

# Universidade Federal de Ouro Preto



## Engenharia de Software II

Sistema de *Empréstimo de Jogos*

Grupo: *caféina++*;

Alunos:	Caio Soares Costa Cibele Oliveira Ferreira Eduardo Matosinhos Florinda Gabriel Caetano Araújo
Professor:	Johnatan Oliveira
Horário:	Seg e Qua - 08:20 - 10:00

Ouro Preto, 1 de abril de 2021

# Conteúdo

<b>1</b>	<b>Histórico de Revisões</b>	<b>1</b>
<b>2</b>	<b>Processo e Software</b>	<b>1</b>
<b>3</b>	<b>Cronograma</b>	<b>1</b>
<b>4</b>	<b>Levantamento de Requisitos</b>	<b>2</b>
4.1	Diagrama de Caso de Uso . . . . .	2
4.2	Descrição de Caso de Uso . . . . .	2
<b>5</b>	<b>Especificação de Requisitos</b>	<b>3</b>
5.1	Requisitos Funcionais . . . . .	3
5.2	Requisitos Não Funcionais . . . . .	5
<b>6</b>	<b>Plano de VVT</b>	<b>5</b>
6.1	Requisitos a serem testados . . . . .	5
6.2	Estratégias e ferramentas de teste . . . . .	6
6.2.1	Tipos de teste . . . . .	6
6.3	Execução do Plano de Teste . . . . .	7
6.3.1	Registrar Jogo . . . . .	7
6.3.2	Registrar Cliente . . . . .	7
6.3.3	Adicionar ao Carrinho . . . . .	7
6.3.4	Efetuar Pagamento . . . . .	7
<b>7</b>	<b>Medição e Qualidade de Software</b>	<b>8</b>
<b>8</b>	<b>Referências</b>	<b>10</b>

## 1 Histórico de Revisões

Data	Versão	Descrição	Autor
10/03/2021	0.0	Justificativa do processo de software	Eduardo
10/03/2021	0.0	Levantamento de requisitos	Caio, Cíbele e Gabriel
20/03/2021	0.0	Especificação de requisitos	Caio, Cíbele e Gabriel
31/03/2021	0.0	Requisitos a serem testados	Eduardo

Tabela 1: Revisões do Documento

## 2 Processo e Software

A equipe “caféina++”, desenvolvedora do software “Game Stonks”, adotou o processo de desenvolvimento ágil Scrum. Esta escolha levou em consideração o fato deste modelo permitir a criação de um software útil rapidamente. Desta forma, desenvolvemos o sistema por uma série de incrementos, em que os usuários finais ou o representante do cliente participam da especificação e validação de cada incremento, garantindo o alcance da maior satisfação do cliente. Em termos teóricos ele se baseia em três pilares, transparência, inspeção e adaptação. Ou seja, todos os papéis, eventos, artefatos, propósitos e valores, são para dar visibilidade ao trabalho em questão, permitir uma validação antes, durante e depois do processo e a qualquer momento realizar ajustes nas demandas a serem executadas.

## 3 Cronograma

Nome	Tarefa	Prazo
Eduardo	Definir tema do projeto	01/03
Gabriel	Criar repositório no GitHub	01/03
Eduardo	Escolha e justificativa do Processo de Desenvolvimento de Software	13/03
Caio, Cíbele e Gabriel	Levantamento dos requisitos do software	13/03
Caio, Cíbele e Gabriel	Especificação de requisitos	20/03
Gabriel	Definição da tecnologia, arquitetura e classes	20/03
Caio, Cíbele, Eduardo e Gabriel	Definir backlog	24/03

Tabela 2: Cronograma

## 4 Levantamento de Requisitos

### 4.1 Diagrama de Caso de Uso

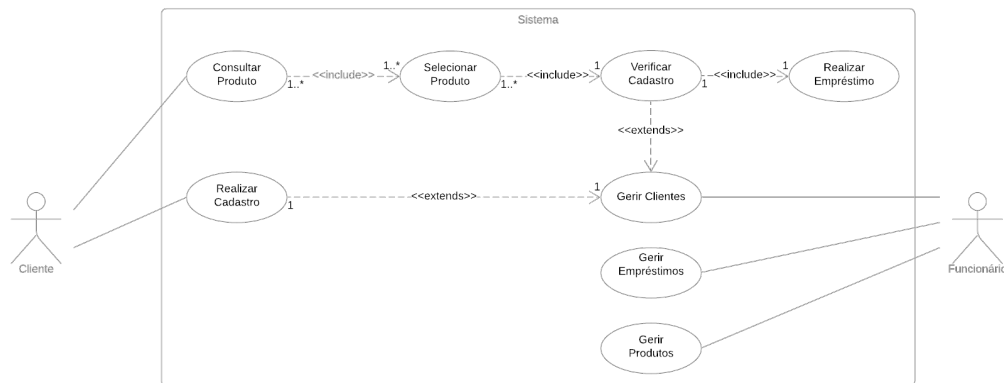


Figura 1: Diagrama de Casos de Uso

### 4.2 Descrição de Caso de Uso

Código	Serviço	Descrição
RF01	Realizar cadastro	Permite o cliente se cadastrar no sistema.
RF02	Consultar produtos	Permite o cliente consultar um jogo no sistema.
RF03	Selecionar produto	Permite o cliente selecionar um jogo para empréstimo.
RF04	Verificar cadastro	Possibilita a verificação do cadastro do cliente no sistema.
RF05	Realizar empréstimo	Efetiva o empréstimo do produto para o cliente que o selecionou.
RF06	Gerir clientes	Gerencia os clientes que estão cadastrados no software.
RF07	Gerir empréstimos	Gerencia os empréstimos realizados.
RF08	Gerir produtos	Gerencia os produtos (jogos) do software.

Tabela 3: Descrição de Casos de Uso

## 5 Especificação de Requisitos

### 5.1 Requisitos Funcionais

**RF01– Realizar cadastro:** Este serviço realiza todas as tarefas relacionadas à autenticação.

- **Criar login:** Campo de texto para login.
- **Criar senha:** Campo de texto para senha.
- **Recuperar senha:** Permite recuperar a senha.
- **Alterar senha:** Permite alterar a senha.
- **Autenticar login e senha:** Valida o login e senha.
- **Sair do sistema:** Realiza o logout.

**RF02– Consultar Produto:** Este serviço possibilita que o usuário consulte se o jogo que ele irá pegar emprestado está disponível.

- **Filtrar Jogos:** Possibilita que o usuário filtre os jogos de acordo com tema, nome e empresa de desenvolvedora.
- **Consultar Disponibilidade:** Possibilita que o usuário veja se um jogo estará disponível para empréstimo.

**RF03– Selecionar Produto:** Este serviço permite ao usuário adicionar um jogo ao carrinho para o empréstimo.

- **Adicionar Jogo:** Permite adicionar jogos ao carrinho.
- **Remover Jogos:** Permite remover jogos do carrinho.
- **Atualizar valor de jogos no carrinho:** Atualiza o valor total dos jogos para empréstimo.

**RF04– Verificar cadastro:** Este serviço faz a verificação do cadastro do usuário e validação das informações

- **Autenticação de login:** Verifica se o usuário está cadastrado no sistema antes de prosseguir com o empréstimo.
- **Autenticação de informações:** Verifica se as informações inseridas pelo usuário estão corretas.

- **Verificar pendências:** Verifica se o usuário possui algum débito registrado no sistema.

**RF05– Realizar Empréstimos:** Este serviço possibilita que o usuário efetue o empréstimo do jogo que deseja.

- **Validar dados do cartão:** Valida os dados de cartão de crédito do usuário.
- **Efetuar pagamento:** Utiliza uma API para realizar o pagamento do empréstimo solicitado pelo usuário.
- **Liberar Empréstimo:** Disponibiliza para o usuário os jogos que ele escolheu por empréstimo.

**RF06– Gerir clientes:** Este serviço realiza todas as tarefas relacionadas ao cadastro de usuários do sistema.

- **Cadastrar usuário:** Permite cadastrar um usuário no sistema.
- **Consultar usuário:** Permite visualizar todos os dados de um usuário registrado.
- **Buscar usuário:** Permite fazer uma busca de um usuário no sistema.
- **Alterar dados do usuário:** Permite atualizar todos os dados relacionados a um usuário.
- **Remover usuário:** Permite excluir o usuário do sistema.

**RF07– Gerir empréstimos:** Este serviço realiza todas as tarefas relacionadas à contabilidade de gastos e lucros do Game Stonks.

- **Consultar empréstimos:** Permite ao administrador visualizar os empréstimos realizados.
- **Consultar lucro por jogo:** Permite ao administrador visualizar lucro por empréstimos de jogos.
- **Consultar resumo financeiro:** Permite ao administrador visualizar o resumo financeiro.
- **Exportar resumo financeiro:** Exporta relatório em formato CSV.
- **Registrar gastos com compra de jogos:** Permite que o administrador adicione os gastos com a compra de jogos.

- **Registrar lucro por jogo:** Permite que o administrador adicione os lucros por empréstimos de jogos.

**RF08– Gerir produtos:** Este serviço realiza todas as tarefas relacionadas ao gerenciamento dos jogos.

- **Registrar jogos:** Permite que o administrador adicione as informações pertinentes aos jogos.
- **Consultar jogos:** Permite ao administrador visualizar os jogos cadastrados no sistema.
- **Editar jogos:** Permite ao administrador editar as informações dos jogos cadastrados no sistema.
- **Excluir jogos:** Permite ao administrador excluir jogos existentes no sistema.

## 5.2 Requisitos Não Funcionais

**RNF01.** A efetivação do empréstimo do jogo só deve ser liberada após o cliente estar logado no sistema. **Informações:** usuário e senha. **Regras:** o cliente terá acesso para contratar, consultar e alterar.

**RNF02** Compatibilidade com os sistemas operacionais Windows, Linux e MacOS.

**RNF03** Tempo limite para processamento de todas as solicitações de empréstimos de jogos.

**RNF04** Um visitante poderá visualizar os jogos que site tem disponível.

**RNF05** Informações pessoais dos usuários não podem ser vistas pelos operadores do sistema.

## 6 Plano de VVT

### 6.1 Requisitos a serem testados

**Teste de desempenho:** O objetivo deste teste consiste em validar se os requisitos de desempenho foram alcançados, como tempos de respostas e outros requisitos sensíveis ao tempo. Para isso deve-se:

- Verificar tempo de resposta para login.
- Verificar a eficácia da aplicação do software diante de um aumento de acessos simultâneos.

- Verificar o tempo de resposta para atualizar o relatório financeiro diante das transações realizadas pelos usuários.

**Teste de segurança:** O objetivo deste teste consiste em verificar a proteção dos dados dos usuários do software, bem como revelar falhas e garantir a qualidade do software. Para isso deve-se:

- Verificar possíveis falhas relacionadas a quebra de autenticação.
- Verificar se há proteção dos dados pessoais de todos os usuários.
- Verificar se há redirecionamento para sites maliciosos.

**Teste de interface de usuário:** O objetivo deste teste consiste em verificar se os componentes da interface estão respondendo às ações do usuário de maneira correta. Para isso deve-se:

- Verificar funcionamento de botões.
- Tratar eventos de timeouts.
- Verificar se a navegação pelo aplicativo reflete os requisitos de negócios.
- Verificar se objetos da interface estão em conformidade com os padrões.

## 6.2 Estratégias e ferramentas de teste

**Ferramenta Utilizada:** Jest é um framework de teste unitário de código aberto em JavaScript, sendo uma das ferramentas de teste unitário mais difundidas dentro da comunidade.

### 6.2.1 Tipos de teste

**Teste Unitário:** O objetivo desse teste é verificar a adequadamente o processamento, recuperação dos dados e a funcionamento das estruturas implementadas. Esse tipo de teste baseia-se em técnicas de caixa branca, ou seja, usa a perspectiva interna do sistema, código fonte, para modelar os casos de teste.

**Técnicas Utilizadas:**

- Executar para cada classe do sistema a instanciação de objetos, acesso aos métodos e atributos.
- Executar todos os métodos de cada classe do sistema verificando a validade dos resultados.



#### **Critérios de Conclusão:**

- Todos os testes planejados foram executados.
- Todos os defeitos identificados foram tratados.

### **6.3 Execução do Plano de Teste**

#### **6.3.1 Registrar Jogo**

**Feature:** Testar a funcionalidade de registrar jogos.

**Cenário:** O administrador deseja realizar o cadastro de um jogo na sistema.

**Given:** O administrador informa os dados pertinentes aos jogos.

**When:** O administrador realiza o cadastro dos jogos no sistema.

**Then:** Os jogos são cadastrados no sistema.

#### **6.3.2 Registrar Cliente**

**Feature:** Testar a funcionalidade de registrar clientes.

**Cenário:** O usuário deseja realizar um empréstimo no sistema e para isso precisa se registrar para prosseguir.

**Given:** O usuário informa seus dados pessoais para o cadastro.

**When:** O usuário tenta realizar um empréstimo no sistema.

**Then:** O usuário preenche os campos com as informações solicitadas.

#### **6.3.3 Adicionar ao Carrinho**

**Feature:** Testar a funcionalidade que cria o carrinho de empréstimo do usuário.

**Cenário:** O usuário irá selecionar os jogos que ele deseja pegar emprestado.

**Given:** O usuário deseja alugar alguns jogos.

**When:** O usuário tenta realizar um empréstimo no sistema.

**Then:** O usuário finaliza a seleção de jogos e fecha o carrinho.

#### **6.3.4 Efetuar Pagamento**

**Feature:** Testar a funcionalidade de efetuar os pagamentos dos serviços prestados pelo sistema.

**Cenário:** O usuário deseja efetuar o pagamento referente ao empréstimo de um jogo.

**Given:** O cliente adiciona um jogo ao carrinho.

**And:** O cliente possui saldo suficiente.

**When:** O cliente conclui a compra.

**Then:** O valor referente ao empréstimo é descontado no saldo do cliente.

**And:** A compra e o pagamento são efetuados com sucesso.

## 7 Medição e Qualidade de Software

O *escomplex*<sup>1</sup> foi utilizado como ferramenta para ajudar na medição da qualidade do software. Essa ferramenta gera um relatório que tem um índice de complexidade do código escrito. Apesar disso, não se deve levar os resultados como valores absolutos para avaliar a complexidade do código desenvolvido. Algumas das métricas utilizadas por essa dependência são:

- Linhas de código: físicas (o número de linhas em um módulo ou função) e lógicas (uma contagem das instruções imperativas).
- Número de parâmetros: Analisados estaticamente a partir da assinatura da função, portanto, nenhuma contabilização é feita para funções que dependem dos argumentos do objeto (quanto menor, melhor).
- Complexidade ciclomática: É definida uma contagem do número de ciclos no gráfico de controle de fluxo do programa.
- Densidade de complexidade ciclomática: proposta como uma modificação da complexidade ciclomática, essa métrica simplesmente a re-expressa como uma porcentagem das linhas lógicas de código. (quanto menor, melhor).
- Medidas Halstead: Essas métricas são calculadas a partir do número de operadores e operandos em cada função (quanto menor, melhor).
- Índice de manutenção: É uma escala logarítmica de infinito negativo a 171, calculada a partir das linhas lógicas de código, a complexidade do ciclomatix e o esforço Halsteal (quanto mais alto, melhor).
- Densidade de primeira ordem: A porcentagem de todas as dependências internas possíveis que são realmente utilizados no projeto (quanto menor, melhor).

---

<sup>1</sup><https://github.com/escomplex/escomplex>

- Custo de mudança: A porcentagem de módulos afetados, em média, quando um módulo do projeto é alterado (quanto menor, melhor).
- Tamanho do núcleo: A porcentagem de módulos que são amplamente dependentes e dependem de outros módulos (quanto menor, melhor).

O *JSHint*<sup>2</sup> é uma ferramenta de análise de código estática usada no desenvolvimento de software para verificar se o código-fonte JavaScript está em conformidade com as regras de codificação.

---

<sup>2</sup><https://jshint.com/docs/>

## 8 Referências

- [1] Sommerville, Ian – Software Engineering, 8th Edition;