# Programación Científica

Temario para enseñar conceptos de programación en la materia Métodos Numéricos, con el objetivos de que los alumnos aprendan a plantear, diseñar y desarrollar un algoritmo utilizando buenas practicas independientemente del lenguaje de programación que utilizan.

Durante el desarrollo de la materia se darán clases especificas a las técnicas de desarrollo de software para reforzar y facilitar el desarrollo de algoritmos de métodos numéricos. En el transcurso de la misma se sugiere que cada alumno tenga un proyecto en un repositorio Git o un Branch en un repositorio común y mantener en el mismo sus ejercicios y las practicas a fin de que los alumnos se acostumbren a trabajar con un repositorio de software (Git).

* **Introducción**
  + Historia de los lenguajes de programación
  + **Programación Estructurada, Programación funcional, Programación Orientada a Objetos.**
  + Donde se utiliza el software
  + Estado del arte en algoritmos de métodos numéricos.
* **Ingeniería de Software**
  + **Definición**
  + De donde surge
  + El rol del Ingeniero de Software
* Herramientas
  + **IDEs, Git, Herramientas de Debug, etc**.
* **Programación (algoritmos y pseudocodigo)**
  + **Tipos de Datos**
  + **Variables**
  + **Estructuras básicas de control (bucles, etc)**
  + **Funciones**
  + **Entrada y Salida de datos**
* **Calidad de Software**
  + **Características de calidad de software (robustez, portabilidad, eficiencia, facilidad de uso, etc).**
  + **Convention y Notación en programación**
  + Aseguramiento de la Calidad de Software
* **Procesos de Desarrollo de Software**
  + Metodologías Tradicionales (modelo espiral, proceso unificado, etc)
  + Metodologías Agiles (XP, SCRUM)
  + Captura de Requerimientos
  + **Análisis, Diseño e Implementación**
    - **Diagrama de Clases y de Flujo**
* **Patrones de Diseño**
  + Tipos de Patrones
  + **Ejemplos de algunos patrones mostrando implementación en algunos lenguajes**
* **Validación y Verificación**
  + **Testing**
  + **Simulación**
  + **Analysis de Performance**
  + **Técnicas de Optimizacion**
  + **Métricas (medir tiempo de ejecución, etc).**
* Librerias:
  + SciPy <http://www.scipy.org/> - https://www.enthought.com/products/epd/
  + IMSL numerical library
* Se recomienda dar ejemplos en pseudocodigo y algunos en los siguientes lenguajes de programación: C/C++, Fortran, Matlab, Java, Python, Erlang (Ing. en Telecomunicaciones).