### Programacion para No Programadores

**Área:** Ingeniería de Software

**Tipo:** Curso de Ampliación

**Unidades:** 5 Unidades Crédito

**Horas académicas:** 70

**Código:** --------

**Fundamentación:**

El conocimiento de la algorítmica y la programación es una base fundamental en el estudio de la computación. En este curso se estudian los conceptos fundamentales de la programación así como técnicas para el diseño de algoritmos y su implementación en lenguajes de programación. Se estudian los conceptos fundamentales de algorítmica, tipos de datos, estructuras de control, clases y métodos. El curso es teórico-práctico: junto al conocimiento teórico impartido el estudiante debe demostrar habilidades para la construcción de programas, utilizando un lenguaje de programación real orientados a objetos

**Objetivos:**

Al finalizar el curso el participante debe ser capaz de:

* Plantear soluciones algorítmicas a distintos problemas.
* Construir programas, utilizando correctamente las primitivas de datos y de control de un lenguaje de programación, así como las unidades de modularización.
* Comprender los aspectos de la programación orientada a objetos y definir clases para resolver problemas sencillos, generando programas escritos en un lenguaje orientado a objetos.

**Perfil del estudiante:**

El curso está dirigido a profesionales con conocimientos básicos o ninguno en programación, así como estudiantes de postgrado en las áreas de ciencias de la computación.

**Contenido Temático:**

1. **Introducción a la Programación:** Organización del Computador. Conceptos de algoritmo, dato, información, lenguaje natural, lenguaje pseudo-formal, lenguaje de programación, programa. Principio de abstracción. Refinamiento progresivo y estrategia divide y vencerás.
2. **Tipos de Datos Elementales**: Variables, Constantes. Concepto de tipo de dato. Clasificación de los Tipos de Datos. Tipos elementales y sus operaciones: entero, caracter, booleano, real, sub-rango, enumerado, cadena de caracteres (*string*). Precedencia de operadores. Conversión de tipos implícita y explícita.
3. **Acciones Elementales**: Declaraciones de variables, constantes y tipos. Instrucción de Asignación. Valor izquierdo y derecho de una variable. Acciones predefinidas. Operación de Lectura. Operación de Escritura.
4. **Estructuras de Control**: Secuenciación. Bloques. Selección simple, compuesta, anidada y múltiple. Estructuras de control iterativas: Repetir, Mientras, Para.
5. **Introducción al Enfoque Orientado a Objetos:** Objetos y Clases. Atributos de clase y de objetos. Mensajes. Encapsulamiento. Descomposición basada en objetos. Relaciones entre clases y objetos
6. **Clases y Métodos**: Declaración de Clases. Atributos. Métodos. Control de acceso. Utilización de métodos. Pase de Parámetros: valor y referencia. Métodos Predefinidos. Métodos Constructores y Destructores. Definición de Objetos. Relaciones entre clases: dependencia, agregación, herencia. Jerarquía de Clases. Redefiniciones de métodos y polimorfismo. Los métodos en los lenguajes procedimentales: Acciones y Funciones, ambientes de referenciación: datos locales y no Locales.
7. **Tipos de Datos Estructurados**: Arreglos unidimensionales y multidimensionales: Declaración y Operaciones. Algoritmos de búsqueda secuencial y binaria. Algoritmos de ordenamiento por selección y por intercambio. Registros. Archivos Secuenciales: Declaración, Operaciones, Archivos de Texto. Algoritmo de mezcla de archivos.

**Evaluación**

* 3 quices 20% c/u
* 1 proyecto 40%

**Observaciones finales**:

* El curso tiene una duración de 70hrs.
* La modalidad es bimodal
* Se utilizaran herramientas de comunicación como Google Meet y el Laboratorio del Postgrado en Ciencias de la Computación.

|  |  |  |
| --- | --- | --- |
| **Docente** | **Coordinador** | **Personal Administrativo** |
| Dr. Andres Sanoja | Dr. Andres Sanoja | Rosiris Morales |

**Bibliografia**

* Paul Barry. 2016. Head First Python (2nd. ed.).
* Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser. 2013. Data Structures and Algorithms in Python (1st. ed.). Wiley Publishing.
* Allen B. Downey. Think Python: How to Think Like a Computer Scientist. Meggie from Hackbright Academy. 2016.
* Coto, Ernesto. *Lenguaje pseudoformal para la construccion de algoritmos*. Escuela de Computación. 2003.
* Fowler, Martin. *UML Gota a Gota*. Addison Wesley Longman. 1999.
* Joyanes, Luis. *Fundamentos de Programación: Algoritmos y Estructuras de Datos*. 2da. Edición. McGraw-Hill, 1996.
* Joyanes, Luis. *Programación Orientada a Objetos*. 2da. Edición. Osborne McGraw-Hill. Madrid, 2000.