



UNICESUMAR – UNIVERSIDADE CESUMAR
CENTRO DE CIÊNCIAS EXATAS TECNOLÓGICAS E AGRÁRIAS
CURSO DE GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DATA WATCHERS

FELIPE WESLEY BASSO
LEONARDO PIMENTEL LOPES FILHO

CURITIBA – PR
2020

FELIPE WESLEY BASSO
LEONARDO PIMENTEL LOPES FILHO

DATA WATCHERS

Trabalho apresentado ao Curso de Graduação em Análise e Desenvolvimento de Sistemas da UNICESUMAR – Universidade Cesumar como requisito parcial para avaliação da Atividade de Estudo Programada (AEP) do 1º semestre de 2020.

CURITIBA – PR

2020

DATA WATCHERS

Felipe Wesley Basso

Leonardo Pimentel Lopes Filho

RESUMO

Data Watchers é um jogo de point click, onde o jogador é envolvido em uma trama e certamente não conseguirá sair sem uma grande dose de raciocínio e, também, sem utilizar da ajuda de algum outro mecanismo de pesquisa. Porém, apesar de se tratar de um jogo baseado no formato de cliques, o fundamental deste jogo não está amarrado apenas à capacidade de o jogador saber clicar com eficiência, mas sim à sua capacidade de desvendar enigmas e, assim, negociar as informações descobertas de acordo com seus respectivos objetivos dentro do jogo.

Palavras-chave: Cliques. Enigmas. Negociação.

SUMÁRIO

1 INTRODUÇÃO	4
2 DESENVOLVIMENTO.....	5
2.1 O QUE É A LEI GERAL DE PROTEÇÃO DE DADOS PESSOAIS	5
2.2 IMPACTO DA LGPD NO CONTEXTO MULTIDISCIPLINAR	6
2.3 DESAFIO PROPOSTO.....	8
2.4 PROPOSTA DE JOGO APRESENTADA	8
2.5 MANUAL DO JOGO.....	8
2.5.1 A HISTÓRIA	8
2.5.2 O JOGO.....	9
2.6 PROJETO DO JOGO	14
2.6.1 PARADIGMA DE ANÁLISE E DESENVOLVIMENTO	14
2.6.2 TECNOLOGIAS UTILIZADAS	15
2.6.3 CICLO DE VIDA E PROCESSO DE SOFTWARE.....	16
3 CONCLUSÃO.....	21
REFERÊNCIAS	22

1 INTRODUÇÃO

Tendo como ponto de partida o desafio de desenvolver um projeto acerca da nova Lei Geral de Proteção de Dados Pessoais, o presente artefato de jogo produzido traz o cenário pelo qual o Brasil irá passar dentro de alguns meses, quando a lei entrar em vigor. As metodologias utilizadas para o desenvolvimento do projeto, desde sua concepção até o lançamento de sua versão finalizada, estão apresentadas na seção de desenvolvimento deste documento, fundamentados em conteúdo teórico devidamente referenciados e ilustrados com diagramas e modelos conceituais.

1.1 OBJETIVO GERAL

Construir um jogo baseado na plataforma Java que apresente conceitos acerca da Lei Geral de Proteção de Dados Pessoais e, também, que apresente seu impacto no contexto multidisciplinar.

1.2 OBJETIVOS ESPECÍFICOS

Descrição dos objetivos específicos do trabalho. Desenvolver um jogo na linguagem de programação Java, utilizando de conceitos e plataformas disponibilizadas pela própria linguagem, tais conceitos como abstração, polimorfismo, encapsulamento e herança, conceitos esses que são os pilares fundamentais do paradigma de orientação a objetos, implementado de maneira sólida na linguagem Java. A aplicação deverá utilizar conexão com um banco de dados relacional para armazenar dados relevantes a respeito do jogo. Antes da entrega final do projeto, assim como durante todo o ciclo de vida de desenvolvimento do software, o mesmo deverá ser submetido a etapas de teste de validação e verificação, utilizando práticas fundamentadas na Engenharia de Software. Afim de manter a performance do jogo, deverão ser estudadas e implementadas estruturas de dados adequadas, que auxiliem na manutenção de sua qualidade e desempenho. O jogo, bem como sua fundamentação teórica, deverá apresentar conceitos sobre a nova Lei Geral de Proteção de Dados Pessoais, os quais possam facilmente ser interpretados pelo jogador e, também, que sejam capazes de transmitir conhecimento sobre alguns fundamentos da LGPD aos *gamers*, que, num futuro próximo, serão direta e indiretamente afetados.

2 DESENVOLVIMENTO

As etapas que compreendem o desenvolvimento do presente trabalho estão descritas nesta seção, divididas em subseções, que diferem entre si de acordo com os tópicos propostos para a elaboração deste projeto.

2.1 O QUE É A LEI GERAL DE PROTEÇÃO DE DADOS PESSOAIS

A respeito da Lei Geral de Proteção de Dados Pessoais, conforme descrito no Artigo 1º da lei nº 13.709, de 14 agosto de 2018:

Esta Lei dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural. (Brasil, 2018).

A partir disso, entende-se que esta lei abrange toda e qualquer operação se caracteriza tratamento de dados pessoais (dentro dos termos que a lei dispõe), seja em meio físico seja em meio digital, por qualquer pessoa, física ou jurídica, cujo direito seja público ou privado.

Segundo a mesma lei, Artigo 5º primeiro parágrafo, um dado pessoal pode ser definido como “informação relacionada a pessoa natural identificada ou identificável”. (Brasil, 2020). Ainda relacionado às considerações da lei, o Artigo 5º, no inciso VI, redige que um ator controlador, dentro do contexto atualmente apresentado, pode ser “pessoa natural ou jurídica, de direito público ou privado, a quem competem as decisões referentes ao tratamento de dados pessoais”. (Brasil, 2020).

Dentro dos termos acima apresentados, e também àqueles trazidos no projeto de lei citado, toda operação realizada com dados pessoais, cujo titular ou controlador tenham direitos ou qualquer outro tipo de vínculo com o território nacional, serão submetidos à aplicação da Lei Geral de Proteção de Dados Pessoais.

2.2 IMPACTO DA LGPD NO CONTEXTO MULTIDISCIPLINAR

Quanto ao contexto multidisciplinar, no que se refere ao curso de análise e desenvolvimento de sistemas da Universidade Cesumar, cabe, porém não somente a este, mas também a todos os cursos afins da área de tecnologia da informação, aprimorar e desenvolver técnicas capazes de garantir a aplicação da LGPD no contexto oportuno.

Como disposto em relação às competências e responsabilidades no que se refere a gestão de segurança da informação, Ferreira, Cabral e Sonnenstrahl (2013) comentam em seu projeto, sobre conceitos e fundamentos para o sucesso da implementação de políticas de segurança da informação dentro da organização, no caso o Instituto Federal de Educação, Ciência e Tecnologia Farroupilha, que, quanto a gestão do instituto citado, o comitê responsável pela Gestão de Segurança da Informação deverá ser integrado com o Comitê de Tecnologia da Informação. Tendo isso em vista, pode-se considerar de extrema importância a presença efetiva do profissional de T.I., devidamente capacitado em relação a Segurança da Informação, uma vez que a equipe de tecnologia da informação deverá atuar nos diversos setores da organização, afim de garantir a aplicação adequada da Lei Geral da Proteção de Dados Pessoais.

Como exemplo disso, pode-se citar diversos cenários que caem em responsabilidade do profissional de informática, tais como alguns listados a seguir:

- A presente lei enfatiza o que pode ser considerado um dado pessoal: é todo dado através do qual é possível identificar uma pessoa natural (Art. 5º § I). Porém, os incisos III e XI do mesmo Artigo discorrem sobre dados anonimizados. Segundo essa referência, caso o dado pessoal necessite ser mantido pela empresa ou instituição de sua posse, o mesmo pode ser submetido a um processo que o torne anônimo, acabando com a possibilidade de relacioná-lo ao seu titular de origem. Através deste cenário é possível observar onde a atuação de um profissional de informática se faz necessária.
- Segundo o Artigo 6º da lei correspondente à LGPD, as atividades de tratamento de dados deverão observar não somente a boa-fé, mas também a alguns princípios, listados em 10 parágrafos no presente artigo. Dentre estes parágrafos, há cenários que garantem ao titular dos dados a consulta facilitada e gratuita aos processos de tratamento (inciso IV e Capítulo III – Dos direitos

do titular). A transparência (§ VI) e segurança (§ VII) dos dados pessoais tratados também são amparados dentre os parágrafos deste artigo. Sendo assim, a atuação do profissional de informática se faz não somente necessária, mas extremamente importante, onde o mesmo, devidamente capacitado, poderá e deverá utilizar:

- ✓ de estruturas de dados que melhores se adequem ao cenário exposto;
 - ✓ de códigos que criptografem determinados dados, de acordo com a necessidade;
 - ✓ de padrões de desenvolvimento que garantem a segurança e integridade das informações tratadas em meio à própria equipe de T.I.;
 - ✓ de controle de acesso do público à dados pessoais, restringindo somente ao necessário;
 - ✓ de facilitar o acesso do titular aos seus dados e aos processos de tratamento aos quais os mesmos são submetidos dentro da organização, entre outros diversos cenários.
- No que se refere a desenvolvimento de software, muitos produtos necessitam de termos e condições de uso, os quais o usuário deve aceitar para poder utilizar o produto. O Capítulo II da LGPD discorre sobre o tratamento de dados pessoais. Na Seção I, que compreende os Artigos 7º, 8º, 9º e 10º, o inciso um (§ 1) do Artigo 8º apresenta a necessidade de que o consentimento do titular dos dados para permitir o tratamento dos mesmos seja expressado de forma clara e objetiva, em cláusula destacada das demais cláusulas contratuais, onde seja expressada a manifestação da vontade do titular de ter os seus dados submetidos a atividades de tratamento, bem como todas as operações que a organização poderá realizar com os dados fornecidos.

Sendo assim, os cenários de exemplo apresentados enfatizam o impacto da Lei Geral de Proteção de Dados Pessoais no contexto multidisciplinar, tanto em um ambiente de aprendizado quanto em um ambiente profissional, onde muitos estudantes já estão inseridos. Neste contexto, o estudante e profissional de tecnologia da informação se faz de extrema importância para garantir a aplicação e manutenção da LGPD no meio digital.

2.3 DESAFIO PROPOSTO

A partir de agosto de 2020 a Lei Geral de Proteção de Dados Pessoais (LGPD ou LGPD) entrará em vigor e muitas empresas e profissionais não possuem o conhecimento necessário para se adequar a essa nova realidade. O desafio será criar uma proposta lúdica (jogo digital) que permita a disseminação dos conhecimentos referentes a LGPD para os diferentes perfis profissionais.

2.4 PROPOSTA DE JOGO APRESENTADA

O jogo que se faz objeto deste trabalho apresenta uma proposta onde o jogador é encorajado a agir de maneira ilegal no que diz respeito à manipulação de dados pessoais, contrariando as indicações da Lei Geral de Proteção de Dados Pessoais, lei esta que está prestes a entrar em vigor no cenário do jogo. Dessa maneira, o jogo desempenha o papel de transmitir conhecimento da nova lei para o jogador.

O jogador passa a observar que dados pessoais não mais poderão, após a vigência da LGPD, receber o mesmo tratamento que recebia há um tempo atrás, pois tudo deverá acontecer conforme a lei redige, atendendo todos os direitos e deveres.

2.5 MANUAL DO JOGO

2.5.1 A HISTÓRIA

Data Watchers se passa em um futuro não muito distante, onde a Lei Geral de Proteção de Dados Pessoais está a horas de ser completamente aplicada no mundo digital. O jogador irá jogar como Thiago, um jovem de 27 anos que trabalha em uma parte do HandBook, aonde seu emprego é comprar ilegalmente dados de determinadas pessoas. Porém, com a LGPD em vigor, seu trabalho está prestes a acabar. Com isso, Thiago é demitido do HandBook e vai para casa. Ao chegar em sua residência, faz coisas normais, que qualquer desempregado faz quando chega em casa: vai para o computador. Quando ele entra em seu e-mail, vê uma mensagem estranha a qual indica o desejo de alguém em comprar dados de

algumas pessoas bastante específicas, sem que ninguém saiba. Thiago, com medo, faz este favor para o desconhecido.

E este é o papel do jogador em Data Watchers: comprar os dados dos usuários específicos solicitados, decodificar enigmas e, ao terminar sua missão, consegue sair com a consciência limpa e sem medo desse desconhecido, ciente da nova lei de proteção de dados pessoais que entrara em vigor.

2.5.2 O JOGO

Antes de tudo, é necessário que o jogador tenha um login e uma senha, para que possa, de fato, acessar o jogo. Caso o jogador ainda não tenha cadastro, o mesmo poderá fazê-lo na própria tela de login na opção de se cadastrar. Estas instruções estão ilustradas nas Fotos 1 e 2 apresentada a seguir. Após o cadastro, basta fazer o login.

Foto 1 – Tela de login

A imagem mostra a interface de login do jogo DataWatchers. O título "DataWatchers" está em uma fonte verde grande e pixelada no topo. Abaixo dele, o texto "Digite os dados abaixo para logar" aparece em uma fonte vermelha menor. Há dois campos de entrada brancos: o primeiro é rotulado "Usuario:" e o segundo "Senha:", ambos em verde. Abaixo dos campos, há um botão cinza com o texto "Login". No canto inferior direito, há o texto "Não tem login entao :" em verde, seguido por um botão cinza com o texto "Cadastre-se".

DataWatchers

Digite os dados abaixo para logar

Usuario:

Senha:

Login

Não tem login entao :

Fonte: Produção autoral.

Foto 2 – Tela de cadastro

Fonte: Produção autoral.

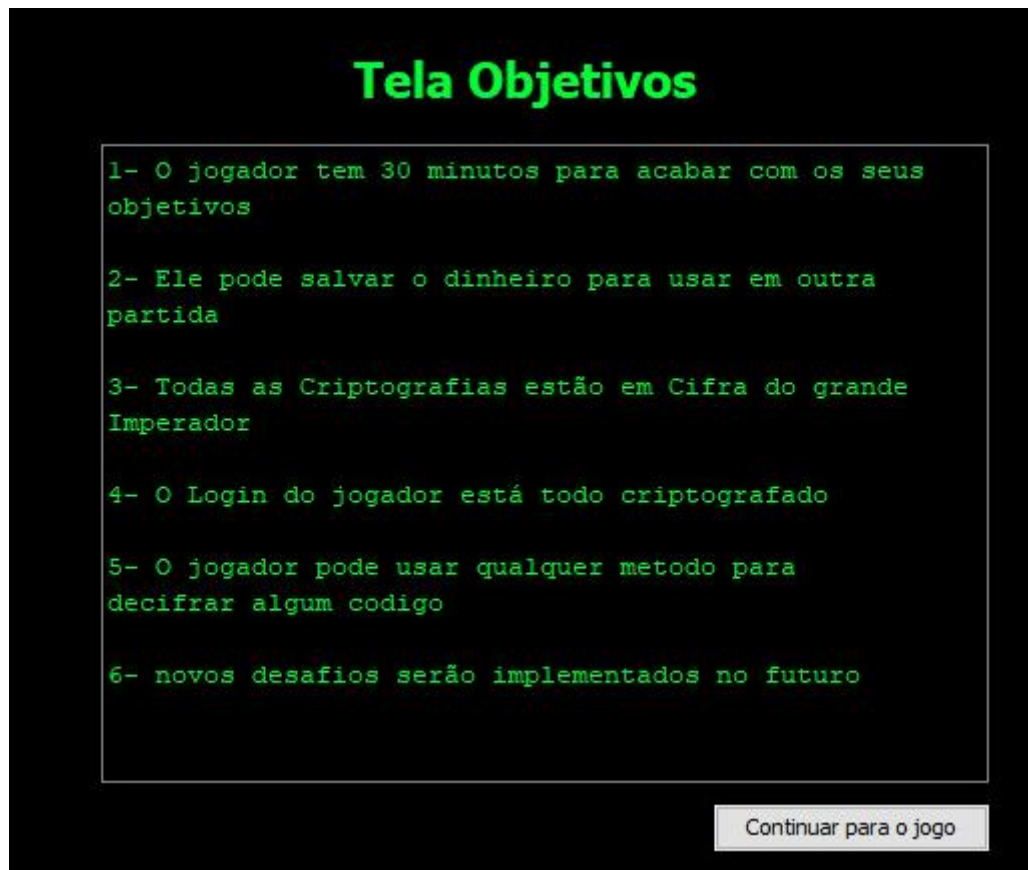
Com a etapa de login concluída, o jogador terá uma breve introdução à história do jogo, onde há um quadro contextualizando o cenário de jogo. Para que o jogo se inicie, basta clicar no botão de novo jogo (“New Game”), conforme apresenta a Foto 3.

Foto 3 – Iniciando um novo jogo

Fonte: Produção autoral.

Antes que o jogo efetivamente seja iniciado, o usuário terá em tela os objetivos do jogo e algumas informações a respeito da manipulação de informações dentro da plataforma do jogo.

Foto 4 – Objetivos do jogo



Fonte: Produção autoral.

A tela principal do jogo, apresentada a seguir na Foto 5, contém alguns botões. O usuário, no início do jogo, terá disponível apenas os seguintes botões:

- +1 BitCoin;
- Comprar Dados;
- Ativar o Click Duplo (que custa 100 BitCoins).

Toda a economia do jogo gira em torno de BitCoins, que o jogador consegue clicando nos botões em áreas específicas da tela. Quando o jogador consegue atingir determinado valor, o dado anônimo de alguém se torna disponível para ser comprado.

O jogo tem um limite de 30 minutos para ser finalizado. Caso o tempo seja excedido, será indicado fim de jogo e a aplicação será encerrada, encerrando também todo o progresso do jogador. Com isso, o desafio é que o jogador consiga obter as quantias o mais rápido possível, para que o tempo não seja excedido.

Foto 5 – Tela principal do jogo



Fonte: Produção autoral.

Do lado direito da figura acima, pode-se observar a tela onde o jogador conseguirá obter BitCoins e, do lado, esquerdo, estão disponíveis os dados para serem comprados.

Na tela apresentada a seguir, um enigma codificado será exibido ao jogador, onde, com a ajuda fornecida pelo jogo e por outros mecanismos de pesquisa, será possível decodificar o enigma apresentado colocando no campo de resposta o resultado obtido. Se o jogador decodificar corretamente a informação, ele poderá tomar posse da mesma, que ficará disponível na tela de dados comprados.

Foto 6 – Tela de decodificação de enigma



Fonte: Produção autoral.


O jogo termina bem sucedido quando o jogador consegue atingir os objetivos apresentados no início e na história do jogo, que é comprar os dados solicitados pelo desconhecido dentro do prazo estipulado de 30 minutos.

Foto 7 – Fim de jogo



Fonte: Produção autoral.

2.6 PROJETO DO JOGO

Este tópico apresenta a interface teórica por trás do desenvolvimento do presente jogo. Este jogo também pode ser encontrado no GitHub. Está armazenado em um repositório público no seguinte link:  github.com/LeonardoLopesFilho/Projeto-DataWatchers

2.6.1 PARADIGMA DE ANÁLISE E DESENVOLVIMENTO

O jogo Data Watchers foi projetado e desenvolvido sobre o paradigma de orientação a objetos. Bezerra (2015) descreve que o paradigma de orientação a objetos consiste em uma abordagem cuja qual visualiza um sistema de software como sendo uma coleção de elementos, onde cada elemento pertencente está interconectado com os demais, elementos estes denominados objetos.

Braga e Masiero (2007) comentam que o paradigma de orientação a objetos oferece diversos modelos através dos quais é possível manter o máximo de fidelidade entre objetos presentes no mundo real e elementos dentro do sistema de software, elementos esses denominados objetos. Esta abordagem garante que o projeto seja desenvolvido de maneira modular, ou seja, de maneira que funcionalidades desacopladas interajam para atingir um determinado objetivo, satisfazendo a necessidade de algum requisito. Sobre isso, Pressman (2016) comenta que “o software é dividido em componentes separadamente especificados e localizáveis, algumas vezes denominados módulos, que são integrados para satisfazer os requisitos de um problema”. (PRESSMAN, 2016, p.234).

Atualmente, o paradigma de orientação a objetos é amplamente difundido no mundo de desenvolvimento de software. Este paradigma é fundamentado em 4 pilares essenciais. Gasparotto (2014) cita a POO corresponde a um padrão de desenvolvimento incorporado fortemente em linguagens de programação como C# e Java. O mesmo autor ainda descreve brevemente cada um dos 4 pilares fundamentais da orientação a objetos:

- Abstração: consiste em observar uma entidade do mundo real e transcrevê-la para dentro de um software, tornando-a uma classe de um tipo específico.

- Encapsulamento: consiste em esconder partes críticas do software, consolidando o conceito de segurança dentro do cenário de desenvolvimento de uma aplicação.
- Herança: consiste na passagem de estruturas de uma classe mãe (ou generalista) para uma classe filha (ou especialista). Este conceito ilustra a prática do reuso de código, ponto extremamente vantajoso no que se refere a desenvolvimento de software, e é um dos grandes diferenciais do paradigma de orientação a objetos em relação ao paradigma procedural, por exemplo.
- Polimorfismo: consiste na reescrita e alteração no funcionamento interno de um método cuja origem é uma classe mãe, método esse obtido através de uma herança.

As classes podem representar um elemento físico, como uma pessoa ou veículo, um elemento negocial, como uma compra ou pedido, ou um elemento de aplicativo, como um botão ou uma caixa de texto.

Ainda sobre o pilar da abstração, as classes trazidas para o ambiente sistêmico devem ser constituídas de estruturas específicas, através das quais serão possíveis fidelizar suas características com as características do objeto observado do mundo real. (PRESSMAN, 2016).

Sobre a estrutura e as características de uma classe contida no sistema de software, Gasparotto (2014) comenta que uma classe deve ser identificada com um nome único, o qual indica um tipo específico. O restante da estrutura desta classe são os atributos e métodos. Atributos (as vezes chamados propriedades) são variáveis que atribuem características àquele tipo. Os métodos, por sua vez, indicam as ações que este tipo pode realizar, e geralmente são apresentados com verbos no infinitivo.

2.6.2 TECNOLOGIAS UTILIZADAS

No cenário da programação do presente jogo, foi utilizado a linguagem Java. O Java além de uma linguagem de programação, também é uma robusta plataforma computacional. Presente no mercado desde 1995, lançado inicialmente pela Sun Microsystems, o Java é a

tecnologia base para que diversas aplicações sejam executadas pelo mundo todo, segundo o site da Oracle, atual proprietária da tecnologia Java.

O Java, enquanto linguagem de programação, implementa fortemente o paradigma de orientação a objetos em toda a sua extensão. (GASPAROTTO, 2014).

Apresentando a característica de tipagem forte, tudo dentro do Java pertence a um tipo específico. Além da tipagem, todos os pilares da orientação a objetos e muitos outros conceitos além também são implementados pela linguagem, trazendo inúmeras vantagens no cenário de desenvolvimento.

O projeto foi desenvolvido utilizando Java Swing. O Swing é uma API de padrão de Interface Gráfica de Usuário (GUI) para ser utilizada com o Java. Sua aplicação é no desenvolvimento de projetos com interface gráfica, onde a renderização de componentes gráficos é de responsabilidade do Swing. (FRIESEN, 2014).

O banco de dados utilizado na aplicação trata-se do Banco de Dados Relacional MySQL. Para que a integração entre o banco de dados e a aplicação Java fosse possível, foi utilizado o Java Database Connectivity (JDBC). Segundo descrito pela Oracle em seu site, na seção de tecnologias da plataforma JavaSE, o JDBC consiste em uma API padrão para estabelecer uma conectividade independente entre um programa escrito em Java e um serviço de banco de dados de amplamente variado que, no caso, foi o banco de dados relacional MySQL.

Quanto a modelagem do projeto, foi adotada a UML. O padrão de Linguagem de Modelagem Unificada (UML), proposto por Booch, Jacobson e Rumbaugh (1990), define uma linguagem padronizada para a elaboração de estruturas de projetos de software, no formato de modelos de documentos e diagramas.

2.6.3 CICLO DE VIDA E PROCESSO DE SOFTWARE

Esta seção traz as etapas compreendidas pelo processo de software, também conhecido como ciclo de vida de desenvolvimento de software.

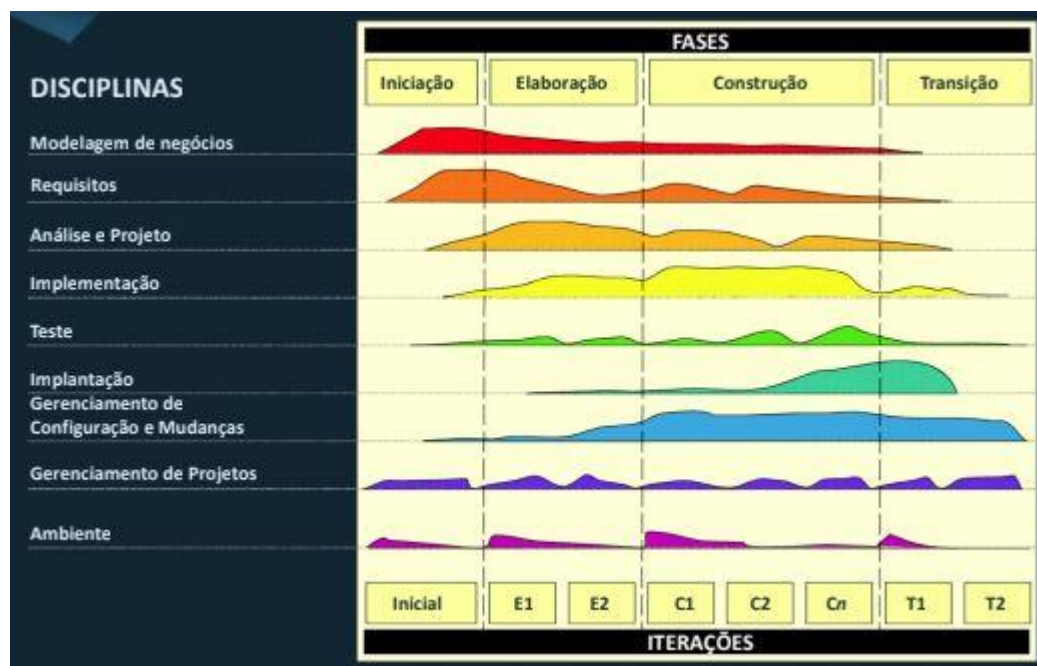
Durante o processo de software, deve-se definir qual será o modelo prescritivo a ser utilizado, e também, porém não obrigatoriamente, alguma modelagem ágil, baseada numa

metodologia composta de conceitos e princípios, afim de complementar o modelo prescritivo escolhido. A respeito disso, Pressman diz que:

Quando você trabalha para construir um produto ou sistema, é importante seguir uma série de passos pré-definidos – um roteiro que ajuda você a criar um resultado oportuno e de alta qualidade. O roteiro que você segue é chamado de processo de software. (PRESSMAN, 2015, p.30).

Nesta etapa do projeto, foi utilizado o processo unificado. O processo unificado consiste num modelo prescritivo baseado em iterações, orientado a casos de uso, sendo representado por um conjunto de atividades ou disciplinas mapeadas ao longo do desenvolvimento, sendo mensuradas através de esforços ao decorrer de cada fase. (ZATTI, 2019). Este processo é altamente compatível com modelagens ágeis, pois trabalha sobre alguns de seus principais princípios. O processo unificado é ilustrado na Foto 8 abaixo.

Foto 8 – Processo Unificado



Fonte: NICOLLETE, 2012 (adaptado).

O paradigma de orientação a objetos prevê a elaboração de diversos artefatos que sejam capazes de representar o sistema como um todo, como por exemplo o banco de dados da aplicação, as classes e atores que atuam dentro do sistema, os fluxos de informação e os casos de uso de cada uma de suas funcionalidades. Para este projeto, foi julgado relevante apresentar graficamente os elementos listados a seguir:

- Quais classes compõem o sistema, e quais são os objetos pertencentes;
- Quais os casos de uso possíveis para cada ator dentro do jogo;
- Modelagem do Banco de Dados representando identificadores primários e relacionamentos de entidades.

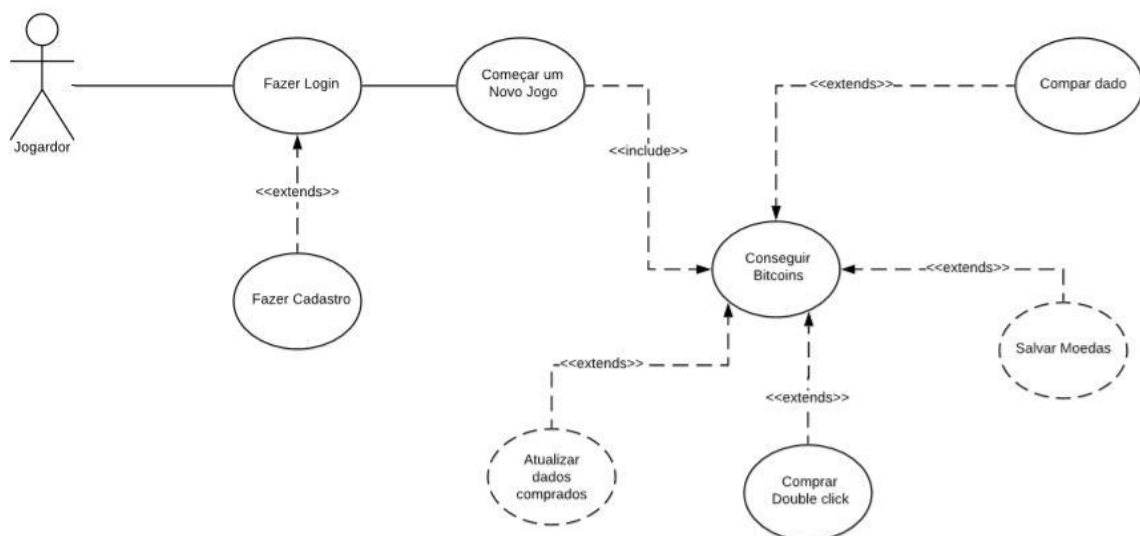
Sendo assim, foi utilizado o padrão UML, proposto por Booch, Jacobson e Rumbaugh (1990), para a representação dos elementos listados anteriormente. Com isso, foram escolhidos os seguintes artefatos para representar a estrutura do presente jogo:

- Diagrama de Casos de Uso;
- Diagrama de Classes;
- Diagrama Entidade-Relacionamento (modelo de banco de dados conceitual);
- Documentos de Especificação de Requisitos Funcionais (*use case*);
- Modelo Entidade-Relacionamento (modelo de banco de dados lógico).

Os documentos elaborados pela equipe que correspondem à especificação de requisitos do projeto de software estão contidos nos apêndices deste trabalho, por se tratarem de documentos detalhados.

Utilizado também como artefato para levantamento de requisitos funcionais, o diagrama de caso de uso apresenta as principais funcionalidades do sistema e, também, como os atores vão interagir com o mesmo. O boneco representa um ator utilizador do sistema, a elipse representa um caso de uso, que consiste basicamente em uma funcionalidade, e as setas representam a comunicação que há entre um ator e um caso de uso.

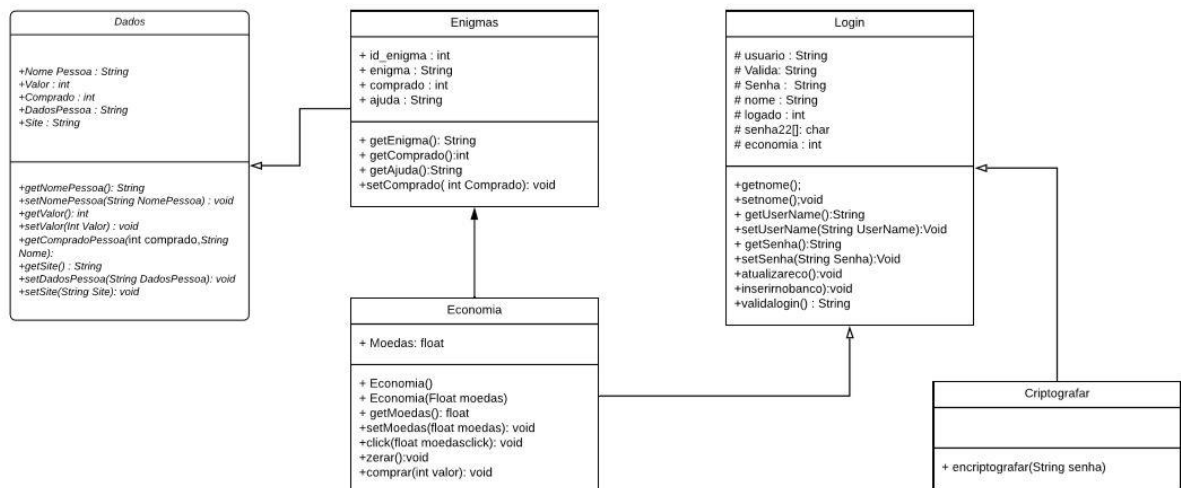
Foto 9 – Diagrama de Casos de Uso



Fonte: Produção autoral.

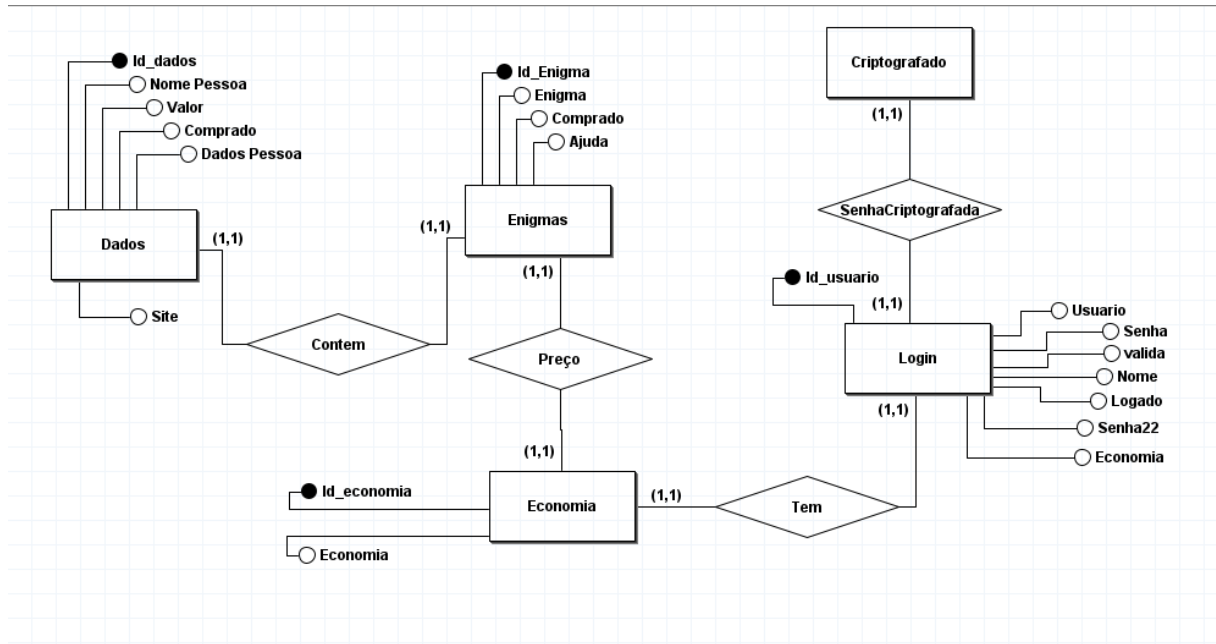
Uma classe é uma definição abstrata de um tipo composto, e um objeto é a instância da própria classe. (PRESSMAN, 2016). Sendo assim, os objetos do mesmo tipo pertencem a mesma classe. Um diagrama de classes é uma representação da estrutura e das relações das classes, que, por sua vez, são representações de objetos de um tipo específico. Na notação UML, uma classe é representada por um retângulo, com três divisões, sendo o nome da classe (identificador), seus atributos ou propriedades e seus métodos.

Foto 10 – Diagrama de Classes



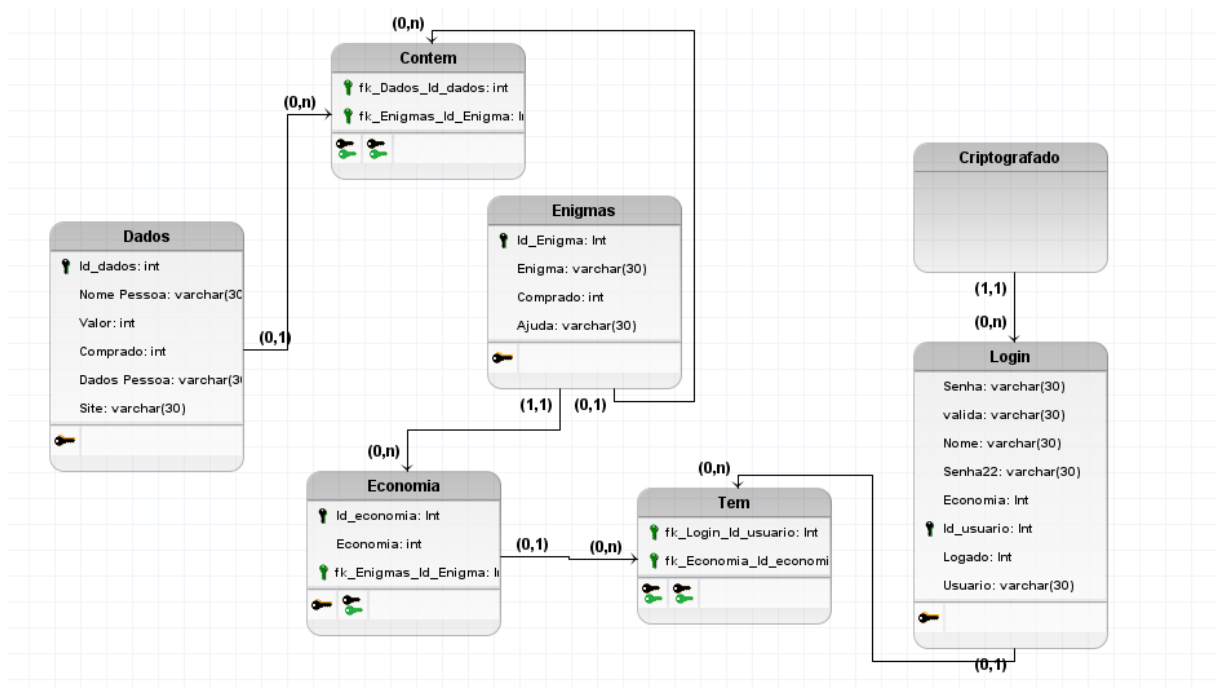
Fonte: Produção autoral.

O DER, sigla para Diagrama Entidade-Relacionamento, é a representação gráfica dos objetos de dados do sistema e das relações estabelecidas entre si. Na Engenharia de Software, um DER é utilizado para projetar e representar a estrutura de um banco de dados relacional através de um modelo conceitual. Nesta representação gráfica, os retângulos representam as entidades, os losangos representam o relacionamento entre as entidades, e os pontos ligados a cada entidade representam os seus atributos (no banco de dados são as colunas da tabela), onde os pontos preenchidos representam a chave primária da entidade (PK).

Foto 11 – Diagrama Entidade-Relacionamento (modelo conceitual)

Fonte: Produção autoral.

O seguinte modelo entidade-relacionamento traz a representação lógica da estrutura relacional do banco de dados, muito semelhante ao DER, diferindo na representação das entidades e na forma como se dão os relacionamentos entre si.

Foto 12 – Modelo Entidade-Relacionamento (modelo lógico)

Fonte: Produção autoral.

3 RESULTADOS E DISCUSSÃO

Com a entrada da nova Lei Geral de Proteção de Dados Pessoais em vigor, é indubitável que a área de tecnologia da informação desempenhará um papel de suma importância, como apresentado em fundamentos citados no desenvolvimento deste instrumento. Contudo, aquele que garantirá a aplicação da lei no seu contexto, em toda a sua amplitude, seja em sua universidade ou no seu trabalho, deverá se atentar e aplicar de maneira eficaz os conhecimentos adquiridos na sua formação. O jogo apresentado ilustra, de maneira simples, que os processos de tratamento de dados, após a implantação da LGPD, nunca mais serão os mesmos. Além disso, a estrutura do jogo procurou ser trabalhada atendendo alguns princípios estabelecidos na referida lei, buscando a anonimização máxima dos dados manipulados durante o jogo.

3 CONCLUSÃO

Sendo essa a segunda e última entrega deste trabalho, pode-se concluir que o desempenho da equipe neste projeto foi satisfatório. Os objetivos definidos no início deste projeto foram parcialmente concluídos, uma vez que algumas melhorias ainda estão em aberto, e algumas melhorias bastante consideráveis foram implementadas desde a última entrega da primeira versão deste trabalho. A equipe encontra-se muito satisfeita com os resultados obtidos até aqui, levando em consideração que, durante o processo de desenvolvimento tanto da parte prática quanto da parte teórica, muito conhecimento foi adquirido e, com certeza, contribuiu muito para o processo de aprendizado não somente do conteúdo abordado, mas além disso, de atividades que futuramente se farão necessárias no dia a dia de qualquer outro trabalho.

REFERÊNCIAS

BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML, Guia do usuário**. 2ª ed., Editora Campus, 2005.

BRAGA, Rosana Vaccare; MASIERO, Paulo Cesar. **Um método completo para Desenvolvimento Orientado a Objetos com UML: da análise à implementação em Java**. São Carlos: ICMC, 2007.

BRASIL. LEI Nº 13.709. Lei Geral de Proteção de Dados Pessoais. Decretada pelo Congresso Nacional e sancionada pela Presidência da República. **Diário Oficial da União**, Brasília em 14 de Agosto de 2018.

FERREIRA, Andrew Roberto Lopes; CABRAL, Heleno Carmo Borges; SONNENSTRAHL, Thiago. **Política de Segurança da Informação**. Santa Maria: _____, 2013. 55 p. Disponível em: <http://www2.fw.iffarroupilha.edu.br/cti/PSI.pdf>. Acesso em: 10 maio 2020.

FRIESEN, Jeff. **Java ME 8 and the Internet of Things**. JavaWorld. Disponível em: <https://www.javaworld.com/article/2848210/java-me-8-and-the-internet-of-things.html>. Acesso em: 26 de maio de 2020.

GASPAROTTO, Henrique Machado. **Os 4 pilares da Programação Orientada a Objetos**. DevMedia. Disponível em < <https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>>. Acesso em: 18 de maio de 2020.

LARMAN, C. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3ª edição, Bookman, 2008.

MORE: **Mecanismo online para referências**, versão 2.0. Florianópolis: UFSC Rexlab, 2013. Disponível em: <http://www.more.ufsc.br/>. Acesso em: 15 de maio de 2020.

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de software: uma abordagem profissional. 7ª e 8ª ed. Porto Alegre: AMGH, 2016. 940 p.

ZATTI, Evandro. **Ciclo de vida e desenvolvimento de software**. Consultado em 13 junho de 2020.

APÊNDICES

Os documentos citados na seção de desenvolvimento estão contidos neste tópico, trazidos como apêndices. Os documentos correspondem às especificações de requisitos.

Tabela 1 – Lista de apêndices

APÊNDICE A	UC001 – Fazer login
APÊNDICE B	UC002 – Cadastrar jogador
APÊNDICE C	UC003 – Começar um novo jogo
APÊNDICE D	UC004 – Atualizar dados comprados
APÊNDICE E	UC005 – Conseguir Bitcoins
APÊNDICE F	UC006 – Comprar Double Click
APÊNDICE G	UC007 – Salvar moedas
APÊNDICE H	UC008 – Comprar dado

Fonte: Produção autoral.

APÊNDICE A

UC001 – Fazer Login	
Função	Fazer login.
Descrição	Requisitar dados do jogador para entrar no jogo.
Entradas	Dados do Jogador como username e senha já cadastrados no sistema.
Origem	Inserção de dados através do preenchimento de um formulário.
Saídas	Login concluído.
Destino	Dados do Jogador vão para o banco de dados para a validação e também para permitir a continuidade no jogo.
Ação	Com o login feito e validado, o jogador já está apto a começar um novo jogo ou continuar um jogo.
Requer	Inserção de login e senha do jogador para validação de sua conta.
Pré-condição	O jogador já deve ter feito o cadastro no jogo.
Pós-condição	Jogador é validado e pode começar a jogar.
Efeitos colaterais	Nenhum.

APÊNDICE B

UC002 – Cadastrar Jogador	
Função	Cadastrar jogador.
Descrição	Cadastrar jogadores no sistema.
Entradas	Dados do jogador: <i>username</i> , e-mail e senha . A identificação interna do jogador (id_jogador) é gerada automaticamente pelo sistema.
Origem	Inserção de dados através do preenchimento de um formulário.
Saídas	Cadastro de jogador.
Destino	Dados do jogador vão para o banco de dados para liberar seu acesso.
Ação	Com os dados completos, efetuar o login para começar a jogar
Requer	Inserção de dados pelo jogador.
Pré-condição	Não ter se cadastrado antes.
Pós-condição	Permissão ao novo jogador para jogar.
Efeitos colaterais	Nenhum.

APÊNDICE C

UC003 – Começar um Novo Jogo	
Função	Começar um novo jogo.
Descrição	Iniciar um jogo do início.
Entradas	Escolher a opção de novo jogo
Origem	Login.
Saídas	Começar um jogo sem nenhum save.
Destino	Os dados do jogo começaram a ser salvos desde o início da gameplay.
Ação	começar um novo jogo.
Requer	Ter realizado o login.
Pré-condição	Login realizado com sucesso.
Pós-condição	Começar o jogo sem nenhum save.
Efeitos colaterais	Nenhum.

APÊNDICE D

UC004 – Atualizar dados comprados	
Função	Atualizar os dados já comprados.
Descrição	Mostrar um botão com o mesmo nome da função para mostrar ao jogador quantos objetivos o mesmo já completou.
Entradas	Nenhuma.
Origem	Usuário já ter comprado algum dado.
Saídas	Número de dados comprados e quantos faltam.
Destino	Dados do jogador vão para o banco para que o objetivo possa ser alcançado.
Ação	Após a compra de algum dado o jogador já pode acessar essa função
Requer	Usuário já ter comprado algum dado.
Pré-condição	O jogador já deve ter feito o cadastro no jogo, ter dinheiro pra comprar algum dado e Usuário já ter comprado algum dado.
Pós-condição	Jogador consegue ver quantos dados foram comprados.
Efeitos colaterais	Nenhum.

APÊNDICE E

UC005 – Conseguir Bitcoins	
Função	Conseguir Bitcoins a partir do botão na tela.
Descrição	Clicar no botão na tela principal a fim de conseguir bitcoins para proseguir no jogo.
Entradas	Apenas o clique do jogador já consegue os Bitcoins.
Origem	Através dos cliques do jogador.
Saídas	1 bitcoin a cada clique.
Destino	Cada bitcoin conseguido vai para a economia do jogador, para possibilitar o jogador comprar os dados e upgrades.
Ação	Quantidade dos Bitcoins é atualizada a cada clique.
Requer	O jogador já esteja logado e com a possibilidade de clicar no botão.
Pré-condição	Jogador esteja logado.
Pós-condição	Quantidade de Bitcoins de determinado jogador recebe sua quantidade inicial somado ao número de cliques.
Efeitos colaterais	Nenhum.

APÊNDICE F

UC006 – Comprar Double Click	
Função	Comprar o upgrade de clique duplo.
Descrição	Após a compra desse upgrade cada clique do jogador contará como 2 cliques e assim conseguirá 2 bitcoins por clique.
Entradas	Ter a quantidade suficiente para comprar esse upgrade.
Origem	Tela principal.
Saídas	A cada clique do jogador contará 2 bitcoins.
Destino	Informações da compra são armazenadas no banco de dados e a quantidade de bitcoins também é atualizada.
Ação	Ativará o clique duplo e o jogador conseguirá 2 bitcoins por clique.
Requer	Que o jogador tenha bitcoins suficiente para comprar esse upgrade.
Pré-condição	Jogador logado e com bitcoins suficientes para a compra.
Pós-condição	Quantidade de bitcoins recebe sua quantidade inicial mais 2 bitcoins adquiridos.
Efeitos colaterais	Nenhum.

APÊNDICE G

UC007 – Salvar Moedas	
Função	Salvar as moedas do jogador.
Descrição	Mostrar um botão com o mesmo nome da função para o jogador salvar a quantidade de moedas que ele tem no momento para seu perfil no jogo, para na próxima vez que ele logar ter a quantidade salva;
Entradas	nenhuma
Origem	Usuário ter algum dinheiro para salvar .
Saídas	BitCoins salvos com sucesso.
Destino	As moedas vão para o perfil do jogador no Banco de Dados .
Ação	Quando o usuário tiver algum dinheiro ele ja pode salvar
Requer	Usuário ter algo para salvar.
Pré-condição	O jogador ter alguma quantidade de bitcoins.
Pós-condição	O jogador tem suas moedas salvas.
Efeitos colaterais	Nenhum.

APÊNDICE H

UC008 – Comprar Dado	
Função	Comprar os dados dos personagens.
Descrição	Comprar os dados dos 5 personagens para completar o objetivo do jogo.
Entradas	Ter a quantidade suficiente para comprar o dado.
Origem	Tela de compra de dados.
Saídas	O jogador terá que resolver o enigma que estará criptografado para ter acesso aos dados das pessoas.
Destino	Informações da compra são armazenadas no banco de dados e a quantidade de bitcoins também é atualizada.
Ação	Após resolver o enigma o jogador terá acesso aos dados antes criptografados.
Requer	Que o jogador tenha bitcoins suficiente para comprar esse dado.
Pré-condição	Jogador logado e com bitcoins suficientes para a compra.
Pós-condição	Quantidade de bitcoins diminui e o jogador consegue o dado comprado.
Efeitos colaterais	Nenhum.

