

# Documentação Scrum

## Sistema Acadêmico Colaborativo com Apoio de IA - PIM

### 1. Introdução e Justificativa da Metodologia

O projeto PIM II, focado no desenvolvimento de um sistema acadêmico colaborativo com apoio de IA, adotou a metodologia ágil Scrum para planejar, organizar e acompanhar as etapas de desenvolvimento.

A escolha do Scrum se justifica pela necessidade de responder de maneira rápida e eficiente aos requisitos do projeto, priorizando a colaboração contínua e a entrega incremental de valor, em ciclos curtos de trabalho (Sprints), conforme os princípios da Engenharia de Software Ágil. O principal objetivo específico de gestão era “Aplicar a metodologia ágil Scrum na organização e acompanhamento das etapas de desenvolvimento”.

### 2. Papéis e Time Scrum

O Time Scrum foi composto pelos membros do grupo do PIM II e distribuídos:

Papel Scrum	Responsabilidade Principal	Membros do Time
<b>Product Owner (PO)</b>	Responsável pelo Product Backlog e alinhamento com as necessidades do cliente (definição de requisitos e prioridades).	Gabrielle Souza
<b>Scrum Master (SM)</b>	Garante a aplicação correta da metodologia, remoção de impedimentos e facilitação dos eventos.	Gabriel Sousa
<b>Time de Desenvolvimento</b>	Equipe responsável por desenvolver o produto de forma colaborativa e entregar o incremento funcional.	Arthur Ferreira, Felipe Faria, Santiago Pacheco e Vinícius Carvalho

### 3. Artefatos Scrum

#### 3.1. Product Backlog Inicial

O **Product Backlog** é a lista priorizada de todas as funcionalidades, requisitos e melhorias que o produto deve ter. Foi definido com base nos requisitos funcionais e não funcionais (SRS) e nas exigências obrigatórias do PIM, categorizado por área técnica.

<b>Categoria</b>	<b>Exemplo de Item do Backlog (Task Name)</b>	<b>Prioridade</b>
<b>Gestão do Projeto</b>	Definir tema e escopo, Levantar requisitos, Criar backlog inicial, Documentar processo ágil.	Alta
<b>Desenvolvimento em C</b>	Estruturas de decisão, Funções externas, Gravar dados em arquivos (.txt), Integração com Python.	Alta
<b>Desenvolvimento em Python</b>	Interface de cadastro (turmas/alunos/atividades), Buscas e relatórios, Integração com C, Interface para usuários.	Alta
<b>Inteligência Artificial</b>	Definir recurso IA (ex: recomendação/análise), Implementar IA, Integrar IA ao sistema.	Alta
<b>Modelagem e Diagramas</b>	DFD (Diagrama de Fluxo de Dados), Diagrama UML de Casos de Uso, Diagrama UML de Classes, Diagrama de Rede.	Alta
<b>Testes e Homologação</b>	Plano de testes, Testar cadastros, Testar integração C-Python, Testar em rede com usuários simultâneos.	Normal
<b>Sustentabilidade</b>	Documentar aplicação de educação ambiental, Contribuição ambiental (relatórios digitais).	Normal

## 4. Execução em Sprints

O desenvolvimento foi organizado em 3 Sprints, focadas na entrega de um incremento funcional e na consolidação da documentação exigida. O Scrum organiza o trabalho em ciclos curtos de 1 a 4 semanas.

### 4.1. Cadência do Ciclo (Time-Box)

O Time Scrum do PIM II estabeleceu uma cadência de 15 dias (duas semanas) para cada Sprint. Esta duração foi definida para equilibrar a complexidade do projeto, a necessidade de feedback rápido e a capacidade de entrega do time.

Sprint	Período Estimado	Objetivos e Entregáveis Principais (Incremento)	Eventos Registrados
<b>Sprint 1: Início e Planejamento</b>	01/09 a 15/09	Foco: Definição do escopo (SRS), Requisitos, Arquitetura e Fundamentação Teórica. Entregáveis: Repositório criado, Escopo definido, Referencial Teórico, SRS.	01/09/2025 (Sprint Planning); 15/09/2025 (Sprint Review/Retrospective).
<b>Sprint 2: Modelagem e Base do Código</b>	16/09 a 02/10	Foco: Modelagem do sistema e desenvolvimento dos módulos em C. Entregáveis: DFDs, Diagramas UML, Diagrama de Rede, Módulos críticos em C.	26/09/2025: Elaboração dos DFDs, Diagramas UML, Arquitetura e Diagrama de Rede; 02/10/2025 (Sprint Review).
<b>Sprint 3: Integração, IA e Testes Finais</b>	03/10 a 01/11	Foco: Implementação final em Python, Integração C-Python, Aplicação da IA, Testes e Documentação ABNT. Entregáveis: Código-fonte funcional (C/Python), Aplicação da IA, Testes em rede local, Plano de Testes/Homologação, Documento ABNT final.	Reuniões de acompanhamento (Daily Scrum/Encontros com Professores) realizadas ao longo do período.

## 5. Definições e Eventos Scrum Essenciais

Para garantir a qualidade e a transparência do processo, o time estabeleceu as seguintes definições:

### 5.1. Definition of Done (DoD - Definição de Pronto)

Um item do **Product Backlog** é considerado Pronto e apto para a Sprint Review somente quando todos os critérios a seguir forem atendidos:

- O código foi escrito e está em conformidade com os padrões de codificação (C/Python).
- Os testes de unidade e de integração (C-Python) foram executados com sucesso.
- O item foi testado e validado pelo Dev (Felipe) conforme o Plano de Testes.
- A funcionalidade está implementada e integrada ao sistema principal (Incremento).
- Os Diagramas (UML, DFD) ou a documentação associada ao item foram atualizados.
- O Product Owner (Gabrielle) aceitou formalmente o resultado.

### 5.2. Daily Scrum (Reunião Diária)

**Frequência e Duração:** Realizada diariamente (de segunda a sexta-feira) com duração máxima de 15 minutos.

**Objetivo:** O Time de Desenvolvimento se reúne para inspecionar o progresso em direção ao Objetivo da Sprint e adaptar o Sprint Backlog conforme necessário.

**Formato:** A reunião se baseava nas três perguntas chaves, focadas no quadro

**Kanban:** o que realizei ontem, o que farei hoje e quais impedimentos estão atrapalhando.

### 5.3. Ação da Retrospectiva

O evento de Sprint Review/Retrospectiva da Sprint 1 resultou em uma ação chave de melhoria:

- **Ação para a Sprint 2:** Dedicar tempo no Sprint Planning para quebrar os itens (como “Modelagem” e “Desenvolvimento em C”) em tarefas menores e mais gerenciáveis, facilitando o rastreo no Kanban e a aplicação do Definition of Done (DoD).

## 6. Sprint Backlog e Gerenciamento Visual (Kanban)

O **Sprint Backlog** é o subconjunto de itens do Product Backlog selecionados para serem concluídos em uma Sprint. O gerenciamento das tarefas foi feito de forma visual, utilizando um quadro Kanban virtual.

### Estrutura do Quadro Kanban

Coluna	Descrição
<b>To Do</b>	Itens do Backlog selecionados para a Sprint, mas ainda não iniciados.
<b>Doing</b>	Item sendo ativamente desenvolvido ou trabalhado por um membro do time.
<b>Done</b>	Item que atendeu à Definition of Done (DoD) e está pronto para o Incremento.

### 6.1. Detalhe da Sprint Backlog 2

O plano da Sprint 2 (16/09 a 02/10) priorizou a modelagem, a base técnica em C e os diagramas exigidos:

Item (Task Name)	Categoria	Assignee	Status (Conclusão)	Referência de Evidência
Diagrama UML de Casos de Uso	Modelagem	Gabrielle	Done	[Diagramas UML - PIMII.asta]
Diagrama UML de Classes	Modelagem	Gabrielle	Done	[Diagramas UML - PIMII.asta]
DFD (Diagrama de Fluxo de Dados)	Modelagem	Gabrielle	Done	Registro: 26/09/2025
Gravar dados (.csv)	Desenv. C	Gabriel	Done	Módulo cadastro.c
Ler dados (.csv)	Desenv. C	Gabriel	Done	Módulo consulta.c

## 7. Incremento e Resultados por Sprint

O Incremento é a soma de todos os itens do Sprint Backlog completados na Sprint e os incrementos de todas as Sprints anteriores, representando a entrega de um produto potencialmente utilizável e de valor.

Sprint	Período	Incremento Entregue (Valor Agregado)	Evidências e Documentação
<b>Sprint 1</b>	01/09 a 15/09	Fundamentação e Escopo: Definição do problema, Stakeholders, Especificação de Requisitos de Software (SRS) e Product Backlog inicial.	[SRS - PIMII.pdf] e [PIMII.docx] (Referencial Teórico)
<b>Sprint 2</b>	16/09 a 02/10	Modelagem e Base de Dados: Criação dos fluxos e arquitetura do sistema (DFDs, Diagramas UML de Classes/Casos de Uso), e desenvolvimento dos módulos em C para manipulação de dados em arquivos (Backend de persistência).	[Diagramas UML - PIMII.asta] e [PIMII.docx] (registro de validação 02/10)
<b>Sprint 3</b>	03/10 a 01/11	Sistema Funcional e Testado: Finalização da interface gráfica em Python, integração total C-Python, implementação da funcionalidade de Inteligência Artificial (RF05), testes em rede e homologação final.	[initial_scrum.csv] (Testes em Rede/IA), [PIMII.docx] (Resultados e Conclusão), Registro

## 8. Métricas e Conformidade no Processo

### 8.1. Métricas de Desempenho do Time (Velocity e Forecast)

Para medir a capacidade de entrega e a previsibilidade do projeto, o time utilizou o conceito de Velocity (Velocidade) para estimar a capacidade de concluir os itens do Backlog.

- **Velocity (Capacidade Média):** Baseado na conclusão das Sprints 1 e 2, o Time de Desenvolvimento atingiu uma capacidade média de entrega de 5 a 6 itens do Backlog por Sprint.
- **Propósito:** Esta métrica foi fundamental para guiar o Sprint Planning da Sprint 3, garantindo que apenas a quantidade de trabalho factível fosse selecionada para ser concluída, priorizando a Integração C-Python e a Implementação da IA.

### 8.2. Sustentabilidade e Requisitos Não Funcionais (RNF)

A documentação ágil também aborda os Requisitos Não Funcionais (RNF):

RNF	Aplicabilidade no Processo Scrum	Evidência/Ação
<b>RNF04 (Scrum)</b>	O processo foi documentado e executado conforme a metodologia Scrum (ver seções 1-7).	Esta documentação, registros de encontros.
<b>RNF09 (Sustentabilidade)</b>	O uso do Scrum reduziu o desperdício de papel (documentação digital) e permitiu a priorização da task "Documentar aplicação de educação ambiental" no Backlog.	Backlog priorizado e registro do tema de sustentabilidade no PIM.
<b>RNF07/RNF08 (Segurança/LGPD)</b>	Os requisitos de segurança e controle de acesso foram incorporados como itens de alta prioridade no Product Backlog e validados durante a Sprint 3.	Testar cadastros e Plano de testes.