

# Bootcamp de Algoritmos y Estructuras de Datos (Parte I)

Código Facilito



## Descripción del bootcamp

Este bootcamp introduce al estudiante a algoritmos y estructuras de datos de nivel **intermedio-avanzado**. A través de mini proyectos; que se desarrollaran en cada clase. Donde el estudiante podrá combinar distintos tipos de algoritmos y estructuras de datos, esto para primero desarrollar la intuición a la hora de escoger una estrategia algorítmica y luego saber cómo aplicarlos.

El bootcamp buscar formar estudiantes con la comprensión de qué los algoritmos no son sólo piezas de software que nos ayudan a desarrollar la lógica, sino también tecnologías que se pueden transformar en productos. Esto está en sintonía con la época en qué vivimos, donde muchas startups, sus productos son, principalmente, algoritmos.

**Líder técnico:** Camilo Chacón Sartori

**Fecha inicio:** 30 de noviembre del 2024

## Objetivos

### General:

- Demostrar con la comprensión de qué los algoritmos no son sólo piezas de software que nos ayudan a desarrollar la lógica, sino también tecnologías que se pueden transformar en productos.

### Específicos:

- Aprender distintas técnicas avanzadas de algoritmos:
  - Exactos (programación dinámica y dividir para conquistar).
  - Aproximación (greedy y metaheurísticas).

- Evolutivos (algoritmos genéticos)
  - Probabilísticos (Monte Carlo).
  - Transmisión [streaming] (algoritmos online).
- Aprender a usar las estructura de datos sofisticadas:
  - Puramente funcionales.
  - Para consultar rangos.
  - Para optimizar memoria.
  - Manipulación de cadena de texto.
- Conocer algunos algoritmos que corresponde de la computación cuántica, como el de Deutsch-Jozsa y el de Shor.

## Perfil de ingreso

Tener conocimientos de algoritmos que las siguientes estructuras de datos: arreglos, listas, tabla hash, árboles y grafos; y que tengas la capacidad de implementarlas en algún lenguaje de programación como puede ser: C, Python, Haskell. Es decir: haber cursado algún curso de algoritmos y estructuras de datos con anterioridad.

## Perfil de egresado

Al completar el bootcamp será capaz de realizar las siguientes actividades:

1. Entender la complejidad del concepto de algoritmo, y cómo este evoluciona en el tiempo.
2. Cuáles son los pro y contra de cada algoritmo y estructura de datos.
3. Comprender qué algoritmos y estructura de datos son más conveniente para desarrollo de un producto.

## Temario del Bootcamp de Algoritmos y Estructuras de Datos

**IMPORTANTE:** Tanto las tecnologías utilizadas como el caso práctico definido en cada clase son propuestas tentativas. Podrían cambiar según el parecer del profesor que dicte la clase.

# I. Fundamentos y Técnicas Avanzadas

## 1. Introducción y repaso de conceptos básicos

- Complejidad algorítmica.
- Estructuras de datos fundamentales.
- **Caso práctico:** Implementación de un sistema de gestión de tareas con análisis de complejidad.
- **Tecnologías utilizadas:** C++ y Python.

## 2. Programación Dinámica

- Principios y aplicaciones.
- Problemas clásicos (p. ej., mochila, subsecuencia común más larga).
- **Caso práctico:** Optimización de rutas de entrega para una empresa de logística.
- **Tecnologías utilizadas:** C++ y Python.

## 3. Dividir para Conquistar

- Estrategia y ejemplos.
- Implementación de algoritmos (p. ej., Merge Sort, Quick Sort).
- **Caso práctico:** Desarrollo de un sistema de búsqueda eficiente para grandes conjuntos de datos.
- **Tecnologías utilizadas:** C++ y Python.

## 4. Optimización con Algoritmos de Aproximación (Algoritmos Greedy y Metaheurísticas)

- Principios y limitaciones.
- Algoritmos de búsqueda local y global.
- **Caso práctico:** Optimización de la planificación de horarios en una institución educativa.
- **Tecnologías utilizadas:** C++, Python y Julia.

# II. Algoritmos Avanzados

## 5. Algoritmos Evolutivos

- Introducción a los algoritmos genéticos.
- Implementación y casos de uso.
- **Caso práctico:** Optimización del diseño de redes de transporte urbano.
- **Tecnologías utilizadas:** Python y Julia.

## 6. Algoritmos Probabilísticos

- Métodos de Monte Carlo.
- Algoritmos de Las Vegas.
- **Caso práctico:** Simulación de riesgos financieros para una empresa de inversiones.
- **Tecnologías utilizadas:** Julia.

## 7. Algoritmos Distribuidos

- Algoritmos de consenso (Paxos).
- Algoritmos para computación paralela (MapReduce, computación distribuida en la nube).
- **Caso práctico:** Implementación de un sistema distribuido para análisis de grandes datos en tiempo real.
- **Tecnologías utilizadas:** C++ y Erlang.

## 8. Algoritmos de Transmisión (*Streaming*)

- Procesamiento de datos en tiempo real.
- *Sketching algorithms*.
- **Caso práctico:** Desarrollo de un sistema de monitoreo de tráfico de red en tiempo real.
- **Tecnologías utilizadas:** C++ y Erlang.

# III. Estructuras de Datos Avanzadas

## 9. Estructuras de Datos Puramente Funcionales

- Listas enlazadas persistentes.
- Árboles balanceados funcionales.
- **Caso práctico:** Implementación de un sistema de control de versiones para documentos colaborativos.
- **Tecnologías utilizadas:** Racket, Haskell y OCaml.

## 10. Estructuras para Consultas de Rango

- Árboles de segmentos.
- Árboles de Fenwick (Árbol Binario Indexado).
- **Caso práctico:** Desarrollo de un sistema de análisis de datos financieros en tiempo real.
- **Tecnologías utilizadas:** C++ y Python.

## 11. Estructuras para Optimización de Memoria

- Tries y árboles de sufijos.
- Filtros de Bloom.
- **Caso práctico:** Creación de un motor de búsqueda eficiente para un diccionario online.
- **Tecnologías utilizadas:** C++ y Python.

## 12. Estructuras para Manejo de Cadenas de Texto

- Algoritmos de búsqueda de patrones.
- Árboles de sufijos y arreglos de sufijos.
- **Caso práctico:** Implementación de un sistema de detección de plagio para una plataforma educativa.
- **Tecnologías utilizadas:** C++ y Python.

## IV. Computación Cuántica

### 13. Introducción a la Computación Cuántica

- Conceptos básicos de computación cuántica.
- Algoritmos de Deutsch-Jozsa y Shor.
- **Caso práctico:** Simulación de un algoritmo cuántico simple y comparación con su contraparte clásica.
- **Tecnologías utilizadas:** Qiskit (Python).

## V. Aplicaciones y Tendencias Futuras

### 14. Tendencias Futuras en Algoritmos y Estructuras de Datos.

- Nuevas direcciones en investigación.
- Impacto de la IA y el aprendizaje automático en el diseño de algoritmos.
- Responsabilidad y ética.
- **Caso práctico:** Diseño de un sistema que integre algoritmos clásicos con técnicas de IA generativa.

### 15. Proyecto Final

- Desarrollo de un proyecto que integre múltiples conceptos del bootcamp.
- Presentación y evaluación de proyectos.
- Discusión sobre la aplicación práctica y comercialización de los proyectos.
- **Caso práctico:** Los estudiantes trabajarán en grupos para desarrollar una solución algorítmica a un problema del mundo real, desde la conceptualización hasta la implementación y presentación del prototipo.

## Otros

En algunas clases se usará Large Language Models (LLM) para entender cómo nos pueden ayudar en la etapa de diseño de nuestros algoritmos.

## Referencias

- General:
  - *Algorithms: A Quest for Absolute Definitions*, de Andreas Blass y Yuri Gurevich.
  - *Introduction to Algorithms*, 4th edition.
  - *Essential Algorithms*, de Rod Stephens, 2nd edition.

- *Introduction to Recursive Programming*, de Manuel Rubio-Sanchez.
- Estructuras de datos avanzadas:
  - *Advanced Data Structures*, de Peter Brass.
  - *Purely Functional Data Structures*, de Chris Okasaki.
- Algoritmos de aproximación:
  - [\*Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison\*](#), de Christian Blum y Andrea Roli.
- Algoritmos distribuidos:
  - *Distributed Systems* 4th edition, de Maarten van Steen y Andrew S. Tanenbaum.
- Algoritmos cuánticos:
  - *Quantum Computer Science: An Introduction*, de N. David Mermin.