

# LPROG 2021

---

17 JUNHO

---

**ISEP**

**Da autoria de: João Beires 1190718**

**José Soares 1190782**

**Lourenço Melo 1190811**

**José Maia 1191419**



---

# LPROG – 2021 – SPRINT D

O nosso grupo é constituído por 4 alunos que não têm LAPR4. Assim, tivemos de adaptar o Sprint D de LPROG ao que nos é possível fazer, tendo em conta que não temos informações em relação a LAPR4 nem EAPLI. Assim, o que podemos fazer fica-nos mais limitado.

Temos um ficheiro Excel (“Catálogos.xlsx”) sobre todos os catálogos mencionados no enunciado. Ainda que alguns dos catálogos viessem de EAPLI e LAPR4, fizemos na mesma com que a nossa gramática.g4 interpretasse todo o tipo de dados de todos os catálogos.

Com isto, no ficheiro Excel é possível ver 4 páginas. A primeira página “Catálogo de Serviços” tem uma coluna “Script” que, ao carregar, abre os scripts em questão facilitando assim a interpretação da automatização, isto é, o script já criado para testar a gramática.

No moodle de LAPR4 foram dados alguns exemplos de funcionalidades:

- Pedido de ausência futura;
- Autorização para Aplicação de Desconto
- Comunicação de Alteração de Residência
- Requerer cotação para venda por grosso

**NOTA:** Todas estas funcionalidades estão 100% funcionais!

Para além das funcionalidade acima mencionadas temos muitas mais a funcionar a 100% para provar que temos o conhecimento deste sprint. Exemplos:

- Ler o rapel do cliente;
- Ler o preço do produto;
- Reportar uma anomalia de comunicação;
- Alterar NIB de um cliente.

Assim, todos os métodos estão criados para todas as funcionalidades possíveis (conferir na gramática .g4)

---

Na gramática, caso falhe um parâmetro, o utilizador é notificado na consola. Deste modo, garante sempre que as funções só façam o que é pedido quando os dados forem corretamente introduzidos. Também temos a confirmação das datas. A data tem de ser uma data válida.

Posteriormente, procedemos à realização do visitor e do listener. Estas duas classes dão “extend” à sua respetiva Base, por exemplo, o visitor dá “extend” à classe “SprintCSEMLAPR4BaseVisitor”. Deste modo a relação entre a gramática.g4, o main as classes visitorAntlr e listenerAntlr é possível.

Explicação sucinta dos métodos utilizados:

Para funcionalidades como:

- Ler/alterar informação do cliente:
  - Utilizamos um ficheiro criado na pasta “Ficheiros” quer seja txt ou xml para obter informações sobre os clientes. Assim, tanto o visitor como o listener são capazes de ler esses ficheiros, cujo nome é enviado por parâmetro, e obter acesso ao antigo, por exemplo, rapel do cliente. Posteriormente procedemos ao print da nova informação, enviada por parâmetro no script, na consola provando que conseguimos ler as informações e tratá-las utilizando os métodos de visitor e listener.
- Reportar anomalias:
  - O script utilizado fornece informações como por exemplo o email para onde vai ser enviada a anomalia. Assim, depois de termos todas as informações sobre as anomalias, passadas por parâmetro, conseguimos enviar um email (funcionalidade a funcionar 100% (confirmar no seu próprio email, basta alterar o email enviado por parâmetro no script)) e este pode conter imagens. Os exemplos que nós fornecemos já têm imagens associadas, imagens estas que estão na pasta “images”. No email pode conter informações recebidas por parâmetro também, confirmado assim que conseguimos tratar as informações através do Formulário.
- Cotações e descontos:
  - Os métodos “requerCotacaoVendaPGrosso” e “autAppDescFinanc” estão também 100% funcionais. O método requer cotação tem em conta todos as confirmações exigidas no moodle de lapr4. É apresentada, tanto em

---

visitor como em listener, as informações todas e resultados na consola após correr este script. Em relação à autorização da aplicação de descontos das finanças é enviado um email a pedir autorização para a aplicação e posteriormente aparece uma janela no ecrã do utilizador a simular que a autorização foi concedida. Esta janela tem informações interpretadas na gramática confirmado assim também que temos os conhecimentos necessários.

- Recursos humanos:
  - Estas três funcionalidades (marcaFerias, justificaFalta, informaAusenciaFutura) estão também 100% funcionais. Estas funcionalidades interpretam também todos os parâmetros e caso algum não esteja correto, tal como todas as funcionalidades, indica na consola a mensagem “NOT OK”. É enviado um email com as informações pretendidas justificando assim o nosso conhecimento necessário para este sprint.

É de salientar que todas as funcionalidades são confirmadas e, caso algum parâmetro não seja escrito corretamente, o sistema escreve uma mensagem na consola a indicar o mesmo! Quando algum parâmetro é introduzido de forma errada é apresentada, tal como pedido, uma mensagem “NOT OK”, dizendo assim que procedemos corretamente às confirmações necessárias para garantir uma gramática e formulário coerentes e conciso.

Em suma, pedimos aos professores que irão avaliar este trabalho que tenham em conta a pouca informação que tivemos acesso ao executar este Sprint D. Ainda assim, tentamos ao máximo evoluir o nosso projeto, aumentar as funcionalidades e confirmações e demonstrar que percebemos bem os conteúdos de ANTLR, Visitor e Listener.

**Última nota:** Ter em conta que para confirmar que os emails estão a funcionar é necessário alterar o remetente nos ficheiros .txt que irão ser utilizados na consola. Por exemplo, caso o professor quisesse confirmar se o método marcaFerias está efetivamente a enviar um email a confirmar a funcionalidade deve abrir o ficheiro marcaFerias.txt e onde se encontra o email [1190782@isep.ipp.pt](mailto:1190782@isep.ipp.pt) deve colocar o seu próprio email. Para esta funcionalidade funcionar corretamente foi criado um email com o nome lprog2021sprintd2de@gmail.com.