**COLÉGIO TORRICELLI**

**ARTHUR BARDEJO FRANCISCO**

**GABRIEL PEREIRA CASSANI**

**GUILHERME EL MOUALLEM DA SILVA**

**LUCAS JOSHUA AGIZ CARDOZO**

**YURI SOUSA OLIVEIRA**

**JOGO EDUCATIVO**

**Guarulhos - SP**

**2014**

**COLÉGIO TORRICELLI**

**ARTHUR BARDEJO FRANCISCO**

**GABRIEL PEREIRA CASSANI**

**GUILHERME EL MOUALLEM DA SILVA**

**LUCAS JOSHUA AGIZ CARDOZO**

**YURI SOUSA OLIVEIRA**

**JOGO EDUCATIVO**

Trabalho de Conclusão de Curso apresentado ao Colégio

Torricelli como parte dos requisitos para obtenção da

Habilitação Profissional de Técnico em Informática.

Professora orientadora: Patrícia Vitorasso.

Trabalho considerado concluído e aprovado em \_\_/\_\_/\_\_\_\_

----------------------------------------------------------------------------------------------------------

Jefferson Ripper

----------------------------------------------------------------------------------------------------------

Sergio Willianns Ponteli

----------------------------------------------------------------------------------------------------------

Patrícia Vitorasso

*Este trabalho é dedicado aos nossos professores,*

*entes queridos e a nós mesmos.*

*Agradecemos aos nossos pais pelo incentivo e o*

*apoio. Agradecemos a todos os componentes do*

*grupo pelo desenvolvimento do trabalho.*

*“O Fracasso é a oportunidade de recomeçar*

*com mais Inteligência.”*

*- Henry Ford*

**RESUMO**

O intuito do Trabalho de Conclusão de curso, foi desenvolver um jogo com intuito que ele seja educativo, no caso o jogo "Geo Quest Brasil", tem como intuito ensinar os usuários um pouco mais sobre a geografia do Brasil.

O jogo possui uma tela ensinando a jogá-lo, uma de menu principal, uma tela para cada fase, sendo que são 3 fases,telas de transição indicando a pontuação do usuário ao final de cada fase, uma tela de registro do usuário ao fim do jogo e uma de game over.

Somente é pedido o cadastro do usuário ao fim do jogo para que sua pontuação seja registrada no banco de dados, e o administrador não possui nenhum privilégio, pois não é necessário.

Índice

[Introdução 10](#_Toc402951992)

[PL01 - Definição do nome e da logomarca 11](#_Toc402951993)

[PL 02 - Pesquisa sobre o jogo 11](#_Toc402951994)

[PL 03 -Planejamento das funções do jogo 11](#_Toc402951995)

[PL 04 - Definição do nome do jogo e roteiro passo a passo 12](#_Toc402951996)

[PL 05 -Definição dos objetivos do jogo e personagens 12](#_Toc402951997)

[Cronograma da Primeira Fase do Jogo 13](#_Toc402951998)

[PL 06 –Diagramas 14](#_Toc402951999)

[Dicionário de Dados 20](#_Toc402952000)

[A 01 – Cronograma da fase de análise do jogo 20](#_Toc402952001)

[A 02 – Limites manuais e automatizados do jogo 20](#_Toc402952002)

[A 03 – Requisitos de segurança do jogo 21](#_Toc402952003)

[A 04 – Requisitos mínimos do jogo 21](#_Toc402952004)

[TELAS DO JOGO 22](#_Toc402952005)

[Tela Inicial - 22](#_Toc402952006)

[Ranking - 22](#_Toc402952007)

[Ajuda - 23](#_Toc402952008)

[Primeira Fase - 23](#_Toc402952009)

[Transição entre a primeira e a segunda fase - 24](#_Toc402952010)

[Segunda Fase- 24](#_Toc402952011)

[Transição entre segunda e terceira fase - 25](#_Toc402952012)

[Terceira Fase - 25](#_Toc402952013)

[Ranking da pontuação final - 26](#_Toc402952014)

[Tela de Game Over 26](#_Toc402952015)

[Imagens dos botões do jogo 27](#_Toc402952016)

[Botão de jogar 27](#_Toc402952017)

[Botão de Ranking 27](#_Toc402952018)

[Botão de Ajuda 27](#_Toc402952019)

[Botão de Sair 27](#_Toc402952020)

[Botão de Menu 28](#_Toc402952021)

[Botão de Repetir 28](#_Toc402952022)

[Botão de Próxima fase 28](#_Toc402952023)

[Botão de salvar 28](#_Toc402952024)

[Programação da Imagem do Jogo. 29](#_Toc402952025)

[Programação do Botão Sair. 33](#_Toc402952026)

[Programação das Imagens. 34](#_Toc402952027)

[Programação dos Botões. 34](#_Toc402952028)

[Programação da Validação 35](#_Toc402952029)

[Conclusões individuais 36](#_Toc402952030)

[Lucas Joshua Agiz Cardozo: 36](#_Toc402952031)

[Arthur Bardejo Francisco: 36](#_Toc402952032)

[Guilherme El Mouallem Da Silva: 36](#_Toc402952033)

[Yuri Sousa Oliveira: 36](#_Toc402952034)

[Gabriel Pereira Cassani: 36](#_Toc402952035)

# Introdução

Nesta parte do trabalho visamos esclarecer como definimos várias etapas de projeto, como os diagramas, cronogramas, as fases de planejamento do jogo, o mascote, o nome do jogo e as funcionalidades.

# PL01 - Definição do nome e da logomarca

O nome da empresa é Right World, pelo fato de estarmos fazendo um jogo educativo para crianças e de querermos passar a imagem de um mundo certo para as crianças que escolherem o nosso jogo.

A logo marca do jogo é feita de maneira para que pareçam rabiscos, com a intenção de que se pareça com que uma criança o fez.



# PL 02 - Pesquisa sobre o jogo

É um jogo destinado a todos que desejarem conhecer mais sobre a geografia e os estados brasileiros, foi feito com a intenção de aumentar o conhecimento sobre os estados de nosso país, para aqueles que jogarem o nosso jogo.

PL 03 -Planejamento das funções do jogo

Tem como função ensinar um pouco sobre a geografia de nossa nação.O jogo será totalmente controlado pelo mouse, com clicks em botões virtuais. botões: Iniciar, ajuda, sair e ranking\*. A acessibilidade do jogo será dada pelo movimento do (e somente ) mouse.

PL 04 - Definição do nome do jogo e roteiro passo a passo

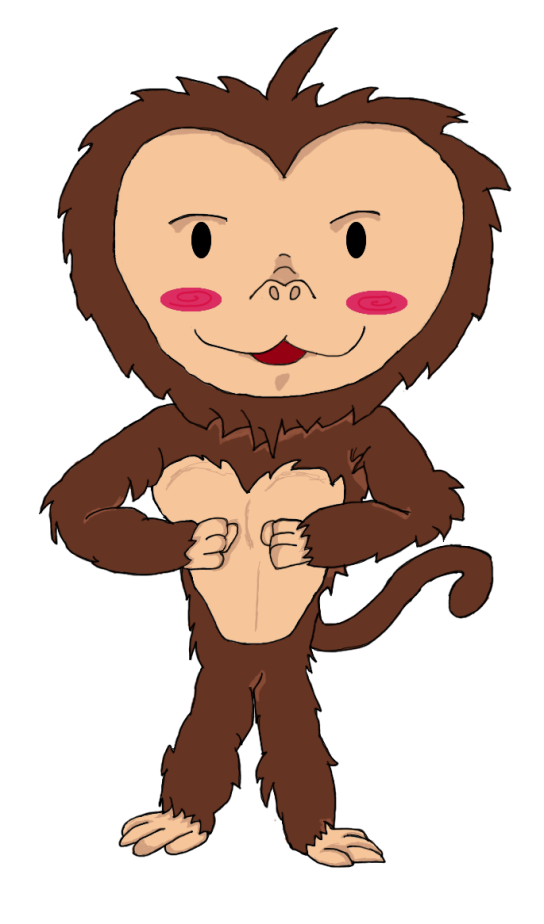
O jogo será constituído por 3 partes, na primeira parte o jogador deve selecionar e clicar o contorno do estado brasileiro a seu respectivo local, na segunda parte devera colocar as siglas correspondentes sobre os estados, e na terceira e última parte deverá arrastar as capitais de cada estado correspondente.



# PL 05 -Definição dos objetivos do jogo e personagens

Os objetivos do jogo são de passar pelas três fases com a maior pontuação possível.

Temos como mascote o Chico, um macaco, nosso mascote é meramente ilustrativo.



# Cronograma da Primeira Fase do Jogo

PL 01 – DEFINIÇÃO DO NOME E LOGOMARCA

Completo

PL 02 – PESQUISA SOBRE O JOGO

Completo

PL 03 – PLANEJAMENTO DAS FUNÇÕES DO JOGO

Completo

PL 04 DEFINIÇÃO DO NOME DO JOGO E ROTEIRO PASSO A PASSO

Completo

PL 05 – DEFINIÇÃO DO OBJETIVO DO JOGO E PERSONAGENS

Completo

CRONOGRAMA

Completo

PL 06 – DIAGRAMAS

Completo

PL 07 – DIAGRAMA DE CONTEXTO DE DADOS (DICIONARIO DE DADOS)

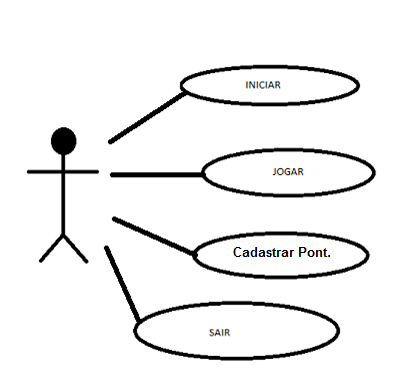
Completo

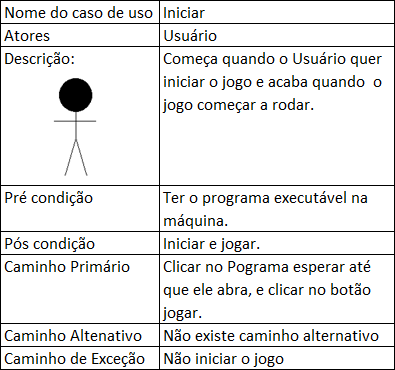
**TAREFA**

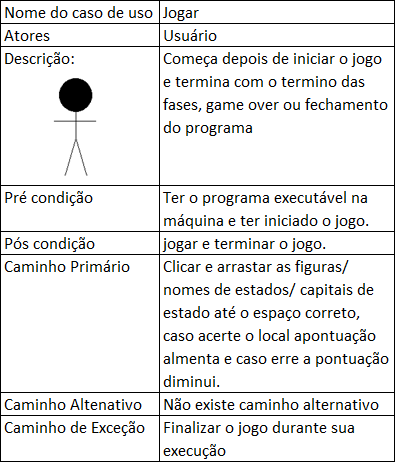
**STATUS**

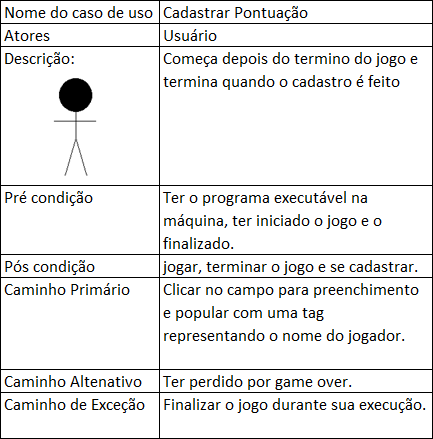
22/mai

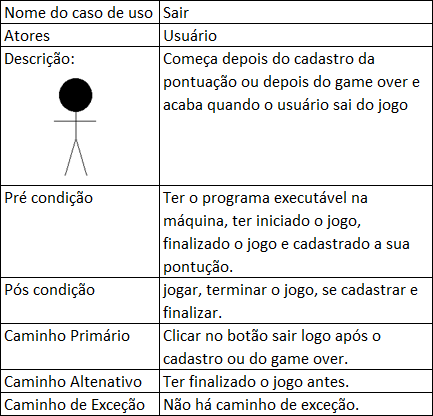
# PL 06 –Diagramas

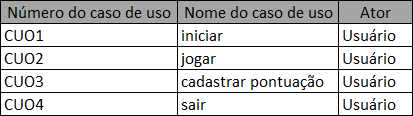


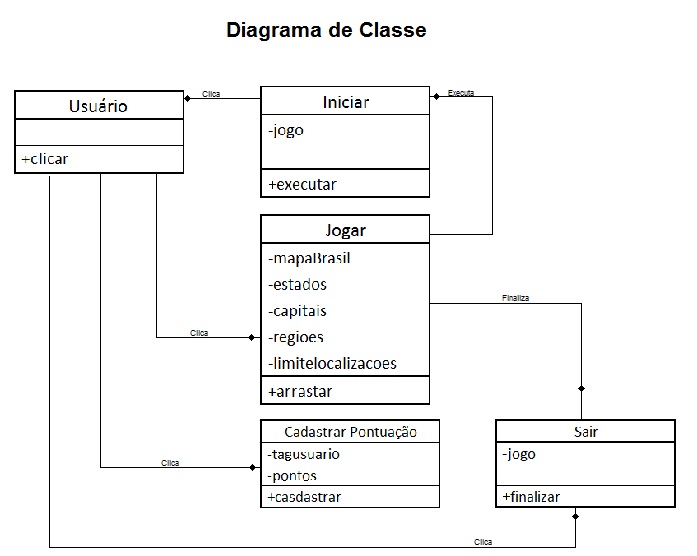




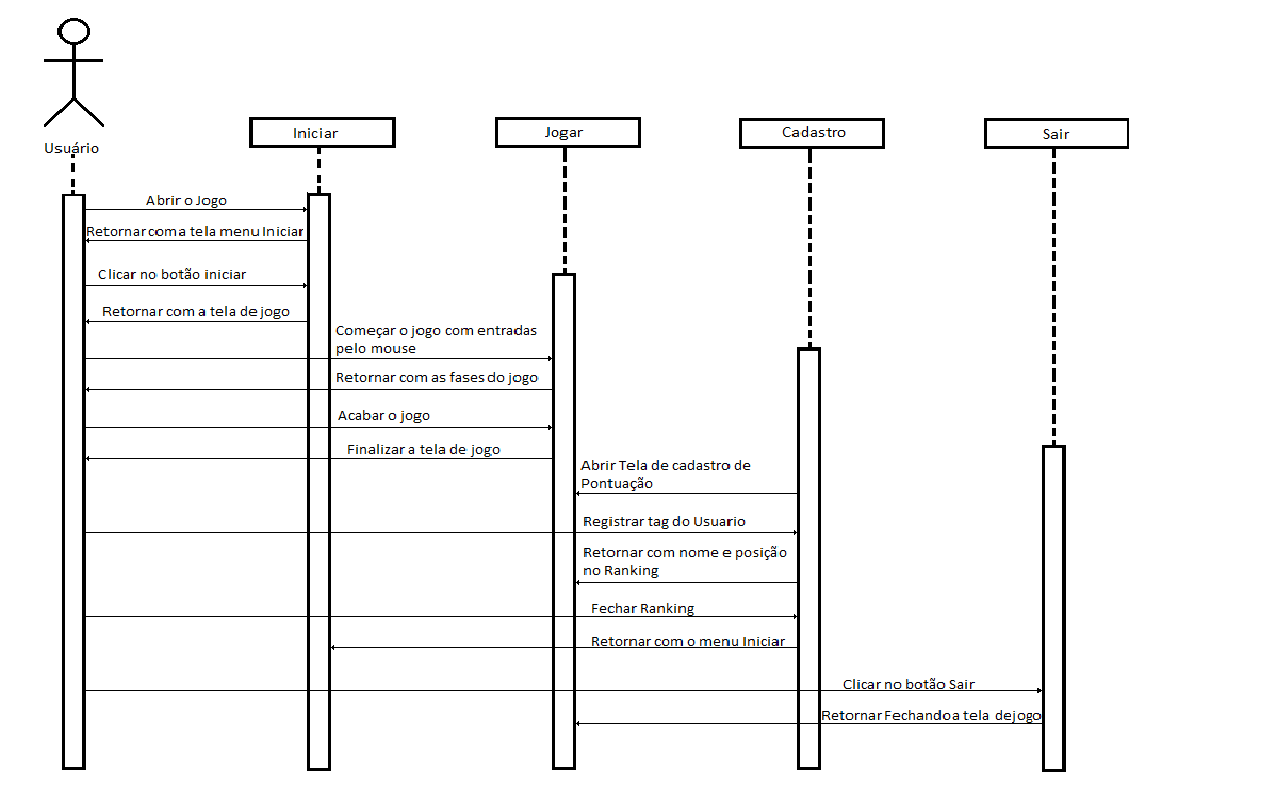


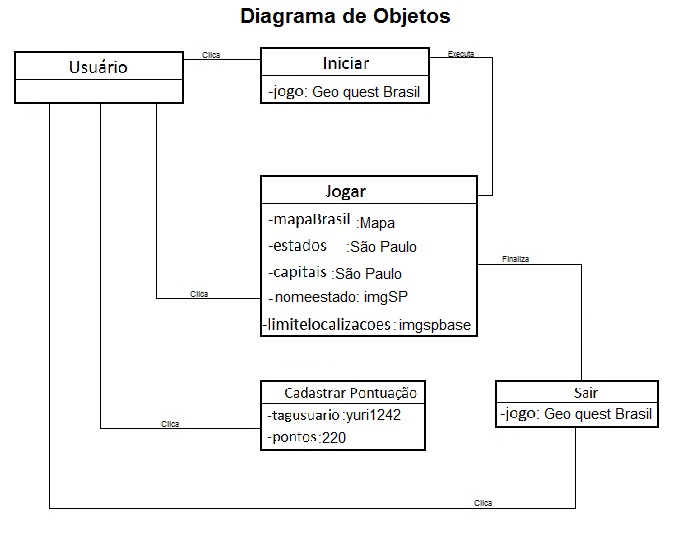


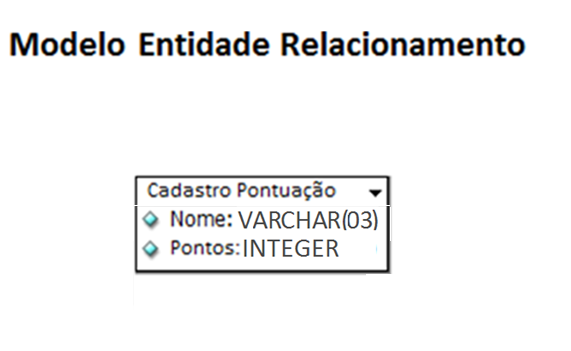




**Diagrama de Sequência**







# Dicionário de Dados

**Tabela: Cadastro Pontuação**

|  |  |  |  |
| --- | --- | --- | --- |
| CAMPO | TIPO | TAMANHO | DESCRIÇAO |
| Nome | Varchar | 03 | Nome do Usuario |
| Pontos | INTEGER | Inteiro Longo | Pontos do Usuario |

Resumo: Essa tabela armazena os dados de Pontuação e nome do usuário no ranking do jogo.

# A 01 – Cronograma da fase de análise do jogo



# A 02 – Limites manuais e automatizados do jogo

Dentro do nosso jogo o usuário somente terá controle sobre o mouse, o qual ele utilizará para clicar sobre as opções e arrastar até a respectiva parte do mapa.

# A 03 – Requisitos de segurança do jogo

Requisitos de Segurança

Não tem requisito algum de segurança, o único requisito é o campo que precisa ser preenchido ao concluir o jogo para que sua pontuação possa ser salva.

# A 04 – Requisitos mínimos do jogo

Requisitos de Hardware e Software

Windows XP ou posterior, com JRE 1.6 ou posterior, ou outros sistemas operacionais do MAC ou Linux,  2GB de Ram, 250MB de espaço no HD, conexão com a internet, computador, mouse, monitor .

# TELAS DO JOGO

## Tela Inicial -



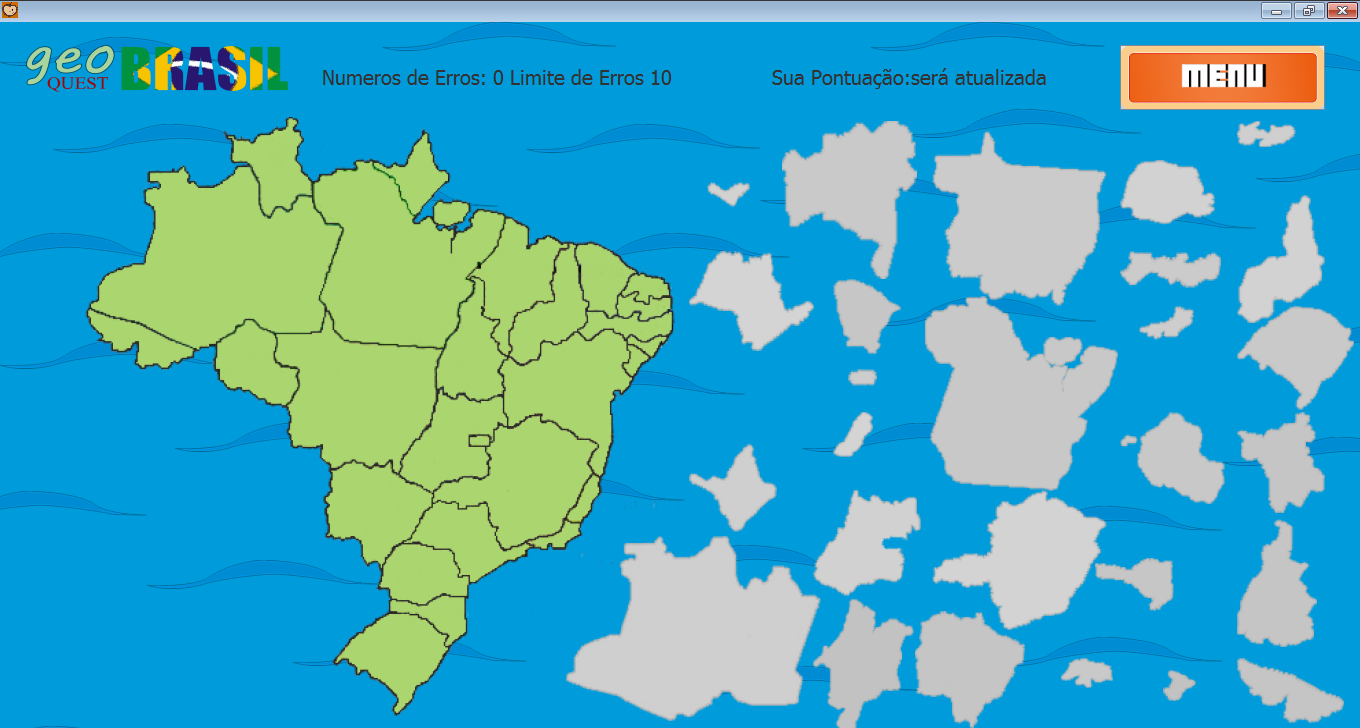
## Ranking -



## Ajuda -



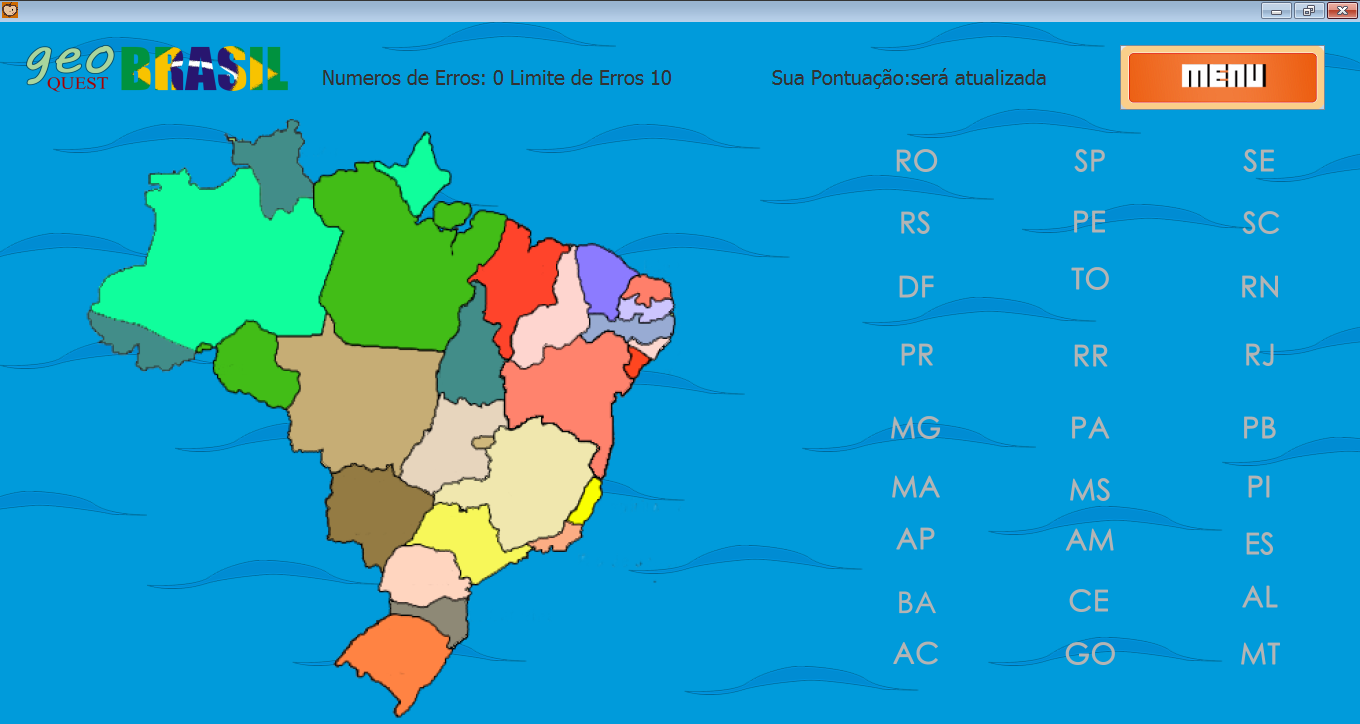
## Primeira Fase -



## Transição entre a primeira e a segunda fase -



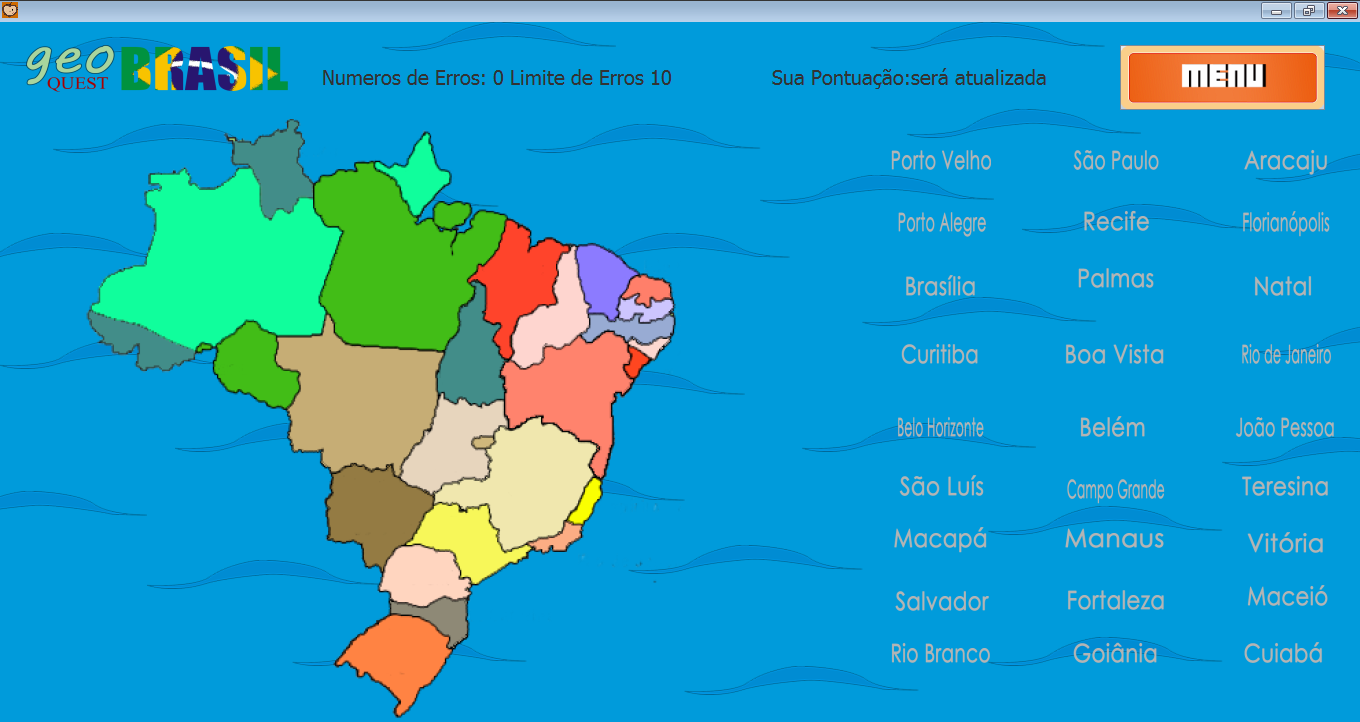
## Segunda Fase-



## Transição entre segunda e terceira fase -



## Terceira Fase -



## Ranking da pontuação final -



## Tela de Game Over



# Imagens dos botões do jogo

## Botão de jogar

Inicia o jogo.



## Botão de Ranking

Mostra o ranking com as pontuações.



## Botão de Ajuda

Ensina o jogador a passar por cada estágio do jogo.



## Botão de Sair

Sai do jogo.



## Botão de Menu

Leva o jogador ao menu principal.



## Botão de Repetir

Volta à primeira fase.



## Botão de Próxima fase

Leva o jogador à próxima fase.



## Botão de salvar

Salva a pontuação do usuário e retorna ao menu.



# Programação da Imagem do Jogo.

A função do código da imagem do jogo e fazer a inserção da imagem dando a ela a ação de mover e parar quando ele realizar a ação de colocar o objeto no lugar certo ela some e é exibida uma outra imagem, no caso o "Acre2", quando intersectar o "Acre3", o "Acre2" some e o "Acre4"é encaixado em seu devido lugar, assim que essa ação é realizada o placar também muda somando 10 no caso do acerto e subtraindo 10 no caso de erro, ocorre um erro quando o objeto não intersecta o outro objeto certo, com isso ele retorna ao seu local de origem, subtraindo 10 pontos e fazendo com que o usuário tenha - 1 chance de erro.

Quando a Pontuação do Jogador chega a 270 pontos, o jogador passa de fase, quando os erros do jogador chegam a 10 a tela de game over é exibida informando o fim de jogo.

Esse comando é repetido 81 vezes na programação do jogo, todos fazendo a mesma função, só mudando os locais de origem e seus nomes.

|  |
| --- |
| **final** JLabel Acre4 = **new** JLabel("");  Acre4.setBounds(86, 284, 107, 65);  ImageIcon imagem970 =**new** ImageIcon (Jogo.**class**.getResource("/Estados/Acre.png"));  Image acress= imagem970.getImage().getScaledInstance(Acre4.getWidth(),Acre4.getHeight() , Image.*SCALE\_SMOOTH*);  Acre4.setIcon(**new** ImageIcon(acress));  Acre4.setVisible(**false**);  contentPane.add(Acre4);    **final** JLabel Acre3 = **new** JLabel("");  Acre3.setBounds(119, 313, 32, 1);  ImageIcon imagem97 =**new** ImageIcon (Jogo.**class**.getResource("/Estados/Acre.png"));  Image acres= imagem97.getImage().getScaledInstance(Acre3.getWidth(),Acre3.getHeight() , Image.*SCALE\_SMOOTH*);  Acre3.setIcon(**new** ImageIcon(acres));  Acre3.setVisible(**false**);  contentPane.add(Acre3);    **final** JLabel Acre2 = **new** JLabel("");  Acre2.addMouseMotionListener(**new** MouseMotionAdapter() {  @Override  **public** **void** mouseDragged(MouseEvent arg0) {  Acre2.setEnabled(**true**);    Component C=arg0.getComponent();    **if**(b1==**false**){b1=**true**;  Point p=**new**  Point(arg0.getPoint());c1=p.x;c2=p.y;}  Point z=**new** Point(arg0.getPoint());  Point q=**new** Point(C.getLocation());  C.setBounds(q.x+(z.x-c1),q.y+(z.y-c2),C.getSize(). width,C.getSize().height);    }  }  );  Acre2.addMouseListener(**new** MouseAdapter() {  @Override  **public** **void** mousePressed(MouseEvent arg0) {    Acre2.setEnabled(**true**);  }  @Override  **public** **void** mouseReleased(MouseEvent arg0) {  b1=**false**;  Acre2.setEnabled(**false**);    Rectangle ac1 = Acre2.getBounds();  Rectangle ac2= Acre3.getBounds();    **if**(ac1.intersects(ac2)){  pontos+=10;  pontoss+=10;  lblponto.setText("Sua Pontuação é: "+String.*valueOf*(pontoss));  Acre4.setVisible(**true**);  Acre2.setVisible(**false**);    **try** {  URL url = **this**.getClass().getClassLoader().getResource("Audio/7.wav");  AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(url);    Clip clip = AudioSystem.*getClip*();    clip.open(audioIn);  clip.start();  } **catch** (UnsupportedAudioFileException e) {  e.printStackTrace();  } **catch** (IOException e) {  e.printStackTrace();  } **catch** (LineUnavailableException e) {  e.printStackTrace();  }  }      **else**{  ponto+=1;  pontoss-=10;  lblponto.setText("Sua Pontuação é: "+String.*valueOf*(pontoss));  lblerro.setText("Numeros de Erros: "+String.*valueOf*(ponto)+" Limite de Erros 10");  Acre2.setBounds(1237, 636, 110, 65);  **try** {    URL url = **this**.getClass().getClassLoader().getResource("Audio/3.wav");  AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(url);    Clip clip = AudioSystem.*getClip*();  clip.open(audioIn);  clip.start();  } **catch** (UnsupportedAudioFileException e) {  e.printStackTrace();  } **catch** (IOException e) {  e.printStackTrace();  } **catch** (LineUnavailableException e) {  e.printStackTrace();  }      }  **if**(pontos==270){  proxima frame = **new** proxima();  frame.txtpont.setText(String.*valueOf*(pontoss));  frame.setVisible(**true**);  frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*);  dispose();          **try** {    URL url = **this**.getClass().getClassLoader().getResource("Audio/5.wav");  AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(url);    Clip clip = AudioSystem.*getClip*();    clip.open(audioIn);  clip.start();  } **catch** (UnsupportedAudioFileException e1) {  e1.printStackTrace();  } **catch** (IOException e1) {  e1.printStackTrace();  } **catch** (LineUnavailableException e1) {  e1.printStackTrace();  }  }  **if**(ponto==10){  Game frame = **new** Game();  frame.setVisible(**true**);  frame.setExtendedState(JFrame.*MAXIMIZED\_BOTH*);  dispose();  **try** {    URL url = **this**.getClass().getClassLoader().getResource("Audio/Erro.wav");  AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(url);    Clip clip = AudioSystem.*getClip*();    clip.open(audioIn);  clip.start();  } **catch** (UnsupportedAudioFileException e) {  e.printStackTrace();  } **catch** (IOException e) {  e.printStackTrace();  } **catch** (LineUnavailableException e) {  e.printStackTrace();  }  }    }  @Override  **public** **void** mouseEntered(MouseEvent arg0) {  Cursor c1=**new** Cursor(Cursor.*HAND\_CURSOR*);  setCursor(c1);  }  @Override  **public** **void** mouseExited(MouseEvent e) {  Cursor c2=**new** Cursor(Cursor.*DEFAULT\_CURSOR*);  setCursor(c2);  }  });  Acre2.setEnabled(**false**);  Acre2.setBounds(1237, 636, 107, 65);  ImageIcon imagem1 =**new** ImageIcon (Jogo.**class**.getResource("/Estados/Acre.png"));  Image Acre= imagem1.getImage().getScaledInstance(Acre2.getWidth(),Acre2.getHeight() , Image.*SCALE\_SMOOTH*);  Acre2.setIcon(**new** ImageIcon(Acre));  contentPane.add(Acre2); |

# Programação do Botão Sair.

O botão Sair tem como única função sair do jogo, quando ele realiza essa função exibida um som e o jogo é fechado.

|  |
| --- |
| JButton btnsair = **new** JButton("");  btnsair.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent arg0) {  dispose();  **try** {    URL url = **this**.getClass().getClassLoader().getResource("Audio/9.wav");  AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(url);    Clip clip = AudioSystem.*getClip*();    clip.open(audioIn);  clip.start();  } **catch** (UnsupportedAudioFileException e1) {  e1.printStackTrace();  } **catch** (IOException e1) {  e1.printStackTrace();  } **catch** (LineUnavailableException e1) {  e1.printStackTrace();  }  }  });  btnsair.setBounds(613, 429, 205, 65);  ImageIcon imagens4 =**new** ImageIcon (Jogo.**class**.getResource("/Img/Bot3.png"));  Image botao4= imagens4.getImage().getScaledInstance(btnsair .getWidth(),btnsair .getHeight() , Image.*SCALE\_SMOOTH*);  btnsair .setIcon(**new** ImageIcon(botao4));  contentPane.add(btnsair); |

# Programação das Imagens.

A programação de todas as imagens é sempre a mesma, com código para que o programador possa alterar o tamanho do objeto direto no Designer.

|  |
| --- |
| JLabel lblmapa4 = **new** JLabel("");  lblmapa4.setBounds(1046, 382, 234, 334);  ImageIcon imagens =**new** ImageIcon (Jogo.**class**.getResource("/Img/Mascote colorido.png"));  Image mapa4= imagens.getImage().getScaledInstance(lblmapa4.getWidth(),lblmapa4.getHeight() , Image.*SCALE\_SMOOTH*);  lblmapa4.setIcon(**new** ImageIcon(mapa4));  contentPane.add(lblmapa4); |

# Programação dos Botões.

A Programação do botão também é sempre a mesma , a diferença que a ação dela que muda, abrindo diferentes frames.

|  |
| --- |
| JButton btnajuda = **new** JButton("");  btnajuda.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent arg0) {  Ajuda pag3 = **new** Ajuda();  pag3.setVisible(**true**);  pag3.setExtendedState(JFrame.*MAXIMIZED\_BOTH*);  dispose();  **try** {  URL url = **this**.getClass().getClassLoader().getResource("Audio/9.wav");  AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(url);  Clip clip = AudioSystem.*getClip*();  clip.open(audioIn);  clip.start();  } **catch** (UnsupportedAudioFileException e1) {  e1.printStackTrace();  } **catch** (IOException e1) {  e1.printStackTrace();  } **catch** (LineUnavailableException e1) {  e1.printStackTrace();  }  }  });  btnajuda.setBounds(613, 353, 205, 65);  ImageIcon imagens3 =**new** ImageIcon (Jogo.**class**.getResource("/Img/Bot4.png"));  Image botao3= imagens3.getImage().getScaledInstance(btnajuda.getWidth(),btnajuda.getHeight() , Image.*SCALE\_SMOOTH*);  btnajuda.setIcon(**new** ImageIcon(botao3));  contentPane.add(btnajuda); |

# Programação para limitar Texto

A Programação do limitar Texto, tem a função de limitar um texto, dentro de TextField (Caixa de Texto.

|  |
| --- |
| **import** javax.swing.text.\*;  **public** **class** LimitadorTexto **extends** PlainDocument {  **private** **int** limite = 0;  **public** LimitadorTexto(**int** l){  **this**.limite = l;  }  **public** **void** insertString(**int** offs, String str, AttributeSet a)  **throws** BadLocationException {  **int** sobra = limite - getLength();  **int** comprimento = ( sobra > str.length() ) ? str.length() : sobra;  **super**.insertString(offs, str.substring(0, comprimento), a);  }  } |

Programação de Ranking do Botão Salvar

A Programação do Botão Salvar, tem a função de salvar a pontuação do Jogador, permitindo-lhe inserir 3 dígitos como nome do Usuário, ele enviará a pontuação para a Frame Ranking.

|  |
| --- |
| JButton btnsalvar = **new** JButton("");  btnsalvar.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent arg0) {  String connectionurl="jdbc:sqlserver://10.0.0.252:1433;databaseName=bancoteste";  **try**{  Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver").newInstance();  Connection con DriverManager.*getConnection*(connectionurl,"aluno","#lab\*01");  System.*out*.println("conectado");  String g="INSERT INTO Rank (nome,pontos) VALUES(?,?)";  java.sql.PreparedStatement stmt=con.prepareStatement(g);  stmt.setString(1,txtnome.getText());  stmt.setString(2,txtpont.getText());  stmt.executeUpdate();  stmt.close();  con.close();  }**catch**(SQLException ex){  System.*out*.println(ex.getMessage());  }**catch**(Exception e){  System.*out*.println("não conectou"+e.getMessage());  }  **try** {  URL url = **this**.getClass().getClassLoader().getResource("Audio/9.wav");  AudioInputStream audioIn = AudioSystem.*getAudioInputStream*(url);  Clip clip = AudioSystem.*getClip*();  clip.open(audioIn);  clip.start();  } **catch** (UnsupportedAudioFileException e1) {  e1.printStackTrace();  } **catch** (IOException e1) {  e1.printStackTrace();  } **catch** (LineUnavailableException e1) {  e1.printStackTrace();  }  Ranking el=**new** Ranking();  el.setVisible(**true**);  el.setExtendedState(JFrame.*MAXIMIZED\_BOTH*);  dispose();  }}); |

# Programação do Ranking

A programação do Ranking, tem a função de receber o valor enviado da Frame Final, e colocar em ordem as 10 primeiros usuários que possuem as melhores pontuações.

|  |
| --- |
| **try**{  Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");  String connectionurl="jdbc:sqlserver://10.0.0.252:1433;databaseName=bancoteste";  Connection con = DriverManager.*getConnection*(connectionurl,"aluno","#lab\*01");    java.sql.Statement stmt =con.createStatement();  ResultSet rs=stmt.executeQuery("SELECT \* FROM Rank ORDER BY pontos DESC;;");  **while**(rs.next()){  String nome=rs.getString("nome");  String pontos=rs.getString("pontos");  nomes1.setText(nome+"");  pontos1.setText(pontos+"");  String ju = pontos1.getText();  String je = nomes1.getText();  String res = je.toUpperCase();  el.append(ju);  yu.append(res);  }  }**catch**(Exception eX){  System.*out*.println("não conectou "+eX.getMessage());  } |

# Conclusões individuais

## Lucas Joshua Agiz Cardozo:

Com este trabalho pude aprender a importância do trabalho em equipe para que as coisas fluam bem, aprendi a ter perseverança e sempre continuar em frente. Não deixar que os erros nos atrapalhem e sim aprender com eles para que se alcance a perfeição baseado no melhor que cada um de nos pode dar.

## Arthur Bardejo Francisco:

Com este trabalho consegui aprender que um grupo não precisa de um líder desde que todos se cobrem, aprendi a trabalhar de uma maneira melhor em grupo e a me relacionar com as pessoas por mais que as diferenças permaneçam no caminho, existe apenas a necessidade de trabalhar em conjunto para se aprimorar de maneira geral, todos são proficientes em algo, basta buscarmos dar o nosso melhor em tudo o que pudermos para ajudarmos o grupo.

## Guilherme El Mouallem Da Silva:

Foi uma experiência muito boa trabalhar com colegas que se esforçaram cada dia para fazer esse trabalho dar certo. Aprendemos a trabalhar em equipe e passamos por todos os desafios que foram impostos a nós. Estou muito satisfeito com o empenho de todos.

## Yuri Sousa Oliveira:

Ao termino do trabalho, pude aprender a como trabalhar em grupo e assim garantindo mais responsabilidade com o integrantes do mesmo, ajudando em todas as suas etapas e realizando tarefas que jamais pensei em fazer, aprendi também a confiar nos companheiros de equipe, pois sem eles não chegaríamos à conclusão deste trabalho.

## Gabriel Pereira Cassani:

Conclui-se que um trabalho de conclusão de curso, não é um simples trabalho para nota, ele serve para testar todas as competências exploradas pelo curso, e também avalia a capacidade do aluno desenvolver um produto para o mercado de trabalho da área, ele simula quase sem querer as dificuldades que o aluno irá passar ao decorrer da sua carreira, e também o ensina a trabalhar em equipe. O TCC é a meu ver um simulador de carreira, e é muito importante para que possamos escolher se ficaremos ou não na área de informática.