

Algoritmos y Estructuras de Datos AED

K1053

2025

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 Inicial

Abr 9, Miércoles

Agenda para esta clase

- Bienvenidos a la Universidad y al Curso
- Presentación inicial, conozcámonos
- El contexto de AED
- 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
- Auxiliar
 - Ing. Juan Manuel Guzmán

Modalidad

- Clases Sincrónicas
 - Presencial Física en Campus
 - Presencial Digital en Meet
 - [Playlist del curso](#) con temas para antes de la clase e intervalo
 - El calendario del curso indica la modalidad
 - <https://josemariasola.wordpress.com/aed/calendar/>
 - Consultas antes y después de clase
- Clases Asincrónicas
 - Excepcionalmente habrá clases donde se indica lectura y trabajo a realizar de manera asincrónica.

Acuerdos durante las Clases

- Presencial Física en Campus
 - Primera fila de banco **libres para el proyector y para otro material** (*si el aula no tiene proyector integrado*)
 - Entrar y salir por puerta trasera (*si el aula tiene*)
 - Celular, en **silencio o vibrador**
- Presencial Digital
 - Micrófono “muteado”
- Para ambas modalidades
 - Levantar la mano o hablar directamente, pero no nos pisemos.

El Contexto de AED

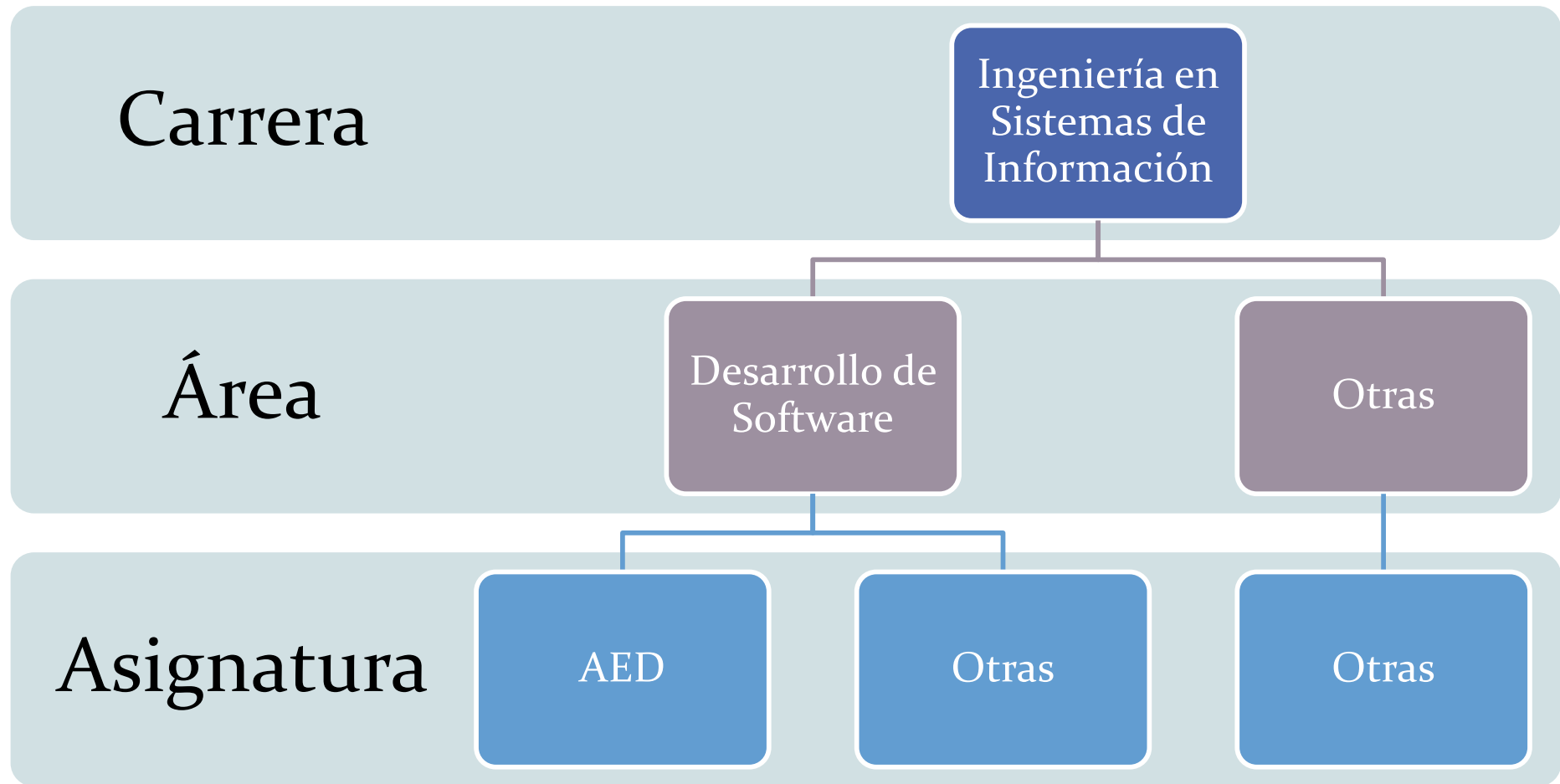
Análisis del Título de la Carrera – Ingeniería en Sistemas de Información

- 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 en las organizaciones
 - Dato valor sintáctico
 - Información valor semántico.
- Sistema de Información
 - Sistema manual y automático con personas, métodos, y tecnología que procesa (crea, recolecta, procesa, transmite, almacena, distribuye, presenta y manipula) información en tiempo y forma para dar soporte a la operación y la toma de decisiones en la organización.
 - En general, los sistemas de información tienen grandes partes implementadas con Sistemas Software
 - La programación y la construcción de Sistemas Software
- Construcción de Sistemas que Permiten a las organizaciones operar.

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

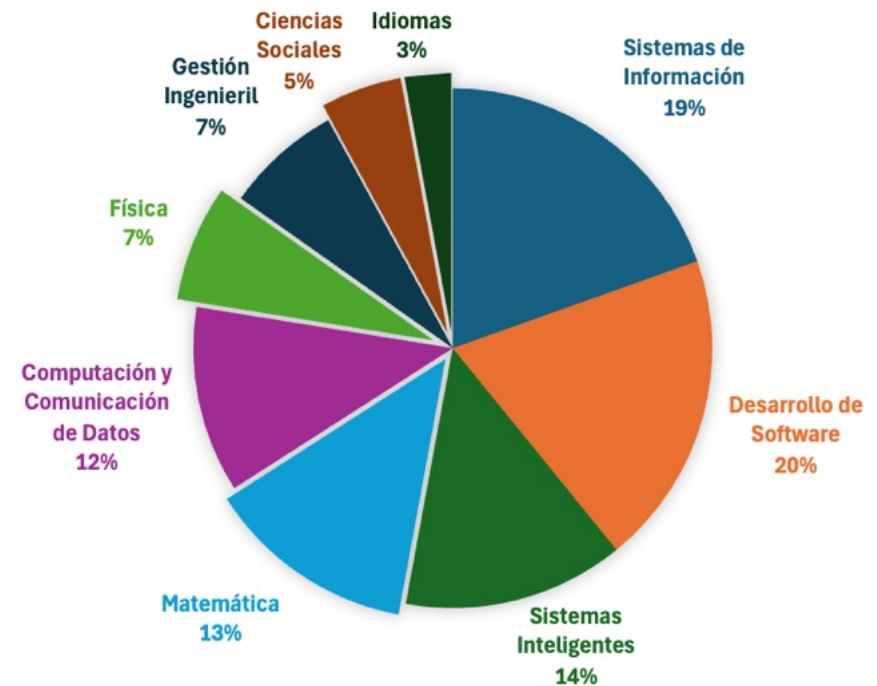


AED en la Carrera



Incidencia de Cada Área




Área de conocimiento	Horas	%
Sistemas de Información	648	20%
Desarrollo de Software	648	20%
Sistemas Inteligentes	456	14%
Matemática	432	13%
Computación y Comunicación de Datos	384	12%
Física	240	7%
Gestión Ingenieril	240	7%
Ciencias Sociales	168	5%
Idiomas	96	3%
Total	3312	100%



- Departamento Básicas (28%)
 - Matemática
 - Física
 - Ciencias Sociales
 - Idiomas
- Departamento Ingeniería Sistemas de Información (72%)
 - Sistemas de Información
 - Desarrollo de Software
 - Sistemas Inteligentes
 - Computación y Comunicación de Datos
 - Gestión Ingenieril
- No incluye (680 h)
 - Electivas (480 h)
 - Tercer nivel: 96 h
 - Segundo nivel: 144 h
 - Quinto nivel 240 h
 - Práctica Profesional Supervisada: 200 h

Relación con Otras Asignaturas

Integración con Otras Asignaturas (AED)

-  Vertical Predecesoras
 - Módulo A
 - Módulo B
-  Horizontal (Paralelas)
 - Lógica y Estructuras Discretas (*ex Matemática Discreta*)
 - Arquitectura de Computadoras
 - Sistemas y Procesos de Negocio (*ex Sistemas y Organizaciones*)
-  Vertical Sucesoras
 - Sintaxis y Semántica de los Lenguajes
 - Paradigmas de Programación
 - Sistemas Operativos
 - Análisis de Sistemas de Información (integradora)
 - Base de Datos (*ex Gestión de Datos*)
 - Desarrollo de Software

Módulo B

- Conjuntos Numéricos
- Ecuaciones e Inecuaciones
- Funciones
- Vectores

Arquitectura de Computadoras

- Objetivos

- Comprender la representación de datos.
- Comprender las estructuras básicas de un computador y su funcionamiento.
- Distinguir la jerarquía de memoria y arquitecturas de microprocesadores.
- Comprender lenguajes de bajo nivel.
- Analizar los recursos computacionales a ser utilizados en el procesamiento, almacenamiento y comunicación de datos.

- Contenidos mínimos

- Sistemas numéricos de distintas bases.
- Operaciones y Conversiones.
- Circuitos lógicos y digitales, códigos y representaciones.
- Algebra de Boole.
- Tecnologías de almacenamiento y dispositivos de entrada y salida.
- Componentes de la arquitectura interna.
- Plataformas de microprocesadores.
- Programación en lenguajes de bajo nivel.

Lógica y Estructuras Discretas

- Objetivos

- Aplicar métodos inductivos, deductivos y recursivos en resolución de situaciones problemáticas.
- Caracterizar estructuras algebraicas y sus propiedades.
- Emplear la teoría de grafos, dígrafos y árboles en resolución de problemas.

- Contenidos Mínimos

- Lógica Simbólica Proposicional y de Predicados de Primer Orden.
- Inducción Matemática.
- Relaciones.
- Estructuras Algebraicas Finitas.
- Teoría de Grafos.
- Teoría de Conjuntos.
- Análisis Combinatorio.

Área Desarrollo de Software

- 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 (648 hs)
 - Lógica y Estructuras Discretas
 - Algoritmos y Estructuras de Datos
 - Sintaxis y Semántica de los Lenguajes
 - Paradigmas de Programación
 - Bases de Datos
 - Desarrollo de Software
 - Ingeniería y Calidad de Software

Integración con otras Asignaturas y Conceptos Principales

- Integración Vertical: Anteriores
 - Módulo B
- Integración Horizontal: Paralelas
 - Área Desarrollo de Software
 - Lógica y Estructuras Discretas
 - Sistemas de Información
 - Sistemas y Procesos de Negocio
 - Computación y Comunicación de Datos
 - Arquitectura de Computadoras
- Integración Vertical: Posteriores
 - Área de Desarrollo de Software
 - Sintaxis y Semántica de los Lenguajes
 - Paradigmas de Programación
 - Computación y Comunicación de Datos
 - 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
- Función
 - Relación entre conjuntos: Existencia y Unicidad
- Orientación a Objetos
 - Objeto: entidad con comportamiento y que mantiene un estado.

La Aprobación y la Regularización

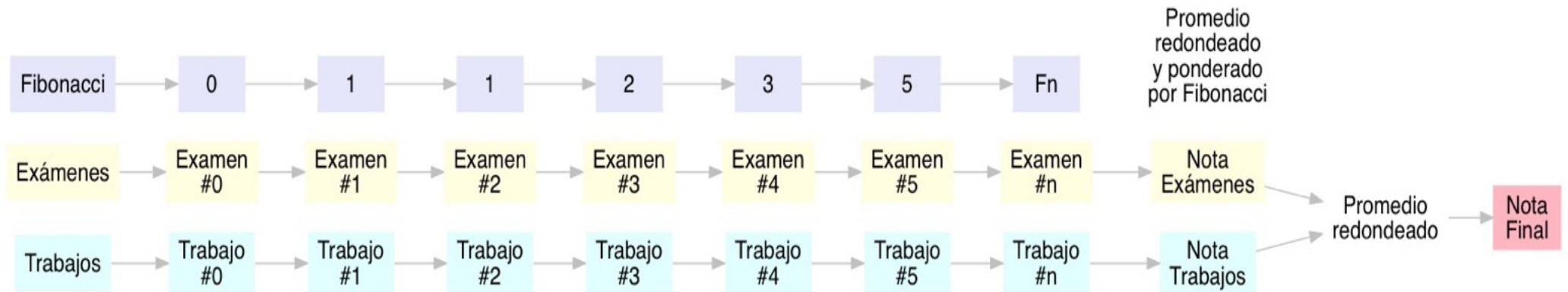
Dinámica de Trabajo

- 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
 - Examen final
 - Correlativas aprobadas (por final o promoción)
 - Algoritmos y Estructuras de Datos
 - Lógica y Estructuras Discretas
 - Cuatro oportunidades
 - ó
 - **Aprobación Directa** (i.e., promoción)

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



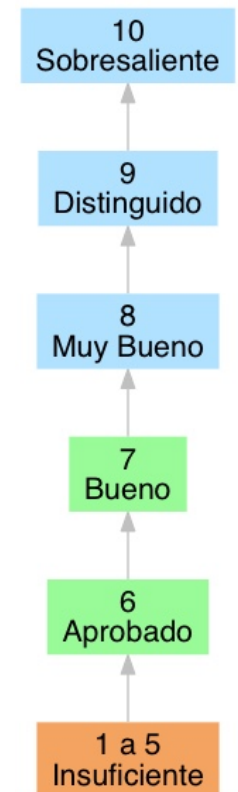
- **Bedelía: 75% de Asistencia**
- **Cátedra: Evaluaciones**
 - Frecuentes, Grupales ó individuales, de aplicación y conceptuales
 - **Dos conjuntos de Evaluaciones**
 - **#1 Trabajos:** no presenciales
 - **#2 Exámenes:** presenciales
 - Fechas establecidas en el calendario
 - **Cada conjunto tiene su nota**
 - **Dos recuperatorios por evaluación en Dic y Feb**
 - Al finalizar el curso **evaluación individual oral ("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).**

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.

Calificaciones según *Niveles de Competencia* y según *Reglamento de Estudio*

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

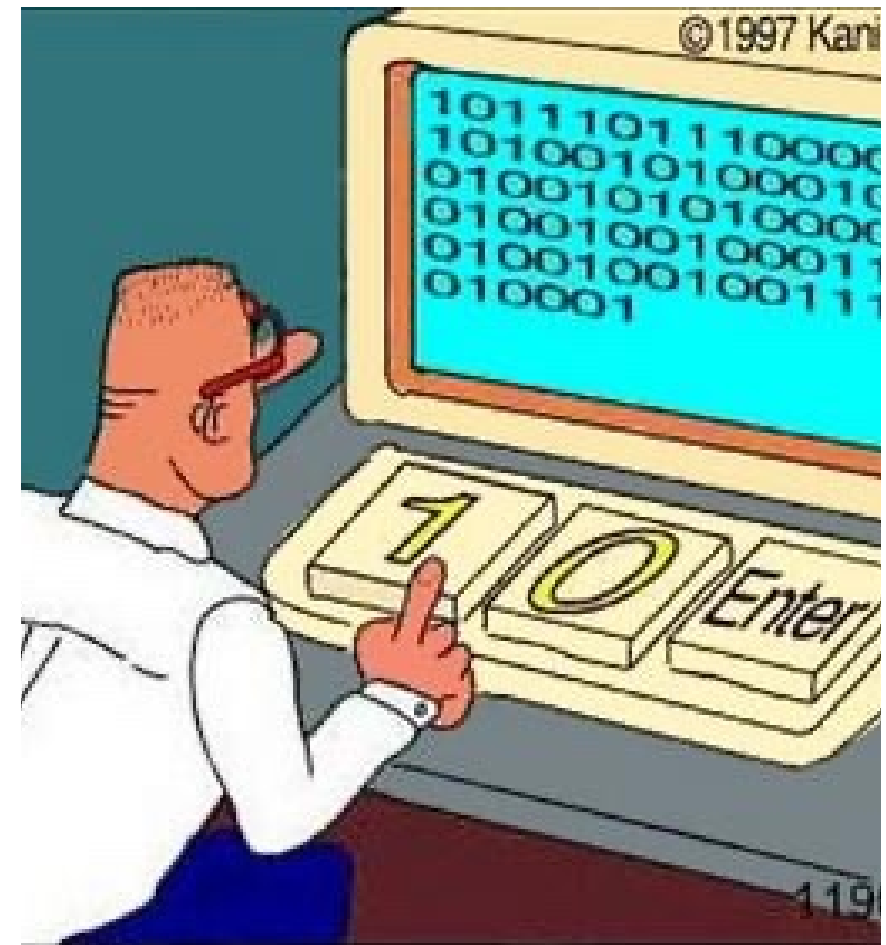


Primer Contacto con la Programación

Lenguajes y Herramientas de Desarrollo

¿Qué es Programar?

- Lenguaje
 - Comunicación
- Lenguaje de programación
 - Define comportamiento
- Lenguaje (ó Código) Máquina (e.g., x86)
 - Lenguaje de Bajo Nivel
 - Opcodes en Binario
 - 1011 0000 0000 0010 0000 00100 0000 0011
 - Opcodes en Decimal
 - 176 2 4 3
 - Opcodes en Hexadecimal
 - B0 02 04 03
 - Mnemónicos en Lenguaje Assembler
 - MOV AL, 2
 - ADD AL, 3
 - Un assembler traduce nemónicos a opcodes, es una traducción casi una a uno.
- Lenguaje C++
 - Un lenguaje de Alto Nivel
 - a = 2;
 - a += 3;
- Compilador
 - Traduce, baja de nivel de abstracción
- Notación más abstracta
 - $a \leftarrow 2$
 - $a \leftarrow a + 3$



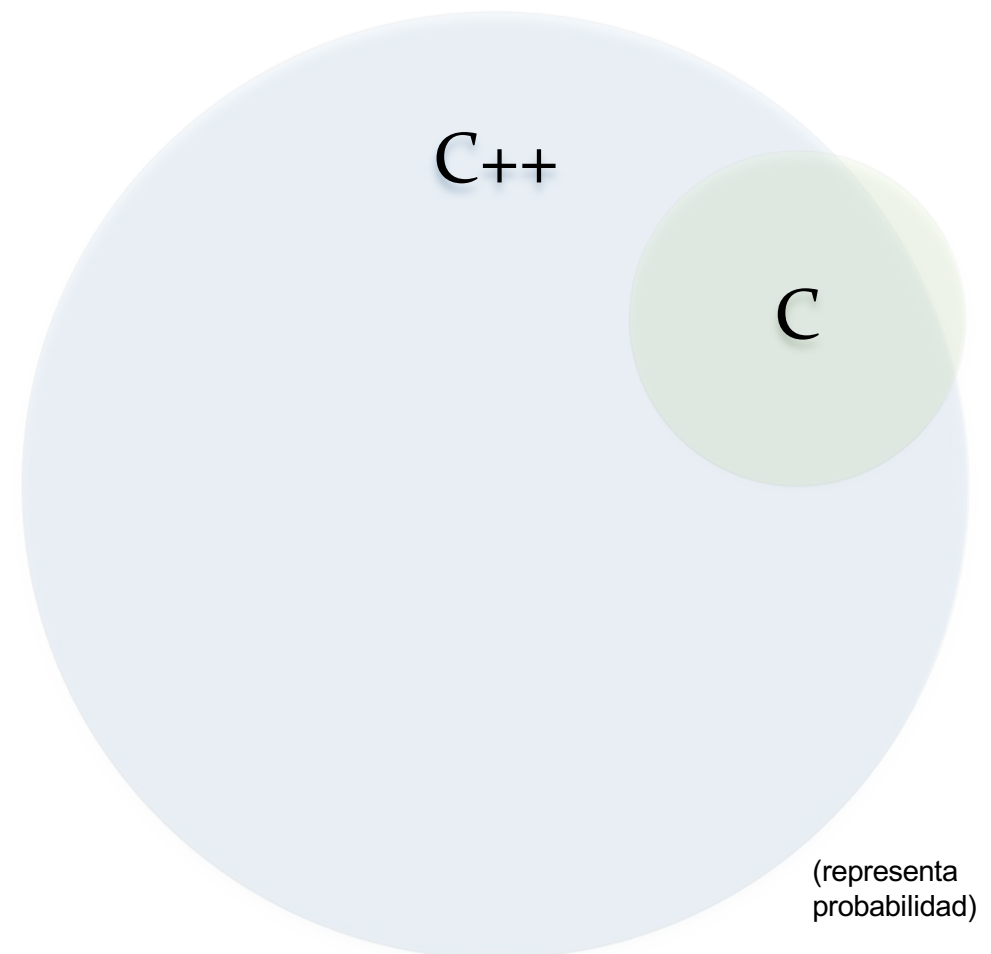
Real programmers code in binary.

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,
- 2020's
 - Standard C23
 - Standard C++23

Conjuntos de infinitos programas válidos de C++ y C



"Hello, World!"

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



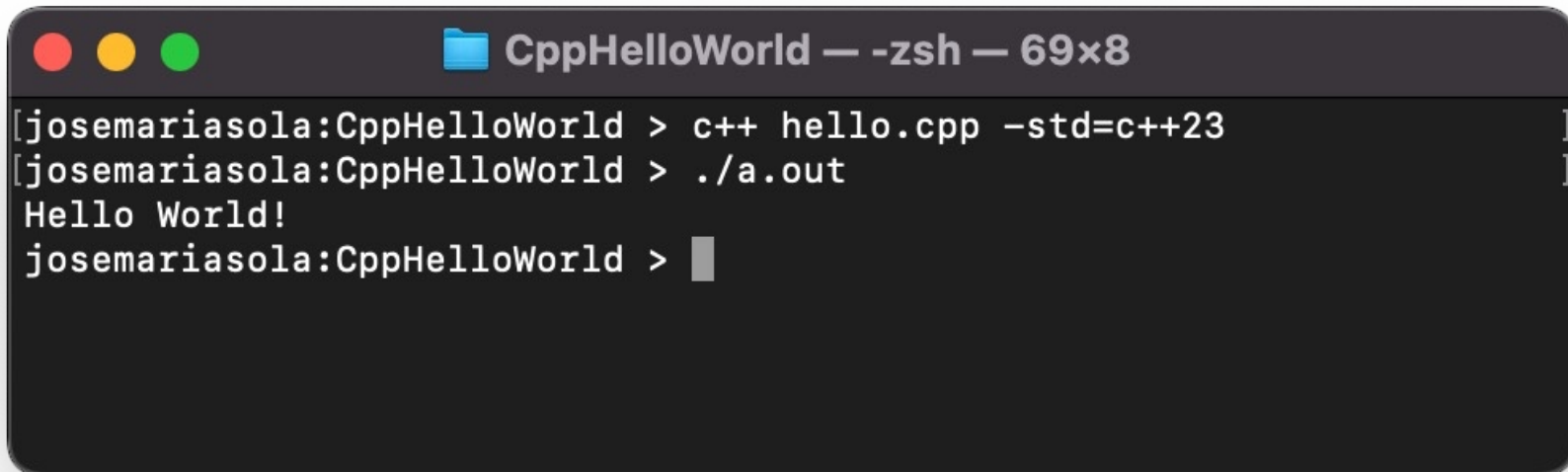
- Propósito
- Comentario encabezado
 - Qué
 - Título descriptivo
 - Quién
 - Número de Equipo e integrantes
 - Cuándo
 - Se actualizó por última vez

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.cpp) ...
... 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 macOS

1. Desde la línea de comando
 1. > `vim hello.cpp` *crear el fuente*
 2. > `c++ hello.cpp` *compilar y 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
- **Otras alternativas para cc son:**
 - > `c++ hello.cpp -o hello` Indica el nombre del ejecutable salida de la compilación (output), en vez del default `a.out`
 - > `c++ hello.cpp -std=c++23` Compila según la *versión del estándar C++23*



```
CppHelloWorld — -zsh — 69x8
[josemariasola:CppHelloWorld > c++ hello.cpp -std=c++23
[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



```
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 C4530: 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

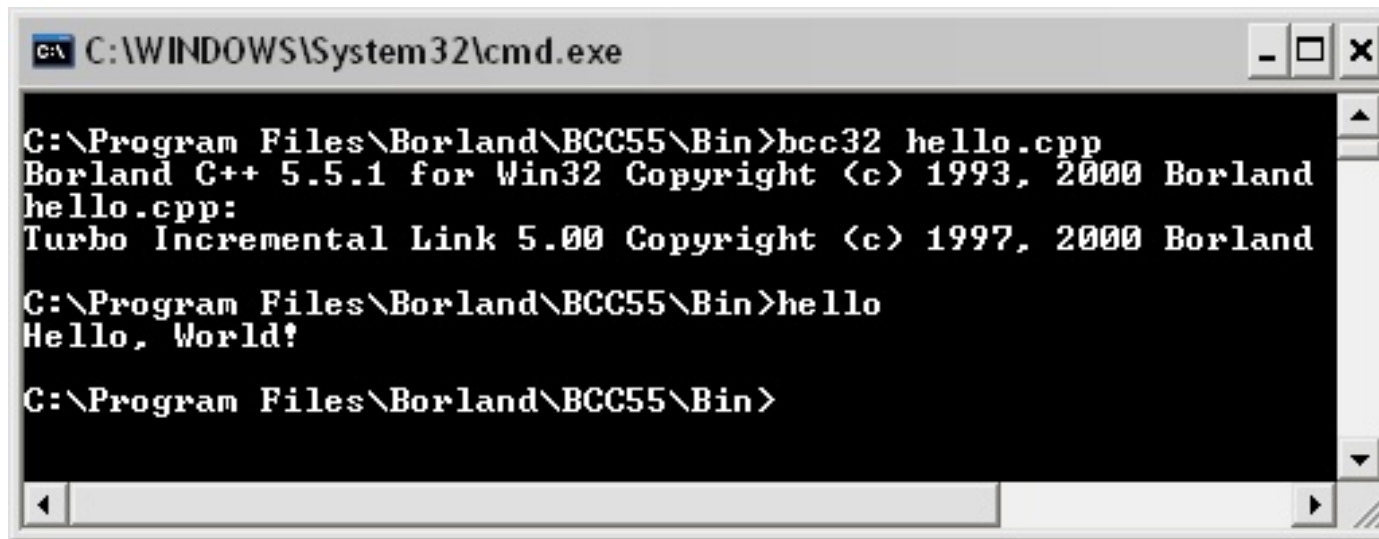
C:\Samples>hello
Hello World!

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



```
C:\WINDOWS\System32\cmd.exe

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 Enviroment*, Entorno Integrado de Desarrollo)
 - Ejemplos
 - Apple Xcode
 - Microsoft Visual Studio
 - Eclipse
- Sin IDE
 - Editor
 - Compilador.

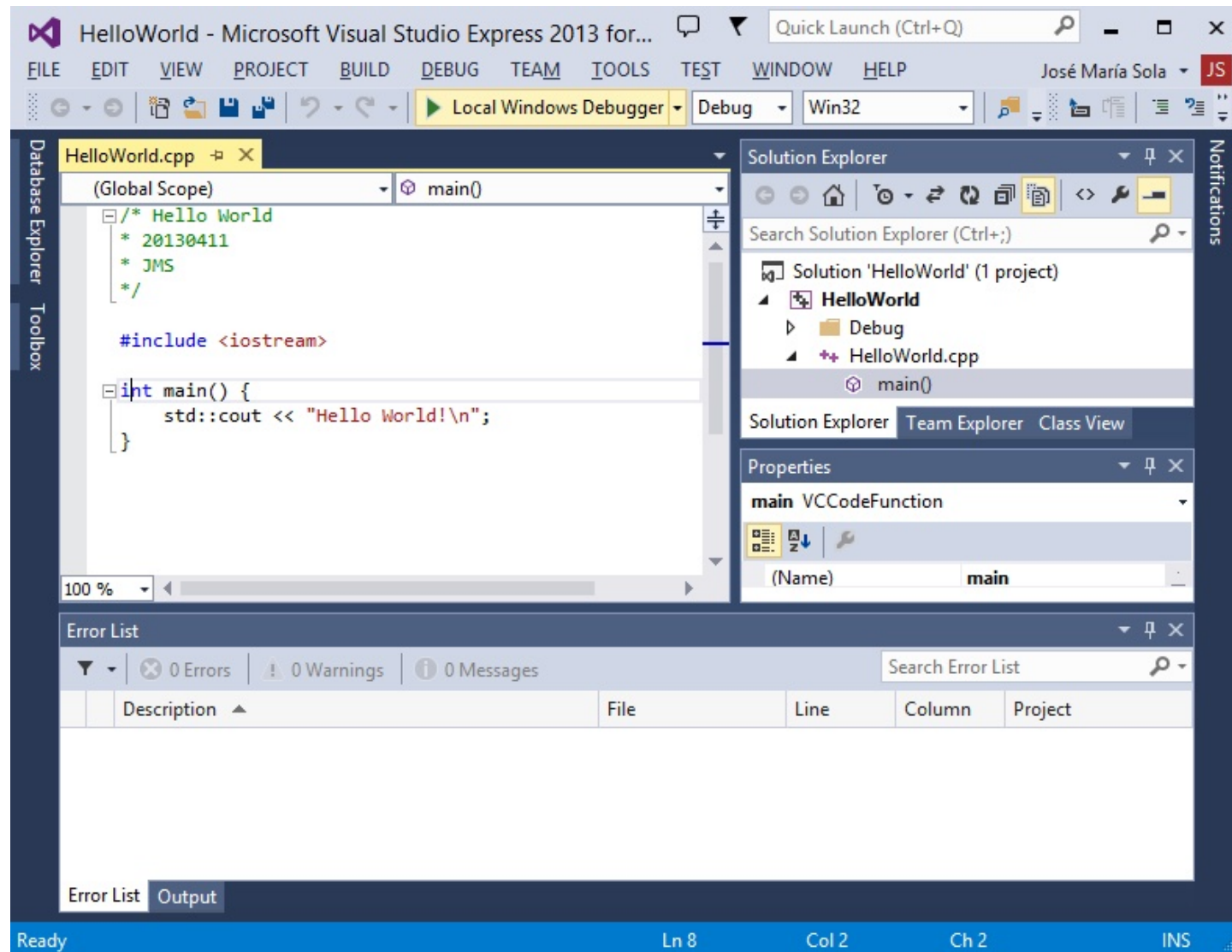
Con IDE

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

Sin IDE
Requiere editor

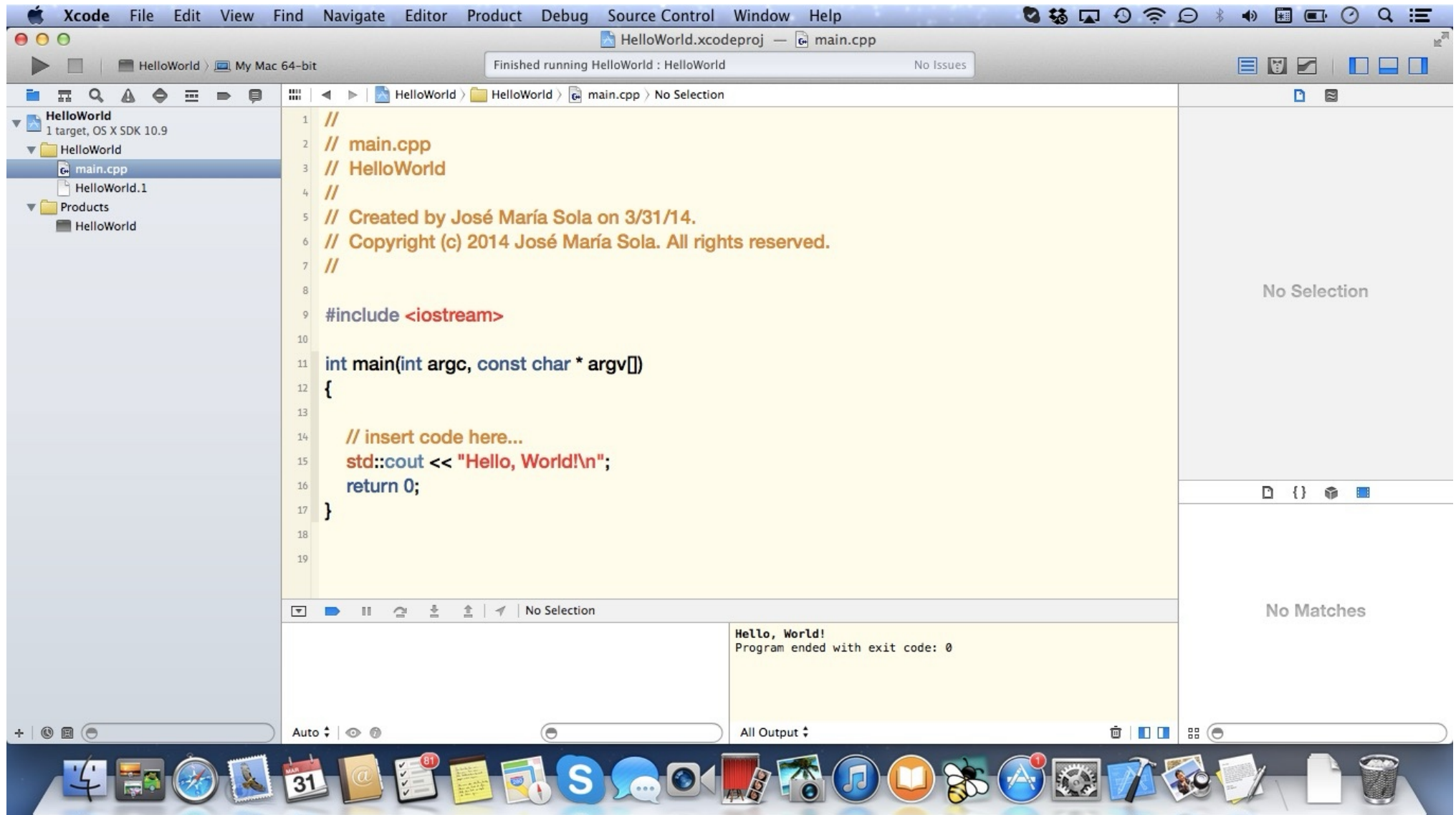
Compilador de C/C++

Ejemplo con IDE Microsoft Visual Studio Express for Windows Desktop



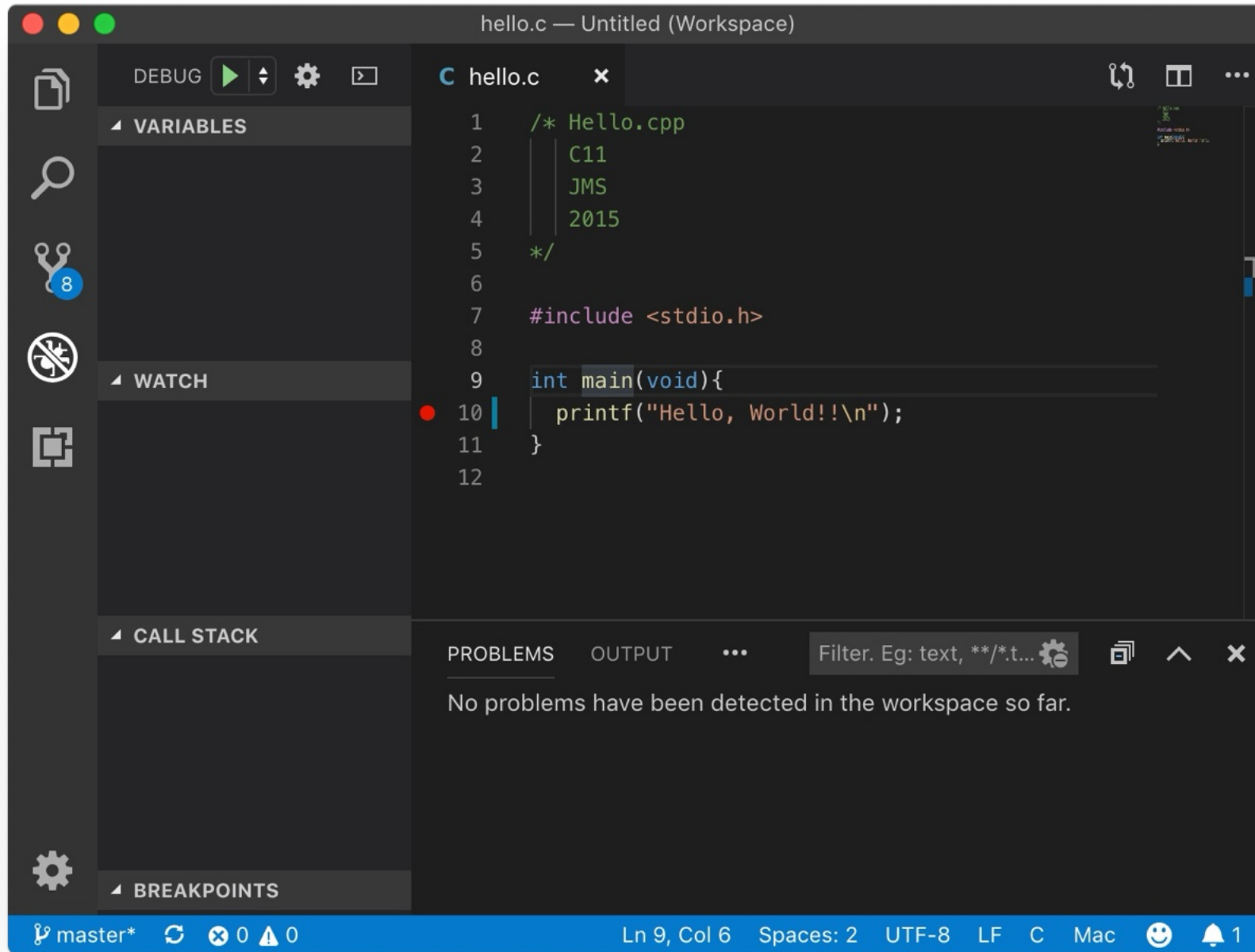
Ejemplo con IDE

Apple XCode



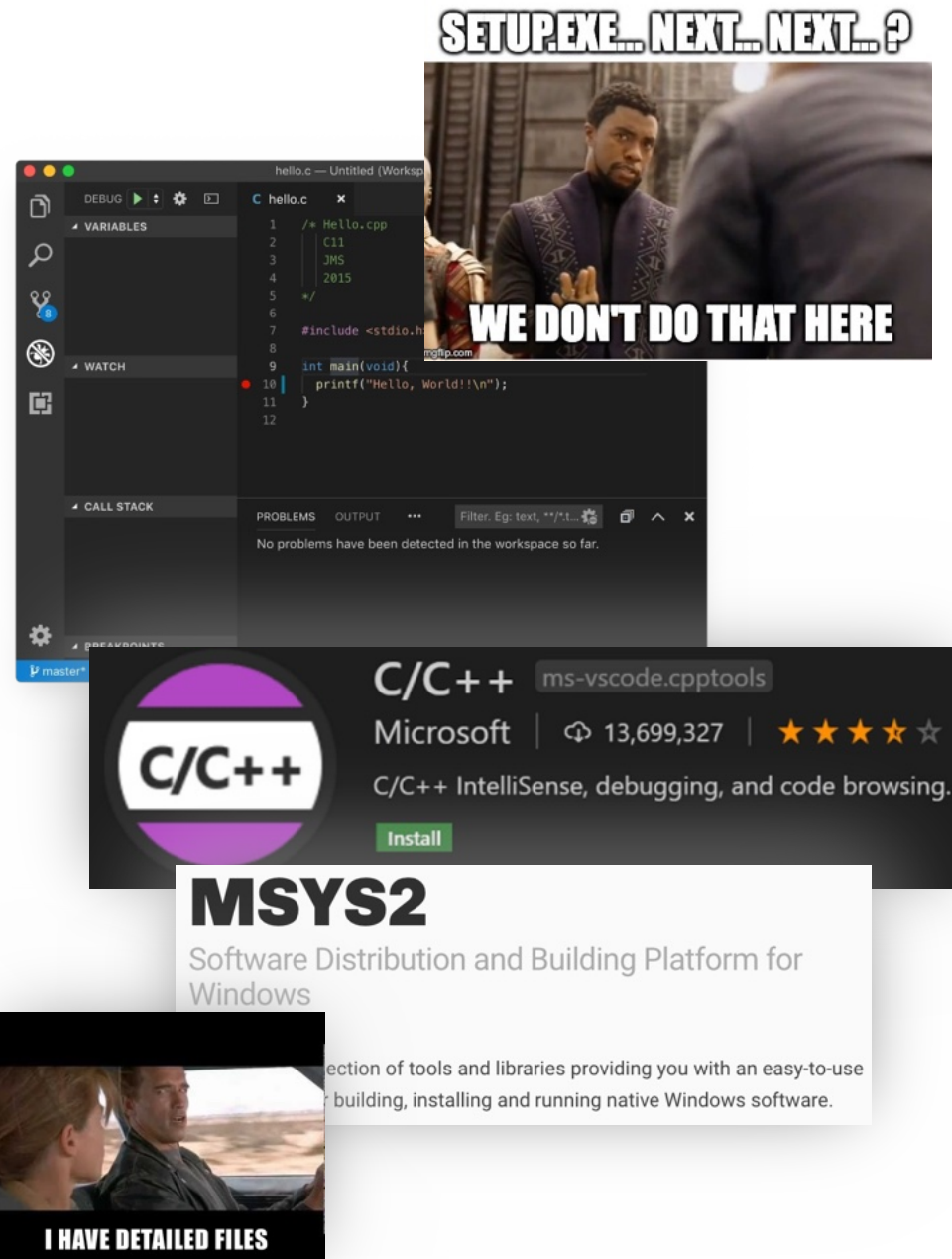
Ejemplo con IDE

Microsoft Visual Studio Code (ejemplo en C, no C++)



Instalación de Microsoft Visual Studio Code

- Es más que un editor de texto, es un editor de código fuente
- Hoy en día es muy popular
- Agrega funcionalidad mediante un sistema de extensiones, para nuestro caso: C/C++
- La extensión no incluye un compilador ☹, por lo que hay que conseguir uno, por ejemplo gcc, clang, o cl
 - Los sistemas Windows pueden obtener gcc mediante MinGW (the Minimalist GNU for Windows)
 - MinGW se puede obtener desde [MSYS2](#)
- [La extensión C/C++ tiene documentación detallada sobre la instalación de todo el tool-chain](#)



Make (C++)

- Nuestro objetivo (*goal*) es fabricar (construir, 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-ake.exe` si instalamos *MinGW*.



```
CppHelloWorld — -zsh — 156x8
[josemariasola:CppHelloWorld > make hello
c++ -std=c++23 -Weverything -pedantic-errors -Wno-c++98-compat -Wno-c++98-compat-pedantic -Wno-padded -Wno-poison-system-directories  hello.cpp -o hello
[josemariasola:CppHelloWorld > ./hello
Hello World!
josemariasola:CppHelloWorld > ]
```

Links a Compiladores C/C++

Usar unos de estos compiladores o cualquier otro, siempre y cuando se lo configure para **C++23**

- Con IDE y Línea de Comandos
 - Microsoft Visual Studio Community 2022
 - <https://www.visualstudio.com/vs/features/cppplusplus/>
 - Microsoft Code
 - <https://code.visualstudio.com/Download>
 - Apple Xcode
 - <https://developer.apple.com/xcode>
 - CodeLite
 - <https://codelite.org>
 - Eclipse IDE for C/C++ Developers
 - <https://www.eclipse.org/downloads/packages/release/2025-03/r/eclipse-ide-cc-developers>
 - Replit (IDE On-Line)
 - <https://replit.com/languages/c>
 - Más antiguos
 - Code::Blocks
 - <http://www.codeblocks.org/downloads/>
 - Dev-C++
 - <http://orwelldvcpp.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>
- Compilador On-line:
 - Compiler Explorer: <https://godbolt.org>

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++23 (o por lo menos 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 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

- El Contexto de AED
 - Ingeniería
 - Sistema
 - Información
 - Sistema de Información
 - Ingeniería en Sistemas de Información
 - Software
 - Programación
 - Problema (Necesidad)
 - Solución
 - Análisis
 - Diseño
 - Implementación
 - Desarrollo
 - Proyecto
 - Organización
- Relación con Otras Asignaturas
 - Áreas de Conocimiento de nuestro plan de estudio
- Repaso Módulo B
 - Conjuntos Numéricos
 - Ecuaciones e Inecuaciones
 - Función
 - Vector
- Primer Contacto con la Programación
 - Lenguaje de Programación
 - Código máquina
 - Assembler
 - C
 - C++
 - Editor
 - Compilador
 - Línea de comando
 - IDE (Integrated Development Enviroment, Entorno Integrado de Desarrollo)

Tareas para la próxima clase

1. Comenzar a trabajar sobre el Trabajo #0

¿Consultas?

Fin de la clase