



Universidad Tecnológica Centroamericana

UNITEC

Laboratorio de Programación III

Ana Gómez

Laboratorio 5



Objetivos

- Utilizar herencia, polimorfismo, casteo y archivos correctamente.
- Hacer uso de funciones en C++.

Desarrollo de la práctica

Para el desarrollo satisfactorio de los ejercicios, siga las instrucciones que a continuación se le presentan, después de cada ejercicio encontrará preguntas que deberá desarrollar en base al ejercicios elaborado.

Ejercicio

¡En hora buena, programador! Usted es ahora el feliz copropietario de un nuevo restaurante. Su socio, Gelbert Olivo, se encargará de la parte administrativa, pero para eso debe de crear un sistema de usuarios primero.

Este sistema será capaz de dejar un registro de todo usuario del restaurante en un archivo ya sea de texto o binario. No es necesario hacer un método de carga del archivo, pero al momento de escribir, debe de estar en el formato correcto para hacer lectura, con los tipos adecuados para los usuarios. *Hint: para guardar correctamente el tipo de cada usuario, deberá utilizar casteo.*

El **usuario** será nuestra clase padre. Cada usuario debe tener su username y password. Adicionalmente, se debe de conocer su nombre, su edad, su ID y su número de teléfono. No puede haber usuarios registrados de menor edad a 18, y las identidades deben de ser únicas. El número de teléfono debe tener al menos 8 dígitos.

Del usuario se desprenderán dos clases hijas: cliente y personal.

Del **cliente** se conoce su dirección y un número entre 1 y 5 que será el rating que tendrán de su restaurante. Debe existir un método que promedie los ratings de todos los clientes, y ese será el número de estrellas que tendrá su restaurante.

Del **personal** se conocerá el año de contratación (este año debe de tener sentido), y el sueldo. Esta clase, a su vez, será padre de:

- **Administrador:** se conoce la cantidad de empleados contratados y empleados despedidos. El administrador tendrá acceso a un set de métodos (a continuación) que deben estar en su clase main.
Métodos (deben estar en el main, no pueden ser void):
 1. Contratar empleado (agregar)
 2. Despedir empleado (eliminar)
 3. Aumentar/Decrementar de acuerdo a la inflación (donde la inflación será un porcentaje ingresado por el usuario)
 4. Conseguir al empleado con menor sueldo. Si hay más de un empleado con un sueldo mínimo igual, imprima la lista completa de estos empleados.
 5. Conseguir al empleado con mayor sueldo. Si hay más de un empleado con un sueldo máximo igual, imprima la lista completa de estos empleados.
 6. Conseguir el promedio de sueldo de todos los empleados.
- **Chef:** se conoce el platillo favorito (string) y el rating del chef (es distinto al rating del restaurante). Cuando el chef está en el login, tiene acceso a dos operaciones:
 1. Gritarle a un lavaplatos: decrementa la motivación de un lavaplatos por un número ingresado por el usuario.
 2. Agradar a un lavaplatos: aumenta la motivación de un lavaplatos por un número ingresado por el usuario.
- **Lavaplatos:** se conoce de él su nivel de motivación (double o int), inicialmente debería de encontrarse en 50. El lavaplatos tiene dos métodos disponibles:
 1. Renunciar. Si la motivación es menor o igual a 25, puede renunciar.
 2. Pedir aumento. Si la motivación es mayor o igual a 100, se le dará el aumento cada vez que la pido (por un número no mayor al saldo actual del lavaplatos).
- **Meseros:** lista de platillos que debe de entregar (**no es necesario guardar esta lista en archivos**), y como métodos (propios de la clase mesero), puede entregar un platillo en específico o todos los platillos.

Ponderación

| Elemento | Puntaje |
|----------------------------------|---------|
| Makefile | 1 |
| Parte de Investigación: Archivos | 3 |
| Métodos | 3 |
| Validaciones | 1 |
| Clases | 2 |

Penalizaciones por:

1. Quiebres en el programa como floating point exception, segmentation fault, what() bad alloc, entre otros.
2. Programa no amigable.
3. Código mal tabulado.
4. Variables no significativas.
5. No hay paridad de línea de código.

Especificaciones de entrega

Deberá enviar los datos del alumno y el repositorio de GitHub.

El repositorio debe llamarse:

Nombre-Apellido-Lab#