




14 DE MARZO DE 2019

# RELACIÓN 2. ESTRUCTURAS DE CONTROL EN ADA

SISTEMAS DE TIEMPO REAL

LAURA AGUILERA CHECA  
IGNACIO AGUILERA GÓMEZ  
GRADO EN INGENIERÍA INFORMÁTICA  
Universidad de Almería



## Contenido

1. Autores .....	2
2. Introducción.....	2
3. Actividades a realizar .....	2
3.1. Ejercicio 1 - INSTRUCCIONES Y ESTRUCTURAS DE CONTROL.....	2

## 1. Autores

Apellidos	Aguilera	Checa
Nombre	Laura	
Titulación	Grado de Ingeniería Informática	

Apellidos	Aguilera	Gómez
Nombre	Ignacio	
Titulación	Grado de Ingeniería Informática	

## 2. Introducción

Con estos ejercicios se pretende realizar un acercamiento del alumno a la programación en ADA. Para ello, se plantea un ejercicio sencillo de utilización de las instrucciones y estructuras básicas de control. Dicho ejercicio deberá de ser remitido al profesor a través del módulo de Tareas de la plataforma Aula Virtual antes de la finalización del plazo de envío indicado. El formato de entrega de los ejercicios deberá de ser un documento PDF en el que se incluyan los comentarios y el código del programa realizados.

## 3. Actividades a realizar

### 3.1. Ejercicio 1 - INSTRUCCIONES Y ESTRUCTURAS DE CONTROL

Escribir un programa en ADA con las siguientes características:

1. Inicializar una variable tipo cadena que contenga los caracteres "ABCDEFG".  
`str : String := "ABCDEFG";`
2. Realizar un bucle para extraer la información de la cadena anterior carácter a carácter e imprimir por pantalla en función del carácter leído:
  - "A" mostrar "Opción 1"
  - "B": mostrar "Opción 2".
  - "C" o "F": mostrar "Opción 3".
  - En otro caso: mostrar "Otra opción".

```
for i in 1..str'Length loop
    c:=str(i);
    case c is
        when 'A' | 'B' => Ada.Text_IO.Put_Line( "Opcion 1" );
        when 'C' | 'D' | 'E' => Ada.Text_IO.Put_Line( "Opcion 2" );
        when 'F' => Ada.Text_IO.Put_Line( "Opcion 3" );
        when others => Ada.Text_IO.Put_Line( "Otra opcion" );
    end case ;
end loop ;
```

```
Opcion 1
Opcion 1
Opcion 2
Opcion 2
Opcion 2
Opcion 3
Otra opcion
[2019-03-15 23:16:18] process terminated succes
```

3. OPCIONAL: Extender el código anterior para admitir una cadena de entrada por teclado  
Las claves para este apartado son:

- Necesitamos una variable para almacenar el string de entrada:  
*str : String (1..1\_000);*
- Necesitamos una variable para almacenar el tamaño de este string:  
*last: Natural;*
- Y necesitamos obtener esta entrada por consola:  
*Ada.Text\_IO.Get\_Line (str, last);*

```
EEAAFFG
Opcion 2
Opcion 2
Opcion 1
Opcion 1
Opcion 3
Opcion 3
Otra opcion
[2019-03-15 23:17:50] process terminated successfully, elapsed time: 09.55s
```