

# Algoritmos y Estructuras de Datos

## AED

*K1051*

*2021*

*Esp. Ing. José María Sola, Profesor*

Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires

Departamento de Ingeniería en Sistemas de Información

# **Clase #01 de 29**

# **Presentación de la Asignatura**

# **y Pautas Generales**

*Abr 8, Jueves*

# Agenda para esta clase

- Bienvenidos a la Universidad y al Curso
- Presentación inicial, conozcámonos
- El contexto de AED
- Intervalo
- La aprobación y la regularización
- Primer contacto con la programación
- Trabajo #0

# Presentación Inicial

Conozcámonos

# Docentes del Curso

- Profesor a Cargo
  - Esp. Ing. José María Sola
- Auxiliares
  - Ing. Joel Edgardo Peralta

# (Distancia) En la Clase y Horarios

- Micrófono “muteado”
- Horarios del curso
  - Jueves 7:00pm
  - Acceso por *Hangouts Meet* desde *Calendar*, si no hay link, no hay clase sincrónica
- Consultas antes y después de clase

# El Contexto de AED

# Análisis del Título de la Carrera – Ingeniería en Sistemas de Información

- Ingeniería
  - Aplicación tecnología—ciencia y técnicas—para resolución problemas
  - Construcción de soluciones
- Sistema
  - Conjunto de elementos relacionados con objetivo común
- Información
  - Datos procesados
  - Materia prima para toma de decisiones
  - Dato valor sintáctico
  - Información valor semántico.
- Sistema de Información
  - Sistema manual o automático con
    - personas, máquinas o métodos que
    - procesa información
      - recolecta, transmite, almacena distribuye, presenta y manipula
      - información para sus usuarios
      - en tiempo y forma
  - En general, los sistemas de información tienen grandes partes implementadas con Sistemas Software
  - La programación y la construcción de Sistemas Software.

# Rol de la Materia en las Actividades del Ingeniero en Sistemas de Información



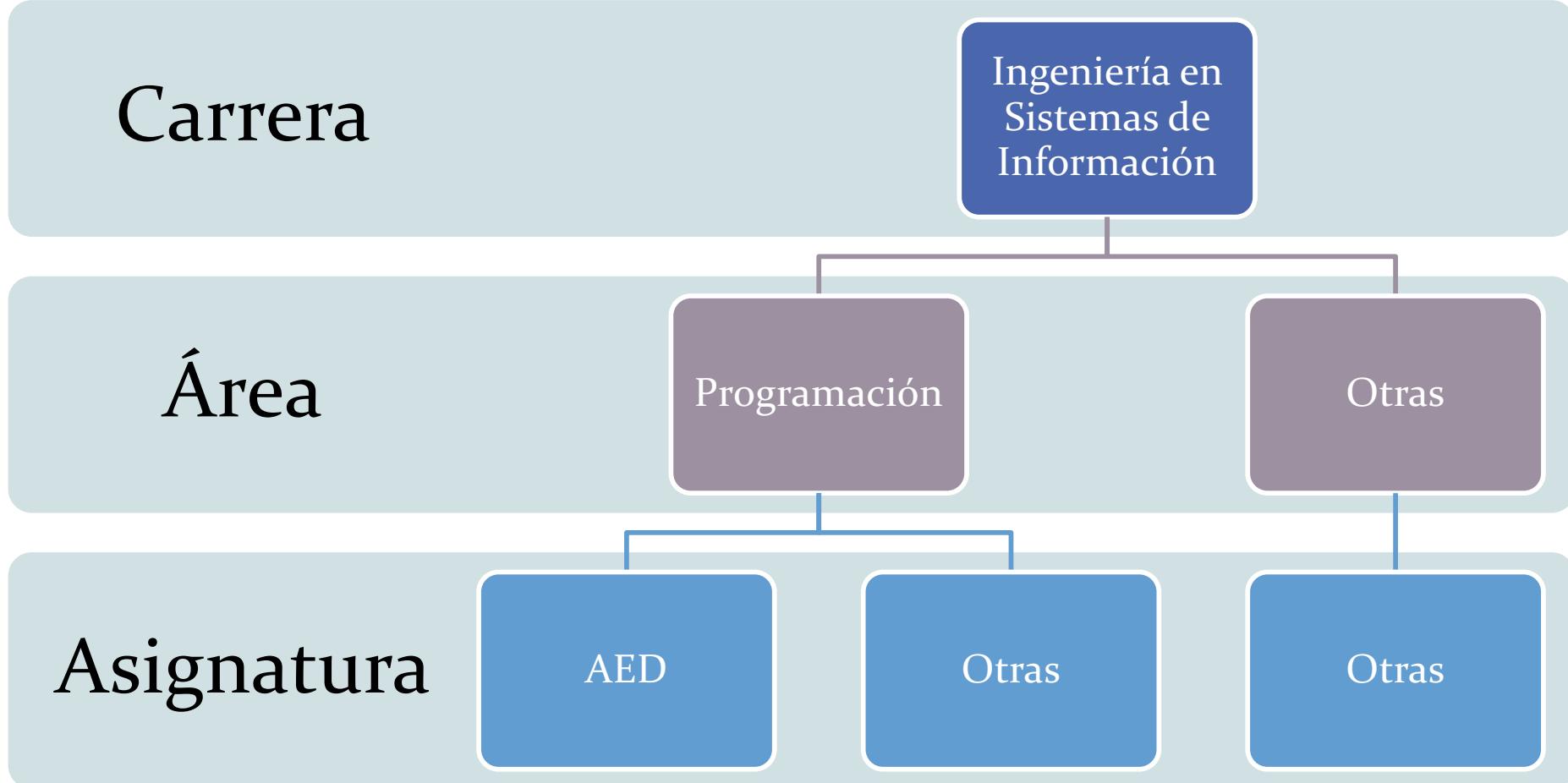
# Las Asignaturas Paralelas

- Matemática Discreta
- Sistemas y Organizaciones

# Matemática Discreta

- Objetivos
  - Aplicar métodos inductivos, deductivos y recursivos en la resolución de situaciones problemáticas y demostraciones matemáticas
  - Comprender los conceptos y procedimientos necesarios para resolver relaciones de recurrencia
  - Aplicar propiedades y funciones definidas en los números enteros y enteros no negativos
  - Caracterizar distintas estructuras algebraicas, enfatizando las que sean finitas y las álgebras de Boole
  - Aplicar propiedades de grafos, dígrafos y árboles en la resolución de situaciones problemáticas
- Contenidos Mínimos
  - Lógica Proporcional Clásica y de Predicados de Primer Orden
  - Teoría de Números
  - Inducción Matemática
  - Relaciones de Recurrencia
  - Estructuras Algebraicas Finitas y Algebra de Boole
  - Grafos, dígrafos y árboles
- Contenidos Extendidos
  - Lenguajes Formales
  - Autómatas Finitos
  - Expresiones Regulares
  - Gramáticas.

# AED en la Carrera



# Área Programación

- Objetivos
  - Formar e informar acerca de metodologías, técnicas y lenguajes de programación, como herramientas básicas para el desarrollo de software y el estudio de disciplinas que permitan crear nuevas tecnologías
- Asignaturas (640 hs)
  - 1. Matemática Discreta (96 hs)
  - 2. Algoritmos y Estructuras de Datos (160 hs)
  - 3. Sintaxis y Semántica de los Lenguajes (128 hs)
  - 4. Paradigmas de Programación (128 hs)
  - 5. Gestión de Datos (128 hs).

# Integración con otras Asignaturas y Conceptos Principales

- Integración Horizontal: Paralelas
  - Área de Programación
    - Matemática Discreta
  - Computación
    - Arquitectura de Computadoras
  - Sistemas de Información
    - Sistemas y Organización
- Integración vertical: Posteriores
  - Área de Programación
    - Sintaxis y Semántica de Lenguajes
    - Paradigmas de Programación
  - Computación
    - Sistemas Operativos
  - Sistemas de Información
    - Análisis de Sistemas
- Abstracción: concepto fundamental
  - Separación, dejar de lado los detalles para enfocar en lo importante
- Tipo de Dato:
  - Conjunto de Valores y conjunto de operaciones sobre ese conjunto de valores
- Orientación a Objetos
  - Objeto: entidad con comportamiento y que mantiene un estado.

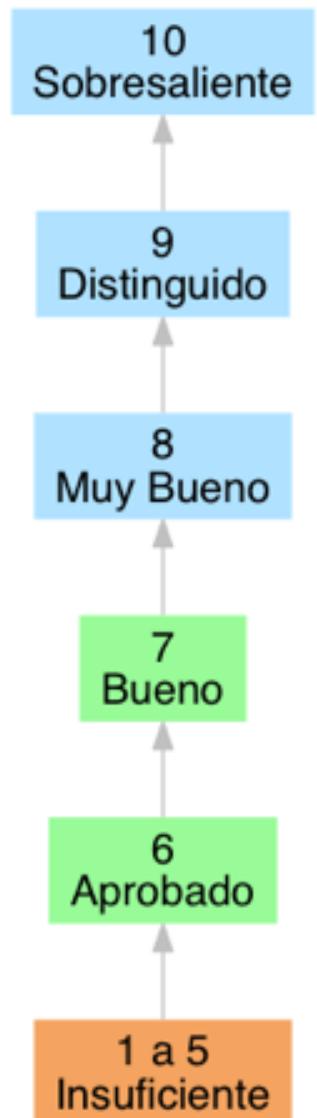
# La Aprobación y la Regularización

# Dinámica de Trabajo

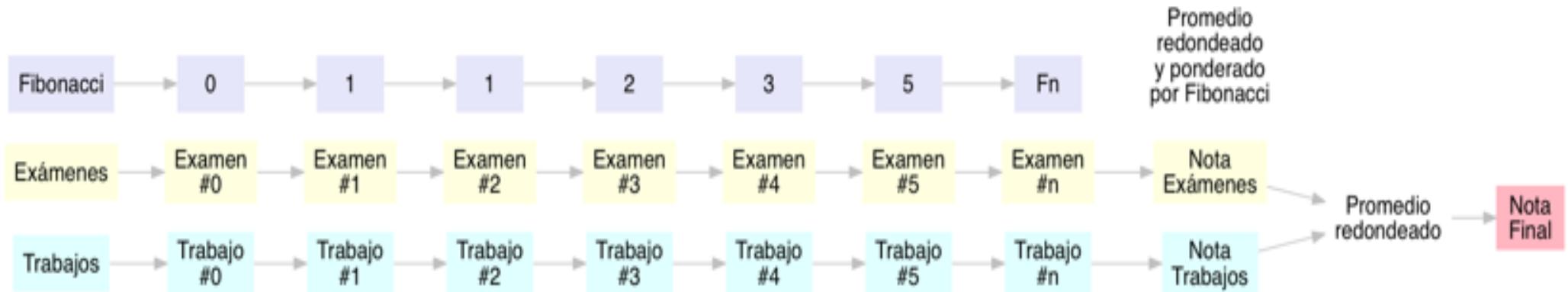
- Indicación de lecturas y ejercicios obligatorios para siguiente clase
- Profundización de conceptos en clase
- Clase interactiva y participativa
  - Se recomienda seguir la clase con la bibliografía y tomar apuntes a la par
- Ejercicios de aplicación en clase
- Consultas
  - Durante clase, y justo antes de iniciar y finalizar la clase
  - En otros horarios, a través del Foro.

# Aprobación – Dos Formas

- Requisito para aprobación
  - Regularización y aprobación del examen final  
ó
  - Aprobación Directa
- Examen final
  - Cuatro oportunidades



# Requisitos para la Regularización (i.e., Firma)



- Bedelía: **75% de Asistencia**
- Cátedra: Evaluaciones
  - Frecuentes, Grupales ó individuales, de aplicación y conceptuales
  - **Dos conjuntos de Evaluaciones**
    - **#1 Trabajos:** no presenciales
    - **#2 Exámenes:** presenciales
    - Fechas establecen durante el curso
    - **Cada conjunto tiene su nota**
    - **Dos recuperatorios por evaluación en Dic y Feb**
  - Al Finalizar el curso evaluación individual oral o escrita ("coloquio") basada en trabajos
  - **Nota Final:** promedio entre conjunto **#1 Exámenes** y conjunto **#2 Trabajos**
  - **Para regularizar, ambas notas mayor o igual a 6(seis).**

# Niveles de Competencia

Nivel	Descripción	Calificaciones
No alcanza	No se observa capacidad de entendimiento.	1, 2, 3
En desarrollo	Logra comprender algunos conceptos, pero no todos.	4, 5
Competente	Comprende todos los conceptos.	6, 7
Promovido	Puede explicar los fundamentos detrás de los conceptos.	8, 9
Avanzado	Propone nuevos fundamentos o conceptos.	10

# Aprobación Directa

- No haber sido reincorporado
- Nota Trabajos y Nota Exámenes, **ambas** mayor o igual a 8
- Hay tres instancias de determinación del estado de Aprobación Directa:
  - En Noviembre, antes de finalizar la cursada
  - En Diciembre
  - En Febrero.



# Intervalo

15 minutos

# Primer Contacto con la Programación

Lenguajes y Herramientas de Desarrollo

# ¿Qué es Programar?

- Lenguaje
- Lenguaje de programación
  - Define comportamiento
- Lenguaje ó Código Máquina
  - Lenguaje de Bajo Nivel
  - Binario
    - **0101010101010101010101**
  - Assembler (Mnemónicos)
    - ADD B C
    - ADD A B
- Lenguaje C++
  - Un lenguaje de Alto Nivel
    - $a = a + b + c;$
    - ó
    - $a += b+c;$
- Compilador
  - Traduce, baja de nivel de abstracción
- Notación más abstracta
  - $a \leftarrow a + b + c$

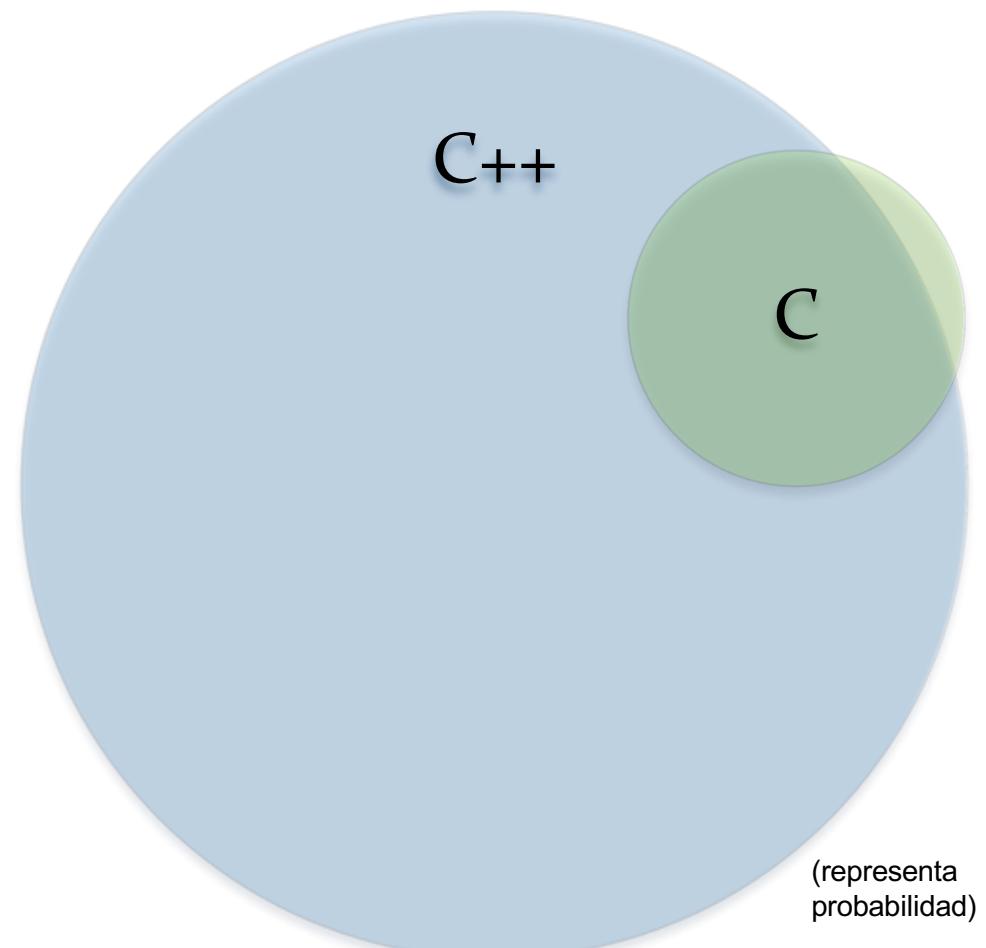


# Sobre los Lenguajes C y C++

## Historia

- 1970's
  - C
  - C With Classes
- 1980's
  - Comienza standard C
  - C++
- 1990's
  - Standard C90
  - Standard C++98
- 2000's
  - Standard C99
  - Standard C++03
- 2010's
  - Standard C11, C18
  - Standard C++11, 14, 17, 20.

**Conjuntos de infinitos  
programas válidos de C++ y C**



# "Hello, World!"

```
/* Hello world  
JMS  
20130411  
*/  
  
#include <iostream>  
  
int main() {  
    std::cout << "Hello, world!\n";  
}
```

- Propósito
- Comentario encabezado
  - Qué
    - Título descriptivo
  - Quién
    - Número de Equipo e integrantes
  - Cuándo
    - Se actualizó por última vez



# Proceso básico para desarrollar programas

1. **Escribir** el programa con un editor de texto (e.g., vi, Notepad, TextPad, Sublime, TextMate, Notepad++, Notepad2). Es convención para los archivos fuente de C++ la extensión sea .cpp (e.g., hello.cpp)
2. **Compilar** el archivo fuente para producir el programa objeto (e.g., c++ hello.c) ...  
... y **Vincular** (link) el programa con las bibliotecas para crear el programa ejecutable; generalmente ocurre junto con el punto anterior.
3. **Ejecutar** el programa (e.g., hello.exe ó ./a.out)
4. ¿Error en 2 ó 3? Volver a 1 y repetir.

# Ejemplo desde línea de comando

## Mac OS X C++20

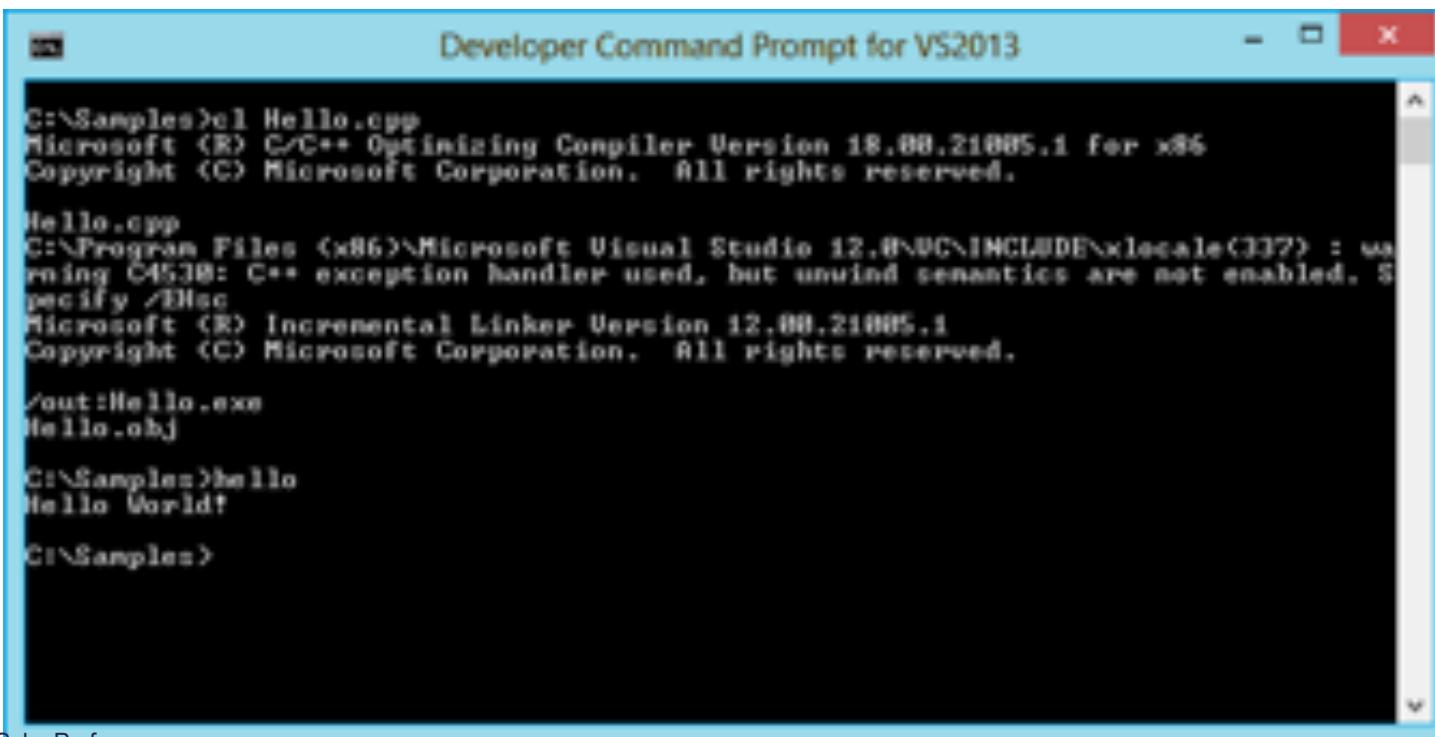
1. Desde la línea de comando
  1. > vi hello.cpp crear el fuente
  2. > c++ hello.cpp -std=c++20 -Weverything crear el ejecutable  
en realidad: Preprocesador → Compilador → Linker
  3. > ./a.out ejecutar  
Hello, World! salida
2. Si hay un error en el paso 2 ó 3, volver al 1 y repetir 2 y 3

```
josemariasola:CppHelloWorld> c++ hello.cpp -std=c++14 -Weverything
josemariasola:CppHelloWorld> ./a.out
Hello, World!
josemariasola:CppHelloWorld>
```

# Ejemplo desde línea de comando

## Compilador Microsoft

1. Desde la línea de comando
  1. > notepad hello.cpp crear el fuente
  2. > cl hello.cpp crear el ejecutable  
en realidad: Preprocesador → Compilador → Linker
  3. > hello.exe ejecutar  
Hello, World! salida
2. Si hay un error en el paso 2 ó 3, volver al 1 y repetir 2 y 3



The screenshot shows a terminal window titled "Developer Command Prompt for VS2013". The command line output is as follows:

```
C:\Samples>cl Hello.cpp
Microsoft (R) C/C++ Optimizing Compiler Version 18.00.21005.1 for x86
Copyright (C) Microsoft Corporation. All rights reserved.

Hello.cpp
C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\INCLUDE\xlocale(337) : warning C4538: C++ exception handler used, but unwind semantics are not enabled. Specify /EHsc
Microsoft (R) Incremental Linker Version 12.00.21005.1
Copyright (C) Microsoft Corporation. All rights reserved.

/out:Hello.exe
Hello.obj

C:\Samples>Hello
Hello World!

C:\Samples>
```

# Ejemplo desde línea de comando

## Compilador Borland

1. Desde la línea de comando
  1. > notepad hello.cpp crear el fuente
  2. > bcc32 hello.cpp crear el ejecutable  
en realidad: Preprocesador → Compilador → Linker
  3. > hello.exe ejecutar  
Hello, World! salida
2. Si hay un error en el paso 2 ó 3, volver al 1 y repetir 2 y 3

The screenshot shows a Windows Command Prompt window titled 'cmd' with the path 'C:\WINDOWS\System32\cmd.exe'. The window displays the following text:  
C:\Program Files\Borland\BCC55\Bin>bcc32 hello.cpp  
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland  
hello.cpp:  
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland  
  
C:\Program Files\Borland\BCC55\Bin>hello  
Hello, World!  
  
C:\Program Files\Borland\BCC55\Bin>

# Herramientas de Desarrollo: Sobre el Compilador y el IDE

- Con IDE (*Integrated Development Environment, Entorno Integrado de Desarrollo*)
  - Ejemplos
    - Apple Xcode
    - Microsoft Visual Studio
    - Eclipse
- Sin IDE
  - Editor
  - Compilador.

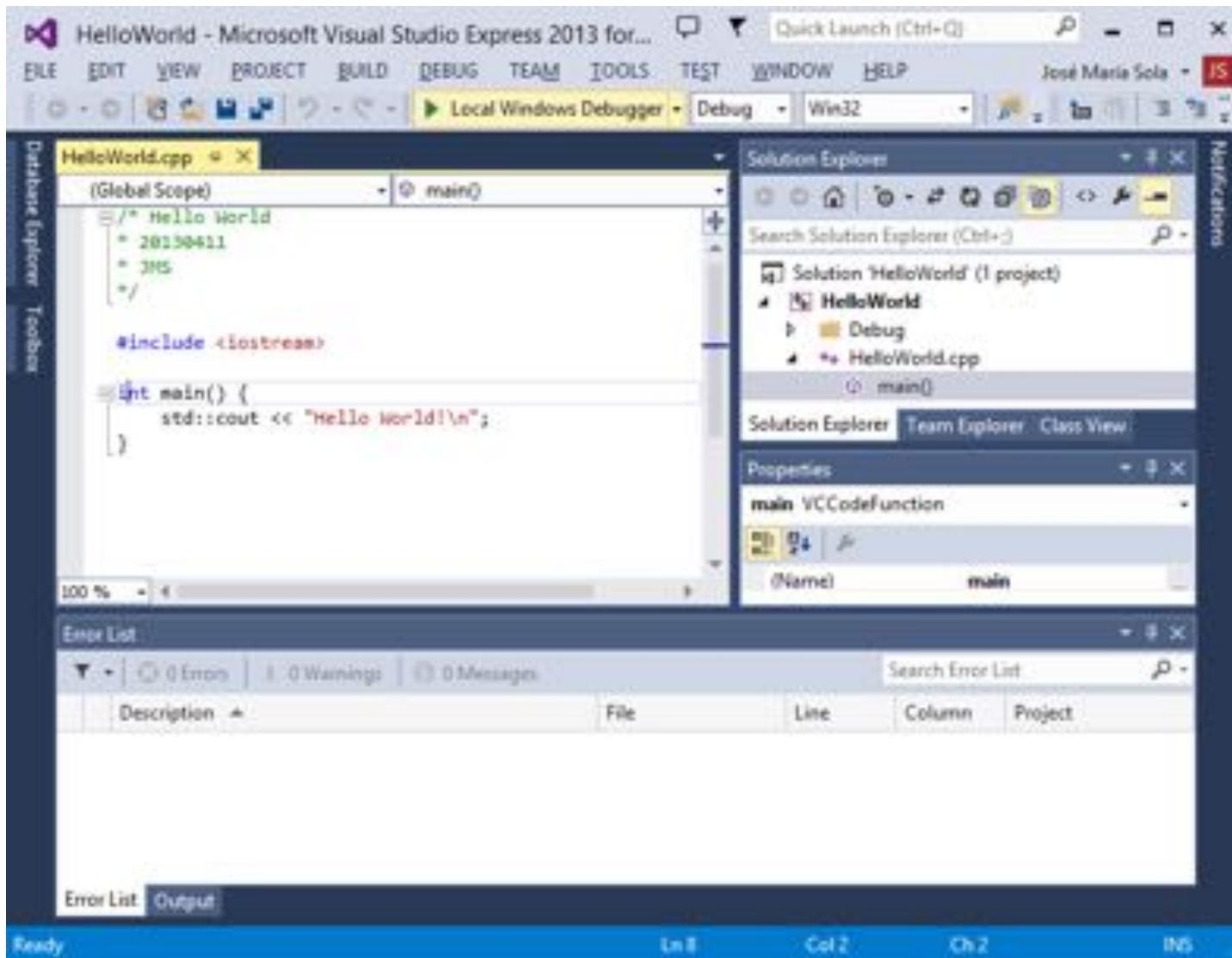
## Con IDE

- Editor
- Depurador
- Gestor de Proyectos y de configuraciones
- Ayuda
- y más...

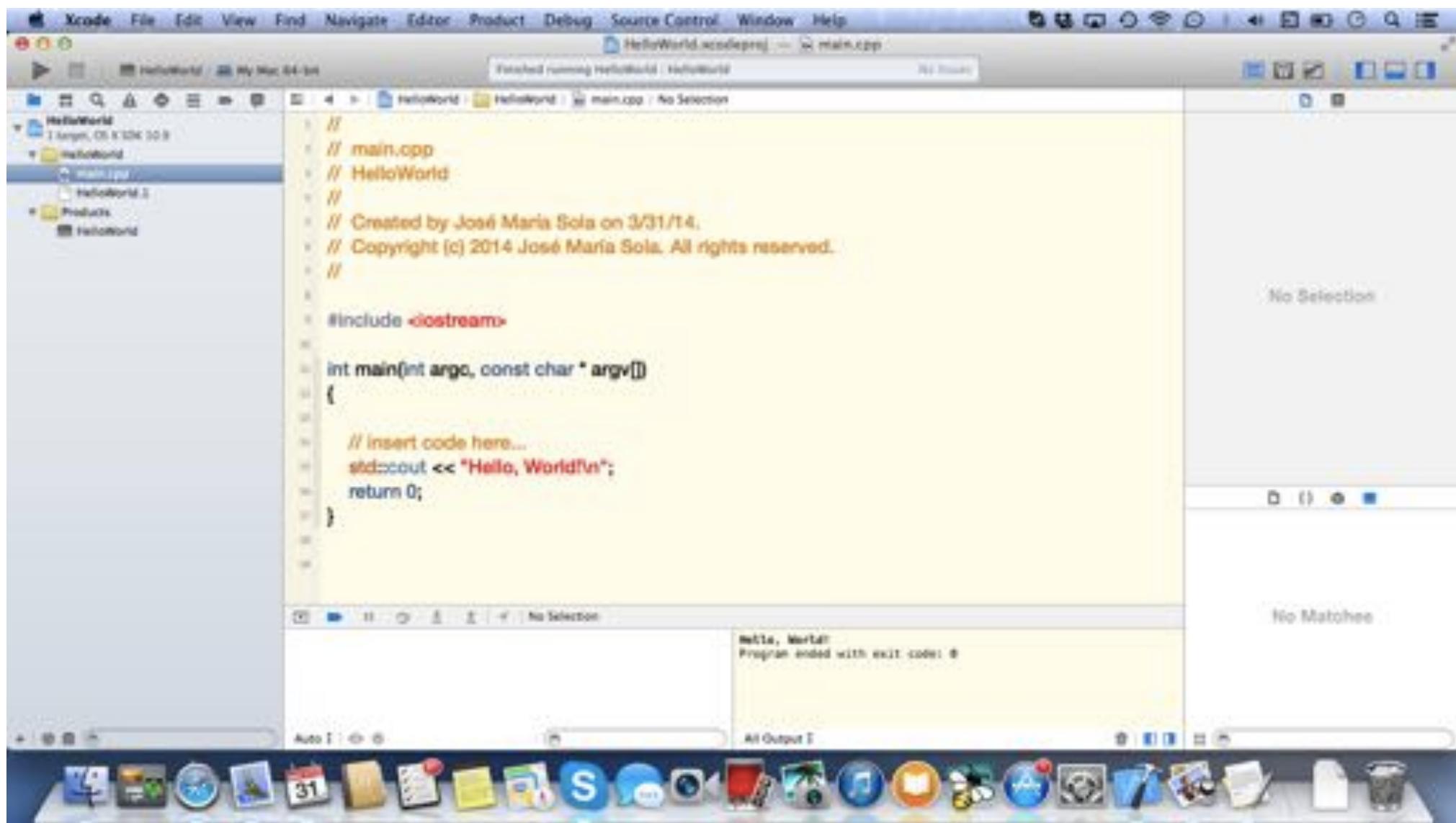
**Sin IDE**  
Requiere editor

Compilador de C/C++

# Ejemplo con IDE Microsoft Visual Studio Express for Windows Desktop



# Ejemplo con IDE Apple XCode



# Ejemplo con IDE

# Microsoft Visual Studio Code

(ejemplo en C, no C++)

The screenshot shows the Microsoft Visual Studio Code interface with a dark theme. The main editor window displays a C program named `hello.c`. The code includes a multi-line comment at the top and a `main` function that prints "Hello, World!!\n". A red dot on the left margin indicates a breakpoint at line 10. The bottom status bar shows the file is in the `master` branch and has 0 changes.

```
/* Hello.cpp
C11
JMS
2015
*/
#include <stdio.h>
int main(void){
    printf("Hello, World!!\n");
}
```

# Make (C++)

- Nuestro objetivo (*goal*) es construir ó hacer (*make*) la versión ejecutable de `hello.cpp`
- Desde la línea de comando podemos lograrlo con el comando `make`, pasándole como argumento el nombre del *goal*, en nuestro caso `hello`
- El comando `make` sabe *makear* un ejecutable a partir de un fuente
- En sistemas *Windows* el análogo es el comando `nmake.exe`, aunque también es posible utilizar el `make.exe` ó `mingw32-make.exe` si instalamos *MinGW*.



```
$ make hello
c++ -std=c++2a -Weverything -pedantic-errors    hello.cpp    -o hello
$ ./hello
Hello, World!
$
```

# Links a Compiladores C/C++

Usar unos de estos compiladores o cualquier otro, siempre y cuando se lo configure para **C++20 (ó 17 ó 14 u 11)**

- Con IDE y Línea de Comandos
  - Microsoft Code
    - <https://code.visualstudio.com/Download>
  - Apple Xcode
    - <https://developer.apple.com/xcode>
  - Replit: IDE On-Line
    - <https://repl.it/>
  - Microsoft Visual Studio Community 2017
    - <https://www.visualstudio.com/vs/features/cplusplus/>
  - CodeLite
    - <https://codelite.org>
  - Eclipse IDE for C/C++ Developers
    - <https://www.eclipse.org/downloads/packages/release/2020-03/r/eclipse-ide-cc-developers-includes-incubating-components>
  - Más antiguos
    - Code::Blocks
      - <http://www.codeblocks.org/downloads/>
    - Dev-C++
      - <http://orwelldevcpp.blogspot.com>
- Sin IDE, solo Línea de Comandos
  - Si tu sistema es un UNIX (macOS, GNU, Linux) es probable que incluya un compilador, probá los comandos **cc** y **gcc** desde la línea de comandos
  - GNU C Compiler (ahora GNU Compiler Collection)
    - <http://gcc.gnu.org/install/binaries.html>
    - Para plataformas Windows
      - <http://www.mingw.org>
      - <http://mingw-w64.org/doku.php>
  - Clang
    - <http://releases.llvm.org/download.html>
  - Embarcadero Free C++ Compiler
    - <https://www.embarcadero.com/free-tools/ccompiler>
  - Más antiguos
    - Borland C++ Compiler version 5.5 Free Download
      - <http://edn.embarcadero.com/article/20633>
        - Using the Borland 5.5 Compiler and command-line tools
      - <http://edn.embarcadero.com/article/20997>
        - Borland C++ 5.5 Free Command-line Tools Supplementary Information
      - <http://edn.embarcadero.com/article/21205>

# Compiladores, Editores y Entornos de Desarrollo: Instalación, Configuración y Prueba

- <https://josemariasola.wordpress.com/papers#CompiladoresInstalacion>
- Introducción a compilador, entornos de desarrollo
- Amar de entorno de desarrollo para C/C++ bajo un entorno Windows, basado en el compilador MinGW y el editor de código fuente Visual Studio Code.

# Trabajo #0

hello.cpp: "Hello, World!"

# Trabajo #0 – "Hello, World!" en C++

- Enunciado en <https://josemariasola.wordpress.com/aed/assignments/>
- Secuencias de Tareas
  - Si no posee una cuenta GitHub, crearla
  - Crear un repositorio público llamado AED
  - Escribir el archivo readme.md que actúa como front page del repositorio personal
  - Crear la carpeta oo-CppHelloWorl
  - Escribir el archivo readme.md que actúa como front page de la resolución
  - Seleccionar, instalar, y configurar un compilador C++11 o mayor
  - Indicar en readme.md el compilador seleccionado.
  - Probar compilador con hello.cpp que envíe a cout la línea Hello, World! o similar
  - Ejecutar el programa, y capturar su salida en un archivo de texto output.txt
  - Publicar en repositorio personal AED \ oo-CppHelloWorld:  
readme.md, hello.cpp, y output.txt
  - Informar el usuario usuario GitHub en la lista indicada en el curso.
- Restricciones
  - La fecha y hora límite de entrega se publica en el calendario
  - La evaluación se hace con lo publicado en GitHub.

# Términos de la clase #01

## Definir cada término con la bibliografía

- Contexto de AED
  - Comportamiento
  - Software
  - Ingeniería
  - Sistema
  - Información
  - Ingeniería en Sistemas de Información
  - Sistema de Información
  - Proyecto
  - Problema (Necesidad)
  - Solución
  - Desarrollo
  - Análisis
  - Diseño
  - Implementación
  - Organización
- Abstracción
- Orientación a Objetos
- Primer Contacto con la Programación
  - Lenguaje de Programación
  - Código máquina
  - Assembler
  - C
  - C++
  - Editor
  - Compilador
  - Línea de comando
  - IDE (Integrated Development Environment, Entorno Integrado de Desarrollo)

# Tareas para la próxima clase

1. Realizar Trabajo #o "Hello, World!".



# ¿Consultas?



# Fin de la clase