

ABAP Moderno: Processamento de Strings

Do Legado ao Funcional: Dominando
Expressões, Regex e Predicados

Objetivos de Aprendizagem



Substituir comandos obsoletos (TRANSLATE, SEARCH) por **Funções Embutidas**.



Aplicar **Expressões Regulares (Regex)** para validação e limpeza de dados complexos.



Utilizar **Funções Predicativas** (matches, contains) para otimizar fluxos lógicos.



Implementar higienização de dados utilizando **encadeamento de métodos**.

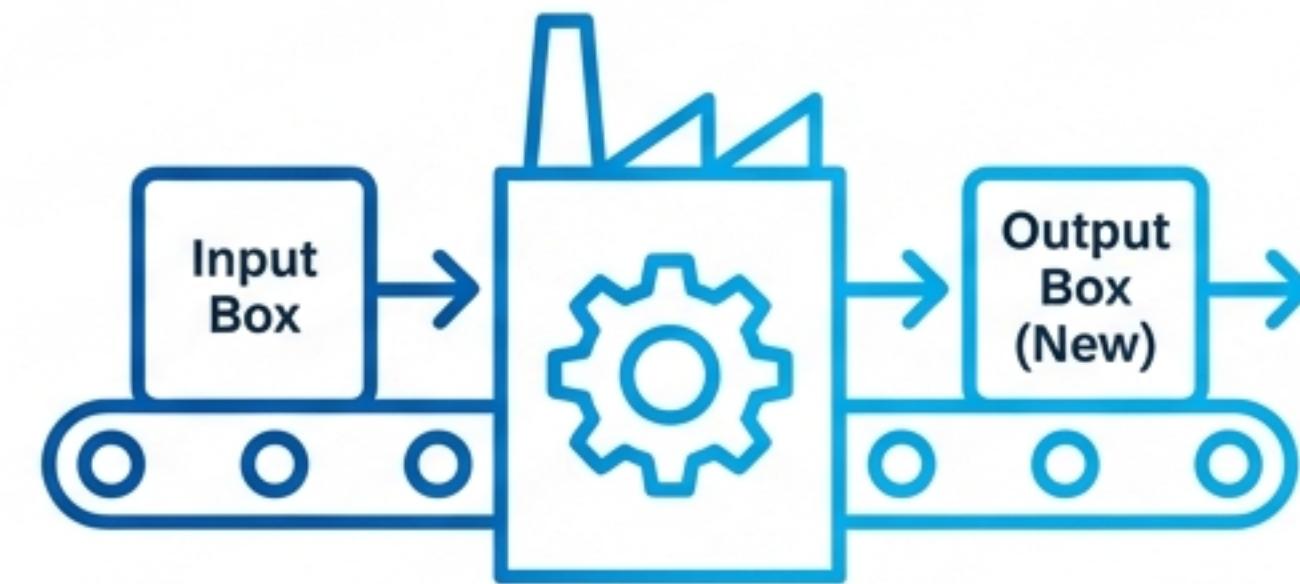
O Fim da Modificação Destruativa

Paradigma Imperativo



No ABAP Clássico, comandos modificavam a variável localmente (ex: TRANSLATE), impedindo o uso fluído.

Paradigma Funcional



No ABAP 7.40+, utilizamos uma biblioteca de funções que **retornam resultados**. Isso habilita o **Encadeamento (Chaining)** e preserva os dados originais.

Funções Embutidas: Case & Space

Maiúsculas/Minúsculas

Legado (Evitar)

```
TRANSLATE lv_text TO UPPER CASE.
```

Moderno (Usar)

```
lv_text = to_upper( lv_text ).
```

Suporta também `to_lower()` e `to_mixed()`.

Limpeza de Espaços

Legado (Evitar)

```
CONDENSE lv_text NO-GAPS.
```

Moderno (Usar)

```
lv_text = condense( val = lv_text  
                    from = ' ' ).
```

Atenção: A função `condense()` realiza *trim* nas pontas e reduz espaços internos múltiplos. Para remover **todos** os espaços, utiliza-se `replace`.

Segurança e Legibilidade em Substrings



Legado (Evitar)

```
lv_part = lv_text+0(5).
```

Sintaxe de Offset rígida. Causa **Short Dump** se o offset for maior que o tamanho da string.



Moderno (Usar)

```
lv_part = substring( val = lv_text  
                     off = 0  
                     len = 5 ).
```

Seguro e legível. Permite o uso de expressões dinâmicas para calcular o offset.

O Poder do Regex: Quando Texto Fixo Não Basta

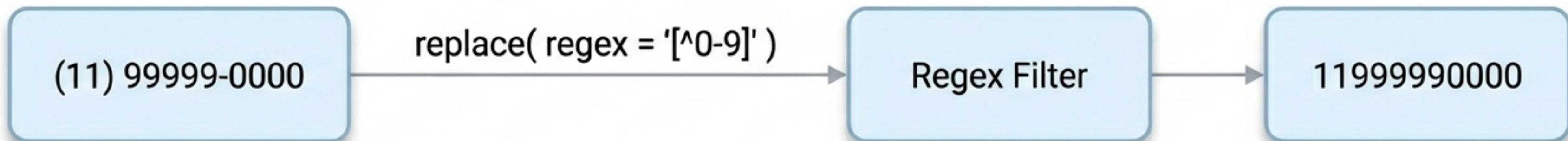


O ABAP moderno suporta Regex padrão POSIX/PCRE para validar padrões abstratos.

```
" Validação de E-mail
IF matches( val = lv_email regex = '^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$' ).
  " Email sintaticamente válido
ELSE.
  " Formato inválido
ENDIF.
```

Regex na Prática: Higienização de Dados

Recebimento de dados 'sujos' que precisam ser salvos apenas como números.

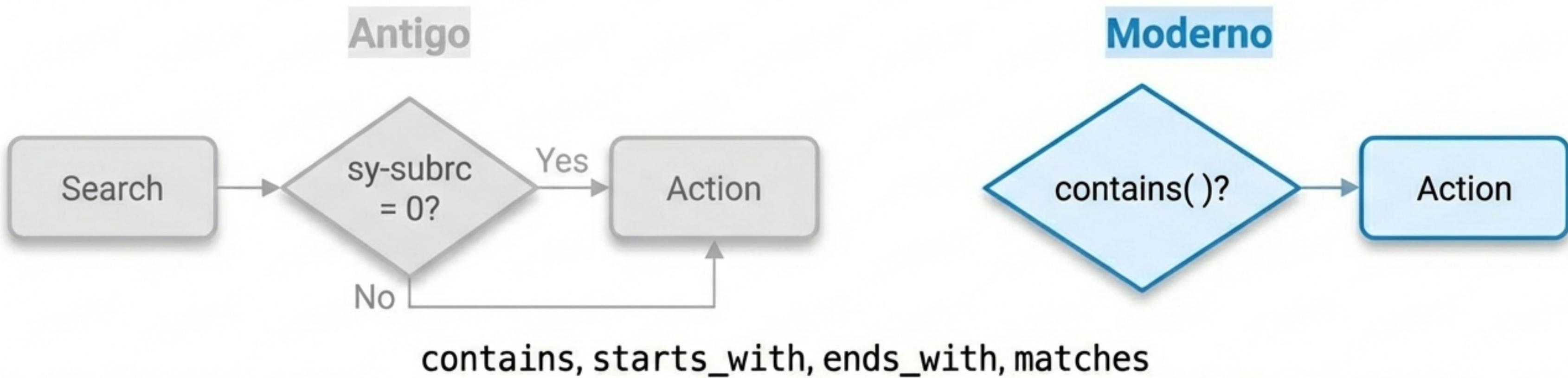


```
DATA(lv_clean) = replace( val    = lv_phone  
                         regex = '[^0-9]'  
                         with   = ''  
                         occ    = 0 ).
```

Insight: O Regex `[^0-9]` significa: Encontre qualquer caractere que **NÃO** (^) seja um dígito (0-9).

Funções Predicativas: Lógica Booleana Direta

Eliminam variáveis auxiliares e verificações de 'sy-subrc'. Retornam um booleano implícito.



Exemplo Prático: Higienização de Dados (Cenário)

Uma classe utilitária recebe dados brutos de uma API legado e precisa padronizá-los.

Data Card

Input Payload

{		
"Nome":	' joão da silva '	Espaços extras, minúsculo
"Email":	'JOAO.SILVA@SAP.COM'	Maiúsculo (Errado)
"SKU":	'MAT-1234-BR'	Mistura ID e Tipo
}		

Implementação da Solução

" 1. Encadeamento: Limpeza e Maiúsculas

```
DATA(lv_name) = to_upper( condense( val = lv_raw_name ) ).
```

" 2. Normalização: Minúsculas

```
DATA(lv_email) = to_lower( lv_raw_email ).
```

" 3. Extração Regex: Apenas dígitos (\d+)

```
find( val = lv_raw_sku regex = '\d+' sub = DATA(lv_extracted_num) ).
```

" 4. Predicados em Condicional (COND)

```
DATA(lv_type) = COND string(
    WHEN starts_with( val = lv_raw_sku sub = 'MAT' ) THEN 'Material Físico'
    WHEN starts_with( val = lv_raw_sku sub = 'SRV' ) THEN 'Serviço'
    ELSE 'Desconhecido'
).
```

Modern Syntax

Glossário Técnico

Built-in Functions: Funções nativas (ex: `strlen`, `to_upper`) utilizáveis em qualquer posição de operando.

Regex: Linguagem formal para descrever e validar padrões complexos de texto.

Chaining (Encadeamento): Chamada de função dentro do resultado de outra (ex: `func1(func2(x))`).

Predicative Function: Função que retorna verdadeiro/falso para uso lógico direto (ex: `matches`).

Tabela Comparativa: Antigo vs. Novo

Operação	Legado (Evitar)	Moderno (Usar)
Maiúsculas	TRANSLATE	`to_upper()`
Busca	SEARCH	`find()`
Padrão	CP (Contains Pattern)	`matches()`
Concatenação	CONCATENATE	String Templates ` { a }{ b } `
Substituição	REPLACE ... INTO	`replace(val = ...)`

Quiz de Fixação

Q1: Qual a diferença fundamental entre a função `condense()` moderna e o comando antigo `CONDENSE NO-GAPS`?

Q2: Qual função predicativa é a mais performática e legível para verificar um sufixo (ex: domínio de email)?

Q3: O que significa o padrão regex `[^0-9]` quando usado em uma limpeza de dados?

Pontos de Atenção

i

Imutabilidade

Funções retornam *novos* valores. Elas não alteram a variável original a menos que a atribuição seja explícita (`x = func(x)`).

i

Performance

Prefira funções específicas (`ends_with`) em vez de Regex genérico para verificações simples.

i

Legibilidade

Embora o encadeamento seja poderoso, evite linhas excessivamente longas. Quebre em passos se necessário para manter a clareza.

Código limpo, seguro e expressivo.
O futuro do ABAP é agora.

Escolha um programa antigo hoje e refatore o tratamento
de strings utilizando funções embutidas e Regex.

Obrigado.