

Nesta aula, implementamos o use case abaixo:

UC1 - Signup

Ator: Passageiro, Motorista

- * deve verificar se o email já existe e lançar um erro caso já exista
- * deve gerar o account_id (uuid)
- * deve validar o nome, email e cpf
- * deve gerar o código de verificação da conta
- * deve enviar um email de verificação da conta com um link contendo o código (por enquanto usando um console.log)

Read Model - GetAccount

Input: account_id

Output: todas as informações da conta

Para a próxima aula, implemente os use cases abaixo utilizando TDD:

UC2 - Solicitar corrida

Ator: Passageiro

Input: passenger_id (account_id), from (lat, long), to (lat, long)

Output: ride_id

Regras:

- * deve verificar se o account_id tem is_passenger true
- * deve verificar se já não existe uma corrida do passageiro em status diferente de "completed", se existir lançar um erro
- * deve gerar o ride_id (uuid)
- * deve definir o status como "requested"
- * deve definir date com a data atual

UC3 - Aceitar corrida

Ator: Motorista

Input: driver_id (account_id)

Output: void

Regras:

- * deve verificar se o account_id tem is_driver true
- * deve verificar se o status da corrida é "requested", se não for, lançar um erro
- * deve verificar se o motorista já tem outra corrida com status "accepted" ou "in_progress", se tiver lançar um erro
- * deve associar o driver_id na corrida
- * deve mudar o status para "accepted"

Read Model - GetRide

Input: ride_id

Output: todas as informações da ride

Considere o modelo de dados:

```
create table cccat13.ride (  
    ride_id uuid,  
    passenger_id uuid,  
    driver_id uuid,  
    status text,  
    fare numeric,  
    distance numeric,  
    from_lat numeric,  
    from_long numeric,  
    to_lat numeric,  
    to_long numeric,  
    date timestamp  
);
```

Considere utilizar um design de 3 camadas com uma API (tanto faz se é REST, GraphQL, gRPC) e o banco de dados (tanto faz se é SQL ou NoSQL):

- * **API** (com os 3 use cases: signup, request_ride e accept_ride e também os read models getAccount e getRide)

- * **Regras de negócio** (não utilizar Domain Model por enquanto)

- * **Acesso à dados** (camada simples de acesso ao banco de dados com as queries que fazem create, retrieve, update e delete)