

Programación para Física y Astronomía

Departamento de Física.

Corodinadora: C Loyola

Profesores C Femenías / F Bugini / D Basantes

Primer Semestre 2025

Universidad Andrés Bello

Departamento de Física y Astronomía



Introducción y Objetivos

Repaso de Contenidos Clave

Actividades de Repaso

Preguntas de Alternativas

Programas Entretenidos

Introducción y Objetivos

- Revisar y consolidar los contenidos previos (Unidades I a IV).
- Resolver dudas y reforzar conceptos fundamentales: sintaxis, estructuras de control, funciones, módulos y uso de librerías.
- Desarrollar actividades interactivas y colaborativas que estimulen el pensamiento algorítmico.
- Practicar mediante ejercicios de opción múltiple y programas entretenidos que integren los conceptos aprendidos.

Repaso de Contenidos Clave

Repaso de Contenidos Clave ∈ Resumen de Unidades y Conceptos

- **Elementos Básicos**

Diseño de programas, uso de GNU/Linux, Google Colab y el intérprete Python.

- **Programación en Python**

Tipos de datos, operaciones, manejo de ficheros, funciones, paquetes y módulos.

- **Iteradores**

Uso de bucles (`while` y `for`).

- **Funciones**

Implementación de funciones.

Actividades de Repaso

A continuación se presentan algunas preguntas de opción múltiple para autoevaluar conocimientos:

1. ¿Cuál es el propósito principal de utilizar funciones en Python?
 - A) Facilitar la lectura y reutilización del código.
 - B) Incrementar la velocidad de ejecución del programa.
 - C) Permitir la interacción con bases de datos.
 - D) Aumentar la complejidad del código.
2. ¿Qué operador se utiliza para multiplicar en Python?
 - A) /
 - B) x
 - C) %
 - D) *

Actividades de Repaso ∈ Ejercicio: Preguntas de Alternativas - Elementos Básicos y GNU/Linux

1. ¿Cuál de los siguientes es un componente esencial de un sistema operativo GNU/Linux?
 - A) El kernel.
 - B) La aplicación de ofimática.
 - C) El navegador web.
 - D) El entorno de escritorio exclusivo.
2. ¿Qué ventaja ofrece Google Colab para quienes inician en la programación?
 - A) Permite acceder a recursos computacionales gratuitos en la nube.
 - B) Requiere la instalación local de paquetes costosos.
 - C) Solo funciona con el intérprete de C++.
 - D) No permite compartir el código entre usuarios.

Actividades de Repaso ∈ Ejercicio: Preguntas de Alternativas - Programación en Python: Tipos de Datos

1. ¿Cuál de los siguientes tipos de datos no existe en python?
 - A) int
 - B) float
 - C) fractal
 - D) complex
2. ¿Qué función se utiliza para determinar el tipo de dato de una variable en Python?
 - A) type().
 - B) int().
 - C) str().
 - D) print().

Actividades de Repaso ∈ Ejercicio: Preguntas de Alternativas - Programación en Python: Operaciones y Funciones

1. ¿Cuál es el operador de potenciación en Python?
 - A) *
 - B) **
 - C) ^
 - D) //
2. ¿Qué función se utiliza para convertir un número flotante a entero?
 - A) int().
 - B) float().
 - C) str().
 - D) bool().

Actividades de Repaso ∈ Ejercicio: Preguntas de Alternativas - Funciones: Conceptos Básicos

1. ¿Cuál es la principal finalidad de utilizar funciones en Python?
 - A) Reutilizar y organizar el código.
 - B) Incrementar la complejidad del programa.
 - C) Reducir la legibilidad del código.
 - D) Evitar el uso de módulos externos.
2. ¿Cómo se define una función en Python?
 - A) Con la palabra reservada **function**.
 - B) Con la palabra reservada **def**.
 - C) Con la palabra reservada **func**.
 - D) Únicamente mediante expresiones lambda.

Actividades de Repaso ∈ Ejercicio: Preguntas de Alternativas - Funciones: Parámetros y Alcance

1. ¿Qué se entiende por parámetro en una función?
 - A) Una variable de salida de la función.
 - B) Un dato de entrada que la función utiliza para operar.
 - C) Una estructura de control interna.
 - D) Un error que se debe manejar.
2. ¿Qué describe el alcance local de una variable?
 - A) Una variable definida dentro de una función y accesible solo en ella.
 - B) Una variable global accesible en todo el programa.
 - C) Una variable definida en otro módulo.
 - D) Una variable que se hereda de una clase.

Juego: Adivina el Número

- El programa genera un número aleatorio entre 1 y 100.
- El usuario tiene 7 intentos para adivinarlo.
- En cada intento, se informa si el número ingresado es mayor o menor que el objetivo.
- Al finalizar, se muestra un mensaje de felicitaciones o se revela el número correcto.

Objetivo: Reforzar estructuras de control (bucles y condicionales) y el uso de la librería `random`.

Actividades de Repaso ∈ Actividad: Suma de una Serie Aritmética (Con Bucle)

Ejercicio: Suma de los primeros N términos (Iterativo)

- Solicitar al usuario el primer término (a), la razón (d) y el número de términos (n).
- Utilizar un bucle **for** para calcular cada término de la serie, donde el i -ésimo término es $a + id$ para $i = 0, 1, \dots, n-1$.
- Sumar cada término calculado y almacenar el resultado en una variable.
- Imprimir la suma total con un mensaje descriptivo.

Objetivo: Reforzar el uso de bucles y operaciones aritméticas en Python.

Ejercicio: Cálculo de la Serie Armónica

- Solicitar al usuario un número entero n .
- Utilizar un bucle **for** para calcular la suma de la serie:

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

- Acumular la suma de cada término (utilizando divisiones en punto flotante).
- Imprimir el resultado con un mensaje descriptivo.

Objetivo: Reforzar el uso de bucles y el manejo de operaciones con números reales en Python.

Ejercicio: Estadísticas Básicas

- Generar un arreglo de 15 números aleatorios utilizando `np.random.rand(15)`.
- Calcular e imprimir la media y la desviación estándar del arreglo.
- Identificar y mostrar los números mayores a la media.

Objetivo: Reforzar el uso de **numpy** y la manipulación de arreglos.

- Si tiene dudas, siempre consulte.
- Se recomienda repasar individualmente y resolver ejercicios adicionales para preparar la evaluación.
- Nunca deje de aprender!

¡A programar y a repasar se ha dicho!