

ToDo – Sprint 2

1 – Mapea requisitos funcionais e não funcionais.

Requisitos funcionais no projeto atual da sprint 2 são:

- Criar item;
- Deletar item;
- Editar item;
- Obter todos itens.

Os requisitos funcionais são os básicos de um crud;

Com isso para se criar um item usamos o método `CreateItemAsync` que precisa receber um request de nome `CreateItemRequestDto` que precisa passar apenas o campo `description` e a partir daí é passado da `application` para o `repository`;

Já no caso de deletar são duas etapas a primeira é obter o item quando clicado em deletar, ele é obtido a partir do ID, e em um método `GetItemAsync` que precisa ser passado o ID em um `IdRequestDto` que na `repository` roda um método com um `select` com `where` usando o ID.

Por fim é passado os dados a tela de deletar, caso clique no botão deletar é acionado o método `DeleteItemAsync`, que precisa ser passado o campo ID em um `IdRequestDto`, por fim vai ao `repository` onde roda um `delete` com o ID.

Já no editar são duas etapas também a primeira para obter os dados do item, depois é criado a tela de editar o item.

Na tela de editar é passado os dados para o método da `service` `EditItemAsync` com o `EditItemRequestDto` que passa o valor do id e do `done` que depois vai para o `repository` rodar um `update`.

Por fim a função Obter todos os itens não possui requisito mas é rodado no método da `service` da `application` `GetItemsAsync`, nele é rodado na `repository` um `select` sem `where`.

Requisitos não funcionais no projeto atual são :

- O sistema deve se comunicar com o banco de dados SQL Server
- Linguagem do sistema é C#
- O sistema deve ter permissão de cadastro no banco de dados
- Deverá ter um banco previamente criado com a estrutura
- Docker com o banco de dados

O sistema precisa de um banco anteriormente criado e populado para assim poder gerar os dados e obter ao menos a estrutura.

2 – As entidades criadas foram:

Metodos da interface devem ser:

GetAllAsync => uma método que retorna um IEnumerable do itemResponseDto, não precisa de ser passado nenhum parâmetro.

AddAsync => um método criado para inserir itens ao banco é preciso passar o parâmetro Item.

EditAsync => um método para editar o item é necessário passar os parâmetros que são o Id e o Done.

DeleteAsync => um método para deletar os itens é necessário passar o Id.

GetByIdAsync => um método para retornar apenas um item é necessário passar o Id.

EditItemModel para obter os dados de edicao;

EditItemRequestDto para passar os dados ID e Done para editar o campo informado no item especifico;

ItemResponseDto para retornar todos campos do cadastro;

IdRequestDto deve receber o Id para retornar apenas um item ou para deletar apenas o item com o Id informado;

CreateRequestDto deve receber o campo Description para criar um novo item.