

Esse documento corresponde à análise do Diagrama de Casos de Classes versão 0.1(alpha). Aponta correções a serem realizadas e possíveis erros. **Observações(OBS)** devem ser analisadas pelo membro responsável pelo artefato!

**Autor(es):** Aroldo Augusto Barbosa Simões

**Data(última modificação):** 17/10/2023

## Documento de Análise do Diagrama de Classes

### Itens a serem verificados

- Sintaxe e semântica:
  - Notação e linguagem de acordo com a UML;
  - Formalidade, linguagem adequada e erros de português.
- Diagrama de Classes:
  - Conformidade dos relacionamentos entre componentes;
  - Nomeação de classes, atributos e métodos;
  - Associações entre classes.
- Outros:
  - Correlação entre Diagrama de Classes e Diagrama de Casos de Uso.
  - Compatível com o padrão ECS(Entity Component System, em português: Sistema de Componente e Entidade).

## Introdução:

- ☒ ~~1-Ortografia;~~
- ☒ ~~2-Texto explicativo.~~

## CS00:

- ☐ **1-Notação no padrão UML;**
  - ☒ ~~Nomeação de classes.~~
  - ☒ ~~Nomeação de atributos.~~
  - ☐ Nomeação de métodos.
    - ☐ **Erro:** Classe “SystemAccessController” método “getQuestsFromDB” está com *return void*.
    - ☐ **Verificar:** Classe “EntityQuest” métodos estão retornando *void*.
- ☐ **2-Linguagem formal e sem erros de português;**
  - ☒ ~~Nomeação de classes.~~
  - ☒ ~~Nomeação de atributos.~~
  - ☒ ~~Nomeação de métodos.~~
  - ☐ Descrição do Caso de Uso.
    - ☐ **Erros:** Tópico ‘Sistemas relacionados’:
      - ☐ **Substituir** ‘...caso de uso bem...’ por ‘...caso de uso, bem...’;
      - ☐ **Retirar** vírgula em ‘...Banco de Dados, e ela...’;
      - ☐ **Substituir** ‘...através de...’ por ‘...por meio de...’/’...mediante...’;
    - ☐ **Erros:** Tópico ‘Quests e Tasks’:
      - ☐ **Substituir** ‘...Quests...’ por ‘...quests...’ quando necessário;
      - ☐ **Substituir** ‘...Tasks...’ por ‘...tasks...’;
      - ☐ **Substituir** ‘...bem sucedido...’ por ‘...bem-sucedido...’;
      - ☐ **Substituir** ‘...que é responsável por...’ por ‘...responsável por...’;
      - ☐ **Substituir** ‘...de acordo com o...’ por ‘...conforme o...’;
      - ☐ **Substituir** ‘Já que temos diferentes tasks que compõem uma quest, e essas tasks...’ por ‘...uma quest e essas tasks por...’;
      - ☐ **Substituir** ‘...xamã, porém mais simplificado...’ por ‘...xamã, porém, mais simplificado...’;
      - ☐ **Substituir** ‘...natureza expansiva uma vez que...’ por ‘...natureza expansiva, uma vez que...’;
- ☐ **3-Classes;**
  - ☒ ~~As classes representam as entidades no ECS.~~
  - ☐ Cada classe possui apenas dados (componentes) e não contém lógica.
    - ☐ **Verificar:** classes como “EntityPlayer” possuem muitos atributos. Favor verificar a correspondência entre Entidade e Componente para estas.
  - ☒ ~~As classes não possuem métodos que alterem o estado do jogo diretamente.~~

☐ **4-Componentes;**

☐ Cada componente é uma estrutura de dados pura, contendo apenas atributos.

☐ Verificar: componente "ComponentItemManager".

☒ ~~Os componentes contêm dados específicos e não possuem comportamento ou lógica associada.~~

☒ **5-Herança;**

☒ ~~Não há hierarquia de herança para os componentes.~~

☒ ~~Herança é usada de forma mínima e apenas para as classes de sistemas.~~

☐ **6-Relacionamentos;**

☐ As entidades (classes) são associadas a seus componentes.

☐ **Verificar:** relacionamento entre "ComponentItemManager" e "EntityItem". Um "ItemManager" não seria uma entidade e um "Item" não seria um componente? Talvez ambos sejam componentes.

☒ ~~Os sistemas operam em grupos de entidades com determinados componentes.~~

☐ **Verificar:** cardinalidade entre "SystemAccessController" e "SystemMainGame".

☐ **Verificar:** relacionamento e cardinalidade entre "EntityPlayer" e "SystemQuestController".

☒ **7-Métodos;**

☒ ~~Os métodos nas classes são apenas para acessar ou manipular os componentes.~~

☒ **8-Operações do Sistema;**

☒ ~~As operações do jogo são realizadas pelos sistemas, não pelas classes.~~

☒ **9-Renderização;**

☒ ~~A renderização está em um sistema separado, não incorporada nas classes.~~

☒ **10-Conformidade com requisitos e regras de negócio.**

## CS01:

- ☒ **1-Notação no padrão UML;**
  - ☒ Nomeação de classes.
  - ☒ Nomeação de atributos.
  - ☒ Nomeação de métodos.
- ☐ **2-Linguagem formal e sem erros de português;**
  - ☒ Nomeação de classes.
  - ☒ Nomeação de atributos.
  - ☒ Nomeação de métodos.
  - ☐ Descrição do Caso de Uso.
    - ☐ **Erros:** Tópico 'Sistemas relacionados':
      - ☐ **Substituir** '...caso de uso bem...' por '...caso de uso, bem...';
      - ☐ **Substituir** '...simples e portanto pode...' por '...simples e, portanto pode...';
      - ☐ **Erro:** Sprite deve ser utilizado no masculino, portanto deve ser usado com preposições e artigos no masculino. Substituir: '...pela sprite...' por '...pelo sprite...' e '...da sprite.' por '...do sprite'.
- ☒ **3-Classes;**
  - ☒ As classes representam as entidades no EGS.
  - ☒ Cada classe possui apenas dados (componentes) e não contém lógica.
  - ☒ As classes não possuem métodos que alterem o estado do jogo diretamente.
- ☒ **4-Componentes;**
  - ☒ Cada componente é uma estrutura de dados pura, contendo apenas atributos.
  - ☒ Os componentes contêm dados específicos e não possuem comportamento ou lógica associada.
- ☒ **5-Herança;**
  - ☒ Não há hierarquia de herança para os componentes.
  - ☒ Herança é usada de forma mínima e apenas para as classes de sistemas.
- ☒ **6-Relacionamentos;**
  - ☒ As entidades (classes) são associadas a seus componentes.
  - ☒ Os sistemas operam em grupos de entidades com determinados componentes.
- ☒ **7-Métodos;**
  - ☒ Os métodos nas classes são apenas para acessar ou manipular os componentes.
- ☒ **8-Operações do Sistema;**

- ☒ ~~As operações do jogo são realizadas pelos sistemas, não pelas classes.~~
- ☒ **9-Renderização;**
  - ☒ ~~A renderização está em um sistema separado, não incorporada nas classes.~~
- ☒ **10-Conformidade com requisitos e regras de negócio.**

## Referências

- [1] LanguageTool - Para correção de textos e verificação ortográfica. Pode ser acessado em: <<https://languagetool.org/pt-BR>>. Último acesso em: 17/10/2023.
- [2] Diagrama de Classes - Sprint 1- Versão do *commit* "[Adição de documentação inicial do Diagrama de Classes](#)". Pode ser acessado no repositório de produto da Equipe 5, link acima. Último acesso em: 17/10/2023.
- [3] ChatGPT. Pesquisa de Checklist para correção de Diagrama de Classes de projeto que utiliza o ECS. Referências: "Game Programming Patterns" por Robert Nystrom; "Entity Systems are the future of MMOG development" por Adam Martin; "Understanding Component-Entity-Systems" por Mike McShaffry. Pode ser acessado em: <<https://chat.openai.com>>. Último acesso em: 17/10/2023.