

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
- Términos de la clase
- Tareas para la próxima 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

Carrera

Ingeniería en
Sistemas de
Información

Área

Programación

Otras

Asignatura

SSL

Otras

Otras

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.

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**
 - **Matemática Discreta**
 - 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

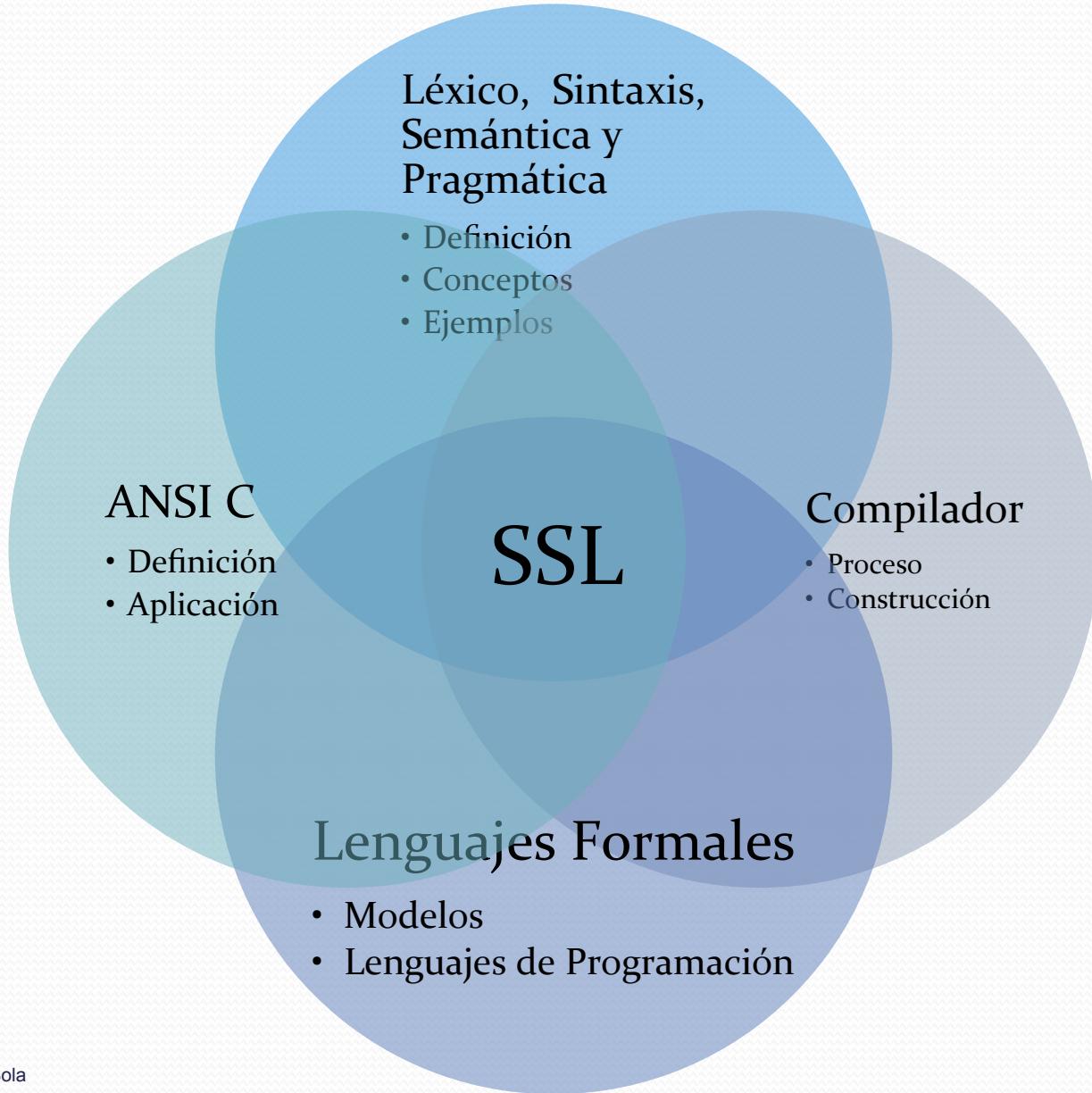
Objetivos

- 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
 - Lic. Ricardo Barca
 - 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
 - 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

Plataforma del Curso

**Clases
Presenciales**

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

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

GitHub
*Papers y
Resoluciones de
cada equipo*

Bibliografía
*K&R1988
MUCH2012*

Google Calendar
*Caledario con detalle
de actividades*

josemariasola.wordpress.com

- Homepage del Curso
- Papers
- Referencia
- Tareas
- Equipos
- Calendario
- Resúmenes de Clases
- 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

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

YAHOO!
GROUPS

Search Conversations

Sign In Mail

Use Groups

Privacy
Feedback
Blog

UTN FRBA SSL
Restricted Group, 1972 members

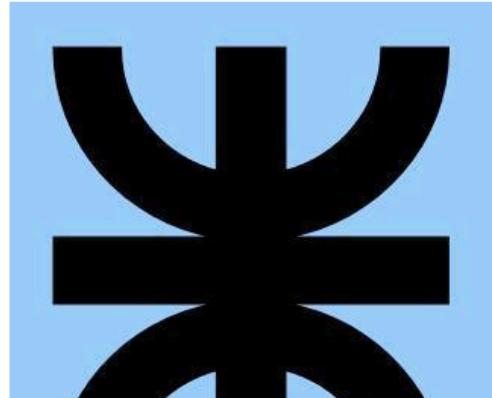
Conversations Photos Files About More ?

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
ICDDA



Prof. Esp. Ing. José María Sola

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 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?

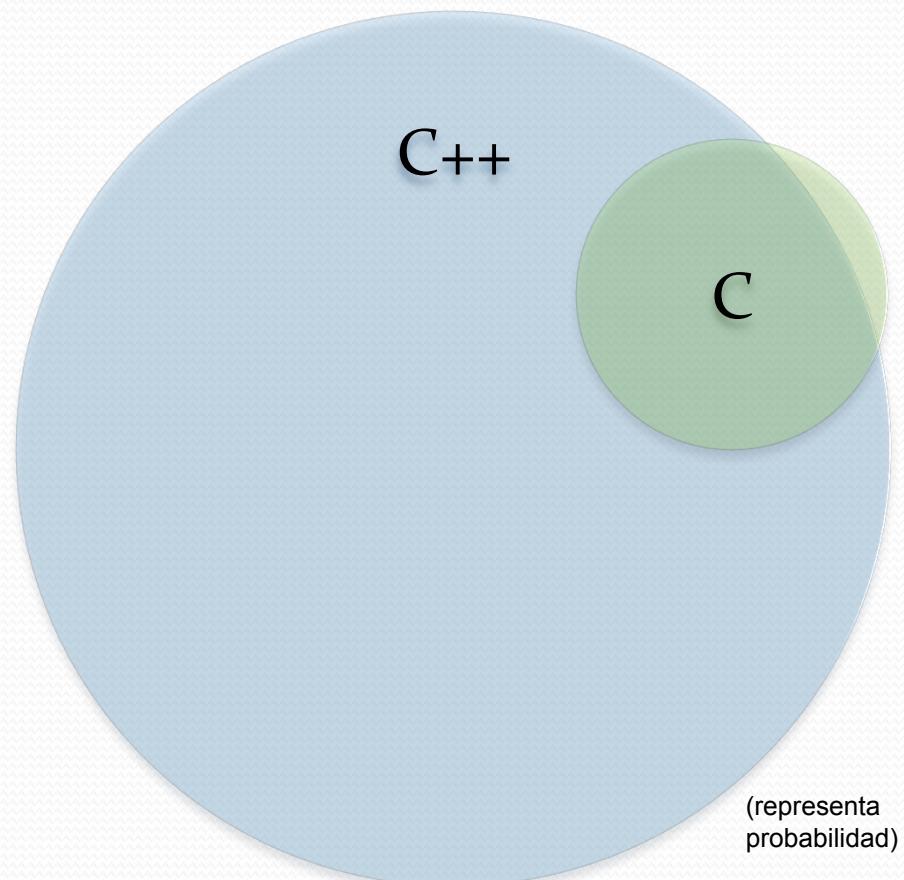
- Programa que hace programas, un meta programa
- Traductor
- Función de Lenguaje a Lenguaje
- Proceso, en etapas
- Baja de Nivel.

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



"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:

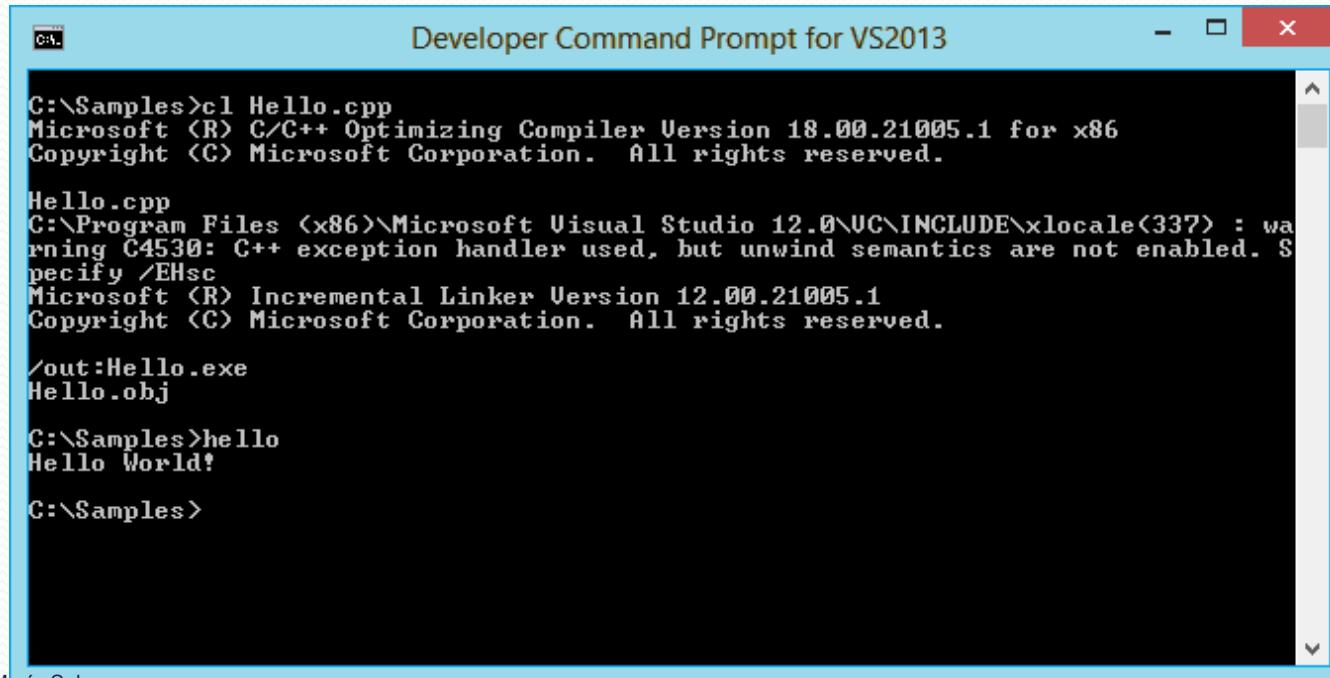
> cc hello.c -std=c11 -Wall -pedantic-errors

The screenshot shows a terminal window with the title bar 'CHelloWorld — -bash — 69x8'. The window contains the following text:

```
[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 'Developer Command Prompt for VS2013' window. The command line output is as follows:

```
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 'cmd C:\WINDOWS\System32\cmd.exe'. The window displays the following text:
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.

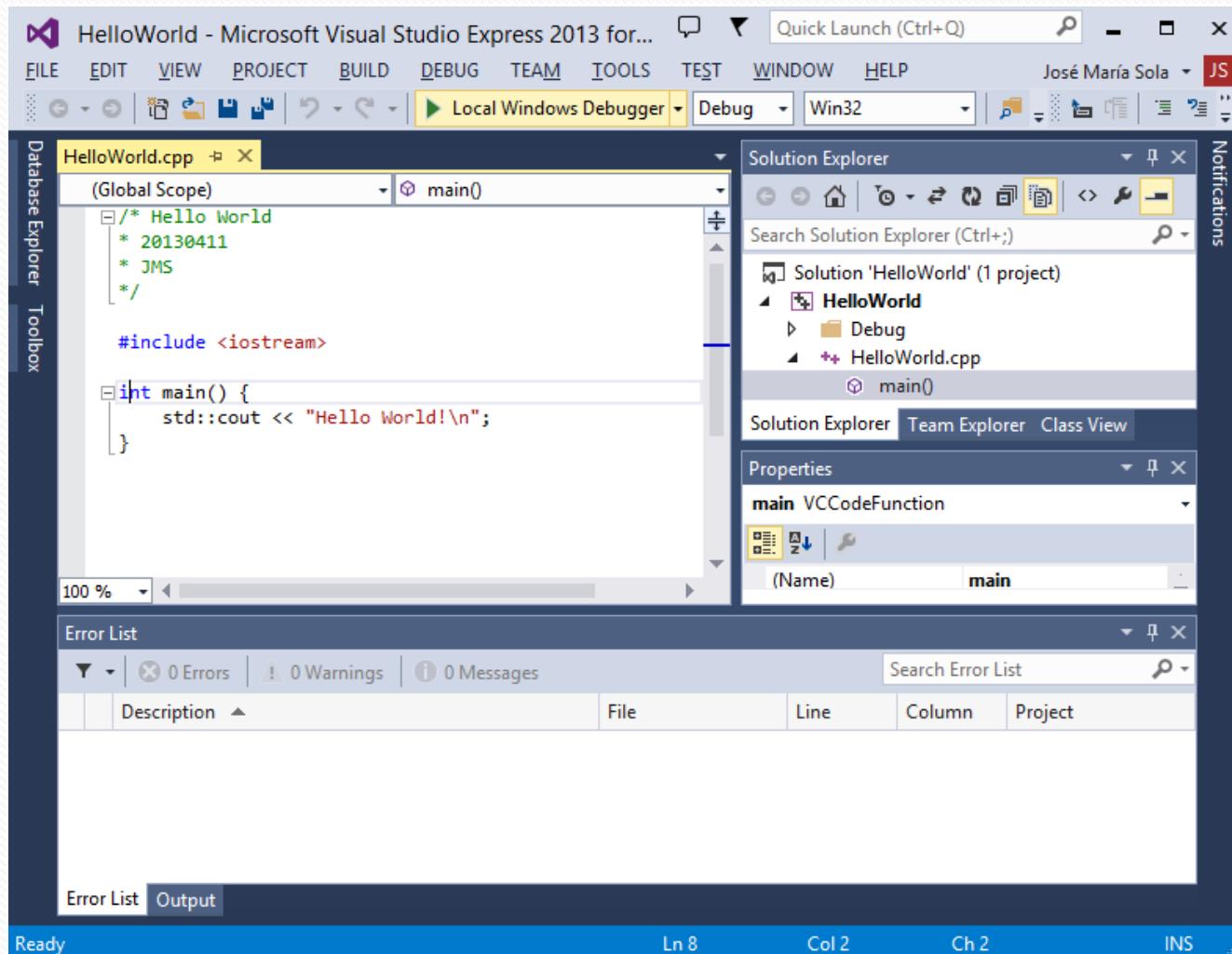
Con IDE

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

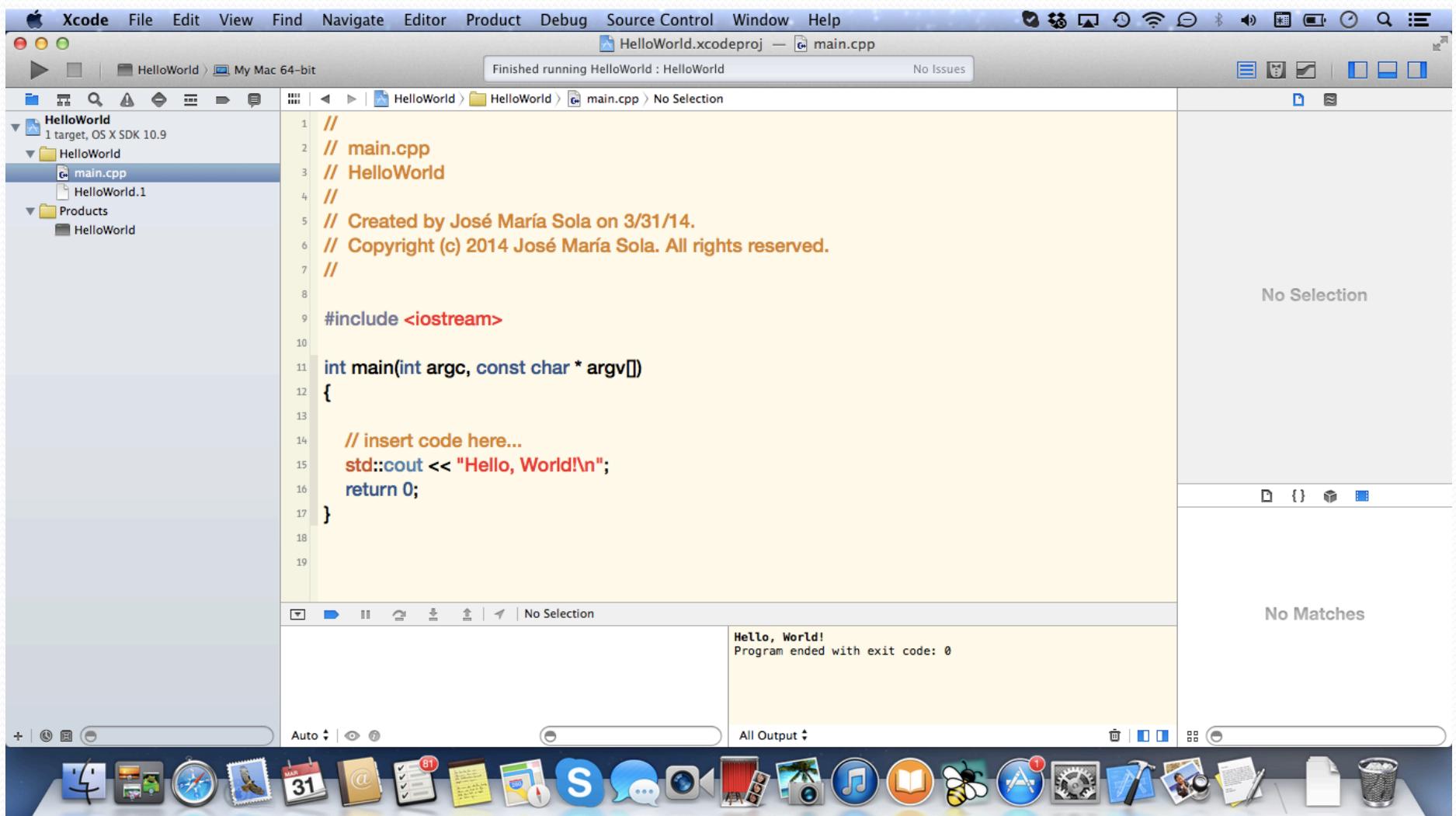
Sin IDE
Requiere editor

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 Xcode IDE interface on a Mac OS X desktop. The menu bar includes Xcode, File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, and Help. The title bar shows "HelloWorld.xcodeproj — main.cpp". A status bar at the bottom indicates "Finished running HelloWorld : HelloWorld" and "No Issues".

The left sidebar displays the project structure:

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

The main editor area shows the content of main.cpp:

```
1 //  
2 // main.cpp  
3 // HelloWorld  
4 //  
5 // Created by José María Sola on 3/31/14.  
6 // Copyright (c) 2014 José María Sola. All rights reserved.  
7 //  
8  
9 #include <iostream>  
10  
11 int main(int argc, const char * argv[]){  
12 {  
13  
14     // insert code here...  
15     std::cout << "Hello, World!\n";  
16     return 0;  
17 }  
18  
19
```

The bottom output window shows the program's output:

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

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>

Trabajo Práctico #0 –

"Hello, World!" en C

- Secuencias de Tarea
 - Solicitar inscripción al Grupo Yahoo.
 - Registrarse en GitHub
 - Crear un repositorio público con el nombre 'CHelloWorld'.
 - Seleccionar, instalar, y configurar un compilador C11.
 - Probar el compilador con un programa '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 'hello.c' y 'output.txt' en GitHub.
 - Enviar a UTNFRBASSL@yahoo-groups.com 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.

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
 - Orientación a Objetos
- SSL
 - Lenguaje de Programación
 - Comportamiento
 - Lenguaje
- Lenguaje Natural
- Lenguaje Formal
- Sintaxis
- Semántica
- Léxico
- Pragmática
- Plataforma del Curso
 - SCV (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. Realizar TP#0 "Hello, World!".



¿Consultas?



Fin de la clase

Clase #02 de 27

Introducción al Lenguaje de Programación C y al Proceso de Compilación

Marzo 20, Lunes

Agenda para esta clase

- Devolución TP #0
- Conformación de Equipos
- Intervalo
- Introducción al Lenguaje de Programación C
- “El Lenguaje de Programación C” aka “El Libro Blanco” aka “K&R”
- El Proceso de Compilación, versión simplificada
- El Preprocesador.

Devolución TP #0

Conformación de Equipos

1. Cinco integrantes como mínimo
2. Por equipo, enviar UTNFRBASSL@yahoo.com apellidos, nombres y usuario github de integrantes.

Intervalo

20 minutos

Introducción al Lenguaje de Programación C

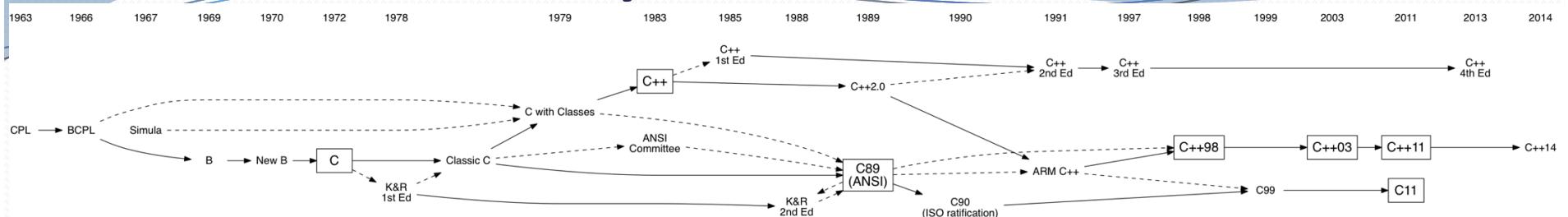
Descripción general

- 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.

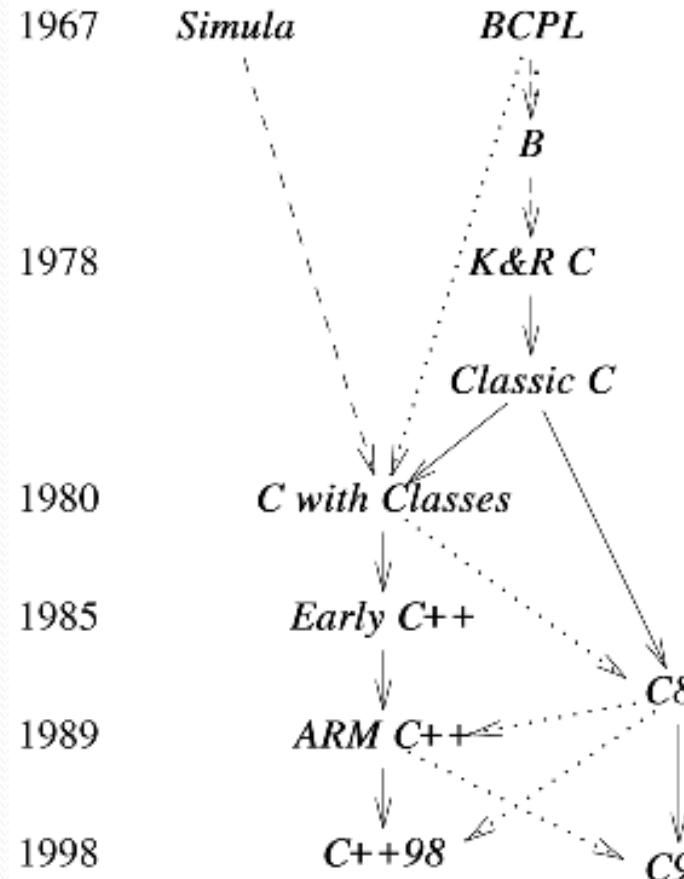
Frases sobre C

- 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)

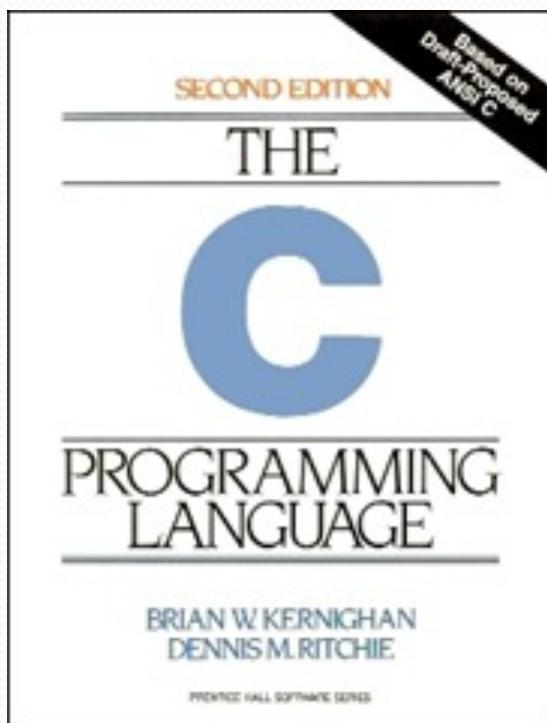


- Otros LP
 - C++
 - Objective-C
 - C#
 - D
 - Java

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”



- 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 alocación de memoria
- “Apéndices”
 - A: Especificación del LP: Semántica (LN) y Sintaxis (BNF)
 - B: Biblioteca estándar.

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

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

```
> cc hello.c
```

```
> hello.exe
Hello, world!
```

Ejercicios

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

El Proceso de Compilación, Versión Simplificada

Hello World

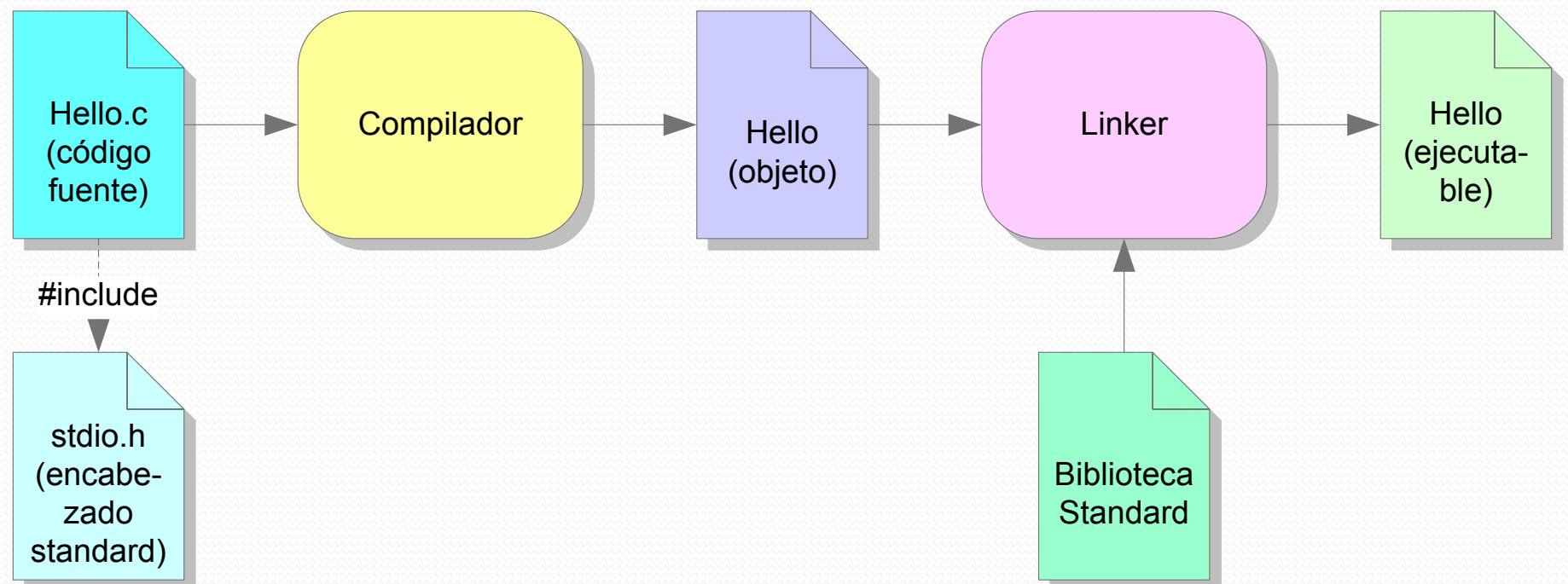
```
/* Hello world
 * JMS
 * 20150402
 */
#include <stdio.h>

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
- Comparacione
 - Contrastar con Máquina Virtual y Bytecode
 - Constrarstar 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.

Términos de la clase #02

Definir cada término con la bibliografía

- Introducción a 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
- K&R Capítulo #1
 - “El Lenguaje de Programación C” aka “El Libro Blanco” aka “K&R”
- Proceso de Compilación Simplificado
 - 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.

Tareas para la próxima clase

1. Corregir TP#0
2. Conformar Equipos: enviar integrantes y usuarios GitHub a UTNFRBASSL@yahooroups.com.
3. Leer de [K&R1988] desde **1.2 Variables y Expresiones Aritméticas** hasta **1.4 Constantes Simbólicas** inclusive.



¿Consultas?



Fin de la clase

Clase #03 de 27

Evaluación #0

Marzo 27, Lunes

Repaso Clase Anterior

- El Proceso de Compilación, Versión Simplificada
- Glosario
- Tareas

Agenda para esta clase

- Análisis y Síntesis de Hello.c
- Autómata Finito para Identificadores
- Sintaxis, Semántica, y Pragmática
- Versiones del Estándar
- Evaluación #o
- Autómata Finito para Identificadores.

Sintaxis, Semántica, y Pragmática

Otras versiones – ¿Mismas 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!" );
}
```

Análisis y Síntesis de Hello.c

Análisis de ‘Hello, World!’

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

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

Incluye información acerca de la biblioteca estándar

Define una función llamada *main* que no recibe valores argumento. Las sentencias de *main* se encierran entre llaves

- Estructura de un programa
 - vs. Pascal
 - Lineal vs. Jerárquico
- Rol de *main*
- Preprocesador
- Biblioteca estándar: Entrada/Salida
 - vs Framework
- Rol de llaves {}
- Función *printf*
- Pasaje de argumentos a funciones
 - Constantes de cadena o literal cadena
 - Secuencia de escape
 - Punto y coma como terminador
 - vs. Pascal
 - Analizar que es “palabra” del LP y que no
 - Identificadores: *main* y *printf*--
¿Cuál es el autómatafinito que los reconoce?

main llama a la función de biblioteca estándar *printf* para imprimir esa secuencia de caracteres. \n representa el carácter nueva-linea

Compiler Explorer - C++

Become a Patron

Source on GitHub

Share

Social ▾

Help ▾

C++ source #1

```
1 /* Hello.cpp
2  C11
3  JMS
4  2015
5 */
6
7 #include <stdio.h>
8
9 int main(void){
10    printf("Hello, World!\n");
11 }
12
```

#1 with x86-64 clang 4.0.0 x

x86-64 clang 4.0.0 -x c -std=c11

11010 .LX0: .text // Intel A A A

```
1 main:                                # @main
2     push   rbp
3     mov    rbp, rsp
4     sub    rsp, 16
5     movabs rdi, .L.str
6     mov    al, 0
7     call   printf
8     xor    ecx, ecx
9     mov    dword ptr [rbp - 4], eax # 4-byte Spill
10    mov    eax, ecx
11    add    rsp, 16
12    pop    rbp
13    ret
14
15 .L.str:
16     .asciz "Hello, World!\n"
17
18
19
```

clang version 4.0.0 (tags/RELEASE_400/final 297782) - cached

#1 with x86-64 clang 4.0.0 x

Compiler exited with result code 0

Generalización de hello.c

Programa General

- Un programa es una secuencia de funciones. Forma general:
main
f
g
...
- Forma general de una función
Tipo Nombre (Parámetros)
{ Cuerpo }
- Cuando se corre (ejecuta) un programa, por convención, main es la primera función invocada por el ambiente de ejecución (e.g., sistema operativo). Todo programa debe tener un main con o sin parámetros

hello.c

- Este programa define solo la función main
- Como toda función, main puede tener o no parámetros
- Este main invoca a printf con una cadena literal (cadena constante) como argumento. "Entre comillas".

Otras versiones de hello.c

Diferencias con ANSI 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";  
}
```

Evaluación #0

Cada alumno resuelve, y luego corrige a su compañero

Tiempos y como responder

- Resolución
 - 4 minutos por pregunta, 20 minutos total
- Corrección
 - 2 minutos por pregunta, 10 minutos total

UTN FRBA – Sintaxis y Semántica de Lenguajes – Evaluación #0 – Evaluación Integral de Asignaturas Anteriores – Tema A					Curso	Fecha		
Quien resuelve debe responder debajo de cada pregunta.					Para la Cátedra			
Apellido, Nombre	Legajo	Equipo			Apellido, Nombre	Legajo	Equipo	Para la Cátedra
Pregunta	B	R	M	B	R	M	Justificación de corrección	
Respuesta								
1 Indique las diferencias entre función y procedimiento. • •								
2 Indique el paradigma de programación aplicado en la 2da (y en la 3era) materia del área de programación. • •								
3 Defina Ingeniería. • •								
4 Defina subprograma. • •								
5 Defina función matemática. • •								
Calificación	Resolución >				< Resolución		Corrección >	

Resolución Evaluación #0

Resolución Evaluación #0 – General

- Defina información
- Defina sistema de información
- Defina abstracción
- Indique la tarea más representativa para la cual un Ingeniero en Sistemas de Información está calificado
- Indique las asignaturas del área de programación
- Indique las asignaturas del área de sistemas de información.

Resolución Evaluación #0 – Matemática Discreta

- Defina función matemática
- Defina grafo
- Defina autómata finito
- Indique diferencias entre secuencia y conjunto
- Defina el concepto matemático secuencia
- Sea $X=\{a,b\}$ indique $P(X)$
- Defina partición de un conjunto
 - 1. $\bigcup P = X$
 - 2. $A \cap B = \emptyset$ if $A \in P, B \in P, A \neq B$

Resolución Evaluación #0 –

Algoritmos y Estructuras de Datos

- Defina algoritmo
- Indique el paradigma de programación aplicado en AED
- Indique las tres estructuras de control de flujo de ejecución de la programación estructurada
- Defina Tipo de dato Abstracto
- Defina programación estructurada
- Indique las diferencias entre función y procedimiento
- Explique las diferencias entre parámetros de entrada, de salida y de entrada/salida
- Explique las dos formas de transferencia de argumentos
- Indique las diferencias entre parámetro y argumento
- Defina archivo
- Indique las diferencias entre una pila y una cola
- Diseñe un algoritmo que dado un arreglo de naturales a y su longitud n calcule el promedio (desarróllelo al dorso).

Términos de la Evaluación #0

- Información
- Sistema de información
- Abstracción
- Área de programación
- Área de sistemas de información
- Función matemática
- Grafo
- Autómata finito
- Secuencia
- Conjunto
- Conjunto Ordenando
- Multiconjunto
- Powerset o Conjunto potencia
- Partición de un conjunto
- Algoritmo
- Paradigma
- Tipo de Paradigma
- Tipo de Paradigma

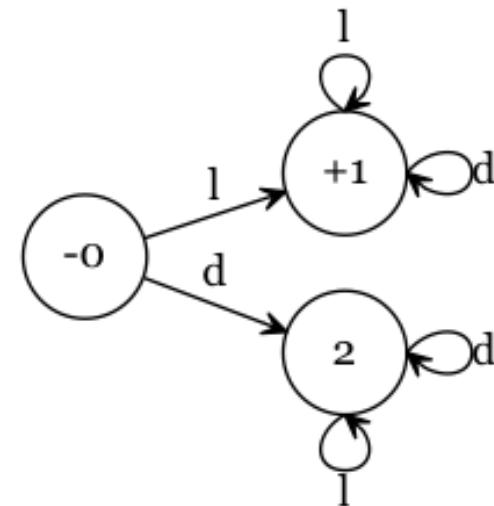
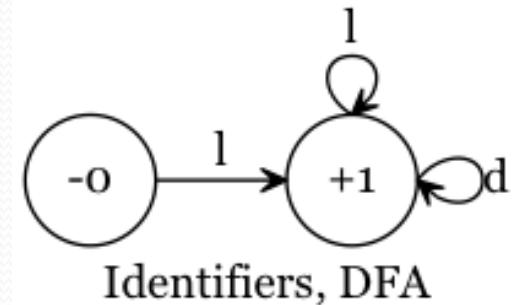
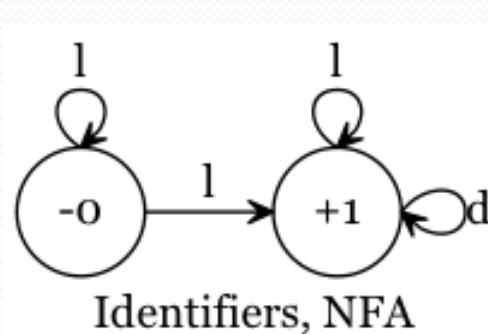
- Declarativo
- Tipo Paradigma Imperativo
- Paradigma Funcional
- Paradigma Lógico
- Paradigma Orientado a Objetos
- Paradigma Procedural
- Lenguaje de programación multiparadigma
- Estilo de programación estructurado
- Tipo de dato
- Estructuras de control de flujo de ejecución de la programación estructurada
- Tipo de dato Abstracto
- Función versus Procedimiento
- Parámetro versus Argumento
- Parámetros de entrada, de

- salida, y de entrada/salida: in, out, inout.
- Transferencia por valor o copia, versus variable, o referencia.
- Archivo
- Memoria secundaria versus memoria primaria
- Pila
- Cola
- Lista
- Implementación contigua
- Implementación enlazada.
- Tratamiento de casos excepcionales
- Arreglos vacíos.

Autómata para Identificadores

Componentes de un Autómata Finito

- Estados
 - Inicial
 - Finales
- Transición
 - Origen
 - Evento
 - Destino
 - Transición reflexiva
- Alfabeto
- Ejemplo:



Identifiers, Complete DFA

Términos de la clase #03

Definir cada término con la bibliografía

- Niveles
 - Sintaxis
 - Semántica
 - Pragmática
- Análisis y Síntesis de hello.c
 - main
 - printf
 - Función del #include
 - puts
 - Secuencia de escape
 - Terminador de sentencia de C
 - Separador de sentencias de Pascal
 - Archivo .h (Encabezado)
 - String literal: Cadena Literal, o Literal de cadena, o Constante cadena, o Cadena constante
 - Lenguaje Ensamblador
 - Estructura lineal de un programa C
 - Estructura jerárquica de un programa Pascal
- Otras Versiones de Hello.c
 - Concatenación de cadenas
- Valor returned por main
- Múltiples invocaciones
- puts
- EXIT_SUCCESS
- Examen #o
- Autómata Finito para Identificadores
 - Identificadores
 - Autómata Finito
 - Estados
 - Inicial
 - Finales
 - Transición
 - Origen
 - Evento
 - Destino
 - Transición reflexiva
 - Alfabeto
 - Dígrafo.

Tareas para la próxima clase

1. Leer de [K&R1988]
1.5 Entrada y Salida de a Caracteres
2. Realizar los ejercicios de [K&R1988] de 1-1 a 1-5 inclusive.



¿Consultas?



Fin de la clase