

O planejamento da arquitetura visa garantir que as implementações de banco de dados, frameworks ou outros componentes que possam acoplar uma classe estejam sempre nas extremidades, assim caso seja necessário trocar, por exemplo, o banco de dados, basta alterar somente a classe que faz implementação do Postgres (AddCategoryPostgresRepo) fazendo com que toda a estrutura do projeto permaneça inalterada.

Dentro das camadas temos uma interface a ser seguida por todas as classes que faz com que a dependência da camada exterior seja invertida. Por exemplo, para ter validação de Email basta o AddAccountController usar os métodos da interface EmailValidator. Aquela classe que implementar o EmailValidator poderá atender ao AddAccountController

Caso seja necessário outras implementações que dependam de frameworks ou banco de dados devemos fazer a inversão de dependência pois a data layer não pode conhecer a infra, sendo que a data usa interfaces que devem ser implementadas pela infra para atender às solicitações.

A camada de infra faz implementações de interfaces voltadas para frameworks ou banco de dados que não sejam adicionados na camada utils.

Contém implementações úteis que não necessariamente tenham que estar em outras camadas.

Domain layer não contém implementação de classes, mas somente interfaces que definem as regras de negócio da aplicação a serem seguidas pelas camadas exteriores

A interface garante que a classe que a implementar respeitará o método de validação e assim será possível a sua utilização em qualquer controller que use o EmailValidator

Para inverter a dependência (o DbAddAccount não apontar diretamente para AccountPostgresRepo) criamos uma interface que deverá ser implementada por alguma classe da infra para atender ao pedido da DbAddAccount, que por sua vez usará a interface AddAccountRepository.

Na data layer fazemos as implementações das regras de negócio. Como a nossa implementação de AddAccount (main layer) é voltada para banco de dados utilizaremos o prefixo Db na classe que implementar a interface AddAccount da main layer

O AddAccountController precisa de alguma classe que implemente a interface definida na camada de domain para permitir que a categoria seja salva em algum local, seja BD, cache ou etc. As classes na data layer são responsáveis por isso

Qualquer controller que implemente a interface Controller pode ser usado no ExpressRouterAdapter

A main layer faz a composição de todos os objetos e por isso é a única que conhece todas as camadas, sendo a que será acoplada em detrimento das demais.

O ExpressRouteAdapter que fica acoplado ao express

----- Interface

