

## **ART03 - Work products e templates associados**

### **→ Visão do projeto:**

Este documento fornece uma visão geral do projeto, incluindo seus objetivos, escopo, requisitos e restrições.

Responsáveis:

- Líder de Projeto
- Product Owner

Descrição Detalhada:

O documento inclui uma lista de todos os requisitos funcionais e não-funcionais desejados para o produto, além das informações necessárias para gerenciar o projeto. Define os parâmetros para o acompanhamento do progresso do projeto, especifica os objetivos de alto nível das iterações e seus marcos, e descreve a organização do projeto e os papéis dos envolvidos. A visão do Projeto será atualizado e expandido à medida que o conhecimento sobre o produto e seus usuários evolui.

Template: [Visão do Projeto](#)

### **→ Modelo de casos de uso:**

Este diagrama apresenta as funcionalidades do sistema, suas relações internas e interações com os usuários.

Responsáveis:

- Design de Software

Descrição Detalhada:

O Design de Software é responsável pela criação inicial do diagrama de casos de uso no nível de domínio. Em seguida, o diagrama de domínio é revisado e o Arquiteto adiciona detalhes ao modelo, resultando em um diagrama no nível de especificação.

Template: Para o diagrama de Casos de Uso, deve-se utilizar a Notação UML. A documentação associada deve seguir o Template de BEZERRA (2002).

### **→ Modelo de arquitetura :**

Modelos que descrevem a arquitetura do sistema, incluindo seus componentes, interfaces e relacionamentos.

Responsável:

- Arquiteto

Descrição Detalhada:

Define a estrutura organizacional do sistema, detalhando os componentes, suas interações, a divisão de responsabilidades, a estrutura organizacional e as tecnologias empregadas.

Template: N/A

### → Diagrama de Classes de Especificação:

Diagrama que descreve e especifica a estrutura interna do sistema, incluindo classes, atributos e métodos.

Responsáveis:

Design de Software

Descrição Detalhada:

O Design de Software é responsável pela criação inicial do diagrama de classes no nível de domínio. Após a revisão do diagrama de domínio, o Arquiteto adiciona detalhes ao modelo, resultando em um diagrama no nível de especificação.

Template: Notação UML

### → Especificação de Classes:

Este documento visa fornecer uma visão clara e estruturada das classes que compõem o sistema, facilitando o processo de desenvolvimento e integração. Nele, são detalhados os atributos, métodos e relacionamentos essenciais de cada classe, bem como breves descrições e observações adicionais que possam auxiliar os desenvolvedores juniores na implementação.

Responsável: Desenvolvedores Seniores

### → Plano de teste:

Documentos que descrevem os casos de teste a serem usados para validar o sistema.

Responsáveis:

- Analista de Qualidade

- Testador

Descrição Detalhada:

Este artefato especifica um conjunto de entradas de teste, condições de execução, resultados esperados e instruções passo a passo para realizar os testes. Para cada Sprint, são gerados testes de integração e usabilidade.

Template: [Plano de Teste](#)

#### → Planos de Iteração (Sprint Backlog):

Artefato que compreende as atividades a serem realizadas durante a sprint.

Responsáveis:

- Product Owner
- Líder de projeto
- Scrum Master

Descrição Detalhada:

O backlog da sprint é um conjunto de itens de trabalho que foram selecionados do backlog do produto para serem desenvolvidos em uma sprint específica. Ele representa o compromisso da equipe de desenvolvimento para a entrega de incrementos de produto durante o período da sprint.

Template: [Sprint Backlog](#)

#### → Resultado de teste:

Documentos que descrevem os resultados obtidos nos testes previamente definidos

Responsável: Analista de Qualidade e Testador

Descrição: Este artefato é a especificação de um conjunto de análises dos resultados referentes aos testes previamente especificados onde se considera se o resultado obtido está em conformidade com o esperado.

Template: [Resultado de Teste](#)

#### → Documentação do usuário (opcional):

Documento que descreve como usar o sistema e fornece instruções para o usuário.

Responsável: Desenvolvedor Back-End

Descrição: A depender do sistema a ser desenvolvido, uma documentação para o usuário com instruções em alto nível de como utilizar o sistema deve ser produzida, como um manual de uso.

→ **Relatório de revisão:**

Documento que descreve os resultados das revisões de código

Responsável: Analista de Qualidade e Testador/ Desenvolvedor Sênior

Descrição: Para cada Sprint duas versões deste artefato são criadas, uma para os Projetistas do Banco de Dados e outra para o Desenvolvedores Júnios do Back-End contendo os pontos positivos e negativos da entrega.

Template: Pull Request no GitHub

→ **Cartões de tarefas:**

Itens que representam as tarefas necessárias para implementar uma história de usuário. Cada tarefa é uma unidade de trabalho atribuída a um membro da equipe e pode ser acompanhada em sistema visual.

Responsável: Líder de Projeto

Descrição: Com base no Backlog da Sprint o Líder de Projeto cria os quadros de tarefas com a descrição das tarefas, os responsáveis pela tarefa e a data de entrega.

Template: Sistema Visual = Trello

→ **Dicionário para desenvolvimento back-end :**

Artefato para que a equipe de desenvolvimento back-end siga um padrão de desenvolvimento.

Responsável: Analista de Qualidade

Descrição: Especifica o nome das funções e das variáveis e como deve ser feita a organização do código.

Template: N/A

## → Ata de Reunião

Documento usado para padronizar as atas de reunião.

Responsável: Scrum Master/ Líder de Projeto

Descrição: Possui os campos com os participantes da reunião, pautas a serem discutidas, quem participou, quem solicitou a reunião e as definições daquelas reuniões.

Template: [Ata de reunião](#)

## → Issue

Forma de registrar e rastrear tarefas, bugs, melhorias e discussões relacionadas a um projeto.

Responsável: Gerente de Configuração e Mudança.

Descrição: Template com passo a passo de como deve ser uma issue do github, com descrição dos comportamentos observados, como deve ser o comportamento correto e com os passos de como resolver o problema

## → Arte de Conceito:

Esse artefato é criado nas fases iniciais do desenvolvimento para ajudar na validação da proposta com os stakeholders.

Responsável:

-Designer de jogos

Descrição:

Os criadores da arte de conceito elaboram representações visuais, como diagramas, wireframes e protótipos, que destacam a funcionalidade e a usabilidade pretendidas do sistema. Essas representações focam em aspectos de alto nível, como a estrutura geral, a interação do usuário e o fluxo de trabalho. O objetivo principal da arte de conceito é coletar feedback e alinhar expectativas antes de avançar para a fase de desenvolvimento mais detalhada.

## → Arte Final:

A arte final é a versão refinada e concluída do um artefato, pronta para ser implementada ou utilizada no desenvolvimento de software.

Responsável:

-Designer de jogos

Descrição:

Os desenvolvedores e projetistas elaboram a arte final a partir da arte de conceito e outras iterações, incluindo especificações detalhadas, documentação técnica, design final e código-fonte. Este artefato foca em aspectos técnicos, como arquitetura, requisitos funcionais e não funcionais, e decisões de design. O objetivo da arte final é garantir que todos os requisitos e diretrizes sejam atendidos, proporcionando uma base sólida para a implementação do software.

### → Diagrama de Implantação

Um diagrama que mostra a disposição física dos componentes de software em um sistema, incluindo hardware e conexões.

Responsável:

Arquiteto

### → Documento de Feedback

Documento que coleta e registra o feedback dos usuários e stakeholders sobre o sistema, funcionalidades, desempenho e experiência de uso.

Responsável:  
Analista de Qualidade e Scrum Master

Descrição:

Este documento é utilizado para registrar e analisar o feedback obtido de diversas fontes, incluindo usuários finais, durante as fases de desenvolvimento e teste do sistema. O feedback é categorizado em positivo, negativo e sugestões de melhoria. Esse documento é crucial para guiar ajustes e melhorias contínuas no projeto, garantindo que o produto final atenda às expectativas dos usuários e stakeholders.

### → **Protótipo de Interface Gráfica**

Protótipo visual do sistema que apresenta a interface gráfica e a experiência do usuário (UX).

Responsável:  
Designer de Jogos

Descrição:


O protótipo de interface gráfica é uma representação visual interativa do sistema, incluindo telas, layouts, botões e outros elementos de interface. Ele é desenvolvido com ferramentas de design como Figma, Sketch ou Adobe XD e permite simular a navegação e a interação do usuário com o sistema. Esse artefato é essencial para validar o design, coletar feedback antecipado e fazer ajustes antes da implementação final.

### → **Plano de gerenciamento de configuração e mudança:**

Documento que descreve como gerenciar as configurações do software, incluindo controle de versão e gerenciamento de mudanças.

Responsável: Gerente de Configuração e Mudança

Descrição: O Plano de Gerenciamento de Configuração possui a descrição das atividades e tarefas do Gerenciamento de Controle de Configuração e Mudança que serão executadas durante o projeto, além de um padrão de commits.

Template:  Plano de Gerenciamento de Configuração e Mudança

### → **Cronograma**

Documento que detalha as atividades e marcos do projeto ao longo do tempo, incluindo datas de início e término, dependências e responsáveis.

Responsável:  
Líder de Projeto

Descrição:

O cronograma do projeto é um plano detalhado que organiza as tarefas e atividades em uma linha do tempo. Ele inclui datas específicas para o início e término de cada atividade, marcos importantes do projeto, dependências entre tarefas e os responsáveis por cada atividade. O cronograma é uma ferramenta essencial para o planejamento e acompanhamento do progresso do projeto, garantindo que todas as atividades sejam realizadas no tempo previsto e que os recursos sejam alocados de maneira eficiente.

### → Plano de Gerência de Comunicação

Documento que define como a comunicação será gerenciada ao longo do projeto, incluindo métodos, frequência, e públicos-alvo.

Responsável:  
Monitor

Descrição:

O plano de gerência de comunicação estabelece as estratégias e métodos para garantir uma comunicação eficaz entre todos os membros da equipe de projeto, stakeholders e outras partes interessadas. Ele especifica os canais de comunicação (e-mail, reuniões, ferramentas de colaboração), a frequência das comunicações, os responsáveis por comunicar diferentes tipos de informações e os públicos-alvo de cada comunicação. Este plano assegura que todas as partes envolvidas estejam bem informadas e alinhadas com os objetivos e progresso do projeto.

### → Plano de Gerência de Configuração

Documento que descreve como os itens de configuração do projeto serão identificados, controlados e auditados.

Responsável:  
Monitor

Descrição:

O plano de gerência de configuração define os processos e procedimentos para identificar, controlar, registrar e auditar todas as alterações nos itens de configuração do projeto. Isso inclui software, hardware, documentação e outros componentes críticos. O plano assegura que todas as mudanças sejam realizadas



de maneira controlada e registrada, minimizando os riscos de inconsistências e erros. Ele inclui políticas para controle de versão, procedimentos para solicitação de mudanças e auditorias regulares para garantir a conformidade com os requisitos de configuração.