

Tarea de entrega Tema 4

Programación de una consigna a través del uso de clases



Por: Veronica González Bravo

Indice

1-Enunciado pag 1

2-Esquema y implementación pag 2

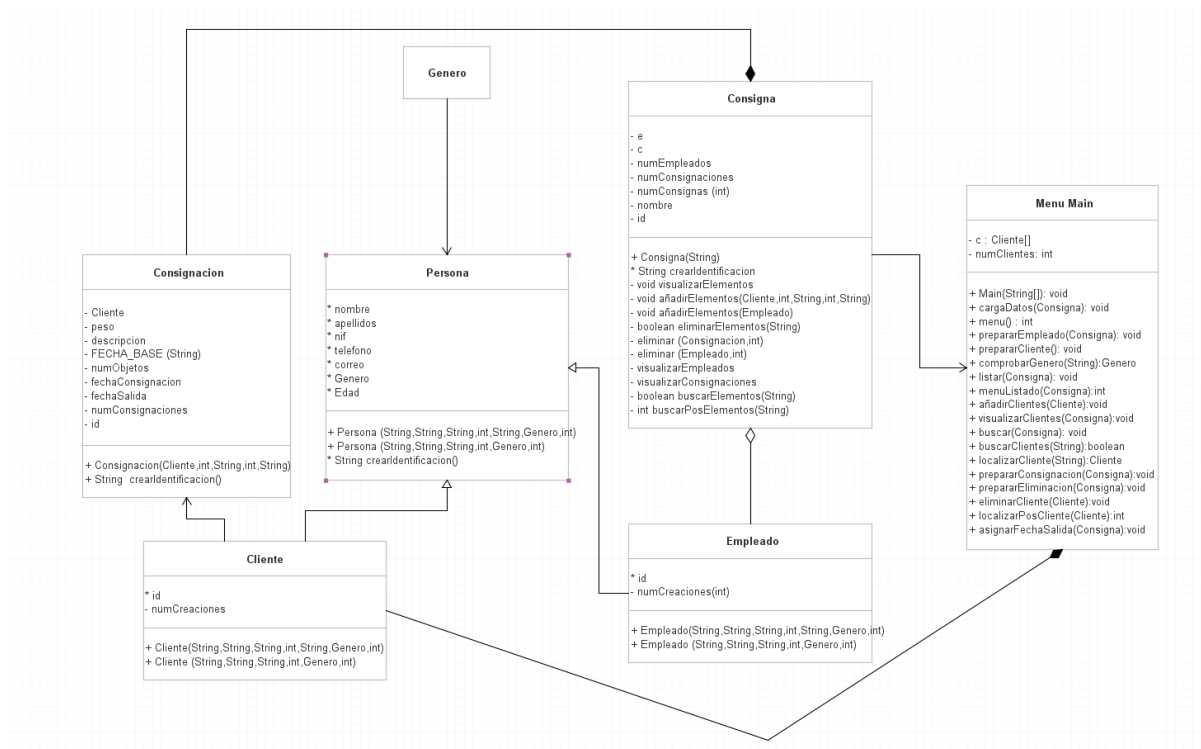
3-Justificacion de diseño pag 3-4

1 Enunciado del problema

Se necesita de la creación de una base de datos de pequeño calibre para una consigna en una cadena de hoteles, para ello se le asigna a un programador la tarea, las siguientes cuestiones que le pone el empleador sobre la mesa son las siguientes:

- La consigna llevará un registro los objetos ingresados con la fecha de ingreso y de extracción una vez efectuada.
- Deberá también guardarse los datos del cliente (Cliente previamente registrado) y asociarlos a la consignación que ha realizado.
- Cada consigna tendrá uno o varios empleados enlazados con las instalaciones, podrán ir dejando o ingresando en las instalaciones a lo largo del día, sin embargo, no es necesario indicar la hora en la que han estado trabajando. (se representara mediante la eliminación del elemento en la lista)
- A pesar de que tanto los empleados como los clientes tienen los mismos datos los clientes estarán enlazados con la consignación y los empleados con la Consigna en forma de agregación.

2 Esquema de implementación



3 Justificación de diseño

- Clase Consignación: Estará conformada por la id de consignación, el peso de los objetos consignados, el numero de los mismos, una descripción y tendrá una fecha de entrada y otra de salida, esta ultima tendrá el dato por defecto de “aun consignado” hasta que se actualize este atributo, ya que damos por hecho que una vez se crea una consignación esta no puede tener fecha de salida.
Esta fecha se actualizara una vez el cliente saque el objeto y la consignación quedara registrada dentro del vector de consignaciones de Consigna.
- Persona: Poseerá los datos básicos que cualquier humano tendrá (Nombre, apellidos, teléfono, correo (por defecto pondrá “ninguno”), id. Esta será abstracta y será el padre de empleado y cliente que se diferenciarán por sus respectivas ids.
- Consigna: Tendrá una composición de consignaciones y agregaciones de empleados ya que los empleados podrán trabajar en otras consignas, las gestiones de eliminación, actualización y lecturas estarán implementadas en la clase Main.
- Empleado: Tendrá una id propia para la realización de las búsquedas y eliminaciones, por el resto heredara todos los atributos de persona. Empleado como existe por si mismo y podra trabajar en otras consignas sera una agregación de Consigna.
- Cliente: Al igual que empleado tendrá su propia id, y estará mencionado dentro de cada asignación, sera elemento de composición dentro del menu main simulando el registro de clientes desde la red de la Empresa de consignas ya que un cliente existe siempre por si mismo y puede ser cliente de varias consignas. Debido a la naturaleza de las pruebas sera una composición dentro de Main, pero en situaciones normales seria una agregación.

- **PruebaMain:** El menu de pruebas main presentara un método llamado cargaDatos() que cargara los datos iniciales dentro de la consigna, presentara un conjunto de opciones para el usuario donde podra, crear,visualizar,eliminar y actualizar datos de las diferentes clases (en este caso solo la fecha de salida)

Entre las funciones básicas estan:

- ☐ **Listar:** Tendrá un submenu donde escoger si queremos ver los empleados, los clientes, las consignaciones, o, en su defecto, todos los datos
 - ☐ **Crear:** Permite insertar nuevos objetos. Clientes,Empleados,etc...
 - ☐ **Borrar:** Permite eliminar un objeto asociado a su id pertinente
 - ☐ **Actualizar Fecha:** Actualizara la fecha de salida, si ya hay una fecha no dejara.
 - ☐ **Buscar:** Buscara un objeto introduciendo previamente su id.
- **IDS:** Las ids de cada objeto se declararán mediante statics ,para definir las ids de los clientes y empleados, el número asociado a la id saldrá del aumento de esta static con cada uso del constructor persona o objeto.
La id se formulara de la siguiente manera:
 - ☐ **Empleado:** Empezará por la letra E seguida del número que marca la Static, aunque se borre un empleado en la consigna seguirá conservando su id ya que se sobrentiende que podra trabajar en otra consigna de la empresa..
 - ☐ **Cliente:** Empezará por la letra C seguido del número que marca su Static.
 - ☐ **Consignación:** Empezara con la letra O seguido del número que marcara su static, si esta se borra la id no podrá ser asignada a otro objeto.

- **Declaración de statics y constantes**

- ☐ **FechaSalida:** Tendrá una constante FECHA_BASE que se usara en el constructor para que escriba “Aun consignado” por defecto
- ☐ **Static numCreaciones:** Sirve para dar una id numérica a empleado y cliente, aunque se elimine una entrada esta siempre ira increchendo, asegurando que no se repita la id.
- ☐ **Static c[] y Static numClientes:** se declararan como statics para que todas las funciones puedan acceder a estos atributos.