



INTRODUCCIÓN

10145 - FUNDAMENTOS DE PROGRAMACIÓN PARA INGENIERÍA

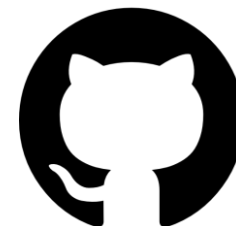


PRESENTACIÓN DEL DOCENTE



CONOCIÉNDONOS

Nombre	John Serrano Carrasco
Correo@usach.cl	john.serrano@usach.cl
Sección	10145-A-2



- Ayudante de Fundamentos de Programación Para Ingeniería (2022-2)
- Ayudante de Estadística Computacional (2022-2 y 2023-1)
- Ayudante de Técnicas de Ingeniería de Software (2023-1)

- Ayudante de Fundamentos de Computación (2024-1 y 2024-2)
- Ayudante de Gestión de Servicios TI (2024-1 y 2024-2)
- Ayudante de Gobernanza y Gestión TI (2024-2)



INFORMACIÓN ADMINISTRATIVA



DATOS DEL CURSO

- 10145 – Fundamentos de Programación para Ingeniería
- 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 del curso:
<https://github.com/PROGRA-FING-USACH/Material>



EQUIPO DE COORDINACIÓN

- Alejandro Cisterna
- Luciano Hidalgo
- Javier Salazar

—Contacto: programacion.diinf@usach.cl



UNIDADES TEMÁTICAS

UNIDAD	TÍTULO	N° DE HORAS PRESENCIALES
1	FUNDAMENTOS DE PROGRAMACIÓN	36
2	FUNCIONES Y ABSTRACCIÓN	24
3	PROGRAMACIÓN PARA LA INGENIERÍA	30
TOTAL	15 SEMANAS	90

HORAS DE DEDICACIÓN QUE REQUIERE LA ASIGNATURA				
TEORÍA (PRESENCIALES)	EJERCICIOS	LABORATORIO	AUTOESTUDIO (NO PRESENCIALES)	TOTALES
42 (2.6 semanales)	0	42 (2.6 semanales)	96 (6 semanales)	180



TEORÍA



MODALIDAD DE TRABAJO



- 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



APLICACIÓN (DURANTE LA CLASE)



Actividades en clase
(evaluada)



Profesor
resume el
contenido y
resuelve
dudas
Estudiantes
resuelven
grupalmente
ejercicio(s)



PRÁCTICA (DESPUÉS DE LA CLASE)

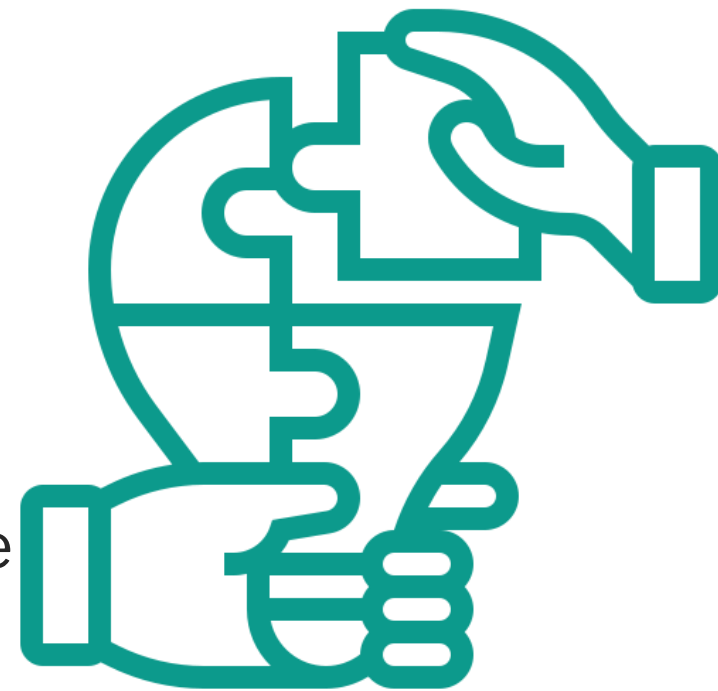


Guías de ejercicios
(Formativas)



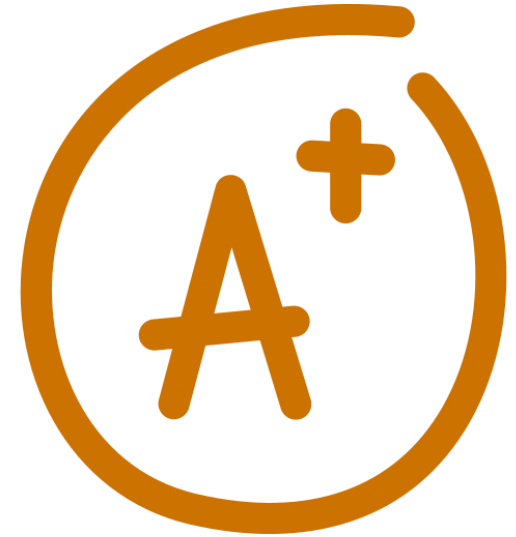
EL ROL DEL PROFESOR

- 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!**



EVALUACIÓN TEORÍA

- El promedio de teoría se calcula según:
 - **[A] Actividades en clases (15%)** – 1 por cada clase (Se calcula el promedio con los 18 mejores)
 - **[E] Ensayos de prueba (5%)** – Ensayos evaluados automáticamente para preparar la prueba
 - **[P] Pruebas (40%)** – Evaluaciones presenciales que se rinden en horario de laboratorio
 - **[T] Tareas (40%)** – Trabajos de larga duración, que deben ser desarrollados en tiempo de trabajo autónomo
- Se requiere adicionalmente un **75% de asistencia** para aprobar





[A] ACTIVIDADES EN CLASES



- Para asegurar que los estudiantes están leyendo y practicando en todas las clases se realiza una actividad evaluada, la cuál puede ser:
 - Test de lectura
 - Preguntas de seguimiento de código
 - Evidenciar el trabajo en clase individual o grupal
- **Cada clase** se realiza una actividad evaluada
- El profesor determina si la hace al inicio, al final o en el transcurso de la clase
- Se calcula esta nota con las **mejores 18 actividades**



[E] ENSAYOS (5%)

$$E = (E1 + E2)/2$$

- La semana anterior a la prueba se liberará un ensayo de prueba
- Se darán 5 días para desarrollarlo
- Al final de la semana se entregarán los puntajes
- Se evaluará **solo el algoritmo** en los ensayos, es decir, **si el programa pasa o no pasa** los casos de prueba



[P] PRUEBAS (40%)

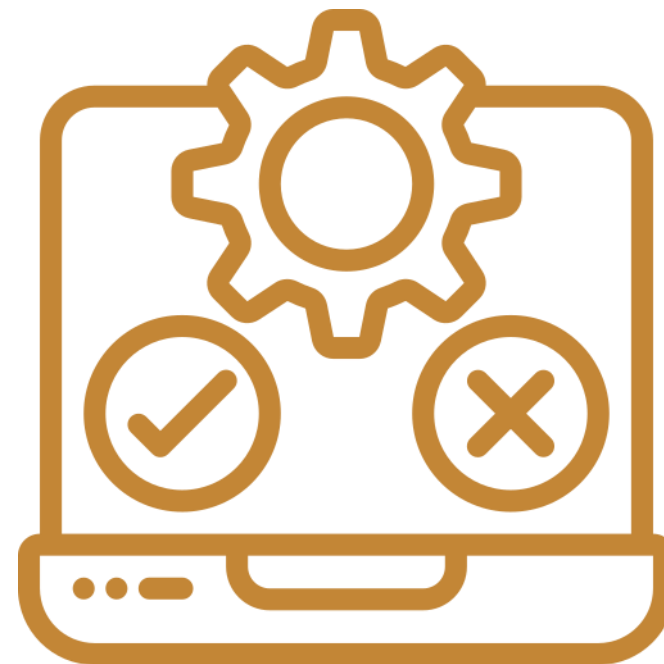
$$P = (P1 + P2)/2$$

- **P1: Semana del 4 de noviembre**

- Elementos básicos de programación (tipos de datos, variables, I/O)
- Estructuras de control (If, While, For)
- Tipos de datos compuestos (Listas/Strings)

- **P2: Semana del 2 de diciembre**

- Funciones (nativas, importadas, propias)
- Archivos
- Toda la materia anterior





[T] TAREAS (40%)

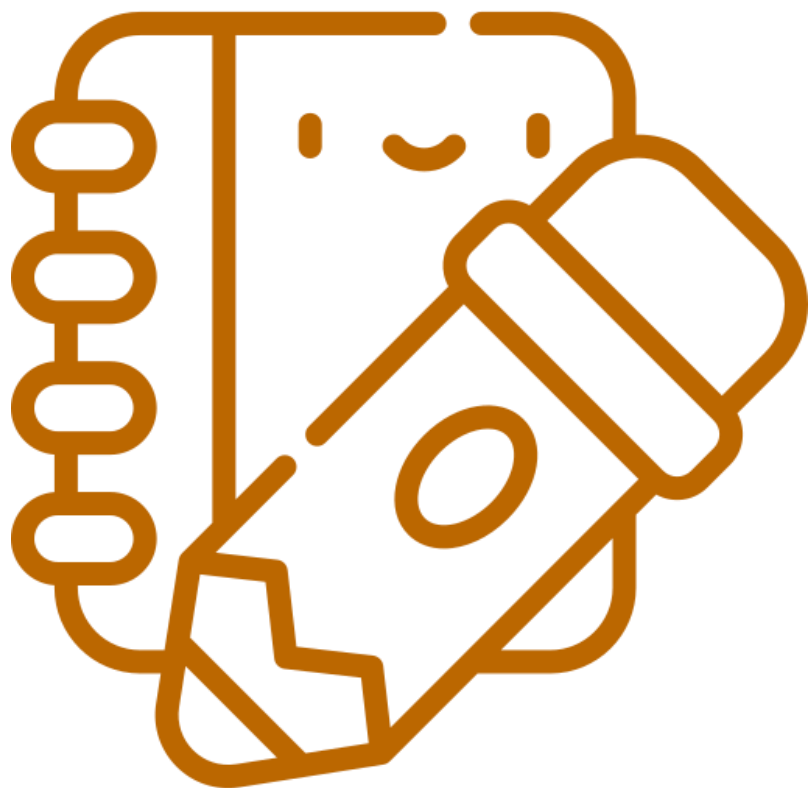
$$T = (T1 + T2)/2$$

- **T1: Entrega 16 de noviembre**

- Elementos básicos de programación (tipos de datos, variables, I/O)
- Estructuras de control (If, While, For)
- Tipos de datos compuestos (Listas/Strings)
- Funciones nativas e importadas

- **T2: Entrega 22 de diciembre**

- Funciones (nativas, importadas, propias)
- Archivos
- Recursión
- Toda la materia anterior





SISTEMA DE EVALUACIÓN

- Todos los ejercicios tienen un enunciado con una especificación precisa que debe ser cumplida en su totalidad para obtener el puntaje
- Por ejemplo, si el enunciado dice: *El programa debe imprimir el mensaje “Hola mundo” con mayúscula al inicio y sin espacios adicionales*, solo se considera correcto:
 - Hola mundo
- Mientras que variaciones como:
 - Hola Mundo
 - hola mundo
 - Hola mundo
- Estarían incorrectas, por no cumplir con la especificación indicada. Un algoritmo corrige automáticamente y asigna parte del puntaje



ASISTENCIA



- Se requiere un mínimo de **75%** de asistencia en orden de aprobar
- Estudiantes quienes no alcancen el mínimo de asistencia reprueban con:
 - **Mínimo(calificación obtenida, 3,5)**
- El profesor está autorizado para tomar asistencia en el momento de la clase que él estime conveniente (incluso **más de una vez** en la misma clase de requerirlo)



EVALUACIÓN DE TEORÍA

- Prueba Recuperativa
 - Tendrá derecho a rendirla toda/o estudiante que, justifique su inasistencia a una evaluación 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



CÁLCULO DE LA NOTA FINAL

- El promedio de teoría se calcula como:
 - $p_{\text{teoría}} = 0.15 * A + 0.05 * E + 0.4 * T + 0.4 * P$
 - Siempre y cuando el estudiante cumpla:
 - $T \geq 4.0$ y $P \geq 4.0$ y Asistencia $\geq 75\%$
 - 0:
 - $p_{\text{teoría}} \geq 4.5$ y Asistencia $\geq 75\%$
- En caso contrario, se debe rendir **Examen [Ex]**:
 - Esta evaluación contempla **toda la materia del semestre**
 - La nota final se calcula: $0.6 * p_{\text{teoría}} + 0.4 * Ex$
 - **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:
 - $\text{nota_final} = 0.5 * p_{\text{teoría}} + 0.5 * p_{\text{laboratorio}}$
 - Si alguna de las notas finales de laboratorio o de cátedra es menor a 4,0, entonces:
 - $\text{nota_final} = \text{minimo}(p_{\text{teoría}}, p_{\text{laboratorio}})$
- En caso de reprobación una (Teoría o Laboratorio), al semestre siguiente el estudiante solo rinde la parte reprobada



CÓDIGO DE HONOR



CÓDIGO DE HONOR



- 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**. Sin mediación de ayuda externa
- Cualquier intento de obtener una **ventaja injusta** en un proceso de evaluación constituye una falta a la integridad académica
- **La/El estudiante debe leer el código de honor y declarar que está en conocimiento de este durante la primera semana del curso**



ARREPENTIMIENTO TEMPRANO

- Usé IA o le pedí ayuda a un compañero, pero me arrepentí ¿Qué puedo hacer?
 - Si cometiste alguna falta al código de honor, puedes acogerte a la **cláusula de arrepentimiento temprano**
 - Con ello, **reconoces haber cometido alguna falta**, pero solicitas que se tenga consideración por tu caso
 - **Esta política aplica solamente dentro del período estipulado y no después de que se ha notificado al estudiante de una sospecha**

¿QUE NO PUEDO HACER?

- Pagar/cobrar para resolver una evaluación
- Trabajar en grupos (o parejas) en actividades evaluadas individuales (“Comparar respuestas”)
- Usar código de internet para una evaluación
- Usar ChatGPT para “tener una idea” o “corregir una parte del código”



image:
[Flaticon.com](https://www.flaticon.com)

SANCIONES ANTE FALTAS A LA ÉTICA



- **Primera falta con arrepentimiento temprano:**
 - 1,0 en la evaluación (**Para ensayos o actividades en clases se aplica al porcentaje completo**)
- **Primera falta sin arrepentimiento temprano:**
 - Reprobación directa de teoría con calificación mínima
- **Faltas recurrentes (considerando sanciones de otras asignaturas o de semestres anteriores):**
 - Reprobación directa de la teoría y laboratorio
 - Anotación en hoja de vida del estudiante



FREQUENTLY ASKED QUESTIONS (FAQ)

- ¿Cuántos casos relacionados a faltas al código de honor son sancionados?
 - El semestre pasado 55 estudiantes (10% aprox.)
- ¿Bajo qué criterios?
 - Principalmente:
 - Uso de herramientas de IA para generar parte de la solución o corregir la solución
 - Copia entre compañeros
 - Uso de un código base (página web, tutorial de youtube)
 - Fuentes externas (Ej: *“Mi abuelita es informática y me ayudó”*)



FREQUENTLY ASKED QUESTIONS (FAQ)

- ¿Y si quiero apelar?
 - La/El estudiante puede apelar a la Subdirección de Docencia del Departamento de Ingeniería Informática, donde una comisión de profesoras/es externos al curso 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
 - Si el estudiante es sancionado con anotación a la hoja de vida, puede solicitar la remoción de su anotación de demérito luego de 4 semestres de ocurrida la falta





¿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



¿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 que 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





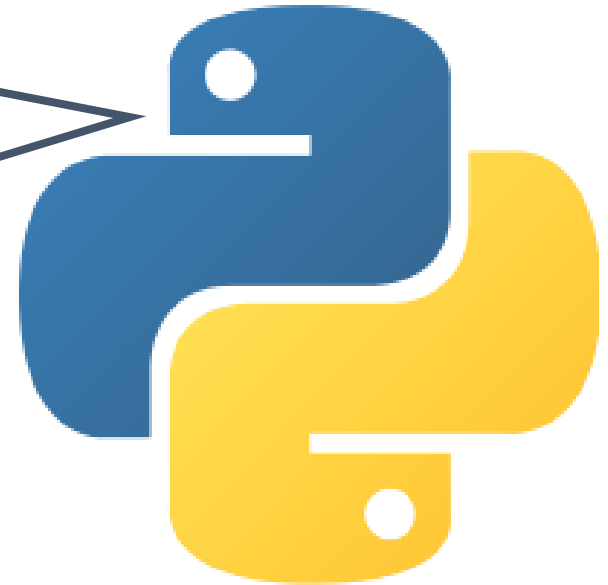
PYTHON

¡¡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

Realizar

pre
ción
Python
e





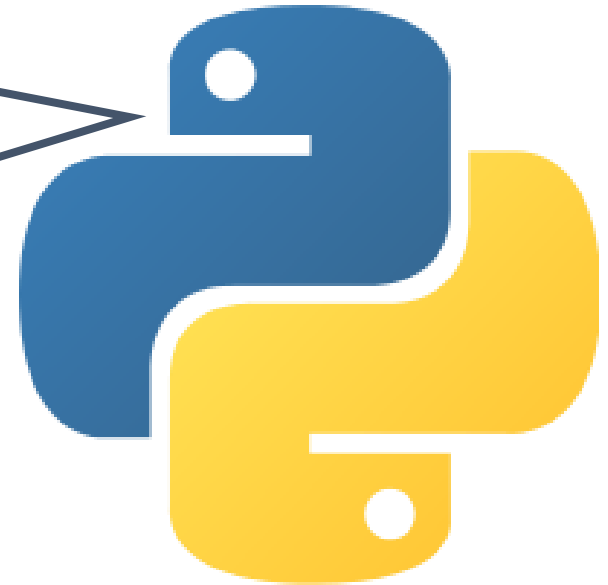
PYTHON

¡Este semestre usaremos **Python 3.10** o superior!

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

Realizar

pre
ción
Python
e



TAREAS PARA TRABAJO AUTÓNOMO



DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

1. Responder el compromiso del código de honor



2. Responder el diagnóstico de la asignatura





TAREAS PARA TRABAJO AUTÓNOMO

3. Revisar el apunte antes de la próxima clase el apunte de “Tipos de datos, operadores y expresiones” (Disponible en: https://github.com/PROGRA-FING-USACH/Material/blob/main/Lecturas/01_Introduccion_a_Python.ipynb)
4. Revisar el Código de Honor en: https://progra-fing-usach.github.io/active/honor_code



¿CONSULTAS?