

# Algoritmos y Estructuras de Datos AED

*K1051*

*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

*Mar 23, Jueves*

# Agenda para esta clase

- Bienvenidos a la Universidad y al curso
- Presentación inicial, conozcámonos
- El contexto de AED
- Sobre la Cátedra y la Asignatura
- La aprobación y la regularización
- Plataforma del Curso
- Sobre el Lenguaje C++
- Primer contacto con la programación
- Trabajo Práctico #0
- 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
  - 5 horas cátedra con intervalo de 15 minutos
    - 3 horas reloj + intervalo de 15 o 20 minutos
  - **19:00** a
    - Intervalo 20:30 o 20:45
  - **23:00**
- Consultas presenciales, antes y después de clase.

# El Contexto de AED

# 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 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



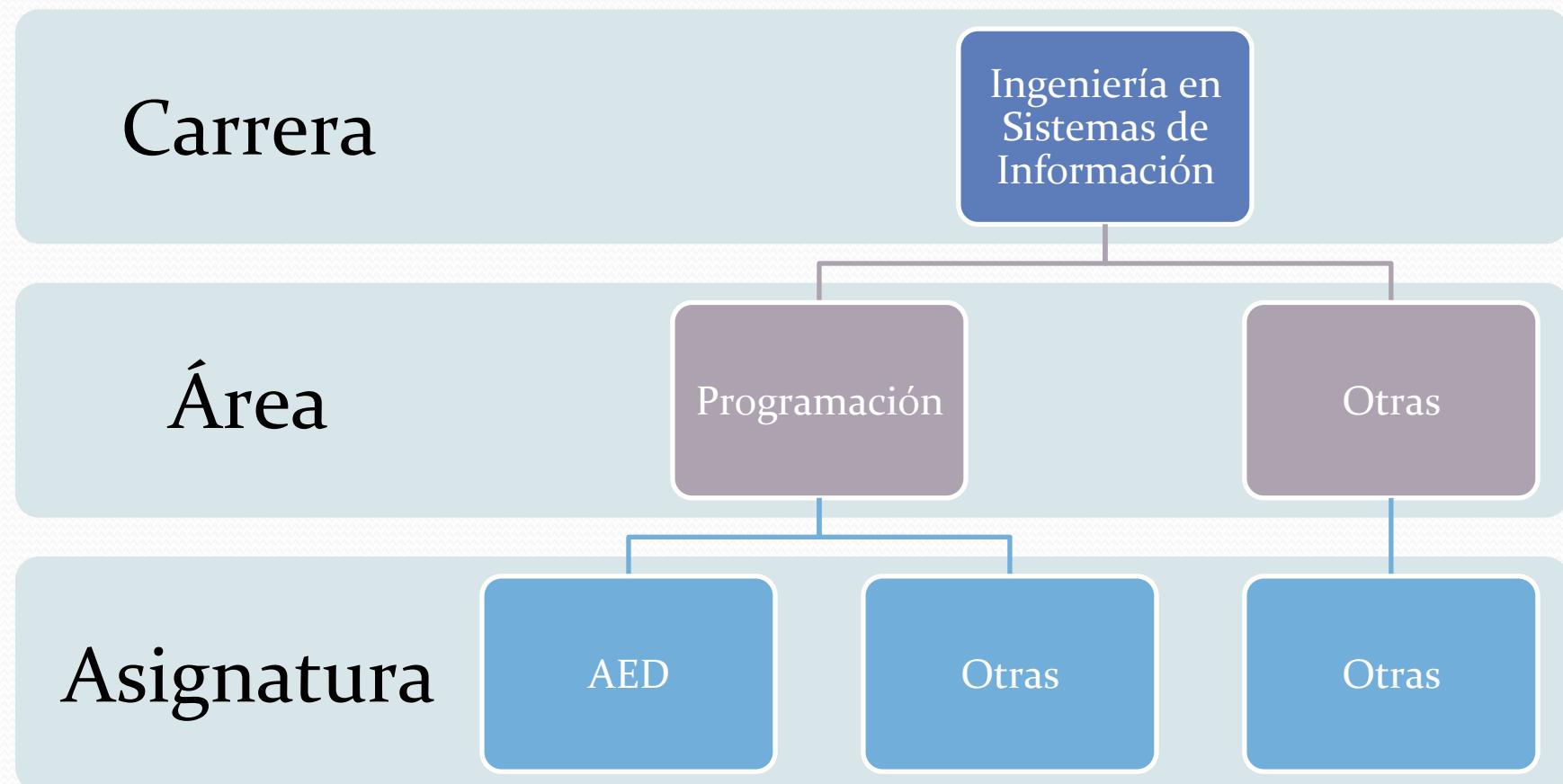
# Las Asignaturas Paralelas

- Matemática Discreta
- Sistemas y Organizaciones

# 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.

# AED 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).

# Integración con otras Asignaturas y Conceptos Principales

- Integración en Área de Programación
  - Integración vertical
    - Paradigmas de Programación
    - Sintaxis y Semántica de Lenguajes
  - Integración Horizontal
    - Matemática Discreta
- 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 – Algoritmos y Estructuras de Datos

- Algoritmos
  - Pasos para resolver un problema general
    - Datos de entrada
    - Lista de pasos
    - Salida resultante
- Datos
  - Valores de un conjunto
  - *Datum*
  - Dato versus Información versus Conocimiento
- Estructura
  - Relaciones entre entidades para almacenar datos
- Tipo de Datos

# Generalidades de AED

## Ordenanza 1150/2007

- Departamento **Ingeniería en Sistemas de Información**
- Bloque **Tecnologías Básicas**
- Área **Programación**
- Nivel **1**
- Horas/semana **5**
- Horas/Año **160**
- Correlatividades
  - Anteriores: **Ninguna**
  - Siguientes
    - Debe estar regularizada para cursar
      - **Análisis de Sistemas (2do)**
      - **Sintaxis y Semántica de los Lenguajes (2do)**
      - **Paradigmas de Programación (2do)**
      - **Sistemas Operativos. (2do, acorde a ordenanza 1150/2007)**
    - Debe estar aprobada para cursar
      - **Diseño de Sistemas (3ro)**
      - **Gestión de Datos (3ro)**
      - **Economía (3ro)**
      - **Redes de Información (4to, acorde a ordenanza 1150/2007)**
      - **Legislación (4to)**

# Programa Sintético de AED

Ordenanza 1150/2007

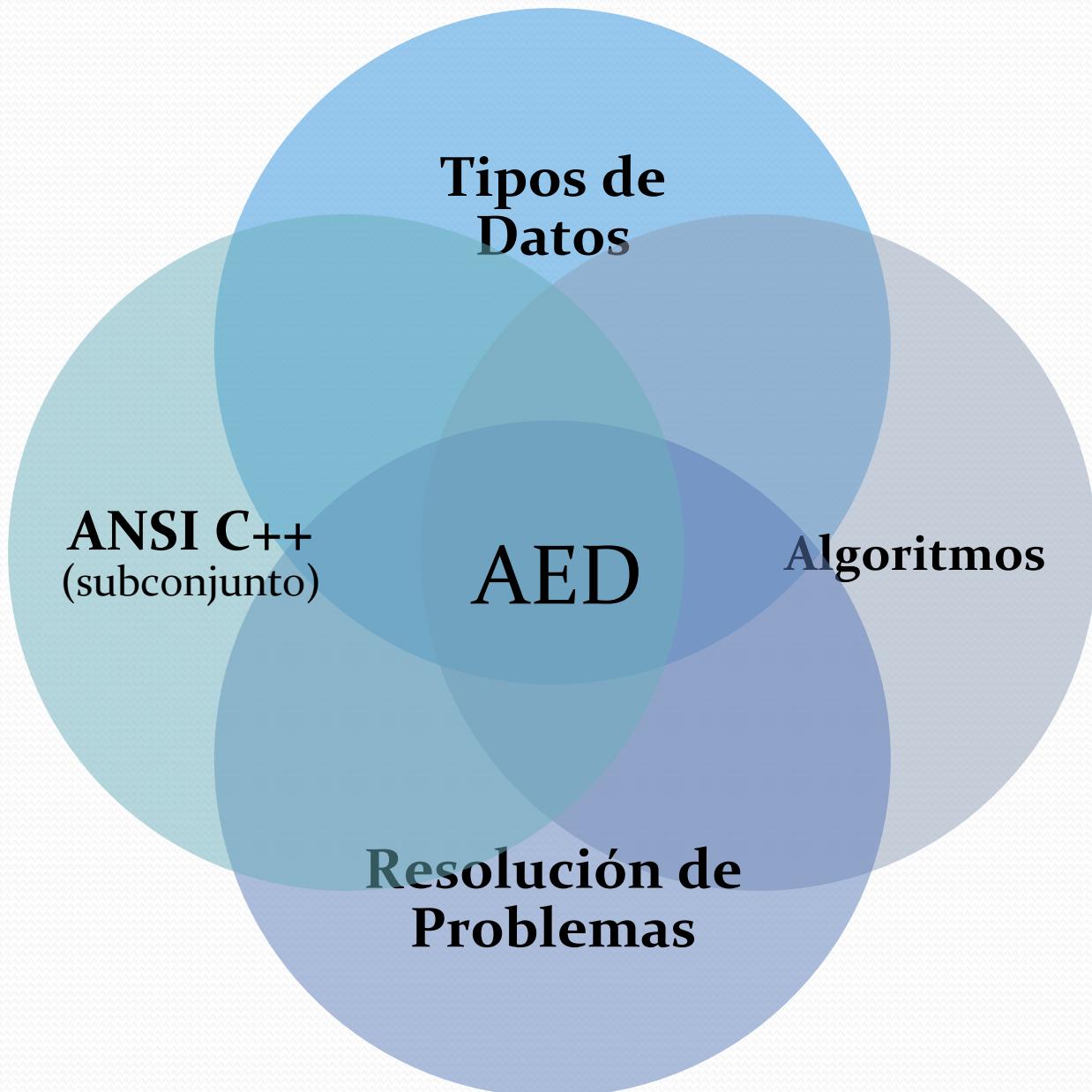
## Objetivos

- 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**
- Noción de **Complejidad Computacional**
- Noción de **Orden de Complejidad**.

# Los Temas de AED



# La Cátedra

- Dr. Mg. Oscar Bruno, Profesor Asociado Ordinario
  - Coordinador de la Cátedra
  - @orbruno
  - [droscarbruno.wordpress.com](http://droscarbruno.wordpress.com)

# La Aprobación y la Regularización

# Aprobación – Examen Final

- Requisito para aprobación
  - Regularización y aprobación del examen final
  - Aprobación Directa
- Examen final
  - 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.

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

- 75% de Asistencia (Bedelía)
- Evaluaciones (Cátedra)
  - Grupales e individuales,
  - De aplicación o conceptuales
  - Evaluación frecuentes
  - Trabajos Prácticos, entregas semanales
  - Dos recuperatorios, en Dic y en Feb
  - Las fechas se establecen durante el curso
  - Fin del curso
    - Evaluación individual oral o escrita ("coloquio")
    - Conocimientos de los trabajos prácticos.

# 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**

**GitHub  
Papers y  
Resoluciones de  
cada equipo**

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

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

**Bibliografía  
droscarbruno.  
wordpress.com**

**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

- Obligatoria
  - <https://droscarbruno.wordpress.com/algoritmos-desarrollo-de-temas/>
- Recomendada
  - Sobre Programación
    - [Avanzado] Kernighan, B; Pike, R. *La práctica de la programación.* 2000. Pearson: México
    - [Inicial] García Molina, J; Montoya Dato, F; otros. *Una introducción a la programación: un enfoque algorítmico.* 2005. Thomson: España
  - Sobre el Lenguaje C++
    - [Avanzado] Stroustrup, B. *El Lenguaje de Programación C++.* 2013. Addison Wesley
    - [Inicial] Deitel, H; Deitel, E. *Cómo programar en C/C++.* 1994. Prentice Hall: México
  - Otros artículos a lo largo del curso, publicados desde la página del curso.

# Grupo Yahoo UTNFRBAAED

## <https://groups.yahoo.com/group/UTNFRBAAED>

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

**YAHOO!**  
GROUPS

Search Conversations

Sign In Mail

Browse Groups

Terms Privacy Guidelines Feedback Help Blog

**UTN FRBA AED**

Restricted Group, 101 members

Conversations Photos Files More ▾ About ?

**About Group**

**Group Description**

UTN FRBA AED  
Grupo Yahoo de la Cátedra de Algoritmos y Estructuras de Datos (AED) de la especialidad Ingeniería en Sistemas de Información en la Facultad Regional Buenos Aires (FRBA) de la Universidad Tecnológica Nacional (UTN).

En el grupo se realizan y responden consultas sobre la asignatura y la Cátedra publica anuncios, novedades, resultados de evaluaciones, ejercitaciones, material bibliográfico y enunciados de trabajos prácticos.

Para suscribirse a este grupo debe:

1. obtener un usuario Yahoo que incluya su nombre y apellido,
2. asociar ese usuario a una casilla de email que también incluya su nombre y apellido y
3. al momento de suscribirse, en el recuadro Comment to, debe ingresar:

- 3.1. nombre y apellido,
- 3.2. legajo y
- 3.3. código de curso

La inscripción a este grupo para el primer cuatrimestre de 2014 finaliza el Lunes 05 de Mayo, al comenzar la séptima semana de clase.

Administradores – Facundo Merighi & José María Sola.-

# Grupo Yahoo UTNFRBAAED

<https://groups.yahoo.com/group/UTNFRBAAED>

- Objetivos
    - 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
  - 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-aed>

- 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-aed*
      - Para los miembros del curso asignamos una repositorio privado, dentro **utn-frba-aed**, a cada equipo
      - <https://github.com/orgs/utn-frba-aed/>
  - Primeros Pasos en GitHub
    1. Leer guía "Git 101"
    2. Crear Usuario
    3. Crear Repositorio.
- <https://josemariasola.wordpress.com/aed/git/>

# Primer Contacto con la Programación

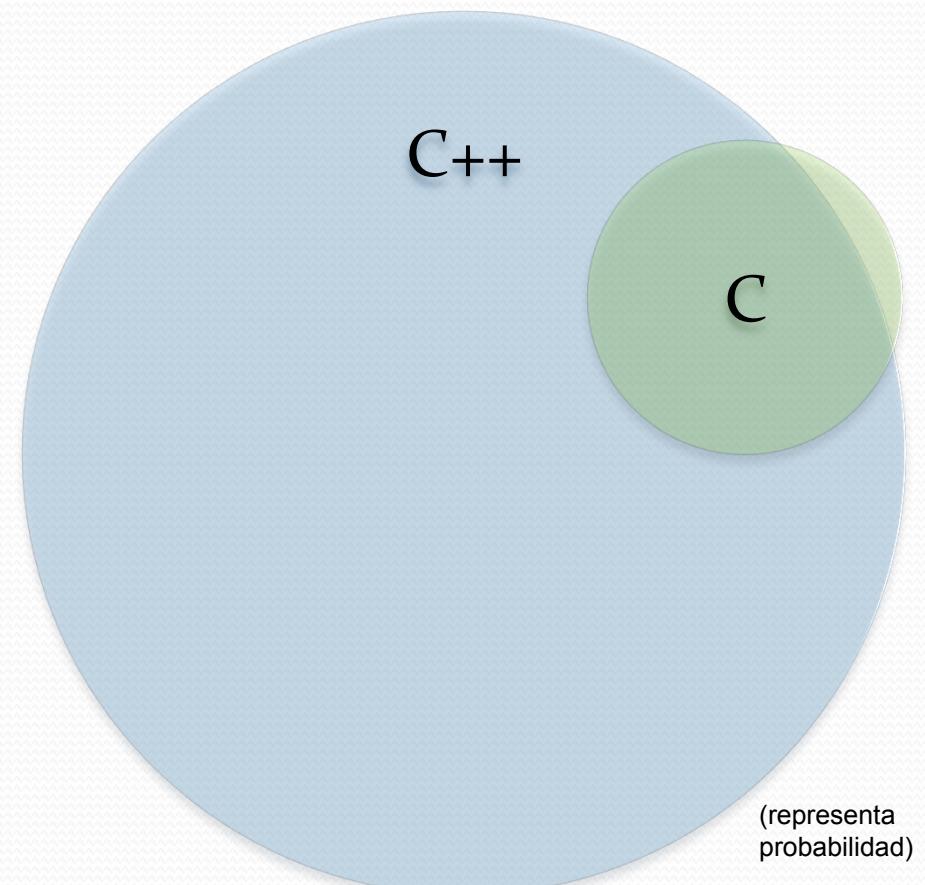
Lenguajes y Herramientas de Desarrollo

# 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!"

- Propósito
- Comentario encabezado
  - Qué
    - Título descriptivo
  - Quién
    - Número de Equipo e integrantes
  - Cuándo
    - Se actualizó por última vez

```
/* Hello world
JMS
20130411
*/
#include <iostream>
int main() {
    std::cout << "Hello, World!\n";
}
```



# 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 .cpp (e.g., hello.cpp)
- 2.** **Compilar** el archivo fuente para producir el programa objeto (e.g., c++ 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 C++14

1. Desde la línea de comando

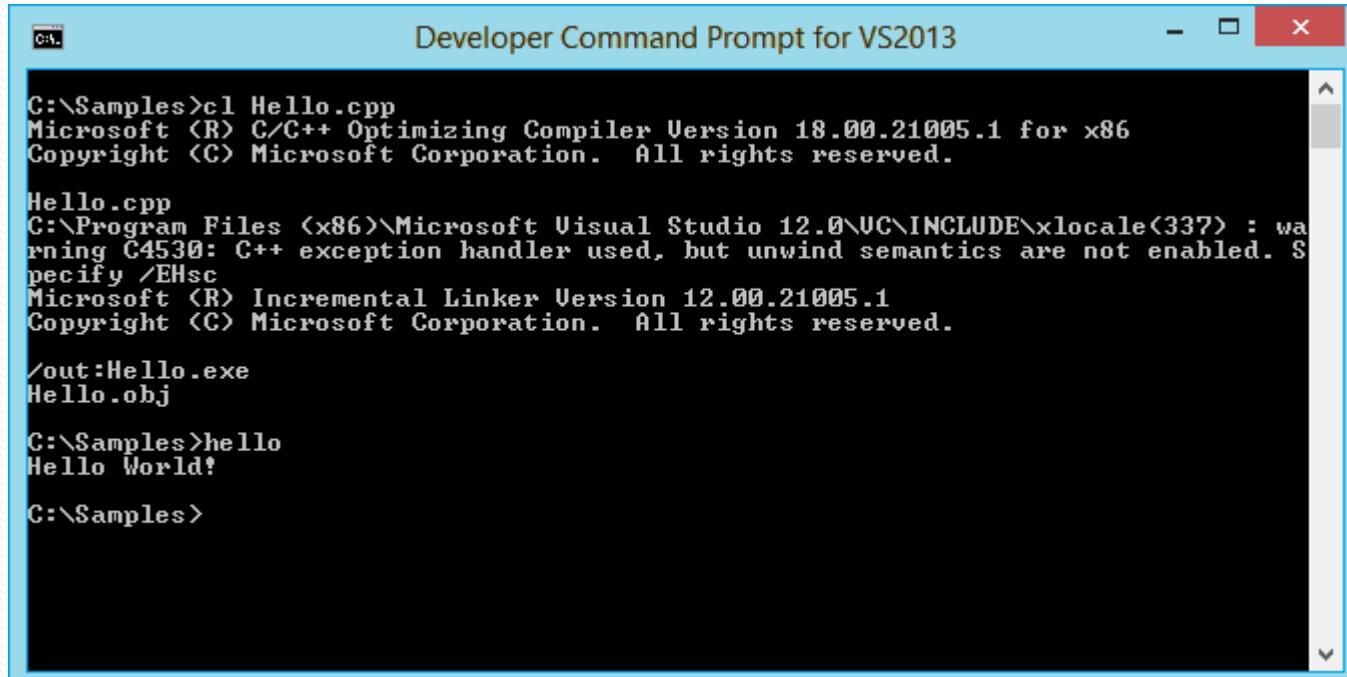
1. `> vi hello.cpp` crear el fuente
2. `> c++ hello.cpp -std=c++14 -Weverything` crear el ejecutable  
en realidad: Preprocesador → Compilador → Linker
3. `> ./a.out` ejecutar  
Hello, World! salida

2. Si hay un error en el paso 2 ó 3, volver al 1 y repetir 2 y 3

```
josemariasola:CppHelloWorld> c++ hello.cpp -std=c++14 -Weverything
josemariasola:CppHelloWorld> ./a.out
Hello, World!
josemariasola:CppHelloWorld>
```

# Ejemplo desde línea de comando Compilador Microsoft

1. Desde la línea de comando
  1. > **notepad hello.cpp** crear el fuente
  2. > **cl hello.cpp** 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.cpp** crear el fuente
  2. > **bcc32 hello.cpp** 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.exe' with the path 'C:\WINDOWS\System32'. The window contains the following text:

```
C:\> 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:\> C:\Program Files\Borland\BCC55\Bin>hello
Hello, World!

C:\>
```

# 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.

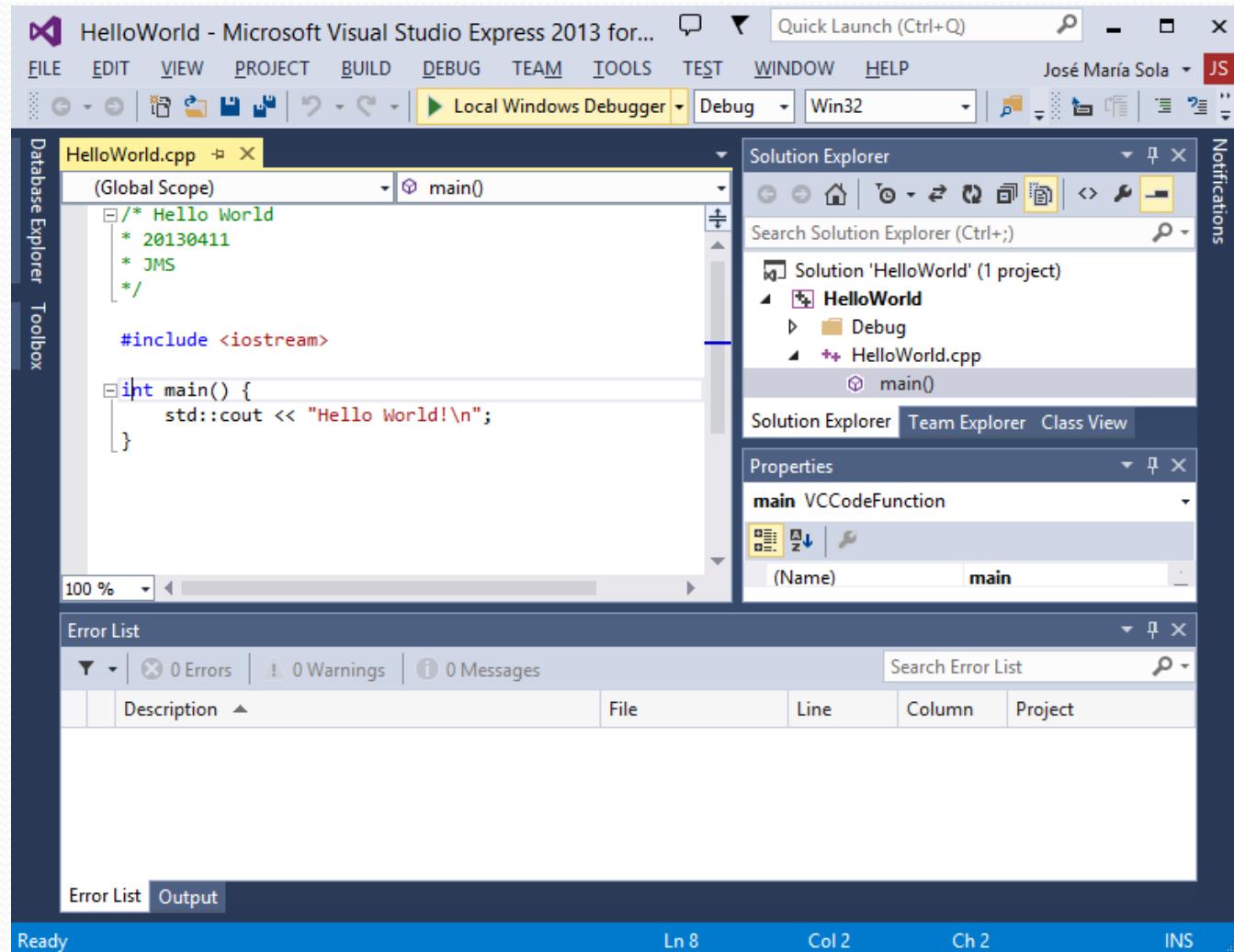
## Con IDE

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

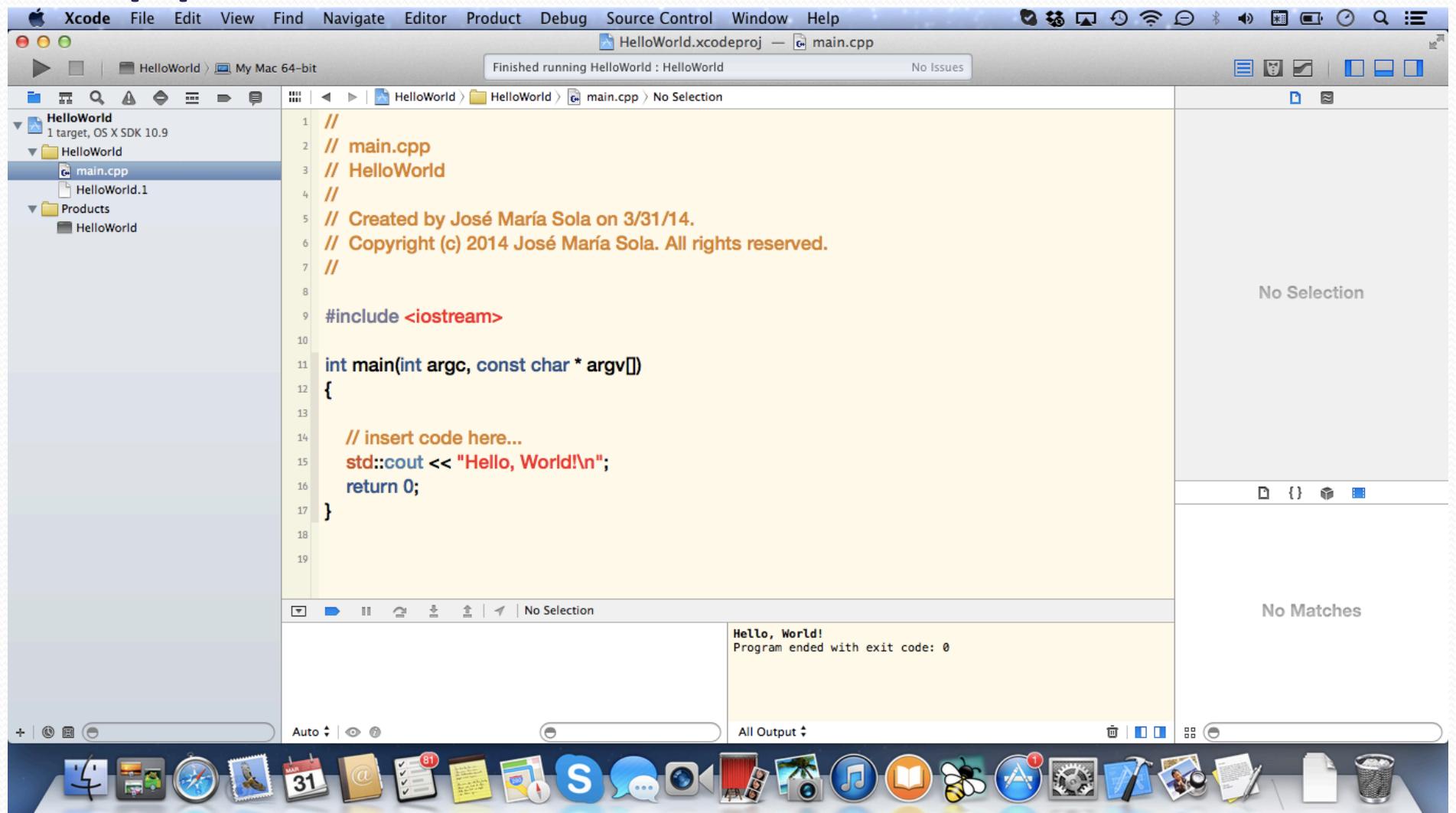
**Sin IDE**  
Requiere editor

Compilador de C/C++

# Ejemplo con IDE Microsoft Visual Studio Express for Windows Desktop



# Ejemplo 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>

# 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 'CppHelloWorld'.
  - Seleccionar, instalar, y configurar un compilador C++14.
  - Probar el compilador con un programa 'hello.cpp' que envíe a 'cout' la línea 'Hello, World!' o similar.
  - Ejecutar el programa, y capturar su salida en un archivo de texto 'output.txt' .
  - Publicar 'hello.cpp' y 'output.txt' en GitHub.
  - Enviar a UTNFRBAAED@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

- Programación
- Lenguaje de Programación
- Comportamiento
- Software
- Ingeniería
- Sistema
- Información
- Ingeniería en Sistemas de Información
- Sistema de Información
- Proyecto
- Problema (Necesidad)
- Solución
- Desarrollo
- Análisis
- Diseño
- Implementación
- Organización
- Abstracción
- Orientación a Objetos
- Algoritmo
- Dato
- Estructura
- Estructura de Datos
- Tipos de Datos
- ISCV (Sistema de Control de Versiones)
- Git
- GitHub
- Repositorio
- C
- C++
- IDE (Integrated Development Environment, Entorno Integrado de Desarrollo)

# Tareas para la próxima clase

1. Realizar TP#0 "Hello, World!".



# ¿Consultas?



**Fin de la clase**

# Clase #02 de 27

## Hello.cpp & Addition.cpp

*Marzo 30, Jueves*

# Agenda para esta clase

- Repaso de clase anterior
- Estado de TP #0
- Análisis de "Hello, World!" Hola.cpp
- TP #1 Adición

# Análisis de Hello World

- Análisis desde el interior al exterior
  - Cadena
  - Operador de envío o inserción
  - Objeto cout
  - Función main
  - #include
- Modelo Consola
  - Flujo de datos
    - De Entrada
      - Origen: Teclado
    - De Salida
      - Destino: Pantalla
  - Procesamiento de renglones

```
/* Hello world
 * JMS
 * 20130411
 */
#include <iostream>

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



# Problema: Adición de dos números

- Problema
  - Escribir los pasos para mostrar la suma de dos números que ingresa el usuario.
- Algoritmo
  - Secuencia de pasos que con datos genera resultados
- Léxico
  - Definición de elementos utilizados en el algoritmo, ejemplo:  $x \in R$
- Resoluciones
  - Representaciones Gráficas
    - Lindsay/Chapin
    - Nassi-Shneiderman (NS)
    - Abstracta: Digrafo
      - ¿Qué representan las flechas?
      - ¿Qué representan los nodos?
  - Textuales
    - Secuencia de pasos en casellano
    - Pseudocódigo
  - Lenguajes de Programación
    - C++.

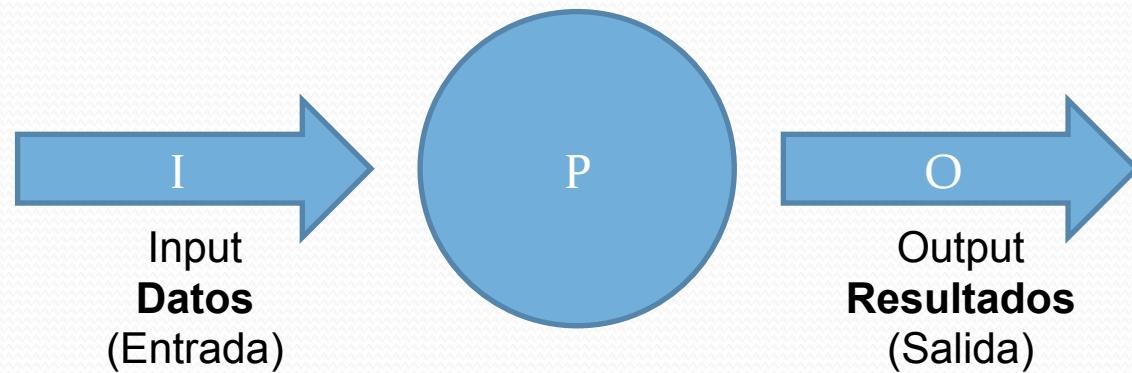
# Adición y Entrada Estándar

```
/* JMS
 * 20130411
 * Adición
 */
#include <iostream>

int main(){
    int a;
    int b;

    std::cin >> a;
    std::cin >> b;
    std::cout << a + b;
}
```

# Proceso



Proceso, requiere  
**recursos**  
espacio y tiempo

# TP #1 – Adición

- Carpeta: Adición
  - Readme.md
    - Enunciado
    - Análisis: Entradas, Proceso, Salidas
    - Algoritmo: Visual o Textual
  - Adición.cpp

# Términos de la clase #02

Definir cada término con la bibliografía

- Hello.cpp
  - Cadena
  - Operador de envío o inserción
  - Objeto cout
  - Función main
  - #include
  - Consola
  - Flujos de datos estándar
  - Flujos de carácter
  - Objeto cin
- Addition.cpp
  - Algoritmo
  - Léxico
  - Representaciones de Algoritmos
    - Lindsay/Chapin
    - Nassi-Shneiderman (NS)
    - Secuencia de pasos en casellano
    - Pseudocódigo
  - Implementación de Algoritmos

# Tareas para la próxima clase

1. Realizar TP#1 Adición.



# ¿Consultas?



**Fin de la clase**