

## **Proyecto.**

**Curso:** Introducción a la Programación.

**Docente:** Alejandro Alfaro Quesada. (aalfaroq@utn.ac.cr)

**Fecha de entrega:** domingo 31 de agosto del 2025.

**Hora de entrega:** 11:59 pm.

## **Indicaciones Generales.**

- La prueba se desarrolla bajo la modalidad de proyecto, con tiempo de inicio y fin predeterminado.
- Toda prueba evaluada es considerada como un único proyecto, es decir, aunque la evaluación sea dividida por secciones, la completitud de uno o más rubros puede afectar la funcionalidad del sistema, y esto, podría influir en la nota final.
- La prueba puede ser desarrollada en un máximo de dos personas. En cuanto a los equipos de trabajo hay dos reglas que se deben contemplar, las cuales son:
  - Para los casos que son parejas, si por alguna razón no puedan continuar trabajando juntos, ambos deben continuar con el código fuente que tienen hasta ese momento; es decir, no se permite que se unan a otro compañero y/o utilizar el código fuente de otro compañero.
  - Aunque el trabajo sea desarrollado entre dos personas, en términos de la evaluación y asignación de la nota, ambos estudiantes serán evaluados de manera individual; lo anterior, para el caso en que el docente compruebe que uno de los estudiantes tuvo poca o nula participación en el desarrollo del proyecto, de acuerdo con las preguntas que se realizarán en el momento de la defensa del proyecto.
- Si algún estudiante es sorprendido en actos fraudulentos, ya sea que los provoque o los consienta, su prueba y la de los demás implicados, será automáticamente anulada.
- El docente ejecuta el código fuente que está en el proyecto enviado para comprobar que realiza lo solicitado. Por lo que, el docente NO modificará ninguna línea de código, por lo tanto, debe asegurarse que el código fuente funciona correctamente.
- El proyecto con el código solo se recibe y se valida desde el Campus Virtual de la UTN. Por cualquier otro medio que se envíe al docente no será revisado.
- Para la explicación del trabajo el estudiante debe estar de manera presencial en la clase, de lo contrario, se rebajan 15 puntos de la nota que obtenga en el momento de la evaluación.

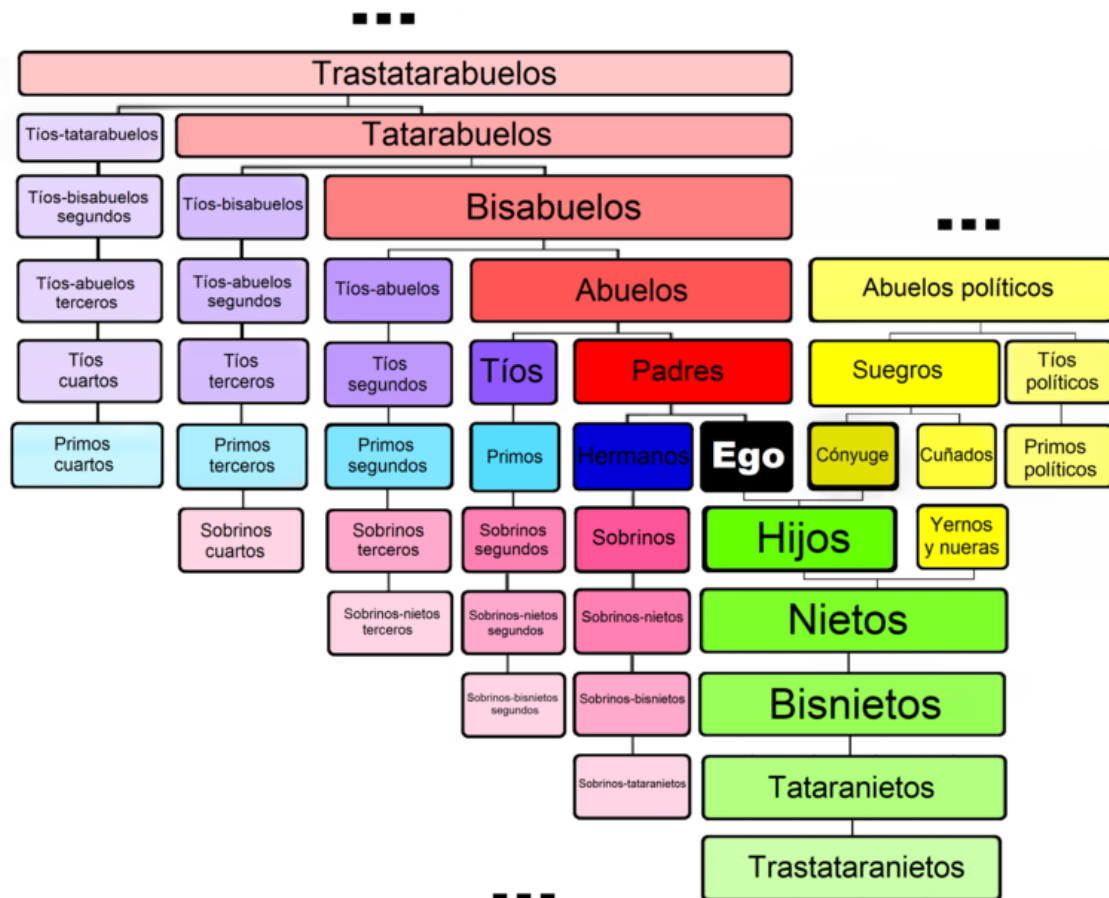
# Árbol Genealógico

**Nota:** Para resolver este ejercicio se prohíbe el uso de librerías como numpy, pandas o similares de alto nivel (si considera alguna y tiene dudas, consulte al docente), debe ser utilizando únicamente las estructuras de datos vistas en clase; además, se prohíbe el uso de archivos de cualquier formato, el uso de las librerías o archivos mencionados invalida por completo la prueba. Para este proyecto **se debe utilizar interfaz gráfica**, no debe haber ninguna acción que ocurra en consola.

## Objetivo

El Sistema de Árbol Genealógico es una aplicación diseñada para registrar, visualizar y gestionar la información de los miembros de su familia a través de generaciones. Permitirá crear su árbol genealógico, agregar información detallada sobre sus familiares y las relaciones familiares, así como eventos relevantes como nacimientos y fallecimientos.





grado cero	primer grado	segundo grado	tercer grado	cuarto grado	quinto grado	sexto grado
ego	padres	abuelos	bisabuelos	tatarabuelos	trastatarabuelos	pentabuelo
	hijos	nietos	bisnietos	tataranietos	trastaranietos	pentanietos
		hermanos	tíos	primos	tíos-bisabuelos	tíos-tatarabuelos
			sobrinos	tíos-abuelos	sobrinos-bisnietos	sobrinos-tataranietos
				sobrinos-nietos	tíos segundos	tíos terceros
					sobrinos-segundos	sobrinos-terceros

### Gestión de registro de familias.

- El sistema debe permitir insertar una nueva familia, para lo cual se le asigna un identificador y un nombre.

### Gestión de integrantes de familias.

- El sistema debe permitir insertar personas a la familia; los datos que se agregan son: cédula, nombre, fecha de nacimiento, fecha de fallecimiento (si corresponde), género, lugar de residencia (serán las provincias), estado civil (serán los normales).
- Para el género, provincias y estado civil, utilice menús para solicitar la información.

### Gestión de relaciones familiares.

- El sistema debe permitir relacionar a las personas de la familia para que se vaya creando el árbol genealógico; los registros deben ser:
  - Registro de padres (pueden ser biológicos o adoptivos).
  - Registro de hijos (biológicos o adoptivos).
  - Registro de hermanos.
  - Uniones de pareja (casados o unión libre).
- Cuando se crea una unión de pareja, debe actualizarse los registros de las personas involucradas para garantizar que se construya el árbol genealógico.
- No se debe agregar la relación de parentesco entre personas, por ejemplo, decir que A es el papá de B; si no, que se debe buscar y proponer una estrategia para relacionar las personas y que se evidencie la construcción del árbol genealógico.

### Gestión de eventos familiares.

- Las personas pueden ir creciendo con un efecto acelerado del tiempo, podrán unirse con otras personas del género opuesto y tener descendencias, lo anterior cumpliendo las siguientes reglas:
  - **Cumpleaños:** Cada 10 segundos (para efectos prácticos) todas las personas cumplen un año más de vida, por lo cual se debe actualizar el dato de cada uno. Este proceso ocurre dentro del código.
  - **Nacimientos:** Las uniones de pareja podrán tener hijos, solo pueden tener hijos las parejas con compatibilidad suficiente; la información del hijo será aleatoria desde el código, es decir, la cédula es autogenerada,

el nombre se asigna, el género es aleatorio, la fecha de nacimiento es la actual, lugar de residencia el de uno de los padres, estado civil es soltero, fallecido por defecto en no, matrimonio por defecto en no, relación familiar hijo y se asocia a los padres.

- **Fallecimientos:** Cada 10 segundos (para efectos prácticos) y aleatoriamente las personas pueden fallecer, siempre y cuando no esté fallecido; por lo cual, se debe actualizar el dato de cada uno. Este proceso ocurre dentro del código.
- **Uniones de pareja:** Las personas podrán decidir unirse a otra persona siempre y cuando:
  - Sean mayor de 18 años.
  - No estar unida a otra persona.
  - Edad compatible (diferencia no mayor a 15 años).
  - Afinidad emocional (puntaje de compatibilidad en intereses; se debe contemplar al menos dos afinidades). Se debe implementar una función que calcule un "Índice de compatibilidad" entre dos personas y permita la unión solo si el puntaje supera un umbral, por ejemplo, 70%.
  - Compatibilidad genética que evite descendencia con riesgos.

### Gestión de efectos colaterales.

- Ciertos eventos familiares tienen impacto sobre otros miembros del árbol:
  - Si muere la madre y padre de hijos menores de 18 años, éstos se deben reasignar a un tutor legal superior (tía, abuela, etc.).
  - Si una persona queda viuda, cambia su estado civil, y tiene una probabilidad menor de volverse a unir.
  - Si una persona pasa muchos años soltera, su salud emocional baja y afecta su esperanza de vida.

### Gestión de historial y línea del tiempo.

- Cada persona debe mantener un historial cronológico de eventos personales y se debe representar de manera visual, como:
  - Año en que nació.
  - Año en que se unió como pareja.
  - Año en que tuvo hijos.
  - Año en que enviudó.
  - Año en que falleció.

## Representación gráfica del árbol genealógico.

- El sistema debe permitir mostrar el árbol genealógico de cada una de las familias agregadas, en donde se visualice claramente el parentesco de una persona con la otra; por ejemplo, no es mostrar que una persona tiene un parentesco de primo, si no que, se visualice de quién es primo para entender las relaciones de la familia.
- Debe realizar una propuesta gráfica que permita entender y visualizar el árbol genealógico con mucha facilidad.

## Búsquedas.

- El sistema debe tener las siguientes consultas:
  1. ¿Cuál es la relación entre persona A y persona B?
  2. ¿Quiénes son los primos de primer grado de X?
  3. ¿Cuáles son todos los antepasados maternos de X?
  4. ¿Cuáles descendientes de X están vivos actualmente?
  5. ¿Cuántas personas nacieron en los últimos 10 años?
  6. ¿Cuáles parejas actuales tienen 2 o más hijos en común?
  7. ¿Cuántas personas fallecieron antes de cumplir 50 años?

## Documentación.

Debe crear y subir junto con la solución un documento (Word) en donde explique lo siguiente:

- **Lógica del algoritmo:** debe explicar claramente cómo funciona el algoritmo que desarrolló para resolver los problemas de las secciones que se listan en el cuadro siguiente; en cada explicación coloque una imagen del código donde realiza el proceso, así como datos que ud considere relevantes para explicar lo que desarrolló.
- Para el momento de la defensa del proyecto debe presentar ejemplos para probar la funcionalidad del sistema de acuerdo con lo solicitado.
- Agregue la siguiente tabla completa:



Sección	¿La sección está completamente terminada y funcional? Indique sí o no.	Si colocó que no, indique una breve descripción del por qué no está completa la sección.
Gestión de registro de familias.		
Gestión de integrantes de familias.		
Gestión de relaciones familiares.		
Gestión de eventos familiares.		
Gestión de efectos colaterales.		
Gestión de historial y línea del tiempo.		
Representación gráfica del árbol genealógico.		
Búsquedas.		