

SPRINT REPORT — SPRINT 1

Equipe:

1. Informações Gerais

Nome do Projeto	D-Lemma
Período da Sprint 1	17/04/25 - 30/04/25
Data de Início	17/04/25
Data de Término	30/04/25
Responsável pelo Preenchimento (Scrum Master ou outro)	Jefferson Antônio Tanajura Silva & Carlos Gil Martins da Silva
Link do repositório do github	Velha Guarda

Membros da equipe (em ordem alfabética) e sua função no desenvolvimento:

NOME	FUNÇÃO
Carlos Gil Martins da Silva	Desenvolvedor Full Stack; Scrum Master e Designer
Antônio Vinícius Silva Dutra	Desenvolvedor Backend
Jeffeson Antônio Tanajura Silva	Desenvolvedor Backend
Kaian de Souza Gonçalves	Desenvolvedor FullStack
Melk Silva Braga	Desenvolvedor Backend
Yago Oliveira Silva	Desenvolvedor FullStack

2. Objetivo Principal da Sprint 1

Descrição:	Essa Sprint marca o começo da estrutura básica do produto. Será feita a modelagem do banco de dados, desenvolvimento do design inicial e criação da página de cadastro.
------------	---

3. Atividades-chave e resultados

(1) Definição da visão do produto

Principais decisões tomadas	Foi decidido começar pela parte conceitual, criando primeiro uma identidade visual para o D-Lemma. As estimativas das tarefas serão feitas com base na quantidade de tempo necessário para cumpri-las, de maneira análoga ao Scrum Point.
Pendências/Ações futuras	

(2) Formação e alinhamento do time

<i>Working agreements</i> (acordos internos)	A daily é realizada a partir das 19h. As ferramentas de comunicação utilizadas são Whatsapp e Discord. O horário de trabalho é flexível, desde que sejam cumpridos os prazos estabelecidos.
Nivelamento em <i>Scrum</i> (se necessário)	
Pendências/Ações futuras	Será necessário treinamento em tecnologias como FIGMA, Next.js e Spring Boot.

(3) Identificação e priorização do *backlog*

<i>Backlog</i> inicial criado	<p>Autenticação e cadastro de usuários: Como usuário, quero fazer login usando meu e-mail e senha, para acessar a plataforma; Como usuário, quero redefinir minha senha caso a esqueça, para recuperar meu acesso à plataforma.</p> <p>Sistema de chat/discussão: Como usuário, quero participar de um chat em tempo real, para debater com os outros usuários o dilema apresentado; Como professor, quero moderar mensagens (editar, apagar ou sinalizar), para manter a</p>
-------------------------------	---

	ordem da discussão.
Técnica de priorização utilizada	A decisão do que será priorizado é tomada pelo Scrum Master se baseando em criar uma estrutura básica minimamente funcional.
Estimativa iniciada	A Estimativa é feita com base no tempo necessário para realizar as tarefas. Esse tempo será configurado nas tarefas em andamento no Jira com a conformidade dos outros membros do projeto.
Pendências/Ações futuras	

(4) Planejamento de arquitetura e infraestrutura

Arquitetura definida	Monolito Modularizado: Ideal para a fase inicial do projeto (MVP), pela simplicidade de deploy e manutenção. Permite futura transição para microsserviços, caso necessário.
Ferramentas de desenvolvimento/ versionamento	Visual Studio Code, GitHub, Figma, Next.js e Spring Boot
Configuração de ambientes	A organização do Github já foi criada com as suas devidas branches
Pendências/Ações futuras	Integração do Banco de Dados com a aplicação; Configurar requisições para enviar os dados do formulário ao backend.

(5) Planejamento de qualidade e testes

Estratégia de testes	<p>GitHub Actions para rodar testes automáticos;</p> <p>Testes de integração entre backend, banco de dados e APIs;</p> <p>Testes funcionais com simulações reais do usuário.</p>
Métricas de qualidade planejadas	<p>Detectar vulnerabilidades, code smells, duplicações;</p> <p>Tempo de resposta das funcionalidades;</p> <p>Falhas detectadas por tipo e camada.</p>
Pipeline de CI/CD inicial	<p>Pipeline Automatizado (GitHub Actions/GitLab CI)</p> <p>Etapa 1: Build do projeto</p> <p>Etapa 2: Execução de testes unitários e geração de cobertura</p> <p>Etapa 3: Lint + análise estática</p> <p>Etapa 4: Deploy automático para ambiente de testes (opcional)</p>
Pendências/Ações futuras	<p>Testar validações de campos (nome vazio, e-mail inválido, senhas diferentes, função não selecionada).</p>

(6) Riscos e gestão de dependências

Riscos Identificados	<p>Tecnológico: incompatibilidade entre bibliotecas utilizadas no front-end e back-end.</p> <p>Recursos Humanos: Indisponibilidade de membros da equipe por motivos de saúde ou agenda.</p> <p>Prazo: Atrasos em entregas devido a tarefas não previstas ou subestimadas.</p>
Plano de mitigação	<p>Tecnológico: adotar bibliotecas amplamente documentadas e com boa comunidade de suporte.</p> <p>Recursos Humanos: Manter comunicação constante entre os membros da equipe,</p>

	<p>encontrando soluções para quaisquer empecilhos que dificultam o andamento do projeto.</p> <p>Atrasos nos prazos: dividir entregas em incrementos menores e mais frequentes.</p>
Dependências externas	<p>Servidor para hospedagem da aplicação;</p> <p>API para cadastro de usuários no Next.js.</p>
Pendências/Ações futuras	Disponibilidade do servidor pelo CIPEC.

(7) Planejamento de lançamentos e *roadmap* macro

Visão geral das 4 Sprints Finais (Planejamento Macro)	<p>Protótipo funcional da interface;</p> <p>Backend e integração inicial;</p> <p>Funcionalidades completas e testes;</p> <p>Refinamento e entrega para homologação.</p>
Marcos (milestones) importantes	<p>Protótipo funcional: interface navegável com fluxos principais.</p> <p>API funcional integrada: Backend em produção parcial.</p> <p>MVP completo: Sistemas com as principais funcionalidades.</p> <p>Entrega para avaliação/homologação: Versão final para revisão por stakeholders.</p>
Alinhamento com <i>stakeholders</i>	
Pendências/Ações futuras	Definição de data para entrega do MVP.

(8) Critérios de pronto (Definition of Done - DoD) e critérios de aceite

Definition of Done (DoD)	<ul style="list-style-type: none">• Código desenvolvido, testado e revisado (Code Review).• Cobertura de testes automatizados adequados (mínimo 70% em código crítico).• Testes manuais realizados (quando aplicável).• Build sem erros e deploy funcional no ambiente de testes.• Documentação atualizada (código, README, Wiki se necessário).• Interface testada e aprovada nos navegadores/sistemas suportados.• Critérios de aceitação específicos da história atendidos.• Nenhum bug crítico pendente.
Critérios de aceite (Gerais) adotados	Segue Link com tabela dos principais critérios de aceites de cada tarefa: Anexos
Pendências/Ações futuras	

(9) Prototipagem ou Prova de Conceito - PoC (se aplicável)

Objetivo do protótipo/PoC	Validar pontos críticos da solução, especialmente relacionados à tecnologia de chat em tempo real, autenticação segura e visualização responsiva.
Resultados obtidos	Confirmação da viabilidade técnica dos principais recursos e feedback do time e stakeholders sobre a experiência do usuário e tecnologias adotadas.
Pendências/Ações futuras	Ajustes na aplicação a depender do feedback recebido.

(10) Organização de fluxos de trabalho

Documentação e <i>Wiki</i>	Documentação no GitHub em código e README.

Pendências/Ações futuras	Versionamento de código e adoção do princípio KISS.

4. Conclusão e *status* final da *Sprint* 1

O que foi concluído com sucesso?	<ol style="list-style-type: none">1. Criação da identidade visual do D-Lemma.2. Criação do repositório GitHub.3. Retrospectiva SPRINT 1 (Disciplina Desenvolvimento de Software).4. Estudo da tecnologia Spring Boot.5. Estudo da tecnologia Next.js.
Desvios ou itens não finalizados	Ficou em aberto a tarefa de desenvolvimento inicial do Design do site no figma, essa atividade não pôde ser concluída devido a fatores externos como (Demais disciplinas da faculdade e afins) que impediram a realização dela.

Lições aprendidas	Tarefas podem acabar tendo seu nível de dificuldade subestimado. Por isso é sempre importante ouvir a opinião de todos sobre a dificuldade de uma tarefa específica.
Próximos passos e preparação para a <i>Sprint 2</i>	Priorizar as tarefas que não puderam ser finalizadas na sprint anterior.

5. Aprovações e revisão

- ***Product Owner:***

Gidevaldo Novais Dos Santos

- ***Scrum Master:***

Carlos Gil Martins da Silva

- **Data de Revisão e Aprovação:**

Anexos (se forem necessários)

- **Documentos referenciados:**

- **Prototipação de telas:**

As telas ainda estão em processo de desenvolvimento.

- **Lista detalhada do *backlog*:**

[Anexos](#)

Observações finais:

1. Intensificar o treinamento nas tecnologias chave (Next.js, Spring Boot, Figma);
2. Garantir o cumprimento das entregas essenciais pendentes, como a prototipação visual;
3. Definir com clareza os marcos de entrega com stakeholders, especialmente a data do MVP.

