UNIVERSIDAD DE COSTA RICA
ESCUELA DE FÍSICA
Curso: FS 0518
Tópicos de Física I
Aplicaciones de la Informática en Física

I - Características generales:

Naturaleza: Teórico-práctico, clases magistrales combinado con prácticas en el laboratorio.
Prof. MSc. Jeudy Blanco Vega
Contacto: jeudy.blanco@cinespa.ucr.ac.cr, jeudyx@gmail.com
Duración: Un semestre
Créditos: 03
Horas por semana: 3
Horario: Jueves de 4pm a 7pm
Aula: Laboratorio 309
Consulta: Jueves de 3pm a 4pm y de 7pm a 8pm

I I- Descripción del curso:

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 de software existentes.
- Profundizar en áreas más avanzadas de la computación científica.

III - Contenido:

1) Introducción al shell de Linux, comandos importantes (semana 1)

2) Control de versiones (semanas 1 y 2)
- Git
- Servicios en la nube (github, bitbucket)

3) Conceptos de Ingeniería de Software y programación (semana 3)
- Ciclo de vida del software.
- Estructuras de datos.
- Bases de datos.
- Redes: Protocolo HTTP (interfaces REST)

4) Programación con python (semanas 4 a 7)
- Shell de python: ipython
- Instalación de paquetes: pip
- Ambientes virtuales: virtualenv
- Ambientes integrados de desarrollo: Pycharm
- Variables y estructuras de datos
- Control de flujo
- Funciones y módulos
- Archivos
- Conexión a redes (librería **requests**)
- Manejo de errores
- Buenas prácticas

5) Librerías científicas en python (semanas 8 a 11)
- Scipy
- Numpy
- MatPlotLib
- Astropy (para procesamiento de archivos FIT)
- Scikit y AstroML (Machine Learning con Python)

6) Aplicación a problemas científicos reales (semanas 12 a 15)
- Simulaciones de N Cuerpos: Implementación en python por métodos de Euler, Euler modificado y Leapfrog.
- Análisis de datos de telescopio Kepler con Astropy

7) Introducción a la Computación de alto desempeño (semana 16)
- Clusters
- MPI
- HPC en python: Mpi4py

IV - Metodología:

Las clases se impartirán en el laboratorio. Se asume que el estudiante sabe utilizar una computadora al menos de forma básica y **ha llevado el curso de principios de informática, por lo que requieren conocimiento previos de programación**. El curso iniciará repasando conceptos básicos del sistema operativo Linux, como lo es el uso del shell y comandos básicos.

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 presentarán problemas científicos reales que serán 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.

V - Evaluación:

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 orientado a resolver un problema científico utilizando lo aprendido a lo largo del curso.

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**: 10% (24 de marzo)
- **Quiz 2**: 10% (31 de marzo)
- **Tarea corta 1**: 10% (21 de abril)
- **Tarea corta 2**: 15% (19 de mayo)
- **Tarea corta 3**: 15% (9 de junio)
- **Proyecto Final**: 40% (4 de julio)

VI - Bibliografía:

- Python para todos, Raúl Gonzalez Duque
- Documentación oficial de Python: http://www.python.org/doc/
- Documentación oficial de Scipy y Numpy: http://docs.scipy.org/doc/
- Documentación oficial de Scikit: http://scikit-learn.org/
- Pro Git: http://gitscm.com/book
- The Linux Command Line: http://linuxcommand.org/tlcl.php
- Learning the Shell: http://linuxcommand.org/learning_the_shell.php
- The 5Minute 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)