

# Clase #04 de 27

## Introducción al Proceso de Compilación

*Abril 20, Lunes*

# Agenda para esta clase

- Repaso Trabajo #0
- Repaso Trabajo #1
- Sobre la Cátedra y la Asignatura
- Plataforma del Curso
- Intervalo
- “El Lenguaje de Programación C” aka “El Libro Blanco” aka “K&R”
- El Proceso de Compilación, versión simplificada
- Otras versiones de hello.c
- El Preprocesador
- Sintaxis, Semántica, y Pragmática
- Trabajo #2: Niveles del Lenguaje: Hello.cpp v Hello.c

# Repaso Trabajo #0

§3. “Hello, World” en C

# Repaso Trabajo #1

§4. Uso del Lenguaje C en mi Día a Día

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

# Programa Sintético de SSL

Ordenanza 1150/2007

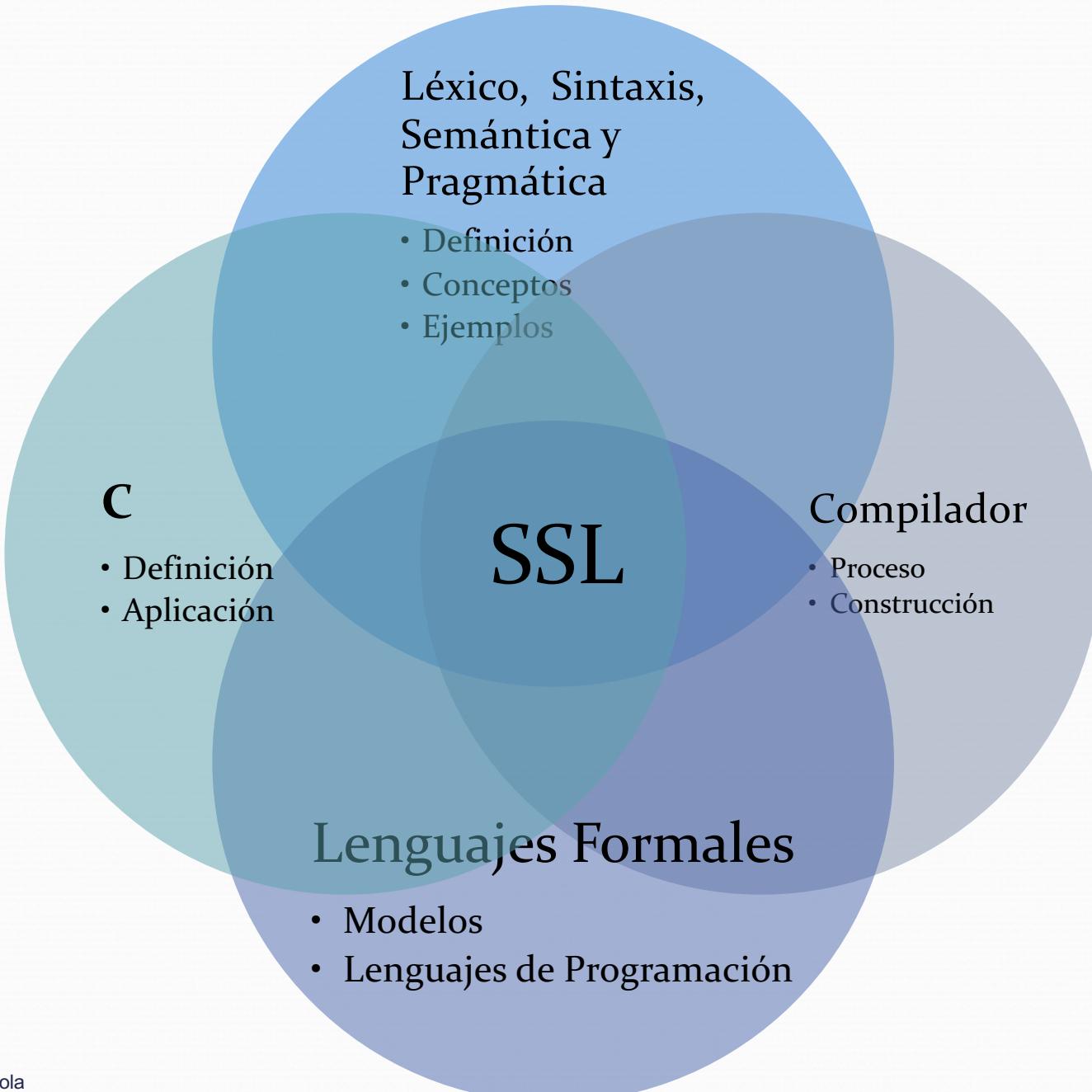
## Objetivos

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

# (Distancia) Plataforma del Curso

- Tipos de Clases
  - **Sincrónica:** Videoconferencias
  - **Asincrónica:** Lecturas y videos
- Herramientas
  - **Campus Virtual:** Centralizador de actividades, anuncios, y consultas
  - **josemariasola.wordpress.com:** Homepage de la asignatura y del curso, material y clases asincrónicas
  - **Google Calendar:** Caledario de clases, actividades y deadlines
  - **Google Meet:** Clases sincrónicas a distancia
  - **YouTube:** Clases asincrónicas
  - **GitHub:** Resoluciones individuales y del cada equipo
  - (Opcional) **Yahoo Groups UTNFRBASSL:** Foro para consultas y anuncios

# josemariasola.wordpress.com

- Homepage del Curso
- Papers
- Referencia
- Enunidades de Trabajos
- Equipos y Notas
- Calendario de clases, éxamenes, deadlines, actividades
- Algunas Slides de Clases
- Algunas Videos de Clases
- Acceso a GitHub.

# Bibliografía

- Bibliografía obligatoria
  - [K&R1988]
    - Biblioteca, Fotocopiadora, y Librerías
  - [MUCH2010], tres volúmenes
    - Homepage.
  - [MUCH2012], tres volúmenes
    - Biblioteca, y Librería del Centro de Estudiantes
- Otros artículos a lo largo del curso, publicados en  
<http://josemariasola.wordpress.com>
- Prioritarios para primera parte del curso
  - Capítulo #1 de [K&R1988]
  - Volumen #1 de [MUCH201X].

# 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

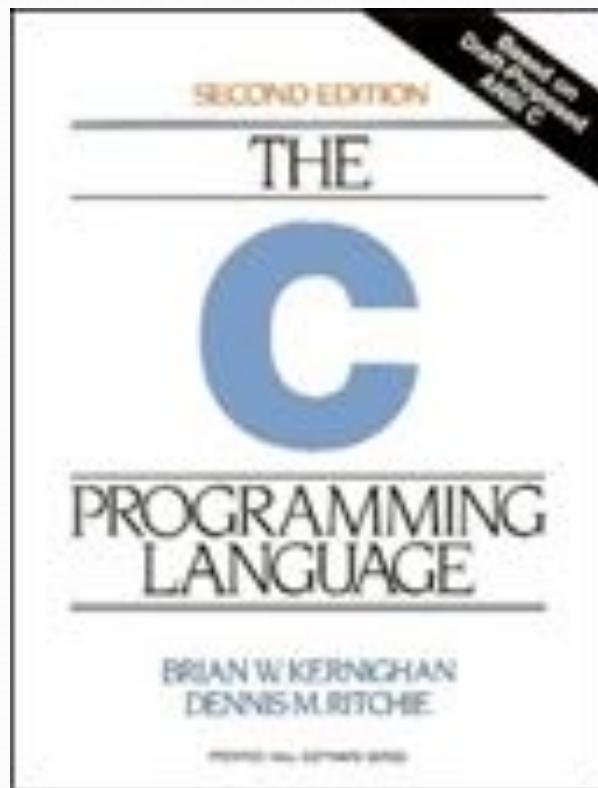
## 10 minutos

# K&R – Capítulo 1

K&R 1.1 Una Introducción Tipo Tutorial – Empezando

# “El Lenguaje de Programación C” aka “El Libro Blanco” aka “K&R”

The C Programming Language, 2nd Edition									
Prefacios e Introducción	#2 Tipos y Expresiones	#3 Control de Flujo	#4 Funciones y Programa	#5 Punteros y Arreglos	#6 Estructuras	#7 Entrada y Salida	Apéndice A Definición del Lenguaje	Apéndice B La Biblioteca Estándar	Apéndice C Resumen de Cambios
<p>#8 Interfaz con Unix Ejemplos de implementación de parte de la Biblioteca</p>									



- Autores:
  - Dennis Ritchie autor del LP y coautor de Unix con Ken Thompson
  - Kernighan
- Requisitos
  - Conocimiento de programación
  - Lectura atenta
- Ejercicios
- Ediciones y usos
  - 1978 1era Edición
    - Manual de referencia
  - 1988 2da Edición
    - ANSI C
- Preliminares (Front Matter)
  - Prefacio
  - Prefacio de la primera edición
  - Introducción
- General, “Ancho”
  - Capítulo 1: Tutorial
- En “profundidad”: Capítulos 2 a 7
  - 2 Tipos, Operadores y Expresiones
  - 3 Control de Flujo
  - 4 Funciones y Estructura de Programa
  - 5 Punteros y Arreglos
  - 6 Estructuras
  - 7 Entrada y Salida
- Ejemplo Integrador
  - 8 Interfaz de Sistema de UNIX – entrada/salida, sistema de archivos y aloación de memoria
- “Apéndices”
  - A: Especificación del LP: Semántica (LN) y Sintaxis (BNF)
  - B: Biblioteca estándar
  - C: Cambios introducidos en la primera versión del estándar.

# "Hello, World!" – Demostración de Compilación y Ejecución

```
#include <stdio.h>
main( ) {
    printf("Hello, world!\n");
}
```

```
$ cc hello.c
```

```
$ ./hello
Hello, world!
```

# Ejercicios

- 1-1. Experimentar con eliminación de partes y compilar
- 1-2. Probar diferentes \c.

# Otras Versiones de Hello.c

# Diferencias con Estándar C (y con C++)

```
main( ){
    puts("Pre Ansi");
}
```

```
#include <stdio.h>

int main(void){
    puts("Post Ansi");
    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>

int main(void){
    puts("Post Ansi");
    return EXIT_SUCCESS;
}
```

```
#include <stdio.h>

int main(void){
    puts("Post Ansi");
}
```

```
#include <iostream>

int main() {
    std::cout << "ANSI C++\n";
}
```

# El Proceso de Compilación, Versión Simplificada

# Hello World

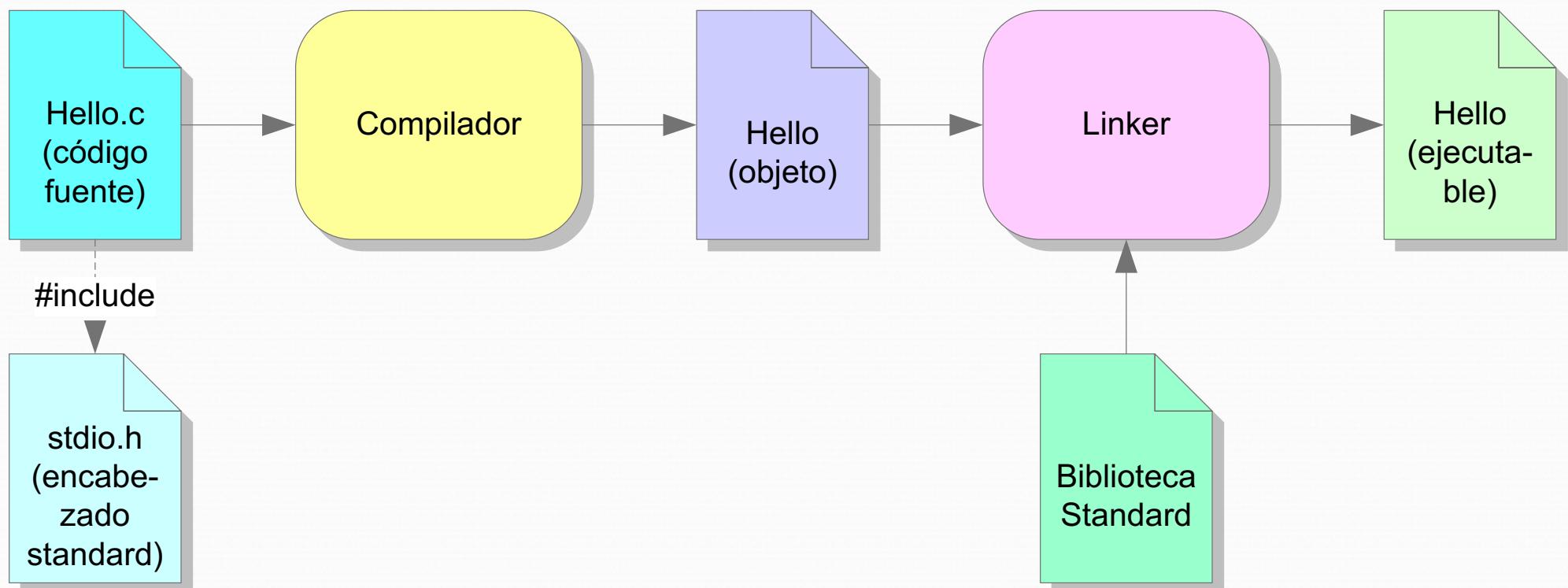
```
/* Hello world
 * JMS
 * 20150402
 */
#include <stdio.h>

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

# Tiempos

- Tiempo de diseño
- Tiempo de traducción (compilación)
  1. Tiempo de Preprocesamiento
  2. Tiempo de Compilación
  3. Tiempo de Vinculación
- Tiempo de ejecución
- Comparaciones
  - Contrastar con Máquina Virtual y Bytecode
  - Contrastar con Lenguajes Interpretados

# El Proceso de Compilación, Versión Simplificada



# El Preprocesador

# Dos Funciones Básicas del Preprocesador

```
#include <_____ .h>
return/*entre*/0;
return0;
return 0;
```

- Incluir archivos, directiva `#include`
- Reemplazar comentarios.

# Léxico, Sintaxis, Semántica, y Pragmática

# Otras versiones – ¿Mismos Léxico, Sintaxis, Semántica, y Pragmática?

```
#include <stdio.h>
main( ) {
    printf( "Hello, world!\n" );
}
```

```
#include <stdio.h>
main( ){
    printf( "Hello," );
    printf( " world!" );
    printf( "\n" );
}
```

```
#include <stdio.h>
main( ){
    printf( "Hello,"
           "world!"
           "\n" );
}
```

```
#include <stdio.h>
main( ){
    puts( "Hello, world!" );
}
```

# Trabajo #2

§5. Niveles del Lenguaje: Hello.cpp v Hello.c

# Niveles del Lenguaje: Hello.cpp v Hello.c

- Enunciado en <https://josemariasola.wordpress.com/ssl/assignments/>
- Objetivos
  - Identificar tecnologías basadas en el Lenguaje C y que usamos en nuestro día a día para estimar el nivel de adopción de C.
- Tareas
  - 1. Listar entre tres y diez tecnologías digitales que usamos en nuestro día a día.
  - 2. Indicar para cada tecnología el repositorio público donde se la desarrolla, si es que lo tiene.
  - 3. Indicar para cada una de esas tecnologías si se desarrollan en C o no.

# Términos de la clase #04

## 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
- K&R Capítulo #1
  - “El Lenguaje de Programación C” aka “El Libro Blanco” aka “K&R”
  - Otras versiones de Hello.c
    - Tipo int implícito
    - Valor returned por main
    - Concatenación de cadenas
    - Múltiples invocaciones
    - puts
    - EXIT\_FAILURE
    - stdlib.h
  - Proceso de Compilación, Versión Simplificada
    - Fuente
    - Compilador
    - Ejecutable
    - Comentarios mínimos en archivos fuente
    - Tiempo de diseño
    - Tiempo de traducción
    - Tiempo de Preprocesamiento
    - Tiempo de Compilación
    - Tiempo de Vinculación (linker)
    - Biblioteca estándar
- Tiempo de ejecución
- Contraste con Framework
- Función del compilador
- Vinculador (Linker)
- Contraste con Máquina virtual
- Bytecode
- Preprocesador
  - #include
  - Tratamiento de comentarios por parte del preprocesador
- Sintaxis, Semántica, y Pragmática
  - Léxico
  - Sintaxis
  - Semántica
  - Pragmática

# Tareas para la próxima clase

1. Conseguir cuenta @frba y vincularla con Campus Virtual
2. Trabajo #2: Niveles del Lenguaje: Hello.cpp v Hello.c  
<https://josemariasola.wordpress.com/ssl/assignments/>
3. Leer “Assembler Fundamental” de  
<https://josemariasola.wordpress.com/ssl/reference/>
4. Leer de [K&R1988] desde **1.2 Variables y Expresiones Aritméticas** hasta **1.4 Constantes Simbólicas** inclusive.



# ¿Consultas?



# Fin de la clase