

Clase #02 de 27 El Compilador & el Lenguaje de Programación C

Marzo 25, Lunes

Agenda para esta clase

- Sobre la Cátedra y la Asignatura
- Plataforma del Curso
- Intervalo
- Primer contacto con el compilador
- Trabajo #0
- Introducción al Lenguaje de Programación C

Sobre la Cátedra y la Asignatura

Análisis del Nombre de la Asignatura – Sintaxis y Semántica de los Lenguajes (de Programación)

- Lenguaje: medio de comunicación
 - Lenguajes Naturales versus Lenguajes Formales
 - Lenguaje de programación: medio de comunicación para definir comportamiento de un Sistema Software
- Sintaxis: Forma, Estructura
 - Reglas de forma
 - ¿Cómo se escribe?
- Semántica: Significado, Sentido
 - Reglas de significado
 - ¿Qué significa?
 - Comportamiento del sistema en función de lo escrito
- ¿Léxico?
 - Componentes de las estructuras sintácticas
- ¿Pragmática?
 - “Bugs”, “Gaps”, Verificación, Validación, Comportamiento esperado.

Generalidades de SSL

Ordenanza 1150/2007

- Departamento **Ingeniería en Sistemas de Información**
- Bloque **Tecnologías Básicas**
- Área **Programación**
- Nivel **2**
- Horas/semana **4**
- Horas/Año **128**
- Correlatividades
 - Anteriores
 - **Algoritmos y Estructuras de Datos (1ro)**
 - **Matemática Discreta (1ro)**
 - Siguientes
 - Debe estar regularizada para cursar
 - **Gestión de Datos (3ro)**
 - Debe estar aprobada para cursar
 - **Ingeniería en Software (4to)**

Programa Sintético de SSL

Ordenanza 1150/2007

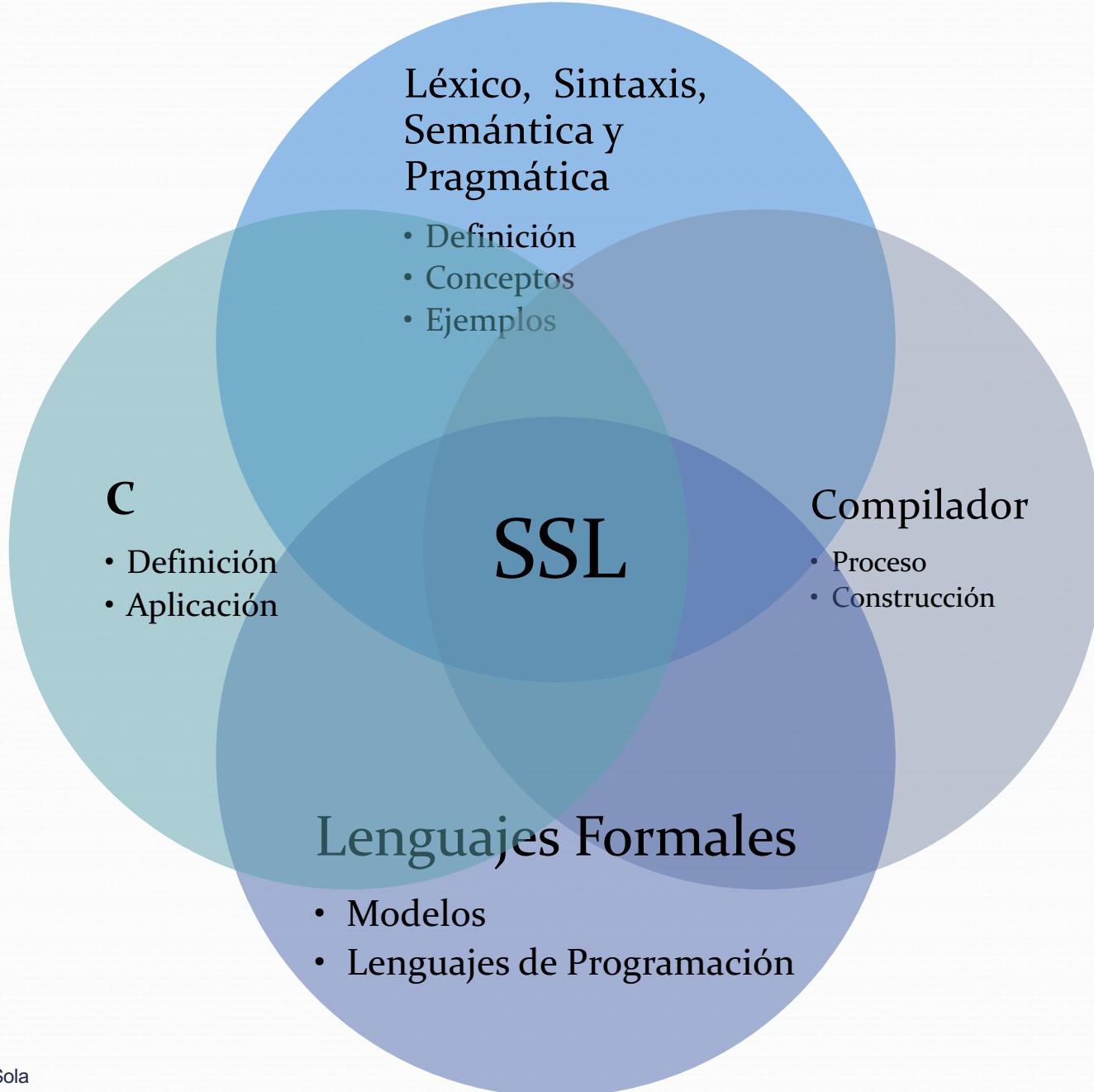
Objetivos

- Conocer los elementos propios de la **sintaxis y semántica** de los **lenguajes de programación**
- Conocer los **lenguajes formales** y **autómatas**
- Comprender conceptos y procedimientos de las **gramáticas libres de contexto** y **gramáticas regulares** para especificar la **sintaxis de los lenguajes de programación**
- Utilizar distintos tipos de autómatas y distintos tipos de notaciones gramaticales
- Comprender el procesamiento de lenguajes y en particular, el **proceso de compilación**

Contenidos Mínimos

- **Gramática y Lenguajes Formales**
- **Jerarquía de Chomsky**
- **Autómatas Finitos**
- **Expresiones Regulares** y su aplicación al **Análisis Léxico**
- **Gramáticas Independientes del Contexto**
- **Autómatas PushDown** y su Aplicación al **Análisis Sintáctico**
- Otros Tipos de Analizadores Sintácticos
- **Máquinas Turing**
- Introducción a las **Semánticas**.

Los Temas de SSL



La Cátedra

- **Profesores Adjuntos**
 - Ing. Adrián Bender
 - Dr. Mgs. Oscar Ricardo Bruno
 - Ing. Santiago Ferreiros
 - Ing. Pablo Méndez
 - Ing. Silvina Ortega
Profesora Adjunta (Carrera Académica)
 - Mgs. Ing. Gabriela Sanromán
 - Lic. Eduardo Zúñiga
 - Ing. Roxana Leituz
 - Esp. Ing. José María Sola
- **Anteriores Profesores**
 - C.C. Jorge Muchnik
Profesor Titular (Ordinario)
 - C.C. Marta Ferrari
Profesora Asociada (Carrera Académica)
 - C.C. Ana María Díaz Bott
Profesora Adjunta (Carrera Académica)
 - C.C. Adriana Adamoli
Profesora Adjunta Interina (Interino)

Plataforma del Curso

Herramientas para facilitar la cursada

Plataforma del Curso

**Clases
Presenciales**

**josemariasola.
wordpress.com**
*Homepage del
curso*

Bibliografía
• *K&R1988*
• *MUCH2012*

**Yahoo Groups
UTNFRBASSL**
*Foro para consultas
y anuncios*

GitHub
*Resoluciones
Individuales y del
cada equipo*

Google Calendar
*Caledario con detalle
de actividades*

josemariasola.wordpress.com

- Homepage del Curso
- Papers
- Referencia
- Trabajos
- Equipos
- Calendario
- Resúmenes de Clases (se actualiza con poca frecuencia)
- Acceso a Foro
- Acceso a GitHub.

Bibliografía

- Bibliografía obligatoria
 - [K&R1988]
 - Biblioteca, Fotocopiadora, y Librerías
 - [MUCH2012], tres volúmenes
 - Biblioteca, y Librería del Centro de Estudiantes
- Otros artículos a lo largo del curso, publicados desde la página del curso
- Prioritarios para primera parte del curso
 - Capítulo #1 de [K&R1988]
 - Volumen #1 de [MUCH2012].

Grupo Yahoo UTNFRBASSL

<https://groups.yahoo.com/group/UTNFRBASSL>

The screenshot shows the Yahoo Groups page for the group "UTN FRBA SSL". The page has a purple header with links for Home, Mail, Search, News, Sports, Finance, Weather, Games, Answers, Screen, Flickr, Mobile, and More. The main navigation bar includes "YAHOO! GROUPS", a search bar for "Search Conversations", and buttons for "Search Groups" and "Search Web". On the far right are "Sign In", "Mail", and a gear icon. The left sidebar has links for "Join Groups", "Privacy", "Feedback", and "Blog". The main content area features a banner image of two stacked books. Below the banner, the group name "UTN FRBA SSL" is displayed, followed by "Restricted Group, 1972 members". A navigation bar below the banner includes "Conversations", "Photos", "Files", "About" (which is underlined), and "More". The "About" section contains "About Group" and a "Join Group" button. The "Group Description" section contains the following text:

UTN FRBA SSL
Grupo Yahoo de la Cátedra de
Sintaxis y Semántica de los
Lenguajes(SSL)
de la especialidad
Ingeniería en Sistemas de Información
en la
Facultad Regional Buenos Aires

FEEDBACK

To the right of the text is a large blue square logo featuring a black stylized letter "X" or a four-pointed star shape.

Grupo Yahoo UTNFRBASSL

<https://groups.yahoo.com/group/UTNFRBASSL>

- Objetivos
 - Foro del curso
 - Canal de comunicación
 - Materiales de lectura
 - Consultas
 - Enunciados
 - Anuncios particulares del curso y generales
- Antes de cada clase se debe revisar si hay novedades en el grupo
- Periodo de suscripción
 - durante las tres primeras semanas del cuatrimestre
 - Si ya está inscripto no es necesario reinscribirse
 - Cuenta de e-mail (de cualquier dominio) con nombre y apellido, asociada al usuario Yahoo
- Requisitos para suscripción
 - Usuario Yahoo con nombre y apellido
 - Cuenta de e-mail (*de cualquier dominio no es necesario sea Yahoo*) con nombre y apellido, asociada al usuario Yahoo
 - Desde la página de suscripción, completar formulario con
 - nombre y apellido,
 - legajo y
 - código de curso
- Comunicación
 - El subject (asunto del mensaje) debe comenzar con el código de curso y debe ser descriptivo
 - Se debe firmar con nombre y apellido
 - No están permitidos los Off-Topic
 - Reglas para el uso del grupo Yahoo en la sección Files (archivos) del propio grupo.

Git y GitHub

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

- Sistema de Control de Versiones (VCS)
 - Gestión de cgos 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 pepositorios 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-ssl**
 - Para los miembros del curso asignamos una repositorio privado, dentro **utn-frba-ssl**, a cada equipo
 - <https://github.com/orgs/utn-frba-ssl/>
- Primeros Pasos en GitHub
 - 1. Leer guía "Git 101"
 - 2. Crear Usuario
 - 3. Crear Repositorio.

<https://josemariasola.wordpress.com/ssl/git/>

Intervalo

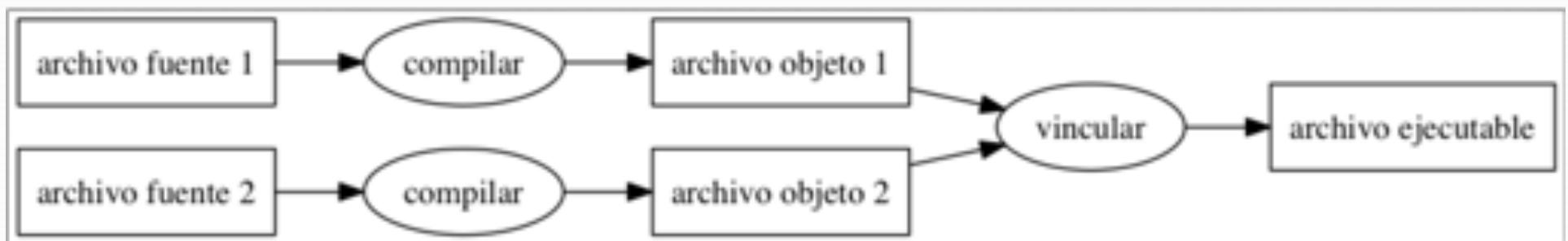
20 minutos

Primer Contacto con el Compilador

Lenguajes y Herramientas de Desarrollo

¿Qué es un Compilador?

- Programa que hace programas, un meta programa
- Traductor
- Función de Lenguaje a Lenguaje: $C: L_1 \rightarrow L_2$
- Baja de Nivel de Abstracción
- Proceso, en etapas: Front End y Back End
- Compilaciones separadas, luego vinculadas

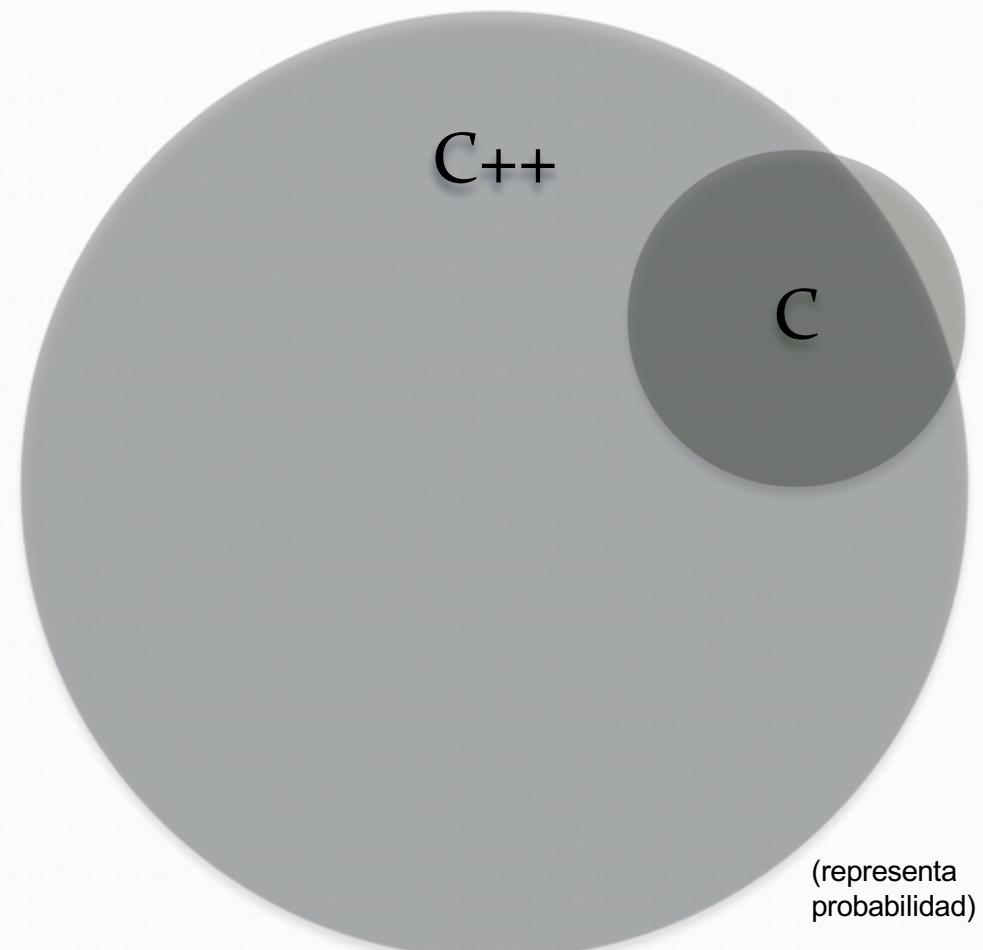


Sobre los Lenguajes C y C++

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.

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



"Hello, World!"

```
/* Hello world
   JMS
   20150402
*/
#include <stdio.h>
int main(void){
    printf("Hello, world!\n");
}
```

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

- 1.1 [K&R1988]
- [https://en.wikipedia.org/wiki/"Hello,_World!"_program](https://en.wikipedia.org/wiki/)

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 .c (e.g., hello.c)
2. **Compilar** el archivo fuente para producir el programa objeto (e.g., cc 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 C11

1. Desde la línea de comando

1. > vi hello.c crear el fuente
 2. > cc hello.c -std=c11 -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

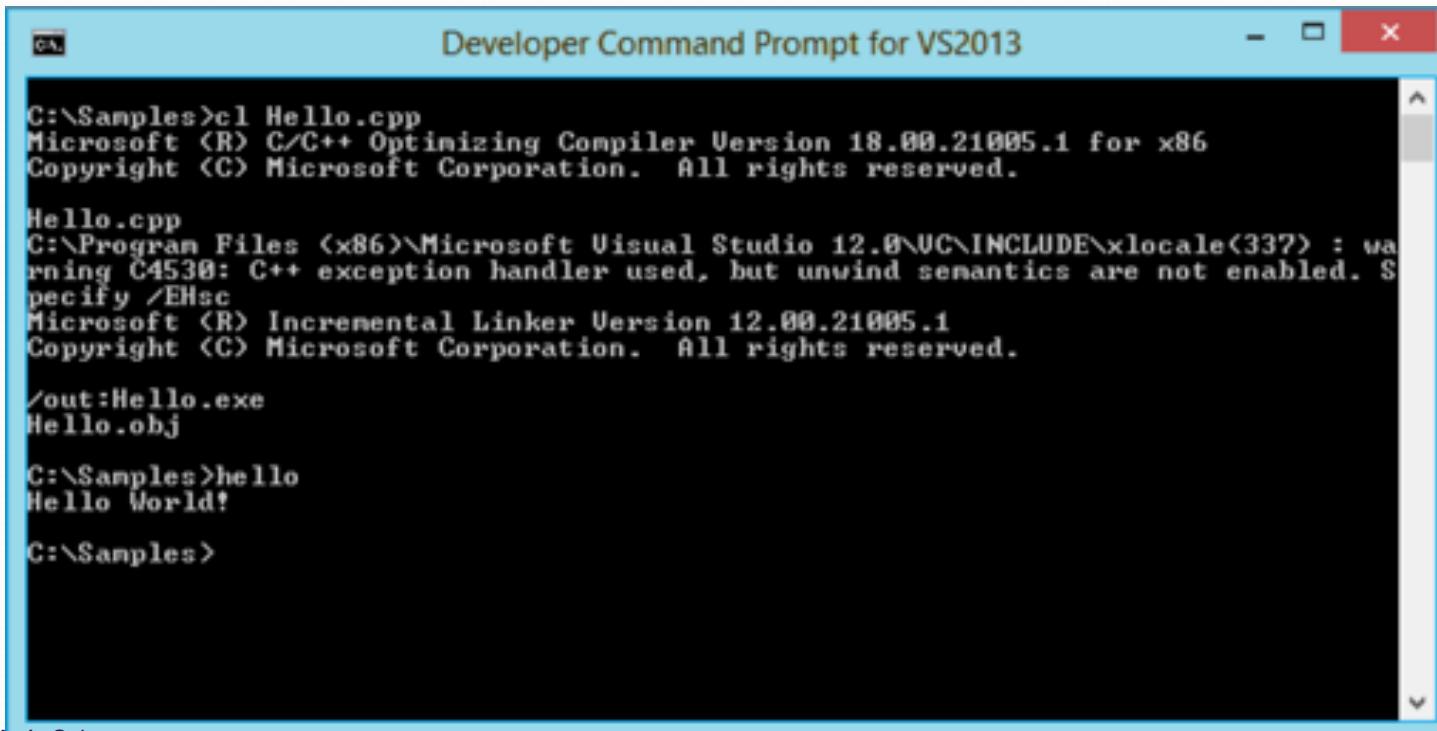
Otra versión para gcc es:

> cc hello.c -std=c11 -Wall -pedantic-errors

```
josemariasola:CHelloWorld> cc hello.c -std=c11 -Weverything
josemariasola:CHelloWorld> ./a.out
Hello, World!
josemariasola:CHelloWorld>
```

Ejemplo desde línea de comando Compilador Microsoft (ejemplo en C++, no C)

1. Desde la línea de comando
 1. > **notepad hello.c** crear el fuente
 2. > **cl hello.c** 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 "Developer Command Prompt for VS2013". The command line shows the steps to compile and run a C++ program:

```
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.c** crear el fuente
 2. > **bcc32 hello.c** 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 'C:\WINDOWS\System32\cmd.exe'. The command line shows the steps of the process:

```
C:\Program Files\Borland\BCC55\Bin>bcc32 hello.c
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
hello.c:
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.

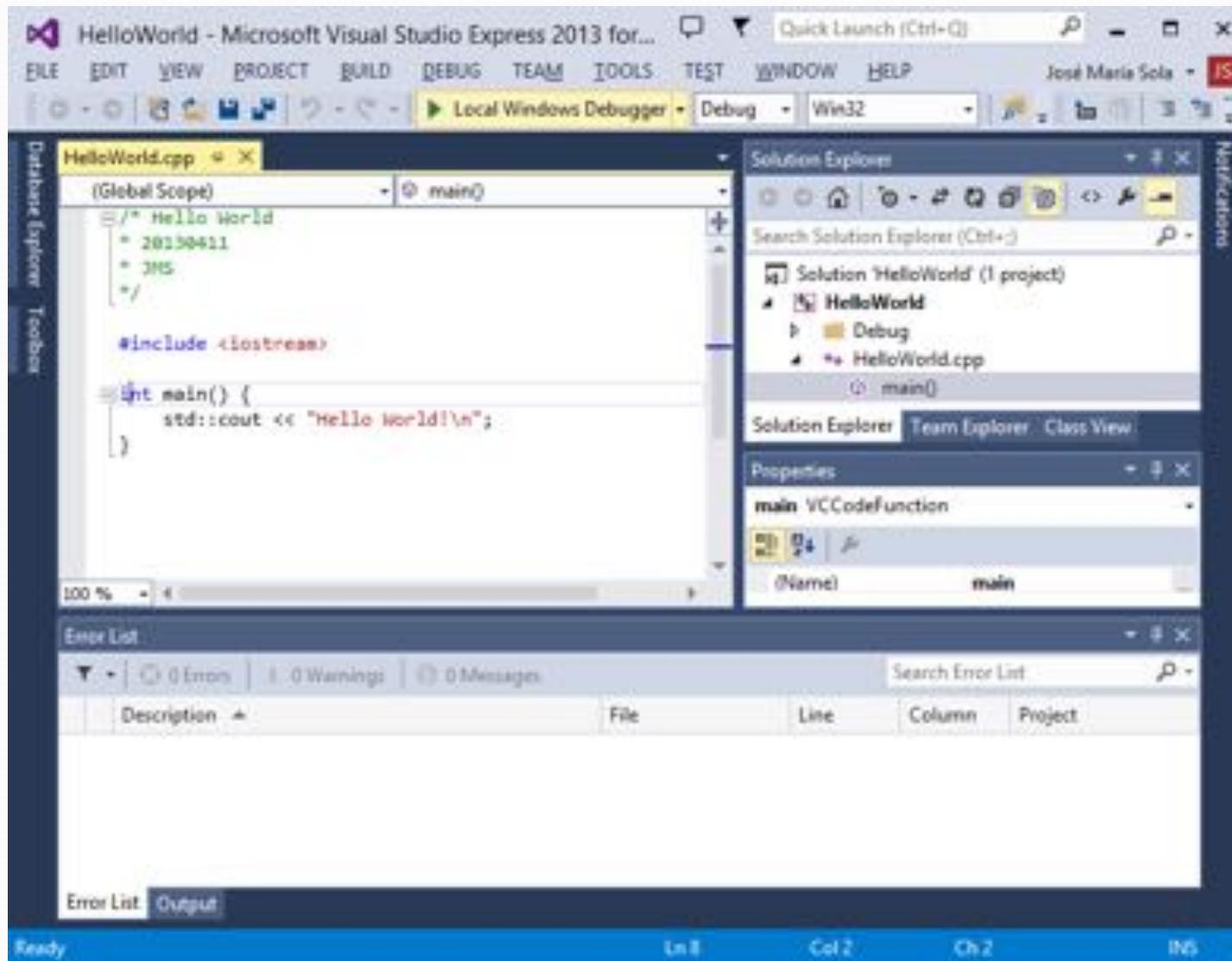
Con IDE

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

Sin IDE
Requiere editor

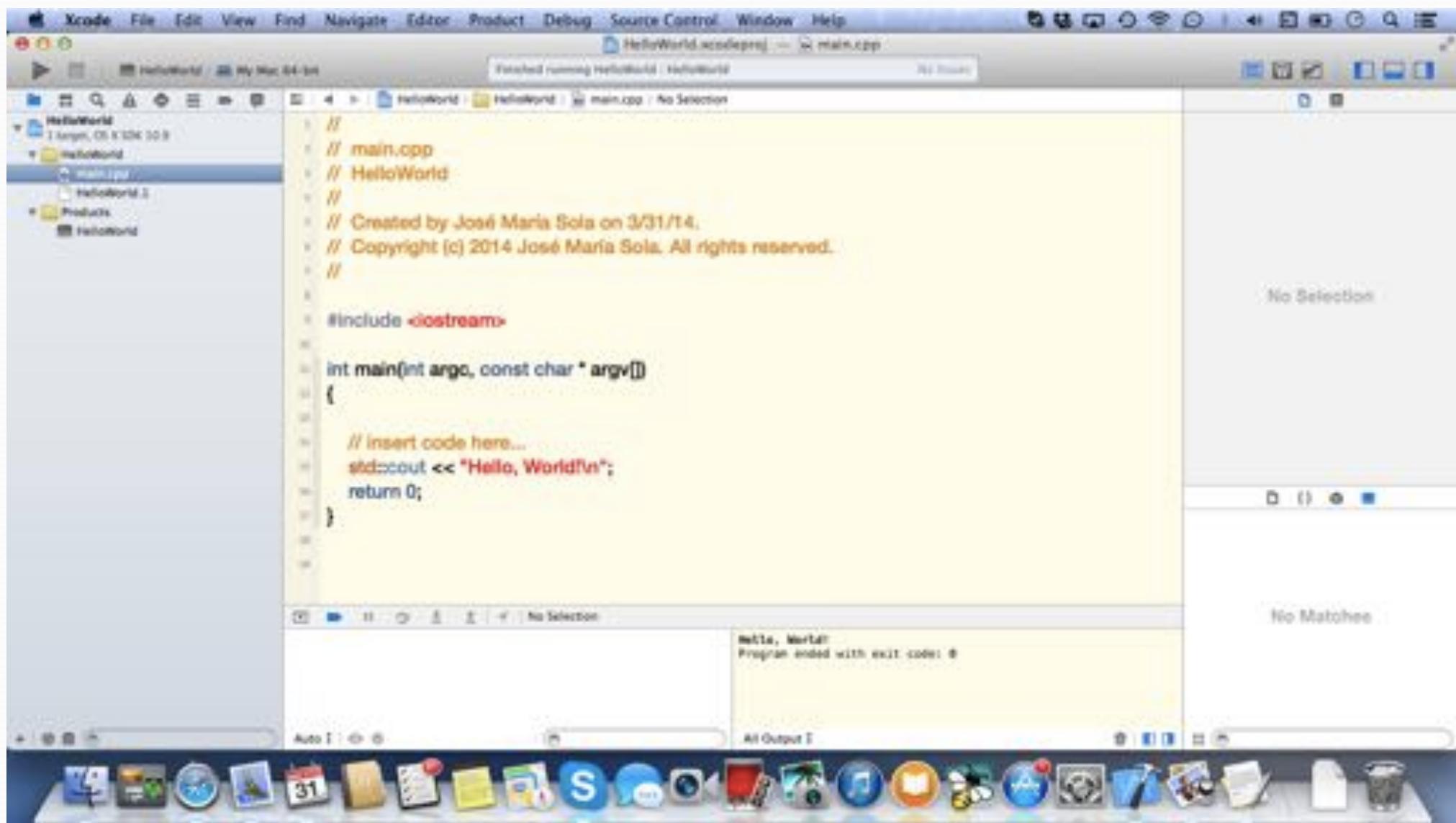
Compilador de C/C++

Ejemplo con IDE Microsoft Visual Studio Express for Windows Desktop (Ejemplo en C++, no C)



Ejemplo con IDE

Apple Xcode (ejemplo en C++, no C)



Ejemplo con IDE

Microsoft Visual Studio Code

The screenshot shows the Microsoft Visual Studio Code interface with a dark theme. The main editor window displays a C file named `hello.c`. The code contains 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 sidebar on the left includes sections for `VARIABLES`, `WATCH`, `CALL STACK`, and `BREAKPOINTS`. The top bar shows the file name `hello.c — Untitled (Workspace)` and various icons for debugging, search, and settings. The bottom status bar shows the file path `master`, line `Ln 9, Col 6`, and encoding `UTF-8`.

```
/* 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.c`
- 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
cc -std=c17 -Weverything -pedantic-errors hello.c -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 **C11 o C18**

- Con IDE y Línea de Comandos
 - Microsoft Visual Studio Community 2017
 - <https://www.visualstudio.com/vs/features/cplusplus/>
 - 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/2019-03/r/eclipse-ide-cc-developers>
 - 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://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/212062>

Trabajo #0

hello.c: "Hello, World!"

Trabajo #0 – "Hello, World!" en C

- Enunciado en josemariasola.wordpress.com
- Secuencias de Tareas
 - Solicitar inscripción al Grupo Yahoo UTNFRBASSL, un par de días para la aprobación
 - Si no posee una cuenta GitHub, crearla
 - Crear un repositorio público llamado SSL
 - Escribir el archivo readme.md que actúa como front page del repositorio personal
 - Crear la carpeta oo-CHelloWorld.
 - Escribir el archivo readme.md que actúa como front page de la resolución.
 - Seleccionar, instalar, y configurar un compilador C11 ó C18
 - Probar compilador con hello.c que envíe a stdout la línea Hello, World! o similar
 - Ejecutar el programa, y capturar su salida en un archivo de texto output.txt
 - Publicar en repositorio personal SSL \ oo-CHelloWorld: readme.md, hello.c, y output.txt
 - Informar por email a UTNFRBASSL@yahoogroups.com el usuario usuario GitHub.
- Restricciones
 - La fecha y hora límite de entrega se publica en el calendario
 - La evaluación se hace con lo publicado en GitHub.

Introducción al Lenguaje de Programación C

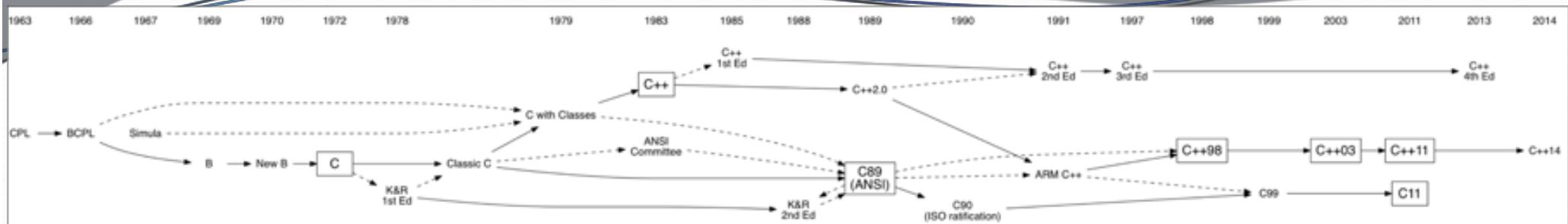
Descripción general

- LP de propósito general, no está especializado
- Economía en las expresiones, pero expresivo
 - Poco texto, mucha información
 - Variabilidad en texto, variabilidad en significado.
- Control de flujo (*¿de qué?*)
- Estructuras de datos
- Gran cantidad de operadores
- flexible
- No es de muy alto nivel (*¿de qué?*)
- No es grande (*¿en qué sentido?*)
- Su falta de restricciones y su generalidad lo hacen efectivo
- Independiente de máquina, portable (procesador y sistema operativo)
- El lenguaje de programación de Unix
- Primer lenguaje de alto nivel eficiente y portable
 - En el momento, menos problemas que
 - Basic, PL/I, Fotran, Cobol, Pascal
 - Comparado con Lisp
 - Vinculación
 - Más rápido
 - Con GC, lo cual no es apropiado para programación de sistemas
 - C es la mejor abstracción de una computadora existente, no de un dispositivo imaginario
 - Suficientes estructuras de control y de datos para resolver problemas, limitadas para que se pueda implementar el compilador.

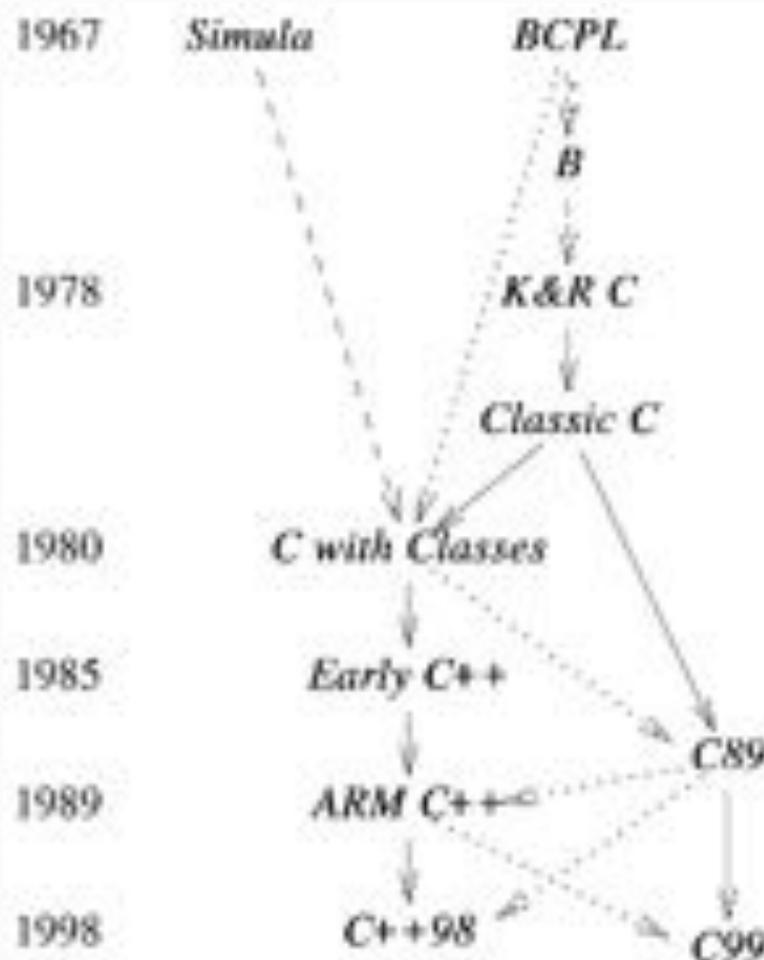
Frases sobre C

- C no es un LP grande, y no le queda bien un libro grande (K&R)
- C tiene vueltas, falencias y un enorme éxito (Ritchie)
- C es un arma filosa, con la se pueden hacer programas eficientes y elegantes o una “carnicería” (Pike)
- C mejora a medida que uno gana experiencia con C (K&R) (Curva de aprendizaje empinada).

Historia de C y LP relacionados



- 1969-1973
 - Elaboración. Basado en B, a su vez, basado en BCPL
 - C Pre estandarización
- 1978
 - K&R 1era edición
- 1983
 - Comienza estandarización
- 1988
 - K&R 2da edición
- 1989-90
 - C89-C90 (1era versión estándar)
- 1999
 - C99 (2da versión estándar)
- 2011
 - C11 (3era versión estándar)
- 2019
 - C18 (4ta versión, sin cambios visibles)



- Otros LP
 - C++
 - Objective-C
 - C#
 - D
 - Java

Términos de la clase #02

Definir cada término con la bibliografía

- SSL
 - Lenguaje de Programación
 - Comportamiento
 - Lenguaje
 - Lenguaje Natural
 - Lenguaje Formal
 - Sintaxis
 - Semántica
 - Léxico
 - Pragmática
 - Bugs
 - Gaps
 - Verificación
 - Validación
- Plataforma del Curso
 - VCS (Sistema de Control de Versiones)
 - Git
 - GitHub
- Repositorio
- Compilador
 - Función del compilador
 - Proceso de compilación
 - Proceso básico para desarrollar programas
 - Lenguaje máquina (bajo nivel de abstracción)
 - Lenguaje de Alto Nivel de Abstracción
 - C
 - C++
 - Hello World (Kernighan)
 - IDE (Integrated Development Environment, Entorno Intedrado de Desarrollo)
 - Utilidad Make
- Introducción al Lenguaje de Programación C
 - Nivel de abstracción
- Independiente de Máquina
- Control de flujo de ejecución
- Historia de C y de ANSI C
- Relación entre C y C++
- ANSI C, C89 ó C90
- C99
- C11
- C18

Tareas para la próxima clase

1. Leer de [K&R1988] *desde la tapa hasta la sección **1.1 Comenzado** inclusive*
2. Entrega Trabajo #0
3. Leer de [K&R1988] *desde **1.2 Variables y Expresiones Aritméticas** hasta **1.4 Constantes Simbólicas** inclusive.*

¿Consultas?



Fin de la clase