

UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS



Hands-on 1: Coding DFAs for validating Expr

Materia: Compiladores

Nombre del estudiante: Juan José Ascencio Tapia

Fecha: 26 de septiembre de 2025

Expresiones alfabéticas:

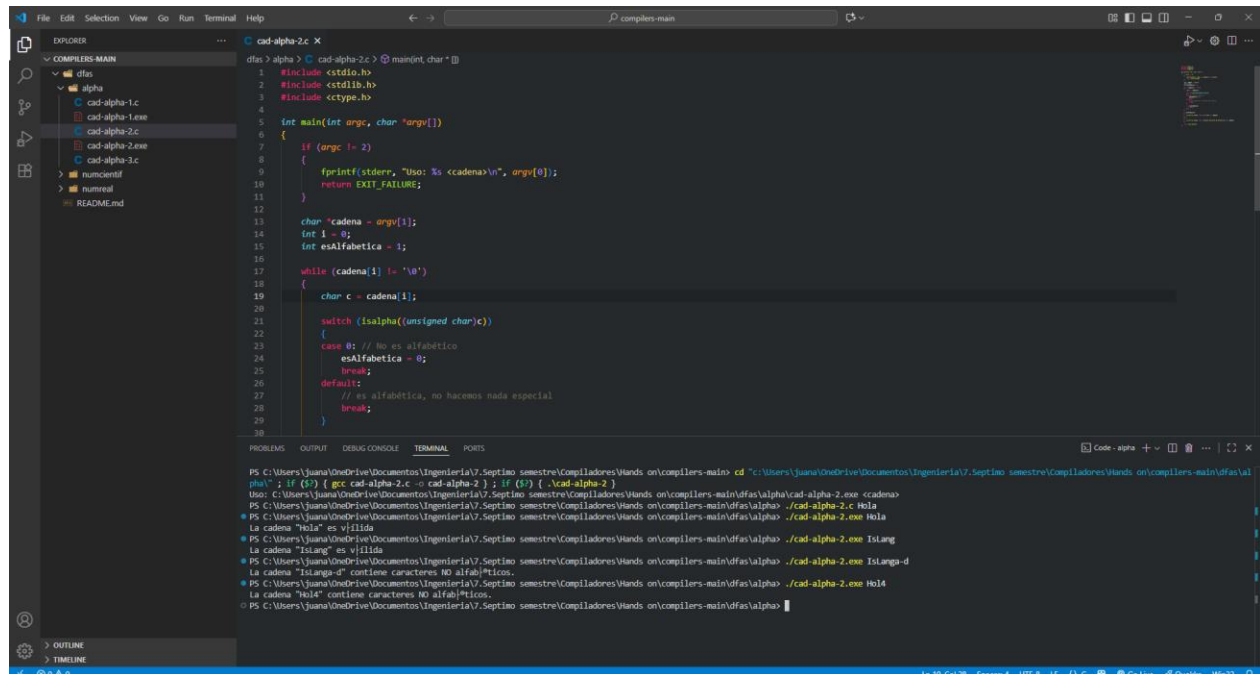


```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     if (argc != 2) {
6         fprintf(stderr, "uso: %s <cadena>\n", argv[0]);
7         return EXIT_FAILURE;
8     }
9
10    char *cadena = argv[1];
11    int i = 0;
12    int esAlfabetica = 1;
13
14    while (cadena[i] != '\0') {
15        switch (cadena[i]) {
16            case 'A': case 'B': case 'C': case 'D': case 'E':
17            case 'F': case 'G': case 'H': case 'I': case 'J':
18            case 'K': case 'L': case 'M': case 'N': case 'O':
19            case 'P': case 'Q': case 'R': case 'S': case 'T':
20            case 'U': case 'V': case 'W': case 'X': case 'Y': case 'Z':
21                // minúsculas
22            case 'a': case 'b': case 'c': case 'd': case 'e':
23            case 'f': case 'g': case 'h': case 'i': case 'j':
24            case 'k': case 'l': case 'm': case 'n': case 'o':
25            case 'p': case 'q': case 'r': case 's': case 't':
26            case 'u': case 'v': case 'w': case 'x': case 'y': case 'z':
27                break;
28            default:
29                esAlfabetica = 0;
30        }
31        i++;
32    }
33
34    if (esAlfabetica) {
35        printf("La cadena '%s' contiene caracteres alfabéticos.\n", cadena);
36    } else {
37        printf("La cadena '%s' contiene caracteres no alfabéticos.\n", cadena);
38    }
39
40    return EXIT_SUCCESS;
41 }
```

```
PS C:\Users\Juana\OneDrive\Documentos\Ingenieria\7. Septimo semestre\Compiladores\Hands on\compilers-main\dfas\alpha> ./validar.exe Hoalafundo
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (./validar.exe:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Juana\OneDrive\Documentos\Ingenieria\7. Septimo semestre\Compiladores\Hands on\compilers-main\dfas\alpha> cd "C:\Users\Juana\OneDrive\Documentos\Ingenieria\7. Septimo semestre\Compiladores\Hands on\compilers-main\dfas\alpha"
PS C:\Users\Juana\OneDrive\Documentos\Ingenieria\7. Septimo semestre\Compiladores\Hands on\compilers-main\dfas\alpha> ./cad-alpha-1.exe <cadena>
La cadena "Hoal" es válida
PS C:\Users\Juana\OneDrive\Documentos\Ingenieria\7. Septimo semestre\Compiladores\Hands on\compilers-main\dfas\alpha> ./cad-alpha-1.exe 8
La cadena "8" contiene caracteres no alfabéticos.
PS C:\Users\Juana\OneDrive\Documentos\Ingenieria\7. Septimo semestre\Compiladores\Hands on\compilers-main\dfas\alpha> ./cad-alpha-1.exe asffe
La cadena "asffe" contiene caracteres no alfabéticos.
PS C:\Users\Juana\OneDrive\Documentos\Ingenieria\7. Septimo semestre\Compiladores\Hands on\compilers-main\dfas\alpha> ./cad-alpha-1.exe Juan26
La cadena "Juan26" contiene caracteres no alfabéticos.
PS C:\Users\Juana\OneDrive\Documentos\Ingenieria\7. Septimo semestre\Compiladores\Hands on\compilers-main\dfas\alpha>
```

En el primer código podemos apreciar que cad-alpha-1.c realiza la comparación mediante un switch cuyos casos son las letras del abecedario en mayúsculas y minúsculas. Este código itera en la cadena, y en caso de que un carácter sea distinto al de los casos, imprime que la cadena contiene caracteres no alfabéticos. Se probó con las cadenas asf4e, Juan26, y 8, retornando que la cadena contiene caracteres no alfabéticos, siendo Hoal, la única cadena válida.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4
5 int main(int argc, char *argv[])
6 {
7     if (argc != 2)
8     {
9         fprintf(stderr, "Uso: %s <cadena>\n", argv[0]);
10        return EXIT_FAILURE;
11    }
12
13    char *cadena = argv[1];
14    int i = 0;
15    int esAlfabetica = 1;
16
17    while (cadena[i] != '\0')
18    {
19        char c = cadena[i];
20
21        switch (isalpha((unsigned char)c))
22        {
23            case 0: // No es alfabético
24                esAlfabetica = 0;
25                break;
26            default:
27                // es alfabética, no hacemos nada especial
28                break;
29        }
30        i++;
31    }
32
33    if (esAlfabetica)
34        printf("La cadena '%s' es válida\n", cadena);
35    else
36        printf("La cadena '%s' contiene caracteres no alfabéticos\n", cadena);
37
38    return EXIT_SUCCESS;
39 }
```

PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7. Séptimo semestre\Compiladores\Wands on\compilers-main\dfas\alpha> gcc cad-alpha-2.c -o cad-alpha-2.exe

PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7. Séptimo semestre\Compiladores\Wands on\compilers-main\dfas\alpha> ./cad-alpha-2.exe Hola

La cadena "Hola" es válida

PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7. Séptimo semestre\Compiladores\Wands on\compilers-main\dfas\alpha> ./cad-alpha-2.exe IsLang

La cadena "IsLang" es válida

PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7. Séptimo semestre\Compiladores\Wands on\compilers-main\dfas\alpha> ./cad-alpha-2.exe IsLang-d

La cadena "IsLang-d" contiene caracteres no alfabéticos

PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7. Séptimo semestre\Compiladores\Wands on\compilers-main\dfas\alpha> ./cad-alpha-2.exe Hol4

La cadena "Hol4" contiene caracteres no alfabéticos

En este código, cad-alpha-2.c se realiza la comparación de manera similar que su versión anterior. Con la diferencia de que aquí el switch se apoya de la función isalpha() para poder realizar las comparaciones de una manera mas limpia. Sus cadenas validas fueron: Hola, EsLang. Sus cadenas no válidas fueron: IsLanga-d, Hol4.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <ctype.h> // isalpha
4
5  #define ES_ALFABETICO(ch) (isalpha((unsigned char)(ch)) ? 1 : 0)
6
7  typedef enum
8  {
9      INICIO,
10     ACEPTACION,
11     ERROR
12 } Estado;
13
14 int main(int argc, char *argv[])
15 {
16     if (argc != 2)
17     {
18         fprintf(stderr, "uso: %s <cadena>\n", argv[0]);
19         return EXIT_FAILURE;
20     }
21
22     const char *cadena = argv[1];
23     Estado estado = INICIO;
24     int i = 0;
25
26     while (cadena[i] != '\0')
27     {
28         char c = cadena[i];
29
30         switch (estado)
31         {
32             case INICIO:
33                 if (ES_ALFABETICO(c))
34                     estado = ACEPTACION;
35                 else
36                     estado = ERROR;
37                 break;
38             case ACEPTACION:
39                 if (ES_ALFABETICO(c))
40                     estado = ACEPTACION;
41                 else
42                     estado = ERROR;
43                 break;
44             case ERROR:
45                 estado = ERROR;
46                 break;
47         }
48         i++;
49     }
50
51     if (estado == ACEPTACION)
52         printf("La cadena '%s' es valida.\n", cadena);
53     else
54         printf("La cadena '%s' no es valida.\n", cadena);
55
56     return EXIT_SUCCESS;
57 }
```

PS C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main> cd "C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main\dfas\alpha" && gcc cad-alpha-3.c -o cad-alpha-3 && if (\$?) { .\cad-alpha-3 }
uso: C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main\dfas\alpha\cad-alpha-3.exe <cadena>
PS C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main\dfas\alpha> .\cad-alpha-3.exe Hola
La cadena 'Hola' es valida.
PS C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main\dfas\alpha> .\cad-alpha-3.exe Ho7!
La cadena 'Ho7!' no es valida.

En esta ultima version del código, se realiza la comparación mediante estados, si el estado es de aceptación o en su defecto de inicio, se dará por valida la cadena, en caso contrario, se declarara invalida. Siendo Hola válida y Ho7! Inválida.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4
5 // expresiones válidas "3", "47", "-8", "-87", "12.7", "12.34", "1e3", "-2.5E-8", ".7e+10", "12.e3"
6
7 typedef enum {
8     S1, S2, S3, S4, S5, S6, S7, S8, S9
9 } State;
10
11 static int es_valida(const char *s) {
12     State st = S1;
13     for (size_t i = 0; i < strlen(s); i++) {
14         unsigned char uc = (unsigned char)s[i];
15         if (uc == '\0') {
16             if (st == S3 || st == S5 || st == S8) {
17                 st = S9;
18                 return 1; /* válido */
19             }
20             return 0; /* inválido */
21         }
22         switch (st) {
23             case S1:
24                 if (uc == '+' || uc == '-') {
25                     st = S2;
26                 } else if (isdigit(uc)) {
27                     st = S3;
28                 } else if (uc == '.') {
29                     st = S4;
30                 }
31             // ... other cases ...
32         }
33     }
34     return 0;
35 }
```

PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7.º semestre\Compiladores\Wands on compilers-main\dfas\numcientif> ./main.exe 45
que dicha ruta es correcta e intento de nuevo.
En línea: 1 Carácter: 1
+ ./main.exe 45
+ ~~~~~
+ CategoryInfo : ObjectNotFound: (C:\main.exe:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7.º semestre\Compiladores\Wands on compilers-main\dfas\numcientif> cd "C:\Users\Juana\OneDrive\Documents\Ingeniería\7.º semestre\Compiladores\Wands on compilers-main\dfas\numreal" && gcc num-real-states.c -o num-real-states && if (\$?) { gcc num-real-states.c -o num-real-states }
PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7.º semestre\Compiladores\Wands on compilers-main\dfas\numreal> ./num-real-states.exe 45
La cadena "45" es válida en notación científica.
PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7.º semestre\Compiladores\Wands on compilers-main\dfas\numreal> ./num-real-states.exe 45.0
La cadena "45.0" es válida en notación científica.
PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7.º semestre\Compiladores\Wands on compilers-main\dfas\numreal> ./num-real-states.exe 45.
La cadena "45." es válida en notación científica.
PS C:\Users\Juana\OneDrive\Documents\Ingeniería\7.º semestre\Compiladores\Wands on compilers-main\dfas\numreal>

De manera similar el código num-real-states.c permite que mediante estados compare si un número es válido en notación científica, siendo objeto de prueba 45,45.0 y 40. Siendo los 3 válidos.

The image shows a Visual Studio Code editor window with the file explorer on the left displaying a project structure. The main editor area shows the source code of `num-real.c`. The code defines a `main` function that takes a string `cadena` and processes it to determine if it is a valid decimal number. It uses a `switch` statement to handle different characters and a `printf` statement to output the result.

```
10 int main(int argc, char *argv[])
11 {
12     for (size_t i = 0; cadena[i] != '\0'; ++i)
13     {
14         switch (cadena[i])
15         {
16             case '0':
17             case '1':
18             case '2':
19             case '3':
20             case '4':
21             case '5':
22             case '6':
23             case '7':
24             case '8':
25             case '9':
26                 break;
27             case '.':
28                 if (st == ERROR)
29                     break;
30                 if (es_estado_aceptacion(st))
31                 {
32                     printf("La cadena \"%s\" es válida.\n", cadena);
33                     return EXIT_SUCCESS;
34                 }
35                 else
36                 {
37                     printf("La cadena \"%s\" NO es un número decimal válido.\n", cadena);
38                     return EXIT_FAILURE;
39                 }
40             default:
41                 break;
42         }
43     }
44 }
```

The terminal at the bottom shows the execution of the program for various inputs:

```
PS C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main\dfas\numreal> ./num-real.exe 45.
La cadena "45." es válida.
PS C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main\dfas\numreal> ./num-real.exe 0
La cadena "0" es válida.
PS C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main\dfas\numreal> ./num-real.exe -84.4
La cadena "-84.4" NO es un número decimal válido.
PS C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main\dfas\numreal> ./num-real.exe 1/4
La cadena "1/4" NO es un número decimal válido.
PS C:\Users\Juana\OneDrive\Documents\Ingenieria\7.Semestre\Compiladores\Wands on\compilers-main\dfas\numreal> ./num-real.exe .45
La cadena ".45" es válida.
```

En el código de `num-real.c` no se permiten números negativos. Teniendo como objeto de prueba: 45.,45,0,-84.4,1/4,.45. Siendo cadenas válidas los números: 45, 0 y .45.