

# Implementación de Autómatas en lenguajes C y Python

2025

## 1 Indicaciones

- Analizar y comprender los tres los autómatas proporcionados, las expresiones que validan y principalmente cómo se implementan dichos autómatas.
- Compilar y ejecutar (se sugiere emplear el lenguaje C o en su defecto Python).
- Prueben diferentes expresiones, válidas y no válidas, de para que sean validadas por los programas respectivos.

## 2 Introducción

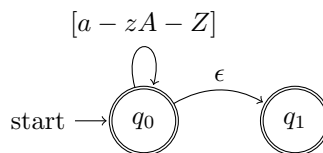
Este documento proporciona una guía técnica para implementar tres autómatas en los lenguajes de programación **C** y **Python**. Los autómatas validarán:

1. Cadenas alfabéticas (solo letras mayúsculas y minúsculas).
2. Números reales (con signo opcional y punto decimal).
3. Sentencias selectivas (por ejemplo, `if-else`).

## 3 Autómatas

### 3.1 Autómata 1: Validación de Cadenas Alfabéticas

Este autómata valida cadenas compuestas únicamente por letras mayúsculas y minúsculas.



### 3.1.1 Código en C

```
#include <stdio.h>
#include <ctype.h>

int validate_alpha(const char *str) {
    while (*str) {
        if (!isalpha(*str)) return 0;
        str++;
    }
    return 1;
}

int main() {
    const char *input = "HelloWorld";
    if (validate_alpha(input)) {
        printf("Cadena válida.\n");
    } else {
        printf("Cadena inválida.\n");
    }
    return 0;
}
```

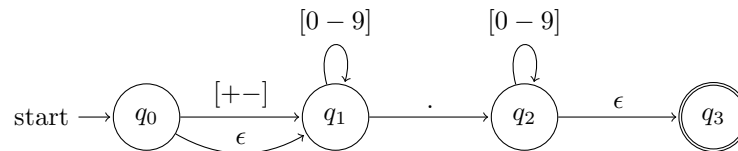
### 3.1.2 Código en Python

```
def validate_alpha(s):
    return s.isalpha()

input_str = "HelloWorld"
if validate_alpha(input_str):
    print("Cadena válida.")
else:
    print("Cadena inválida.")
```

## 3.2 Autómata 2: Validación de Números Reales

Este autómata valida números reales con signo opcional y punto decimal.



### 3.2.1 Código en C

```
#include <stdio.h>
```

```

#include <ctype.h>

int validate_real(const char *str) {
    int has_dot = 0;
    if (*str == '+' || *str == '-') str++;
    while (*str) {
        if (*str == '.') {
            if (has_dot) return 0;
            has_dot = 1;
        } else if (!isdigit(*str)) return 0;
        str++;
    }
    return has_dot;
}

int main() {
    const char *input = "-123.456";
    if (validate_real(input)) {
        printf("Número válido.\n");
    } else {
        printf("Número inválido.\n");
    }
    return 0;
}

```

### 3.2.2 Código en Python

```

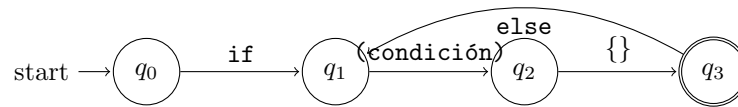
def validate_real(s):
    try:
        float(s)
        return True
    except ValueError:
        return False

input_str = "-123.456"
if validate_real(input_str):
    print("Número válido.")
else:
    print("Número inválido.")

```

## 3.3 Autómata 3: Validación de Sentencias Selectivas

Este autómata valida sentencias selectivas simples como if-else.



### 3.3.1 Código en C

```

#include <stdio.h>
#include <string.h>

int validate_if_else(const char *str) {
    return strstr(str, "if") != NULL && strstr(str, "else") != NULL;
}

int main() {
    const char *input = "if (x > 0) { } else { }";
    if (validate_if_else(input)) {
        printf("Sentencia válida.\n");
    } else {
        printf("Sentencia inválida.\n");
    }
    return 0;
}

```

### 3.3.2 Código en Python

```

def validate_if_else(s):
    return "if" in s and "else" in s

input_str = "if x > 0: pass else: pass"
if validate_if_else(input_str):
    print("Sentencia válida.")
else:
    print("Sentencia inválida.")

```