



FITTRACK

Aplicativo Web de Rastreamento de Fitness
Entrega Final

INTRODUÇÃO E OBJETIVO

🎯 O PROBLEMA

A dificuldade em manter um registro consistente e organizado de atividades físicas e métricas de saúde dispersas.

💡 A SOLUÇÃO

Um sistema centralizado para registrar treinos, calcular calorias e visualizar o progresso ao longo do tempo.

PÚBLICO-ALVO

- ✓ Entusiastas de fitness e atletas amadores.
- ✓ Pessoas buscando monitoramento de saúde.
- ✓ Usuários que desejam histórico de evolução.



REQUISITOS FUNCIONAIS



AUTENTICAÇÃO

O usuário deve conseguir se cadastrar, fazer login e gerenciar seu perfil com dados de saúde (peso, altura).



ATIVIDADES

Registro de treinos com tipo, duração, intensidade e cálculo automático de calorias.

DASHBOARD

Visualização de histórico de atividades e estatísticas de desempenho (resumo).

Diagrama de Sequência

Mostra o fluxo do processo de registrar uma atividade: o usuário envia os dados pela interface, o backend valida e salva no banco de dados, e o sistema retorna uma confirmação de sucesso.

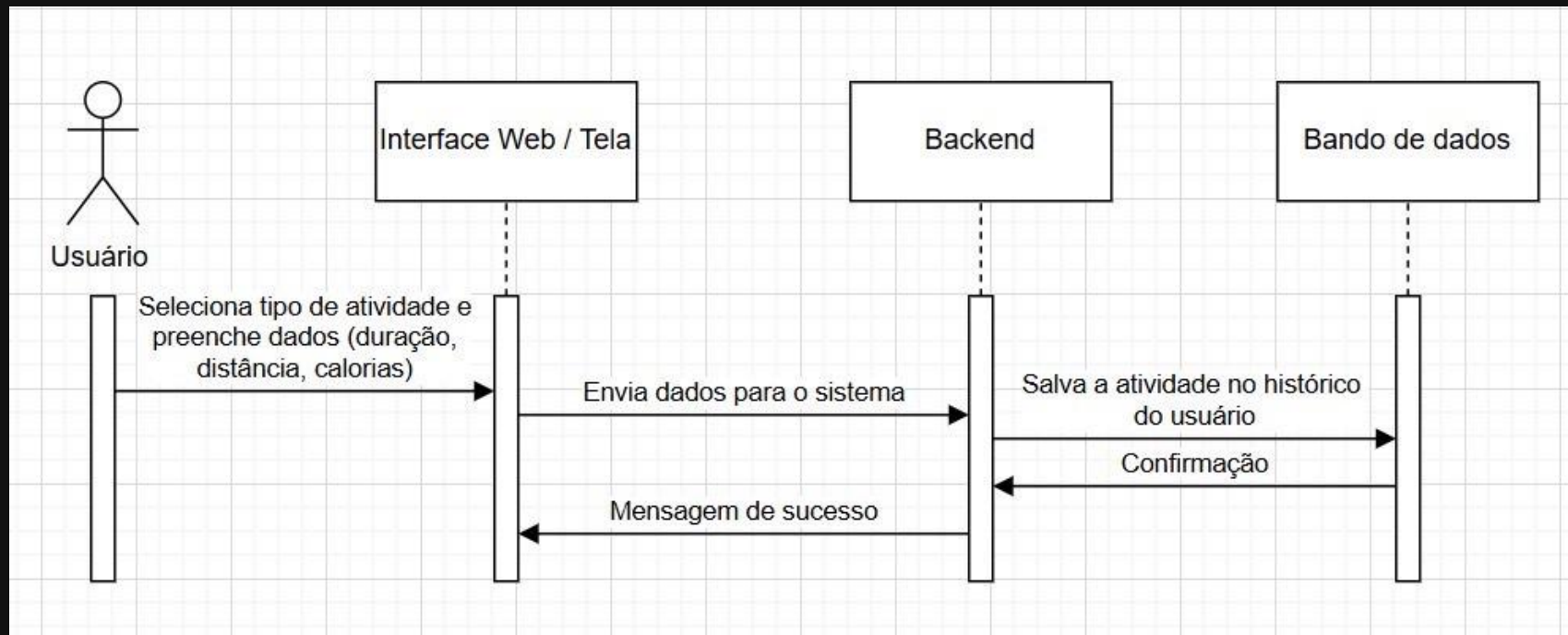


Diagrama de Caso de Uso

Representa as ações que o usuário pode realizar dentro do sistema de metas, como criar metas recorrentes ou marcá-las como concluídas. Também mostra funcionalidades internas, como reiniciar metas automaticamente e enviar notificações.

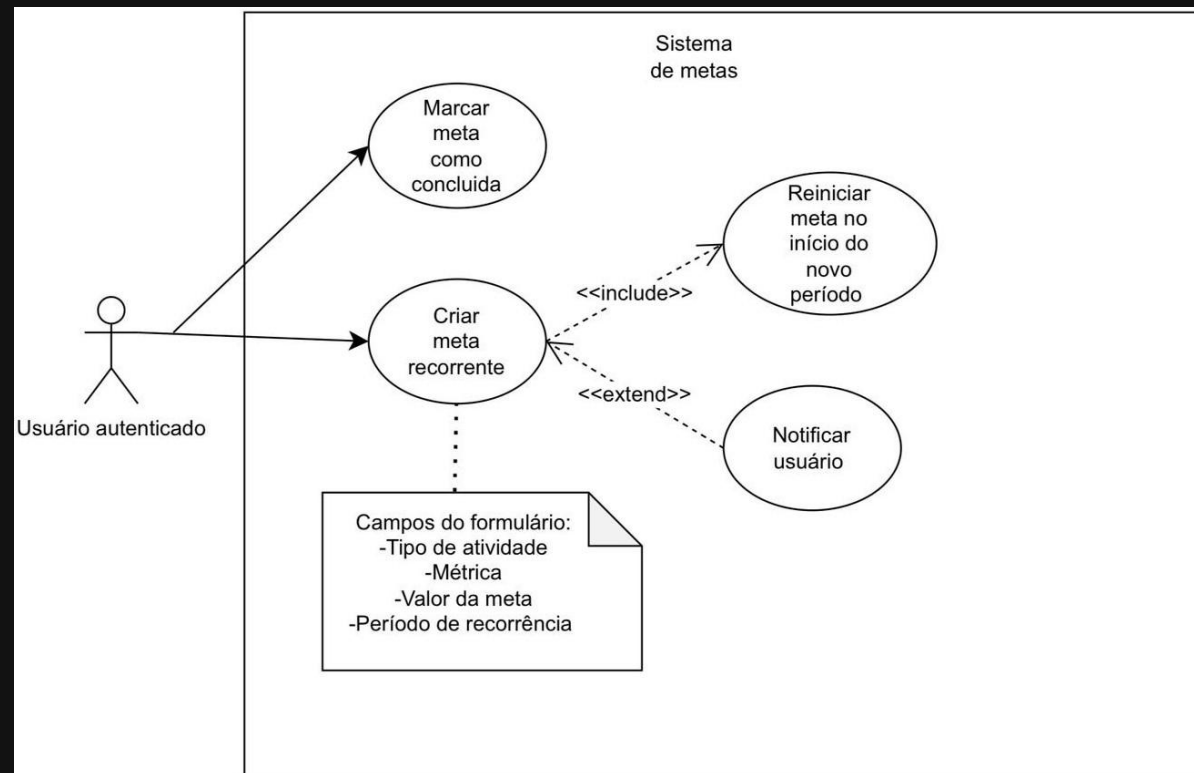


Diagrama de Camadas

Divide o sistema em três níveis: apresentação (interfaces web e mobile), camada de negócios (lógica do sistema, como atividades e metas) e persistência (bancos de dados). Ilustra como essas camadas se comunicam entre si.

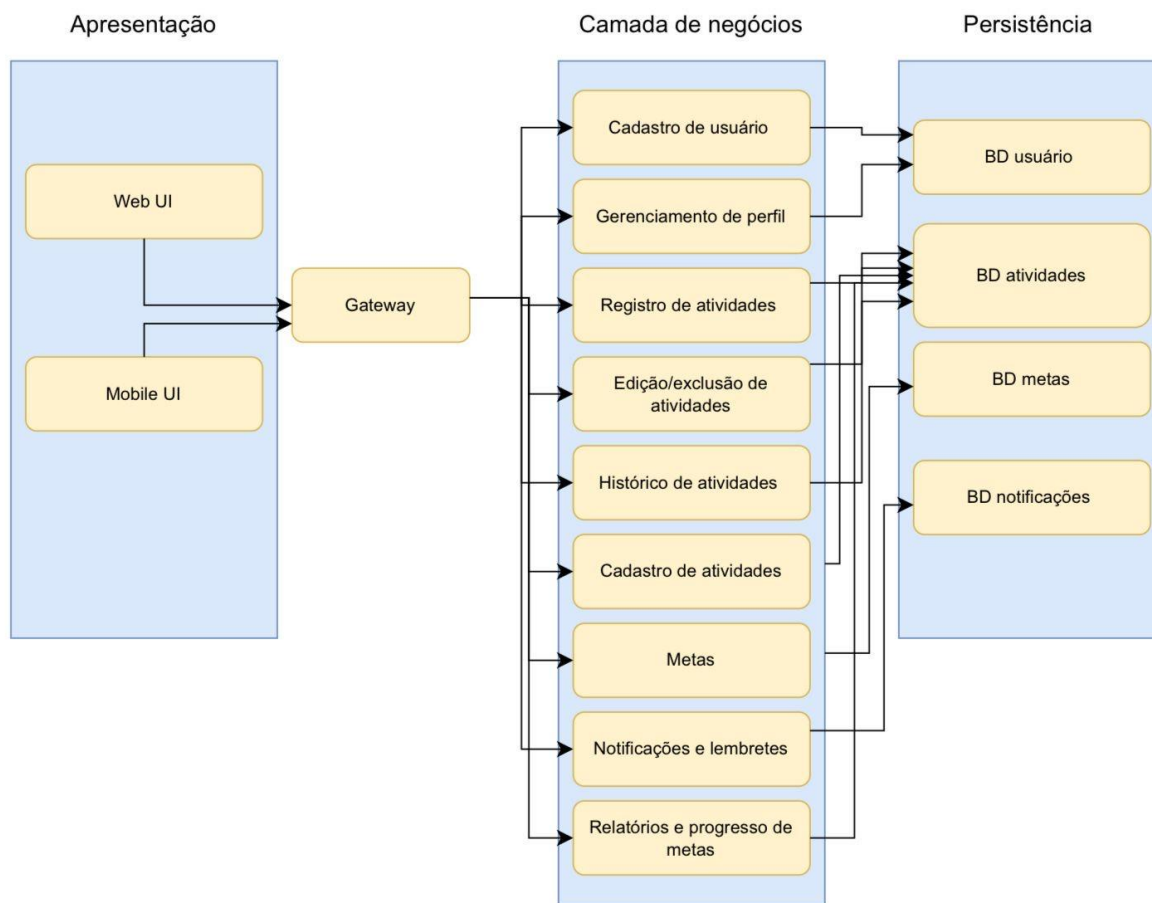
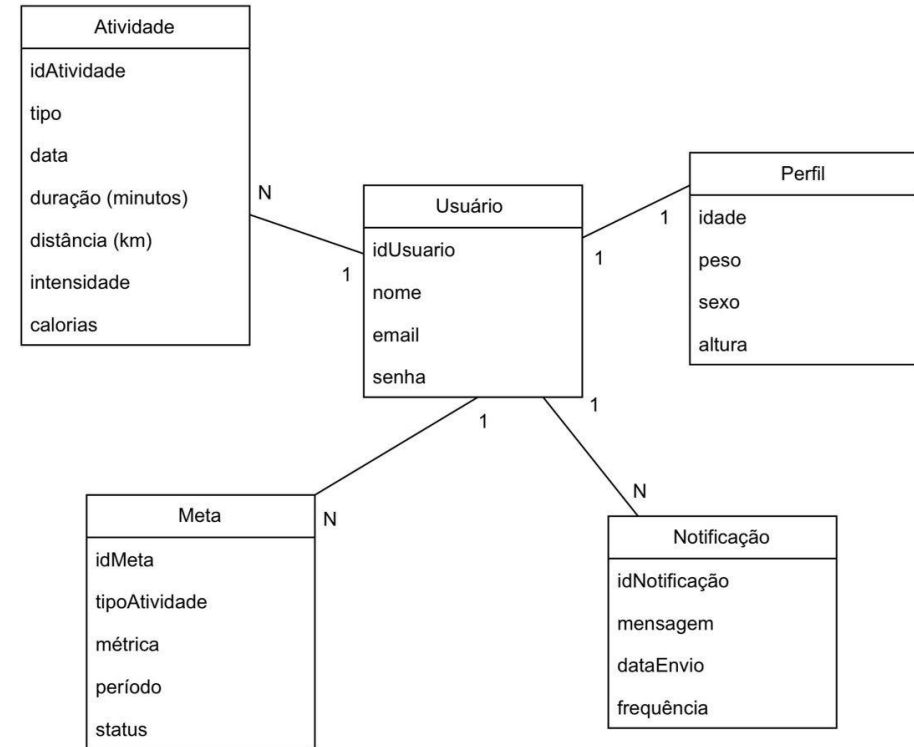


Diagrama de Classes

Mostra os principais dados do sistema e como eles se relacionam.

O usuário possui perfil, atividades, metas e notificações. Cada classe contém seus atributos essenciais, representando como as informações são organizadas no banco de dados.



| Principais Histórias de Usuário

US01 – Cadastro de Usuário

“Eu, como usuário, quero criar uma conta, para registrar minhas atividades e acompanhar meu progresso.”

US02 – Registrar Atividade Física

“Eu, como usuário, quero registrar manualmente minhas atividades (corrida, caminhada, ciclismo, etc.), para acompanhar o que fiz durante a semana.”

US03 – Editar e Excluir Atividades

“Eu, como usuário, quero editar ou excluir registros de atividades, para corrigir informações inseridas incorretamente.”

US04 – Criar Meta Recorrente

“Eu, como usuário, quero definir metas semanais ou mensais, para manter minha disciplina e medir minha evolução.”

US05 – Ver Histórico de Atividades

“Eu, como usuário, quero ver um histórico organizado das minhas atividades, para analisar meu progresso ao longo do tempo.”

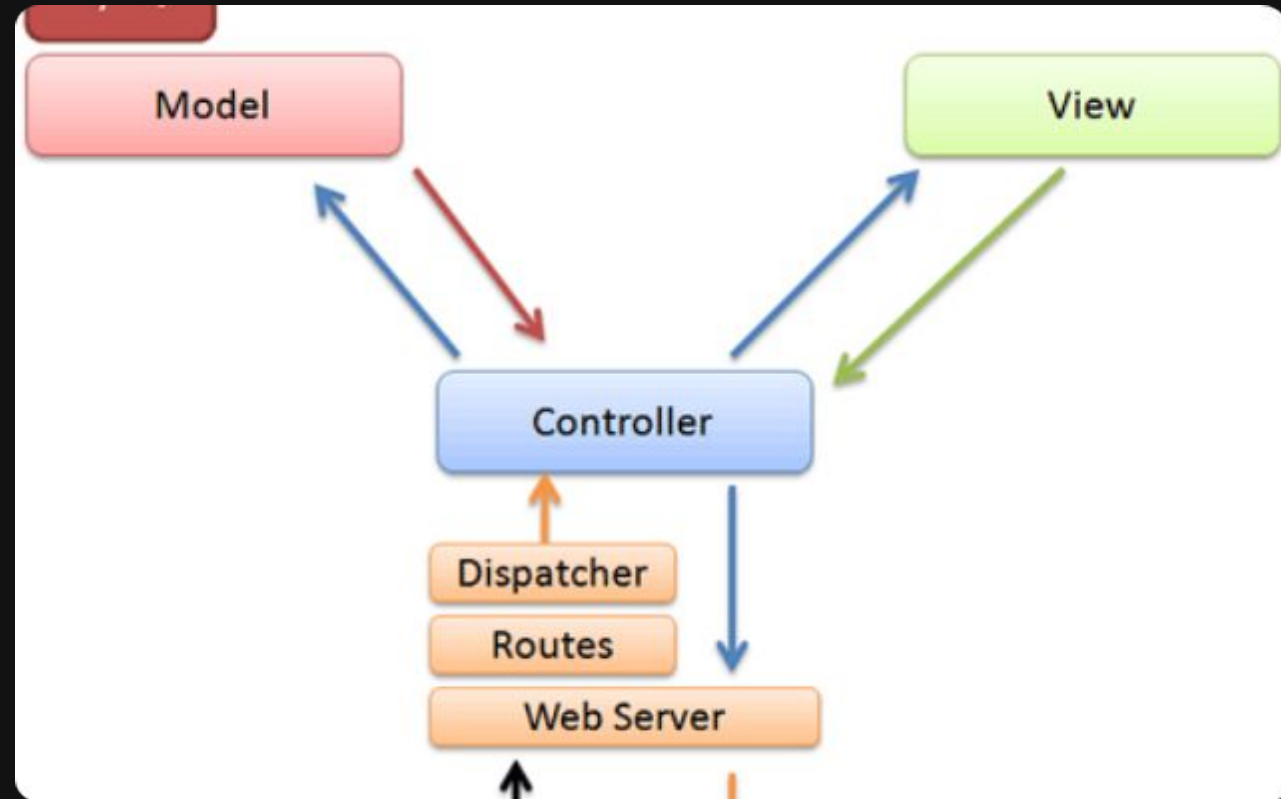
US06 – Receber Lembretes e Notificações

“Eu, como usuário, quero receber lembretes, para não esquecer de realizar minhas atividades e metas.”

ARQUITETURA MVC

O projeto segue o padrão **Model-View-Controller** para separação de responsabilidades.

- ✓ **Model (Python):** Definição das tabelas e regras de negócio (`models.py`).
- ✓ **View (HTML/JS):** Interface do usuário e renderização (`templates/`, `js/views`).
- ✓ **Controller (Flask/JS):** Gerenciamento de rotas e lógica de requisições.



IMPLEMENTAÇÃO: BACKEND

TECNOLOGIAS

Python Flask SQLAlchemy

ESTRUTURA DE PASTAS

- ✓ backend/models/: Modelos ORM (Usuario, Atividade).
- ✓ backend/routes/: Endpoints (Auth, Atividades).
- ✓ backend/config.py: Configurações de ambiente.

CÓDIGO (ROTAS)

```
# Exemplo de Rota Flask
@app.route('/api/atividades', methods=['POST']) def
criar_atividade(): data = request.json
nova_atividade = Atividade( tipo=data['tipo'],
duracao=data['duracao'],
calorias=calcular_calorias(...) )
db.session.add(nova_atividade) db.session.commit()
return jsonify(nova_atividade.to_dict()), 201
```

| FRONTEND & BANCO DE DADOS

FRONTEND INTERATIVO

Desenvolvido com **Vanilla JS (ES6+)**.

- ✓ **APIClient.js:** Abstração para chamadas ``fetch``.
- ✓ **Controllers.js:** Lógica de manipulação do DOM.
- ✓ **Utils.js:** Cálculos de IMC e validadores.

PERSISTÊNCIA (SQLITE)

Tabelas Relacionais:

- ✓ **Usuarios:** ID, Nome, Email, Hash Senha, Dados Saúde.
- ✓ **Atividades:** ID, FK_Usuario, Tipo, Duração, Intensidade.

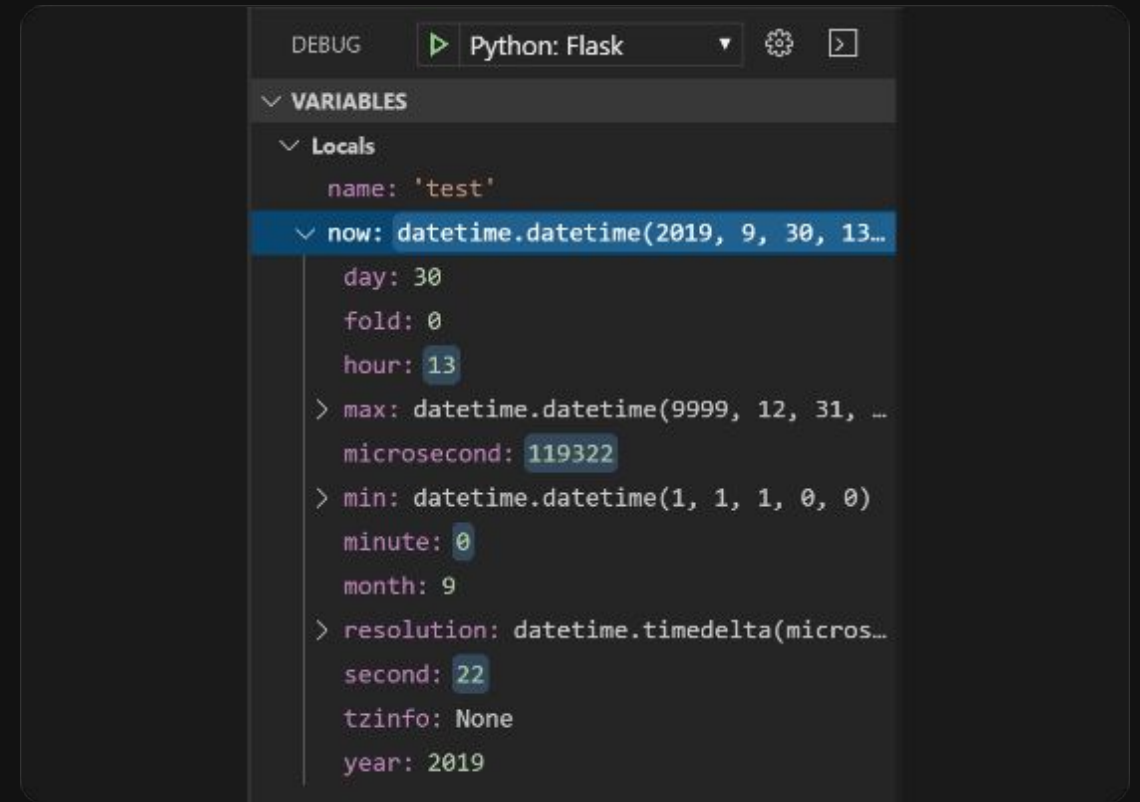
*Migração futura planejada para PostgreSQL.

I TESTES E QUALIDADE

ESTRATÉGIA DE TESTES

Focamos em garantir a integridade dos dados e a segurança das rotas.

- ✓ **Testes Automatizados:** Utilização do pytest (configurado em pytest.ini) para validação de lógica de backend.
- ✓ **Testes Manuais:** Validação de fluxos de interface via Live Server.
- ✓ **Tratamento de Erros:** Feedback visual para o usuário (Flash messages).





Step

5489/7000 steps



Heart Rate

120 bpm

This Day

13940 steps

Light

149 bpm

Intensive

45 bpm

Aerobic

80 bpm

Anaerobic

82 bpm

DEMONSTRAÇÃO

Apresentação dos fluxos principais do sistema.

6:00

7:00

8:00

9:00

10:00

11:00

12:00

1:00

2:00

6:00 PM

7:00 PM

8:00 PM

9:00 PM

10:00 PM

11:00 PM



Blood Tracking



Daily healthy overview

(Vídeo disponível no repositório)



Cholesterol Levels



Iron Levels



Sugar Level



Heart Rate

Sunday, 10 Apr, 2025

11:00 AM



Max 78 kg

weight

75

| CONCLUSÕES

APRENDIZADOS

Implementação prática de autenticação (Hash/Sessions) e estruturação robusta de uma API RESTful.

DESAFIOS

Gerenciamento de sessão com Flask-Session e configuração de CORS para comunicação Front-Back.

PRÓXIMOS PASSOS

Implementar JWT para suporte mobile, migrar banco para PostgreSQL e adicionar gráficos interativos (Chart.js).



ACESSE O PROJETO

joaovictor915/FitTrack

O repositório contém documentação completa, scripts e testes.

IMAGE SOURCES



<https://betterexplained.com/wp-content/uploads/rails/mvc-rails.png>

Source: stackoverflow.com



<https://code.visualstudio.com/assets/docs/python/flask-tutorial/debug-local-variables.png>

Source: code.visualstudio.com



<https://cdn.dribbble.com/userupload/45740344/file/a79b8edcab26df09d7806b9db15b50fe.png?resize=1600x1200>

Source: dribbble.com