

# Clase #02 de 29

## Análisis de “Hello, World!” & Flujos, Archivos y Bytes

*Abril 16, Miércoles*

# Agenda para esta clase

- Trabajo #0
- Análisis de hello.cpp
- Flujos, Archivos y Bytes

# Progreso de Trabajo #0

# Progreso de Trabajo #0

## "Hello, World!" en C++

---

### 4.1. Objetivos

- Demostrar capacidad para editar, compilar, y ejecutar programas C mediante el desarrollo de un programa simple. C++.
- Tener un primer contacto con las herramientas necesarias para abordar la resolución de los trabajos posteriores.
- Creación de repositorio personal git.
- Armado de equipo de trabajo.

### 4.2. Temas

- Sistema de control de versiones.
- Lenguaje de programación C++.
- Proceso de compilación.
- Pruebas.

### 4.3. Problema

Adquirir y preparar los recursos necesarios para resolver los trabajos del curso.

- git
- readme.md del repositorio
- readme.md de la carpeta del trabajo
- Compilador seleccionado
- Versión del lenguaje utilizada: -std=c++23
- .gitignore de ejecutables
  - Microsoft
    - \*.exe
  - Unix
    - \*
    - !/\*\*/
    - !\*. \*
- Envío por la salida estándar y redireccionamiento, sin modificar el programa fuente
- Equipo de trabajo

# Análisis de Hello.cpp

# Análisis de hello.cpp

```
/* Hello world
 * JMS
 * 20130411
 */

#include <iostream>

int main() {
    std::cout << "Hello, world!\n";
}
```



- Modelo Consola
  - Flujo de datos
    - De Entrada
      - Origen: Teclado
    - De Salida
      - Destino: Pantalla
  - Redireccionamiento de la salida con >
  - Procesamiento de renglones
- Análisis desde el interior al exterior
  - Cadena ó String
    - \n
    - endl
  - Operador de envío o inserción
  - Objeto en variable cout
    - Namespace std
    - Scope
    - Carpeta
    - “¡Pero yo quiero usar using namespace std!”
      - Implicancias
      - Using std::cout
  - Función main
    - ¿Quien la invoca?
    - Valor de retorno por defecto
  - #include
    - Header
    - La Biblioteca Estándar

# Intervalo

15 minutos



# Sobre la Cátedra y la Asignatura



# La Cátedra

- Dr. Mg. Oscar Bruno, Profesor Asociado Ordinario
  - Coordinador de la Cátedra
  - **@orbruno**
  - **droscarbruno.wordpress.com**

# Análisis del Nombre de la Asignatura – Algoritmos y Estructuras de Datos

- Algoritmos
  - Pasos para resolver un problema general
    - Datos de entrada
    - Lista de pasos
    - Salida resultante
- Datos
  - Valores de un conjunto
  - *Datum*
  - Dato versus Información versus Conocimiento
- Estructura
  - Relaciones entre entidades para almacenar datos
- Tipo de Datos

# Programa Sintético



Ministerio de Educación  
Universidad Tecnológica Nacional  
Rectorado

REGISTRADO

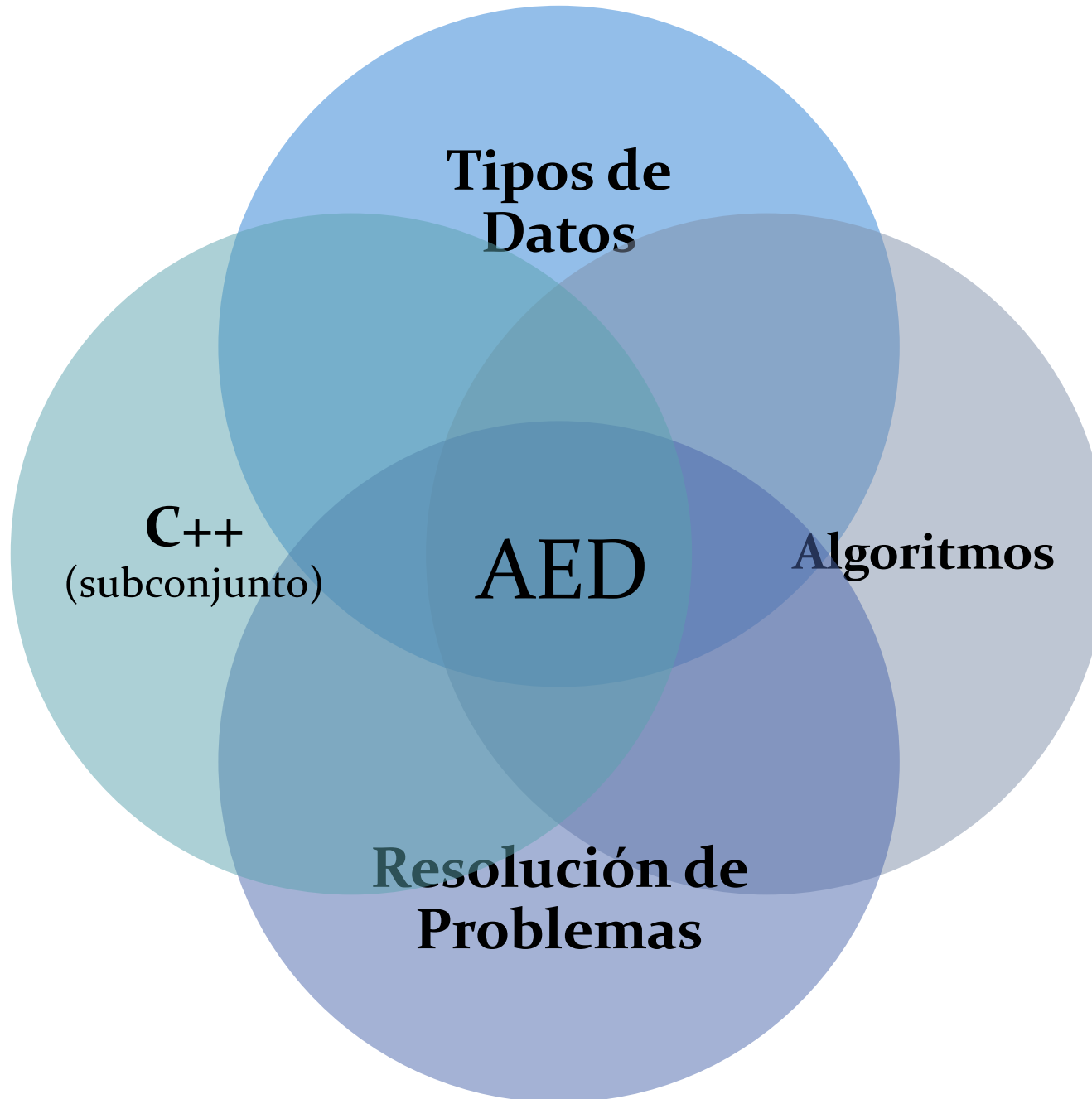
PABLO A. HUEL  
JEFE DE DEPARTAMENTO  
APOYO AL CONSEJO SUPERIOR

"Las Malvinas son argentinas"

- Ordenanza 1878: Diseño Curricular de Ingeniería en Sistemas de Información -Plan 2023-
- CE1.3: Especificar, proyectar y desarrollar software para la elaboración de soluciones informáticas con el propósito de resolver problemas estratégicos y operativos, así como de servicios y de negocios, en el marco de una actividad económica que sea social y ambientalmente sustentable
- RTF: Reconocimiento de créditos o Trayectos Formativos
- Ordenanza 1877 :Régimen de Correlatividades, Equivalencias Para Materias de la Carrera Ingeniería en Sistemas de Información -Plan 2023-
  - AED Cursada para cursar y rendir
    - Para Sintaxis y Semántica de los Lenguajes
    - Paradigmas de Programación
    - Análisis de Sistemas de Información (integradora)
  - AED Cursada para cursar y rendir
    - Base de Datos
    - Desarrollo de Software
    - Diseño de Sistemas de Información (integradora)

Carrera:	Ingeniería en Sistemas de Información	N° de orden:	6
Asignatura:	ALGORITMOS Y ESTRUCTURAS DE DATOS	Horas cátedra semanales:	5
Departamento:	Ingeniería en Sistemas de Información	Horas reloj total:	120
Bloque:	Tecnologías Básicas	Nivel:	1
Área:	Desarrollo de Software	RTF	10
Competencias	Específicas		
	CE1.3		
Objetivos			
<ul style="list-style-type: none"><li>- Identificar problemas algorítmicos.</li><li>- Comprender el proceso de desarrollo de software.</li><li>- Resolver problemas aplicando soluciones algorítmicas y estructuras de datos.</li></ul>			
Contenidos mínimos			
<ul style="list-style-type: none"><li>- Programación Imperativa y Concepto de algoritmo.</li><li>- Concepto de Dato.</li><li>- Tipos de Datos Simples.</li><li>- Tipo Abstracto de datos.</li><li>- Estructuras de Control Básicas.</li><li>- Estrategias de Resolución de problemas.</li><li>- Estructuras de Datos.</li><li>- Abstracciones con procedimientos y funciones.</li><li>- Estructuras de Datos lineales y no lineales.</li><li>- Algoritmos de Búsqueda, Recorrido y Ordenamiento.</li><li>- Archivos de Acceso Secuencial y Aleatorio.</li><li>- Recursividad.</li></ul>			

# Los Temas de AED



# Plataforma del Curso

Herramientas para facilitar la cursada

# Plataforma del Curso

- Tipo de Clases
  - **Sincrónica**, por videoconferencias, con participación y conversaciones sobre los temas
- Herramientas
  - **Aulas Virtuales:**  
Centralizador de actividades, anuncios, y consultas
  - **josemariasola.wordpress.com:**  
Homepage de la asignatura y del curso, material y actividades asincrónicas
  - **Google Calendar:** Caledario de clases, actividades y deadlines
  - **Google Meet:** Clases sincrónicas a distancia
  - **YouTube:** Quizás algún video
  - **GitHub:** Resoluciones Individuales y del cada equipo

# Bibliografía

- Obligatoria
  - <https://droscarbruno.wordpress.com/algoritmos-desarrollo-de-temas/>
- Recomendada
  - Sobre Programación
    - [Avanzado] Kernighan, B; Pike, R. *La práctica de la programación*. 2000. Pearson: México
    - [Inicial] García Molina, J; Montoya Dato, F; otros. *Una introducción a la programación: un enfoque algorítmico*. 2005. Thomson: España
  - Sobre el Lenguaje C++
    - [Avanzado] Stroustrup, B. *El Lenguaje de Programación C++*. 2013. Addison Wesley
    - [Inicial] Deitel, H; Deitel, E. *Cómo programar en C/C++*. 1994. Prentice Hall: México
  - Otros artículos a lo largo del curso, publicados desde la página del curso.



# josemariasola.wordpress.com

- Homepage del Curso
- Papers
- Referencia
- Enunciados de Trabajos
- Equipos y Notas
- Calendario de clases, exámenes, deadlines, actividades
- Algunas Slides de Clases
- Algunos Videos de Clases.

# Git y GitHub

## <https://github.com/orgs/utn-frba-aed>

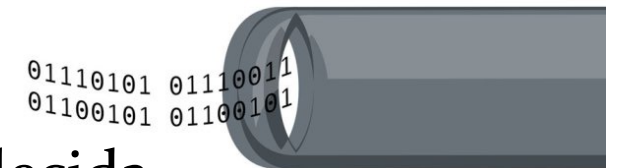
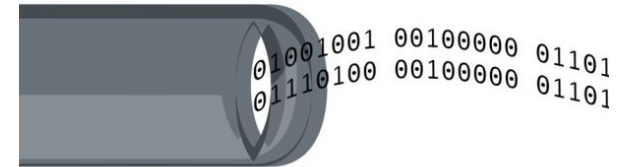
- Sistema de Control de Versiones (VCS)
  - Gestión de cambios sobre producto o su configuración
  - Ejemplos: Git, Mercurial, y Subversion
- Git
  - VCS distribuido, 2005 por Linus Torvalds para el kernel de Linux
- GitHub
  - Servicio que implementa Git
  - Otros similares: Bitbucket, GitLab, Gitorious, CloudForge
- ¿Por qué GitHub en el Curso?
  - Tiene repositorios públicos, y privados para universidades
  - Es, para muchos, el primer contacto con un VCS distribuido, y con lenguajes de marca livianos, como markdown
  - Es más eficiente que trabajos impresos
  - Posee interfaces web, de escritorio, y móvil
- ¿Para qué?
  - Avance de cada estudiante y de cada equipo.
  - Interacción entre alumnos y con la cátedra similar a red social
  - Seguimiento de correcciones y mejoras
- Repositorios durante el curso
  - Primeras clases
    - Repositorio individual
  - Luego, uso de la *Organización utn-frba-aed*
    - Para los miembros del curso asignamos un repositorio privado, dentro **utn-frba-aed**, a cada equipo
    - <https://github.com/orgs/utn-frba-aed/>
- Primeros Pasos en GitHub
  1. Leer guía "Git 101"
  2. Crear Usuario
  3. Crear Repositorio.

- <https://josemariasola.wordpress.com/aed/git/>

# Flujos, Archivos, y Bytes

# Salida Estándar, cout y Redirección

- El objeto cout es un destino, es la salida estándar
- Ese destino es un flujo por donde viajan (salen) datos
- El Sistema Operativo enlaza la salida estándar a
  - La pantalla
  - Una consola o terminal
  - Otro proceso
  - Un archivo
  - Una impresora
  - Otro dispositivo
  - O lo que el usuario del Sistema Operativos decida
- El programa sabe que envía datos por cout por un extremo, pero no sabe a qué está conectado al otro extremo.



# Archivo y Byte

- Un **archivo** contiene un **secuencia de bytes**
- La **interpretación** del significado de la secuencia la dan los **programas**
- Los programas se **conectan** los archivos mediante **flujos**
- Las **extensiones** de archivos son solo **convenciones**
- Ejemplos para ver con un editor de texto o de bytes
  - hello.cpp
  - hello.exe
  - readme.md
  - hello.jpg



- Un **Byte** es un **número entero** en el **rango de cero a 255**
- Los números se pueden representar en diferentes **notaciones** como números romanos o arábigos y en diferentes **bases**
- Ejemplo: **42**
  - Castellano: cuarenta y dos
  - Francés: quarante-deux
  - Romano: XLII
  - Decimal: 42
  - Hexadecimal: 2A
  - Octal: 52
  - Binario: 101010
  - ASCII: \*
  - No importa la representación, el concepto 42 es universal
- Para representar valores entre cero y 255 se necesitan **ocho dígitos binarios o bits**:  $2^8=256$
- Byte también es un **tipo de dato**: es un **conjunto de valores** y otro de **operaciones sobre esos valores**

# Alfabeto ASCII: Bits

<div> <div> <div>b<sub>7</sub></div> <div>b<sub>6</sub></div> <div>b<sub>5</sub></div> </div> <div> <div>b<sub>4</sub></div> <div>b<sub>3</sub></div> <div>b<sub>2</sub></div> <div>b<sub>1</sub></div> </div> <div> <div>Column</div> <div>Row</div> </div> </div>					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
					0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	
1	1	0	1	13	CR	GS	—	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	—	o	DEL

<https://danq.me/2024/07/21/ascii/>



# Alfabeto ASCII: Sectores

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	00	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	01	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	02	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	03	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	04	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	05	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	06	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	07	[BELL]	39	27	'	71	47	G	103	67	g
8	08	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	09	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	0A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	0B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	0C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	0D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	0E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	0F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

<https://danq.me/2024/07/21/ascii/>



# Visualizaciones: ASCII, Decimal, Hexadecimal

```
$ cat StreamsFilesBytes.cpp
#include <iostream>

int main() {
    std::cout << "ABCDEFGH\n0123456\n !\"#$%\a\n";
}

$ c++ StreamsFilesBytes.cpp
$ ./a.out
ABCDEFGH
0123456
!\"#$%

$ ./a.out > output.txt
$ cat output.txt
ABCDEFGH
0123456
!\"#$%

$ hexdump -v -e '8/1 "%3_c "' -e '"\n"' output.txt
A  B  C  D  E  F  G  \n
0  1  2  3  4  5  6  \n
!  "  #  $  %  \a  \n

$ hexdump -v -e '8/1 "%03u "' -e '"\n"' output.txt
065 066 067 068 069 070 071 010
048 049 050 051 052 053 054 010
032 033 034 035 036 037 007 010

$ hexdump -v -e '8/1 "%02x "' -e '"\n"' output.txt
41 42 43 44 45 46 47 0a
30 31 32 33 34 35 36 0a
20 21 22 23 24 25 07 0a

$ hexdump -C output.txt
00000000 41 42 43 44 45 46 47 0a 30 31 32 33 34 35 36 0a |ABCDEFGH.0123456.|
00000010 20 21 22 23 24 25 07 0a |!\"#$%..|
```

Propósito	Unix	Windows
Representación carácter	<code>cat filename</code>	<code>type filename</code>
Redirección	<code>comando &gt; destino</code>	
Representación numérica	<code>hexdump -C filename</code>	<code>debug filename</code> <code>certutil -encodehex filename stdout</code>

# Términos de la clase #02

## Definir cada término con la bibliografía

- Progreso de Trabajo #0
  - readme.md
  - .gitignore
- Análisis de Hello.cpp
  - Consola
  - Flujos de datos estándar
  - Flujos de carácter
  - Redireccionamiento de la salida con >
  - Procesamiento de renglones
  - Cadena ó String
  - \n
  - endl
  - Operador de envío o inserción
  - Objeto en variable cout
  - Namespaces
  - Scope
  - Using
  - Función main
  - Invocación de main
  - Valor de retorno
  - #include
  - Header
- La Biblioteca Estándar
- Sobre la Asignatura
- Plataforma del Curso
- Flujos, Archivos, y Bytes
  - cout
  - Destinos y orígenes
  - Sistema operativo y redirección
  - Abstracción del programa
  - Archivo
  - Extensiones como convención
  - Byte
  - Notaciones y Bases
  - Bit
  - Tipo de dato
  - Tipo de dato Byte
  - Código ASCII
  - Comandos para visualizar bytes como texto ASCII, hexadecimal, y decimal



# Tareas para la próxima clase

1. Investigar sobre assert
2. Investigar sobre tipos de datos.

**¿Consultas?**

**Fin de la clase**