

## Requisitos do sistema

### RF1- 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.

#### RF1.1 - 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)

#### RF1.2 - PERMITIR EDIÇÃO DE TAREFAS

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

#### RF1.3 - 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.

### RF2 - OFERECER ALERTAS E NOTIFICAÇÕES:

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

#### RF2.1 - CONFIGURAÇÃO DE ALERTAS

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

#### RFF2.2 - ENVIAR LEMBRETES AUTOMÁTICAS

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

### **RF3 - DEFINIR TAREFAS COM MAIS PRIORIDADES:**

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

#### **RF3.1 - EXIBIR LISTA DE PRIORIDADES**

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

#### **RF3.2 - 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.

### **RF4 - PERMITIR O REGISTRO DAS TAREFAS CONCLUÍDAS OU EM ANDAMENTO:**

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

#### **RF4.1 - MOSTRAR ATIVIDADES CONCLUÍDAS**

O usuário poderá marcar tarefas como concluídas na página inicial.

#### **RF4.2 - GERAR RELATÓRIOS DE PRODUTIVIDADE**

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

### **RF5 - GARANTIR ACESSIBILIDADE:**

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

#### **RF5.1 - DISPONIBILIZAR LOGIN E SINCRONIZAÇÃO**

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

#### **RF5.2 - 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:

### RNF1- UM BOM DESEMPENHO:

- **RNF1.1 - TEMPO DE RESPOSTA:** O sistema deverá responder a comandos do usuário em no máximo 2 segundos.
- **RNF1.2 - ESTABILIDADE:** O sistema deve ser capaz de suportar um número crescente de usuários sem perda significativa de desempenho.

### RNF2- TER SEGURANÇA:

- **RNF2.1 - 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 usuário contra acessos.

### RNF3- SER MAIS ACESSÍVEL:

- **RNF3.1 - INTERFACE INTUITIVA:** A interface do sistema deverá ser fácil de navegar, com opções claramente rotuladas e fluxos de trabalho intuitivos.
- **RNF3.2 - ACESSIBILIDADE:** O sistema deverá ser acessível a qualquer tipo de usuário, conforme as diretrizes de acessibilidade da Web.

### RNF4- SEGURANÇA PARA O USUÁRIO:

- **RNF4.1 - DISPONIBILIDADE:** O sistema deverá ter um tempo de atividade de 99.9%, garantindo alta disponibilidade aos usuários.
- **RNF4.2 - CORRIGIR POSSÍVEIS ERROS:** Em caso de falhas, o sistema deve ser capaz de recuperar-se rapidamente, garantindo que os dados dos usuários não sejam perdidos.

### RNF5- SEGURANÇA SEMPRE ATUALIZADA:

- **RNF5.1 - FACILIDADES DE ATUALIZAÇÃO:** O código-fonte do sistema deverá ser modular e bem documentado para facilitar atualizações e correções futuras.
- **RNF5.2 - FACILIDADES NA MANUTENÇÃO:** 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.

### RNF1- 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.

- **RNF1.1- 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.

### RNF2- 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.

### RNF3- 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.

- **RNF3.1- INTEGRAÇÃO:** Permite usar funcionalidades integradas ao google tasks para melhor funcionamento do sistema a ser integrado permitindo uma melhora nas informações que serem transmitidas para que o usuario tenha uma ótima experiência de uso do mesmo.
- **RNF3.2- AUTENTICAÇÃO:** Usa o Google para fazer login e autenticar usuários no sistema.

### RNF4- XAMPP:

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

- **RNF4.1- SERVIDOR LOCAL:** Inclui o Apache para testar o site no seu computador.
- **RNF4.2- BANCO DE DADOS LOCAL:** Vem com o MySQL para armazenar dados localmente.
- **RNF4.3- FACILIDADE:** É fácil de instalar e configurar para começar a testar o sistema.

**RNF5- SISTEMA DEVERÁ TER UM PROTEÇÃO CONTRA MALWARES:**

## Processo de Teste

### 1. Verificação e Validação:

#### 1.1 Referência para testes: ISO 9126.

#### 1.2 Itens abordados:

##### 1.2.1 Funcionalidade:

1.2.1.1 RF1: Trata sobre a função de organização do projeto para uma melhor visão do sistema que será produzido para entendimento do conceito sobre a rotina diária, o mesmo se relaciona com o RNF1 para verificar e validar os presentes itens foi utilizado no conceito de Caixa Preta:

- Teste funcional onde o testador avalia se os itens apresentados nos requisitos foram adotados no sistema preparo para organizar as tarefas/rotina diária;
- Na tela primeira tela de cadastro foi testado o requisito de cadastro de usuario onde o (testador) criou um registro de Nome, Email e Senha e ao clicar em registrar as informações apareceria uma mensagem de cadastro bem sucedido.

- 
- Na tela de Login é onde o usuário tem a possibilidade de logar e ter acesso ao sistema que foi projetado, onde o (testador) deve inserir o Email e a Senha previamente cadastrados para ter o devido acesso ao sistema e ao clicar em fazer login com as devidas informações inseridas o sistema deveria o redirecionar para a próxima tela de visão geral assim descrito no RF1.

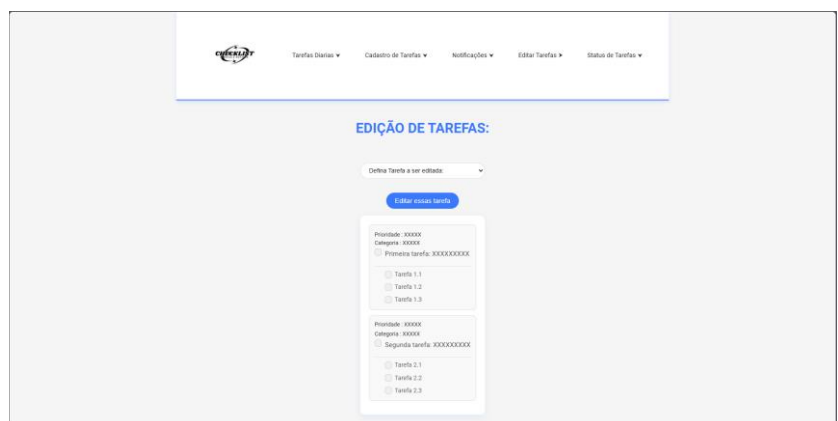
- 
- Na tela de visão geral de tarefas o (testador) deverá analisar se as tarefas estão bem organizadas por seu grau de prioridade e fácil de identificar as tarefas cadastradas pelo usuário, o (testador) deverá analisar se as caixas de marcação estão devidamente funcionais e analisar se ao marcar uma caixa de conclusão e clicar em marcar como concluído a tarefa irá para outra categoria de “marcados como concluído”.

- 
- Na tela de cadastro de tarefas o (testador) deverá analisar se a os campos então sendo preenchidos corretamente e ao clicar em cadastrar tarefa irá aparecer uma mensagem de tarefa cadastrada com sucesso.

- 
- Na tela de configurações de notificações vai ao encontro direto da RF2 onde o testador irá fazer uma breve configuração de notificação e avaliar se as opções estão sendo devidamente marcadas e se aparece as tarefas previamente cadastradas e ao clicar em confirmar dar uma mensagem de sucesso.

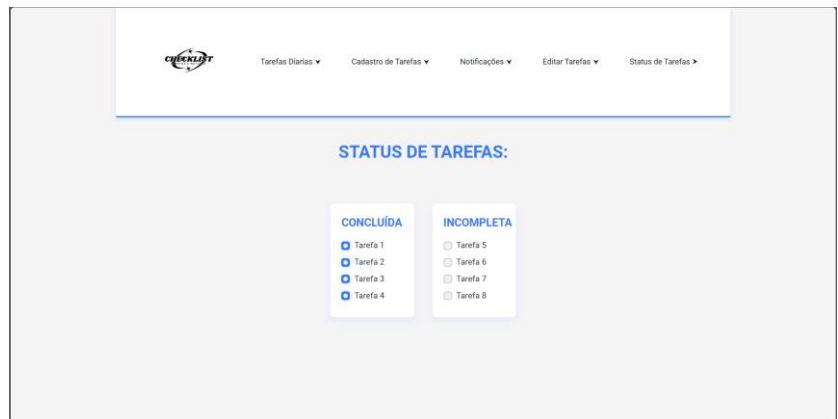


- 
- Na tela de edição vai direto ao encontro ao RF1.2 onde o testador vai verificar se as tarefas que estão aparecendo são as mesmas que foram previamente cadastradas e ainda estão pendentes e que ao selecionar a tarefa e clicar em editar a tarefa deve aparecer de uma forma onde o testador pode modificar a(s) tarefas e posteriormente ao termino da edição ver se a tarefa foi editada com sucesso.



- 
- Em Status de tarefa vai direto ao RF4 onde testador devera observar quais tarefas foram concluídas e quais ainda estão em andamento e se o relatório está sendo produzido pelo administrador.





### 1.2.2 Confiabilidade:

- Esse item se relaciona com a o RFN1 onde prevê a instabilidade do sistema assim verificando até mesmo se o site está devidamente codificado, e o testador deve garantir que tudo está em ordem de acordo com o que estava previsto nos requisitos.
- O testador deverá observar se o sistema está bem responsivo e agradável ao ser utilizado dentro do possível ao usuário final.
- Observar o tempo de resposta do site vai direto ao encontro aos requisitos que serão necessários para que o sistema não seja muito lento, o que atrapalha tanto o testador como os futuros utilizadores do mesmo.
- No sistema o testador irá marcar as tarefas e verificar se está sendo marcado como concluído, e verificar quanto o sistema demora para resposta e posteriormente informar para o desenvolvedor sobre o teste informado para garantir que o requisito seja devidamente concluído.

- Testar o RNF2 é fundamental para que em uma integração futura o sistema deve garantir a sua maturidade com relação ao projeto que está sendo testado, e a segurança é uma parte fundamental do sistema pois garantir segurança ao usuário é garantir a melhor expectativa para o mesmo.
- Para além disso o RNF4 é outra parte importante onde o testador vai observar justamente o código combinando assim o teste de caixa-branca, garantir que o sistema está tratando os erros corretamente com logs e suporte é essencial para o seu funcionamento.

#### 1.2.3 Eficiência:

- O sistema será eficiente ao garantir três pilares essenciais: desempenho, segurança e acessibilidade. Em termos de rapidez, a plataforma responderá aos comandos em até dois segundos, proporcionando uma experiência fluida mesmo para um grande número de usuários. Além disso, a organização de tarefas será intuitiva, permitindo facilmente o cadastro, edição e visualização das atividades diárias.
- A segurança é outro fator fundamental, com autenticação robusta e proteção avançada de dados. O sistema terá alta disponibilidade, operando com 99,9% de tempo ativo, e contará com recuperação rápida em caso de falhas, evitando perda

de informações e garantindo confiabilidade no uso contínuo.

- Por fim, a acessibilidade será um diferencial, oferecendo uma interface clara e compatível com diferentes dispositivos, incluindo um modo offline. A navegação será simples e adaptável, permitindo que qualquer usuário organize suas tarefas de forma eficiente, com suporte a notificações e organização inteligente por prioridade.

#### 1.2.4 Manutenibilidade:

##### 1.2.4.1 Tutoriais do sistema:

##### 1.2.4.1.1 Recomendações:

- O sistema tem que ter um breve tutorial de utilização do sistema onde será feito uma primeira apresentação de como funciona o sistema, para o usuário ficar mais apto a poder explorar suas funcionalidades.
- Para ser mais didático recomenda-se que o tutorial seja apresentado ao realizar o login para não haver confusões em relações a como o sistema tem que ser utilizado e ou observado.

## 2. Planejamento:

### 2.1 Tipos de teste:

#### 2.1.1 Teste Estrutural (Caixa -Branca):

- Para testar a estrutura interna do sistema, um teste estrutural (Caixa-Branca) pode ser aplicado ao processo de cadastro de tarefas. Esse teste verificará se a lógica do código armazena corretamente as informações fornecidas pelo usuário e se valida adequadamente os campos obrigatórios. Ao preencher o formulário com dados como nome da tarefa, descrição, categoria, data e prioridade, o sistema deve processar a entrada, garantir que não haja erros no banco de dados e retornar uma resposta confirmando o sucesso do cadastro. Caso algum campo obrigatório esteja ausente, a aplicação deve exibir uma mensagem de erro adequada.

#### 2.1.2 Teste Funcional (Caixa-Preta):

- Um teste funcional (Caixa-Preta) pode ser aplicado ao sistema para verificar se a funcionalidade de envio de alertas e notificações está operando corretamente. O teste avaliará se, ao configurar um lembrete para uma tarefa, o sistema envia a notificação no horário definido pelo usuário. O teste será conduzido inserindo diferentes horários e verificando se a notificação é recebida no tempo correto, seja por e-mail ou pelo site. Além disso, será testado se o usuário consegue modificar ou desativar os alertas sem erros. O objetivo é garantir que a funcionalidade atenda aos requisitos

esperados sem considerar a implementação interna do código.

#### 2.1.3 Teste Unitário:

- O teste unitário será aplicado em funções específicas do sistema, como a criação de tarefas. Ele verificará se, ao fornecer dados válidos, a função salva corretamente as informações e retorna um status de sucesso. Também testará cenários de erro, como a tentativa de cadastrar uma tarefa sem nome ou data, garantindo que o sistema exiba mensagens adequadas.

#### 2.1.4 Teste Individual:

- Será realizado em módulos isolados antes da integração, como o sistema de notificações. O teste verificará se os lembretes são configurados corretamente e enviados no horário definido, avaliando se as notificações aparecem na interface do sistema ou são entregues via e-mail sem falhas.

#### 2.1.5 Aceitação:

- Este teste validará se o sistema atende às expectativas dos usuários finais, garantindo que seja intuitivo, funcional e eficiente na organização de rotinas. Usuários testarão recursos como cadastro e edição de tarefas, priorização, alertas e acessibilidade, verificando se todas as funcionalidades operam conforme os requisitos especificados.

