

# Proyecto lógico

Lenguajes de Programación  
Universidad Nacional  
Eddy Ramírez  
II-Semestre 2020

23 de octubre de 2020

## 1. El contador de microondas ¿dañado?

En general, cuando se quieren representar números en un BCD, se realiza mediante prender o apagar siete leds que representen los dígitos.

Para efectos de esta tarea, se puede asumir que los leds son nombrados tal como se muestra en la figura 1.

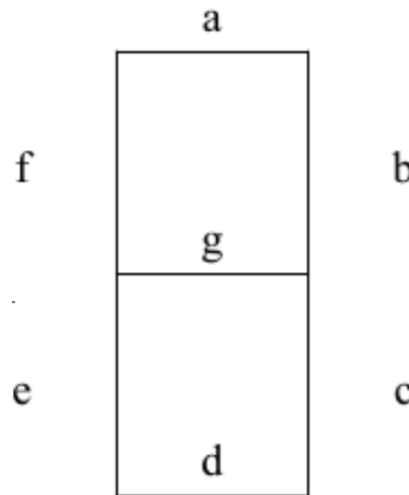


Figura 1: Leds en un display

Entonces, si se quiere mostrar un dígito en particular, se deben encender ciertos leds y apagar el resto para que luzcan como se expresan los dígitos

en decimal. Para ello se muestra la siguiente tabla (donde Y es prendido y N es apagado):

Números:	a	b	c	d	e	f	g
0	Y	Y	Y	Y	Y	Y	N
1	N	Y	Y	N	N	N	N
2	Y	Y	N	Y	Y	N	Y
3	Y	Y	Y	Y	N	N	Y
4	N	Y	Y	N	N	Y	Y
5	Y	N	Y	Y	N	Y	Y
6	Y	N	Y	Y	Y	Y	Y
7	Y	Y	Y	N	N	N	N
8	Y	Y	Y	Y	Y	Y	Y
9	Y	Y	Y	Y	N	Y	Y

Figura 2: Tabla de valores BCD

Resulta que mediante esta configuración se programan los displays de cronómetros o microondas, las cuales deben mostrarse descendientemente.

Este problema ocurrió en la realidad en una empresa, la cual estaba probando los temporizadores exhaustivamente como parte del control de calidad de los microondas (no se quería que el microondas no se detuviera transcurrido el tiempo asignado). Se probaba una y otra vez que cualquier secuencia de números de la forma min:dsec:usec (minutos, decenas de segundos y unidades de segundos) fuera siempre una secuencia decremental. Nótese que alguien podría poner 90 segundos y alguien más podría poner 130 (un minuto con 30 segundos), e igualmente debe de durar el mismo tiempo.

El problema que enfrentaron es que fueron tan exhaustivos en las pruebas que a veces había leds que se quemaban, por lo que lo que se desplegaba en pantalla podía no ser exactamente lo que se esperaba, sin embargo, sí se trataba de una secuencia correcta. Otro problema que enfrentaron fue que a veces durante una secuencia de decremento, habían leds que se quemaban durante la prueba, pero ésta debía indicar si lo mostrado correspondería con una secuencia descendiente o no.

Su trabajo corresponde con hacer en prolog un programa que, dada una lista de strings (de 21 chars) que representan 3 dígitos de un BCD, indicar si se tratan de una secuencia descendiente o no. Prolog debe indicar ‘yes’ o ‘no’ según corresponda. Debe considerar que es posible que se hayan quemado

algunos leds antes o durante la prueba (a partir de cierta posición en la lista, el led aparecería quemado).

### 1.1. Ejemplos

```
- decremento(["YYYYYYYYYYYYYYYYYY"] ).
yes

- decremento(["NNNNNNNNNNNNNNNNNNNN", "NNNNNNNNNNNNNNNNNNNN"] ).
yes

- decremento(["YYYYYYYYYYYYYYNYNY", "YYYYYYYYYYYYYYNYNYNY"] ).
yes

- decremento(["YYNNYYNNYYNNYYYYYYYY", "YYNNYYNNYYNNYYYYYYYY"] ).
no
```

## 2. Documentación

Todo proyecto debe venir con la documentación en  $\text{\LaTeX}$  que conste de:

- Portada

Con el logo oficial de la institución, título del proyecto, nombre del curso, nombre de los estudiantes, nombre del profesor y fecha.

- Resumen ejecutivo

En una página (no más, no menos) un resumen del proyecto, su solución y los resultados de las pruebas

- Introducción

Una descripción del documento y en qué consistía el proyecto.

- Marco teórico

Una descripción de los lenguajes utilizados (antecedentes, historia, usos), una breve documentación de las bibliotecas utilizadas (si las hubo) y de los ide's utilizados.

- Descripción de la solución

Detalle del algoritmo de solución para cada caso (backtracking y genéticos) y problemas encontrados durante el desarrollo del proyecto y su respectiva solución.

- Resultados de pruebas

Una vez finalizado el proyecto, debe de tener realizados diversos casos de prueba, tomar tiempo de ejecución (usando el time de cada lenguaje) y mostrar gráficas del contraste de resultados utilizando cada posible solución. (No se esperan menos de 10 contrastes en casos de prueba para cada algoritmo)

- Conclusiones

Un análisis de las gráficas y del proyecto en general. Así como una explicación teórica de los resultados.

- Aprendizajes

De manera individual, qué aprendió cada integrante de los equipos con la realización de este proyecto, a la luz de los objetivos del curso.

- Bibliografía (en formato IEEE –sólo la bibliografía)

Se rebajará un punto por cada falta ortográfica presente en la documentación. Si el documento no tiene bibliografía o no está hecho en Latex, se otorgará una nota de cero.

### 3. Rúbrica

Se asignarán dos rúbricas, una para la parte programada  $p$  y otra para la documentación  $d$ . La nota del proyecto será el promedio geométrico de ambas notas ( $\sqrt{p \times d}$ ).

#### 3.1. Programas lógicos

Producto	Valor
Respeto al paradigma de programación	20 %
Algoritmo sin leds quemados	20 %
Algoritmo con leds quemados	30 %
Algoritmo con leds quemados en caliente	30 %
<b>Total</b>	100 %

Tener presente que el respeto al paradigma de programación, debe mantenerse a lo largo de todo el proyecto. El compilador que debe de utilizarse es Gprolog o se obtendrá una nota de cero.

### 3.2. Documentación

Tener presente que la ortografía resta 1 punto por cada falta y no tener bibliografía implica una nota de cero, lo mismo que no hacerla en L<sup>A</sup>T<sub>E</sub>X.

Producto	Valor
Portada	5 %
Resumen ejecutivo	10 %
Introducción	5 %
Marco teórico	10 %
Descripción de la solución	20 %
Resultados de pruebas	35 %
Conclusiones	10 %
Aprendizajes	5 %
<b>Total</b>	<b>100 %</b>

## 4. Aspectos Técnicos

- El trabajo debe de ser realizado de forma individual, en parejas o tríos.
- Toda la programación debe realizarse en Gprolog y será revisado sobre Linux.
- **Todas las relaciones principales deben tener un comentario antes del nombre de la misma indicando lo que recibe como parámetro y lo que ancla.**
- Los nombres de los archivos deben ser las tres primeras letras de los apellidos de los integrantes. Ejemplo: *ram.pl*.

## 5. Aspectos Administrativos

- La fecha de entrega de este proyecto es el 23 de noviembre de 2020 a la hora de revisión.
- El medio de entrega será por correo electrónico a eddy.ramirez.jimenez@una.cr

- El asunto del correo debe ser: 2020-02 EIF-400 Proyecto Lógico [Nombre del estudiante(s)]<sup>1</sup> y debe enviarse con copia a todos los miembros del grupo.

---

<sup>1</sup>Siendo nombre del estudiante el nombre de cada estudiante, no debe llevar los paréntesis cuadrados