

Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:	M. I. Marco Antonio Martínez Quintana
Asignatura:	Fundamentos de programación
Grupo:	3
No de Práctica(s):	#8
Integrante(s):	Fausto Ángel Reséndiz Álvarez
No. de Equipo de cómputo empleado:	No aplica
No. de Lista o Brigada:	38
Semestre:	2021-1
Fecha de entrega:	27/11/2020
Observaciones:	
-	
_	CALIFICACIÓN:

Objetivo

Elaborar programas en lenguaje C que incluyan las estructuras de selección if, if-else, switch y ternaria (o condicional) para la resolución de problemas básicos.

Introducción

Las estructuras de control de flujo en un lenguaje especifican el orden en que se realiza el procesamiento de datos, en otras palabras, las acciones que se deben realizar y las que no.

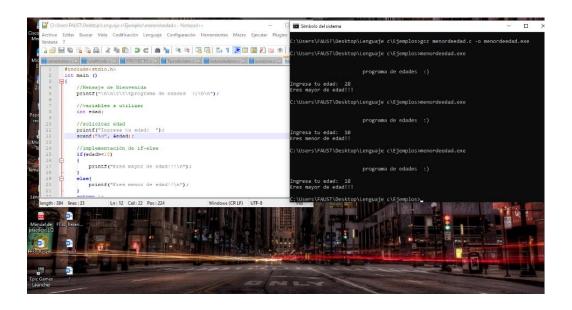
Las estructuras de selección permiten realizar una u otra acción con base en una expresión lógica.

Tipos de estructuras:

- *if*
- if else
- *Switch case*

Desarrollo

Estructuras de selección: if -else Switch



```
| Consecution Description programme (Superior No. 1985) | Consecution |
```

```
### Comparison Continues of Comparison Continues of Cont
```

```
Constraint Modification (Lifemplos per menu.comenu.ene

Constraint Modification (Lifemplos per menu.comenu.ene

Stenvenidos a muestra calculadora :)
) puna
) mutigalicación
) mutigalicación
) mutigalicación
constraint (Lifemplos permu, ene

Stenvenidos a realizar: 1

Desego a fundro se paracido por consisto, 44

La suma de 18 y 18 es; 28

La suma de 18 y 18 es; 28

Stenvenidos a muestra calculadora :)
) mutigalicación
) mutigal
```

```
### Simport of the control of the co
```

Conclusión

Las estructuras de control nos permiten darle variación a las acciones que colocamos en el desarrollo del código permitiéndole al programa decidir la acción a realizar en base a lo que definimos anteriormente.

Dándole un valor muy importante a este tipo de estructuras y es crucial que nosotros como programadores dominemos estas estructuras para así lograr un mejor y aun más complejo desarrollo de programas.