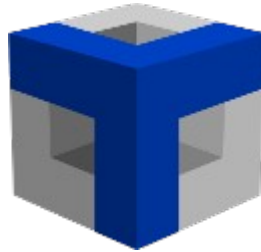


David de Almeida Bezerra Júnior



TETRIS IDE

DESENVOLVIMENTO VISUAL JAVA

Análise de Desenvolvimento e Projeto
Escopo, Diagramas e Descrição de Funcionalidades

Versão 1.3

2020

CONTROLE DE VERSÃO

Data	Versão do TetrisIDE	Alterações
04/05/2020	1.3	Criação do documento.

Índice

1 INTRODUÇÃO.....	1
2 PROJETO.....	2
2.1 Estudo de Ferramentas Concorrentes.....	2
2.2 Escopo.....	3
2.3 Processo de Desenvolvimento de Software.....	5
2.4 Conexão com o Banco de Dados.....	6
2.5 A Organização do Código-fonte Gerado.....	7
2.6 Processo de Verificação de Problemas.....	8
2.7 Apresentação do Sistema.....	9
2.7.1 Barra de ferramentas, Botão Seletor de Janela Principal do Projeto, Configurações e Pannel de Log.....	9
2.7.2 Explorador de Projetos.....	11
2.7.3 Explorador de Janelas.....	11
2.7.4 Gerenciador de Banco de Dados.....	12
2.7.5 Área de trabalho e Visualizador de Código-fonte gerado.....	12
2.7.6 Paleta de Objetos.....	14
2.7.7 Inspetor de Objetos.....	14
2.7.8 Lista de Componentes da Janela.....	15
2.7.9 Componentes Externos.....	16
2.8 Análise de Requisitos.....	16
2.8.1 Requisitos Não-Funcionais.....	17
2.8.2 Requisitos Funcionais.....	17
2.9 Casos de Uso.....	18
2.9.1 Diagrama de Casos de Uso.....	20
2.10 Diagramas de Atividades.....	20
9.5 Descrição dos Casos de Uso.....	21
2.11 Descrição dos Casos de Uso.....	21
9.5.1 UC1 – Iniciar Sistema.....	21
2.11.1 UC1 – Iniciar Sistema.....	21
9.5.1.1 DA1 – Diagrama de Atividades.....	22
2.11.1.1 DA1 – Diagrama de Atividades.....	22
2.11.1.2 Protótipo da Tela.....	23
9.5.2 UC2 – Manter Projeto.....	23
2.11.2 UC2 – Manter Projeto.....	23
9.5.2.1 DA2 – Diagrama de Atividades.....	25
2.11.2.1 DA2 – Diagrama de Atividades.....	25
9.5.2.2 Protótipo da Tela.....	25
2.11.2.2 Protótipo da Tela.....	25
9.5.3 UC3 – Abrir Projeto.....	26
2.11.3 UC3 – Abrir Projeto.....	26
9.5.3.1 DA3 – Diagrama de Atividades.....	26
2.11.3.1 DA3 – Diagrama de Atividades.....	26

2.11.3.2 Protótipo da Tela.....	27
9.5.4 UC4 – Fechar Projeto.....	27
2.11.4 UC4 – Fechar Projeto.....	27
9.5.4.1 DA4 – Diagrama de Atividades.....	28
2.11.4.1 DA4 – Diagrama de Atividades.....	28
9.5.4.2 Protótipo da Tela.....	28
2.11.4.2 Protótipo da Tela.....	28
9.5.5 UC5 – Gerar Código-fonte.....	29
2.11.5 UC5 – Gerar Código-fonte.....	29
9.5.5.1 DA5 – Diagrama de Atividades.....	29
2.11.5.1 DA5 – Diagrama de Atividades.....	29
9.5.6 UC6 – Compilar Código-fonte.....	30
2.11.6 UC6 – Compilar Código-fonte.....	30
9.5.6.1 DA6 – Diagrama de Atividades.....	30
2.11.6.1 DA6 – Diagrama de Atividades.....	30
9.5.7 UC7 – Executar Projeto.....	31
2.11.7 UC7 – Executar Projeto.....	31
9.5.7.1 DA7 – Diagrama de Atividades.....	31
2.11.7.1 DA7 – Diagrama de Atividades.....	31
9.5.7.2 Protótipo da Tela.....	32
2.11.7.2 Protótipo da Tela.....	32
9.5.8 UC8 – Exportar Projeto.....	32
2.11.8 UC8 – Exportar Projeto.....	32
9.5.8.1 DA8 – Diagrama de Atividades.....	33
2.11.8.1 DA8 – Diagrama de Atividades.....	33
9.5.8.2 Protótipo da Tela.....	34
2.11.8.2 Protótipo da Tela.....	34
9.5.9 UC9 – Importar Projeto.....	34
2.11.9 UC9 – Importar Projeto.....	34
9.5.9.1 DA9 – Diagrama de Atividades.....	35
2.11.9.1 DA9 – Diagrama de Atividades.....	35
9.5.9.2 Protótipo da Tela.....	36
2.11.9.2 Protótipo da Tela.....	36
9.5.10 UC10 – Visualizar Janela.....	36
2.11.10 UC10 – Visualizar Janela.....	36
9.5.10.1 DA10 – Diagrama de Atividades.....	37
2.11.10.1 DA10 – Diagrama de Atividades.....	37
9.5.10.2 Protótipo da Tela.....	37
2.11.10.2 Protótipo da Tela.....	37
2.11.11 UC11 – Manter Janela.....	37
9.5.11.1 DA11 – Diagrama de Atividades.....	38
2.11.11.1 DA11 – Diagrama de Atividades.....	38
9.5.12 UC12 – Manter componentes na Janela.....	39
2.11.12 UC12 – Manter componentes na Janela.....	39
9.5.12.1 DA12 – Diagrama de Atividades.....	40
2.11.12.1 DA12 – Diagrama de Atividades.....	40
9.5.12.2 Protótipo da Tela.....	40
2.11.12.2 Protótipo da Tela.....	40
9.5.13 UC13 – Manter Propriedades e Eventos do componente.....	41
2.11.13 UC13 – Manter Propriedades e Eventos do componente.....	41

9.5.13.1 DA13 – Diagrama de Atividades.....	42
2.11.13.1 DA13 – Diagrama de Atividades.....	42
9.5.13.2 Protótipo da Tela.....	42
2.11.13.2 Protótipo da Tela.....	42
9.5.14 UC14 – Manter Banco de Dados.....	43
2.11.14 UC14 – Manter Banco de Dados.....	43
9.5.14.1 DA14 – Manter Banco de Dados.....	44
2.11.14.1 DA14 – Manter Banco de Dados.....	44
9.5.14.2 Protótipo da Tela.....	45
2.11.14.2 Protótipo da Tela.....	45
2.11.15 UC15 – Adicionar Componente.....	45
9.5.15.1 DA15 – Diagrama de Atividades.....	46
2.11.15.1 DA15 – Diagrama de Atividades.....	46
9.5.15.2 Protótipo da Tela.....	47
2.11.15.2 Protótipo da Tela.....	47
2.11.16 UC16 – Selecionar Janela Principal.....	47
2.11.16.1 DA16 – Diagrama de Atividades.....	48
2.11.16.2 Protótipo da Tela.....	48
2.11.17 UC17 – Visualizar Código-Fonte.....	49
2.11.17.1 DA17 – Diagrama de Atividades.....	49
2.11.17.2 Protótipo da Tela.....	50
2.11.18 UC18 – Visualizar Código-Fonte.....	50
2.11.19 UC19 – Gerar Janela a partir de Banco de Dados.....	50
2.11.19.1 DA19 – Diagrama de Atividades.....	52
2.11.19.2 Protótipo da Tela.....	52
2.11.20 UC20 – Clonar Janela.....	53
2.11.20.1 DA20 – Diagrama de Atividades.....	54
2.11.20.2 Protótipo da Tela.....	54
2.11.21 UC21 – Espelhar Banco de Dados.....	55
2.11.21.1 DA21 – Diagrama de Atividades.....	56
2.11.21.2 Protótipo da Tela.....	56
2.11.22 UC22 – Configurar sistema.....	57
2.11.22.1 DA22 – Diagrama de Atividades.....	57
2.11.22.2 Protótipo da Tela.....	58
2.11.23 UC23 – Configurar Banco de Dados.....	58
2.11.23.1 DA23 – Diagrama de Atividades.....	59
2.11.23.2 Protótipo da Tela.....	59
2.12 Diagrama de Classes.....	60

1 INTRODUÇÃO

O TetrisIDE é uma ferramenta RAD (Rapid Application Development) que proporciona um desenvolvimento de aplicações Java para desktop sem codificação manual, suportando completa integração com o Sistema Gerenciador de Banco de Dados MySQL. Com esta simples e elegante IDE, você pode ser 80% mais rápido do que em um desenvolvimento convencional (utilizando um editor com codificação manual), criando sua solução de software.

O TetrisIDE foi criado por David de Almeida Bezerra Júnior buscando reduzir tempo e custo no desenvolvimento de software. Esta tarefa foi feita através da redução da carga de codificação manual que, na programação Java, é mais lenta que em outras linguagens, como Delphi/Lazarus/Object Pascal.

Este documento objetiva-se em apoiar o desenvolvedor no processo de desenvolvimento da ferramenta, através de descrições, ilustrações e diagramas UML.

O TetrisIDE convém ser produzido na linguagem Java e para Java.

2 PROJETO

Nesta sessão serão abordados tópicos que comporão o projeto do software ferramenta.

2.1 Estudo de Ferramentas Concorrentes

Desde a criação do conceito VPL (Visual Programming Language), diversas ferramentas surgiram, buscando a implementação deste diferente modelo de desenvolvimento de softwares, porém nem todas elas se difundiram tão bem na comunidade de desenvolvedores quanto o GeneXus e o Softwell Maker.

A primeira inclui uma forma de produção de aplicações baseada em conhecimento de negócio, gerando sistemas corporativos com uma transparência muito bem-feita entre o desenvolvedor e as funções automaticamente implementadas pela IDE, que através de uma linguagem declarativa e visual, consegue transformar a especificação não normalizada do usuário desenvolvedor em um schema bem normalizado num SGDB.

No GeneXus é possível gerar aplicações Web, para Microsoft Windows e softwares para plataformas móveis (Android, iOS e Blackberry).

O Maker, solução da Softwell, na verdade surgiu em 2002 dentro da Freire Informática Ltda., uma software house sediada na Bahia que atua na área pública e desenvolveu mais de 100 soluções baseadas na plataforma criada. Em 2004 a Softwell Solutions Ltda. foi criada para levar a frente a evolução da ferramenta. Ainda no Maker, quando o usuário necessita desenvolver uma função/operação, esta é representada em um fluxograma, que é chamado de linguagem em dataflow.

Todas as duas são excelentes soluções, porém um aspecto que ambas têm em comum, e que o TetrísIDE vem sanar, é o fato de que quando o programador necessita utilizar algo que o sistema não possui, ele mesmo tem de programar, seja em linguagem escrita ou em fluxogramas.

O TetrísIDE possuirá um módulo de adição de componentes

externos, no qual, quando houver a necessidade de utilizar funções diversas não existentes na IDE, o desenvolvedor poderá adicionar novas opções a partir de módulos criados por outros programadores, através de uma linguagem definida. Logo, o desenvolvedor final não necessitaria trabalhar com programação, e sim com uma montagem de software, trabalhando somente na parte formulário e especificação.

O TetrisIDE trabalhará somente em Java para desktops, utilizando a priori o SGDB MySQL Server Community 5.5, porém através deste módulo de adição de componentes será possível oferecer suporte a outros sistemas gerenciadores de banco de dados e suas diversificadas funções.

2.2 Escopo

O TetrisIDE convém em uma ferramenta de desenvolvimento rápido de aplicações (RAD) voltada à programação por manipulação de formulários, extinguindo todo e qualquer trabalho manual para a geração de código-fonte.

O sistema será desenvolvido na linguagem Java e terá o seu foco em computadores desktop, podendo ser executado em qualquer máquina que tenha o JRE (Java Runtime Environment) instalado. Seguindo esta linha, o software produzirá somente software em Java, utilizando o gerenciador de banco de dados MySQL Community, logo será uma aplicação multiplataforma.

Através da interface do programa será possível criar projetos de software e desenvolvê-los de uma forma totalmente gráfica, contendo um *Explorador de Projetos*, *Explorador de Janelas*, *Paleta de Objetos*, *Inspetor de Objetos*, *Gerenciador de Banco de Dados*, *Área de trabalho*, *Visualizador de Código-fonte Gerado*, *Lista de Componentes da Janela*, *Barra de ferramentas*, *Botão Seletor de Janela Principal do Projeto*, *Configurações*, *Painel de Log* e *Gerenciador de Componentes Externos do Projeto*.

No *Explorador de Projetos* estará contida a lista de projetos existentes no espaço de trabalho, podendo abrir, alterar o nome ou excluir o item selecionado, além de executar ou exportar o projeto aberto. Já o *Explorador de Janelas* será uma lista de todas as janelas do sistema (projeto) aberto no *Explorador*

de *Projetos*. Nele será possível abrir, criar, clonar, renomear e excluir janelas (formulários).

Sendo o *Explorador de Janelas* diretamente ligado ao *Explorador de Projetos*, o *Inspetor de Objetos* será o utilitário responsável por controlar as Propriedades e Eventos de cada janela e de seus itens (botões, campos de texto, etc.).

A *Paleta de Objetos* será composta por todos os itens adicionáveis às janelas do software em desenvolvimento. Estes itens serão: rótulos, campos de texto, caixas de combinações, listas, áreas de texto, painel de edição, painéis, barras de ferramentas, painel de abas, imagens, botões, tabelas, campos de checagem, campos de escolha, barras de menus, timers, procedures, variables e relatórios.

Na *Área de Trabalho* será visualizada (graficamente) a janela que estiver em edição no momento. É aqui também onde os componentes serão manipulados e arrastados. Ainda aqui, no ato da edição da janela, será possível mapear diretamente as colunas e tabelas do banco de dados em seus componentes (campos de texto, tabelas, etc).

No *Visualizador de Código-fonte Gerado* o usuário visualizará o código-fonte Java gerado, fruto da edição da janela aberta na *Área de Trabalho*.

No *Gerenciador de Banco de Dados* o usuário poderá manipular as tabelas e campos do schema do banco de dados de forma gráfica, não necessitando de criação/edição manual de schema no DB.

Através da *Lista de Componentes da Janela*, o usuário poderá organizar (ordem de criação dos componentes) ou selecionar componente da janela para edição.

A partir da *Barra de ferramentas* será possível Criar, Salvar, Abrir, Fechar, Executar e Exportar (para código Java) o projeto.

O *Botão Seletor de Janela Principal do Projeto* exibirá uma lista com as janelas do projeto para seleção da janela principal.

O botão *Configurações* exibirá tela de configuração do TetrisIDE, definindo LookAndFeel dos projetos e idioma da aplicação.

No *Painel de Log* exibirá retorno de compilação do projeto aberto.

Pelo Gerenciador de Componentes Externos, será possível adicionar e remover componentes externos (.jar) ao projeto.

2.3 Processo de Desenvolvimento de Software

Através do Tetris, o desenvolvedor trabalhará manipulando formulários e funções visualmente, não necessitando programar nem mesmo pelo modelo de fluxogramas. Assim, a criação se concentrará em arrastar e soltar componentes da *Paleta de Objetos* para a janela na *Área de Trabalho* e modificar propriedades e manipular eventos (visualmente) de componentes pelo *Inspetor de Objetos*, sempre espelhando as modificações com o *Gerenciador de Banco de Dados*, o *Explorador de Janelas* e o *Explorador de Projetos*. Abaixo, na figura 1, temos um exemplo deste processo.

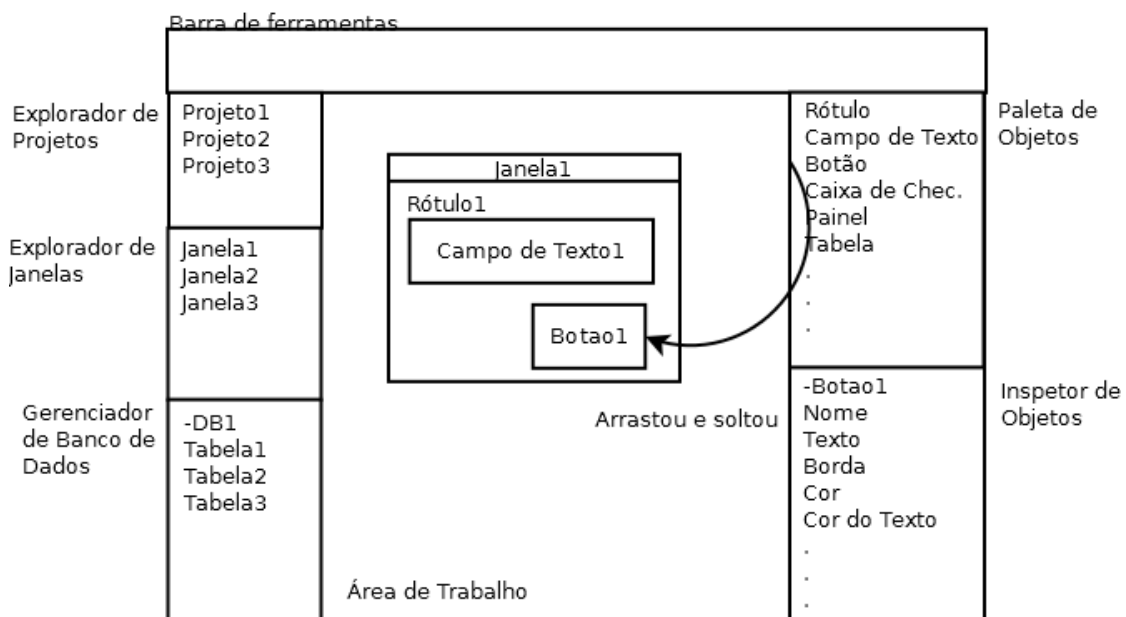


Figura 1- Wireframe da tela principal do TetrisIDE demonstrando a função de arrastar e soltar.

Após a confecção do link das janelas, o desenvolvedor poderá gerar e executar o programa criado a partir do botão *Executar*, presente na *Barra de Ferramentas*, e acessível a partir da tecla *F9*. Sendo executada a função, o programa automaticamente traduzirá toda a criação visual feita pelo programador

em código-fonte Java, organizando todas as classes em seus devidos pacotes, e importando as extensões para dentro do projeto. Logo, o JDK (Java Development Kit) será requisitado para a compilação daquele código agora gerado. Após ter os bytecodes gerados, a classe principal é executada. Depois disso, através da função Exportar Executável Jar, a IDE empacotará todas as classes compiladas e os arquivos necessários em um arquivo “.jar”, que é o considerado executável para a linguagem Java. Segue este processo abaixo na figura II.



Figura II- Processo de geração do programa executável Java, através da compilação e empacotamento do código-fonte gerado a partir das janelas criadas.

2.4 Conexão com o Banco de Dados

Com o sistema gerenciado de banco de dados (SGDB) MySQL Server Community 5.5 será feita a conexão através do driver JDBC, e todo o processo de criação e geração do schema será automático.

Quando criado um novo projeto, haverá uma classe que estruturará o banco de dados junto ao servidor de forma transparente, não sendo perceptível ao usuário programador ao testar o software produzido. Essa transparência é importante, porque a partir dela é que se eliminam mais linguagens no processo de desenvolvimento: o SQL (Structured Query Language), a DDL (Data Definition Language) e a DML (Data Manipulation Language).

Dentro desta linha de pensamento, o usuário não necessitará conhecer normas e normalização de dados em databases, pois o Tetrís cuidará de todos estes detalhes, deixando o foco de raciocínio do programador voltado à lógica de negócio da aplicação a ser construída. Entretanto, ele precisará conhecer sobre tipos de dados, pois o programa e seus campos são definidos através de limites onde, por exemplo, um campo de idade não pode aceitar “JOAO” como valor, ficando restrito somente a números inteiros. A figura III abaixo explica como se procedem os processos atualização do schema para com o banco de dados.

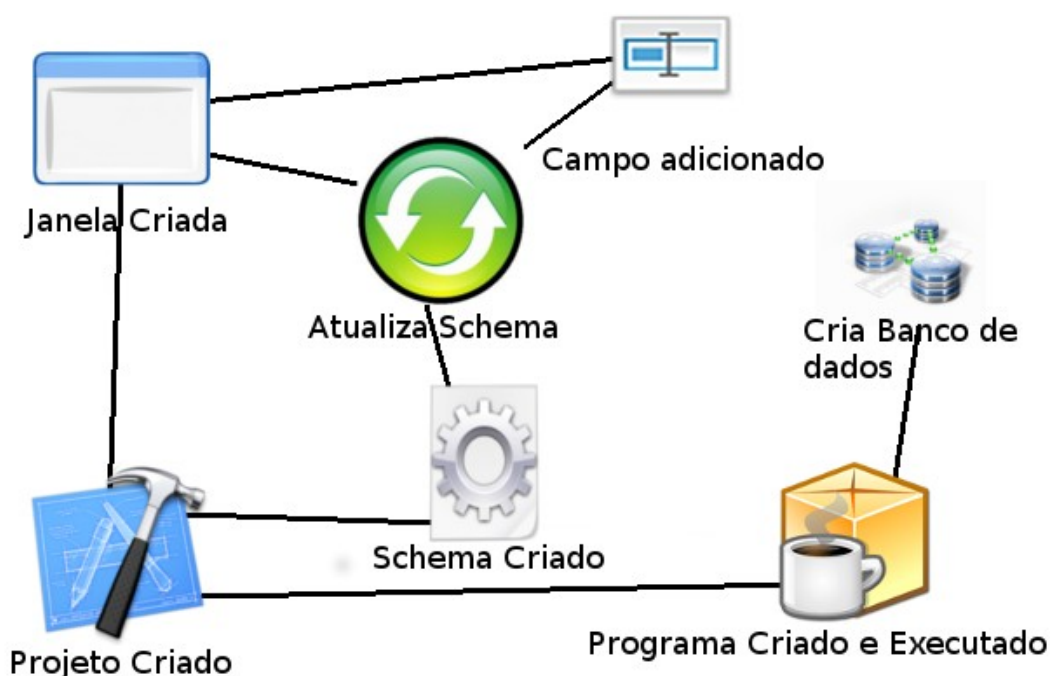


Figura III- Processo de sincronia e atualização do schema do banco de dados feito pelo TetrísIDE ao manipular um projeto.

2.5 A Organização do Código-fonte Gerado

Toda a estrutura das classes e pacotes dos programas produzidos pelo Tetrís obedecerão as normas do modelo de camadas *MVC* (*Model*, *View* e *Control*), sendo que a única diferença será que o nível *Visão* (*View*) e o *Controle* (*Control*) estarão em um só lugar: no próprio *Visão*.

A disposição dos pacotes será simples e organizada, sendo que o

grupo mestre será o *tetris*, seguido do conjunto que levará o nome do projeto, e logo as camadas modelo e visão. Haverá dois grupos separados, um chamado *relatorios*, abrigando todos os elementos essenciais para a execução dos documentos e impressões gerados pelo sistema, e outro chamado *componentes*, este será responsável por abrigar todas as classes auxiliares que acompanharão o Tetris. A tabela 5 demonstra esta estrutura.

Pacote	Internos	
Tetris	Pacote	Internos
	nome do projeto	modelo, visao, res
componentes	visao, modelo, imagem	
relatórios	visao, modelo	

Tabela 1- Organização em pacotes das classes dos projetos gerados pelo TetrisIDE.

2.6 Processo de Verificação de Problemas

Estando muito bem coesa e muito bem apoiada pela manipulação de formulários (em vez de codificação manual), a IDE não necessitará alocar recursos do computador para realizar verificação de sintaxe, concentrando toda a sua atenção no módulo de estúdio visual oferecido. Porém, ao ser lançada uma execução pelo usuário programador, o projeto será compilado pelo javac (programa compilador presente no Java Development Kit), dando um retorno sobre o processo. Recebendo uma resposta, o Tetris mostrará ao programador o sucesso ou o fracasso da operação, exibindo as dicas (fornecidas pelo JDK) sobre o problema ocorrido.

Por parte da ferramenta os problemas para com a compilação não deverão ocorrer, a não ser por passagem de parâmetros incorreta por parte do usuário desenvolvedor, ou por qualquer ação/agente externo, como por exemplo, quando o usuário mover os arquivos necessários pelo projeto, ou simplesmente adicionar um plugin mal estruturado na IDE.

Mais uma vez, o uso da ferramenta proposta consegue remover

mais uma preocupação do processo de desenvolvimento, deixando mais explícito que o foco se concentra nas regras de negócio.

2.7 Apresentação do Sistema

Como já foi dito, o TetrísIDE está disposto em *Explorador de Projetos*, *Explorador de Janelas*, *Paleta de Objetos*, *Inspetor de Objetos*, *Gerenciador de Banco de Dados*, *Área de trabalho*, *Visualizador de Código-fonte Gerado*, *Lista de Componentes da Janela*, *Barra de ferramentas*, *Botão Seletor de Janela Principal do Projeto*, *Configurações*, *Painel de Log* e *Gerenciador de Componentes Externos do Projeto*. Seguindo esta estrutura, nesta sessão serão abordados aspectos mais aprofundados de cada elemento citado.

2.7.1 Barra de ferramentas, Botão Seletor de Janela Principal do Projeto, Configurações e Painel de Log



Figura IV- Barra de ferramentas do TetrísIDE.

O software busca ser o mais intuitivo e minimalista possível. Por isso, ele traz os comandos considerados essenciais em sua barra de ferramentas, extinguindo qualquer opção desnecessária presente nos menus das diversas aplicações existentes. Opções de edição, como copiar, recortar e colar são óbvias para o usuário (utilizando CTRL + C, CTRL + X e CTRL + V para as citadas operações, respectivamente, ou acessando através de popup menus), logo, se fossem inseridas em uma barra de menus ou na própria barra de ferramentas fariam somente volume, rebuscando a usabilidade da IDE.

Visto que todas as opções são autoexplicativas, caso o usuário queira criar um projeto, basta clicar no botão *Novo*. Mas se ele precisar abrir um já criado (selecionado a partir do *Explorador de Projetos*), seria bastante clicar no

botão *Abrir*, ou efetuar dois cliques no *Explorador de Projetos*. Assim o botão *Fechar* fecha o projeto, o *Salvar* salva o progresso, o *Executar* gera o código-fonte, compila e executa o programa, o *Exportar* exporta o Projeto, Código-fonte Java ou Executável Jar para um local indicado e o *Sobre* mostra informações sobre a IDE e opções de ajuda. O botão com uma chave inglesa abre a janela de *Configurações*, e o botão com um terminal mostra o *Painel de Log* de compilação. Você pode escolher a janela principal do projeto através do *Botão Seletor de Janela Principal do Projeto*.



Figura VI- Janela de Configurações.

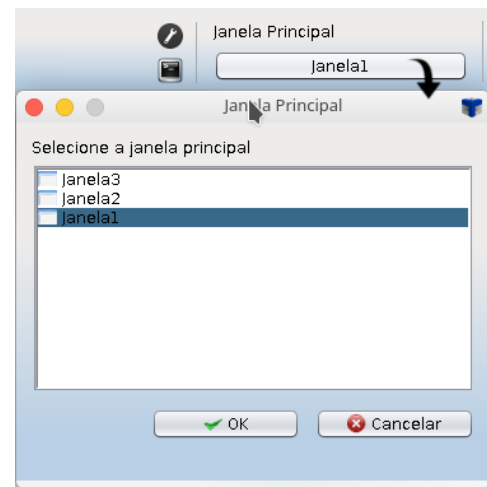


Figura V- Seleção de Janela Principal.

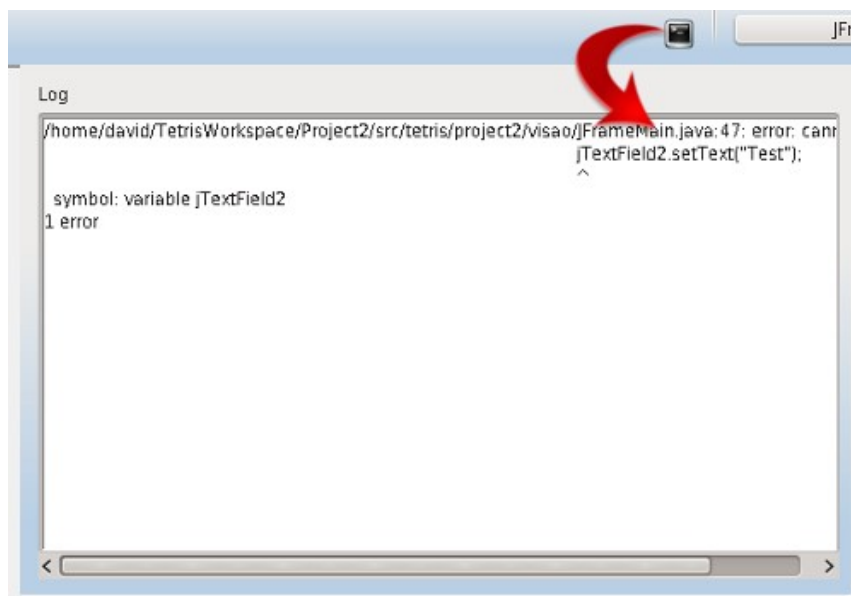


Figura VII- Painel de Log de Compilação.

2.7.2 Explorador de Projetos

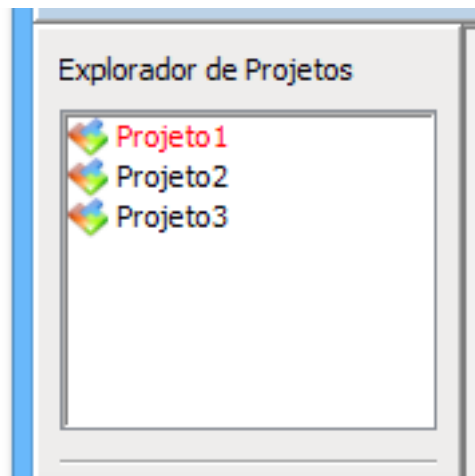


Figura VIII- Explorador de Projetos.

Através desta lista, que se encontra no lado esquerdo da IDE, é possível manipular os projetos criados, todos eles armazenados em um workspace (local de trabalho), que nada mais é do que um diretório com recursos e configurações da ferramenta.

Por aqui é possível abrir, fechar, excluir, importar, exportar e executar projetos.

2.7.3 Explorador de Janelas

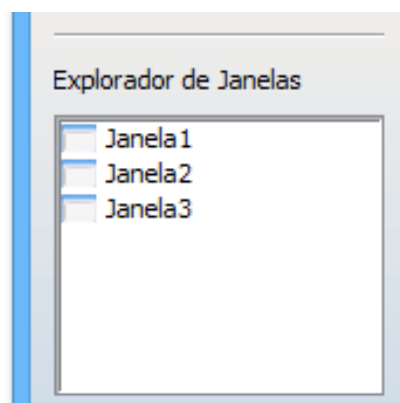


Figura IX- Explorador de Janelas.

Esta lista tem a função de possibilitar a manipulação das janelas do

projeto em desenvolvimento, sendo que o item selecionado automaticamente será o componente que estará em edição na *Área de Trabalho* da ferramenta. Por ela são oferecidas as funções de abrir, criar, renomear, clonar e excluir janelas.

2.7.4 Gerenciador de Banco de Dados



Figura X- Gerenciador de Banco de Dados.

A partir daqui, o usuário programador pode manipular as tabelas do banco de dados do sistema, podendo inserir, modificar e excluir colunas, isso tudo sem precisar escrever uma linha de código SQL, DDL ou DML. Apesar de sua aparência simples, por aqui o desenvolvedor do projeto pode selecionar as colunas que desejar e gerar janelas no projeto prontas para manipulação.

Por aqui também é possível espelhar um banco de dados presente no Sistema Gerenciador de Banco de Dados MySQL, gerando suas tabelas e colunas no projeto.

2.7.5 Área de trabalho e Visualizador de Código-fonte gerado

A Área de trabalho é a mesa do programador, onde ele visualiza e manipula o resultado dos seus esforços.

Aqui é exibida a janela que está em edição atual, mostrando todo o seu conteúdo para a manipulação, onde é possível arrastar e posicionar itens em seu contêiner. Através da Paleta de Objetos, arrasta-se para cá os itens desejados.

O Visualizador de Código-fonte gerado exibe o código Java resultado da manipulação da janela.

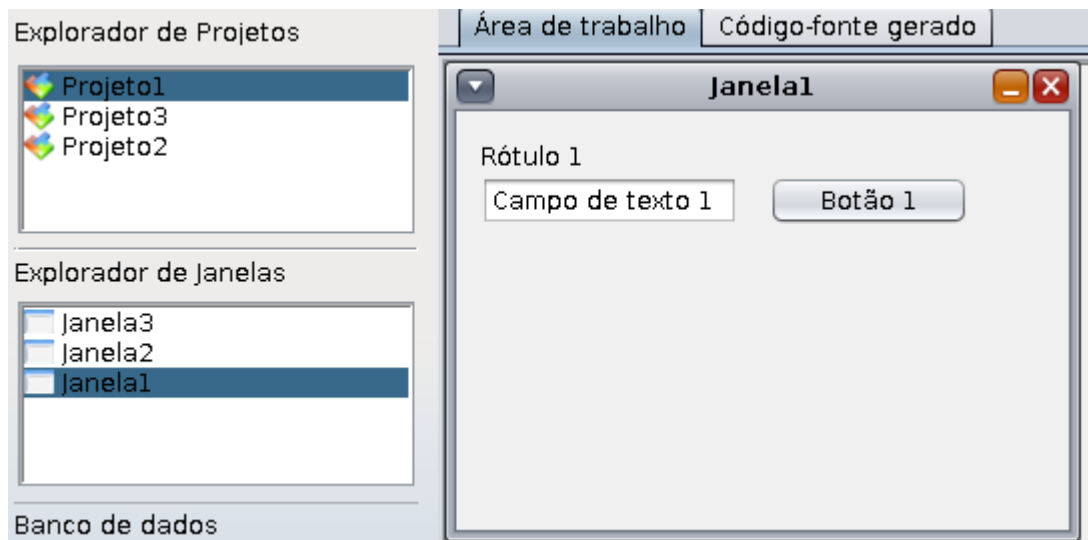


Figura XI- Área de trabalho.

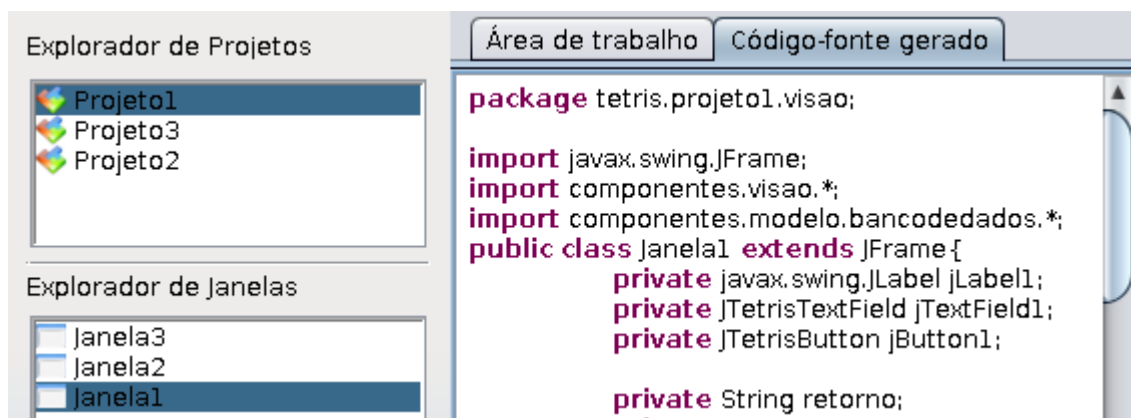


Figura XII- Visualizador de Código-fonte gerado.

2.7.6 Paleta de Objetos

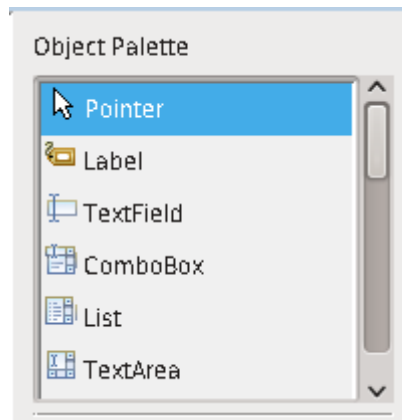


Figura XIII- Paleta de Objetos.

Como toda IDE com um módulo de estúdio visual, o Tetris concede ao programador a possibilidade de adicionar componentes na janela em edição, podendo assim confeccionar o seu programa de forma rápida e intuitiva.

Como padrão, a Paleta de Objetos será composta pelos componentes: rótulo, campo de texto, caixa de combinações, lista, área de texto, painel de edição, painel, barra de ferramentas, painel de abas, imagem, botão, tabela, campo de checagem, campo de escolha, barra de menus, timer, procedure, variable e relatório.

Seu uso é extremamente simples, sendo necessário somente clicar no objeto que se deseja adicionar (na Paleta de Objetos) e clicar no local da janela que se quer manipular.

2.7.7 Inspetor de Objetos

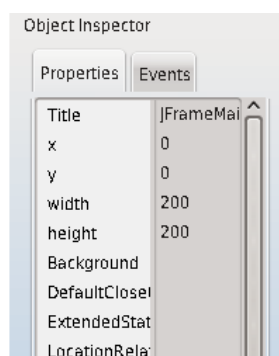


Figura XIV- Inspetor de Objetos.

Composto por duas abas, que são a *Propriedades* e a *Eventos*, o *Inspetor de Objetos* é responsável por permitir a manipulação dos atributos e métodos de determinado componente na tela, possibilitando visualizar as alterações diretamente na *Área de Trabalho* ao realizar alguma modificação em qualquer valor de atributo.

É uma ferramenta essencial para a criação visual, pois é por ela que o usuário tem o poder de personalização.

Na aba de *Eventos* é possível definir ações para os métodos dos objetos, como por exemplo: evento *OnClick* do *botao1* abre a *Janela2*.

2.7.8 Lista de Componentes da Janela

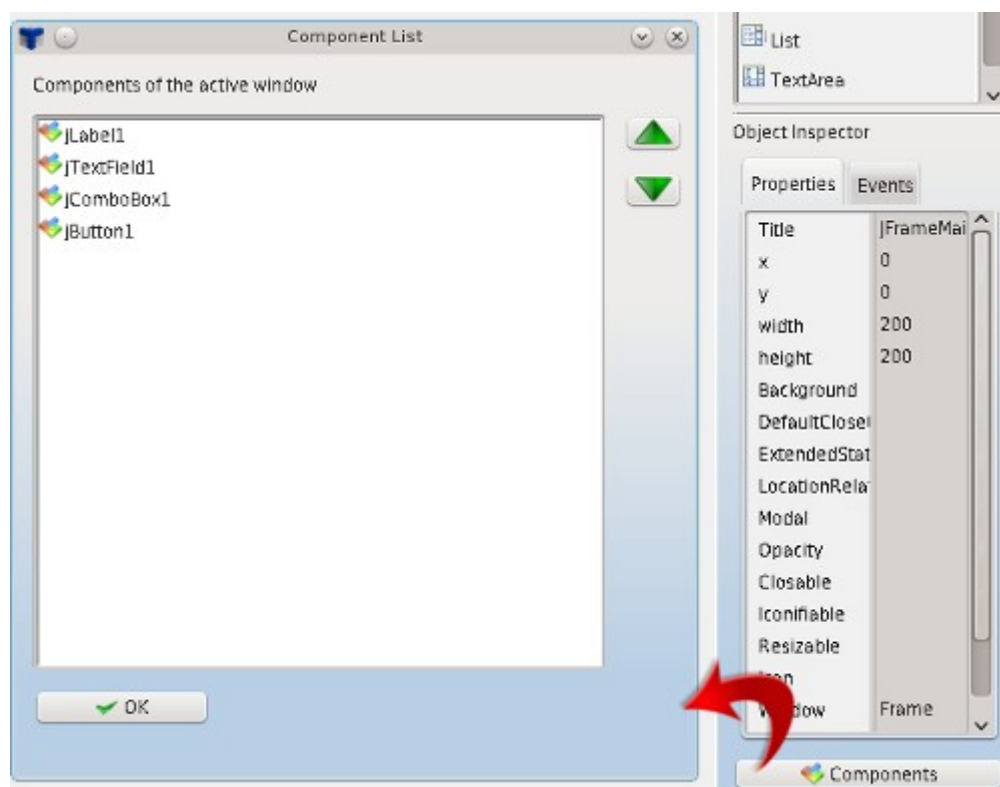


Figura XV- Lista de Componentes da Janela.

Mostra uma lista com todos os componentes da janela em edição. O desenvolvedor pode modificar a ordem de criação e mudar o foco de edição. É acessível através do botão Componentes na barra lateral direita ou pressionando **CTRL + F**.

2.7.9 Componentes Externos

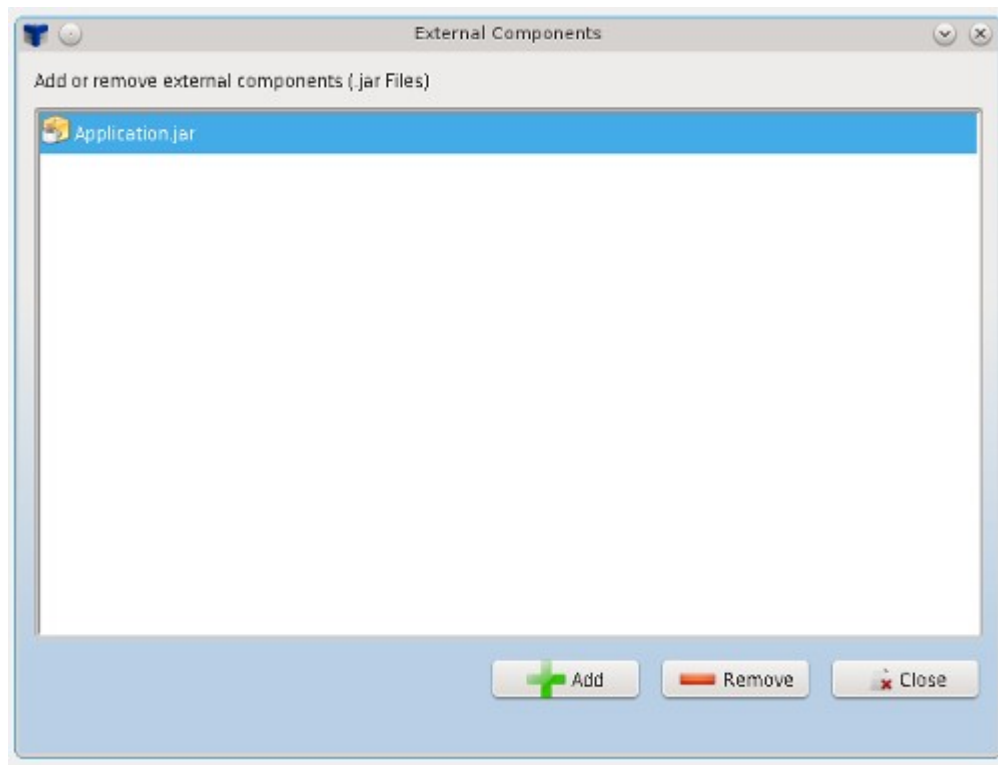


Figura XVI- Janela de Manipulação de Componentes Externos do Projeto.

Acessa-se através do popup menu do Explorador de Projetos, selecionando Componentes Externos. A janela que aparece nos traz um gerenciador de arquivos JAR externos, proporcionando aos projetos a possibilidade de ter componentes externos.

2.8 Análise de Requisitos

Seção destinada a dirimir detalhes sobre especificações de requisitos do programa aplicativo TetrisIDE.

Este documento é destinado a quaisquer profissionais que sejam responsáveis pela análise, implementação e testes dos requisitos do sistema.

2.8.1 Requisitos Não-Funcionais

Na tabela 2 estão listados todos os requisitos característicos necessários para o sistema ser desenvolvido e funcionar.

Tabela II- Requisitos Não-Funcionais do TetrisIDE.

Requisit o	Descrição
RNF1	O software será desenvolvido em linguagem Java.
RNF2	O software gerará programas em linguagem Java.
RNF3	O software gerará aplicações que conseguem se comunicar com o gerenciador de banco de dados MySQL.
RNF4	O software será multiplataforma.
RNF5	O software gerará aplicações multiplataforma.
RNF6	O software necessitará de um hardware mínimo com: <ul style="list-style-type: none">• Processador Intel Celeron;• 1 GB de memória RAM;e• 10 MB de espaço em disco (somente para a aplicação).
RNF7	O software necessitará do Java Development Kit (JDK) 7 ou superior.

2.8.2 Requisitos Funcionais

A tabela 3 mostra os requisitos funcionais do TetrisIDE, que descrevem determinadas ações (ou funções) que o sistema deverá suportar.

Tabela III- Requisitos Funcionais do TetrisIDE.

Requisit o	Descrição
RF1	O sistema deve exibir tela principal, atualizando as listas das ferramentas.
RF2	O desenvolvedor deve Manter Projeto .
RF3	O desenvolvedor deve Abrir Projeto .
RF4	O desenvolvedor deve Fechar Projeto .

RF5	O desenvolvedor deve Gerar Código-fonte do Projeto.
RF6	O desenvolvedor deve Compilar Código-fonte do Projeto.
RF7	O desenvolvedor deve Executar o Projeto .
RF8	O desenvolvedor deve Exportar Projeto .
RF9	O desenvolvedor deve Importar Projeto .
RF10	O desenvolvedor deve Manter Janela .
RF11	O desenvolvedor deve Visualizar Janela .
RF12	O desenvolvedor deve Manter Componentes na Janela .
RF13	O desenvolvedor deve Manter Banco de Dados .
RF14	O desenvolvedor deve Manter Propriedades e Eventos do componente selecionado.
RF15	O desenvolvedor deve Adicionar Componentes Externos .
RF16	O desenvolvedor deve Selecionar Janela Principal .
RF17	O desenvolvedor deve Visualizar Código-fonte .
RF18	O desenvolvedor deve Visualizar Log .
RF19	O desenvolvedor deve Gerar Janela a partir do Banco de Dados .
RF20	O desenvolvedor deve Clonar Janela .
RF21	O desenvolvedor deve Espelhar Banco de Dados .
RF22	O desenvolvedor deve Configurar Sistema .
RF23	O desenvolvedor deve Configurar Banco de Dados .

2.9 Casos de Uso

Tabela IV- Casos de Uso do TetrisIDE.

Código	Descrição
UC1	Iniciar Sistema.
UC2	Manter Projeto.
UC3	Abrir Projeto.
UC4	Fechar Projeto.
UC5	Gerar Código-fonte.
UC6	Compilar Código-fonte.
UC7	Executar Projeto.
UC8	Exportar Projeto.

UC9	Importar Projeto.
UC10	Visualizar Janela.
UC11	Manter Janela.
UC12	Manter componentes na Janela.
UC13	Manter Propriedades e Eventos do componente.
UC14	Manter Banco de Dados.
UC15	Adicionar Componentes.
UC16	Selecionar Janela Principal.
UC17	Visualizar Código-Fonte.
UC18	Visualizar Log.
UC19	Gerar Janela a partir de Banco de Dados.
UC20	Clonar Janela.
UC21	Espelhar Banco de Dados.
UC22	Configurar sistema.
UC23	Configurar Banco de Dados.

2.9.1 Diagrama de Casos de Uso

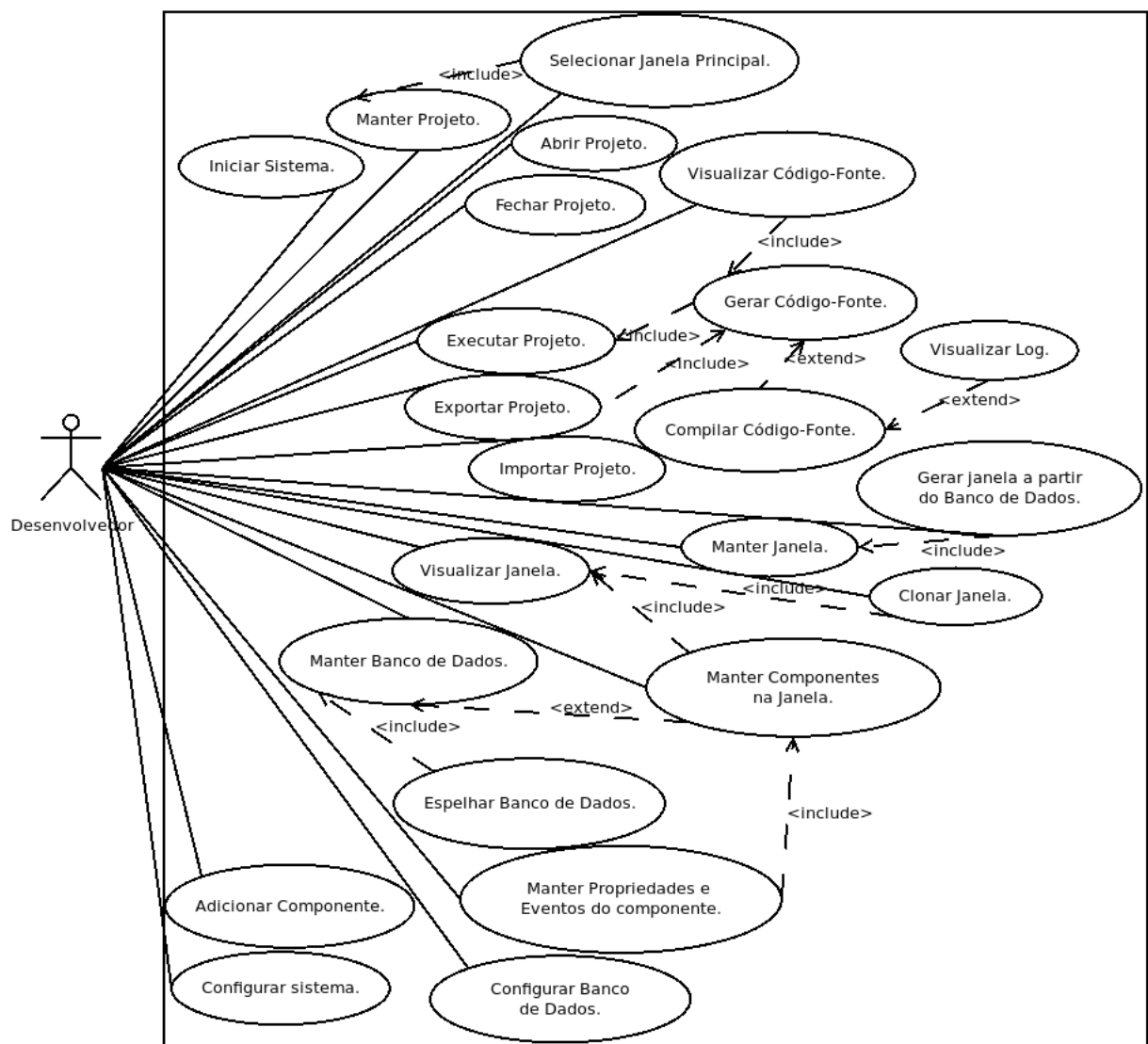


Figura XVII- Diagrama de Casos de Uso do TetrísIDE.

2.10 Diagramas de Atividades

Tabela V- Relação de Diagramas de Atividades do TetrísIDE.

Código	Descrição
DA1	Iniciar Sistema.
DA2	Manter Projeto.
DA3	Abrir Projeto.
DA4	Fechar Projeto.
DA5	Gerar Código-fonte.

DA6	Compilar Código-fonte.
DA7	Executar Projeto.
DA8	Exportar Projeto.
DA9	Importar Projeto.
DA10	Visualizar Janela.
DA11	Manter Janela.
DA12	Manter componentes na Janela.
DA13	Manter Propriedades e Eventos do componente.
DA14	Manter Banco de Dados.
DA15	Manter Componentes Externos.
DA16	Selecionar Janela Principal.
DA17	Visualizar Código-Fonte.
DA18	Visualizar Log.
DA19	Gerar Janela a partir de Banco de Dados.
DA20	Clonar Janela.
DA21	Espelhar Banco de Dados.
DA22	Configurar sistema.
DA23	Configurar Banco de Dados.

2.11 Descrição dos Casos de Uso

Aqui se encontra, de forma detalhada, a descrição de cada caso de uso do TetrisIDE, assim também como os seus diagramas de atividades e os referentes fluxos.

2.11.1 UC1 – Iniciar Sistema

Tabela VI - UC1 – Iniciar Sistema

Objetivo	Iniciar o sistema e preencher as listas de ferramentas.
Requisitos	RF1
Atores	-
Prioridade	Alta

Pré-condições	-
Frequência de Uso	Alta
Pós-condições	Mostrar janela do sistema pronta para uso.
Campos	-
Fluxo principal	<ol style="list-style-type: none"> 1. A tela de splash é iniciada; 2. O sistema verifica os arquivos e programas necessários; 3. Caso haja algo errado, o sistema exibe uma mensagem ao usuário, finaliza o sistema e pula para o passo 5; 4. Caso não, o sistema exibe a janela principal e preenche as listas de Explorador de Projetos e Paleta de Objetos;e 5. Encerra-se o fluxo principal.
Regras de negócio	A lista no Explorador de Projetos é preenchida com a lista de pastas presentes no workspace (pasta definida como local de trabalho).

2.11.1.1 DA1 – Diagrama de Atividades

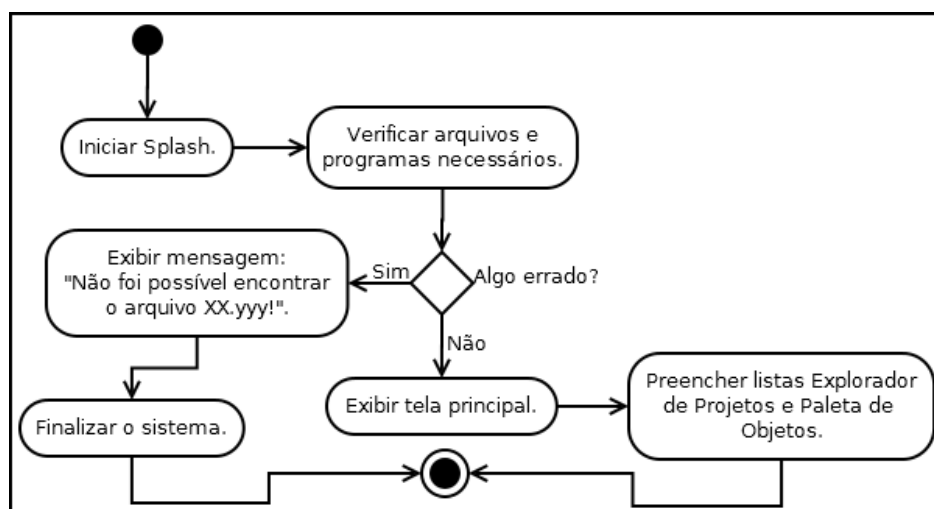


Figura XVIII - Diagrama de Atividades DA1 – Iniciar Sistema.

2.11.1.2 Protótipo da Tela

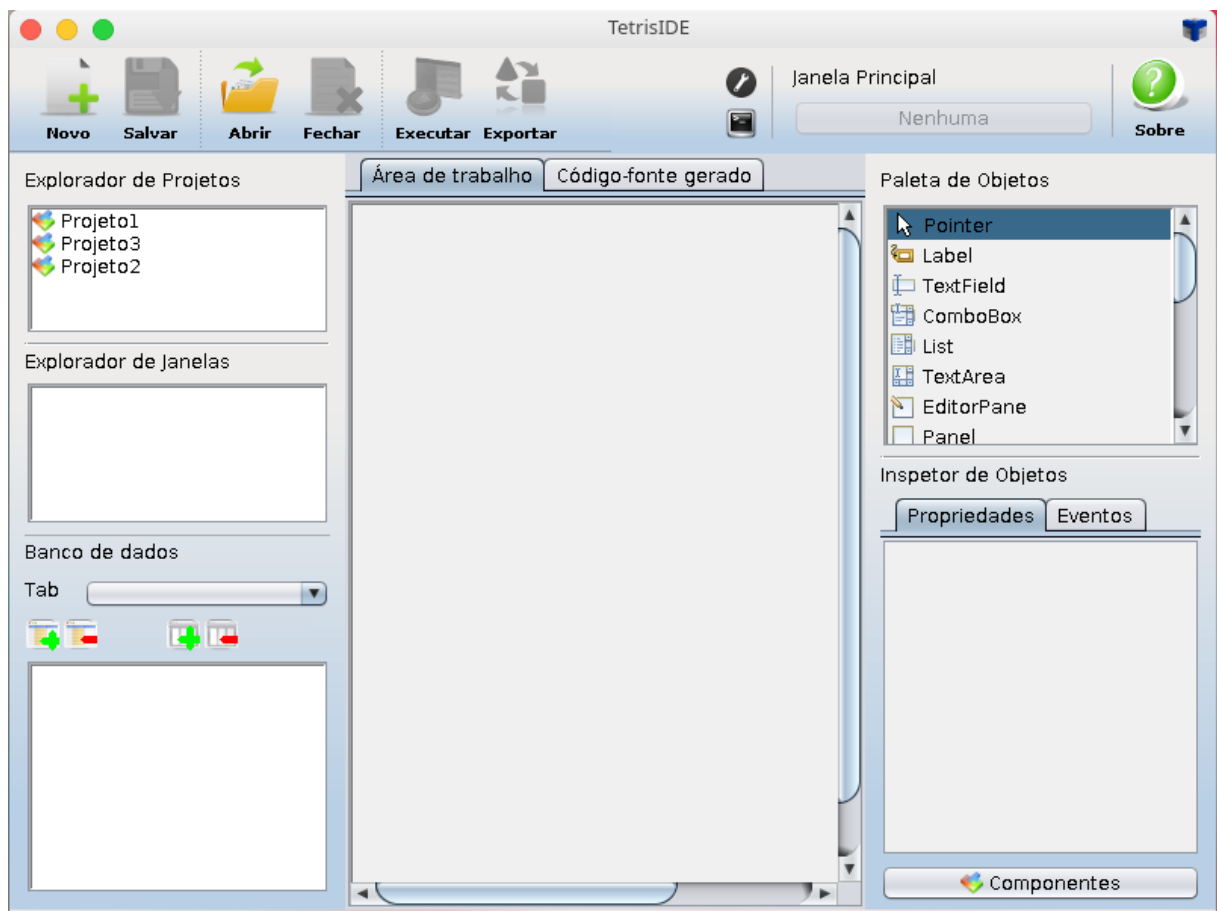


Figura XIX - Protótipo da tela principal do TetrísIDE.

2.11.2 UC2 – Manter Projeto

Tabela VII - UC2 – Manter Projeto

Objetivo	Manter um projeto.
Requisitos	RF2
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Sistema esteja iniciado.
Frequência de Uso	Média
Pós-condições	Mostrar projeto modificado/criado.
Campos	Nome (Campo de Texto)
Fluxo principal	1. O Desenvolvedor clica com o botão direito do mouse no

	<p>Explorador de Projetos; [FA1] [FA2]</p> <ol style="list-style-type: none"> 2. O Desenvolvedor seleciona a opção “Novo”; 3. O sistema exibe a janela com o nome “Novo Projeto”; 4. O Desenvolvedor preenche os campos corretamente; 5. O Desenvolvedor seleciona a opção “Criar”; 6. O sistema cria o projeto; 7. O sistema abre o projeto;e 8. Encerra-se o fluxo principal.
Fluxo Alternativo 1	<p>[FA1]</p> <ol style="list-style-type: none"> 1. O Desenvolvedor seleciona a opção “Renomear”; 2. O sistema exibe a janela com o nome “Renomear Projeto”; 3. O Desenvolvedor preenche os campos corretamente; 4. O Desenvolvedor seleciona a opção “Ok”; 5. O sistema renomeia o projeto selecionado;e 6. Dirige-se ao 8º passo do fluxo principal.
Fluxo Alternativo 2	<p>[FA2]</p> <ol style="list-style-type: none"> 1. O Desenvolvedor seleciona a opção “Excluir”; 2. O sistema pergunta se o Desenvolvedor realmente deseja excluir aquele projeto; 3. Se a resposta for positiva, o sistema exclui o projeto;e 4. Dirige-se ao 8º passo do fluxo principal.
Regras de negócio	<p>O projeto é criado no workspace (pasta definida como local de trabalho).</p>

2.11.2.1 DA2 – Diagrama de Atividades

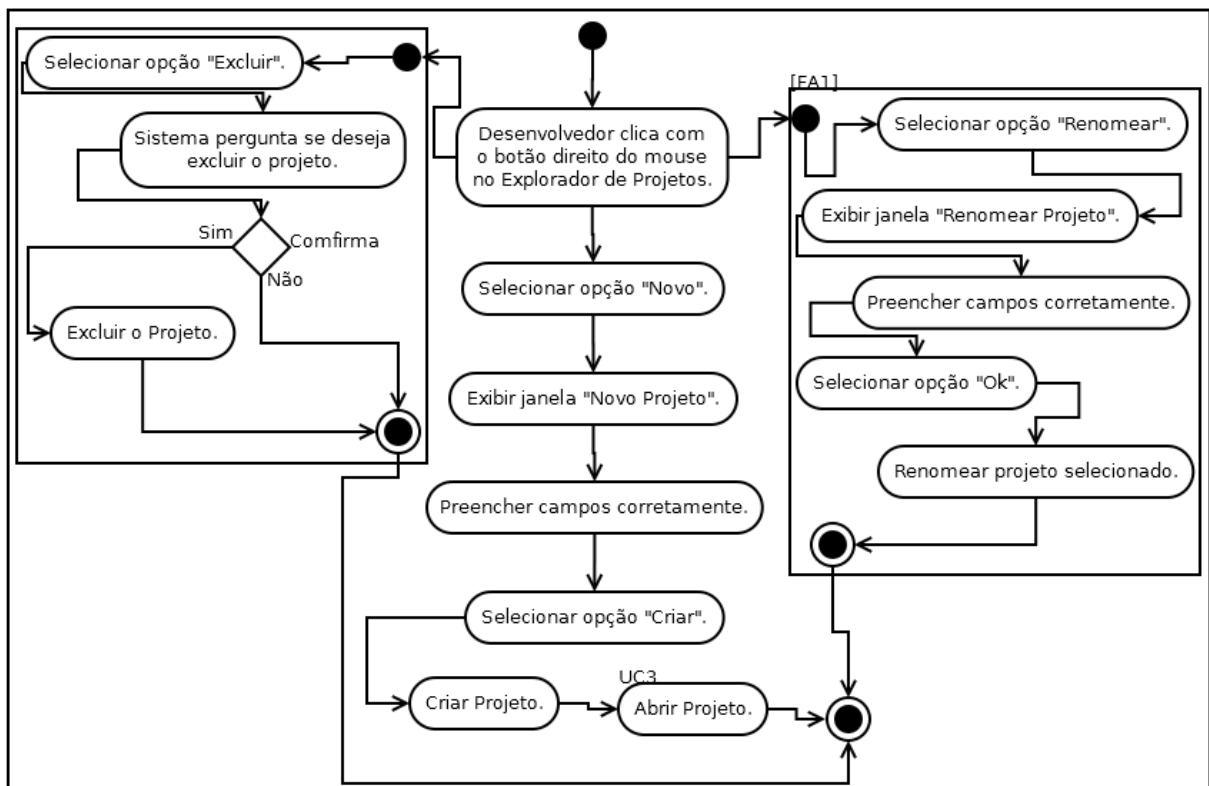


Figura XX - Diagrama de Atividades DA2.

2.11.2.2 Protótipo da Tela

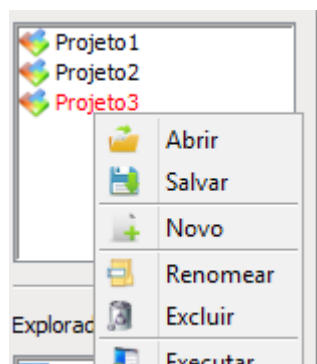


Figura XXI - Protótipo do menu do Explorador de Projetos do Tetrís.

2.11.3 UC3 – Abrir Projeto

Tabela VIII - UC3 – Abrir Projeto

Objetivo	Abrir um projeto.
Requisitos	RF3
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver projetos listados no Explorador de Projetos.
Frequência de Uso	Alta
Pós-condições	Mostrar as janelas do projeto no Explorador de Janelas e o banco de dados no Gerenciador de Banco de Dados.
Campos	-
Fluxo principal	1. O Desenvolvedor clica com o botão direito do mouse no projeto, localizado no Explorador de Projetos; 2. O Desenvolvedor seleciona a opção “Abrir”; 3. O sistema abre o projeto; 4. O sistema preenche o Explorador de Janelas com as janelas no projeto; 5. O sistema preenche o Gerenciador de Banco de Dados com o banco de dados do projeto;e 6. Encerra-se o fluxo principal.
Regras de negócio	O Projeto a ser aberto está presente no workspace (pasta definida como local de trabalho).

2.11.3.1 DA3 – Diagrama de Atividades

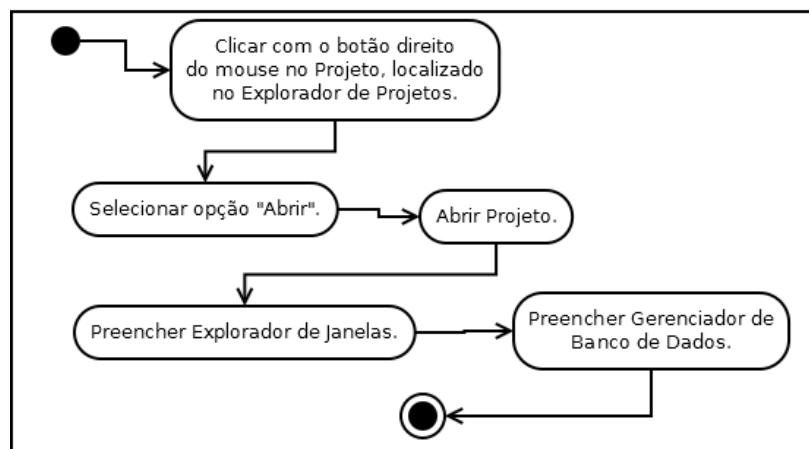


Figura XXII - Diagrama de Atividades DA3.

2.11.3.2 Protótipo da Tela

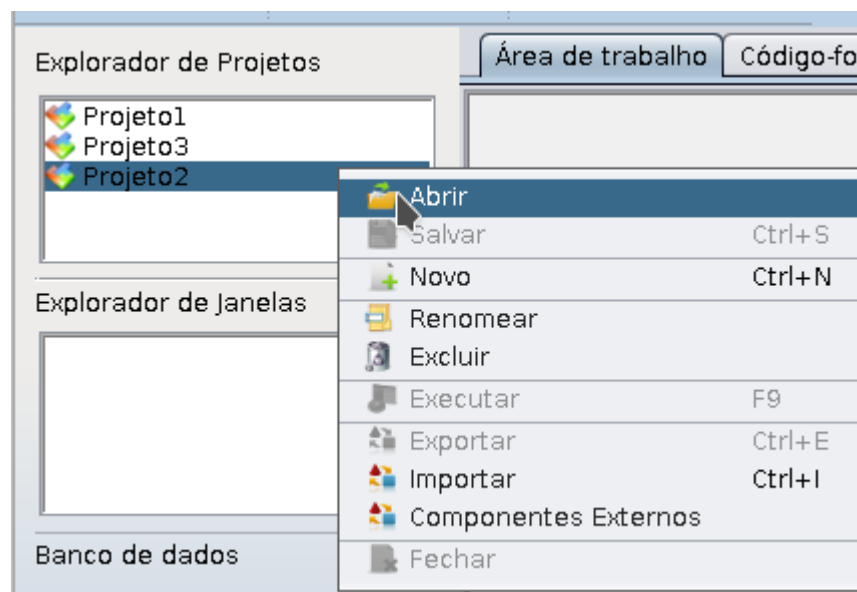


Figura XXIII- Protótipo do menu do Explorador de Projetos abrindo um projeto.

2.11.4 UC4 – Fechar Projeto

Tabela IX - UC4 – Fechar Projeto

Objetivo	Fechar um projeto.
Requisitos	RF4
Atores	Desenvolvedor
Prioridade	Média
Pré-condições	Haver um projeto aberto.
Frequência de Uso	Média
Pós-condições	Limpar a lista do Explorador de Janelas e a Área de Trabalho.
Campos	-
Fluxo principal	<ol style="list-style-type: none">1. O Desenvolvedor clica com o botão direito do mouse no projeto, localizado no Explorador de Projetos;2. O Desenvolvedor seleciona a opção "Fechar";3. O sistema pergunta se "Deseja realmente fechar o projeto";4. Caso negativo, o sistema pula para o passo 8;5. O sistema limpa o Explorador de Janelas;6. O sistema limpa a Área de Trabalho;

	7. O sistema fecha o projeto;e 8. Encerra-se o fluxo principal.
Regras de negócio	-

2.11.4.1 DA4 – Diagrama de Atividades

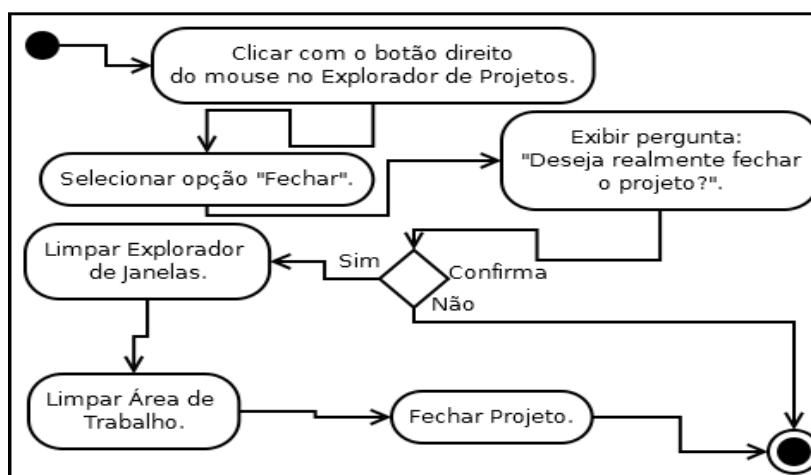


Figura XXIV - Diagrama de Atividades DA4.

2.11.4.2 Protótipo da Tela

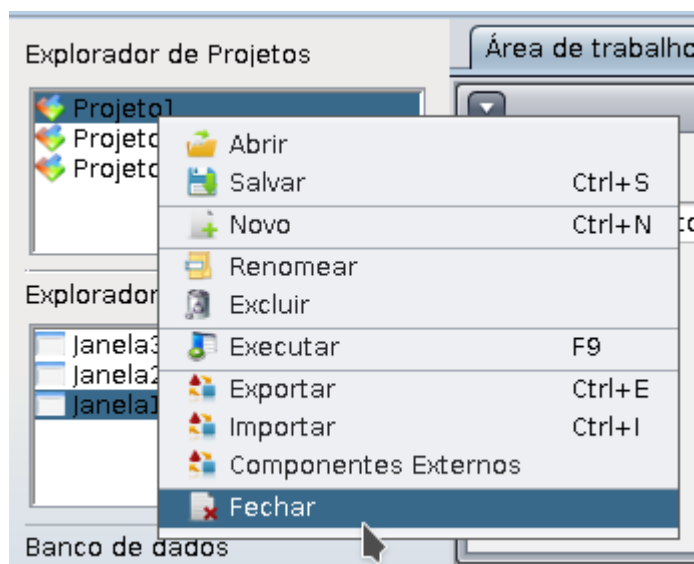


Figura XXV- Protótipo do menu fechando um projeto.

2.11.5 UC5 – Gerar Código-fonte

Tabela X - UC5 – Gerar Código-fonte

Objetivo	Gerar código-fonte em Java a partir do projeto.
Requisitos	RF5
Atores	-
Prioridade	Alta
Pré-condições	Haver um projeto aberto.
Frequência de Uso	Alta
Pós-condições	Gerar todo o código-fonte do projeto..
Campos	-
Fluxo principal	<ol style="list-style-type: none"> 1. O sistema verifica se existe pasta "src" no local do projeto; 2. Caso não exista, o sistema cria; 3. O sistema gera arquivos necessários para o projeto; 4. O sistema copia as imagens do projeto para a pasta "tetris\"(nome_do_projeto)\res", dentro da pasta "src"; 5. O sistema analisa cada janela e componente criado; 6. O sistema gera o código-fonte a partir da análise; 7. Caso haja algum problema, exibe a mensagem com a especificação do erro;e 8. Encerra-se o fluxo principal.
Regras de negócio	O código é gerado na pasta "src", dentro da pasta com o nome do projeto, localizado no workspace (pasta definida como espaço de trabalho).

2.11.5.1 DA5 – Diagrama de Atividades

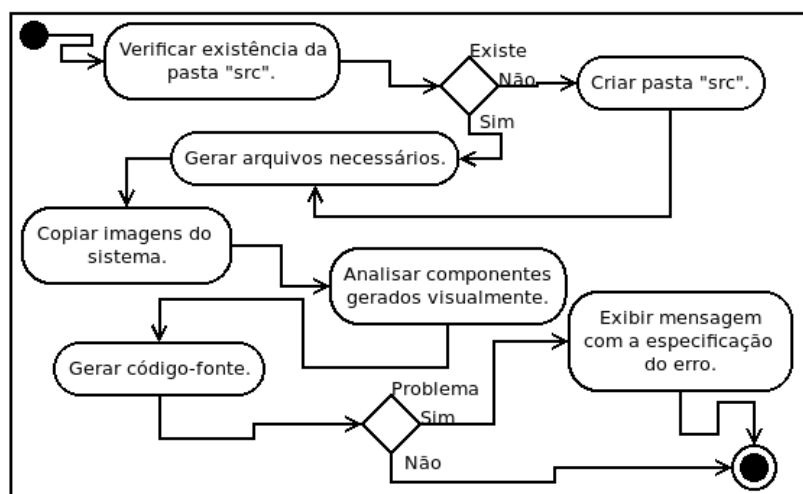


Figura XXVI - DA5 – Diagrama de Atividades.

2.11.6 UC6 – Compilar Código-fonte

Tabela XI - UC6 – Compilar Código-fonte

Objetivo	Compilar código-fonte, gerando os bytecodes em Java do projeto.
Requisitos	RF6
Atores	-
Prioridade	Alta
Pré-condições	Haver código-fonte do projeto já gerado.
Frequência de Uso	Alta
Pós-condições	Gerar bytecodes a partir do código-fonte do projeto.
Campos	-
Fluxo principal	<ol style="list-style-type: none"> 1. O sistema verifica se o JDK está instalado; 2. Caso não esteja, o sistema exibe a mensagem “O JDK não foi encontrado! Reinstale-o novamente!”, e pula para o passo 6; 3. O sistema compila o código-fonte gerado em bytecodes, através do JDK; 4. Se houver algum erro, o sistema exibe uma mensagem com a especificação do erro e pula para o passo 6; 5. O sistema organiza as classes geradas e os arquivos do projeto na pasta “bin”, dentro da pasta com o nome do projeto, localizada no workspace (espaço de trabalho);e 6. Encerra-se o fluxo principal.
Regras de negócio	Os bytecodes são gerados na pasta “bin”, dentro da pasta com o nome do projeto, localizado no workspace (espaço de trabalho).

2.11.6.1 DA6 – Diagrama de Atividades

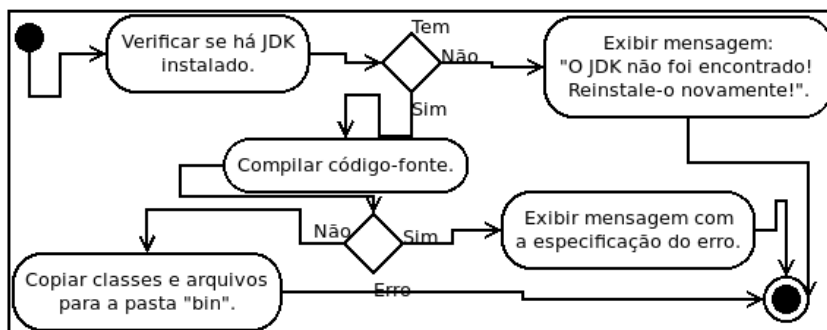


Figura XXVII - DA6 – Diagrama de Atividades.

2.11.7 UC7 – Executar Projeto

Tabela XII - UC7 – Executar Projeto

Objetivo	Executar programa gerado pelo projeto.
Requisitos	RF5, RF6 e RF7
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver projeto aberto.
Frequência de Uso	Alta
Pós-condições	Iniciar o programa fruto do projeto.
Campos	-
Fluxo principal	<ol style="list-style-type: none"> 1. O Desenvolvedor seleciona a opção Executar na Barra de ferramentas; 2. O sistema gera código-fonte das janelas do projeto; [UC5] 3. O sistema compila código-fonte gerado; [UC6] 4. O sistema executa a aplicação; 5. Encerra-se o fluxo principal;
Regras de negócio	O JRE (Java Runtime Environment) é necessário para executar a aplicação. O comando utilizado é o "java -cp (diretório bin do projeto):(diretório do workspace do TetrisIDE):(demais diretórios de componentes):. tetris/(nome do projeto)/Principal".

2.11.7.1 DA7 – Diagrama de Atividades

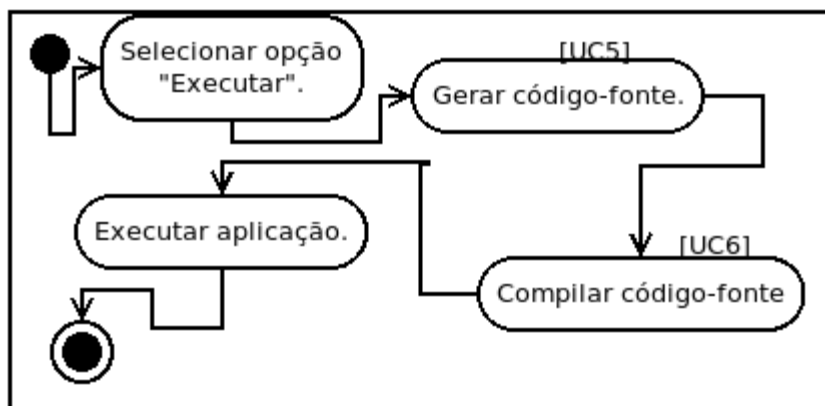


Figura XXVIII - DA7 – Diagrama de Atividades.

2.11.7.2 Protótipo da Tela



Figura XXIX - Tela inicial. Executando Projeto.

2.11.8 UC8 – Exportar Projeto

Tabela XIII - UC8 – Exportar Projeto

Objetivo	Exportar projeto TetrísIDE, código-fonte Java ou executável Jar.
Requisitos	RF8
Atores	Desenvolvedor
Prioridade	Média
Pré-condições	Haver código gerado a partir do projeto já salvo.
Frequência de Uso	Média
Pós-condições	Copiar código-fonte Java, projeto TetrísIDE ou executável Jar no local indicado pelo Desenvolvedor, .
Campos	-
Fluxo principal	1. O Desenvolvedor seleciona a opção Exportar na Barra de ferramentas; 2. Desenvolvedor seleciona opção de exportação; 3. Desenvolvedor seleciona local para a exportação; [FA1]

	<p>[FA2]</p> <p>4. O sistema copia o diretório do projeto no workspace para o local indicado pelo Desenvolvedor;e</p> <p>5. Encerra-se o fluxo principal.</p>
Fluxo Alternativo	<p>[FA1]</p> <p>1. O sistema copia o conteúdo do diretório src do projeto aberto para o local indicado pelo Desenvolvedor;e</p> <p>2. Dirige-se ao passo 5 do fluxo principal.</p> <p>[FA2]</p> <p>1. O sistema compacta os arquivos de bytecodes do projeto em arquivo Jar executável;</p> <p>2. O sistema copia o arquivo Jar para o local indicado pelo Desenvolvedor;e</p> <p>3. Dirige-se ao passo 5 do fluxo principal.</p>
Regras de negócio	-

2.11.8.1 DA8 – Diagrama de Atividades

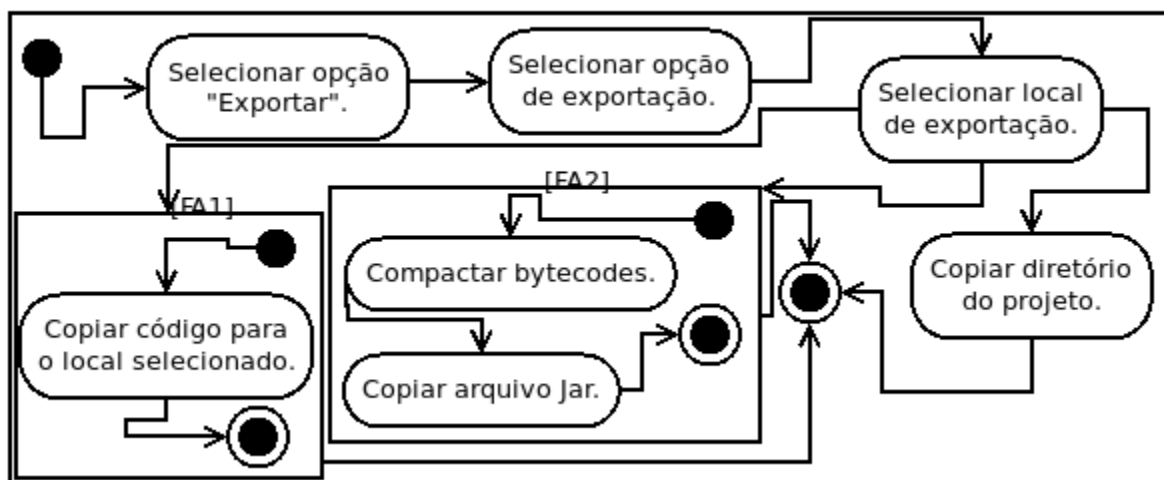


Figura XXX - DA8 – Diagrama de Atividades.

2.11.8.2 Protótipo da Tela

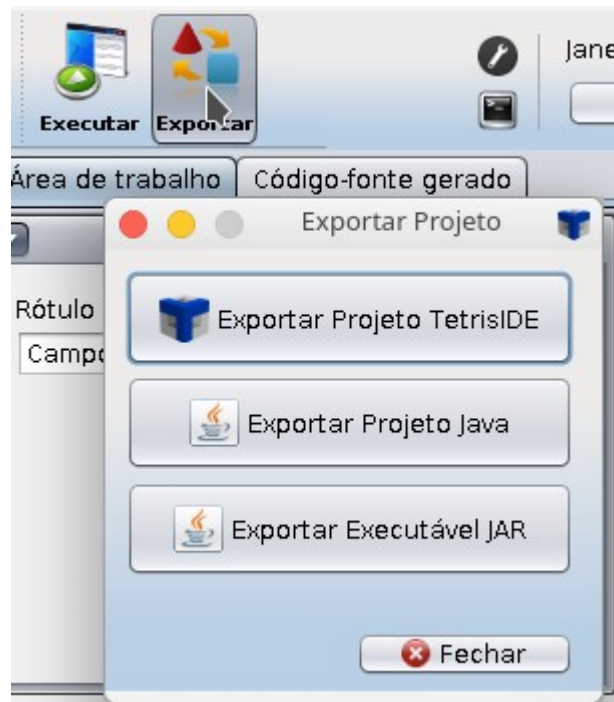


Figura XXXI - Protótipo da tela inicial. Exportando um projeto.

2.11.9 UC9 – Importar Projeto

Tabela XIV - UC9 – Importar Projeto

Objetivo	Importar projeto para o workspace (ambiente de trabalho).
Requisitos	RF9
Atores	Desenvolvedor
Prioridade	Média
Pré-condições	Haver um projeto TetrisIDE no local selecionado.
Frequência de Uso	Baixa
Pós-condições	Copiar projeto no workspace.
Campos	-
Fluxo principal	<ol style="list-style-type: none">1. O Desenvolvedor clica com o botão direito do mouse no Explorador de Projetos;2. O Desenvolvedor seleciona a opção Importar;3. O Desenvolvedor seleciona o local do projeto TetrisIDE a importar;4. O sistema verifica se há um projeto TetrisIDE naquele

	<p>local;</p> <p>5. Caso não haja, o sistema exibe a mensagem “Não há um projeto TetrisIDE no local indicado!” e pula para o passo 9;</p> <p>7. O sistema copia o projeto para o workspace;</p> <p>8. O sistema atualiza o Explorador de Projetos;e</p> <p>9. Encerra-se o fluxo principal.</p>
Regras de negócio	<p>O projeto, no diretório selecionado, é identificado por um arquivo, logo a ausência dele descaracteriza o diretório como um projeto TetrisIDE.</p>

2.11.9.1 DA9 – Diagrama de Atividades

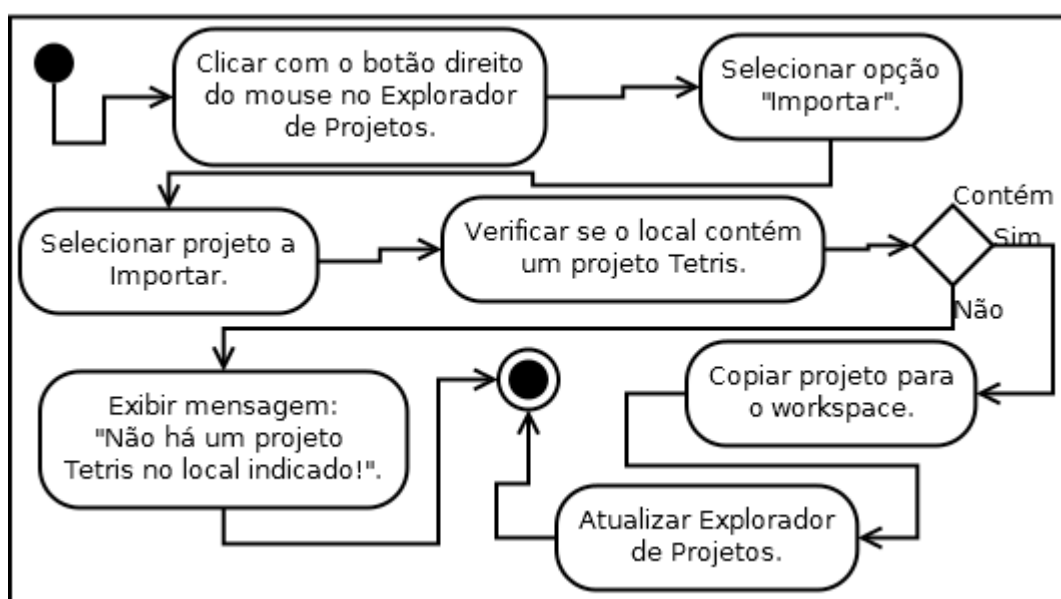


Figura XXXII - DA9 – Diagrama de Atividades.

2.11.9.2 Protótipo da Tela

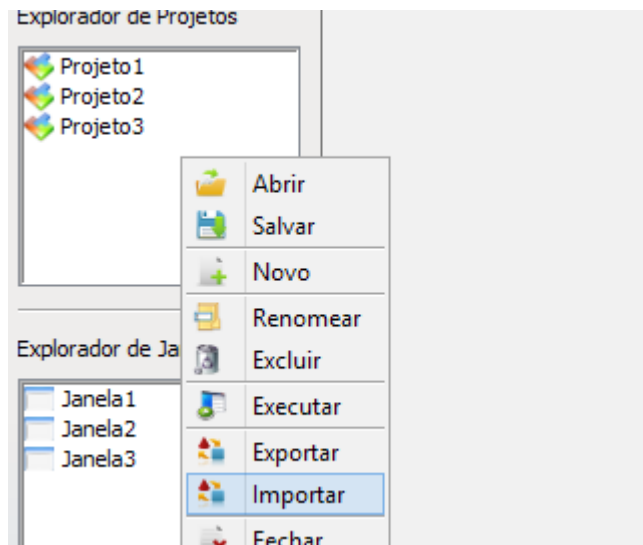


Figura XXXIII - Protótipo da tela principal. Importando um projeto.

2.11.10 UC10 – Visualizar Janela

Tabela XV - UC10 – Visualizar Janela

Objetivo	Visualizar uma janela do projeto na Área de Trabalho.
Requisitos	RF10
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver janelas no Explorador de Janelas.
Frequência de Uso	Alta
Pós-condições	Mostrar janela selecionada.
Campos	-
Fluxo principal	1. O Desenvolvedor clica com o botão direito do mouse no Explorador de Janelas; 2. O Desenvolvedor seleciona a opção Abrir; 3. O sistema exibe a janela na Área de Trabalho;e 4. Encerra-se o fluxo principal.
Regras de negócio	-

2.11.10.1 DA10 – Diagrama de Atividades

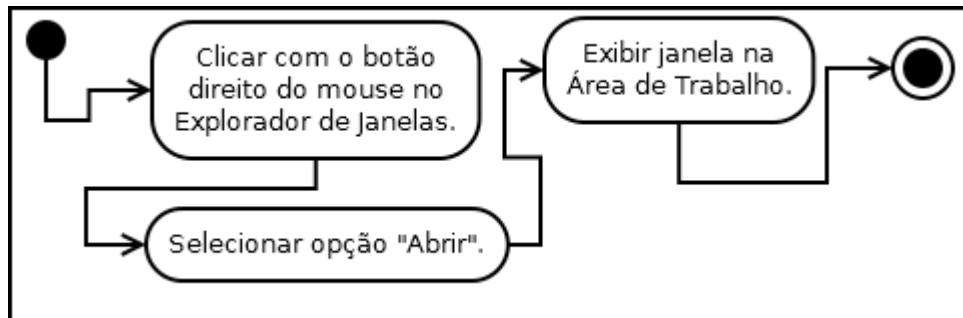


Figura XXXIV - DA10 – Diagrama de Atividades.

2.11.10.2 Protótipo da Tela

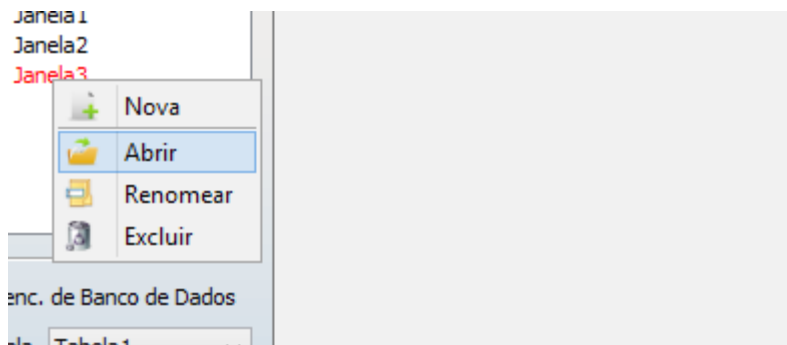


Figura XXXV - Protótipo da tela principal. Abrindo uma janela.

2.11.11 UC11 – Manter Janela

Tabela XVI - UC11 – Manter Janela

Objetivo	Manter uma janela do projeto.
Requisitos	RF11
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver uma janela aberta.
Frequência de Uso	Alta
Pós-condições	Criar/Alterar/Excluir uma janela.
Campos	-
Fluxo principal	1. O Desenvolvedor clica com o botão direito do mouse no

	<p>Explorador de Janelas; [FA1] [FA2]</p> <ol style="list-style-type: none"> 2. O Desenvolvedor seleciona a opção “Nova”; 3. O Desenvolvedor entra com o nome e o título da janela; 4. O sistema cria a janela;e 5. Encerra-se o fluxo principal.
Fluxo Alternativo	<p>[FA1]</p> <ol style="list-style-type: none"> 1. O Desenvolvedor seleciona a opção “Excluir”; 2. O sistema pergunta se deseja realmente excluir a janela; 3. Caso negativo, pula para o passo 5 do fluxo principal;e 4. O sistema exclui a janela e vai para o passo 5 do fluxo principal. <p>[FA2]</p> <ol style="list-style-type: none"> 1. O Desenvolvedor seleciona a opção “Renomear”; 2. O Desenvolvedor entra com o novo nome da janela; 3. O sistema altera o nome da janela;e 4. Pula para o passo 5 do fluxo principal.
Regras de negócio	-

2.11.11.1 DA11 – Diagrama de Atividades

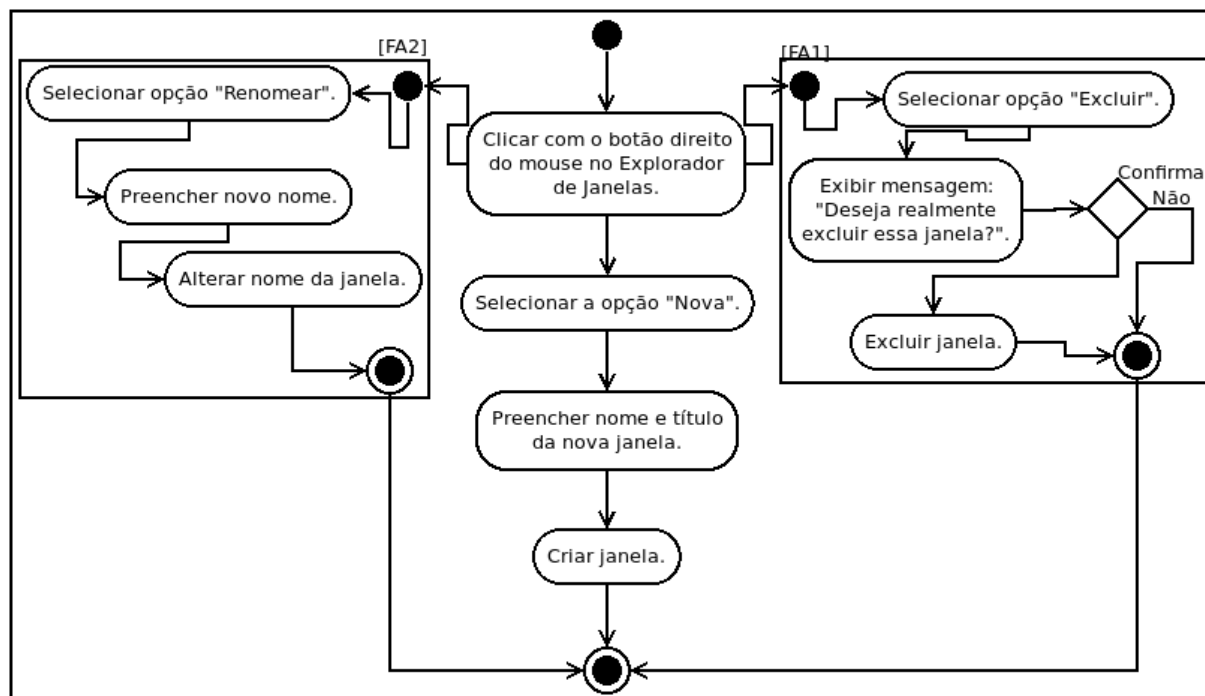


Figura XXXVI - DA11 – Diagrama de Atividades.

2.11.12 UC12 – Manter componentes na Janela.

Tabela XVII - UC12 – Manter componentes na Janela

Objetivo	Manter componentes numa janela do projeto na Área de Trabalho.
Requisitos	RF12
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver janela aberta na Área de Trabalho.
Frequência de Uso	Alta
Pós-condições	Criar/Modificar/Excluir componente na janela aberta.
Campos	-
Fluxo principal	<ol style="list-style-type: none">1. O Desenvolvedor abre a janela; [FA1] [FA2]2. O Desenvolvedor seleciona um componente na Paleta de Objetos;3. O Desenvolvedor clica na janela;4. O sistema cria o componente na janela clicada;e5. Encerra-se o fluxo principal.
Fluxo Alternativo	<p>[FA1]</p> <ol style="list-style-type: none">1. O Desenvolvedor seleciona um componente na janela;2. O Desenvolvedor pressiona o botão “Delete”;3. O sistema exclui o componente da janela;e4. Dirige-se ao passo 5 do fluxo principal. <p>[FA2]</p> <ol style="list-style-type: none">1. O Desenvolvedor seleciona um componente na janela;2. O Desenvolvedor altera uma propriedade/evento no Inspetor de Objetos;3. O sistema modifica o componente;e4. Dirige-se ao 5 passo do fluxo principal.
Regras de negócio	-

2.11.12.1 DA12 – Diagrama de Atividades

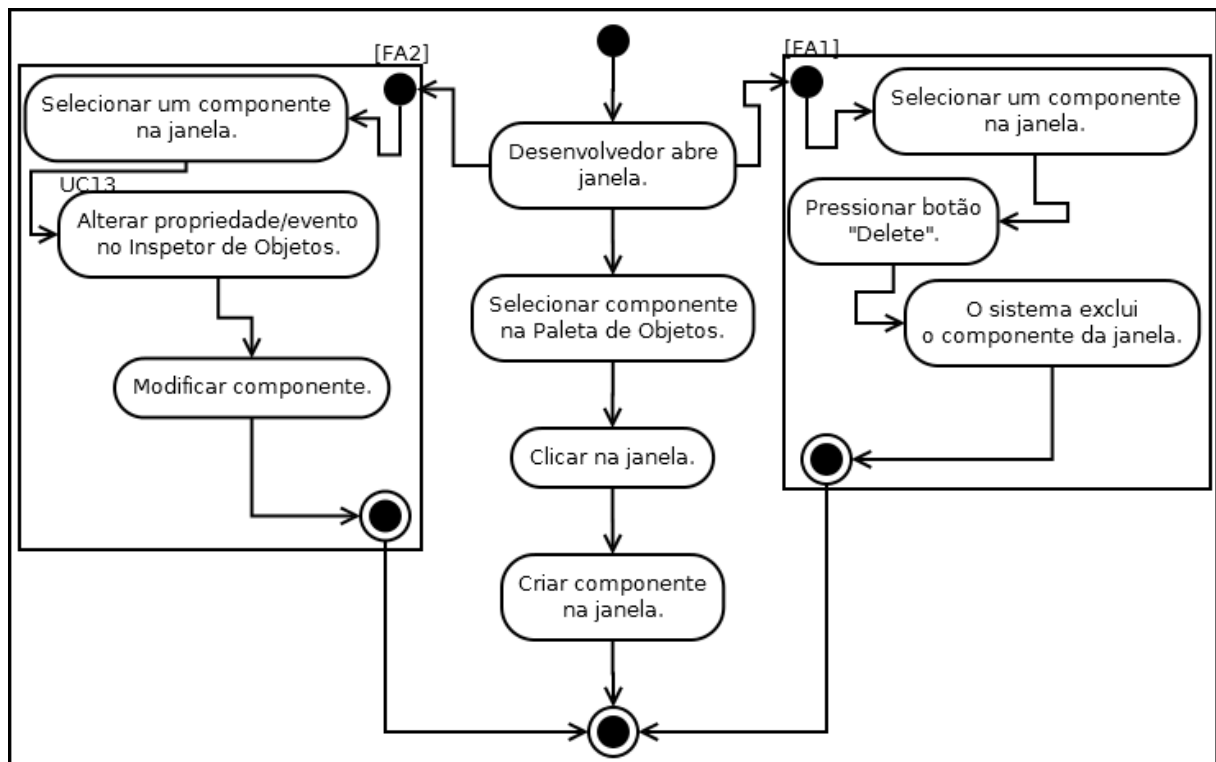


Figura XXXVII - DA12 – Diagrama de Atividades.

2.11.12.2 Protótipo da Tela

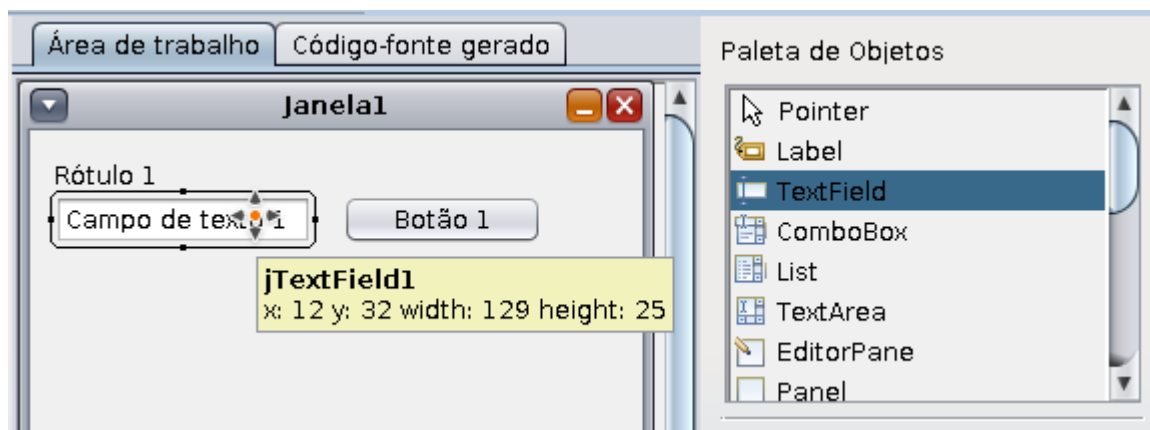


Figura XXXVIII - Protótipo da tela principal. Adicionando componente a uma janela.

2.11.13 UC13 – Manter Propriedades e Eventos do componente.

Tabela XVIII - UC13 – Manter Propriedades e Eventos do componente.

Objetivo	Manter propriedades/eventos de um componente numa janela do projeto na Área de Trabalho.
Requisitos	RF14
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver componentes numa janela aberta na Área de Trabalho.
Frequência de Uso	Alta
Pós-condições	Modificar componente na janela aberta.
Campos	-
Fluxo principal	<ol style="list-style-type: none">1. O Desenvolvedor clica no componente; [FA1]2. O Desenvolvedor seleciona uma propriedade do componente no Inspetor de Objetos;3. O Desenvolvedor altera o valor da propriedade;4. O sistema modifica o componente;e5. Encerra-se o fluxo principal.
Fluxo Alternativo	[FA1] <ol style="list-style-type: none">1. O Desenvolvedor seleciona um evento do componente no Inspetor de Objetos;2. O Desenvolvedor adiciona funções no evento do componente;e3. Dirige-se ao 4 passo do fluxo principal.
Regras de negócio	As funções a selecionar dos eventos são pré-equipadas no TetrisIDE, porém o desenvolvedor tem liberdade de implementar suas rotinas além das pré-definidas.

2.11.13.1 DA13 – Diagrama de Atividades

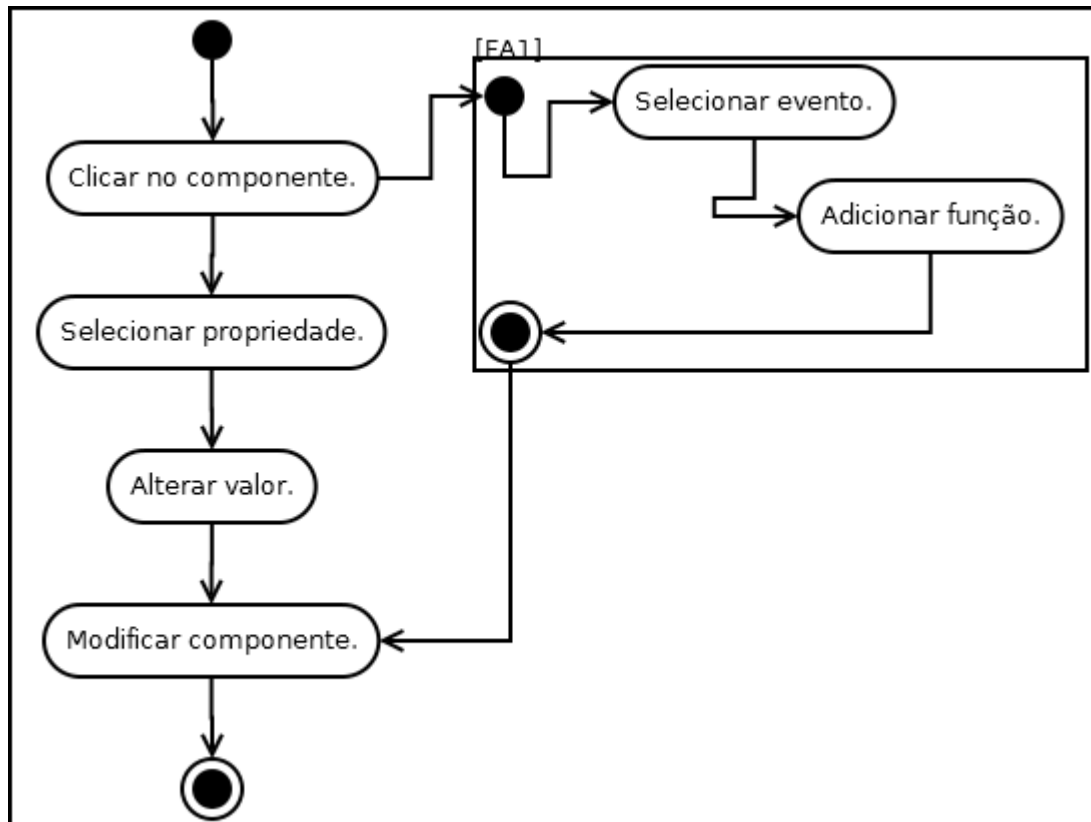


Figura XXXIX - DA13 – Diagrama de Atividades.

2.11.13.2 Protótipo da Tela

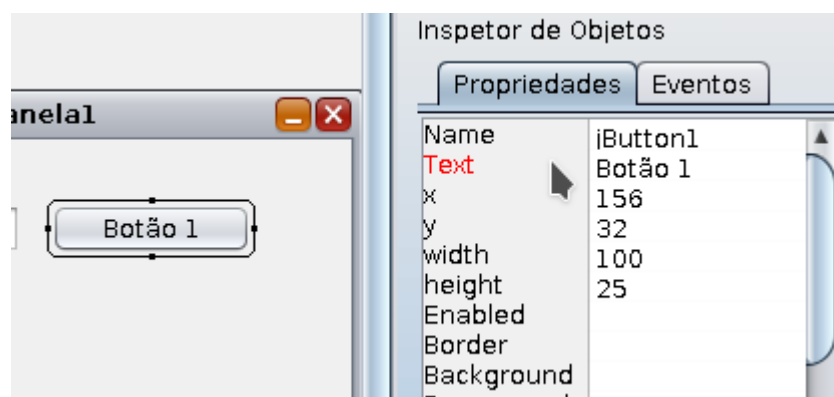


Figura XL - Protótipo da tela principal. Alterando propriedade texto em um botão.

2.11.14 UC14 – Manter Banco de Dados

Tabela XIX - UC14 – Manter Banco de Dados.

Objetivo	Manter o banco de dados do projeto.
Requisitos	RF13
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver um projeto aberto.
Frequência de Uso	Alta
Pós-condições	Modificar propriedades do schema do banco de dados do projeto.
Campos	-
Fluxo principal	<div><div>[FA1] [FA2]</div><div>[FA3] 1. O Desenvolvedor clica no botão “Adicionar tabela” ; 2. O Desenvolvedor preenche um nome para a tabela; 3. O sistema cria a tabela; 4. O Gerenciador de Banco de Dados é atualizado;e 5. Encerra-se o fluxo principal.</div></div>
Fluxo Alternativo	<div><div>[FA1] 1. O Desenvolvedor clica no botão “Remover tabela”; 2. O sistema pergunta se deseja realmente remover a tabela; 3. Caso negativo, dirige-se ao passo 5 do fluxo principal; 4. O sistema exclui a tabela do schema do banco de dados;e 5. Pula para o passo 4 do fluxo principal.</div><div>[FA2] 1. O Desenvolvedor seleciona uma tabela; 2. O Desenvolvedor clica no botão “Adicionar coluna”; 3. O Desenvolvedor entra com o nome e tipo da nova coluna; 4. O sistema cria a coluna;e 5. Pula para o passo 4 do fluxo principal.</div><div>[FA3] 1. O Desenvolvedor seleciona uma tabela; 2. O Desenvolvedor clica no botão “Remover coluna”; 3. O sistema pergunta se deseja realmente remover a tabela; 4. Caso negativo, dirige-se ao passo 5 do fluxo principal; 5. O sistema exclui a coluna da tabela;e 6. Pula para o passo 4 do fluxo principal.</div></div>
Regras de negócio	As funções a selecionar dos eventos são pré-equipadas no TetrisIDE, porém o desenvolvedor pode implementar suas

rotinas além das pré-definidas.

2.11.14.1 DA14 – Manter Banco de Dados

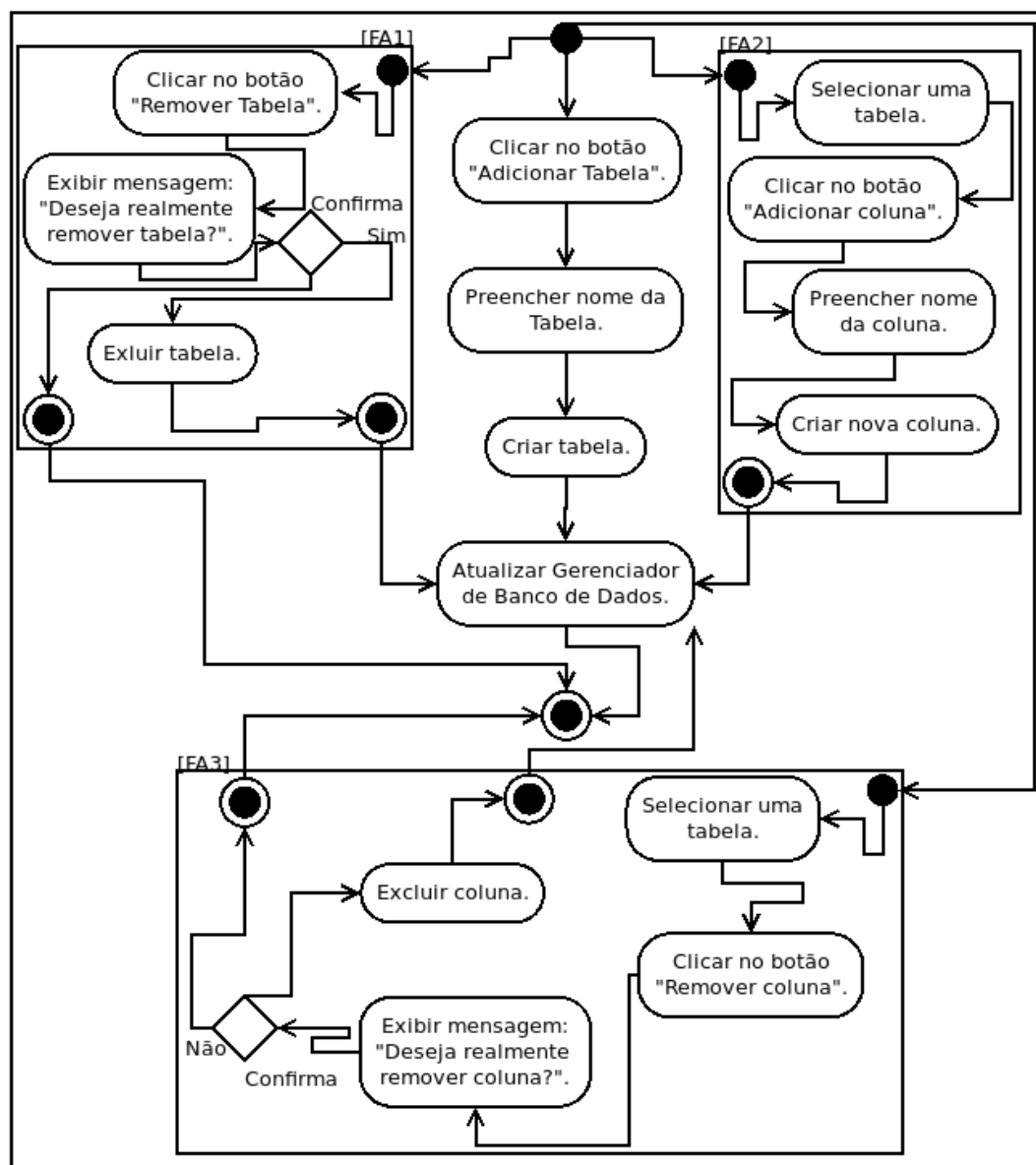


Figura XLI - DA14 – Diagrama de Atividades.

2.11.14.2 Protótipo da Tela

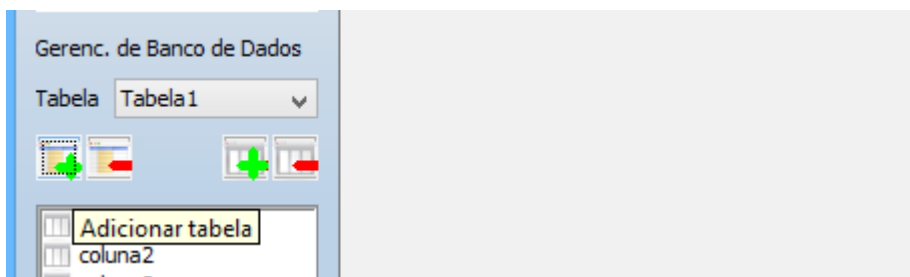


Figura XLII - Protótipo da tela principal. Adicionando tabela.

2.11.15 UC15 – Adicionar Componente

Tabela XX - UC15 – Adicionar Componente

Objetivo	Adicionar um componente ou função ao projeto externo ao TetrísIDE.
Requisitos	RF15
Atores	Desenvolvedor
Prioridade	Baixa
Pré-condições	Haver um projeto aberto.
Frequência de Uso	Baixa
Pós-condições	Ter o novo componente adicionado ao projeto.
Campos	-
Fluxo principal	<ol style="list-style-type: none">1. O Desenvolvedor clica com o botão direito do mouse no projeto, localizado no Explorador de Projetos;2. O Desenvolvedor seleciona a opção “Componentes Externos”;3. O sistema abre a janela com o nome “Componentes Externos”;4. O sistema preenche uma lista com os componentes externos adicionados no projeto; [FA1]5. O Desenvolvedor seleciona a opção “Adicionar”;6. O Desenvolvedor seleciona o componente que deseja adicionar;7. O sistema copia o componente para a pasta “comp” do projeto;e8. Encerra-se o fluxo principal.
Fluxo Alternativo	[FA1] <ol style="list-style-type: none">1. O Desenvolvedor seleciona um componente na lista;

	<p>2. O Desenvolvedor seleciona a opção “Remover”;</p> <p>3. O sistema pergunta se deseja realmente remover o componente;</p> <p>4. Caso negativo, dirige-se ao passo 8 do fluxo principal;</p> <p>5. O sistema exclui o componente do projeto;e</p> <p>6. Pula para o passo 8 do fluxo principal.</p>
Regras de negócio	Os componentes externos são guardados na pasta “comp”.

2.11.15.1 DA15 – Diagrama de Atividades

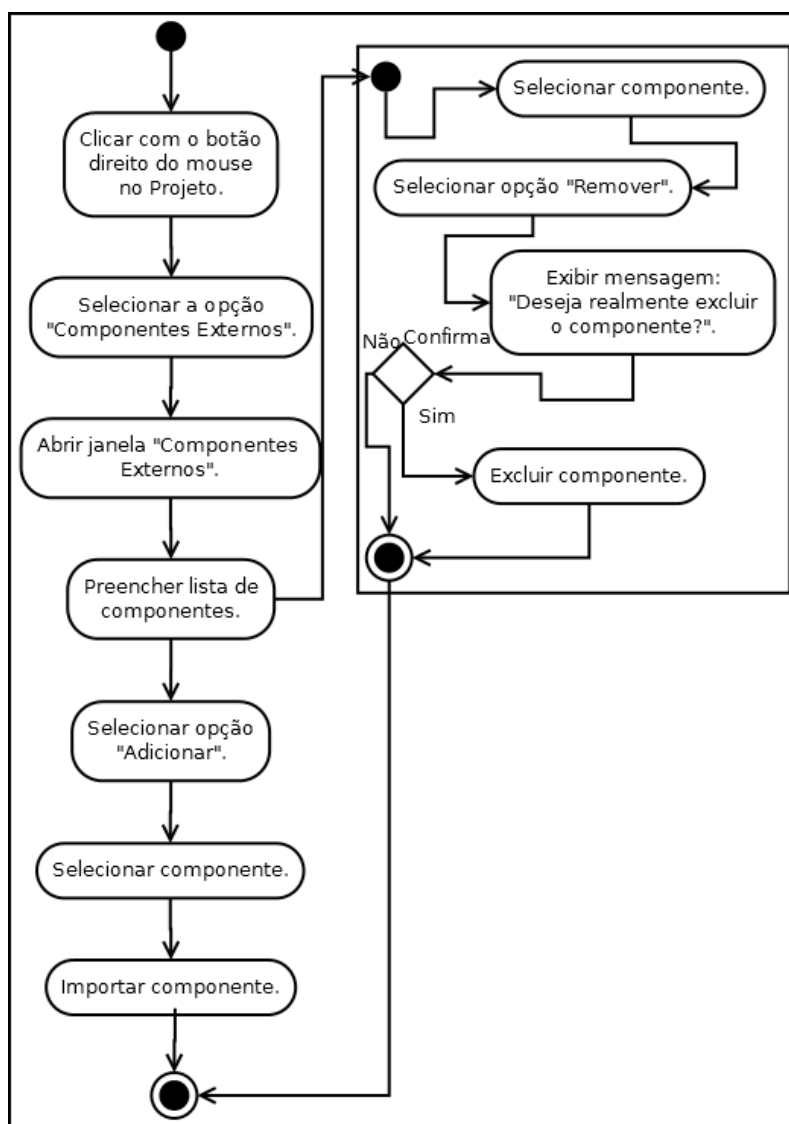


Figura XLIII - DA15 – Diagrama de Atividades.

2.11.15.2 Protótipo da Tela

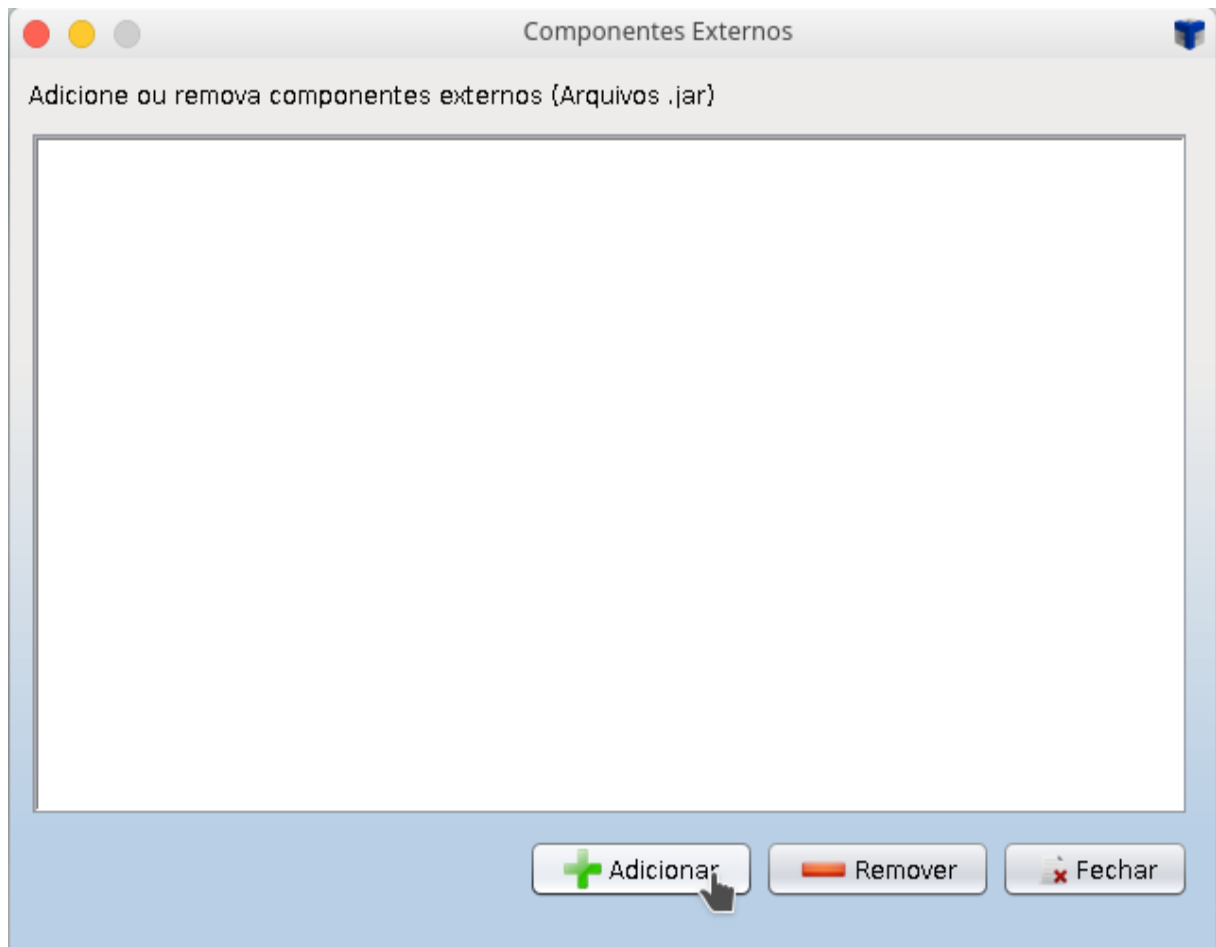


Figura XLIV - Protótipo da janela de Componentes Externos.

2.11.16 UC16 – Selecionar Janela Principal

Tabela XXI - UC16 – Selecionar Janela Principal.

Objetivo	Selecionar a Janela Principal do projeto.
Requisitos	RF16
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver um projeto aberto.
Frequência de Uso	Média
Pós-condições	Ter a Janela Principal definida no projeto.
Campos	-

Fluxo principal	<ol style="list-style-type: none"> 1. O Desenvolvedor clica no botão seletor de Janela Principal; 2. O sistema exibe a lista de janelas do projeto; 3. O Desenvolvedor seleciona a Janela Principal;e 4. Encerra-se o fluxo principal.
Regras de negócio	-

2.11.16.1 DA16 – Diagrama de Atividades

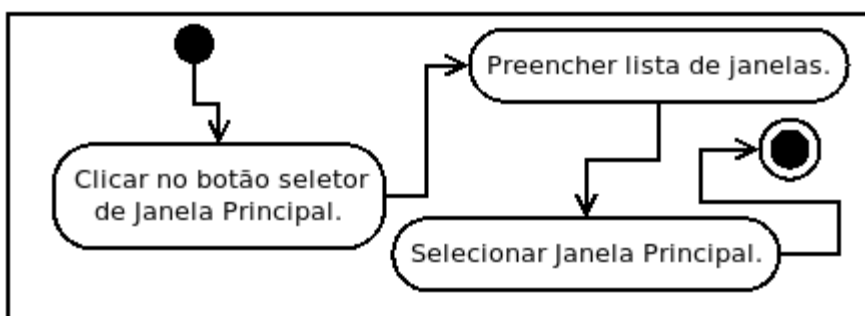


Figura XLV - DA16 – Diagrama de Atividades.

2.11.16.2 Protótipo da Tela

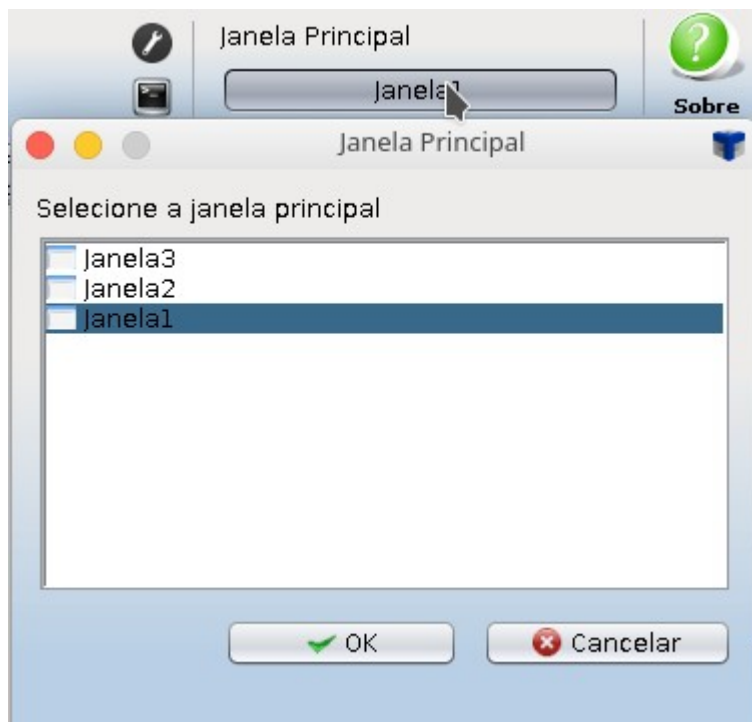


Figura XLVI- Protótipo da janela de Componentes Externos.

2.11.17 UC17 – Visualizar Código-Fonte

Tabela XXII - UC17 – Visualizar Código-Fonte.

Objetivo	Visualizar o código-fonte Java da janela aberta.
Requisitos	RF17, RF5
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver uma janela aberta na área de trabalho.
Frequência de Uso	Baixa
Pós-condições	Ter exibido o código-fonte Java da janela aberta.
Campos	-
Fluxo principal	1. O Desenvolvedor seleciona a aba “Código-fonte gerado”; 2. O sistema gera o código-fonte da janela; [UC5] 3. O sistema exibe o código-fonte da janela;e 4. Encerra-se o fluxo principal.
Regras de negócio	-

2.11.17.1 DA17 – Diagrama de Atividades

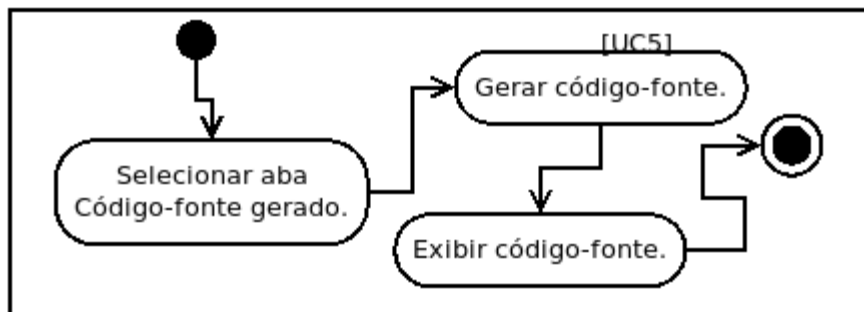


Figura XLVII- DA17 – Diagrama de Atividades.

2.11.17.2 Protótipo da Tela



Figura XLVIII- Protótipo da janela de Componentes Externos.

2.11.18 UC18 – Visualizar Código-Fonte

Tabela XXIII - UC18 – Visualizar Log.

Objetivo	Visualizar o log de compilação do projeto.
Requisitos	RF18
Atores	Desenvolvedor
Prioridade	Alta
Pré-condições	Haver o Desenvolvedor compilado o projeto.
Frequência de Uso	Alta
Pós-condições	Ter exibido o log de compilação do projeto.
Campos	-
Fluxo principal	1. O sistema exibe o log de compilação do projeto no Painel de Log;e 2. Encerra-se o fluxo principal.
Regras de negócio	O retorno gerado pelo compilador “javac” do JDK (Java Development Kit) é o log exibido no painel de log.

2.11.19 UC19 – Gerar Janela a partir de Banco de Dados.

Tabela XXIV - UC19 – Gerar Janela a partir de Banco de Dados.

Objetivo	Gerar uma janela para manipulação de tabela de banco de dados a partir das colunas definidas no projeto do banco de
-----------------	---

	dados.
Requisitos	RF19
Atores	Desenvolvedor
Prioridade	Média
Pré-condições	Haver colunas listadas no Gerenciador de Banco de Dados.
Frequência de Uso	Média
Pós-condições	Ter a janela criada na lista do Explorador de Janelas.
Campos	-
Fluxo principal	<ol style="list-style-type: none"> 1. O Desenvolvedor seleciona as colunas da tabela no Gerenciador de Banco de Dados; 2. O Desenvolvedor clica com o botão direito do mouse no Gerenciador de Banco de Dados; [FA1] 3. O Desenvolvedor seleciona a opção “Criar janela a partir de colunas selecionadas”; 4. O Desenvolvedor seleciona o modelo de janela que será criado; 5. O sistema gera a janela;e 6. Encerra-se o fluxo principal.
Fluxo alternativo	[FA1] <ol style="list-style-type: none"> 1. O Desenvolvedor seleciona a opção “Criar janela a partir de tabela inteira”;e 2. Dirige-se ao passo 4 do fluxo principal.
Regras de negócio	-

2.11.19.1 DA19 – Diagrama de Atividades

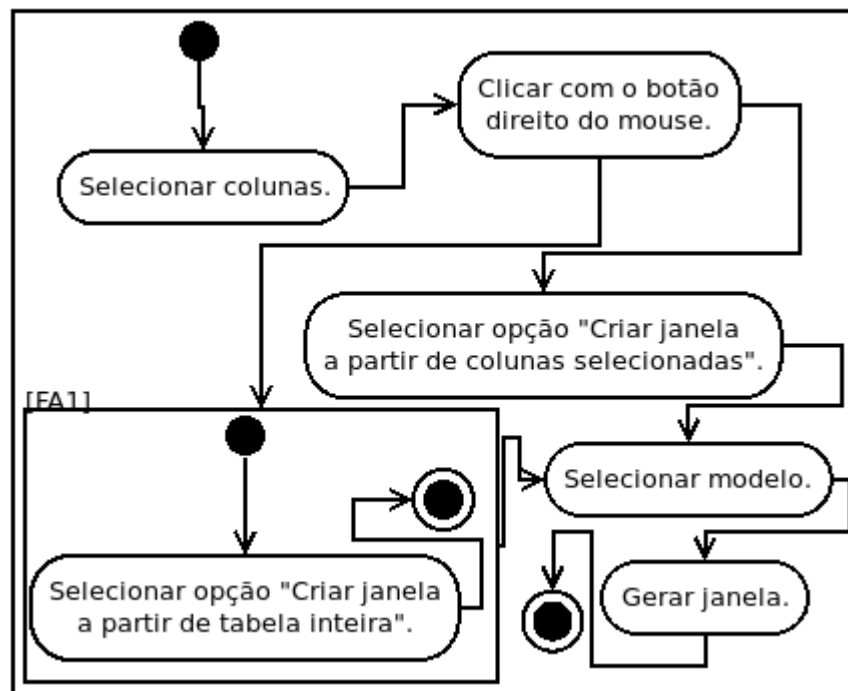


Figura XLIX- DA19 - Diagrama de Atividades.

2.11.19.2 Protótipo da Tela

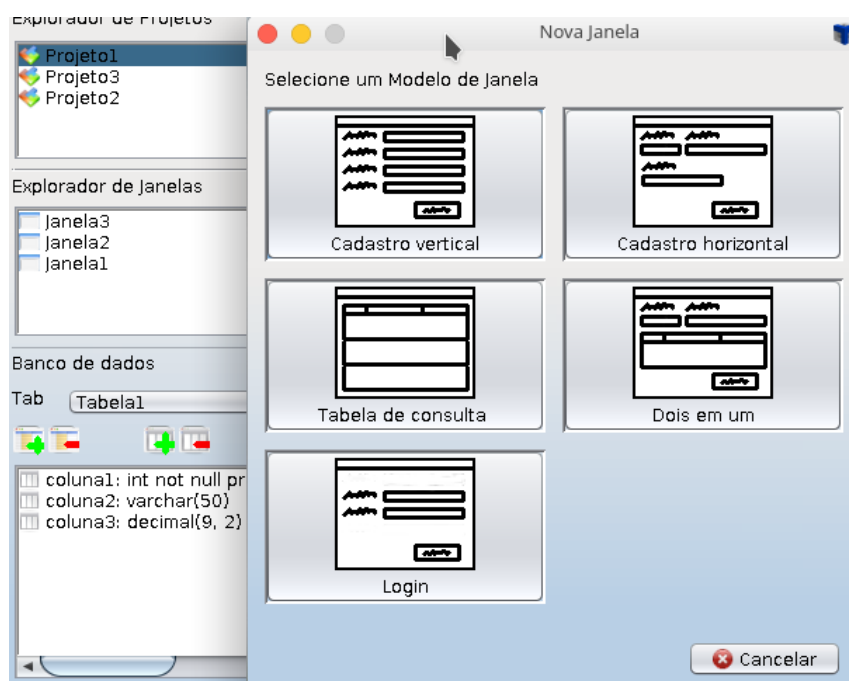


Figura L- Protótipo da tela de Gerar Janela a partir de Banco de Dados.

2.11.20 UC20 – Clonar Janela.

Tabela XXV - UC20 – Clonar Janela.

Objetivo	Gerar uma janela clone de outra já existente no projeto.
Requisitos	RF20
Atores	Desenvolvedor
Prioridade	Baixa
Pré-condições	Haver ao menos uma janela no projeto.
Frequência de Uso	Baixa
Pós-condições	Janela clonada adicionada ao Explorador de Janelas.
Campos	-
Fluxo principal	<ol style="list-style-type: none">1. O Desenvolvedor seleciona a janela a ser clonada;2. O Desenvolvedor clica com o botão direito do mouse no Explorador de Janelas;3. O Desenvolvedor seleciona a opção “Clonar Janela”;4. O Desenvolvedor preenche o nome e título na janela clone;5. O sistema verifica se já existe uma janela com o nome desejado no projeto;6. Caso sim, exibe mensagem “Já existe uma janela com esse nome!” e dirige-se ao passo 8;7. Clona janela;e8. Encerra-se o fluxo principal.
Regras de negócio	-

2.11.20.1 DA20 – Diagrama de Atividades

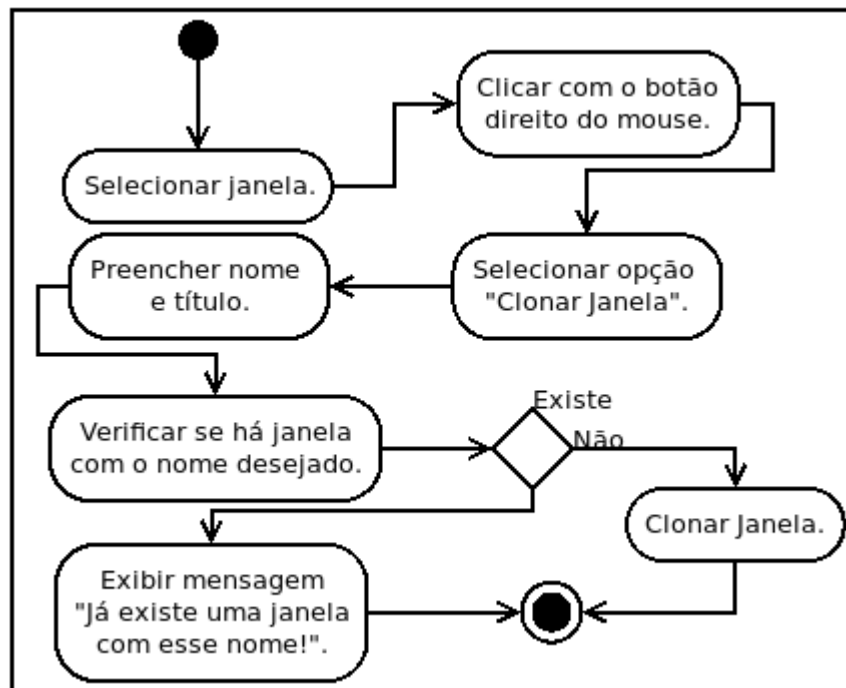


Figura LI- DA20 - Diagrama de Atividades.

2.11.20.2 Protótipo da Tela

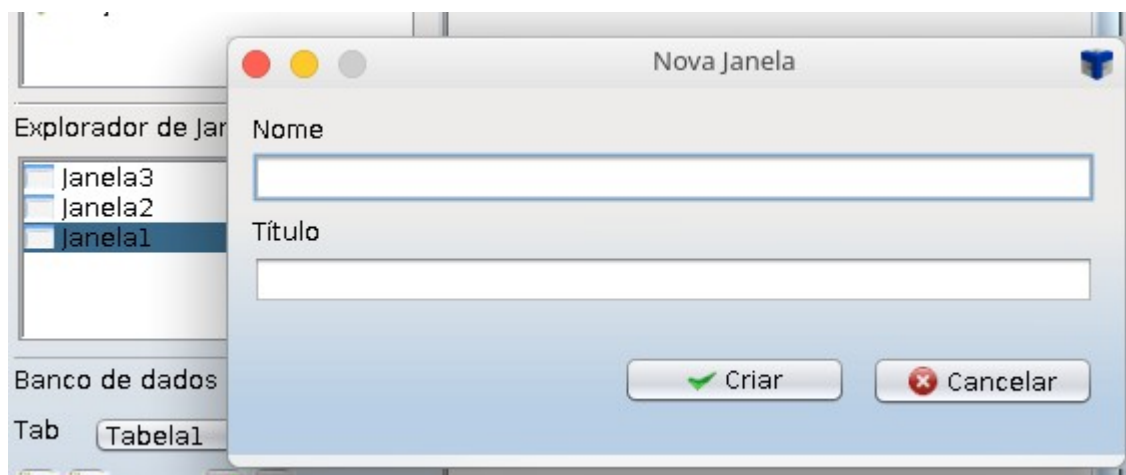


Figura LII- Protótipo da Tela de Clonar janela.

2.11.21 UC21 – Espelhar Banco de Dados.

Tabela XXVI - UC21 – Espelhar Banco de Dados.

Objetivo	Espelhar o banco de dados do Sistema Gerenciador de Banco de Dados no projeto aberto.
Requisitos	RF21
Atores	Desenvolvedor
Prioridade	Baixa
Pré-condições	Haver um projeto aberto.
Frequência de Uso	Baixa
Pós-condições	Schema do banco de dados do projeto idêntico ao banco de dados espelhado do SGDB.
Campos	-
Fluxo principal	<ol style="list-style-type: none">1. O Desenvolvedor clica com o botão direito do mouse no Gerenciador de Banco de Dados;2. O Desenvolvedor seleciona a opção “Espelhar banco de dados no projeto”;3. O Desenvolvedor preenche as informações de conexão com o SGDB;4. O Desenvolvedor seleciona o botão “Carregar lista de banco de dados”;5. O sistema preenche a lista de banco de dados;6. O Desenvolvedor seleciona o banco de dados a ser espelhado;7. O Desenvolvedor seleciona o botão “Espelhar”;8. O sistema espelha o banco de dados;e9. Encerra-se o fluxo principal.
Regras de negócio	Este processo é feito somente com SGDB MySQL/MariaDB.

2.11.21.1 DA21 – Diagrama de Atividades

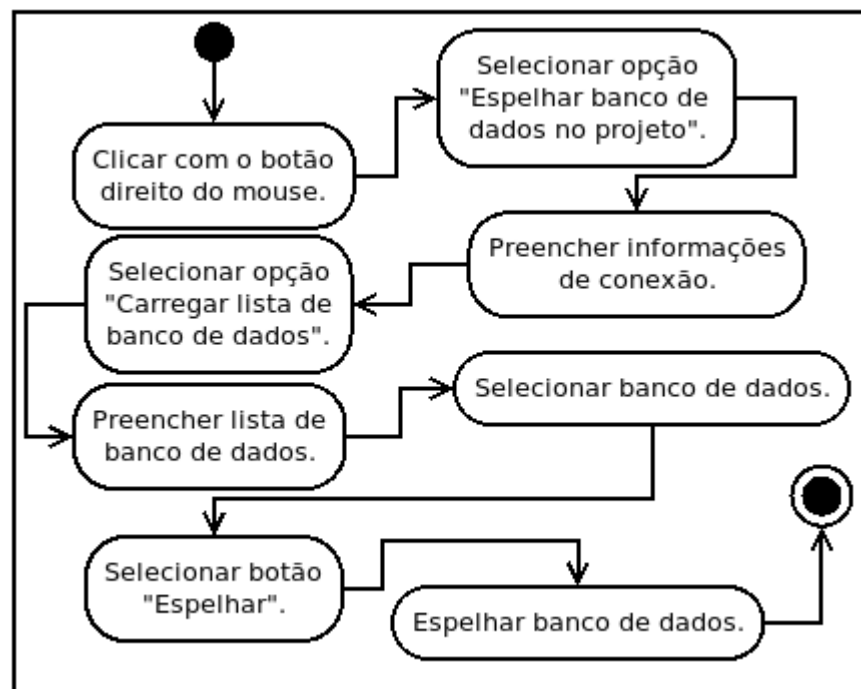


Figura LIII- DA21 - Diagrama de Atividades.

2.11.21.2 Protótipo da Tela

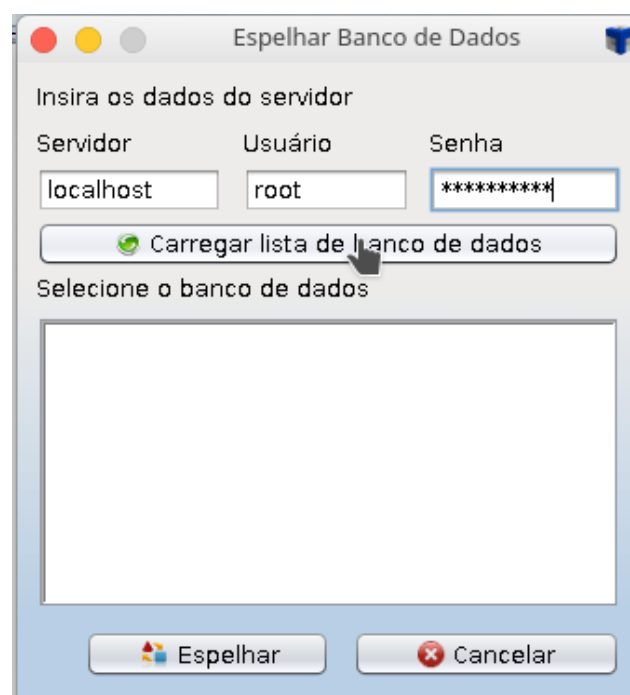


Figura LIV- Protótipo da Tela de Espelhar Banco de Dados.

2.11.22 UC22 – Configurar sistema.

Tabela XXVII - UC22 – Configurar sistema.

Objetivo	Configurar parâmetros gerais do TetrisIDE.
Requisitos	RF22
Atores	Desenvolvedor
Prioridade	Média
Pré-condições	-
Frequência de Uso	Baixa
Pós-condições	Modificar aspectos paramentados no TetrisIDE.
Campos	-
Fluxo principal	1. O Desenvolvedor seleciona o botão de “Configurações”; 2. O Desenvolvedor altera os dados desejado; 3. O Desenvolvedor salva as configurações;e 4. Encerra-se o fluxo principal.
Regras de negócio	As configurações são salvas no workspace do TetrisIDE em um arquivo nomeado de “conf”.

2.11.22.1 DA22 – Diagrama de Atividades

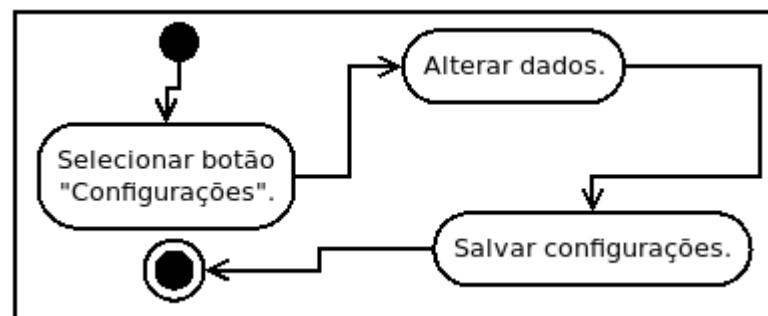


Figura LV- DA22 - Diagrama de Atividades.

2.11.22.2 Protótipo da Tela

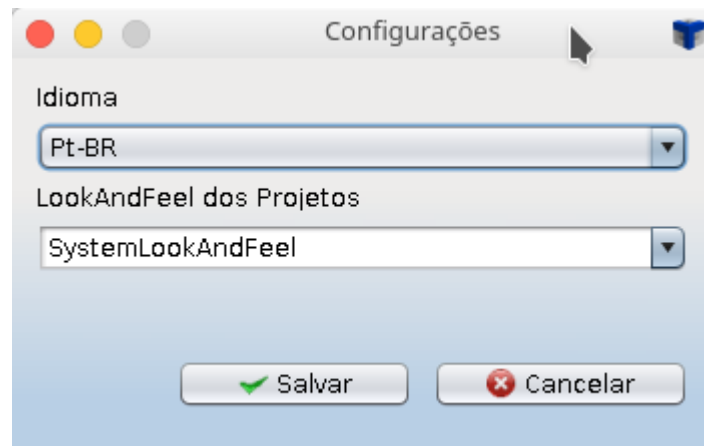


Figura LVI- Protótipo da Tela de Configurar sistema.

2.11.23 UC23 – Configurar Banco de Dados.

Tabela XXVIII - UC23 – Configurar Banco de Dados.

Objetivo	Configurar parâmetros de conexão ao banco de dados do projeto aberto.
Requisitos	RF23
Atores	Desenvolvedor
Prioridade	Média
Pré-condições	Haver um projeto aberto.
Frequência de Uso	Baixa
Pós-condições	Modificar aspectos de conexão ao banco de dados do projeto.
Campos	-
Fluxo principal	1. O Desenvolvedor clica com o botão direito do mouse no Gerenciador de Banco de Dados; 2. O Desenvolvedor seleciona a opção “Configurações de Banco de Dados”; 3. O Desenvolvedor altera os dados; 4. O Desenvolvedor salva as configurações;e 5. Encerra-se o fluxo principal.
Regras de negócio	Os termos \$server e \$database são utilizados como variáveis que serão preenchidas em tempo de execução.

2.11.23.1 DA23 – Diagrama de Atividades

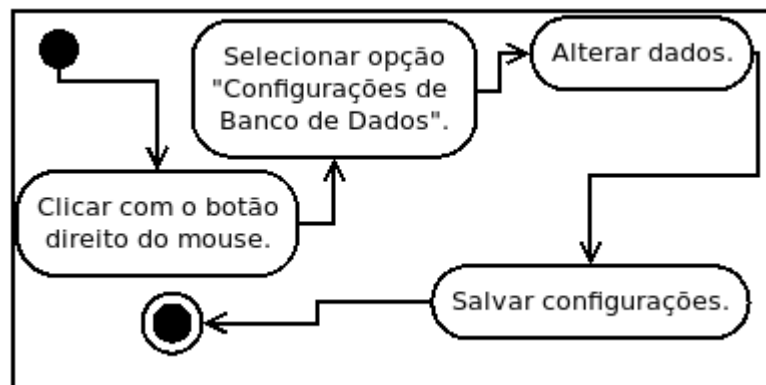


Figura LVII- DA23 - Diagrama de Atividades.

2.11.23.2 Protótipo da Tela

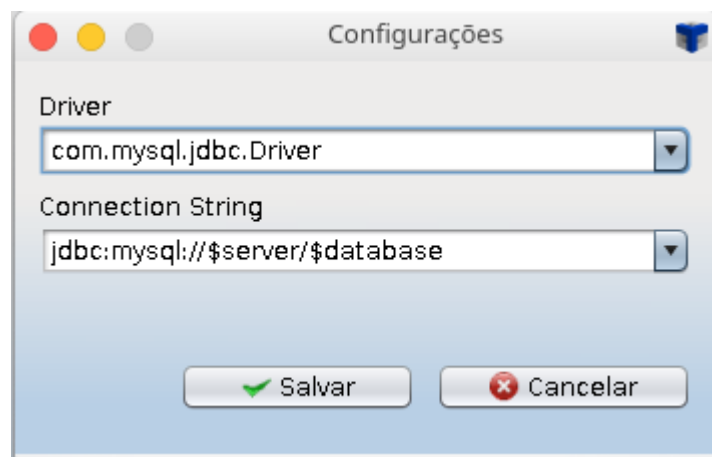


Figura LVIII- Protótipo da Tela de Configurar Banco de Dados.

2.12 Diagrama de Classes

Abaixo segue o Diagrama de Classes contendo as entidades manipuladas em um projeto do TetrisIDE.

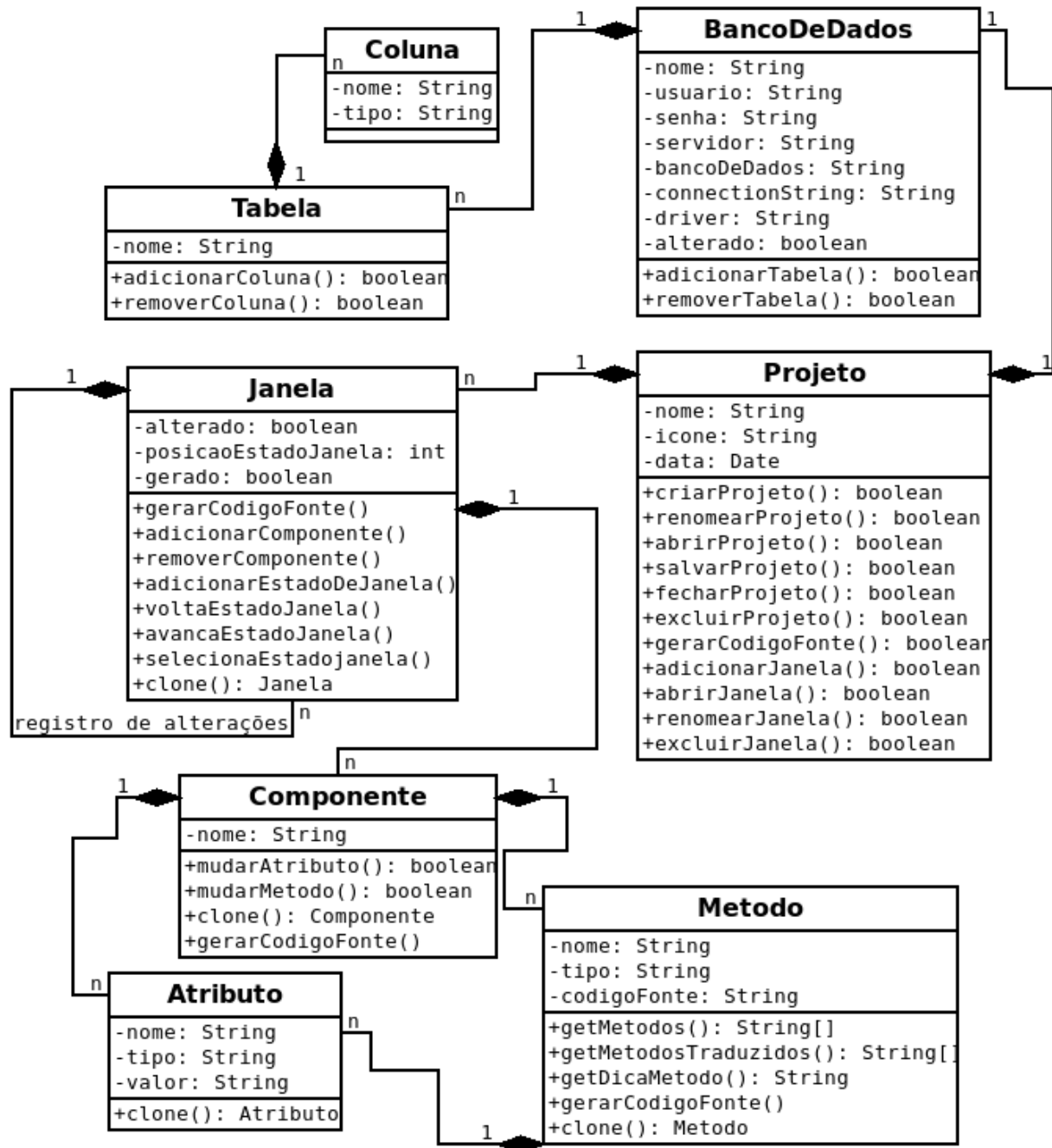


Figura LIX- Diagrama de Classes.