



Universidade Estadual Do Centro-Oeste
Setor De Ciências Exatas E De Tecnologia – SEET
Departamento De Ciência Da Computação – DECOMP
Disciplina: 2334 – Engenharia de Software III



Prof. Dr. Marcos Antonio Quináia

Engenharia de Software III – Jogo de Damas para Web

**Lucas Fernando Frighetto
Mauricio Specht**

Guarapuava, 28 de julho de 2015.

Sumário

Glossário.....	3
1.Introdução.....	4
2.Especificação de Requisitos.....	4
2.1.Requisitos de Usuário.....	4
2.2.Requisitos de Sistema.....	5
3.Ferramentas para desenvolvimento.....	6
4.Diagramação UML.....	6
4.1.Diagrama de casos de uso.....	7
4.1.1. Descrição dos casos de uso.....	8
4.2.Diagrama de classes.....	9
4.3.Disgrama de sequência.....	9
5.Projeto do Banco de Dados.....	10
6.Especificação de confiança.....	11
7.Especificação de proteção.....	12
8.Evolução do Sistema.....	12
9.Conclusão.....	12
Referências.....	12

Glossário

- **Chat:** Aplicação de conversação em tempo real.
- **Glassfish:** Servidor para a plataforma Java.
- **HTML:** Linguagem de programação utilizada para desenvolver páginas web.
- **Java:** Linguagem de programação orientada a objetos.
- **Lobby:** Ambiente para os jogadores interagirem entre si.
- **Postgre:** Sistema gerenciador de banco de dados.
- **SO:** Sistema Operacional.
- **SGBD:** Sistema de Gerenciamento de Banco de Dados.

1. Introdução

A Engenharia de Software surgiu com o objetivo de melhorar a qualidade dos produtos de software e aumentar a produtividade no processo de desenvolvimento. Esta trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software (Falbo, 2005).

Com o passar dos tempos, a difusão do campo tecnológico vem aumentando a competitividade nessa área. A produtividade passou a ser um aspecto fundamental para a sobrevivência de qualquer organização.

Com o intuito de preparar os alunos para o mercado de trabalho cada vez mais competitivo, as disciplinas de Engenharia de Software, Programação Orientada a Objetos, Banco de Dados e Interação Humano-Computador buscam simular a produção e utilização de um software comercial voltado para web, utilizando-se de clientes fictícios para tal abordagem e aproximando assim os alunos do campo de trabalho da engenharia e desenvolvimento de software.

O caso descrito nesse trabalho trata-se de um jogo de damas para web, tendo como cliente fictício os próprios instrutores das disciplinas. A aplicação deverá envolver um sistema de contas que possuem pontuação, um lobby para os jogadores escolherem seus adversários e o tabuleiro do jogo, com seus devidos estados.

Para o desenvolvimento do projeto, serão utilizadas técnicas de desenvolvimento, projeto e processo mais viáveis para este caso. Além disso, a implementação do sistema deve contar com a utilização de padrões de programação viáveis para este caso. Serão aplicados ainda, diversos conhecimentos referentes a Engenharia de Software, Interação Humano-computador, Programação Orientada a Objetos, Banco de dados, entre outras disciplinas, adquiridos durante o curso.

Encontram-se descritas aqui as atividades desenvolvidas na etapa 1 do projeto, que compreende, principalmente, a especificação do software em questão.

2. Especificação de Requisitos

Segundo Kotonya e Sommerville (2000), o termo Engenharia de Requisitos foi criado para cobrir todas as atividades envolvidas na descoberta, documentação, e manutenção de um conjunto de requisitos para um sistema baseado em computador.

Dentre outras atividades, a engenharia de requisitos compreende o levantamento e elicitação dos requisitos do software, a qual representa a etapa inicial de desenvolvimento do projeto. Para tal, a equipe fez uso de reuniões informais e discussões entre os integrantes da equipe e o cliente que, nesse caso, é o professor da disciplina de Engenharia de Software, a fim de sanar todas as dúvidas referentes ao sistema e diminuindo assim ao máximo a necessidade de futuras mudanças no projeto.

Foram identificados os requisitos inerentes ao usuário e ao sistema, tendo como base os diagramas UML elaborados a partir dessas reuniões.

2.1. Requisitos de Usuário

Segundo Sommeervile (2011), requisitos do usuário descrevem as funções e restrições do sistema de forma abstrata. Ou seja, descreve os requisitos do software do ponto de vista das necessidades do usuário, em linguagem natural e de fácil compreensão mesmo para quem não

tem conhecimento técnico sobre o funcionamento e o desenvolvimento do sistema. **Abaixo** encontram-se descritos os principais requisitos de usuários levantados nessa etapa.

- Um sistema web para o jogo;
- Interface Online;
- Impedir ataques externos que prejudiquem o andamento dos jogos.
- Proteger os dados dos usuários
- O jogo deve seguir as mesmas regras do jogo de damas nativo do SO Windows (Microsoft, 2015c).
- Quando for a vez do jogador, ele deve clicar em uma peça e escolher a posição onde deseja movê-la.
- Os jogadores devem possuir contas e pontuação de acordo com seu desempenho.
- Deve haver um ambiente para os jogadores interagirem entre si a fim de iniciar uma partida, denominado *lobby*.
- O usuário deve poder escolher o adversário com quem deseja jogar através do *lobby*.
- Os jogadores podem desistir durante o jogo ou pedir empate.

2.2. Requisitos de Sistema

Os requisitos de sistema são a versão complementar dos requisitos dos usuários, desenvolvidas pelos próprios desenvolvedores ou projetistas de software a fim de detalhar como cada requisito do usuário deve ser implementado no sistema. Esses requisitos têm por objetivo documentar as restrições e métodos oferecidos pelo sistema e pode servir, opcionalmente, como contrato entre as partes por definirem de forma clara o que será implementado e de que forma isso ocorrerá (Sommerville, 2011). A seguir encontram-se descritos os principais requisitos do software em questão.

- Um servidor para o sistema. GlassFish de preferência (Glassfish, 2015).
- Interface em HTML para o usuário.
- Utilizar padrões de projeto que aumentem a proteção (Miazaki, 2015).
- Utilização do Banco de dados PostgreSQL para as contas e histórico das contas (PostgreSQL, 2015).
- O jogador deve selecionar a peça que deseja mover, bem como a posição de destino através do clique do mouse.
- O movimento de captura é obrigatório quando possível.
- Para criar uma conta, basta fazer login com uma conta inexistente.
- A conta deverá ser composta por letras ou números de no mínimo de 3 caracteres e no máximo de 15.
- A senha poderá possuir caracteres especiais de no mínimo de 6 caracteres e no máximo de 20.
- Para iniciar uma partida, deve-se selecionar um jogador e clicar em desafiar. Isto ativará o seu status de desafiante no *lobby* para o jogador selecionado, se a ação for recíproca, a partida iniciará.
- A interface do *lobby* será em forma de lista, onde cada item dessa lista é um jogador que possui os atributos nome, pontuação e status de desafiante.
- O jogo termina quando é a vez do oponente e ele não pode mais realizar movimentos, seja por não possuir mais peças ou por ter todas as suas peças restantes trancadas.
- Se houver mais de uma captura na mesma jogada, é obrigatório capturar o maior número de peças.
- Se houver a possibilidade de fazer duas jogadas diferentes que capturarão o mesmo

número de peças, é obrigatório capturar as peças que contenham o maior número de damas.

- A captura pode ser realizada quando uma peça do oponente está imediatamente próxima da peça do jogador e seguida de uma casa vazia na diagonal, a peça do jogador é movida para esta casa vazia e a peça que estava imediatamente próxima é removida
- Uma peça normal torna-se uma dama quando atinge o lado oposto do tabuleiro.
- O jogo será composto por 24 peças, 12 brancas e 12 vermelhas, o primeiro movimento é sempre das peças vermelhas.
- O tabuleiro é de forma quadrada e contém 64 casas, estas com alternância entre as cores pretas e brancas. A grande diagonal de cor preta do tabuleiro deve ficar sempre à esquerda de ambos os jogadores, para que seja branca a primeira casa inferior a direita.
- Cada jogador começa com 12 peças (de uma só cor), as peças devem ser posicionadas nas casas pretas mais próximas do lado do jogador.
- Peças normais podem mover-se somente para frente, a dama pode mover-se para frente e para trás, e ambas podem mover-se apenas nas diagonais e para casas vazias.
- O movimento de captura é obrigatório quando possível.
- Apenas damas podem capturar peças movendo-se para trás.
- Damas e peças simples não podem saltar sobre peças de sua mesma cor, nem sobre duas peças de uma só vez, e também não podem passar duas vezes sobre a mesma peça.



3. Ferramentas para desenvolvimento

A escolha das ferramentas a serem utilizadas no projeto foi a próxima atividade realizada pela equipe. Para a realização deste projeto, serão utilizadas diferentes ferramentas para fins distintos. Dentre elas, podemos citar:

- MS – Visio Profissional (Microsoft Corporation, 2015a): Utilizado para a elaboração do diagrama de casos de uso.
- MS – Project 2013 (Microsoft Corporation, 2015b): Esta ferramenta será utilizada para a elaboração do cronograma, bem como o plano do projeto de software.
- MySQL Workbench (Oracle, 2015b): Essa ferramenta será utilizada para a elaboração do projeto do banco de dados.
- PostgreSQL (PostgreSQL, 2015): Será utilizado como Sistema de Gerenciamento de Banco de Dados (SGBD).
- NetBeans 8.0 (Oracle, 2015c): Esta IDE será utilizada para a codificação das funcionalidades dos componentes e desenvolvimento das janelas.
- Linguagem de Programação JAVA (Oracle, 2015d): Codificação do sistema.
- LibreOffice (LibreOffice, 2015): A plataforma LibreOffice será utilizada para o desenvolvimento dos relatórios do projeto.

4. Diagramação UML

A UML (Unified Modeling Language) é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de orientação a objetos. Esta linguagem tornou-se, nos últimos anos, a linguagem-padrão de modelagem de software adotada internacionalmente pela indústria de Engenharia de Software (Guedes, 2006).

Por mais simples que seja, todo e qualquer sistema deve ser modelado antes de se

iniciar sua implementação, isso porque os sistemas de informação costumam possuir a propriedade de crescer, isto é, aumentar em tamanho, complexidade e abrangência. Assim, um sistema deve ser bem documentado para que possa ser mantido com facilidade, rapidez e correção. Modelar um sistema é uma forma eficiente de documentá-lo, mas a modelagem é apenas uma das vantagens oferecidas pela UML.

Neste trabalho foram criados os diagramas de Casos de Uso, Classes e de Sequência, a fim de facilitar o entendimento, a implementação e a visão do sistema como um todo.

4.1. Diagrama de casos de uso

Para auxiliar o processo de elicitação dos requisitos junto ao usuário bem como facilitar a visão do sistema pelos desenvolvedores, um diagrama de casos de uso do sistema de locação foi elaborado. A figura 1 representa os casos de uso do software.

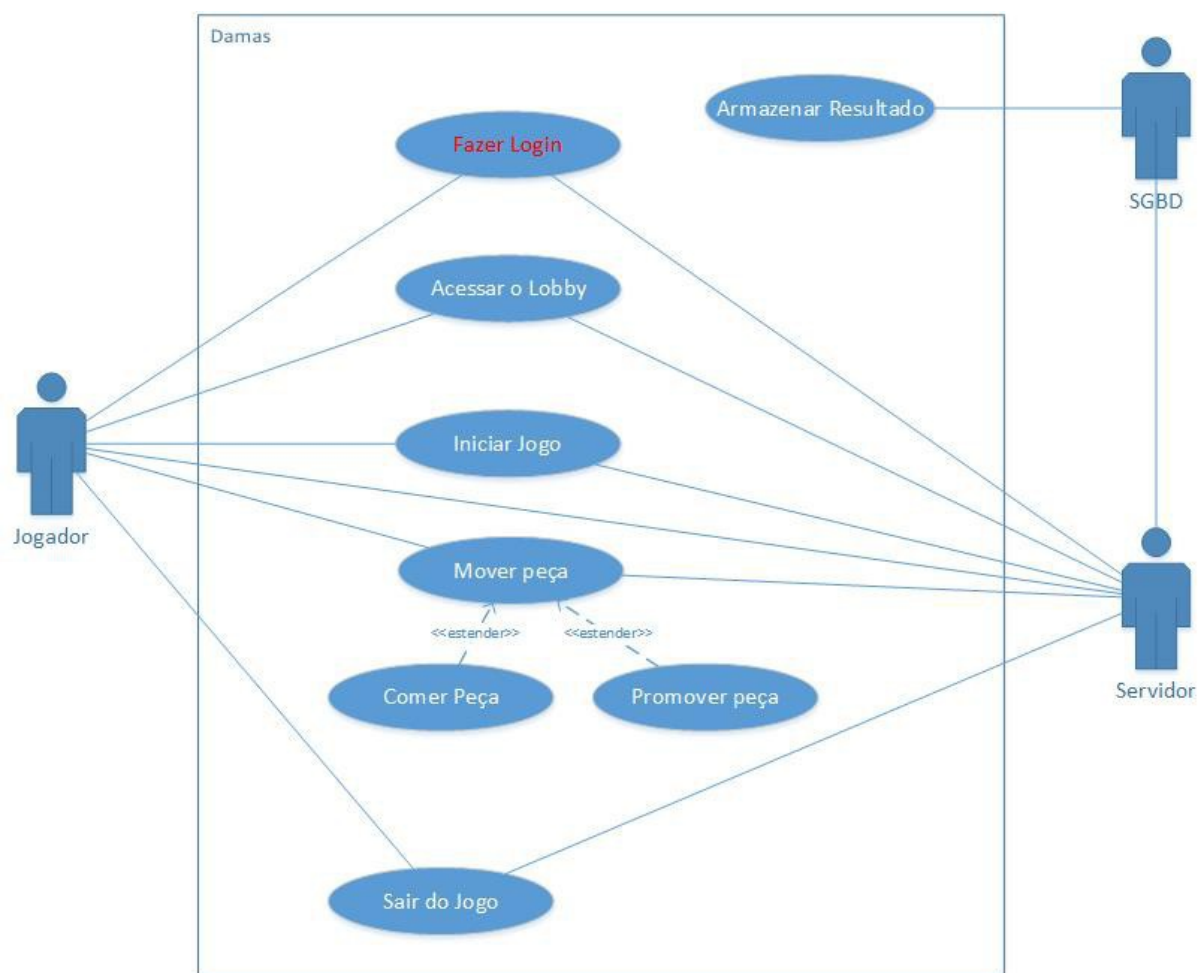


Figura 1: Diagrama de Casos de Uso do Software.

4.1.1 Descrição dos casos de uso do sistema

A descrição dos casos de uso possibilitam um melhor entendimento do funcionamento e resposta do sistema. Isso porque, em alguns casos, apenas o diagrama de casos de uso pode não ser capaz de comunicar a funcionalidade e o comportamento do sistema para o cliente, por não haver comunicação ou explicação.

Além disso, a descrição dos casos de uso define melhor e de maneira mais clara o escopo do sistema, evitam interpretações equivocadas do diagrama e facilitam a interpretação de como programar e implementar esse sistema. A Tabela 1 descreve os principais casos de uso do sistema.



Tabela 1: Descrição dos casos de uso do sistema.

Nome	Armazenar Resultado
Atores	Servidor, SGBD.
Descrição	Os resultados das partidas e informações dos jogadores e lobby são armazenados.
Pré-requisitos	
Dados	Dados dos jogadores, dados da partida.
Resposta	
Comentário	
Nome	Fazer Login
Atores	Jogador, Servidor, SGBD.
Descrição	O jogador informa seu login e senha para acessar o Lobby.
Pré-requisitos	O jogador deve ter uma conta já cadastrada.
Dados	Login e Senha do jogador.
Resposta	Acessar Lobby..
Comentário	
Nome	Acessar Lobby
Atores	Jogador, Servidor, SGBD.
Descrição	Acessa o Lobby para poder iniciar uma partida.
Pré-requisitos	Deve ter efetuado o Login.
Dados	Dados do Lobby.
Resposta	Acesso ao Lobby.
Comentário	
Nome	Iniciar Jogo
Atores	Jogador, Servidor, SGBD.
Descrição	O jogador inicia uma partida a partir do Lobby.
Pré-requisitos	deve ter feito o Login. Deve haver mais jogadores online,
Dados	Dados do jogador; Dados do Lobby.
Resposta	Iniciar partida
Comentário	
Nome	Sair do jogo
Atores	Jogador, Servidor.
Descrição	O jogador pode optar por abandonar o jogo. Isso resulta em derrota automática.
Pré-requisitos	O jogador deve estar em uma partida.
Dados	Dados da partida, dados do jogador.
Resposta	Volta para o Lobby.
Comentário	

4.2. Diagrama de classes

O diagrama de classes será útil no desenvolvimento do sistema, pois é nele que estarão definidas todas as classes do sistema, todos os atributos, as operações que o sistema deverá disponibilizar e as associações entre as classes. Dessa forma, o diagrama de **classe** (figura 6) serve como um modelo para objetos através da representação da estrutura e o relacionamento das classes.

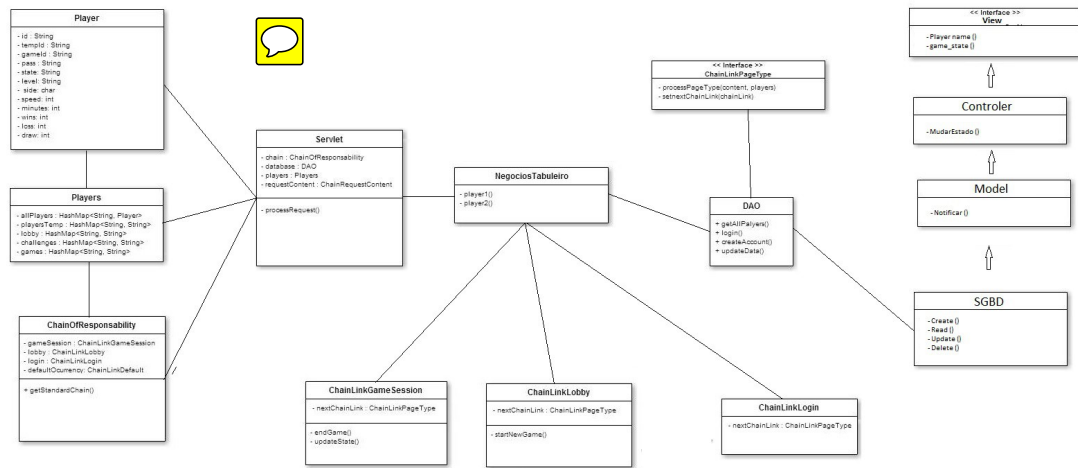


Figura 2: Diagrama de classes do sistema.

4.3. Diagrama de sequência

O diagrama de sequência consiste em demonstrar como é realizada a troca de mensagens durante a execução de uma operação do sistema [20]. Foram desenvolvidos somente os diagramas das quatro principais sequências. Tendo em vista que o restante quase não possui sequência, sendo somente chamada de função. A figura 7 ilustra o diagrama da sequência para o login no sistema.

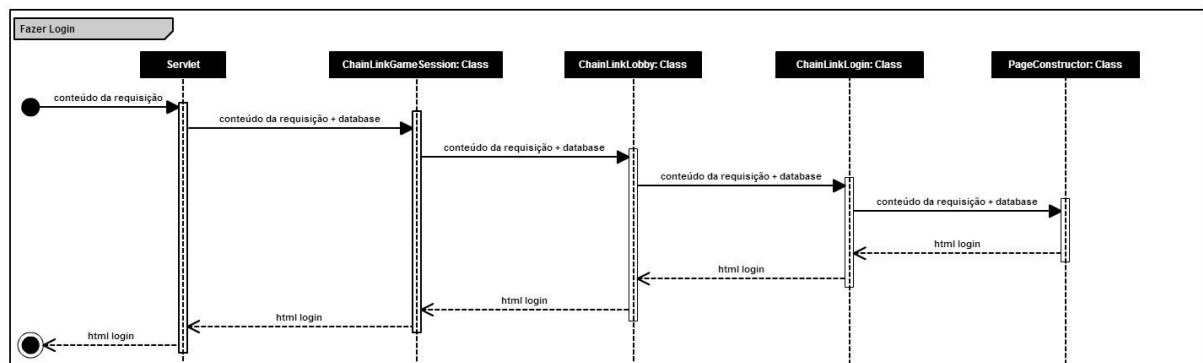


Figura 3: Diagrama de sequência para login no sistema.

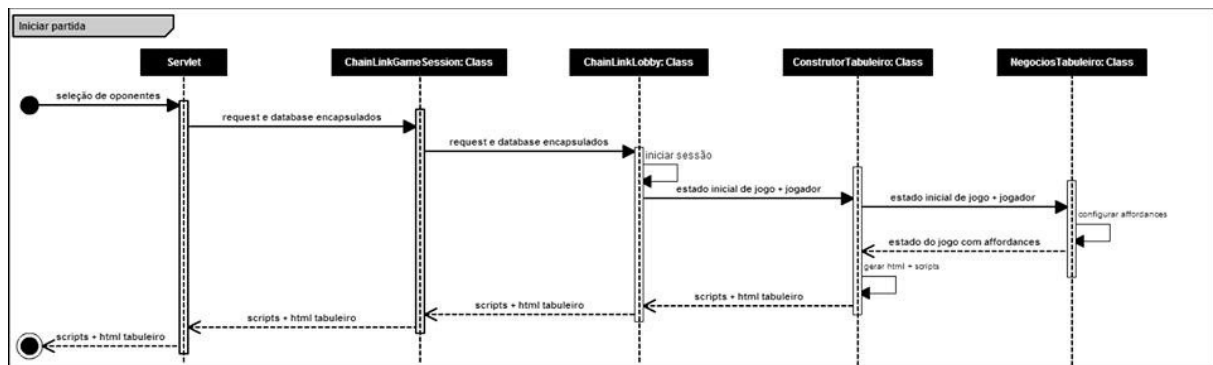


Figura 4: Diagrama de sequência para iniciar nova partida.

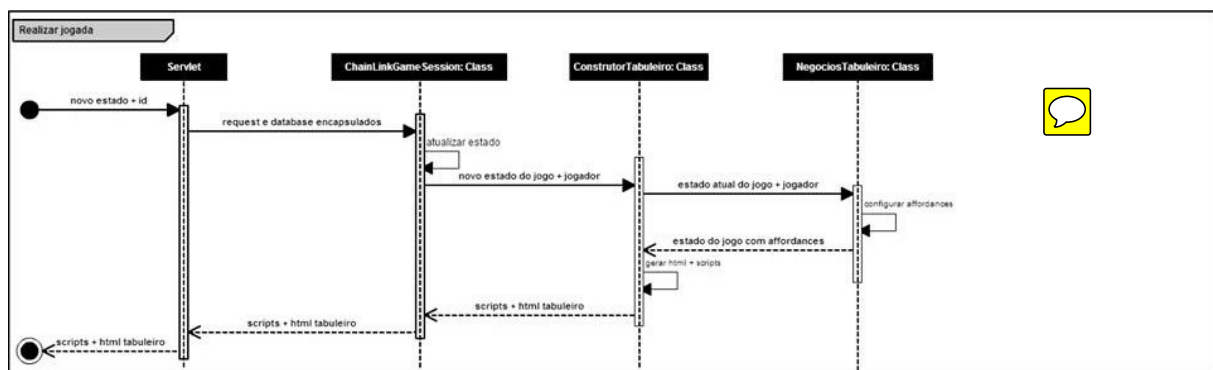


Figura 5: Diagrama de sequencia para realizar jogada.

5. Projeto do Banco de Dados

O projeto de banco de dados consistiu do desenvolvimento de um diagrama de entidade-relacionamento (ER), desenvolvido utilizando-se a ferramenta MySQL Workbench (Oracle, 2015b).

O modelo entidade-relacional tem o objetivo de facilitar o desenvolvimento do projeto de banco de dados; Basicamente, permite representar a estrutura lógica geral de um banco de dados. O modelo ER consiste de três componentes: conjunto de entidades, conjunto de relacionamento e atributos (Silberschatz et al., 2006).

O conjunto de entidades pode ser definido como sendo um conjunto de entidades do mesmo tipo que compartilham as mesmas propriedades ou atributos. O conjunto de relacionamentos consiste na ligação que define a interação entre as entidades, definindo assim a dinâmica dos negócios. Um atributo é uma característica que expressa uma entidade, sendo assim, os atributos estão contidos dentro do conjunto de entidades (Silberschatz et al., 2006).

Por se tratar de uma aplicação simples, com interação simplificada entre jogadores e sistema, no projeto desenvolvido são encontradas apenas duas entidades, sendo elas Player e Game. A entidade Player trata dos atributos dos jogadores, enquanto a Game possui apenas sua identificação e o estado do tabuleiro. A figura 18 representa o diagrama de entidade de relacionamento (ER) do banco de dados.

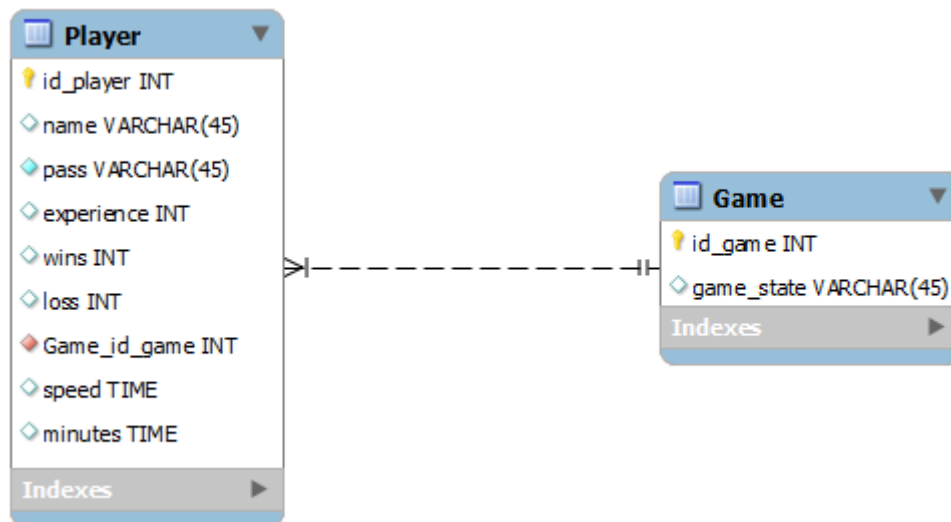


Figura 6: Diagrama de Entidade e Relacionamento (EER) do Banco de Dados.

6. Especificação de confiança

Confiança de software remete a quanto os softwares, críticos ou não críticos, são confiáveis (Sommerville, 2011). Basicamente, a Engenharia de Confiança está preocupada com técnicas que aumentam a confiança e diminui os riscos de falhas.

Entende-se que para que a confiança de um software seja garantida, processos e técnicas confiáveis devem ser aplicadas desde o projeto do software até sua conclusão. Contudo, muitas vezes o custo resultante desses processos são relativamente altos, levando os clientes a tolerarem certo grau de confiança, de modo a diminuir os custos.

Desta forma, por não se tratar de um sistema crítico, ou que resulte em grandes perdas e prejuízos em caso de falha, o atual sistema busca utilizar técnicas, padrões de projeto e implementação, além de processos confiáveis em sua produção, porém de forma a economizar tempo e diminuir os custos finais do projeto. Algumas características como desempenho, segurança e disponibilidade são de baixa prioridade, por se tratar de um jogo.

Pretende-se aplicar a este projeto, o uso de técnicas de engenharia de software, as melhores linguagens de programação para esta aplicação e o melhor gerenciamento da qualidade, a fim de aumentar o máximo a confiança do jogo, evitando também custos elevados para o sistema.

Além dessas opções de projeto, os processos de produção desse sistema deve estar preocupada com a prevenção de defeitos, a fim de que o sistema seja desenvolvido de tal forma que os erros humanos sejam evitados ou minimizados ao máximo e os defeitos do sistema minimizados. O processo será organizado visando a identificação de componentes suscetíveis a falhas antes da entrega do sistema.

Além disso, para as próximas etapas do trabalho pretende-se aplicar técnicas de verificação e validação sobre o sistema, a fim de detectar e corrigir os defeitos que possam existir. O sistema contará ainda, com políticas de tolerância a defeitos, ou seja, o sistema deve tratar erros comuns dentro do cenário do jogo sem que haja danos ao sistema como um todo.

7. Especificação de proteção

A Engenharia de Proteção é um dos focos principais de qualquer sistema voltado para internet, e não é diferente para o caso do jogo de damas. Esta atividade inclui ferramentas, técnicas e métodos para apoiar o desenvolvimento e manutenção de sistemas, a fim de evitar ataques externos mal intencionados que visam danificar o sistema ou prejudicar seu funcionamento (Sommerville, 2015).

Para garantir a proteção do sistema do jogo de damas, pretende-se implementar um sistema de controle de usuários com gerenciamento de permissões, baseadas principalmente em login e senha para cada usuário. Cada usuário terá permissão apenas para realizar as atividades a qual se destina, seja ele jogador ou administrador do sistema.

Pretende-se ainda, para as próximas etapas do projeto estudar a aplicabilidade da instalação de aplicações específicas para a proteção do sistema, além do uso de middleware e a configuração mais adequada desses sistemas para que as vulnerabilidades do sistema possam ser identificadas e minimizadas.

Por fim, o correto tratamento e monitoramento de ataques deve ser tido como prioridade do sistema, visto que ataques dificilmente podem ser evitados com total garantia. Desta forma, é essencial que acessos não autorizados ao sistema sejam neutralizados, aplicando técnicas e estratégias de projeto para resistir e se recuperar com o mínimo de danos ao sistema.

8. Evolução do Sistema

Processo de evolução de um software significa projetar um software para que o mesmo se adapte com facilidade às mudanças que surgirão futuramente, de forma natural. Os maiores custos de um projeto de software, para grandes organizações, encontra-se no projeto de evolução do mesmo, visto que os prejuízos causados por mudanças não esperadas para o sistema podem ser muito maiores (Sommerville, 2011).

No presente trabalho, o sistema é projetado para focar em dois tipos de instância, uma é o lobby, que é única no sistema, a outra é o tabuleiro (o jogo) que pode conter n instâncias. Como projeto de evolução para o software, pretende-se implementar o lobby de modo a deixar lacunas para possíveis implementações adicionais. Estas implementações podem ser referentes a interação entre jogadores, adicionar novos contatos, determinar diferentes estilos de jogos, entre outros.

Além do lobby, o tabuleiro também deve ser projetado de modo que sua modificação ou incrementação seja fácil e sem prejuízos para os demais componentes. Alguns exemplos de novas atribuições do tabuleiro pode ser exibir jogadas anteriores, ou verificar possíveis jogadas, por exemplo.

Tanto para o tabuleiro, quanto para o lobby, pretende-se alcançar essas características de evolução aplicando diferentes métodos e padrões, tanto de implementação quanto de projeto do sistema, a fim de garantir atributos de encapsulamento de componentes e um baixo acoplamento entre os vários componentes do sistema, de modo que futuras mudanças afetem apenas componentes específicos, e não todo o sistema (Miazaki, 2015).

9. Conclusão

Diversos fatores influenciam no desenvolvimento de um software, bem como na produtividade e qualidade do produto final. Tendo isso em vista, é notável a aplicabilidade dos métodos e

técnicas, não só de Engenharia de Software, mas de todos os campos de estudos que englobam o processo de desenvolvimento de um software.

Com o desenvolvimento desta primeira etapa do projeto, a qual engloba a elicitação de requisitos de sistema e usuário, modelagem UML, projeto de banco de dados, especificação de proteção e confiança e a definição dos métodos, ferramentas e processos, foi possível aproximar o conhecimento de sala de aula com a prática do projeto de software, possibilitando aos alunos aplicação desses conhecimentos e a preparação para o mercado de trabalho.

Referências



- Guedes, G. T. A (2006) “UML - Uma Abordagem Prática”, Ed. Novatec, segunda edição.
- Sommerville, I. (2011) “Engenharia de software” Editora: Pearson Education 9ª edição.
- Kotonya, G.; Sommerville, I. (2000) “Requirements Engineering”, John Wiley & Son Ltd.
- Falbo, R. de A. (2005) “Engenharia de Software”, Disponível online em: <<http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.pdf>>. Acessado em 25 de julho de 2015.
- Miazaki, M. (2015) “Princípios de Design”, Programação Orientada a Objetos 3, notas de aula.
- Oracle Corporation and/or its affiliates (2015a) “Java JVM”. Disponível em: <https://www.java.com/pt_BR/download/> Acessado em 25 de julho de 2015.
- Microsoft Corporation (2015a) “Visio Professional 2013”. Disponível em: <<http://office.microsoft.com/pt-br/visio/>> Acessado em 24 de julho de 2015.
- Microsoft Corporation [2] (2015b) “Deliver Winnning Projects”. Disponível em <<http://office.microsoft.com/en-us/project/>>Acessado em 24 de julho de 2015.
- Oracle Corporation and/or its affiliates (2015b) “MySQL Workbench”. Disponível em: <<http://www.mysql.com/products/workbench/>> Acessado em 24 de julho de 2015.
- PostgreSQL – The world's most advanced open source database (2015) “PostgreSQL”. Disponível online em <www.postgresql.org> Acessado em 25 de julho de 2015.
- Oracle Corporation and/or its affiliates (2015c) “NetBeans IDE”. Disponível em: <<https://netbeans.org/>> Acessado em 24 de julho de 2015.
- Oracle Corporation and/or its affiliates (2015d) “Java”. Disponível em: <<http://www.oracle.com/technetwork/pt/java/index.html>>Acessado em 24 de julho de 2015
- LibreOffice - A melhor suite office livreLibreOffice (2015) “LibreOffice”. Disponível em: <<https://pt-br.libreoffice.org/>>Acessado em 24 de julho de 2015.
- Microsoft Corporation (2015c) “Damas para Internet: Como jogar”. Disponível online em: <<http://windows.microsoft.com/pt-br/windows7/internet-checkers-how-to-play>>. Acessado em 26 de abril de 2015.
- GlassFish - World's first Java EE 7 Application Server (2015) “Glassfish Server”. Disponível em: <<https://glassfish.java.net/>> Acessado em 25 de julho de 2015.
- Silberschatz, A.; Korth, H. F.; Sudarshan, S. (2006) “Sistemas de Banco de Dados”, Ed. Elsevier, tradução da 5ª edição, 133-174 p.