



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAMPINA GRANDE
CURSO TÉCNICO EM INFORMÁTICA INTEGRADO

JÔNATAS TAVARES DOS SANTOS
LUCAS SILVA NASCIMENTO
MARIA GABRIELA PEREIRA DA SILVA
MARIA LETÍCIA SILVA MENDES

GRUPO 7 - TIC TAC TOE
RELATÓRIO FINAL

CAMPINA GRANDE - PB

2019

SUMÁRIO

1 CRONOGRAMA	2
2 O PROJETO	3
2.1 CLASSES DO SISTEMA	3
2.2 PERSISTÊNCIA DE DADOS	5
2.3 FUNCIONALIDADES EXCLUÍDAS	5
2.4 INTERFACE GRÁFICA	6
3 DIVISÃO DE TAREFAS	7
3 DIAGRAMA UML	11

1 CRONOGRAMA

G7 - TIC TAC TOE		
DATA	AÇÃO	ESTADO
04/09/2019	Prazo final para a entrega do cronograma	Concluído
12/09/2019	Classe Jogador e Partida (Atributos, gets e sets)	Concluído
03/19/2019	Código para um jogo contra o computador (sem interface)	Concluído
10/10/2019	Código para um jogo contra outro jogador (sem interface)	Concluído
24/10/2019	Função cadastrar	Concluído
07/11/2019	Funções: recuperar senha, listar jogadores, excluir usuário e autenticar	Concluído
21/11/2019	Funções: buscar jogadores por nome, manter pontuação, exibir ranking, manter histórico, listar partidas, imprimir relatório de partidas	Concluído
27/11/2019	Prazo final para a entrega do projeto	Concluído
28/11/2019	Apresentação final do projeto	Em andamento

2 O PROJETO

Tic Tac Toe, em português, Jogo da Velha, é um jogo de dois jogadores que possui regras extremamente simples. Durante o jogo, os participantes revezam sua vez de jogar, representando com “X” e “O”, as posições jogadas em um tabuleiro 3 x 3. Objetiva-se formar uma sequência de três jogadas de um mesmo símbolo em uma mesma linha, coluna, ou em diagonal, ao mesmo tempo em que busca impedir que o adversário faça o mesmo. O jogador que consegue fazer isso primeiro, ganha o jogo, caso não existam mais jogadas possíveis e nenhum dos jogadores tenha atingido o objetivo, a partida é declarada empatada.

O projeto consiste no desenvolvimento de um sistema que proporciona ao usuário que ele jogue uma partida contra outro jogador ou o Computador. Para isso o usuário deve possuir um cadastro no sistema possuindo nome de usuário e senha, dessa forma, ao ser autenticado no sistema, ele pode ter acesso aos dois modos de jogo; a lista com todos os jogadores cadastrados no sistema, e seus respectivos pontos; o ranking desses jogadores por pontuação ou número de vitórias; aos dados de sua conta, podendo alterá-los ou excluir sua conta. O sistema é composto por classes, dentre elas: Central, Jogador, Partida e Tabuleiro.

2.1 CLASSES DO SISTEMA

A classe Jogador representa os vários usuários contidos no sistema, portanto ela possui como atributos login, senha, dataCadastro, vitoria e pontos. Ela contém um Construtor, hashCode, ToEquals, ToString, getters e setters referentes aos atributos presentes.

A classe Tabuleiro representa o tabuleiro do jogo, usando uma matriz 3x3 para tal, além disso, ela possui uma lista de posições, que armazena todas as posições já ocupadas, um atributo chamado “posição” que representa a posição jogada no momento, e um atributo chamado “parar”, que serve para encerrar a partida quando o jogo chegar ao fim. A classe possui além de um Construtor, hashCode, ToEquals, ToString, getters e setters, ela também possui as funções: “zerarTabuleiro” que percorre o tabuleiro limpando ao colocar um espaço em branco em todas as suas posições; “zerarPosições” que percorre a lista de posições zerando-a para uma nova partida; “jogada1” que realiza a jogada do primeiro jogador (usuário cadastrado), ela verifica a posição passada por parâmetro vinda da Interface, e inserindo-a na posição equivalente no tabuleiro; “jogada2” que realiza a jogada do Computador, a partir do mesmo princípio da jogada1, entretanto não há posição passada como parâmetro, pois ela é

sorteada aleatoriamente na função; a “jogada3” é similar à jogada1 mudando apenas o símbolo inserido, já que ela representa o jogador2, na opção de dois jogadores; “verificarVitoria”, essa função através de um aninhamento de if’s verifica todas as possibilidades de haver uma vitória; e “verificarEmpate” que verifica se há empate através de um loop.

A classe Partida contém como atributos jogador1, jogador2, vencedor, atual, tabuleiro (é uma instância à um objeto do tipo Jogador), modo que simboliza se o jogo será contra um jogador (true) ou computador (false); possui Construtor, hashCode, ToEquals, ToString, getters e setters, além disso contém as funções: “zerar” que zera a lista de posições e limpa o tabuleiro para uma nova partida; a função “jogada1” chama a função jogada1 do Tabuleiro; assim como as funções jogada2 e jogada3 chamam as respectivas funções do Tabuleiro; “jogou” verifica se na posição passada como parâmetro já está preenchida; “parar” que retorna o valor do getParar contido no Tabuleiro; “fimJogada” que verifica se houve vitória ou empate, atribuindo os pontos adquiridos pelo jogador, além disso ela setta um novo valor à variável parar, para que não haja uma próxima jogada;

A classe Central comunica-se com a Interface, ela apresenta praticamente todas as funcionalidades tidas no projeto. Ela é uma classe do tipo static que contém uma partida, um jogador1, um jogador2 e uma lista de Jogadores que vai estar ligada com o Banco de Dados, a qual vai receber todos os jogadores existentes no Banco de Dados; juntamente com seus getters e setters. A classe contém também as funções: “login” que percorre a lista de jogadores procurando o usuário, e verificando se os seu nome e senha são compatíveis com o cadastrado no sistema; a função “login2” tem a mesma base da função de login, entretanto ela se refere ao jogador2, quando ele vai autenticar para o jogo contra dois jogadores; a função “cadastrar” cadastra um novo jogador no sistema criado na função “criarJogador”, desde que ele não exista no sistema e possua senha com no mínimo 6 caracteres; “removerJogador” que exclui a conta do jogador do banco de dados; “rankingPontos” busca os três primeiros colocados no ranking de acordo com os pontos através de três loops, o qual cada um procura o jogador da lista com maior pontuação, exceto o que já foi escolhido como primeiro e segundo; a função “rankingVitorias” segue a mesma logica da função anterior, só que ao invés de pontos, utiliza o critério de quantidade de vitórias; “iniciar” atualiza o valor do jogador atual na partida, zera o tabuleiro e a lista de posições, chamando a função zerar da partida;

“iniciar2” funciona do mesmo modo da iniciar, só que para o modo de dos jogadores, dessa forma, ele atualiza também o valor do jogador2; “editarUsuario” atualiza o valor do novo nome do usuário logado, e a função “editarSenha” atualiza o valor da nova senha do usuário logado; as funções de “jogada1”, “jogada2” e “jogada3” chamam as funções correspondentes na Partida, levando em conta que a “jogada1” faz a distinção dos dois modos de jogo, para que a atualização do jogador atual fosse feita de forma distinta, e a “jogada2” não atualiza o valor atual como nas outras duas; “verificarGanhou” que chama a função de fimJogada da partida; e as funções “jogou”, “parar”, e “atual” chamam as suas respectivas funções da classe Partida. Vale salientar que por interligarmos a interface apenas com a Central algumas funções que precisávamos chamar na interface que estavam contidas em outras classes, foram criadas funções na Central que chamavam a próxima função respectiva, até que se chegasse à função desejada, como foi o caso de, por exemplo, as funções: “jogou”, “parar”, etc.

2.2 PERSISTÊNCIA DE DADOS

A classe BancoDeDados desempenha a função de ler e escrever os dados dos jogadores em seu respectivo arquivo XML. Para facilitar o processo de serialização e deserialização dos objetos, utilizamos a biblioteca XStream. Contém duas funções estáticas que são acessadas pela classe Central na inicialização e atualização dos valores. A função “lerArquivo”, ao inicializar a execução do programa, irá buscar todos os jogadores salvos no XML e retorná-los a Central, caso o arquivo seja inexistente ele gerará um novo arquivo, através da chamada da função “gerarArquivo”. Esta, irá receber como parâmetro um vetor ArrayList de Jogadores, e salvará os dados atualizados no arquivo, ela é repetida a cada movimentação na conta do usuário de forma a manter os dados atualizados.

2.3 FUNCIONALIDADES EXCLUÍDAS

Diante do projeto inicial proposto algumas funcionalidades foram excluídas; inicialmente tivemos a oportunidade de excluir três funções do nosso projeto, foram elas: jogar contra o computador no nível difícil, enviar mensagem para outro jogador e ler mensagens enviadas por outro jogador. Após a mudança de professora foram excluídas as funções: recuperar senha, manter histórico de partidas, listar partidas, imprimir relatório de partidas, buscar jogador por nome, e manter o arquivo armazenando informações na nuvem.

2.4 INTERFACE GRÁFICA

A interface gráfica do projeto foi feita para plataforma desktop utilizando Swing. Escolhemos um tom de azul mais claro para nos servir de tema base, também utilizamos alguns desenhos bem simples para não passar uma imagem tão pesada e cansativa para os olhos dos usuários do jogo. Ao todo temos 12 telas igualmente planejadas e priorizando a navegação entre elas.

Ao executar o programa, o usuário vai se deparar com uma tela onde poderá realizar seu login, ou criar uma nova conta, caso este ainda não possua uma. Ao clicar no botão de criar uma nova conta será solicitado ao usuário um nome de usuário único, uma senha de no mínimo 6 caracteres, e será pedido que ele repita a senha digitada anteriormente. Caso o nome de usuário não exista em nosso banco de dados e as senhas digitadas correspondam, será exibida a mensagem de conta criada com sucesso, caso contrário, o usuário será auxiliado para preencher os campos corretamente.

Imediatamente após a criação da conta o usuário poderá entrar no sistema digitando o nome de usuário e senha corretamente nos campos correspondentes. Uma tela de menu será exibida. Na barra de menu superior, o usuário poderá conferir a listagem de todos os usuários cadastrados no sistema; verificar dados de sua conta, podendo editá-los clicando no ícone de lápis na lateral direita da tela; excluí-la permanentemente; e sair da conta.

O usuário poderá verificar o ranking dos 3 primeiros jogadores, ordenados por pontuação e também por número de vitórias. Uma tela de instruções a respeito do jogo também poderá ser consultada pelo usuário a partir desta mesma tela. Para não gerar dúvidas e complicações na vida do usuário, os botões estão dispostos de forma simples, direta e de fácil acesso.

O botão *1 Jogador* inicia uma partida entre o jogador e o computador. Ao ser clicado, uma tela com um tabuleiro será aberta, e o jogador inicia a partida escolhendo onde o seu ícone (“X”) deverá ser posicionado, e logo após, automaticamente, o computador fará sua jogada de forma aleatória utilizando do outro ícone (“O”). Caso vença, o jogador ganhará 1 ponto; caso empate, o jogador ganhará meio ponto; caso perca, o jogador perderá meio ponto.

Já o botão *2 Jogadores*, após ser clicado, abrirá uma segunda tela de login, onde serão pedidas as credenciais do oponente. Após os dois estarem logados o jogo iniciará, sendo o

primeiro a jogar, o jogador que está com login “mestre”, utilizando do ícone “X”, e em revezamento, o segundo jogador com o ícone “O”. O tabuleiro é exibido até que alguém vença a partida, até que ela seja declarada empatada ou até o usuário fechar a aba, encerrando a partida. O jogador que vencer ganha 4 pontos; o jogador que perder perde 4 pontos; em caso de empate ninguém ganha ponto.

Casos de erro e funcionalidades adicionais, como por exemplo permitir que o usuário jogue novamente, estão sendo tratadas em todas as telas. Para o funcionamento de todas essas telas, elas irão usar funções das classes Central, e de suas ramificações com as demais classes do sistema, para fazer as alterações necessárias. As telas estão dispostas de forma que uma leva a outra, a navegação entre elas e a hierarquia de classes foi priorizada durante todo o trabalho;

3 DIVISÃO DE TAREFAS

O projeto foi desenvolvido durante as monitorias, aulas da disciplina e nas residências os integrantes do grupo. No decorrer do desenvolvimento cada aluno foi tomando posição de determinada parte do projeto e ficando encarregados da mesma, abaixo nós temos uma relação entre função e o(s) responsável(is) pela criação da mesma.

Classe Jogador: Jônatas, Lucas, Gabriela e Letícia.

Classe Tabuleiro – função:

HashCode, ToEquals, ToString, getters e setters: Jônatas, Lucas, Gabriela, e Letícia

zerarTabuleiro: Gabriela.

zerarPosicoes: Gabriela.

jogada1: Jônatas e Letícia.

jogada2: Jônatas e Letícia.

jogada3: Jônatas e Letícia.

verificarVitoria: Jônatas, Gabriela e Letícia.

verificarEmpate: Jônatas, Gabriela e Letícia.

Classe Partida – função:

HashCode, ToEquals, ToString, getters e setters: Jônatas, Lucas, Gabriela, e Leticia

zerar: Gabriela

jogou: Gabriela

parar: Leticia

jogada1: Jônatas e Leticia

jogada2: Jônatas e Leticia

jogada3: Jônatas e Leticia

fimJogada: Jônatas e Leticia

Classe Central – função:

removerJogador: Jônatas

iniciar1: Jônatas

iniciar2: Jônatas

editarUsuario: Gabriela

editarSenha: Gabriela

jogou: Gabriela

verificarGanhou: Gabriela

parar: Leticia

atual: Leticia

criarJogador: Leticia

cadastrar: Leticia

login1: Jônatas, Gabriela, e Leticia

login2: Jônatas, Gabriela, e Leticia

jogada1: Jônatas, Gabriela, e Leticia

jogada2: Jônatas, Gabriela, e Leticia

jogada3: Jônatas, Gabriela, e Leticia

rankingPontos: Jônatas, Gabriela, e Leticia

rankingVitorias: Jônatas, Gabriela, e Leticia

Outras produções:

Implementação do XML: Jônatas

Interface: Lucas

Produção dos relatórios e diagrama UML: Leticia

Participação na criação da Classe Tabuleiro				
Função	Jônatas	Lucas	M ^a Gabriela	M ^a Leticia
HashCode	X	X	X	X
ToEquals	X	X	X	X
ToString	X	X	X	X
Getters e Setters	X	X	X	X
zerarTabuleiro			X	
zerarPosicoes			X	
parar				X
jogada1	X			X
jogada2	X			X
jogada3	X			X
verificarVitoria	X		X	X
verificarEmpate	X		X	X

Participação na criação da Classe Partida				
Função	Jônatas	Lucas	M ^a Gabriela	M ^a Letícia
HashCode	X	X	X	X
ToEquals	X	X	X	X
ToString	X	X	X	X
Getters e Setters	X	X	X	X
zerar			X	
jogou			X	
parar				X
jogada1	X			X
jogada2	X			X
jogada3	X			X
fimJogada	X			X

Participação na criação da Classe Central				
Função	Jônatas	Lucas	M ^a Gabriela	M ^a Letícia
removerJogador	X			
inicio1	X			
inicio2	X			
editarUsuario			X	
editarSenha			X	
jogou			X	
verificarGanhou			X	
parar				X
atual				X
criarJogador				X

cadastrar				X
login1	X		X	X
login2	X		X	X
jogada1	X		X	X
jogada2	X		X	X
jogada3	X		X	X
rankingPontos	X		X	X
rankingVitorias	X		X	X

3 DIAGRAMA UML

