DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

INTRODUCCIÓN

10110 - FUNDAMENTOS DE COMPUTACIÓN Y PROGRAMACIÓN

CONOCIÉNDONOS

- DOCENTE

Nombre	PROFESOR
Correo@usach.cl	CONTACTO
Sección	SECCIÓN

INFORMACIÓN ADMINISTRATIVA

DATOS DEL CURSO

- 10110 – Fundamentos de Computación y Programación
- Vía oficial de comunicación para el curso: https://uvirtual.usach.cl/
- La inscripción a la plataforma se hará automáticamente la segunda semana del curso
- Repositorio temporal de material: https://github.com/PROGRA-FING-USACH/2023-1/

EQUIPO DE COORDINACIÓN

- Alejandro Cisterna
  Alejandro.Cisterna@usach.cl
- Luciano Hidalgo
  Luciano.Hidalgo@usach.cl
- Javier Salazar
  Javier.Salazar.L@usach.cl

UNIDADES TEMÁTICAS

UNIDAD	TÍTULO	N° DE HORAS PRESENCIALES
1	FUNDAMENTOS DE PROGRAMACIÓN	24
2	FUNCIONES Y ABSTRACCIÓN	15
3	PROGRAMACIÓN PARA LA INGENIERÍA	24
TOTAL	15 SEMANAS	63

HORAS DE DEDICACIÓN QUE REQUIERE LA ASIGNATURA				
TEORÍA (PRESENCIALES)	EJERCICIOS	LABORATORIO	AUTOESTUDIO (NO PRESENCIALES)	TOTALES
42 (2.6 semanales)	0	21 (1.3 semanales)	97 (6 semanales)	160

TEORÍA

MODALIDAD DE TRABAJO

image: Flaticon.com

- Cómo se puede ver, el curso exige 6 horas de dedicación temporal **fuera del aula** a la semana
- Con esto en mente, trabajaremos con aula invertida
- En particular, entre los deberes de la/el estudiante esta:
  - **Prepararse** para la sesión de clases: Leyendo el material, resumiendo los contenidos y anotando sus dudas
  - **Participar** durante la clase, junto con su grupo y resolviendo sus dudas
  - Desarrollar los ejercicios de **práctica** que quedan para la semana

ESTRUCTURA DE CLASE

PREPARACIÓN (PREVIO A LA CLASE)

Lectura – Google Colab

Guía de práctica - Replit

APLICACIÓN (DURANTE LA CLASE)

Test de entrada

Profesor resume el contenido y resuelve dudas

Estudiantes resuelven grupalmente ejercicio(s)

PRÁCTICA (DESPUÉS DE LA CLASE)

Estudiantes envían respuestas de la guía para calificación

EL ROL DEL PROFESOR

- El profesor de teoría está para resolver dudas y plantear desafíos
- De todos modos, el profesor **resumirá** y **recordará** los elementos esenciales para el trabajo del día
- La idea es que no esté contándoles nuevamente la materia que (se supone) ya leyeron
- **¡La única forma de aprender a programar es practicando!**

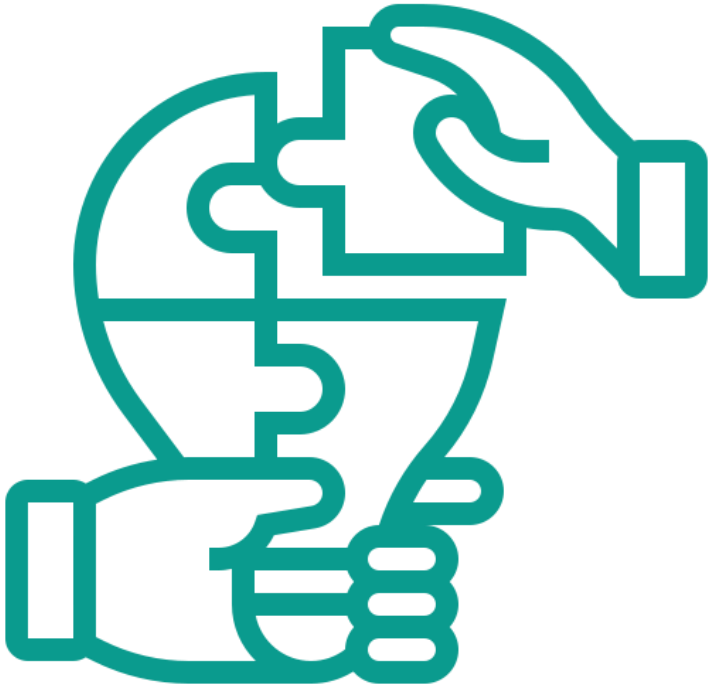


image: Flaticon.com

EVALUACIÓN TEORÍA

- El promedio de teoría se calcula según:
  - **Test de entrada (15%)** – 1 por cada lectura (14 en total)
  - **Guías de ejercicios (25%)** – 11 Guías al semestre
  - **3 Tareas (10%, 20% y 30%)** – se publican a las 8 AM y tiene plazo de 5 o 6 días (dependiendo de la tarea) para entregarla
- Adicionalmente se requiere un **75% de asistencia como mínimo para aprobar** la asignatura
- Todas las evaluaciones de teoría son **individuales**

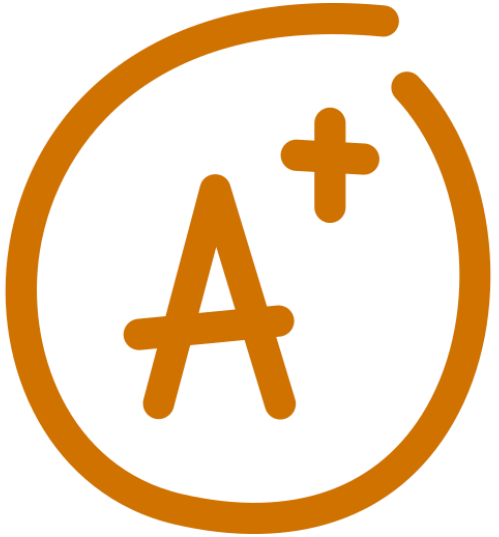


image: Flaticon.com

TEST DE ENTRADA

image: Flaticon.com

- Para asegurar que los estudiantes están leyendo previo a la clase, al inicio de esta se controlará la lectura con un test de entrada
- Durante el semestre son 14 lecturas, cada una con un test de entrada
- Cada test corresponde a 2 o 3 preguntas de selección múltiple
- Se **elimina la peor de las 14 notas**
- Los test se hacen al inicio de la clase, por lo que estudiantes atrasados o ausentes al momento del test **obtienen nota mínima** en este

GUÍAS DE EJERCICIOS

- 11 Guías semanales de 7 ejercicios
- Cada guía tiene una semana de plazo para ser desarrollada
- El estudiante puede probar el código y revisar sus respuestas en la plataforma `replit`

GUÍAS DE EJERCICIOS

- Respecto a la nota:
  - Cada ejercicio vale 1 punto y se corrigen automáticamente. Si la/el estudiante tiene el ejercicio bueno consigue un punto, en este **caso no hay puntaje parcial por ejercicios incompletos** o **que superen algunos (pero no todos)** los *testcases*
  - Para obtener un 7,0 en guías, se requieren 67 puntos del máximo de 77
  - Para el 4,0 se requieren 40 puntos
- Es decir, existen 10 puntos extra, por lo mismo, **la no entrega de una guía significa la pérdida de los puntos de esa semana**

TAREAS

- **Tarea 1 (10%): 17 de abril**
  - Elementos básicos de programación (tipos de datos, variables, I/O)
  - Estructuras de control (If, While, For)
  - Tipos de datos compuestos (Listas/Strings)
- **Tarea 2 (20%): 08 de mayo**
  - Funciones (nativas, importadas, propias)
  - Recursión
- **Tarea 3 (30%): 19 de junio**
  - Archivos (Texto plano)
  - Pandas
  - Pyplot

image: Flaticon.com

EVALUACIÓN TEORÍA

- Asistencia mínima exigida
  - Para aprobar la teoría del curso, se exige un **75% mínimo de asistencia**
  - Estudiantes que no alcancen el mínimo de asistencia exigida, aún en caso de que consigan la nota mínima de aprobación serán calificados con nota 3,5 bajo el concepto de NCA (No cumple asistencia)
  - Si el estudiante reprobase por asistencia y por calificación se aplicará la menor entre las dos notas
  - El profesor puede pasar lista en **cualquier momento de la clase (varias veces si lo desea)**, si el estudiante no está presente en ese punto, **se considera ausente**

image: Flaticon.com

EVALUACIÓN DE TEORÍA

- Prueba Recuperativa
  - Tendrá derecho a rendirla toda/o estudiante que, justifique su inasistencia o no entrega de Tarea y sea autorizado por la **Secretaría de Docencia de la Facultad de Ingeniería**
  - La evaluación recuperativa sólo considera los contenidos de la evaluación a la que la/el estudiante ha faltado
  - La/El estudiante solo puede rendir **UNA prueba recuperativa** durante el semestre
  - La **prueba recuperativa** se rinde **presencialmente** la última semana del semestre

EVALUACIÓN ADICIONAL OPTATIVA (EAO)

- Estudiantes que a final de semestre no hayan obtenido un 4,0 con el promedio ponderado de las evaluaciones tiene derecho a rendir la **evaluación adicional optativa**, siempre y cuándo cumpla con los requisitos:
  - 75% asistencia
  - No tener sanciones por faltas al código de honor del curso
  - Promedio mínimo de presentación 3,0

- Esta evaluación contempla **toda la materia del semestre**
  - La nota final se calcula 60% promedio del semestre + 40% nota EAO
  - No se eliminan notas para el cálculo del promedio final

CALIFICACIÓN FINAL

- Se calculará la nota final de la asignatura de acuerdo a:
  - Si las notas finales de teoría y de laboratorio son **ambas mayores** o iguales a **4,0** entonces:
    - Nota Final = **0,5** * Nota Cátedra + **0,5** * Nota Laboratorio
  - Si alguna de las notas finales de laboratorio o de cátedra es menor a 4,0, entonces:
    - Nota Final = **Menor Nota**(Nota Cátedra, Nota Laboratorio)
  - Si la/el alumna/o aprueba sólo teoría o sólo laboratorio, la nota se guarda por **un semestre** y al semestre siguiente **sólo rinde** la parte de la asignatura que reprobó

CÓDIGO DE HONOR

CÓDIGO DE HONOR

image: Flaticon.com

- En orden de aprobar el curso, las y los estudiantes deben demostrar un comportamiento ético y acorde a los principios de integridad estudiantil
- En particular, se espera que todo trabajo que la/el estudiante presente este semestre sea de su única y exclusiva autoría
- Cualquier intento de obtener una ventaja injusta en un proceso de evaluación constituye una falta a la integridad académica
- La/El estudiante arriesga sanciones en caso de no ceñirse a estos principios que van desde nota 1,0 en la evaluación a la reprobación directa del curso
- **La/El estudiante debe leer el código de honor y declarar que está en conocimiento de este durante las primeras dos semanas del curso**

¿QUE NO PUEDO HACER?

- Pagar/cobrar para resolver una evaluación
- Copiar a otras personas o plagiar
- Compartir código durante una evaluación
- Trabajar en grupos (o parejas) en actividades evaluadas
- Entre otras

image: Flaticon.com

SANCIONES ANTE FALTAS A LA ÉTICA

- **Primera falta:**
  - 1,0 en la evaluación
  - Pierde el derecho a rendir EAO
  - Se informa a la dirección de docencia de la FING y a la subdirección de su Dpto. Académico para anotación en hoja de vida

- **Segunda falta (considerando sanciones de otras asignaturas o de semestres anteriores):**
  - Reprobación directa de la teoría o laboratorio (según donde se haya cometido la falta)
  - Se informa a la Dirección de Docencia para inicio de procedimiento disciplinario

¿CÓMO SE DETERMINAN LAS SANCIONES?

- Ante sospechas de faltas al código de honor, la coordinación determinará una comisión revisora, compuesta por 3 miembros del cuerpo docente para revisar el caso
- Dicha comisión citará a las/los estudiantes involucrados para recolectar antecedentes
- Con ello se emitirá un informe con el cuál la coordinación determinará si hubo una falta y la sanción correspondiente
- Se informará del resultado al estudiante vía correo institucional

¿Y SI QUIERO APELAR?

image: Flaticon.com

- La/El estudiante puede apelar a la Subdirección de Docencia del Departamento de Ingeniería Informática, donde una comisión de tres profesoras/es externos revisarán el caso

- Una vez revisada la apelación, se le informará vía correo institucional del resultado de ella, instruyendo a la coordinación del curso del cambio, según corresponda

¿QUÉ APRENDERÉ EN LA ASIGNATURA?

OBJETIVOS DEL CURSO

- El principal objetivo del curso es **desarrollar la capacidad de resolución de problemas usando herramientas de programación**
- Para ello, se requiere el desarrollo del pensamiento computacional y sus habilidades asociadas
- ¿Qué es el pensamiento computacional?
- ¿Qué habilidades necesito desarrollar para fortalecer mi pensamiento computacional?

PENSAMIENTO COMPUTACIONAL

- Es difícil de definir, pero diremos que el pensamiento computacional podría entenderse cómo:
  - Un **conjunto de habilidades** que permiten **abstraer un problema** y expresar su solución de forma tal que un computador pueda resolverlo
  - El proceso de **reconocer aspectos de la computación** en el mundo que nos rodea y aplicar técnicas y herramientas para entender y razonar acerca de sistemas y procesos naturales o artificiales
  - Una **orientación mental** para formular problemas y transformar entradas en las salidas deseadas a través del uso de algoritmos para realizar las transformaciones


Generalización (Reconocimiento de patrones)
Abstracción
Modelamiento
PENSAMIENTO COMPUTACIONAL
Pensamiento algorítmico
Descomposición
Pensamiento lógico
Evaluación

¿PARA QUÉ SIRVE EL PENSAMIENTO COMPUTACIONAL?

- La principal ventaja de desarrollar el pensamiento computacional es que este nos vuelve mejores a la hora de **resolver problemas**
- Al manejar herramientas como el **modelamiento** y la **abstracción**, somos capaces de seleccionar los aspectos relevantes de un problema y representarlo en términos sencillos
- **Generalizar** y **descomponer** nos permite adecuar soluciones exitosas de otros contextos a nuevos cuerpos de conocimiento, generando innovaciones

¿PARA QUÉ SIRVE EL PENSAMIENTO COMPUTACIONAL?

- Las habilidades de **pensamiento lógico** y **algorítmico** nos permiten encontrar oportunidades para la automatización y entender estados de procesos, maquinarias y otros elementos del mundo real

- Las habilidades de **evaluación** nos apoyan en el desarrollo de pensamiento crítico y nos permiten generar criterios de evaluación del éxito de procesos, proyectos, soluciones y otros elementos del mundo real

PROGRAMACIÓN

- Un **usuario de aplicaciones** computacionales no es lo mismo que un **usuario profesional** de computadores
- En este curso aprenderás a **comunicarte** con el computador
- La programación básicamente es: **explicarle** al computador qué es lo quieres que él haga, en un **lenguaje que el computador entienda**
- No es una ciencia, ¡es una **habilidad**!, como tocar un instrumento o saber conducir un vehículo

SISTEMA BINARIO

- Un *bit* es la unidad mínima de la electrónica digital, **dos estados de voltaje** y nosotros los representamos con los valores 1 y 0
- Esto significa que el computador, en un nivel muy básico, **solo opera sobre 0's y 1's**
- Esto es importante, porque finalmente un computador es una máquina cuyo objetivo es realizar **operaciones aritméticas** (como sumas, restas, multiplicaciones y divisiones) y **lógicas** (como conjunciones, disyunciones y negaciones) sobre bits
- ¡La ventaja es que el computador realiza estas operaciones muy rápidamente y por eso puede obtener resultados a cálculos complejos en tiempo récord!

SISTEMA BINARIO

- A raíz de la velocidad de cálculo de los computadores, los programadores se las han ingeniado para representar sólo con 0's y 1's elementos más complejos, tales como:
  - Letras
  - Números enteros y no enteros
  - Imágenes
  - Vídeos

- A fin de poder aprovechar las **capacidades de cálculo** para manejar texto, manipular imágenes, realizar cálculos complejos, reproducir vídeo y juegos de vídeo, entre otros

LENGUAJES DE PROGRAMACIÓN

- Los lenguajes de programación son la forma en que comunicamos a los computadores las **instrucciones** que queremos que **ejecuten**

- Como los humanos no podemos usar binario puro para darle instrucciones al computador, y el computador no es capaz (aún) de seguir a la perfección instrucciones en lenguaje natural, los **lenguajes de programación** son puntos intermedios entre la forma en que se comunica la máquina y nuestro lenguaje

- Existen lenguajes de programación para diversos propósitos, de distinta complejidad y documentación disponible, por eso en este curso partiremos aprendiendo un lenguaje de **propósito general** y **multiparadigma** llamado **Python**

PYTHON

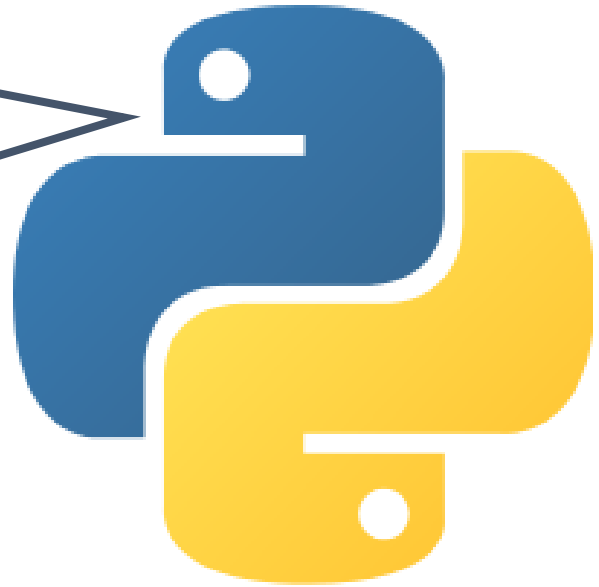
Python es un lenguaje:

- **De propósito general**, por lo que es útil para realizar múltiples tareas y desarrollar distintos tipos de software, y no está orientado a un conjunto de problemas en específico
- **Multiparadigma**, es decir, permite trabajar sobre distintos enfoques o paradigmas de programación
- **Interpretado**, lo que implica que el código en Python se transforma en código de máquina cuando se necesita la instrucción en específico

PYTHON

¡¡Y lo más importante de todo!!

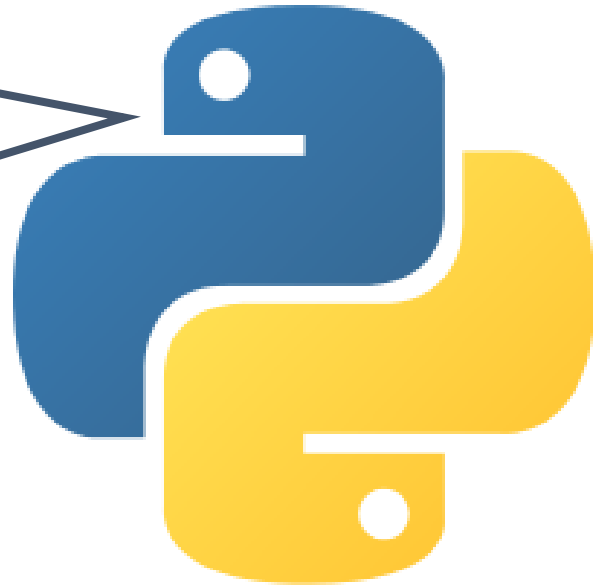
Python es un excelente lenguaje para aprender, a programar pues al ser un **lenguaje de alto** nivel, su sintaxis se parece más a la de **nuestro lenguaje** que al de la máquina



PYTHON

¡Este semestre usaremos Python 3.9 o superior!

Pueden revisarlo en el sitio oficial de Python
https://www.python.org/



...r

...stintos

...n se
...sita la

TAREAS PARA TRABAJO AUTÓNOMO

1. Revisar el apunte:
   - Tipos de datos, operadores y expresiones en Google Colab (Disponible en: https://github.com/PROGRA-FING-USACH/2023-1/blob/main/Lecturas/01_Introduccion_a_Python.ipynb)
2. Revisar el Código de Honor y otros documentos en: https://github.com/PROGRA-FING-USACH/2023-1/tree/main/Documentos
3. Suscribirse al calendario de evaluaciones en Google Calendar **(Opcional):** https://calendar.google.com/calendar/u/4?cid=Y18yZWU4YjU1NmE1NTM1Y2FmYmFkNzBlNTJlOWQ3ZWJlNjBlN2M1Njk5MjQwNTE5N2ZmNGYyMmE4ZmQzMDliNjRiQGdyb3VwLmNhbGVuZGFyLmdvb2dsZS5jb20

¿CONSULTAS?