

# Segunda Evaluación Ayudantía POO

### Enunciado:

***“Un restaurant necesita un pequeño sistema en Python para modelar a los trabajadores. Se te pide implementar clases usando herencia, herencia múltiple y polimorfismo. ”***

#### Clases

1. Clase base Trabajador - Debe tener un atributo nombre. - Debe implementar un método tarea() que retorne un mensaje genérico indicando que realiza tareas generales.
2. Clase Chef que hereda de Trabajador - Debe agregar el atributo especialidad. - Debe sobrescribir tarea() indicando qué prepara el chef.
3. Clase Mesero que hereda de Trabajador - Debe agregar el atributo seccion. - Debe sobrescribir tarea() indicando qué sección atiende.
4. Clase AyudanteChefMesero (Herencia múltiple) - Debe heredar de Chef y Mesero. - Constructor recibe: nombre, especialidad, seccion. - Debe sobrescribir tarea() combinando ambas funciones.
5. Polimorfismo En cada instancia de los trabajadores debe llamar la funcion tarea() Demostrando que reciben un mismo método pero gracias a cada atributo propio se operan de forma diferente **(Pista)**

## Evaluación

### Clase Base Trabajador (5 pts)

| Nivel                | Descripción  | Pts |
|----------------------|--|-----|
| Logrado              | Clase correcta con atributo <code>nombre</code> y método <code>tarea()</code> funcional. | 5   |
| Medianamente logrado | Falta atributo o <code>tarea()</code> incompleto.  | 3   |
| No logrado           | Clase incorrecta o inexistente.  | 0   |

### 2. Clase Chef (7 pts)

| Nivel                | Descripción  | Pts |
|----------------------|--|-----|
| Logrado              | Hereda bien, agrega <code>especialidad</code> y sobrescribe <code>tarea()</code> . | 7   |
| Medianamente logrado | Hereda pero sobrescritura o atributo incompleto.                                   | 4   |
| No logrado           | No implementa la clase Chef.   | 0   |

### 3. Clase Mesero (7 pts)

| Nivel                | Descripción   | Pts |
|----------------------|---|-----|
| Logrado              | Hereda bien, agrega <b>seccion</b> y sobrescribe <b>tarea()</b> . | 7   |
| Medianamente logrado | Hereda pero con fallas en atributos o sobrescritura.              | 4   |
| No logrado           | No implementa la clase Mesero.                                    | 0   |

### 4. Herencia múltiple: AyudanteChefMesero (10 pts)

| Nivel                | Descripción   | Pts |
|----------------------|---|-----|
| Logrado              | Herencia múltiple correcta, constructor y <b>tarea()</b> combinada. | 10  |
| Medianamente logrado | Herencia funciona pero constructor o <b>tarea()</b> incompletos.    | 6   |
| No logrado           | Implementación incorrecta o inexistente.                            | 0   |

### 5. Polimorfismo: muestra la funcion tarea() en todos las instancias (5 pts)

| Nivel                | Descripción   | Pts |
|----------------------|---|-----|
| Logrado              | Muestra correctamente las funciones de todas las instancias | 5   |
| Medianamente logrado | Las logra llamar pero con algunos errores                   | 3   |
| No logrado           | No llama ninguna funcion tarea()                            | 0   |



## 6. Programa principal (6 pts)

| Nivel                | Descripción  | Pts |
|----------------------|--|-----|
| Logrado              | Crea Chef, Mesero y Ayudante; usa <code>mostrar_tarea()</code> . | 6   |
| Medianamente logrado | Crea objetos pero no los prueba bien.                            | 3   |
| No logrado           | No ejecuta o faltan objetos.                                     | 0   |

## 7. Estilo, legibilidad y orden (5 pts)

| Nivel                | Descripción   | Pts |
|----------------------|---|-----|
| Logrado              | Buenas prácticas, indentación correcta, nombres claros. | 5   |
| Medianamente logrado | Detalles menores de estilo/orden.                       | 3   |
| No logrado           | Código desordenado o difícil de leer.                   | 0   |

**TOTAL : 45 PTS.**