

## Proyecto de Estructuras de Datos: Algoritmo de Compatibilidad de Parejas

Este proyecto final tiene como objetivo desarrollar un algoritmo para encontrar parejas compatibles a partir de sus gustos y descripciones. El algoritmo debe ser capaz de identificar las parejas con mayor afinidad con base a la información proporcionada. El proyecto se enfocará en la evaluación de la estructura de datos utilizada y las estrategias de solución implementadas para lograr una búsqueda eficiente y precisa.

### Objetivos:

- Diseñar e implementar un algoritmo para encontrar parejas compatibles a partir de gustos y descripciones.
- Evaluar la eficiencia y precisión del algoritmo utilizando conceptos de estructuras de datos y estrategias de búsqueda.
- Desarrollar un programa parametrizable que permita leer datos desde un archivo y ejecutar el algoritmo con diferentes parámetros.

### Descripción del Problema:

El problema consiste en encontrar parejas compatibles a partir de un conjunto de datos que contiene información sobre los gustos y descripciones de las personas. La compatibilidad se determinará en base a la similitud de los intereses y características personales. El algoritmo debe ser capaz de identificar las parejas con mayor afinidad entre todas las posibles combinaciones.

### Metodología:

1. **Recolección de datos:** Se utilizará un conjunto de datos predefinido que contenga información sobre los gustos y descripciones de las personas. El conjunto de datos puede incluir información como aficiones, preferencias musicales, películas favoritas, valores, etc.
2. **Preprocesamiento de datos:** Los datos se preprocesarán para eliminar valores faltantes, normalizar los datos y prepararlos para su uso en el algoritmo.
3. **Selección de la estructura de datos:** Se evaluarán diferentes estructuras de datos para almacenar la información de las personas y sus características. Algunas opciones a considerar incluyen:
  - **Listas:** Ofrecen flexibilidad para agregar, eliminar y modificar elementos dinámicamente.
  - **Árboles:** Facilitan la búsqueda y clasificación de datos jerárquicos.
  - **Grafos:** Representan relaciones entre elementos, permitiendo identificar conexiones y similitudes.
4. **Diseño del algoritmo:** Se diseñará un algoritmo para encontrar parejas compatibles utilizando la estructura de datos seleccionada. El algoritmo debe considerar una o más, de diferentes estrategias de búsqueda, como:
  - **Búsqueda por fuerza bruta:** Evalúa todas las posibles combinaciones de parejas.
  - **Búsqueda binaria:** Busca parejas compatibles de manera eficiente en estructuras ordenadas.
  - **Algoritmos de emparejamiento:** Utilizan técnicas de optimización para encontrar parejas con mayor afinidad.
5. **Evaluación del algoritmo:** Se evaluará el rendimiento del algoritmo utilizando diferentes métricas, como tiempo de ejecución, precisión y memoria utilizada. Se compararán los resultados obtenidos con diferentes estructuras de datos y estrategias de búsqueda.
6. **Desarrollo del programa:** Se implementará un programa parametrizable que permita leer datos desde un archivo, seleccionar la estructura de datos y la estrategia de búsqueda, y ejecutar el algoritmo para encontrar parejas compatibles. El programa debe ser flexible y adaptable a diferentes formatos de datos.

### Lenguaje de programación:

El lenguaje de programación es libre, a elección del estudiante

### Evaluación:

El proyecto se evaluará en base a los siguientes criterios:

- **Corrección funcional:** El algoritmo debe ser capaz de encontrar parejas compatibles de manera precisa y eficiente.
- **Elección de la estructura de datos:** La estructura de datos seleccionada debe ser adecuada para el problema y debe justificarse adecuadamente.
- **Diseño del algoritmo:** El algoritmo debe estar bien diseñado y documentado, utilizando estrategias de búsqueda apropiadas.
- **Implementación del programa:** El programa debe ser legible, modular y fácil de usar, permitiendo la parametrización de la estructura de datos y la estrategia de búsqueda.

### Visión:

Este proyecto permitirá al estudiante aplicar los conceptos aprendidos en Estructuras de Datos para resolver un problema real y significativo. El estudiante tendrá la oportunidad de evaluar diferentes estructuras de datos y estrategias de búsqueda, y desarrollar un programa parametrizable que pueda ser utilizado para encontrar parejas compatibles en diferentes escenarios. Además, el proyecto fomentará el desarrollo de habilidades de análisis, diseño, implementación y evaluación de algoritmos.

### Recomendaciones:

- Se recomienda que el estudiante investigue y explore diferentes algoritmos de emparejamiento y estrategias de búsqueda para encontrar la solución más adecuada para el problema.
- Es importante utilizar técnicas de programación adecuadas, como la modularidad, la documentación y la gestión de errores, para garantizar la calidad y mantenibilidad del código.
- Se recomienda probar el programa con diferentes conjuntos de datos para evaluar su rendimiento y precisión en diferentes escenarios.
- Es importante documentar el proceso de desarrollo del proyecto, incluyendo la selección de la estructura de datos, el diseño del algoritmo, la implementación del programa y la evaluación del rendimiento.

### Entrega del Proyecto:

- Se puede trabajar por grupo y se entregará de forma oral y escrita.
- Se debe solicitar una reunión de 15 minutos, puede ser presencial o virtual. para este proposito se debe solicitar una reunión antes del 29 de mayo a las 11:00 pm, proponiendo un día y una hora, y esperando confirmación por parte del profesor, para confirmar hora.
- En <https://docs.google.com/spreadsheets/d/1omm8gRvXds7Mn77B5iWavVWevGHY9mJjiBIKQJaOxZk/edit?usp=sharing> se reserva y hay que noticiar confirmar.
- Cada equipo es responsable de la citación.

# Dating

## Features:

- \* Perfil (Info personal)
- \* Matching (Perfil - Perfil, valor de compatibilidad)

## Preprocesamiento

Anchura de perfiles

Contiene los parámetros para construir perfiles  
"gustos y descripciones"

Perfiles

- Información personal { Nombre, gustos, desc. }
- Datos para computar compatibilidad  
↳ Vector de características

Matches / Puntuaje P-P

- \* Puntuaje de compatibilidad  
↳ Comparación coseno (Ángulo)  
$$\frac{A \cdot B}{\|A\| \|B\|}$$
  
A y B vectores característicos de dos perfiles

A misos

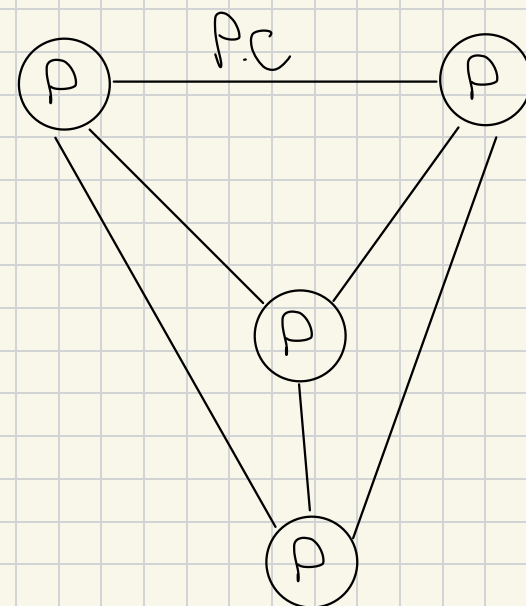
Perfil A  $\xrightarrow{\text{Puntuaje Comp}}$  Perfil B

Notas: \* Gustos y descripciones cómo los valores vector de características

Ó uso en conjunto cualitativo y cuantitativo para generar el p.c

\* Monitorean tiempo de cómputo y uso de memoria

Mapa de compatibilidad



## Emparejamiento

Grafo:

Buscar vecinos óptimos

(mayor P.C.)

Nearest Neighbor

=>

Retornan Mejores matches para un cierto perfil

## Recolección de datos

- 1 Leer archivos de entrada
- 2 Extraer y almacenar perfiles

¿ Cómo convertir?

Jacobo  
Música  
Ciencia  
Deporte

Perfil

Nombre = "Jacobo"

características = [0.1, 1.3, 5.0, ...]

Características = Rangos de personalidad

- \* Apertura a la experiencia
- \* Escrupulosidad
- \* Amabilidad
- \* Neuroticismo
- \* Empatía
- \* Asunción de riesgos
- \* Comunicación Asertiva

Valores  $\in [-1, 1]$

0.5  
-0.6  
0.7  
-0.6  
0.6  
0.9  
1

Representación  
de una persona

Vector para cálculo de  
compatibilidad

Gustos = Intereses con impacto en rangos de personalidad

Cada gusto impacta típicamente uno o más rangos de personalidad

Base de datos de gustos con:

Gusto ( descripción, impacto)

e.g.

Música,  $\begin{bmatrix} 0.0 \\ +0.5 \\ -0.1 \\ 0 \\ -1 \end{bmatrix}$

Estos valores producen el vector de características de una persona

Ej.

Juan tiene los siguientes intereses:

A: Peppercorn

B: Ciencia

C: Viajar

D: Jardinería

Cada interés tiene impacto en cómo se describe la personalidad de Juan

A  $\begin{pmatrix} +1 \\ -1 \\ 0 \\ +0.5 \\ 0 \end{pmatrix}$  B  $\begin{pmatrix} 0 \\ +1 \\ -0.7 \\ 0 \\ +0.2 \end{pmatrix}$  C  $\begin{pmatrix} 0 \\ 0 \\ +1 \\ +1 \\ 0 \end{pmatrix}$  D  $\begin{pmatrix} -0.6 \\ -0.2 \\ +0.4 \\ 0 \\ -1 \end{pmatrix}$

$\rightarrow A + B + C + D = \begin{pmatrix} 0.4 \\ -0.2 \\ 0.7 \\ 1.5 \\ -0.8 \end{pmatrix}$  X representa la personalidad de Juan

X luego se normaliza en  $[-1, 1]$  y está listo para procesarse

Módulos

