

Programacion II [PRACTICA]

1
0
0
1
0
1
0
1
0
1
0
1
0
1
0
0
0

1
0
1
0
0
1
0
1
0
1
0
0
1
0
1
0

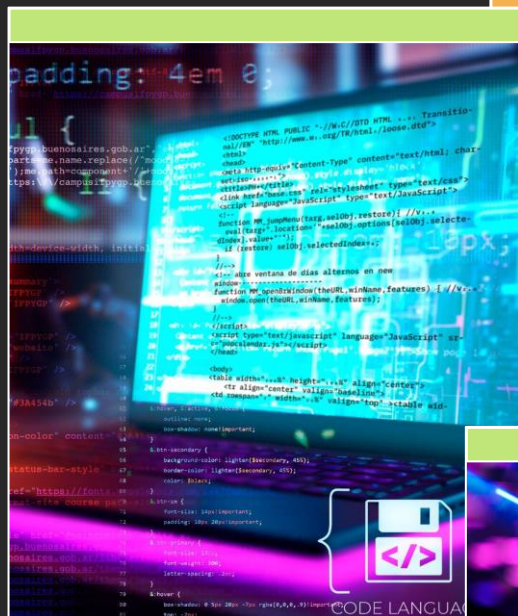


CLASES

| | |
|-----------|---------|
| Martes | 19 a 22 |
| Miercoles | 19 a 22 |



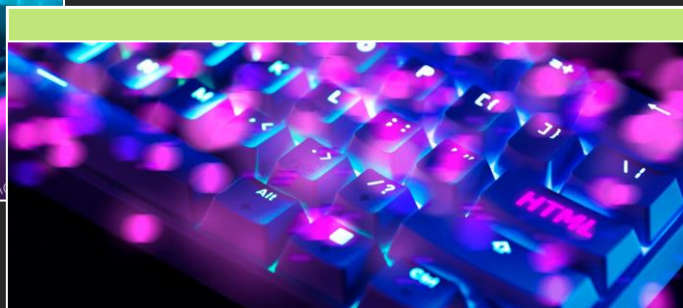
1
0
1
0
1
0
1
0
1
1
0
0
1
1
1
0
1
0



01

INTRODUCCION

1
1
0
0
1
0
1
0
1
0
1
1
1
0



DIAS Y HORARIOS



01

TEORIA

Ing. De la Puente Matias

Lunes

18 a 21 hs (virtual)

02

PRACTICA

Ing. Oviedo Carlos

Martes

19 a 22 hs (presencial)

COMISION A

COMISION B

Miércoles

19 a 22 hs (presencial)

1
0
1
0
1
0
1
1
1
0
0
0
0
1
1
1
0

1
1
0
0
1
0
1
0
1
0
1
0
1
1
1
0

Condiciones de Regularización y/o Promoción ●●

Regularización

1
0
1
0
1
0
1
1
1
0
0
0
1
1
1
0

01

Asistencia 70%

02

*Aprobar exámenes con
una nota mínima de 6*

03

*Aprobar trabajos
prácticos con una nota
mínima de 6*

Promoción

1
1
0
0
1
0
1
0
1
0
1
0
1
0
1
1
0

01

Asistencia 80%

02

*Aprobar exámenes con
una nota mínima de 8*

03

*Aprobar trabajos
prácticos con una nota
mínima de 8*

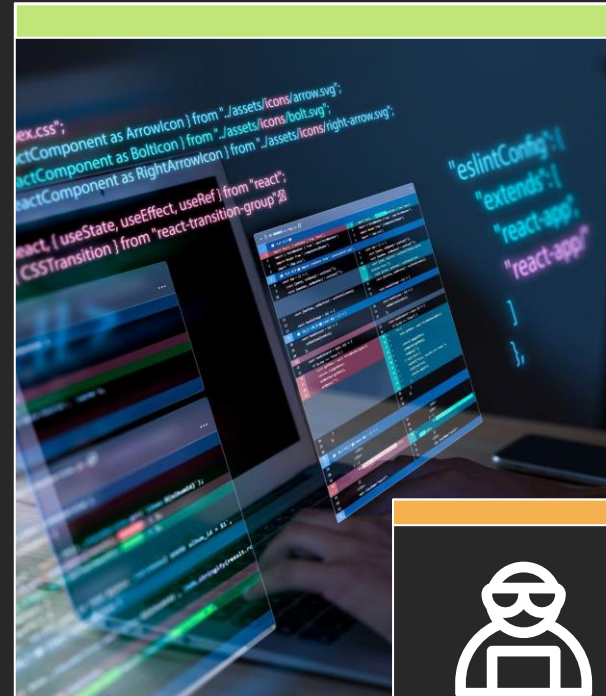
Parciales & Recuperatorios



Evaluación teórica presencial de tipo cuestionario:

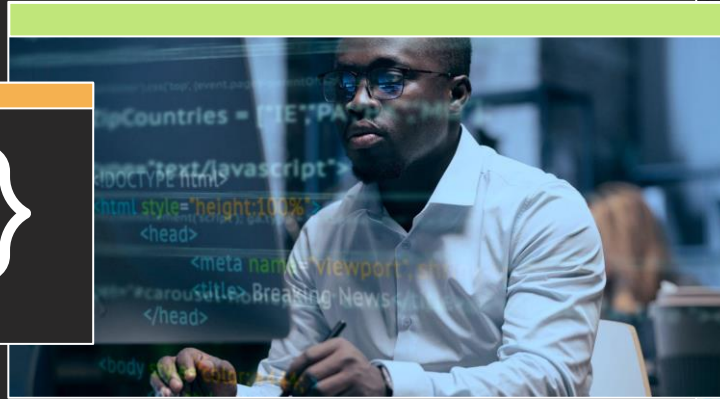
- Completar.
- Opciones Múltiple.
- Verdadero/Falso.
- Unir con flechas

De entre 15 a 20 preguntas a realizarse en una hora.



1
0
1
0
0
1
0
1
0
1
0
0
1
0
1
0

Trabajos Prácticos Individuales



1
0
1
0
0
1
0
1
0
0
1
0
1
0

Los trabajos prácticos individuales consistirán en la resolución de 10 ejercicios de forma individual en el transcurso de 5 días aproximadamente. Se dispondrá de un tiempo extra de entrega de 12 horas. La entrega se realizará únicamente en el campus virtual de la materia.

No se permiten entregas fuera de los plazos establecidos.

1
0
0
1
0
1
0
1
0
1
0
1
0
0

Fechas Importantes



1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0

| | |
|--------------------------|---|
| <u>Septiembre</u> | 09/09 al 14/09: Semana de exámenes finales. Entrega del primer trabajo práctico. |
| <u>Octubre</u> | 21/10: Primer parcial. Entrega del segundo trabajo práctico. |
| <u>Noviembre</u> | 18/11: Recuperatorio. 25/11 al 27/11: Entrega del trabajo práctico grupal. |

1
0
1
0
0
1
0
0
1
0
1
0
1
1
1
1

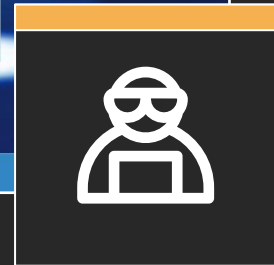
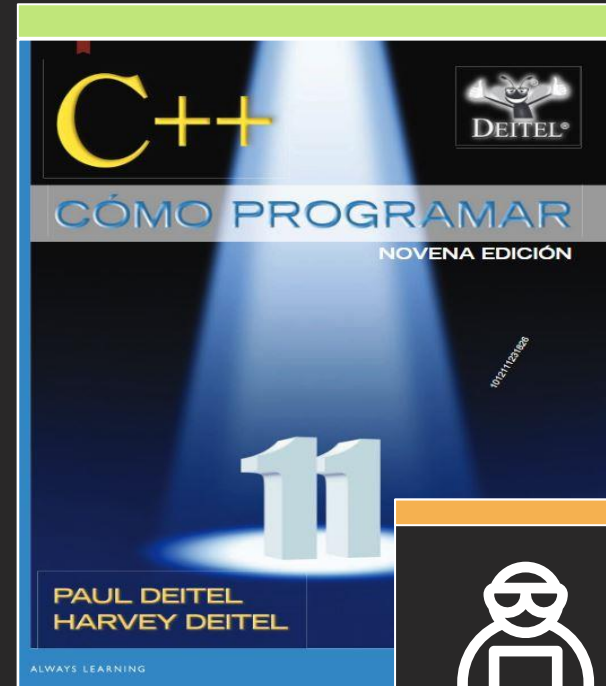
Bibliografía



- **C++ Como Programar 9º Edición**
Paul Deitel / Harvey Deitel

Bibliografía Complementaria

- **Programación en C++ Un Enfoque Practico**
Schaum
- **C++ Soluciones de Programación**
Hern Schildt



1
0
1
0
0
1
0
1
0
1
0
0
1
0
1
0

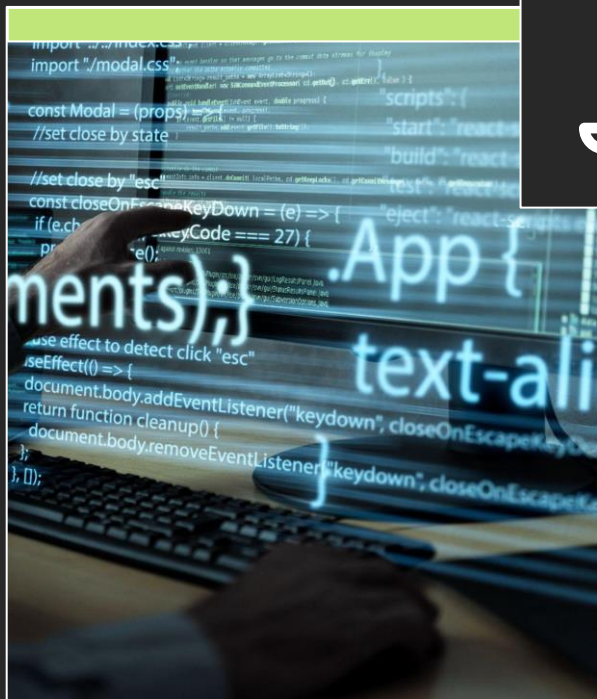


02

IDE

1
1
0
0
1
0
1
0
1
0
1
0
1
1
1
0

1
0
1
0
1
0
1
1
1
0
0
0
1
1
1
0





Qt Creator es un IDE multiplataforma programado en C++, JavaScript y QML creado por Trolltech el cual es parte de SDK para el desarrollo de aplicaciones con Interfaces Gráficas de Usuario (GUI por sus siglas en inglés) con las bibliotecas Qt, Los sistemas operativos que soporta en forma oficial son:

GNU/Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado. Además hay una versión para Linux con gcc 3.3.

Mac OS X 10.4 o superior, requiriendo Qt 4.x

Windows XP y superiores, requiriendo el compilador MinGW y Qt 4.4.3 para MinGW.

1
0
0
1
0
1
0
1
0
1
0
1
0
0

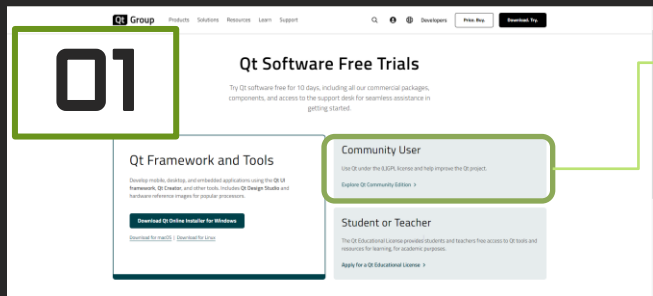
1
0
1
0
0
1
0
1
0
0
1
0
1
0

Descargar QT Creator

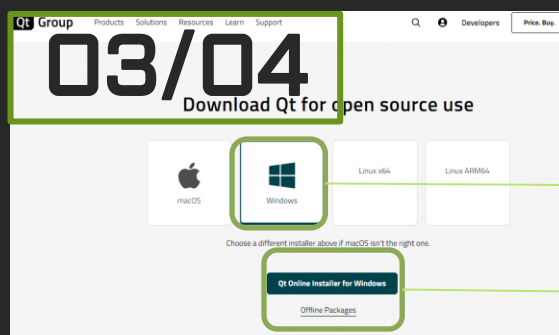
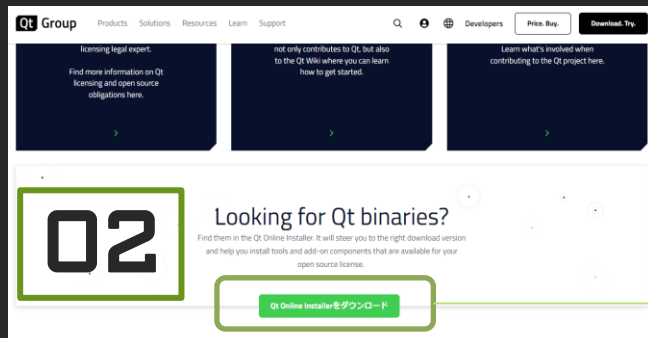


Para descargar el IDE QT:

<https://www.qt.io/download-dev>



1. Ingresar a Community User
2. Qt Online Installer
3. Seleccionar Sistema Operativo
4. Descargar:
 - QT Online Installer
 - Offline Packages



1
0
1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0



TUTORIALES DE QT Creator (INSTALAR)

Tutoriales en Youtube para Instalar QT Creator para Windows y Linux

1
0
0
1
0
1
0
0
1
0
1
0
1
0
1
0

| | |
|-------------------------------------|---|
| <u>DuarteCorporation Tutoriales</u> | https://www.youtube.com/watch?v=x9XeCq9tOI0 |
| <u>Qt With Ketan</u> | https://www.youtube.com/watch?v=Aw7gozXQE4c |
| <u>Joaquin Lara Sierra</u> | https://www.youtube.com/watch?v=VpDxfR7iFYc |
| <u>Gabriel Carvalho</u> | https://www.youtube.com/watch?v=tNbLFqZ11q8 |
| <u>ZacsTech</u> | https://www.youtube.com/watch?v=YZgb1f8-LJU |
| <u>LinuxSimply</u> | https://www.youtube.com/watch?v=mVKdomkl5Uk |

1
0
1
0
0
1
0
0
1
0
1
1
1
0
1
1
1

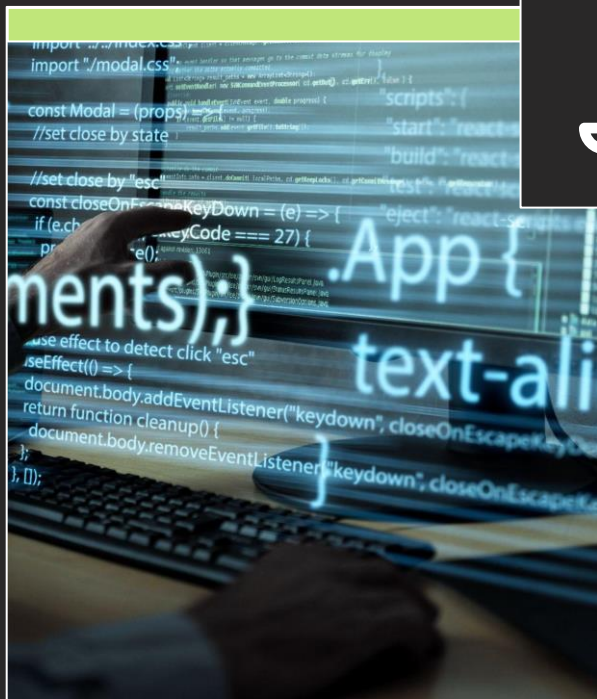


03

REPASAMOS

1
1
0
0
1
0
1
0
1
0
1
0
1
1
1
0

1
0
1
0
1
0
1
1
1
0
0
0
1
1
1
0



Repasamos C



- | | | | | |
|---|---|-----|--|---|
| 1 | 1. ¿Cuál es la diferencia entre los tipos de datos int, float, y double en C? | | dirección de memoria apuntada por un puntero? | |
| 0 | | | | |
| 0 | 2. ¿Qué tipo de dato utilizarías para almacenar un carácter? | 9. | ¿Cuál es la diferencia entre una función que retorna un valor y una que no lo hace (void)? | |
| 1 | | | | |
| 0 | 3. ¿Y para un número muy grande? | 10. | ¿Cómo pasas un puntero como argumento a una función y qué ventaja tiene esto? | 1 |
| 1 | | | | 0 |
| 0 | 4. ¿Cuándo utilizarías una estructura if/else en lugar de un simple if? | 11. | ¿Qué es una estructura (struct) en C y cuándo es útil? | 1 |
| 1 | | | | 0 |
| 0 | 5. ¿Qué ventaja ofrece el uso de switch en comparación con múltiples if/else? | 12. | ¿Cómo puedes acceder a los miembros de una estructura? | 0 |
| 0 | | | | 0 |
| 1 | 6. ¿En qué casos es más apropiado usar un ciclo for en lugar de un while? | 13. | ¿Qué pasos debes seguir para leer desde un archivo en C? | 1 |
| 0 | Explica la diferencia entre do while y while. | | | 1 |
| | | | | 0 |
| | 7. ¿Qué es un puntero en C y cómo se declara? | 14. | Explica la diferencia entre las funciones fscanf y fprintf. | 1 |
| | | | | 1 |
| | 8. ¿Cómo accedes al valor almacenado en la | | | 1 |

Ejercicio de Repaso: Calculadora Básica



Escribe un programa en C que implemente una calculadora básica. El programa debe:

1. Pedir al usuario que ingrese dos números.

2. Mostrar un menú con las opciones:

A. Sumar,

B. Restar,

C. Multiplicar,

D. Dividir.

3. Leer la opción seleccionada por el usuario.

4. Usar una estructura switch para realizar la operación correspondiente.

5. Mostrar el resultado en pantalla.

6. Preguntar al usuario si desea realizar otra operación (utilizando un bucle do-while).

OBJETIVO :

Desarrollar una calculadora básica que permita al usuario realizar operaciones aritméticas (suma, resta, multiplicación, división) utilizando funciones, estructuras condicionales y bucles.

NOTA: Validar todos los ingresos de datos del usuario.

Ejercicio de Repaso: Promedio de Calificaciones



Escribe un programa en C que calcule el promedio de las calificaciones de un grupo de estudiantes. El programa debe:

1. Pedir al usuario que ingrese la cantidad de estudiantes.
2. Declarar un vector para almacenar las calificaciones.
3. Utilizar un bucle for para leer las calificaciones desde el teclado.
4. Implementar una función que reciba el vector y la cantidad de estudiantes, y devuelva el promedio de las calificaciones.
5. Mostrar el promedio en pantalla.

OBJETIVO :

Trabajar con vectores y funciones para calcular el promedio de un conjunto de calificaciones ingresadas por el usuario.

NOTA: Validar todos los ingresos de datos del usuario.

1
0
0
1
0
1
0
1
0
1
0
1
0
1
0

1
0
1
0
0
1
0
0
0
1
0
1
0
1
1

Ejercicio de Repaso: Transpuesta de una Matriz



Escribe un programa en C que calcule la transpuesta de una matriz. El programa debe:

1. Pedir al usuario que ingrese el tamaño de la matriz (número de filas y columnas).
2. Declarar una matriz bidimensional y leer sus elementos desde el teclado.
3. Implementar una función que reciba la matriz y sus dimensiones, y calcule la transpuesta.
4. Mostrar la matriz transpuesta en pantalla.

OBJETIVO:

Manejo de matrices y funciones para calcular la transpuesta de una matriz.

NOTA: Validar todos los ingresos de datos del usuario.

1
0
0
1
0
1
0
1
0
0
1
0
1
0
0

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1

Ejercicio de Repaso: Operaciones con Punteros



Escribe un programa en C que realice las siguientes operaciones utilizando punteros:

1. Declarar dos variables enteras e inicializarlas con valores ingresados por el usuario.
2. Implementar una función que reciba dos punteros enteros y los intercambie (swap).
3. Implementar una función que reciba dos punteros enteros y calcule su suma y producto, almacenando los resultados en variables pasadas por referencia.
4. Mostrar los resultados de las operaciones en pantalla.

OBJETIVO :

Ejercitación al uso de punteros y funciones en C.

NOTA: Validar todos los ingresos de datos del usuario.

1
0
0
1
0
1
0
0
0
1
0
1
0

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1

Ejercicio de Repaso: Gestión de Inventario con Estructuras ●●●

Escribe un programa en C que permita gestionar un inventario de productos. El programa debe:

1. Definir una estructura para representar un producto (nombre, código, precio, cantidad).

2. Declarar un vector de productos.

3. Implementar funciones para:

a. Agregar un nuevo producto.

b. Mostrar todos los productos.

c. Buscar un producto por código.

4. Guardar y cargar el inventario desde un archivo.

5. Utilizar un menú para que el usuario pueda seleccionar las distintas opciones.

OBJETIVO :

Uso de estructuras y archivos para gestionar un inventario simple.

NOTA: Validar todos los ingresos de datos del usuario.

1
0
0
1
0
1
0
1
0
0
1
0
1
0
1
0

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1