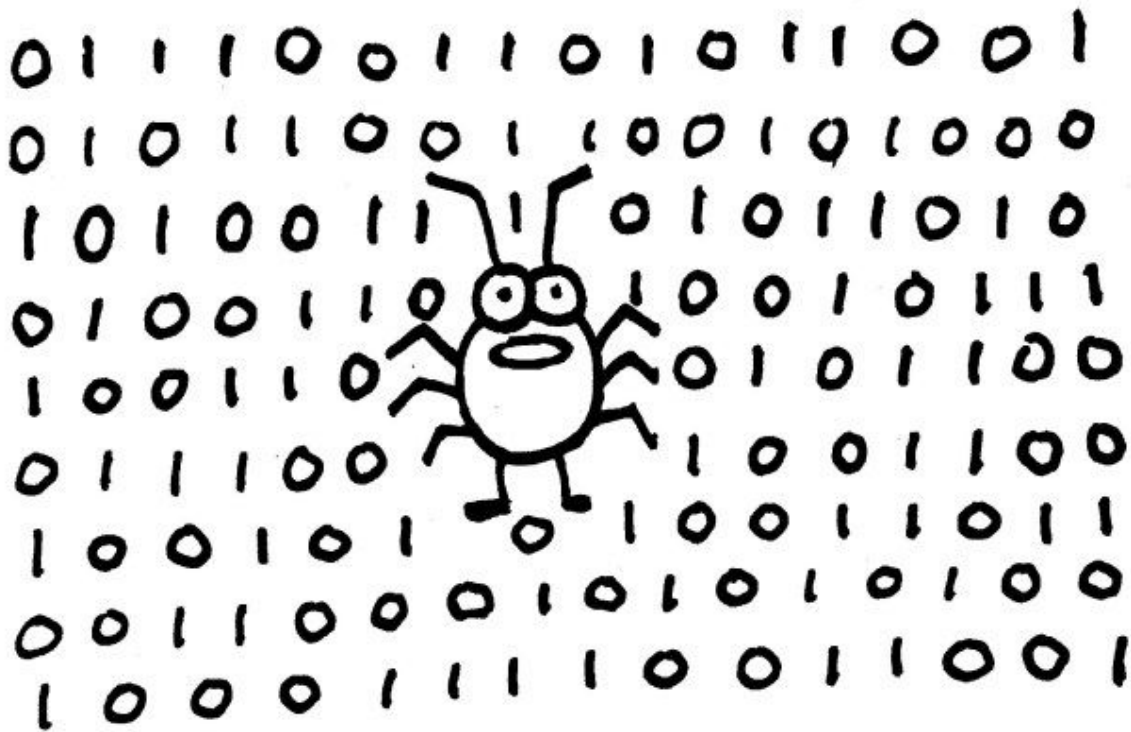




Ciências  
ULisboa

## 2º Projecto

### Verificação e Validação de Software



Carla Mata fc45109

Gonçalo Cardoso fc46784

Pedro Gaspar fc46411

## Erros Corrigidos

- 1) Era possível adicionar uma sale a um customer inexistente. Para corrigir este erro, adicionamos um CustomerHelper ao AddSalePageController e populamos este sale helper com o getCustomerByVat do CustomerService. Desta forma quando é introduzido um vat inexistente é devolvida uma mensagem de erro explícita.
- 2) Era possível adicionar addresses duplicados ao mesmo cliente. Para corrigir este erro, foi adicionado um método auxiliar no CustomerService que é utilizado antes do endereço ser adicionado para verificar se o endereço a adicionar já existe para o cliente, devolvendo um erro correspondente para o caso.
- 3) Ao apagar um cliente as sales, deliveries e addresses associadas a este não eram apagadas. Foi necessário a criação de um método para remover as sales e as consequentes deliveries que se encontra na classe SaleService. Para isto foi necessário também a criação de métodos para a remoção nos RowDataGateways da Sale, Delivery e Address.

## Testes Extra do Exercício 2

Para os testes extra adicionamos os seguintes:

- Ao adicionar uma sale o número total de sales aumenta por um.
- Quando uma sale é apagada a sua delivery também é, se esta existir.

## Exercício 3

O mockito é normalmente usado de fazer uma classe mock para vários tipos de classes tais como:

- Ligações a bases de dados
- Serviços Web
- Classes Lentas
- Classes com efeitos secundários
- Classes com comportamento não determinístico

Sendo que o tipo de classe sugerida para fazer um mock é um serviço web, esta seria uma proposta possível de concretizar, mas, dado que os serviços desenvolvidos são enumerados, não é possível, pois essa é uma das limitações do mockito. Para que seja possível desenvolver um mock dos services, bastaria fazer um refactoring de forma a torná-los não enumerados