

Programa del curso

Clase #01

IIC1103 – Introducción a la Programación

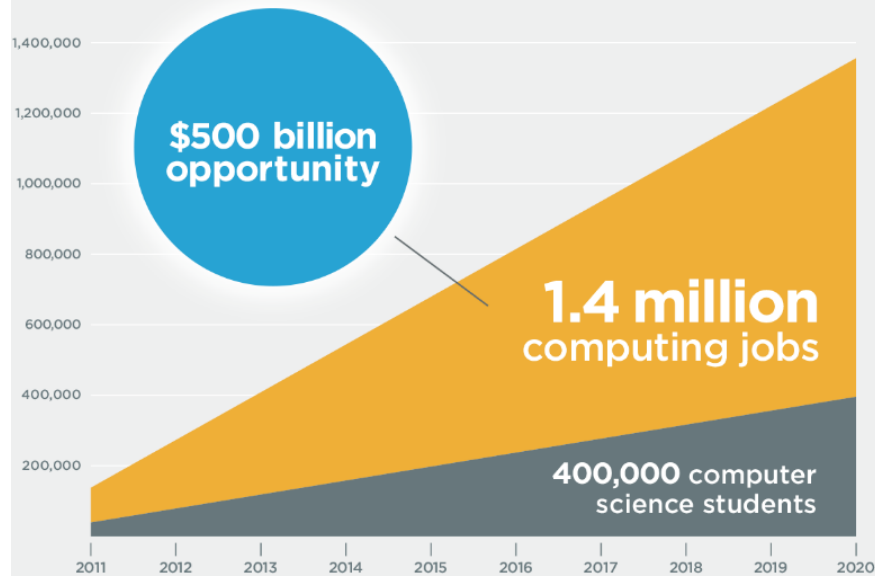
Marcos Sepúlveda (marcos@ing.puc.cl)



PRESENTACIÓN

COMPUTACIÓN Y EL FUTURO

1,000,000 more jobs than students by 2020

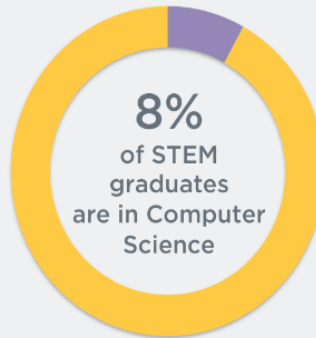


Computer science
is a top paying
college degree
and computer
programming
jobs are growing
at 2X the
national average.

The value of a computer science education



A computer
science major
can earn
40% more
than the
college average.



The
“STEM”
problem
is in
computer
science

Computing jobs are the #1 source of
new wages in the United States.



500,000 current openings: These jobs are in **every** industry and **every** state, and they're projected to grow at twice the rate of all other jobs.



Profesor del DCC UC crea algoritmo de inteligencia artificial que formula alimentos a partir de plantas



Profesor de Ingeniería UC presenta sistema recomendador de música en línea



Académico del DCC UC, Domingo Mery, opina sobre Biometría en El Mercurio



Investigadores de Ingeniería UC presentan estudio sobre adopción de MOOCs en América Latina y Europa



ENTI 2015 revela que las TI reaccionan lento ante necesidades de empresas



Diplomado en Inteligencia de Negocios gradúa a su primera generación



Académico del DCC UC publica estudio comparativo sobre edad curricular y año académico de escolares chilenos



Profesor del DCC UC, Cristian Ruz, opina sobre tecnologías para apoyar facturación electrónica



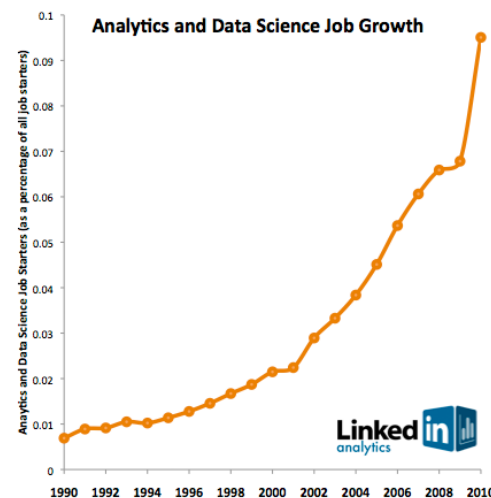
Ingeniería y Enfermería UC emprenden proyecto interdisciplinario



Entre curvas, líneas y estrellas

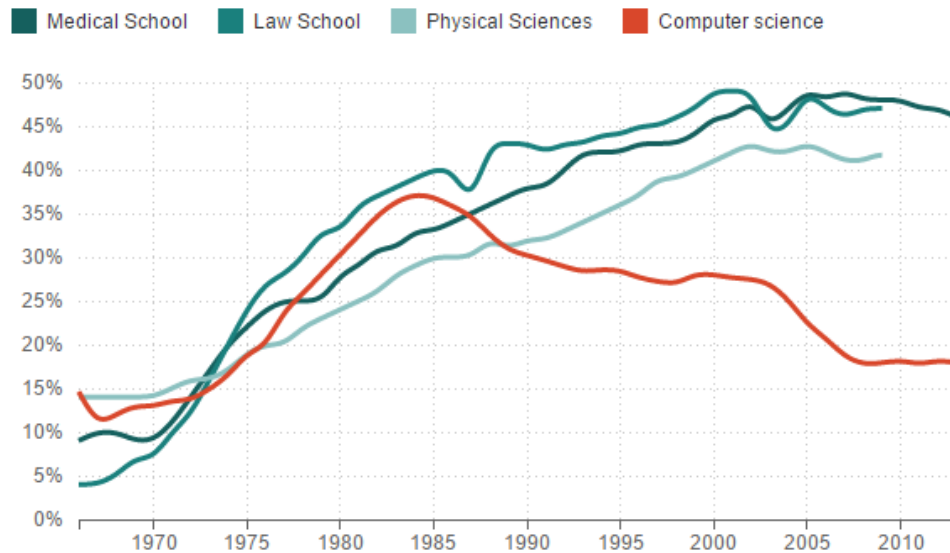
Faculty Position: Big Data for Transportation and Logistics Systems

The School of Engineering at Pontificia Universidad Católica de Chile invites applications for a full-time position in the area of Big Data for Transportation and Logistics Systems.



What Happened To Women In Computer Science?

% Of Women Majors, By Field



Source: National Science Foundation, American Bar Association, American Association of Medical Colleges

Credit: Quoc Trung Bui/NPR

13 octubre 2017:
www.chilewic.cl

VI ENCUENTRO DE
*Mujeres en
computación*

UNA CELEBRACIÓN ACM

Auditorio D'Etigny Beauchef
Av. Beauchef 851, Santiago

Viernes 13 de octubre - 9:00 a 18:30 horas

Inscripciones gratuitas
Más información en www.chilewic.cl

CHILEWIC
WOMEN IN COMPUTING



EL CURSO

Preguntas frecuentes...

¿Qué hago aquí?

Aprender

- Adquirir conocimiento, y luego
- Aplicar conocimiento

¿Por qué?

¿Para qué?

Resolver problemas
usando computación

Preguntas frecuentes...

¿Sirve?

Avanzamos utilizando la ayuda de la computación

Diferentes áreas (medicina, economía, arquitectura, ingeniería...etc.)

¿Dónde se aplica?

¿Es difícil?

Depende de ti. La clave es estudiar y practicar !

¿Qué haremos?

Algoritmo

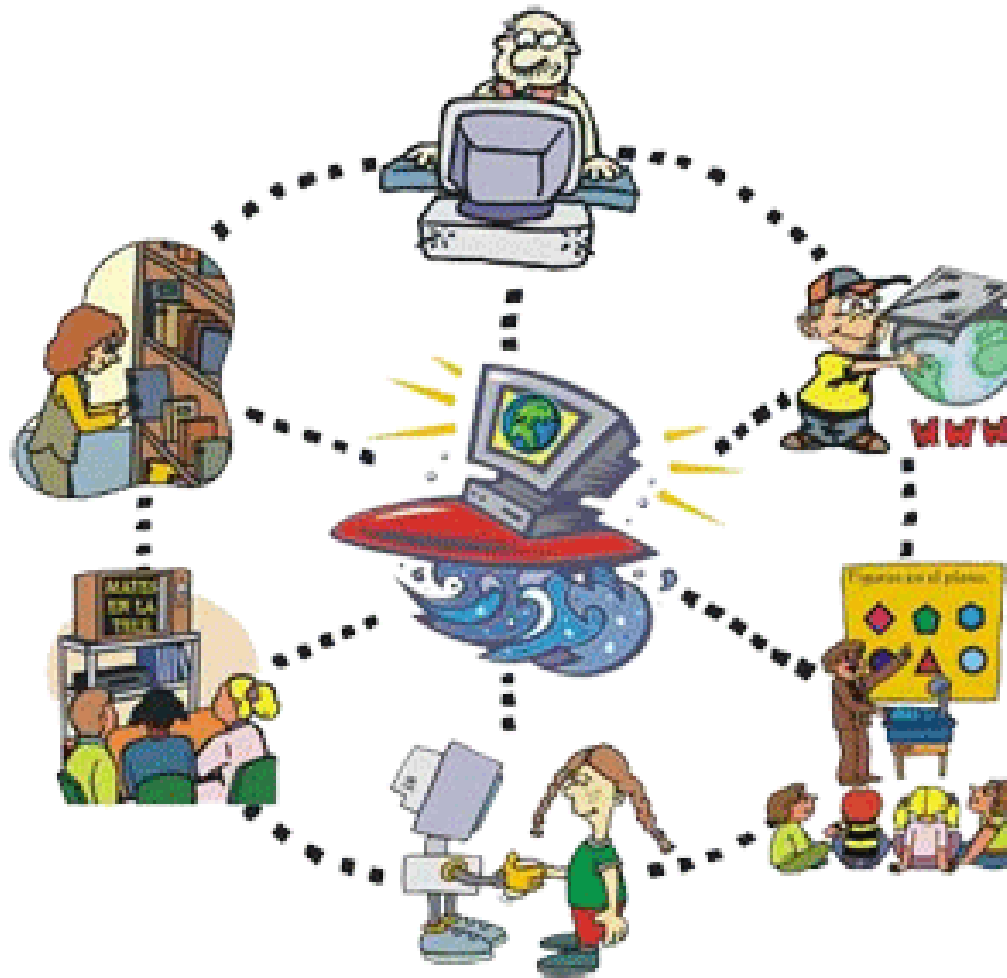
+

Lenguaje
Programación

=

Programa
Computacional

¿Dónde se aplica?



Empresas exitosas



Aplicaciones exitosas



Aplicaciones en la organización



Inteligencia de negocio (BI)



Gestión de Contenidos (ECM/CMS)



Relaciones con el Cliente (CRM)



Distribuciones Empresariales



Gestión Documental (DMS)



Comercio Electrónico



Recursos Empresariales (ERP/HRM)



Gestión Financiera



Servicios e infraestructuras



Software Colaborativo



Internet



Ofimática



Gestión de Proyectos



Punto de Venta



Utilidades

Tendencia hacia las aplicaciones móviles



Aspectos académicos

- ▶ Objetivos
- ▶ Competencias
- ▶ Contenidos
- ▶ Evaluación
- ▶ Referencias

Objetivos

- ▶ Desarrollar en los estudiantes las competencias necesarias para que sean capaces de resolver problemas de diversa índole mediante la programación de computadores.
- ▶ Se espera que los estudiantes aprendan a elaborar modelos y algoritmos que permitan resolver estos problemas usando el enfoque de orientación a objetos y a escribir programas que implementen estos algoritmos utilizando el lenguaje de programación Python (versión 3).

Competencias

Al finalizar el curso, los estudiantes deberán ser capaces de:

- ▶ Explicar conceptos básicos relativos a un programa computacional, tales como algoritmos, variables, expresiones, control de flujo, funciones, listas, strings, clases y objetos.
- ▶ Aplicar técnicas fundamentales para la resolución de diversos problemas con ayuda del computador, como identificar los datos relevantes de un problema y las relaciones entre ellos, modelar estos datos en una representación para un computador y descomponer problemas grandes en varios problemas más pequeños.
- ▶ Aplicar el razonamiento algorítmico para generar la solución a un problema como una secuencia de pasos bien definidos, incluyendo pasos condicionales, repetición de pasos, llamadas a funciones, y recursión.
- ▶ Llevar a cabo el proceso de desarrollo de programas, escribiendo y depurando programas orientados a objetos que satisfagan ciertas especificaciones, usando el lenguaje de programación Python.
- ▶ Usar un entorno de desarrollo de software para escribir, compilar y depurar programas.

Contenidos

1. Introducción a los algoritmos
2. El lenguaje de programación Python
3. Variables, expresiones
4. Control de flujo
5. Funciones
6. Programación orientada a objetos. Clases
7. Strings
8. Archivos
9. Listas
10. Ordenación y búsqueda
11. Recursión

Objetivos

- ▶ “Aplicar el razonamiento algorítmico para genera la solución a un problema como una secuencia de pasos bien definidos”
 - Nueva forma de razonar
 - Nuevo idioma
- ▶ **¡NO es el sistema que funciona en otros cursos!**
 - Solo Memorizar => REPRUEBO
 - Solo Entender => REPRUEBO
 - Estudiar los últimos días => REPRUEBO
- ▶ **Practicar, Practicar, Practicar ... cada semana**
 - No hace falta estudiar para el examen 😊

Evaluación

La evaluación del curso estará compuesta por:

- ▶ 2 Interrogaciones:
 - **I1** (lunes 25 septiembre 2017, 18:30 horas)
 - **I2** (martes 24 octubre 2017, 18:30 horas)
- ▶ 1 Examen:
 - **E** (jueves 23 noviembre 2017, 15:30 horas)
- ▶ 3 tareas:
 - **T1, T2, T3**
- ▶ Nota de participación:
 - **NP** (definida por cada sección)
- ▶ La nota final (**NF**) del curso se calcula en base a la siguiente fórmula:
 - **$NF = 15\% I1 + 15\% I2 + 30\% E + 10\% T1 + 10\% T2 + 10\% T3 + 10\% NP$**

Evaluación

- ▶ Para aprobar el curso, debes cumplir las siguientes condiciones:
 - Promedio ponderado de evaluaciones mayor o igual a 4.0:
 $(15\% \text{ I1} + 15\% \text{ I2} + 30\% \text{ E}) / 60\% \geq 4.0$
 - Promedio de tareas mayor o igual a 4.0:
 $(\text{T1} + \text{T2} + \text{T3}) / 3 \geq 4.0$
 - Promedio final mayor o igual a 4.0:
 $\text{NF} \geq 4.0$
- ▶ En otro caso, la nota final del curso será el mínimo entre **NF** y la nota **3.9**.
- ▶ En caso de inasistencia **justificada ante la Dirección de Pregrado** a **una** interrogación, esta nota será reemplazada por la nota de examen.

Nota de participación – sección 2

- ▶ Sistema de puntajes por actividades prácticas:
 - Laboratorios
 - Hackerrank
 - Pequeños ejercicios en clases
 - Ya explicaremos cómo ganar puntos



Fuente:
<https://m2.behance.net/rendition/pm/18733915/dispatch/df107dd98d8733d53c4fa6d750bf5343.png>

Evaluación

► Curso **COORDINADO!**

- Evaluaciones coordinadas; no dependen sólo de mí
- Correcciones coordinadas; no dependen de mí
- Tareas coordinadas; no dependen de mí
- Administración coordinada; no depende de mí

► Soy el **entrenador**, no el árbitro

- No influyo en las notas
- No influyo en las sanciones

► Coordinación

- Académica
- Administrativa



https://pbs.twimg.com/profile_images/714767079962976257/w8Dg0o2m.jpg

Entregas de tareas: no esperar al último día

- ▶ Entregar hasta las X hrs $\neq X+1$ segundo
 - Note que la hora de su reloj, computador, etc., puede estar desfasada con respecto al servidor.
- ▶ Entregar archivos requeridos
 - Si se solicitan archivos $.X \neq .Y$
- ▶ Entregar tarea que funcione
- ▶ **Realizar entregas parciales**
- ▶ Entregar trabajo propio y proteger trabajo propio

Entregas: no esperar al último día

- ▶ “Se me cortó la luz/internet y no pude enviar mi tarea”
 - ▶ “Me enfermé un día antes del plazo y no pude hacer la tarea”
 - ▶ “Voy a estar enfermo en dos días más”
 - ▶ “Pero solo le faltaba ‘:’ para que funcionara!”
 - ▶ “¡No me lo recordaron!”
 - ▶ “Se me perdió el archivo de la tarea”
-
- ▶ **Solución MUY simple a todo esto:
¡ trabajar con tiempo !**

Nacional

Copia masiva en la UC: 18 alumnos han reconocido falta durante examen

Universidad abrió un sumario por la transmisión de datos vía celular en prueba de Ingeniería.

Judith Herrera Cabello

/ 15/05/2015 - 05:00

317



14

url <http://fw.to/6FeEeNk>

-A

+A



Honestidad académica

- ▶ Cualquier forma de deshonestidad académica (por ejemplo, copia de código o copia en una evaluación escrita) podrá ser sancionada con la máxima rigurosidad permitida por el reglamento, esto es un **1.1 final en el curso**. Se informará a la Dirección de Pregrado de la Escuela del alumno para que ésta aplique las sanciones adicionales que estime convenientes.
- ▶ Este curso suscribe el Código de Honor (<http://www.ing.uc.cl/codigodehonor>) establecido por la Escuela de Ingeniería. En particular, si un estudiante no lo cumple, se le aplicará la Política de Integridad Académica del Departamento de Ciencia de la Computación.

Política de Integridad Académica del Departamento de Ciencia de la Computación

- ▶ Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile (y alumnos de los cursos dictados por esta Escuela) deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería.
- ▶ Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho **individualmente** por el alumno, **sin apoyo en material de terceros**. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.
- ▶ En particular, si un alumno copia un trabajo, o si a un alumno se le prueba que compró o intentó comprar un trabajo, **obtendrá nota final 1.1 (uno coma uno) en el curso** y se solicitará a la Dirección de Docencia de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral. Por “copia” se entiende incluir en el trabajo presentado como propio partes hechas por otro alumno. En este caso, la sanción anterior se aplicará a todos los alumnos involucrados. Por “compra” se entiende presentar como propio un trabajo hecho por otra persona.
- ▶ Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, **siempre y cuando se incluya la referencia correspondiente**.
- ▶ Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile. Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.

Política de Integridad Académica del Departamento de Ciencia de la Computación



Copia

▶ TRABAJO INDIVIDUAL!!

- Tareas, Hackerrank, y evaluaciones

▶ TOLERANCIA 0%

- Independientemente de **otras secciones**
- Independientemente de **otros semestres**
- Independientemente de **otros cursos**
- Independientemente de **otros alumnos**

▶ DURA SANCIÓN

- Puede llegar a la **reprobación** del curso con **1.1** y un **procedimiento sumario** severo

Copia

► Copia:

- Dar a alguien
- Recibir de alguien
- Facilitar pasivamente o por negligencia
- Trabajar juntos
- Comprar ayuda
- Modificar un trabajo ajeno
- Total o parcial
- ...

Copia

- ▶ “Solo quería ayudar a mi amigo”
- ▶ “No, nos conocemos”
- ▶ “Trabajé en su computador, y quedó allí”
- ▶ “No tenía internet, y se lo pasé para que la enviara”
- ▶ “Siempre trabajamos juntos”
- ▶ “Intenté explicárselo, pero como no tenía tiempo se lo envié”
- ▶ “Es el único amigo que tengo”
- ▶ “Me robó la tarea”
- ▶ “Le dije que la cambiara”
- ▶ “Solo le expliqué una pequeña cosa”
- ▶ “Se lo expliqué, pero no le mostré el código”
- ▶ “Es que tuve problemas personales/familiares/...”
- ▶ “Otros semestres/secciones/cursos/ ...”

Copia

- ▶ Sistema automático
 - Detecta cambiar nombres, orden, textos, ...
 - Código es como huella dactilar
- ▶ Curso coordinado
 - Se escapa de mis manos (no importa cuán bueno sea el alumno)
- ▶ Laboratorios todos los días
 - Muchos ayudantes dispuestos a apoyarte

Copia

- ▶ ¿Hay alguna duda?
- ▶ ¿Se ha entendido lo que es copia?
- ▶ ¿Se entiende la sanción posible?
- ▶ ¿Hay algún comentario?

Clases

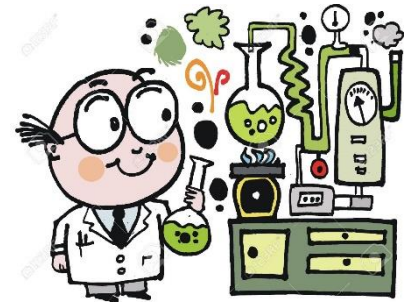
- ▶ Orientadas a resolución de problemas
 - Como afrontar/descomponer el problema
- ▶ Dinámicas y participativas



Fuente: <http://jorruva.blogspot.cl/2015/03/que-me-gustaria-que-pasara-siempre-en.html>

Laboratorios

- ▶ Resolver ejercicios en plataforma Hackerrank
- ▶ Flexible
 - Hay todos los días
 - Todos los días que quieras
 - Especialmente antes de tareas y evaluaciones
 - Control de asistencia
 - Parte de la evaluación de la **nota de participación**
 - Sin ir a los laboratorios el curso se hace más difícil
- ▶ Ayudantes
 - Desarrollan problemas en pizarra resueltos paso a paso
 - Resuelven dudas con ejercicios Hackerrank, tareas, evaluaciones, cátedra, ...
 - Apoyo individual y grupal
- ▶ Instalaciones
 - Salas de clase (con zapatillas) ← **MUY RECOMENDABLE**
 - Tu propio notebook y tu propio ambiente de desarrollo
 - Laboratorios
- ▶ Calendario
 - Empiezan la **semana del 21 de agosto**



Rol del alumno

► Venir preparados a clase

- Conceptos básicos de la última clase bien asimilados
- Serán aplicados en la clase

► Tener todo al día

- Hackerrank
- Tareas

► Practicar, practicar, practicar

- ¿Por qué no reservarse tiempo para practicar en computador después de cada clase?
- ¿Por qué no reservarse dos laboratorios a la semana?

INFORMACIÓN GENERAL

Equipo Docente

► Coordinación

- iic1103@uc.cl
- **Problemas, evaluaciones, certificados, tareas, ...**

► Profesor

- Marcos Sepúlveda – marcos@ing.puc.cl
- **Contenidos académicos**

► Ayudante de cátedra

- Juan Aguillón - jjaguillon@uc.cl
- **Aspectos sobre nota de participación**

Sitios web

- ▶ Toda la información relacionada al curso estará disponible en el sitio web del curso en SIDING:
 - Programa
 - Clases
 - Ejemplos clases
 - Evaluaciones anteriores
 - Avisos
 - Notas

- ▶ Las notas parciales se publicarán en Google Drive

Referencias

1. Python software foundation, *Python v3 Documentation*, <http://docs.python.org/3>.
2. V. Ceder. *The quick python book*. Manning Publications Co., 2010.
3. A. B. Downey. *Think Python: How to think like a computer scientist*. Green Tea Press, 2013.
Versiones web:
 - <http://greenteapress.com/wp/think-python>
 - <http://interactivepython.org/courselib/static/thinkcspy/index.html>
4. J. M. Zelle. *Python programming: An introduction to computer science*. Franklin, Beedle & Associates, Inc., 2nd edition, 2010.
5. *Py-Libre, Apunte interactivo para el curso Introducción a la Programación*, <http://runest.ing.puc.cl>.

Otras posibles referencias en Castellano

- ▶ “Apuntes del Curso IIC1103 - Introducción a la Programación con Python” – Adrián Soto, Mauricio Arriagada
 - SIDING – “Material Complementario”

- ▶ <https://wiki.python.org/moin/SpanishLanguage>
 - Buen punto de partida

- ▶ “Tutorial de Python” - release 3 – Guido van Rossum
 - <http://docs.python.org.ar/tutorial>
 - <http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf>
 - Traducido al Argentino

Créditos

- ▶ El material de esta sección fue preparado a partir de las clases del semestre 2014-01 del profesor Mauricio Arriagada (marriagb@uc.cl).



¡ Gracias Mauricio !

CIERRE

Al cierre

- ▶ Los laboratorios empiezan la **semana del 21 de agosto**
- ▶ Mirar <https://www.youtube.com/watch?v=U3CGMyjzlvM> para el martes