

Sintaxis y Semántica de los Lenguajes

SSL

K2051

2017

Prof. Esp. Ing. José María Sola

Universidad Tecnológica Nacional

Facultad Regional Buenos Aires

Departamento de Ingeniería en Sistemas de Información

Clase #01 de 27

Presentación de la Asignatura y Pautas Generales

Marzo 13, Lunes

Agenda para esta clase

- Bienvenidos a segundo año y al curso
- Presentación inicial, conozcámonos
- El contexto de SSL
- Sobre la Cátedra y la Asignatura
- La aprobación y la regularización
- Plataforma del Curso
- Sobre el Lenguaje C
- Primer contacto con el compilador
- Tareas para la próxima clase
- Términos de la clase.

Presentación Inicial

Conozcámonos

Docentes del Curso

- Profesor a Cargo
 - Esp. Ing. José María Sola
- Auxiliares
 - Srta. Elizabeth Sosa

En la Clase y Horarios

- Primeras 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 la tiene)
- Celular, en **silencio o vibrador**
- Horarios del curso
 - 4 horas cátedra con intervalo de 15 minutos
 - 3 horas reloj + intervalo de 15 o 20 minutos
 - **19:45 a**
 - Intervalo 21:00
 - **23:00**
- Consultas presenciales, antes y después de clase.

El Contexto de SSL

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



Otras Asignaturas

- Asignaturas Anteriores
 - Algoritmos y Estructura de Datos
 - Matemática Discreta
 - Sistemas y Organizaciones (no correlativa)
- Asignaturas Paralelas
 - Paradigmas de Programación
 - Análisis de Sistemas
 - Sistemas Operativos (segundo nivel, según el plan)

Algoritmos y Estructuras de Datos

- Objetivos
 - Identificar problemas algorítmicos.
 - Conocer el proceso de diseño e implementación de software
 - Aplicar las herramientas fundamentales representativas de los procesos, integrando la sintaxis elemental de un lenguaje de programación en el laboratorio asociado
- Contenidos Mínimos
 - Concepto de Dato
 - Tipos de Datos Simples
 - Tipo Abstracto de datos
 - Estructuras de Control Básicas: secuencial, condicional, cíclica
 - Estrategias de Resolución
 - Estructuras de Datos: registros, arreglos y archivos
 - Abstracciones con procedimientos y funciones
 - Pasaje de Parámetros
 - Estructuras de Datos lineales (Pilas-Colas)
 - Algoritmos de Búsqueda, Recorrido y Ordenamiento
 - Archivos de Acceso Secuencial y Aleatorio: organizaciones y accesos.
 - Procesamiento Básico
 - Recursividad
 - Nociones de Complejidad Computacional
 - Noción de Orden de Complejidad.

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

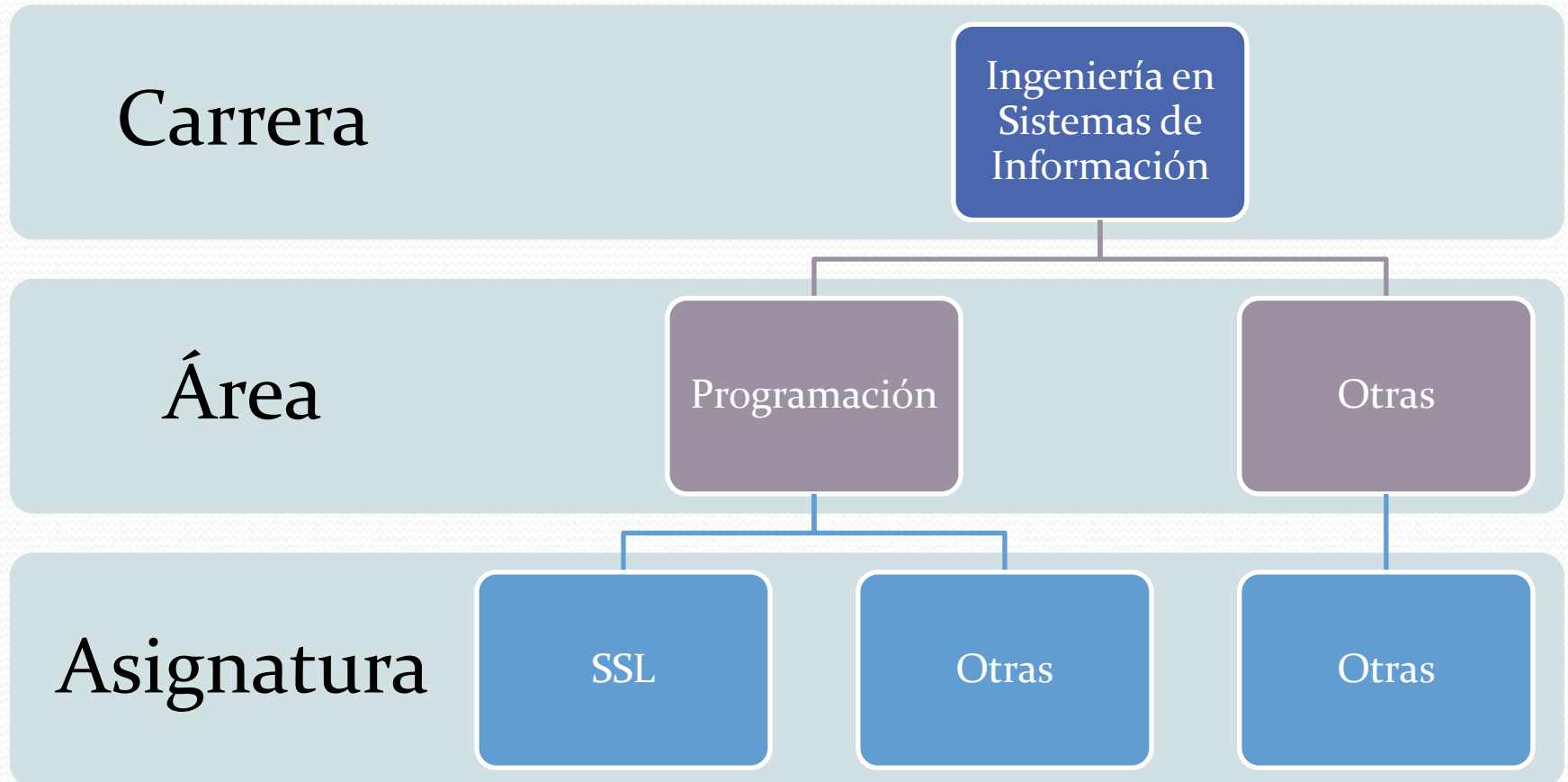
- Contenidos Mínimos

- 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.

Repaso de Conceptos de Asignaturas Anteriores

- Área de Sistemas de Información
 - Sistemas y Organizaciones (no correlativa)
 - Sistema
 - Organización
 - Información
 - Dato
 - Proceso
 - Área de programación
 - Matemática Discreta (correlativa)
 - Función
 - Autómata
 - Lógica
 - Números aleatorios
 - Grafos
- Algoritmos y Estructura de Datos (correlativa)
 - Algoritmo
 - Dato
 - Estructura de Datos
 - Función
 - Programa
 - Lenguaje
 - Proceso
 - Procedimiento
 - Parámetro
 - Argumento.

SSL 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).

Objetivos de cada Asignatura del Área

- MD, AyEdD, SSL
- Paradigmas de Programación
 - Comprender los fundamentos de los paradigmas de programación básicos que son utilizados por los lenguajes de programación actuales
 - Conocer el modelo formal o semiformal subyacente de cada paradigma y la forma en que el mismo es incorporado en un lenguaje de programación concreto.
 - Aplicar los diferentes paradigmas en la resolución de problemas
- Gestión de Datos
 - Desarrollar los conceptos de estructuración de los datos en dispositivos de almacenamiento.
 - Describir metodologías para el modelado de datos.
 - Conocer modelos actuales para la persistencia de grandes volúmenes de datos.
 - Desarrollar los conceptos relacionados con la consistencia, integridad y seguridad de la información. Aplicar técnicas y métodos para el tratamiento concurrente de los datos.

Integración con otras Asignaturas y Conceptos Principales

- Integración en Área de Programación
 - Integración vertical
 - Matemática Discreta
 - Algoritmos y Estructuras de Datos
 - Gestión de Datos
 - Integración Horizontal
 - Paradigmas de Programación
- Integración con otras áreas
 - Computación
 - Sistemas de Información
- Abstracción (concepto fundamental)
 - Separación, dejar de lado los detalles para enfocar en lo importante
- Orientación a Objetos
 - Objeto: entidad con comportamiento y que mantiene un estado.

Sobre la Cátedra y la Asignatura

Intervalo

20 minutos

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**
 - **Matemática Discreta**
 - Siguietes
 - 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

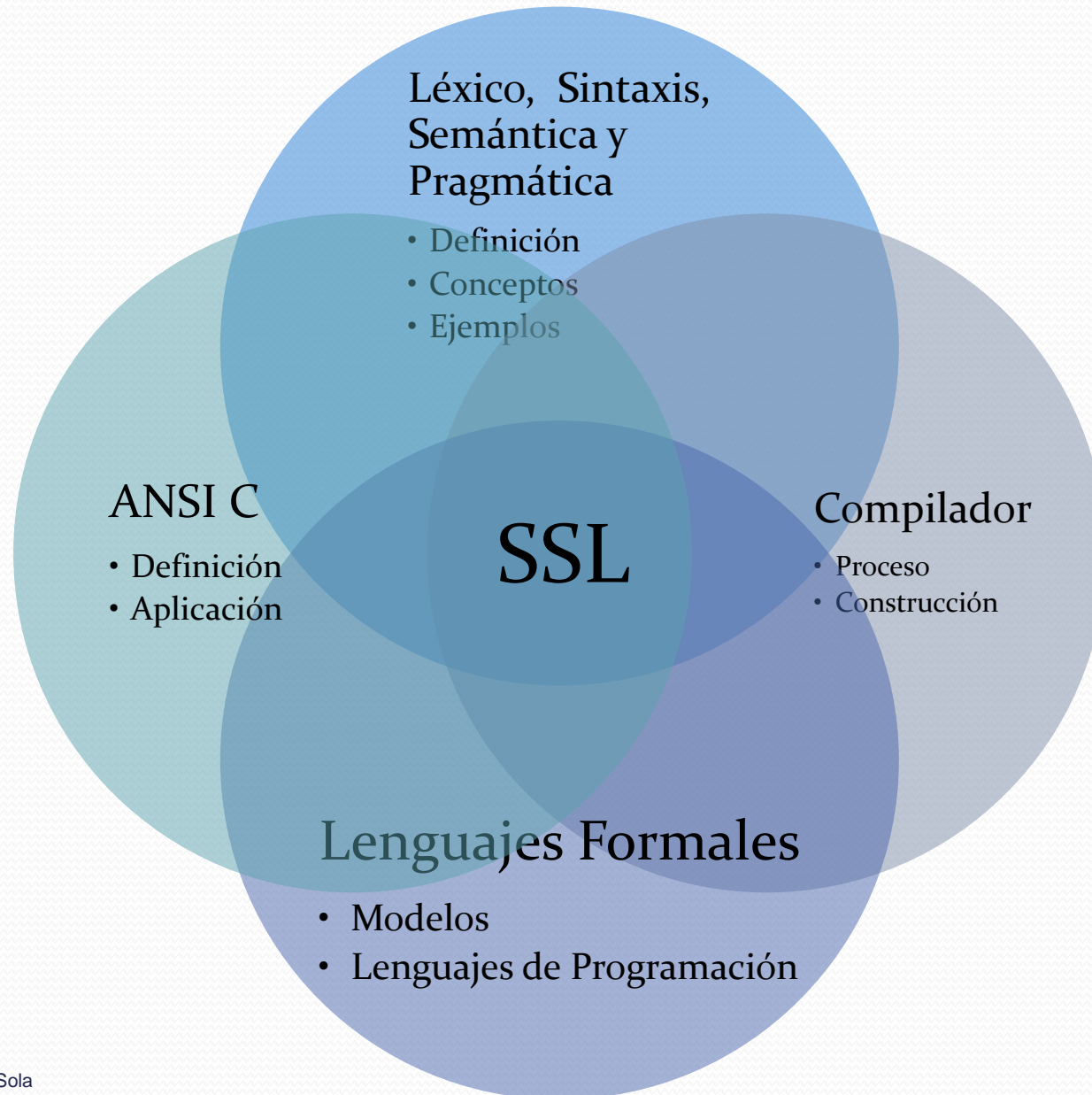
Objetivos

- 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**
 - **Lic. Ricardo Barca**
 - **Ing. Adrian Bender**
 - **Dr. Mgs. Oscar Ricardo Bruno**
 - **Ing. Santiago Ferreiros**
 - **Ing. Pablo Méndez**
 - **Ing. Silvina Ortega**
Profesora Adjunta (Carrera Académica)
 - **Mgs. Ing. Gabriela Sanromán**
 - **Esp. Ing. José María Sola**
 - **Lic. Eduardo Zúñiga**
- **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)

La Aprobación y la Regularización

Aprobación – Examen Final

- Requisito para aprobación
 - Regularización y aprobación del examen final
 - Aprobación Directa
- Examen final
 - Luego de un ciclo lectivo
 - Correlativas aprobadas
 - Algoritmos y Estructuras de Datos
 - Matemática Discreta
 - Cuatro oportunidades

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

- 75% de Asistencia (Bedelía)
- Evaluaciones (Cátedra)
 - Características
 - Grupales e individuales
 - De aplicación o conceptuales
 - Evaluación frecuentes
 - Trabajos Prácticos, entregas semanales
- Instancias
 - Una evaluación escrita individual en el primer cuatrimestre, otra en el segundo
 - Trabajos prácticos
 - Comunes a todos los cursos
 - Propios del curso
 - Dos recuperatorios, en Dic y en Feb
 - Las fechas se establecen durante el curso
 - Al Finalizar el curso
 - Evaluación individual oral o escrita ("coloquio")
 - Conocimientos de los trabajos prácticos.

Aprobación Directa

- No haber sido reincorporado
- Todo aprobado y, para las evaluaciones con nota numérica, con nota mayor o igual a 8
- No haber recuperado ningún parcial o sólo un examen parcial.
- Se debe cumplir en tiempo y forma con todas las entregas de los TP de comunes a todos los curso
- Los TP del curso se aprueban en la primera oportunidad o en el primer recuperatorio
- Hay dos instancias de determinación del estado de Aprobación Directa:
 - En Noviembre, antes de finalizar la cursada
 - Durante Diciembre
- Quienes lleguen a los segundos recuperatorios de Febrero-Marzo, no logran la aprobación directa
- Se trata como caso especial quien tenga una nota con 7(siete) y cumple con el resto de los requisitos.

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 en Yahoo Groups.

Plataforma del Curso

Herramientas para facilitar la cursada

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>

Home Mail Search News Sports Finance Weather Games Answers Screen Flickr Mobile More ▾

YAHOO!
GROUPS

Search Conversations Search Groups Search Web

Sign In Mail

Use Groups

Privacy
Feedback
Blog

UTN FRBA SSL
Restricted Group, 1972 members

Conversations Photos Files **About** More ▾

About Group [+ Join Group](#)

No activity in last 7 days

Group Description

UTN FRBA SSL

Grupo Yahoo de la Cátedra de

Sintaxis y Semántica de los
Lenguajes(SSL)

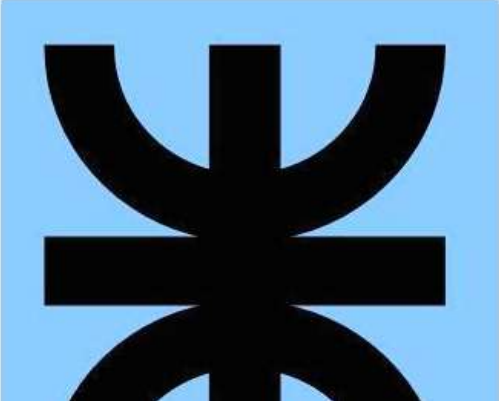
de la especialidad

Ingeniería en Sistemas de Información

en la

Facultad Regional Buenos Aires

UTN FRBA



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) 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 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-ssl**
 - Para los miembros del curso asignamos un 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://github.com/utn-frba-aed/Papers/raw/master/Git 101.pdf](https://github.com/utn-frba-aed/Papers/raw/master/Git%20101.pdf)
- <https://guides.github.com/activities/hello-world/>

josemariasola.wordpress.com

- Homepage del Curso
- Papers
- Referencia
- Tareas
- Equipos
- Calendario
- Resúmenes de Clases
- Acceso a Foro
- Acceso a GitHub.

Plataforma del Curso

josemariasola.wordpress.com

Homepage del curso

GitHub

*Papers y
Resoluciones de
cada equipo*

**Clases
Presenciales**

Yahoo Groups UTNFRBASSL

Foro para consultas y anuncios

Google Calendar

Caledario con detalle de actividades

Primer Contacto con el Compilador

Lenguajes y Herramientas de Desarrollo

¿Qué es un Compilador?

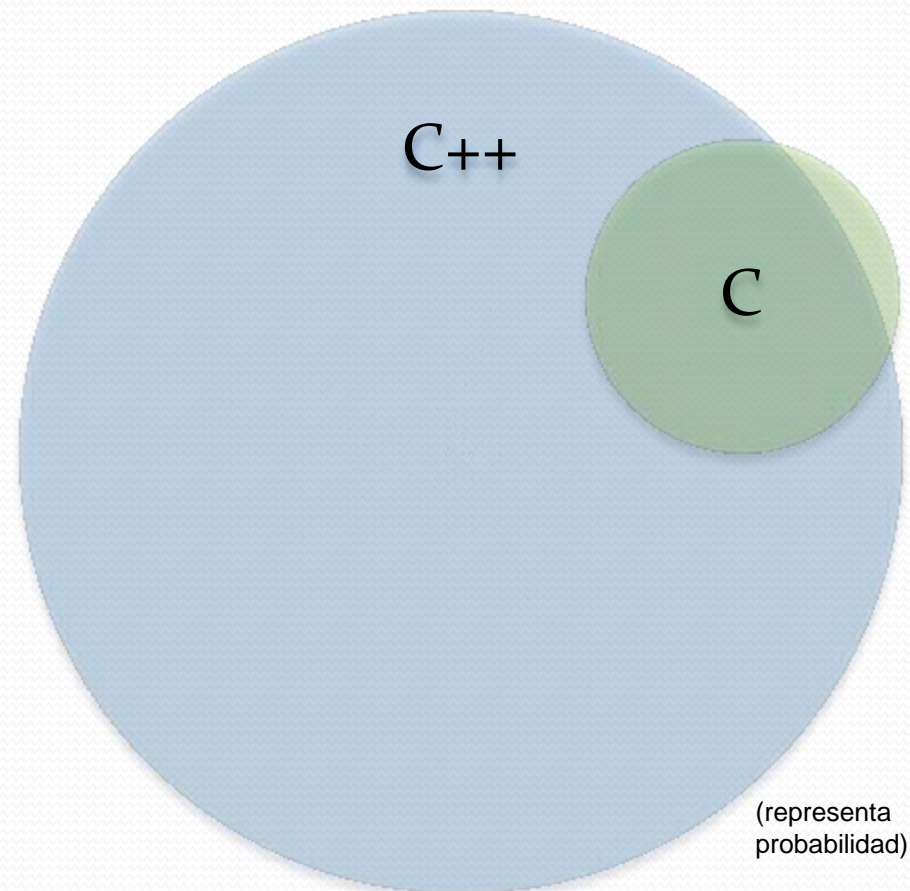
- Programa que hace programas, un meta programa
- Traductor
- Función de Lenguaje a Lenguaje
- Proceso, en etapas
- Baja de Nivel

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
 - Standard C++11, 14, 17

Conjuntos de infinitos programas válidos de C++ y C



(representa probabilidad)

"Hello, World!"

```
/* Hello world
 * JMS
 * 20150402
 */

#include <stdio.h>

int main(void){
    printf("Hello, world!\n");
}
```

- 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 .c (e.g., hello.c)
2. **Compilar** el archivo fuente para producir el programa objeto (e.g., cc hello.c)
3. **Vincular** (link) el programa con las bibliotecas para crear el programa ejecutable; generalmente ocurre junto con el punto anterior.
4. **Ejecutar** el programa (e.g., hello.exe ó ./a.out)

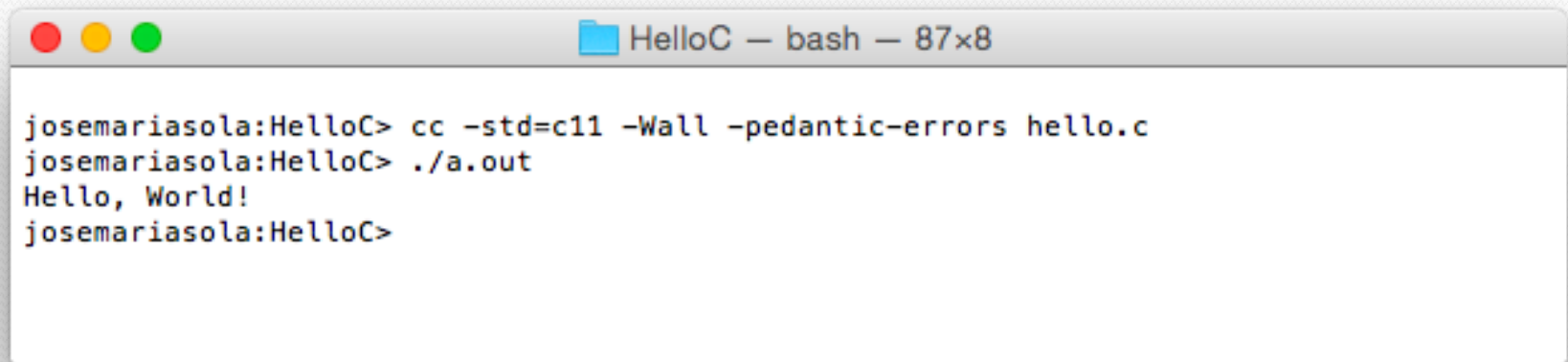
Ejemplo desde línea de comando

Mac OS X C11

1. Desde la línea de comando
 1. `> vi hello.c` crear el fuente
 2. `> cc -std=c11 -Wall -pedantic-errors hello.c`
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 clang es:

`> cc -std=c11 -Weverything hello.c`

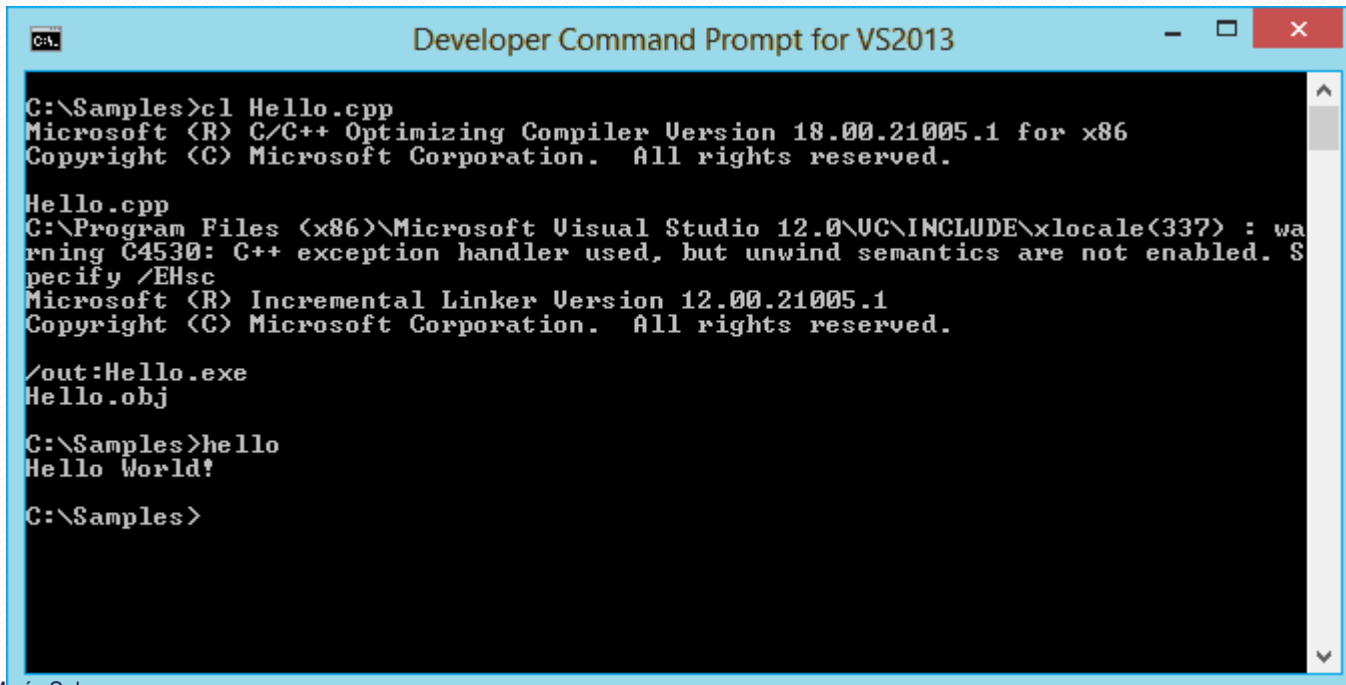


```
josemariasola:HelloC> cc -std=c11 -Wall -pedantic-errors hello.c
josemariasola:HelloC> ./a.out
Hello, World!
josemariasola:HelloC>
```

Ejemplo desde línea de comando

Compilador Microsoft

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



```
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



```
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 C++ Express Edition
 - Eclipse
- Sin IDE
 - Editor
 - Compilador.

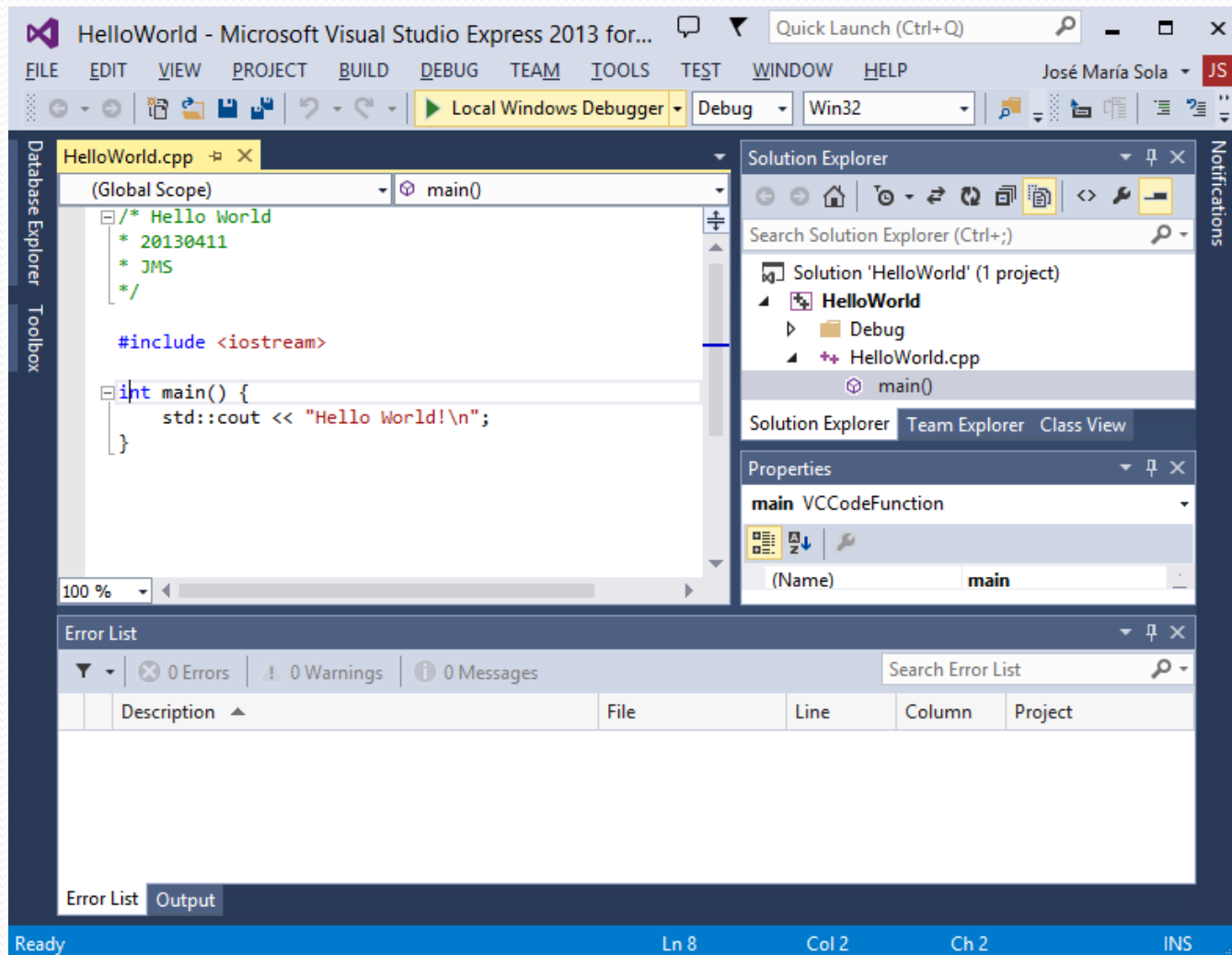
Con IDE

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

Sin IDE
Requiere editor

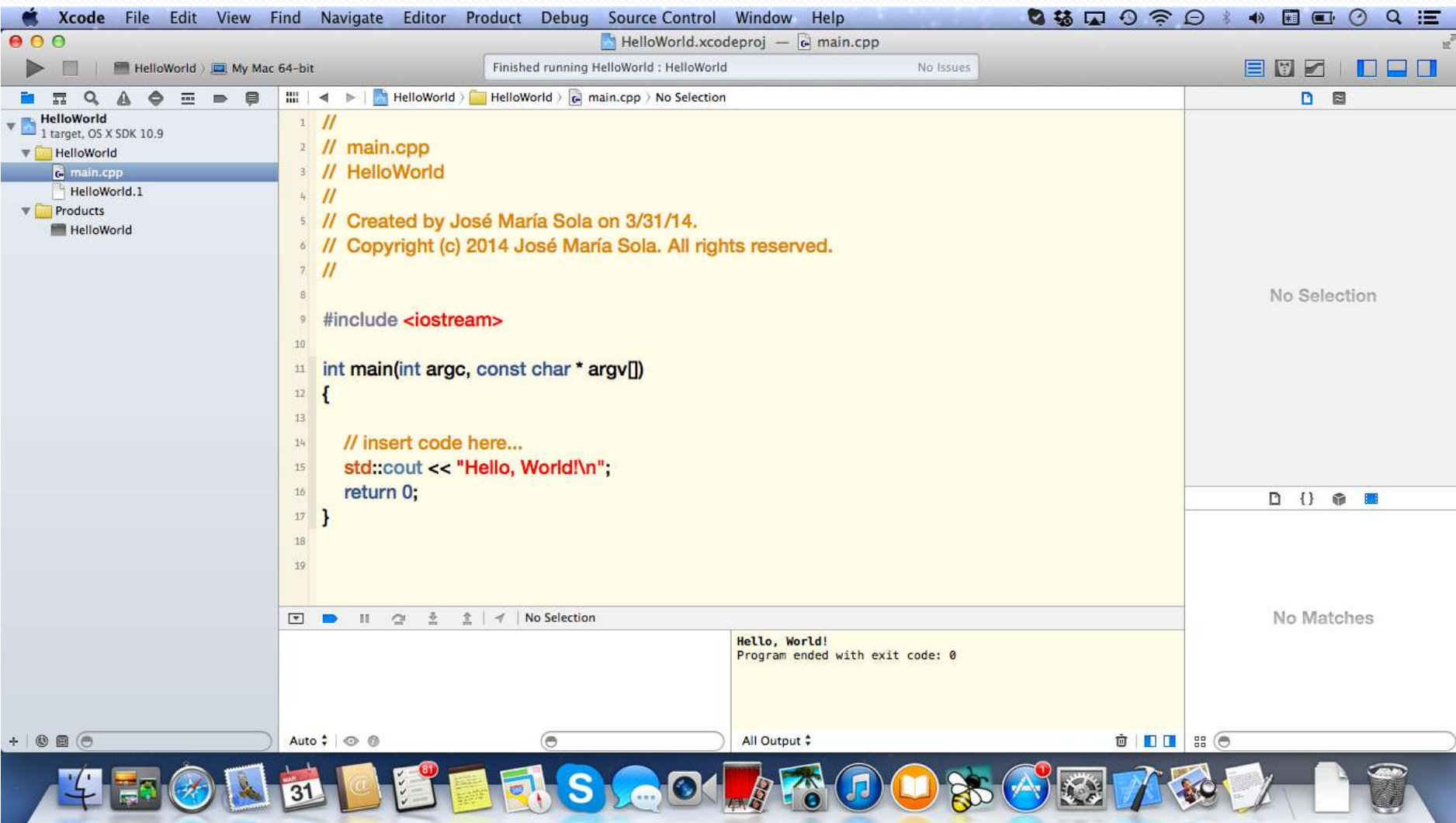
Compilador de C/C++

Ejemplo con IDE Microsoft Visual Studio Express for Windows Desktop



Ejemplo con IDE

Apple Xcode



Links a Compiladores C/C++

- Con IDE y Línea de Comandos
 - Microsoft Visual Studio Express 2015 for Windows Desktop
 - <https://www.visualstudio.com/downloads/#d-express-windows-desktop>
 - Apple Xcode
 - <https://developer.apple.com/xcode>
 - Dev-C++
 - <http://orwelldevcpp.blogspot.com>
 - Eclipse IDE for C/C++ Developers
 - <http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/neon2>
 - Code::Blocks
 - <http://www.codeblocks.org/downloads/>
- Sin IDE, solo Línea de Comandos
 - GNU C Compiler (ahora GNU Compiler Collection)
 - <http://gcc.gnu.org/install/binaries.html>
 - Free C++ Compiler
 - <https://www.embarcadero.com/free-tools/ccompiler>
 - 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>

Primer Trabajo Práctico:

TP #0 – Hola Mundo

- Secuencias de Tarea

- 1. Inscribirse al Grupo Yahoo
 - 2. Seleccionar e instalar compilador C11
 - 3. Probar compilador con el programa Hello World
- Todo código fuente debe comenzar con el comentario

```
/* Qué  
 * Quién  
 * Cuándo  
 */
```

Reemplazar quién por el nombre completo del autor o autores , cuándo con la fecha última modificación, y qué con una breve descripción del programa.

- 4. Ejecutar el programa, y capturar su salida en un archivo de texto
 - 5. Registrarse y publicar hello.c y salida.txt en GitHub
 - 6. Enviar al grupo Yahoo usuario y repositorio GitHub
- Restricciones
 - La fecha límite de entrega es el día de la segunda clase a las 13:00 hs
 - La evaluación se hace con lo publicado en GitHub.

Tareas para la próxima clase

1. Solicitar inscripción al Grupo Yahoo
2. Crear usuario GitHub
3. Leer 1.1 de [K&R1988]
4. Realizar TP#0 "Hello, World!"
 - TP obligatorio para quienes no tienen conocimiento de programación en C o C++.

Términos de la clase #01

Definir cada término con la bibliografía

- Programación
- Programa
- Lenguaje de Programación
- Comportamiento
- Software
- Ingeniería
- Sistema
- Información
- Dato
- Estructura de Datos
- Ingeniería en Sistemas de Información
- Sistema de Información
- Proyecto
- Problema (Necesidad)
- Solución
- Desarrollo
- Análisis
- Diseño
- Implementación
- Organización
- Abstracción
- Orientación a Objetos
- Lenguaje
- Lenguaje Natural
- Lenguaje Formal
- Sintaxis
- Semántica
- Léxico
- Pragmática
- SCV (Sistema de Control de Versiones)
- Git
- GitHub
- Repositorio
- Compilador
- Proceso
- Función del compilador
- Lenguaje máquina (bajo nivel de abstracción)
- Lenguaje de Alto Nivel de Abstracción
- C
- C++
- Hello World (Kernighan)
- IDE (Integrated Development Enviroment, Entorno Intedrado de Desarrollo).

¿Consultas?

Fin de la clase