



DEPARTAMENTO DE  
**INGENIERÍA  
INFORMÁTICA**  
UNIVERSIDAD DE SANTIAGO DE CHILE

# Análisis de Algoritmos y Estructuras de Datos

## ENUNCIADO LABORATORIO NRO. 2

*Profesor. Cristián Sepúlveda S.*

Noviembre 2022

## Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Resultados de aprendizaje (RdeA)</b>	<b>2</b>
<b>3</b>	<b>Instrucciones</b>	<b>2</b>
<b>4</b>	<b>Evaluación y fechas</b>	<b>4</b>
<b>5</b>	<b>Descripción del problema</b>	<b>4</b>

## 1 Introducción

La teoría de colas estudia mediante métodos formales las líneas de espera. Se modelan las colas para poder predecir, por ejemplo, su longitud y el tiempo de espera. Esta teoría es considerada una rama de la investigación de operaciones, debido a que los resultados se usan generalmente con el fin de tomar decisiones administrativas sobre los recursos necesarios para proporcionar un servicio. La teoría de colas tiene importantes aplicaciones en telecomunicaciones, ingeniería de transporte, informática, y particularmente en ingeniería industrial, en el diseño de fábricas, tiendas, oficinas y hospitales, así como en la gestión de proyectos.

## 2 Resultados de aprendizaje (RdeA)

- Formular algoritmos para problemas computacionales.
- Calcular y analizar la eficiencia de algoritmos computacionales.
- Desarrollar la capacidad de comunicarse efectivamente en español de forma escrita.
- Demostrar capacidad crítica en el análisis de resultados.

## 3 Instrucciones

1. El trabajo debe ser realizado de forma individual.
2. Describir en pseudocódigo los algoritmos propuestos, utilizando como estructura de datos lineales **solamente listas enlazadas**. La utilización de cualquier otra estructura de datos lineal para resolver el problema, por ejemplo arreglos, implicará la **calificación mínima** en el laboratorio.
3. Calcular la complejidad de tiempo para cada algoritmo que maneje listas enlazadas.
4. Implementar los algoritmos propuestos utilizando el lenguaje de programación C.
5. Generar un reporte con las siguientes secciones:
  - **Introducción**  
Se proporciona el contexto y la motivación para el trabajo. Se explica brevemente la teoría relevante con suficiente detalle como para introducir leyes, ecuaciones o teoremas relevantes. Se indica claramente el/los objetivo/s.

- **Método**  
Se describen el equipo, los materiales y los procedimientos utilizados en los experimentos. Se describen procesamientos o cálculos realizados sobre los datos utilizados.
- **Algoritmos propuestos**  
Se describen en pseudocódigo, de forma ordenada, los algoritmos utilizados para resolver el problema.
- **Cálculo de complejidad**  
Se describen los cálculos realizados para obtener la complejidad de tiempo y el orden de complejidad de los algoritmos propuestos que manejen listas enlazadas.
- **Conclusiones**  
Se recuerda al lector qué problema se estaba investigando. Se resumen los hallazgos en relación con el problema/hipótesis. Se identifican brevemente las implicaciones generales de los principales hallazgos.
- **Apéndice** (opcional)  
Se agrega información que ayuda a los lectores a comprender el proceso de investigación.
- **Referencias**  
Se enumeran los detalles de todas las publicaciones citadas en el texto, permitiendo a los lectores localizar las fuentes de forma rápida.

## 4 Evaluación y fechas

Entregable: informe en formato PDF e implementación en lenguaje C.

Ponderaciones: informe 60 %. Implementación 40 %.

Canal: Campus virtual.

Fecha de entrega: **27 de noviembre**.

## 5 Descripción del problema

La vacunación masiva de la población ha sido una de las principales medidas para enfrentar la pandemia mundial de COVID-19. Con el fin de optimizar el proceso de inmunización, se requiere simular un servicio de vacunación.

El servicio simulado consta de un módulo al que llegan pacientes a intervalos regulares de  $X$  unidades de tiempo. Al momento de llegar, los pacientes son identificados por un número mayor que cero que representa el orden de llegada al vacunatorio  $(p_1, p_2, p_3, \dots, p_n)$ . Además, se almacena el instante de llegada, siendo este un valor mayor o igual que cero y menor o igual que el valor del tiempo de la simulación  $(0, X, 2X, 3X, \dots)$ . En cada unidad de tiempo se debe verificar si se debe atender a los pacientes, para esto se debe esperar una orden aleatoria indicando que la reposición de dosis para la vacunación de los pacientes fue completada. Al momento de ser recibida la orden, se debe atender a todos los pacientes que están esperando en ese instante, asumiendo un tiempo de atención de 1 unidad de tiempo por cada paciente. En caso de que durante un periodo de atención llegue un nuevo paciente, este no será vacunado, debiendo esperar hasta que finalice el periodo de atención y que se active una nueva orden.

Se le solicita a usted la construcción de un programa en lenguaje C que simule el comportamiento de la fila del módulo de vacunación. Para su programa considere:

- Se debe construir una función nombrada **atender**, con la siguiente sintaxis:

```
atender(probabilidad): bool
```

la función debe retornar, con cierta probabilidad dada, un valor booleano que determine si se atiende o no a los pacientes en espera en ese instante.

- El programa deberá recibir como entrada el valor del tiempo de simulación, el valor del intervalo de llegada de los pacientes y la probabilidad (valor entre 0 y 1) de que la función `atender()` retorne un valor verdadero.
- El programa deberá entregar como salida los tiempos en los que se activó la orden aleatoria de atención (retorno verdadero de la función `atender`), el tiempo en el que fue atendido cada paciente, el tiempo de espera de cada paciente atendido, el tiempo promedio de espera de los pacientes vacunados, número de pacientes vacunados y número de pacientes no vacunados.
- La solución propuesta debe utilizar al menos una lista enlazada.

- Todas las estructuras lineales utilizadas para la solución deben ser listas enlazada, pilas o colas (no arreglos).
- Cada paciente debe tener asociada una estructura que mantenga al menos el tiempo de llegada y el tiempo de atención.
- Cualquier supuesto que realice para su solución, regístrela en la sección algoritmos propuestos del informe.

#### Entrada de ejemplo:

Tiempo de simulación: 45 unidades de tiempo

Intervalo de llegada: 10 unidades de tiempo

Probabilidad de la función `atender()`: 0.1

#### Salida de ejemplo:

Suponiendo que la función `atender()` retornó verdadero en las unidades de tiempo 15 y 39 (retornando falso en las restantes unidades de tiempo). La salida es:

- Tiempos de activación atención: 15, 39
- El momento de atención de cada paciente es: p1: 15, p2: 16, p3: 39, p4: 40, p5: -1 (indica sin atención)
- El tiempo de espera de cada paciente es: p1: 15, p2: 6, p3: 19, p4: 10, p5: -1 (indica sin atención)
- Tiempo promedio espera de pacientes atendidos: 12,5 unidades de tiempo
- Pacientes vacunados: 4
- Pacientes no vacunados: 1

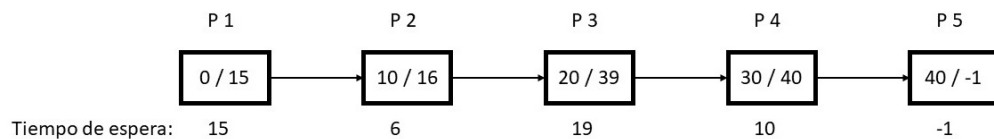


Figura nro. 1: representación de la simulación de ejemplo.