

Laboratório de Engenharia de Software

LES

Desenvolvimento de Software

Versão: 21/11/2017

*Prof. Antonio Sergio Bernardo*

RESUMO x

1. INTRODUÇÃO.......,..... x

2. ANÁLISE DE NEGÓCIOS

2.1 Análise da Situação Atual x

2.1.1 Descrição do Macro-Processo x

2.1.2 Diagrama de Processos As Is x

2.1.3 Diagnóstico da Situação Atual x

2.1.4 Descrição das Mudanças Necessárias x

2.2 Planejamento de Melhorias x

2.2.1 Diagrama de Processos To Be x

2.2.2 Descrição dos Processos To Be x

Fornecimento de Refeições x

Preparação de Refeições x

Recebimento de Mercadorias x

2.2.3 Plano de Implementação (Word) x

3. SISTEMA DE INFORMAÇÃO

3.1 Requisitos Não Funcionais....................................................................................................................... x

3.1.1 Requisitos de Desempenho............................................................................................................ x

3.1.2 Requisitos de Armazenamento x

3.1.3 Requisitos de Hwm Sw, Rede x

3.1.4 Requisitos Outros (Anexo) x

3.2 Requisitos Funcionais x

3.2.1 Diagramas de Casos de Uso x

3.2.2 Descrição dos Casos de Uso x

3.3 Projeto Detalhado do Software x

3.3.1 Arquitetura da Aplicação x

3.3.2 Tecnologias a Serem Utilizadas x

3.3.3 Componentes do Sw x

3.3.4 Modelo de Dados x

3.3.5 Protótipo Back-End x

3.3.6 Protótipo Front-End x

3.3.7 Refinamento dos Requisitos de Hw, Sw, Rede x

3.4 Codificação x

3.4.1 Script de Tabelas x

3.4.2 Código Fonte (CD Anexo) x

3.4.3 Testes e Aprovação x

4. GERENCIAMENTO DA TRANSIÇÃO

4.1 Preparação.............................................................................................................................................. x

4.1.1 Adequação de Recursos (Hw, Sw, Rede) x

4.1.2 Instalação do Software x

4.1.3 Configurações de Acesso e Permissões x

4.1.4 Aspectos Operacionais x

4.2 Efetivação ............................................................................................................................................ x

4.2.1 Treinamentos x

4.2.2 Implementação das Mudanças x

4.3 Finalização........................................................................................................................................... x

4.3.1 Monitoramento e Refinamento (Ajustes) x

4.3.2 Planejamento de Novo Ciclo (Versão) x

5. CONCLUSÃO x

6. BIBLIOGRAFIA x

7. GLOSSÁRIO x

**2Play**

RESUMO

2Play nasceu com a ideia de um e-commerce brasileiro de jogos e aplicativos digitais para os sistemas Windows, Mac e Linux a um preço acessível aos usuários. A 2Play também fornece serviços facilitado como a disponibilidade de atualização de seus produtos.

Foi feita por meio de softwares de código aberto, sendo eles o AngularJS, o NodeJS e o MongoDB.

1. INTRODUÇÃO

A 2Play foi pensada para ser uma plataforma orientada a documentos e com um banco de dados não relacional, já que dessa forma traria uma maior flexibilidade de estruturação, inserção e acesso aos dados, além de ter uma escalabilidade mais fácil.

Para o front-end, seria utilizado o AngularJS e o Bootstrap, dois frameworks de código aberto para auxiliar no desenvolvimento da interface, facilitando a construção da aplicação web e dinamizando seus componentes.

Para o back-end temos o NodeJS, já que por a 2Play se tratar de uma aplicação web e de sua alta escalabilidade, seria necessário um interpretador de código capaz de manipular milhares de dados de forma simultânea.

Neste documento é apresentada toda a documentação, os diagramas de casos de uso, os requisitos e a modelagem dos sistemas.

3.1 Requisitos Não Funcionais

3.1.1 Requisitos de Desempenho

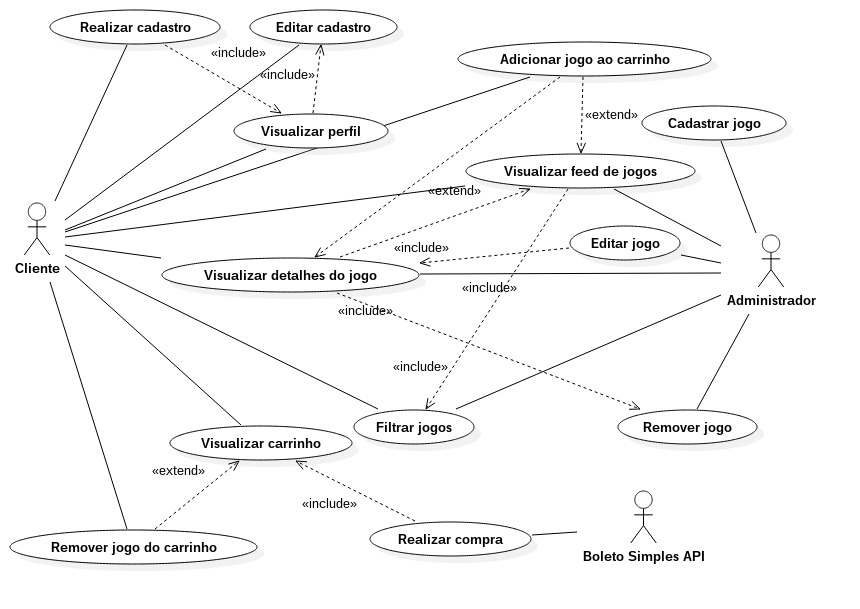
3.1.2 Requisitos de Armazenamento

3.1.3 Requisitos de Hw, Sw, Rede

3.1.4 Requisitos Outros (Anexo)

* 1. Requisitos Funcionais

3.2.1 Diagramas de Casos de Uso



3.2.2 Descrição dos Casos de Uso

|  |
| --- |
| **Caso de uso:** Realizar cadastro |
| **Atores:** Cliente |
| **Interessados:** Cliente que deseja ter acesso as compras |
| **Pré-condições:** Cliente não possui cadastro no sistema |
| **Pós-condições:** Usuário é redirecionado para o feed de jogos |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente entra em cadastro de usuário 2. O cliente insere seus dados pessoais 3. O sistema valida os dados 4. O cliente salva os dados no sistema |
| **Tratamento de exceções:**  **2.** **Os dados estão inválidos**  2.1 Volta passo 2 |

|  |
| --- |
| **Caso de uso:** Editar cadastro |
| **Atores:** Cliente |
| **Interessados:** Cliente que deseja atualizar seu cadastro |
| **Pré-condições:** Cliente deve possuir cadastro no sistema |
| **Pós-condições:** Mensagem de atualização é mostrada ao usuário |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente visualiza o perfil 2. O cliente insere seus dados pessoais 3. O sistema valida os dados 4. O cliente salva os dados no sistema |
| **Tratamento de exceções:**  **2.** **Os dados estão inválidos**  2.1 Volta passo 2 |

|  |
| --- |
| **Caso de uso:** Visualizar perfil |
| **Atores:** Cliente |
| **Interessados:** Cliente que deseja visualizar seu perfil |
| **Pré-condições:** Estar cadastrado no sistema |
| **Pós-condições:** Perfil é mostrado ao usuário |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente entra em visualizar perfil |
| **Tratamento de exceções:** |

|  |
| --- |
| **Caso de uso:** Adicionar jogo ao carrinho |
| **Atores:** Cliente |
| **Interessados:** Cliente que deseja comprar um jogo |
| **Pré-condições:** O cliente deve possuir cadastro no sistema |
| **Pós-condições:** O cliente deve poder continuar a navegação |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente seleciona adicionar ao carrinho 2. O jogo é adicionado ao carrinho |
| **Tratamento de exceções:** |

|  |
| --- |
| **Caso de uso:** Cadastrar jogo |
| **Atores:** Administrador |
| **Interessados:** O administrador do sistema |
| **Pré-condições:** O administrador deve possuir todos os dados do jogo |
| **Pós-condições:** Mensagem de jogo cadastrado deve ser mostrado |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O administrador deve selecionar cadastrar jogo 2. O administrador deve inserir os dados 3. O sistema deve validar os dados 4. O administrador salva os dados |
| **Tratamento de exceções**  **2 O sistema deve validar os dados**  2.1 Volta ao passo 2 |

|  |
| --- |
| **Caso de uso:** Visualizar feed de jogos |
| **Atores:** Administrador, cliente |
| **Interessados:** Cliente interessado em novidades |
| **Pré-condições:** |
| **Pós-condições:** |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O usuário entra em visualizar feed. |
| **Tratamento de exceções** |

|  |
| --- |
| **Caso de uso:** Editar Jogo |
| **Atores:** Administrador |
| **Interessados:** O administrador do Sistema |
| **Pré-condições:** O jogo deve estar cadastrado |
| **Pós-condições:** Mostrar a mensagem de confirmação e redirecionar para o jogo com os dados atualizados. |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O Administrador deve selecionar editar jogo 2. O Administrador deve inserir os dados atualizados 3. O Sistema atualiza os dados 4. O Administrador salva a ação |
| **Tratamento de exceções** |

|  |
| --- |
| **Caso de uso:** Visualizar detalhes do jogo |
| **Atores:** Cliente |
| **Interessados:** Cliente interessado em detalhes do produto |
| **Pré-condições:** Ter o produto selecionado |
| **Pós-condições:** |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente deve clicar em visualizar detalhes 2. O sistema mostra os detalhes |
| **Tratamento de exceções** |

|  |
| --- |
| **Caso de uso:** Remover jogo |
| **Atores:** Administrador |
| **Interessados:** Administrador do Sistema |
| **Pré-condições:** Ter o produto selecionado |
| **Pós-condições:** Exibir mensagem de exclusão sucedida e redirecionar ao feed de jogos |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O administrador deve clicar em remover jogo 2. O sistema deve pedir uma confirmação 3. O administrador confirma a ação |
| **Tratamento de exceções** |

|  |
| --- |
| **Caso de uso:** Filtrar jogos |
| **Atores:** Administrador, Cliente |
| **Interessados:** Usuário a fim de filtrar jogo por categoria |
| **Pré-condições:** |
| **Pós-condições:** Mostrar os jogos na categoria |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O usuário deve selecionar as categorias pretendidas 2. O Sistema busca os jogos relacionados 3. O Sistema mostra os jogos encontrados |
| **Tratamento de exceções**   1. **Não foi encontrado nenhum jogo na categoria**   O sistema mostra a mensagem de “jogos não encontrados”, e o fluxo volta a posição 1 |

|  |
| --- |
| **Caso de uso:** Visualizar carrinho |
| **Atores:** Cliente |
| **Interessados:** Cliente interessado em ver o que pôs no carrinho |
| **Pré-condições:** |
| **Pós-condições:** Mostrar o carrinho |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente clica em visualizar carrinho 2. O sistema mostra o carrinho |
| **Tratamento de exceções** |

|  |
| --- |
| **Caso de uso:** Remover jogo do carrinho |
| **Atores:** Cliente |
| **Interessados:** |
| **Pré-condições:** O cliente precisa estar visualizando o carrinho e o mesmo não pode estar vazio |
| **Pós-condições:** Excluir o jogo pretendido |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O Cliente clica em remover jogo 2. O sistema retira o jogo do carrinho. |
| **Tratamento de exceções** |

|  |
| --- |
| **Caso de uso:** Realizar compra |
| **Atores:** Cliente, Gerador de boleto |
| **Interessados:** Cliente fechando a compra |
| **Pré-condições:** O carrinho não pode estar vazio, o cliente deve estar logado. |
| **Pós-condições:** Gerar o boleto |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente clica em fechar compra 2. O Sistema mostra os detalhes da compra e pede confirmação 3. O cliente confirma 4. O Gerador gera o boleto |
| **Tratamento de exceções** |

* 1. Projeto Detalhado do Sofware

3.3.1 Arquitetura da Aplicação

3.3.2 Tecnologias a Serem Utilizadas

Node 8.11.1, Angular5, Express5.8, Boostrap4, MongoDB

3.3.3 Componentes do Sw

3.3.4 Modelo de Dados

Não Relacional

3.3.5 Protótipo Front-End

Angular acessando back end

3.3.6 Protótipo Back-End

Acessando o mongo