

CHECKLIST.com.br

**TAREFAS E ROTINAS**

**DOCUMENTO DE IDEALIZAÇÃO 01**

**AUTOR: GUILHERME BENJAMIN ÚLTIMA ATUALIZAÇÃO: 03/12/2024**

**HISTÓRICO DE REVISÕES DO DOCUMENTO**

|  |  |  |  |
| --- | --- | --- | --- |
| **DATA** | **VERSÃO** | **DESCRIÇÃO DA ALTERAÇÃO** | **AUTOR** |
| 03/12/2024 | 1 | CRIAÇÃO DE IDEIA E DO DOCUMENTO | GUILHERME BENJAMIN |
| 05/12/2024 | 1 | DESENVOLVIMENTO DE ANÁLISE | GUILHERME BENJAMIN |
| 09/12/2024 | 1 | ENTREGA DO TRABALHO | GUILHERME BENJAMIN |
| 22/12/2024 | 2 | REVISÃO DA SEGUGUNDA VERSÃO | GUILHERME BENJAMIN |
| 23/12/2024 | 2 | IMPORTANDO MODELOS DO BANCO DE DADOS | GUILHERME BENJAMIN |
|  |  |  |  |

**NOMES DOS DESENVOLVEDORES DO PROJETO.**

|  |  |  |
| --- | --- | --- |
| **PAPEL** | **NOME** | **EMAIL** |
| ANALISTA DE REQUISITOS | GUILHERME BENJAMIN | [guihermeb.ltda@gmail.com](mailto:guihermeb.ltda@gmail.com) |

**Tema:**

Desenvolver um site de gerenciamento de tarefas:

**Objetivo Geral:**

Desenvolver um sistema para gerenciar suas tarefas e rotinas diárias.

**Objetivo Específicos:**

* Fazer um estudo sobre a os problemas enfrentados pelo estudantes de vários niveis de ensino como fundamental, médio e ou superior, bem como a falta de organização de trabalhos, atividades e como se pode solucionar;
* Fazer o levantamento dos requisitos funcionais e não funcionais;
* Selecionar os diagramas mais adequados com o perfil levantado;
* Fazer os wireframe’s para validar os requisitos do sistema;
* Implementar o front-end e back-end;

**Introdução:**

Atualmente, tanto pessoas quanto profissionais enfrentam dificuldades em organizar suas rotinas diárias, o que impacta negativamente na agilidade e eficiência das atividades realizadas. A "rotina" refere-se às atividades diárias habituais que alguém realiza, enquanto as "tarefas de trabalho" são as responsabilidades específicas associadas a uma função ou área de atuação. Muitos negligenciam a importância de manter uma rotina estruturada, gerando acúmulo de tarefas, prazos perdidos e baixa produtividade.

Principalmente estudantes de todas as origens enfrentam dificuldades para organizar suas tarefas e trabalhos acadêmicos, especialmente quando precisam gerenciar prazos e atividades de forma eficiente. No entanto, a maioria das ferramentas disponíveis, como Trello ou Asana, pode parecer complexa e excessiva para suas necessidades. Por isso, é essencial contar com uma solução simples e intuitiva, que facilite a organização da rotina de estudos sem exigir um longo tempo de aprendizado ou configurações complicadas.

Essa situação é também problemática em áreas que demandam organização rigorosa, como saúde, educação, gestão de projetos e negócios. Identificou-se, portanto, a necessidade de um sistema que auxilie na organização e otimização das rotinas e tarefas de trabalho, oferecendo ferramentas práticas e acessíveis para que usuários consigam planejar, acompanhar e ajustar suas atividades de forma eficiente, criando assim uma abordagem mais personalizada para cada usuário, dependendo de sua área de atuação.

# Requisitos do sistema

### R1- POSSIBILITAR A ORGANIZAÇÃO DE ROTINAS DIÁRIAS:

O sistema deverá oferecer uma interface intuitiva onde o usuário possa cadastrar, editar e acompanhar suas rotinas diárias de forma estruturada. No site será possível organizar tarefas por categorias, prioridades e horários.

#### R1.1 - DISPONIBILIZAR CADASTRO DE TAREFAS

Na interface do sistema, o usuário deverá ter acesso a um formulário para cadastrar novas tarefas. O cadastro deverá incluir os campos:

* + Nome da tarefa
  + Descrição detalhada
  + Categoria (ex.: Trabalho, Pessoal, Estudos)
  + Data e hora de início
  + Prioridade (Baixa, Média, Alta)

#### R1.2 - PERMITIR EDIÇÃO DE TAREFAS

As tarefas cadastradas deverão ser editáveis. O usuário poderá modificar os detalhes da tarefa, inclusive sua data, hora e prioridade.

#### R1.3 - EXIBIR VISÃO GERAL DA ROTINA

O sistema deverá exibir uma visão geral das tarefas cadastradas, organizadas por dia, em um formato específico. Essa visão deverá destacar tarefas de alta prioridade.

### R2 - OFERECER ALERTAS E NOTIFICAÇÕES:

O sistema deverá enviar alertas e notificações para lembrar o usuário das tarefas ao longo do dia.

#### R2.1 - CONFIGURAÇÃO DE ALERTAS

O usuário poderá configurar alertas personalizados para cada tarefa, como lembretes antes do horário da(s) tarefa(s).

#### R2.2 - ENVIAR LEMBRETES AUTOMÁTICAS

O sistema deverá enviar notificações no horário escolhido pelo usuário, via e-mail ou pelo próprio site quando estiver aberto.

### R3 – DEFINIR TEREFAS COM MAIS PRIORIDADES:

O sistema deverá oferecer recursos que ajudem o usuário a priorizar suas tarefas.

#### R3.1 - EXIBIR LISTA DE PRIORIDADES

Na visão geral, as tarefas deverão ser destacadas por nível de prioridade, utilizando cores ou ícones diferentes.

#### R3.2 – OTIMIZAÇÃO DE SISTEMA

O sistema poderá ser otimizado nas atividades caso existam mais de uma atividade no mesmo horário ou muitas tarefas acumuladas no mesmo dia.

### R4 - PERMITIR O REGISTRO DAS TAREFAS CONCLUIDAS OU EM ANDAMENTO:

O sistema deverá registrar o status de cada tarefa e permitir que o usuário acompanhe seu progresso.

#### R4.1 – MOSTRAR ATIVIDADES CONCLUDAS

O usuário poderá marcar tarefas como concluídas na página inicial.

#### R4.2 - GERAR RELATÓRIOS DE PRODUTIVIDADE

O sistema deverá oferecer relatórios semanais ou mensais, indicando o número de tarefas concluídas, atrasadas e pendentes.

### R5 - GARANTIR ACESSIBILIDADE:

O sistema deverá ser acessível tento via navegador e mobile, com uma interface de fácil utilização.

#### R5.1 - DISPONIBILIZAR LOGIN E SINCRONIZAÇÃO

O usuário deverá ter uma conta única para acessar o sistema, com sincronização automática de dados.

#### R5.2 - OFERECER MODO OFFLINE

Será possível cadastrar e visualizar tarefas mesmo sem conexão à internet. Dados serão sincronizados automaticamente quando a conexão for restabelecida

# Requisitos não Funcionais:

### R1- UM BOM DESENPENHO:

* **R1.1 - TEMPO DE RESPOSTA**: O sistema deverá responder a comandos do usuário em no máximo 2 segundos.
* **R1.2 - ESTALIBILIDADE**: O sistema deve ser capaz de suportar um número crescente de usuários sem perda significativa de desempenho.

### R2- TER SEGURANÇA:

* **R2.1 - AUTENTICAÇÃO**: Os usuários deverão ser autenticados de maneira segura, utilizando senhas fortes e, idealmente, autenticação multifator.
* **R2.2 - PROTEÇÃO DE DADOS**: Todas as informações do usuário deverão ser armazenadas de forma segura, protegendo o usuario contra acessos.

### R3- SER MAIS ACESSIVEL:

* **R3.1 - INTERFACE INTUITUITIVA**: A interface do sistema deverá ser fácil de navegar, com opções claramente rotuladas e fluxos de trabalho intuitivos.
* **R3.2 - ACESSIBILIDADE**: O sistema deverá ser acessível a qualquer tipo de usuario, conforme as diretrizes de acessibilidade da Web.

### R4- SEGURANÇA PARA O USUARIO:

* **R4.1 - DISPONIBILIDADE**: O sistema deverá ter um tempo de atividade de 99.9%, garantindo alta disponibilidade aos usuários.
* **R4.2 - CORRIJIR POSSIVEIS ERROS**: Em caso de falhas, o sistema deve ser capaz de recuperar-se rapidamente, garantindo que os dados dos usuários não sejam perdidos.

### R5- SEGURANÇA SEMPRE ATUALIZADA:

* **R5.1 - FACILIDADES DE ATUALIZAÇÃO**: O código-fonte do sistema deverá ser modular e bem documentado para facilitar atualizações e correções futuras.
* **R5.2 - FACILIDADES NA MANUNTENÇÃO**: O sistema deverá contar com mecanismos de monitoramento e geração de logs para ajudar na identificação e solução de problemas.

# Requisitos Não Funcionais: Ferramentas Utilizadas:

Para desenvolver o sistema, vamos usar várias ferramentas que ajudam em diferentes partes do processo. Abaixo, destaco como cada uma delas funciona.

## R1- Microsoft Visual Studio Code (VS Code):

O **VS Code** é um editor de texto para escrever e editar código. Ele é leve, fácil de usar e tem várias extensões para ajudar no desenvolvimento.

* **R1.1- ESCRITA DE CÓDIGO:** Facilita escrever e organizar o código e destaque de sintaxe.
* **R1.2- DEPURAÇÃO:** Permite encontrar e corrigir erros no código.
* **R1.3- INTEGRAÇÃO COM GIT:** Facilita o controle de versão do código.

## R2- MySQL:

O **MySQL** é um sistema de banco de dados usado para armazenar dados.

* **R2.1- ARMAZENAMENTO DE DADOS:** Guarda informações de forma organizada.
* **Consultas:** Usamos SQL para fazer perguntas ao banco de dados, como buscar ou atualizar dados.
* **R2.2- SEGURANÇA:** Ajuda a proteger os dados com permissões e backups.
* **R2.3- FACILIDADE:** É uma ferramenta de criação de banco de dados mais fácil de ser usado tanto para modelagem como para a elaboração de um base dados mais simples.

## R3- APIs do Google:

As **APIs do Google** são ferramentas que permitem adicionar serviços do Google no seu sistema, como o Google Tasks ou login com Google.

* **R3.1- INTEGRAÇÃO:** Permite usar funcionalidades integradas ao google tasks para melhor funcionamento do sistema a ser integrado permitindo uma melhora nas informações que seram transmitidas para que o usuario tenha uma ótima experiência de uso do mesmo.
* **R3.2- AUTENTICAÇÃO:** Usa o Google para fazer login e autenticar usuários no sistema.

## R4- XAMPP:

O **XAMPP** é um pacote de programas que ajuda a testar o sistema localmente antes de colocá-lo no ar.

* **R4.1- SERVIDOR LOCAL:** Inclui o Apache para testar o site no seu computador.
* **R4.2- BANCO DE DADOS LOCAL:** Vem com o MySQL para armazenar dados localmente.
* **R4.3- FACILIDADE:** É fácil de instalar e configurar para começar a testar o sistema.

[**PROTÓTIPOS DE INTERFACE:**](https://www.analisederequisitos.com.br/requisitos-funcionais-e-requisitos-nao-funcionais-o-que-sao/)

*Figura 1 Exemplo de protótipo da tela de cadastro:*



*Figura 2 Exemplo de protótipo da tela de cadastro:*

Interface gráfica do usuário, Aplicativo

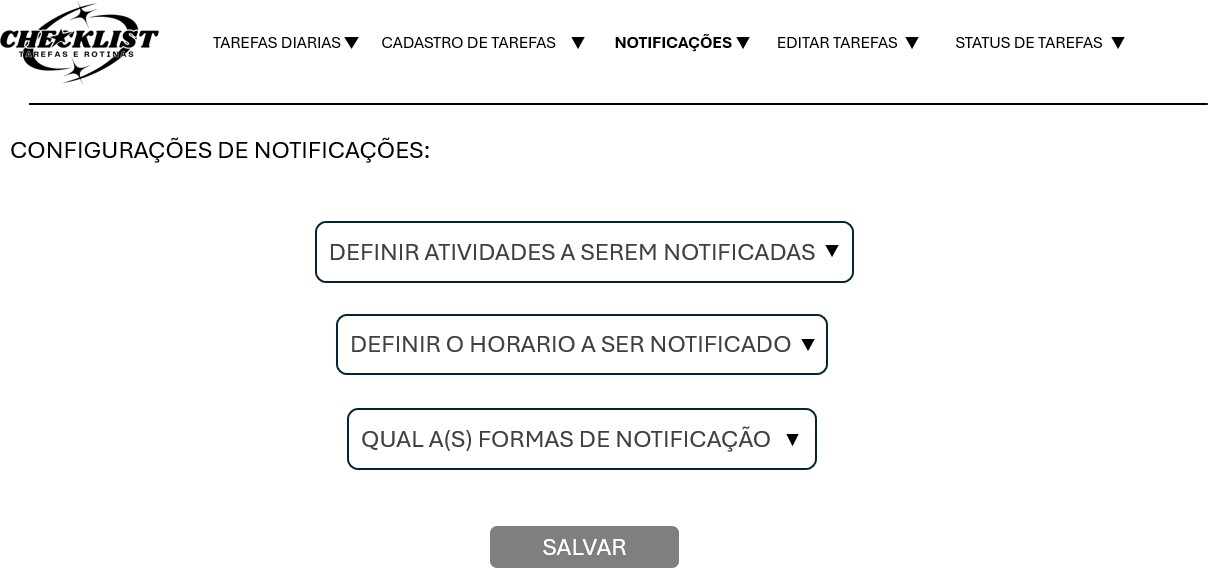
O conteúdo gerado por IA pode estar incorreto.

*Figura 3 Exemplo de protótipo da tela visão geral:*

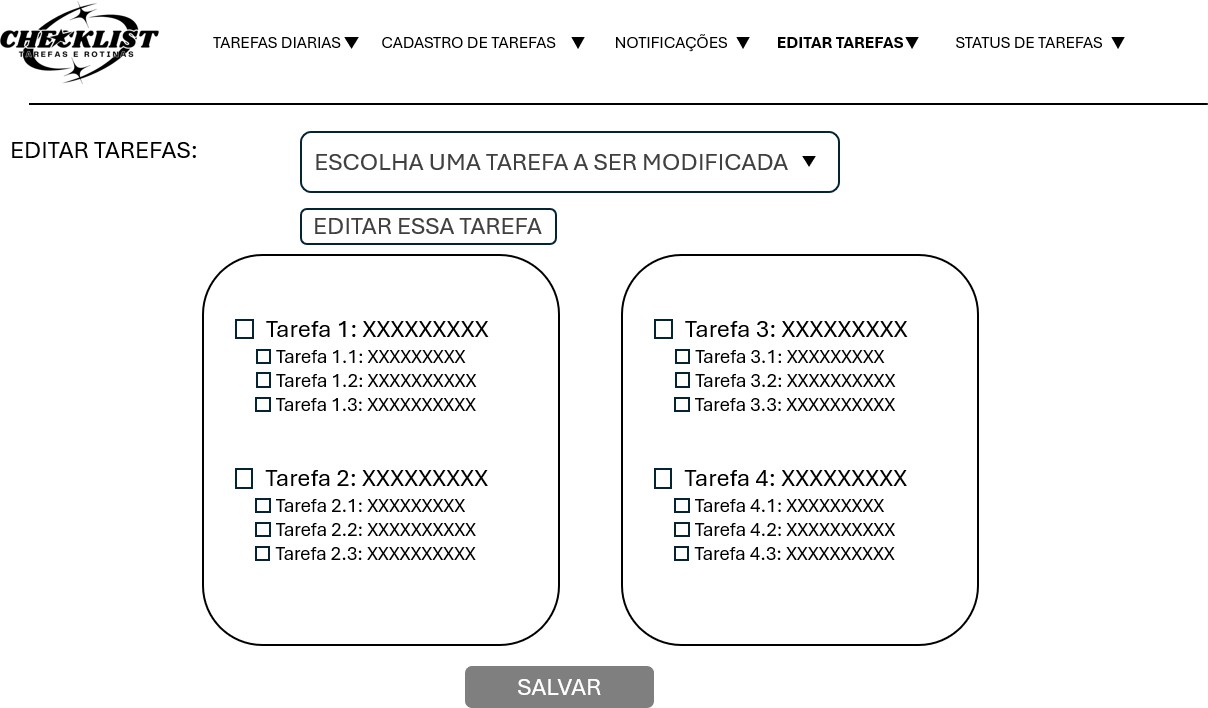
*Figura 4 Exemplo de protótipo da tela de cadastro de tarefas:*



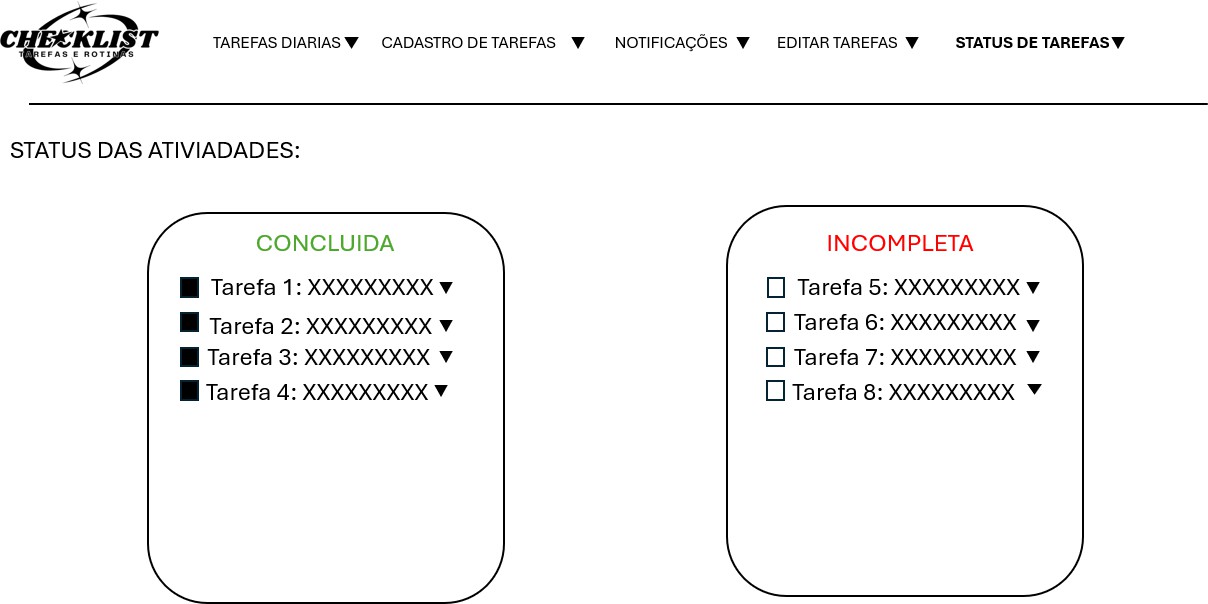
*Figura 5 Exemplo de protótipo da tela de notificações:*

**

*Figura 6 Exemplo de protótipo da tela de edição de tarefas:*



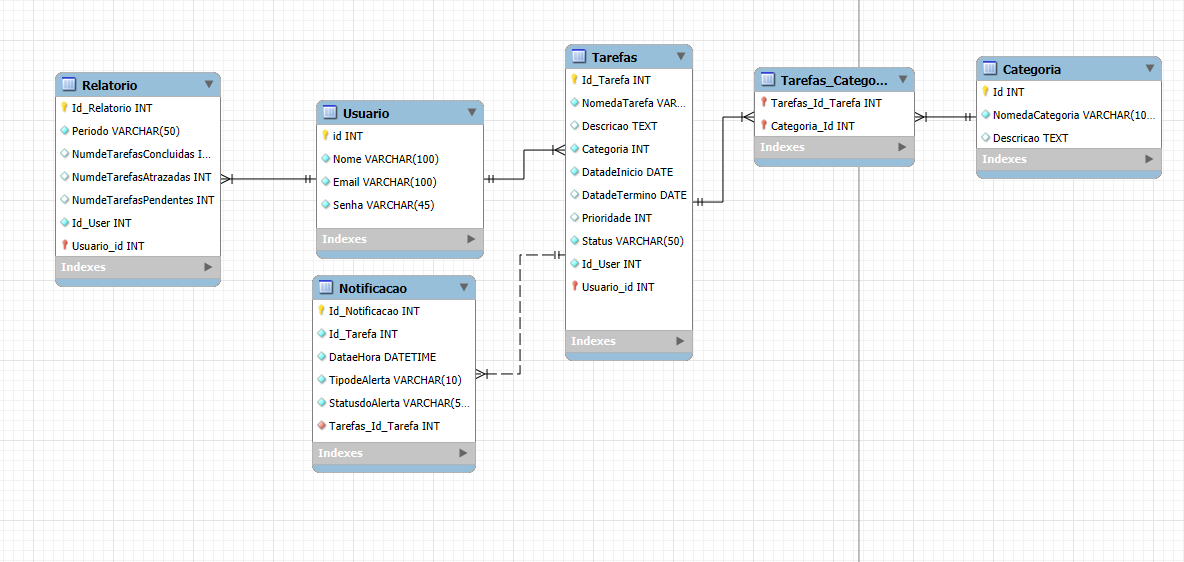
*Figura 7 Exemplo de protótipo da tela de status da tarefa:*

**

# Ilustrações da modelagem do Banco de Dados:

## 1°Ilustração caso de uso:

**2°Ilustração modelagem de entidades e tabelas:**

****

**Ambas as modelagens serão utilizadas para ter uma melhor base de dados e tomar como base no desenvolvimento do projeto.**