

MODELO DE SYLLABUS

MACROPROCESO DOCENCIA

CIENCIAS BÁSICAS Y TECNOLOGÍAS  
PROGRAMA DE FÍSICA

*Syllabus*

**Programación**

**1. Descripción**

La asignatura Programación de la Facultad de Ciencias Básicas y Tecnologías se describe como un espacio académico que hace parte de las Actividades Académicas Básicas del Programa de Física. En esta materia se estudian los fundamentos de la programación secuencial y se busca principalmente desarrollar en el estudiante un amplio conjunto de herramientas conceptuales inherentes a las ciencias de la computación y en particular en Física que le permitan desarrollar, implementar y proponer soluciones a problemas de carácter científico. Adicionalmente, la asignatura de programación es considerada como el punto de partida hacia asignaturas avanzadas de los semestres superiores tales como: Métodos Numéricos y Física computacional entre otras.

**2. Justificación**

Hoy en día el uso de los computadores y lenguajes de programación son considerados por la comunidad científica como una herramienta indispensable y fundamental en la formación de todo profesional en Física. Puede decirse sin pérdida de generalidad que los lenguajes de programación son una herramienta bastante versátil la cual permite realizar una gran cantidad de tareas de manera eficiente, como por ejemplo: el procesamiento de datos, cálculos numéricos, visualización gráfica y cálculo simbólico, entre muchas otras. Adicionalmente, gracias al uso de lenguajes de programación es posible hoy en día simular -modelar- fenómenos físicos bajo condiciones de interés, predecir posibles comportamientos de la naturaleza aún cuando los experimentos no se pueden realizar debido a limitaciones de equipos o por las condiciones mismas del experimento.

**3. Competencias propias del espacio académico, núcleo o cátedra.**

**Competencias conceptuales:**

MODELO DE SYLLABUS

MACROPROCESO DOCENCIA

- Identifique los conceptos fundamentales programación y su uso en las diferentes áreas de la Física.
- Desarrolle el pensamiento lógico-matemático e identifique situaciones donde puedan resolverse problemas usando el cálculo científico.
- Interprete las estructuras básicas de un lenguaje de programación, además de diseñar algoritmos.
- Comprenda las estructuras lógicas de los lenguajes de programación.
- Implemente algoritmos básicos en un lenguaje de programación de manera eficiente a partir de un pseudo código.

**Competencias procedimentales:**

- Implemente algoritmos para dar solución a problemas básicos de la Física.
- Resuelva problemas de cálculo científico mediante algoritmos secuenciales o recursivos.
- Manipule correctamente archivos de entrada o salida y esté en capacidad de representarlos gráficamente.
- Manipule y desarrolle rutinas y librerías de uso general en los programas de cómputo científico.

**Competencias genéricas:**

- Desarrolla la comprensión lectora y la lectura crítica.
- Lee conceptos y ejercicios de la disciplina en una lengua extranjera (inglés).
- Comunica las ideas y conceptos con el lenguaje propio de la programación.

**4. Administración del espacio académico**

**Espacio académico:** Programación

**Horas semanales:** 6 (seis)

**Total de horas por semestre:** 192 (noventa y seis)

**Metodología:** Seminario presencial

| Generalidades               | Detalle   |
|-----------------------------|-----------|
| Código                      | 120710206 |
| Tipo de Actividad Académica | Básica    |

MODELO DE SYLLABUS

MACROPROCESO DOCENCIA

|                                |                                                                                                                                                                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ubicación                      | Quinto semestre                                                                                                                                                                                                                          |
| Naturaleza                     | Teórico-Práctica                                                                                                                                                                                                                         |
| Contenidos                     | Unidad 1. Introducción y Principios Básicos I<br>Unidad 2. Principios Básicos II<br>Unidad 3. Arreglos y Estructuras<br>Unidad 4. Programacion Orientada a Objetos<br>Unidad 5. Herencia y Polimorfismo<br>Unidad 6. Librerías de Clases |
| Créditos                       | 3 (Tres)                                                                                                                                                                                                                                 |
| Evaluación                     | Cuantitativa y cualitativa                                                                                                                                                                                                               |
| Horas de docencia directa      | 4 (Cuatro)                                                                                                                                                                                                                               |
| Horas teórico-prácticas        | 2 (Dos)                                                                                                                                                                                                                                  |
| Horas de trabajo independiente | 9 (Nueve)                                                                                                                                                                                                                                |
| Horas de asesoría              | 6 (Seis)                                                                                                                                                                                                                                 |
| Habitable                      | Sí                                                                                                                                                                                                                                       |
| Validable                      | Sí                                                                                                                                                                                                                                       |
| Homologable                    | Sí                                                                                                                                                                                                                                       |
| Requisitos                     | Ecuaciones Diferenciales                                                                                                                                                                                                                 |

## 5. Procesos integrativos:

Los estudiantes tienen la opción de pertenecer a los semilleros de la Facultad y/o vincularse a los grupos de investigación del Programa mediante procesos de iniciación científica.

## 6. Contenidos

### UNIDAD 1 - INTRODUCCIÓN Y PRINCIPIOS BÁSICOS I. – 3 Semanas

- Breve discusión sobre aspectos de los lenguajes de programación: compilados vs. Interpretados, programación secuencial, programación en paralelo, programación basada en CPUs vs. GPUs, ANSI C.
- Diseño de algoritmos y pseudo código, tipos de datos, especificadores de formato, operadores y expresiones, entrada y salida de datos por teclado.
- Operaciones aritméticas, operaciones con enteros, punto flotante, doble precisión, operadores de incremento y decremento, manejo de cadena de caracteres, conversiones de tipo.
- Variables locales, globales, programa principal y parámetros, problemas resueltos y propuestos.

MODELO DE SYLLABUS

MACROPROCESO DOCENCIA

UNIDAD 2 - PRINCIPIOS BÁSICOS II – 3 Semanas

- Uso de operadores lógicos y relacionales, uso de operadores comparacionales y operador condicional, construcción de ciclos: FOR, WHILE, DO, DO-WHILE, BREAK, CONTINUE, SWITCH, ENUM.
- Contrucción y manejo de funciones, constantes simbólicas, introducción a los punteros.
- Manejo de rutinas sin argumentos de entrada, rutinas con paso de parámetros por valor, rutinas con paso de parámetros por dirección.
- Uso de librerías externas y adaptación al programa principal, problemas resueltos y propuestos.

UNIDAD 3 - ARREGLOS Y ESTRUCTURAS – 2 Semanas

- Arreglos unidimensionales, bidimensionales y multidimensionales, inicialización de arreglos, paso de arreglos a rutinas.
- Manejo de archivos: lectura y escritura de archivos en disco duro, estructuras y arreglos de estructuras.
- Funciones de asignación dinámica malloc(), free(), problemas resueltos y propuestos.

UNIDAD 4 - PROGRAMACION ORIENTADA A OBJETOS (POO) – 3 Semanas

- Entrada y salida en C++, tipos de datos: elementales, enumerados, derivados, compuestos, constantes y variables.
- Paso de parámetros a rutinas, estructuras de selección, estructuras de repetición, estructuras de salto.
- Clases, objetos y métodos, clases sin secciones privadas (STRUCT), clases con secciones privadas, entrada y salida de datos, operadores NEW, DELETE, inicializadores.
- Constructores por defecto y construcción con parámetros con valor por defecto, destructores, operadores de asignación, clase cadena para el manejo de cadena de caracteres, variables miembro STATIC y funciones miembro STATIC.
- Recursividad, función MAIN(), problemas resueltos y propuestos.

UNIDAD 5 - HERENCIA Y POLIMORFISMO – 3 Semanas

- Definición de herencia, variables y funciones miembro PROTECTED, constructores de las clases derivadas.

**MODELO DE SYLLABUS**

**MACROPROCESO DOCENCIA**

- Herencia simple y múltiple, sobrecarga de operadores, funciones operador, asignación e inicialización.
- Clase base virtuales, conversión entre objetos de clases base y derivadas.
- Concepto de polimorfismo
- Funciones virtuales, funciones virtuales puras, clases abstractas, destructores virtuales, problemas resueltos y propuestos.

**UNIDAD 6 - LIBRERIAS DE CLASES – 2 Semanas**

- Diseño de librerías.
- Clases contenedor, clases para aplicaciones, clases de interface.
- Manejo de memoria, problemas propuestos y resueltos.
- Opcional: uso de las librerías científicas GSL del proyecto GNU, uso de rutinas científicas del Numerical Recipes C y Numerical Recipes C++, problemas resueltos básicos y propuestos.

## **7. Metodología**

El espacio académico de la asignatura de programación se desarrollará mediante exposiciones del profesor acompañadas de explicaciones de código fuente que sirva para orientar discusiones grupales en clase, desarrollo de ejercicios de aplicación sencilla y trabajo por parte del estudiante relacionadas con problemas propuestos en clase. En principio, los temas de la asignatura están enfocados para orientar el aprendizaje del lenguaje de programación en C y C++, sin embargo puede ser también orientada con otros lenguajes de programación como Python, Matlab, Mathematica, Fortran 90, Java, Octave, SAGE según el interés particular del docente y los estudiantes.

## **8. Evaluación**

Se realizarán cuatro exámenes escritos, cada uno de ellos con igual valor porcentual (20% cada uno), de acuerdo con las normas establecidas en el Estatuto Estudiantil de la Universidad del Quindío. Además, el profesor podrá usar otros elementos de evaluación como elaboración de programas de cómputo y simulaciones de fenómenos físicos, que serán acordados y concertados con los estudiantes. En el caso de los programas y las simulaciones, se deben sustentar en horas preestablecidas o en las horas de asesoría y no deben afectar el desarrollo de las clases.

**MODELO DE SYLLABUS**

**MACROPROCESO DOCENCIA**

Para las evaluaciones escritas, se espera una retroalimentación que se hará en el salón de clases o mediante la publicación de la solución de los exámenes. Para los trabajos de programación y las simulaciones se realizará una corrección personalizada de los mismos.

## **9. Bibliografía**

### Texto guía:

- García, J. Y colaboradores. (1998). Aprenda C++ como si estuviera en primero. Universidad de Navarra.
- Oliag, S. T. (1995). Curso de programación: Apuntes de Clase.
- Paez, M.J. (2004). C y C++ de Afán. Universidad de Antioquia.

### Textos de lectura complementaria:

- Abell, M.L. & Braselton. J. P.(2009). Mathematica by Example.Deitel (1994). Como Programar en C/C++. Prentice Hall.
- Ataway, S. (2009). Matlab: A Practical Introduction to Programming and Solving Problems.
- Chapman, S. J. (1999). Matlab Programmimg for engineers.
- Ford, W. & Topp, W. (1999). Introduction to Computing Using C++ and Object Tecnology, Prentice Hall.
- Galassi, M. (2009). GNU Scientific Library Reference Manual.
- Guttag, J. (2016). Introduction to Computation and Programming Using Python: with Application to Understanding Data.
- Lambert , K.A. (2011). The Fundamentals of Python: First Programs, Cengage Learning.
- Teukolsky, S. (1986). Numerical Recipies in C: The Art of Scientific Computing.
- Wellin, P. R. (2013). An introduction to programming with Mathematica.

## **10. Historial de revisión**

Se genera el Syllabus de la asignatura Cálculo Integral para el segundo semestre del año 2018.

Revisión: Agosto 6 de 2018 – Acta 006 – Consejo Curricular de Física

## **11. Vigencia del syllabus:**

**MODELO DE SYLLABUS**

**MACROPROCESO DOCENCIA**

La vigencia del Syllabus es a partir del segundo semestre de 2018.

Responsables:

Este Syllabus fue elaborado por:

Prof. Paulo César Carmona Tabares