

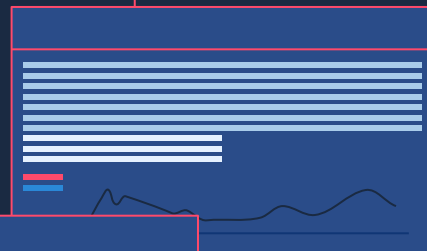


Tecnicatura en Programación Universitaria

Comisión

B

Clase 01



Clase Practica

Martes 09 de Abril 2024

Ing. Oviedo Codigoni Carlos Nicolas



Equipo de Cátedra

01

Teoría

Ing. De la Puente Matias

02

Practica Comisión A

Ing. Radlovacki Yhibo

03

Practica Comisión B

Ing. Oviedo Carlos

Horarios de Clases

01

Teoría

Comisión A: Jueves 18hs a 21hs

Comisión B: Lunes 18hs a 21hs

02

Practica Comisión A

Martes 19hs a 22hs

03

Practica Comisión B

Martes 19hs a 22hs

Condiciones para:

REGULARIZACIÓN

01

Asistencia 70%

02

Aprobar exámenes con una nota mínima de 6

03

Aprobar trabajos prácticos con una nota mínima de 6

PROMOCIÓN

01

Asistencia 80%

02

Aprobar exámenes con una nota mínima de 8

03

Aprobar trabajos prácticos con una nota mínima de 8

Parciales & Recuperatorios

Serán evaluaciones teóricas presenciales tipo cuestionario entre 15 a 20 preguntas de opciones múltiples entre otras a realizarse en hora y media.



Trabajos Prácticos



Ejercicios

En cada trabajo práctico habrá que hacer la resolución de 10 Ejercicios de forma **individual** que estarán en la GAP (Guía de Trabajo Práctico)



Tiempo Extra

Pasado el plazo de entrega se dispondrá de un plazo extra de 12hs para entregar el trabajo práctico



Tiempo de Entrega

Se deberán entregar los trabajos prácticos en el transcurso de los 5 días aproximadamente



Entregas

Todos los Trabajos Prácticos solo se podrán entregar en el campus virtual de la materia Programación I

!! Fechas Importantes !!

Fecha	Objetivo	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
28/04	Entrega Primer Trabajo Práctico				28								
13/05	Primer Parcial – Comisión B					13							
16/05	Primer Parcial – Comisión A					16							
26/05	Entrega Segundo Trabajo Práctico					26							
16/06	Entrega Tercer Trabajo Práctico						16						
24/06	Segundo Parcial – Comisión B						24						

!! Fechas Importantes !!

Fecha	Objetivo	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
27/06	Segundo Parcial – Comisión A						27						
01/07	Recuperatorio – Comisión B							01					
04/07	Recuperatorio – Comisión A							04					

Links Importantes

Campus virtual

01



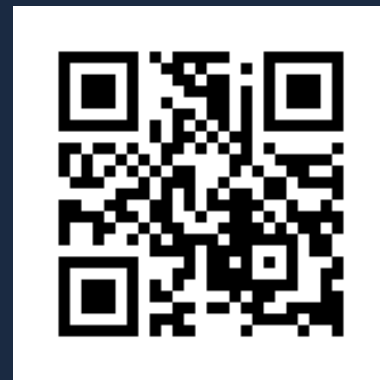
Autogestión

02



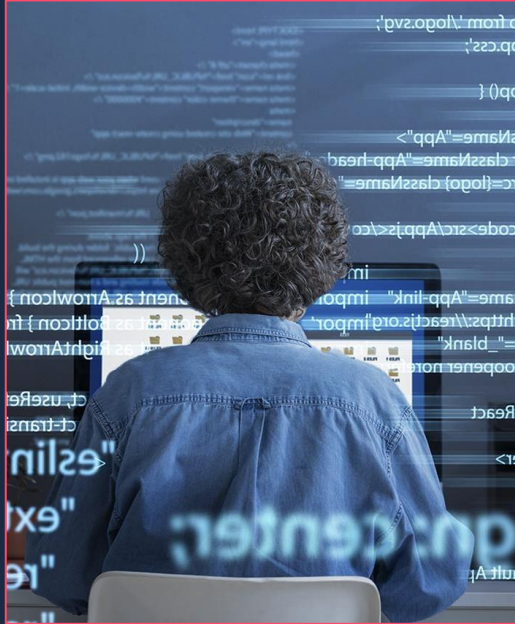
Materia

03

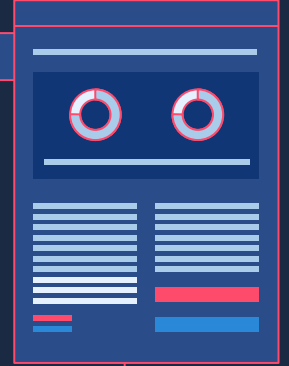


Discord

<https://discord.gg/uBxRwWDuGn>



Programación en C

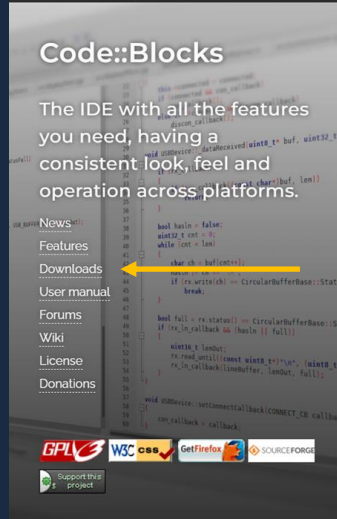


Software



CodeBlocks

<https://www.codeblocks.org/>



[Code::Blocks / Downloads](#)

Downloads

There are different ways to download and install Code::Blocks on your computer:

• [Download the binary release](#)

This is the easy way for installing Code::Blocks. Download the setup file, run it on your computer and Code::Blocks will be installed, ready for you to work with it. Can't get any easier than that!

• [Download a nightly build](#)

There are also more recent so-called nightly builds available in the [forums](#). Please note that we consider nightly builds to be stable, usually, unless stated otherwise.

• Other distributions usually follow provided by the community (big "Thank you!" for that!). If you want to provide some, make sure to announce in the forums such that we can put it on the official C.B homepage.

• [Download the source code](#)

If you feel comfortable building applications from source, then this is the recommend way to download Code::Blocks. Downloading the source code and building it yourself puts you in great control and also makes it easier for you to update to newer versions or, even better, create patches for bugs you may find and contributing them back to the community so everyone benefits.

Code::Blocks es un entorno de desarrollo integrado de código abierto, que soporta múltiples compiladores, que incluye GCC, Clang y Visual C++. Se desarrolló en C++ usando wxWidgets como el kit de herramientas GUI. Utilizando una arquitectura de complemento, sus capacidades y características están definidas por los complementos proporcionados. A la fecha octubre de 2017, Code::Blocks está orientado hacia C, C++ y Fortran. Tiene un sistema de compilación personalizado y un soporte de construcción opcional.

¡Hola Mundo!

Lo Clave :

- *Librerías*
- *Función Principal*
- *Punto y Coma*
- *Función printf()*

Ejercicio:

Realizar un Programa que muestre en pantalla tu nombre y apellido

Primer Programa en Lenguaje C

```
1  // Librerías
2  #include <stdio.h>
3  // Funcion Principal
4  void main(void)
5  {
6      // Imprimir "Hola Mundo" en la consola
7      printf("Hola Mundo!\n");
8  }
```

- **#include <stdio.h>:**
Esta línea incluye la librería estándar de entrada y salida de C, que contiene la función printf() utilizada para imprimir texto en la consola.
- **int main(void):**
La función principal main() contiene el código principal
- **printf("Hola Mundo!\n");**
Esta línea imprime "Hola Mundo" seguido de un salto de línea en la consola.

Conociendo a la función Printf

Caracteres Especiales por Naturaleza

\	escapar el siguiente caracter
\\	imprimir una barra invertida
"	comienzo o fin de cadena
\"	imprimir comillas dobles
'	comienzo o fin de una constante de caracter
\'	imprimir comilla simple
%	comenzar especificación de formato
\\%	imprimir un signo de porcentaje

Porcentaje & Formatos

%c	imprimir un carácter
%d	un número decimal (base 10)
%e	un N.º de punto flotante exponencial
%f	un N.º de punto flotante
%g	un N.º de punto flotante en formato general
%i	un entero en base 10
%o	un N.º en octal (base 8)
%s	una cadena de caracteres
%u	un N.º decimal no signado (base 10)
%x	un N.º en hexadecimal (base 16)
%%	un signo porcentaje (también funciona \\%)

Caracteres especiales alternativos

\a	alerta auditiva (campana -bell-)
\b	retroceso
\f	form feed
\n	nueva línea
\r	retorno de carro
\t	tabulación
\v	tabulación vertical

Ejemplos

printf	imprime
("%d",0)	0
("%d",-7)	-7
("%d",1560133635)	1560133635
("%d",-2035065302)	-2035065302