**1. Leitura do Cenário (30 minutos): - Leia atentamente o cenário do projeto, compreendendo as necessidades e expectativas do cliente em relação ao sistema de gerenciamento de tarefas. - Identifique os principais stakeholders e suas respectivas responsabilidades.**

O que cliente espera é que o sistema ajude o trabalho das equipes a ser mais eficiente e ter uma melhor organização das gestões de tarefas, dando a elas uma melhor comunicação entre si.

**Cliente/contratante:** aquele que solicita o desenvolvimento do sistema e é responsável por ditar e fornecer as informações principais que o sistema deve ter.

**Usuários/equipe de Gestores:** São aqueles que vão usar o sistema no dia a dia para gerenciar suas tarefas.

**Desenvolvedores:** São aqueles que criam funcionalidades do sistema de acordo com os requisitos, eles também testam e corrigem bugs.

**2-Identificação de Requisitos Funcionais (45 minutos): - Com base no cenário, identifique e descreva pelo menos 10 requisitos funcionais do sistema. - Para cada requisito funcional, defina claramente o comportamento esperado do sistema. - Exemplos de requisitos funcionais incluem a criação de tarefas, atribuição de tarefas a membros da equipe, e envio de notificações automáticas.**

**Criação de tarefas:** O usuário deve preencher um formulário para criar uma tarefa, detalhando a tarefa podendo colocar descrição, prazo e o nível de prioridade e responsáveis que será salva e aparecerá na lista de tarefas.

**Atribuir Tarefas:** O usuário deve conseguir acessar a tarefa, alterar o que precisar e salvar as mudanças. podendo atribuir novos responsáveis.

**Atualização do progresso:** O usuário deve conseguir mudar o status de uma tarefa para concluída, e a lista de tarefas vai ser atualizada automaticamente.

**Gerar relatórios:** O usuário deve receber relatórios detalhados sobre o andamento das tarefas e sobre onde é preciso melhorar

**Pesquisa de tarefas:** O usuário deve conseguir achar tarefas na lista de forma rápida usando uma barra de pesquisa ou filtros, como data de vencimento ou prioridade.

**Importar e exportar dados:** O usuário deve poder importar dados para análise e exportar relatórios ou conjuntos de dados em outros formatos.

**Comunicação integrada:** O sistema deve mandar atualizações e notificações pelas ferramentas de comunicação que que a empresa disponibilizara sobre as tarefas podendo fazer menções e comentários em tempo real, aumentado a colaboração e a troca de informação dentro da equipe.

**Integração em plataformas**: O sistema será integrado em plataformas de comunicação já utilizadas pelos membros da equipe tais como o Slack, Microsoft Teams, ou e-mail, sendo assim sincronizado e tornando mais accessível a outros canais de comunicação

**Criação de subtarefas:** O usuário deve conseguir adicionar, editar e ver subtarefas ligadas a uma tarefa principal, e marcar cada subtarefa como concluída.

**Notificações de alertas:**  O usuário deve conseguir definir alertas para receber notificações em momentos específicos ou quando houver atualizações nas tarefas.

**Comentários:**  O sistema deve mandar atualizações e notificações pelas ferramentas de comunicação que já estão integradas.

**Monitoramento e relatório de tarefas:** O sistema deve monitorar o status atual das tarefas atribuindo marcados para indicar se alguma alteração foi realizada e o status da tarefa está concluída, pendente ou em andamento. Além disso o sistema deve prover periodicamente relatórios sobre a produtividade e histórico de alteração de cada tarefas.

**3. Identificação de Requisitos Não Funcionais (45 minutos): - Identifique e descreva pelo menos 5 requisitos não funcionais do sistema. - Exemplos de requisitos não funcionais incluem desempenho do sistema, segurança, usabilidade, e compatibilidade com diferentes dispositivos.**

**Desempenho:** O sistema deve processar e mostrar os resultados em menos de 2 segundos, mesmo com uma grande carga de dados de usuários.

**Segurança:** Dados pessoais e informações de tarefas devem ser criptografados, e o sistema precisa pedir uma autenticação forte, como a de múltiplos fatores além de e controle de acesso baseado em funções (RBAC) para que os usuários tenham acesso as informações das tarefas.

**Escalabilidade**: O sistema deve funcionar de forma eficiente e estável, mesmo com o aumento do número de usuários e dados. Incluindo a capacidade de suporte a centenas de usuários aos mesmos tempos, podendo adicionar de novas funções sem precisa de atualizações para isso.

**Backup e Recuperação:** O sistema deve fazer backups automaticamente e ter boas opções para restaurar dados quando necessário.

**facilidade de manutenção:** A manutenção e atualização do sistema devem ser simples, permitindo correções e melhorias rápidas.

**Sistema intuitivo/Multiplataformas:** o sistema deve ser intuitivo e consistente trazendo facilidade de aprendizados a novos usuários. o sistema também deve poder ser acessados de diferentes dispositivos, mantendo o desempenho em todos.

**4. Análise Crítica e Melhorias (30 minutos): - Analise criticamente os requisitos levantados. - Sugira melhorias no processo de levantamento de requisitos ou em algum requisito específico, justificando sua resposta.**

**Prioridade em atribuir as tarefas:** Poder definir o nível de prioridade da tarefa.

**Especificação na criação de tarefas:** especificar e detalhar os campos a serem preenchidos na tarefa.

**Agendar relatórios:** poder agendar e personalizar relatórios de forma periódica

**Implantação de filtros em Pesquisa de tarefas:** Além de usar uma palavra-chave o uso de filtros facilita as pesquisas dando sugestões.

**5. Documentação Final (30 minutos): - Organize todos os requisitos levantados em um documento estruturado. - Certifique-se de que o documento esteja claro e de fácil entendimento para todos os stakeholders envolvidos.**

**1. Descrição geral do sistema de Gerenciamento de Tarefas**

O sistema de gerenciamento de tarefas é uma plataforma fácil e intuitiva de usar, feita para ajudar pessoas e equipes a organizarem, acompanhar e completar tarefas de forma eficiente. Ele permite criar, alterar, designar tarefas, acompanhar o progresso e comunicar-se bem com outros usuários.

**2. Identificação dos requisitos**

**Funcionais:**

**Criação de Tarefas:** O usuário adiciona uma nova tarefa dando a ela uma discrição, prazo e designação. A tarefa será salva e aparecerá na lista de tarefas.

**Atribuir Tarefas:** O usuário pode abrir uma tarefa e mudar suas informações. Depois de fazer as mudanças além de mudar os responsáveis por ela., o usuário pode salvar e ver as atualizações na lista de tarefas.

**Atualização do progresso:** O usuário pode marcar uma tarefa como concluída. A lista de tarefas será atualizada para mostrar que a tarefa está concluída.

**Gerar relatórios:** O sistema mostra relatórios detalhados sobre o progresso das tarefas. Esses relatórios ajudam a ver onde é preciso melhorar.

**Pesquisa de tarefas:** O usuário pode encontrar tarefas rapidamente na lista. É possível pesquisar usando uma barra de busca ou filtros como data de vencimento e prioridade.

**Importar e exportar dados:** O usuário pode importar dados para análise e exportar relatórios ou conjuntos de dados. É possível exportar os dados em vários formatos diferentes.

**Comunicação integrada:** O usuário receberaatualizações e notificações pelas ferramentas de comunicação que que a empresa disponibilizara sobre as tarefas.

**Comunicação em múltiplos canais:** Além dos meios disponíveis pelo sistema deve ser integrado em outras plataformas de comunicação. Exemplo (Slack, Microsoft Teams, ou e-mail,)

**Criação de subtarefas:** O usuário pode adicionar, editar e ver subtarefas que estão relacionadas a uma tarefa principal. Cada subtarefa pode ser marcada como concluída separadamente

**Notificações de alertas:** O usuário pode configurar alertas para receber notificações em horários específicos ou quando as tarefas forem atualizadas. As notificações são enviadas de acordo com as preferências de alerta que o usuário escolheu.

**Comentários:** O sistema permite que os usuários adicionem e vejam comentários nas tarefas. Quando os comentários são atualizados, as ferramentas de comunicação integradas enviam notificações sobre essas mudanças.

**Não Funcionais:**

**Desempenho:** O sistema deve ser rápido e eficiente. O sistema deve processar e mostrar os resultados de forma ágil, mesmo com vários usuários acessando ao mesmo tempo.

**Segurança:** O usuário deve fazer uma verificação para de 2 fatores para poder ter acesso.

**Escalabilidade**: O sistema deve suportar o crescimento de usuários diários sem perca de desempenho

**Backup e Recuperação:** O sistema faz backups automaticamente. Há boas opções para restaurar os dados quando for preciso

**Facilidade de Manutenção:** Manter e atualizar o sistema deve ser fácil. Isso permite fazer correções rápidas e melhorar o sistema de forma contínua.

**3. Propriedades dos Requisitos e Estabelecimento de Prioridade**

**3.1. Essencial:** Requisitos sem os quais o sistema não funciona. São essenciais e precisam ser implementados sem falta. **Criação de Tarefas, atualização de Progresso e Segurança**

**3.2. Importante:** Requisitos que, se não forem incluídos, ainda deixam o sistema funcionando, mas não da melhor forma. Devem ser uma prioridade, mas o sistema pode ser lançado sem eles. **Geração de Relatórios, pesquisa de tarefas e notificações de alertas**

**3.3. Desejável:** Requisitos que não afetam as funções principais do sistema. Podem ser adicionados em versões futuras se não houver tempo para incluí-los agora. **Criação de subtarefas, comunicação integrada e importação e exportação de dados**

**6.0. projeto, como a escalabilidade, desempenho, segurança, e as preferências da equipe de desenvolvimento. No entanto o cliente solicitou um estudo de viabilidade sobre as seguintes linguagens de programação, dessa forma aponte as vantagens e desvantagens de cada tecnologia e quais as melhores escolhas para o projeto (e indique os motivos):**

1. JavaScript/TypeScript (com Node.js) - **Vantagens:**

velocidade, simplicidade, possui mais ferramentas e muitas bibliotecas

**Desvantagens:**

assincronismo, performance

2. Python – **Vantagens:**

Compatibilidade entre plataformas, Versatilidade, integrações a outros tipos de linguagem de programação, desenvolvimento rápido e eficiente

**Desvantagens:** suporte a dispositivos moveis limitado, desempenho em relações a outras tipos de linguagem.

3. Java – **Vantagens:**

Portabilidade parta outras plataformas, segurança e gerenciamento de memória.

**Desvantagens:**

Curva de aprendizado, código pode ser mais extenso

4. C# (com .NET) **Vantagens:**

Integração com o windows,performace, possui ferramentas robustas como Visual Studio e.NET Framework.

**Desvantagens:**

possui uma plataforma Limitada principalmente para o Windows, menos comunidades e bibliotecas

5. Go (Golang) **Vantagens:**

Simples e fácil de entender, possui uma performance rápida e eficiente.

**Desvantagens:**

Possui menos bibliotecas e frameworks, linguagem relativamente nova

E sobre os seguintes bancos de dados:

1. PostgreSQL **Vantagens:**

Robusto e possui escalabilidade

**Desvantagens:**

Complexo para iniciantes

2. MySQL/MariaDB **Vantagens:**

Rápido e é amplamente usado

**Desvantagens:**

Possui menos recursos que alguns outros como PostgreSQL

3. MongoDB **Vantagens:**

Possui escalabilidade, consegue armazenar dados e uma flexibilidade maior

**Desvantagens:**

Menos consistente e menos suporte para transações

4. SQLite **Vantagens:**

Possui um banco de dados embutido não necessita de um servidor, e mais leve suporta transações.

**Desvantagens:**

Não e muito adequado para cargas de trabalho com muitos acessos concorrentes, não é ideal para sistemas de grande escalabilidade.

5. Microsoft SQL Server **Vantagens:**

Possui recursos mais avançados, tem integração com recursos e ferramentas da Microsoft. Consegue lidar com grande volume de dados.

**Desvantagens:**

É mais caro, possui uma plataforma limitada**.**

**Escolha:**

A melhor escolha de linguagem seria javaScript/TypeScript (com Node.js), mesmo ele podendo ser um pouco mais difícil de gerenciar ele possui, muitos dos requisitos que o cliente pede, como escalabilidade muito boa para aplicações em web, segurança e desempenho dele seria melhor pois muitas equipes ja estão familiarizadas.

Já banco de dados eu escolheria MySQL/MariaDB , pois ele e mais rápido e possui suporte amplo, apesar de ter menos recursos que alguns outros ele também se encaixa nos requisitos de escalabilidade ainda mais com o MariaDB , segurança bem adequada, e um desempenho bom.