

# Clase #03 de 27

## Introducción a Léxico, Sintaxis, Semántica y Pragmática

*Abril 19, Lunes*

# Agenda para esta clase

- Sobre la Cátedra y la Asignatura
- Plataforma del Curso
- Intervalo
- Otras versiones de hello.c
- Introducción a Léxico, Sintaxis, Semántica, y Pragmática

# 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)**
  - 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

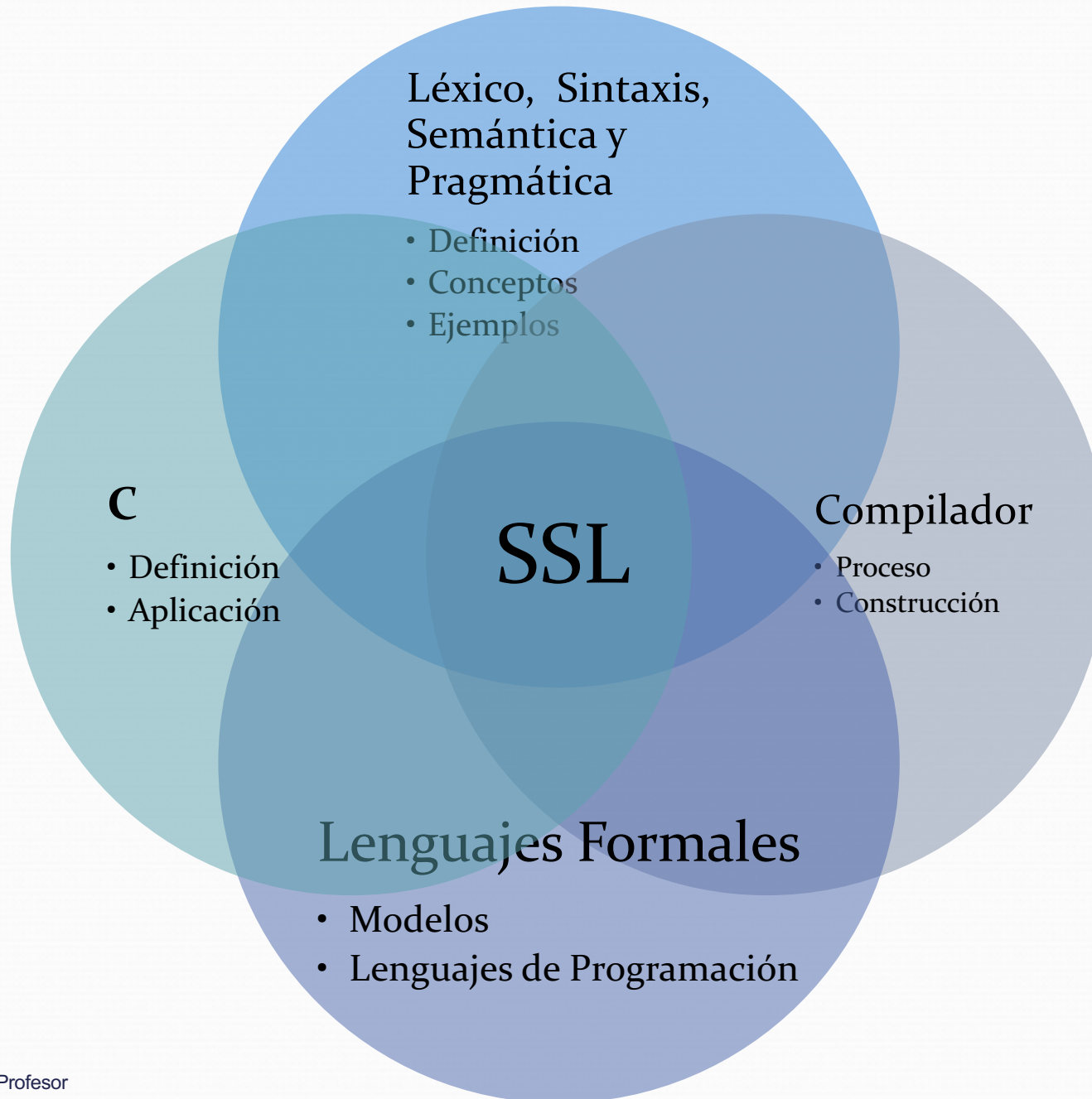
## Objetivos

- 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**
  - **Dr. Mgs. Oscar Ricardo Bruno** (*co-coordinador*)
  - **Esp. Ing. José María Sola** (*co-coordinador*)
  - **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

# Plataforma del Curso

- Tipo de Clases
  - **Sincrónica**, por videoconferencias, con participación y conversaciones sobre los temas
- Herramientas
  - **Aulas Virtuales:**  
Centralizador de actividades, anuncios, y consultas
  - **josemariasola.wordpress.com:**  
Homepage de la asignatura y del curso, material y actividades asincrónicas
  - **Google Calendar:** Caledario de clases, actividades y deadlines
  - **Google Meet:** Clases sincrónicas a distancia
  - **YouTube:** Quizás algún video
  - **GitHub:** Resoluciones Individuales y del cada equipo

# 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].

# josemariasola.wordpress.com

- Homepage del Curso
- Papers
- Referencia
- Enunciados de Trabajos
- Equipos y Notas
- Calendario de clases, exámenes, deadlines, actividades
- Algunas Slides de Clases
- Algunos Videos de Clases.

# 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 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://josemariasola.wordpress.com/ssl/git/>



# Intervalo

15 minutos

# 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";  
}
```

# Introducción a Léxico, Sintaxis, Semántica, y Pragmática

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

```
#include <iostream>

int main() {
    std::cout << "Hello, world\n";
}
```

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

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

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

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



# Términos de la clase #03

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
- Otras versiones de Hello.c
  - Tipo int implícito
  - Valor retornado por main
  - Concatenación de cadenas
  - Múltiples invocaciones
  - Operación corrimiento en C y en C++: <<
  - Operación inserción en C++ <<
  - cout
  - stdout
  - printf versus puts
  - printf versus fprintf
  - puts
  - EXIT\_FAILURE
  - stdlib.h
- Introducción a:
  - Léxico
  - Sintaxis
  - Semántica
  - Pragmática

# Tareas para la próxima clase

1. Leer [Interfaces y Make](#).

# ¿Consultas?