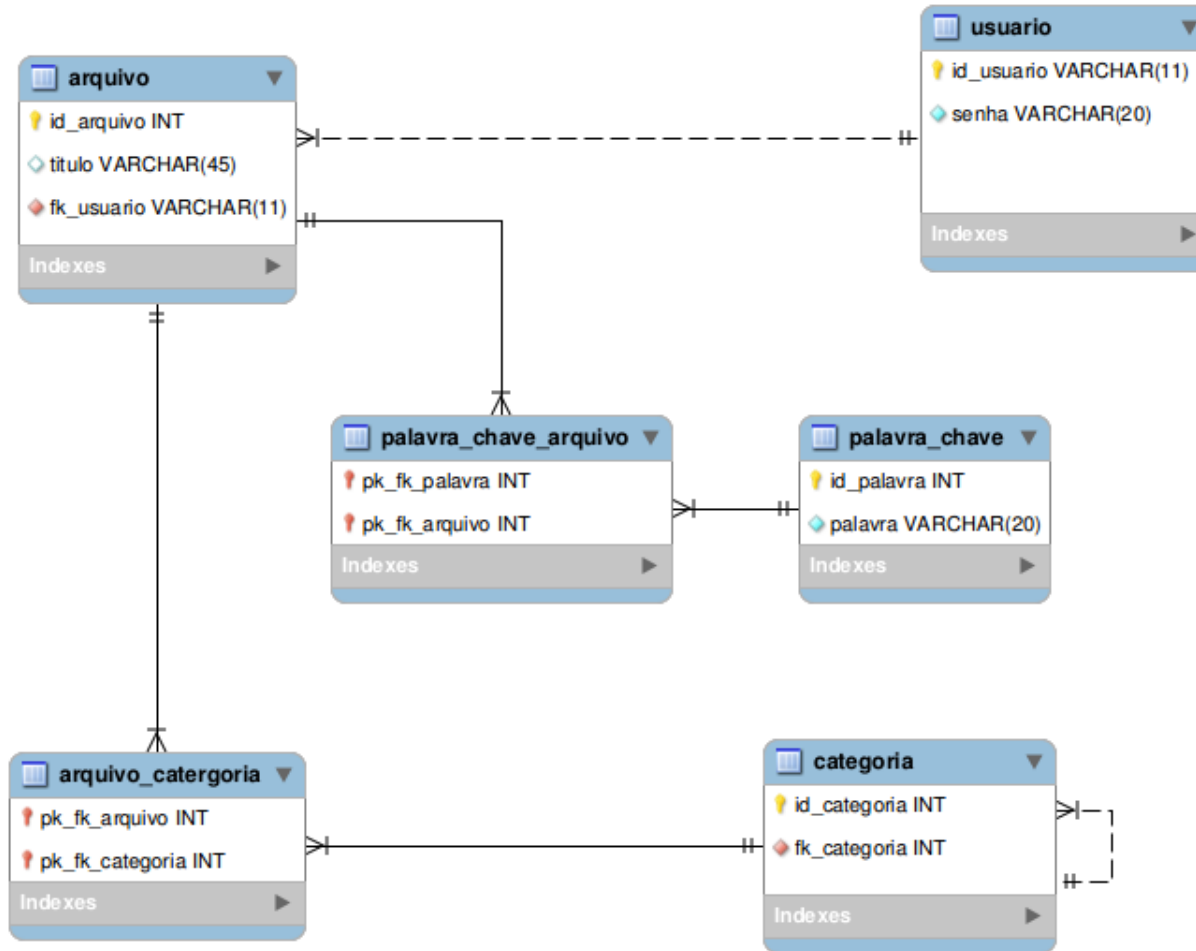
Lista de Padrões de Implementação

- na linha 88, retornar diretamente o que o método remover() retorna:
Ex: return remover();
- o atributo vetor deve ser instanciado no construtor.
- colocar eh_ no início dos métodos que apenas fazem um teste lógico: Regra 2.
- método modificar_objeto nunca retorna false. No final, deve retornar false e se modificou, retornar true: Regra 6.
- modificar nome método transformar_array para para_array: Regra 2.

Resultados obtidos até o momento

		resumo/membros na principal		
223	Gabriel	Sobre a equipe (GUI)	Feature	Leve
224	Gabriel	Criação do Banco de Dados	Feature	Critico
225	Gabriel	Resumo PENSE	Enhancement	Moderado
226	Garren	Implementação Codificador (Bencode)	Feature	Moderado
227	Garren	Implementação Decodificador (Bencode)	Feature	Moderado
228	Garren	Criar Diagrama de Classes	Planning	Moderado
229	Garren	Implementação Requisicao	Feature	Moderado
230	Guilherme	Documentação do CSS no Wiki	Planning	Moderado
231	Guilherme	Documentação do FXML no Wiki	Planning	Moderado
232	Leonardo	Resumo PENSE	Enhancement	Moderado
233	Otávio	Revisão de código (quarta de manhã)	Bug	Moderado
234	Otávio	Estudar Framework Codelgniter (falar com Gabriel)	Planning	Critico
235	Toffolo	Adicionar tags na barra de pesquisa	Feature	Moderado
236	Toffolo	Ir para a pagina de ajuda	Enhancement	Leve
237	Toffolo	Fazer imagens para logoff	Enhancement	Leve
238	Toffolo	Fazer reedição do resumo para o PEnSE	Enhancement	Moderado
239	Toffolo	Revisão de código (quarta de manhã)	Bug	Moderado
240	Toffolo	Verificar bugs	Bug	Leve

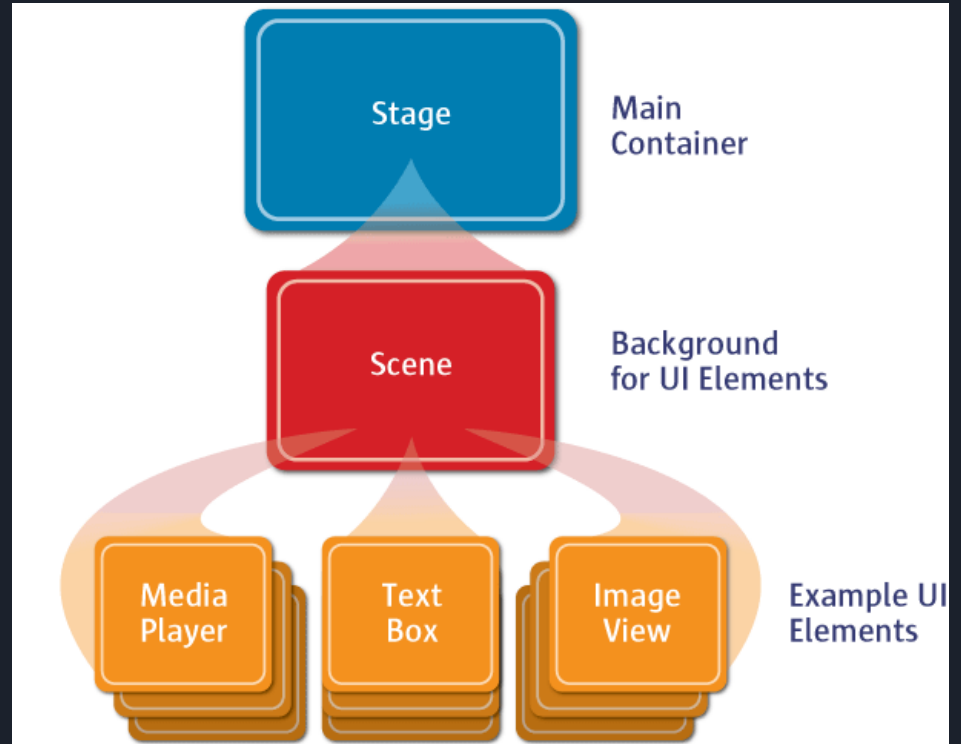
O Banco de Dados



JavaFX

É a nova biblioteca gráfica da Oracle

Possui a seguinte hierarquia de componentes:





Código do Exemplo Mostrado

```
public class JavaFXApplication1 extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        Button btn = new Button();  
        btn.setText("Hello World");  
        btn.setOnAction((ActionEvent event) -> {  
            System.out.println("Hello World!");  
        });  
        StackPane root = new StackPane();  
        root.getChildren().add(btn);  
        Scene scene = new Scene(root, 300, 250);  
        primaryStage.setTitle("Hello World!");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```

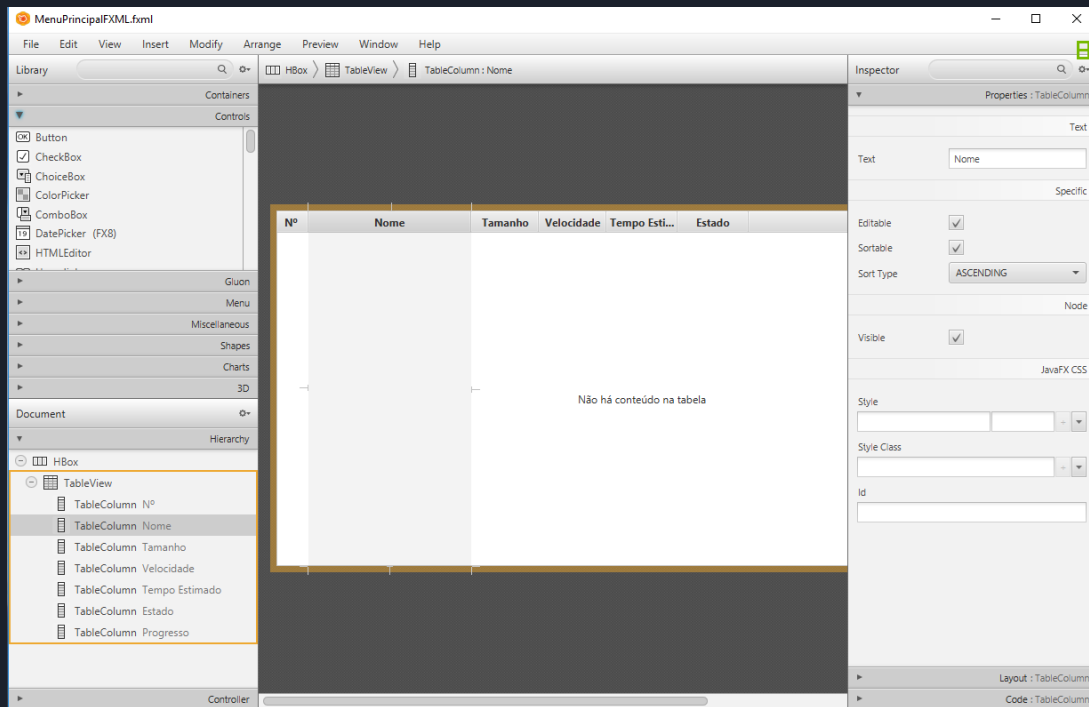


Aplicação FXML do JavaFX

O FXML fornece uma alternativa apropriada para criar gráficos e é ideal para definir a interface do usuário de uma aplicação JavaFX, uma vez que a estrutura hierárquica de um documento XML corresponde fielmente à estrutura do gráfico de cenário do JavaFX.

Gerando Componentes Gráficos

Para gerar os arquivos da parte gráfica da nossa aplicação, a ferramenta *Scene Builder* é utilizada.



Aplicação FXML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import java.lang.*?>
4 <?import java.util.*?>
5 <?import javafx.scene.*?>
6 <?import javafx.scene.control.*?>
7 <?import javafx.scene.layout.*?>
8
9 <HBox stylesheets="/iftorrent/gui/menuPrincipal/MenuPrincipalCSS.css"
10       xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
11       fx:controller="iftorrent.gui.menuPrincipal.MenuPrincipalControlador">
12   <children>
13       <TableView fx:id="tabela">
14           <columns>
15               <TableColumn fx:id="numero" prefWidth="35.0" maxWidth="35.0" text="N° " />
16               <TableColumn fx:id="nome" prefWidth="183.0" text="Nome" />
17               <TableColumn fx:id="tamanho" maxWidth="76.0" text="Tamanho" />
18               <TableColumn fx:id="velocidade" maxWidth="76.0" text="Velocidade" />
19               <TableColumn fx:id="tempo_estimado" maxWidth="120.0" text="Tempo Estimado" />
20               <TableColumn fx:id="estado" maxWidth="282.0" text="Estado" />
21               <TableColumn fx:id="progresso" prefWidth="282.0" maxWidth="282.0" text="Progresso" />
22           </columns>
23       </TableView>
24   </children>
25 </HBox>
```

Controlando o FXML

Cada componente da interface tem seu próprio código FXML e cada um destes possui o seu controlador.

```
16 public class MenuPrincipalControlador implements Initializable {
17     /**
18      * Tabela que será usada para mostrar os itens.
19      */
20     @FXML
21     TableView tabela;
22
23     /**
24      * Método inicialize utilizado para inicializar o componente MenuPrincipal.
25      *
26      * @param url O diretório do objeto root ou null caso o mesmo não seja
27      * encontrado.
28      * @param rb O recurso para localizar o objeto root ou null caso o mesmo não
29      * seja encontrado.
30      */
31     @Override
32     public void initialize(URL url, ResourceBundle rb) {
33         HBox.setHgrow(tabela, Priority.ALWAYS);
34     }
35 }
36
```



Agrupando os FXMLs

Para agrupar os FXMLs, um novo código deste mesmo tipo foi criado.

!-- Pannel que junta e organiza todos os componentes da GUI -->

```
<VBox fx:id="painel1"
  xmlns:fx="http://javafx.com/fxml/1"
  fx:controller="iftorrent.gui.janelaPrincipal.JanelaPrincipalControlador">
  <children>
    <!-- Inclui a barra menu no topo da aplicação -->
    <fx:include fx:id="barra_menu" source="/iftorrent/gui/barraMenu/BarraMenuFXML.fxml" />
    <!-- Pannel para organizar os componentes entre a barra menu e a barra inferior -->
    <HBox fx:id="painel2">
      <children>
        <!-- Inclui a barra lateral no lado esquerdo da aplicação -->
        <fx:include fx:id="barra_lateral" source="/iftorrent/gui/barraLateral/BarraLateralFXML.fxml" />
        <!-- Pannel que organiza os componentes principais da aplicação -->
        <VBox fx:id="painel3">
          <children>
            <!-- Inclui a barra de ferramentas -->
            <fx:include fx:id="barra_de_ferramentas" source="/iftorrent/gui/barraDeFerramentas/BarraDeFerramentasFXML.fxml" />
            <!-- Pannel que organiza a área de exibição dos arquivos torrents e as suas propriedade -->
            <VBox fx:id="painel4">
              <children>
                <!-- Inclui o menu principal -->
                <fx:include fx:id="menu_principal" source="/iftorrent/gui/menuPrincipal/MenuPrincipalFXML.fxml" />
                <!-- Inclui as abas de propriedades referentes a um arquivo torrent -->
                <fx:include fx:id="abas_propriedades" source="/iftorrent/gui/abasPropriedades/AbasPropriedadesFXML.fxml" />
              </children>
            </VBox>
          </children>
        </VBox>
      </children>
    </HBox>
    <!-- Inclui a barra inferior na parte de baixo da aplicação -->
    <fx:include fx:id="barra_inferior" source="/iftorrent/gui/barraInferior/BarraInferiorFXML.fxml" />
  </children>
</VBox>
```

Captura Red

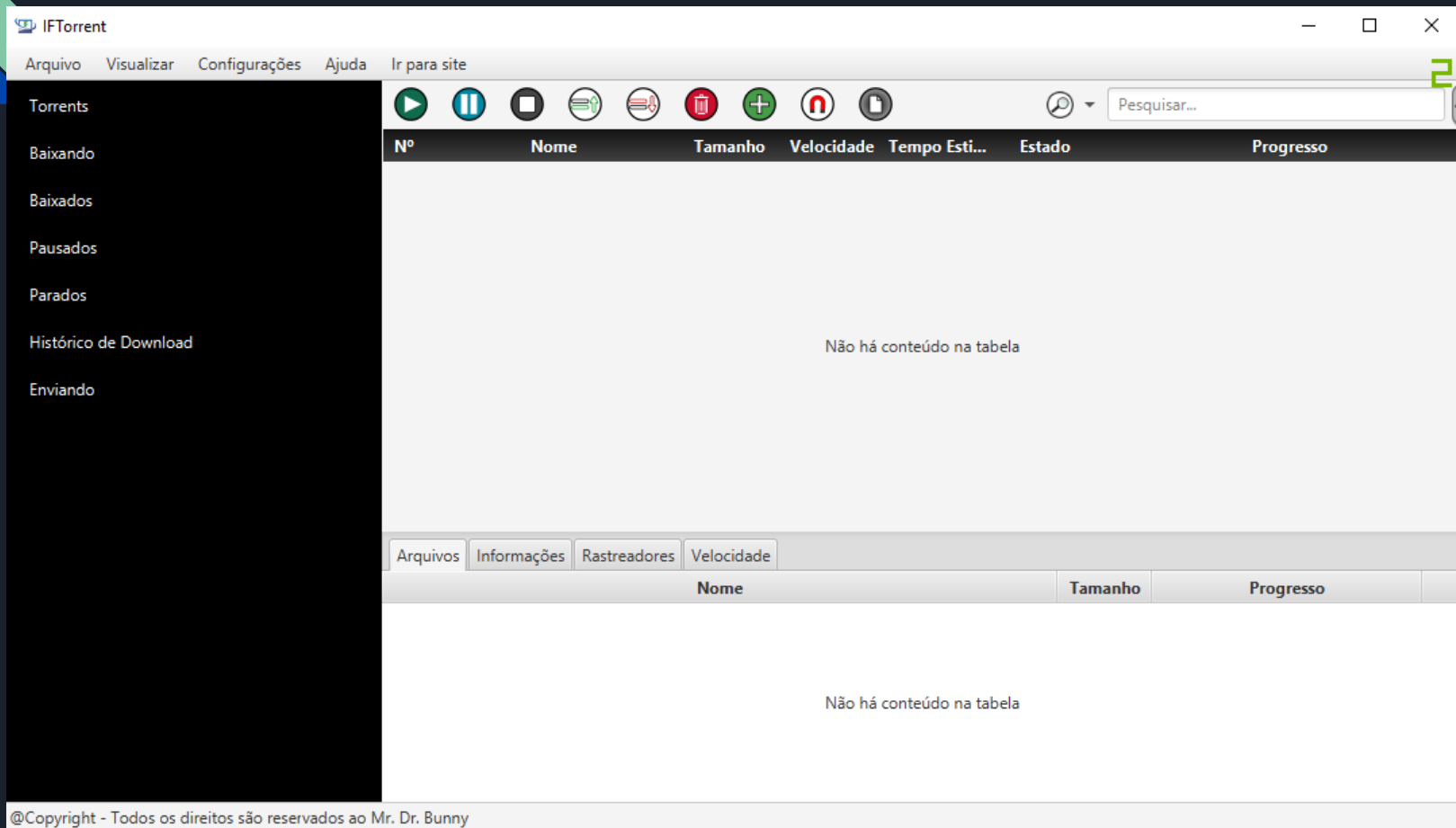


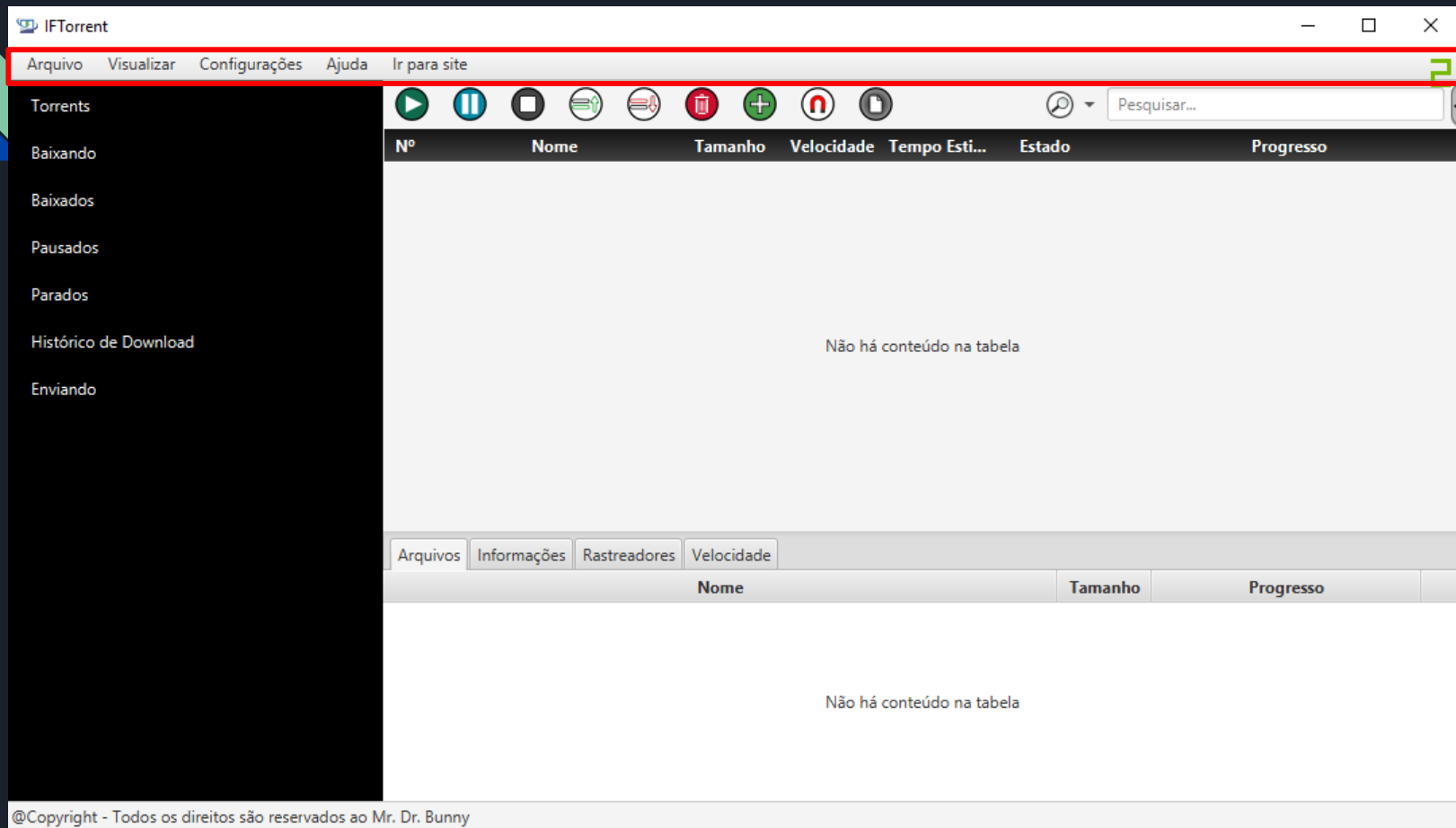
Carregando o FXML principal

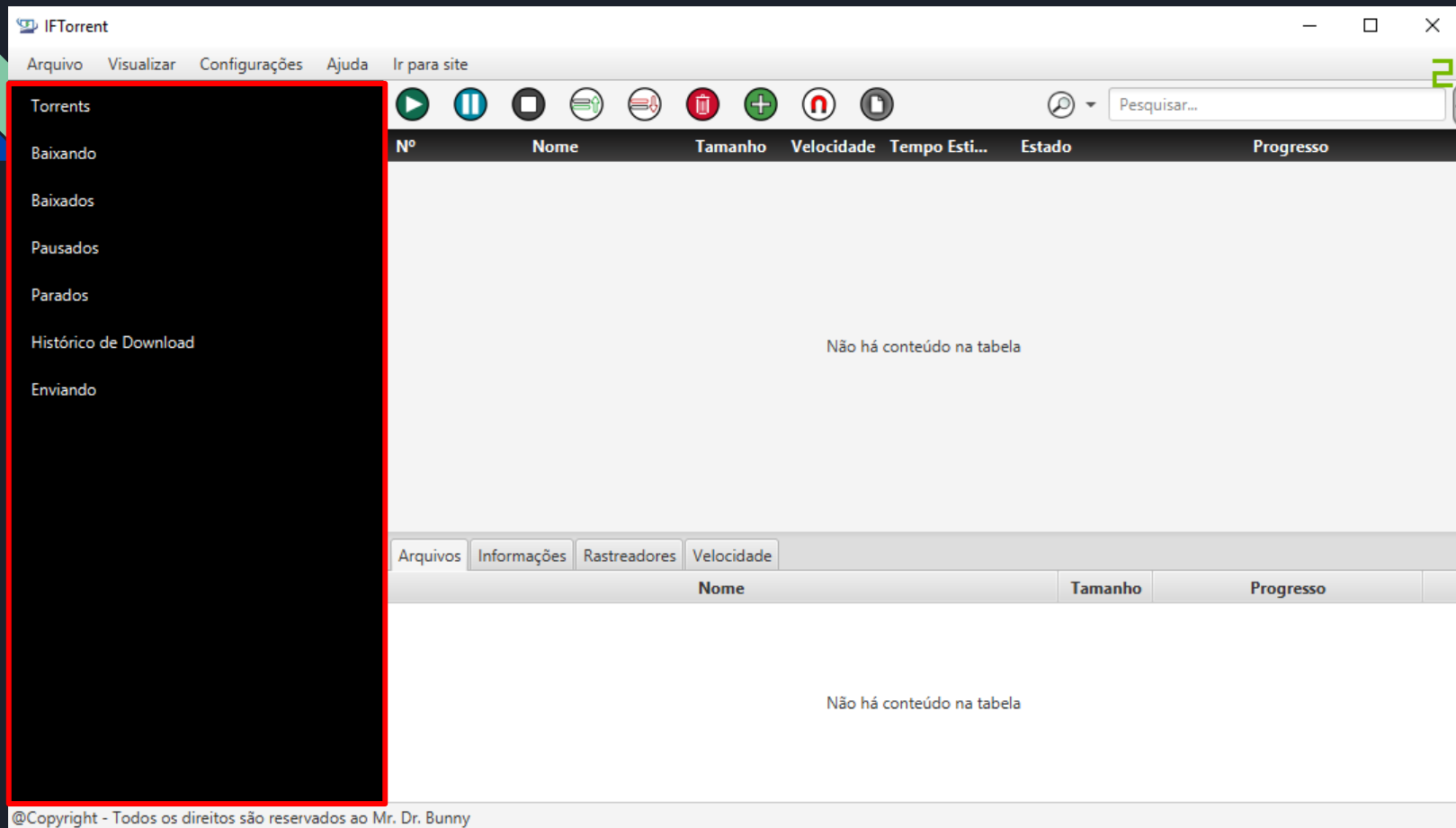
Após os códigos FXML terem sido criados e agrupados, eles precisam ser mostrados.

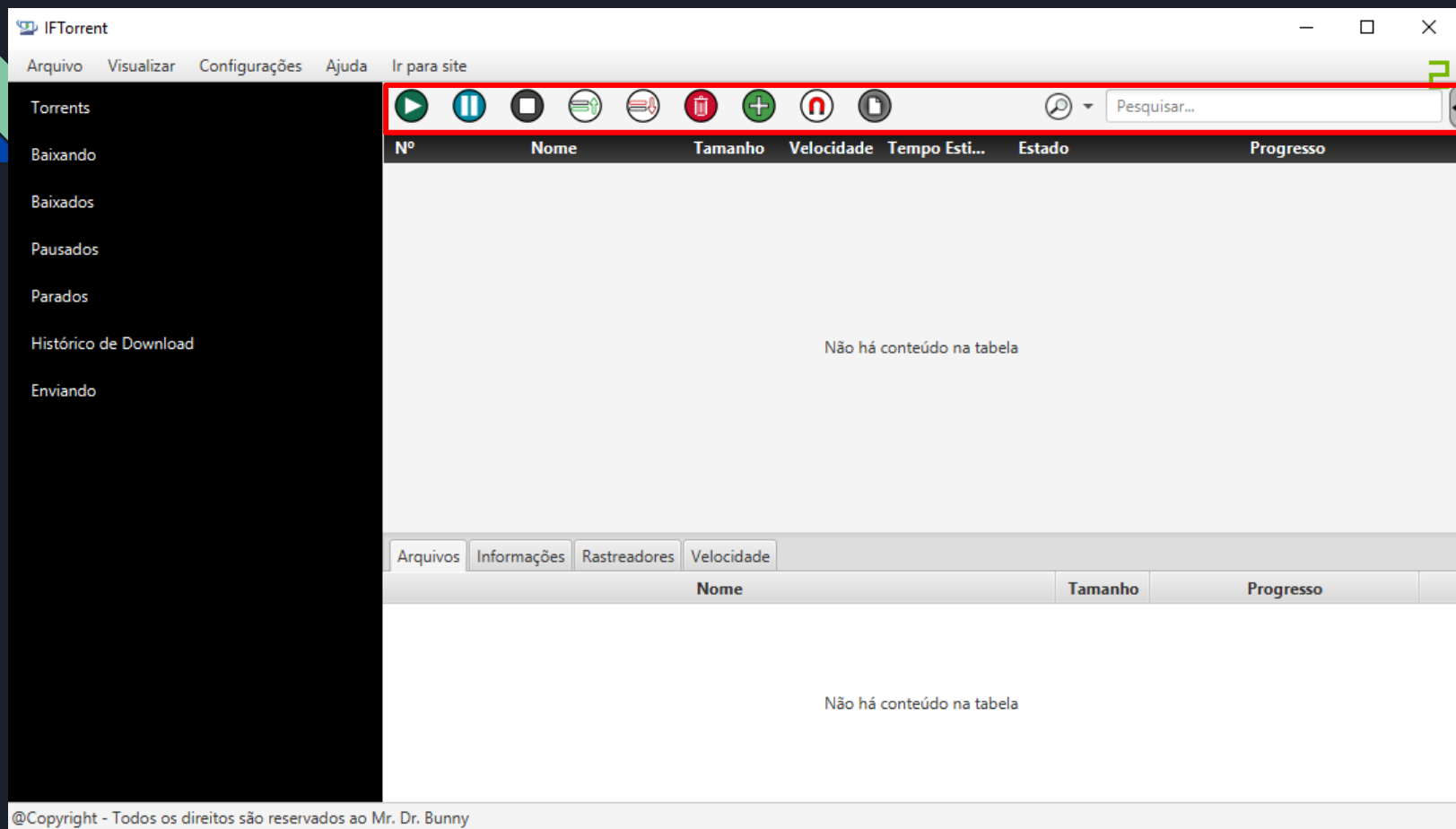
```
18 public class JanelaPrincipalMain extends Application {
19
20     @Override
21     public void start(Stage stage) throws Exception {
22         Comunicador.setMain(this);
23         Parent root = FXMLLoader.load(getClass().getResource("/iftorrent/gui/janelaPrincipal/JanelaPrincipalFXML.fxml"));
24         Scene scene = new Scene(root);
25
26         stage.setTitle(JANELA_TITULO1);
27         stage.getIcons().add(new Image(new File(URL_ICONE1).toURI().toURL().toExternalForm()));
28         stage.setScene(scene);
29         stage.show();
30     }
31
32     /**
33      * @param args the command line arguments
34      */
35     public static void main(String[] args) {
36         launch(args);
37     }
38
39 }
```

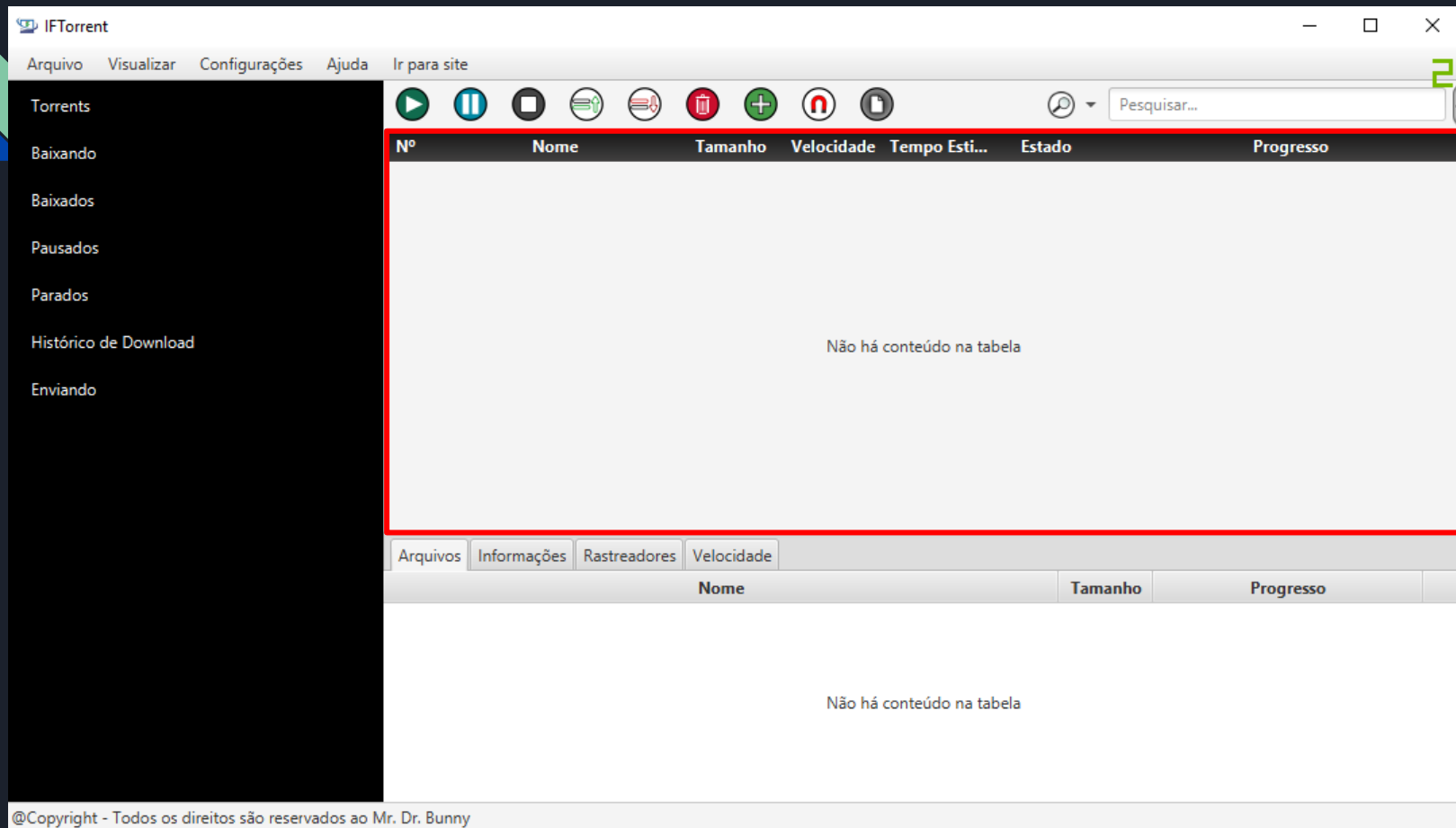
A Interface

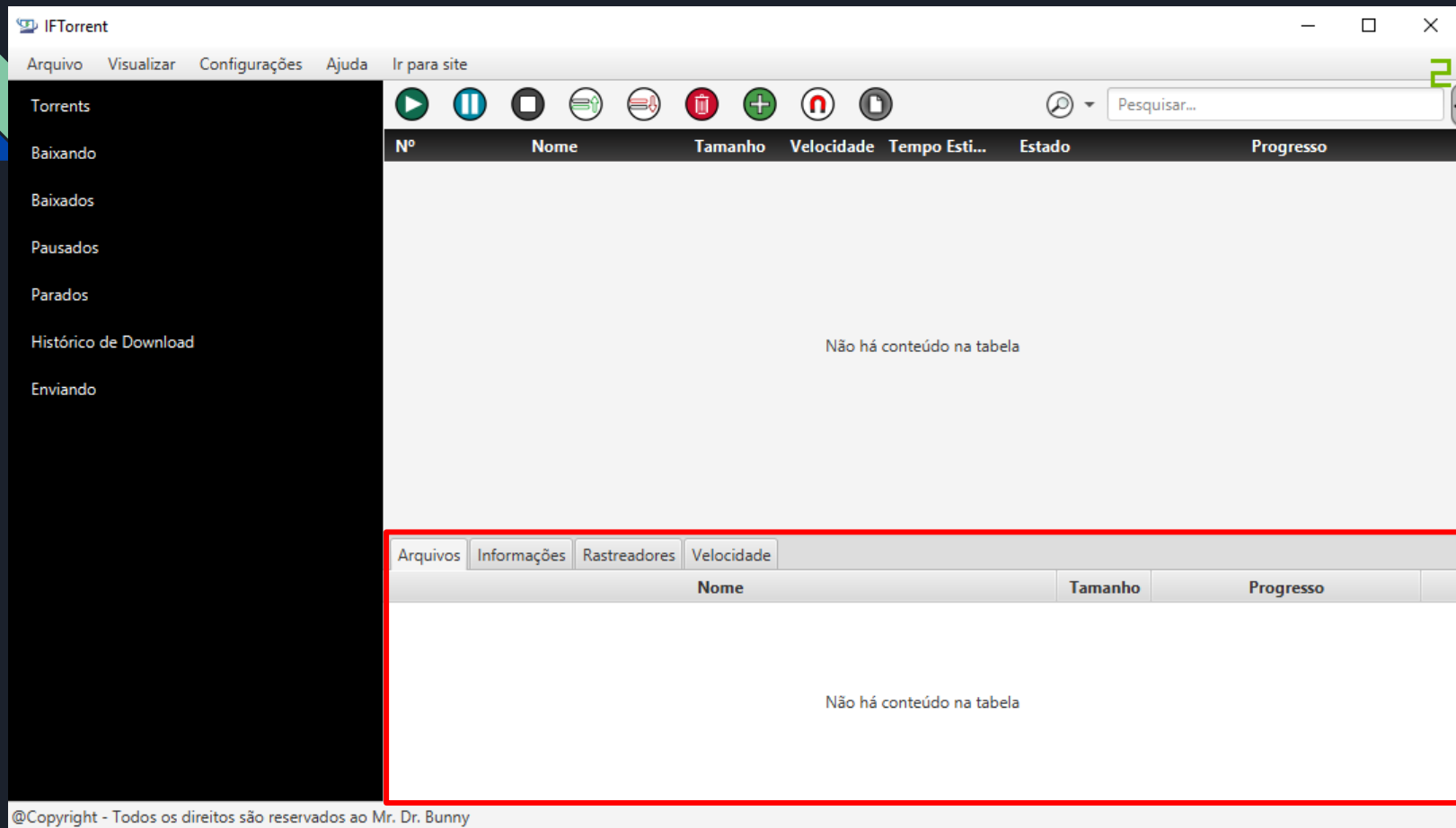












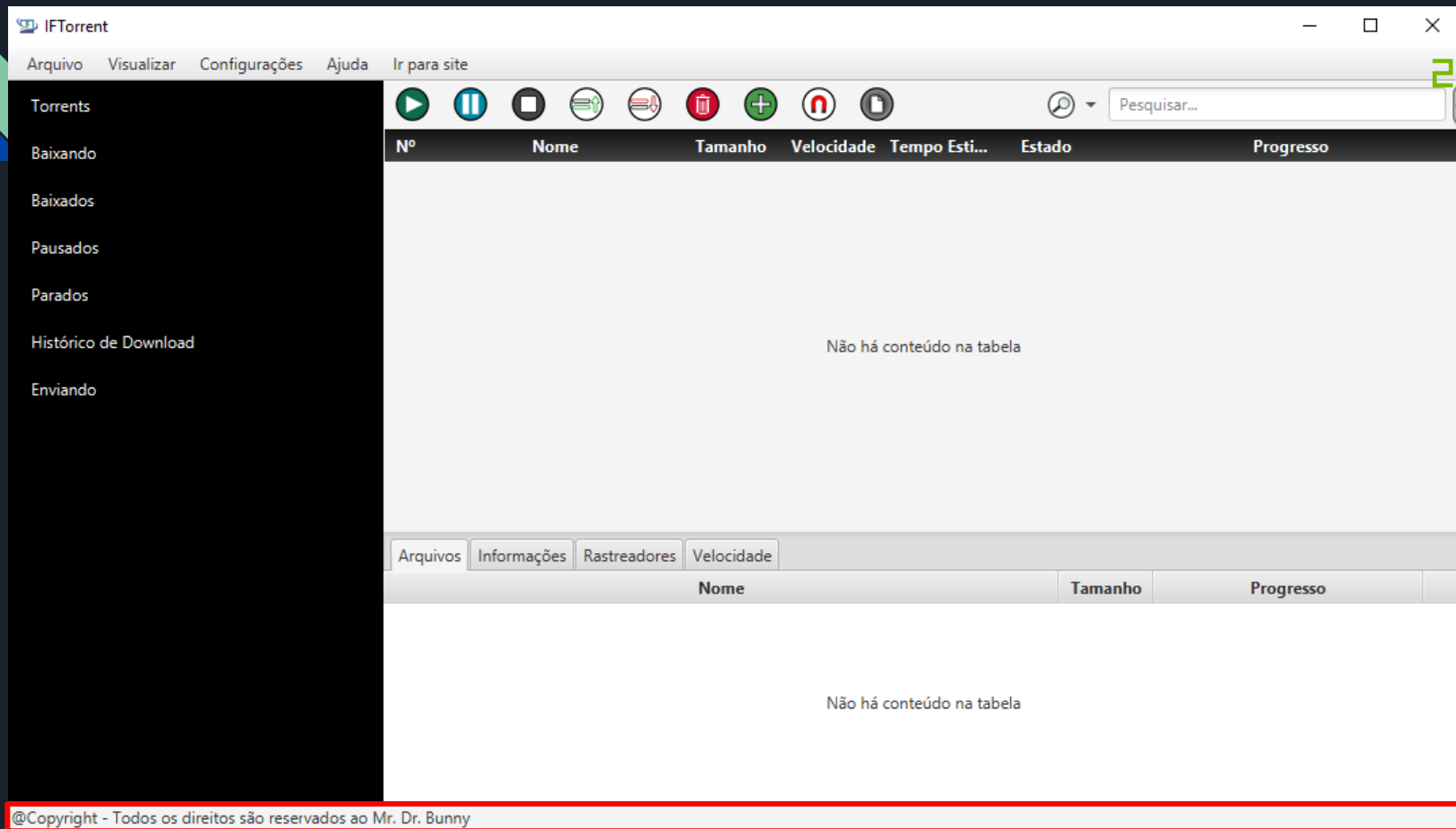
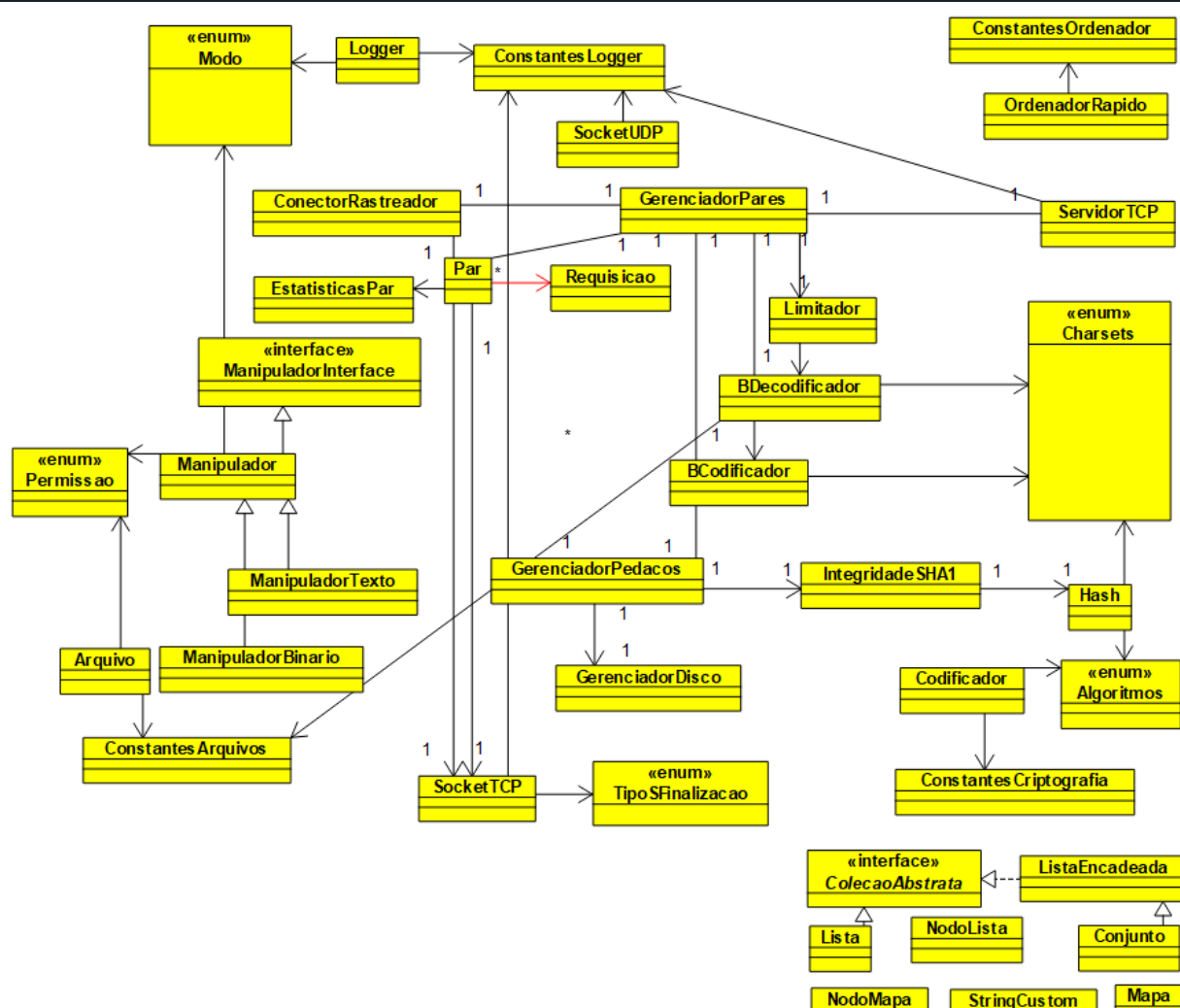


Diagrama de classes.





Atualmente

- Framework JRebirth;
- Maven;
- Biblioteca Bt;
- Wiki.



Obrigado pela atenção

Ficamos agora disponíveis para possíveis questionamentos