



Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

**ProcSoftDigital - Um protótipo computacional para dinamizar o ensino de
processo de software**

Brendo Peres Bizetto

**CASCADEL
2019**

Brendo Peres Bizetto

ProcSoftDigital - Um protótipo computacional para dinamizar o ensino de processo de software

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. Ivonei Freitas da Silva

CASCADEL
2019

Brendo Peres Bizetto

ProcSoftDigital - Um protótipo computacional para dinamizar o ensino de processo de software

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Ivonei Freitas da Silva (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof^a. Claudia Brandelero Rizzi
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Victor Francisco Araya Santander
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 20 de dezembro de 2019

AGRADECIMENTOS

Agradeço primeiramente aos meus familiares, mas principalmente à minha mãe pelo apoio, por acreditar em mim, por estar ao meu lado nos momentos felizes e principalmente nos difíceis, por financiar esses anos de graduação para que eu tivesse uma educação de qualidade.

Ao meu padrinho e pai, Marcus, minha avó Maria e minha tia Téia, que dentre todos que me apoiaram, sempre estiveram presentes e acreditaram em mim, me ajudaram a crescer e amadurecer como pessoa e melhorar como ser humano.

Também a todos meus professores, pelo conhecimento adquirido, mas especialmente ao meu orientador André Luiz Brun, pela sua competência, por estar sempre presente nos momentos de dúvida e por nunca medir esforços para me auxiliar. Aos membros da minha banca Cláudia Brandelero Rizzi e Victor Francisco Araya Santander pelo tempo gasto e pelos conhecimentos transmitidos.

A todos meus amigos, principalmente aos que fiz durante o tempo de graduação, os quais levarei para o resto da vida. Principalmente aos amigos que sempre me auxiliaram e estiveram ao meu lado na graduação, nos trabalhos, nas dúvidas, nos rolês, sou muito grato a todos vocês.

Lista de Figuras

2.1	Camadas da Engenharia de Software (MOURA, 2014)	5
2.2	Ambiente principal do SE RPG 2.0 (AMBROSIO, 2008)	16
2.3	Tabuleiro SimuleS (FIGUEIREDO et al., 2006)	17
2.4	Tabuleiro do ProcSoft (MOURA, 2014)	18
3.1	Grupos de atividades. Adaptado de (MOURA, 2014)	21
3.2	Grupos de atividades. Adaptado de (MOURA, 2014)	22
3.3	Carta de Atividade (MOURA, 2014)	23
3.4	Carta de Atividade (MOURA, 2014)	24
3.5	Carta de Ferramenta (MOURA, 2014)	25
3.6	Carta de Revés (MOURA, 2014)	26
3.7	Carta de Pergunta (MOURA, 2014)	27
3.8	Casos de Uso	28
3.9	Tabuleiro do ProcSoftDigital	30
3.10	Dado do ProcSoftDigital	31
3.11	Adicionar Time do ProcSoftDigital	31
3.12	Adicionar nome do time e participantes do ProcSoftDigital	31
3.13	Card do Time no ProcSoftDigital	32
3.14	Caixa de seleção de Papeis no ProcSoftDigital	32
3.15	Atividade no ProcSoftDigital	33
3.16	Botão perguntas no ProcSoftDigital	33
3.17	Exemplo de Pergunta no ProcSoftDigital	33
3.18	Exemplo de Resposta no ProcSoftDigital	34
3.19	Gerenciar Dinheiro no ProcSoftDigital	34

3.20	Adicionar atividade no ProcSoftDigital	34
3.21	Card com atividade adicionada no ProcSoftDigital	35
3.22	Ferramentas no ProcSoftDigital	35
3.23	Carta de sorte no ProcSoftDigital	36
3.24	ProcSoftDigital	36
3.25	Diagrama de fluxo	37
3.26	Diagrama de Pacotes	38
4.1	Gráfico de faixa etária	40
4.2	Gráfico relativo à quem joga jogos digitais	41
4.3	Gráfico relativo à quem joga jogos não digitais	41
4.4	Gráfico relacionado a usabilidade do jogo	42
4.5	Gráfico relacionado à confiança do jogo	42
4.6	Gráfico relacionado ao Desafio do jogo	43
4.7	Gráfico relacionado à Satisfação do jogo	43
4.8	Gráfico relacionado à Interação Social do jogo	44
4.9	Gráfico relacionado à Diversão do jogo	44
4.10	Gráfico relacionado à Atenção Focada no jogo	44
4.11	Gráfico relacionado à Relevância do jogo	45
4.12	Gráfico Percepção da aprendizagem do jogo	45
A.1	Perguntas sobre os participantes	50
A.2	Perguntas sobre Usabilidade	51
A.3	Perguntas sobre Confiança	51
A.4	Perguntas sobre Desafio	51
A.5	Perguntas sobre Satisfação	52
A.6	Perguntas sobre Interação Social	52
A.7	Perguntas sobre Diversão	52
A.8	Perguntas sobre Atenção Focada	52
A.9	Perguntas sobre Relevância	53
A.10	Perguntas sobre Aprendizagem	53

A.11 Perguntas Abertas	53
----------------------------------	----

Lista de Tabelas

2.1	Processos de um perfil básico da ISO/IEC 29110	10
3.1	Tabela de valores dos Papéis	24
3.2	Tabela de valores das Ferramentas	25

Sumário

Lista de Figuras	v
Lista de Tabelas	viii
Sumário	ix
Resumo	xi
1 Introdução	1
2 Fundamentação Teórica	4
2.1 Engenharia de Software	4
2.1.1 Processos de Software	6
2.1.2 Normas e Modelos de Qualidade de Processo de Software	8
2.1.3 Norma ISO/IEC 29110	8
2.1.4 Gamification	13
2.1.5 Trabalhos Relacionados	15
2.1.6 Limitações dos trabalhos relacionados	19
2.1.7 Considerações Finais	19
3 Apresentação do ProcSoft, ProcSoftDigital	20
3.1 ProcSoft e seus requisitos	20
3.1.1 Tabuleiro	21
3.1.2 Atividades	22
3.1.3 Os Papéis	23
3.1.4 Ferramentas	24
3.1.5 Cartas de Sorte/Revés	25
3.1.6 A Casa \$\$	26
3.1.7 Perguntas	26

3.2	ProcSoftDigital	27
3.2.1	Dinâmica do Jogo	29
3.3	A aplicação JavaScript	30
4	Metodologia de Avaliação e Resultados	39
4.1	Aplicação do Jogo	39
4.2	Informações sobre os participantes	40
4.3	Avaliação do Jogo	42
4.4	Questões abertas	45
5	Resultados e Discussões	48
5.1	Conclusões	48
5.2	Trabalhos Futuros	49
A	Questionário	50
A.1	Questões sobre os participantes	50
A.2	Questões sobre o jogo	51
A.2.1	Questões sobre Usabilidade	51
A.2.2	Questões sobre Confiança	51
A.2.3	Questões sobre Desafio	51
A.2.4	Questões sobre Satisfação	52
A.2.5	Questões sobre Interação Social	52
A.2.6	Questões sobre Diversão	52
A.2.7	Questões sobre Atenção Focada	52
A.2.8	Questões sobre Relevância	53
A.3	Questões sobre Aprendizagem	53
A.3.1	Questões Abertas	53
	Referências	54

Resumo

Tendo em mente que o ensino de Engenharia de Software e Processos de Software é muito conteudista (MOURA, 2014), causando desinteresse nos alunos em conjunto da não assimilação do conteúdo, temos o jogo ProcSoft (MOURA, 2014), cujo objetivo é dinamizar o ensino destes conteúdos, promovendo um melhor aprendizado do mesmo. Objetivando contribuir com as ações deste jogo, foi desenvolvida uma versão digital do mesmo, chamada de ProcSoftDigital, que dentre as funcionalidades gerais do jogo ProcSoft, conta com uma revisão da norma ISO/IEC 29110 (ABNT, 2012), em que o jogo foi baseada, devido ao fato de a mesma ter um foco em pequenas e médias empresas, revisão do jogo ProcSoft, junto com a sua versão digital. Para seu desenvolvimento, foram utilizadas a linguagem de programação JavaScript (MOZILLA, 2019) e o framework Quasar (QUASARDEV, 2019). Os resultados mostram que o jogo é uma dinâmica que agrada os alunos, pois assim eles tem uma interação maior com os colegas, e aprendem como cada atividade é importante em um processo de software de uma forma dinâmica.

Palavras-chave: educação em engenharia de software, processo de software, norma ISO/IEC 29110, ferramenta, gamificação.

Capítulo 1

Introdução

Os jogos além de possuírem um caráter lúdico, divertido e de entretenimento, buscam também desenvolver funções em aspectos afetivos, cognitivos e sociais das pessoas que jogam (SANTOS, 2019).

De acordo com Daniela (SANTOS, 2019) quando falamos do aspecto afetivo, o jogo nos mostra um ambiente onde as pessoas devem compreender e respeitar a vez do próximo e até mesmo de quem ganha ou perde a partida, levando em consideração que nem sempre quem ganha é o melhor.

A respeito do aspecto social, os jogos nos auxiliam em relacionamentos interpessoais, junto ao fato de aprendermos a lidar com regras e limitantes impostos em todos os ambientes, sejam nos jogos ou no nosso dia a dia como cidadão.

Quando tratamos do aspecto cognitivo, lidamos com o jeito que nos mostra as competências acadêmicas desenvolvidas pelos estudantes, como habilidades de raciocínio, liderança, comunicação, administração, inteligência emocional, estratégias e assim por diante.

Para a aquisição de conhecimento, há diversos jogos que facilitam a aprendizagem. Como exemplo, na área de engenharia de software, podemos citar o SE RPG 2.0 (AMBROSIO, 2008), o SimuleS (FIGUEIREDO et al., 2006) e o ProcSoft (MOURA, 2014). Jogos esses, que objetivam ter uma maior compreensão de conteúdos de forma dinâmica e eficiente, tentando fazer com que os alunos adquiram o conhecimento de uma maneira diferente buscando manter o interesse dos mesmos no assunto relacionado (MOURA, 2014).

Em especial o ProcSoft (MOURA, 2014), é um jogo educacional, com uma estrutura parecida com a do conhecido Banco Imobiliário®, ou seja, um jogo de tabuleiro com alguns participantes, cartas e tarefas a serem cumpridas ao decorrer do jogo. Seu objetivo é ensinar

conceitos básicos sobre a engenharia de software, unido a conceitos relacionados a definição de um processo de software para uma organização utilizando normas e modelos de qualidade de software de maneira informal e descontraída, voltada para as Very Small Entity (VSE), ou pequenas organizações.

Considerando que VSEs têm alta importância econômica para um país, haja vista dados como o reportado por (NOGUEIRA; PEREIRA, 2018), onde 97,8% das empresas formais brasileiras são micro e pequenas empresas, a engenharia de software para essas organizações passa a ser de interesse de estudos e normas para padronização do desenvolvimento de software neste cenário.

No cenário de VSEs surge a norma ISO/IEC 29110 (ABNT, 2012), na qual define-se: Os perfis de ciclo de vida de Software e os Guias, Normas e Relatórios Técnicos para pequenas organizações. O objetivo dessa norma é atender as necessidades das VSEs, fazendo com que as mesmas alcancem seus objetivos de qualidade sem depender de longos prazos e altos investimentos (ABNT, 2012).

No contexto do aprendizado de engenharia de software para VSEs, este trabalho delimita-se no melhoramento do ensino de processo de Software, devido ao fato de que o mesmo é muito conteudista (MOURA, 2014), contendo inúmeras tecnologias, práticas que englobam linguagens de programação, banco de dados, ferramentas, plataformas, padrões de projeto, gerenciamento de projetos, entre outras tantas, que geralmente causa desinteresse dos alunos em conjunto da não assimilação do conteúdo.

A proposta então, é a de que o jogo promova um melhor aprendizado dos conceitos de Processo de Software de uma forma lúdica. Levando em consideração que os alunos passam grande parte do seu tempo em frente ao computador, a versão digital do jogo, no caso, o ProcSoft, despertaria maior engajamento no tópico de processo de software.

Assim, o objetivo principal deste trabalho é desenvolver o ProcSoftDigital, um protótipo computacional para dinamizar o ensino de processo de software, contendo nele, desafios como, verificar a aderência da norma ISO/IEC 29110 no jogo ProcSoft. A ferramenta computacional também foi avaliada e testada, por meio de questões estruturadas, quanto à sua jogabilidade e se a mesma auxiliou o estudante no aprendizado.

O Capítulo 2 trata dos conceitos sobre Engenharia de Software e Processos de Software,

a norma ISO/IEC 29110, utilizada para definição e criação de um processo de software e *gamificação*, juntamente com os trabalhos relacionados. O Capítulo 3 explica o jogo ProcSoft com mais detalhes, mostrando os requisitos que foram utilizados para o desenvolvimento do ProcSoftDigital, mostrando como é sua jogabilidade junto com sua forma de avaliação.

Capítulo 2

Fundamentação Teórica

Neste capítulo são apresentados alguns conceitos necessários para o entendimento do jogo proposto nessa monografia.

A princípio são apresentados conceitos de Engenharia de Software e Processos de Software, a norma ISO/IEC 29110, utilizada para definição e criação de um processo de software e *gamification*. Por fim, serão apontados os trabalhos relacionados.

2.1 Engenharia de Software

Diversas definições sobre o que é Engenharia de Software surgiram ao longo dos anos. A definição inicial foi dada por Fritz Bauer (NAUR; RANDELL, 1968):

- Engenharia de Software é o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais.

Uma segunda definição foi a publicada por Ian Sommerville (SOMMERVILLE, 2011), e diz:

- A Engenharia de Software é uma disciplina da engenharia relacionada com todos os aspectos da produção de um software, desde os estágios iniciais de especificação do sistema até sua manutenção, depois que este entrar em operação.

Uma terceira definição proposta pela IEEE (IEEE. . . , 1990), e essa mais abrangente, afirma que:

- Engenharia de Software é a aplicação de uma abordagem sistemática, disciplinada e quantificável ao projeto, desenvolvimento, operação e manutenção de software, e o estudo dessas abordagens; isto é, a aplicação da engenharia ao software.

Entretanto, Pressman (PRESSMAN, 2010) diz que uma abordagem “sistemática, disciplinada e quantificável“ para uma equipe, pode não ser para outra. E que deve ser levado em conta a disciplina, mas também adaptabilidade e agilidade. Para ele, a engenharia de software é uma tecnologia em camadas, e que qualquer abordagem da engenharia (não somente engenharia de software) deve ter como foco a qualidade, como pode ser visto na figura a seguir.



Figura 2.1: Camadas da Engenharia de Software (MOURA, 2014)

Um dos alicerces da engenharia de software é a camada de *processos*. O processo é o ligamento que mantém as camadas coesas e possibilita o desenvolvimento do software de forma racional e dentro do prazo. O processo define uma metodologia que deve ser estabelecida para a entrega efetiva de tecnologia de engenharia de software. O processo de software constitui a base para o controle do gerenciamento de projetos (por exemplo: PMBOK) e estabelece o contexto no qual são aplicados métodos técnicos, são gerados produtos derivados (modelos, documentos, dados, etc.), são estabelecidos marcos, a qualidade é garantida e mudanças são geridas de forma apropriada.

Os *métodos* da engenharia de software fornecem as informações técnicas para como os softwares devem ser desenvolvidos. Os métodos envolvem uma ampla gama de tarefas, que incluem: comunicação, análise de requisitos, modelagem de projeto, construção de programa, testes e suporte. Estes métodos baseiam-se em um conjunto de princípios básicos que governam cada área da tecnologia e incluem atividades de modelagem e outras técnicas descritivas.

As *ferramentas* da engenharia de software fornecem apoio automatizado ou semiautomatizado para o processo e para os métodos. Quando as ferramentas são integradas de modo que

as informações criadas por ela possam ser utilizadas por outra, é estabelecido um sistema para suporte ao desenvolvimento de software.

2.1.1 Processos de Software

Processo de software é definido por Sommerville (SOMMERVILLE, 2011) como sendo um conjunto de atividades e resultado associados que produz um produto de software. Existem quatro atividades fundamentais de processo, são elas:

- *Especificação de software*: clientes e engenheiros definem o software a ser produzido e as restrições para sua operação.
- *Desenvolvimento de software*: o software é projetado e programado.
- *Validação de software*: na qual o software é verificado para garantir que é o que o cliente deseja.
- *Evolução de software*: o software é modificado para se adaptar às mudanças dos requisitos do cliente e do mercado.

Diferentes sistemas carecem de diferentes processos de desenvolvimento. Suas atividades podem ser organizadas de inúmeras maneiras e descritas em níveis distintos de detalhes, para diversos tipos de software. Contudo, o uso inadequado de um processo de software pode reduzir a qualidade ou a utilidade do produto e/ou aumentar os custos do seu desenvolvimento.

Para Pressman (PRESSMAN, 2010), processo é um conjunto de atividades, ações e tarefas realizadas na criação do produto de trabalho onde as atividades se esforçam para atingir um objetivo amplo e são utilizadas independentemente do campo de aplicação, do tamanho do projeto, da complexidade de esforços ou do grau de rigor com que a engenharia de software será aplicada. As ações envolvem um conjunto de tarefas que resultam num artefato de software fundamental. Uma tarefa se concentra em um objetivo pequeno, porém, bem definido e produz um resultado tangível.

Diante disso, percebemos que num contexto da engenharia de software, um processo não é uma prescrição rígida de como desenvolver um software. Ao contrário, é uma abordagem adaptável que possibilita às pessoas realizarem o trabalho de selecionar e escolher o conjunto

apropriado de ações e tarefas. Sempre tendo em mente que o objetivo é de entregar software dentro do prazo e com qualidade suficiente para satisfazer seus *stakeholders*, que são as pessoas ou empresas interessadas no produto.

Uma estrutura de processo estabelece a base para um processo completo de engenharia de software, identificando um pequeno número de atividades de estrutura que são aplicáveis a todos os projetos de software, independentemente de seu tamanho ou complexidade. Além disso, a estrutura de processo engloba um conjunto de atividades abrangentes que são aplicáveis em todo o processo de software. Uma estrutura genérica de processos para engenharia de software engloba cinco atividades (PRESSMAN, 2010):

- *Comunicação*: É de vital importância que ter uma comunicação boa com o cliente contribui com entendimento dos objetivos de todas as partes interessadas e a fazer o levantamento de seus requisitos.
- *Planejamento*: Definir um plano de projeto, descrevendo suas tarefas técnicas, riscos, recursos, produtos resultantes a ser produzidos e um cronograma de trabalho.
- *Modelagem*: Criar modelos, esboços e protótipos para ter uma ideia do todo e assim entender melhor as necessidades do software e o projeto que irá atender a essas necessidades.
- *Construção*: Essa atividade combina geração de código e testes necessários para revelar erros na codificação.
- *Emprego*: O software completo é entregue ao cliente, que avalia o produto entregue e fornece seu feedback.

As atividades metodológicas de um processo de software são complementadas por uma série de atividades de apoio, que em geral são aplicadas ao longo de um projeto, ajudando a equipe a gerenciar, a controlar o progresso, a qualidade, as mudanças e os riscos. Algumas destas atividades de apoio são (PRESSMAN, 2010):

- *Controle e acompanhamento do projeto*: Onde a própria equipe avalia o progresso do projeto e toma as medidas necessárias para cumprir o cronograma.

- *Administração de riscos*: Avalia riscos que possam afetar o resultado ou a qualidade do projeto/produto.
- *Garantia de qualidade de software*: Atividades que garantem a qualidade do software.
- *Revisões técnicas*: Tem como objetivo encontrar erros antes que se propaguem para atividades posteriores.
- *Medição*: Define e coleta medidas. Auxilia na entrega do software de acordo com os requisitos.
- *Gerenciamento da configuração de software*: Gerencia os efeitos das mudanças ao longo do processo.
- *Gerenciamento da reusabilidade*: Define critérios para reuso, sejam eles de código ou componentes.
- *Preparo e produção de artefatos de software*: Engloba as atividades necessárias para criar artefatos como modelos, documentos, logs, formulários e listas, por exemplo.

2.1.2 Normas e Modelos de Qualidade de Processo de Software

O processo de software é um conjunto de processos, dentre eles o de desenvolvimento. Mas há outros, tais como os de gerência de projetos e de garantia da qualidade. Para o sucesso na definição e melhoria dos processos de software, é fundamental que vários aspectos sejam considerados. Vários modelos e normas de qualidade têm surgido com o objetivo de apoiar a busca por processos de maior qualidade, apontando os principais aspectos de qualidade que devem ser considerados (FALBO, 2014). No contexto de empresas muito pequenas, destaca-se a norma ISO/IEC 29110.

2.1.3 Norma ISO/IEC 29110

Garantir a qualidade do que é produzido é sabidamente um fator crítico de sucesso no negócio das pequenas empresas. Mais que isso, é necessário ter padrões de qualidade equiparáveis aos padrões internacionais. A norma ISO/IEC 29110 é um instrumento que pode propiciar às micro e pequenas empresas (MPE) desenvolvedoras de produtos e serviços de software ter um

processo produtivo de maior qualidade e, com isso, aumentar a satisfação dos seus clientes, sua competitividade e sua capacidade de acessar novos mercados (ABNT, 2012).

Apesar do foco inicial da norma ter sido as micro e pequenas empresas, a sua versão final está endereçada a pequenas entidades (ou VSE, do inglês *very small entities*), que podem ser, por exemplo, um departamento ou equipe dentro de uma grande organização, ou, até mesmo, uma pequena empresa.

O propósito da norma ISO/IEC 29110 - Engenharia de Software - Perfis de Ciclo de Vida para VSE (*Very Small Entities*), é fazer com que as VSE alcancem seus objetivos de qualidade, sem, necessariamente, ter que demandar projetos de longo prazo e altos investimentos para adoção das normas relevantes ao seu contexto.

Espera-se que a aplicação da ISO/IEC 29110 possibilite que os gestores de pequenas empresas desenvolvedoras de software respondam positivamente as seguintes perguntas (ABNT, 2012):

- Sei o que está sendo feito e por quê?
- Posso garantir que cada membro da minha equipe tem o mesmo entendimento do que está sendo feito?
- Tenho controle sobre a integração do que dois ou mais desenvolvedores produzem?
- Posso desenvolver um produto dentro do prazo e do orçamento estabelecidos?
- Posso garantir que o software produzido, tecnicamente, faz o que deveria fazer?
- Posso garantir que o software produzido faz o que o cliente quer que ele faça?
- Quando as mudanças ocorrem – e sei que elas ocorrem – estou preparado para conhecer os seus impactos e para incorporá-las ao trabalho da forma adequada?
- Sei o ponto exato em que o meu projeto de desenvolvimento é encerrado e passa a ser um projeto de manutenção de produto?
- Estou pronto para um aumento de demanda de clientes e produtos?

Respostas positivas às perguntas anteriores podem aumentar a segurança do atendimento aos requisitos dos clientes, a conformidade dos processos aos requisitos da norma e a construção mais rápida de produtos e serviços cada vez melhores.

Deste modo, além de proporcionar uma certificação que pode ter reconhecimento internacional e facilitar o acesso ao mercado externo, a adoção da série ISO/IEC 29110 por uma VSE pode propiciar outros benefícios, como, por exemplo, o estabelecimento de processos internos de gestão e implementação de software adequados ao seu contexto, aumento da confiança e satisfação dos clientes, maior qualidade do produto ou serviço de software, aumento de patrocínio para a melhoria de processos e diminuição dos riscos de desenvolvimento.

Os processos de ciclo de vida descritos na ISO/IEC 29110, entretanto, não têm intenção de restringir ou desencorajar seu uso em organizações maiores. Podem ser usados pelas VSE tanto ao adquirir e utilizar um sistema de software, quanto ao criá-lo e/ ou fornecê-lo para uma terceira parte. Tais processos podem ser aplicados a qualquer nível na estrutura de um sistema de software e a qualquer estágio no ciclo de vida, e não têm intenção de impedir ou desestimular o uso de processos adicionais que as VSE considerem úteis.

A Tabela 2.1 a seguir apresenta os processos que compõem o perfil básico da ISO/IEC 29110. Uma organização deve adaptá-lo às suas necessidades e, também, escolher que atividades considerar e como executá-las (MOURA, 2014).

Tabela 2.1: Processos de um perfil básico da ISO/IEC 29110

Processo	Atividade	Tarefas
Gerenciamento do Projeto	Planejamento do Projeto	Revisar o escopo do Projeto
		Definir com clientes as instruções de cada entrega
		Identificar as tarefa
		Estabelecer o tempo de duração para cada tarefa
		Identificar e documentar os recursos
		Definir a composição da equipe, definindo papéis e responsabilidades
		Definir o cronograma estimado de cada tarefa do Projeto
		Calcular o custo/esforço do Projeto
		Identificar os Riscos do Projeto
		Documentar como será o controle de versão do projeto
		Criar Plano do Projeto integrando o que foi definido anteriormente
		Incluir a descrição do produto, escopo, objetivos e entregas no Plano do Projeto

Processo	Atividade	Tarefas
		Verificar e obter a aprovação do Plano do projeto
		Revisão e aceitação do Projeto
		Estabelecer o Repositório do projeto, usando a estratégia de controle de versão
	Execução do Plano do Projeto	Monitorar a Execução e Criar Relatório de Progresso
		Analisar e Estimar Pedidos de Mudança
		Conduzir Reuniões com Equipe
		Fazer backup de acordo com a Estratégia de Controle de Versão
	Controle e Avaliação do Projeto	Avaliar o progresso do Projeto de acordo com o Plano do Projeto
		Definir ações de correção para os problemas e os riscos identificados. Documentá-las.
		Identificar mudanças nos requisitos e/ou no Plano de Projeto para analisar potenciais riscos e problemas para a conclusão do Projeto
	Fechamento do Projeto	Formalizar a completude do projeto de acordo com as Instruções de Entrega estabelecidas no Plano do Projeto. Conseguir a assinatura do termo de aceite
		Atualizar o repositório do Projeto
	Implementação do Software	Início da Implementação do Software
Montar/ Atualizar o ambiente de Implementação		
Análise dos requisitos de Software		Definir as tarefas para os membros da equipe de acordo com seus papéis
		Documentar ou atualizar as Especificações de Requisitos
		Verificar e obter a aprovação do documento de Especificação de Requisitos
		Validar se a especificação de Requisitos satisfaz e atende as expectativas. Os resultados encontrados são documentados em Validação de Resultados
		Documentar a versão inicial do Manual do usuário ou atualizar
		Verificar e obter a aprovação do manual do usuário, verificar consistência com a Especificação de requisitos
		Incorporar o documento de Especificação de Requisitos e o Manual do usuário na configuração do software na baseline
		Definir tarefas para a equipe
Arquitetura de Software e Detalhamento do Design		Entender as Especificações de Requisitos
		Documentar ou atualizar o Design de Software. Analisar a Especificação de requisitos para gerar a arquitetura do software. Gerar o documento de Rastreabilidade
		Verificar e obter a aprovação do Design de Software. Os resultados serão documentados no documento de Verificação de resultados e se precisar de mudanças começar um pedido de mudanças
		Criar ou atualizar Casos de Teste e Procedimentos de Testes para testes integrados baseados na Especificação de Requisitos e Design de Software

Processo	Atividade	Tarefas
		Verificar e obter aprovação para os casos de testes e os procedimentos de teste. Documentar em Verificação de resultados
		Atualizar o documento de Rastreabilidade incorporando os Casos de teste e os Procedimentos de Teste
		Incorporar o documento de Design de Software e o de Rastreabilidade as configurações de Software como parte da baseline. Colocar casos de testes e procedimentos de teste no Repositório do Projeto
	Construção do Software	Definir as tarefas para os membros da equipe
		Entender o Design de Software
		Construir ou atualizar os componentes de Software baseado no detalhamento do Design de Software
		Criar ou atualizar os casos de teste de unidade e aplicá-los para ver se os componentes de Software implementam o que foi detalhado no design de software
		Corrigir os defeitos encontrados até que o sucesso dos testes unitários sejam alcançados
		Atualizar a rastreabilidade incorporando os componentes de software construídos ou modificados
		Incorporar os componentes de software e a rastreabilidade nas configurações de software como parte da baseline
	Integração de Software e Testes	Definir as tarefas para os membros da equipe
		Entender casos e procedimentos de testes. Definir/atualizar o ambiente de testes
		Integrar o software usando componentes de software e atualizar Casos e Procedimentos de testes para integração quando necessário
		Realizar testes de software usando Casos e Procedimentos de testes para integração e produção do software e documentar os resultados no "Test Report"
		Corrigir os defeitos encontrados e realizar testes de regressão até o critério de saída ser atingido
		Atualizar o documento de rastreabilidade, se propício
		Documentar o guia de operação do Produto ou atualizar o existente
		Verificar e obter a aprovação do Guia de Operação do Produto. Os resultados encontrados são documentados em Resultados de Verificação e as correções feitas antes do DES aprovar
Criar a Documentação do Usuário do Software ou atualizar a existente.		
Verificar e obter aprovação da Documentação do Usuário do Software. Os resultados encontrados são documentados em Verificação de resultados e as correções feitas até o documento ser aprovado pelo Cliente		

Processo	Atividade	Tarefas
		Incorporar Casos e procedimentos de Testes, Test Report, Guia de Operação do produto e Documentação do Usuário de Software para as configurações de software como parte da baseline
	Entrega do Produto	Definir as tarefas para os membros da equipe
		Entender Configuração de Software
		Criar o documento de Manutenção ou atualizar o existente
		Verificar e obter a aprovação do documento de Manutenção. Os resultados documentados na Verificação de Resultados e as correções são feitas até o documento ser aprovado pelo TL
		Incorporar o Documento de Manutenção como baseline para a configuração de Software
		Realizar entrega de acordo com as Instruções de entrega

2.1.4 Gamification

Jogos englobam um conjunto de atividades que envolvem um ou mais participantes, geralmente instigando as suas habilidades físicas e/ou mentais, requisitando que os jogadores sigam um conjunto de regras para alcançar um objetivo definido previamente (SILVA, 2010).

Cada tipo de jogo possui características distintas, os jogos de tabuleiros por exemplo, podem ser jogados a qualquer momento e em qualquer lugar, enquanto demais jogos podem estar sujeitas a condições que frequentemente não é possível controlar, como tempo e local (PEREIRA.; FUSINATO.; NEVES, 2009).

O elemento interação é fundamental para promover uma participação ativa, impedindo então que o jogador fique entediado. A flexibilidade proporciona ao jogador um certo controle e exploração do conteúdo, traçando assim um caminho particular. A competição trata-se do desafio, conflito e/ou solução de problemas que exigem habilidades do jogador como forma de atingir os objetivos (PRENSKY, 2001).

De acordo com Prensky (PRENSKY, 2001), os elementos usabilidade e grau de realidade equivalem a representação do quão fácil e intuitivo é o jogo e a forma como os efeitos dramáticos se relacionam para fornecer ao jogador situações similares às do mundo real.

O método da gamificação surgiu com o rápido desenvolvimento de novas tecnologias e como as mesmas influenciaram o processo de educação tradicional e também moldaram o papel dos estudantes e professores hoje. À medida que os alunos são expostos à uma grande quantidade de informações todos os dias, é cada vez mais desafiador sentar-se durante todo o período e

concentrar-se nas aulas dos professores. Por outro lado, sabe-se que os jogadores podem se sentar por horas sem perder o interesse e a energia, já que jogar é muito dinâmico e divertido (MASLAĆ; RAŠIĆ; VRANESIC, 2018).

Originada como método aplicado em programas de marketing e aplicações para web, com a finalidade de motivar, engajar e fidelizar clientes e usuários (ZICHERMANN; CUNNINGHAM, 2011), a gamificação pressupõe a utilização de elementos tradicionalmente encontrados nos games, como narrativa, sistema de feedback, sistema de recompensas, conflito, cooperação, competição, objetivos e regras claras, níveis, tentativa e erro, diversão, interação, interatividade, entre outros, em outras atividades que não são diretamente associadas aos games, com a finalidade de tentar obter o mesmo grau de envolvimento e motivação que normalmente encontramos nos jogadores quando em interação com bons games (FARDO, 2013).

Atualmente, a gamificação encontra na educação formal uma área bastante fértil para a sua aplicação, pois lá ela recupera os indivíduos que carregam consigo muitas aprendizagens advindas das interações com os games. Encontra também uma área que necessita de novas estratégias para dar conta de indivíduos que cada vez estão mais inseridos no contexto das mídias e das tecnologias digitais e se mostram desinteressados pelos métodos passivos de ensino e aprendizagem utilizados na maioria das escolas (FARDO, 2013).

De acordo com Marcelo Luis Fardo (FARDO, 2013), uma vantagem é a de proporcionar um sistema em que os estudantes consigam visualizar o efeito de suas ações e aprendizagens, na medida em que fica mais fácil compreender a relação das partes com o todo, como acontece nos games. Da mesma forma, um dos objetivos principais de introduzirmos uma experiência assim é para que os indivíduos sintam que seus objetivos contribuem para algo maior e mais importante, que suas ações fazem sentido dentro de uma causa maior, que é o que os elementos dos games podem proporcionar se utilizados de forma cuidadosa, da mesma forma que conseguem dentro dos mundos virtuais.

Em geral, os jogos devem ser aplicados após a matéria específica ser lecionada, pois assim o conteúdo estará em processo de assimilação, e o jogo ajudaria no aprendizado. Para medi-lo é recomendado que se aplique um questionário sobre a matéria antes e depois do jogo, assim consegue-se avaliar resultados e se eles são significativos o suficiente para indicar que um jogo realmente auxilie o processo de aprendizagem (MOURA, 2014).

2.1.5 Trabalhos Relacionados

Ao longo do levantamento bibliográfico para realização desta monografia, destacamos a seguir os trabalhos que mais se identificam: SE RPG 2.0 (AMBROSIO, 2008), por já possuir uma versão digital, simulES (FIGUEIREDO et al., 2006), pelo fato de também simular um processo de desenvolvimento de software e o ProcSoft (MOURA, 2014), por ser o jogo escolhido para o desenvolvimento da versão digital.

2.1.5.1 SE RPG 2.0

O SE RPG 2.0 (AMBROSIO, 2008) é uma nova versão do jogo intitulado de Software Engineering-Roleplaying Game. A ferramenta propicia o desenvolvimento de software, envolvendo o jogador em uma trama onde há personagens a serem interpretados, necessidades de gerenciamento de tempo, custo e eventuais situações não esperadas, bem como a avaliação do cliente quanto ao resultado do produto (AMBROSIO, 2008).

O jogo contém uma sequência de passos que simulam o funcionamento de uma empresa de software, onde o jogador deverá decidir qual projeto irá desenvolver. Após, é necessário definir o modelo de desenvolvimento (cascata e iterativo), linguagem de implementação e a contratação de personagens que irão fazer parte da sua equipe. Cada personagem possui habilidades próprias, e caberá ao jogador escolher quais mais se identificam com as características necessárias para o desenvolvimento do projeto.

Durante o andamento do jogo, o jogador deverá distribuir tarefas para sua equipe. Ao concluir a etapa de codificação o projeto poderá ser entregue e assim um relatório com informações a respeito do cumprimento das metas, satisfação do cliente e se o modelo escolhido foi correto para o desenvolvimento será entregue (AMBROSIO, 2008).

O cenário do jogo apresenta 3 ambientes, sendo eles: recepção e sala de reuniões, sala da direção e sala da produção, que seria o principal ambiente do jogo como mostra na figura 2.2.

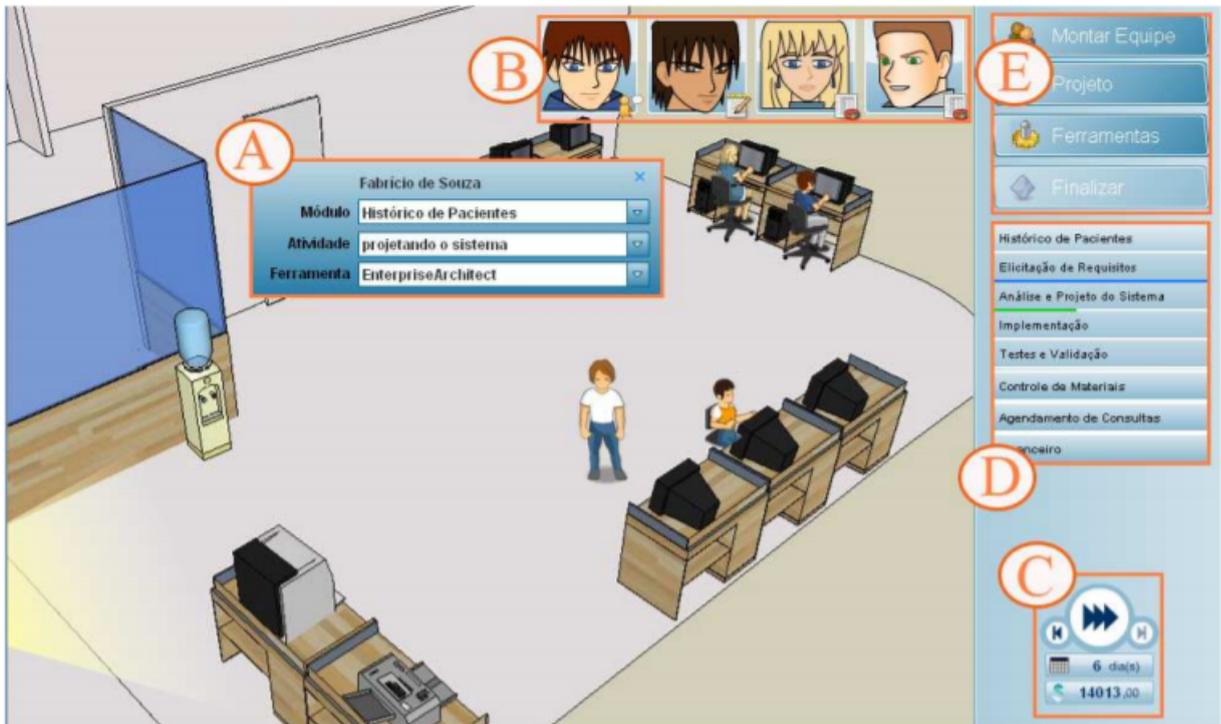


Figura 2.2: Ambiente principal do SE RPG 2.0 (AMBROSIO, 2008)

O ambiente se refere a sala de desenvolvimento e é composta por: (a) atribuição de tarefas; (b) representação das personagens; (c) controle de tempo; (d) processo das atividades do projeto; (e) edição de equipe.

2.1.5.2 SimulES

O SimulES (FIGUEIREDO et al., 2006) - Simulador de Uso da Engenharia de Software é um jogo educacional de cartas que simula o processo de desenvolvimento de software. No jogo, ocorre uma disputa entre 4 e 8 jogadores, onde toma-se como objetivo terminar um projeto de software, e o vencedor será quem implantá-lo primeiro.

O projeto de software escolhido por cada jogador é feito de forma aleatória, através de um dos cartões de projetos disponíveis. As informações desse cartão devem ficar visíveis para todos os jogadores. Cada jogador possui seu próprio tabuleiro como visto na figura 2.3, que deve ser montado após a escolha do projeto.

	Engenheiro ES1 Janaina	Engenheiro ES2 Carlos	Engenheiros de Software		
	Profissional veterano, mas com pouca habilidade no desenvolvimento. Salário: 40 K Habilidade 1 Maturidade 4	Experiência em eng. de software, mas não é amigável à equipe. Salário: 70 K Habilidade 5 Maturidade 1	Engenheiro 3	Engenheiro 4	Engenheiro 5
Requisitos					
Desenhos					
Códigos					
Rastros					
Ajudas					

Figura 2.3: Tabuleiro Simules (FIGUEIREDO et al., 2006)

As cartas do baralho são dispostas em 4 montes, sendo eles: engenheiros de software, problemas e conceitos, artefatos brancos e artefatos cinzas. A cada rodada, o jogador deverá lançar o dado, e de acordo com o número sorteado, o mesmo deverá pegar no monte de cartas, um número equivalente ao valor do dado, não se fazendo necessário pegar apenas cartas de um dos montes. Essas cartas irão simular os problemas e desafios encontrados em um projeto de software de uma empresa.

O monte que contém o tipo de carta "engenheiro de software", é o principal método de evolução no jogo. Levando em conta que os engenheiros possuem 4 opções de tarefas: construir artefatos, inspeção de artefatos, corrigir defeito e integrar artefatos em um módulo.

O final do jogo é dado quando um jogador conseguir completar os componentes necessários de acordo com o cartão de projeto que por ele foi selecionado no início da partida. Após, é feita a validação dos requisitos por parte do cliente, e se não for encontrado nenhum defeito, o jogador é dado como vencedor.

2.1.5.3 ProcSoft

O ProcSoft é um jogo educacional baseado na norma ISO/IEC 29110, explicada anteriormente. Segundo Moura (MOURA, 2014), o jogo foi definido visando apoiar alunos a criarem

processos de software que contemplassem as atividades de um ciclo fundamental de desenvolvimento de software e atividades de apoio como gerência de projetos e gerência de configuração.

ProcSoft é um jogo de tabuleiro não digital, e seu *design* final foi baseado no popular jogo comercial Banco Imobiliário®. Sendo necessários para execução jogo, o tabuleiro, cartas, dados, peões e peças para marcação de propriedades. Um exemplo do tabuleiro pode ser visto na figura 2.4 mostrada a seguir.

São até 4 equipes compostas por 2 a 4 jogadores. Os mesmos simulam uma equipe de desenvolvimento de software que necessita criar um processo de software adequado para a construção de software.

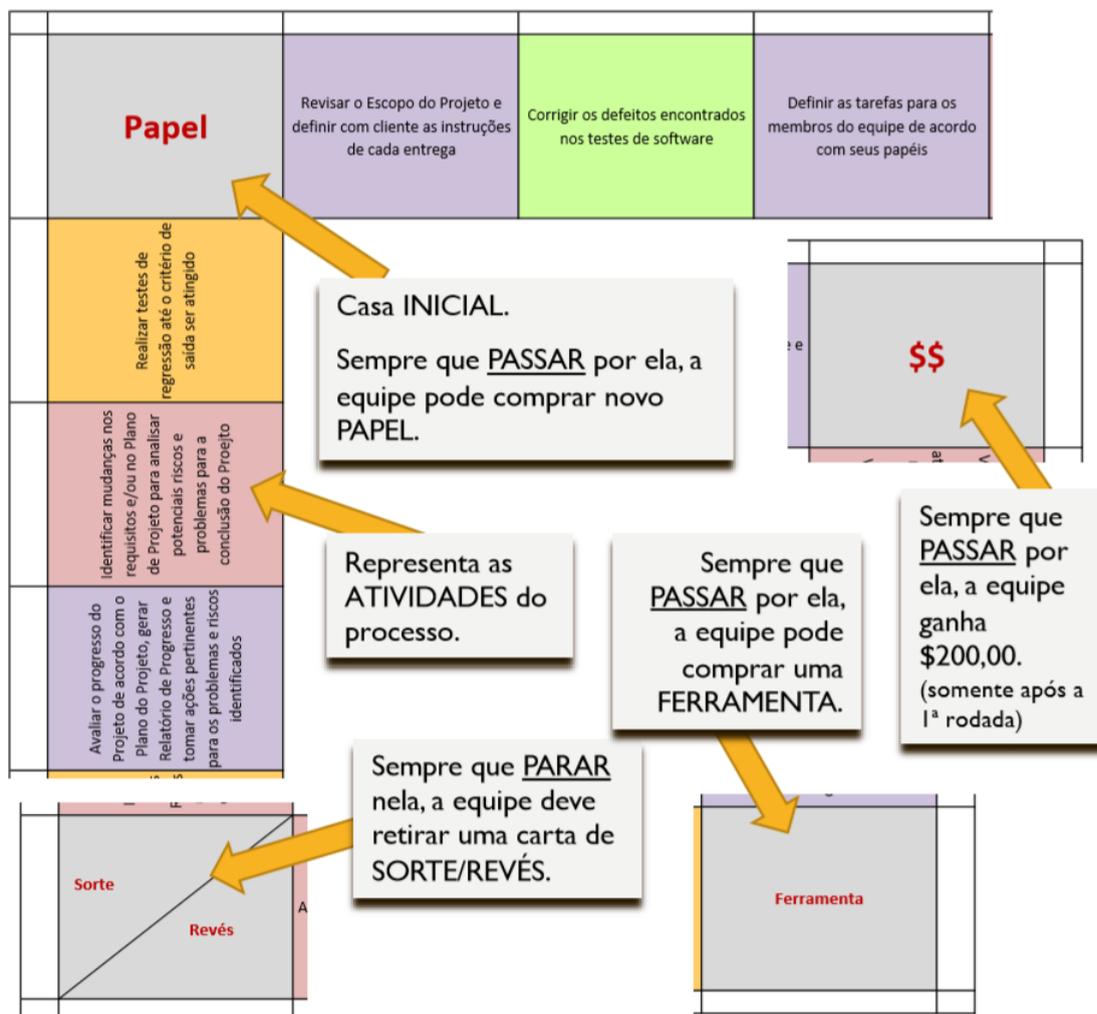


Figura 2.4: Tabuleiro do ProcSoft (MOURA, 2014)

De acordo com Moura (MOURA, 2014), o objetivo do jogo é capacitar os participantes a

entenderem quais são as atividades comuns a um processo de software e sua relação com os papéis existentes em projeto de software e as ferramentas de apoio que podem ser utilizada. Ainda, a expectativa é a de que os participantes entendam o quão importante é a atribuição de atividades e papéis específicos, a necessidade de ferramentas adequadas para auxiliar a execução das atividades e importância do uso de engenharia de software para a execução de um processo.

2.1.6 Limitações dos trabalhos relacionados

Embora os trabalhos citados simulem parte ou todo o processo de desenvolvimento de software, eles focam em ensinar como desenvolver um software e os seus processos para empresas no geral, ou seja, por exemplo, não levam em consideração o tamanho dessas organizações. O ProcSoft possui um diferencial, afinal o mesmo é baseado na norma ISO/IEC 29110, que é voltada para o desenvolvimento de software para as VSE (*Very Small Entities*), entretanto, não possui uma versão digital. Detalhes do jogo ProcSoft e seus requisitos funcionais serão explicados em breve, visto que o objetivo desta monografia é desenvolver um protótipo digital para o mesmo.

2.1.7 Considerações Finais

Este capítulo apresentou uma breve revisão sobre engenharia de software, processos de software, gamificação, a norma ISO/IEC 29110 e mostrou também alguns trabalhos relacionados. No próximo capítulo será apresentado o jogo ProcSoft e será explicado como sua versão digital, o ProcSoftDigital será desenvolvido.

Capítulo 3

Apresentação do ProcSoft, ProcSoftDigital

Neste capítulo o jogo ProcSoft será explicado com mais detalhes, ensinando como jogar e mostrando os requisitos que serão utilizados no ProcSoftDigital. Serão mostradas também algumas telas do jogo ProcSoftDigital, será explicado como será sua jogabilidade e a sua forma de avaliação.

3.1 ProcSoft e seus requisitos

Como dito anteriormente, ProcSoft é um jogo de tabuleiro não digital, e seu *design* final foi baseado no popular jogo comercial Banco Imobiliário®. Sendo necessários para execução jogo, alguns requisitos indispensáveis (MOURA, 2014):

- 28 cartas de papéis;
- 16 cartas de ferramentas;
- 44 cartas de atividades;
- 1 gabarito de processos;
- 20 cartões sorte/revés;
- 4 peões;
- 2 dados;
- 180 notas de dinheiro;

- 1 tabuleiro;
- 60 cartas com perguntas sobre engenharia de software;
- 4 cartilhas com as regras.

3.1.1 Tabuleiro

O tabuleiro utiliza atividades de processos baseados na norma ISO/IEC 29110, distribuídos em 44 casas, de forma a utilizar as tarefas mais relevantes e evitar repetições. Tais atividades são distribuídas em 4 diferentes grupos e cores exemplificados na figura 3.1, sendo eles:

- Atividades executadas pelo Gerente – Roxo;
- Atividades executadas pelo Analista – Rosa;
- Atividades executadas pelo Desenvolvedor – Verde;
- Atividades executadas pelo Testador – Laranja.

Revisar o Escopo do Projeto e definir com cliente as instruções de cada entrega	Corrigir os defeitos encontrados nos testes de software	Definir as tarefas para os membros do equipe de acordo com seus papéis	Formalizar a completude do projeto e realizar entrega de acordo com o Plano do Projeto
---	---	--	--

Figura 3.1: Grupos de atividades. Adaptado de (MOURA, 2014)

No tabuleiro, também são encontradas 4 casas com conteúdos diferentes das atividades de processos, sendo essas casas: Sorte/Revés, \$\$ (casa que indica "dinheiro"), papel e ferramentas. Além disso, ao parar em cada casa de atividade, o jogador deverá responder uma das 60 perguntas sobre engenharia de software.

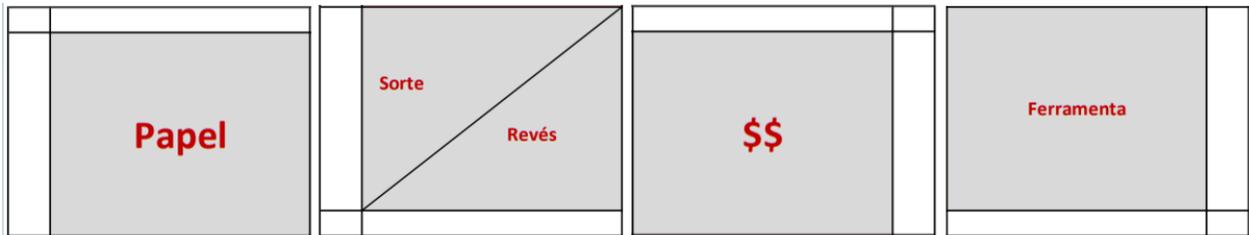


Figura 3.2: Grupos de atividades. Adaptado de (MOURA, 2014)

Essas casas com conteúdos diferenciados, tem como objetivo não só dar dinâmica ao jogo, mas também criar facilidades e dificuldades aos jogadores, mostrando necessidades e acontecimentos que podem ocorrer durante a idealização dos processos de software de uma empresa.

3.1.2 Atividades

As atividades representam as tarefas existentes na norma ISO/IEC 29110 para a criação de processos. Tarefas essas selecionadas por Moura (MOURA, 2014), disposta de forma aleatória no tabuleiro e pertencentes a 4 tipos:

- Planejamento;
- Execução e Monitoração;
- Análise e Projeto;
- Codificação/Testes/Entrega.

Toda vez que o jogador parar em uma casa de atividade, ele pode comprá-lá por \$100. Mas para que o preço seja esse, ele deve responder a pergunta sobre engenharia de software corretamente, ou o valor da atividade subirá para \$200. Ao comprá-la, o jogador se torna o detentor daquela atividade. Se não possuir a ferramenta adequada para ela, será cobrado \$50 a mais. Ao fim do jogo, se não possuir a carta ferramenta, sua pontuação cairá pela metade. Entretanto, se possuir a carta de papel Sênior correspondente à atividade ganhará 50 pontos por atividades correspondente.

As equipes deverão identificar as Atividades corretamente ao tipo em que ela pertence. No jogo, as Atividades não precisam ser colocadas na ordem dentro do tipo ao que ela pertence. A Figura 3.5 apresenta um exemplo de carta de Atividade.

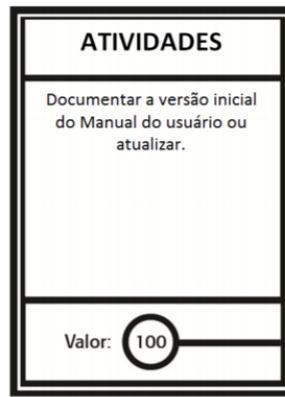


Figura 3.3: Carta de Atividade (MOURA, 2014)

3.1.3 Os Papéis

Os Papéis representam cargos e responsabilidades que existem em uma empresa real de desenvolvimento de software. São cargos essenciais em uma equipe para que possam executar as atividades de Engenharia de Software definidas de acordo com o processo estabelecido pela empresa e também consigam gerenciar o projeto ao longo de sua duração até a obtenção do produto final.

Antes de o jogo começar, os grupos deverão definir quem interpretará cada papel no grupo: Gerente, Analista, Desenvolvedor e Testador. Na primeira rodada as equipes poderão comprar 2 papéis. Os papéis funcionam como critério de capacidade para poder avançar no jogo, uma vez que algumas ferramentas só poderão ser manipuladas por alguém que tenha um determinado papel e de acordo com seu nível (Júnior, Sênior). A equipe só poderá comprar uma carta de papel Sênior se já possuir a de Júnior. A carta de Gerente só possui um nível (equivalente ao Sênior). Caso uma equipe tenha menos pessoas que os papéis previstos no jogo, alguns membros da equipe deverão acumular papéis.

O grupo poderá escolher se quer promover um dos papéis toda vez que passar pela casa de papéis (casa inicial do jogo). Para promover, é preciso pagar um valor especificado na carta de papel. Cada papel possui uma bonificação diferente e dá acesso a mais ferramentas, podendo assim acumular mais pontos. O responsável, por representar a equipe no tabuleiro, será o que estiver com o papel de Gerente, caso a equipe não tenha comprado este papel, um membro será

escolhido pela equipe para representá-la. Um exemplo de carta de Papéis está na figura a seguir. E seus valores na tabela 3.1:

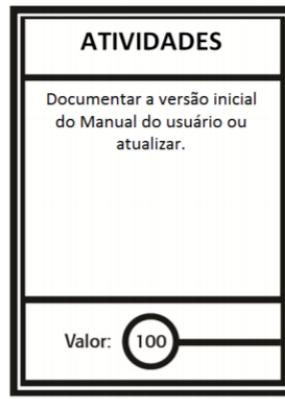


Figura 3.4: Carta de Atividade (MOURA, 2014)

Tabela 3.1: Tabela de valores dos Papéis

Papel	Tipo 1	Valor 1	Tipo 2	Valor 2
Gerente	-	-	-	\$300
Analista	Júnior	\$100	Sênior	\$300
Desenvolvedor	Júnior	\$100	Sênior	\$300
Testador	Júnior	\$100	Sênior	\$300

3.1.4 Ferramentas

As Ferramentas são cartas que ajudam a realizar as tarefas que se encontram no tabuleiro. Elas são de diferentes tipos: Gerenciamento, Análise e Projeto, Codificação e Testes. Elas podem ser adquiridas ao longo do jogo para auxiliar no processo. Algumas ferramentas só podem ser utilizadas por determinados papéis (Gerente, Analista, Desenvolvedor ou Testador) e níveis (Júnior, Sênior ou Uso Geral). Elas variam de preço de acordo com a sua qualidade/posição no mercado/etc. Você poderá adquirir ferramentas sempre que passar pela casa Ferramentas no tabuleiro. A Figura 3.5 apresenta um exemplo de carta de Ferramentas. Os valores das ferramentas são apresentados na tabela 3.2.

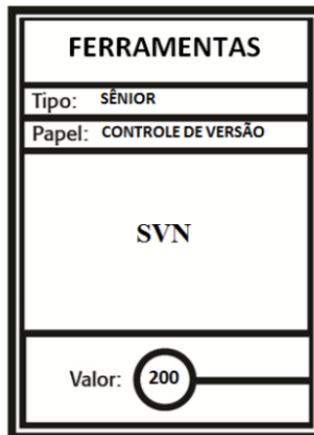


Figura 3.5: Carta de Ferramenta (MOURA, 2014)

Tabela 3.2: Tabela de valores das Ferramentas

Tipo	Valor
Uso Geral	\$100
Júnior	\$150
Sênior	\$200

3.1.5 Cartas de Sorte/Revés

As equipes deverão tirar Sorte/Revés sempre que pararem na casa de mesmo nome. Uma carta deve ser tirada e a ordem executada imediatamente. Depois, a carta deverá ser devolvida para baixo da pilha. Toda vez que um jogador adversário parar em uma Atividade que outra equipe possui, a equipe também deverá retirar uma carta de Sorte/Revés. Caso dê sorte o bônus será revertido para quem possui a Atividade, caso dê Revés a penalidade será para a equipe que parou na Atividade de outra equipe (nesse caso o dinheiro será dado ao banco). A Figura 3.6 apresenta exemplo de Carta de Sorte/Revés.

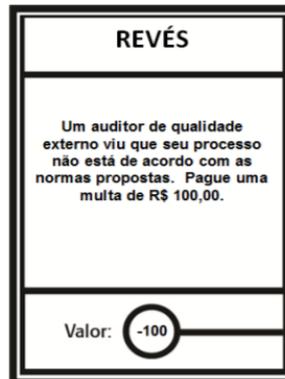


Figura 3.6: Carta de Revés (MOURA, 2014)

3.1.6 A Casa \$\$

A casa \$\$ (Dinheiro) é utilizada como uma casa bônus, pois cada vez que um peão alcançar esta casa, a equipe receberá uma bonificação, exceto na primeira rodada. O objetivo desta casa é poder auxiliar as equipes a comprarem e evoluírem atividades, ferramentas e papéis ao longo do jogo.

3.1.7 Perguntas

As cartas de perguntas contêm 60 perguntas de diversas áreas da Engenharia de Software. O objetivo principal é auxiliar o ensino uma parte do conteúdo da disciplina de forma dinâmica e divertida e atrelar este conhecimento ao jogo, utilizando-as de maneira a incrementar a jogabilidade.

Antes de poder comprar uma carta de um determinado tipo, é necessário responder uma pergunta sobre Engenharia de Software. Se a equipe souber responder, paga o valor de face. Não sabendo responder, paga o dobro pela atividade. As perguntas podem ter respostas diferentes do gabarito desde que as equipes participantes concordem que a resposta dada também é válida.

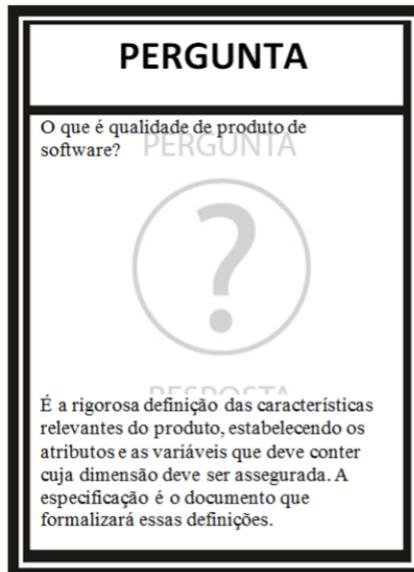


Figura 3.7: Carta de Pergunta (MOURA, 2014)

3.2 ProcSoftDigital

ProcSoftDigital é um jogo inteiramente baseado no jogo ProcSoft e na norma ISO/IEC 29110. Nesta seção serão feitas algumas conexões com o jogo ProcSoft e será explanado a forma correta de jogar o mesmo.

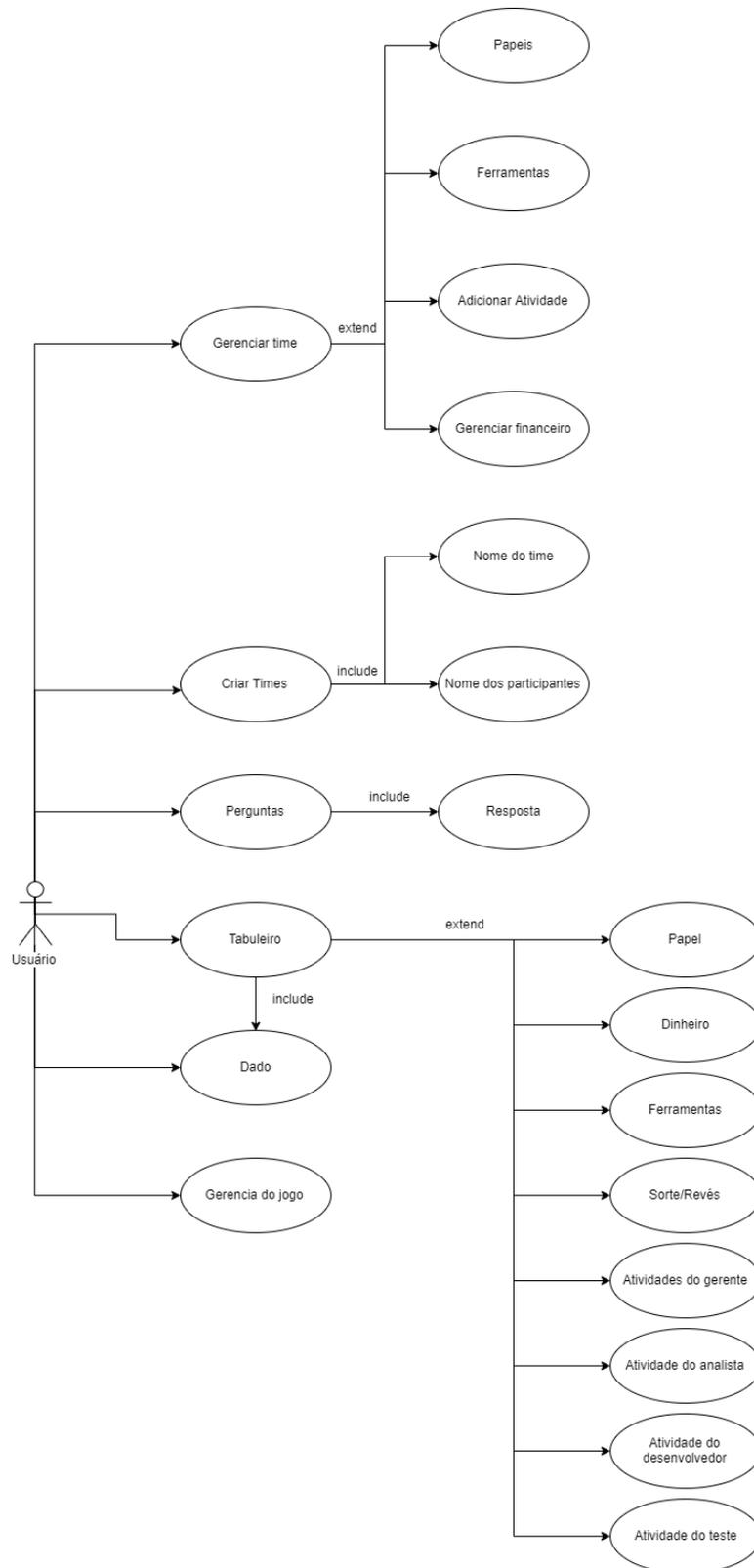


Figura 3.8: Casos de Uso

O jogo ProcSoftDigital foi desenvolvido em JavaScript (MOZILLA, 2019), uma linguagem de programação que permite implementar funcionalidades mais complexas em páginas web. A cada momento uma página web faz mais do que apenas mostrar informações estáticas para você - elas mostram em tempo real conteúdos atualizados, ou mapas interativos, vídeos, entre outras funcionalidades.

Foi utilizado também o framework Quasar (QUASARDEV, 2019) que é um framework baseado em Vue.js de código aberto licenciado pelo MIT, que permite que você, como desenvolvedor web, crie rapidamente sites/aplicativos responsivos em diversas maneiras. Nenhuma outra ferramenta de desenvolvimento de jogos foi escolhida devido ao conhecimento já obtido com a linguagem JavaScript.

3.2.1 Dinâmica do Jogo

A princípio, cada equipe deve receber \$2100. Porém, como estamos trabalhando em sua versão digital, não temos mais a contagem das notas, por exemplo, não é necessário entregar 2 notas de \$500, 5 notas de \$100 e 12 notas de \$50, apenas o valor total de cada equipe.

Ao início do jogo, cada equipe joga o dado uma vez, e a que sortear o maior número começa jogando, seguido pela equipe que tirar o segundo maior e assim por diante. Ao jogar os dados, se os número forem pares, o jogador pode jogar novamente.

O jogo segue o padrão do banco imobiliário, então a cada vez que cair numa casa de atividade, você pode comprá-la ou não, do mesmo modo como é feito com as propriedades no banco imobiliário, porém, para comprá-la é necessário responder a pergunta ligado a engenharia do software, selecionada de forma aleatória, de uma das 60 perguntas já estabelecidas. Se responder corretamente, a equipe obtém a atividade correspondente.

Como dito anteriormente, ao cair na casa Papéis, você poderá dar um *upgrade* nas suas cartas, pagando o valor estipulado na tabela 3.1. Ao parar na casa \$\$ o jogador receberá uma bonificação de \$200 toda vez que passar, exceto a primeira. Na Sorte/Revés, deverá ser pego uma carta no baralho correspondente e averiguar o que diz na mesma. E na casa de ferramentas, o jogador poderá comprar uma ferramenta que o auxiliará desenvolvimento do produto, porém, algumas ferramentas só poderão ser utilizadas por jogadores que possuírem papéis específicos.

Se o jogador ficar muito indvidado ele poderá declarar falência, porém, deverá passar por

várias etapas, antes de simplesmente declará-la. Mais detalhes no Anexo 1, que contém todas as regras detalhadas do jogo.

O tempo indicado para o fim do jogo é de 60 minutos, porém, se os jogadores optarem por um tempo menor ou maior, não haverá complicações. O vencedor será aquele que conseguir encaixar o maior número de atividades coerentes nas fases corretas do desenvolvimento, utilizando da melhor forma os papéis e ferramentas para conseguir uma maior bonificação.

As regras do ProcSoftDigital são as mesmas do ProcSoft, as mudanças evidentes seriam no design do jogo, incluindo tabuleiro, cartas, dados e a moeda do jogo.

3.3 A aplicação JavaScript

O tabuleiro do jogo ProcSoftDigital foi feito com 48 botões utilizando HTML e CSS. Como no ProcSoft (MOURA, 2014), os tabuleiros também seguem um padrão de cores, sendo eles:

- Roxo - Para atividades realizadas por gerentes;
- Vermelho - Para atividades realizadas pelo Analista;
- Verde - Para atividades realizadas pelo Desenvolvedor;
- Laranja - Para atividades realizadas pelo Testador.

Abaixo podemos ver o tabuleiro desenvolvido:

PAPEL	2	3	4	5	6	7	8	9	10	11	DINHEIRO
44											13
43											14
42											15
41											16
40											17
39											18
38											19
37											20
36											21
35											22
S/R	33	32	31	30	29	28	27	26	25	24	FERRAM.

Figura 3.9: Tabuleiro do ProcSoftDigital

Ao iniciar o jogo, assim como no ProcSoft (MOURA, 2014), é necessário jogar o dado para iniciar a partida, e para a movimentação das equipes pelo tabuleiro. Para isso temos um botão ao lado do tabuleiro que ao ser clicado, gera um número aleatório entre 1 e 6, e é mostrado na tela ao seu lado, como podemos ver a seguir:



Figura 3.10: Dado do ProcSoftDigital

Após as equipes terem seus times definidos, e o sorteio da ordem de jogo for realizado, adicionamos os times. Para fazer isso, temos o botão mostrado na figura 3.11.



Figura 3.11: Adicionar Time do ProcSoftDigital

Ao clicar no botão apresentado na figura 3.11, um *modal* como na figura 3.12 é aberto, onde podemos adicionar um nome para o time e os participantes do mesmo.

O modal é uma caixa de diálogo com o título "Nome do time". Possui dois campos de texto: o primeiro, rotulado "Time 1", está vazio; o segundo, rotulado "Nome dos participantes", contém o texto "Ivonei, Claudia, Victor". Na base do modal, há dois botões: "CANCEL" e "ADICIONAR TIME".

Figura 3.12: Adicionar nome do time e participantes do ProcSoftDigital

Quando o botão Adicionar Time no *modal* é selecionado, um *card* para o time é criado, como pode ser visto na figura a seguir:

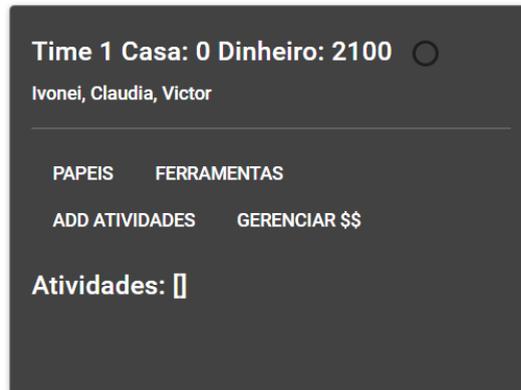


Figura 3.13: Card do Time no ProcSoftDigital

Como podemos ver no *Card*, o dinheiro do Time já é inicializado em \$2100. O primeiro passo então é selecionar o time, marcando o *Radio button* do lado do dinheiro do time. Após ter feito isso, deve se escolher os 2 papéis que cada time pode selecionar no início do jogo, isso é feito clicando em “PAPEIS“ no *card*. Abrindo então uma caixa de seleção para os times selecionarem os papéis desejados, cobrando o valor de cada uma, indicados na tabela 3.1.

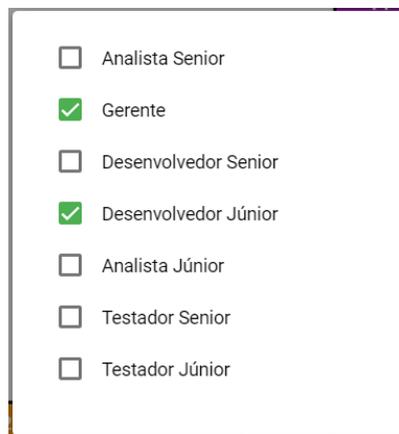


Figura 3.14: Caixa de seleção de Papeis no ProcSoftDigital

Após feitas essas ações, temos o início do jogo. Ao jogar o dado e parar em uma das atividades, por exemplo, a atividade 6 que é da cor verde, e portanto, representa a atividade de um Desenvolvedor, verificamos se o time possui esse Papel. Se a resposta for negativa, a vez passa para o próximo time, porém, se for positiva, como é o caso, clicamos na mesma, onde um *modal* é aberto e mostrado uma das atividades que são realizadas por um desenvolvedor segundo a norma ISO/IEC 29110.

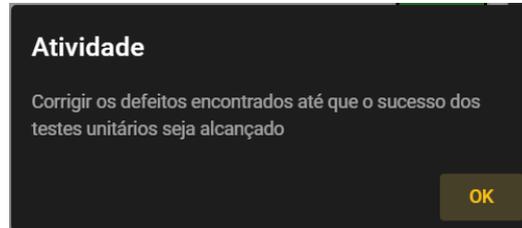


Figura 3.15: Atividade no ProcSoftDigital

Porém, para que essa atividade seja adquirida pelo time, é necessário responder uma pergunta relacionada a Engenharia de Software, onde temos 60, e são escolhidas aleatoriamente ao clicar no botão “PERGUNTAS”.



Figura 3.16: Botão perguntas no ProcSoftDigital

Ao clicar no botão 3.16, um *modal* com uma das perguntas é aberto, como por exemplo, a pergunta a seguir:

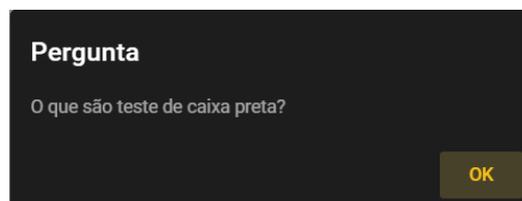


Figura 3.17: Exemplo de Pergunta no ProcSoftDigital

Nesse momento, o time que está na vez deve respondê-la corretamente, pois se errar deverá pagar \$200 pela atividade que custa \$100. Existem duas formas de avaliar se a resposta está correta, a primeira e mais indicada é que o Professor e os outros times aceitem a resposta e digam se a mesma está correta ou errada, causando uma discussão entre os participantes e diferentes pontos de vista. A segunda forma, seria utilizar a resposta proposta pelo jogo, que pode ser vista ao clicar no botão “OK” do *modal* mostrado na figura 3.17. Ao clicar no mesmo, a resposta é mostrada da seguinte forma:

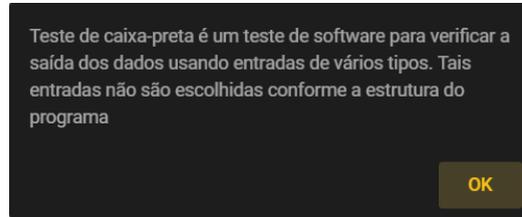


Figura 3.18: Exemplo de Resposta no ProcSoftDigital

Após a resposta ser avaliada, então devemos entregar a atividade e cobrar o valor, seja ele de \$100 caso a resposta esteja correta ou \$200 se estiver incorreta. Para alterar o valor, clicamos no botão “GERENCIAR \$\$”, assim abrimos um modal onde podemos adicionar ou diminuir fundos da equipe que está na vez.

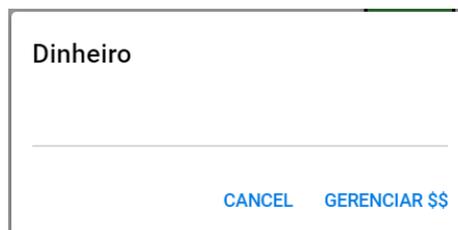


Figura 3.19: Gerenciar Dinheiro no ProcSoftDigital

Nesse momento também deve ser adicionada a atividade ao *card* do time. Para que isso seja feito, devemos clicar em “ADD ATIVIDADES”, e colocar o número da casa a qual a atividade pertence, como pode ser vista a seguir:

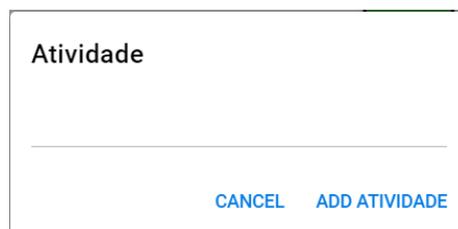


Figura 3.20: Adicionar atividade no ProcSoftDigital

O resultado final do *card* fica como mostra a imagem a seguir:



Figura 3.21: Card com atividade adicionada no ProcSoftDigital

O botão de “FERRAMENTAS” no *card* do time funciona da mesma forma que o de “PAPEIS”, e toda vez que o time passar pela casa “ferram.” no tabuleiro, poderá comprar uma ferramenta.

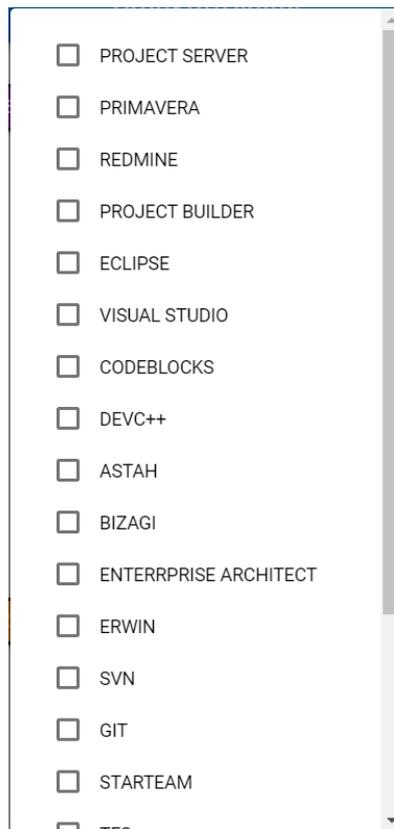


Figura 3.22: Ferramentas no ProcSoftDigital

Algumas outras funcionalidades seriam as de ao clicar no botão “Dinheiro” no tabuleiro, é acrescido \$200 no Dinheiro do time selecionado. Outra seria a de ao parar na casa “S/R”,

que significa Sorte ou Revés, e clicar neste botão, é mostrado uma das cartas de sorte ou revés propostas pelo jogo, como por exemplo o da figura abaixo:

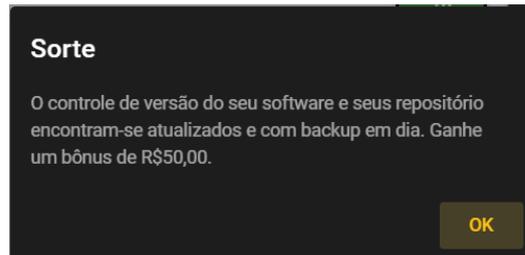


Figura 3.23: Carta de sorte no ProcSoftDigital

Vale lembrar que ao parar em uma atividade pertencente a outro time, também é necessário clicar no "S/R", caso seja "Sorte", o Banco que deverá pagar a quantia ao dono da atividade, e se for "Revés", o time que parou na atividade é quem deve realizar o pagamento.

Segue uma imagem do jogo mostrado por completo:

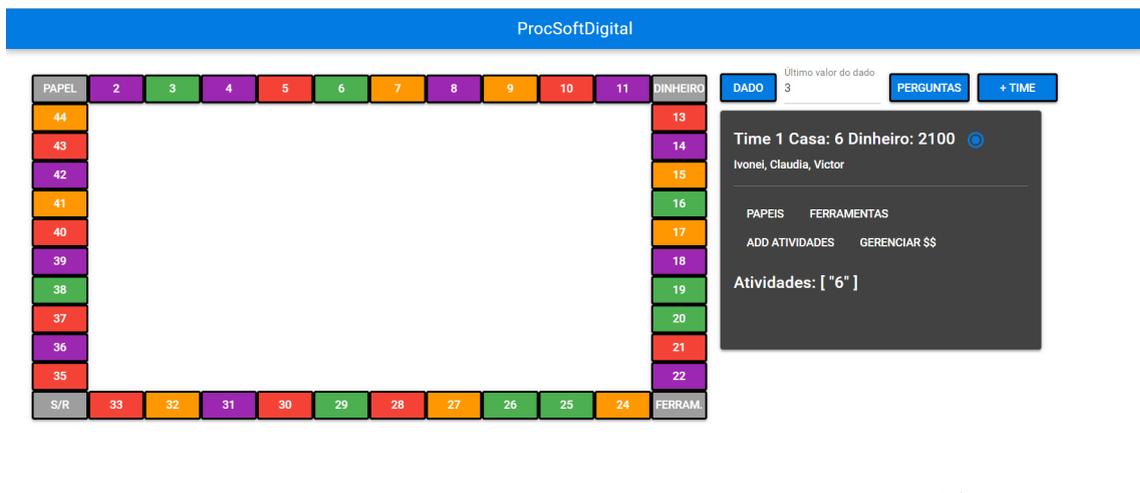


Figura 3.24: ProcSoftDigital

É possível também ver um diagrama de fluxo simplificado do funcionamento e dinâmica do jogo:



Figura 3.25: Diagrama de fluxo

Quanto às perguntas, atividades, cartas de sorte/revés, papéis e ferramentas, todas as informações estão localizadas em um arquivo chamado de enums.js, onde são postas as informações contidas nas mesmas, como por exemplo, valores das ferramentas, dos papéis, das cartas de ferramentas, juntamente com as respostas indicadas pra cada uma das perguntas.

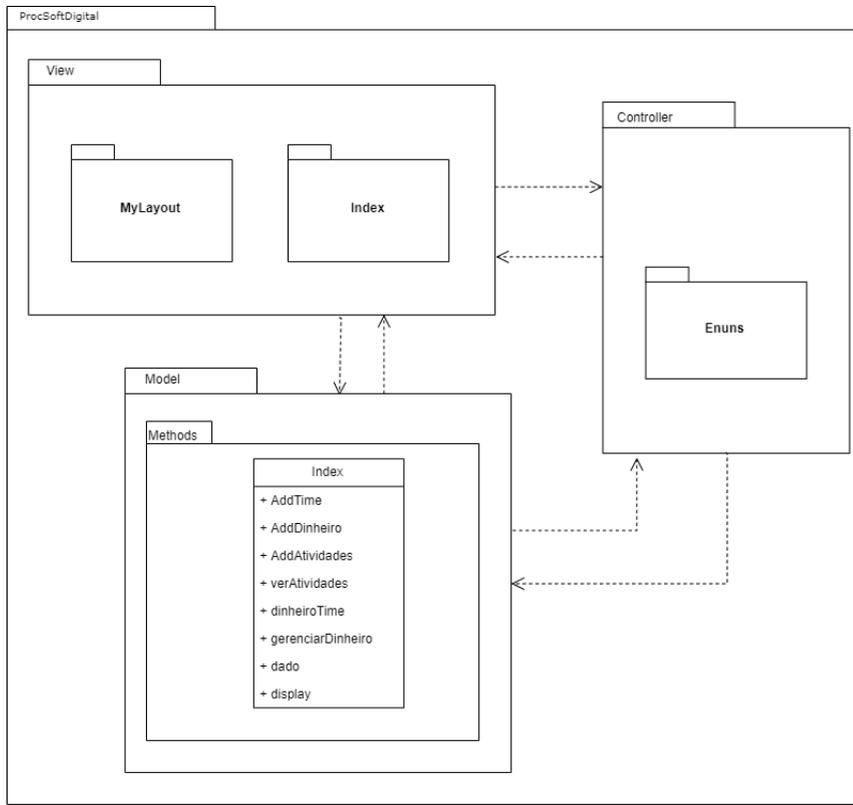


Figura 3.26: Diagrama de Pacotes

Capítulo 4

Metodologia de Avaliação e Resultados

Este capítulo apresenta os resultados obtidos após um estudo de viabilidade (SHULL; CARVER; TRAVASSOS, 2001) do jogo, bem como a sua forma de avaliação. O estudo de viabilidade visa entender se a ferramenta apresenta resultados utilizáveis dentro de um tempo apropriado (SHULL; CARVER; TRAVASSOS, 2001). Resultado positivo nessa avaliação de viabilidade proporciona que a ferramenta possa ser avaliada em trabalhos futuros com outras metodologias, tais como estudos de caso, experimentos, etc.

4.1 Aplicação do Jogo

O jogo ProcSoftDigital foi aplicado na turma de Processo de Engenharia de Software II, do curso de Ciência da Computação da Universidade Estadual do Oeste do Paraná (UNIOESTE), campus Cascavel. Nove alunos participaram da aplicação do jogo. Eles foram divididos em 3 equipes com 3 jogadores cada, escolhidos pelos próprios alunos por afinidade com os colegas.

A aplicação do jogo teve início às 13:30hs. Os 10 minutos iniciais foram de apresentação do jogo, suas regras e como jogá-lo. Como todos os alunos tinham familiaridade com o jogo Banco Imobiliário, eles entenderam rapidamente a ideia de como o jogo funciona. Após os minutos iniciais, iniciamos o jogo, que foi aplicado por 50 minutos, ou seja, até às 14:30hs. Entre 14:30hs até as 14:45hs foi aplicado o questionário sobre a ferramenta.

Para verificar se o jogo pode ser útil e motivador para os alunos, foi necessário avaliar a percepção em relação ao jogo e também se foi possível aprender algum conteúdo relacionado à Engenharia de Software. Para isto foi utilizado um questionário adaptado de (MOURA, 2014),

cujo trabalho aplicou-o para avaliar o jogo ProcSoft. Este questionário está descrito no Apêndice A.

4.2 Informações sobre os participantes

A faixa etária dos 9 participantes é homogênea, o conteúdo abordado sobre a norma ISO/IEC 29110 e sobre as VSE não haviam sido explicados em sala, apenas os conteúdos geral sobre processos de software e engenharia de software, então as atividades que continham no tabuleiro eram todas novas para os participantes, alguns dados dos alunos podem ser vistos nos gráficos seguintes:



Figura 4.1: Gráfico de faixa etária

Também foi perguntado aos participantes quanto a sua interação com jogos, sendo eles digitais ou não, obtendo assim, os gráficos relativos à quem joga jogos digitais com frequência:

Com que frequência você costuma jogar jogos digitais?

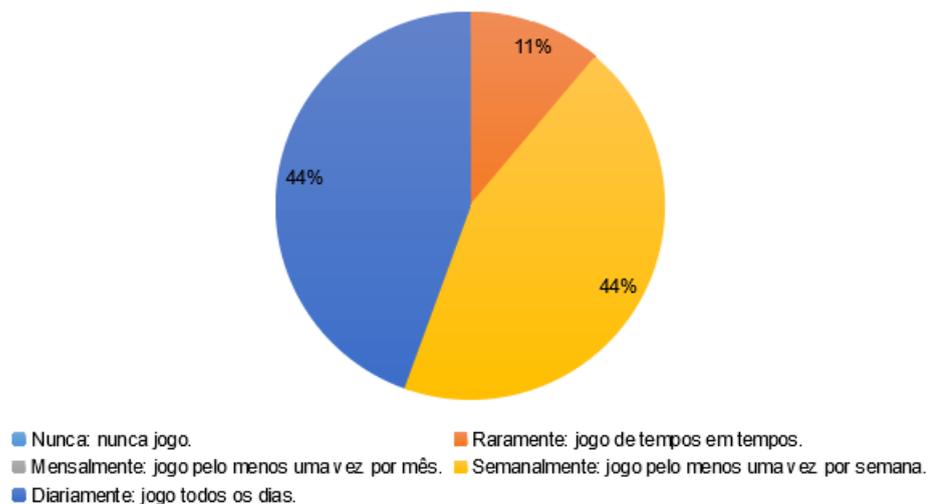


Figura 4.2: Gráfico relativo à quem joga jogos digitais

E relativo aos que jogam jogos não digitais, ou seja, jogos de cartas, tabuleiro, entre outros:

Com que frequência você costuma jogar jogos não-digitais (de cartas, tabuleiro, etc)?

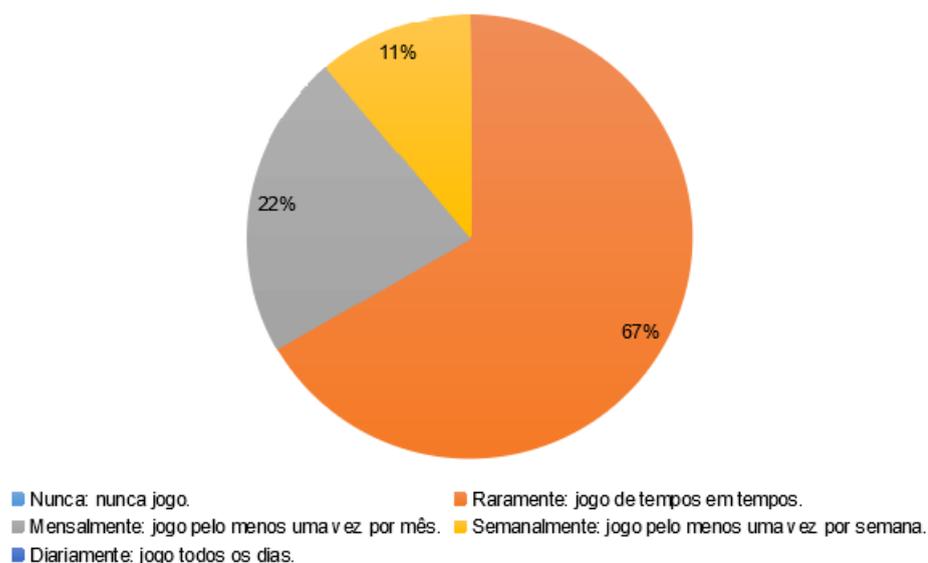


Figura 4.3: Gráfico relativo à quem joga jogos não digitais

Com essas informações, conseguimos verificar que os alunos tem mais interesse em jogos digitais do que em não digitais, nos mostrando a ideia de ter uma versão do ProcSoft (MOURA, 2014) digital seria algo mais interessante para todos os alunos da area da Ciência da Computação.

4.3 Avaliação do Jogo

Após aplicado o questionário, foram feitos gráficos para analisar cada resposta e possíveis melhorias para o jogo quanto a Usabilidade, Confiança, Desafio, Satisfação, Diversão, Atenção focada, Relevância e Percepção da Aprendizagem.

Em termos de usabilidade do jogo, tivemos as seguintes respostas:

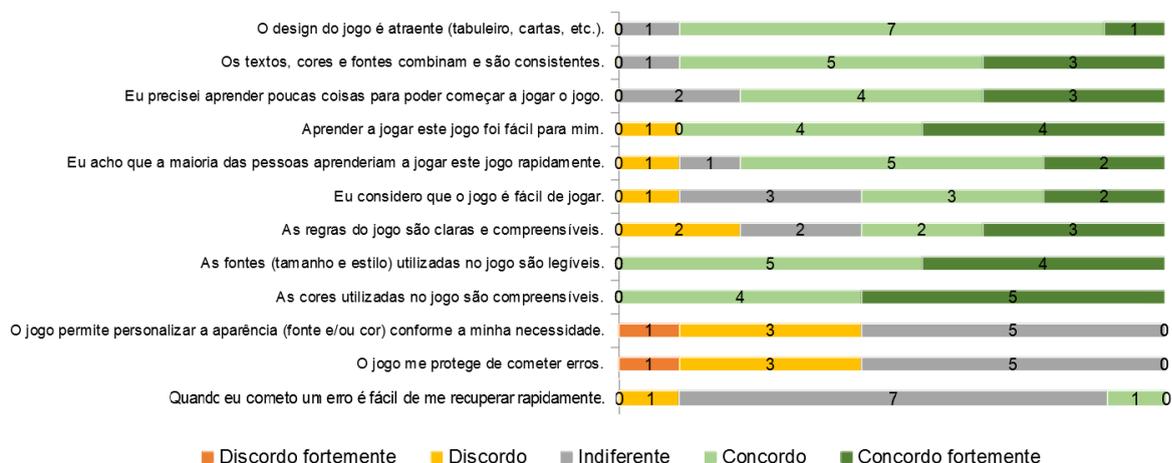


Figura 4.4: Gráfico relacionado a usabilidade do jogo

Quando falamos sobre a usabilidade do jogo, podemos ver que os alunos no geral consideraram o jogo bonito em termos de design, com suas fontes e cores combinando. A figura 4.4 também nos mostra que 8 dos 9 alunos acharam o jogo fácil de compreender e sem a necessidade de um conhecimento básico sobre jogos para jogá-lo, porém um pouco confuso quanto as regras, e limitado quanto à prevenção de erros, afinal, para que o jogo aconteça é necessário ter um “Banco“, assim como no Banco Imobiliário, que faz todo o gerenciamento do dinheiro dos times, e do jogo em geral, cuidando de jogar os dados, fazer as compras dos papéis e ferramentas, junto com todas as funcionalidades mostradas acima.

Quanto à confiança dos jogadores ao jogarem:

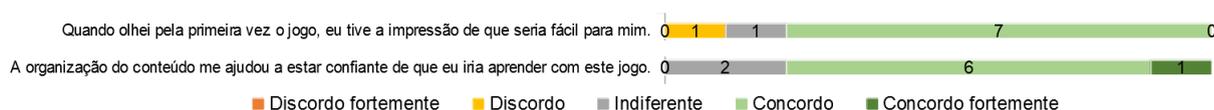


Figura 4.5: Gráfico relacionado à confiança do jogo

Como os participantes já conheciam o jogo Banco Imobiliário, quando foi dito que jogo

era similar ao mesmo, e que foi baseado nele, os alunos se sentiram mais confiantes, afinal, a ideia da jogabilidade com tabuleiro, compras de atividades ao invés de propriedades, e também o sorte/revés que já existem no Banco Imobiliário os deixou com a ideia de jogar algo que já estão acostumados e que já dominam parecia ser fácil.

Quanto ao Desafio do jogo:

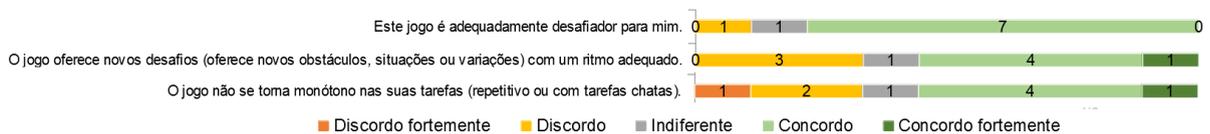


Figura 4.6: Gráfico relacionado ao Desafio do jogo

O jogo foi considerado desafiador por 7 dos 9 alunos, acreditando que a ideia de responder perguntas novas a cada rodada, a ideia de compra de atividades seria desafiadora, devendo sempre estar relebrando conteúdos da engenharia de software e aprendendo novos com as atividades propostas pela norma ISO/IEC 29110. Contudo, 33% dos alunos acreditam que os desafios não evoluem, continuam sempre os mesmos, no caso, os citados acima. Tivemos também 33% que consideraram as atividades propostas pelo jogo monótonas ou repetitivas, nos levando a repensar algumas atividades, apesar do pouco tempo de jogo.

Quanto à Satisfação do jogo:

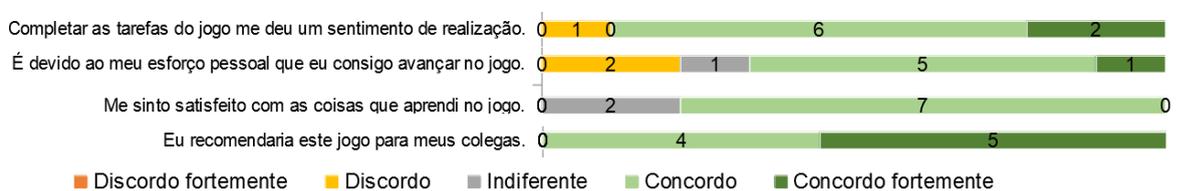


Figura 4.7: Gráfico relacionado à Satisfação do jogo

Dois dos 9 alunos consideraram que não é por esforço próprio que se consegue evoluir no jogo e somente por sorte, entretanto, os demais pensaram o contrário, nos mostrando que eles acreditam que por exemplo, ao responder uma pergunta e pagar mais barato em uma atividade os auxiliará a evoluir no jogo. Os participantes gostaram do jogo e todos responderam que possivelmente o indicariam para outros colegas.

Quanto à Interação Social do jogo:



Figura 4.8: Gráfico relacionado à Interação Social do jogo

Um ponto abordado foi sobre a interação com os outros alunos, nesse ponto tivemos muitas respostas positivas. O jogo promoveu momentos de cooperação entre sua equipe, algo considerado importante, pois dificilmente trabalharemos sozinhos durante um processo de software.

Tratando da Diversão do jogo:

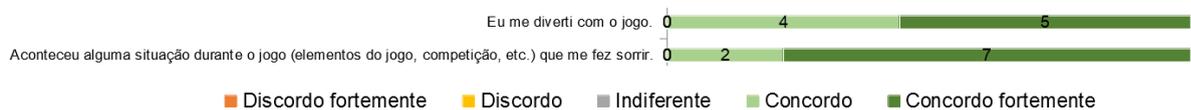


Figura 4.9: Gráfico relacionado à Diversão do jogo

No geral, um jogo deve ser divertido, as pessoas devem se sentir alegres jogando, e nesse ponto o jogo cumpriu completamente seu papel. Como pode ser visto na figura 4.9 acima, todos os participantes riram e se divertiram durante o andamento do jogo.

Sobre à Atenção Focada no jogo:

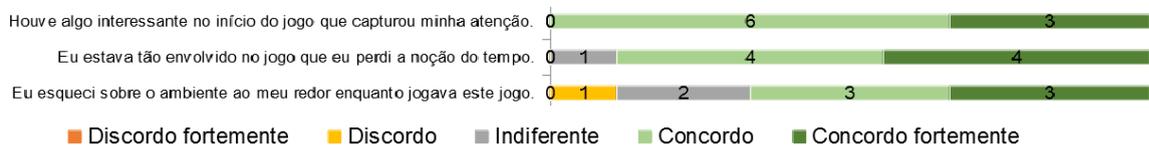


Figura 4.10: Gráfico relacionado à Atenção Focada no jogo

O gráfico da figura 4.10, nos mostra que o jogo cativou a atenção dos participantes, levando-os a prestar atenção apenas no jogo e no que estavam fazendo, esquecendo do tempo e do ambiente externo, ou seja, a partir do momento que o jogo se iniciou, os alunos esqueceram seus problemas, e focaram apenas no que estavam fazendo naquele momento.

Quanto à Relevância do jogo:



Figura 4.11: Gráfico relacionado à Relevância do jogo

Quanto a relevância do jogo, 2 dos 9 alunos afirmaram que não se interessavam pelo conteúdo do jogo, levando a acreditar que os mesmos não se interessam pelo conteúdo da engenharia de software e processos de software. Apesar disso, 7 dos 9 afirmaram que preferem aprender o conteúdo utilizando o jogo do que com outra forma de ensino. Os outros 2 participantes ficaram neutros em suas respostas.

Quanto à Percepção da aprendizagem do jogo:

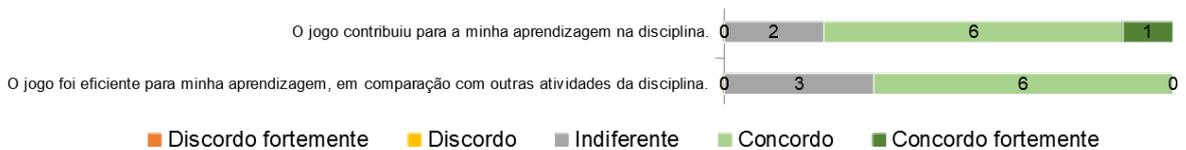


Figura 4.12: Gráfico Percepção da aprendizagem do jogo

Quanto a aprendizagem através do jogo, 7 dos 9 participantes disseram que o jogo os auxiliou no aprendizado do conteúdo. Os demais ficaram neutros sobre se realmente o jogo foi eficiente para o aprendizado dos alunos.

4.4 Questões abertas

Além das questões de assinalar, foram feitas 3 perguntas abertas, para que os participantes dessem sua opinião sobre o jogo. Abaixo seguem as questões e as respostas mais pertinentes, junto da análise de cada uma:

“Cite 3 pontos fortes do jogo:“

- Aprendizagem de forma interativa;
- O jogo foi competitivo;

- Foi divertido jogá-lo;
- Bom teste para avaliar os conhecimentos da disciplina;
- Design simples e bonito;
- Perguntas desafiantes.

As principais respostas foram: Aprendizagem de forma interativa e a competitividade do jogo. Isso nos mostra que a competição entre os jovens os motiva a melhorar, e junto disso, a aprendizagem sendo comentada e discutida pelos alunos também os ajudou.

“Dê 3 sugestões para a melhoria do jogo:“

- Automatização dos processos no jogo;
- Mais clareza em algumas regras;
- Adicionar animações;
- Mostrar no tabuleiro quais casas foram compradas;
- Adicionar sons;
- Melhorar a visualização.

Mais da metade dos participantes citou a automatização dos processos, ou seja, o fato de ter alguém como gerente durante o jogo não foi muito agradável, e passiva de erros, então o melhor seria automatizar o jogo, como por exemplo, tirar o botão de gerenciar dinheiro e ao invés de colocar lá o valor, apenas clicar em comprar, e o dinheiro descontaria automaticamente do dinheiro do seu time. Outros pontos pertinentes foram os de mais clareza em algumas regras, e a adição de animações, como por exemplo, um dado 3D na tela.

“Comentários adicionais:“

- Muito bom! Gostei;
- Jogo muito bom, porém, tive azar;
- Ri muito durante a atividade, foi divertido.

Nos comentários, era pra ser dito algo que não foi perguntado, e muitos deixaram em branco, e os mais pertinentes foram os citados acima, elogiando o jogo ou dizendo que foi divertido, porém, alguns não contaram com a sorte.

Capítulo 5

Resultados e Discussões

Neste trabalho de conclusão de curso foi desenvolvido um protótipo funcional do ProcSoft (MOURA, 2014), chamado de ProcSoftDigital (BIZETTO, 2019), com o objetivo de contribuir para um melhor aprendizado dos conceitos de Processo de Software de uma forma lúdica, bem como lembrar alguns conceitos da Engenharia de Software, apresentando detalhes sobre seu desenvolvimento e conclusões obtidas na aplicação do jogo. Por fim, são propostos alguns trabalhos futuros relacionados com o tema.

5.1 Conclusões

A avaliação da versão inicial do jogo mostrou que os alunos foram receptivos ao mesmo. Por meio dos questionários foi possível perceber que a experiência foi positiva, afinal, é uma forma dinâmica de se aprender processos de software e como cada atividade é importante para o desenvolvimento de um software, seja ela a atividade de um gerente, desenvolvedor, testador ou analista e que com as melhorias adequadas o jogo pode se tornar algo útil para o ensino.

Cabe destacar desafios relacionados ao desenvolvimento do jogo. Algumas limitações foram causadas pela escolha da linguagem. Acredita-se que com uma ferramenta ou linguagem específica para o desenvolvimento de jogos, sua implementação possa ser mais rápida e deixe o jogo mais atrativo, com animações, sons, entre outras coisas que existem em jogos digitais.

Cabe destacar também que as perguntas ligadas à Engenharia de Software utilizadas devem sempre ser estudadas e atualizadas, juntamente com as ferramentas utilizadas dentro do jogo, para que o mesmo sempre se mantenha em processo de renovação.

5.2 Trabalhos Futuros

- Automatização dos processos no jogo, tentando evitar ao máximo que exista uma pessoa para gerenciar o mesmo;
- Criação de uma versão online, onde os jogadores possam jogar cada um em uma máquina, seja cada um de sua casa com seus amigos, ou durante aulas no laboratório;
- Acrescentar animações no jogo, para que o mesmo seja mais dinâmico para quem estiver jogando;
- Criar um sistema para adicionar questões, e removê-las, com a utilização de um Banco de Dados e conectado com o jogo.

Apêndice A

Questionário

A.1 Questões sobre os participantes

Informações Demográficas	
Instituição:	
Curso:	
Disciplina:	
Faixa etária:	<input type="checkbox"/> Menos de 18 anos <input type="checkbox"/> 18 a 28 anos <input type="checkbox"/> 29 a 39 anos <input type="checkbox"/> 40 a 50 anos <input type="checkbox"/> Mais de 50 anos
Gênero:	<input type="checkbox"/> Masculino <input type="checkbox"/> Feminino
Com que frequência você costuma jogar jogos digitais?	<input type="checkbox"/> Nunca: nunca jogo. <input type="checkbox"/> Raramente: jogo de tempos em tempos. <input type="checkbox"/> Mensalmente: jogo pelo menos uma vez por mês. <input type="checkbox"/> Semanalmente: jogo pelo menos uma vez por semana. <input type="checkbox"/> Diariamente: jogo todos os dias.
Com que frequência você costuma jogar jogos não-digitais (de cartas, tabuleiro, etc)?	<input type="checkbox"/> Nunca: nunca jogo. <input type="checkbox"/> Raramente: jogo de tempos em tempos. <input type="checkbox"/> Mensalmente: jogo pelo menos uma vez por mês. <input type="checkbox"/> Semanalmente: jogo pelo menos uma vez por semana. <input type="checkbox"/> Diariamente: jogo todos os dias.

Figura A.1: Perguntas sobre os participantes

A.2 Questões sobre o jogo

A.2.1 Questões sobre Usabilidade

Experiência do Jogador					
Afirmações	Marque uma opção conforme sua avaliação				
	Discordo Totalmente	Discordo	Nem discordo, nem concordo	Concordo	Concordo Totalmente
O design do jogo é atraente (interface, gráficos, etc.).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Os textos, cores e fontes combinam e são consistentes.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu precisei aprender poucas coisas para poder começar a jogar o jogo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aprender a jogar este jogo foi fácil para mim.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eu considero que o jogo é fácil de jogar.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
As regras do jogo são claras e compreensíveis.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
As fontes (tamanho e estilo) utilizadas no jogo são legíveis.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
As cores utilizadas no jogo são compreensíveis.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
O jogo permite personalizar a aparência (fonte e/ou cor) conforme a minha necessidade.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
O jogo me protege de cometer erros.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quando eu cometo um erro é fácil de me recuperar rapidamente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura A.2: Perguntas sobre Usabilidade

A.2.2 Questões sobre Confiança

Quando olhei pela primeira vez o jogo, eu tive a impressão de que seria fácil para mim.	<input type="checkbox"/>				
A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo.	<input type="checkbox"/>				

Figura A.3: Perguntas sobre Confiança

A.2.3 Questões sobre Desafio

Este jogo é adequadamente desafiador para mim.	<input type="checkbox"/>				
O jogo oferece novos desafios (oferece novos obstáculos, situações ou variações) com um ritmo adequado.	<input type="checkbox"/>				
O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas).	<input type="checkbox"/>				

Figura A.4: Perguntas sobre Desafio

A.2.4 Questões sobre Satisfação

Completar as tarefas do jogo me deu um sentimento de realização.	<input type="checkbox"/>				
É devido ao meu esforço pessoal que eu consigo avançar no jogo.	<input type="checkbox"/>				
Me sinto satisfeito com as coisas que aprendi no jogo.	<input type="checkbox"/>				
Eu recomendaria este jogo para meus colegas.	<input type="checkbox"/>				

Figura A.5: Perguntas sobre Satisfação

A.2.5 Questões sobre Interação Social

Eu pude interagir com outras pessoas durante o jogo.	<input type="checkbox"/>				
O jogo promove momentos de cooperação e/ou competição entre os jogadores.	<input type="checkbox"/>				
Eu me senti bem interagindo com outras pessoas durante o jogo.	<input type="checkbox"/>				

Figura A.6: Perguntas sobre Interação Social

A.2.6 Questões sobre Diversão

Eu me diverti com o jogo.	<input type="checkbox"/>				
Aconteceu alguma situação durante o jogo (elementos do jogo, competição, etc.) que me fez sorrir	<input type="checkbox"/>				

Figura A.7: Perguntas sobre Diversão

A.2.7 Questões sobre Atenção Focada

Houve algo interessante no início do jogo que capturou minha atenção.	<input type="checkbox"/>				
Eu estava tão envolvido no jogo que eu perdi a noção do tempo.	<input type="checkbox"/>				
Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.	<input type="checkbox"/>				

Figura A.8: Perguntas sobre Atenção Focada

A.2.8 Questões sobre Relevância

O conteúdo do jogo é relevante para os meus interesses.	<input type="checkbox"/>				
É claro para mim como o conteúdo do jogo está relacionado com a disciplina.	<input type="checkbox"/>				
O jogo é um método de ensino adequado para esta disciplina.	<input type="checkbox"/>				
Eu prefiro aprender com este jogo do que de outra forma (outro método de ensino).	<input type="checkbox"/>				

Figura A.9: Perguntas sobre Relevância

A.3 Questões sobre Aprendizagem

Percepção da Aprendizagem					
Afirmações	Marque uma opção conforme sua avaliação				
	Discordo Totalmente	Discordo	Nem discordo, nem concordo	Concordo	Concordo Fortemente
O jogo contribuiu para a minha aprendizagem na disciplina.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura A.10: Perguntas sobre Aprendizagem

A.3.1 Questões Abertas

Cite 3 pontos fortes do jogo: _____

Dê 3 sugestões para a melhoria do jogo: _____

Comentários adicionais: _____

Figura A.11: Perguntas Abertas

Referências Bibliográficas

ABNT. *GUIA DE IMPLEMENTAÇÃO: DESENVOLVIMENTO DE SOFTWARE PARA PEQUENAS ORGANIZAÇÕES*. [S.l.]: ABTN, SEBRAE, 2012.

AMBROSIO, F. K. *SE•RPG 2.0: Uma nova versão do software*. RIO DE JANEIRO: [s.n.], 2008.

BIZETTO, B. P. *ProcSoftDigital*. 2019. Consultado na INTERNET: <https://github.com/brendoperes1/ProcSoftDigital>.

FALBO, R. de A. *Engenharia de Software: Notas de Aula*. Espírito Santo: [s.n.], 2014.

FARDO, M. L. A gamificação aplicada em ambientes de aprendizagem. *CINTED-UFRGS*, 2013.

FIGUEIREDO, E. M. L. et al. *SimulES: Um Jogo para o Ensino de Engenharia de Software*. RIO DE JANEIRO: [s.n.], 2006.

IEEE Standard Glossary of Software Engineering Terminology. 1990.

MASLAĆ, K. A.; RAŠIĆ, M.; VRANESIC, P. Influence of gamification on student motivation in the educational process in courses of different fields. *Econtro Nacional de Pesquisa em Educação em Ciências*, 2018.

MOURA, V. *Jogo para ensino de processo de software*. Rio de Janeiro: [s.n.], 2014.

MOZILLA. *JavaScript*. 2019. Consultado na INTERNET: <https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript>.

NAUR, P.; RANDELL, B. *Nato Science Committee*. 1968.

NOGUEIRA, M. O.; PEREIRA, L. de S. As empresas de pequeno porte no brasil: Os dilemas da produtividade e da informalidade. 2018.

PEREIRA., R. F.; FUSINATO., P. A.; NEVES, M. C. D. Desenvolvendo um jogo de tabuleiro para o ensino de física. *Econtro Nacional de Pesquisa em Educação em Ciências*, 2009.

PRENSKY, M. *Digital Game-Based Learning*. [S.l.]: McGraw-Hill, 2001.

PRESSMAN, R. S. *Software Engineering: A practitioner's approach*. 9. ed. New York: McGraw-Hill, 2010.

QUASARDEV. *Quasar*. 2019. Consultado na INTERNET: <https://quasar.dev/introduction-to-quasar>.

SANTOS, D. S. dos. *O Papel dos jogos de tabuleiro na aprendizagem*. 2019. Consultado na INTERNET: <https://meuartigo.brasilecola.uol.com.br/pedagogia/o-papel-dos-jogos-tabuleiro-na-aprendizagem.htm>.

SHULL, F.; CARVER, J.; TRAVASSOS, G. H. An empirical methodology for introducing software processes. *SIGSOFT Softw. Eng. Notes*, ACM, New York, NY, USA, v. 26, n. 5, p. 288–296, set. 2001. ISSN 0163-5948. Disponível em: <<http://doi.acm.org/10.1145/503271.503248>>.

SILVA, A. C. *Jogo educacional para apoiar o ensino de técnicas para elaboração de testes de unidade*. São José: [s.n.], 2010.

SOMMERVILLE, I. *SOFTWARE ENGINEERING*. 9. ed. United States of America: Pearson Education, Inc, 2011.

ZICHERMANN, G.; CUNNINGHAM, C. *Gamification by Design*. [S.l.]: O'Reilly Media, Inc, 2011.