

# Sintaxis y Semántica de los Lenguajes SSL

*K2051*

*2018*

*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 19, Lunes*

# Agenda para esta clase

- Bienvenidos a Segundo año y al Curso
- Presentación inicial, conozcámonos
- El contexto de SSL
- Intervalo
- Sobre la Cátedra y la Asignatura
- La aprobación y la regularización
- Plataforma del Curso
- Primer contacto con el compilador
- Trabajo #0

# Presentación Inicial

Conozcámonos

# Docentes del Curso

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

# 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



# Integración Vertical: Asignaturas Anteriores

- Algoritmos y Estructura de Datos
- Matemática Discreta
- Sistemas y Organizaciones (no correlativa)

# 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.

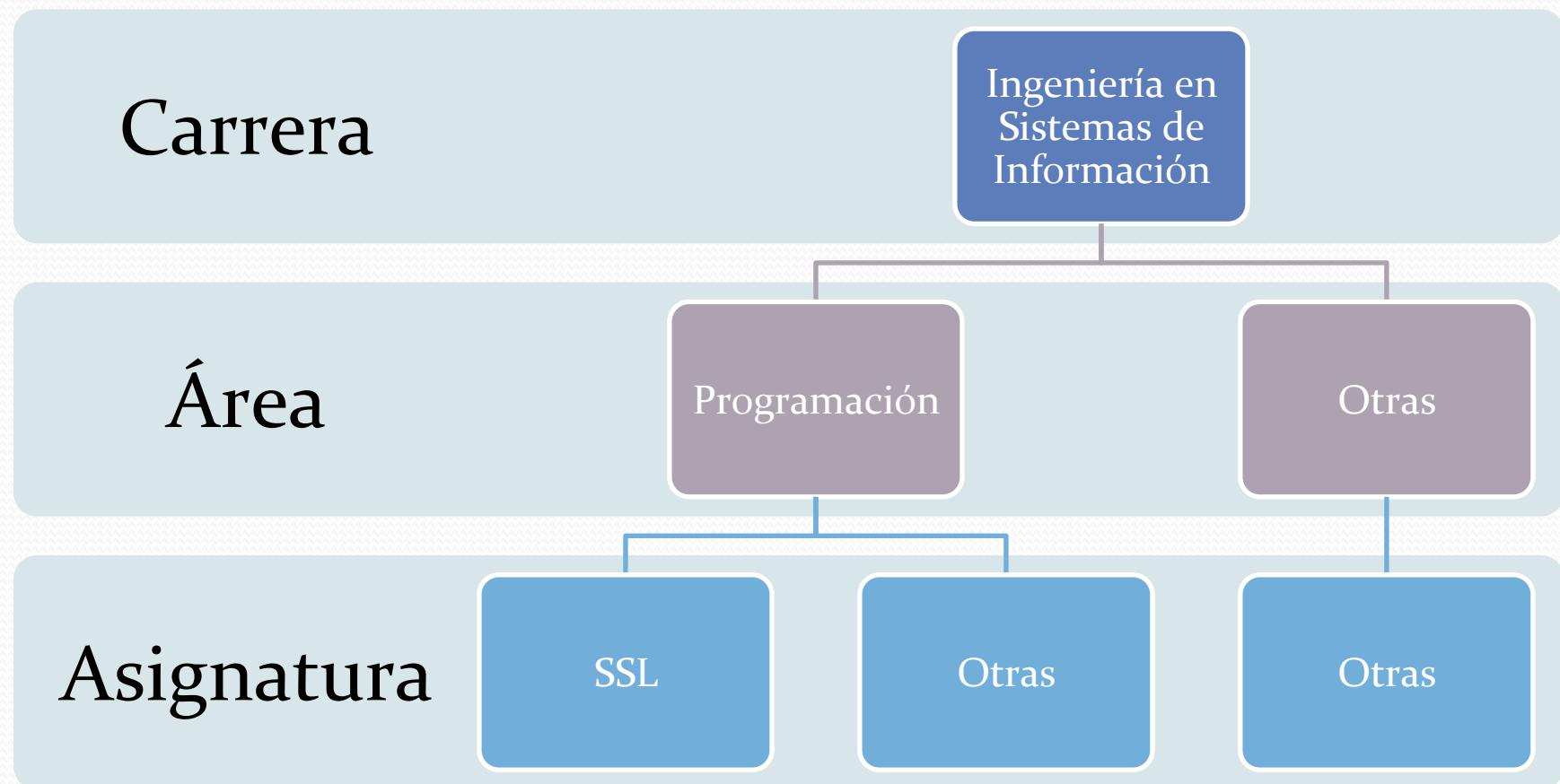
# Algoritmos y Estructuras de Datos

- 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.

# 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 vertical: Anteriores
    - Área de Programación
      - Matemática Discreta
      - Algoritmos y Estructuras de Datos
  - Integración Horizontal: Paralelas
    - Área de Programación
      - Paradigmas de Programación
    - Computación
      - Sistemas Operativos
    - Sistemas de Información
      - Sistemas y Organización
      - Análisis de Sistemas
      - Diseño de Sistemas
  - Integración vertical: Posteriore
    - Área de Programación
      - Gestión de Datos
      - Ingeniería en Software
- 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.



# Intervalo

20 minutos

# 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

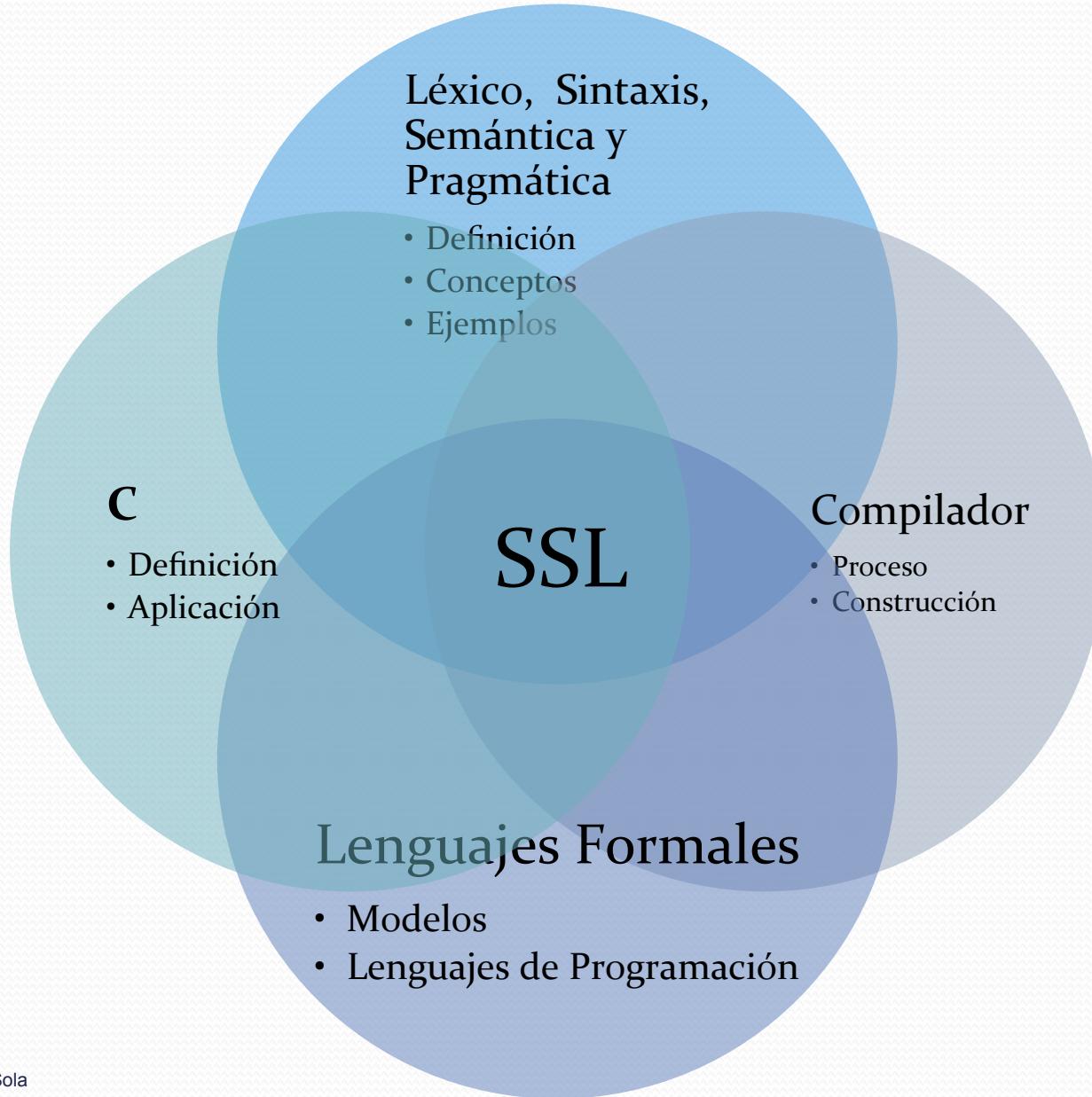
## Objetivos

- 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
  - 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)*

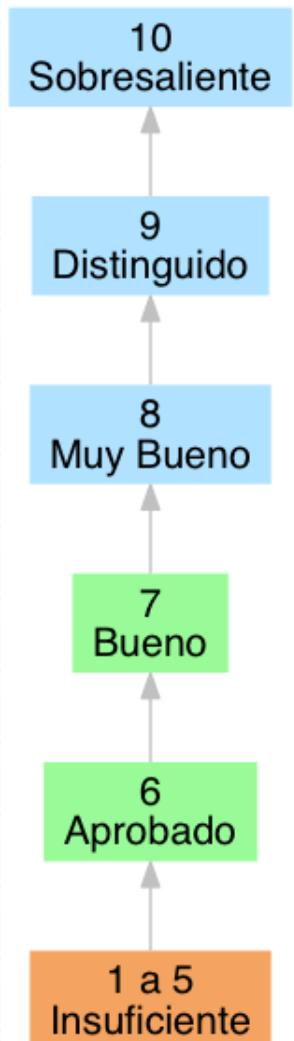
# La Aprobación y la Regularización

# Dinámica de Trabajo

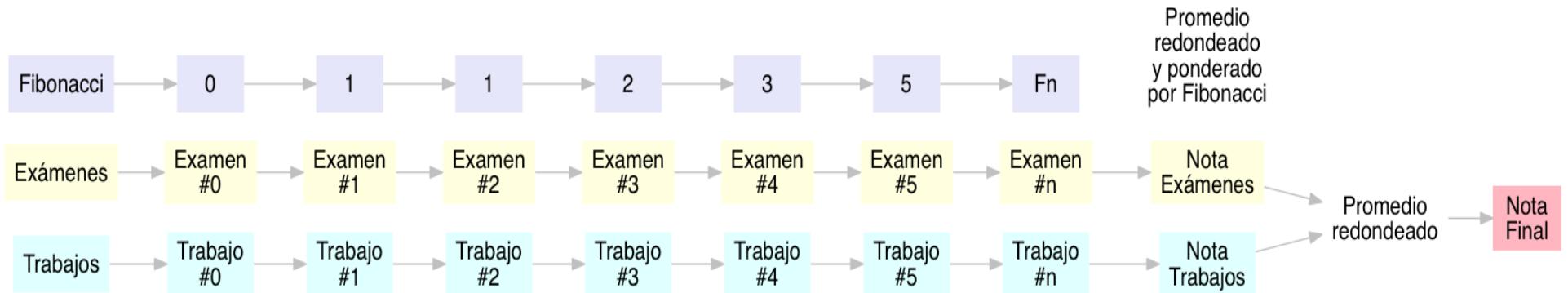
- Indicación de lecturas y ejercicios obligatorios para siguiente clase
- Profundización de conceptos en clase
- Clase interactiva y participativa
  - Se recomienda seguir la clase con la bibliografía y tomar apuntes a la par
- Ejercicios de aplicación en clase
- Consultas
  - Durante clase, y justo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

# Aprobación – Examen Final

- Requisito para aprobación
  - Regularización y aprobación del examen final  
ó
  - 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)



- 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**.

# Aprobación Directa

- **No haber sido reincorporado**
- **Nota final mayor o igual a 8 (muy bueno)**
- No haber recuperado ningún parcial o sólo un examen parcial.
- Se debe cumplir en tiempo y forma con todos los trabajos
- 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.

# Plataforma del Curso

Herramientas para facilitar la cursada

# Plataforma del Curso

**Clases  
Presenciales**

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

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

**GitHub**  
*Resoluciones  
Individuales y del  
cada equipo*

**Bibliografía**  
• *K&R1988*  
• *MUCH2012*

**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 interface for the group "UTN FRBA SSL". The top navigation bar includes links for Home, Mail, Search, News, Sports, Finance, Weather, Games, Answers, Screen, Flickr, Mobile, and More. The main search bar has fields for "Search Conversations", "Search Groups", and "Search Web". On the right, there are "Sign In", "Mail", and a gear icon for settings.

The group's banner image features a stack of books. The group name "UTN FRBA SSL" and description ("Restricted Group, 1972 members") are displayed above the banner. Below the banner, there are tabs for Conversations, Photos, Files, About (which is selected), and More.

The "About Group" section shows "No activity in last 7 days" 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

ICDDA

A large black logo consisting of three interlocking arches or stylized letters is displayed on the right side of the group description area.

# 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 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 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/>

# Primer Contacto con el Compilador

Lenguajes y Herramientas de Desarrollo

# ¿Qué es un Compilador?

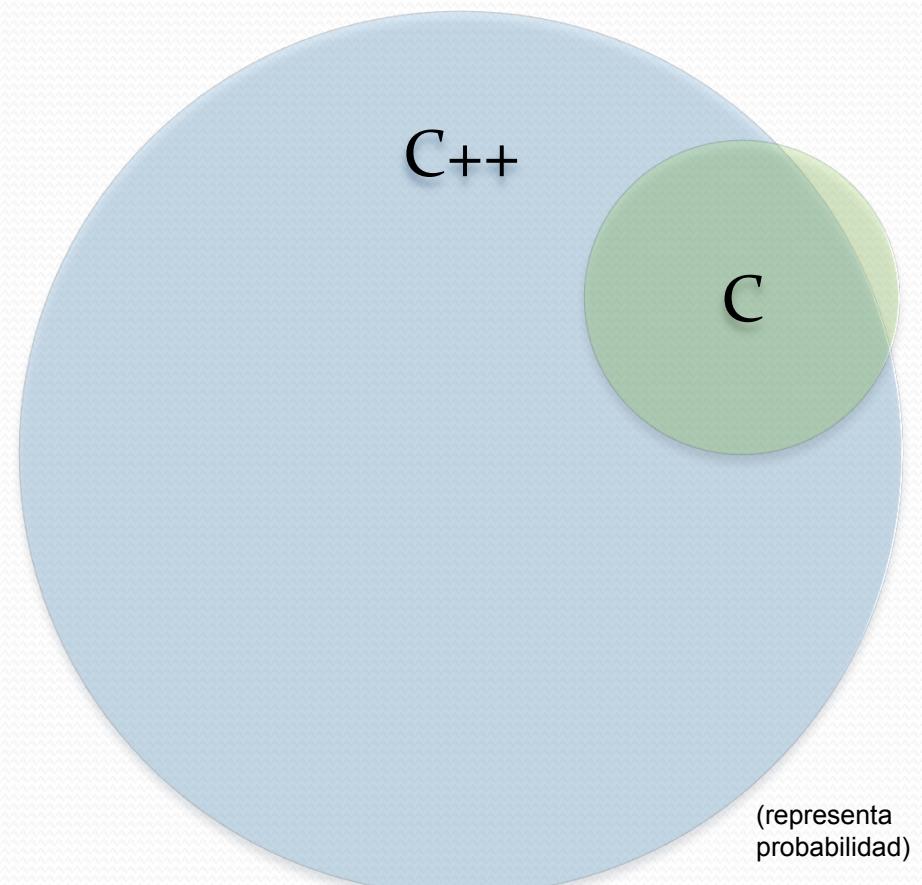
- Programa que hace programas, un meta programa
- Traductor
- Función de Lenguaje a Lenguaje
- Baja de Nivel
- Proceso, en etapas.

# 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**



# "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

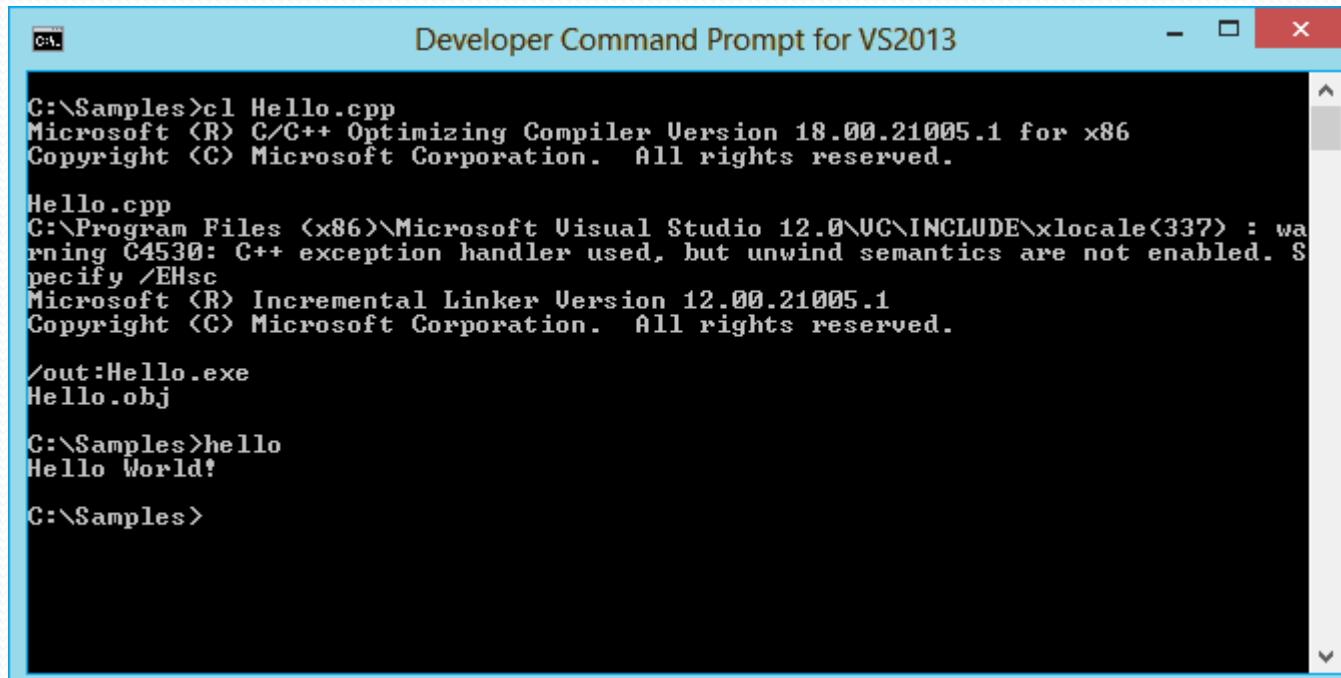
Otra versión para gcc es:

> **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 terminal window titled "Developer Command Prompt for VS2013". The command line shows the following sequence of commands and their output:

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

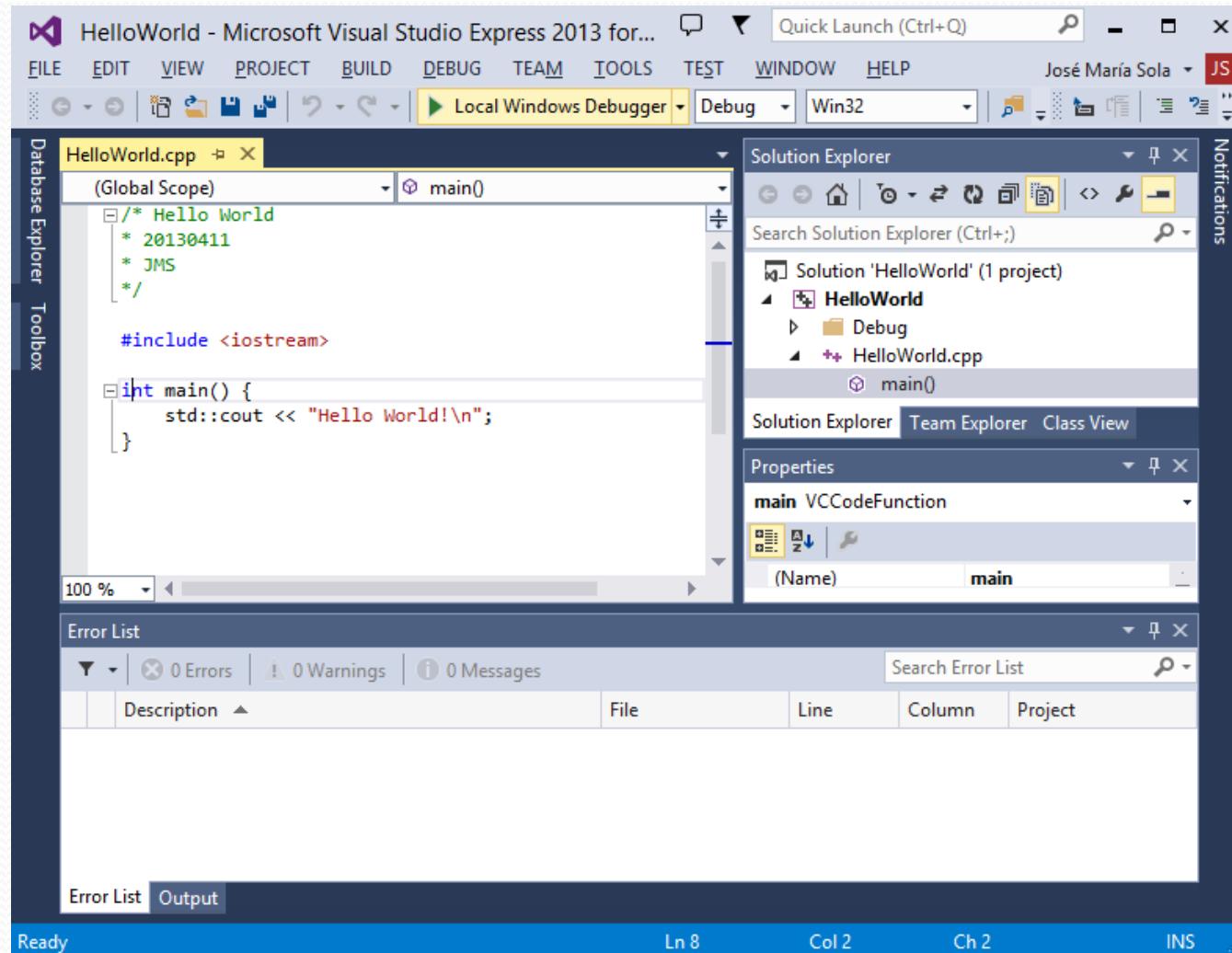
## Con IDE

- 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 en C++, no C)



# Ejemplo con IDE Apple Xcode (ejemplo en C++, no C)

The screenshot shows the Apple Xcode IDE interface. The top menu bar includes Xcode, File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, and Help. The title bar displays "HelloWorld.xcodeproj — main.cpp". A status bar at the top right indicates "Finished running HelloWorld : HelloWorld" and "No Issues".

The left sidebar shows the project structure:

- HelloWorld (target, OS X SDK 10.9)
- HelloWorld (group)
  - main.cpp (selected)
  - HelloWorld.1
- Products

The main editor area contains the following C++ code:

```
//  
// main.cpp  
// HelloWorld  
//  
// Created by José María Sola on 3/31/14.  
// Copyright (c) 2014 José María Sola. All rights reserved.  
//  
#include <iostream>  
  
int main(int argc, const char * argv[])  
{  
  
    // insert code here...  
    std::cout << "Hello, World!\n";  
    return 0;  
}
```

The bottom output window shows the console output:

```
Hello, World!  
Program ended with exit code: 0
```

The Dock at the bottom of the screen contains icons for various Mac OS X applications.

# Links a Compiladores C/C++

Usar unos de estos compiladores o cualquier otro, siempre y cuando se lo configure para **C11**

- Con IDE y Línea de Comandos
  - Microsoft Visual Studio Community 2017
    - <https://www.visualstudio.com/vs/features/cplusplus/>
  - Apple Xcode
    - <https://developer.apple.com/xcode>
  - CodeLite
    - <https://codelite.org>
  - Eclipse IDE for C/C++ Developers
    - <http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/oxygen3>
  - Code::Blocks
    - <http://www.codeblocks.org/downloads/>
  - Dev-C++
    - <http://orwelldevcpp.blogspot.com>
- Sin IDE, solo 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>

# Trabajo #0

hello.c: "Hello, World!"  
\_

# Trabajo #0 — "Hello, World!" en C

- Enunciado en [josemariasola.wordpress.com](http://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
  - 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](mailto: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.

# Términos de la clase #01

## Definir cada término con la bibliografía

- Contexto
  - Ingeniería
  - Sistema
  - Información
  - Sistema de Información
  - Ingeniería en Sistemas de Información
  - Software
  - Programación
  - Problema (Necesidad)
  - Solución
  - Análisis
  - Diseño
  - Implementación
  - Desarrollo
  - Proyecto
  - Organización
  - Abstracción
  - Tipo de Dato
  - Orientación a Objetos
- 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 Integrado de Desarrollo).

# Tareas para la próxima clase

1. Leer de [K&R1988] *desde la tapa hasta la sección **1.1 Comenzado** inclusive*
2. Comenzar Trabajo #o "Hello, World!".



# ¿Consultas?



**Fin de la clase**