



Posgrado
Facultad de Ingeniería

Carrera: Doctorado en Ingeniería. Acreditado a las tres menciones

Curso de Posgrado: Programación científica y aplicaciones con Python

Carga Horaria: 90 hs

Docente/s a cargo: Dr. Leandro Bugnon, Dr. Sebastián Vanrell, Dr. Matias Gerard

Semestre: 2 de 2019

Características del curso

1. **Carga horaria:** la cantidad de horas reloj: 90 hs
2. **Curso teórico:** curso donde se desarrolla en forma expositiva una temática propia de la disciplina:
3. **Curso teórico-práctico:** curso que articula la modalidad del curso teórico con una actividad de la práctica con relación a la temática de estudio. Lo teórico y lo práctico se dan simultáneamente en forma interrelacionada: x
4. **Carácter:** si son del ciclo común o del ciclo electivo: Electivo

Programa Analítico de foja: a foja:

Bibliografía de foja: a foja:

Aprobado Resoluciones de Consejos Directivos:

Fecha:

Modificado/Anulado/ Res. Cs. Ds.:

Fecha:

Carece de validez sin la certificación del Director/a del Doctorado:



Facultad de Ingeniería
Oro Verde, E. R.
República Argentina

PROGRAMA ANALÍTICO

Módulo básico:

- a. Fundamentos: Introducción al lenguaje Python. Tipos y estructuras de datos. Bloques de código y estructuras de control. Funciones y clases. Depuración de errores. Manejo de archivos. Módulos, paquetes e importación de librerías. Manejo de excepciones. Creación, edición y ejecución de scripts. Documentación y sitios de consulta.
- b. Computación numérica (NumPy): Introducción a NumPy. Estructuras de datos: matrices, arreglos n -dimensionales y propiedades. Índices y selección. Operaciones. Importación y exportación de datos.
- c. Graficación (Matplotlib): Introducción a Matplotlib. Elementos de una gráfica. Tipos de gráficas 2D y 3D. Múltiples gráficas en una figura. Personalización y edición avanzada de gráficas y figuras. Exportación en formato de imágenes.

Módulo avanzado:

- d. Manejo de datos estructurados (Pandas): Introducción a Pandas. Estructuras de datos: Index, Series y DataFrames. Índices y selección. Operaciones con Series y DataFrames. Datos faltantes y valores duplicados. Métodos para combinación de datos estructurados: Concat, Append, Merge y Join. Operaciones globales y locales: mapeo, agregado y agrupamiento. Visualización de datos estructurados.
- e. Procesamiento de señales e imágenes y estadística general (SciPy, Scikit-Image): Introducción a SciPy. Integración numérica. Optimización. Interpolación. Convolución. Transformada de Fourier. Diseño de filtros. Introducción a scikit-image. Procesamiento básico de imágenes. Manejo de archivos. Distribuciones estadísticas. Estadística descriptiva. Tests estadísticos.
- f. Aprendizaje automático (Scikit-Learn, PyTorch). Introducción a Scikit-Learn. Bloques de procesamiento en cascada. Exploración y selección de hiper-parámetros y validación de modelos. Figuras de mérito. Introducción a PyTorch. Operaciones con tensores. Estructuras neuronales básicas. Arquitecturas de redes neuronales y modos de entrenamiento.
- g. Tópicos selectos: Creación de módulos propios y publicación en repositorios públicos o privados. Interacción con el sistema operativo y otros lenguajes de programación. Manejo independiente de proyectos y entornos virtuales. Paralelización. Procesamiento avanzado de imágenes.
- h. Aspectos de Arquitectura de Aplicación en Python: Arquitectura de Software: Principios de la Arquitectura de Software, Definición, Arquitectura vs Diseño, Características. Modelos de Referencia de

Arquitectura: Modelo de Capas y Arquitectura Limpia. Consideraciones de Diseño: Tipo de Aplicación, Estrategia de Despliegue, Tecnologías, Atributos de Calidad, Aspectos Transversales. Capa de Persistencia: Tipos de Tecnologías para persistir datos. Introducción a SQLAlchemy. SQLAlchemy Core, Lenguaje SQL y manejo de tablas. SQLAlchemy ORM, Introducción al mapeo de Objetos. Capa de Presentación: Arquitecturas de Aplicaciones Web. Framework Web Flask. Estructura básica de una aplicación. Templates. Web Forms. Base de datos. Capa de Servicios: Conceptos de Arquitectura basada en Servicios. Servicios Rest. Introducción de la APIs REST con Flask.



Facultad de Ingeniería
Oro Verde, E. R.
República Argentina

BIBLIOGRAFIA

- Fundamentos del lenguaje Python y cómputo científico
- VanderPlas, J. (2016). *Python data science handbook: essential tools for working with data*. O'Reilly Media, Inc.
- Reitz, K., & Schlusser, T. (2016). *The Hitchhiker's Guide to Python: Best Practices for Development*. O'Reilly Media, Inc.
- Varoquaux, G., Gouillart, E., & Vahtras, O. (editores) (2017). *Scipy lecture notes*.
- González Duque, R. (2014). *Python para todos*.
- Mahdavan, Samir (2015). *Mastering Python for Data Science*. Packt Publishing.

- Arquitectura de software
- Bass, L., Clements, P., & Kazman, R. (2013). *Software Architecture in Practice*. Addison-Wesley Professional.
- Microsoft patterns, & Practices. (2009). *Microsoft application architecture guide*. Microsoft Press.
- Pillai, A. B. (2017). *Software Architecture with Python*. Packt Publishing Ltd.
- Chun, W. J. (2012). *Core Python Applications Programming* (3rd ed.). Prentice Hall Press.
- Grinberg, M. (2018). *Flask web development: developing web applications with python*. O'Reilly Media, Inc..
- Copeland, R. (2008). *Essential SQLAlchemy*. O'Reilly Media, Inc.

- Documentación oficial de referencia:
- Python: <https://docs.python-guide.org/>
- NumPy: <https://docs.scipy.org/doc/numpy/contents.html>
- Matplotlib: <https://matplotlib.org/users/index.html>
- Pandas: <http://pandas.pydata.org/pandas-docs/stable/>
- SciPy: <https://docs.scipy.org/doc/scipy/reference/>

- Scikit-Image: <http://scikit-image.org/docs/stable/>
- Scikit-Learn: https://scikit-learn.org/stable/user_guide.html
- PyTorch: <https://pytorch.org/docs/stable/index.html>



Facultad de Ingeniería
Oro Verde, E. R.
República Argentina

PLANIFICACIÓN DEL CURSO

Objetivo General:

Se espera que el alumno se familiarice con el lenguaje de programación Python, utilizando módulos básicos y específicos, para poder enfrentar satisfactoriamente los desafíos que se le presenten en sus tareas de investigación y desarrollo.

Objetivos Particulares:

- Adquirir conocimientos y habilidades básicas en el lenguaje de programación Python, para el desarrollo e implementación de algoritmos.
- Conocer el ecosistema Python y sus utilidades en el manejo de datos y aplicaciones científicas.
- Adquirir y ejercitar el uso de herramientas avanzadas para desarrollar eficientemente tareas en áreas tales como minería de datos, procesamiento de imágenes o señales, y aprendizaje automático.
- Adquirir la capacidad de analizar la documentación disponible para resolver rápidamente nuevos desafíos que se le presenten en el uso diario.

Conocimientos previos requeridos (Si correspondiese).

- Conocimientos básicos de programación
- Conocimientos de álgebra vectorial
- Conocimientos recomendados para un mejor aprovechamiento del curso (no excluyente): procesamiento de señales e imágenes, aprendizaje automático o minería de datos, según el perfil y los intereses del alumno.

Fecha tentativa de inicio del dictado y duración del Curso (en semanas).

Agosto de 2019 (12 clases semanales de 5 hs. c/u, más 30 hs de actividades extra áulicas)

Cupo de alumnos (cantidades mínima y máxima).

5-30

Lugar:

Oro Verde

Día(s) y horario(s) tentativo(s) de dictado: Martes, de 9 a 12 y de 13 a 15 hs

Profesores:

Docentes responsables:

- Dr. Leandro Bugnon
- Dr. Sebastián Vanrell
- Dr. Matías Gerard

Docente colaborador

- Ing. Víctor Valotto

Colaboradores:

- Bioing. Juan Cruz Gazo Locan
- Bioing. Leandro Mayrata
- Bioing. Ramiro Gatti

Los docentes responsables estarán presentes en todas las clases teórico-prácticas. Cada tema del cronograma estará asignado a un docente, quien estará a cargo de impartir la parte teórica y guiar la parte práctica. Los demás docentes participarán durante la teoría y práctica, asistiendo al docente responsable y colaborando en la explicación y resolución de los ejercicios de las guías prácticas. La clase de la semana 11 será responsabilidad del Ing. Valotto.

Condiciones de Regularidad y Promoción:

Será responsabilidad del alumno asistir a las clases teórico-prácticas y realizar los ejercicios prácticos propuestos en cada semana.

La evaluación periódica consistirá en la entrega de 4 guías de ejercicios de aplicación, agrupadas por bloques temáticos. Cada una estará compuesta por un conjunto de ejercicios específicos de los temas dados en cada bloque temático, y otro conjunto compuesto por ejercicios de integración de los temas desarrollados en el curso hasta ese momento. Las guías se realizarán en grupos de hasta 3 personas, y se entregará un informe por grupo para ser evaluado.

La evaluación final se realizará a partir de una propuesta e implementación en Python de algoritmos o soluciones para resolver un problema de aplicación de interés para el alumno, previamente discutida y acordada con los docentes.

Infraestructura necesaria:

- Disponibilidad de una PC o notebook por alumno con conexión a internet para todas las clases.
- Cañón proyector.
- Pizarra y marcadores.

Cronograma del Curso:**Semana 1:**

Fundamentos I. Presentación de los objetivos y alcances del curso. Introducción al entorno de desarrollo que utilizaremos en el curso. Ejercicios de programación en Python. (Gerard-Bugnon-Vanrell)

Semana 2:

Fundamentos II. (Gerard-Vanrell-Bugnon)

Semana 3:

Computación numérica con Numpy. (Bugnon-Vanrell-Gerard)

Semana 4:

Graficación de datos. (Vanrell-Gerard-Bugnon)

Semana 5:

Manejo de datos estructurados con Pandas. Visualización avanzada de datos. (Bugnon-Gerard-Vanrell)

Semana 6:

Procesamiento de señales e imágenes y estadística general I. (Gerard-Vanrell-Bugnon)

Semana 7:

Procesamiento de señales e imágenes y estadística general II. (Bugnon-Gerard-Vanrell)

Semana 8:

Aprendizaje automático en Python, validación y test estadísticos. (Vanrell-Bugnon-Gerard)

Fecha de definición de alcances y objetivos de la propuesta para la evaluación final.

Semana 9:

Introducción a PyTorch. Implementación de redes neuronales. (Bugnon-Gerard-Vanrell)

Semana 10:

Tópicos selectos (Vanrell-Gerard-Bugnon)

Semana 11:

Arquitectura de software para aplicaciones en Python (Valotto)

Semana 12:

Presentación breve de las soluciones implementadas por los alumnos como evaluación final (15 min cada uno).

Fechas tentativas de dictado: Todos los martes desde el 6 de agosto al 22 de octubre, 5 hs. cada clase. Además de 30 hs. de actividades extra áulicas.

Fecha Evaluación: Semana 12

Instancia de recuperatorio: 15-30 días después de la semana 12.