

|  | **Projeto de Desenvolvimento de Software - Projetão** |
| --- | --- |
| Shaypado | |
| 06/12/2023 | **Plano de Projeto** |

# Visão Geral

O projeto Shaypado é uma iniciativa que visa proporcionar uma experiência única e motivadora para os entusiastas e iniciantes de atividades físicas. Diante da crescente preocupação com a saúde e bem-estar, o Shaypado surge como uma solução abrangente para motivar os usuários a se engajarem em exercícios e acompanharem de perto seu progresso.

A era moderna trouxe consigo um estilo de vida sedentário, contribuindo para diversos problemas de saúde. O Shaypado nasce da necessidade de combater esse padrão, incentivando as pessoas a adotarem um estilo de vida ativo, buscando construir sua própria identidade e fidelizar seus usuários. A justificativa do projeto está ancorada na promoção da saúde física, oferecendo aos usuários uma ferramenta personalizável que os mantenha motivados, conectados e comprometidos com seus objetivos fitness.

# Atores / Papéis (Stakeholders)

| **Instrutor**:  Profissional que é responsável por fornecer treinamentos e orientações aos usuários.  **Ações Principais**:  Consultar o histórico dos alunos; Adicionar novos treinos para os alunos; Gerenciar turmas, etc. | **Usuário Comum:**  Pessoa que utiliza o aplicativo para treinamento físico e acompanhamento de sua evolução.  **Ações Principais**:  Consultar suas informações; Entrar em contato com instrutores; Cadastrar-se no sistema; Customizar seu pet no app; Receber lista de exercícios de um instrutor; Comprar planos; etc. | **Administrador**:  Os responsáveis pela gestão interna do app e administração dos sistemas.  **Ações Principais**:  Cadastrar exercícios para o app; Cadastrar treinos para o app; Gerenciar planos para usuários; Gerenciar usuários e instrutores cadastrados; etc. |
| --- | --- | --- |

.

# Histórias de Usuários

| **Plataforma** | **Usuário** | **Ação** | **História** |
| --- | --- | --- | --- |
| app/web | instrutor | Consultar | O instrutor pode consultar o histórico dos alunos |
| app/web | instrutor | Cadastrar | O instrutor pode se cadastrar no sistema |
| app/web | instrutor | Excluir | O instrutor pode excluir sua própria conta |
| app/web | instrutor | Editar | O instrutor pode editar suas informações |
| app/web | instrutor | Adicionar | O instrutor pode adicionar um novo treino para os alunos |
| app/web | instrutor | Modificar | O instrutor pode modificar suas configurações |
| app/web | instrutor | Adicionar | O instrutor pode adicionar alunos a turma |
| app/web | instrutor | Criar | O instrutor pode criar uma turma |
| app/web | instrutor | Remover | O instrutor pode remover um aluno da turma |
| app/web | instrutor | Excluir | O instrutor pode excluir uma turma |

| **Plataforma** | **Usuário** | **Ação** | **História** |
| --- | --- | --- | --- |
| app | Usuário comum | Consultar  Informações | O usuário pode consultar suas informações |
| app | Usuário comum | Consultar Instrutor | O usuário pode consultar as informações do instrutor |
| app | Usuário comum | Cadastrar | O usuário pode se cadastrar no sistema |
| app | Usuário comum | Editar | O usuário pode editar suas informações na conta |
| app | Usuário comum | Excluir | O usuário pode excluir sua própria conta |
| app | Usuário comum | Customizar | O usuário pode customizar o seu pet no app |
| app | Usuário comum | Cadastrar Exercícios | O usuário pode cadastrar seus próprios exercícios |
| app | Usuário comum | Editar Exercícios | O usuário pode editar sua lista de treinos para praticar |
| app | Usuário comum | Excluir Exercícios | O usuário pode excluir treinos da sua lista |
| app | Usuário comum | Publicar | O usuário tem a opção de publicar sua evolução |
| app | Usuário comum | Consultar Evolução | O usuário pode consultar sua evolução no app |
| app | Usuário comum | Receber | O usuário pode receber uma lista de exercícios pronta de um instrutor |
| app | Usuário comum | Utilizar IA | O usuário pode utilizar IA para verificar se está executando corretamente seus exercícios |
| app | Usuário comum | Utilizar IA | O usuário pode utilizar IA para verificar a quantidade de gordura corporal |
| app | Usuário comum | Comprar Plano | O usuário pode comprar um plano |
| app | Usuário comum | Comprar skills | O usuário pode comprar itens para customizar seu pet |
| app | Usuário comum | Modificar | O usuário pode alterar as configurações do app |
| app | Usuário comum | Sair | O usuário pode sair de uma turma |
| app | Usuário comum | Consultar | O usuário pode consultar instrutores próximos |
| app | Usuário comum | Recuperar | O usuário pode recuperar sua senha |
| app | Usuário comum | Feedback | O usuário pode compartilhar feedback sobre um treino |

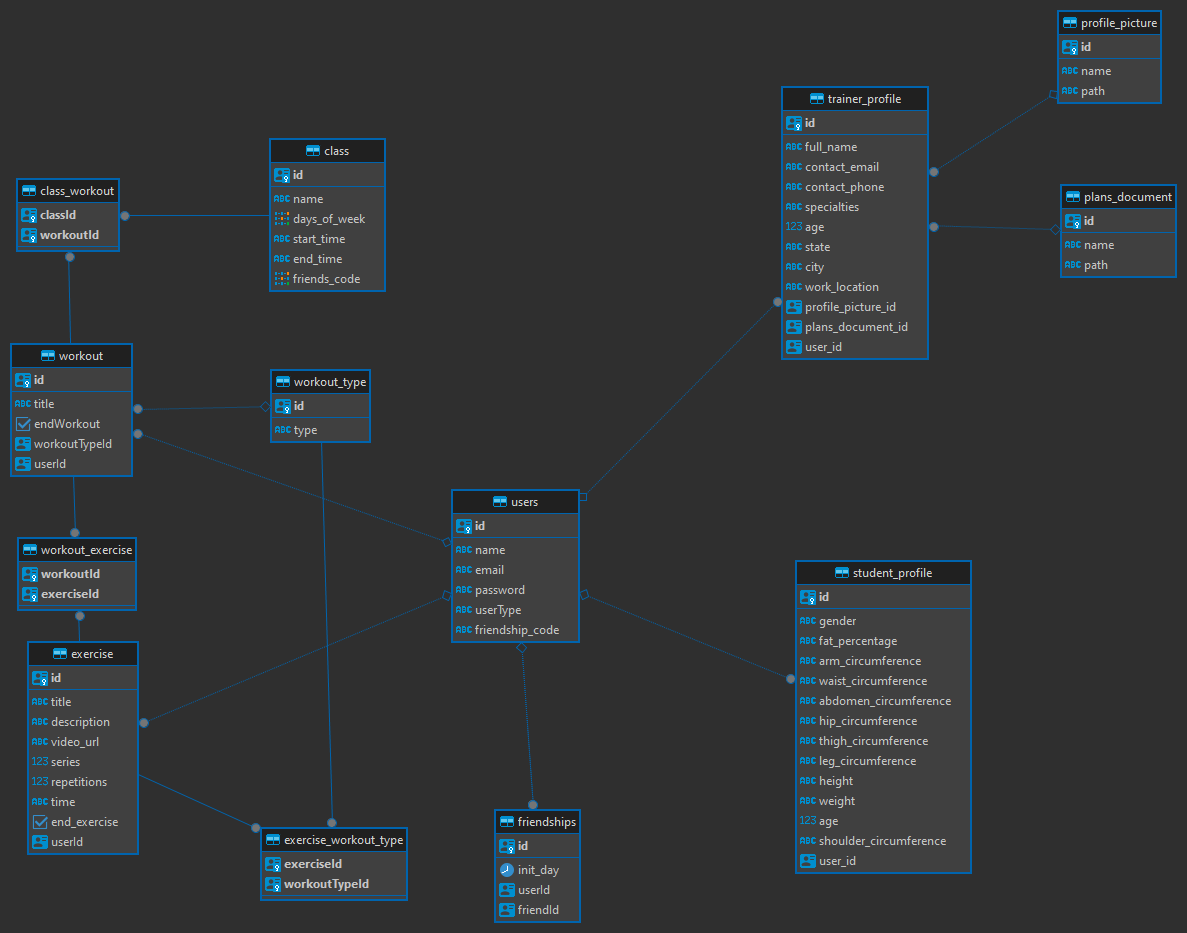
| **Plataforma** | **Usuário** | **Ação** | **História** |
| --- | --- | --- | --- |
| web | Administrador | Cadastrar Exercícios | O administrador pode cadastrar exercícios que estarão disponíveis no app |
| web | Administrador | Cadastrar treinos | O administrador pode cadastrar treinos que estarão disponíveis no app |
| web | Administrador | Gerenciar | O administrador pode gerenciar os planos para o usuário comum e instrutor |
| web | Administrador | Gerenciar | O administrador pode gerenciar os usuários e instrutores cadastrados no sistema |
| web | Administrador | Cadastrar pets | O administrador pode cadastrar os itens que serão vendidos na loja do app |

# Protótipo

[Link para o protótipo do Mobile](https://www.figma.com/file/qF13NQKSQWs12iMc4VJ3j6/shaypado---ihc?type=design&mode=design&t=6nqbj9AgAD7xXD7d-0)

[Link para o protótipo do Web](https://www.figma.com/file/qz8w23TopALiRdmgUuMKLe/shaypado?type=design&node-id=409-1239&mode=design)

# MER



# Arquitetura de Software

Qual a arquitetura do projeto e métodos de desenvolvimento

1. Mobile: Desenvolvimento Mobile nativo. Iremos utilizar a arquitetura Model-View-ViewModel(MVVM), que é a mais utilizada no desenvolvimento mobile e a arquitetura recomendada pela Google para desenvolvimento Android.
2. IA: python, Tensorflow, pytorch, opencv - Não há arquitetura definida pois é feita testando
3. Back-end: Desenvolvimento da RestAPI em Python utilizando o FastApi como framework - Uso da Arquitura MVC (Sem as View)
4. Front: Tecnologias React + JS, axios para requisições e React router doom para rotas

# Cronograma (Sprint Backlogs)

|  | **Macro Etapa** | **Data Início** | **Data Fim** |
| --- | --- | --- | --- |
| 1 | Definir o escopo, os objetivos e os requisitos do projeto. Escolher as ferramentas e as tecnologias a serem utilizadas. Dividir o projeto em módulos e atribuir as tarefas aos membros da equipe. | 27/11/2023 | 01/12/2023 |
| 2 | Desenvolver o protótipo da interface gráfica do usuário (GUI). E toda documentação inicial para desenvolvimento. | 04/12/2023 | 05/12/2023 |
| 3 | Reunião com Stakeholders para definição de funcionalidades. | 06/12/2023 | 06/12/2023 |
| 4 | Desenvolver o protótipo da interface gráfica do usuário (GUI). E toda documentação inicial para desenvolvimento. | 06/12/2023 | 08/12/2023 |
| 5 | Entrega da documentação da primeira VA | 08/12/2023 | 08/12/2023 |
| 6 | Desenvolvimento dos sistemas | 11/12/2023 | 15/12/2023 |
| 7 | Reunião com a equipe (Recap da Sprint) | 15/12/2023 | 15/12/2023 |
| 7 | Desenvolvimento dos sistemas | 18/12/2023 | 22/12/2023 |
| 8 | Reunião com a equipe (Recap da Sprint) | 22/12/2023 | 22/12/2023 |
| 8 | Férias de fim de ano | 23/12/2023 | 07/01/2024 |
| 9 | 1º Release parcial dos produtos | 08/01/2024 | 08/01/2024 |
| 10 | Desenvolvimento e correção de bugs na primeira release. | 08/01/2024 | 12/01/2024 |
| 11 | Reunião com stakeholders e possíveis interessados | 12/01/2024 | 12/01/2024 |
| 12 | Reunião com a equipe (Recap da Sprint) | 12/01/2024 | 12/01/2024 |
| 13 | Desenvolvimento dos sistemas. Implementar as melhorias e as alterações solicitadas pelo cliente. Atualizar o protótipo da GUI e documentação. | 15/01/2024 | 19/01/2024 |
| 14 | Reunião com a equipe (Recap da Sprint) | 19/01/2024 | 19/01/2024 |
| 15 | Desenvolvimento dos sistemas. Corrigir os bugs e as falhas encontradas. | 22/01/2024 | 26/01/2024 |
| 16 | Reunião com a equipe (Recap da Sprint) | 26/01/2024 | 26/01/2024 |
| 17 | Desenvolvimento dos sistemas. Corrigir os bugs e as falhas encontradas. | 29/01/2024 | 02/02/2024 |
| 18 | Reunião com a equipe (Recap da Sprint) | 02/02/2024 | 02/02/2024 |
| 19 | 2º Release parcial dos produtos | 05/02/2024 | 05/02/2024 |
| 20 | Reunião com stakeholders e possíveis interessados | 05/02/2024 | 05/02/2024 |
| 21 | Desenvolvimento e correção de bugs da segunda release. Analisar o feedback do cliente e identificar as correções e os ajustes finais. Revisar o código e a documentação do projeto. Realizar os testes de qualidade e de garantia do sistema. | 06/02/2024 | 08/02/2024 |
| 22 | Reunião com a equipe (Recap da Sprint) | 09/02/2024 | 09/02/2024 |
| 23 | Desenvolvimento das aplicações e correções de erros. | 15/02/2024 | 16/02/2024 |
| 24 | Reunião com a equipe (Recap da Sprint) | 16/02/2024 | 16/02/2024 |
| 25 | Desenvolvimento das aplicações e correções de erros. | 19/02/2024 | 23/02/2024 |
| 26 | Reunião com a equipe (Recap da Sprint) | 23/02/2024 | 23/02/2024 |
| 27 | 3º Release dos produtos - Versão beta | 26/02/2024 | 26/02/2024 |
| 28 | Desenvolvimento e correção de erros para versão final. | 26/02/2024 | 28/02/2024 |
| 29 | 4º Release dos produtos - Versão final | 29/02/2024 | 29/02/2024 |
| 30 | Entrega final dos produtos | 01/03/2024 | 01/03/2024 |

# Riscos

| **Riscos** | | |
| --- | --- | --- |
| 1 | Body Fat System | Há um risco em relação à entrega deste produto, pois atualmente não dispomos de uma base de dados disponível para uso. A implementação será realizada sem a utilização de treinamento de redes neurais, o que pode apresentar desafios. Inicialmente, a solução será experimental. Conforme os usuários utilizam o aplicativo, planejamos coletar dados para posteriormente treinar nossas redes neurais, aprimorando assim o produto ao longo do tempo. |
| 2 | App, API e Sistema Web | Devido à complexidade tanto do aplicativo móvel quanto do sistema back-end e web, existe a possibilidade de não conseguirmos implementar todas as funcionalidades dentro do prazo estimado. |

# Produtos Esperados

| **Produtos** | |
| --- | --- |
| 1 | App: Um aplicativo móvel destinado ao acompanhamento de exercícios físicos para indivíduos que praticam treinos independentes e querem organizar suas rotinas de treino ou buscam suporte, bem como para aqueles que preferem treinar com a orientação de um instrutor. O app oferece a capacidade para instrutores gerenciarem os treinos dos usuários de forma eficiente. |
| 2 | Sistema Web: Um sistema web projetado para que instrutores possam gerenciar seus alunos, acessar relatórios detalhados sobre o desempenho de cada aluno. Além disso, uma área administrativa que permite ao administrador gerenciar o produto, adicionar exercícios, informações sobre os pets, entre outros recursos. |
| 3 | API: A API é um componente vital deste ecossistema integrado, proporcionando uma estrutura back-end robusta que conecta o aplicativo móvel, o sistema web e o inovador Body Fat System. Desenvolvida para facilitar a comunicação entre esses elementos, a API oferece funcionalidades essenciais para o gerenciamento eficiente de entidades, possibilitando a interação fluida entre usuários, instrutores e administradores. |
| 4 | Body Fat System: O Body Fat System é uma solução inovadora que utiliza IA para calcular e monitorar a porcentagem de gordura corporal. Projetado para atender às necessidades de indivíduos preocupados com a saúde e fitness, o sistema será utilizado no aplicativo móvel para permitir o acompanhamento do histórico de medições para uma análise detalhada do progresso ao longo do tempo. |
| 5 | Página estática: A página estática do produto é uma fonte informativa e acessível para usuários em busca de uma visão geral rápida e abrangente sobre o produto. Com um design intuitivo e informações concisas, a página estática destaca os benefícios, recursos principais e casos de uso do produto. |

# Equipe do projeto

| **Nome** | **Função** | **E-mail** | **Telefone** |
| --- | --- | --- | --- |
| Ana Beatriz | Designer | anabiaavanderlei@gmail.com | (87) 9 8102-5842 |
| Gabriel Viana | IA | gabrielalmeidavn@gmail.com | (81) 9 8961-8295 |
| Gabriel Melo | Mobile | gabrielmelodemenezes@gmail.com | (81) 9 8202-7999 |
| Maria Virgínia | Back-end | mvmendonca.s3301@gmail.com | (87) 9 8138-4687 |
| Pedro Medeiros | Front-end | pemedeiros0110@gmail.com | (11) 9 5479-6825 |
| Pedro Vinícius | Mobile | pedro.vinicius.melo.silva@gmail.com | (87) 9 8106-7233 |
| Rogério Lacerda | Front-end | nenorogerinho@gmail.com | (81) 9 9657-7835 |
| Thiago Fabrício | Back-end | thiago.fabricio@ufape.edu.br | (87) 9 9945-3778 |
| Victor Hugo | Mobile | victor.hugooo99@gmail.com | (87) 9 9139-1066 |