

Título del proyecto: el desafío de los gutierrez

Alumnos:

Aldo Manfredi (manfredialdo.1979@gmail.com)

Bruno Pighin (brunopighin@hotmail.com)

Materia: Programación I

Profesor: AUS Bruselario, Sebastián

Fecha de Entrega: 9 de junio de 2025

INTRODUCCION

En este trabajo logramos integrar los conceptos teóricos sobre árboles como estructura de datos con su aplicación práctica mediante la resolución de un problema realista, la inseguridad alimentaria en los hogares argentinos.

A razón de este tema, implementamos un programa que simula las decisiones cotidianas de una familia con recursos limitados, utilizando un árbol binario de decisiones.

Este enfoque permite no sólo ejercitar técnicas de programación, como la recursividad y el manejo de estructuras anidadas, sino también reflexionar sobre la utilidad de los árboles en la resolución de problemas del mundo real.

MARCO TEORICO

El presente trabajo se fundamenta en la aplicación de estructuras de datos avanzadas, específicamente los árboles de decisión binarios, para modelar escenarios complejos de toma de decisiones. Un árbol de decisión es una representación jerárquica que utiliza nodos para formular preguntas o condiciones, y ramas para indicar los posibles caminos basados en las respuestas ("sí" o "no" en este caso). Estos caminos conducen eventualmente a nodos hoja, que representan los resultados o conclusiones finales del proceso decisorio.

La profundidad de un nodo se define por su distancia desde el nodo raíz (la primera pregunta), mientras que la longitud se refiere al número de pasos en un camino específico hasta una hoja. Mediante el uso de la recursividad, el árbol puede ser recorrido eficientemente para evaluar cada trayectoria. La incorporación de la simulación aleatoria permite explorar la totalidad de las posibles combinaciones de decisiones, ofreciendo una visión integral de los diversos desenlaces en un contexto de incertidumbre. Este enfoque facilita el análisis y la comprensión de sistemas donde múltiples factores interconectados influyen en el resultado final.

algunos ejemplos que se van a trabajar...

Ejemplo nodo raíz: ¿Queda plata en la billetera?

Ejemplo nodo Hoja: ¡Solidaridad doble! Hoy comemos y ayudamos a otros.

Ejemplo nodo rama: ¿Podemos al menos conseguir algún producto de segunda marca?

CASO PRACTICO

Este código simula las decisiones diarias de una familia típica en un contexto de crisis, inflación e incertidumbre..., los Gutiérrez son quienes sobreviven para conseguir alimentos en ese panorama. Usa un árbol de decisiones para representar las diferentes opciones y sus consecuencias, desde ver si tienen dinero hasta buscar ayuda comunitaria. Cada paso en la simulación es una respuesta aleatoria a una pregunta, y al final, el programa sugiere un "menú del día" basado en el resultado obtenido.

código

```
# el desafio de los gutierrez
```

```
# progrmaacion1 utn tup
```

```
# trabajo pracitco integrador version02
```

```
# datos avanzados, arboles de decision
```

```
# aldo manfredi // bruno pighin
```

```
"""
```

La familia gutierrez, como muchas en Argentina, enfrenta diariamente el desafío

de dudar sobre su alimentación en un contexto de alta inflación, y recursos

limitados. Cada mañana, se encuentran ante una serie de decisiones difíciles que

determinarán si podrán poner algo en la mesa ese día.

No hay un camino fácil, y cada elección tiene consecuencias directas en su bienestar.

Nos piden, una forma de simular estas complejas decisiones para entender las posibles trayectorias y desenlaces, así como los factores que influyen en su situación.

Esto es solo una simulacion del día a día de la familia.

"""

```
import random
```

```
# Árbol binario
```

```
manual_cesta_basica = [
```

```
    "¿Queda plata en la billetera?",
```

```
    [
```

```
        "¿Hay alguna oferta o descuento en el super de la esquina?",
```

```
        [
```

```
            "¿podre comprar algo esencial (pan y/o leche y/o arroz)?",
```

```
            [
```

```
                "¿Podemos al menos conseguir algun producto de segunda marca?",
```

```
                [
```

```
                    "¿hay espacio para guardar varios cosas?",
```

```
                    [
```

```
                        "¿Podemos compartir con otro hogar cercano?",
```

```
                        "¡Solidaridad doble! Hoy comemos y ayudamos a otros.", # 'solidaridad'
```

```
                        "¡Aseguramos provisiones, pero solo para casa!" # 'provisiones'
```

],

"¡Compramos lo justo! No hay lugar para almacenar más." # 'compramos lo justo'

],

"¡Sobrevivimos hoy, pero mañana será otro desafío." # 'sobrevivimos'
(corregida la tilde, coincide)

],

"El descuento no alcanza, ¡solo podemos comprar lo barato!" # No hay una clave directa, pero podemos ver si una palabra común sirve.

],

[

"¿Podemos pedir fiado en la verdulería?",

[

"¿El verdulero ya nos ha fiado esta semana?",

"¡Al límite de la confianza, pero hoy se come, gracias al fiado!", # 'fiado'

"¡Gracias a la confianza del barrio, salvamos el día con el fiado!" # 'fiado'

],

"¡**GRAVE**: No hay ofertas, no alcanza ni fiado, ni piedad, ni opción para hoy."
'grave'

]

],

[

"¿Queda algo de comida de anda saber cuando en esa heladera casi vacía?",

[

```

"¿Se puede repartir entre todos aunque sea poco?",

[

  "¿Algún vecino puede aportar algo a la olla?",

  "¡Comunidad unida! Hoy comemos todos." # 'comemos todos'

  "¡Comemos poco, pero comemos juntos!" # 'comemos juntos'

],

"¡**CRÍTICO**: No alcanza, algunos pasarán hambre hoy." # 'crítico'

],

[

  "¿Hay comedor comunitario cerca?",

  [

    "¿Esta abierto hoy, hay lugar?",

    "¡Recibimos la ayuda vital del comedor, pero quedamos hambrientos!", #
'ayuda vital'

    "Ya no queda ni las migas.. El comedor esta saturado, preferimos dormir
vacíos intentaremos mañana." # 'saturado'

  ],

  "¡**DESASTRE TOTAL**: No hay comida, dinero ni ayuda visible." # 'desastre'

]

]

]

```

```
menu_del_dia = {

    "solidaridad": "asado con amigos y mate compartido",

    "provisiones": "guiso de lentejas y pan casero",

    "compramos lo justo": "fideos con manteca",

    "sobrevivimos": "te con leche tibia y galletitas de agua",

    "fiado": "sopa improvisada con una alita de pollo que sobro de hace dos dias",

    "grave": "café negro y sin azucar... hay que agradecer?",

    "crítico": "un vaso de agua y una mirada al cielo",

    "desastre": "nos miramos las caras y nos vamos a dormir... gracias",

    "comemos todos": "arroz con aceite y un cuarto de choripan para cada uno... gracias",

    "comemos juntos": "pan con manteca compartido, un fiasco!",

    "ayuda vital": "una bolsita chiquita tipo vianda comunitaria caliente",

    "saturado": "mate cocido frio con pan duro"

}

def obtener_menu(resultado_final):

    # busca palabras clave en el resultado final del arbol para sugerir un

    # menu del dia.

    texto = resultado_final.lower()

    for clave, menu in menu_del_dia.items():

        if clave in texto:

            return menu

    return "lo que tengo no alcanza.. hoy: nada... por ahora."
```

```
def evaluar_arbol(arbol, decidir, historial=None):

    # recorre recursivamente el arbol de decisiones, tomando y registrando
    # cada decisión.

    if historial is None:

        historial = []

    if isinstance(arbol, str):

        historial.append(("RESULTADO", arbol))

        return historial

    # if not isinstance(arbol, list) or len(arbol) != 3:

    #     raise ValueError("Estructura del árbol inválida. Debe ser [pregunta, rama_si,
    # rama_no].")

    pregunta, rama_si, rama_no = arbol

    decision = decidir(pregunta)

    historial.append((pregunta, "sí" if decision else "no"))

    siguiente_rama = rama_si if decision else rama_no

    return evaluar_arbol(siguiente_rama, decidir, historial)

def respuestas_aleatorias(pregunta):
```



```

# simula un "sí" o "no" al azar para cada pregunta del arbol.

respuesta = random.choice([True, False])

print("\n" + pregunta)

print("Respuesta aleatoria:", "sí" if respuesta else "no")

return respuesta


def imprimir_historial(historial):

    # muestra la secuencia de decisiones tomadas y el menú final determinado.

    print("\n=== Decisiones tomadas ===")

    resultado_final = ""

    for paso in historial:

        if paso[0] == "RESULTADO":

            resultado_final = paso[1]

        else:

            print(f"{paso[0]} → {paso[1]}")

    print(f"\nResultado final: {resultado_final}")

    menu = obtener_menu(resultado_final)

    print(f"Menu del dia: hoy {menu}.")


def inicio():

    # Principal. Ejecutar simulación

```

```

print("=== Simulacion aleatoria con menu del dia ===")

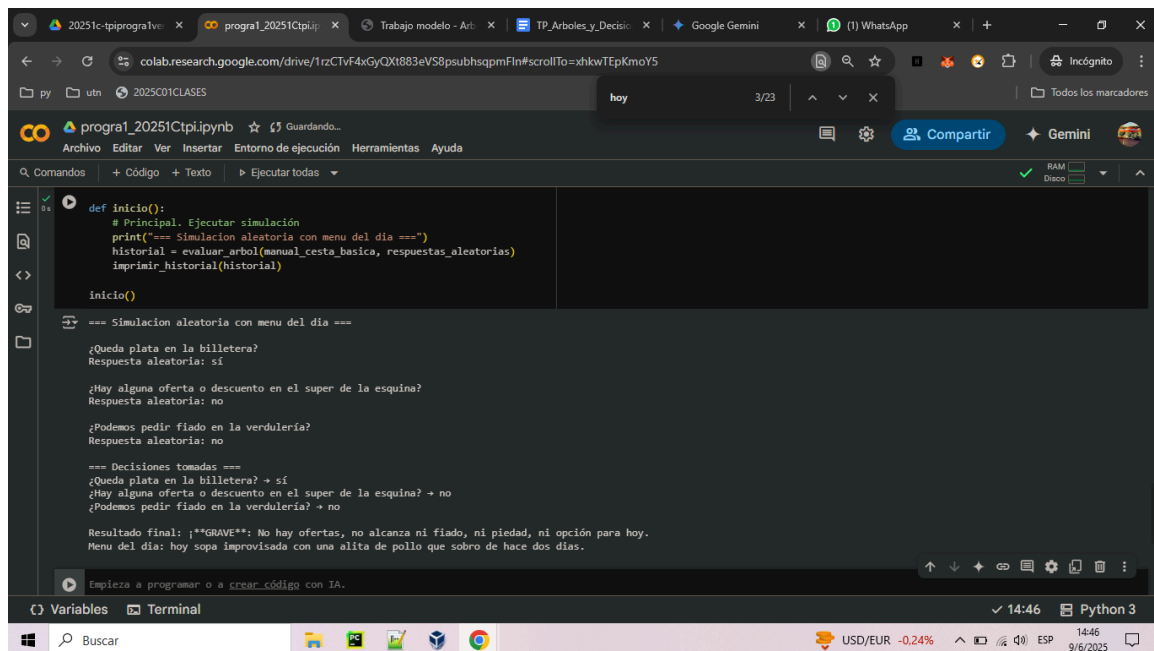
historial = evaluar_arbol(manual_cesta_basica, respuestas_aleatorias)

imprimir_historial(historial)

inicio()

```

ejemplo captura resultado



```

def inicio():
    # Principal. Ejecutar simulación
    print("=== Simulacion aleatoria con menu del dia ===")
    historial = evaluar_arbol(manual_cesta_basica, respuestas_aleatorias)
    imprimir_historial(historial)

    inicio()

=== Simulacion aleatoria con menu del dia ===

¿Queda plata en la billetera?
Respuesta aleatoria: sí

¿Hay alguna oferta o descuento en el super de la esquina?
Respuesta aleatoria: no

¿Podemos pedir fiado en la verdulería?
Respuesta aleatoria: no

=== Decisiones tomadas ===
¿Queda plata en la billetera? + sí
¿Hay alguna oferta o descuento en el super de la esquina? + no
¿Podemos pedir fiado en la verdulería? + no

Resultado final: **¡GRAVE**! No hay ofertas, no alcanza ni fiado, ni piedad, ni opción para hoy.
Menu del día: hoy sopa improvisada con una alita de pollo que sobro de hace dos días.

```

METODOLOGIA UTILIZADA

El código utiliza la metodología de Árboles de Decisión para simular la toma de decisiones. Representa un escenario complejo (la situación alimentaria de una familia) como una serie de preguntas ramificadas. Cada pregunta lleva a una de dos opciones ("sí" o "no"), avanzando por el árbol hasta llegar a un resultado final o "hoja". La simulación es aleatoria, lo que permite explorar diferentes caminos y resultados posibles de forma automática. Finalmente, asocia cada resultado final con un "menú del día" para contextualizar las consecuencias.

ALGUNOS CONCEPTOS TRABAJADOS

Árbol de Decisión: La estructura principal que modela las decisiones y sus posibles desenlaces.

Recorridos de Árbol (Recursividad): La función `evaluar_arbol` utiliza la recursividad para navegar por la estructura del árbol, moviéndose de pregunta en pregunta.

Simulación: El uso de `random.choice` para las respuestas permite simular aleatoriamente el proceso de toma de decisiones, generando diferentes escenarios en cada ejecución.

Estructuras de Datos Anidadas (Listas): El árbol se define como una lista de listas anidadas, que permite representar la jerarquía y ramificaciones.

Mapeo (Diccionario): El `menu_del_dia` es un diccionario que mapea palabras claves de los resultados finales del árbol a descripciones de comidas, interpretando los desenlaces.

RESULTADOS OBTENIDOS

La simulación del "Desafío de los Gutiérrez" nos permitió observar las diversas trayectorias que la familia podría tomar en su búsqueda diaria de alimentos. Al ejecutar el código múltiples veces, se generaron distintos escenarios, cada uno culminando en un "Menú del Día" particular.

Por ejemplo, una ejecución podría mostrar una secuencia como la siguiente:

"¿Queda plata en la billetera?" → Sí

"¿Hay alguna oferta o descuento en el super de la esquina?" → Sí

"¿podré comprar algo esencial (pan y/o leche y/o arroz)?" → Sí

"¿Podemos al menos conseguir algún producto de segunda marca?" → Sí

"¿hay espacio para guardar varios días?" → Sí

"¿Podemos compartir con otro hogar cercano?" → Sí

Resultado final: "¡Solidaridad doble! Hoy comemos y ayudamos a otros." Menú del día: asado con amigos y mate compartido.

En otro caso, la simulación podría revelar una situación más compleja:

"¿Queda plata en la billetera?" → No

"¿Queda algo de comida de “anda saber cuando” en esa heladera casi vacía?"
→ Sí

"¿Se puede repartir entre todos aunque sea poco?" → No

Resultado final: "¡CRÍTICO: No alcanza, algunos pasarán hambre hoy." Menú del día: un vaso de agua y una mirada al cielo.

Estos ejemplos demuestran cómo la aleatoriedad de las respuestas guía a la familia por diferentes ramas del árbol, desde situaciones de alivio y colaboración hasta escenarios de extrema dificultad. La diversidad de los resultados resalta la complejidad y la incertidumbre inherentes a su día a día.

CONCLUSIONES (semi reflexion final)

Este modelo de árbol de decisiones simula de forma efectiva el problema alimentario de una familia en Argentina, mostrando cómo cada elección, incluso las aleatorias, conduce a resultados muy variados. La simulación subraya la dependencia y vulnerabilidad de factores externos, pero también destaca la resistencia y el valor de la solidaridad comunitaria. Más allá de la técnica, el código ofrece una ventana a una realidad difícil, invitando a la empatía y a la reflexión sobre la seguridad alimentaria de manera muy simplista...

Ademas el trabajo permitió: - Integrar los conceptos teóricos sobre árboles con una aplicación práctica. - Comprender la utilidad de los árboles en la representación de procesos de decisión complejos. - Aplicar técnicas de recursividad y estructuras anidadas en Python. - Reflexionar sobre problemáticas sociales reales mediante el uso de herramientas informáticas.

BIBLIOGRAFIA

apuntes de catedra unidad 7 datos complejos, Estructura de datos avanzadas arboles. Apunte teórico sobre árboles.

<https://tup.sied.utn.edu.ar/course/view.php?id=12§ion=67#:~:text=Apunte%20te%C3%B3rico%20sobre%20%C3%A1rboles>

ANEXO (links github, youtube,

```
C:\Users\bruno> OneDrive\Escritorio\progra1tpi.py > ...
120 def respuestas_aleatorias(pregunta):
121     print("\n" + pregunta)
122     print("Respuesta aleatoria:", "sí" if respuesta else "no")
123     return respuesta
124
125 def imprimir_historial(historial):
126     # muestra la secuencia de decisiones tomadas y el menú final determinado.
127     print("\n=== Decisiones tomadas ===")
128     resultado_final = ""
129     for paso in historial:
130         if paso[0] == "RESULTADO":
131             resultado_final = paso[1]
132         else:
133             print(f"{paso[0]} -> {paso[1]}")
134
135     print(f"\nResultado final: {resultado_final}")
136     menu = obtener_menu(resultado_final)
137     print(f"Menu del día: hoy {menu}.")
138
139 def inicio():
140     # Principal. Ejecutar simulación
141     print("=== Simulación aleatoria con menú del día ===")
142     historial = evaluar_arbol(manual_cesta_basica, respuestas_aleatorias)
143     imprimir_historial(historial)
144
145 inicio()
146
147
148
149
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\bruno> & C:/Users/bruno/AppData/Local/Programs/Python/Python312/python.exe c:/Users/bruno/OneDrive/Escritorio/progra1tpi.py
=== Simulación aleatoria con menú del día ===

¿Queda plata en la billetera?
Respuesta aleatoria: no

¿Queda algo de comida de anda saber cuando en esa heladera casi vacia?
Respuesta aleatoria: no

¿Hay comedor comunitario cerca?
Respuesta aleatoria: no

=== Decisiones tomadas ===
¿Queda plata en la billetera? -> no
¿Queda algo de comida de anda saber cuando en esa heladera casi vacia? -> no
¿Hay comedor comunitario cerca? -> no

Resultado final: ¡**DESASTRE TOTAL**: No hay comida, dinero ni ayuda visible.
Menu del día: hoy nos miramos las caras y nos vamos a dormir... gracias.
PS C:\Users\bruno>
```

```

120 def respuestas_aleatorias(pregunta):
121     print("\n" + pregunta)
122     print("Respuesta aleatoria:", "sí" if respuesta else "no")
123     return respuesta
124
125 def imprimir_historial(historial):
126     # muestra la secuencia de decisiones tomadas y el menú final determinado.
127     print("\n=== Decisiones tomadas ===")
128     resultado_final = ""
129     for paso in historial:
130         if paso[0] == "RESULTADO":
131             resultado_final = paso[1]
132         else:
133             print(f"{paso[0]} → {paso[1]}")
134
135     print(f"\nResultado final: {resultado_final}")
136     menu = obtener_menu(resultado_final)
137     print(f"Menu del día: hoy {menu}.")
138
139 def inicio():
140     # Principal. Ejecutar simulación
141     print("=== Simulacion aleatoria con menu del día ===")
142     historial = evaluar_arbol(manual_cesta_basica, respuestas_aleatorias)
143     imprimir_historial(historial)
144
145 inicio()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu del día: hoy sopa improvisada con una alita de pollo que sobro de hace dos días.

PS C:\Users\bruno> & C:/Users/bruno/AppData/Local/Programs/Python/Python312/python.exe c:/Users/bruno/OneDrive/Escritorio/progra1tpi.py

=== Simulacion aleatoria con menu del día ===

¿Queda plata en la billetera?

Respuesta aleatoria: sí

¿Hay alguna oferta o descuento en el super de la esquina?

Respuesta aleatoria: sí

¿podre comprar algo esencial (pan y/o leche y/o arroz)?

Respuesta aleatoria: no

=== Decisiones tomadas ===

¿Queda plata en la billetera? → sí

¿Hay alguna oferta o descuento en el super de la esquina? → sí

¿podre comprar algo esencial (pan y/o leche y/o arroz)? → no

Resultado final: El descuento no alcanza, ¡solo podemos comprar lo barato!

Menu del día: hoy lo que tengo no alcanza.. hoy: nada... por ahora..

PS C:\Users\bruno> |

```

120 def respuestas_aleatorias(pregunta):
123     print("\n" + pregunta)
124     print("Respuesta aleatoria:", "sí" if respuesta else "no")
125     return respuesta
126
127 def imprimir_historial(historial):
128     # muestra la secuencia de decisiones tomadas y el menú final determinado.
129     print("\n=== Decisiones tomadas ===")
130     resultado_final = ""
131     for paso in historial:
132         if paso[0] == "RESULTADO":
133             resultado_final = paso[1]
134         else:
135             print(f"{paso[0]} → {paso[1]}")
136
137     print(f"\nResultado final: {resultado_final}")
138     menu = obtener_menu(resultado_final)
139     print(f"Menu del día: hoy {menu}.")
140
141
142 def inicio():
143     # Principal. Ejecutar simulación
144     print("=== Simulación aleatoria con menú del día ===")
145     historial = evaluar_arbol(manual_cesta_basica, respuestas_aleatorias)
146     imprimir_historial(historial)
147
148
149 inicio()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

¿Queda plata en la billetera?
Respuesta aleatoria: no

¿Queda algo de comida de anda saber cuando en esa heladera casi vacia?
Respuesta aleatoria: sí

¿Se puede repartir entre todos aunque sea poco?
Respuesta aleatoria: sí

¿Algun vecino puede aportar algo a la olla?
Respuesta aleatoria: no

=== Decisiones tomadas ===
¿Queda plata en la billetera? → no
¿Queda algo de comida de anda saber cuando en esa heladera casi vacia? → sí
¿Se puede repartir entre todos aunque sea poco? → sí
¿Algun vecino puede aportar algo a la olla? → no

Resultado final: ¡Comemos poco, pero comemos juntos!
Menu del día: hoy pan con manteca compartido, un fiasco!.
PS C:\Users\bruno>

```

youtube <https://youtu.be/ZvxxTleui0Y>

github <https://github.com/manfredialdo/programacion1tpi>