



Tecnológico de Monterrey

Tecnológico de Monterrey

Campus Estado de México

“Programación Orientada a Objetos”

Alan Josué Melgar Fuentes A01752228

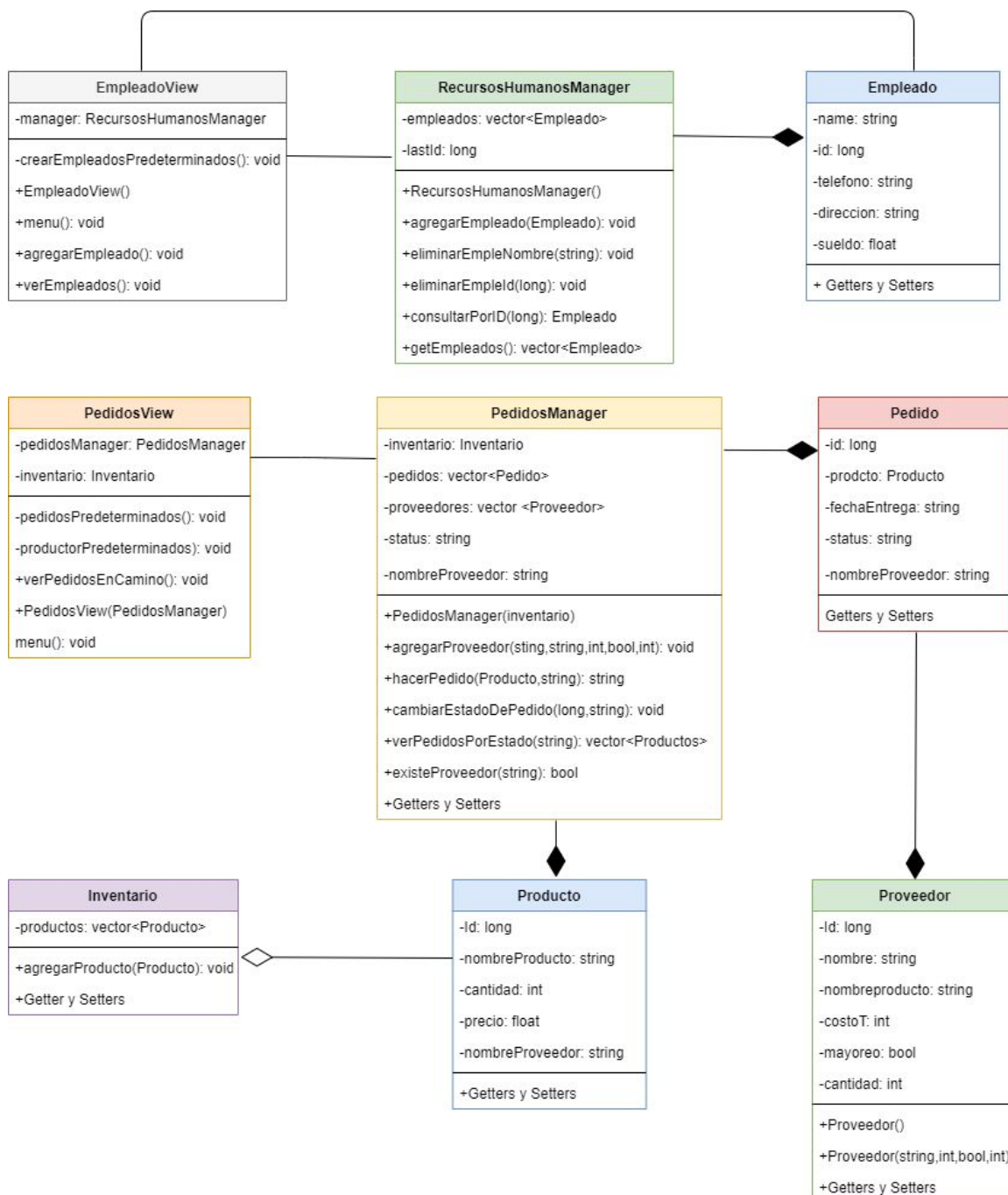
Fernando Emilio Nava Cano A01745597

Explicación:

Este proyecto tiene como objetivo crear una base de datos que esté manejada por el manager o el jefe de la empresa. En ella el usuario podrá administrar sus empleados, así como también el inventario y los pedidos a proveedores. Esta aplicación se ajustará manteniendo un orden entre todos sus elementos, para que el usuario de este programa no tenga problemas de algún tipo al usarla y sea de fácil acceso a nuevos usuarios. Carece de movimientos confusos y un menú en el que es fácil de navegar.

Liga de Github:

https://github.com/A01752228/ProyectoFinal_POO



El proyecto sirve para tener el control de los empleados de una PYME donde una persona en específico llevará el control de los empleados así como su podrá ingresar un nuevo empleado, borrar un empleado ya sea por nombre o por id, también podrá consultar a los empleados de la empresa, cuál es su salario y sus datos personales.

En el segundo apartado del proyecto algún empleado podrá solicitar pedidos que tenga con sus respectivos proveedores, podrá ver el estatus de los pedidos en camino y cuando le llegue un paquete podrá cambiar el estatus del pedido por id y de esta manera ingresarlo en el inventario de productos, también va a poder ver el inventario de los productos, la cantidad de productos y el precio de los productos y por último podrá hacer nuevos pedidos a los proveedores con los que tiene algun contrato y si es ingresado un proveedor desconocido el programa te arrojará error.

Clases:

Las clases se hicieron con el objetivo de repartir las tareas a cada una de ellas.

Empleado

- Es donde se declararon las variables (id,nombre,telefono, direccion, sueldo) y se generan los getters y setters.

Recursos Humanos Manager

- En esta clase se hacen los procesos eliminar un empleado y consultar un empleado.
 - **agregarEmpleado(Empleado)** = Agrega a un objeto de la clase empleado a un vector que existe en esta clase para poder acceder a sus datos en cualquier momento.
 - **eliminarEmpleNombre(string nombre)** = Recibe un Nombre y checa en la base de datos (vector de empleados) si existe un empleado con ese nombre y lo elimina.
 - **eliminarEmpleId(long Id)** = Recibe un Id y como el método anterior busca el empleado y lo elimina del vector.

Empleado View

- En esta clase se agrega un empleado, también puedes ver los empleados que ya están en la empresa y también puedes visualizar las opciones que tienes para realizar una acción dentro de la clase Recursos Humanos Manager.
 - **agregarEmpleado():** En este método puedes agregar un empleado nuevo dentro de tu empresa así como ingresar sus datos personales y el sueldo.
 - **verEmpleados():** En esta clase vas a ver los empleados de la empresa, así como sus datos personales y el sueldo.
 - **menu():** Vas a visualizar el menú de opciones que puedes realizar dentro de esta clase.

Pedido

- Aquí se declara el nombre del producto, la fecha de entrega, el estatus del pedido y el nombre del proveedor que te va a entregar el pedido, también se generan los getters y setters de cada una de estas variables.

Pedidos Manager

- En esta clase existen todos los métodos con los que se construirán la base de datos. Agregar pedidos, agregar proveedores, cambiar los estados de los pedidos, así como

eliminar los pedidos, son las cosas que esta clase se encarga de hacer. Todos los objetos de proveedores e inventario se agregan a esta clase.

- **hacerPedido(Producto, string fecha)** = Este método te permite agregar un pedido a la vez ingresando el producto que vas a pedir y la fecha en que llegará. Y lo agrega a un vector de los pedidos.
- **cambiarEstadoDePedido(long Id)** = Ingresando el Id del pedido que quieres editar, este método te permitirá cambiar el estado del pedido con ese Id a “En camino” o a “Recibido”
- **verPedidosPorEstado(string estado)** = Este método te permite ver la lista de pedidos por estado, elige ente “En camino” o “Recibido”

Pedidos View

- En esta clase se muestran las opciones que vas a visualizar en la consola, al igual que vas a poder ver los pedidos que están en camino y los productos que ya están en el inventario, también vas a ver el menú que dependiendo de la opción seleccionada se realiza una acción en específico.
 - **verPedidosEnCamino():** En este método se pueden visualizar los pedidos en camino, que proveedor lo trae, la cantidad de producto que trae , el nombre del producto y el precio que va a costar el pedido.
 - **pedidosPredeterminados():** Aquí se declaran pedidos para hacer la simulación de los pedidos en camino, se le agrega, nombre, precio, cantidad y proveedor.
 - **menu():** Aquí puedes visualizar el menú de opciones a realizar dentro de esta clase.

Proveedor

- En esta clase se declaran los datos de un proveedor, esto incluye; Su nombre, que es lo que proveen, su cargo por el transporte, si se vende en mayoreo o no, y su cantidad.

Inventario

- Aquí se declara un vector de los productos recibidos y se agregan al inventario si es un producto existente se acumula la cantidad y si no lo es se le crea su propio espacio en el inventario.

Producto

- En esta clase se declara el nombre del producto, la cantidad, el precio del producto y el nombre del producto además de la declaración de los getters y setters.

Pruebas

```
Bienvenido al menu de empleados
Opciones:
1) Agregar empleado:
2) Ver empleados:
3) Eliminar empleado por nombre:
4) Eliminar empleado por ID:
Ingresa no para salir
Ingresa una opcion
1
Ingresa el nombre de un empleado:Jorge Uresti
Ingresa el telefono de un empleado:5567876987
Ingresa la direccion de un empleado:Tec
Ingresa la su
eldo de un empleado:5000
Deseas agregar otro empleado: (y/n):n
```

```
1) Ver pedidos en camino
2) Cambiar estatus de pedido por ID
3) Hacer Pedido:
4) Ver inventario:
5) Ver Proveedores:
Ingresa no para salir
1
[[Pedidos en camino]]
ID: 1    Nombre del producto: Shampoo    Cantidad: 1000    Proveedor: Pantene
ID: 2    Nombre del producto: Jabon      Cantidad: 3000    Proveedor: Dove
ID: 3    Nombre del producto: Taladros   Cantidad: 200     Proveedor: Black&Decker
ID: 4    Nombre del producto: Laptop     Cantidad: 15      Proveedor: HP
ID: 5    Nombre del producto: Playeras   Cantidad: 70      Proveedor: Levis
```

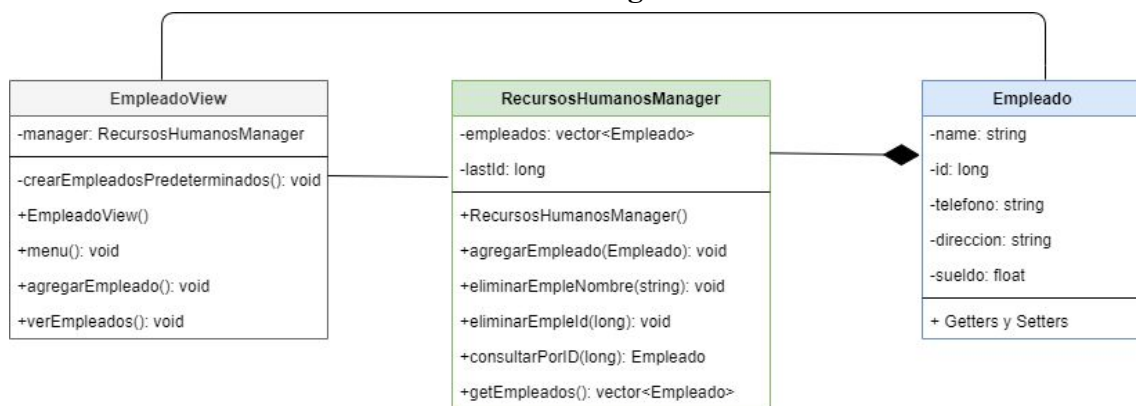
```

Que quiere visualizar
1) Ver pedidos en camino
2) Cambiar estatus de pedido por ID
3) Hacer Pedido:
4) Ver inventario:
5) Ver Proveedores:
Ingresa no para salir
5
-----
Nombre: Pantene
Calidad: 9
Producto que provee: Shampoo
-----

-----
Nombre: Dove
Calidad: 8
Producto que provee: Jabon
-----

```

Alan Josué Melgar Fuentes:



Hice las clases Empleados View y Recursos Humanos Manager, en estas clases doy de alta un empleado, eliminarlo un empleado ya sea por nombre o por Id, también puedes visualizar los empleados que están en la empresa . En los empleados puede consultar el sueldo del empleado, su id, la dirección y el teléfono.

Mi parte individual del proyecto lleva de manera estructurada que empleados hay en tu empresa, da de alta un empleado y puedes eliminarlo de esta lista en caso de despido.

agregarEmpleado(): En esta a poder dar de alta un empleado solicitando su nombre, telefono, direccion.

verEmpleados(): En esta metodo vas a poder visualizar los empleados que ya están en la empresa y si agregar algún empleado se actualizará y también lo podrá visualizar.

eliminarEmpleadoPorNombre(string): En esta metodo vas a poder eliminar un empleado de la todos los empleados que cuenta tu empresa por su nombre.

<pre>Ingresa el nombre de un empleado:Alan Josué Ingresa el telefono de un empleado:5568798564 Ingresa la direccion de un empleado:Mexico Ingresa la su eldo de un empleado:1502 Deseas agregar otro empleado: (y/n):n</pre>	<pre>----- ID: 6 Nombre: Alan Josué Telefono: 5568798564 Direccion: Mexico Sueldo: 1502</pre>
--	---

```
-----
ID: 6
Nombre: Alan Josué
Telefono: 5568798564
Direccion: Mexico
Sueldo: 1502
Ingresa el nombre del empleado que quieres borrar:Alan Josué
```

Aprendí a estructurar un proyecto para una PYME y que no es fácil llevar el control de un sistemas para el control de productos y empleados.

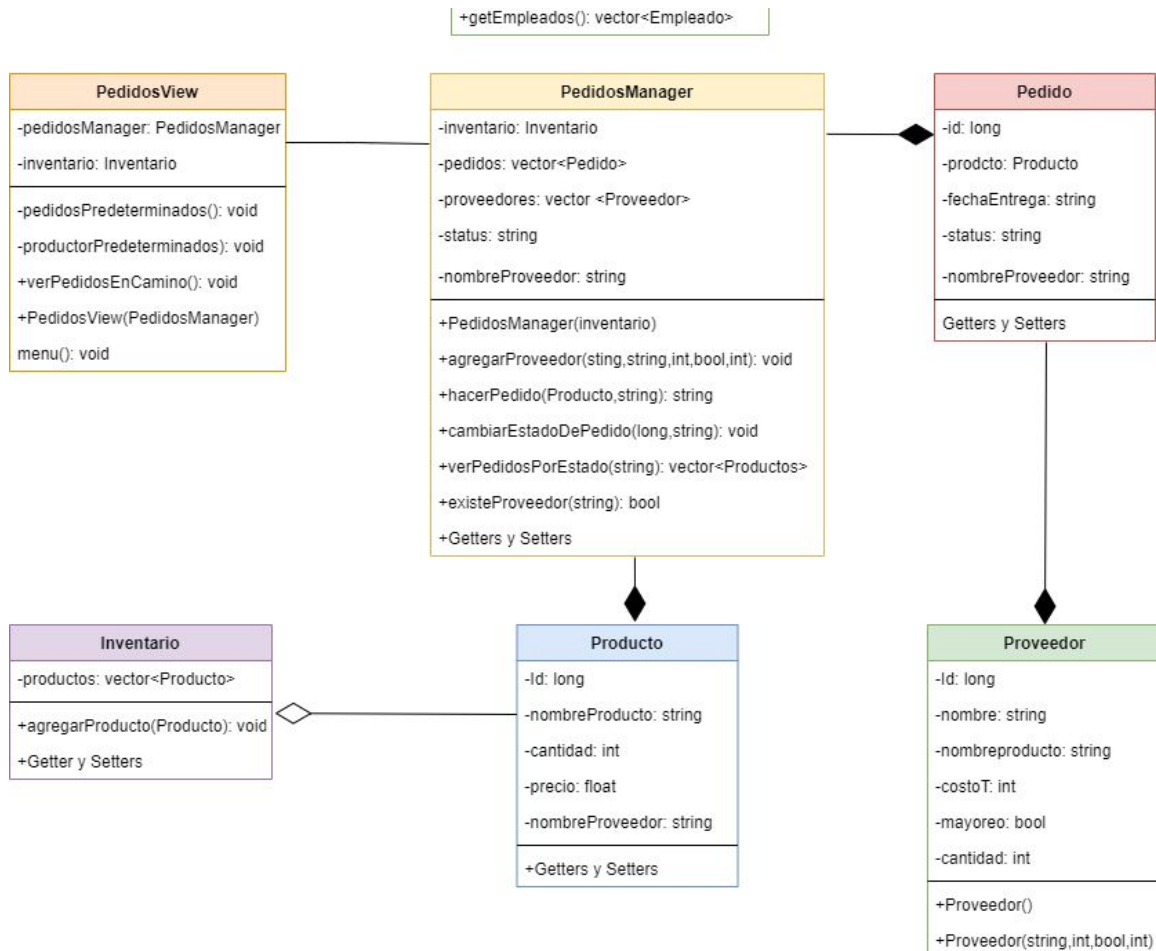
La forma en cómo me comunicaba con mi equipo de trabajo y como plasmé mis ideas en el proyecto.

Lo que más me gustó fue trabajar en equipo con mi compañero Fernando, ya que si trabajo y esto hizo que el proyecto fuera mucho más fácil y rápido.

Lo que menos me gustó fue el tiempo que tuvimos después de la explicación de las clases y cómo se estructuraba, también que no contábamos con tantos conocimientos para hacer un mejor proyecto.

Fernando Emilio Nava Cano

En el proyecto, la clase de proveedores fue mi parte de la programación individual, después de crearla y darle sus variables lo integre al trabajo en equipo en los pedidos. Esto incluyó crear nuevos métodos para la clase “Pedidos Manager”, los métodos que agregué te permiten agregar un proveedor al vector de proveedores, ver todos los proveedores y saber si un proveedor existe en el vector. Estos métodos los llame en el método “menu()” de la clase “Pedidos View”. El propósito de lo que agregué es darle opción al usuario de administrar más datos de sus pedidos.



existeProveedor(string nombre) : Este método te dice si el proveedor con el nombre que introdujiste existe en el vector de Proveedores.

agregarProveedor(string nombre, string nombreproducto, int costosT, bool mayoreo, int calidad) : Este método recibe todos los datos necesarios para agregar un objeto de proveedor y lo agrega al vector.

verPedidosPorEstado(string) : Te permite ver los pedidos que tienen el estado que recibe como parámetro.

En este proyecto aprendí a estructurar una base de datos con programación orientada objetos en c++, así como a hacer vectores y manipularlos. También aprendí a no subestimar cualquier cosa, no importa que tan simple se vea.

En mi opinión hay algunas cosas que se pueden mejorar con mis métodos, pude haber agregado uno más que permitiera un poco más de manipulación o algo similar. Aunque yo termino satisfecho con lo que hice.

Mi parte favorita de este proyecto, sinceramente fue aprender todo lo que vimos y hacer mi propia clase y manipular dicha clase con objetos en un vector. Aunque también me agradó el trabajo en equipo y la ayuda de mi compañero.

Lo que no me gustó de este proyecto fue cuando 1 línea de código hizo que el programa terminara sin ninguna razón aparente, eventualmente se arregló, pero fue muy estresante ver que el código terminaba y no marcaba ninguna clase de error.