***“Año del Fortalecimiento de la Soberanía Nacional”***

****

**Profesor:**

Ricardo Martin Dulanto Ramirez

**Curso:**

Programación Orientada a Objetos II

**Integrantes:**

Ayala Segundo, Juan Carlos

Fonseca Vicente, Sebastian Javier

Paz Alejos, Dayana Sophia

**2022-II**

**Tabla de contenidos**

* Introducción
* Objetivos
* Descripción del problema
  + Requerimientos de la propuesta
  + Actores
* Diseño de la solución
  + Proceso Seleccionado (UML)
  + Diagrama de clases
  + Solución propuesta
* Desarrollo de la solución
  + Paradigma POO
  + Empleo de interfaz gráfica
  + Manejo de listas
  + Empleo de patrón de diseño
* Conclusiones
* Referencias

**Introducción-.**

No es una sorpresa que el delivery se ha vuelto una parte importante en la vida diaria en gran parte de la población, esto es debido a los múltiples beneficios que nos ofrece como lo es la rapidez de hacer un pedido, el tiempo ahorrado o su gran variedad de métodos de pago.

En los últimos años se ha hecho más común el delivery mediante aplicaciones que por llamadas telefónicas, esto ha producido una gran creación de app tanto como web como para móviles (android, IOs). Nosotros como grupo nos enfocaremos en la creación de una interfaz para una hamburguesería.

La hamburguesería en la que nos estamos enfocando se llama JSM, la cual nos contrató para automatizar su empresa y le podamos ayudar a que sus pedidos estén de una manera más ordenada.

**Objetivos -.**

Nuestro trabajo tiene diferentes tipos de objetivos:

* Objetivos Primarios:
  + La interfaz tiene que ser intuitiva y fácil para el comprador.
  + Se tiene que tener todos los implementos necesarios para evitar futuros problemas en la compra.
  + Se tiene que implementar distintos métodos de pago para aumentar nuestros pedidos gracias a la flexibilidad que tiene nuestro local.
  + Por último, nuestro proyecto tiene que ser seguro tanto para los clientes como para los trabajadores, así ambos puedan hacer y recibir un pedido de manera exitosa.
* Objetivos Secundarios:
  + La interfaz debe tener un efecto impactante frente a los ojos de los clientes.
  + Se debe evitar que ocurra errores mientras la aplicación está en curso, por lo que se tiene que ver cualquier alternativa y corregirla.
  + El cliente puede dar su opinión respecto a su experiencia pidiendo en JSM.

**Descripción del problema -.**

**Requerimiento de la propuesta:** Hemos encontrado que, a pesar del incremento del uso del servicio de delivery, no se han tomado las medidas necesarias para asegurar la seguridad tanto de la empresa como la del comprador.

Tenemos el lado de la empresa. Existen casos donde restaurantes o tiendas han recibido compras hasta de cientos de soles y que terminan siendo un fraude por no tener un sistema de logueo. Esto llega a producir pérdidas exageradas tanto en dinero como en utilidades, esto se ve peor dentro del sector de comidas ya que estas no se pueden devolver después de ser cocinadas.

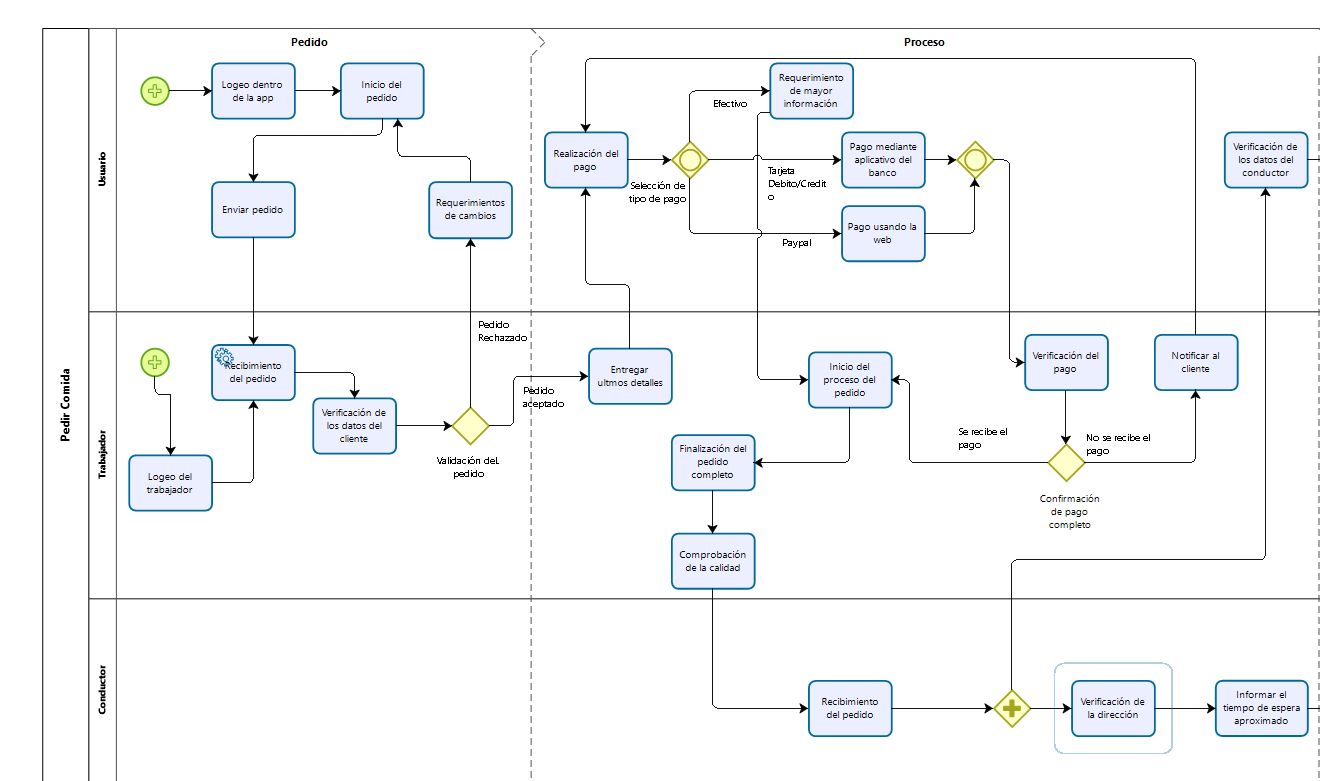
Cuanto a la seguridad del cliente, actualmente no se tiene una información más completa de quién es el driver de su producto. Durante estos últimos años (2020-2022) han ocurrido un montón de casos de robo gracias a no saber ni siquiera el nombre del conductor. Tenemos el caso de un carnicero donde denuncia haber perdido aproximadamente unos 800 soles en mercadería. Ocurrió cuando el repartidor fue a entregar un paquete y momentos después, mientras el carnicero no estaba observando, se llevó la carne valorizada en 800 soles. Fue gracias a las cámaras de seguridad que se pudo identificar al conductor, ya que la aplicación no daba información alguna.

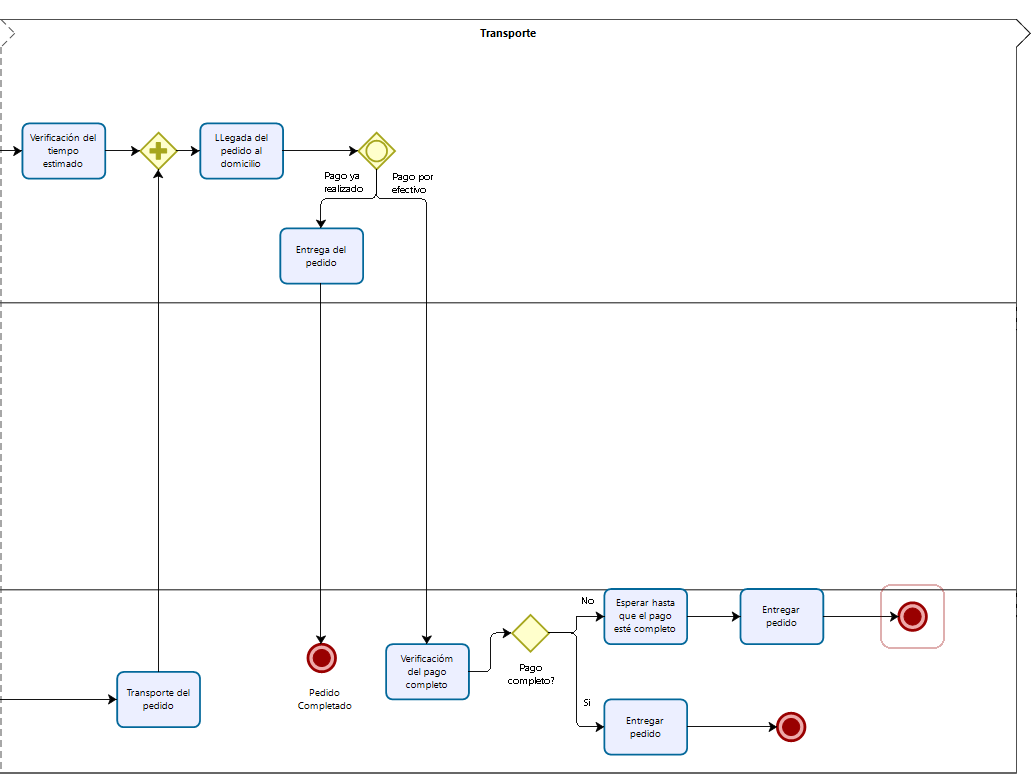
**Actores:**

Los actores que consideramos en este proyecto son:

* Trabajador: En el caso del trabajador, estos son los que reciben los pedidos y se encargan de enviar un motorizado hacia el domicilio del usuario para que se pueda realizar un envío satisfactoriamente.
* Usuario: En el caso del usuario, este se encargará de usar nuestro proyecto, en el cual ellos deciden el pedido, extras, etc. Digita su dirección y algún especial que deseen obtener. Por último, este recibirá su pedido y el usuario tendrá una experiencia satisfactoria.

Un actor no puede funcionar sin el otro, ya que ambos se necesitan para hacer y recibir una orden, se beneficia el trabajador, el usuario y sobre todo la empresa.





Link del diagrama de procesos: <https://drive.google.com/file/d/1OH894ZEN4fchuuxP288gYSics0Ptnn6o/view?usp=sharing>

**Diseño de la solución -.**

Proceso de seleccionado(UML):

**CLIENTE:**

-Dni

-Nombre

-Apellido

-Fecha\_de\_Nacimiento

-Sexo

-Email

-Dirección

-Distrito

-Provincias

-País

**TRABAJADOR:**

-Dni

-Nombre

-Apellido

-Fecha\_de\_Nacimiento

-Sexo

-Email

-Dirección

-Distrito

-Provincias

-País

-Sueldo

**USUARIO:**

-Id

-Nickname

-Password

**PRODUCTO:**

-Id\_producto-Clave

-NombreProducto

-Precio

**PAGO:**

-Codigo-Clave

-Situación\_Entrega

**FACTURA:**

-NumeroFactura-Clave

-Fecha

-NumeroPedido

-PrecioUnitario

-Productos-Multivalorado

-Importe

-ImporteTotal

-NombreCliente-Multivalorado

**BOLETA:**

-Nombre\_Cliente

-Cantidad-Multivalorado

-Productos-Multivalorado

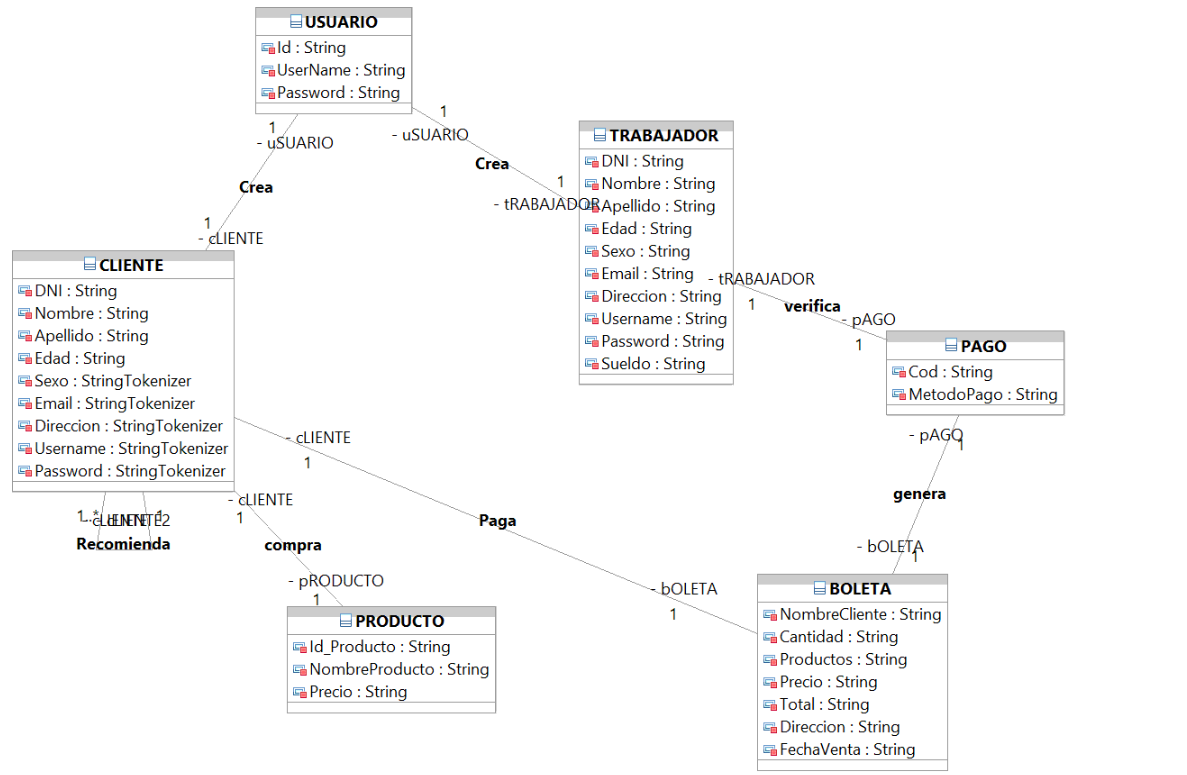
-Precio

-Total

-Dirección-Derivado

-FechaVenta

Diagrama de clases:



Solución de propuesta:

Como ya explicamos, las aplicaciones de delivery aún tienen una gran brecha de inseguridad tanto para el cliente como la empresa. Por lo que tenemos dos soluciones que estarán implementadas en nuestra interfaz.

En primer lugar, la interfaz tiene un login donde se tienen que registrar cada comprador y trabajador para poder hacer uso de esta. Para registrarse se tiene que poner información necesaria como un dni para asegurar que, quien esté creando la cuenta piensa usarlo para correctamente y no para hacer bromas o fraudes. Con esto evitaremos futuros problemas y se podrá saber de cual cuenta es que se hacen los pedidos.

Por otro lado, para tener un envío más seguro pondremos información completa del conductor, a cual le corresponda el envío de la comida. Con esto se podrá saber quién es la persona que está realizando el envío y evitar futuros problemas.

**Desarrollo de la solución.-**

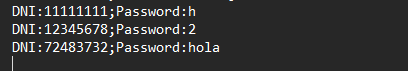
* Paradigma POO
* Empleo de interfaz gráfica

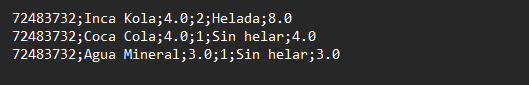






* Manejo de listas





* Empleo de patrón de diseño

El patrón de diseño utilizado fue netamente con líneas de código, importando java swing, para que pueda verse de una manera presentable y tenga una función exitosa al momento de su uso.

**Conclusiones**

Gracias a nuestro proyecto, se pudo entender mejor la lógica de la programación orientada a objetos. Se puede concluir que, por medio de distintas funcionalidades se puede llegar a realizar algo complejo que puede facilitar la vida diaria de una persona.

**Referencias**

<https://www.youtube.com/watch?v=NAjfZgILjuk>