# Definição de Projeto

## Socifit

# I. Escopo

O sistema tem como objetivo fornecer uma solução completa para o gerenciamento de uma academia especializada em musculação e reabilitação física, permitindo o controle de professores, alunos, agendas, treinos e notificações. Ele facilitará o cadastro e acompanhamento dos alunos em processo de reabilitação, oferecendo planos de treino personalizados. A agenda integrada permitirá o agendamento de aulas e sessões, enquanto o sistema de notificações manterá todos os envolvidos informados sobre atualizações. Dessa forma, o sistema ajudará os profissionais da academia a oferecer suporte personalizado e eficiente aos alunos durante sua recuperação.

# II. Tecnologia e frameworks principais

As tecnologias foram separadas em dois tipos: as tecnologias e frameworks que vão ser utilizadas no app desktop e as tecnologias que serão utilizadas no backend, que vai ser consumido pelo app desktop.

## Desktop

- ### Electron

  O Electron permite desenvolver aplicações desktop usando tecnologias web, como HTML, CSS e JavaScript, o que reduz a curva de aprendizado para a equipe. Ele é ideal para criar aplicativos cross-platform (Windows, macOS, Linux) com uma base de código única.

- ### Javascript/Typescript

  A escolha de JavaScript/TypeScript como linguagem para o frontend e backend reflete a experiência da equipe, garantindo maior produtividade. O TypeScript, em particular, oferece tipagem estática e melhorias de desempenho no desenvolvimento, além de reduzir erros. Essa consistência entre as tecnologias facilita o compartilhamento de código entre o cliente (desktop) e o servidor (backend).

- ### Next.js

  O Next.js é um framework React que oferece funcionalidades como renderização híbrida (SSR e SSG), roteamento simples e otimizações automáticas. Ele é utilizado no sistema desktop para construir a interface de forma eficiente, aproveitando a facilidade de criar layouts dinâmicos e responsivos com ótima performance.

- ### TailwindCSS

  O TailwindCSS é uma biblioteca de CSS que promove o desenvolvimento de interfaces com design consistente e altamente customizável. Sua abordagem baseada em classes utilitárias reduz a escrita de CSS manual e acelera o desenvolvimento da interface, especialmente em combinação com frameworks como Next.js.

### Backend

- Javascript/Typescript - Node.js

  A utilização do Node.js permite construir um backend escalável e eficiente, utilizando a mesma linguagem que o frontend (JavaScript/TypeScript). A experiência prévia da equipe com essas tecnologias, junto com a ampla comunidade e disponibilidade de bibliotecas, facilita o desenvolvimento rápido e confiável.

- Nest.js

  O Nest.js é um framework estruturado que oferece ferramentas e boas práticas para desenvolvimento backend. Ele promove modularidade, padronização de código e suporte nativo ao TypeScript, tornando a aplicação mais fácil de manter e escalável.

- Prisma (ORM)

  O Prisma é um ORM moderno que simplifica a interação entre a aplicação backend e o banco de dados. Ele oferece migrações automatizadas, suporte a TypeScript e uma interface declarativa para consultas, facilitando o gerenciamento de dados e mantendo a consistência no desenvolvimento.

- SQLite

  O SQLite foi escolhido como banco de dados devido à sua simplicidade, leveza e fácil integração com o Prisma. Ele é ideal para sistemas desktop e pequenos servidores, pois o banco é armazenado em um único arquivo, o que elimina a necessidade de configurar ou manter um servidor de banco de dados dedicado.

## III. Ferramentas

As ferramentas utilizadas serão os sistemas escolhidos para ajudar tanto no desenvolvimento quanto na organização, repositório de código e comunicação.

### 1. Comunicação

- Discord

  Escolhido pela facilidade de organizar a informação em vários canais de interesse.

- Whatsapp

  Utilizado para comunicações mais informais.

### 2. Repositório e Gerenciamento de tarefas

- Gitlab

  Foi escolhido pela centralização de funcionalidades como criação de atividades, repositório de código e CI/CD mais consolidado dentro do mercado, tudo isso numa só ferramenta.

### 3. IDE

- VSCode

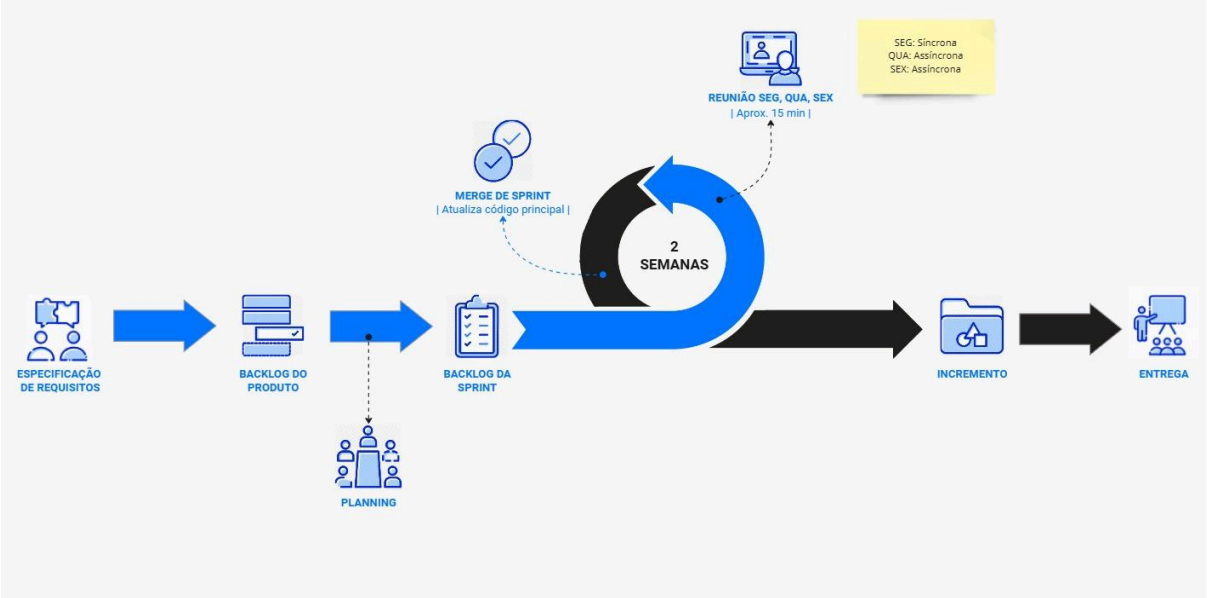
  IDE escolhida para trabalhar graças à quantidade de extensões e ferramentas voltadas para o desenvolvimento em Javascript/Typescript.

## IV. Arquitetura

- Backend

  Para o backend, o padrão de arquitetura escolhido foi o Controller-Service-Repository, onde serão criados os endpoints nos controllers para serem utilizados pela aplicação desktop, terá os repositories para acessar o banco de dados e os services para manter as regras de negócio do sistema.

## V. Processo




- Passos que foram definidos sobre como será o processo utilizado pela equipe:

  1. O processo que estaremos utilizando será baseado no Scrum, com reuniões síncronas todas as segundas-feiras e reuniões assíncronas todas as quartas-feiras e sextas-feiras. No começo de cada sprint, será feita uma sprint planning para selecionar as histórias do backlog do projeto e trazê-las para o backlog da sprint como tarefas.

  2. Antes de cada sprint, será realizada a especificação de requisitos com o cliente e atualização do protótipo e do diagrama de banco de dados, quando necessário.

3. No final da sprint, será realizado o merge da sprint e será gerado o incremento do produto, o qual será validado com o cliente. Depois de validado, será separado o que precisa de correções e serão criadas tarefas para a próxima sprint. Assim, vamos construindo incrementalmente nosso software durante a disciplina de projeto.