

# Anteproyecto: Sistema de Asignación de Mesas Priorizada para Recepcionista Robot

Jonatan Aguilar (20240565) - Desarrollador Principal Por Descarte

## 1. Propósito del Sistema

Cada grupo de clientes genera una "solicitud de asignación". El sistema toma esta solicitud y la coloca en una **lista de espera inteligente**. Cuando una mesa queda libre, el sistema consulta esta lista para encontrar al grupo de mayor prioridad y le asigna la mesa más adecuada según su tamaño, cumpliendo con las reglas del restaurante.

## 2. Características Principales (Funcionalidades de la Aplicación)

La aplicación proporcionará las siguientes acciones que un usuario (o el robot simulado) podrá realizar:

- **Registrar Llegada de Grupo:** Ingresar un nuevo grupo de clientes al sistema, especificando su nombre, si tienen reserva, su tipo y el número de personas.
- **Reportar Mesa Vacía:** Indicar al sistema que una mesa específica ha terminado de ser usada y está lista para ser reasignada.
- **Ver Lista de Espera:** Mostrar los grupos que actualmente esperan una mesa, ordenados según su prioridad actual.
- **Ver Estado de Mesas:** Consultar qué mesas están libres y cuáles están ocupadas por algún grupo.
- **Avanzar Tiempo:** Simular el paso de 5 minutos en el restaurante, lo cual puede afectar la prioridad de los grupos en espera.

## 3. Limitaciones y Reglas del Restaurante Simulado

El sistema operará dentro de las siguientes restricciones y configuraciones específicas:

- **Mesas:** Un total de 12 mesas con capacidades fijas:
  - 4 mesas para 6 personas (Mesas 1-4)
  - 6 mesas para 4 personas (Mesas 5-10)
  - 2 mesas para 2 personas (Mesas 11-12)
- **Tamaño de Grupos:** Solo se aceptan grupos de 1 a 6 personas.
- **Capacidad de la Lista de Espera:** La lista de espera tiene un límite de 15 grupos.
- **Asignación Flexible por Espera:** Si un grupo ha esperado 15 minutos o más, el sistema tiene flexibilidad para asignarle una mesa disponible que sea más grande de lo estrictamente necesario para su grupo, si no hay otra opción más adecuada disponible.

Se desarrollarán dos versiones de esta implementación:

- **Versión 1 (Estructuras de Datos):** Utilizará estructuras de datos de Python y objetos.
- **Versión 2 (Primitivas):** Estará restringida a utilizar únicamente variables primitivas (`int`, `float`, `str`, `bool`, `None`) y control de flujo (`if`, `while`, `for`). Esta versión manejará un número **fijo y muy limitado** de clientes en espera y simulará la lógica usando comparaciones y asignaciones directas a variables hardcodeadas para contrastar la complejidad sin abstracciones.