ELEMENTOS DE MATEMÁTICA Y LÓGICA

ENTREGA FINAL ABP

Integrantes:
González Javier Alexis
Jiménez Mariel Emilse
Lazarte Mansicidor Alexis Saúl

“Mi Agenda de Tareas Hogareñas”

PROBLEMÁTICAS/NECESIDADES:
Distribución de tareas en el hogar para todos los miembros del hogar.

FUNDAMENTACIÓN:
Actualmente, el ritmo de vida implica que los adultos de los hogares deban salir a trabajar y a la vez gestionar las tareas hogareñas.

Tradicionalmente esta división genera una carga mental que se ha atribuido a las mujeres de los hogares, así mismo no son remuneradas y no se transparenta su cuantificación. De esta manera resulta difícil ponderarlas.

En este nuevo paradigma la asignación de las tareas se vuelve una situación que requiere revisión y transparencia para garantizar la equidad en las tareas de cuidado, entre otras.

Encontramos por ejemplo: hacer compras, sacar la basura, cocinar, llevar al médico a los niños, llevar y retirar de la escuela a los niños, comprar regalos de cumpleaños, llevar al veterinario a las mascotas, vacunar a los niños, etc.

Objetivo General:

Implementar una aplicación de tareas que permita la automatización y distribución equitativa de las mismas entre los distintos miembros del hogar.
Permitiendo así, por ejemplo, realizar una serie de automatizaciones de acuerdo a tareas repetitivas previamente establecidas en horarios y/o momentos del día específico.

Objetivos Específicos:

- *Desarrollar un software que permita mediante el uso de librerías del tiempo, definir franjas horarias, especificar tareas y asignarlas según las variables disponibles.*

- *Visibilizar las tareas realizadas en los hogares a partir de la cuantificación (lista) y la demanda real de cada usuario.*

- *Automatizar la distribución equitativa de las tareas agregadas de acuerdo a la cantidad de las mismas en función de la disponibilidad y cantidad de los miembros del hogar.*

OBJETIVO ESPECÍFICO	ACCIONES
Desarrollar un software que dé solución a la problemática	Mediante el lenguaje de programación Python, se desarrollará dicha aplicación a través del uso de la programación orientada a objetos (POO), la cual consta del uso de clases, métodos e instancias para dicho fin. Para ello se definen franjas horarias, especifican tareas y se asignan según las variables disponibles.
Visibilizar las tareas realizadas en los hogares a partir de la cuantificación (lista) y la demanda real de cada usuario.	1. Habilitar el ingreso del detalle de las tareas a los usuarios de la aplicación 2. Permitir visualizar el listado completo de las tareas disponibles, asignadas o a asignar 3. Generar un informe periódico de las tareas y usuarios, empleando herramientas gráficas
Automatizar la distribución equitativa de las tareas agregadas	4. Definir variables que se correspondan a cada miembro del grupo familiar; 5. Definir tareas listadas, franjas horarias y métodos a ejecutar; 6. Definir parámetros de asignación en base a rol del hogar, disponibilidad temporal y habilidades.

El desarrollo de esta aplicación ha cumplimentado dos objetivos planteados hacia el inicio de este proyecto, quedando pendiente la automatización de asignación de tareas a cada miembro de la familia que utilice el desarrollo.

Sin embargo nos motiva a seguir mejorando y trabajando sobre el mismo ya que implica adquirir mayores conocimientos y aplicarlos en una idea propia que estamos convencidos que hará la diferencia.

Vinculación con los conceptos propios del espacio curricular “Elementos de Matemática y Lógica”:

1. Teoría de Conjuntos

La Teoría de Conjuntos es el estudio de colecciones bien definidas de objetos, llamadas conjuntos, y las relaciones entre ellos.

Conjunto Universal (U)	La lista completa de todas las tareas, `self.tareas`, modela el Conjunto Universal (U) de la aplicación.
Subconjuntos y Complemento (U∖C)	Se generan dos subconjuntos a partir de U: C (Tareas Completadas) y P (Tareas Pendientes). P se calcula como el **Complemento** de C respecto a U (P=U∖C).
Cardinalidad	|**A**| Corresponde a la cantidad de elementos del Conjunto.

2. Lógica Proposicional

La Lógica Proposicional estudia las sentencias declarativas (proposiciones) que pueden ser verdaderas o falsas, y cómo se combinan usando conectivos lógicos.

Proposiciones	Dos atributos de la clase `Tarea` actúan como proposiciones: **P**: `self.completada` ("La tarea está completada"), y **Q**: `self.es_urgente` ("La tarea es urgente").
Conjunción (∧) y Negación (¬)	El filtro de prioridades (Opción 4) aplica la fórmula lógica de **Conjunción (AND)** y **Negación (NOT)**: **Q∧¬P**. Esto se traduce en el código como: `t.es_urgente and not t.completada` (Urgente **Y** NO completada).

3. Funciones y Lógica de Predicados

Este punto incluye la relación entre elementos y las estructuras formales para la búsqueda y demostración.

Funciones Inyectivas (Inyectividad)	Una función es inyectiva si a cada elemento distinto del dominio le corresponde un elemento distinto del codominio. El sistema de asignación de IDs garantiza la inyectividad: a cada tarea distinta se le asigna un ID único (`self.id`).

Cuantificador Existencial (∃)	El concepto corresponde a "Existe al menos un elemento que cumple una propiedad". En `marcar_completada`, el código busca si **existe** (∃) una tarea en el conjunto universal U tal que su ID coincida con el proporcionado (`if tarea.id == tarea_id:`).
Modus Ponens	Es una regla de inferencia lógica: Si A→B es verdadera y A es verdadera, entonces B debe ser verdadera. Se refleja en la ejecución de `marcar_completada`: Si (el ID coincide) **entonces** (ejecutar la acción de `tarea.completada = True`).

En cuanto a las funcionalidades por desarrollar, que corresponden a la Automatización de la asignación de las tareas a los miembros (usuarios disponibles) de acuerdo a su tiempo, habilidad, etc, delimitamos que los siguientes conceptos serían indispensables para pensar en el desarrollo:

I. Bases de Conteo

El problema de asignar tareas a personas con restricciones es fundamentalmente un problema de **Conteo**. Por lo que utilizamos el **Principio de Dirichlet (Principio del Palomar)**

> Si tenemos N tareas y M personas, y N>M, sabes que al menos una persona tendrá que recibir más de una tarea. Esto es útil para **verificar la equidad mínima** antes de la asignación.

II. Álgebra Lineal: Vectores y Matrices

El método más eficiente y riguroso para modelar y resolver un problema de asignación con múltiples variables y restricciones es mediante **Álgebra Lineal**, específicamente el concepto de **Matrices**.

Vectores	Cada persona y cada tarea se modelarían como un **Vector de Atributos**. **Vector Persona:** [Rol,Disponibilidad (h), HabilidadLavar, HabilidadComprar] **Vector Tarea:** [Tiempo requerido (h),Rol requerido,Complejidad]
Matrices	Se usaría una **Matriz de Adyacencia o Matriz de Costos** para representar todas las posibles asignaciones y sus "costos" (ej. el tiempo que le tomaría a esa persona, o un "costo" alto si no está capacitada).

La matriz Am×n tendría m (miembros) filas y n (tareas) columnas. El valor Aij sería el costo de asignar la tarea j al miembro i.

La asignación final de tareas a miembros corresponde a **Funciones Sobreyectivas (Sobreyectividad)**
Para garantizar que **todas** las tareas sean asignadas, la función de asignación f debe ser **sobreyectiva** (o al menos, la imagen de la función debe ser igual al conjunto de tareas). Esto asegura que no queden tareas sin dueño.

Anexo:

Para poder correr y visualizar correctamente la aplicación desarrollada, encontrará dentro de la carpeta a descomprimir, el archivo 'Instrucciones.txt' con el paso a paso para la instalación del lenguaje Python en su sistema operativo.

Bibliografía:

Instituto Superior Politécnico Córdoba (ISPC). (2025). *Unidad 1: Conjuntos*. [Apuntes de clase en línea]. Campus Virtual ISPC. https://acceso.ispc.edu.ar/course/view.php?id=3829

Instituto Superior Politécnico Córdoba (ISPC). (2025). *Unidad 2: Elementos de lógica*. [Apuntes de clase en línea]. Campus Virtual ISPC. https://acceso.ispc.edu.ar/course/view.php?id=3829

Instituto Superior Politécnico Córdoba (ISPC). (2025). *Unidad 3: Demostraciones*. [Apuntes de clase en línea]. Campus Virtual ISPC. https://acceso.ispc.edu.ar/course/view.php?id=3829

Instituto Superior Politécnico Córdoba (ISPC). (2025). *Unidad 4: Bases de conteo*. [Apuntes de clase en línea]. Campus Virtual ISPC. https://acceso.ispc.edu.ar/course/view.php?id=3829

Instituto Superior Politécnico Córdoba (ISPC). (2025). *Unidad 5: Funciones*. [Apuntes de clase en línea]. Campus Virtual ISPC. https://acceso.ispc.edu.ar/course/view.php?id=3829

Instituto Superior Politécnico Córdoba (ISPC). (2025). *Unidad 6: Fundamentos del Álgebra y el álgebra lineal*. [Apuntes de clase en línea]. Campus Virtual ISPC. https://acceso.ispc.edu.ar/course/view.php?id=3829