

UNIVERSIDAD DR. ANDRÉS BELLO
REGIONAL CHALATENANGO
FACULTAD DE TECNOLOGÍA E INNOVACIÓN



CARRERA: INGENIERÍA EN SISTEMAS Y COMPUTACIÓN.

DOCENTE: ING. JONATHAN FRANCISCO CARBALLO CASTRO.

ASIGNATURA: FUNDAMENTOS DE PROGRAMACIÓN.

CONTENIDO: ACTIVIDAD ASINCRONA S - 16.

ALUMNOS: WENDY MARISOL ESPINOZA RIVAS.
KENIA GUADALUPE FIGUEROA ORTEGA.
ROMILIO DAVID GALDÁMEZ ESCOBAR.
AMANDA YURIDIA PORTILLO MARROQUÍN.
BRANDON JOSUÉ RIVERA ALEMÁN.

CICLO: 01-23

Chalatenango, 19 de mayo de 2023

Descripción de los conocimientos obtenidos en la cátedra de Fundamentos de Programación.

1. Algoritmos:

Comprendimos los conceptos fundamentales de un algoritmo, incluyendo su estructura, su secuencia lógica y pasos necesarios para resolver un problema. Practicamos en el diseño de algoritmos para solucionar diversos problemas, comprendimos de la importancia de desarrollar algoritmos eficientes y claros, así como la capacidad de descomponer problemas complejos en pasos pequeños y manejables.

2. Diagramas de flujo:

Aprendimos a como representar visualmente la secuencia de pasos de un algoritmo utilizando símbolos y flechas. Practicamos la creación de análisis de diagramas de flujo para diferentes escenarios y problemas. Adquirimos la habilidad para visualizar y comunicar de manera clara y estructurada la lógica de un algoritmo, facilitando su comprensión.

3. Definición e implementación de estructura de control:

Comprendimos las estructuras de control como if-else, bucles entre otros, y su importancia para controlar el flujo de ejecución de un programa. Practicamos la implementación de estructuras de control en programas, utilizando condiciones y bucles para controlar el comportamiento del programa. Adquirimos la habilidad para diseñar programas con una lógica de ejecución un poco compleja, permitiendo la toma de decisiones y la repetición de acciones según fuera necesario.

4. Características de un lenguaje de programación (Python):

Nos familiarizamos con Python como lenguaje de programación, incluyendo sus características como legibilidad, expresividad y facilidad de uso. Tuvimos la experiencia en la escritura de programas en Python, utilizando su sintaxis y características específicas. Obtuvimos conocimiento de las ventajas y

características distintivas de Python, también aprendimos del aprovechamiento de sus características para escribir programas eficientes y legibles.

5. Conociendo las variables:

Comprendimos las variables como contenedores de datos en Python, su declaración y asignación de valores. Practicamos en la creación y manipulación de variables en Python, incluyendo operaciones matemáticas y concatenación de cadenas. Adquirimos la capacidad para poder almacenar y manipular datos en Python y comprendimos la importancia de elegir nombre para variables que sean descriptivos y significativos.

6. Introducción a las estructuras de control:

Aprendimos de las estructuras de control básicas, como if, else y elif, y su uso para controlar el flujo de ejecución de un programa en Python. Practicamos en la implementación de estructuras de control condicionales en Python para tomar decisiones basadas en condiciones específicas. Adquirimos la habilidad para desarrollar programas que ejecuten diferentes instrucciones según las condiciones establecidas, permitiendo una mayor flexibilidad y adaptabilidad en el comportamiento del programa.

7. Operadores de comparación:

Conocimos de los operadores de comparación en Python, como igualdad (==), desigualdad (!=), mayor que (>), menor que (<), mayor o igual que (>=) y menor o igual que (<=). Practicamos con la utilización de operadores de comparación para evaluar condiciones y tomar decisiones en estructuras de control. Adquirimos la capacidad para realizar comparaciones entre valores y evaluar condiciones en programas Python, lo que permite tomar decisiones lógicas y controlar el flujo del programa.

8. Operadores lógicos:

Conocimos de los operadores lógicos en Python, como and, or, not, y su uso para combinar y evaluar múltiples condiciones. Pusimos en práctica la utilización de

operadores lógicos para construir expresiones condicionales más complejas y tomar decisiones basadas en múltiples condiciones. Adquirimos la habilidad para realizar evaluaciones lógicas y combinar condiciones en programas Python, permitiendo una lógica más avanzada en las estructuras de control.

9. Estructura de control if-else:

Comprendimos de la estructura de control if-else en Python y su uso para ejecutar diferentes bloques de código según una condición. Practicamos en la implementación de la estructura if-else en programas Python para tomar decisiones y controlar el flujo de ejecución. Adquirimos capacidad para escribir programas que ejecuten diferentes instrucciones según una condición según una condición específica, lo que permite una mayor flexibilidad y adaptabilidad en el comportamiento del programa.

10. Conociendo los bucles:

Nos familiarizamos con los bucles en Python, como el bucle While y el bucle For, y su uso para repetir un bloque de código varias veces. Practicamos en la implementación de bucles en programas Python para ejecutar iterativamente un conjunto de instrucciones. Adquirimos la habilidad para crear programas que realicen tareas repetitivas de manera eficiente y controlada, lo que facilita la automatización de procesos y el procesamiento de grandes cantidades de datos.

11. Introducción a los bucles anidados:

Comprendimos de los bucles anidados en Python, como el bucle for anidado y el bucle while anidado y su uso para ejecutar bucles dentro de otros bucles. Pusimos en práctica la implementación de bucles anidados en programas Python para realizar iteraciones complejas y estructuradas. Adquirimos habilidad para resolver problemas que requieren múltiples niveles de iteración y realizar tareas más complejas utilizando bucles anidados de manera eficiente.

12. Entornos virtuales:

Aprendimos de los entornos virtuales en Python y su importancia para gestionar y aislar las dependencias de los proyectos. También adquirimos conocimientos a cómo hacer uso de los entornos virtuales para mantener proyectos de Python independientes y evitar conflictos de dependencias, lo que facilita la gestión y el desarrollo de aplicaciones.

13. Introducción a los Arrays:

Nos familiarizamos con los Arrays en Python, también conocidos como listas y su uso para almacenar y manipular conjuntos de datos ordenados. Pusimos en práctica la creación, indexación y manipulación de Arrays en Python, incluyendo la adición, eliminación y modificación de elementos. Adquirimos la habilidad para utilizar Arrays, para organizar y procesar datos de manera eficiente.

14. Métodos de listas:

Adquirimos conocimientos de los métodos de listas incorporados en Python y su uso para realizar operaciones comunes en listas. Conocimos diferentes métodos como `count()`, `append()`, `remove()`, `join()`, `split()`, `replace()`, que sirven para agregar, eliminar etc., elementos en una lista. Comprendimos como aprovechar los métodos de listas para manipular y transformar eficientemente los elementos dentro de una lista en Python.