

# Template Validator - Documentação Técnica

## Objetivo do Trabalho

O objetivo deste trabalho foi construir um sistema que valida templates de propostas (com placeholders) utilizando **ANTLR** para garantir que todas as informações necessárias estão corretamente especificadas.

A ferramenta foi desenhada para:

- Validar a sintaxe de templates textuais.
  - Verificar a presença de campos obrigatórios (placeholders).
  - Gerar substituições para esses placeholders com base no conteúdo da proposta (`Content`).
  - Ser integrada no sistema existente da Shodrone (projeto Java).
- 

## Pressupostos e Decisões

- Os templates devem conter placeholders no formato `${field}`.
  - Todos os campos obrigatórios definidos no enum `RequiredFields` devem estar presentes pelo menos uma vez.
  - Usamos ANTLR para gerar o parser e o visitor.
  - A validação não requer análise semântica do conteúdo, apenas sintática e estrutural.
  - A gramática assume que os templates são essencialmente texto com campos interpolados.
- 

## Estrutura da Solução

ANTLR Grammar (template.g4)

```
grammar template;

template: (field | TEXT)* EOF;

field: FIELD;

FIELD: '${' FIELD_NAME '}';

TEXT: ~[$]+ | '$' ~[{}]+;

FIELD_NAME: [a-zA-Z_][a-zA-Z0-9_]*;

WS: [ \t\r\n]+ -> skip;
```

---

## Componentes Desenvolvidos

### TemplatePlugin

- Classe principal de validação.
- Usa ANTLR para parsear o conteúdo do template.
- Verifica sintaxe e presença de campos obrigatórios.
- Gera mapa de placeholders substituíveis (`buildPlaceholderMap()`).
- Usa métodos de formatação específicos para cada tipo de conteúdo.

### TemplateFieldVisitor

- Estende `templateBaseVisitor` gerado por ANTLR.
- Visita todos os campos `${...}` do template e coleta seus nomes.

### RequiredFields

- Enum com os grupos de campos obrigatórios:
  - `CUSTOMER`, `SHOW_DATE`, `SHOW_LOCATION`, `DURATION`, `FIGURES`, `DRONES`, `VIDEO`, `MANAGER`
- Permite várias formas alternativas por grupo (e.g., "cliente", "customer").

---

## Exemplo de Template Utilizado

Válido

Exmos. Senhores \${customerName},

Envio da sua proposta para o espetáculo de \${showDate}.

Local: \${showLocation}

Duração: \${duration}

Figuras apresentadas:  
\${figures}

Modelos de drones utilizados:  
\${drones}

Video preview do show: \${video}

Obrigado pela sua preferência,  
\${manager}

© 2025 Shodrone. Todos os direitos reservados.

**Inválido**

Exmos. Senhores \${customerName},

Envio da sua proposta para o espetáculo de \${showDate}.

Local: \${showLocation}

Duração: \${duration}

Video preview do show: \${video}

Obrigado pela sua preferência,  
\${manager}

© 2025 Shodrone. Todos os direitos reservados.

*Faltam os drones e as figuras...*

# Exemplo de Erros Detetados

Se faltar algum campo obrigatório:

```
Missing required field group: VIDEO  
Missing required field group: SHOW_DATE
```

Se houver erro de sintaxe:

```
Line 2:17 - mismatched input 'clientName' expecting '}'
```

---

## Ficheiros de Teste Incluídos

- `valid_template.txt` – Template completo e válido.
- `missing_fields.txt` – Template com campos obrigatórios em falta.
- `syntax_error.txt` – Template com erros de formatação nos placeholders.

---

## Conclusão

Este validador permite garantir que os templates de propostas utilizados no sistema possuem todos os dados necessários antes de serem enviados. A utilização de ANTLR permitiu definir uma gramática clara e extensível, suportando facilmente alterações futuras.

A estrutura modular e orientada ao domínio (com apoio de `Content`, `Location`, `Figure`, `DroneModel`) garante flexibilidade e integração com o restante sistema Java.