

ADVENTURE_4

Setup

Para realizar a quarta parte do projeto deve começar por **resolver na totalidade o exercício prático da aula-spring**.

Para ser efetuado por um único elemento do grupo

Obtenha a nova versão do código disponibilizada em <https://github.com/tecnico-softeng/reference.git>. Instale essa versão no ramo master e crie uma cópia para o ramo develop onde o grupo irá realizar as suas atividades de desenvolvimento. Para isso faça

```
$ git fetch reference
$ git checkout master
$ git merge reference/master
$ git push origin master
```

de seguida deve assegurar-se que o código da segunda parte foi entregue através do ramo third-deliver, mudando para esse ramo

```
$ git checkout third-deliver
```

pode então apagar o ramo develop, mudar para o ramo master, criar um novo ramo develop a partir do master e colocar a nova versão do develop em origin

```
$ git branch -d develop
$ git checkout master
$ git checkout -b develop
$ git push origin develop --force
```

deve agora correr os testes existentes

```
$ mvn clean test
```

É também necessário configurar correctamente o **Travis CI** e o **Codecov** (esta tarefa deve ser uma issue). Para tal, deve:

- Obter o token Repository Upload Token disponível no site do Codecov - https://codecov.io/gh/tecnico-softeng/es18LL_NN-project/settings (onde LL é al ou tg e NN é o número do projecto) - e actualizar a variável **CODECOV_TOKEN** no ficheiro **.travis-ci.yml** com esse token.
- Actualizar os badges presentes no ficheiro README.md para reflectirem os dados deste projecto.
 - Para obter o novo badge do Travis CI deve abrir o projecto - https://travis-ci.com/tecnico-softeng/es18LL_NN-project - e clicar no badge. De seguida deve escolher a opção Markdown e substituir o badge que se encontra no README.md com esse valor.
 - Para obter o novo badge do Codecov deve abrir o projecto - https://codecov.io/gh/tecnico-softeng/es18LL_NN-project/settings/badge - e substituir o badge que se encontra no README.md com o valor em Markdown.

Para ser efetuados pelos restantes elementos do grupo, após as alterações anteriores

Após a nova versão do código estar no GitHub do projeto deve actualizar os ramos locais:

```
$ git fetch --all
$ git checkout master
$ git pull
$ git branch -d develop
$ git checkout -b develop origin/develop
```

deve agora correr os testes existentes

```
$ mvn clean test
```

e abrir o projeto no Eclipse.

Enunciado

A quarta parte do projeto corresponde a transformar a aplicação da terceira parte (terceiro *sprint*) em 6 aplicações web, aplicando o padrão arquitetural camadas e usando a tecnologia Spring Boot e Thymeleaf, que implementam o padrão modelo-vista-controlador. Os módulos broker, activity, hotel, e bank, já se encontram parcialmente implementados, em que é necessário adaptar a implementação existente às alterações funcionais efetuadas durante este ano no projeto. Note que pode haver bugs neste código que deverá identificar e resolver. Adicionalmente deve implementar na totalidade os módulos car e tax para serem aplicações web, em que o primeiro deve executar no porto 8084 e 8086.

As interfaces web a implementar são (cada sub-ítem corresponde a uma interface utilizador). Note que pode ser necessário efetuar outras alterações sempre que existam bugs:

- Módulo activity:
 - UI de criação de fornecedores de atividades passa a incluir o nif e iban dos fornecedores. Esta informação também deve estar na lista de fornecedores.
 - UI de criação de criação de oferta passa a incluir a o custo.
 - UI de reserva, que inclui a criação de reserva e a listagem de todas as reservas associadas a uma oferta.
- Módulo bank:
 - Necessário efetuar algumas adaptações, por exemplo, devido aos valores serem do tipo double.
- Módulo broker:
 - UI de criação de broker para passar a incluir os nifs e o iban. Esta informação deve surgir na lista de brokers.
 - UI de criação de clientes no contexto de um broker
 - UI de criação de aventuras deve passar a ocorrer no contexto de um cliente.
 - UI de criação de reservas bulk passa a ter nif e iban, essa informação deve surgir na lista
- Módulo car:
 - Implementar todas as UI para o módulo, criação e listagem dos seus elementos.
 - RentACar - criar e listar
 - Vehicle (Car, Motorcycle) - criar e listar para cada RentACar
 - Renting - criar e listar para cada Vehicle, fazer checkout
- Módulo hotel:
 - UI de criação de hotel para incluir o nif, iban e preços, devendo esta informação deve também surgir na lista de hotéis.
 - UI de reserva de quarto para incluir o nif e o iban.
- Módulo tax:
 - Implementar todas as UI para o módulo, criação e listagem dos seus elementos.
 - TaxPayer - criar e listar
 - ItemType - criar e listar
 - Invoice - criar e listar para cada TaxPayer

Sempre que durante a execução de uma operação for lançada uma excepção, deve ser apresentada uma mensagem de erro indicando apenas que não foi possível efetuar a operação.

Os grupos devem começar por reunir para se subdividirem em subgrupos e identificarem as tarefas, associá-las ao backlog, e definir dependências de precedência entre elas, de forma a calendarizá-las no período disponível. Cada grupo deve-se dividir em sub-grupos de 3 alunos, em que cada um deles, trabalha num módulo. **É obrigatório indicar no README.md quais os elementos que trabalham em cada um dos módulos, indicando o número de aluno, o nome e o username no GitHub.**

Para a entrega deverão fazer:

```
$ git checkout develop
$ git checkout -b fourth-deliver
$ git tag ADVENTURE_4
$ git push origin --tags fourth-deliver:fourth-deliver
```

Esta tag colocada deve ter uma data anterior à data limite de entrega, **dia 6 de Maio pelas 20:00**.

Durante os laboratórios, os alunos dos grupos serão avaliados com base no trabalho desenvolvido durante cada uma das semanas. Para isso é necessária a criação de um sprint no GitHub, usando a interface de Project, em que cada tarefa deve ser representada por um Note. Neste caso o projeto a criar deve-se chamar Sprint Fourth. Uma vez criado o projeto, escolhendo como template pré-definido "Kanban (automatic)", deve ser alterado o nome da coluna To Do para Backlog,. Nesta coluna serão colocadas todas as tarefas a realizar, as quais serão definidas durante a reunião do grupo. Cada tarefa é criada como uma Note, contendo a descrição da tarefa a realizar, e deve ser convertida num Issue, por forma a que os commits possam ser-lhe associados. Cada aluno que vá realizar uma tarefa move o respetivo note da coluna de Backlog para a coluna In progress, e atribui-se essa tarefa no Issue respetivo. Quando a tarefa é terminada, o commit deve ser associado ao respetivo Issue, passando este para o estado fechado. Para se associar um commit a um Issue deve colocar-se na mensagem de commit close #n, em que n é o número do Issue. Quando o Issue é fechado o respetivo Note é automaticamente passado do Issue In progress para a coluna Done. **Cada commit deve estar associado a uma única tarefa.**

[Página Inicial](#)



[Grupos](#)

[Avaliação](#)

[Bibliografia](#)

[Horário](#)

[Métodos de Avaliação](#)

[Objectivos](#)

[Planeamento](#)

[Programa](#)

[Turnos](#)

[Anúncios](#)



[Sumários](#)



[Notas](#)

[Resultado dos QUC](#)

[Teóricas](#)

[Laboratórios](#)

[Horário de Dúvidas](#)

[Documentação](#)

[Projeto](#)

[ADVENTURE_1](#)

[ADVENTURE_2](#)

[ADVENTURE_3](#)

[ADVENTURE_4](#)

[FAQ](#)

[ADVENTURE_5](#)

[Extra-Mile
\(Prémio
Novabase\)](#)

[Época Especial](#)

[Notas intercalares](#)