

SoftFit

# **VISÃO DO PRODUTO E PROJETO**

Versão [\[1.0\]](#)

## Histórico de Revisão

Data	Versão	Descrição	Autor
31/01/2022	1.0	Primeira versão do documento. Em que todos os quadros, tanto de problema, quanto de produto foram preenchidos. A metodologia escolhida e suas justificativas também são adicionadas pela primeira vez no documento. As lições aprendidas da Unidade 1 são descritas nessa versão do documento.	Victor Hugo Oliveira Leão; Felipe Alef Pereira Rodrigues

## Sumário

<b>1</b>	<b><i>VISÃO GERAL DO PRODUTO .....</i></b>	<b><i>4</i></b>
1.1	<b>Declaração do Problema.....</b>	<b>4</b>
1.2	<b>Declaração de Posição do Produto .....</b>	<b>4</b>
1.3	<b>Objetivos do Produto.....</b>	<b>5</b>
<b>2</b>	<b><i>ABORDAGEM DE DESENVOLVIMENTO DE SOFTWARE .....</i></b>	<b><i>5</i></b>
2.1	<b>Metodologia .....</b>	<b>5</b>
2.2	<b>Processo.....</b>	<b>6</b>
2.3	<b>Procedimentos .....</b>	<b>5</b>
<b>3</b>	<b><i>LIÇÕES APRENDIDAS .....</i></b>	<b><i>7</i></b>
3.1	<b>Unidade 1 .....</b>	<b>7</b>
3.2	<b>Unidade 2 .....</b>	<b>7</b>
3.3	<b>Unidade 3 .....</b>	<b>7</b>
3.4	<b>Unidade 4 .....</b>	<b>7</b>
3.5	<b>Unidade 5 .....</b>	<b>7</b>
<b>4</b>	<b><i>REFERÊNCIAS BIBLIOGRÁFICAS.....</i></b>	<b><i>7</i></b>

# VISÃO DO PRODUTO E PROJETO

## 1 VISÃO GERAL DO PRODUTO

### 1.1 Declaração do Problema

O problema	É a falta de uma solução digital e consolidada para o gerenciamento das questões humanas da academia. A partir disso, foi identificada a oportunidade de desenvolvimento de um software para o nicho de academias.
Afeta	O administrador, os alunos e os profissionais (professores/Personal Trainers).
Cujo impacto é	Para o administrador: a falta de concentração dos dados financeiros e de controle de frequência de seus alunos matriculados em uma única plataforma; a dificuldade de comunicação imediata e automática de avisos para com os alunos. Para os alunos: terem suas avaliações e treinos registrados em fichas físicas, que podem ser facilmente perdidas ou danificadas; a necessidade de se dirigir a uma unidade física da academia para resolver problemas financeiros ou solucionar dúvidas ou consultas simples. Para os profissionais: não possuírem sua rotina e ferramentas de gerenciamento e auxílio de alunos em um único lugar.
Uma solução de sucesso seria	Um software em que os alunos pudessem administrar seus treinos e avaliações, resolver questões que envolvam seu estado financeiro e entrar em contato com a academia. Para o administrador, uma solução de sucesso seria aquela em que os principais dados dos alunos e professores estivessem concentrados e que o possibilitaria com poucos cliques enviar avisos para os alunos por e-mail. Já para os professores, ter sua rotina em uma única tela juntamente com a possibilidade de auxiliar os alunos de forma on-line.

### 1.2 Declaração de Posição do Produto

Para	Academias
Que	Buscam uma solução em que possam realizar o gerenciamento de questões humanas e entregarem uma experiência digital aos alunos e professores
O SoftFit	É uma aplicação Web responsiva
Que	Entrega soluções digitais para as 3 partes: administrador, aluno e professor.
Ao contrário	Das academias tradicionais em que: <ol style="list-style-type: none"><li>1. os dados, treinos e avaliações do aluno são disponíveis de forma física;</li><li>2. Os alunos precisam se locomover até uma unidade física para solucionar problemas financeiros e tirarem dúvidas;</li><li>3. Os professores só conseguem criar e administrar treinos se estiverem com os alunos ao seu lado;</li><li>4. Os administradores não tem os principais dados dos alunos concentrados e com a disponibilidade de entrarem em contato de forma imediata e automática acerca de avisos importantes;</li></ol>
Nosso produto	Possui as seguintes funções: <ol style="list-style-type: none"><li>1. Para o administrador, há a disposição dos dados, tanto pessoais e financeiros, quanto de frequência de seus alunos matriculados.</li></ol>

	<p>Além disso, o administrador também consegue enviar avisos importantes e entrar em contato por e-mail com os frequentadores da academia.</p> <p>2. Para os alunos, haverá seus treinos e avaliações (medidas) em uma única página (com a possibilidade de vídeos demonstrativos), também é possível entrar em contato com a academia por e-mail e chats para poderem tirar suas dúvidas e controlarem seu estado financeiro.</p> <p>3. Para os professores terão sua rotina disposta e ferramentas de criação e auxílio de treinos a partir dos dados dos alunos.</p>
--	---

### 1.3 Objetivos do Produto

*O principal objetivo do produto é proporcionar uma experiência digital intuitiva e consolidada para os alunos, professores e administrador da academia.*

## 2 ABORDAGEM DE DESENVOLVIMENTO DE SOFTWARE

### 2.1 Metodologia

*Para o estabelecimento da metodologia de desenvolvimento de software, algumas questões foram previamente respondidas:*

- **Questões Técnicas:**
  - **Qual é o tamanho do sistema que está sendo desenvolvido?** Relativamente extenso, uma vez que envolve funções para o aluno, o professor e o administrador da academia.
  - **Que tipo de sistema está sendo desenvolvido?** Aplicação Web responsiva.
  - **Qual a vida útil prevista para o sistema?** 4 meses (tempo da disciplina).
  - **O sistema está sujeito a controle externo?** Sim, o dono/administrador da academia adquiriria o software, logo o produto não estaria somente sob influência dos desenvolvedores.
- **Questões Humanas:**
  - **Qual é o nível de competência dos projetistas e programadores do time de desenvolvimento?** A princípio, todos que compõem o time possuem um nível de competência igualmente básico-intermediário.
  - **Como está organizado o time de desenvolvimento?** Não há uma divisão concreta além de Product Owner e desenvolvedores.
  - **Quais são as tecnologias disponíveis para apoiar o desenvolvimento do sistema?** Trello, Discord, Microsoft Teams e GitHub.
- **Questões Organizacionais:**
  - **É importante ter uma especificação e um projeto (design) bem detalhados antes de passar para a implementação - talvez por motivos contratuais?** De certa forma, não é necessário uma detalhação de especificações e design, apesar de haver um prévio conhecimento do que será construído e algumas funções do projeto.
  - **É realista uma estratégia de entrega incremental, na qual o software é entregue aos clientes ou outros stakeholders e um rápido feedback é obtido?** Considerando que teremos um Product Owner, isto é, alguém que trabalharia analisando as entregas do grupo e ver se está de acordo com o pensado, é sim realista.
  - **Os representantes do cliente estarão disponíveis e dispostos a participar do time de desenvolvimento?** Basicamente, a equipe terá um “representante de cliente”, então sim.
  - **Existem questões culturais que possam afetar o desenvolvimento do sistema?** Não

A partir das respostas desenvolvidas, considera-se que por não haver uma necessidade da criação de documentos formais sendo criados dentro das atividades e, uma vez que os requisitos e o projeto serão desenvolvidos conjuntamente, uma **abordagem ágil** seja ideal.

Sendo assim, a metodologia ágil **Kanban** foi escolhida a partir dos seguintes motivos:

- Seguir uma filosofia contínua de desenvolvimento e entrega e, portanto, admitir uma maior flexibilidade e leveza para os desenvolvedores;
- Ter uma organização visível clara que possibilita ao grupo, sem a necessidade de reunião constantes, entender os processos e em que etapa o desenvolvimento está como um todo;
- Por não ter a necessidade de uma divisão de funções específicas, o grupo pode, como um todo, se auto-organizar e trabalhar com maior autonomia;
- A priorização de atividades é bem intuitiva com o quadro e com a limitação de WIP (trabalhos em andamento).

Dessa forma, apesar dos horários rígidos que os integrantes do grupo, como alunos, possam ter, com o Kanban, o grupo pode trabalhar de forma ágil, mas sem a necessidade de comprometer diários de horários.

## 2.2 Processo e Procedimentos

<i>Atividade</i>	<i>Objetivo</i>	<i>Papel</i>	<i>Método</i>	<i>Ferramenta</i>
Criar quadro Kanban	Organizar o quadro que a equipe utilizará para se orientar durante o desenvolvimento	Product Owner e Equipe de Desenvolvimento	Requisitos – Em desenvolvimento - Entrega	Trello
Estabelecer o prazo de entregas	Organizar os requisitos a serem entregues em relação ao tempo	Product Owner e Equipe de Desenvolvimento	Analisar datas da disciplina com os requisitos definidos	Trello
Estabelecer escopo do produto	Definir as características que devem estar presentes no produto de software	Product Owner e Equipe de Desenvolvimento	EAP – Estrutura Analítica de Projeto	Trello
Estudar tecnologias	Adquirir o conhecimento necessário para o desenvolvimento do sistema	Equipe de Desenvolvimento	Individual (Cada integrante seguirá o método de estudo que achar mais eficiente)	Materiais digitais
Definir design das interfaces	Criar um modelo das interfaces que serão desenvolvidas	Product Owner e Equipe de Desenvolvimento	Pesquisa de mercado e Boas práticas de UI e UX.	Canva
Codificação das Interfaces (Front-End)	Desenvolver as telas que os usuários do sistema terão acesso	Equipe de Desenvolvimento	Seguir o design das interfaces feito previamente	Linguagens HTML5/CSS3
Codificação das funções do sistema (Back-End)	Desenvolver as funcionalidades do programa em si	Equipe de Desenvolvimento	Desenvolvimento dos requisitos definidos no escopo do produto.	Python e Django
Tornar o produto responsivo	Tornar o produto utilizável tanto no desktop, quanto no dispositivo	Desenvolvedor	Utilização do PWA	PWA

	<i>móvel</i>			
<i>Realizar Versionamento</i>	<i>Acompanhar o progresso do projeto</i>	<i>Equipe de Desenvolvimento</i>	<i>Controlar as versões no GitHub</i>	<i>GitHub</i>
<i>Realizar Teste Unitário</i>	<i>Encontrar possíveis falhas no software</i>	<i>Desenvolvedor</i>	<i>Criação de classes de teste</i>	<i>PyUnit</i>
<i>Realizar Refatoração</i>	<i>Melhorar a qualidade do código</i>	<i>Equipe de Desenvolvimento</i>	<i>Extração de método, extração de classe, renomeação de método, classe ou variável, extração de interface</i>	<i>GitHub</i>
<i>Entregar produto final</i>	<i>Entregar o software em funcionamento para o cliente final</i>	<i>Product Owner e Equipe de Desenvolvimento</i>	<i>Hospedagem do produto final na Internet</i>	<i>GitPages</i>

### 3 LIÇÕES APRENDIDAS

#### 3.1 Unidade 1

A unidade 1 trouxe diversas lições e aprendizagens no que convém ao começo de um projeto e conceitos iniciais de Engenharia de Software. Entretanto, podemos destacar dois pontos que tiveram maior impacto no grupo como um todo:

- Primeiramente, a lição referente à escolha da metodologia de desenvolvimento de software. As questões levantadas por Sommerville ajudaram de forma substancial na determinação da abordagem ágil para a realização do projeto, uma vez que a equipe, a priori, não sabia como justificar de forma correta a escolha de uma abordagem e não de outra.
- O esclarecimento do que é o Product Owner, quais são suas funções e características também teve grande impacto no grupo, uma vez que pudemos escolher um integrante para trabalhar na função de P.O., o que acreditamos que será de grande ajuda durante todo o projeto.

#### 3.2 Unidade 2

#### 3.3 Unidade 3

#### 3.4 Unidade 4

#### 3.5 Unidade 5

### 4 REFERÊNCIAS BIBLIOGRÁFICAS

1. Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira.