

Fokus

I - Informações Gerais

I.I - Objetivo

O site Fokus tem como objetivo principal guiar os usuários que precisam definir claramente suas tarefas e metas, criando hábitos saudáveis. Com ele, os usuários conseguem criar e monitorar suas metas para atingir seus desejos pessoais, como ser mais saudável, melhorar seu condicionamento físico ou definir suas rotinas.

I.II - Personas

- Persona 1:

Seção	Detalhes da Persona
Nome	Márcio Silva
Citação	<i>"Preciso deixar minha rotina semanal bem definida, caso contrário, começo a procrastinar e a esquecer do que devo fazer."</i>
Comportamento	Usa o telefone para registrar seus deveres. Precisa de uma interface simples mas que seja completa para seu objetivo.
Objetivos	Registrar e acompanhar seus deveres recorrentes.
Dores/frustrações	Falta de uma interface eficiente. Sites simples demais que não atendem ao seu objetivo.
Necessidades	Sistema de notificação. Interface limpa e minimalista. Opções completas.

- Persona 2:

Seção	Detalhes da Persona
Nome	Rafaela Vitória
Citação	“Preciso começar a ter hábitos melhores. Escolhi mudar de vida e preciso de ajuda para me organizar e continuar motivada.”
Comportamento	Usa notas comuns para tentar organizar seus hábitos novos. Precisa de um site específico para esse caso.
Objetivos	Organizar seus novos hábitos para continuar motivada.
Dores/frustações	Falta de lembretes para suas tarefas. Falta de um lugar organizado para não se sentir perdida.
Necessidades	Sistema de notificação. Site organizado.

I.III - Resultados

Com a utilização do Fokus espera-se que os usuários consigam definir e monitorar seus hábitos, os ajudando na criação de novos modos de vida, bem como na regulação de sua rotina, por meio de uma interface minimalista, ainda assim completa.

II - Requisitos Técnicos

II.I - Requisitos Funcionais

- O usuário pode se cadastrar, conectar, fazer logout e deletar sua conta.
- O usuário pode atualizar seus dados.
- O usuário deve poder fazer um CRUD básico de Hábitos.
- O usuário deve poder fazer um CRUD básico de Metas.
- O usuário deve poder trocar seu nome, email e senha.

- O usuário deve poder separar suas metas por tags.

II.II - Requisitos não Funcionais

- Ações de autenticação e cadastro devem demorar menos de 4 segundos.
- O usuário deve entender perfeitamente o que pode e o que não pode fazer na utilização do site.

III - Design

III.I - Identidade Visual

Para a construção da identidade visual, foi observado que usuários preferem um site mais minimalista, com uma interface limpa, mas que ainda carregue opções suficientes para conseguir fazer tarefas um pouco mais complexas.

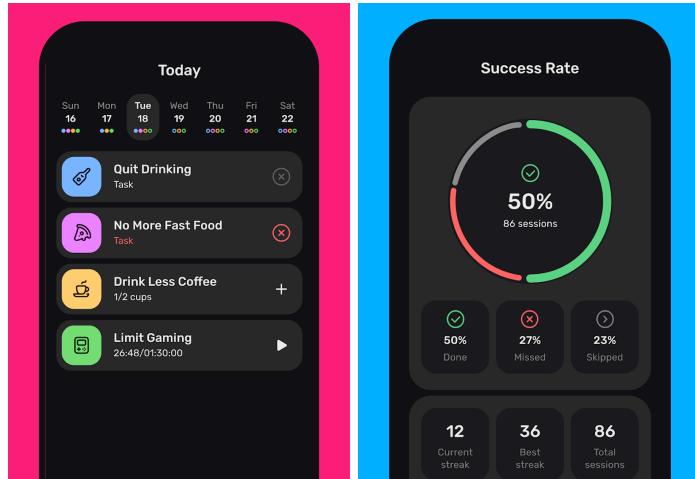
A paleta de cores do Fokus irá se basear em tons de verdes, sendo a cor de destaque principal, e tons de carmesim, sendo a cor de destaque secundária:



III.II - Análise e Pesquisa

Antes de iniciar o site, foi feita uma pesquisa em sites e aplicativos semelhantes para entender seus pontos fortes e fracos, visando um diferencial do Fokus em relação a outros produtos no mercado.

- **Motivated: Habit Tracker:** o primeiro produto observado foi o aplicativo “Motivated: Habit Tracker”. No aplicativo, é possível observar uma interface minimalista e objetiva, o que reflete os valores que se procuram para o site.

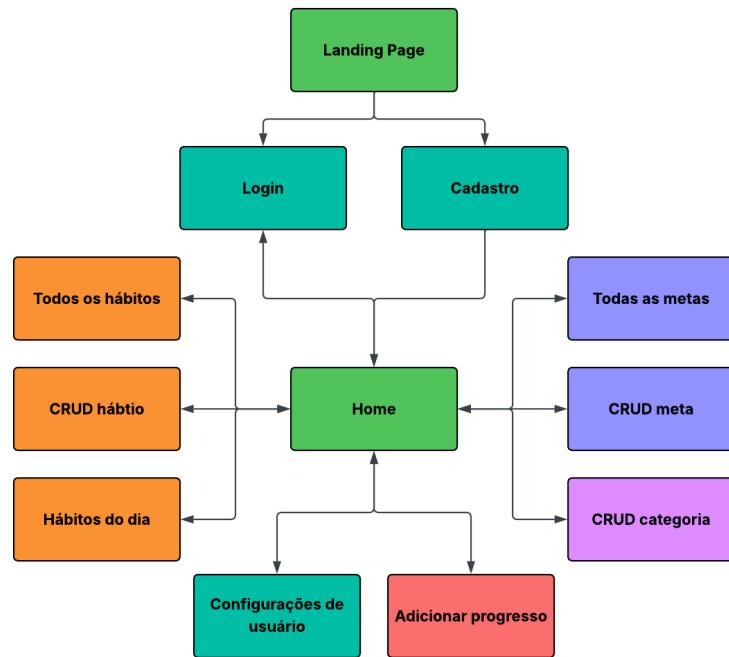


III.III - Fluxograma das Páginas

Com uma visão inicial de como o site deve se comportar visualmente, fica necessária uma representação de baixo nível do fluxograma de navegação entre as páginas. Considerando um nível de protótipo inicial, as seguintes páginas participarão do fluxograma:

- Landing Page.
- Cadastro.
- Login.
- Home.
- Ver detalhes/editar configurações de usuário.
- Ver detalhes/adicionar/editar hábitos.
- Ver detalhes/adicionar/editar metas.
- Ver tarefas cadastradas.
- Ver metas cadastradas.

Com isso em mente, o seguinte fluxograma foi criado:



IV- Banco de Dados

IV.I - Definição dos Schemas

Após a definição do fluxograma, foi iniciada a parte relacionada ao banco de dados do Fokus. Inicialmente, foi definida os schemas presentes e seus atributos, ambos representados abaixo através de uma tabela:

User:

User	
_id	ObjectId
name	String
email	String
password	String
themeMode	Enum('light', 'dark')

Habit:

Habit	
_id	ObjectId
userId	ObjectId (ref: 'User')
title	String
progressImpactValue?	Number
unitOfMeasure?	String
weekDays	Array[Enum('seg', 'ter', 'qua', 'qui', 'sex', 'sab', 'dom')]
Reminder?	Date Null
streak	Number
bestStreak	Number
color	String (Hexadecimal)
icon	String (React Icon name)

Goal:

Goal	
_id	ObjectId
userId	ObjectId (ref: 'User')
categoryId	ObjectId (ref: 'Category')
title	String
type	Enum('quant', 'quali')
currentValue	Number?
targetValue	Number?
unitOfMeasure	String?
tasks	Array[ObjectId] (ref: 'Task')
deadline	Date?
isActive	Boolean
Color	String (Hexadecimal)

Progress Log:

ProgressLog	
_id	ObjectId
goalId	ObjectId (ref: 'Goal')
valueAchieved	Number
dateAchieved	Date

Category:

Category	
_id	ObjectId
userId	ObjectId (ref: 'User')
name	String

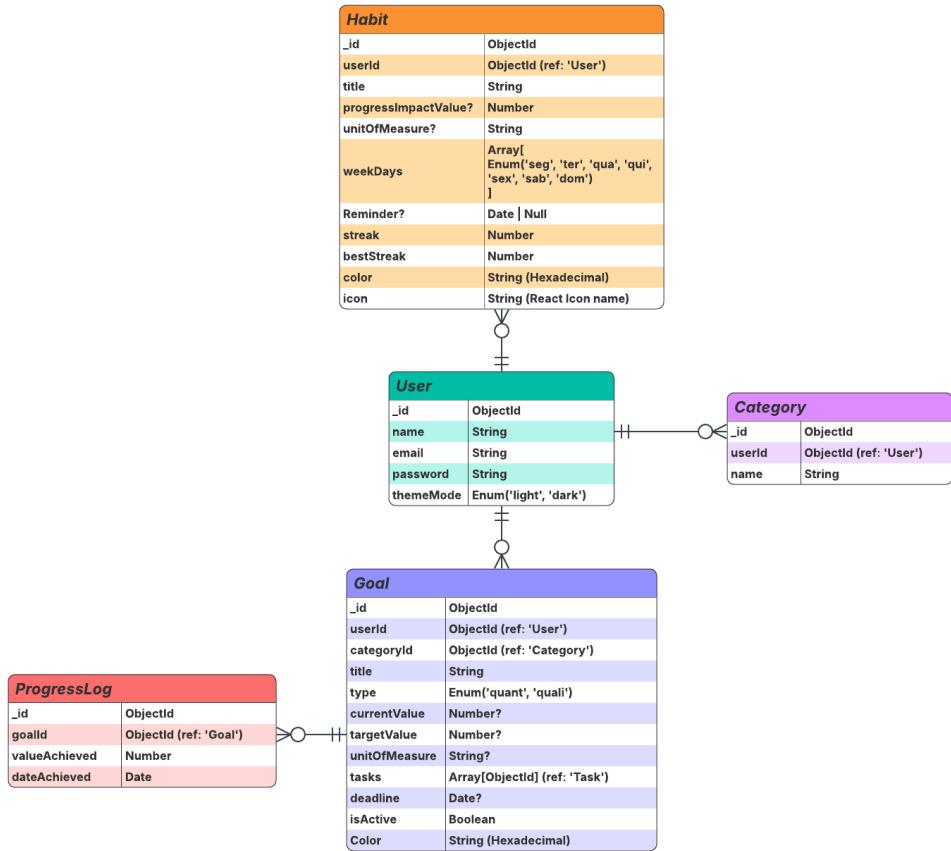
IV.II - Definição dos Índices

Para melhorar o tempo de resposta do site, é importante a utilização de índices no MongoDB, que agilizam o processo de busca dos dados requisitados. Pensando nas ações mais comuns que o site pedirá, os seguintes índices fazem-se necessários:

Schema	Índice	Objetivo
<i>Goal</i>	createdAt: 1	Filtrar metas atingidas ou novas
<i>Goal</i>	deadline: 1	Filtrar data limites que estão próximas
<i>Goal</i>	tasks: 1	Atualizar eficientemente as metas após realizar uma tarefa
<i>Goal</i>	categoryId: 1	Filtrar metas por categorias
<i>Habit</i>	createdAt: 1	Filtrar hábitos antigos ou novos
<i>Habit</i>	userId: 1	Filtrar hábitos do usuário
<i>Habit</i>	userId: 1 & weekDays: 1	Filtrar hábitos por dia da semana

IV.III - Relacionamento entre Schemas

Com os schemas bem definidos, o próximo passo foi definir como os schemas iriam conversar entre si, tarefa realizada através de um diagrama graficamente parecido com o Diagrama Entidade-Relacionamento (DER), ressaltando que, no caso do Fokus, o banco de dados utilizado será o MongoDB, um banco não-relacional, ao contrário do que sugere o DER:



V- Servidor

V.I - Definição das Rotas

Por fim, Foi preciso definir as rotas presentes no site e o que elas esperam no seu corpo da requisição e o que entregam no corpo da resposta, além de seus métodos e suas URLs:

- **Rotas para ações de cadastro e login:**
 - POST auth/register/
 - Precisa dos dados do novo usuário no corpo da requisição.
 - Retorna o novo usuário e salva seu token no Session Storage.
 - POST auth/login/
 - Precisa dos dados do usuário já cadastrado no corpo da requisição.

- Retorna o usuário e salva seu token no Session Storage.

- **Rotas para ações de ‘User’:**
 - GET users/
 - Rota protegida.
 - Sem corpo na requisição.
 - Retorna todos os usuários registrados no corpo da resposta.
 - GET users/<userId>
 - Sem corpo na requisição.
 - Precisa do ‘userId’ nos parâmetros da requisição.
 - Retorna o usuário encontrado ou ‘null’ se não for encontrado.
 - PATCH users/<userId>
 - Precisa dos dados atualizados do antigo usuário no corpo da requisição.
 - Precisa do ‘userId’ nos parâmetros da requisição.
 - Retorna o usuário atualizado ou ‘null’ se não for encontrado.
 - DELETE users/<userId>
 - Sem corpo na requisição
 - Precisa do ‘userId’ nos parâmetros da requisição.
 - Retorna o usuário deletado ou ‘null’ se não for encontrado.

- **Rotas para ações de ‘Goal’:**
 - GET goals/
 - Rota protegida.
 - Sem corpo na requisição.
 - Retorna todas as metas registradas no corpo da resposta.
 - GET goals/categories/<categoryId>
 - Sem corpo na requisição.
 - Retorna todas as metas relacionadas com o ID de categoria passado no corpo da resposta.

- GET goals/<goalId>
 - Sem corpo na requisição.
 - Precisa do ‘goalId’ nos parâmetros da requisição.
 - Retorna a meta encontrada ou ‘null’ se não for encontrada.
- POST goals/
 - Precisa dos dados da nova meta no corpo da requisição.
 - Retorna a meta criada.
- PATCH goals/<goalId>
 - Precisa dos dados atualizados da antiga meta no corpo da requisição.
 - Precisa do ‘goalId’ nos parâmetros da requisição.
 - Retorna a meta atualizada ou ‘null’ se não for encontrada.
- DELETE goals/<goalId>
 - Sem corpo na requisição
 - Precisa do ‘goalId’ nos parâmetros da requisição.
 - Retorna a meta deletada ou ‘null’ se não for encontrada.

- **Rotas para ações de ‘Category’:**

- GET categories/
 - Rota protegida.
 - Sem corpo na requisição.
 - Retorna todas as categorias registradas no corpo da resposta.
- GET categories/<categoryId>
 - Sem corpo na requisição.
 - Precisa do ‘categoryId’ nos parâmetros da requisição.
 - Retorna a categoria encontrada ou ‘null’ se não for encontrada.
- POST categories/
 - Precisa dos dados da nova categoria no corpo da requisição.
 - Retorna a categoria criada.

- PATCH categories/<categoryId>
 - Precisa dos dados atualizados da antiga categoria no corpo da requisição.
 - Precisa do ‘categoryId’ nos parâmetros da requisição.
 - Retorna a categoria atualizada ou ‘null’ se não for encontrada.
- DELETE categories/<categoryId>
 - Sem corpo na requisição
 - Precisa do ‘categoryId’ nos parâmetros da requisição.
 - Retorna a categoria deletada ou ‘null’ se não for encontrada.

- **Rotas para ações de ‘Habit’:**

- GET habit/
 - Rota protegida.
 - Sem corpo na requisição.
 - Retorna todas as tarefas registradas no corpo da resposta.
- GET habit/<habitId>
 - Sem corpo na requisição.
 - Precisa do ‘habitId’ nos parâmetros da requisição.
 - Retorna a tarefa encontrada ou ‘null’ se não for encontrada.
- POST habit/
 - Precisa dos dados da nova tarefa no corpo da requisição.
 - Retorna a tarefa criada.
- PATCH habit/<habitId>
 - Precisa dos dados atualizados do antigo hábito no corpo da requisição.
 - Precisa do ‘habitId’ nos parâmetros da requisição.
 - Retorna a tarefa atualizada ou ‘null’ se não for encontrada.
- DELETE habit/<habitId>
 - Sem corpo na requisição
 - Precisa do ‘habitId’ nos parâmetros da requisição.
 - Retorna a tarefa deletada ou ‘null’ se não for encontrada.