

# Introducción al curso Algoritmia y Complejidad

Ernesto Rodriguez - Juan Roberto Alvaro Saravia

Universidad Francisco Marroquin

*ernestorodriguez@ufm.edu - juanalvarado@ufm.edu*

# Objetivos del curso

- Familiarizarse con varios algoritmos que se utilizan en las ciencias de la computación
- Aprender a analizar problemas con el proposito de poder aplicar algoritmos para solucionarlos
- Aprender a razonar sobre la complejidad y rendimiento de una solución algorítmica
- Familiarizarse con el analisis costo/beneficio de las diferentes opciones para resolver problemas
- Introducir al estudiante a la teoría de computabilidad y complejidad
- Familiarizar al estudiante con el concepto de intratabilidad y ayudarle a manejarla

- El curso se dividirá en dos módulos:
  - Algoritmos y complejidad
  - Teoría de la computabilidad y complejidad

- Analisis asintotico y notación
- Familias de algoritmos en ciencias de la computación:
  - Ordenamiento
  - Búsqueda
  - Optimización
  - Numericos
  - Algoritmos aleatorizados
- Aplicaciones de algoritmos:
  - Almacenamiento
  - Optimización de procesos
  - Inteligencia artificial
  - Graficos
  - Seguridad
- Estos algoritmos aparecen en entrevistas muy amenudo

# Teoria de la Computabilidad y Complejidad

- Lenguajes formales:
  - No-interesantes
  - Interesantes
  - Decidibles
  - Enumerables
  - Regulares
- Decidibilidad y el *Halting Problem*
- Modelos de computación:
  - Definición de computabilidad
  - Maquinas de Turing
  - Calculo- $\lambda$
  - Funciones recursivas
- Clases de computabilidad y herarquia computable
- Clases de complejidad:
  - Lenguajes polinomiales (P)
  - Lenguajes polinomiales no determinísticos (NP)
  - Lenguajes exponenciales
  - Por espacio y por tiempo

- Licenciatura de Jacobs University Bremen, enfocado en Machine Learning.
- Maestria de Utrecht University, enfocado en Lenguajes de Programación.
- Trabajo en varias empresas, incluyendo Microsoft.
- Dos publicaciones academicas.
- Varios proyectos open source.
- Intereses: Hackathons Cryptomonedas, Programación funcional, Machine Learning, Computación Teórica.

# Formato del curso

- Una hoja de trabajo (casi) semanal.
- Se utilizara Git y Github para entregar trabajos.
- Todo trabajo escrito se realizara con Latex.
- Dos exámenes parciales teóricos.
- Exámen final.
- No es un curso de programación, pero se utilizara la programación para entender los temas.
- Se utilizara Python 3 como lenguaje de programación

## Herramientas

- Visual Studio Code
- Git
- Github
- Latex
- Python
- Manjaro Linux

No importa a que campo de la computación se dediquen, Linux siempre estará ahí.

## Recursos

- Programa del Curso
- Repositorio del curso [3]
- Introduction to Algorithms [2]
- Latex Wiki [4]
- Git tutorial [1]
- Python tutorial



- La asistencia a clase es requisito universitario, pero somos adultos y si preferimos aprender por su cuenta, lo acepto. Sin embargo, recomiendo que asistan a clase.
- Durante clase, respetar al profesor y a sus compañeros. No interrumpir ni burlarse de otras personas.
- Ser respetuoso y constructivo a la hora de criticar.
- No se tolera el plagio.  
Es más difícil aprender a copiar que aprender a programar.
- Ser puntuales a la hora de entregar tareas.

# Referencias



Software Freedom Conservancy.

Git.

<https://git-scm.com/docs/gittutorial>.



T. Cormen, C. Leiserson, R. Rivest, and C. Stein.

*Algorithms*.

Mit Press, 3 edition, 2009.



Ernesto Rodriguez.

Algoritmos ufm 2018.

<https://github.com/netogallo/algoritmos-ufm-2018>.



Wikibooks.

Latex.

<https://en.wikibooks.org/wiki/LaTeX>.