

# Projeto de Bloco: Engenharia Disciplinada de Softwares

Engenharia Disciplinada de Softwares

**AT**



**Instituto Infnet**

Antonio Eduardo Silveira Demarchi

Professor: Armênio Cardoso

[23E4\_5]

**Rio de Janeiro**

**2024**

## Sumário

<b>Capa do Projeto</b> .....	<b>1</b>
<b>Índice</b> .....	<b>2</b>
<b>Especificação</b> .....	<b>3</b>
Visão do Projeto.....	3.1
Contexto e Problema.....	3.2
Controle de Versão.....	3.3
<b>Escopo</b> .....	<b>4</b>
Introdução.....	4.1
Situação.....	4.2
Justificativa.....	4.3
Público-Alvo.....	5
Riscos para o Negócio.....	5
Nomenclatura.....	5
<b>Funcionalidades Principais</b> .....	<b>6</b>
<b>Requisitos Funcionais</b> .....	<b>7</b>
<b>Análise</b> .....	<b>10</b>
Diagrama Caso de Uso.....	10
Modelo de Dominio.....	11
Diagrama de Classe.....	12
<b>Arquitetura</b> .....	<b>13</b>
Requisitos Não Funcionais.....	13
Stack Tecnologia.....	14
Desenho da Arquitetura da Aplicação.....	14
Mínimo Produto Viável - MVP.....	15
<b>Projeto</b> .....	
Tarefas.....	xx
Diagrama de Sequência.....	16
Protótipo de Telas.....	17
<b>Construção</b> .....	
Plano do Projeto.....	20
Repositório do GitHub.....	21
TDD.....	22
<b>Testes</b> .....	<b>25</b>
Testes unitários orientados ao TDD.....	xx
Plano de testes exploratórios.....	xx
<b>Plano de Implantação:</b> .....	<b>24</b>

# ESPECIFICAÇÃO DO PROJETO

<b>Nome</b>	DevDelivery - Conectando Estudantes e Investidores
<b>Autor</b>	Antonio Eduardo Silveira Demarchi
<b>Bloco</b>	Engenharia Disciplinada de Softwares
<b>Área</b>	Negócio Educacional

## Contexto e Problema

<b>Problema</b>	Empresas buscam talentos e muitas vezes não sabem onde encontrar os melhores estudantes.
<b>Contexto</b>	Networking - Inovação Tecnológica Brasileira Espaço virtual que reúna os melhores trabalhos e projetos desenvolvidos por estudantes de engenharia e suas expectativas.

## CONTROLE DE VERSÕES

N Versão	Data de revisão	Resumo das alterações	Modificado por
1.0	24/10/2023	Criação	Antonio Eduardo S. Demarchi
1.1	18/11/2023	Formatação	Antonio Eduardo S. Demarchi
1.2	26/02/2024	Contexto	Antonio Eduardo S. Demarchi
1.3	18/03/2024	Visão e correções	Antonio Eduardo S. Demarchi
1.4	03/04/2024	Melhorias nos Diagramas	Antonio Eduardo S. Demarchi

## Escopo do Projeto

### Introdução

O Devlivery é um ambiente colaborativo que visa conectar estudantes de engenharia com potenciais investidores de maneira eficaz.

Os projetos têm a responsabilidade de apresentar o aluno, gerando possíveis propostas de parceria.

Permitindo que os alunos mantenham um portfólio de produtos e um perfil com suas principais habilidades.

Ganhando experiência prática e obtendo reconhecimento de seu trabalho.

Os projetos podem variar desde o desenvolvimento de aplicativos, produtos inovadores, pesquisas aprofundadas até sistemas complexos para diversas áreas de negócio.

Através de seu perfil os estudantes não apenas ganham exposição para seus projetos, mas também têm a oportunidade de colaborar com outros, aprender com especialistas do setor e potencialmente garantir financiamento para suas ideias.

### Situação

A Engenharia de Software é uma área em constante evolução, caracterizada por um ambiente altamente competitivo e dinâmico. Nesse cenário, muitos estudantes encontram desafios para evidenciar suas competências técnicas e se destacar entre seus pares.

A demanda por profissionais qualificados na área está em ascensão, tornando essencial a existência de um portfólio robusto que vá além das plataformas convencionais, como o GitHub.

### Justificativa:

**Necessidade de Exposição:** Os estudantes precisam de uma plataforma para exibir seus projetos, habilidades e conquistas de forma organizada e acessível.

**Incentivar a Participação em Projetos:** Estimular os estudantes a participarem de projetos acadêmicos, extracurriculares e de pesquisa, promovendo a colaboração, a inovação e o crescimento contínuo.

**Propostas de trabalho:** Um portfólio bem elaborado pode aumentar suas chances de conseguir estágios remunerados, bolsas ou oportunidades de trabalho freelance.

**Fornecer Orientações de Arquitetura:** Com o auxílio do professor os alunos terão um estímulo a desenvolver projetos com arquiteturas modernas, design modular e melhores práticas.

### **Público-Alvo**

- Estudantes [Alunos] - Aluno do curso de Engenharia de Software.
- Investidores [Contratante, Empresas, Recrutadores] - em busca de talento;
- Professores e Instituto [Admin] - Usuário administrador do Sistema;

### **Riscos para o Negócio**

- **Prazo de entrega curto:** Desenvolvimento de uma solução completa leva tempo.
- **Adoção do Usuário:** Como a plataforma é voltada para um público específico (estudantes de engenharia de software), pode haver resistência ou falta de interesse em adotar a plataforma. Isso pode ser devido à falta de conhecimento sobre a plataforma, falta de tempo ou simplesmente preferência por outras plataformas já estabelecidas.
- **Financiamento:** A construção e manutenção de uma plataforma digital pode ser cara. Se o financiamento não for suficiente, pode haver dificuldades para manter a plataforma funcionando de maneira eficaz.
- **Segurança de Dados:** Como a plataforma lida com informações pessoais e projetos dos estudantes, a segurança dos dados é uma preocupação importante.
- **Qualidade dos Projetos:** A qualidade dos projetos postados na plataforma é importante para atrair investidores.
- **Engajamento dos investidores:** A plataforma depende do interesse e do engajamento dos investidores. Se os investidores não estiverem interessados nos projetos ou se não virem valor na plataforma, isso pode limitar o sucesso da plataforma.

### **Nomenclatura:**

RF = Requisito Funcional (É o conjunto de entradas, seu comportamento e sua saída, ou seja, envolve cálculos, lógicas de trabalho, manipulação e processamento de dados, entre outros).

# Funcionalidades Principais

<b>Projetos</b>	<ul style="list-style-type: none"><li>- Publicar projeto</li><li>- Deletar projeto</li><li>- Excluir projeto</li><li>- Filtrar projetos</li></ul> <p>(título, objetivo, descrição, links,,ator,data)</p>
<b>Proposta</b>	<ul style="list-style-type: none"><li>- Enviar proposta</li><li>- Obter lista de propostas</li><li>- Responder proposta</li><li>- Solicitar Contato</li></ul> <p>(nome, detalhes da proposta,tempo estimado, valor)</p>
<b>Negócio</b>	<ul style="list-style-type: none"><li>- Listar negócio</li><li>- Criar negócio</li><li>- Gerenciar negócio</li><li><ul style="list-style-type: none"><li>- Listar serviço</li><li>- Cadastrar serviço</li><li>- Deletar serviço</li></ul></li><li><ul style="list-style-type: none"><li>- Listar produto</li><li>- Cadastrar produto</li><li>- Deletar produto</li></ul></li></ul>

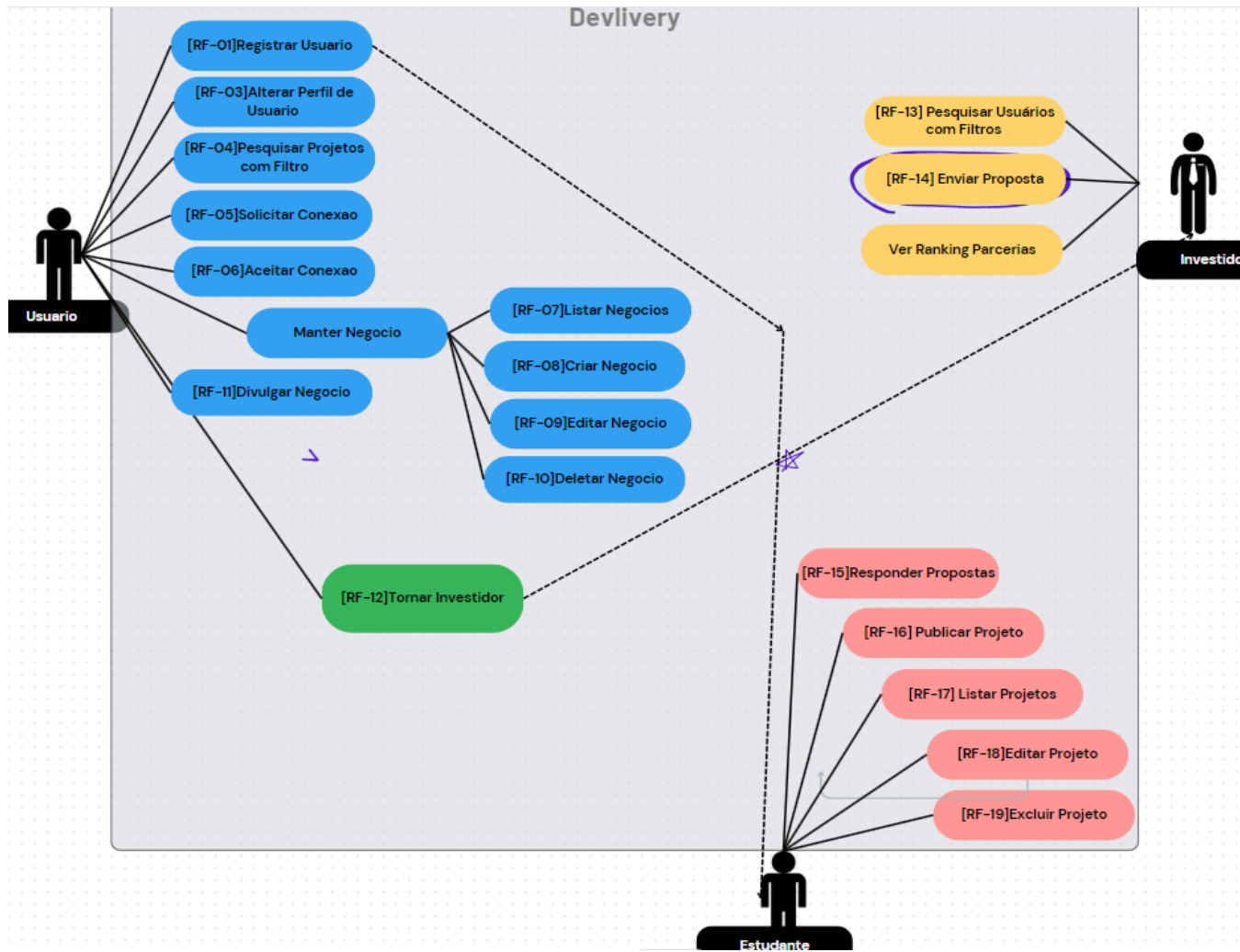
## Requisitos Funcionais

ID	Descrição	Nível de Complexidade	Prioridade
[RF-01]	A plataforma deve permitir que o usuário crie um perfil informando um email válido (verificação de email para alunos @al.infnet.) e uma senha segura.	4	Essencial
[RF-02]	A plataforma deve permitir que faça login e acesse seu perfil.	2	Essencial
[RF-03]	A plataforma deve permitir que o usuário altere informações em seu perfil.	2	Desejável
[RF-04]	A plataforma deve permitir que o usuário pesquise projetos com filtro avançado.	2	Essencial
[RF-05]	A plataforma deve permitir que o usuário solicite conexão a outro.	4	Desejável
[RF-06]	A plataforma deve permitir que o usuário aceite conexões na rota de solicitações.	3	Essencial
[RF-07]	A plataforma deve permitir a exibição da lista de negócios com ordenação.	2	Importante

<b>[RF-08]</b>	A plataforma deve permitir que o usuário cadastre um negócio.	4	Desejável
<b>[RF-09]</b>	A plataforma deve permitir a o usuário editar seus negócios.	4	Desejável
<b>[RF-10]</b>	A plataforma deve permitir que o usuário delete seus negócios.	5	Desejável
<b>[RF-11]</b>	A plataforma deve permitir que o usuário divulgue seus negócios.	4	Desejável
<b>[RF-12]</b>	A Plataforma deve oferecer a opção de se tornar um Investidor.	2	Essencial
<b>[RF-13]</b>	Pesquisar usuários com filtro avançado.	2	Essencial
<b>[RF-14]</b>	A plataforma deve permitir que o Investidor faça proposta de parceria a um estudante.	4	Desejável
<b>[RF-15]</b>	A plataforma deve permitir que o estudante responda suas propostas. Caso aceite, incrementa pontos no perfil do estudante e	3	Desejável

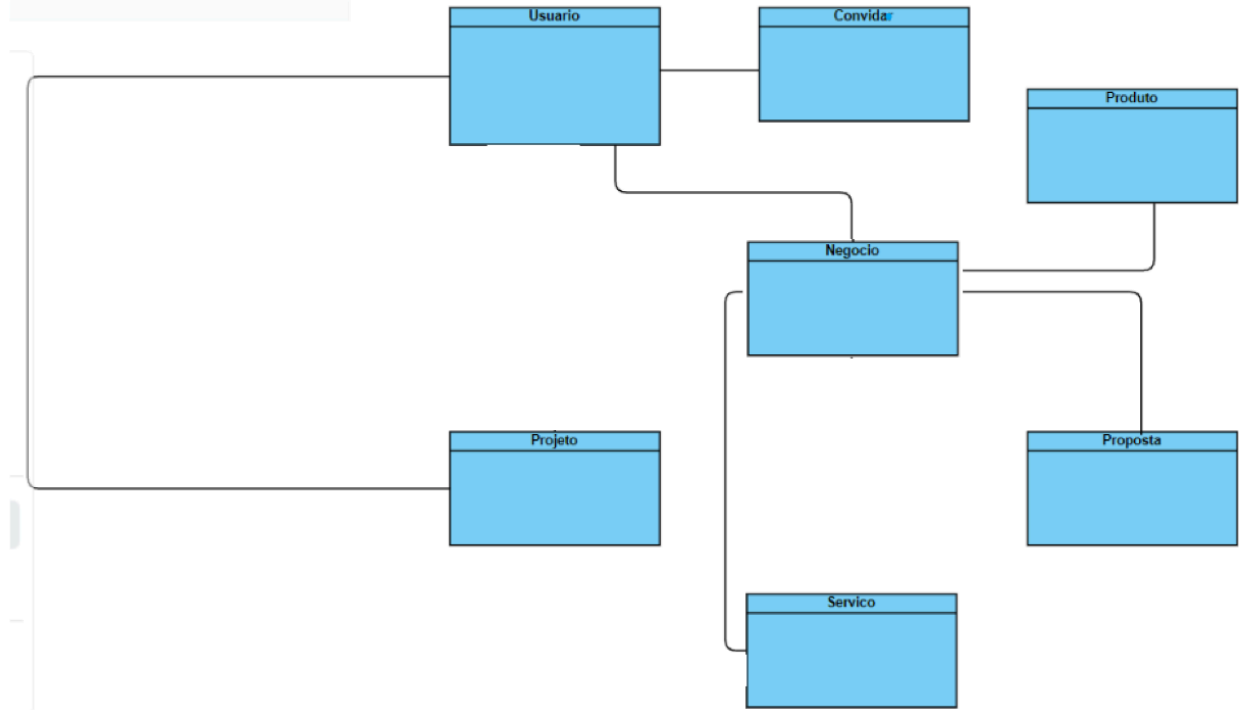
	investidor.		
<b>[RF-16]</b>	A plataforma deve permitir que o estudante publique detalhes de seus projetos com link para os arquivos.	3	Essencial
<b>[RF-17]</b>	A plataforma deve permitir a o usuário ver a lista de projetos com filtro.	2	Importante
<b>[RF-18]</b>	A plataforma deve permitir a o estudante editar seus projetos.	3	Desejável
<b>[RF-19]</b>	A plataforma deve permitir a o estudante excluir seus projetos.	2	Importante

# Análise - Diagrama Caso de Uso



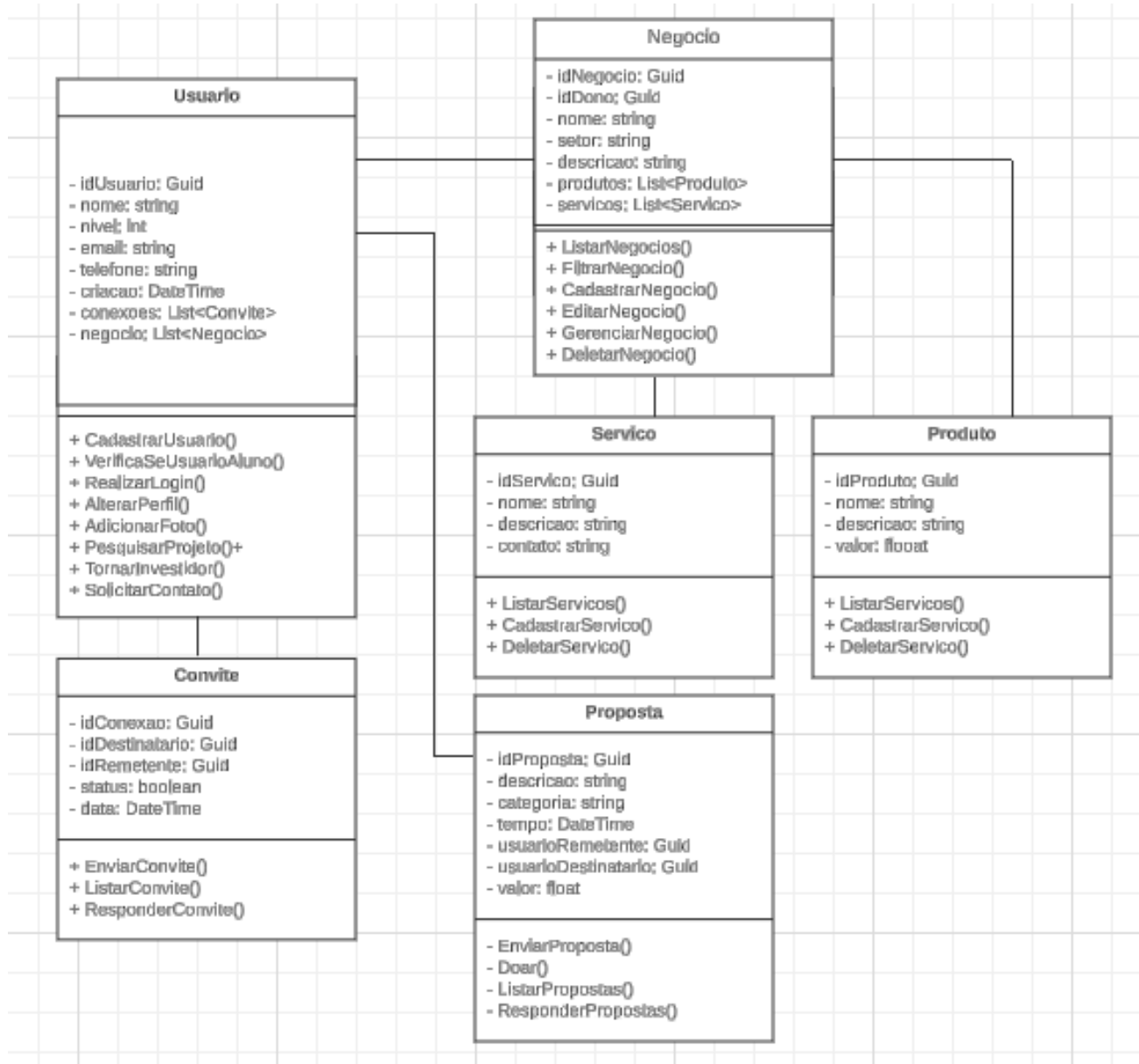
\* Caso o usuário seja criado com email "@al.infnet" o sistema deve automaticamente alterar seu perfil para "Aluno".

# Modelo de Dominio



\*Plus: grupos e mensagem.

# Diagrama de Classe



## Arquitetura - Requisitos Não Funcionais

Requisito	Descrição	Justificativa
<b>Usabilidade</b>	<p>O sistema deve ser fácil de usar para que o usuário com poucos cliques realize seus casos de uso.</p> <p>Compatibilidade com diversas resoluções</p>	<p>Nem todos os usuários são experientes com tecnologia, então para evitar o abandono da plataforma é necessário uma boa experiência responsiva.</p>
<b>Segurança</b>	<p>O sistema deve ser protegido contra acesso não autorizado, permitindo que cada usuário tenha acesso às funcionalidades de seu grupo (role).</p> <p>Para implementar esse requisito o sistema deverá trocar tokens durante a comunicação de requisições não públicas.</p>	<p>É necessário proteger os dados dos usuários e suas informações.</p>
<b>Desempenho</b>	<p>A plataforma deve ser rápida.</p>	<p>A plataforma deve ser rápida para lidar com um grande número de projetos.</p>

# Stack Tecnologia

Camada	Tecnologia
Apresentação (Front-End)	Angular (Typescript) Tailwind, Flowbite, Flaticon
Negócio (Back-End)	C# (.NET) Padrões: DDD, Clean Code e DI
Dados (Banco-Dados)	SQL Server (Microsoft)

## Justificativa

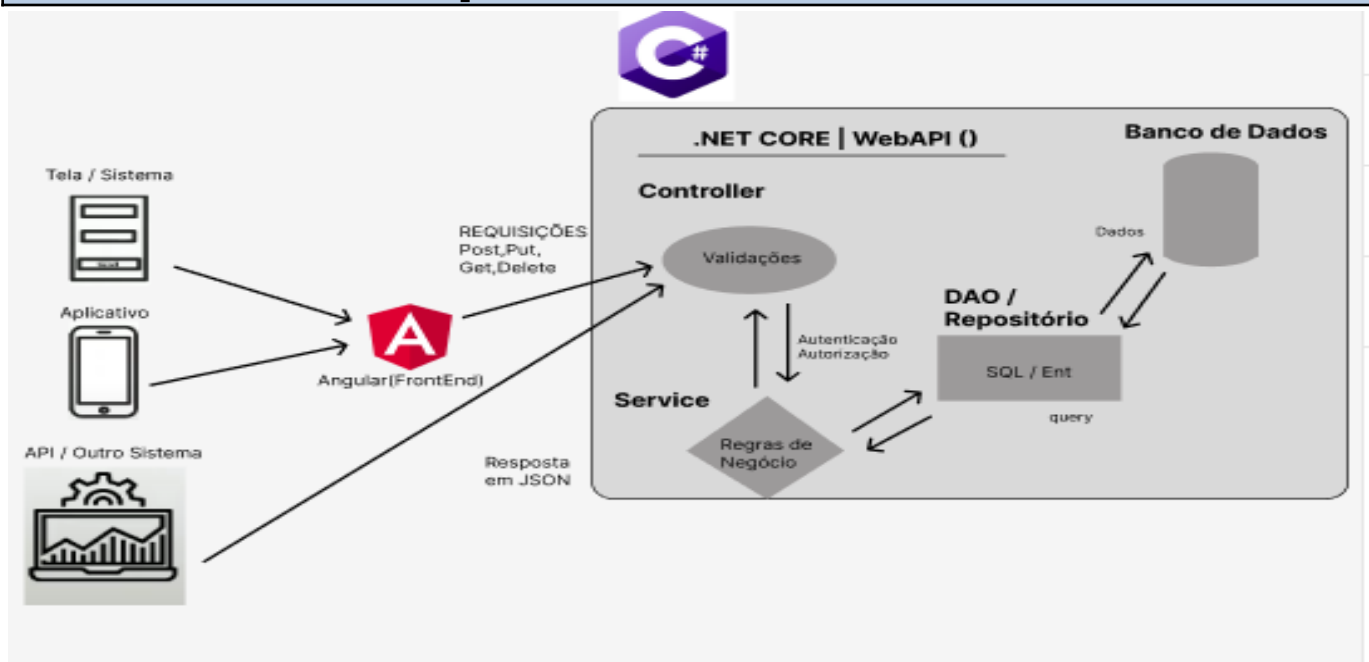
**Performance e Escalabilidade** uma stack conhecida por sua alta performance, modernidade e baixo consumo de recursos.

**Integração** : O .NET Core é uma estrutura moderna e flexível que se integra bem com o Angular.

Comunicação entre o back-end e o front-end é simplificada, permitindo uma experiência de desenvolvimento mais fluida.

**Comunidade e Atualizações** correções de segurança e acesso a bibliotecas e ferramentas de terceiros.

## Arquitetura - Desenho



# Mínimo Produto Viável - MVP

[RF1]	A plataforma deve permitir que uma pessoa <b>cadastre</b> um perfil aluno ou investidor, informando um email válido, telefone e uma senha segura.
Necessário salvar as informações de cada usuário em seu perfil e identificação e autorização do usuário.	

[RF4]	A plataforma deve permitir que um usuário <b>pesquise</b> projetos com filtro (nome, curso, email...)
Necessário separar projetos por categoria, onde o usuário encontra o aluno que tem capacidade para atender sua necessidade específica. Aprimorando a busca com um filtro na tela e botões específicos para cada categoria	

[RF-16]	A plataforma deve permitir que o usuário preencha um formulário de projeto e <b>publique</b> em seu portfólio. (Título, Público, Categoria, Descrição, Bloco, Objetivo, Links [código-fonte, fotos, Anexos, documentação]).
Para os alunos atraírem investidores é necessário a criação do catálogo de projetos da plataforma. Neste catálogo será possível que o aluno cadastre informações sobre seus projetos e caso necessário adicionar links externos (Ex.: código fonte, vídeo apresentação, foto...).	

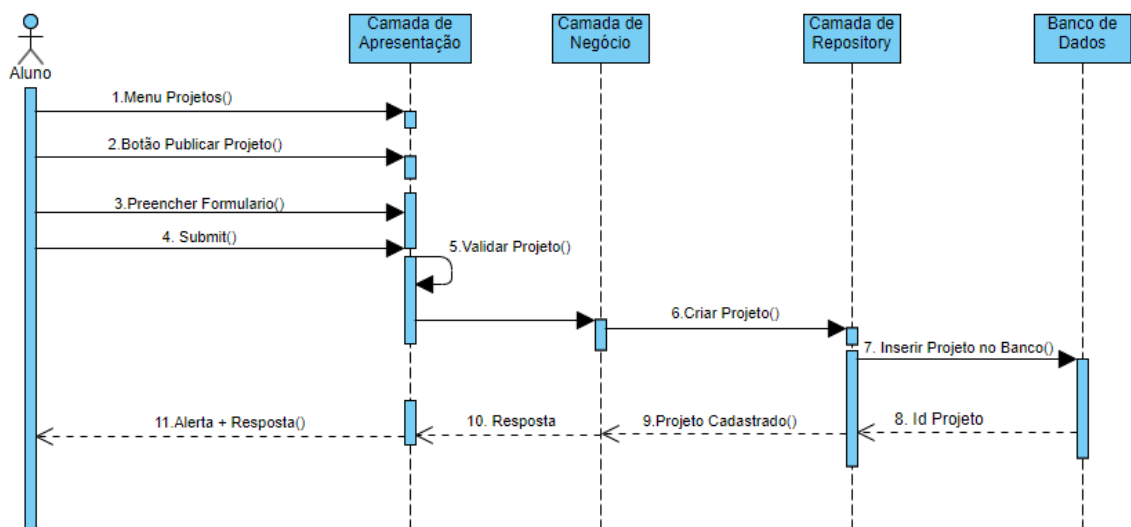
[RF-17]	A plataforma deve permitir que o usuário <b>pesquise</b> projetos filtrando os parâmetros de entrada.
No catálogo de projetos o usuário terá acesso ao portfólio, onde será possível identificar as principais qualidades e experiências de cada aluno.	

[RF-14]	A plataforma deve permitir que o Investidor faça <b>proposta</b> de parceria a um estudante (interação).

<b>[RF-15]</b>	<p>A plataforma deve permitir que o estudante <b>responda</b> suas propostas.</p> <p>Caso aceite, incremente pontos no perfil do estudante e investidor.</p>
----------------	--

## Diagrama de Sequência

### [Cadastrar Projeto]



**Ator->Sistema: Solicita cadastro de projeto**

**Sistema->Ator: Exibe formulário de cadastro**

**Ator->Sistema: Preenche detalhes do projeto**

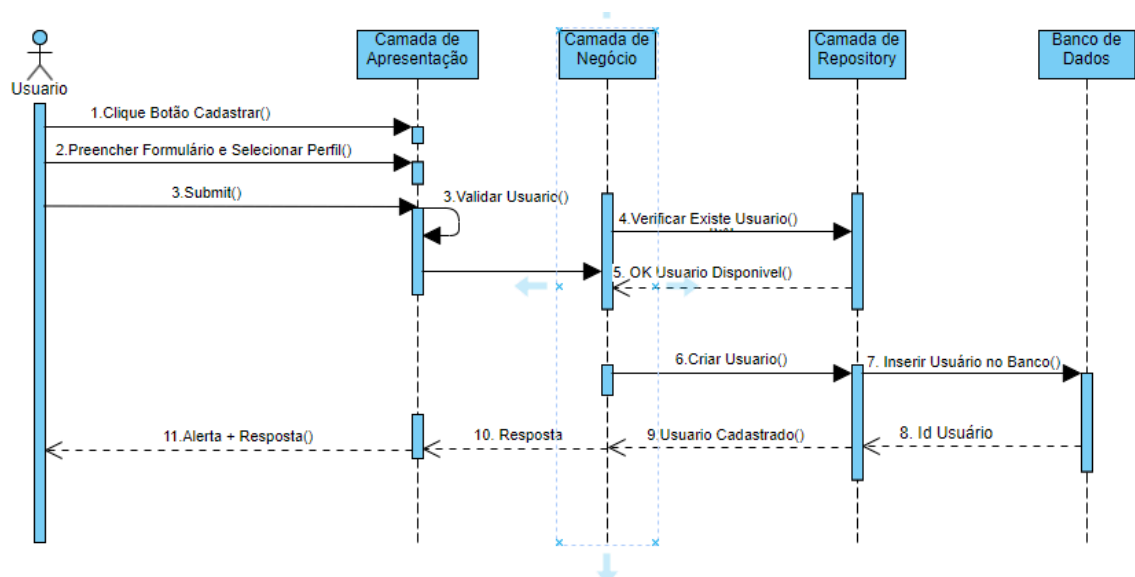
**Sistema->Ator: Valida dados**

**Sistema->Projeto: Cria novo projeto**

**Sistema->Banco de Dados: Armazena projeto**

**Sistema->Ator: Confirma sucesso do cadastro**

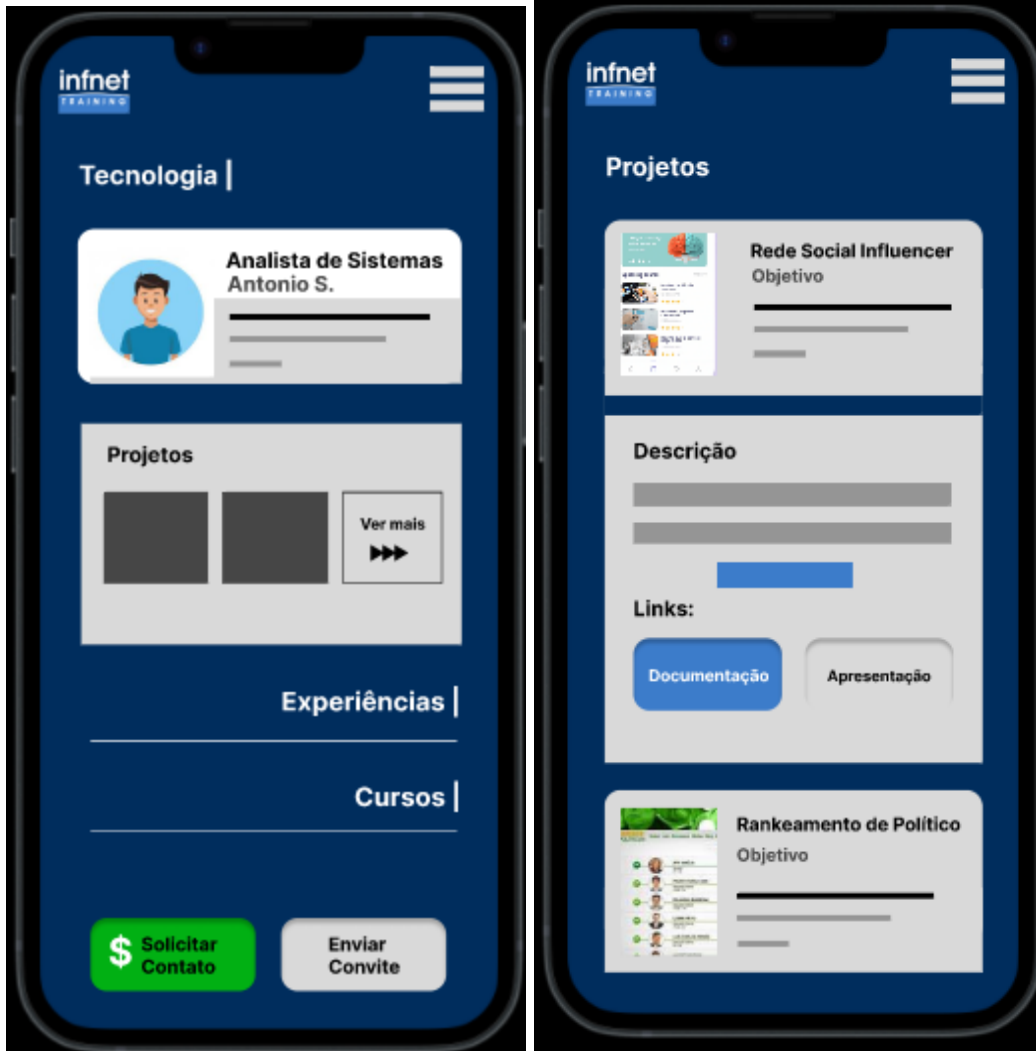
## [Cadastrar Usuário]



## Protótipo de Tela





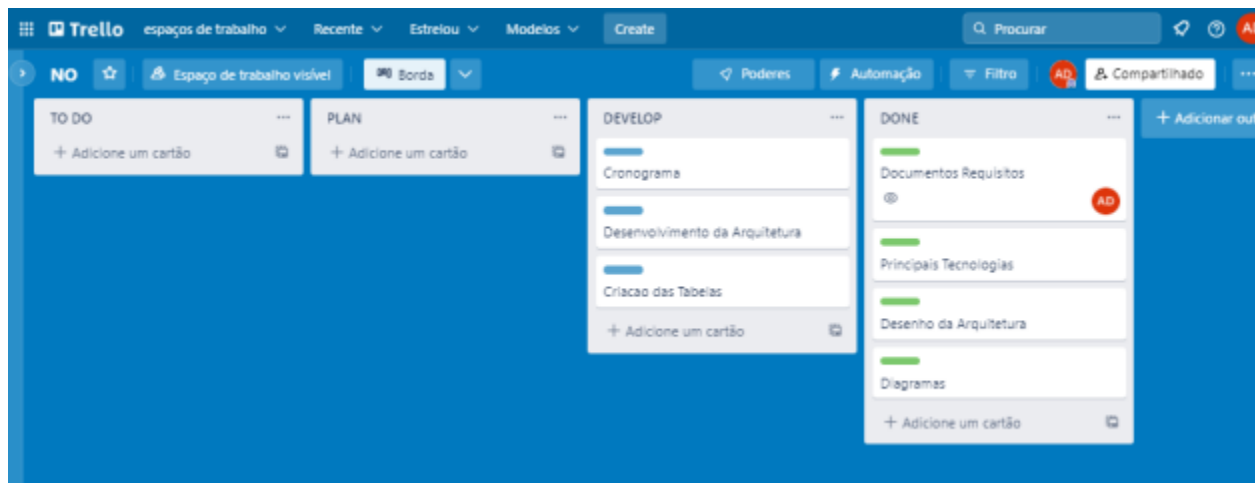


**Link Protótipo:**

<https://www.figma.com/file/bT7ERxSwT56VKBGUG80z4A/Untitled?type=design&node-id=1-33&mode=design&t=3U7vuSUzjE4rDGU2-0>

# Tarefas (Iteração Time)

SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 5	SEMANA 6	SEMANA 7	SEMANA 8
Entregar a EAP e o Cronograma Melhorar os critérios de sucesso (1.1.3)	<b>Desenho</b>	Prototipação (1.2.4)	<b>Codificação (1.3)</b>	Desenvolvimento das telas de prototipação (Proto-End)	Desenvolvimento da Camada de Negócios (1.3.5)	Desenvolvimento da Camada de Banco (1.3.6)	<b>Testes 1.4</b>
Documentar marcos do projeto Sucesso (1.1.5)	Desenho da Base de Dados (1.2.1)	Diagrama de Relacionamento (1.2.5)	Rever a documentação		Crud - Habilidades (1.3.5.1)		QA testing
<b>Finalizar Planejamento</b>	Desenho Arquitetura (1.2.2)		Desenvolvimento da Arquitetura (1.3.1)		Crud - Serviços (1.3.5.2)		
<b>Documentar o Planejamento</b>		Diagrama de caso de uso (1.2.5)	Criação de tabelas (1.3.2)		Crud - Produtos (1.3.5.2)	Desenvolvimento de Validações (1.3.7)	Obter aprovação (1.5.1)
Estudar a etapa de Desenho (1.2)	Criar documentação de dependências (1.2.2)	<b>Finalizar documentação</b>	Desenvolvimento das Entidades (1.3.3)	Gestão de Acessos (1.3.4)	<b>Finalizar Camada de Negócios</b>	Implementações de Segurança (1.3.8)	<b>Deploy (1.5.2)</b>
				<b>Finalizar Interface</b>	Confirmar Relacionamentos	<b>Finalizar Software</b>	<b>Importante</b>
							<b>Feedback</b>



## GITHUB (link)

[https://github.com/DemarchiWorking/Devdelivery\\_PortfolioPB\\_API.git](https://github.com/DemarchiWorking/Devdelivery_PortfolioPB_API.git)

[https://github.com/DemarchiWorking/Devdelivery\\_PortfolioPB\\_Frontend.git](https://github.com/DemarchiWorking/Devdelivery_PortfolioPB_Frontend.git)

[https://github.com/DemarchiWorking/Devdelivery\\_Portfolio\\_PB.git](https://github.com/DemarchiWorking/Devdelivery_Portfolio_PB.git)

Atividades que orientam a construção de software e o desenvolvimento ágil. Promovendo colaboração, transparência e o desenvolvimento iterativo orientado a valor, tornando uma pequena equipe (<15) altamente colaborativa e auto-organizada.

Seu diferencial é se adaptar a mudanças nos requisitos e nas necessidades dos clientes de forma simples, diferente de outros planos mais rígidos.

# Testes (TDD)

Minha experiência com Desenvolvimento Orientado a Testes (TDD) foi bastante enriquecedora. Infelizmente, devido ao tamanho do meu projeto e o prazo curto, não tive a oportunidade de implementar o TDD no projeto completo.

No entanto, entendo a importância dessa prática para garantir a qualidade do código e a confiabilidade das funcionalidades.

- Primeiro, escrevemos um teste;
- Executamos o teste e verificamos se ele falha
- Depois, escrevemos o código para fazer o teste passar

## Plano de Teste para o Backend com xUnit:

Arrange: Criamos um mock do repositório de projetos e instanciamos o Projeto

Act: Chamamos o método CadastrarProjeto com um projeto de teste.

Assert: Verificamos se o método Adicionar do repositório foi chamado exatamente uma vez.

Verificar se as condições ou propriedades estão corretas.

Exemplo Xunit:

```
[Fact]
public void Test_CadastrarProjeto_Valido()
{
    var projeto = new CadastroProjetoModel
    {
    };
    var result = _projetoController.CadastrarProjeto(projeto);

    Assert.True(result is OkResult);
}
```

## Plano de Teste para o Front-End com Cypress:

Cadastro de Aluno: Cenário: Preencher o formulário de cadastro com informações válidas (id, xpath,class...)

- Teste: Acessar a página de cadastro.
- Preencher os campos obrigatórios (nome, e-mail, senha).
- Clicar no botão “Cadastrar”.
- Verificar se o aluno é redirecionado para a página de perfil.

Exemplo:

```
describe('Cadastro de Projeto', () => {  
  it('Deve cadastrar um novo projeto', () => {  
    cy.visit('/cadastro-projeto');  
    cy.get('#nome').type('Projeto A');  
    cy.get('#descricao').type('Descrição do Projeto A');  
    cy.get('button[type="submit"]').click();  
  });  
});
```

Testes unitários garantem que cada parte do código funcione conforme o esperado. Isso ajuda a manter a qualidade do software e a evitar regressões.



Documentação: Os testes unitários descrevem como as diferentes partes do sistema devem se comportar e podem ser usados como referência para outros desenvolvedores.

Integração contínua: Testes unitários são uma parte fundamental da integração contínua. Eles são executados automaticamente sempre que há uma alteração no código, garantindo que problemas sejam detectados rapidamente.

# Implantação - Passos

- 1. Criar uma conta na AWS:**
  - Acesse o Console AWS e crie uma nova conta.
- 2. Escolha uma região da AWS:**
  - Selecione a região da AWS onde deseja implantar sua aplicação.
- 3. Crie uma instância EC2:**
  - No Console da AWS selecione o serviço EC2.
  - Crie uma nova instância EC2
- 4. Implantar o backend .NET (API):**
  - Fazer a publicação do código .NET para a instância EC2.
  - Instale o .NET Core
- 5. Configure o servidor web:**
  - Configure um servidor web.
- 6. Crie um bucket S3:**
  - Acesse o console da AWS e crie um novo bucket .
  - Configure as permissões.
  - Faça upload dos arquivos do Angular para o bucket.
- 7. Configure o banco de dados SQL:**
  - Crie uma instância de banco de dados RDS.
  - Importe seu banco de dados SQL para a instância RDS.
  - Faça upload dos arquivos do Angular para o bucket.
- 8. Execute as migrações**
  - Execute a migração com o comando:
    - `dotnet ef database update`
    - `dotnet ef migrations add PrimeiraMigracao --project`
- 9. Configure o domínio e o DNS:**
  - Registre um domínio.
  - Configurar registros DNS para apontar para o IP público.

## Apresentação - Bonus

Profissão:		
Nome:	Ana Luiza	
Sexo:	Feminino	
Idade:	23	
Localização:	Campos	
		

O produto é ?	O produto <b>não</b> é ?
Aplicação web para estudantes divulgarem seus projetos.	Plataforma de Curso ou Chat.

Como **estudante**, Quero divulgar meu trabalho e encontrar parcerias para meu negócio.

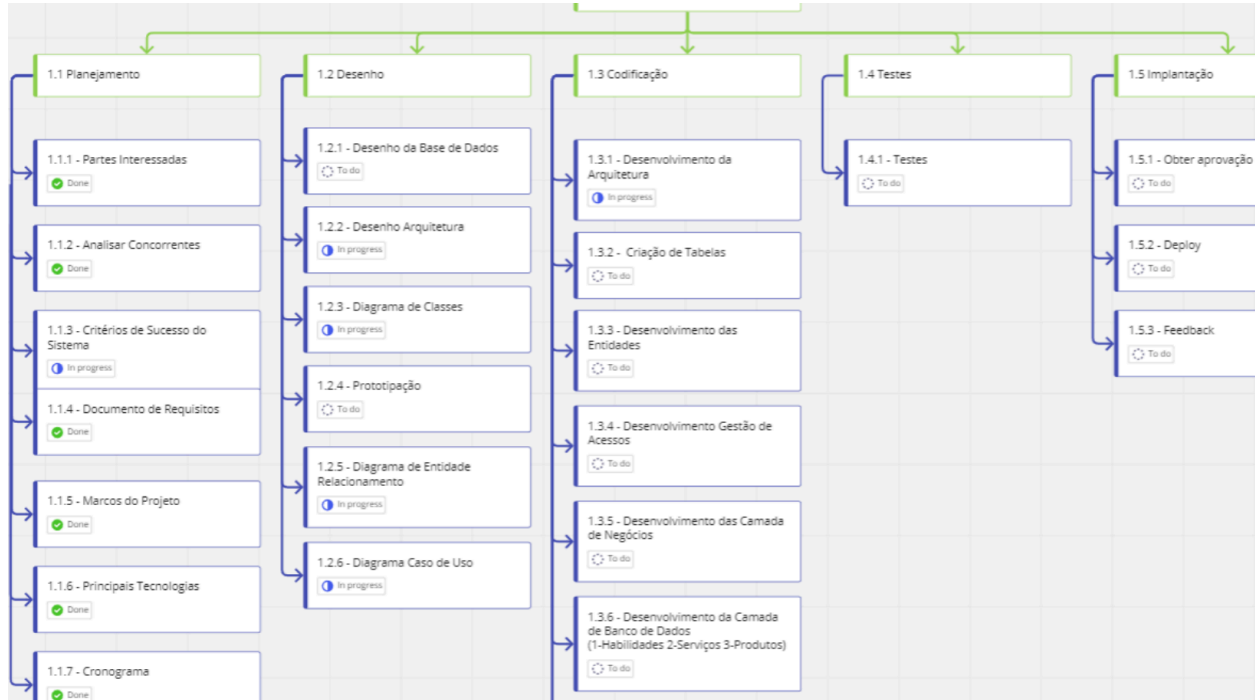
Gravidade	5	A funcionalidade principal
Urgência	5	Necessária para o lançamento
Tendência	5	Evolutivo

Como **investidor**, Quero ter acesso a uma avaliação grátis e acesso a conselhos de negócios dos melhores alunos .

Gravidade	5	A funcionalidade principal
Urgência	5	Necessária para o lançamento
Tendência	5	Evolutivo

Painel			
N	Certezas	Suposições	Dúvidas
1.1	Escopo Inicial	Necessidade Mercado	Aceitação dos Usuários
1.2	Tecnologias	Escalabilidade	Monetização
1.3	Aquisição de Usuários	Segurança	Feedback
1.4	Catálogos Projetos	Design e Usabilidade	Indicação
1.5	Lista de Alunos	Negócios	Investidores

# Cronograma



## Melhorias Futuras.

- Tipagem nas variáveis do front (any)
- Regras de negócio de validações concretas.
- Aplicar TDD em novas features.