

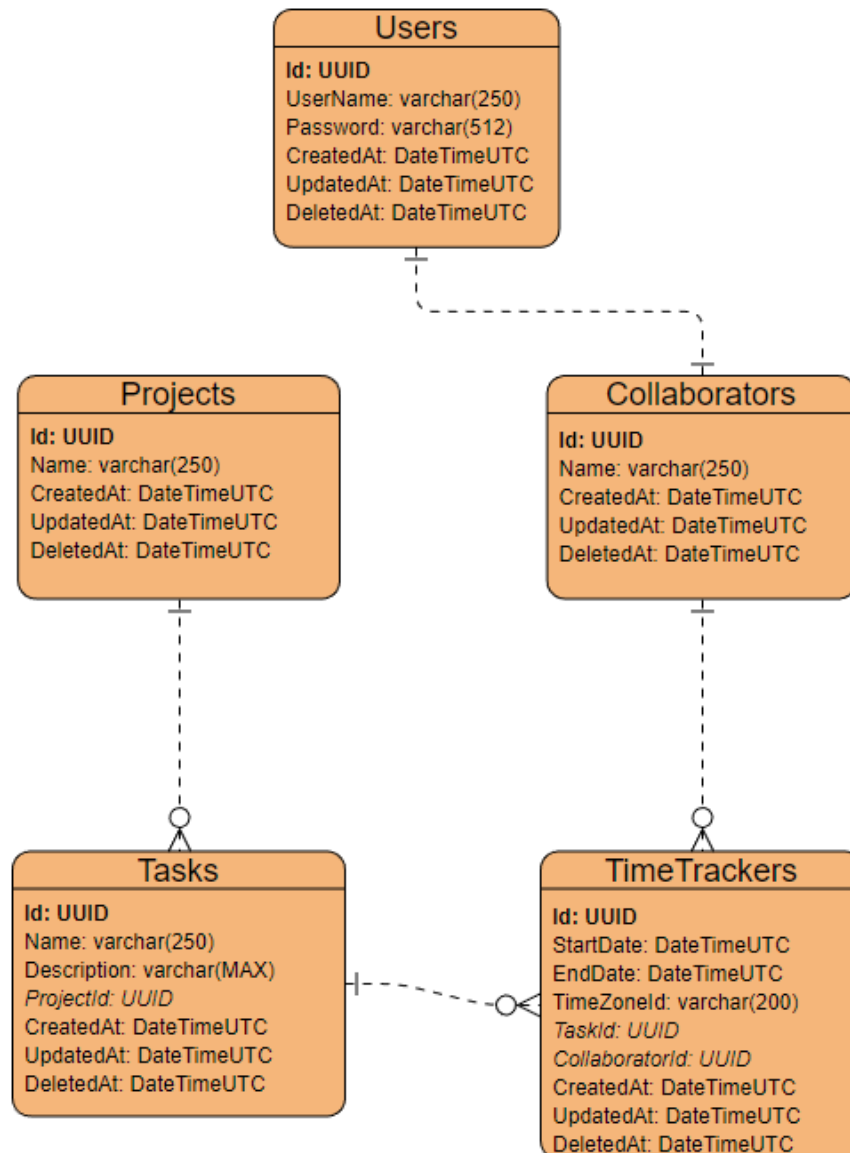


Desafio 1 - Criar sistema para controle de tarefas com apontamento de horas

Itens esperados da aplicação

- Cadastro de Tasks (CRUD)
- Cadastro de Projetos (CRUD)
- Interface para listagem de tarefas;
 - Visualização do tempo gasto no dia corrente;
 - Visualização do tempo gasto no mês corrente;
 - Associação colaborador a uma task;
- Apontamento de tempo gasto nas tasks;
 - DataHora inicial e DataHora final;

Diagrama do banco de dados



Regras de negócio

- ☐ Tabelas com os tipos relacionados ao banco de dados utilizado.;
- ☐ **Username** dever ser único;
- ☐ **Password** deve ser criptografado;
- ☐ A tabela de “Collaborators” deve ter um vínculo forte com usuário;
- ☐ O tipo de banco de dados pode ser SQL ou No-SQL;
- ☐ Não é possível incluir um time tracker que colida o intervalo de tempo;

Ex: se existir [07:00:00|10:00:00], não é possível incluir um intervalo que comece ou termine entre 07:00:00 às 10:00:00 inclusive.

- ☐ Não é obrigatório um colaborador para uma task;
- ☐ É obrigatório que uma task tenha associação a um projeto;
- ☐ Uma task pode ter vários `timetrackers`
- ☐ O tempo de início deve ser menor ou igual ao término;
- ☐ O time zone local sempre deve ser enviado na requisição de inclusão do tempo;
- ☐ Deve-se ter validação dos campos no frontend e backend;
- ☐ O backend deve funcionar como uma `api rest` ;
- ☐ Utilizar padrões de projetos e arquiteturas;
- ☐ A autenticação deve ser feita usando padrão de token JWT;
- ☐ Aplicação deve ser responsiva;
- ☐ Erros devem ser exibidos de maneira clara ao usuário;
- ☐ O total de horas em tasks dentro de um dia não deve ultrapassar 24hrs;
- ☐ Não é necessário tela para cadastro de usuários e colaboradores. Isso deve ser feito direto no banco de dados;

User Stories

- ☐ Usuário pode ver lista de projetos;
- ☐ Usuário pode adicionar/editar e excluir projetos;
- ☐ Usuário pode ver lista de tarefas;
- ☐ Usuário pode adicionar/editar e excluir tarefas;
- ☐ Usuário pode associar tarefa a um projeto;
- ☐ Usuário pode associar tarefa a um colaborador;
- ☐ Usuário deve ver de forma destacada o tempo gasto no dia corrente no formato `Horas:Minutos` ;
- ☐ Usuário deve ver de forma destacada o tempo total gasto no mês corrente no formato `Horas:Minutos` ;
- ☐ Usuário deve ver uma lista de todas as tasks;

- ☐ Usuário pode filtrar a lista por projeto;
- ☐ Usuário pode filtrar a lista por colaborador;
- ☐ Usuário pode selecionar uma tarefa e iniciar/parar o `timetracker` de uma tarefa através de um botão;

Funcionalidades bônus

- ☐ Tela padrão de login com validação de autenticação, deve ser apresentada antes da tela de tarefas;
- ☐ Criar documentação da API, swagger ou similares;
- ☐ Utilização de ORM;
- ☐ Usuário pode ver o tempo gasto dia-a-dia(do mês corrente) no formato `Horas:Minutos` (Tela de relatório);
- ☐ Usuário pode filtrar por projeto ou usuário;

Informações importantes

1. A aplicação deve ser disponibilizada em um repositório GIT de maneira pública.
2. Importante utilizar o mínimo possível de Scaffolding (geração automatizada de código), pois o objetivo do teste é analisar o seu conhecimento e organização para codificação.
3. Apenas a **nível de conhecimento**, aqui trabalhamos com C#, ASP.NET, *ASP.NET CORE*, NodeJS, JavaScript, Bootstrap, *VueJs*, EntityFramework e AutoMapper/StructureMap, SQL Server, Postgres, MongoDB, Redis, RabbitMQ.
4. Questões tecnológicas/frameworks, ficam à sua escolha (sugerimos você utilizar o que mais conhece).