

Universidad de Costa Rica
Escuela de Física
Tópicos de Física I (Aplicaciones de Informática en Física)
FS0518
I Semestre, 2013

Créditos 3
Horas por semana: 3
Horario: Lunes de 4pm a 7pm
Aula: Laboratorio 309

Profesor: **Jeudy Blanco Vega**

Descripción y Objetivos

El objetivo principal del curso es brindarle al estudiante los conocimientos y herramientas fundamentales que le permitan aprovechar las tecnologías de computación para aplicarlas en su futura vida profesional y académica.

Para esto, se verán principios y técnicas de Ingeniería de Software aplicables al ámbito científico: fundamentos de programación, herramientas para el control de versiones, técnicas y herramientas de depuración y “profiling”, utilización e implementación de librerías científicas.

Durante el curso, se hará uso herramientas open source disponibles y de amplia utilización en el mundo de la computación científica, como lo es el sistema operativo Linux, el lenguaje de programación Python, entre otros. Se escoge Python como lenguaje de programación por su legibilidad, portabilidad, amplia difusión y creciente uso en aplicaciones científicas, al punto de ofrecer alternativas libres a conocidos paquetes propietarios.

Al finalizar el curso, el estudiante será capaz de:

- Empezar a implementar soluciones computacionales para diferentes problemas en sus proyectos de investigación.
- Evaluar soluciones y paquetes existentes.
- Profundizar en áreas más avanzadas de la computación científica.

Metodología

Las clases se impartirán en el laboratorio. Se asume que el estudiante sabe utilizar una computadora al menos de forma básica, pero no se requieren conocimientos previos de programación. El curso iniciará repasando conceptos básicos del sistema operativo Linux, como lo es el uso del shell y comandos básicos. Luego se verá el sistema de control de versiones que se usará a lo largo del curso (git), tanto para ejemplos como para tareas y proyectos. Luego se verá una introducción a conceptos fundamentales de Ingeniería del

Software, para posteriormente entrar de lleno a la programación con Python. Una vez se tengan los conocimientos y habilidades de programación fundamentales, se presentará un problema científico real que será desarrollado gradualmente en clase aplicando todo lo aprendido.

Durante las primeras 2 horas de cada clase se darán lecciones magistrales de los diferentes temas del curso, y durante la tercera hora, cuando así sea conveniente, se desarrollarán ejercicios prácticos relacionados con los temas vistos en clase.

Evaluaciones

Dada la naturaleza del curso, las evaluaciones se realizarán **preferiblemente** en la computadora, dentro del laboratorio. Estas consistirán en quices (teóricos y prácticos), tareas cortas, un proyecto programado y una exposición, en donde se dará al estudiante la posibilidad de escoger un tema dentro del área de la Física que sea de su interés (Meteorología, Astrofísica, etc) o bien un tema específico sugerido por el profesor, e investigar aplicaciones reales de la computación científica relacionados con el tema escogido.

Las tareas cortas se trabajarán de forma individual, y el proyecto programado podrá trabajarse en grupos de máximo 2 personas. La entrega de estos se hará por medio de repositorios de código privados en Bitbucket, respetando que la última actualización sea hecha antes de la fecha límite de entrega. El proyecto programado se revisará junto al profesor en un horario a convenir, los estudiantes deberán explicar la solución implementada y en caso de haberse trabajado en grupo, el aporte de cada integrante.

- **Quiz 1:** 5% (semana 2)
- **Quiz 2:** 5% (semana 4)
- **Tarea corta 1:** 15% (semana 6)
- **Quiz 3:** 5% (semana 9)
- **Proyecto:** 35% (semana 13)
- **Tarea corta 2:** 15% (semana 15)
- **Exposición:** 20% (semana 16)

Cronograma

Semanas 1 - 2:

- Linux (shell, comandos importantes).
- Git como herramienta de control de versiones (servicios en la nube: Github y Bitbucket)

Semanas 3 - 4:

- Conceptos de Ingeniería de Software
 - Ciclo de vida del software.
 - Estructuras de datos.
 - Pruebas y aseguramiento de calidad.

- Bases de datos.
- Conceptos generales de programación.

Semanas 5 - 7:

- Introducción a la programación con python
 - ipython
 - Variables y estructuras de datos
 - Control de flujo
 - Funciones y módulos
 - Archivos
 - Manejo de errores
 - Buenas prácticas

Semanas 8 - 9:

- Programación orientada a objetos con python

Semanas 10 - 12:

- Aplicación a caso real: Problema de N cuerpos.
 - Implementación en python por métodos de Euler, Euler modificado y Leapfrog.
 - Algoritmos optimizados Vs Fuerza Bruta

Semana 13:

- Depuración y “profiling” con python
- Pruebas al software con python

Semanas 14 - 15:

- Librerías científicas en python (Scipy, Numpy, Matplotlib, PyFITS)

Semana 16:

- Exposiciones

Bibliografía y referencias

- Python para todos, Raúl Gonzalez Duque (libro)
- Documentación oficial de Python (<http://www.python.org/doc/>)
- Documentación oficial de Scipy y Numpy (<http://docs.scipy.org/doc/>)
- Pro Git (<http://git-scm.com/book>)
- The Linux Command Line (<http://linuxcommand.org/tlcl.php>)
- Learning the Shell (http://linuxcommand.org/learning_the_shell.php)
- The 5-Minute Essential Shell Tutorial (<http://community.linuxmint.com/tutorial/view/100>)
- The Art of Computational Science (<http://www.artcompsci.org/>)
- Basic concepts of software engineering and maintenance (G. Giuliani -ICTP)
- Introduction to testing scientific software (S. Cozzini - Democritos and SISSA/eLAB)