



**UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM ENGENHARIA DE SOFTWARE**

WELLINGTON LUCAS MOURA

**DESENVOLVIMENTO DE UMA SOLUÇÃO WEB DE APOIO AO JOGO
EDUCACIONAL “CONCURSO DE IDEIAS DE NEGÓCIO”**

**QUIXADÁ
2016**

WELLINGTON LUCAS MOURA

DESENVOLVIMENTO DE UMA SOLUÇÃO WEB DE APOIO AO JOGO
EDUCACIONAL “CONCURSO DE IDEIAS DE NEGÓCIO”

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Engenharia de Software da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel. Área de concentração: Computação.

Orientador: Prof. Msc. Camilo Camilo Almendra.

Coorientadora: Prof^a. Dra. Paulyne Matthews Jucá

QUIXADÁ
2016

WELLINGTON LUCAS MOURA

**DESENVOLVIMENTO DE UMA SOLUÇÃO WEB DE APOIO AO JOGO
EDUCACIONAL “CONCURSO DE IDEIAS DE NEGÓCIO”**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Engenharia de Software da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel. Área de concentração: Computação.

Aprovado em: 04 / fevereiro / 2016.

BANCA EXAMINADORA

Prof. Msc. Camilo Camilo Almendra
Universidade Federal do Ceará (UFC)

Profª. Dra. Paulyne Matthews Jucá
Universidade Federal do Ceará (UFC)

Prof. Msc. Regis Pires Magalhães
Universidade Federal do Ceará (UFC)

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca do Campus de Quixadá

M884d Moura, Wellington Lucas
Desenvolvimento de uma solução web de apoio ao jogo educacional “Concurso de ideias de
negócio/ Wellington Lucas Moura. – 2016.
85 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de
Bacharelado em Engenharia de Software, Quixadá, 2016.
Orientação: Prof. Msc. Camilo Camilo Almendra
Área de concentração: Computação

1. Software – Desenvolvimento 2. Jogos educativos 3. Engenharia de software I. Título.

CDD 371.937

À minha mãe, Marlene.

Aos meus familiares.

Aos meus professores.

À Priscila Santos.

AGRADECIMENTOS

Agradeço à minha mãe Marlene, que me tornou a pessoa que sou e por todo o apoio oferecido durante minha formação.

Agradeço à minha família, que também me apoiou quando necessário.

Agradeço à minha namorada, Priscila Santos, pelo incentivo, apoio e compreensão em todos os momentos.

Agradeço ao professor Camilo Almendra, por ter me orientado com excelência durante esse período e pelas excelentes aulas ministradas que contribuíram para minha formação.

Agradeço à professora Paulyne Matthews Jucá, por me aceitar como seu bolsista neste projeto, por fazer parte da minha banca examinadora e por ter contribuído para minha formação como aluno e pessoa.

Agradeço ao professor Regis Pires Magalhães, pelo tempo, pelas críticas e sugestões que me ajudaram a tornar o trabalho cada vez melhor.

Agradeço à professora Andréia Libório Sampaio e ao professor Críston Pereira de Souza, por terem sido meus orientadores nas bolsas de Iniciação Acadêmica e Monitoria de Matemática, respectivamente, e que tiveram papel fundamental na minha formação.

Agradeço à professora Diana Braga e ao professor Wladimir Tavares, por terem sido tão importantes para minha formação.

Agradeço a professora Germana Rolin, pela paciência e dedicação em validar os resultados do projeto.

Em nome de Rayson Santos, Paula Santos, Wellyngton Amaral, Alexsandro Oliveira, Luan Lima, Mauro Roberto, Felipe Neves, Yago Alves, Neto Deolino, Lucas Vieira, agradeço a todos que de alguma maneira contribuíram para o resultado deste trabalho.

"To learn more about science, turn off your electronic device, go outside and look around a bit. Nature is calling you." (Niel deGrasse Tyson)

RESUMO

Com a gamificação da disciplina de Empreendedorismo surgiu a necessidade de uma solução web de apoio a execução do jogo empregado na disciplina. O objetivo desse trabalho é desenvolver uma solução web que apoie a configuração, acesso a informação, controle de rodadas e fases de gerenciamento do jogo, de acordo com os requisitos levantados. Para tanto foi definido um processo base que norteia o andamento do projeto, contendo as fases de análise e especificação de requisitos, análise e projeto, implementação, e verificação e validação. Como arcabouço tecnológico foram utilizadas ferramentas presentes no Núcleo de Práticas em Informática, juntamente com ferramentas utilizadas pela comunidade acadêmica da Universidade Federal do Ceará campus Quixadá. Para o desenvolvimento da solução web desejada, foram analisados e validados requisitos para compor o *backlog* da solução, foi definida arquitetura para guiar o desenvolvimento, foram desenvolvidos todos os requisitos especificados no *backlog* e foram verificadas e validadas funcionalidades desenvolvidas. Este trabalho apresenta um relato dos resultados obtidos pela execução desse processo. A metodologia e tecnologias adotadas nesse trabalho servem para orientar alunos no desenvolvimento de soluções web semelhantes.

Palavras-chave: Engenharia de Software. Desenvolvimento Web. Spring MVC. Jogo Sériio. Gamificação.

ABSTRACT

There is a need for a web solution to support the gamification of Entrepreneurship course. The objective of this work is to develop a web solution that supports the configuration, access to information, control rounds and stages of game management, according to the raised requirements. For this purpose a basic process has been defined that guides the progress of the project, including the stages of analysis and specification of requirements, analysis and design, implementation, and verification and validation. As technological framework used tools were present in Informatics Practices Center, along with tools used by the academic community at the Federal University of Ceará campus Quixadá. For the development of the desired web solution, requirements were analyzed and validated to make up the backlog of the solution, was defined architecture to guide the development, were developed all the requirements specified in backlog and were verified and validated developed features. This paper presents an account of the results of the implementation of this process. The methodology and technologies adopted in this work serve to guide students in developing similar web solutions.

Keywords: Software Engineering. Web Development. Spring MVC. Serious Game. Gamification.

LISTA DE ILUSTRAÇÕES

Figura 1 – Visão geral do processo	19
Figura 2 – Visão geral do <i>Scrum</i>	20
Figura 3 – Processo de Análise e Design	27
Figura 4 – Dependências entre pacotes	28
Figura 5 – Processo de implementação	29
Figura 6 – Processo de Verificação e Validação	30
Figura 7 – Visão da ferramenta easyBacklog.....	33
Figura 8 – Tela de login.....	36
Figura 9 – Tela de detalhes de um jogo.....	37
Figura 10 – Tela inicial visão do docente.....	37
Figura 11 – Protótipo ferramenta Pingendo	38
Figura 12 – Organização do projeto no Eclipse	41
Figura 13 – Repositório remoto no GitHub.....	42
Figura 14 – Formulário criado na solução.....	44
Figura 15 – Detalhes de uma equipe	45
Figura 16 – Detalhes de uma rodada	46
Figura 17 – Listagem ultimas submissões de uma rodada	47
Figura 18 – Pedido de reabertura rodada.....	49
Figura 19 – Avaliação do aluno com gabarito.....	49
Figura 20 – Histórico de notas usuário e equipe	51
Figura 21 – Ranking investidores rodada.....	52
Figura 22 – Ultimas submissões visão do aluno	53
Figura 23 – Listagem de empresas	54
Figura 24 – Visão Lógica de Pacotes	67
Figura 25 – Padrão Front Controller	68
Figura 26 – Padrão de comunicação entre classes e interfaces	69
Figura 27 – Visão das classes de modelo	70
Figura 28 – Cadastrar novo jogo	71
Figura 29 – Diagrama de estado de máquina de uma rodada.....	72
Figura 30 – Visão de classes de modelo.....	73
Figura 31 – Visão de apostas em uma rodada	74
Figura 32 – Histórico - Usuário Rodada.....	75
Figura 33 – Nota equipe rodada	75
Figura 34 – Visão de entregas para uma rodada.....	76
Figura 35 – Serviços de uma rodada	77

LISTA DE QUADROS

Quadro 1 – Caso de teste criar rodada.....	55
Quadro 2 – Execução caso de teste criar rodada	56

SUMÁRIO

1 INTRODUÇÃO.....	13
2 TRABALHOS RELACIONADOS	16
3 FUNDAMENTAÇÃO TEÓRICA	18
3.1 Processo de Desenvolvimento de Software	18
3.1.1 Análise e Especificação de Requisitos	20
3.1.2 Análise e Projeto.....	21
3.1.3 Implementação.....	21
3.1.4 Testes	22
3.2 Desenvolvimento web.....	22
3.2.1 Spring Framework	23
3.3 Ferramentas de Desenvolvimento.....	23
4 PROCEDIMENTOS METODOLÓGICOS	25
4.1 Especificação de Requisitos.....	25
4.1.1 Compreender as necessidades dos envolvidos	25
4.1.2 Detalhar histórias de usuários e preparar o <i>Backlog</i> do produto	25
4.1.3 Validação e priorização das histórias de usuário.....	25
4.1.4 Definir <i>Sprints</i>	26
4.2 Prototipação	26
4.3 Análise e <i>Design</i>	26
4.3.1 Análise arquitetural e definição de arquitetura.....	27
4.4 Preparar ambiente de desenvolvimento	28
4.5 Implementação da solução.....	28
4.5.1 Definição de pronto	29
4.5.2 Implementar itens do <i>Backlog</i>	29
4.6 Verificação e Validação	29
4.6.1 Especificar casos de teste	30
4.6.2 Executar testes	30
4.6.3 Validar solução com o cliente	31
5 DESENVOLVIMENTO.....	32
5.1 Analisar requisitos	32
5.1.1 Análise inicial dos requisitos	32
5.1.2 Prototipação	35
5.1.3 Validação dos requisitos.....	38
5.1.4 Definição de <i>Sprints</i>	39
5.2 Projetar arquitetura	40
5.3 Implementar solução.....	41
5.3.1 Ambiente de desenvolvimento	41
5.3.2 Implementação da solução.....	42
5.4 Planejar e executar testes	55
5.4.1 Especificar casos de teste	55
5.4.2 Executar testes	56
5.4.3 Validar solução	58

6 CONSIDERAÇÕES FINAIS	59
6.1 Trabalhos Futuros	61
REFERÊNCIAS	62
APÊNDICES	64
APÊNDICE A – Documento de Arquitetura.....	64
ANEXOS	78
ANEXO A – Documento de Visão	78
ANEXO B – Roteiro de Entrevista.....	86

1 INTRODUÇÃO

A aplicação de práticas de jogos vem sendo utilizada para suprir a necessidade de maior dinamismo nas salas de aula (JUCÁ; ROLIM, 2013). Nesse contexto, a disciplina de empreendedorismo ofertada nos cursos do Campus Quixadá da Universidade Federal do Ceará, desde 2012 vem adotando técnicas de gamificação na sua metodologia. Gamificação pode ser definida como a aplicação de dinâmicas de jogos, em ambientes diferentes do contexto convencional dos jogos, com o objetivo de tornar mais atrativas as atividades realizadas, aumentando o engajamento das pessoas (JUCÁ; ROLIM, 2013). A disciplina de empreendedorismo foi gamificada a fim de promover um maior engajamento dos alunos (JUCÁ et al., 2014).

O jogo Concurso de Ideias de Negócio (CIN), fruto do projeto de extensão Jogo de Empreendedorismo (JEM), tem como principal objetivo a construção de ideias de negócio por alunos organizados em equipes, que ao longo do jogo constroem e melhoram suas ideias a partir de entregas parciais e feedback do docente. Cada entrega parcial é efetuada durante uma rodada. O jogo promove a competição de melhor equipe da rodada, melhor equipe geral, melhor avaliador da rodada e melhor investidor geral. Individualmente, os alunos podem apostar nas melhores ideias de negócios dentre todas as equipes. Dessa forma, o jogo promove a busca por excelência na elaboração das ideias de forma colaborativa, assim como promove a análise crítica individual dos alunos.

Anteriormente na disciplina, a turma era dividida em equipes, onde cada uma elaborava sua ideia de negócio. Com esse processo, os alunos não tomavam conhecimento detalhado sobre as ideias de negócio das outras equipes, perdendo assim a oportunidade de aprender com o trabalho desenvolvido pelos colegas (JUCÁ et al., 2014).

“O jogo é constituído de um conjunto de fases, ou desafios, que são primeiramente apresentados ao aluno. Depois, os alunos devem executar o desafio e liberar para os demais alunos o resultado do desafio levando em consideração o seu negócio proposto.” (JUCÁ; ROLIM, 2013, p. 4). O docente da disciplina de Empreendedorismo que deseje aplicar o jogo, necessita de apoio ferramental específico para lidar com as diversas informações e arquivos trocados entre o docente, alunos e equipes. Além disso o docente atua como configurador e gerenciador da execução do jogo. A participação dos alunos no jogo é usada como forma de avaliação na disciplina, em conjunto com outras formas de avaliação. Para o docente é importante poder acompanhar o desempenho dos alunos de forma individual ao longo do jogo. Para o aluno é importante ter acesso às informações e artefatos trocados

entre participantes do jogo. O aluno da disciplina se envolve com o jogo de duas maneiras: como membro de uma equipe e como investidor individual. Como membro de equipe o aluno colabora para construir os artefatos das rodadas em equipe, e recebe notas em grupo que compõem sua avaliação final. Como investidor, o aluno faz avaliação individual de artefatos de outras equipes, inclusive a sua, e realiza apostas de valores em equipes que acreditar serem as melhores na rodada. O aluno deve ser capaz de acompanhar o racional de cálculo de suas notas, incluindo a nota em equipe e a nota de avaliação de artefatos.

Devido à falta de ferramenta específica para controle do jogo, inicialmente foi utilizada a ferramenta Moodle¹ juntamente com formulários do Google² para suprir essa necessidade. Através do Moodle, o docente controla datas de início e fim de cada fase e utiliza funções de repositório para a troca de arquivos digitais entre docente e alunos. Já os formulários são usados para coletar as avaliações dos alunos em planilhas. Os rankings também são disponibilizados através do Moodle, onde os alunos podem acessar os resultados a todo instante. Pela natureza do jogo, cada rodada envolve muitas informações inseridas pelos envolvidos, e realizar a tabulação e organização dessas informações nas planilhas, além das trocas de e-mails, é muito suscetível a erros e demanda muito esforço em horas de trabalho do docente (Anexo A).

Para tentar resolver as necessidades apresentadas na execução do jogo CIN este trabalho propõe o desenvolvimento de um ambiente de apoio baseado na web. Anteriormente foi iniciado um projeto com essa finalidade cujo foco principal foram os cálculos das rodadas. Devido à iniciativa de construção anterior, alguns artefatos podem ser reusados e adaptados, como modelagem da base de dados, identificação de entidades, facilitando assim a fase inicial do desenvolvimento deste trabalho. A iniciativa anterior tem seu código fonte³ publicado na Internet.

De acordo com levantamento de requisitos realizado junto à docente responsável por ministrar a disciplina, percebeu-se que a parte de configuração e troca de informações são partes essenciais para uma solução que apoie o jogo. Esse levantamento também foi realizado no contexto de um projeto de extensão da universidade, e resultou no documento de visão disponível no Anexo A.

O objetivo desse trabalho é desenvolver uma solução web que apoie a configuração, acesso a informação, controle de rodadas e fases de gerenciamento do jogo, de

¹ <http://moodle.quixada.ufc.br/>

² <https://www.google.com/forms/about/>

³ <https://github.com/Marcosepi/game-empendedorismo>

acordo com os requisitos levantados no documento de visão, Anexo A. Este trabalho continua o esforço anterior adotando um novo *framework* de desenvolvimento. Para isso serão utilizadas técnicas e ferramentas consolidadas para a especificação dos requisitos, análise e projeto, implementação e testes da solução.

Além da introdução, este trabalho está organizado da seguinte forma: na Seção 2, são apresentados os trabalhos relacionados; na Seção 3, a fundamentação teórica é apresentada, onde serão definidos o processo de desenvolvimento, o *framework* adotado e ferramentas de apoio ao desenvolvimento do projeto; na Seção 4, são discutidos os procedimentos metodológicos utilizados para execução deste trabalho; na Seção 5, são apresentados e discutidos os resultados de todo o processo de desenvolvimento para a solução web; e na Seção 6 são apresentadas as considerações finais e os trabalhos futuros.

2 TRABALHOS RELACIONADOS

Savi e Ulbricht (2008) apresentam o potencial que os jogos digitais educacionais possuem para dinamizar o ambiente de aprendizado nas salas de aulas, buscando tirar proveito do poder de atração que os jogos têm, devido a sua natureza. Com base nesse trabalho, alguns benefícios dos jogos digitais educacionais podem ser aplicados ao CIN, como, efeito motivador, facilitador de aprendizado, socialização e comportamento *expert*. Essas características fornecem embasamento para que seja despendido esforço voltado à criação de uma ferramenta, que apoie a execução do jogo empregado na disciplina de empreendedorismo, buscando melhorar a experiência de interação dos alunos e professor da disciplina com o jogo.

Em Oliveira, Cintra e Neto (2013), é proposto um jogo sério multiusuário voltado ao ensino da Gestão de Riscos em Projetos de Software. Nomeado como eRiskGame, o jogo é caracterizado como sendo persistente, baseado em navegadores web, englobando tarefas que um gerente de projetos deve executar em uma organização, com foco na gestão de riscos. Esse trabalho assemelha-se ao nosso pelo fato de ser proposto o desenvolvimento de um jogo sério, podendo ser visto como uma solução, para o ensino de uma disciplina, baseado na web e multiusuário. Difere, porém, pelo fato de que o desenvolvimento de uma solução web para o jogo CIN usará uma arquitetura de desenvolvimento específica, com técnicas e ferramentas utilizadas em projetos do Núcleo de Práticas em Informática (NPI), enquanto eRiskGame é um Jogo Persistente Baseado em Navegadores (JPBN)⁴ e utiliza técnicas de Inteligência Artificial, incluindo agentes inteligentes e lógica *fuzzy*.

Em Juca et al. (2014), são apresentados os resultados da gamificação da disciplina de empreendedorismo, os dados são obtidos em decorrência da primeira aplicação da disciplina após sua gamificação. A gamificação se deu através da aplicação de um jogo que neste trabalho é chamado de “Concurso de Ideias de Negócio”. O jogo é dividido em cinco fases, cada fase corresponde a uma parte do plano de negócio, onde o aluno elabora um documento correspondente a cada fase, que são “Identificação de uma oportunidade de negócio”, “Descrição da empresa, produtos e serviços, mercados e competidores”, “Análise e Mix de Marketing e Comercialização”, “Análise Estratégica”, e “Plano Financeiro”. Juca et al. (2014) propõe como trabalho futuro o desenvolvimento de ferramentas mais adequadas para apoiar a execução do jogo, devido à ausência de ferramenta específica. Esse trabalho está

⁴ Persistent Browser-Based Game – Defining a genre, <http://www.pbbg.org>

relacionado ao objetivo do nosso, tendo em vista que estamos dando continuidade ao projeto de construir uma ferramenta que apoie adequadamente a execução do jogo.

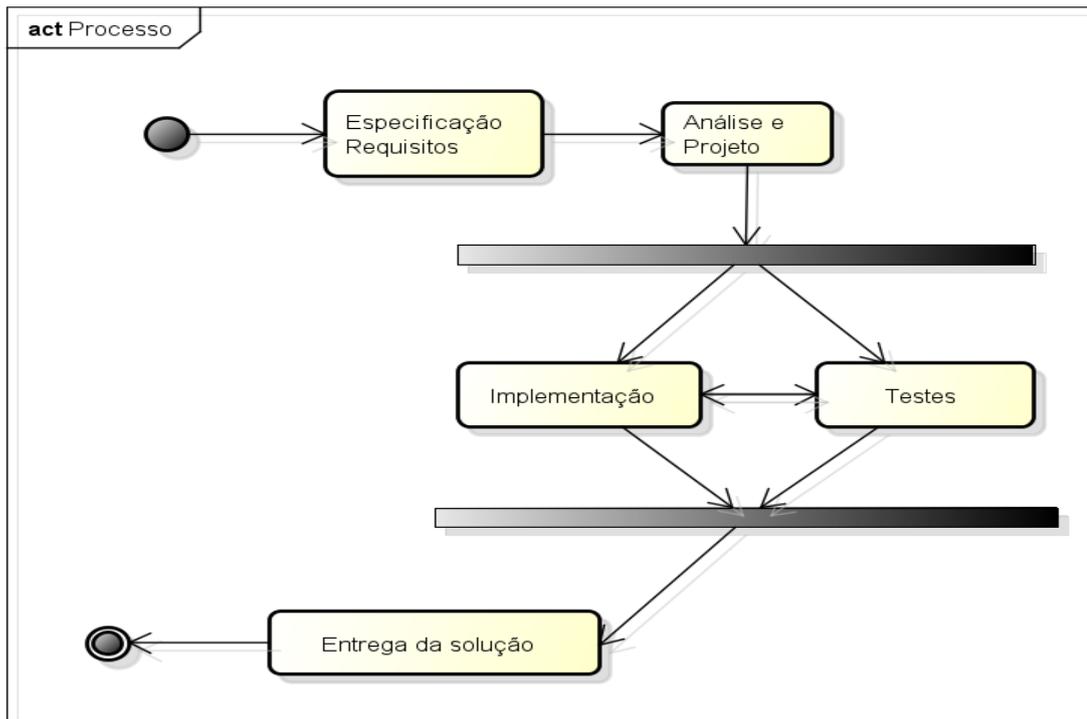
3 FUNDAMENTAÇÃO TEÓRICA

Na fundamentação teórica serão abordados conceitos utilizados para o desenvolvimento do projeto. Na primeira subseção será definido o processo de desenvolvimento, como está dividido e como foi adaptado. No contexto de desenvolvimento web, a segunda subseção apresenta o que é *Spring Framework* e quais vantagens agrega ao desenvolvimento de uma aplicação web. Na terceira subseção serão apresentadas as ferramentas de apoio ao desenvolvimento do projeto, como *frameworks* e ferramentas *case*.

3.1 Processo de Desenvolvimento de Software

Processo de desenvolvimento de software é o que define as atividades a serem realizadas, qual sequência de atividades devem ser seguidas, quais documentos ou artefatos gerados em cada etapa e papéis com suas respectivas responsabilidades. Existem muitos modelos de processos, e nenhum é ideal para todos os casos, ou seja, processos precisam ser adaptados. Porém todos os processos possuem atividades em comum, são essas: especificação de software; projeto e implementação; validação; e evolução (SOMMERVILLE, 2011). Para o desenvolvimento da solução web pretendida optou-se por definir um processo, em que se engloba as atividades base a qualquer processo de desenvolvimento de software. Conforme ilustrado na Figura 1, pode-se notar que as principais atividades se encontram definidas no processo. Com relação à evolução do software, o processo foi elaborado para esse fim, cada fase do processo gerará artefatos que poderão ser consultados por futuros desenvolvedores que precisarem evoluir o código fonte da solução.

Figura 1 – Visão geral do processo

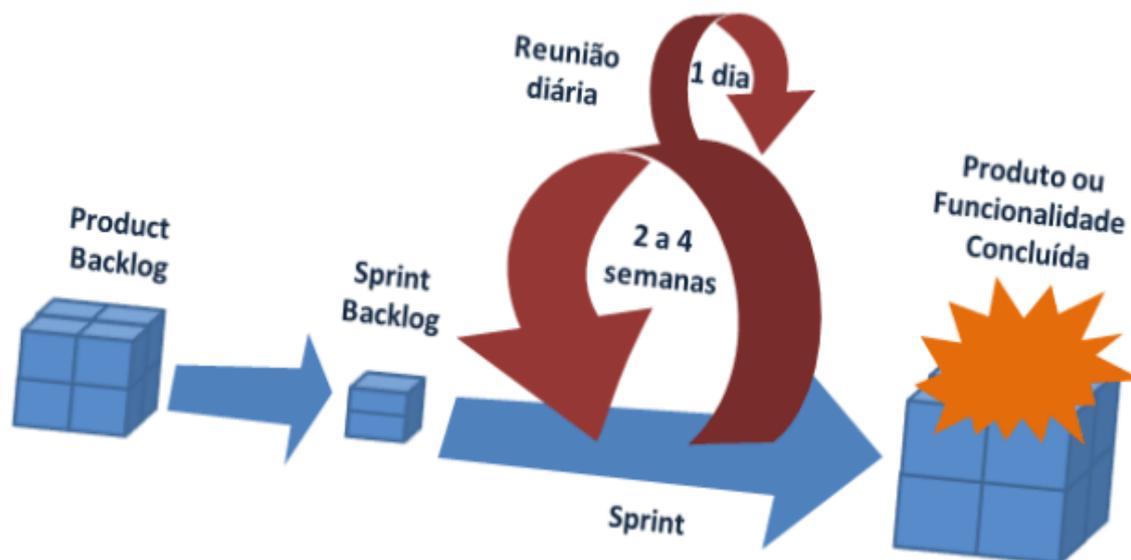


Fonte: Elaborado pelo autor

O tipo de diagrama utilizado para representação do processo foi o de atividades, que é orientado a fluxos de controle. O diagrama foi utilizado para representar como as fases do processo se relacionam (BEZERRA, 2007).

Para o desenvolvimento do projeto de software serão utilizadas práticas da metodologia *Scrum*. “*Scrum* não é um processo ou uma técnica para construir produtos; em vez disso, é um *framework* dentro do qual você pode empregar vários processos ou técnicas.” (SCHWABER; SUTHERLAND, 2013, p. 3). O *framework* foi escolhido por ser uma metodologia ágil que visa iterações curtas para promover visibilidade ao desenvolvimento. Entre os eventos do *Scrum*, serão utilizados, *Sprint*, Reunião de Planejamento da *Sprint* e Revisão da *Sprint*. Entre os artefatos do *Scrum*, serão utilizados o *Backlog* do Produto, *Backlog* da *Sprint* e o Incremento. Uma prática importante inerente às metodologias ágeis utilizada no projeto é a de Histórias de Usuários, que será apresentada com mais detalhes no decorrer das próximas seções, assim como os artefatos e práticas do *Scrum*. Uma visão geral do *Scrum* é apresentada na Figura 2.

As próximas subseções explicam superficialmente os conceitos de cada fase do processo, tendo em vista que cada um deles é suficientemente complexo e não é o foco desse trabalho apresentar detalhadamente cada um dos conceitos.

Figura 2 – Visão geral do *Scrum*

Fonte: Adaptado de <http://www.mindmaster.com.br/scrum/>

3.1.1 Análise e Especificação de Requisitos

Requisitos são necessidades do cliente que precisam ser atendidas com o sistema pretendido. Especificar requisitos é o processo de elucidar as necessidades do cliente e documentá-las, com isso obtém-se os documentos que compõem os requisitos do sistema (SOMMERVILLE, 2011).

A visão de um produto é o nível mais alto de abstração dos requisitos, onde são alinhados os interesses dos *stakeholders*, delimitando o escopo do produto, tanto positivo quanto negativo (WIEGERS, 2003). A visão do produto está representada através do documento de visão, Anexo A. O Documento de visão é escrito focando nas características essenciais descritas pelo cliente, coletando os requisitos de negócio em um único documento, que define o cenário para o desenvolvimento posterior (KRUCHTEN, 2003; WIEGERS, 2003).

Para o processo definido, a fase de Especificação de Requisitos gera um refinamento do documento de Visão e a construção do *Backlog* do produto, ambos são artefatos importantes para o decorrer de todo o processo. A partir do documento de visão são especificadas histórias de usuários, que juntas compõem o *Backlog* do produto. *Backlog* do Produto é uma lista com todos os requisitos do produto, é a origem única dos requisitos para qualquer mudança a ser feita. Um *Backlog* do Produto nunca está completo, é dinâmico e evolui de acordo com a evolução do produto (SCHWABER; SUTHERLAND, 2013).

História de usuário é a descrição de requisitos que agregam valor a um *stakeholder*. As histórias de usuário são compostas por três aspectos: uma descrição da história, conversas sobre as histórias, e confirmações ou critérios de aceitação (COHN, 2011). A descrição das histórias, ou cartão, é usada no planejamento e como lembrete, apresentando claro valor de negócio para o *stakeholder*. Conversas são detalhes expostos durante a Elicitação, ajudando a definir o escopo da história. Confirmações servem para clientes avaliarem uma história, podendo ser utilizadas para determinar se uma história está pronta (JEFFRIES, 2001).

3.1.2 Análise e Projeto

A fase de análise e projeto foca na viabilidade da continuação do projeto, mapeia os requisitos de forma a descrever como implementar o sistema. Análise é o fluxo de projeto que mapeia os requisitos em um conjunto de classes e subsistemas. Essa transformação para o processo adotado é dirigida pelas histórias de usuário. Focando em delimitar e controlar os requisitos funcionais da solução. O projeto tem como objetivo fazer um refinamento da análise com base nos requisitos não funcionais da solução, buscando cobrir todos os requisitos (KRUCHTEN, 2003).

O projeto de arquitetura está preocupado em como o sistema será dividido, definindo uma estrutura geral. O resultado do projeto de arquitetura é o Documento de Arquitetura, que capta várias visões arquiteturais do sistema (SOMMERVILLE, 2011). Apesar de não ser uma tarefa simples, documentar a arquitetura trará benefícios para quando futuros membros de equipe precisarem compreender e avaliar o sistema, podendo através desse documento gerar análises sobre seu provável desempenho (GORDON, 1998).

Essa fase permite selecionar a melhor estratégia de implementação. Define-se uma arquitetura de referência que melhor se adeque às características dos requisitos funcionais e não funcionais, além do modelo de dados a ser adotado (KRUCHTEN, 2003). Um projeto pode adotar uma arquitetura de referência, que serve como um modelo para o sistema que está sendo desenvolvido (SOMMERVILLE, 2011). O desenvolvimento de um sistema segue os padrões da arquitetura adotada, como divisão de classes por pacotes, nomenclaturas, etc.

3.1.3 Implementação

Implementação é a fase onde será construída ou codificada a solução propriamente dita. Essa é fase que gera uma versão executável do sistema (SOMMERVILLE,

2011). A implementação será realizada com entregas incrementais, onde alguns incrementos desenvolvidos são entregues ao cliente de acordo com o valor já agregado ao projeto.

É importante que se defina quando um item do *Backlog* do produto ou incremento é dado como “Pronto”, de forma a alinhar esse conceito entre os envolvidos. Essa definição impacta em muitas decisões durante o projeto. Um item é dado como “Pronto” quando ele atende às especificações determinadas entre os membros da equipe (SCHWABER; SUTHERLAND, 2013). No escopo deste projeto um item é dado como pronto quando atende aos critérios de aceitação das histórias de usuário e é validado pelo cliente.

3.1.4 Testes

Entenda teste como verificação e validação. Verificação é o processo de analisar se o software está sendo construído de maneira correta, validação é o processo de analisar se o software construído é o que o cliente espera (PRESSMAN, 2011, p. 402).

Nesta etapa, serão executados testes funcionais à medida que novos itens forem desenvolvidos, tentando agregar maior consistência à solução desenvolvida. Para isso algumas técnicas de teste de software serão empregadas, principalmente o teste de fronteira. Teste de fronteira é uma das atividades mais importantes para testes funcionais, onde são verificados os extremos de algum conjunto de dados, onde os erros frequentemente ocorrem (PRESSMAN, 2011, p. 408). A validação de novas funcionalidades e incrementos serão feitas mediante os critérios de aceitação definidos para as histórias de usuários, em primeira instancia. Funcionalidades serão tidas como válidas após a aprovação do cliente.

3.2 Desenvolvimento web

O desenvolvimento de jogos educacionais mais sofisticados demanda muitos custos, chegando a envolver muitas áreas da computação, como banco de dados, computação gráfica, redes de computadores e estrutura de dados, além de necessitar de equipes multidisciplinares, tornando difícil o desenvolvimento devido ao custo dos recursos humanos necessários (SAVI; ULBRICHT, 2008). Por isso, o desenvolvimento de uma solução web, com uma plataforma de desenvolvimento mais conhecida pela comunidade do campus UFC Quixadá, foi escolhida. A subseção seguinte apresenta o principal componente dessa plataforma de desenvolvimento, *Spring Framework*.

3.2.1 Spring Framework

“*Framework* é uma estrutura genérica estendida para se criar uma aplicação ou subsistema mais específico” (SOMMERVILLE, 2011, p. 300). *Frameworks* dão suporte a serviços genéricos comuns a um conjunto de aplicações de mesmo tipo. Como os frameworks usam padrões de projeto eles permitem a extensibilidade (SOMMERVILLE, 2011). “*Spring Framework* é um *framework* voltado para o desenvolvimento de aplicações corporativas para a plataforma Java” (WEISSMANN, 2012, p. 3).

O *Spring Framework* é uma tecnologia madura no mercado. Sua arquitetura é modular, o que permite o uso apenas dos módulos que uma aplicação necessite, assim evitando desperdícios de recursos. Por ser baseado nos conceitos de inversão de controle e injeção de dependências, a lógica de domínio da aplicação fica desacoplada do *framework* (JOHNSON et al., 2015). Esse *framework* é adotado como base de desenvolvimento pela maioria dos projetos do NPI e está em processo de consolidação nas disciplinas de desenvolvimento web do Campus UFC Quixadá. O *Spring* possui uma série de recursos, que permitem a evolução da aplicação para aplicação com redes sociais, acesso a uma diversidade de fontes de dados, provimento de web *services*, implementação de testes unitários, etc. Por este ser um projeto desenvolvido por pesquisadores do Campus, é mais provável que os próximos desenvolvedores a darem manutenção no projeto sejam alunos ou professores do Campus, o que facilitará a realização de manutenção e melhorias na aplicação, devido a utilização do *framework*.

3.3 Ferramentas de Desenvolvimento

Iniciar um projeto do zero acarreta uma certa problemática, caso o desenvolvedor não tenha um arcabouço tecnológico definido à sua disposição. Por isso, em sua maioria, as tecnologias utilizadas no projeto fazem parte do arcabouço tecnológico do NPI: Spring Framework, Web (HTML, CSS, JS, jQuery), Git, Maven, Requisitos, Debugging, etc. (Almendra; Magalhães; Almeida, 2015). Outras tecnologias são provenientes de conhecimento prévio das disciplinas ofertadas no campus Quixadá.

A *Integrated Development Environment* (IDE), Eclipse versão JEE⁵, para desenvolvedores web, é a principal ferramenta de apoio ao desenvolvimento da solução web proposta. A IDE foi escolhida pelo fato de ser destinada ao desenvolvimento web com Java e amplamente utilizada, pois possui em seu pacote de complementos ferramentas importantes

⁵ <https://eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/lunasr2>

que auxiliam no desenvolvimento. Um complemento importante é o plug-in EGit, que facilita a integração do projeto com o servidor remoto GitHub⁶, fornecendo uma perspectiva para o gerenciamento de mudanças do projeto. GitHub é o repositório online onde o código fonte será armazenado.

Para o gerenciamento de dependências do projeto foi utilizado o gerenciador do Eclipse baseado no Maven. Apache Maven⁷ é uma ferramenta de gerenciamento, construção e implantação de projetos.

Para o gerenciamento do *Backlog* foi utilizada a ferramenta easyBacklog⁸ que é online, gratuita, divide as histórias em temas, capaz de exportar o *Backlog* como .xls, além da possibilidade de adicionar contribuintes ao projeto. A ferramenta dá suporte a divisão das histórias em *Sprints*, que são ciclos de desenvolvimento, e gera gráficos estatísticos para o gerenciamento das *Sprints*.

A ferramenta utilizada para acompanhamento e controle de mudanças foi o Redmine⁹, que é a mesma ferramenta adotada pela UFC Campus Quixadá para o gerenciamento dos seus projetos. Diferente do easyBacklog, o Redmine é utilizado para acompanhamento de atividades em geral, não se limitando ao gerenciamento do *Backlog* do produto.

Para o gerenciamento de testes a ferramenta utilizada foi o TestLink¹⁰, que, entre outras coisas, dá suporte à criação de planos de testes, casos de testes e registra resultados de execução de testes. A ferramenta também permite exportar os dados para o formato CSV.

O Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado foi o PostgreSQL¹¹, e o servidor web da aplicação é o Apache Tomcat. Ambos são tecnologias presentes na arquitetura¹² do NPI.

Para estilização das páginas JSP do projeto foi utilizado o *framework* Bootstrap¹³, que é amplamente utilizado em projetos tanto no NPI, quanto na comunidade acadêmica do campus.

⁶ <https://github.com>

⁷ <https://maven.apache.org/>

⁸ <https://easybacklog.com/>

⁹ <http://adm.quixada.ufc.br/>

¹⁰ <http://testlink.npi.quixada.ufc.br/>

¹¹ <http://www.postgresql.org/>

¹² <https://github.com/npi-ufc-qxd/wiki/wiki/Auto-estudo>

¹³ <http://getbootstrap.com/>

4 PROCEDIMENTOS METODOLÓGICOS

A capacidade de gerenciar, fornecer maior visibilidade aos participantes, diminuir o esforço em horas para configuração e divulgação de resultados são pontos fundamentais de uma solução web para o jogo CIN. Um meio para se conseguir isso é através do desenvolvimento de uma solução seguindo um processo em que suas fases sejam respeitadas, fornecendo documentação essencial, utilizando-se de ferramentas que possuam boa maturidade no mercado e dando suporte para evolução da solução.

Este trabalho apresenta o desenvolvimento de uma solução web utilizando os processos e ferramentas descritos na seção 3 deste trabalho. A seguir esses passos são detalhados.

4.1 Especificação de Requisitos

O projeto conta com um Documento de Visão, exposto no Anexo A, por isso não foi necessário o levantamento dos requisitos do zero. Sendo assim, a fase de especificação de requisitos seguiu os passos citados nas subseções seguintes.

4.1.1 Compreender as necessidades dos envolvidos

Nesta etapa do projeto, foram compreendidas as necessidades dos envolvidos com o sistema, conhecidos como *stakeholders*, de acordo com especificações existentes no Documento de Visão, Anexo A. Através de experiência própria do autor, que foi aluno da disciplina de Empreendedorismo, muitas dúvidas de como funciona o jogo na prática foram sanadas.

4.1.2 Detalhar histórias de usuários e preparar o *Backlog* do produto

As histórias de usuários foram inseridas no *Backlog* do produto, onde foram complementadas com os campos “Para que” e com os critérios de aceitação, que definem respectivamente, uma ação que o usuário espera que o sistema possua e os critérios para uma história ser considerada completa.

4.1.3 Validação e priorização das histórias de usuário

Nesta etapa as histórias de usuário foram validadas junto ao cliente e atualizadas. A priorização seguiu os conceitos apresentados em Cohn (2009), onde são definidas estratégias para priorização de histórias.

4.1.4 Definir *Sprints*

De acordo com a priorização das histórias, foram definidas *Sprints*, de modo a proporcionar um melhor agrupamento de funcionalidades para o desenvolvimento em cada ciclo.

4.2 Prototipação

Prototipação é uma técnica usada em muitas situações durante o desenvolvimento de sistemas, baseia-se num esboço inicial do *layout* do sistema, onde os principais objetivos são obter *feedback* do cliente a respeito da organização dos elementos, e levantamento de novas necessidades. Protótipos não precisam ser muito detalhados e fieis à realidade, porém servem como guias para o desenvolvimento de interfaces gráficas e levantamento de requisitos.

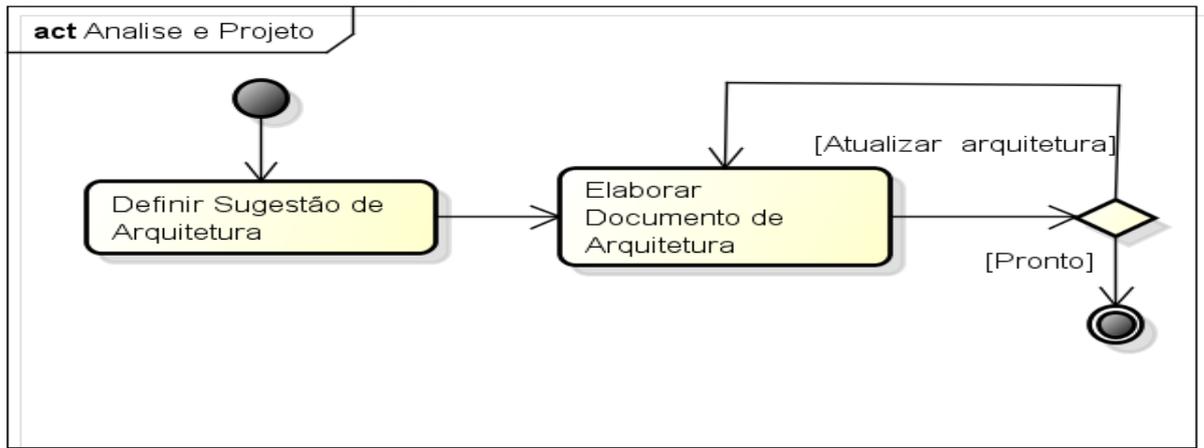
A prototipação foi usada de acordo com a divisão das primeiras *Sprints*, onde foram desenvolvidos protótipos para funcionalidades especificadas nas histórias. A técnica de prototipação foi escolhida devido a facilidade de validar requisitos funcionais e não funcionais do sistema.

Após o desenvolvimento das primeiras funcionalidades, a prototipação pode ser opcional, visto que fica fácil discutir alterações ou novas funcionalidades diretamente na interface já desenvolvida.

4.3 Análise e *Design*

Com os requisitos definidos, priorizados e divididos em *Sprints* iniciais, foram feitos a análise e o *design* da solução a ser criada. A Figura 3 apresenta o processo definido para essa etapa.

Figura 3 – Processo de Análise e Design



Fonte: Elaborado pelo autor

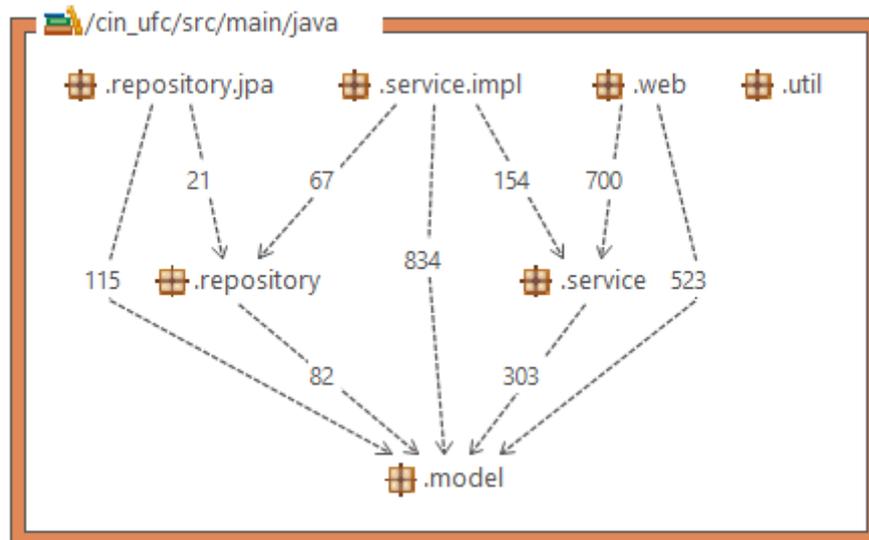
4.3.1 Análise arquitetural e definição de arquitetura

Foi adotada a arquitetura de referência definida pelo *Spring MVC*, que é o *Model View Control (MVC)*, padrão amplamente utilizado por aplicações web. O projeto segue os padrões já utilizados no NPI, além dos padrões definidos pela própria comunidade do *Spring*.

Nesta etapa foi gerado um documento de arquitetura com as visões lógica, de processos e de implementação. O documento foi atualizado à medida que se achou necessário. Detalhes das visões arquiteturais foram criados, após algumas *Sprints* concluídas, que podem servir como apoio ao aprendizado de futuros desenvolvedores (LARMAN, 2007).

O plug-in STAN, do Eclipse, foi utilizada para verificar as conformidades com a arquitetura de referência adotada. O plug-in mostra o relacionamento entre componentes em diferentes níveis de visão do projeto, possibilitando assim, a análise do acoplamento do projeto e se está acontecendo alguma violação nos relacionamentos entre classes ou entre pacotes. A Figura 4 apresenta uma visão gerada pelo STAN no nível de pacotes.

Figura 4 – Dependências entre pacotes



Fonte: Elaborado pelo autor

4.4 Preparar ambiente de desenvolvimento

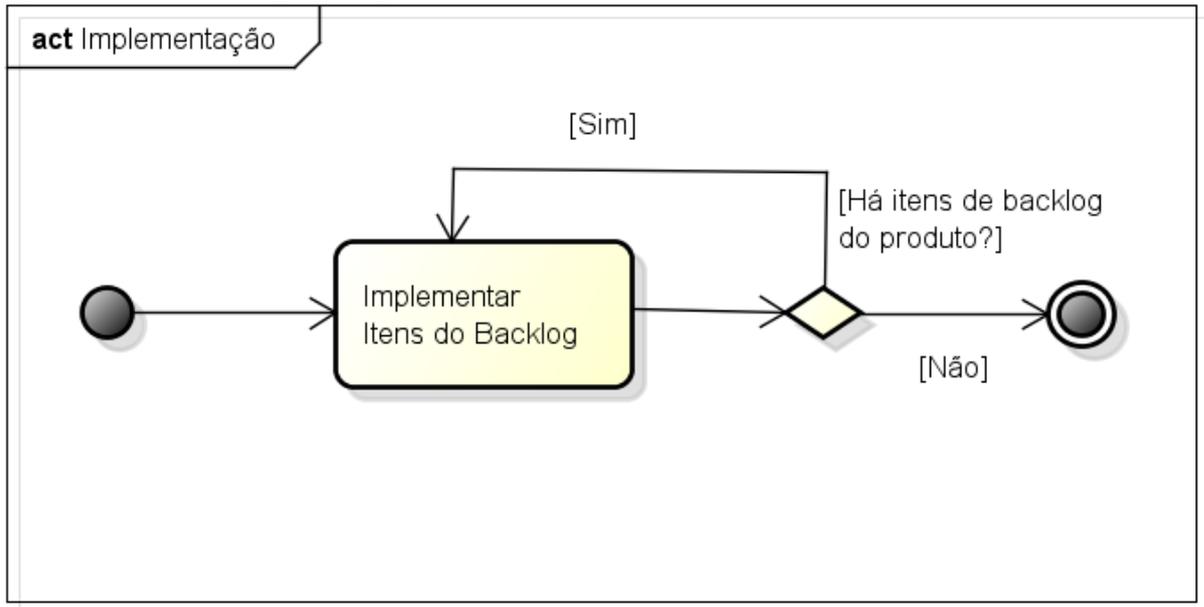
Nesta etapa foi preparado o ambiente de desenvolvimento, com ferramentas que suportam e atendem às necessidades de desenvolvimento. Focando sempre em ferramentas livres e gratuitas, para não gerar dificuldades a quem for manter o projeto posteriormente.

A configuração do ambiente de desenvolvimento foi feita utilizando-se as ferramentas listadas na seção 3.3 com Windows 8.1 64 bits como sistema operacional, porém o uso desse Sistema Operacional (SO) não é obrigatório, pois são utilizadas ferramentas multiplataformas.

4.5 Implementação da solução

O processo de implementação segue o fluxo ilustrado na Figura 5.

Figura 5 – Processo de implementação



Fonte: Elaborado pelo autor

4.5.1 Definição de pronto

Um incremento será dado como “pronto” após a especificação dos casos de testes, desenvolvimento do incremento, execução dos testes, integração do incremento com o *branch* de desenvolvimento e validação com o cliente. Os passos citados foram definidos com o Gerente de Projeto, que é o orientador deste trabalho.

4.5.2 Implementar itens do *Backlog*

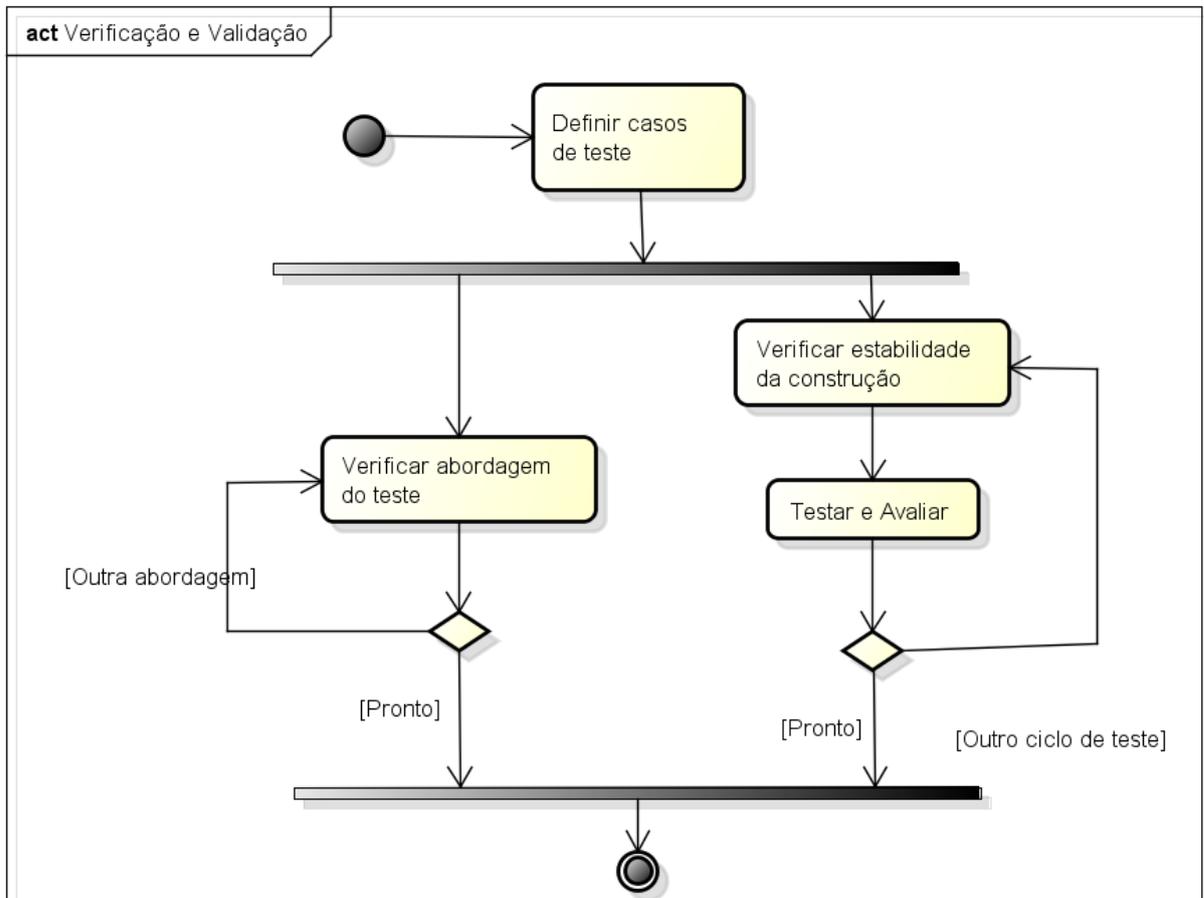
Com o que vai ser implementado definido nas *Sprints*, foi feita a codificação dos componentes. Esta etapa não se limitou apenas à construção novos itens, também foram feitas atividades de adaptação de código fonte existente, execução de testes, além de fornecer *feedback* para atualização do *design* da solução. Padrões de projeto foram analisados no decorrer dessa etapa e empregados onde necessário.

Os componentes foram integrados continuamente, para manter a consistência do sistema, como um todo, no decorrer das *Sprints*. Como para o desenvolvimento foi utilizado o *Spring Framework* com a estrutura de projetos utilizada pelo NPI, esta etapa também consistiu em manter os padrões utilizados pelo NPI, com o intuito de facilitar futuras manutenções.

4.6 Verificação e Validação

O processo de verificação e validação segue o fluxo apresentado na Figura 6.

Figura 6 – Processo de Verificação e Validação



Fonte: Elaborado pelo autor

4.6.1 Especificar casos de teste

Os casos de teste foram feitos de acordo com os critérios de aceitação definidos para cada história de usuário. Com a especificação dos casos de teste foi possível ter uma melhor noção do que se deve testar no sistema, pois os casos de teste são baseados nos requisitos do sistema, agregando consistência ao sistema a cada funcionalidade testada e validada de acordo com seus respectivos casos. Esses foram criados e controlados utilizando-se a ferramenta TestLink. A especificação de casos de testes é um ativo importante do projeto, visando facilitar sua continuidade por futuras equipes.

4.6.2 Executar testes

Os testes foram executados concorrentemente ao desenvolvimento, à medida que funcionalidades foram sendo implementadas logo em seguida foram testadas. O objetivo dessa abordagem é garantir uma maior qualidade nas funcionalidades desenvolvidas para a solução.

4.6.3 Validar solução com o cliente

Nesta etapa o cliente avaliou as funcionalidades que foram desenvolvidas, fornecendo *feedback* e informando se o sistema correspondia às suas expectativas. A validação foi um processo que se repetiu várias vezes, buscando saber se o sistema realmente está atendendo às necessidades e expectativas do cliente.

5 DESENVOLVIMENTO

Nesta seção são apresentados, em forma de relato pessoal, as atividades realizadas e os resultados alcançados durante todo o processo de desenvolvimento da solução web pretendida.

5.1 Analisar requisitos

Esta seção trata da análise de requisitos para preparação do *Backlog* do produto. Nas próximas subseções abordarei as etapas dessa fase do processo.

5.1.1 Análise inicial dos requisitos

Inicialmente para a preparação do *Backlog* analisei ferramentas que dessem suporte a essa tarefa, cogitei a utilização planilhas eletrônicas, mas como já era de conhecimento prévio de disciplinas do campus, escolhi a ferramenta online easyBacklog devido aos fatores citados na seção 3.3.

Mantive os temas para agrupar histórias de usuários de acordo com o Documento de Visão do projeto, Anexo A. Com isso, criei temas na ferramenta e adicionei histórias em seus respectivos temas.

Inicialmente baseado no Documento de Visão, preenchi os campos “Para que”, das histórias, de acordo com minha expectativa para cada perfil de ator em relação ao resultado de um requisito definido em uma história. Defini critérios de aceitação para todas as funcionalidades presente no *Backlog* até então. Acrescentei comentários com o intuito de ajudar na tomada de decisões sobre a funcionalidade pretendida. A Figura 7 – Visão da ferramenta easyBacklog mostra a estrutura de uma história de usuário inserida na ferramenta easyBacklog. É possível observar como o *Backlog* fica organizado na ferramenta, com o tema “Equipe” e suas respectivas histórias.

Figura 7 – Visão da ferramenta easyBacklog

Concurso de Ideias de Negócio Wellington Lucas Moura Filter Snapshots Export

Backlog Stats 8 7 6 5 4 3 2 1 Total do Backlog

Tema	ID	História de Usuário	Critérios de Aceitação	Comentários
Configuração 11 stories collapsed				
Código: CON				
Equipe 1 story collapsed				
Código: EQU				
EQU1		Como membro de equipe Eu quero visualizar informações dos demais membros Para que seja possível algum feedback interno da equipe	a) Deve ser exibido apenas as informações aos membros da equipe b) Nenhuma informação é mostrada se a pessoa não estiver associada a uma equipe c) Uma pessoa consegue ver a lista de demais membros	Controle de acesso às informações da equipe, como artefatos submetidos, informações gerais dos outros membros, nome, email, etc.
EQU2		Como membro de equipe Eu quero visualizar informações de contato do docente Para que se estabeleça um meio de comunicação	a) Deve ser exibido aos alunos o email, ou outro tipo de contato disponibilizado pelo docente	Exibir informações no rodapé da página
EQU3		Como membro de equipe Eu quero editar as informações da minha equipe Para que o dados estejam consistentes	a) O campo matrícula não pode ser editado pelo aluno	Campos: logo e nome da empresa;
EQU4		Como membro de equipe	a) O campo ideia de negócio pode ser	O aluno deve ser capaz de adicionar uma

Fonte: Elaborado pelo autor

Nesse processo identifiquei algumas histórias que julguei faltarem na versão inicial do documento, além de realocar histórias que se encaixavam melhor em outros temas. O *Backlog* inicialmente com 28 histórias, foi acrescido de mais histórias, totalizando 37. Os temas e suas respectivas histórias ficaram organizados como se segue, onde as novas histórias estão marcadas como (Novo) e as remanejadas como (Remanejada de):

Configuração

- CON1: Como docente, eu posso cadastrar participantes;
- CON2: Como docente, eu posso cadastrar/atualizar as informações gerais do jogo;
- CON3: Como docente, eu posso organizar equipes de participantes para um jogo;
- CON4: Como docente, eu posso revisar a ideia de negócio de uma equipe;
- CON5: Como aluno, eu posso cadastrar-me no jogo (Novo);
- CON6: Como administrador, eu posso cadastrar docentes (Novo);
- CON7: Como docente, eu posso importar participantes;
- CON8: Como docente, eu posso cadastrar um novo jogo (Novo);

Equipe

- EQU1: Como membro de equipe, eu posso visualizar informações dos demais membros;

- EQU2: Como membro de equipe, eu posso visualizar informações de contato do docente;
- EQU3: Como membro de equipe, eu posso editar as informações da minha equipe;
- EQU4: Como membro de equipe, eu posso editar a ideia de negócio;
- EQU5: Como membro de equipe, eu posso submeter artefato para entrega da rodada;
- EQU6: Como membro de equipe, eu posso visualizar o histórico de submissão para uma entrega de rodada;
- EQU7: Como membro de equipe, eu posso visualizar o saldo da equipe;
- EQU8: Como membro de equipe, eu posso visualizar o resultado da equipe numa rodada;
- EQU9: Como membro de equipe, eu posso adquirir um serviço (Novo);

Avaliação Individual do Aluno

- AVA1: Como aluno, eu posso visualizar as submissões finais da rodada de todas equipes;
- AVA2: Como aluno, eu posso submeter uma avaliação individual de uma entrega de equipe;
- AVA3: Como aluno, eu posso visualizar minhas avaliações individuais anteriores;
- AVA4: Como docente, eu posso visualizar as avaliações individuais dos alunos;
- AVA5: Como aluno, eu posso visualizar minha nota individual;
- AVA6: Como docente, eu posso configurar um formulário de múltipla escolha e gabarito (Novo);

Aposta

- APO1: Como investidor, eu posso apostar dinheiro, do valor liberado para a rodada, em uma ou mais equipes na rodada;
- APO2: Como investidor, eu posso visualizar minhas apostas anteriores;
- APO3: Como docente, eu posso visualizar todas as apostas dos alunos;

Avaliação do Docente

- AOD1: Como docente, eu posso salvar uma avaliação para uma entrega de equipe;

- AOD2: Como docente, eu posso configurar o fator de aposta para uma equipe;

Controle

- CTR1: Como docente, eu posso publicar as avaliações e fatores para as entregas de uma rodada;
- CTR2: Como docente, eu posso gerar e publicar os rankings;
- CTR3: Como docente, eu posso encerrar o jogo;
- CTR4: Como docente, eu posso iniciar o jogo (Novo);
- CTR5: Como docente, eu posso encerrar período de apostas (Novo);
- CTR6: Como docente, eu posso encerrar período de submissão de artefatos (Novo);
- CTR7: Como docente, eu posso iniciar uma rodada (Remanejada de Configuração);
- CTR8: Como docente, eu posso iniciar uma rodada final (Remanejada de Configuração);
- CTR9: Como participante, eu posso me logar no jogo (Novo);

5.1.2 Prototipação

Antes de revisar os requisitos com a cliente, criei protótipos não funcionais relacionados com histórias de usuários existentes até então. Esses protótipos foram criados para sanar dúvidas sobre organização inicial de elementos no layout do sistema, direcionar o desenvolvimento, validar requisitos existentes e obter novos requisitos.

Realizei uma pesquisa para definição de qual ferramenta utilizar no desenvolvimento de protótipos, onde encontrei algumas opções:

- Pencil¹⁴, ferramenta gratuita, poucas opções, interfaces muito realistas geralmente com temas do sistema operacional Windows XP, possui poucos componentes para download e que não são atualizados, descontinuado em 2012;
- moqups.com¹⁵, ferramenta online, possui poucos componentes, as funcionalidades que interessam, como adicionar usuários para edição, opções de exportar os protótipos, estão disponíveis apenas na versão paga.

¹⁴ <http://pencil.evolus.vn>

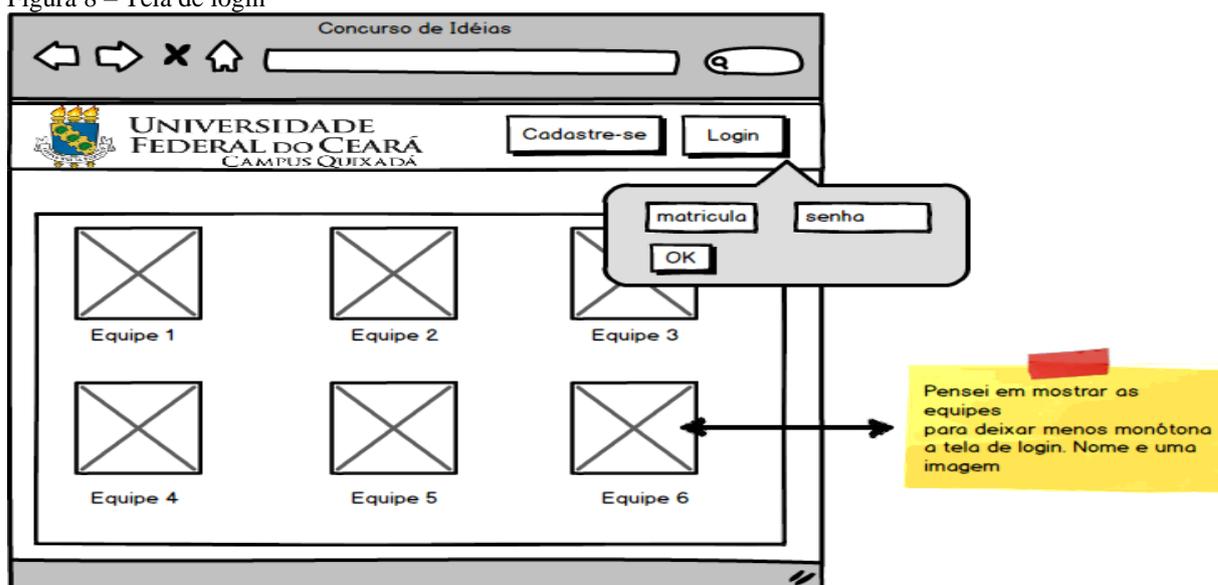
¹⁵ <https://moqups.com>

- Balsamiq Mockups 3¹⁶, versão *trial*, disponibiliza os recursos da versão paga por 30 dias, que são muitos, entre as vantagens está a exportação dos protótipos como PDF e PNG;
- Pingendo¹⁷, ferramenta que cria protótipos gerando o código HTML e CSS Bootstrap referente ao layout definido para a página. Possibilitando o aproveitamento dos layouts para uso no projeto.

Em geral as ferramentas de prototipação disponibilizam versões gratuitas com algum limitante, podendo ser quantidade de dias, quantidade de *frames* usados, entre outros.

Após utilização, análise das funcionalidades e recursos das ferramentas, optei pela utilização da Balsamiq Mockups 3, com a qual elaborei alguns protótipos, a seguir três deles são apresentados nas Figuras Figura 8, Figura 9 e Figura 10. Esses protótipos foram elaborados com o intuito de levantar novos requisitos e esclarecer dúvidas a respeito do layout da solução.

Figura 8 – Tela de login

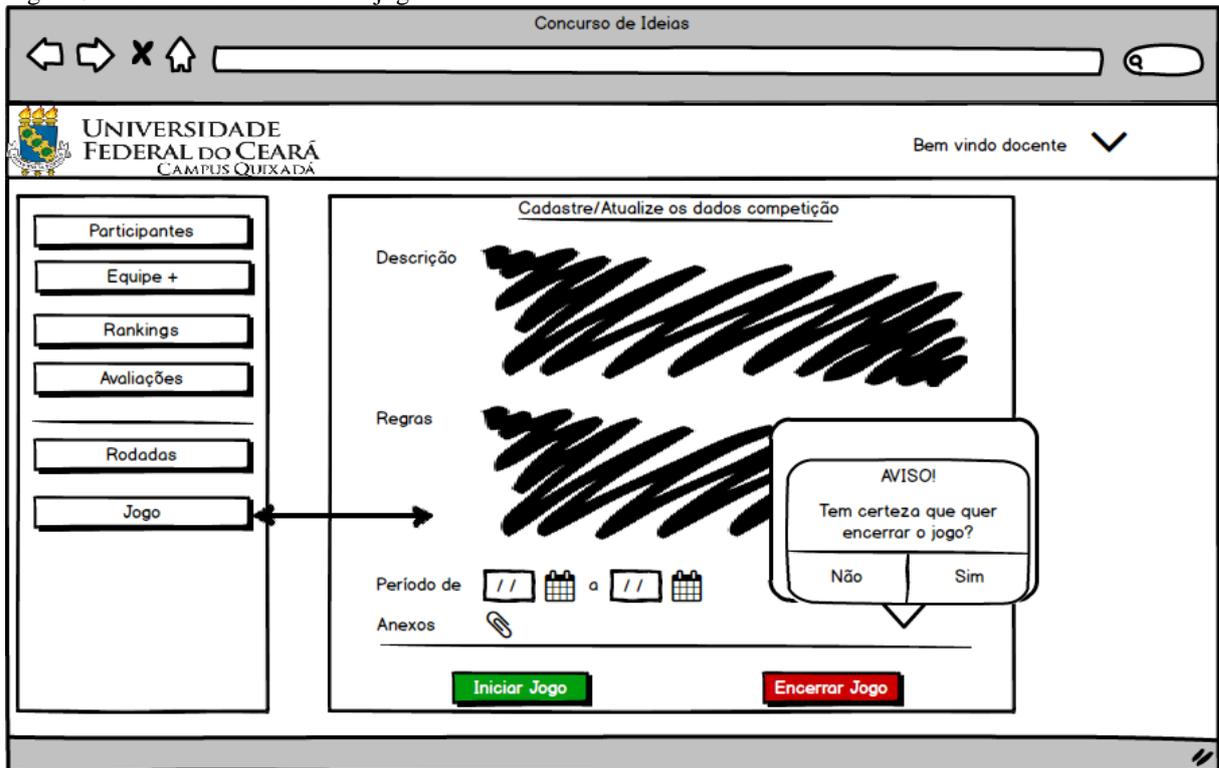


Fonte: Elaborado pelo autor

¹⁶ <https://balsamiq.com/>

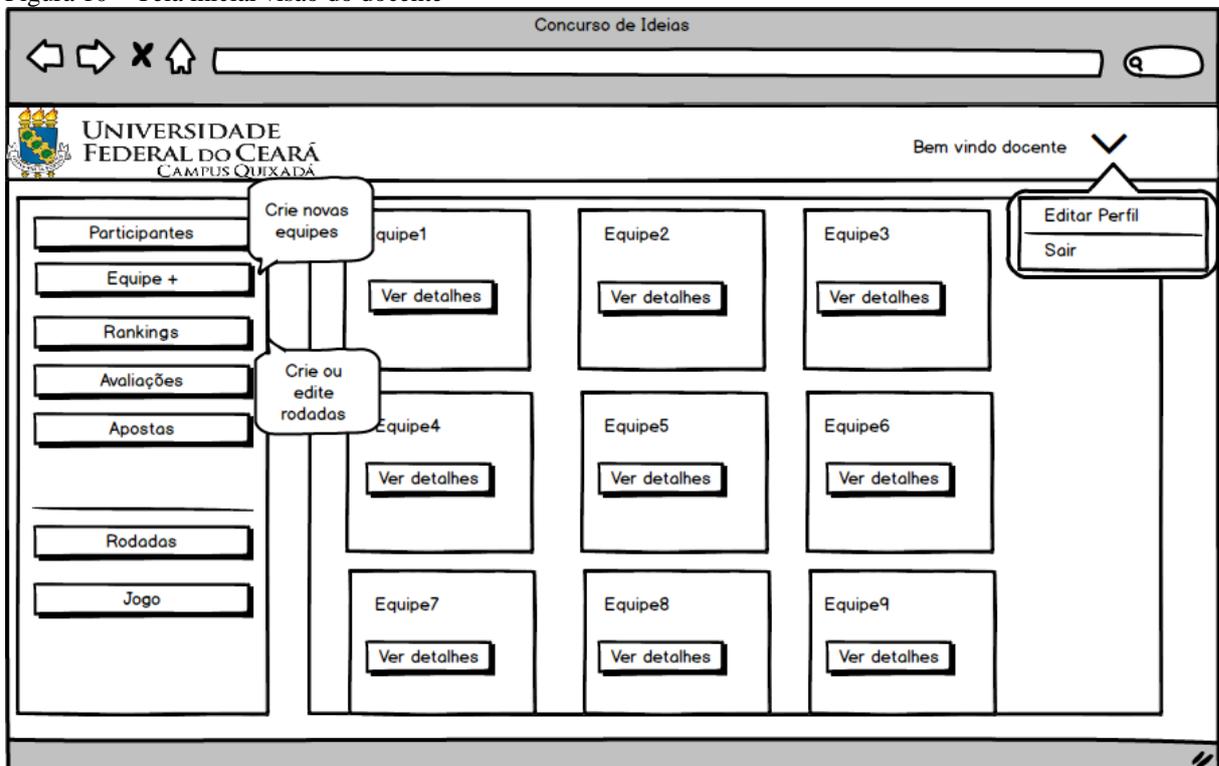
¹⁷ <http://pingendo.com/>

Figura 9 – Tela de detalhes de um jogo



Fonte: Elaborado pelo autor

Figura 10 – Tela inicial visão do docente



Fonte: Elaborado pelo autor

Inicialmente tive dificuldade em mapear histórias de usuários para representações em protótipos, devido à minha falta de experiência nesse tipo de atividade. Porém com a

criação dos primeiros protótipos foi possível notar a falta de algumas histórias de usuários que representam requisitos básicos do sistema, como efetuar login, por exemplo.

Cogitei inicialmente utilizar layouts baseados em HTML e CSS gerados a partir dos protótipos feitos na ferramenta Pingendo como ilustrado na Figura 11. Porém durante o desenvolvimento percebi que ao se criar um novo protótipo na ferramenta é gerado um novo CSS para a página, duplicando o CSS já existente em outra página e com as mesmas características. Dessa forma para adaptar a estilização gerada pela ferramenta seria necessário muito esforço, o que retardaria o desenvolvimento. Também não se fez necessária a utilização do HTML gerado, tendo em vista que se tornou mais simples modularizar o layout do sistema, reutilizando-se assim os componentes existentes para manter um padrão da interface nas novas páginas.

Figura 11 – Protótipo ferramenta Pingendo



Fonte: Elaborado pelo autor

Após o desenvolvimento das primeiras funcionalidades, tornou-se mais cômodo sanar dúvidas sobre layout diretamente nas telas criadas para o sistema. Assim a criação de protótipos não se fez necessária a partir do momento que novas telas foram criadas.

5.1.3 Validação dos requisitos

Para melhor detalhamento das histórias de usuários, validar novas histórias inseridas e consequentemente confirmação dos requisitos, foi feita uma reunião com a cliente. Primeiramente foi criado um roteiro (Anexo B) para a reunião, com o intuito de não passarem despercebidos pontos importantes desse momento.

Meu conhecimento sobre o negócio ajudou em diversos aspectos desta e das demais etapas. Conhecimento este adquirido durante a primeira aplicação da gamificação da disciplina de Empreendedorismo, que aconteceu em minha turma de Engenharia de Software,

e participação de algumas reuniões para levantamento de requisitos para o desenvolvimento da primeira iniciativa do projeto Jogo de Empreendedorismo, onde requisitos para o sistema eram discutidos.

A reunião ficou dividida em dois dias, pois o tempo despendido durante a revisão dos requisitos ultrapassou as expectativas, que era uma reunião de 2h (duas horas). O tempo extra deveu-se ao tamanho do *Backlog* do produto e meu aprofundamento nos detalhes de cada história. Utilizei protótipos para enriquecer as reuniões em busca de identificar e validar novos requisitos, assim como ajudar na tomada de decisões iniciais para o desenvolvimento. Durante o processo novas histórias surgiram, juntamente com novos critérios de aceitação para histórias existentes, o que é natural durante o processo de validação dos requisitos, e histórias que inseri foram aceitas pela cliente como requisitos válidos para a solução esperada.

Ao final da validação, o *Backlog* do produto ficou com 40 (quarenta) histórias de usuários divididas nos mesmo seis temas iniciais. As novas histórias acrescentadas foram:

Aposta

- APO4: Como docente, eu posso delimitar valor a ser liberado para uma rodada;

Controle

- CTR10: Como docente, eu posso reabrir o período de submissão de artefatos para equipe específica;
- CTR11: Como aluno, eu posso escolher uma opção de aposta para a última rodada.

5.1.4 Definição de *Sprints*

De acordo com a validação dos requisitos por parte da cliente, busquei definir *Sprints* que agrupassem histórias de forma a desenvolver um fluxo de uso para o sistema. Por exemplo, o usuário precisa primeiramente criar seu cadastro no sistema, para se logar, e então poder criar jogos. Esta forma de definir *Sprints* foi utilizada durante todo o processo de desenvolvimento.

Seguindo o processo definido para implementação, foram definidas 8 (oito) *Sprints*, que englobam 39 (trinta e nove) histórias de usuários de diversas complexidades.

- A primeira *Sprint* trata do controle de fluxo inicial do sistema, englobando histórias de cadastro de usuários, login no sistema, cadastro e atualização de jogos, e configuração de formulários;
- A segunda *Sprint* concentra-se na configuração de equipes para um jogo;

- A terceira *Sprint* concentra-se no fluxo de controle inicial para uma rodada, iniciar e encerrar um jogo e submeter artefatos para rodada;
- A quarta *Sprint* concentra-se nas atividades de visualização das últimas submissões de uma rodada, e avaliação para entregas de uma rodada por alunos e docentes;
- A quinta *Sprint* aborda funcionalidades diversas, mas concentra-se na criação de um serviço pago para uma rodada, que é a reabertura de submissões para uma equipe específica;
- A sexta *Sprint* concentra-se no cálculo de notas para equipes e alunos, além de gerar e publicar rankings de uma rodada, publicar avaliações e configurar fatores de apostas.
- A sétima *Sprint* concentra-se em apostas, ou investimentos, realizados pelos alunos nas equipes que efetuaram entregas na rodada.
- A oitava *Sprint* concentra-se na criação de uma consultoria para uma rodada, assim como atividades de configuração para melhorar a efetividade do sistema.

5.2 Projetar arquitetura

Inicialmente para o desenvolvimento da solução, decidi utilizar a arquitetura presente no NPI, o que facilitou em vários aspectos o desenvolvimento. Com essa adoção de arquitetura base, pude tirar proveito de configurações iniciais dos projetos pertencentes ao portfólio do Núcleo de Práticas.

Criei um Documento de Arquitetura, disponível no Apêndice A, em que expus visões arquiteturais lógica, de processos e de desenvolvimento.

Durante todo o processo de desenvolvimento, gerei novas modelagens que me ajudaram no entendimento de requisitos e forneceram detalhes de como proceder com o desenvolvimento da solução. A modelagem de novas entidades surgiu pela necessidade de entender como novas entidades do sistema se relacionam, não somente pela alimentação do documento de arquitetura presente no Apêndice A, que poderá ser utilizado por futuros desenvolvedores como referência.

Sendo assim, este momento de arquitetar a solução, de forma que se adeque aos requisitos, foi contemplado durante todo o processo de desenvolvimento. Modelando e inserindo essas novas modelagens no documento de arquitetura para posterior consulta.

5.3 Implementar solução

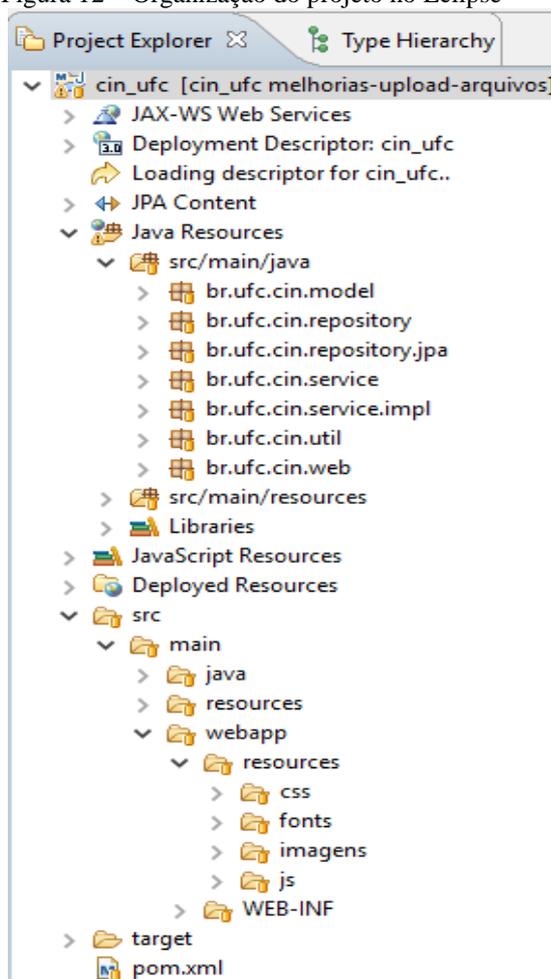
Esta seção trata da implementação da solução de acordo com os requisitos definidos para a solução. Nas próximas subseções apresento detalhes sobre as etapas de configuração inicial do ambiente de desenvolvimento e da implementação de todas as *Sprints* definidas para o sistema.

5.3.1 Ambiente de desenvolvimento

Utilizando-se das ferramentas apresentadas na seção 3.3 criei um projeto *Maven* na IDE Eclipse, onde é possível escolher qual tipo de projeto deseja-se gerenciar utilizando-se do *Maven*. No meu caso foi um projeto *Spring MVC*, de forma que se assemelhasse aos projetos desenvolvidos no NPI. Configurações adicionais foram acrescentadas ao projeto, pois na IDE não é criado um projeto com hierarquia de diretórios e modo de configuração iguais a estratégia utilizada no núcleo de práticas.

Na Figura 12 pode-se observar como o projeto ficou organizado na IDE.

Figura 12 – Organização do projeto no Eclipse

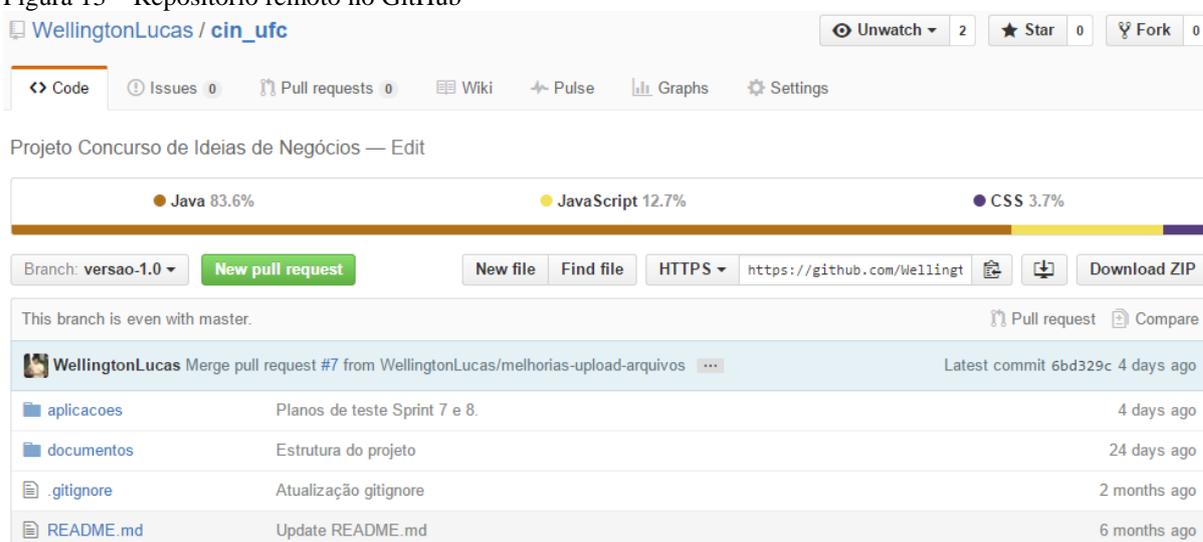


Fonte: Elaborado pelo autor

Criei um repositório remoto no GitHub para a solução, onde a aplicação¹⁸ desenvolvida está disponível. Ao passo que novas funcionalidades foram sendo desenvolvidas, por questões de segurança e integridade, foram sendo adicionadas ao repositório remoto, garantindo assim que caso algum problema ocorresse na máquina de trabalho o repositório remoto teria a última versão desenvolvida. O uso do plug-in EGit mais a ferramenta Git foram de grande valia para o controle de versões do sistema, onde juntamente com o GitHub pude sempre que necessário voltar o sistema para versões anteriores, de acordo com o histórico de mudanças registradas no GitHub.

A Figura 13 mostra como o repositório remoto é apresentado.

Figura 13 – Repositório remoto no GitHub



Fonte: Elaborado pelo autor

5.3.2 Implementação da solução

Para a implementação da solução inicialmente tive que configurar corretamente o *framework* adotado para o desenvolvimento, o que gerou alguns problemas que demandaram alguns dias de pesquisa para serem resolvidos. Alguns dos problemas surgiram porque me baseei nas configurações dos projetos do NPI, e algumas configurações não interpretei e adaptei corretamente. Problemas ocasionados pela falta de experiência em iniciar um projeto deste tipo do zero, mesmo tendo me baseado em projetos já existentes. Apesar dos problemas ocorridos em configurar inicialmente o projeto, possuir um exemplo de projeto com configuração pronta ajudou a diminuir o tempo gasto nesta etapa, permitindo assim que o desenvolvimento pudesse ser iniciado de maneira relativamente rápido.

¹⁸ https://github.com/WellingtonLucas/cin_ufc

Resolvidos os problemas de configurações iniciais, passei para o desenvolvimento propriamente dito. Onde pude aproveitar dados iniciais do projeto anterior de solução, como mapeamento entre entidades, porém tudo mudou a partir do momento que decidimos criar um sistema que gerencie jogos e não um sistema que gerencia um jogo. Com essa decisão a complexidade da solução aumentou.

Outra importante decisão de negócio tomada no desenvolvimento trata-se do papel atribuído aos usuários que utilizarão o sistema, inicialmente pretendia-se utilizar dois papéis, “PROFESSOR” e “ALUNO”, porém esta alternativa limitaria o uso da solução. Uma alternativa empregada para se obter maior universalização da solução web foi, a não atribuição de papéis fixos para usuários, ou seja, se um usuário criar um Jogo, então ele será “professor” desse jogo, se um usuário for vinculado a um Jogo, então ele será “aluno” desse jogo.

A seguir discutirei os resultados obtidos em cada *Sprint* definida para a solução web pretendida.

a) *Sprint 1*

A primeira *Sprint* definida conta com as seguintes histórias de usuários:

- CON1: Como participante, eu posso cadastrar-me no jogo;
- CON2: Como docente, eu posso cadastrar/atualizar as informações gerais de um jogo;
- CON8: Como docente, eu posso criar um novo jogo;
- AVA6: Como docente, eu posso configurar um formulário de múltipla escolha e gabarito; e
- CTR9: Como participante, eu posso me logar no jogo.

Esta *Sprint* concentra-se nas configurações iniciais do fluxo de uso do sistema. Onde interessados em utilizar a solução podem cadastrar-se, acessa-la, criar ou atualizar jogos de sua autoria e configurar formulários que podem ser utilizados em qualquer jogo que o interessado tenha criado.

A criação, visualização, edição, exclusão e a possibilidade de copiar formulários tem grande impacto no desenvolvimento da solução. Através dos formulários criados na aplicação alunos e docente poderão avaliar entregas de equipes para uma determinada rodada. Esta funcionalidade tem grande relevância porque elimina a necessidade da criação de formulários através do uso de outras aplicações.

Na Figura 14 é apresentada a tela de detalhes de um formulário criado na solução.

Figura 14 – Formulário criado na solução

Fonte: Elaborado pelo autor

Durante esta *Sprint* iniciei o desenvolvimento de funcionalidades que pertencem ao escopo da *Sprint* definida posteriormente, como a vinculação de usuários a um jogo e organização de equipes, porém estas funcionalidades foram remanejadas.

Após o término do desenvolvimento desta *Sprint* as funcionalidades foram verificadas pelo gerente, onde foi definido que todas as histórias, exceto AVA6, estavam como aceitas, permanecendo AVA6 como em progresso. Na mesma ocasião foram definidas as *Sprints* 2, 3 e 4.

b) *Sprint* 2

A segunda *Sprint* definida conta com as seguintes histórias de usuários:

- CON5: Como docente, eu posso vincular participantes a um jogo;
- CON3: Como docente, eu posso organizar equipes de participantes para um jogo;
- EQU1: Como membro de equipe, eu posso visualizar as informações dos demais membros;
- EQU3: Como membro de equipe, eu posso editar as informações da minha equipe; e
- EQU4: Como membro de equipe, eu posso editar a ideia de negócio.

Esta *Sprint* concentra-se na configuração de equipes para um jogo. Onde docentes podem criar, vincular participantes, ver detalhes, editar e excluir equipes do jogo. Membros

de uma equipe além de visualizar os detalhes da sua equipe, também podem editar os dados da sua equipe e ver informações básicas dos demais membros da equipe.

Para a história CON5 além da vinculação de participantes a um jogo, também desenvolvi a opção de desvinculação, apesar de inicialmente não especificado explicitamente na história de usuário, notei a necessidade da funcionalidade de desvinculação. Nos comentários dessa história acrescentei posteriormente nota sobre esta funcionalidade.

A configuração de equipes para o jogo tem grande importância para os objetivos da solução web, cada equipe representa uma empresa. Através de equipes, alunos podem submeter entregas para uma rodada, o que é fundamental para o andamento do jogo, pois essas entregas serão avaliadas pelo docente, formando assim uma nota que fará parte da média final dos alunos na disciplina.

A Figura 15 apresenta como uma equipe está organizada em um jogo na visão de um docente.

Figura 15 – Detalhes de uma equipe

The screenshot displays a web application interface for a game. At the top, there is a blue header with the text 'CIN Concurso de Idéias de Negócio' on the left and 'Olá Wellington!' with a home icon and a dropdown arrow on the right. A left sidebar contains a menu with options like 'Home Jogo', 'Empresas', 'Rodadas', 'Rankings', 'Participantes', 'Formulários', 'Avaliações', 'Histórico de Notas', 'Vincular Membros', and 'Voltar'. The main content area is titled 'Detalhes da Empresa' and features a large black Apple logo. To the right of the logo, it shows 'Nome: Apple Store' and 'Saldo (R\$): 0'. Below the logo, there is a section for 'Ideia de Negócio' with the text 'Nossa ideia de negócio consistem em três aspectos:' and a 'Leia mais' link. At the bottom of this section are three buttons: 'Editar' (blue), 'Inativar' (orange), and 'Excluir' (red). Below this is a 'Membros' section with a table listing team members.

Nome	Sorenome	Curso	Email
Priscila	Silva	Engenharia de Software	pri@gmail.com

Below the table, there is a 'Desvincular' button with a red 'x' icon.

Fonte: Elaborado pelo autor

c) *Sprint 3*

A terceira *Sprint* definida conta com as seguintes histórias de usuários:

- CTR4: Como docente, eu posso iniciar um jogo;
- CTR3: Como docente, eu posso encerrar um jogo;
- EQU2: Como membro de equipe, eu posso visualizar informações de contato do docente;
- CTR7: Como docente, eu posso iniciar uma rodada; e

- EQU5: Como membro de equipe, eu posso submeter para entrega de rodada.

Esta *Sprint* concentra-se no fluxo de controle inicial para uma rodada, iniciar e encerrar um jogo e submeter artefatos para rodada.

A história CTR7 engloba todo o fluxo de controle para uma rodada, pois não existe outra história que especifique as demais funcionalidades relacionadas ao gerenciamento da rodada, logo é uma história que possui grande impacto no sistema. Não criei mais histórias fragmentando CTR7 porque achei desnecessário, devido ao meu conhecimento dos requisitos do sistema.

O controle de rodadas é fundamental para o sistema, pois através das rodadas o jogo atinge seus objetivos, que é a construção de um Plano de Negócio (PN), fazendo que todos os alunos da disciplina leiam e avaliem as demais entregas dos colegas para uma rodada, ganhando assim visões diferentes sobre como construir um PN. Cada rodada representa uma etapa na construção do documento. Muitas funcionalidades do sistema estão relacionadas a uma rodada e dependem do seu bom funcionamento. No decorrer das *Sprints* essas funcionalidades estão melhor especificadas.

A Figura 16 mostra como uma rodada está representada no sistema na visão de um docente.

Figura 16 – Detalhes de uma rodada

Rodada: Rodada 2

Nome: Rodada 2

Início: 07/01/2016

Término: 14/01/2016
A rodada se encerra 14/01/2016 - 00:00:00

Submissões: 08/01/2016
O prazo de submissões se encerra 08/01/2016 - 00:00:00

Prazo de avaliações: 10/01/2016
O prazo de avaliações se encerra 10/01/2016 - 00:00:00

Valor de aposta: R\$ 1.000

All in: Sim Não

Descrição: Descrição da empresa, produtos e serviços, mercado e competidores.
[Leia mais](#)

Formulário: Descrição da empresa, produtos e serviços, mercados e competidores

Modelo: Modelo.docx

Modelo: Selecione...

Fonte: Elaborado pelo autor

d) *Sprint* 4

A quarta *Sprint* definida conta com as seguintes histórias de usuários:

- AVA1: Como aluno, eu posso visualizar as submissões finais da rodada de todas as equipes;
- AOD1: Como docente, eu posso salvar uma avaliação para uma entrega de equipe;
- AVA2: Como aluno, eu posso submeter uma avaliação individual de uma entrega de equipe; e
- CTR6: Como docente, eu quero encerrar o período de submissão de artefatos.

Esta *Sprint* concentra-se nas atividades de visualização e avaliação das últimas submissões de uma rodada, por alunos e docentes, além da delimitação de um prazo para submissões.

Inicialmente adaptei a configuração de uma rodada para suportar uma data final para as submissões, com isso uma rodada passa a estabelecer uma data limite para que alunos possam submeter suas respectivas entregas.

Após a data limite do prazo de submissão, alunos e docente poderão acessar as últimas submissões das equipes para uma rodada, com isso poderão avaliar as respectivas submissões. Para avaliação de uma entrega é disponibilizado um formulário de múltipla escolha criado pelo professor do jogo, a associação entre formulário e rodada acontece no momento da criação ou edição de uma rodada, onde torna-se obrigatório que o docente crie primeiramente o formulário para efetuar o cadastro de uma rodada.

Na Figura 17 é apresentada uma visão da listagem das últimas entregas de uma rodada.

Figura 17 – Listagem ultimas submissões de uma rodada

Últimas Submissões				
10	resultados por página		Pesquisar	<input type="text"/>
Empresa	Documento	Data de Submissão	Avalie	
Apple Store	entrega-rodada-1.doc	11/01/2016 - 21:10:03	Avaliar	
Software Livre Quixadá - SLQ	entrega-rodada-1.doc	11/01/2016 - 21:11:12	Avaliar	
Organizações UFC	entrega-rodada-1.doc	11/01/2016 - 21:11:59	Avaliar	

Fonte: Elaborado pelo autor

Adotei uma estratégia para facilitar o discernimento de quais avaliações já foram efetuadas pelo usuário. A visão de sistema apresentada Figura 17 pertence a um docente, porém para ambas as visões a listagem é muito semelhante, para alunos diferencia-se por aparecer a opção de efetuar investimentos numa determinada empresa, que será apresentada mais adiante em sua respectiva *Sprint*. Percebe-se que existe um botão de cor verde enquanto os demais estão azuis, a cor verde está representando que a entrega já foi avaliada e o azul as faltantes.

Cada entrega possui uma data e horário específico, que representam o momento em que algum membro de equipe efetuou uma entrega. De acordo com o momento em que cada entrega foi efetuada também é gerado um histórico de entrega para cada equipe, o que garante uma melhor segurança por parte do professor em afirmar qual entrega foi a última de cada empresa, caso seja necessário.

Após o termino desta *Sprint* foram definidas juntamente com o gerente as *Sprints* 5, 6 e 7. Neste momento as histórias das *Sprints* desenvolvidas foram revisadas, onde notou-se que precisava-se verificar melhor o controle de acesso do sistema, para todas as funcionalidades que necessitassem, além da correção de pendências notadas na revisão.

e) ***Sprint 5***

A quinta *Sprint* definida conta com as seguintes histórias de usuários:

- EQU6: Como membro de equipe, eu posso visualizar o histórico de submissões para uma entrega de rodada;
- CTR10: Como docente, eu posso reabrir o período de submissão de artefatos para equipe específica; e
- AVA4: Como docente, eu posso visualizar as avaliações individuais dos alunos.

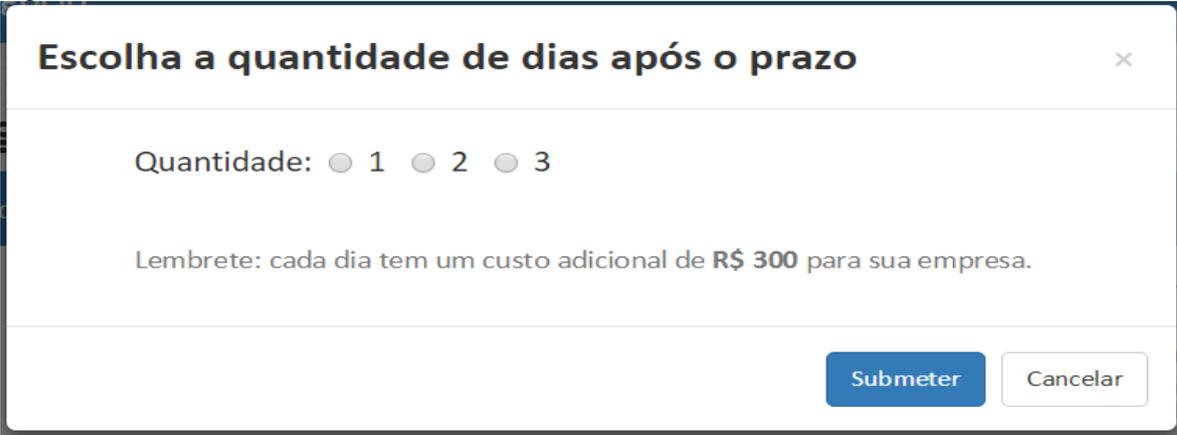
Esta *Sprint* concentra-se em funcionalidades diversificadas do sistema, a visualização do histórico de submissões de uma equipe, a criação de um serviço pago para uma rodada, que é a reabertura de submissões para uma equipe específica, e visualização de como os alunos avaliaram as empresas.

Iniciei o desenvolvimento pela funcionalidade de exibir o histórico de submissões para uma equipe, funcionalidade que exibe uma linha do tempo com o usuário que efetuou a entrega, o arquivo submetido, a rodada e a data de submissão. Esta funcionalidade pode ser utilizada para sanar dúvidas sobre qual foi a última entrega de uma rodada para uma equipe.

Reabrir uma rodada para equipe específica é um serviço oferecido pelo docente do jogo, onde membros de uma equipe podem solicitar uma quantidade de dias após o prazo de submissões, para que a equipe possa efetuar entregas neste prazo. Esta funcionalidade impacta diretamente o prosseguimento de uma rodada, se uma equipe requisitar a reabertura isso implica que o período, de avaliações das entregas, só iniciará após o término do prazo de prorrogação requisitado.

A Figura 1 apresenta a tela de pedido de reabertura de uma rodada, informando o custo de cada dia após o prazo.

Figura 18 – Pedido de reabertura rodada



Escolha a quantidade de dias após o prazo

Quantidade: 1 2 3

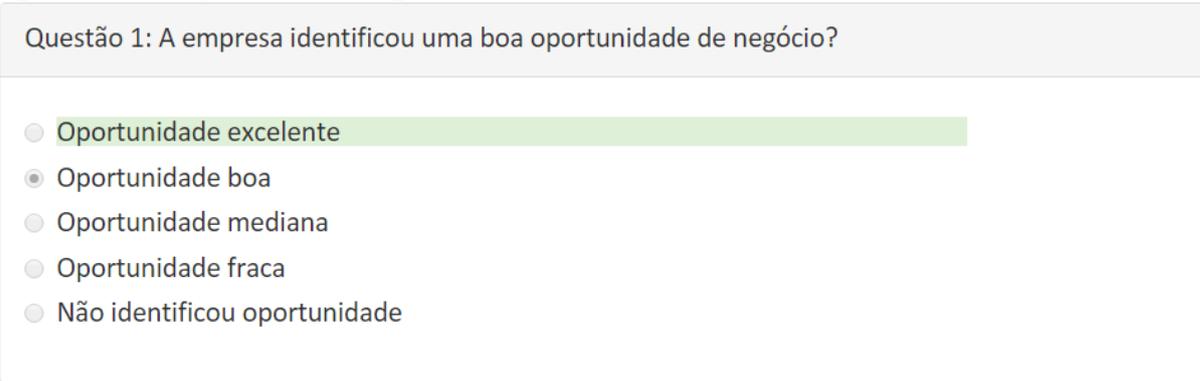
Lembrete: cada dia tem um custo adicional de **R\$ 300** para sua empresa.

Submeter Cancelar

Fonte: Elaborado pelo autor

A visualização das avaliações dos alunos é importante para que o docente possa obter feedback de como os alunos estão avaliando as entregas dos demais colegas. Na página de detalhes de um aluno é exibida a opção de visualizar avaliações, onde são listadas todas as avaliações que o aluno efetuou no jogo até então. Para a visualização é exibido o respectivo formulário utilizado na rodada, com as respostas do aluno juntamente com as respostas do docente, destacadas, para que se possa comparar os resultados. Na Figura 19 é apresentada a avaliação do aluno com o respectivo gabarito do docente destacado.

Figura 19 – Avaliação do aluno com gabarito



Questão 1: A empresa identificou uma boa oportunidade de negócio?

Oportunidade excelente

Oportunidade boa

Oportunidade mediana

Oportunidade fraca

Não identificou oportunidade

Fonte: Elaborado pelo autor

f) *Sprint 6*

A sexta *Sprint* definida conta com as seguintes histórias de usuários:

- AVA3: Como aluno, eu posso visualizar minhas avaliações individuais anteriores;
- AVA5: Como aluno, eu posso visualizar minha nota individual;
- EQU8: Como membro de equipe, eu posso visualizar os resultados da equipe numa rodada;
- CTR1: Como docente, eu posso publicar as avaliações e fatores para as entregas de uma rodada;
- CTR2: Como docente, eu posso gerar e publicar os rankings; e
- AOD2: Como docente, eu posso configurar o fator de aposta para uma equipe.

Esta *Sprint* concentra-se no cálculo de notas para equipes e alunos, um dos pontos críticos do sistema, além de gerar e publicar rankings de uma rodada, publicar avaliações e configurar fatores de apostas.

Inicialmente adaptei a visualização das avaliações individuais dos alunos, por parte do docente, também para o aluno, buscando o mesmo propósito.

O cálculo de notas para os alunos é uma das funcionalidades mais importantes para a solução web, pois as notas fazem parte da média final de um aluno na disciplina de Empreendedorismo. Esta funcionalidade eliminará a necessidade do uso de planilhas eletrônicas para efetuar cálculos complexos sobre as avaliações dos alunos.

Existe um requisito relacionado a forma como o cálculo das notas dos alunos deveria ser realizado, porém possuía uma lacuna que poderia prejudicar o bom funcionamento do jogo. Analisando os requisitos desenvolvi uma solução para esses cálculos que eliminava a lacuna anterior. Posteriormente apresentei a nova estratégia à cliente, sendo aprovada a nova forma de avaliação.

O cálculo dos resultados de uma equipe numa rodada também representa uma das funcionalidades mais importantes do sistema, pois fornece uma das notas que compõem a média final dos alunos na disciplina. Em reunião com a cliente apresentei a estratégia adotada durante o desenvolvimento da funcionalidade, onde a cliente forneceu *feedback* positivo a respeito. Na Figura 20 é apresentado o histórico de notas de um aluno, onde são listadas suas notas individuais e da equipe que participa.

Figura 20 – Histórico de notas usuário e equipe

Visualizar histórico Priscila Silva

Suas Notas	
Rodada	Nota
Rodada 1	6,5
Rodada 2	9,12
Média: 7,81	

Notas da Empresa				
Rodada	Fator de aposta	Nota		
Rodada 1	2	10		
Rodada 2	2	10		
Média: 10				

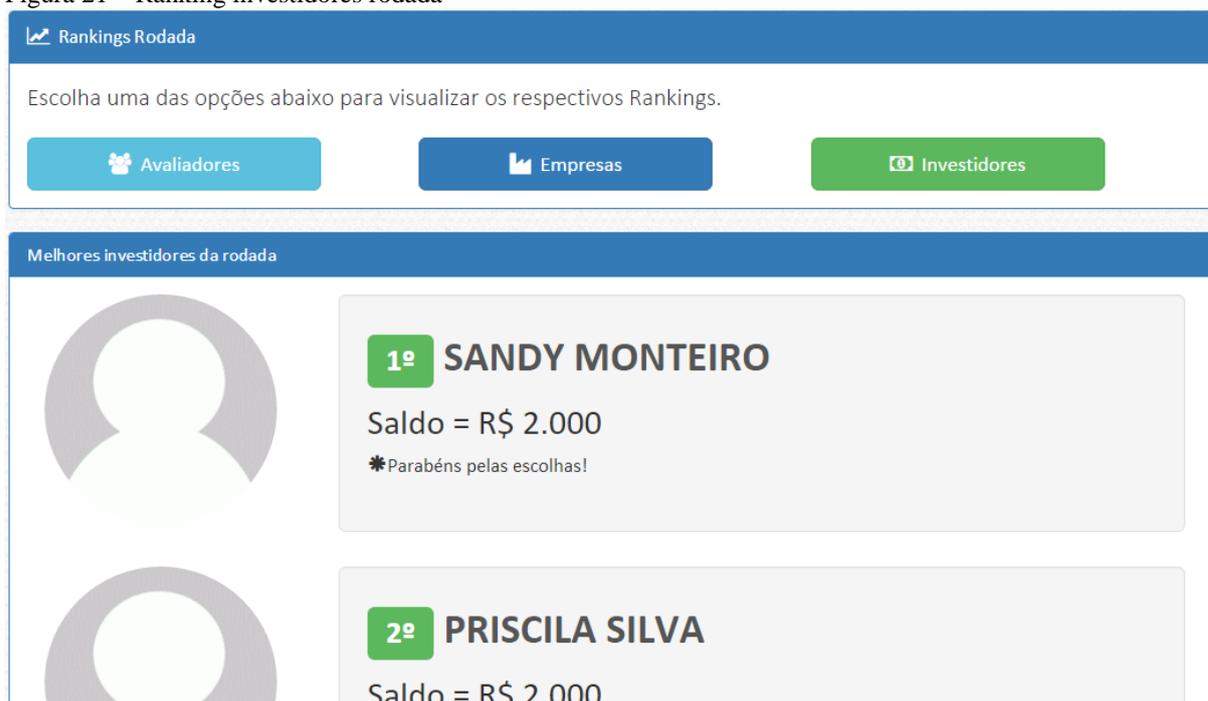
Fonte: Elaborado pelo autor

A publicação de avaliações e fatores de aposta para visualização final ocorre apenas após a publicação dos rankings da rodada.

Gerar e publicar os rankings de uma rodada é parte importante da gamificação da disciplina, mostrando como os alunos estão em relação as avaliações, investimentos e as posições de suas equipes em relação aos investimentos recebidos na rodada, assim como rankings gerais do jogo, que mostram os melhores investidores e as melhores equipes. De acordo com a posição no ranking, alunos e equipes recebem prêmios ao final do jogo.

Na Figura 21 é apresentado o ranking com os melhores investidores da rodada, onde é apresentado em ordem decrescente com base no saldo acumulado na rodada por cada aluno.

Figura 21 – Ranking investidores rodada



Fonte: Elaborado pelo autor

g) *Sprint 7*

A sétima *Sprint* definida conta com as seguintes histórias de usuários:

- APO4: Como docente, eu posso delimitar valor liberado para uma rodada;
- APO1: Como investidor, eu posso apostar dinheiro, do valor liberado para a rodada, em uma ou mais equipes em uma rodada;
- EQU7: Como membro de equipe, eu posso visualizar o saldo da equipe;
- APO3: Como docente, eu posso visualizar todas as apostas dos alunos; e
- CTR5: Como docente, eu posso encerrar o período de apostas;

Esta *Sprint* concentra-se nas apostas, ou investimentos, realizados pelos alunos nas equipes que efetuaram entregas na rodada.

Existem dois tipos de rodadas definidas para sistema, que se diferenciam com base no valor liberado para apostas, onde um docente pode explicitamente definir um valor liberado para a rodada, ou definir a rodada como “*All in*”, que se trata de liberar todo o saldo acumulado no jogo para ser o valor de aposta dos alunos na rodada. Inicialmente alterei a configuração de uma rodada para receber o valor liberado para apostas, juntamente com uma opção “*All in*” que delimita se os investidores poderão, ou não, utilizar todo o saldo que acumularam durante o jogo para efetuar apostas na rodada.

O período de apostas é paralelo ao período de avaliações de uma rodada, ou seja, ao se encerrar o período de avaliações também se encerra o período de apostas.

Poder apostar, ou investir, em equipes numa rodada constitui parte importante do andamento do jogo. Através dos valores é que se definem retornos para equipes e alunos, em função do fator de apostas das equipes, definem-se os rankings que se baseiam no retorno dos investimentos, consequentemente definindo o melhor investidor e a melhor equipe do jogo.

A Figura 22 apresenta a listagem das últimas submissões de uma rodada na visão de um aluno. Através a opção “Apostar” o aluno pode utilizar o valor liberado para investir na equipe que achar conveniente. O saldo do aluno é exibido na tela para que o aluno saiba quanto ainda pode investir nas equipes.

Figura 22 – Últimas submissões visão do aluno

Rodada: Rodada 2 - Seu saldo é R\$ 0

Últimas Submissões

10 resultados por página Pesquisar

Empresa	Documento	Data de Submissão	Avalie	Aposte
Empresa S	Entrega R2.odt	10/01/2016 - 21:17:29	Avaliar	Apostar
Empresa P	Entrega-reabertura.odt	10/01/2016 - 21:30:40	Avaliar	Apostar

Mostrando de 1 até 2 de 2 registros Anterior **1** Próximo

Fonte: Elaborado pelo autor

A funcionalidade de gerar e publicar rankings, *Sprint* anterior, foi concluída somente após o encerramento desta *Sprint*, pois um dos rankings envolve os valores investidos pelos alunos na rodada.

h) *Sprint 8*

A sexta *Sprint* definida conta com as seguintes histórias de usuários:

- EQU9: Como membro de equipe, eu posso adquirir um serviço;
- APO2: Como investidor, eu posso visualizar minhas apostas anteriores;
- CON9: Como docente, eu posso adicionar uma imagem de apresentação para o jogo;
- CON10: Como participante, eu posso alterar meus dados de cadastro;

- CON11: Como participante, eu posso alterar minha imagem de perfil; e
- CON12: Como membro de equipe, eu posso adicionar uma imagem que defina minha empresa.

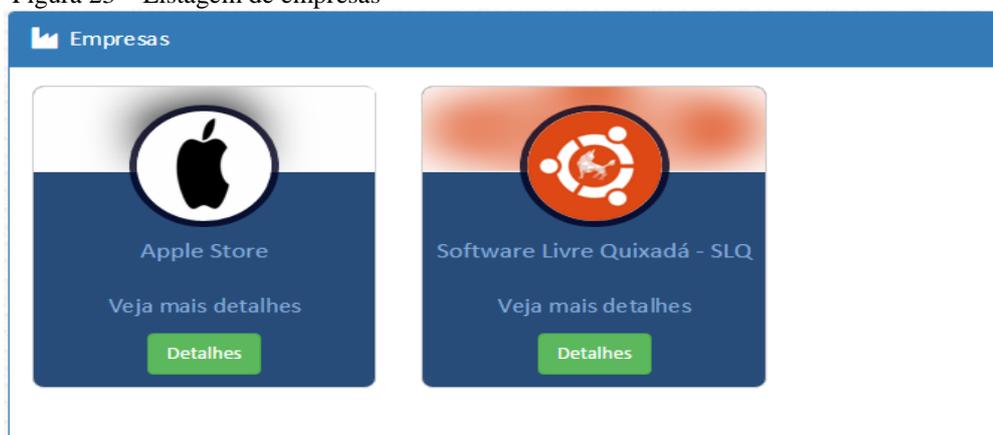
Esta *Sprint* concentra-se na criação de um serviço, para uma rodada, pelo docente, mais especificamente uma consultoria. Também é abordada a visualização das apostas anteriores efetuadas pelo aluno, assim como atividades de configuração para melhorar a efetividade do sistema.

Adquirir uma consultoria faz parte de um dos serviços oferecidos numa rodada. Para uma consultoria optou-se por criar uma descrição e um valor, que representem ao aluno do que se trata e quanto custa uma consultoria. Para esse serviço é mostrada uma lista de pedidos para o docente, onde serão apresentadas solicitações efetuadas pelas equipes. O valor de uma consultoria somente será debitado, do saldo de uma equipe na rodada, após a confirmação de aceite pelo docente.

As histórias de configuração CON9, CON10, CON11 e CON12 foram acrescentadas ao *Backlog* nesta última *Sprint*, buscando melhorar a apresentação e uso do jogo. Com essas histórias o docente pode adicionar uma identidade visual ao seu jogo, os usuários do sistema podem adicionar fotos em seus perfis e alterar seus dados cadastrais caso achem necessário, e membros de uma equipe podem definir um logo que represente sua empresa.

Requisitos de definição de imagens para jogos, equipes e usuários são opcionais, porém melhoram a apresentação de várias funcionalidades do sistema, como os rankings, o perfil do aluno, listagem dos jogos e equipes. Definir fotos para usuários também é importante para o docente, pois assim identifica com maior facilidade de qual aluno se trata cada perfil no jogo. A Figura 23 apresenta a listagem das empresas, ou equipes, com seus logos fictícios.

Figura 23 – Listagem de empresas



Fonte: Elaborado pelo autor

5.4 Planejar e executar testes

Durante o desenvolvimento do sistema foram executados testes de desenvolvedor, à medida que novas funcionalidades foram desenvolvidas, porém esses testes são muito específicos e não abordam cenários complexos durante o uso do sistema.

De acordo com os requisitos especificados para a solução, foram definidos e executados casos de teste com a apoio da ferramenta TestLink. Nesta seção serão apresentados os resultados obtidos com o processo de verificação e validação para a solução.

5.4.1 Especificar casos de teste

Inicialmente defini um projeto de testes na ferramenta, nesse projeto foram definidas suítes de testes que estão mapeadas de acordo com as histórias de usuários definidas no *Backlog* da solução, totalizando 35 (trinta e cinco) suítes. Os casos de teste foram especificados e agrupados de acordo com uma suíte de testes, buscando sempre abordar os testes de fronteira das funcionalidades, além do fluxo padrão de uso. Ao todo foram especificados 157 (cento e cinquenta e sete) casos de teste, o documento que contém todos os casos se encontra no repositório¹⁹ remoto do projeto.

O Quadro 1 representa um caso de teste presente no documento de testes do projeto.

Quadro 1 – Caso de teste criar rodada

Caso de Teste CIN-63: CTR7 - Criar uma rodada - Dados válidos		
Autor:	Wellington Lucas Moura	
<u>Objetivo do Teste:</u> Verificar se um usuário dono de um jogo consegue criar uma rodada, preenchendo todos os campos de forma válida.		
<u>Pré-condições:</u> Usuário estar logado no sistema, possuir um jogo e possuir formulário (s) criados.		
<u>#:</u>	<u>Ações do Passo:</u>	<u>Resultados Esperados:</u>
1	Escolha a opção de criar uma nova rodada;	Sistema redireciona para a página de criar rodada;
2	Preencha o campo "Nome da Rodada" com o valor "Identificação de uma oportunidade de negócio";	
3	Preencha o campo "Data de Início" com o valor "12/03/2016";	
4	Preencha o campo "Data de Término" com o valor "23/03/2016";	
5	Preencha o campo "Prazo de	

¹⁹ https://github.com/WellingtonLucas/cin_ufc/tree/master/documentos

	Submissão" com o valor "17/03/2016";	
6	Preencha o campo "Prazo de avaliação" com o valor "20/03/2016";	
7	Preencha a descrição da rodada com o valor "Rodada de identificação de uma oportunidade de negócio";	
8	Selecione um formulário para a rodada; e	
9	Confirme a criação da rodada.	O sistema cria uma nova rodada com os dados especificados e fornece feedback ao usuário.
<u>Tipo de execução:</u>	Manual	
<u>Prioridade:</u>	Médio	

Fonte: Elaborado pelo autor

Encontrei dificuldades em criar testes com a ferramenta, pela acessibilidade ruim, e pela forma como são criados casos de teste, porém o TestLink possui alguns recursos que ajudaram na criação dos testes, que foram sendo descobertos ao longo do uso. Um recurso que ajudou foi a possibilidade de copiar casos de testes, bastante útil para casos semelhantes, assim como o agrupamento de casos de testes em suítes de testes.

5.4.2 Executar testes

Após a definição dos casos de testes, criei oito planos de testes, cada um referente a uma *Sprint*, para cada plano de teste defini uma *Baseline* com as respectivas suítes de testes.

Ao executar os casos de teste pude perceber a necessidade de criar novos casos de testes e acrescentá-los nas suas respectivas suítes, buscando garantir que determinados cenários do sistema estão sendo verificados. Em termos de desenvolvimento e testes, uma clara vantagem surgiu, o testador também é o desenvolvedor das funcionalidades, com isso, corrigi funcionalidades à medida que erros surgiam.

O Quadro 2 apresenta a execução de um caso de teste. Todos os planos de testes podem ser consultados no repositório remoto do projeto.

Quadro 2 – Execução caso de teste criar rodada

Caso de Teste CIN-63: CTR7 - Criar uma rodada - Dados válidos				
Autor:	Wellington Lucas Moura			
<u>Objetivo do Teste:</u> Verificar se um usuário dono de um jogo consegue criar uma rodada, preenchendo todos os campos de forma válida.				
<u>Pré-condições:</u> Usuário estar logado no sistema, possuir um jogo e possuir formulário (s) criados.				
<u>#:</u>	<u>Ações do Passo:</u>	<u>Resultados</u>	<u>Notas de</u>	<u>Status de</u>

		<u>Esperados:</u>	<u>execução</u>	<u>execução</u>
1	Escolha a opção de criar uma nova rodada;	Sistema redireciona para a página de criar rodada;	O sistema se comportou como esperado.	Passou
2	Preencha o campo "Nome da Rodada" com o valor "Identificação de uma oportunidade de negócio";			Passou
3	Preencha o campo "Data de Início" com o valor "12/03/2016";			Passou
4	Preencha o campo "Data de Término" com o valor "23/03/2016";			Passou
5	Preencha o campo "Prazo de Submissão" com o valor "17/03/2016";			Passou
6	Preencha o campo "Prazo de avaliação" com o valor "20/03/2016";			Passou
7	Preencha a descrição da rodada com o valor "Rodada de identificação de uma oportunidade de negócio";			Passou
8	Selecione um formulário para a rodada; e			Passou
9	Confirme a criação da rodada.	O sistema cria uma nova rodada com os dados especificados e fornece feedback ao usuário.	O sistema se comportou como esperado.	Passou
<u>Tipo de execução:</u>	Manual			
<u>Prioridade:</u>	Médio			
<u>Baseline</u>	<i>Sprint 3</i>			
<u>Testador</u>	Wellington Lucas Moura			

Fonte: Elaborado pelo autor

Muitas funcionalidades apresentaram defeitos, descobertos nos casos de testes, o que mostra a importância da execução de testes em um projeto de software. Alguns defeitos encontrados estão relacionados ao caso em que usuários editam o HTML de uma página, para tentar manipular os dados submetidos numa requisição. Para esse tipo de vulnerabilidade foram criados casos de teste específicos, que buscam invalidar esse tipo de artifício no sistema. Por exemplo um problema identificado foi, usuários poderiam submeter pedidos de

reabertura de rodada após o prazo de submissão; e outro foi, usuários poderiam submeter entregas após o prazo de submissões. Para isso bastaria copiar o código HTML da página durante um período válido e copiá-lo após os prazos definidos. Após a identificação, essas funcionalidades passaram a verificar os prazos da rodada, evitando assim possíveis tentativas de burlar as regras do sistema.

5.4.3 Validar solução

Até a quarta *Sprint* as funcionalidades foram revisadas junto ao gerente do projeto, posteriormente as funcionalidades foram tratadas diretamente com a cliente. Ao longo de 11 (onze) meses foram realizadas duas validações com o gerente de projetos e três validações com a cliente. Pontualmente as funcionalidades foram revisadas junto a pesquisadora responsável pelo Projeto de Extensão que engloba o desenvolvimento da solução web pretendida.

Validar a solução com a cliente trouxe muitos benefícios ao projeto. Foram realizadas reuniões durante o processo para mostrar o funcionamento do sistema e validar funcionalidades. Durante essas reuniões surgiram melhorias que precisavam ser abordadas no sistema, além de erros que se mostravam em casos específicos, que eram prontamente anotados para serem corrigidos posteriormente. Foram verificadas diversas funcionalidades do sistema, dando-se maior atenção a geração de notas e rankings, pois são funcionalidades críticas da solução.

As múltiplas perspectivas das validações forneceram valiosos *feedbacks* em prol do bom funcionamento da solução.

6 CONSIDERAÇÕES FINAIS

Com a gamificação da disciplina de Empreendedorismo, ofertada na UFC Campus Quixadá, surgiu a necessidade de uma solução web específica e unificada que automatizasse a configuração, acesso a informações, controle de rodadas e fases de gerenciamento do jogo aplicado na disciplina. Como maneira de suprir a necessidade existente, foi desenvolvida uma solução web para apoiar a execução do jogo CIN.

Inicialmente foi definido um processo base que guiou todo o desenvolvimento do projeto, buscando facilitar futuras manutenções.

Para garantir que o projeto entregasse algo alinhado aos interesses do usuário, foi elaborado um processo iterativo para o desenvolvimento dos requisitos. Esse processo iniciou com uma análise dos requisitos existentes do projeto. Posteriormente foram definidos novos requisitos buscando definir melhor o escopo da solução. Os requisitos foram revisados durante o desenvolvimento junto com o cliente, servindo de suporte para planejamento das *Sprints*.

Para melhor entendimento e facilitar a manutenção futura da base de código, optou-se pela utilização da arquitetura de referência utilizada nos projetos do NPI. Como resultado para esta etapa foi gerado o documento de arquitetura presente no Apêndice A, que foi atualizado gradativamente durante o processo de desenvolvimento. A solução foi desenvolvida seguindo o modelo arquitetural adotado.

Ao todo foram concluídas 39 (trinta e nove) histórias de usuários, ou seja, todas as histórias que compõem o *Backlog* do produto, divididas em 8 (oito) *Sprints*. A solução permite o uso de qualquer funcionalidade que esteja especificada na versão final do *Backlog*, que pode ser consultado no documento de visão, Anexo A ou detalhadamente no repositório remoto do projeto.

Para tentar garantir um bom funcionamento para a solução foram especificados 157 (cento e cinquenta e sete) casos de teste, que estão divididos em 35 (trinta e cinco) suítes de testes, cada suíte representa uma história de usuário presente no *Backlog*. A quantidade de histórias difere da quantidade de casos de teste. Isso ocorre porque existem suítes que englobam mais de uma história. Após a definição, as suítes foram divididas em 8 (oito) planos de testes, onde cada um representa uma *Sprint*, para então serem executados. Ao final do processo de testes todos os casos passaram, o que permite assegurar 157 (cento e cinquenta e sete) casos em que a solução web se comporta como esperado. Para maiores detalhes, os casos de testes e suas execuções podem ser encontrados no repositório remoto do projeto.

Durante o processo de desenvolvimento houve momentos para validar a solução com *stakeholders*, principalmente com a cliente, que forneceram *feedbacks* de melhorias para funcionalidades e de usabilidade. Na última validação realizada a cliente deu aval para que a solução seja utilizada na próxima turma de Empreendedorismo.

No decorrer do processo foi possível perceber que as tecnologias adotadas facilitaram o desenvolvimento da solução web. A utilização do *Spring Framework* em conjunto com o padrão de desenvolvimento utilizado no NPI facilitou a separação de interesses no código fonte, organização do projeto, norteador o desenvolvimento de forma geral. A ferramenta easyBacklog facilitou muito a visualização e manutenção do *Backlog* da solução e suas *Sprints*. A utilização do Bootstrap facilitou na estilização das páginas, pois é amplamente utilizado por comunidades na web, que fornecem *templates* com várias funcionalidades úteis ao desenvolvimento. O uso do Bootstrap também permite que a aplicação possa ser utilizada através de dispositivos móveis, mantendo as características da solução. O uso da ferramenta Redmine mostrou-se desnecessário durante o processo, devido o projeto contar com apenas um desenvolvedor o acompanhamento das atividades ter sido realizado em reuniões periódicas com o gerente e a ferramenta easyBacklog possuir controle de status para as histórias de usuários.

Espera-se que a solução supra as necessidades dos professores da disciplina de Empreendedorismo, podendo ser utilizada não apenas no âmbito local do campus Quixadá, mas sendo um produto que possa ser incorporado em disciplinas de Empreendedorismo em todo Brasil. Essa possibilidade é real devido a forma como a solução foi desenvolvida, abstraindo seu uso para qualquer pessoa que utilize a mesma metodologia empregada na gamificação da disciplina de Empreendedorismo. A solução está disponível como código aberto.

As metodologias e tecnologias adotadas nesse projeto servem para orientar futuros desenvolvedores de soluções web semelhantes, apresentando sugestões que podem ser perfeitamente adotadas. Vale salientar que o jogo CIN é relativamente complexo para se desenvolver, o que atesta a eficácia das tecnologias adotadas, pelo menos para esse projeto. O desenvolvimento de cada funcionalidade representou novos desafios, pois a solução web gerencia o funcionamento de um ou mais jogos, que possuem muitos estados diferentes e que geralmente estão intimamente ligados.

6.1 Trabalhos Futuros

Ao final desse processo de desenvolvimento foram observadas algumas possibilidades de trabalhos futuros, são eles:

- Utilização da solução em uma disciplina para verificar sua eficácia em um cenário real de uso;
- Incorporar novas funcionalidades de métricas de uso da solução, visando analisar o comportamento dos alunos na disciplina, incluindo-se geração de relatórios com os dados analisados;
- Avaliação da usabilidade da solução, buscando sugestões de melhorias na interação dos usuários; e
- Integração com redes sociais, permitindo que usuários possam efetuar cadastro e publicar resultados através das redes sociais.

REFERÊNCIAS

- ALMENDRA, Camilo Camilo. MAGALHAES, Regis Pires. ALMEIDA, Calos Diego Andrade. **Métodos Ágeis em um Núcleo de Práticas Acadêmico: Relato de Experiência.** In: XXIII Workshop sobre Educação em Computação (WEI), 2015, Recife. XXXV Congresso da Sociedade Brasileira de Computação. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2015/009.pdf>>. Acesso em 25 de jan. 2015
- BEZERRA, E. Modelagem de Atividades. In: BEZERRA, E. (Org.). **Princípios de Análise e Projeto de Sistemas com UML: Um guia prático para modelagem de sistemas.** Rio de Janeiro, RJ: Elsevier, 2007. p. 307-312.
- COHN, Mike. **Desenvolvimento de software com scrum: aplicando métodos ágeis com sucesso.** Porto Alegre, RS: Bookman, 2011. 496 p.
- COHN, Mike. **User Stories Applied: For Agile Software Development.** Crawfordsville, Indiana: Pearson Education, 2009. 268 p.
- JEFFRIES, Ron. Essential XP: Card, conversation, confirmation. **XP Magazine**, v. 30, 2001. Disponível em: <<http://ronjeffries.com/xprog/articles/expcardconversationconfirmation/>>. Acesso em 27 de mai. 2015.
- JUCÁ, Paulyne Matthews. ROLIM, Germana Ferreira. **Gamificação na Disciplina de Empreendedorismo.** In: XXI Workshop sobre Educação em Computação (WEI), 2013, Maceió. XXXIII Congresso da Sociedade Brasileira de Computação. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/0012.pdf>>. Acesso em: 23 de abr. 2015.
- JUCÁ, Paulyne Matthews et al. **Aplicação da Gamificação na Disciplina de Empreendedorismo.** In: XXII Workshop sobre Educação em Computação (WEI), 2014, Brasília. **Anais...** Brasília: XXXIV Congresso da Sociedade Brasileira de Computação, 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2014/008.pdf>>. Acesso em: 23 de abr. 2015.
- KRUCHTEN, Philippe. **Introdução ao RUP. rational unified process** . Rio de Janeiro, RJ: Ciência Moderna, c2003. xv, 255 p.
- OLIVEIRA, Carlos D. C.; CINTRA, Marcos E.; NETO, Francisco Milton Mendes. **Jogo sério para o ensino da Gestão de Riscos em Projetos de Softwares usando Inteligência Artificial.** RENOTE, v. 11, n. 1, 2013. Disponível em: <<http://www.seer.ufrgs.br/renote/article/viewFile/41619/26400> >. Acesso em: 23 de abr. 2015.
- PRESSMAN, Roger S. **Engenharia de software: Uma Abordagem Profissional.** 7. ed. Porto Alegre: AMGH Ed., 2011. 780 p.
- SAVI, Rafael; ULBRICHT, Vania Ribas. Jogos digitais educacionais: benefícios e desafios. **RENOTE**, v. 6, n. 1, 2008.
- SCHWABER, Ken; SUTHERLAND, Jeff. The scrum guide. **Scrum Alliance**, 2013. Disponível em <<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 25 de mai. 2015.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo, SP: Pearson Prentice Hall, 2011. 521 p.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3. ed. Porto Alegre: Bookman, 2007. 695 p.

ROB JOHNSON et al. **Spring Framework Reference Documentation**. Disponível em: <http://docs.spring.io/spring/docs/4.0.x/spring-framework-reference/html/>. Acesso em: 25 de mar. 2015.

WEISSMANN, Henrique Lobo. **Vire o jogo com Spring Framework**. São Paulo, SP: Casa do Código, 2012. 252 p.

WIEGERS, Karl Eugene. Chapter 5: Establishing the Product Vision and Project Scope. In: WIEGERS, Karl Eugene (Org.). **Software Requirements**. Redmond, Washington: Microsoft Press, 2003.

APÊNDICES

APÊNDICE A – Documento de Arquitetura



UNIVERSIDADE FEDERAL DO CEARÁ

Documento de Arquitetura de Software Concurso de Ideias de Negócio - CIN

Histórico da Revisão

Data	Versão	Descrição	Autor
31/05/2015	0.1	Elaboração do documento	Wellington Lucas Moura
29/12/2015	0.2	Adição da visão de implementação	Wellington Lucas Moura
30/12/2015	1.0	Acréscimo de novos diagramas na visão de Processos e Implementação	Wellington Lucas Moura

1. Introdução

1.1. Finalidade

Este documento apresenta a arquitetura proposta para a solução web de apoio ao jogo Concurso de Ideias de Negócio (CIN), utilizado na disciplina de Empreendedorismo na UFC – Campus Quixadá. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

1.2. Escopo

Este documento trata da arquitetura da solução web para o jogo Concurso de Ideias de Negócio, que visa apoiar a execução do jogo na disciplina de Empreendedorismo na UFC campus Quixadá.

1.3. Definições, Acrônimos e Abreviações

Model View Control	MVC
Spring Framework	SF
Núcleo de Práticas em Informática	NPI
Concurso de Ideias de Negócio	CIN
Unified Modeling Language	UML

1.4. Visão Geral

Este documento está organizado em tópicos relacionados às diferentes visões arquiteturais abordadas.

2. Arquitetura da Aplicação

Esse documento apresenta a arquitetura como uma série de visualizações/visões: visão lógica, visão de processos e visão de execução. Essas visualizações são apresentadas utilizando-se UML.

2.1. Representação arquitetural

O sistema está sendo desenvolvido tendo como base a arquitetura adotada no NPI. Onde utiliza-se uma Arquitetura de Camadas juntamente com o Padrão Repository.

2.2. Objetivos e Restrições da Arquitetura

A arquitetura proposta tem como objetivo separar os interesses de cada camada da aplicação, disponibilizar um sistema funcional e com potencial de adaptabilidade a mudanças.

2.3. Critérios da Avaliação Arquitetural

Os critérios utilizados para a seleção da solução arquitetural foram:

- Modularidade;
- Modificabilidade;
- Manutenibilidade;

3. Metas e Restrições da Arquitetura

Softwares Utilizados

Para a execução do sistema será requisitado a instalação do (s) seguinte (s) software (s):

- Mozilla Firefox v. 38 ou superior / Google Chrome v.43 ou superior;

Caso algum software indicado não seja instalado, a aplicação pode não funcionar como esperado, ou não funcionar.

Existem alguns importantes requisitos e restrições do sistema que possuem uma influência significativa na arquitetura: São elas:

- Todas as funções devem estar disponíveis através dos navegadores mais populares;
- O sistema deve garantir a confidencialidade dos dados cadastrados;
- O sistema deve possuir diferentes níveis hierárquicos de acesso.

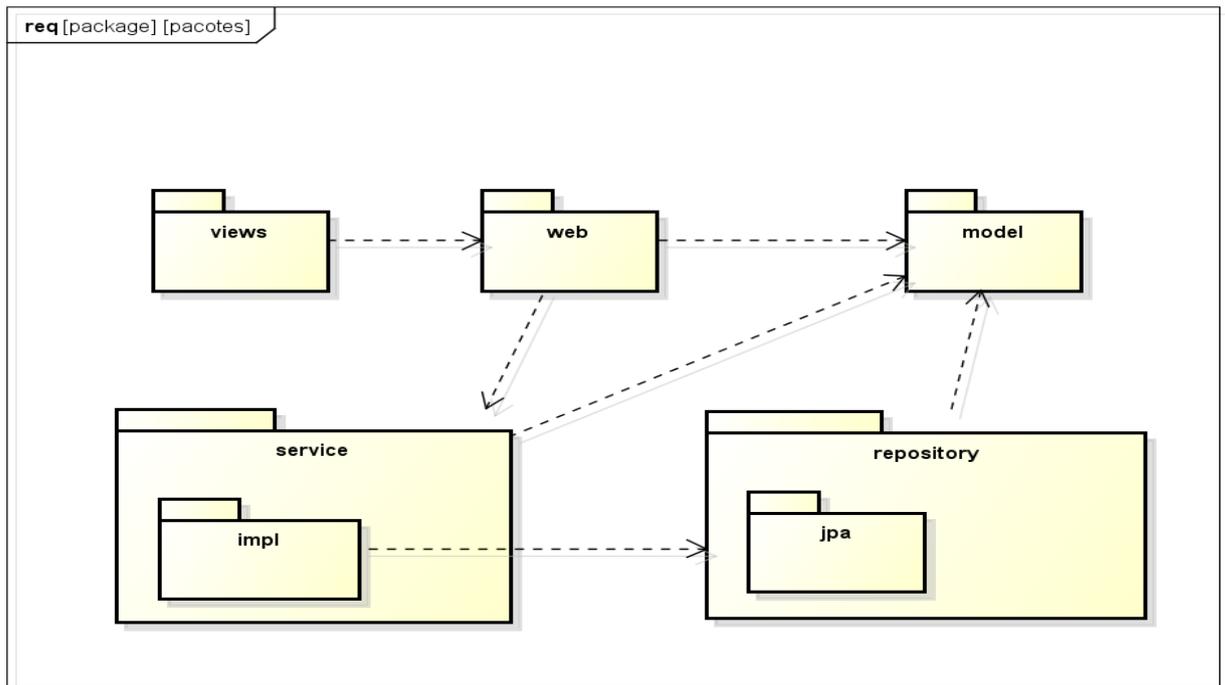
4. Visualização Lógica

Esta seção apresenta diferentes tipos de visões lógicas, com níveis de abstrações distintas, para que se possa obter uma visibilidade de comunicação entre pacotes e classes.

4.1. Visão de pacotes

A visão lógica da solução CIN é composta por 7 diretórios principais:

Figura 24 – Visão Lógica de Pacotes



Fonte: Elaborado pelo autor

- **views**

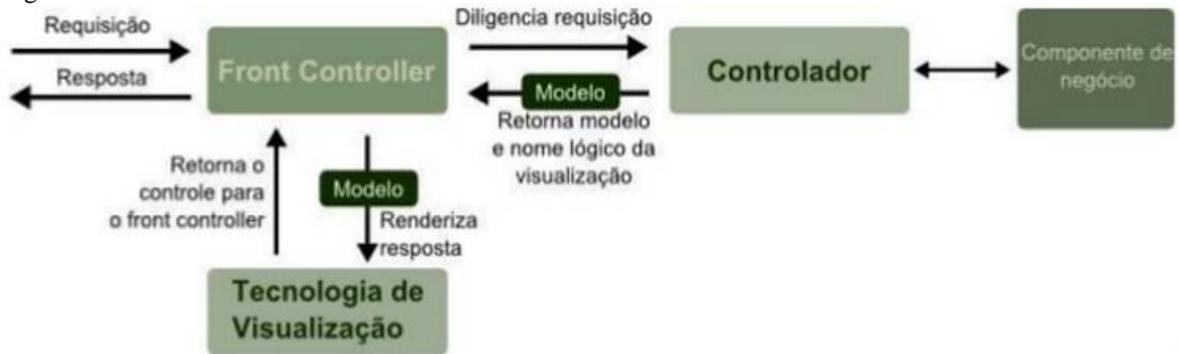
Neste diretório estão todos os subdiretórios e arquivos “.jsp” necessários para gerar a visualização do sistema. Aqui serão inseridos qualquer novo arquivo “.jsp” necessário para a visualização de informações.

- **web**

Neste diretório são implementados os controladores, responsáveis por fazer um “link” entre as regras de negócio da aplicação com a visão. SF usa o padrão Front Controller. O Front Controller intercepta todas as requisições para direcionar ao controlador adequado. A figura 2 apresenta o padrão Front Controller.

É importante que qualquer novo controlador seja inserido neste diretório para manter o padrão adotado de separação de interesses. Essa é uma decisão de projeto adotada para facilitar futuras manutenções de funcionalidades.

Figura 25 – Padrão Front Controller



Fonte: Adaptado de (WEISSMANN, 2013)

- **model**

Contém as classes de negócio da aplicação. Ex.: Usuario, Jogo, Rodada, etc. Qualquer nova entidade mapeada para o projeto deve ser inserida neste pacote.

- **repository**

Contém as interfaces de persistência das entidades do sistema. Ex.: UsuarioRepository. Novas interfaces de repositório devem ser inseridos nesse diretório.

- **repository.jpaa**

Contém as classes que implementam as interfaces de repository. Ex.: JpaUsuarioRepository. Ao ser inserida alguma interface no pacote repository deve-se incluir aqui uma implementação para ela.

- **service**

Contém as interfaces responsáveis pela lógica relacionada às entidades do sistema. Ex.: UsuarioService. Novas interfaces de serviço devem ser incluídas nesse pacote, à medida que se faça necessário.

- **service.impl**

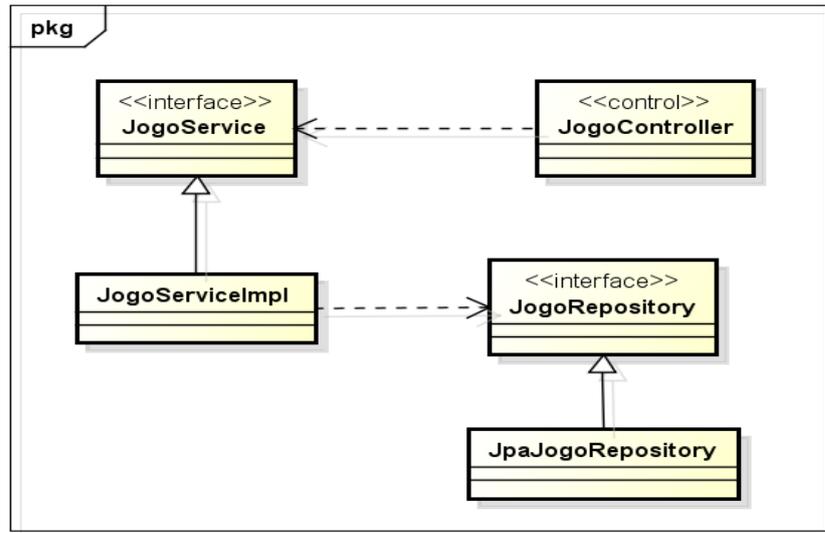
Contém as classes que implementam as interfaces de service. Ex.: UsuarioServiceImpl.

4.2. Visão de Classes

Apresenta como as classes se comunicam entre si na aplicação.

4.2.1. Interação entre classes dos pacotes repository, service e web

Figura 26 – Padrão de comunicação entre classes e interfaces



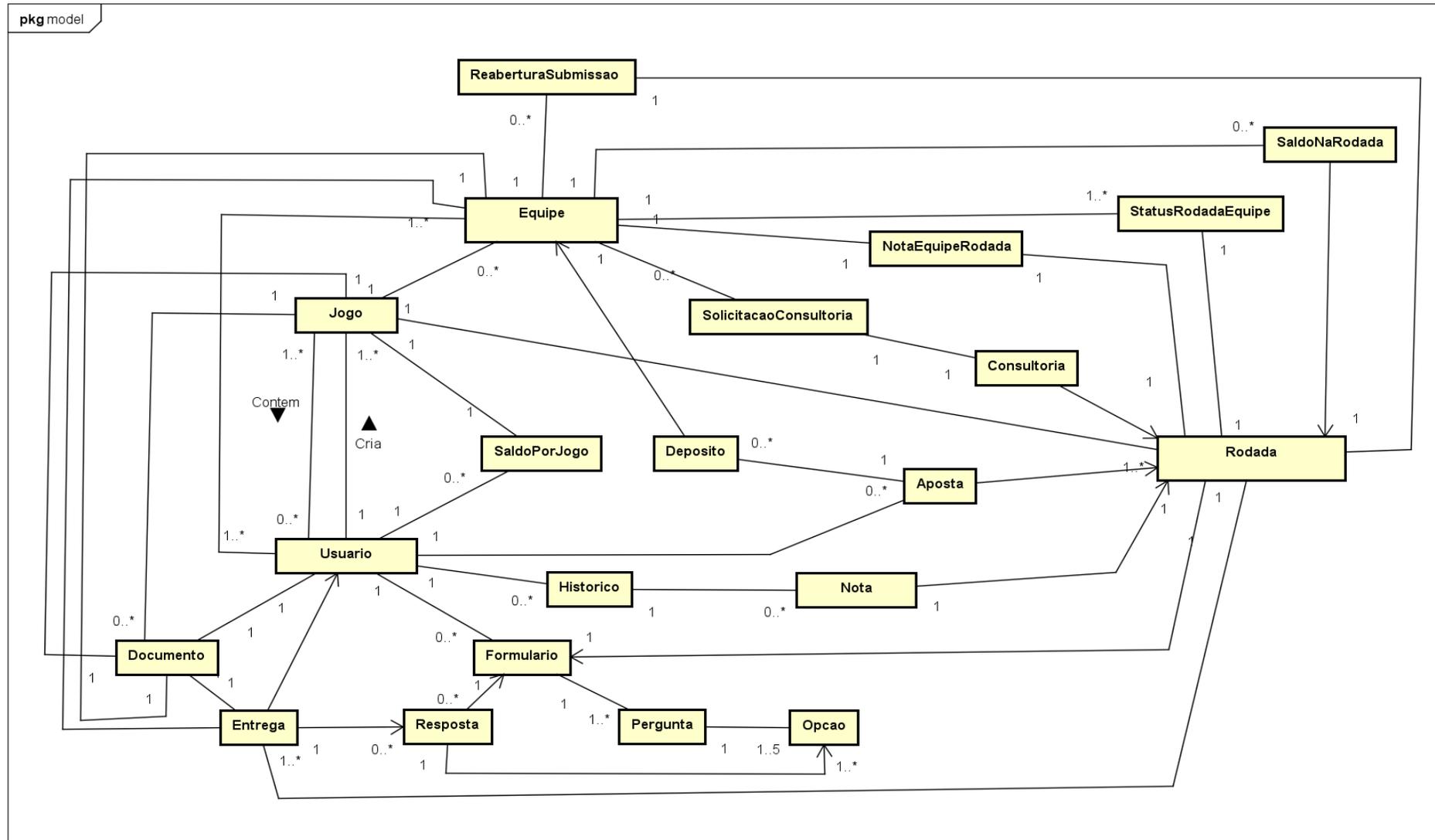
Fonte: Elaborado pelo autor

As classes se relacionam dessa forma para aumentar a abstração dos dados, sendo necessário que uma classe, que precise de um serviço, conheça apenas as interfaces que fornecem o serviço.

4.2.2. Divisão de classes

Mostra a visão geral de como as classes do sistema se comunicam. A Figura 27 apresenta a visão de classes de modelo do sistema, apresentando as respectivas cardinalidades.

Figura 27 – Visão das classes de modelo

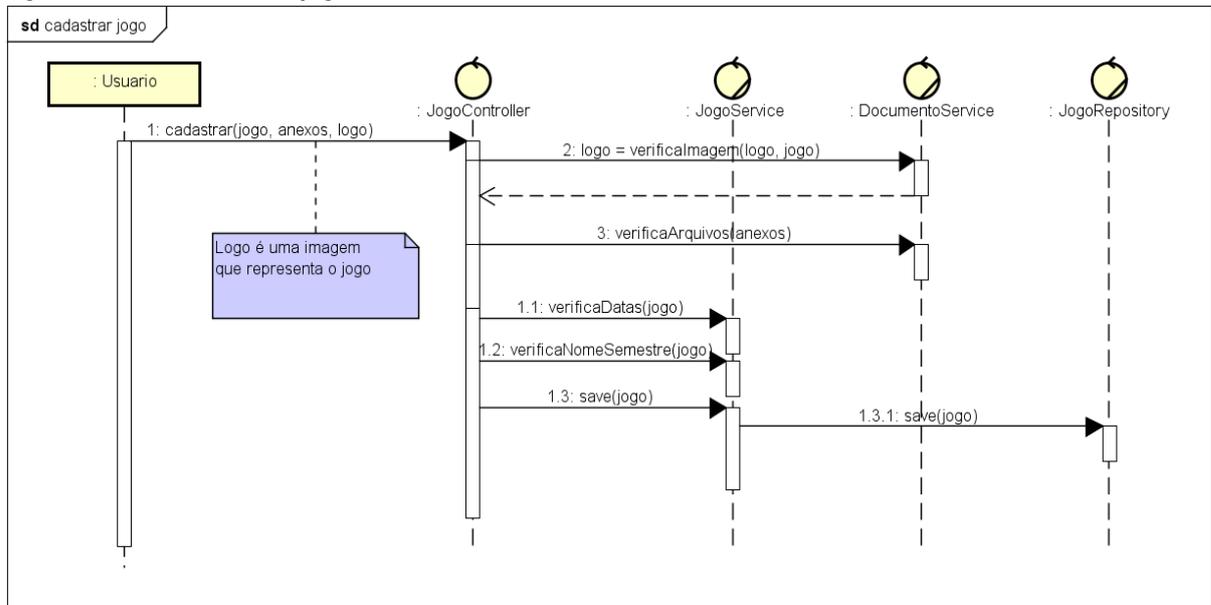


Fonte: Elaborado pelo autor

5. Visão de Processos

Esta seção descreve a decomposição do sistema em processos, sendo evidenciado até então, o processo de cadastrar um novo jogo no sistema. No qual um professor logado no sistema faz uma requisição para se criar um novo jogo, o fluxo é apresentado na Figura 17.

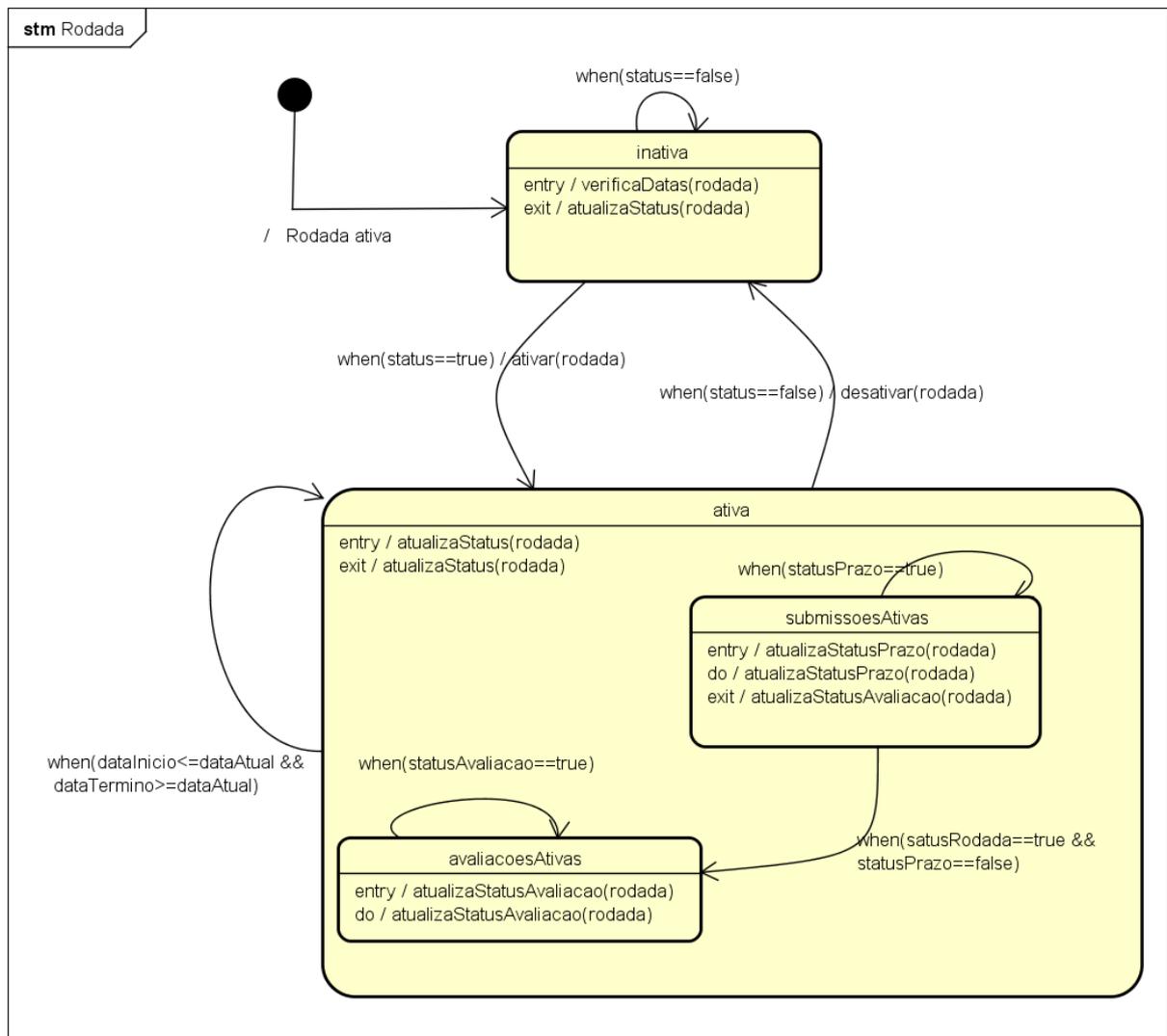
Figura 28 – Cadastrar novo jogo



Fonte: Elaborado pelo autor

A seguir é apresentado um diagrama de máquina de estados de uma rodada, considerando-se que é a entidade que possui mais estados no jogo e a maior parte da lógica do sistema ocorre neste modelo.

Figura 29 – Diagrama de estado de máquina de uma rodada



Fonte: Elaborado pelo autor

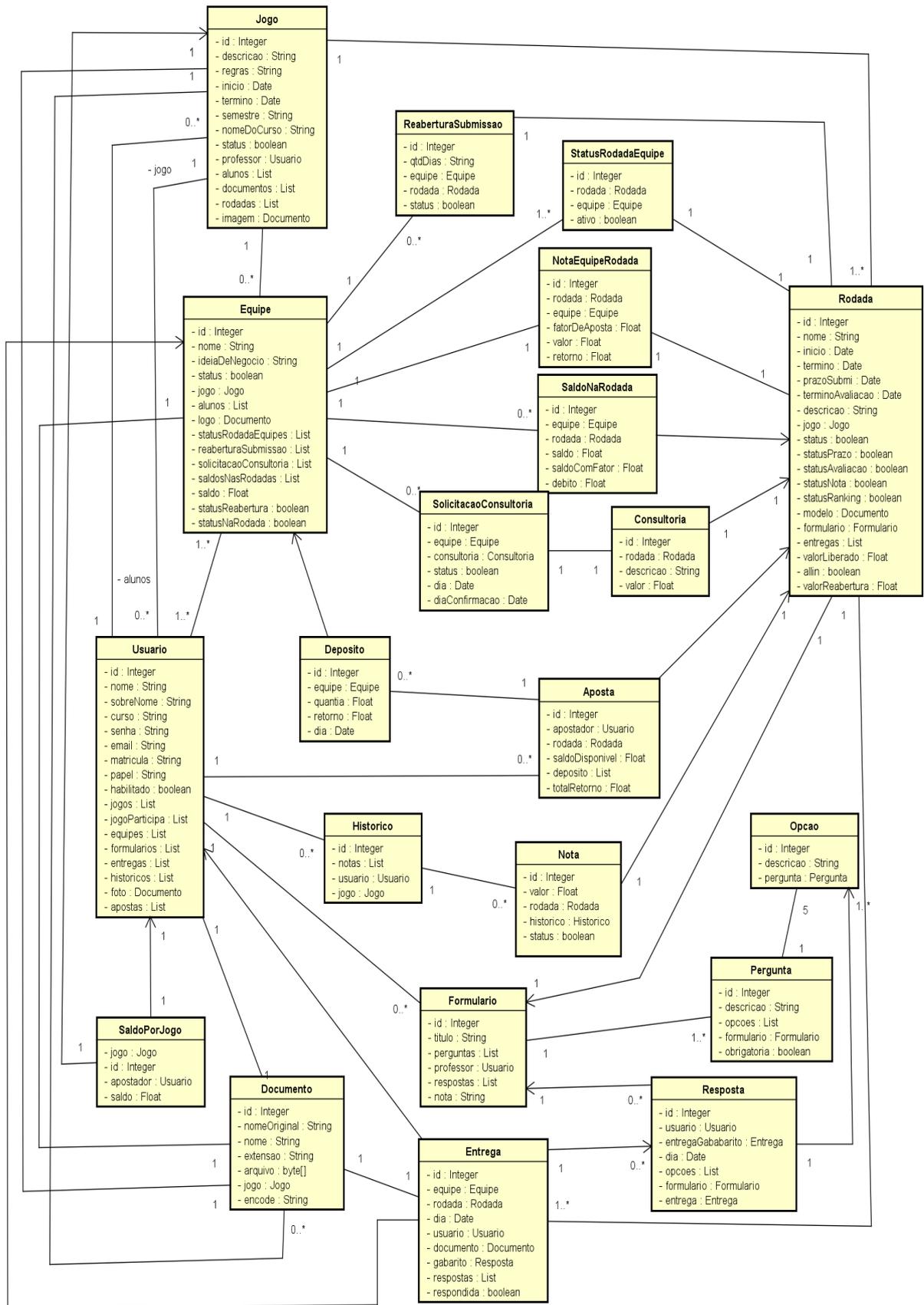
6. Visão de Implementação

Esta seção descreve a estrutura geral do modelo de implementação, a divisão do software em subsistemas no modelo de implementação e todos os componentes significativos do ponto de vista da arquitetura.

6.1. Visão geral das classes de modelo

Na Figura 30 estão representadas todas as entidades de modelo do projeto, com seus respectivos atributos e dependências.

Figura 30 – Visão de classes de modelo

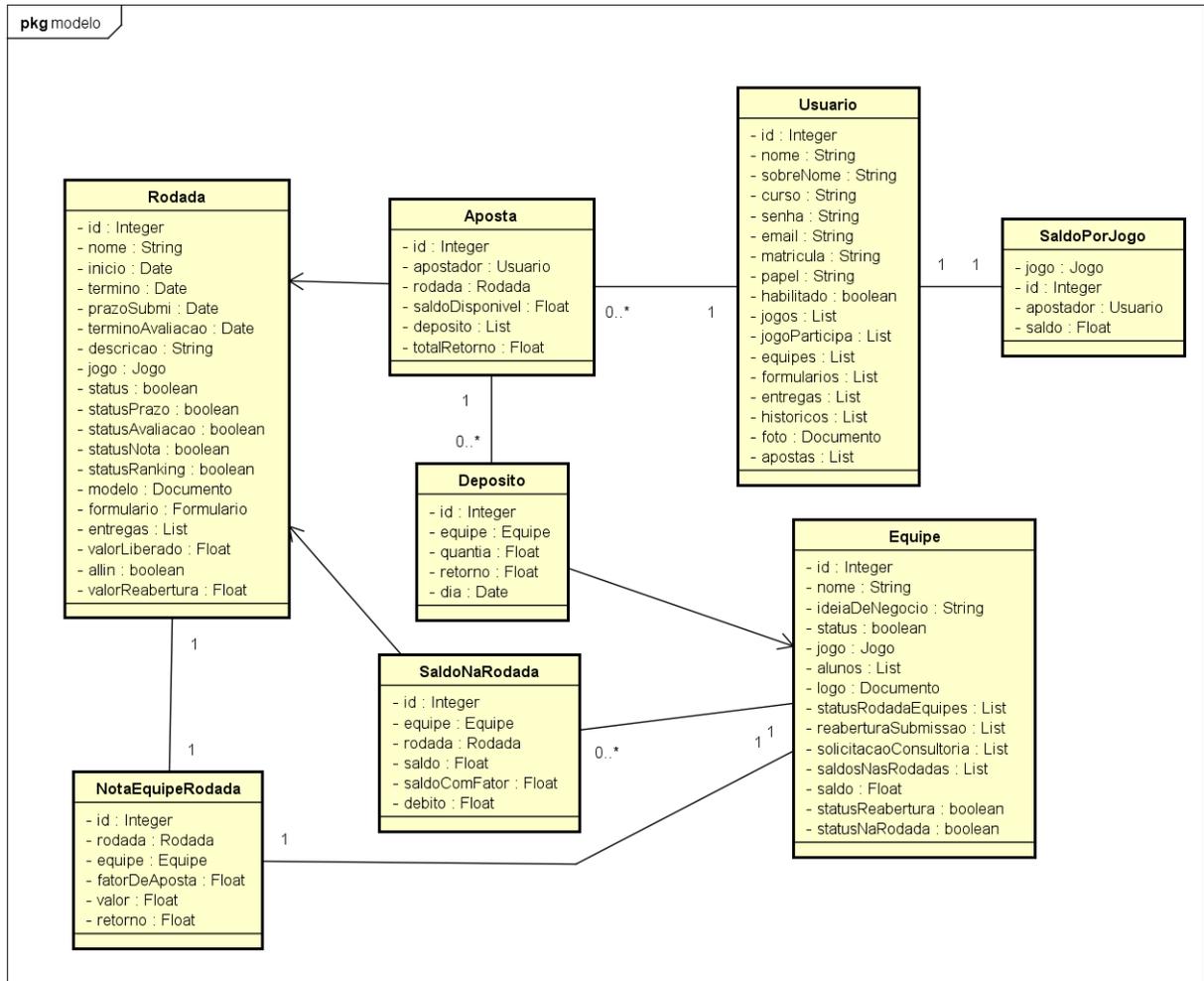


Fonte: Elaborado pelo autor

6.2. Visão de apostas numa rodada

Na Figura 31 são apresentadas as entidades de modelo associadas ao processo de apostas/investimentos numa rodada.

Figura 31 – Visão de apostas em uma rodada

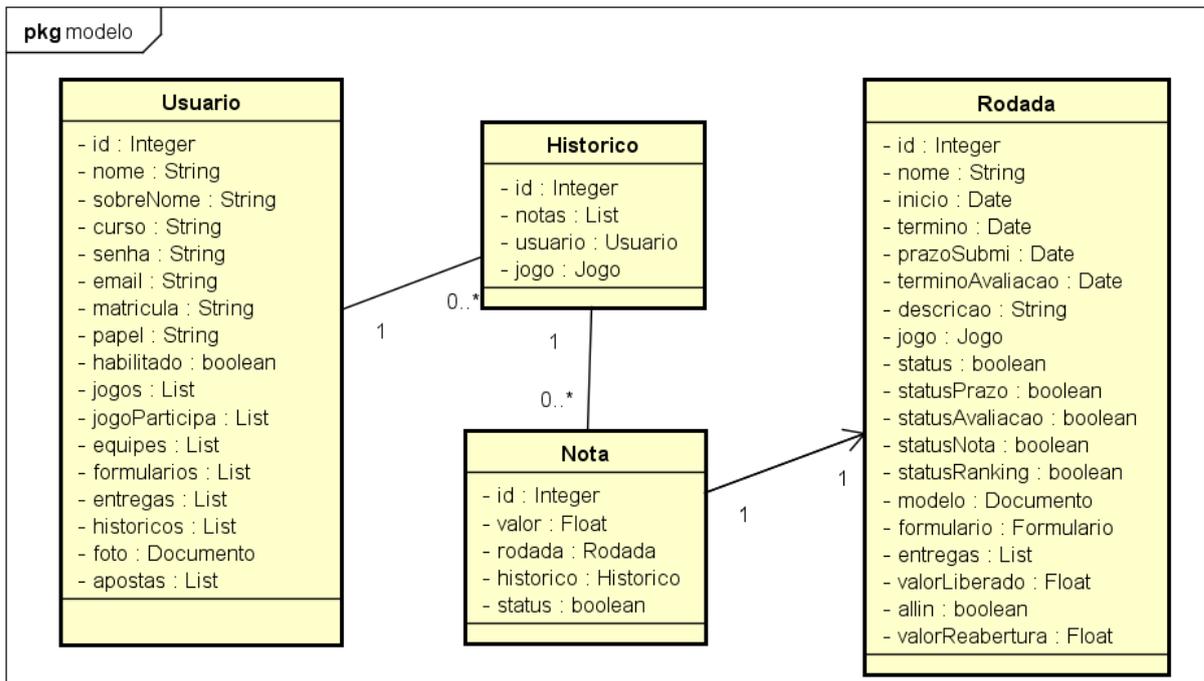


Fonte: Elaborado pelo autor

6.3. Visão do histórico de notas de Usuario

Na Figura 32 são apresentadas as entidades de modelo associadas ao processo de notas de um usuário, durante uma rodada.

Figura 32 – Histórico - Usuário Rodada

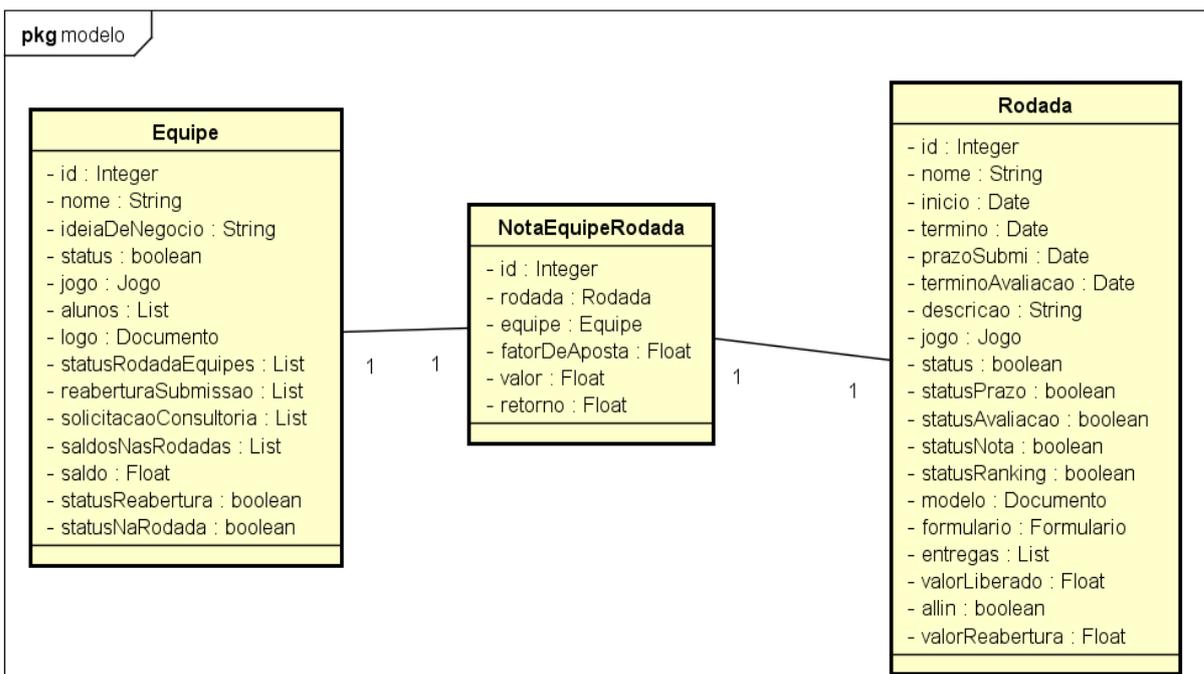


Fonte: Elaborado pelo autor

6.4. Visão das notas de equipe

Na Figura 33 são apresentadas as entidades de modelo associadas ao processo de notas de uma equipe, para uma rodada.

Figura 33 – Nota equipe rodada

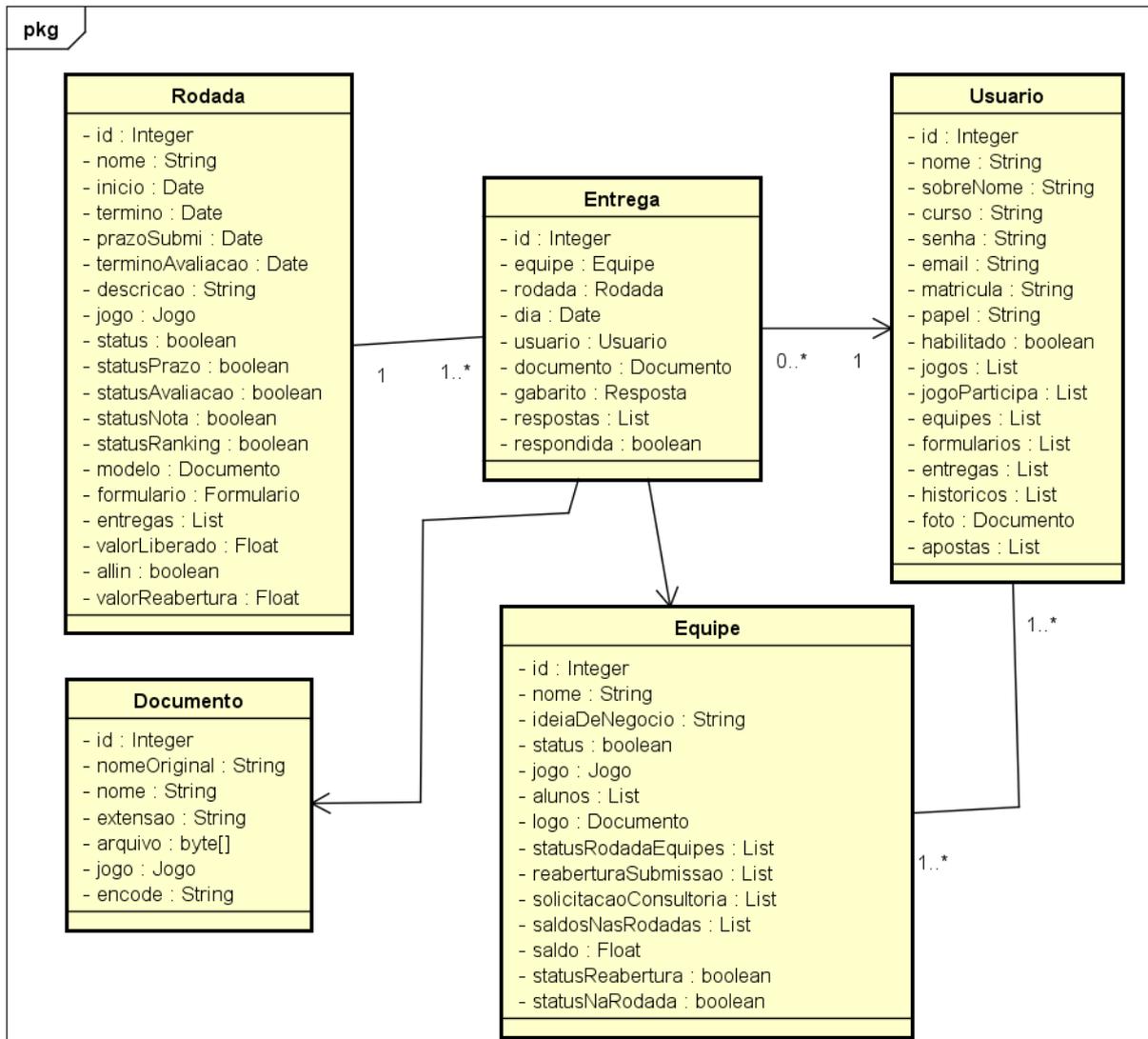


Fonte: Elaborado pelo autor

6.5. Visão de entrega para uma rodada

Na Figura 34 são apresentadas as entidades envolvidas na entrega de uma equipe para uma rodada.

Figura 34 – Visão de entregas para uma rodada

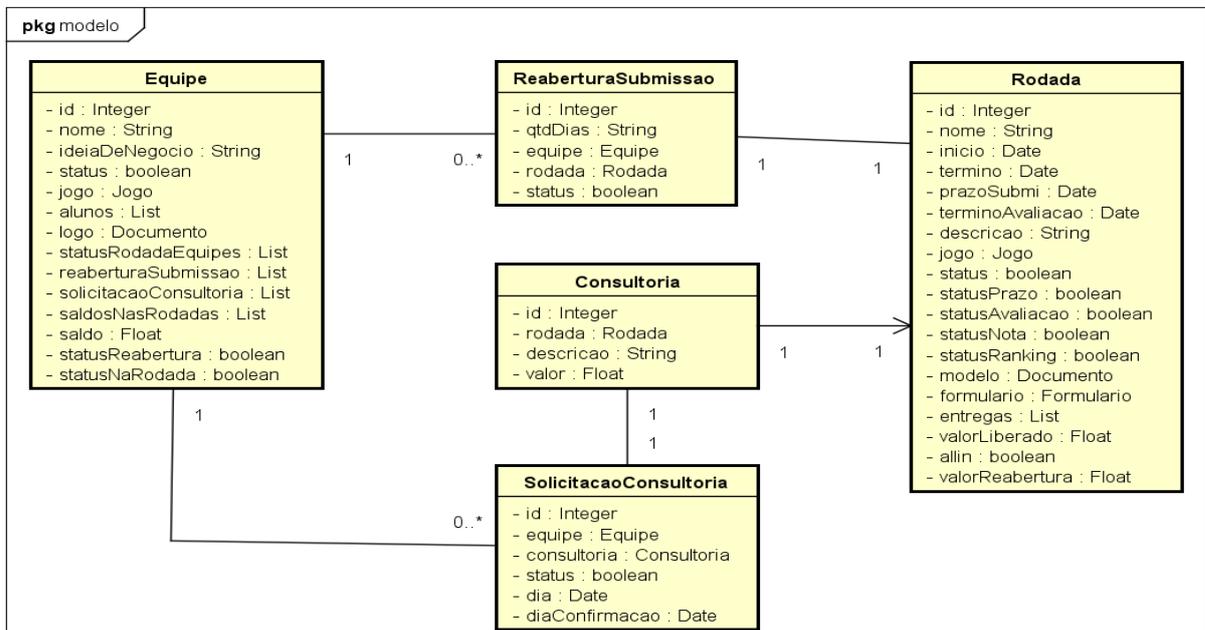


Fonte: Elaborado pelo autor

6.6. Visão de serviços oferecidos pelo professor

Na Figura 35 são apresentadas as entidades envolvidas na oferta de serviços para uma rodada.

Figura 35 – Serviços de uma rodada



Fonte: Elaborado pelo autor

7. Qualidade

- Modularidade:
 - Os componentes estão separados por seus estereótipos, cada classe pertencente ao sistema tem um pacote específico que agrupa classes que possuem o mesmo estereótipo.
- Modificabilidade:
 - Por possuir componentes com distribuição modular, separados por interesses, onde a camada de apresentação ou visão, negócio e gerenciamento de dados são encapsuladas, aumenta-se a modificabilidade. Onde pode-se alterar a lógica interna de algum componente sem que, em muitos casos, tenha-se impacto em outros níveis.
- Manutenibilidade:
 - Como há separação de interesses no momento de se criar um componente ou classe, seguindo padrões de nomenclatura, buscando manter a coesão entre pacotes, facilita-se assim a manutenibilidade da arquitetura.

REFERÊNCIAS

WEISSMANN, Henrique Lobo. **Vire o jogo com Spring Framework**. São Paulo, SP: Casa do Código, 2012. p. 133.

ANEXOS

ANEXO A – Documento de Visão



UNIVERSIDADE FEDERAL DO CEARÁ

DOCUMENTO DE VISÃO

Jogo Concurso de Ideias de Negócio

HISTÓRICO

Data	Versão	Responsável	Alteração
20/02/2015	0.1	Camilo Almendra	Versão inicial do documento
24/03/2015	0.3	Wellington Lucas	Formatações e correções do documento
09/06/2015	0.4	Wellington Lucas	Mapeamento das histórias de usuário
12/01/2016	1.0	Wellington Lucas	Atualização do <i>backlog</i>

1 INTRODUÇÃO

A finalidade deste documento é fazer a coleta e análise das necessidades e recursos de alto nível do Jogo Concurso de Ideias de Negócio. O foco do documento é identificar as necessidades dos envolvidos e do público-alvo. Os detalhes de como o CIN irá satisfazer essas necessidades serão descritos no documento de casos de uso e nas especificações suplementares.

2 POSICIONAMENTO

2.1 DESCRIÇÃO DO PROBLEMA

O problema de	Apoiar a execução de um jogo sério envolvendo dezenas de participantes e equipes em uma disciplina de graduação
Afeta	Docentes e Alunos participantes
Cujo impacto é	O grande esforço de executar as etapas do jogo de forma manual, e a limitação de acesso a informações pelos envolvidos
Uma boa solução seria	Um sistema que automatizasse a configuração, acesso a informações, controle de rodadas e fases e gerenciamento do jogo

2.1 SENTENÇA DE POSIÇÃO DO PRODUTO

Para	Docente da disciplina
Que	Necessita gastar menos tempo e dar mais visibilidade aos alunos das fases do jogo Confiança dos resultados
O (nome do produto)	Concurso de Ideias de Negócio
Que	Configura, Divulga e Gerencia as Fases do Jogo
Ao contrário da	Realização de um processo quase manual baseado em planilhas, pastas, compartilhamentos em fóruns ou serviços de hospedagem de arquivos, e troca de e-mails não estruturada
Nosso produto	Possibilitará uma diminuição do esforço em horas para configuração, divulgação e gerência do jogo.

3 PERFIL E DESCRIÇÃO DOS STAKEHOLDERS

3.1 STAKEHOLDERS DO PROJETO

1. Docente que aplica o jogo em disciplina;
2. Alunos envolvidos no jogo;

3.2 PERFIL DOS STAKEHOLDERS DO PROJETO

- DOCENTE:

Um docente da disciplina de Empreendedorismo que deseje aplicar o Jogo de Empreendedorismo necessita de apoio ferramental para lidar com as diversas informações e arquivos trocados entre o docente e os alunos e equipes. Além disso, o docente atua como configurador e gerenciador da execução do jogo, intervendo na definição de parâmetros em tempo de execução – que afetam os resultados do jogo e precisam ter seus efeitos divulgados de forma rápida e clara aos participantes.

Outro aspecto importante é que a participação no Jogo pelos alunos é usada como forma de avaliação da disciplina, em conjunto com outras formas de avaliação (prova, etc.). Para o docente é importante poder acompanhar o desempenho dos alunos de forma individual ao longo do jogo.

Pela natureza do jogo, cada rodada envolve inúmeras informações (arquivos digitais, parâmetros de configuração, apostas individuais, notas de avaliação, etc.) inseridas por docente, equipes e alunos individualmente. Realizar a tabulação e organização dessas informações em planilhas e troca de e-mails é muito suscetível a erros e demanda muito esforço em horas de trabalho do docente.

- ALUNO DA DISCIPLINA:

Um aluno da disciplina se envolve com o jogo de duas formas: como membro de uma equipe e como investidor individual. Como membro de equipe um aluno colabora para construir os artefatos das rodadas em equipe, e recebe notas em grupo que compõe sua avaliação final. Como investidor individual, um aluno faz avaliação individual de artefatos de outras equipes e da sua própria e realiza apostas de valores em equipes que acreditar serem as melhores naquela rodada.

Um aluno precisa ser capaz de acompanhar o racional do cálculo de suas notas, incluindo a nota em equipe e a nota da avaliação de artefatos. As apostas geram apenas retorno financeiro e não afetam a nota de avaliação na disciplina. O jogo prevê uma

premiação de caráter lúdico para o melhor aluno apostador do jogo inteiro. Além da melhor equipe.

4 DESCRIÇÃO DO JOGO

O Concurso de Ideias de Negócios (CIN) é um jogo em equipe onde o objetivo principal é construir artefatos (e.g. Plano de Negócio) de excelência, que sejam reconhecidos como tal por docentes e os demais alunos participantes. Individualmente, os alunos além de buscarem um bom desempenho para sua equipe também atuam como avaliadores e investidores. Como avaliadores os alunos devem realizar a avaliação de todos os artefatos gerados pelas demais equipes e a sua, seguindo um instrumento objetivo de avaliação comum a todos. Seu desempenho como avaliador resulta em uma nota, já seu desempenho em equipe compõe outra nota, as notas são independentes. Além disso, como investidor os alunos podem apostar em equipes que acreditem terem entregado artefatos de excelência, essa etapa não compõe nenhuma das notas. Ao final do jogo os alunos concorrem ao prêmio de melhor investidor, que possui caráter lúdico, sem implicações na nota final. As equipes também são classificadas de acordo com os investimentos.

Fases

O jogo possui basicamente três fases: Configuração, Rodadas e Rodada Final. Na configuração são organizadas as equipes e seus membros e escolha de temas de negócio. Essa rodada final é semelhante as demais, exceto pelas apostas que não são limitadas, podendo o apostador, usar todo o valor que conseguiu até então

Rodada

Cada rodada possui as seguintes etapas:

- Apresentação do Modelo da etapa
- Atendimento docente para as equipes:

Atendimento presencial ou remoto do docente para as equipes, em horários previamente divulgados.

Equipes podem realizar a compra de atendimento extra, usando seu saldo de dinheiro da equipe.

- Preparação e submissão dos artefatos pelas equipes;

A equipe pode submeter várias versões do artefato até a data/hora limite. Todos os participantes podem acompanhar quais artefatos e quem da equipe os submeteu. O docente também deve ter acesso ao arquivo, para lê-lo e realizar as correções.

- Distribuição dos artefatos das equipes pelo docente;
Após o término das submissões, os artefatos finais de todas as equipes ficam disponíveis para todos os participantes, sem qualquer alteração do docente. Ao longo das rodadas, os artefatos enviados em rodadas anteriores ficam também disponíveis para análise dos participantes.
- Avaliação individual dos artefatos pelos participantes;
Cada participante deve avaliar as submissões das outras equipes e da sua própria, seguindo um instrumento de avaliação padronizado. A proximidade de sua avaliação comparada à avaliação do docente estabelecerá sua Nota de Avaliação Individual.
- Aposta individual dos investidores (participantes) nas equipes;
É facultado aos participantes realizar apostas em equipes, acreditando que os artefatos gerados são os melhores da rodada. A aposta é realizada a partir do valor definido para cada participante, independente do saldo. Os melhores investidores serão os que acumularem mais saldo ao final do jogo.
- Avaliação dos artefatos pelo docente e divulgação para cada equipe;
O docente realiza a avaliação das entregas das equipes usando o instrumento padrão de avaliação, mesmo instrumento que os alunos utilizam. O resultado da avaliação de uma equipe é disponibilizado para os membros da equipe apenas.
- Configuração do Fator de Aposta para cada equipe pelo docente;
A partir do desempenho geral e específico das equipes, o docente configura o Fator de Aposta para cada equipe. Esse fator influencia o valor em dinheiro que a equipe acumulará a partir da sua pontuação no instrumento e das apostas que foram feitas pelos investidores. O fator também influencia a taxa de retorno das apostas para os investidores.
- Cálculo dos rankings e Distribuição/resgate de apostas
São divulgados ao final de cada rodada quatro rankings:
 - Melhor Equipe da Rodada
 1. Ordem decrescente;
 2. Equipe com mais dinheiro conseguido na rodada;
 3. O cálculo é o somatório de todo o valor apostado naquela equipe, multiplicado pelo fator de aposta.
 - Exemplo: fator de aposta 2,0, e 5.000,00 em apostas: $5.000 \times 2,0 = 10.000$
 - Melhor Equipe Geral

1. Ordem decrescente
2. Equipe com mais dinheiro acumulado em todas as rodadas
3. O cálculo é soma dos valores da rodada atual e das anteriores
 - Melhor Avaliador da Rodada
 1. Apenas os cinco primeiros
 2. Participante que mais se aproximou da avaliação do docente
 3. Cálculo: lista por notas em valor decrescente todos os alunos, os cinco melhores desempenhos são listados. No caso de empate, mais de cinco alunos podem ser listados
 - Melhor Investidor Geral
 1. Ordem decrescente;
 2. Participante com maior saldo em dinheiro, acumulado de todas as rodadas;
 3. Cálculo é feito pelo valor que ele apostou numa equipe multiplicado pelo fator de aposta;
 4. Exemplo: aposta na equipe A de 500,00, fator 2,0, retorno de 1.000,00. Aposta na equipe B de 500,00, fator -1,0, retorno -500,00. Valor final retornado $1.000 - 500 = 500$.

Ao final da rodada são também distribuídos ou resgatados valores referentes às apostas. Cada equipe poderá verificar seu saldo, seu ranking e sua nota de grupo, assim como cada participante poderá verificar seu saldo, seu ranking e sua nota individual.

4.1 Ambiente de execução do jogo

O jogo é executado em conjunto com uma disciplina semestral de Empreendedorismo, servindo como parte da avaliação da disciplina. O jogo possui objetivos de equipe e individuais. Tipicamente, existem de 6 a 9 equipes com até 5 membros, com um total em torno de 30 a 40 participantes. O período de execução do jogo dura cerca de 2 meses, incluindo todas as rodadas. O jogo é configurado e controlado pelo docente, normalmente apenas uma pessoa.

O tema do jogo é “Ideia de Negócio”, e as equipes concorrem para produzir a melhor ideia de negócios possível. Ao longo das rodadas é exigida a entrega de diferentes partes do plano de negócios, em sintonia com o andamento das aulas didáticas. As entregas são documentos em formato PDF ou Microsoft/Open Office.

As equipes cooperam entre si através de encontros presenciais e/ou quaisquer outros meios de comunicação (e-mails, listas, mídias sociais, etc.) que acharem conveniente. A comunicação intra-equipes não é monitorado ou suportada pelo ambiente de apoio ao jogo.

O acesso ao ambiente do jogo deve ser disponibilizado através da Web para livre acesso em qualquer horário ou localidade.

5 CENÁRIOS

Configuração

- CON1: Como participante, eu posso me cadastrar no sistema;
- CON2: Como docente, eu posso cadastrar/atualizar as informações gerais do jogo;
- CON3: Como docente, eu posso organizar equipes de participantes para um jogo;
- CON5: Como docente, eu posso vincular participantes a um jogo;
- CON8: Como docente, eu posso cadastrar um novo jogo;
- CON9: Como docente, eu posso adicionar uma imagem de apresentação para o jogo;
- CON10: Como participante, eu posso alterar meus dados de cadastro;
- CON11: Como participante, eu posso alterar minha imagem de perfil; e
- CON12: Como membro de equipe, eu posso adicionar uma imagem que defina minha empresa.

Equipe

- EQU1: Como membro de equipe, eu posso visualizar informações dos demais membros;
- EQU2: Como membro de equipe, eu posso visualizar informações de contato do docente;
- EQU3: Como membro de equipe, eu posso editar as informações da minha equipe;
- EQU4: Como membro de equipe, eu posso editar a ideia de negócio;
- EQU5: Como membro de equipe, eu posso submeter artefato para entrega da rodada;
- EQU6: Como membro de equipe, eu posso visualizar o histórico de submissão para uma entrega de rodada;
- EQU7: Como membro de equipe, eu posso visualizar o saldo da equipe;
- EQU8: Como membro de equipe, eu posso visualizar o resultado da equipe numa rodada;
- EQU9: Como membro de equipe, eu posso adquirir um serviço;

Avaliação Individual do Aluno

- AVA1: Como aluno, eu posso visualizar as submissões finais da rodada de todas equipes;
- AVA2: Como aluno, eu posso submeter uma avaliação individual de uma entrega de equipe;
- AVA3: Como aluno, eu posso visualizar minhas avaliações individuais anteriores;
- AVA4: Como docente, eu posso visualizar as avaliações individuais dos alunos;
- AVA5: Como aluno, eu posso visualizar minha nota individual;
- AVA6: Como docente, eu posso configurar um formulário de múltipla escolha e gabarito;

Aposta

- APO1: Como investidor, eu posso apostar dinheiro, do valor liberado para a rodada, em uma ou mais equipes na rodada;
- APO2: Como investidor, eu posso visualizar minhas apostas anteriores;
- APO3: Como docente, eu posso visualizar todas as apostas dos alunos;
- APO4: Como docente, eu posso delimitar valor a ser liberado para uma rodada;

Avaliação do Docente

- AOD1: Como docente, eu posso salvar uma avaliação para uma entrega de equipe;
- AOD2: Como docente, eu posso configurar o fator de aposta para uma equipe;

Controle

- CTR1: Como docente, eu posso publicar as avaliações e fatores para as entregas de uma rodada;
- CTR2: Como docente, eu posso gerar e publicar os rankings;
- CTR3: Como docente, eu posso encerrar o jogo;
- CTR4: Como docente, eu posso iniciar o jogo;
- CTR5: Como docente, eu posso encerrar período de apostas;
- CTR6: Como docente, eu posso encerrar período de submissão de artefatos;
- CTR7: Como docente, eu posso iniciar uma rodada;
- CTR9: Como participante, eu posso me logar no jogo;
- CTR10: Como docente, eu posso reabrir o período de submissão de artefatos para equipe específica;

ANEXO B – Roteiro de Entrevista

REVISÃO DO BACKLOG

ROTEIRO

- Apresente à cliente o conceito de História de Usuário, explicando as 3 partes: cartão (história), comentários e critérios de aceitação. Ressalte que a história é uma representação das necessidades do sistema.
- Apresente à cliente o conceito de Temas, como forma de agrupar as histórias.
- Apresente à cliente a ferramenta easyBacklog, e mostre onde as estórias e suas partes, e os temas ficam armazenados.
- Depois, proceda com a revisão. Siga tema a tema, revisando as histórias de cada grupo antes de partir para outro. O que focar na revisão:
 - Os termos/nomes usados estão corretos?
 - As histórias estão corretas? Faltou alguma história?
 - Os detalhes (comentários) estão corretos? Faltou algum detalhe?
 - Os critérios de aceitação estão corretos? Faltou algum critério?
 - Existe alguma parte do Jogo que não foi contemplada?
- Por fim, solicite à cliente que indique algumas histórias que gostaria de ver serem implementadas inicialmente (no máximo 5).

OBSERVAÇÕES

1. Oportunamente, a discussão sobre uma história pode ser ilustrada com um protótipo de tela (não precisa ser usando ferramenta, pode ser a mão).
2. Não há problema em deixar pontos com dúvidas ou em aberto (para posterior discussão), desde que isso fique indicado nos comentários.
3. A medida em que forem sendo achados problemas, deve-se editar diretamente o *Backlog*.