

## Inteligencia Artificial - Laboratorio 8 -

### Instrucciones:

- Esta es una actividad en grupos de no más de 4 integrantes.
  - Este grupo aún no existe en Canvas por lo que deberán unirse a uno con el nombre de **Grupo A [Hasta 3 Integrantes]** -
- Sólo es necesario que una persona del grupo suba el trabajo a Canvas.
- No se permitirá ni se aceptará cualquier indicio de copia. De presentarse, se procederá según el reglamento correspondiente.

### Tasks 1 - Teoría

Responda las siguientes preguntas de forma clara y concisa, pueden subir un PDF o bien dentro del mismo Jupyter Notebook.

1. Investigar el algoritmo AC-3 y su relación con el algoritmo de backtracking search
2. Defina en sus propias palabras el término "Arc Consistency"

### Task 2 - CSP con Backtracking, Beam y Local Search

En este ejercicio, implementará tres algoritmos diferentes de satisfacción de restricciones para resolver un problema de programación de exámenes para cuatro estudiantes que toman siete exámenes diferentes. El problema implica calendarizar los exámenes para los estudiantes respetando diversas limitaciones y preferencias.

Restricciones:

- Todos los exámenes deberán realizarse en días diferentes, concretamente lunes, martes y miércoles.
- Ningún estudiante deberá tener más de un examen por día.
- Los estudiantes que toman el mismo curso no pueden tener exámenes programados para el mismo día.

Para este ejercicio **puede utilizar librerías**, pero se sugiere intentar hacer los algoritmos por ustedes mismos. Para programar su solución pueden considerar (pero esto **no significa** que sea una guía definitiva) lo siguiente:

1. Definir las variables:
  - a. Definir variables que representen exámenes (cursos).
  - b. Defina el dominio para cada variable, especificando los días posibles (lunes, martes, miércoles) para cada examen.
2. Definir las restricciones:
  - a. Defina restricciones para garantizar que todos los exámenes se realicen en días diferentes, que ningún estudiante tenga más de un examen por día y que los estudiantes que toman el mismo curso no tengan exámenes el mismo día.
3. Implemente el algoritmo de backtracking search:
  - a. Implemente el algoritmo de backtracking search para encontrar una solución.
  - b. Utilice el retroceso para explorar el espacio de búsqueda y encontrar una asignación válida de variables.
4. Implemente el algoritmo de beam search:
  - a. Implemente el algoritmo de beam search para encontrar una solución.
  - b. Utilice una heurística para seleccionar "camino" prometedores y explorar el espacio de búsqueda de manera eficiente.
5. Implemente el algoritmo de local search:
  - a. Implemente el algoritmo de local search para encontrar una solución.
  - b. Utilice una heurística para mejorar iterativamente la solución actual explorando soluciones vecinas.

De cada uno de los algoritmos implementados, tome el tiempo que le toma encontrar una solución, y **compare** no solo el **tiempo**, sino también la **solución encontrada** de cada uno. Escriba sus conclusiones como parte de una celda "markdown" en su Jupyter Notebook.

## Inteligencia Artificial - Laboratorio 8 -

---

### Entregas en Canvas

1. Link al repositorio de los integrantes del grupo.
  - a. Deberán subir el código también a Canvas por temas de Acreditación

### Evaluación

1. [1 pts.] Task 1 (0.5 pts cada pregunta)
2. [3 pts.] Task 2 - Implementación de algoritmos (1pt cada algoritmo)
3. [1 pts.] Task 2 - Conclusiones

Total 5 pts.