### “UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO”

**FACULTAD DE CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



**GRUPO A**

**PROYECTO**

**SISTEMA DE INFORMACIÓN PARA GESTIONAR LA COMPRA, VENTA E INVENTARIO DE LA ROSTICERÍA POLLOS “EL BUEN GUSTO”**

**MATERIA:** SISTEMAS DE INFORMACIÓN 1

**SIGLA:** INF342

**GRUPO:** SA

**DOCENTE:** ING. GARZON CUELLAR ANGELICA

**INTEGRANTES:**

* Huarachi Laime Raúl 217169392
* Llanos Duran Junior Javier 216083001
* Maldonado Gutierrez Daniel 217027891
* Mamani Soliz Christian Celso 218030940
* Zeballos Carvallo Oscar Daniel 218060556

**GESTIÓN:** 1/2021

**SANTA CRUZ – BOLIVIA**

**Sistema de información para gestionar la compra y venta de la rosticería restaurante pollos “el buen gusto”**

Contenido

[“UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO” 1](#_Toc74895560)

[1. PERFIL 1](#_Toc74895561)

[1.1. INTRODUCCIÓN 1](#_Toc74895562)

[1.2. ANTECEDENTE 2](#_Toc74895563)

[1.3. JUSTIFICACIÓN 2](#_Toc74895564)

[1.4. DESCRIPCIÓN DEL PROBLEMA 2](#_Toc74895565)

[1.5. FORMULACIÓN DEL PROBLEMA 5](#_Toc74895566)

[1.6. OBJETIVOS 5](#_Toc74895567)

[1.6.1. Objetivo general 5](#_Toc74895568)

[1.6.2. Objetivos específicos 5](#_Toc74895569)

[1.7. ALCANCE 6](#_Toc74895570)

[1.8. ENTREVISTA 7](#_Toc74895571)

[2. ELEMENTOS DEL SISTEMA BASADO EN COMPUTADORAS 11](#_Toc74895572)

[2.1. HARDWARE 11](#_Toc74895573)

[2.1.1. Servidor 11](#_Toc74895574)

[2.1.2. Cliente 11](#_Toc74895575)

[2.1.3. Medios de comunicación 12](#_Toc74895576)

[2.1.4. Otros Dispositivos 12](#_Toc74895577)

[2.2. SOFTWARE 12](#_Toc74895578)

[2.2.1. Servidor 12](#_Toc74895579)

[2.2.2. Cliente 12](#_Toc74895580)

[2.2.3. Otro software adicional 12](#_Toc74895581)

[2.3. DATOS 12](#_Toc74895582)

[2.4. PROCESOS 13](#_Toc74895583)

[2.5. GENTE/USUARIO 13](#_Toc74895584)

[2.6. DOCUMENTO 13](#_Toc74895585)

[3. TECNOLOGIA PARA EL DESARROLLO DEL SOFTWARE 14](#_Toc74895586)

[3.1. Estrategia para el desarrollo del software 14](#_Toc74895587)

[3.2. Metodología para el desarrollo del software 15](#_Toc74895588)

[3.2.1. Características del PUDS 15](#_Toc74895589)

[3.2.2. Características de UML 17](#_Toc74895590)

[3.3. HERRAMIENTAS DE DESARROLLO 18](#_Toc74895591)

[3.3.1. Software 18](#_Toc74895592)

[3.3.2. Hardware 18](#_Toc74895593)

[4. POSIBLES COSTOS 19](#_Toc74895594)

[5. POSIBLES BENEFICIOS 19](#_Toc74895595)

[5.1. TIEMPO 19](#_Toc74895596)

[5.2. ESFUERZO 19](#_Toc74895597)

[5.3. COSTOS 20](#_Toc74895598)

[6. POSIBLES CLIENTES 20](#_Toc74895599)

[7. DISEÑO DE DATOS 21](#_Toc74895600)

[7.1 DISEÑO LOGICO 21](#_Toc74895601)

[7.1.1 Diagrama de clases 21](#_Toc74895602)

[7.1.2 Mapeo 22](#_Toc74895603)

[7.1.3 Tabla de volumen 25](#_Toc74895604)

[7.1.4 Script 31](#_Toc74895605)

[7.1.5 Diagrama relacional 42](#_Toc74895606)

[7.1.6 Actualización de tuplas 43](#_Toc74895607)

[7.1.7 Consultas 51](#_Toc74895608)

[7.1.8 Procedimientos almacenados 57](#_Toc74895609)

[7.1.9 Disparadores 58](#_Toc74895610)

[8. ISHIKAWA 59](#_Toc74895611)

[8.1 Identificar Problemas 59](#_Toc74895612)

[8.1.1 Lista de problemas 59](#_Toc74895613)

[8.1.2 Depurar la lista de problemas 60](#_Toc74895614)

[8.1.3 Propietarios de Problemas 60](#_Toc74895615)

[8.1.4 Análisis del Problemas 61](#_Toc74895616)

[8.1.5 Estimación y cuantificación del problema 62](#_Toc74895617)

[8.1.6 Alternativas de cambio 63](#_Toc74895618)

[8.1.7 Conclusión y Recomendación 64](#_Toc74895619)

[8.1.8 Diseñar el diagrama de Ishikawa 64](#_Toc74895620)

[8.2 Diseñar el diagrama de Ishikawa 64](#_Toc74895621)

[8.3 Identificar las causas 65](#_Toc74895622)

[8.4 Analizar y discutir del Diagrama 66](#_Toc74895623)

[9 MODELO DE NEGOCIO 67](#_Toc74895624)

[9.1 Diagrama de actividad proceso de compra de producto 67](#_Toc74895625)

[9.2 Diagrama de actividad proceso de venta de producto 68](#_Toc74895626)

[9.3 Diagrama de actividad proceso de inventariar stock 69](#_Toc74895627)

[ANEXOS 70](#_Toc74895628)

# PERFIL

## INTRODUCCIÓN

En los últimos tiempos los sistemas de información dentro de los restaurantes han sido de gran ayuda para automatizar la gestión de ventas, ofrecer un mejor servicio de atención al cliente y brindar una supervisión detallada sobre el manejo del mismo.

Un restaurant de forma manual nos genera problemas como: pérdida de tiempo en la atención al cliente, disminución de ganancias, errores al realizar el cálculo manual, lentitud en los procesos de venta y precisar de más personal.

El presente proyecto realizará las actividades de compra de los productos, que se usarán en la producción y venta de los platos de comida. Se ayudará a optimizar la toma de decisiones que está estrechamente relacionado con los procesos de planificación y control, logrando así sus objetivos.

De acuerdo a lo mencionado proponemos desarrollar un “Sistema De Información Para Gestionar Las Compras y Ventas De La Rosticería "Restaurante Pollos El Buen Gusto””. La gestión de compras llevará un registro de los gastos en el abastecimiento de insumos y la cantidad adquirida del mismo, la gestión de ventas registrará y facturará los ingresos monetarios del restaurante, así mismo mostrará la cantidad de ventas en el día.

El presente sistema de información se implementará para corregir los problemas anteriormente mencionados, con respecto a los pedidos y cobro del mismo.

## ANTECEDENTE

Hace 7 años la rosticería “Restaurante Pollos El Buen Gusto” se fundó y abrió sus puertas como una nueva opción para el pueblo cruceño, se encuentra ubicada en el municipio de Puerto Quijarro, Av. Luis Salazar de la Vega al frente del Hotel Silvia.

El negocio fue fundado por la Sra. Rosa Condori junto a sus hijos en la localidad de Puerto Quijarro. Desde entonces ha ido en creciendo hasta la actualidad donde fue cedido a su hijo Jonathan Ismael Zambrana Condori.

La idea nació por curiosidad y probar como les iba con una rosticería. Al principio no todo fue bueno ya que no tenían experiencia, pero poco a poco se fueron ganando un puesto en el mercado de la gastronomía.

## JUSTIFICACIÓN

Muchos restaurantes que pasan de ser pequeños donde solo van los vecinos a

medida que pasa el tiempo comienzan a ganar popularidad y obtienen una mayor actividad, esto hace que el control de la información sea más difícil de manejar.

Si no queremos perder información o desaprovechar el potencial que la rosticería está ganando, necesitamos una organización que lleve el control de cada proceso (ventas, producción, atención de clientes, etc.), y manualmente, aunque no es imposible, seguro que necesitara muchas más personas, tiempo, gastos y esfuerzos que un sistema eficiente y completo como el presente. Y bueno por ello es que realizamos como grupo este sistema para el proyecto, ya en la introducción se podrán reconocer la mayoría de los servicios y funciones que el sistema será capaz de realizar y así mismo su alcance.

El presente es un sistema de información que se responde a las necesidades de un entorno abrumado por la presión de corriente intensa de información y la pone en orden para sacarle provecho.

## DESCRIPCIÓN DEL PROBLEMA

La “rosticería restaurante Pollos El Buen Gusto” no cuenta con un sistema de información que le permita administrar la distribución, compras y ventas de los platos. Se generan muchas dificultades a la hora de procesar los datos en términos de tiempo y las posibles falencias que este trabajo manual implica; sin mencionar el riesgo de que estos datos se pierdan. La información acerca de los administradores, la desorganización al momento de vender y hacer el talonario con la cantidad de ventas. Son datos que la empresa maneja de forma manual, esto, aparte de generar lentitud al momento del registro de ventas y consulta de las mismas, puede generar que no haya una buena atención al cliente, el cual puede causar problemas al negocio. Debido a estas dificultades en el manejo de datos dentro del restaurante, se desarrolla el presente proyecto, que se va a poder identificar en los siguientes procesos de desarrollo.

**Proceso Personal**

Empleados

A la hora de contratar algún personal que trabaje en el restaurant, aunque normalmente suelen llegar por recomendaciones de amigos, se debe anotar el nombre completo, número de carnet, número de teléfono, fecha de ingreso. Además, al empleado se le paga mensualmente.

* Como no hay una capacitación de personal, esto puede llegar a causar pérdidas de clientes.

**Gestión de Compras**

La rosticería “Restaurante Pollos El Buen Gusto” realiza sus pedidos de insumos diarios para la preparación del menú, pero esto no lleva un control exacto de cuántos productos comprar debido a que el registro de ventas es manual y puede haber errores.

* Como no tiene un informe pueden comprar menos de lo esperado o más insumos, causando pérdidas.
* Las compras las realiza según las carencias de productos que tiene la rosticería “Restaurante Pollos El Buen Gusto”, no lleva un registro adecuado para futuros sondeos de gastos en un tiempo determinado.
* El dueño de la rosticería “Restaurante Pollos El Buen Gusto” no lleva un control de los proveedores, por lo cual realiza un sondeo de los precios para escoger el precio y calidad más óptima, sin embargo esto resulta una pérdida de tiempo y esfuerzo.

**Gestión de Ventas**

El registro de las ventas, entre el cliente y el cajero, se realiza de manera manual por ticket.

En primera instancia el cliente se dirige a caja, y consulta los precios del plato y bebida, en caso de haber pedidos, el empleado rellena la Nota de pedido, con las especificaciones del cliente y los clientes pagan al contado, esta nota pasa al cocinero, quien verifica de manera manual, y el mesero es quien alista el pedido, ya sea para mesa o llevar.

Si el cliente pide para la mesa existen inconvenientes de los empleados de cocina al momento de recibir los comprobantes de las ventas (ticket) por parte de los empleados que reciben el pedido, porque hay casos en los que pierdan el ticket y pueda generar un duplicado o confusión en los pedidos.

Si el cliente pide para llevar, el ticket le es entregado, en este caso el cliente es el responsable de llevar el pedido al empleado de cocina.

* El registro de pedido se hace de forma manual, lo que genera demora en atención al cliente
* Posibles errores en la cantidad de pedido a la hora de la entrega, ya que el registro y manipulación del pedido es manual, puede haber errores a la hora de la entrega, como al equivocarse de pedido.
* Variación en el precio del menú en la venta por motivos de variabilidad en el precio de insumos en el mercado.
* Pérdida de notas de pedido por parte del cajero hacia el cocinero, lo cual genera pérdidas de clientes.
* Variación en el precio de menú en la venta por motivos de competencia existente en el rubro y alteraciones de los precios por parte de los empleados.
* También puede ocurrir el caso contrario, es decir, que se llegue al punto de que el material se haya agotado y no se tenga disponibilidad, trayendo incomodidades para el cliente ya que se alargan los plazos de entrega de determinado producto, donde el único afectado realmente es la compañía, lo cual se traduce en pérdida de imagen.

**Proceso Administrativo**

El control administrativo es una de las funciones gerenciales, como lo es la planificación, organización, supervisión y dirección. Es una función importante, porque ayuda a verificar los errores y tomar las medidas correctivas para minimizar la desviación de los estándares y lograr los objetivos establecidos de la organización de la manera deseada. Como no existe un sistema esto puede causar problemas en el manejo de sus funciones.

El dueño de la rosticería se encarga de designar las funciones que cada empleado va realizar.

**Proceso de inventario**

El restaurant el buen gusto hace un registro manual de los productos como ser: pollos, bebidas Analcohólicas, desechables plásticos, arroz, fideos, verduras, alimentos y otros.

En cuanto el producto del pollo es recibido 3 veces a la semana, pero esto puede variar según el consumo, el producto es marinado y posteriormente cocinado para su venta donde cada presa o conjunto de presas puede representar un plato para su venta.

La bebida es recibida una vez por semana donde la mayor cantidad de un mismo producto depende de las ventas de la anterior semana, posteriormente una parte de las bebidas son refrigeradas y las otras son almacenadas cerca del refrigerador para poder así abastecer en caso de que haga falta o disminuya el stock.

El problema es que todos estos procesos son registrados manual y mentalmente donde pude ocurrir un descuido de los empleados al abastecer los refrigeradores o en reabastecimiento de la cantidad de pollos cocinados.

## FORMULACIÓN DEL PROBLEMA

El presente proyecto desarrollará el registro de los pedidos de productos dentro de la rosticería, registro de los empleados, el detalle de las ventas realizadas en la jornada de trabajo, así como el registro de los procesos de compra.

## OBJETIVOS

### Objetivo general

Desarrollar un Sistema de Información para gestionar la compra, venta e inventario de la rosticería “restaurante pollos el buen gusto”

### Objetivos específicos

* Recolectar información sobre la rosticería “restaurante pollos el buen gusto”, utilizando técnicas de cuestionario y entrevistas.
* Analizar un sistema apropiado para solucionar los problemas existentes en el área de ventas, el cual se realizará en el Lenguaje Unificado de Modelado (UML) y el Proceso Unificado de Desarrollo de Software.
* Diseñar las diferentes arquitecturas haciendo uso del lenguaje unificado de modelado (UML) siguiendo el proceso de desarrollo unificado (PUDS), además una base de datos para el registro y almacenamiento de los datos en MySQL.
* Implementar la interfaz del software acorde a los requerimientos del usuario haciendo uso del framework laravel y utilizando el patrón de arquitectura de MVC (Modelo, Vista, Controlador).
* Probar el sistema ingresando n cantidad de datos obtenidos de la empresa para observar su comportamiento y buscar posibles fallas.

## ALCANCE

Considerando la formulación del problema indicada anteriormente, donde se establece los procesos de negocio que se pretende solucionar, en esta sección se establecen las funciones que el sistema implementará para poder cubrir estos procesos.

**• Módulo Compras:**

* **Gestionar Nota de compra:** Este proceso está encargado de gestionar las notas de compras de producto a los proveedores.
* **Gestionar Proveedor:** Este proceso está encargado de gestionar a los proveedores.

**• Módulo Ventas:**

* **Gestionar Pedido:** Este proceso está encargado de gestionar notas de ventas por pedidos
* **Gestionar Recibo:** Este proceso está encargado de generar recibo de los pedidos con los respectivos datos de los clientes (para los clientes) y del mismo recibo se genera un ticket para el cocinero.

**• Módulo inventario:**

* **Gestionar Producto Menú:** Este proceso está encargado de registrar los productos (menú) que el restaurante ofrece.
* **Gestionar Categoría Menú:** Este proceso está encargado de clasificar los productos del menú en Categorías.
* **Gestionar Producto:** Este proceso está encargado de clasificar los productos que el restaurant recibe.

**• Módulo Personal:**

En este módulo exclusivamente se gestionará los datos de toda persona externa (proveedores) e interna (los empleados) de la empresa.

* **Gestionar Empleado:** Este proceso está encargado de gestionar todos los datos de los empleados del restaurante.
* **Gestionar Cargo:** Este proceso está encargado registrar los roles de los empleados en la empresa.

**• Módulo Administrar Usuario:**

* **Gestionar Usuario:** Este proceso está encargado de registrar como usuario (administrador, cajero).
* **Gestionar Acción:** Este proceso está encargado registrar el conjunto de acciones que el usuario ejecutará en el sistema.
* **Gestionar Bitácora:** Este proceso está encargado de controlar las acciones de cada usuario en el sistema (login, logout, etc.).

## ENTREVISTA

**Objetivo:** Recolectar información primordial para el desarrollo del sistema

**Entrevista # 1**

**Lugar:** Reunión virtual a través de zoom **Duración:** 45 min.

**Datos de la Empresa**

**Nombre:** Restaurante pollos el buen gusto

Privada

**Datos de la Entrevistada**

**Nombre:** Luz clarita Zambrana Condori

**Cargo:** Administradora

**Datos del Entrevistador**

**Nombre:** Daniel Maldonado Gutierrez

**Anexos**

Entrevista #1

**1. ¿Cuál es su nombre completo?**

**R.** Luz clarita Zambrana Condori

**2. ¿Cuál es el nombre del negocio del cual es administrador?**

**R.** Restaurante Pollos el buen Gusto

**3. ¿Dónde se encuentra ubicado actualmente?**

**R.** Puerto Quijarro

Av. Luis Salazar de la Vega

Al frente del hotel Silvia

**4. ¿Cuántas personas trabajan en el negocio y que trabajos realizan?**

**R.** Somos 7 personas, 4 personas ajenas, que son: Mesero, servidor, lava plato, cocinero y 3 personas que son mi madre mi hermana y yo. Mi madre es la que pone el orden y a hacer las cosas a los ayudantes.

Mi hermana es servidora en momentos de abundante clientela, y yo me encargo de la parte financiera, atendiendo la caja y ayudando a atender a la gente.

**5. ¿Cómo se fundó el negocio?**

**R.** Nosotros nos metimos en el negocio de la pollería por metiche, ya que queríamos saber cómo nos iría, y al principio nos fue mal, hasta el punto de meternos en deudas, ya que el pollo no tenía ese saber que tiene y las salsas eran las comunes que puedes llegar a encontrar en cualquier lado, así que nos fue mal al principio, como sabrás, mi madre es muy sociable y hace muchas amistades, y una de sus amigas le voy en esa situación nos ayudó a experimentar, a cambiar la receta, y mi madre llego a crea una nueva receta, también cambiamos la salsa, esta salsa que tenemos actualmente la creo mi madre, después de experimentar a la gente le gusto y empezamos a tener más clientela.

**6. ¿Quién es el que elige a los empleados, que requisitos pide el lugar para contratar, tiene un formulario de preguntas antes de contratar?**

**R.** los contratamos según la edad, referencias de la agencia y experiencia en sus trabajos anteriores

**7. Al momento de contratar, ¿Qué datos pide el restaurant, pide dejar algún documento que lo identifique?**

**R.** Mayormente dejan su carnet de identidad, y se les devuelve cuando ya quieren retirarse del trabajo.

**8. ¿Quién tiene acceso a los datos, documentos, etc…?**

**R.** Los documentos de los empleos lo maneja la propietaria del negocio o sea la dueña o también puede manejarlos la segunda al mando.

**9. De los empleados del restaurant ¿Cuánto se le paga a los empleados, cuantos días a la semana se trabaja, cuantas horas se trabaja, tiene un horario de llegada o de entrada, los empleados trabajan cama adentro y a quién realiza las compras de los productos el restaurant?**

**R.** Le llegamos a pagar 2000bs por mes, se trabaja 6 días a la semana, se puede llegará a trabajar 14 horas o menos según la clientela que haiga o sea si acabamos temprano se van temprano, su hora de llegada es a las 8a.m., mayormente contratamos empleamos cama adentro, las compras lo llega a realizar aquella que maneja el dinero ya que así puede administrar cuando se compró y para cuanto alcanza.

**10. ¿De quién compra el pollo, o algún otro producto que se necesite para el funcionamiento del restaurant, como mayonesa, kétchup, salsas especiales, ingredientes (arroz, papa, aceite, etc.), bebidas (coca cola, fanta, etc… ya sea de dos litros, 1 litro, popular, retornable, etc…)?**

**R.** compramos de un solo lugar, porque así se nos facilita, también nos hacen el servicio de traer las cosas que compramos ya que llegan hacer muchas.

**11. ¿En caso de que no ocupe todo lo que compro y se pase, existe algún reembolso de parte del productor que entrega los productos? (ya sea el vendedor de los pollos)**

**R.** No existe el reembolso, nosotros calculamos los vamos a vender y los tenemos que comprar, ya como sabes llevamos mucho en este negocio y se nos facilita el saber cuánto arroz, fideo o papá vamos a comprar.

**12. ¿El restaurant cuenta con reservas y entrega a domicilio (delivery) o estuvo contando con este servicio debido a la pandemia?**

**R.** No contamos con entrega a domicilio, ya q en este lugar la gente aún no conoce eso, están acostumbrándose a comprar físicamente y extraer sus productos del lugar de venta

**13. En caso de que el cliente rompa algo, o destruya algo, ¿Qué acciones toma el restaurant?**

**R.** Depende de la situación, en el cliente rompa la cosa, ya que mayormente son los niños quienes lo rompen y pues son niños y no tomamos ninguna acción.

**14. ¿El restaurant cuenta con pago por internet, por tarjeta, etc.? O todo es efectivo, en caso de que no, qué opinión tiene sobre esto, o piensa implementarlo más adelante, al igual que el servicio de delibera**

**R.** No contamos con la plataforma virtual, el restaurante aún no ha tomado esa opción.

**15. ¿Quién es el que administra y como es el proceso de administración actualmente en el restaurant?**

**R.** Actualmente, la que maneja la parte financiera soy yo, y la que se encarga de manejar a los ayudantes a instruirlos en sus labores es mi madre.

**16. En la parte financiera, se maneja las pérdidas, las ganancias, el capital y el capital y ganancias de las bebidas. Un breve relato de como es el proceso de preparar los platos de comida**

**R.** Pues sólo es arroz, fideo cocido con salsa y papa frita.

**17. ¿Quiénes se encargan de preparar los platos?**

**R.** Los que se encarga en preparar cada plato son los cocineros, ya que a ellos, se le enseña

**18. ¿Cuánto tiempo tardan en preparar o si tienen un horario para preparar la comida?**

**R.** Pues se tardan 3 horas en preparar.

**19. ¿Cuántos empleados se encargan de la cocina?**

**R.** De cocineros tenemos 3.

**20. ¿Qué se hace con lo que sobre del día?**

**R.** pues mayormente se calcula cuánto vamos a poner en el día, y tomamos mucho en cuenta la hora, pero si no llegara a sobrar se lo damos de sobra a los perro o lo vendemos de sobra de perro.

**21. Si es que falta comida, ¿Qué es lo que hacen? ¿Vuelven a preparar más?**

**R.** No, llegamos a preparar más, simplemente acabamos lo que ponemos en venta ya que la preparación dura más de 1 hora y la clientela no puede esperar.

**22. ¿Quiénes son los que sirven la comida?**

**R.** El cocinero, mi hermana, y mi madre.

# ELEMENTOS DEL SISTEMA BASADO EN COMPUTADORAS

## HARDWARE

### Servidor

* Procesador AMD Ryzen 7 3700X 4.4Ghz, 8 Núcleos reales, 16 Hilos.
* Placa Base ASRock x570 Steel Legend Wifi ax.
* Chipsec AMD Ryzen SOC.
* Tarjeta Gráfica Radwon RX 580 Series, 8 Gbytes.
* Memoria RAM DDR4 16 GBytes 1333 Mhz.
* Disco Duro WDC de 1TBytes.
* Disco Solido HP de 500 GBytes.
* Pantalla de 32 Pulgadas Marca Samsung LCD.
* 2 Puertos HDMI y 9 Puertos USB.
* Lector/Quemador de CDS/DVDS.

### Cliente

* Procesador core i3 6ta generación
* memoria ram 8 gb
* disco duro 1Tb
* Velocidad de reloj de 3.7 GHz

### Medios de comunicación

* Conector RJ45
* Router TPLink

### Otros Dispositivos

* Impresora Canon PIXMA MP250

## SOFTWARE

### Servidor

* Sistema operativo Windows 8.1
* Gestor de Base de Datos MYSQL workbench
* Lenguaje de programación PHP

### Cliente

* Sistema Operativo Windows 8, Linux y Mac os
* Navegador (Chrome, Mozilla, Opera, etc)

### Otro software adicional

* Microsoft Office Word

## DATOS

El sistema de datos manejará los siguientes datos:

* Usuario

(username, password, email)

* Empleados

(nombre, apellido, carnet de identidad, teléfono/celular, dirección)

* Proveedor

(nombre, apellido, carnet de identidad, teléfono/celular, dirección)

* Productos

(nombre, precio)

* Venta

(pedidos, precio, cantidad, fecha)

* Inventario

(nombre\_producto, stock, fecha)

* Nota de compra

(nombre\_producto, cantidad, costo, fecha, proveedor)

* Perdida del inventario

(código, fecha, costo\_total)

## PROCESOS

* **Proceso de gestión de compra. -** En el momento que llega los productos de los proveedores, el administrador llena la nota de compra y envía dichos productos a determinado almacén
* **Proceso de Gestión de ventas. -** El cliente se aproxima al cajero y hace su pedido posteriormente el cliente paga por el pedido y se le es entregado un ticket con el cual puede hacer el intercambio por el producto del menú.
* **Proceso de Gestión de inventario. –** Los productos recibidos son registrados y almacenados para su venta.
* **Proceso de administración de personal. –** Cuando se contrata personal se debe registrar los datos más importantes.
* **Proceso de gestión de proveedor. -**  Se registra cada proveedor en el sistema.
* **Proceso de gestión de usuarios. –** Un administrador o dueño podrá adicionar, modificar, inactivar o eliminar un usuario del sistema iniciando sesión en el sistema y haciendo uso del módulo de gestión de usuario, registro de nombre de usuarios, password y correo electrónico.

## GENTE/USUARIO

El equipo humano que desarrolla el sistema está conformado por:

* + - * + Huarachi Laime Raúl
        + Llanos Duran Junior Javier
        + Maldonado Gutierrez Daniel
        + Mamani Soliz Christian Celso
        + Zeballos Carvallo Oscar Daniel

## DOCUMENTO

El sistema debe generar información física referente a:

* Nota de compra
* Tickets de pedidos
* Control de inventario

# TECNOLOGIA PARA EL DESARROLLO DEL SOFTWARE

## Estrategia para el desarrollo del software

La estrategia para el Desarrollo de Software del presente proyecto tendrá un enfoque

centrado en dos herramientas de análisis y diseño: El Proceso Unificado de Desarrollo de

Software (PUDS) y el Lenguaje Unificado de Modelado (UML).

El Proceso Unificado "es un proceso de desarrollo de software configurable

que se adapta a través de los proyectos variados en tamaños y complejidad.

El Proceso Unificado guía al equipo de proyecto en la administración del

desarrollo iterativo de un modo controlado mientras se balancean los

requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto.

Este proceso describe los diversos pasos involucrados en la captura de los

requerimientos y el establecimiento de una guía arquitectónica en un periodo

corto, para diseñar y probar el sistema realizado de acuerdo a los

requerimientos y a la arquitectura solicitada.

El proceso Unificado describe que entregables producir, cómo desarrollarlos y

también provee patrones.

Lenguaje de Modelado Unificado (UML) es el lenguaje de modelado orientado

a objetos estándar de la industria para especificar, visualizar, construir y

documentar los elementos, artefactos de los sistemas de software, así como

para modelado del negocio y de otros sistemas que no son de software.

Simplifica el proceso complejo de análisis y diseño de software, facilitando un

plano para la construcción.

## Metodología para el desarrollo del software

### Características del PUDS

El Proceso Unificado ha adoptado un enfoque que se caracteriza por:

* Interacción con el usuario continua desde un inicio
* Mitigación de riesgos antes de que ocurran
* Liberaciones frecuentes
* Aseguramiento de la calidad
* Involucramiento del equipo en todas las decisiones del proyecto
* Anticiparse al cambio de requerimientos.

**Dirigido por los casos de uso**

En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc. el proceso dirigido por casos de uso es el RUP.

**Centrado en la arquitectura**

El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura software de un sistema. La analogía con la construcción es clara, cuando construyes un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc.

La arquitectura de Sistema es la visión general común en la que todos los empleados (desarrolladores y otros usuarios) deben estar de acuerdo. La arquitectura da una clara perspectiva del sistema completo, necesaria para controlar el desarrollo. Se necesita arquitectura para:

* Comprender el Sistema
* Organizar el desarrollo
* Fomentar la reutilización
* Hacer evolucionar el Sistema

**Iterativo e Incremental**

Un producto software puede conllevar mucho tiempo resulta más práctico dividirlo en pequeñas partes o mini proyectos, cada mini proyecto es una iteración que acaba en un incremento las iteraciones hacen referencia a pasos en el flujo de trabajo y los incrementos al crecimiento del producto. La iteración trata un grupo de casos de uso que juntos amplían la utilidad del producto desarrollado hasta ahora e identifica los riesgos más importantes.

En cada iteración, se identifican y especifican los casos de uso más relevantes, se crea un diseño utilizando la arquitectura como guía, se implementa el diseño como componentes y se verifica que dichos componentes satisfacen los casos de uso. 25

**Fases del PUDS**

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

**Fase de Inicio**

En esta fase se establece la oportunidad y alcance el proyecto. Se identifican todas las entidades externas con las que se trata (actores) y se define la interacción en un alto nivel de abstracción: se deben identificar todos los casos de uso, y se deben describir algunos en detalle. La oportunidad del negocio incluye: definir los criterios de éxito, identificación de riesgos, estimación de recursos necesarios, y plan de las fases incluyendo hitos.

**Fase de elaboración**

Definir y validar una arquitectura estable. Se hace un refinamiento de la Visión del sistema, basándose en nueva información obtenida durante esta fase, se establece una sólida comprensión de los casos de uso más críticos que definen las decisiones arquitectónicas y de planificación. Creación de los planes de desarrollo detallados para las iteraciones de la fase de construcción.

**Fase de construcción**

Gestión de los recursos, optimización y control de los procesos de construcción del software. Se completa el desarrollo de los componentes y/o subsistemas, probándolos contra un conjunto definido de criterios aprobados al inicio del proyecto.

**Fase de transición**

Ejecución de los planes de implantación. Se finalizan los manuales de usuario y mantenimiento. Pruebas del sistema en el entorno de explotación. Creación de una reléase del sistema. Validación del sistema por los usuarios. Ajuste fino del sistema según la validación con el usuario. Se facilita la transición el sistema al personal de mantenimiento. Se pone el producto a disposición del usuario final.

### Características de UML

Lenguaje estándar para el modelado de software lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. Lenguaje usado por el Proceso Unificado. Lenguaje que permite a los desarrolladores visualizar el producto de su trabajo (Artefactos) en esquemas o diagramas estandarizados.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

* **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
* **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.

Define una serie de diagramas:

* **Diagrama de Caso de Uso:** muestra las relaciones entre el sujeto (sistema) y los casos de uso.
* **Diagrama de Clase:** Modela la estructura estática de las clases en el sistema.
* **Diagrama de Secuencia:** Representa una interacción, poniendo el foco en la secuencia de los mensajes que se intercambian.
* **Diagrama de Componentes:** Modela los componentes que componen una aplicación, sistema o empresa.

## HERRAMIENTAS DE DESARROLLO

### Software

**Servidor Web:** Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.11 y la noción de sitio virtual. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido

**Motor de base datos:** MySQL, es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

**Editor de base datos:** PhpMyAdmin, es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos. Se encuentra disponible bajo la licencia GPL.

**Programación Web:** Que se realizara en PHP (bajo el framework de Laravel versión 8), HTML y CSS.

PHP es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. El cual es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet y CSS (Hojas de Estilo en Cascada) es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla permitir la separación de los contenidos de los documentos escritos en HTML.

### Hardware

* Procesador Core i3 o mayor
* Velocidad 3.0GHZ o mayor
* Memoria RAM 2 GB recomendado 4GB
* Disco Duro 120 recomendado, 500 dependiendo de la cantidad de información que se va almacenar y los programas a instalar
* Conexión a internet 1MB

# POSIBLES COSTOS

Considerando el desarrollo del proyecto por 5 personas en 2 meses, se consideran los siguientes costos:

|  |  |
| --- | --- |
| **EQUIPOS COMPUTACIONALES Y MATERIALES EN GRAL.**  **(COSTO EN DOLARES)** | **Costos** |
| Laptops computacionales: | **2500$** |
| Material de escritorio | **1000$** |
| **Licencias y Software utilizados** |  |
| Sistema Operativo Windows10 C/Licencia | **72$** |
| Ubuntu | **0$** |
| Enterprise Architect 13.0 | **499$** |
| MariaDB(php)/MySql Workbench Standard C/Licencia | **0$** |
| **Tiempo y dedicación** |  |
| Estudio de viabilidad | **50$** |
| Implementación y pruebas de sistema | **50$** |
| Tiempo dedicado al proyecto (3 meses) | **5142$** |
| **Total** | **9313$** |

# POSIBLES BENEFICIOS

## TIEMPO

* Agilizar la búsqueda y el almacenamiento de datos.
* Evitar pérdida de tiempo en los pedidos recopilando información que está almacenada en bases de datos.
* Brindar la información en tiempo real de cualquier producto del menú.
* Facilita la verificación del orden de los pedidos, y así no genera confusión entre los trabajadores que están sirviendo el pedido.

## ESFUERZO

* Facilitar el manejo de información.
* Acelerar el registro de venta de los productos a ofrecer
* Una Interfaz sencilla y de rápido acceso, para el mejor entendimiento del Usuario.
* La administración utiliza la información gestionada por nuestro sistema para identificar los diferentes tipos de productos del menú, mejorar la variedad de productos.

## COSTOS

* Reduce el costo en la compra de facturas manuales.

# POSIBLES CLIENTES

Con la gran cantidad de empresas del sector productivo que existe hoy en día con métodos de almacenamiento y procesamiento de información rudimentario (pluma y papel). Este sistema podría ser de gran utilidad en empresas que necesiten un control sobre sus facturas venta y también que manejan gran cantidad de personal como ser Empresas del mismo rubro, patios de comida, pensiones, etc.

Este software puede ser empleado a muchos pequeños y medianos restaurantes de cualquier comida, que necesiten un buen control al registrar sus ventas, compra y manejo de sus inventarios. Este es el caso de muchos restaurantes en la ciudad de Santa Cruz o en el resto del país.

# DISEÑO DE DATOS

## DISEÑO LOGICO

### Diagrama de clases



### Mapeo

1 persona

**pk**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CI | nombre | apellido | celular | direccion | sexo | Tipo\_p |

2 proveedor

**pk**

|  |  |  |
| --- | --- | --- |
| Ci | nombre\_negocio | direccion\_negocio |

3 trabajador

**Pk fk**

|  |  |  |  |
| --- | --- | --- | --- |
| Ci | fecha\_inicio | estado | cod\_cargo |

4 cargo

**pk**

|  |  |  |  |
| --- | --- | --- | --- |
| cod\_cargo | descripcion | sueldo | perfil\_usuario |

5 usuario

**Pk fk**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Username | psswrd | email | nivel\_acceso | Ci |

6 Producto

**pk**

|  |  |  |
| --- | --- | --- |
| Codigo | nombre | tipo\_producto |

7 nota\_de\_compra

**Pk fk fk**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Codigo | Fecha\_hora | total | Username | Cod\_proveedor |

8 tipo

**Pk**

|  |  |  |
| --- | --- | --- |
| Cod\_tipo | descripcion | abreviatura |

9 detalle\_compra

**Pk fk pk fk fk**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| codigo\_nota\_compra | Codigo\_producto | cantidad\_producto | precio\_total | precio\_unitario | descripcion | codigo\_tipo |

10 lote

**Fk fk**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cod\_lote | fecha\_vencimiento | fecha\_elaboracion | cantidad | codigo\_producto\_l | codigo\_nota\_compra\_l |

11 bebida

**pk**

|  |  |  |
| --- | --- | --- |
| Codigo\_producto | estado | precio |

12 pedido

**Pk fk**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cod\_pedido | fecha\_hora | costo\_total | llevar | Username |

13 detalle\_pedido

**Pk fk fk fk fk**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Codigo | Cantidad\_plato | precio | codigo\_pedido | codigo\_plato | codigo\_contenido | Codigo\_bebida |

14 detalle\_presa

**Pk fk pk fk**

|  |  |  |
| --- | --- | --- |
| cod\_producto | cod\_detalle\_pedido | cantidad |

15 plato

**pk**

|  |  |  |
| --- | --- | --- |
| Cod\_plato | nombre | precio |

16 contenido

**Pk**

|  |  |  |  |
| --- | --- | --- | --- |
| Cod\_contenido | nombre | precio | estado |

17 contenido\_producto

**Pk fk pk fk**

|  |  |  |
| --- | --- | --- |
| cod\_plato | cod\_contenido | cantidad |

18 ingreso\_egreso

**Pk fk**

|  |  |  |  |
| --- | --- | --- | --- |
| Codigo | fecha\_hora | tipo | Nombre\_usuario |

19 detalle\_ingreso\_egreso

**Pk fk pk fk**

|  |  |  |
| --- | --- | --- |
| codigo\_ingreso\_egreso | codigo\_producto | cantidad |

20 stock

**Pk pk**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cod\_stock | cod\_producto | cantidad | ingreso | stock | Fecha\_hora |

21 bitacora

**Pk fk**

|  |  |  |  |
| --- | --- | --- | --- |
| Codigo | fecha\_hora | descripcion | Username |

**Normalizacion**

1 Persona

**pk**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CI | nombre | apellido | direccion | sexo | Tipo\_p |

**Celular\_persona**

|  |  |
| --- | --- |
| **Ci** | celular |

### Tabla de volumen

**PERSONA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| ci | Varchar(10) | 1-10 | NO | **PK** | Código del Carnet de Identidad de la persona |
| nombre | Varchar(30) | 1-30 | NO |  | Nombre de la persona |
| apellido | Varchar(30) | 1-30 | NO |  | Apellido de la persona |
| direccion | Varcha(40) | 1-40 | NO |  | Dirección de la persona |
| Sexo | Char | 1 | No |  | Sexo de la persona(M o F) |
| tipo\_p | Char | 1 | NO |  | Tipo de persona (trabajador o proveedor) |

**CELULAR\_PERSONA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| ci | Varchar(10) | 1-10 | NO | **PK,FK** | Código del Carnet de identidad de la persona |
| celular | int | 4 | SI | **PK** | Número de celular de la persona (sin contar el código de país) |

**PROVEEDOR**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| ci | Varchar(10) | 1-10 | NO | **PK, FK** | Código del Carnet de Identidad del proveedor |
| nombre\_negocio | Varchar(30) | 1-30 | NO |  | Nombre del negocio al que pertenece el proveedor |
| dirección\_negocio | Varchar(40) | 1-30 | NO |  | Dirección del negocio al que pertenece el proveedor |

**CARGO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| cod\_cargo | int | 4 | NO | **PK** | Código de los cargos que existen |
| descripcion | Varchar(30) | 1-30 | NO |  | Descripción del cargo |
| sueldo | Float | 4 | NO |  | Sueldo que ofrece el cargo |
| perfil\_usuario | boolean | 1 | NO |  | Perfil usuario que determina si el cargo puede tener uso del sistema |

**TRABAJADOR**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| ci | Varchar(10) | 1-10 | NO | **PK, FK** | Código del Carnet de Identidad de la persona |
| fecha\_inicio | Datetime | 8 | NO |  | Fecha de inicio o registro del trabajador |
| Estado | Boolean | 1 | No |  | Indica si el trabajador sigue trabajando, true si aun sigue trabajando, false si ya no trabaja en el restaurant |
| cod\_cargo | int | 4 | NO | **FK** | Código del cargo que ocupa el trabajador |

**USUARIO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| Username | Varchar(20) | 1 a 20 | NO | **PK** | Código del usuario |
| Psswrd | Varchar(30) | 1 a 30 | NO |  | Contraseña de usuario |
| email | Varchar(50) | 1 a 50 | NO |  | correo |
| Nivel\_aceso | int | 4 | NO |  | Indica el nivel de permisos que tendrá el usuario |
| Ci | Varchar(10) | 1 a 10 | NO | FK | Llave foránea de la tabla  usuario |

**PRODUCTO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| codigo | Int | 4 | NO | **PK** | Código del producto |
| nombre | Varchar(30) | 1 a 30 | NO |  | Nombre del producto |
| tipo\_producto | char | 1 | NO |  | Tipo de producto, p si es producto presa, b si es producto bebida, i si es producto ingrediente |

**NOTA\_DE\_COMPRA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| codigo | int | 4 | no | **PK** | Código de la nota de compra |
| Fecha\_hora | datetime | 8 | no | - | Fecha de la nota de compra |
| total | float | 4 | no | - | Costo total de la nota de compra |
| username | varchar | 20 | no | **FK** | Usuario que registra la compra |
| Cod\_proveedor | varchar | 10 | no | **FK** | Proveedor a quien se le hace la compra |

**TIPO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| cod\_tipo | int | 4 | no | **PK** | Código del tipo de unidad |
| descripcion | varchar | 40 | no | - | Descripción de la unidad |
| abreviatura | varchar | 5 | no | - | Abreviatura del tipo |

**DETALLE\_COMPRA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| cod\_nota\_compra | int | 4 | NO | **PK, FK** | Código de la tabla nota de compra |
| cod\_producto | int | 4 | NO | **PK, FK** | Código de la tabla producto |
| cantidad\_producto | float | 4 | No |  | Cantidad del producto |
| Precio\_total | float | 4 | No |  | Precio total de la compra del producto |
| Precio\_unitario | float | 4 |  |  | Precio por unidad de producto |
| descripcion | Varchar(40) | 1 a 40 |  |  | Detalles de los productos |
| cod\_tipo | int | 4 | NO | **FK** | Tipo de unidad en la que se compra |

**LOTE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| cod\_lote | int | 4 | no | **PK** | Código dellote |
| Fecha\_vencimiento | datetime | 8 | No | - | Fecha de vencimiento del lote |
| Fecha\_elaboracion | Datetime | 8 |  | - | Fecha que se elabora el lote |
| Cantidad | Int | 4 | No |  | Cantidad de producto que viene en el lote |
| Codigo\_producto\_l | Int | 4 | No | **Fk** | Llave foránea con detalle de compra |
| Codigo\_nota\_compra\_l | int | 4 | no | **Fk** | Llave foránea con detalle de compra |

**BEBIDA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD (BYTE) | NULO | LLAVE | DESCRIPCION |
| codigo\_producto | Int | 4 | NO | **PK** | Código de bebida |
| estado | Boolean | 8 | NO |  | True si hay bebida, false si se acabo la bebida |
| precio | FLOAT | 4 | NO |  | Precio de venta de la bebida en la rosticeria |

**PEDIDO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD(Byte) | NULO | LLAVE | DESCRIPCION |
| Cod\_pedido | Int | 4 | NO | **PK** | Codigo del pedido y llave primaria |
| Fecha\_hora | Datetime | 8 | NO |  | Fecha y hora de pedido |
| Costo\_total | Float | 4 | NO |  | El costo total de pedido |
| Llevar | boolean | 1 | NO |  | True si es para llevar, false si no es para llevar ( comer en la rosticería) |
| Username | Varchar(15) | 1 a 15 | NO | FK | Llave foránea del usuario que hace el pedido |

**PLATO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD(Byte) | NULO | LLAVE | DESCRIPCION |
| Cod\_plato | Int | 4 | NO | **PK** | Código del plato y llave primaria |
| Nombre | Varchar(20) | 1 a 20 | NO |  | Nombre del plato |
| Precio | Float | 4 | NO |  | Precio del plato |

**CONTENIDO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD(BYTE) | NULO | LLAVE | DESCRIPCION |
| Cod\_contenido | Int | 4 | NO | **pk** | Llave primaria de contenido y codigo |
| Nombre | Varchar(20) | 1 a 20 | No |  | Nombre del contenido |
| Precio | Float | 4 | No |  | El precio\_contenido |
| Estado | Boolean | 1 | No |  | Estado en el que se encuentra, true si hay, false si no hay, o se agoto |

**CONTENIDO\_PRODUCTO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD(Byte) | NULO | LLAVE | DESCRIPCION |
| Cod\_plato | Int | 4 | NO | **PK- FK** | Código del plato |
| Cod\_contenido | Int | 4 | NO | **PK-FK** | Código del contenido |
| Cantidad | int | 4 | NO |  | Cantidad del contenido\_producto en el plato |

**DETALLE\_PEDIDO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD(Byte) | NULO | LLAVE | DESCRIPCION |
| Código | Int | 4 | NO | **PK** | Código del detalle pedido |
| Cantidad\_plato | Int | 4 | NO |  | Cantidad de producto |
| Precio | Float | 4 | No |  | Precio del detalle\_pedido |
| Codigo\_pedido | int | 4 | no |  | Codigo del pedido al que pertenece |
| Codigo\_plato | Int | 4 | si |  | Código del plato |
| Codigo\_contenido | int | 4 | Si | PK/FK | Código del contenido |
| Codigo\_bebida | Int | 4 | Si | PK/FK | Código de la bebida |

**DETALLE\_PRESA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTOS | TIPO DE DATOS | AMPLITUD(BYTE) | NULO | LLAVE | DESCRIPCION |
| Cod\_producto | Int | 4 | No | PK/FK | Código del producto |
| Cod\_  detalle\_pedido | Int | 4 | No | PK/FK | Código del detalle producto |
| Cantidad | Int | 4 | No |  | Cantidad de los pollos en presa |

**INGRESO\_EGRESO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTO | TIPOD DE DATOS | AMPLITUD(BYTE) | NULO | LLAVE | DESCRIPCION |
| Código | Int | 4 | NO | PK | Llave primaria y código del ingreso\_egreso |
| Fecha\_Hora | datetime | 8 | No |  | El tiempo del ingreso y egreso |
| Tipo | char | 1 | No |  | El tipo, i si es ingreso, e si es egreso, y c si es cantidad inicial |
| Nombre\_usuario | Varchar(15) | 1 a 15 | Si | FK | Llave foránea del usuario que hace el ingreso o egreso |

**DETALLE\_INGRESO\_EGRESO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTO | TIPO DE DATOS | AMPLITUD(BYTE) | NULO | LLAVE | DESCRIPCION |
| Cod\_ingreso  \_egreso | Int | 4 | No | PK/FK | Llave foránea del código |
| Cod\_producto | Int | 4 | No | PK/FK | Llave foránea del producto |
| Cantidad | Int | 4 | No |  | Cantidad de ingreso y egreso |

**STOCK**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTO | TIPO DE DATOS | AMPLITUD(BYTE) | NULO | LLAVE | DESCRIPCION |
| Cod\_stock | Int | 4 | No | PK | Llave primaria y código del stock |
| Cod\_producto | Int | 4 | No | Pk | Llave foránea del producto |
| Cantidad | Int | 4 | No |  | Cantidad inicial del stock |
| Ingreso | Int | 4 | No |  | Ingreso que hace del stock luego de la cantidad inicial |
| Stock | Int | 4 | No |  | Stock disponible del producto |
| Fecha\_hora | datetime | 8 | No |  | Fecha y hora de la ultima actualizacion |

**BITACORA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUTO | TIPO DE DATOS | AMPLITUD(BYTE) | NULO | LLAVE | DESCRIPCION |
| Codigo | Int | 4 | No | PK | Llave primaria y código de la bitácora |
| Fecha\_Hora | Datetime | 8 | No |  | Fecha y hora en que se hace esa bitacora |
| Username | Varchar(15) | 1 a 15 | No | FK | Llave foránea con el usuario |

### Script

create database polleria;

use polleria;

create table persona(

ci varchar(10) not null,

nombre varchar(20) not null,

apellido varchar(40) not null,

direccion varchar(40) not null,

sexo char not null,

tipo\_p char not null,

primary key(ci)

);

create table celular\_persona(

ci varchar(10) not null,

celular int,

primary key(ci, celular),

foreign key(ci) references persona(ci)

on delete no action on update no action

);

create table proveedor(

ci varchar(10) not null,

nombre\_negocio varchar(30) not null,

direccion\_negocio varchar(40),

primary key(ci),

foreign key(ci) references persona(ci)

on delete cascade

on update cascade

);

create table cargo(

cod\_cargo int not null,

descripcion varchar(30) not null,

sueldo float not null,

perfil\_usuario boolean ,

primary key(cod\_cargo)

);

create table trabajador (

ci varchar(10) not null,

fecha\_inicio datetime not null,

estado boolean not null,

cod\_cargo int not null,

primary key(ci),

foreign key(ci) references persona(ci)

on delete cascade

on update cascade,

foreign key(cod\_cargo) references cargo(cod\_cargo)

on delete cascade

on update cascade

);

create table usuario(

username varchar(20) not null,

psswrd varchar(30) not null,

email varchar(50) not null,

nivel\_acceso int not null,

ci varchar(10) not null,

primary key(username),

foreign key(ci) references trabajador(ci)

on delete cascade

on update cascade

);

create table producto(

codigo int not null,

nombre varchar(30) not null,

tipo\_producto char not null,

primary key(codigo)

);

create table nota\_de\_compra(

codigo int not null AUTO\_INCREMENT,

fecha\_hora datetime not null,

total float,

username varchar(15) not null,

cod\_proveedor varchar(10) not null,

primary key(codigo),

foreign key(username) references usuario(username)

on delete cascade

on update cascade,

foreign key(cod\_proveedor) references proveedor(ci)

on delete cascade

on update cascade

);

create table tipo(

cod\_tipo int not null,

descripcion varchar(40) not null,

abreviatura varchar(5) not null,

primary key(cod\_tipo)

);

create table detalle\_compra(

cod\_nota\_compra int not null,

cod\_producto int not null,

cantidad\_producto float not null,

precio\_total float not null ,

precio\_unitario float,

descripcion varchar(40),

cod\_tipo int not null,

primary key(cod\_producto,cod\_nota\_compra),

foreign key(cod\_producto) references producto(codigo)

on delete cascade

on update cascade,

foreign key(cod\_nota\_compra) references nota\_de\_compra(codigo)

on delete cascade

on update cascade,

foreign key(cod\_tipo) references tipo(cod\_tipo)

on delete cascade

on update cascade

);

create table lote(

cod\_lote int not null,

fecha\_vencimiento datetime not null,

fecha\_elaboracion datetime,

cantidad int not null,

codigo\_producto\_l int not null,

codigo\_nota\_compra\_l int not null,

primary key(cod\_lote),

foreign key(codigo\_producto\_l) references detalle\_compra(cod\_producto)

on delete cascade

on update cascade,

foreign key(codigo\_nota\_compra\_l) references detalle\_compra(cod\_nota\_compra)

on delete cascade

on update cascade

);

create table bebida(

codigo\_producto int not null,

estado boolean not null,

precio float not null,

primary key(codigo\_producto),

foreign key(codigo\_producto) references producto(codigo)

on delete cascade

on update cascade

);

create table pedido(

cod\_pedido int not null AUTO\_INCREMENT,

fecha\_hora datetime not null,

costo\_total float not null,

llevar boolean not null,

username varchar(15) not null,

primary key(cod\_pedido),

foreign key(username) references usuario(username)

on delete cascade

on update cascade

);

create table plato(

cod\_plato int not null,

nombre varchar(20) not null,

precio float not null,

primary key(cod\_plato)

);

create table contenido(

cod\_contenido int not null,

nombre varchar(20) not null,

precio float not null,

estado boolean not null,

primary key(cod\_contenido)

);

create table contenido\_producto(

cod\_plato int not null,

cod\_contenido int not null,

cantidad int not null,

primary key(cod\_plato,cod\_contenido),

foreign key(cod\_plato) references plato(cod\_plato)

on delete cascade

on update cascade,

foreign key(cod\_contenido) references contenido(cod\_contenido)

on delete cascade

on update cascade

);

create table detalle\_pedido(

codigo int not null,

cantidad\_plato int not null ,

precio float not null,

codigo\_pedido int not null,

codigo\_plato int,

codigo\_contenido int,

codigo\_bebida int ,

primary key(codigo),

foreign key(codigo\_pedido) references pedido(cod\_pedido)

on delete cascade

on update cascade ,

foreign key(codigo\_plato) references plato(cod\_plato)

on delete cascade

on update cascade,

foreign key(codigo\_bebida) references bebida(codigo\_producto)

on delete cascade

on update cascade,

foreign key(codigo\_contenido) references contenido(cod\_contenido)

on delete cascade

on update cascade

);

create table detalle\_presa(

cod\_producto int not null,

cod\_detalle\_pedido int not null,

cantidad int not null,

primary key(cod\_producto,cod\_detalle\_pedido),

foreign key(cod\_producto) references producto(codigo)

on delete cascade

on update cascade,

foreign key(cod\_detalle\_pedido) references detalle\_pedido(codigo)

on delete cascade

on update cascade

);

create table ingreso\_egreso(

codigo int not null auto\_increment,

fecha\_hora datetime not null,

tipo char not null,

nombre\_usuario varchar(15) not null,

primary key(codigo) ,

foreign key(nombre\_usuario) references usuario(username)

on delete cascade

on update cascade

);

create table detalle\_ingreso\_egreso(

codigo\_ingreso\_egreso int not null,

codigo\_producto int not null,

cantidad int not null,

primary key(codigo\_ingreso\_egreso,codigo\_producto),

foreign key(codigo\_ingreso\_egreso) references ingreso\_egreso(codigo)

on delete cascade

on update cascade,

foreign key(codigo\_producto) references producto(codigo)

on delete cascade

on update cascade

);

create table stock(

cod\_stock int not null auto\_increment,

cod\_producto int not null,

cantidad int not null,

ingreso int not null,

stock int not null,

fecha\_hora datetime not null,

primary key(cod\_stock,cod\_producto),

foreign key(cod\_producto) references producto(codigo)

on delete cascade

on update cascade

);

create table bitacora(

codigo int not null auto\_increment,

fecha\_hora datetime not null,

descripcion varchar(30) not null,

username varchar(15) not null,

primary key(codigo),

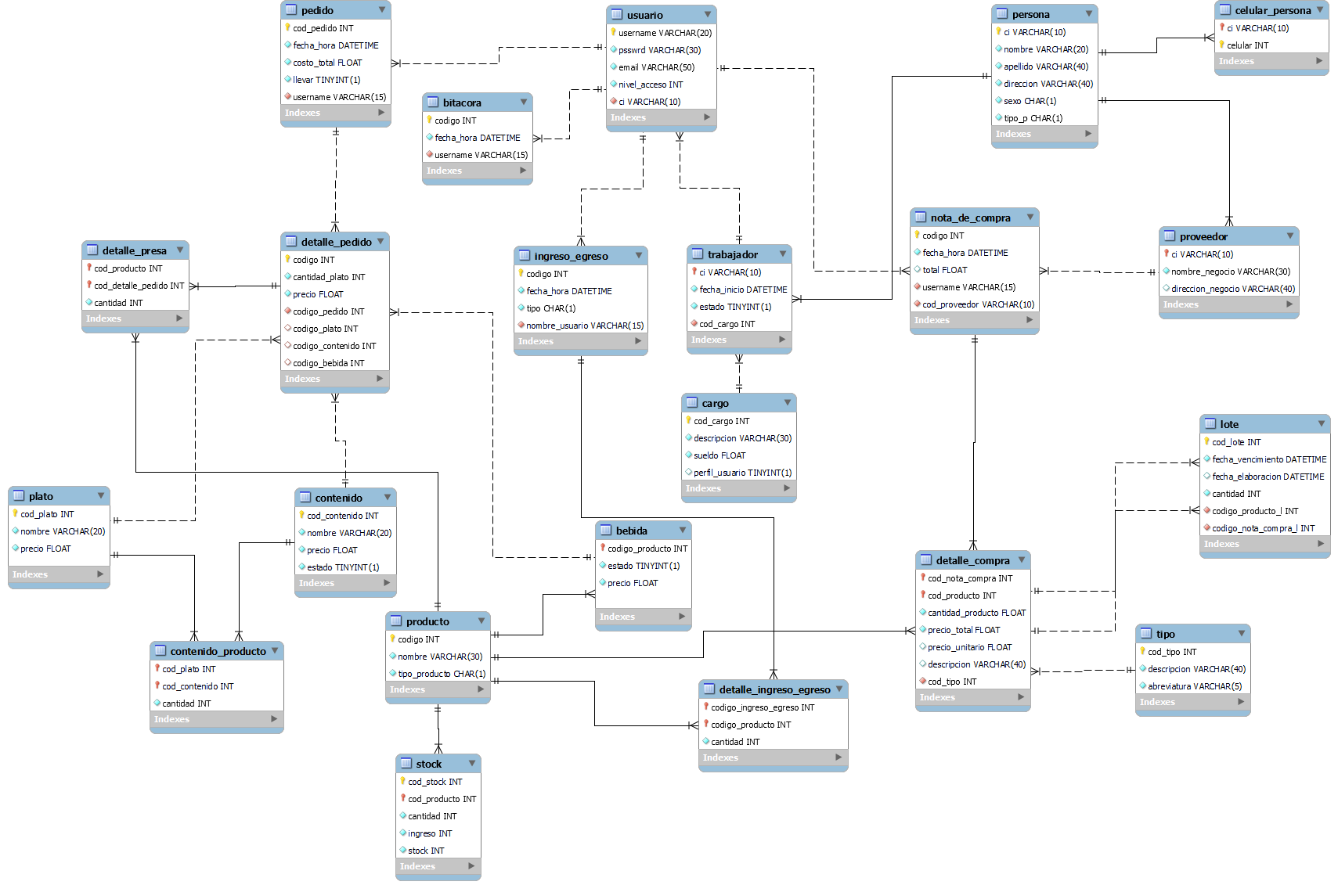
foreign key(username) references usuario(username)

on delete cascade

on update cascade

);

### Diagrama relacional



### Actualización de tuplas

insert into persona values('1112354c','Joaquin','Chumacero','Plan 3000','m','p');

insert into persona values('1112659A','Alejandra','Mercedes','Av. Plan 3000','f','t');

insert into persona values('8945414L','Alex','Farfán ','Av. Busch','m','p');

insert into persona values('54548455','Alejandra','Soto Aguilar','Av. Arroyito','f','p');

insert into persona values('51545454','Max','Fernandez Céspedes ','Av. Cañoto frente av. Busch','m','t');

insert into persona values('98745451','Romina','Nahir Durán','Av. Plan 3000','f','t');

insert into persona values('9897156','Juan','Sans','Av. Busch','m','p');

insert into persona values('4568798','Martha','Coehlo','Av. Busch','f','t');

-- celular\_persona(

-- ci varchar(10), celular int

insert into celular\_persona values('1112354c',72737292);

insert into celular\_persona values('1112659A',72393182);

insert into celular\_persona values('8945414L',6983323);

insert into celular\_persona values('54548455',62738281);

insert into celular\_persona values('51545454',62778213);

insert into celular\_persona values('98745451',73628732);

insert into celular\_persona values('9897156',77263828);

insert into celular\_persona values('4568798',66253652);

insert into celular\_persona values('4568798',61290393);

-- proveedor

-- ci varchar(10), nombre\_negocio varchar(20), direccion\_negocio varchar(40)

insert into proveedor values('1112354c','polleria san juan','av. plan 3 mil');

insert into proveedor values('8945414L','coca cola','Calle Israel mendia #4');

insert into proveedor values('54548455','distr. vasquez','calle israel mendia #9');

insert into proveedor values('9897156','verduleria buenas','calle israel mendia #8');

-- cargo(

-- cod\_cargo int, descripcion varchar(30), sueldo float not null, perfil\_usuario boolean

insert into cargo values(1,'Administrador',2000,true);

insert into cargo values(2,'Cajero',2000,true);

insert into cargo values(3,'Mesero',1850,true);

insert into cargo values(4,'Cocinero',1850,true);

-- trabajador

-- ci varchar(10), fecha\_inicio datetime, estado boolean, cod\_cargo int

insert into trabajador values('1112659A','2021-04-02 16:06:00',true,1);

insert into trabajador values('51545454','2021-04-02 16:08:00',true,2);

insert into trabajador values('98745451','2021-04-02 16:11:00',true,3);

insert into trabajador values('4568798','2021-06-05 16:01:00',true,4);

-- usuario

-- username varchar(15), psswrd varchar(20), email varchar(50),nivel\_acceso int

-- ci varchar(10)

insert into usuario values('Alejandrita2000','Aleja9878','alejandra@gmail.com',1,'1112659A');

insert into usuario values('Max007','45454sasa','Maxbell32\_2@gmail.com',2,'51545454');

insert into usuario values('Rominita21','romini121','romi@gmail.com',3,'98745451');

-- producto

-- codigo int, nombre varchar(20), tipo\_producto char

insert into producto values(10001,'Arroz','i');

insert into producto values(10002,'Papa','i');

insert into producto values(10003,'Cebollita Verde','i');

insert into producto values(10004,'Mortadela','i');

insert into producto values(10005,'Zanahoria','i');

insert into producto values(10006,'Aceite','i');

insert into producto values(10007,'Pimenton','i');

insert into producto values(10008,'Carbón','i');

insert into producto values(10009,'Salsa Soya','i');

insert into producto values(10010,'Pollo Crudo','i');

insert into producto values(10011,'Pierna','p');

insert into producto values(10012,'Pecho','p');

insert into producto values(10013,'Contra','p');

insert into producto values(10014,'Ala','p');

insert into producto values(10015,'Coca Cola mini','b');

insert into producto values(10016,'Coca Cola peque','b');

insert into producto values(10017,'Coca Cola retornable personal','b');

insert into producto values(10018,'Coca Cola personal','b');

insert into producto values(10019,'Coca Cola retornable 1 Lt','b');

insert into producto values(10020,'Coca Cola 2 Lt','b');

insert into producto values(10021,'Coca Cola 3 Lt','b');

insert into producto values(10022,'Fanta mini','b');

insert into producto values(10023,'Fanta peque','b');

insert into producto values(10024,'Fanta retornable personal','b');

insert into producto values(10025,'Fanta personal','b');

insert into producto values(10026,'Fanta 2 Lt','b');

insert into producto values(10027,'Fanta 3 Lt','b');

insert into producto values(10028,'Sprite mini','b');

insert into producto values(10029,'Sprite personal','b');

insert into producto values(10030,'Sprite retornable 1 Lt','b');

insert into producto values(10031,'Sprite 2 Lt','b');

insert into producto values(10032,'Sprite 3 Lt','b');

insert into producto values(10033,'Tropifrut popular','b');

insert into producto values(10034,'Tropifrut 1 Lt','b');

insert into producto values(10035,'Popular mendocina','b');

insert into producto values(10036,'Agua Vital personal','b');

insert into producto values(10037,'Agua Vital 1 Lt','b');

insert into producto values(10038,'Agua Vital 2 Lt','b');

insert into producto values(10039,'Aquario 2 Lt','b');

insert into producto values(10040,'Aquario 3 Lt','b');

-- nota\_de\_compra

-- codigo int, fecha\_hora datetime, total float, usernameU varchar(15), cod\_proveedor varchar(10)

insert into nota\_de\_compra values(101,'2021-06-12 08:30:23',3225,'Alejandrita2000','1112354c');

insert into nota\_de\_compra values(102,'2021-06-12 09:00:00',680,'Alejandrita2000','9897156');

insert into nota\_de\_compra values(103,'2021-06-11 09:25:21',1636,'Alejandrita2000','8945414L');

select \* from nota\_de\_compra;

-- tipo

-- cod\_tipo int, descripcion varchar(40), abreviatura varchar(5)

insert into tipo values(1001,'caja','cj');

insert into tipo values(1002,'kilogramo','kg');

insert into tipo values(1003,'quintal','qq');

insert into tipo values(1004,'arroba','@');

insert into tipo values(1005,'gramo','g');

insert into tipo values(1006,'paquete','pq');

insert into tipo values(1007,'unidad','u');

insert into tipo values(1008,'docena','dc');

-- detalle\_compra

-- cod\_nota\_compra int, cod\_producto int,cantidad\_producto float, precio\_total float,

-- precio\_unitario float, descripcion varchar(40), cod\_tipo int

insert into detalle\_compra values(101,10010,30,3225,107.5,'',1001);

insert into detalle\_compra values(102,10001,2,500,250,'',1006);

insert into detalle\_compra values(102,10006,1,100,100,'',1001);

insert into detalle\_compra values(102,10002,2,80,40,'',1004);

insert into detalle\_compra values(103,10015,4,80,20,'',1001);

insert into detalle\_compra values(103,10016,4,60,15,'',1006);

insert into detalle\_compra values(103,10017,4,92,23,'',1001);

insert into detalle\_compra values(103,10018,6,150,25,'',1006);

insert into detalle\_compra values(103,10019,2,90,45,'',1001);

insert into detalle\_compra values(103,10020,10,580,58,'',1006);

insert into detalle\_compra values(103,10021,8,584,73,'',1006);

-- lote

-- cod\_lote int, fecha\_vencimiento datetime, fecha\_elaboracion datetime,

-- cantidad int, codigo\_producto\_l int, codigo\_nota\_compra\_l int

insert into lote values(9001,'2021-12-12','2021-05-03',2,10015,3);

insert into lote values(9002,'2021-08-23','2021-02-02',2,10015,3);

insert into lote values(9003,'2021-09-20','2021-03-19',4,10016,3);

insert into lote values(9004,'2021-10-15','2021-04-23',4,10017,3);

insert into lote values(9005,'2021-10-21','2021-04-21',6,10018,3);

insert into lote values(9006,'2021-07-30','2021-01-15',2,10019,3);

insert into lote values(9007,'2021-08-10','2021-02-15',8,10020,3);

insert into lote values(9008,'2021-07-15','2021-01-15',2,10020,3);

insert into lote values(9009,'2021-12-21','2021-06-10',5,10021,3);

insert into lote values(9010,'2021-08-01','2021-02-20',5,10021,3);

-- bebida

-- codigo\_producto int, estado boolean not null, precio float

insert into bebida values(10015,true, 2);

insert into bebida values(10016,true,3);

insert into bebida values(10017,true,5);

insert into bebida values(10018,true,6);

insert into bebida values(10019,true,10);

insert into bebida values(10020,true,12);

insert into bebida values(10021,true,15);

insert into bebida values(10022,false,2);

insert into bebida values(10023,false,3);

insert into bebida values(10024,false,5);

-- pedido

-- cod\_pedido int, fecha\_hora datetime, costo\_total float

-- llevar boolean, username varchar(15)

insert into pedido values(1,'2021-06-13 11:30:20',56,true,'Max007');

insert into pedido values(2,'2021-06-13 11:38:30',10,false,'Max007');

insert into pedido values(3,'2021-06-13 11:42:12',54,false,'Max007');

insert into pedido values(4,'2021-06-13 11:45:32',40,true,'Max007');

insert into pedido values(5,'2021-06-13 11:50:23',60,true,'Max007');

insert into pedido values(6,'2021-06-13 11:55:25',38,true,'Max007');

insert into pedido values(7,'2021-06-13 12:14:54',140,false,'Max007');

insert into pedido values(8,'2021-06-13 12:23:34',102,false,'Max007');

insert into pedido values(9,'2021-06-13 11:34:21',144,false,'Max007');

insert into pedido values(10,'2021-06-13 13:32:53',30,false,'Max007');

insert into pedido values(11,'2021-06-13 13:34:32',180,false,'Max007');

insert into pedido values(12,'2021-06-13 13:38:45',100,false,'Max007');

-- create table plato(

-- cod\_plato int, nombre varchar(20), precio float

insert into plato values(11001,'Económico',10);

insert into plato values(11002,'Cuarto',18);

insert into plato values(11003,'Medio',40);

insert into plato values(11004,'Entero',60);

-- contenido

-- cod\_contenido int, nombre varchar(20), precio float, estado boolean

insert into contenido values(12001,'Porción Presa',7,true);

insert into contenido values(12002,'Porción papa',5,true);

insert into contenido values(12003,'Porción arroz',7,true);

insert into contenido values(12004,'Porción plátano maduro',5,true);

insert into contenido values(12005,'Porción fideo',5,true);

-- contenido\_producto(

-- cod\_plato int, cod\_contenido int, cantidad int

insert into contenido\_producto values(11001,12001,1);

insert into contenido\_producto values(11001,12002,1);

insert into contenido\_producto values(11001,12003,1);

insert into contenido\_producto values(11001,12004,1);

insert into contenido\_producto values(11002,12001,2);

insert into contenido\_producto values(11002,12002,1);

insert into contenido\_producto values(11002,12003,2);

insert into contenido\_producto values(11002,12004,1);

insert into contenido\_producto values(11003,12001,4);

insert into contenido\_producto values(11003,12002,4);

insert into contenido\_producto values(11003,12003,2);

insert into contenido\_producto values(11003,12004,2);

insert into contenido\_producto values(11004,12001,8);

insert into contenido\_producto values(11004,12002,6);

insert into contenido\_producto values(11004,12003,4);

insert into contenido\_producto values(11004,12004,4);

-- detalle\_pedido

-- codigo int, cantidad\_plato int, precio float, codigo\_pedido int

-- codigo\_plato int, codigo\_contenido int, codigo\_bebida int

insert into detalle\_pedido values(5001,2,20,1,11001,null,null);

insert into detalle\_pedido values(5002,2,36,1,11002,null,null);

insert into detalle\_pedido values(5003,1,10,2,11001,null,null);

insert into detalle\_pedido values(5004,3,54,3,11002,null,null);

insert into detalle\_pedido values(5005,1,40,4,11003,null,null);

insert into detalle\_pedido values(5006,1,60,5,11004,null,null);

insert into detalle\_pedido values(5007,2,20,6,11001,null,null);

insert into detalle\_pedido values(5008,1,18,6,11002,null,null);

insert into detalle\_pedido values(5009,5,50,7,11001,null,null);

insert into detalle\_pedido values(5010,5,90,7,11002,null,null);

insert into detalle\_pedido values(5011,3,30,8,11001,null,null);

insert into detalle\_pedido values(5012,4,72,8,11002,null,null);

insert into detalle\_pedido values(5013,8,144,9,11002,null,null);

insert into detalle\_pedido values(5014,3,30,10,11001,null,null);

insert into detalle\_pedido values(5015,10,180,11,11002,null,null);

insert into detalle\_pedido values(5016,10,100,12,11001,null,null);

-- detalle\_presa

-- cod\_producto int, cod\_detalle\_pedido int, cantidad int

insert into detalle\_presa values(10011,5001,1);

insert into detalle\_presa values(10013,5001,1);

insert into detalle\_presa values(10014,5002,1);

insert into detalle\_presa values(10012,5002,1);

insert into detalle\_presa values(10011,5003,1);

insert into detalle\_presa values(10011,5004,1);

insert into detalle\_presa values(10012,5004,3);

insert into detalle\_presa values(10013,5004,1);

insert into detalle\_presa values(10014,5004,1);

insert into detalle\_presa values(10011,5005,1);

insert into detalle\_presa values(10012,5005,1);

insert into detalle\_presa values(10013,5005,1);

insert into detalle\_presa values(10014,5005,1);

insert into detalle\_presa values(10011,5006,2);

insert into detalle\_presa values(10012,5006,2);

insert into detalle\_presa values(10013,5006,2);

insert into detalle\_presa values(10014,5006,2);

insert into detalle\_presa values(10014,5007,1);

insert into detalle\_presa values(10013,5007,1);

insert into detalle\_presa values(10012,5008,1);

insert into detalle\_presa values(10014,5008,1);

insert into detalle\_presa values(10012,5009,2);

insert into detalle\_presa values(10013,5009,2);

insert into detalle\_presa values(10011,5009,1);

insert into detalle\_presa values(10012,5010,5);

insert into detalle\_presa values(10011,5010,5);

insert into detalle\_presa values(10011,5011,1);

insert into detalle\_presa values(10014,5011,1);

insert into detalle\_presa values(10013,5011,1);

insert into detalle\_presa values(10012,5012,4);

insert into detalle\_presa values(10011,5012,2);

insert into detalle\_presa values(10014,5012,2);

insert into detalle\_presa values(10012,5013,8);

insert into detalle\_presa values(10011,5013,4);

insert into detalle\_presa values(10014,5013,4);

insert into detalle\_presa values(10011,5014,2);

insert into detalle\_presa values(10014,5014,1);

insert into detalle\_presa values(10012,5015,10);

insert into detalle\_presa values(10011,5015,5);

insert into detalle\_presa values(10014,5015,5);

insert into detalle\_presa values(10011,5016,3);

insert into detalle\_presa values(10013,5016,3);

insert into detalle\_presa values(10012,5016,2);

insert into detalle\_presa values(10014,5016,2);

### Consultas

-- 1.- Mostrar la cantidad total de pedidos de todo un día

select count(\*) as Cantidad\_Pedidos from pedido

where date(fecha\_hora)=curdate();

-- 2.- Listar la cantidad de trabajadores según cargo

select count(trabajador.ci), cargo.descripcion from trabajador, cargo

where trabajador.cod\_cargo=cargo.cod\_cargo

group by cargo.descripcion;

-- 3.- Mostrar todas las bebidas Coca Cola con sus precio de venta

select producto.nombre as Nombre\_Producto, bebida.precio as Precio\_Venta from producto, bebida

where producto.codigo=bebida.codigo\_producto and producto.nombre like 'Coca Cola%';

-- 4.- Mostrar las cantidades de presas pedidas, ordenar de mayor a menor

select producto.nombre as Presa, sum(detalle\_presa.cantidad) as cantidad from producto, detalle\_presa, detalle\_pedido

where producto.tipo\_producto='p' and producto.codigo=detalle\_presa.cod\_producto

and detalle\_pedido.codigo=detalle\_presa.cod\_detalle\_pedido

group by producto.nombre

order by sum(detalle\_presa.cantidad) desc;

-- 5.- Mostrar todos los trabajadores hombres

select persona.ci, persona.nombre, persona.apellido from persona

where persona.tipo\_p='t' and persona.sexo='m';

-- 6.- Mostrar todos los proveedores hombres

select persona.ci, persona.nombre, persona.apellido from persona

where persona.tipo\_p='p' and persona.sexo='m';

-- 7.- Mostrar la cantidad de nota\_de\_compra del mes pasado

/\*select count(\*) as cantidad\_Compra from nota\_de\_compra

where month(fecha\_hora)=month(date(curdate()))-1;

\*/

select count(\*) as cantidad\_Compra from nota\_de\_compra

where date\_format(fecha\_hora, "%Y-%m") = date\_format(now() - interval 1 month, "%Y-%m");

-- 8.- Mostrar el costo total más bajo de un pedido del día de hoy

select \* from pedido;

select min(costo\_total) from pedido

where date(fecha\_hora)=curdate();

-- 9.- Mostrar los productos que tengan una cantidad mínima de 30 en Stock

select \* from stock;

select stock.cod\_stock, producto.nombre, stock.cantidad from stock, producto

where stock.cod\_producto=producto.codigo and stock.cantidad>=30;

-- 10.- Mostrar el costo\_Total de los pedidos del día

select sum(costo\_total) as Venta\_Día from pedido

where date(pedido.fecha\_hora)=curdate();

-- 11.- Mostrar la cantidad de cada platos que se vendío entre 2021-06-11 y 2021-06-15

select plato.nombre, sum(detalle\_pedido.cantidad\_plato) from plato, detalle\_pedido, pedido

where detalle\_pedido.codigo\_plato=plato.cod\_plato and pedido.cod\_pedido=detalle\_pedido.codigo\_pedido

and date(pedido.fecha\_hora)>='2021-06-11' and date(pedido.fecha\_hora)<='2021-06-15'

group by plato.nombre;

-- 12.- Mostrar el trabajador mas antiguo

select persona.nombre, min(trabajador.fecha\_inicio) from persona,trabajador

where persona.ci=trabajador.ci and persona.tipo\_p='t'

group by persona.nombre;

-- 13.- Mostrar todos los proveedores

select \* from persona

where persona.tipo\_p='p' ;

-- 14.- mostrar el pedido on mayor precio

select max(pedido.costo\_total) from pedido;

-- 15.- Mostrar todas las trabajadores mujeres

select \* from persona

where persona.tipo\_p='t' and persona.sexo='f';

-- 16.-Mostrar todos los proveedores mujeres

select \* from persona

where persona.tipo\_p='p' and persona.sexo='f';

-- 17.- Mostrar a una persona segun su apellido

select \* from persona

where persona.apellido='Sans';

-- 18.- Mostrar el costo total más alto de un pedido del día de hoy

select MAX(costo\_total) from pedido

where fecha\_hora between curdate() and

(select DATE\_SUB(CURDATE(), INTERVAL -1 DAY));

-- 19-.mostrar cuantos trabajadores tiene la rosticeria

select count( trabajador.ci) as 'Nro de trabajadores'

from trabajador;

-- 20.- Mostrar todos los pedidos del día

select \* from pedido

where fecha\_hora between curdate() and (select DATE\_SUB(CURDATE(), INTERVAL -1 DAY));

-- 21.- Mostrar los bebidas que tengan un precio menor a 10

select \* from bebida;

select producto.codigo, producto.nombre, bebida.precio from bebida,producto

where bebida.codigo\_producto=producto.codigo and bebida.precio<10;

-- 22.- Mostrar el costo\_total de nota\_Compra de un més y año dado

select \* from nota\_de\_compra;

select nota\_de\_compra.codigo,nota\_de\_compra.total from nota\_de\_compra

where month(date(fecha\_hora))=4 and year(date(fecha\_hora))=2021;

-- 23.- Mostrar la cantidad bebidas y de presas que se vendió en el día

select sum(detalle\_pedido.codigo\_bebida) as Cantidad\_bebida, sum(detalle\_presa.cantidad) as Cantidad\_Presas

from bebida, detalle\_presa, detalle\_pedido, producto, pedido

where pedido.cod\_pedido = detalle\_pedido.codigo\_pedido

and bebida.codigo\_producto = producto.codigo and detalle\_presa.cod\_producto = producto.codigo

and detalle\_pedido.codigo\_bebida = bebida.codigo\_producto

and detalle\_presa.cod\_detalle\_pedido = detalle\_pedido.codigo and pedido.fecha\_hora = now()

-- 24.- Mostrar la cantidad de bebidas y presas que se vendío entre dos fechas dadas

select sum(detalle\_pedido.codigo\_bebida) as Cantidad\_bebida, sum(detalle\_presa.cantidad) as Cantidad\_Presas

from bebida, detalle\_presa, detalle\_pedido, producto, pedido

where pedido.cod\_pedido = detalle\_pedido.codigo\_pedido

and bebida.codigo\_producto = producto.codigo and detalle\_presa.cod\_producto = producto.codigo

and detalle\_pedido.codigo\_bebida = bebida.codigo\_producto

and detalle\_presa.cod\_detalle\_pedido = detalle\_pedido.codigo and pedido.fecha\_hora between 'Fecha1' and 'Fecha2'

-- 25.- Mostrar el proveedor mas antiguo

select proveedor.ci, proveedor.nombre\_negocio

from proveedor, nota\_de\_compra

where proveedor.ci = nota\_de\_compra.ci\_proveedor and nota\_de\_compra.fecha\_hora in (select min(nota\_de\_compra.fecha\_hora)

from proveedor, nota\_de\_compra

where proveedor.ci = nota\_de\_compra.ci\_proveedor)

-- 26.- Mostrar el ultimo proveedor al que se le compro

select proveedor.ci, proveedor.nombre\_negocio

from proveedor, nota\_de\_compra

where proveedor.ci = nota\_de\_compra.ci\_proveedor and nota\_de\_compra.fecha\_hora in (select max(nota\_de\_compra.fecha\_hora)

from proveedor, nota\_de\_compra

where proveedor.ci = nota\_de\_compra.ci\_proveedor)

-- 27.- Mostrar el proveedor al que más se le compra

select proveedor.nombre\_negocio,ci\_proveedor

from proveedor,nota\_de\_compra

where proveedor.ci =nota\_de\_compra.ci\_proveedor

group by proveedor.nombre\_negocio

order by ci\_proveedor desc

-- 28.- Mostrar los nombre de los trabajadores ordenados albafeticamente

select trabajador.ci, persona.nombre

from trabajador, persona

where trabajador.ci = persona.ci

group by trabajador.ci

order by persona.nombre ASC

-- 29.- Mostrar el costo\_Total de los pedido del dia

select sum(precio\_total)

from nota\_de\_compra , detalle\_compra ,producto

where detalle\_compra.cod\_nota\_compra = nota\_de\_compra.codigo and detalle\_compra.cod\_producto = producto.codigo and nota\_de\_compra.fecha\_hora = 'fecha'

-- 30.- Mostrar el costo\_Total de los pedido entre dos fechas dadas

select sum(precio\_total)

from nota\_de\_compra , detalle\_compra ,producto

where detalle\_compra.cod\_nota\_compra = nota\_de\_compra.codigo and detalle\_compra.cod\_producto = producto.codigo and nota\_de\_compra.fecha\_hora = 'fecha' -- between 'fecha1' and 'fecha2'

### Procedimientos almacenados

use polleria;

-- 1 muestra todos los trabajadores

delimiter //

create procedure mostrar\_trabajadores()

begin

select persona.nombre,trabajador.ci from trabajador, persona

where trabajador.ci = persona.ci ;

end //

delimiter ;

-- llamar al procedimiento

call mostrar\_trabajadores();

-- 2 mostrar el stock de presas

delimiter //

create procedure cantidad\_presas()

begin

select stock.stock, producto.nombre

from stock, producto

where producto.codigo = stock.cod\_producto and producto.tipo\_producto = 'p';

end //

delimiter ;

-- llamar al procedimiento

call cantidad\_presas();

-- 3 mostrar el stock de bebidas

delimiter //

create procedure cantidad\_bebidas()

begin

select stock.stock, producto.nombre

from stock, producto

where producto.codigo = stock.cod\_producto and producto.tipo\_producto = 'b';

end //

delimiter ;

-- llamar al procedimiento

call cantidad\_bebidas();

### Disparadores

-- trigger para descontar bebidas

delimiter //

create trigger act\_bebida

before insert

on detalle\_pedido

for each row

begin

update stock set stock.stock = stock.stock - new.cantidad\_plato

where new.codigo\_bebida= stock.cod\_producto;

end //

delimiter ;

# ISHIKAWA

## Identificar Problemas

### Lista de problemas

**P1** Precario manejo de orden de pedidos, lo que genera desorden en las entregas

**P2** Perdida de tiempo en generar los reportes de ventas del día (se realiza de manera manual el conteo de ticket).

**P3** Lentitud en la atención al cliente, es causado por que se realizan los pedidos manualmente

**P4** Deficiencia en el control de menú disponible

**P5** Insuficiente personal para la atención al cliente, lo que genera largas filas

**P6** Deficiente control del estado de los productos en las compras, puede o no generar factura

**P7** Déficit de gestión de productos (materia prima).

**P8** Incertidumbre en el manejo de las existencias del stock disponible

**P9** Dificultad en identificar los productos que están próximos a caducar

**P10** Falta de registro de los proveedores que suministran con productos al restaurante.

**P11** Demora en realizar los pedidos de suministro a los proveedores, ya que, al no saber la existencia de stock disponible, no se puede hacer un pedido con anticipación.

**P12** Perdida de información acerca de los productos que se compraron, ya que en ocasiones no se emitieron notas de compra.

**P13** Inexistencia de registro de las acciones en caja, no conocer las acciones del cajero.

**P14** Inadecuado manejo de la información, los datos se guardan en un cuaderno de forma manual

**P15** Deficiente gestión del personal en el restaurante

**P16** Déficit en la atención al cliente por la falta de capacitación del personal

**P17** Perdida de información de los movimientos realizados en caja y administración.

### Depurar la lista de problemas

**P1** Precario manejo de orden de pedidos, lo que genera desorden en las entregas

**P2** Perdida de tiempo en generar los reportes de ventas del día (se realiza de manera manual el conteo de ticket).

**P3** Lentitud en la atención al cliente, es causado por que se realizan los pedidos manualmente

**P4** Deficiencia en el control de menú disponible

**P7** Déficit de gestión de productos (materia prima).

**P8** Incertidumbre en el manejo de las existencias del stock disponible

**P10** Falta de registro de los proveedores que suministran con productos al restaurante.

**P12** Perdida de información acerca de los productos que se compraron, ya que en ocasiones no se emitieron notas de compra.

**P13** Inexistencia de registro de las acciones en caja, no conocer las acciones del cajero.

**P14** Inadecuado manejo de la información, los datos se guardan en un cuaderno de forma manual

**P15** Deficiente gestión del personal en el restaurante

**P17** Perdida de información de los movimientos realizados en caja y administración.

### Propietarios de Problemas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Propietarios/  Problemas | Cliente | Proveedor | Administrador | cajero |
| P1 |  |  |  |  |
| P2 |  |  |  |  |
| P3 |  |  |  |  |
| P4 |  |  |  |  |
| P7 |  |  |  |  |
| P8 |  |  |  |  |
| P10 |  |  |  |  |
| P12 |  |  |  |  |
| P13 |  |  |  |  |
| P14 |  |  |  |  |
| P15 |  |  |  |  |
| P17 |  |  |  |  |

**P1** Precario manejo de orden de pedidos, lo que genera desorden en las entregas

**P2** Perdida de tiempo en generar los reportes de ventas del día (se realiza de manera manual el conteo de ticket).

**P3** Lentitud en la atención al cliente, es causado por que se realizan los pedidos manualmente

**P4** Deficiencia en el control de menú disponible

**P7** Déficit de gestión de productos (materia prima).

**P8** Incertidumbre en el manejo de las existencias del stock disponible

**P10** Falta de registro de los proveedores que suministran con productos al restaurante.

**P12** Perdida de información acerca de los productos que se compraron, ya que en ocasiones no se emitieron notas de compra.

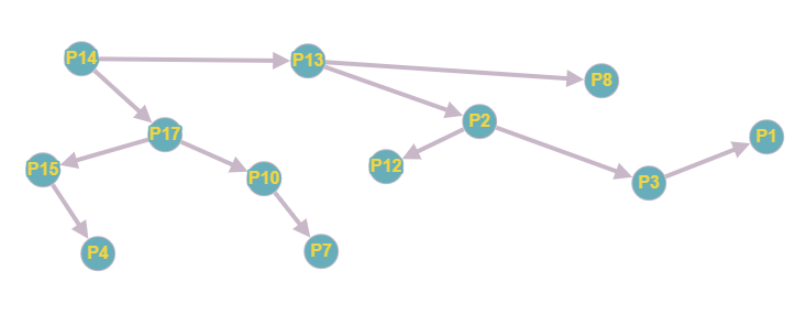
**P13** Inexistencia de registro de las acciones en caja, no conocer las acciones del cajero.

**P14** Inadecuado manejo de la información, los datos se guardan en un cuaderno de forma manual

**P15** Deficiente gestión del personal en el restaurante

**P17** Perdida de información de los movimientos realizados en caja y administración.

### Análisis del Problemas



### Estimación y cuantificación del problema

|  |  |
| --- | --- |
| **Problemas** | **Cuantificación/justificación** |
| **P1** Precario manejo de orden de pedidos, lo que genera desorden en las entregas. | Al no tener un manejo adecuado de los pedidos realizados, genera errores en el nro de orden del pedido. |
| **P2** Perdida de tiempo en generar los reportes de ventas del día (se realiza de manera manual el conteo de ticket). | Al terminar la jornada provoca retraso y un no tan preciso control diario de las ventas. |
| **P3** Lentitud en la atención al cliente, es causado por que se realizan los pedidos manualmente. | Al no tener un sistematizado generador de pedidos, existe retraso en la toma de pedidos, causando molestías y lentitud. |
| **P4** Deficiencia en el control de menú disponible. | Al no contar con un stock o alguna referencia de los platos del día, pueden ocasionar errores en el menú que se ofrece a los clientes. |
| **P7** Déficit de gestión de productos (materia prima). | Al no contar con un control de los productos que se necesitaran para la preparación de los platos de comidas generan una mala preparación de los mismos. |
| **P8** Incertidumbre en el manejo de las existencias del stock disponible. | Al no contar con un Stock real que se actualice puede generar confusiones, y falta de coherencia al tomar la orden o al preparar el menú del día. |
| **P10** Falta de registro de los proveedores que suministran con productos al restaurante. | Al no tener un control de los proveedores este genera retrasos en la preparación, y pérdidas en la atención. |
| **P12** Perdida de información acerca de los productos que se compraron, ya que en ocasiones no se emitieron notas de compra. | Al no conocer los detalles de compra, no se podrá tener un control real de los ingresos y egresos del día. |
| **P13** Inexistencia de registro de las acciones en caja, no conocer las acciones del cajero. | Al no tener un sistema que marque los privilegios del cajero, éste puede ocasionar daños a la información del lugar. |
| **P14** Inadecuado manejo de la información, los datos se guardan en un cuaderno de forma manual. | Al no tener el manejo de información genera problema para llevar un adecuado control. |
| **P15** Deficiente gestión del personal en el restaurante. | Al no contar con un control adecuado del personal, no se podrá tener una correcta atención, y habrá retrasos y molestias. |
| **P17** Perdida de información de los movimientos realizados en caja y administración. | Al hacer cálculos, registros y facturas de forma manual uno puede cometer errores y pérdida de tiempo en realizar dicho procesos. |

### Alternativas de cambio

A1. Se puede evitar pérdida de tiempo al momento de registrar datos de los productos.

A2. Se puede evitar la mala interpretación de los datos escritos de cada producto y del menú de ese modo no se tendrá errores de transcripción, registrando directamente.

A3. Para evitar demoras en la búsqueda de los datos de los empleados, se puede manejar toda esa información digitalmente.

A4. Para evitar demoras en la búsqueda de los datos del menú y las porciones, se puede manejar toda esa información digitalmente.

A5. Para evitar la búsqueda manual de los productos comprados de los lotes uno por uno, se consultará digitalmente en las planillas de producto.

A6. Organizar y clasificar los productos del stock del menú, tipo de producto y fecha de ingreso; siguiendo un sistema estándar decodificación.

A6. Invertir recursos que manipulen el control de entrada y salida de los productos o porciones en el stock.

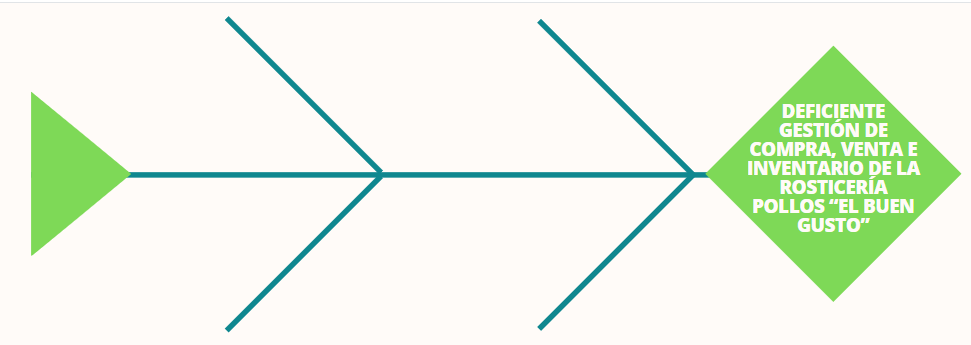
A7. Invertir recursos en infraestructura para mejorar la atención y evitarla incomodidad del cliente.

A8. Implementar un sistema de información para gestionar la venta, compra e inventario de los productos que se compran y del menú del día.

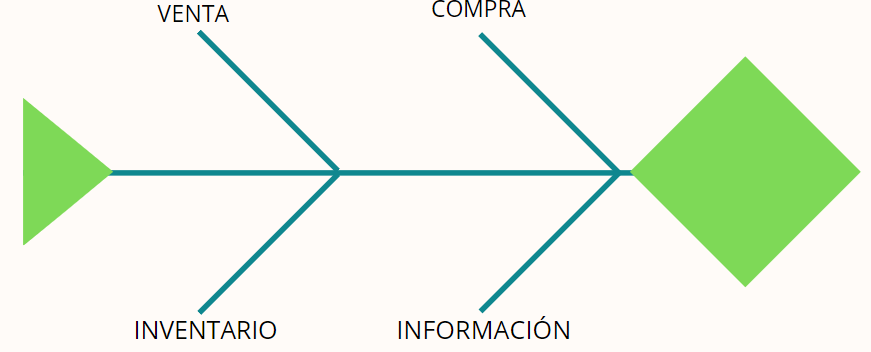
### Conclusión y Recomendación

En conclusión, la alternativa A8 es la que el cliente acepta como la más viable y la que recomendamos para resolver todos los problemas que se refieren a la compra, venta y la administración de inventario del menú del día sobre de la rosticería. Porque eso es lo principal para una mejor atención y mejor organización en general. Y una vez se estabilice en esta alternativa ya podrá optar a otras como es el de la infraestructura, por ejemplo.

### Diseñar el diagrama de Ishikawa



## Diseñar el diagrama de Ishikawa



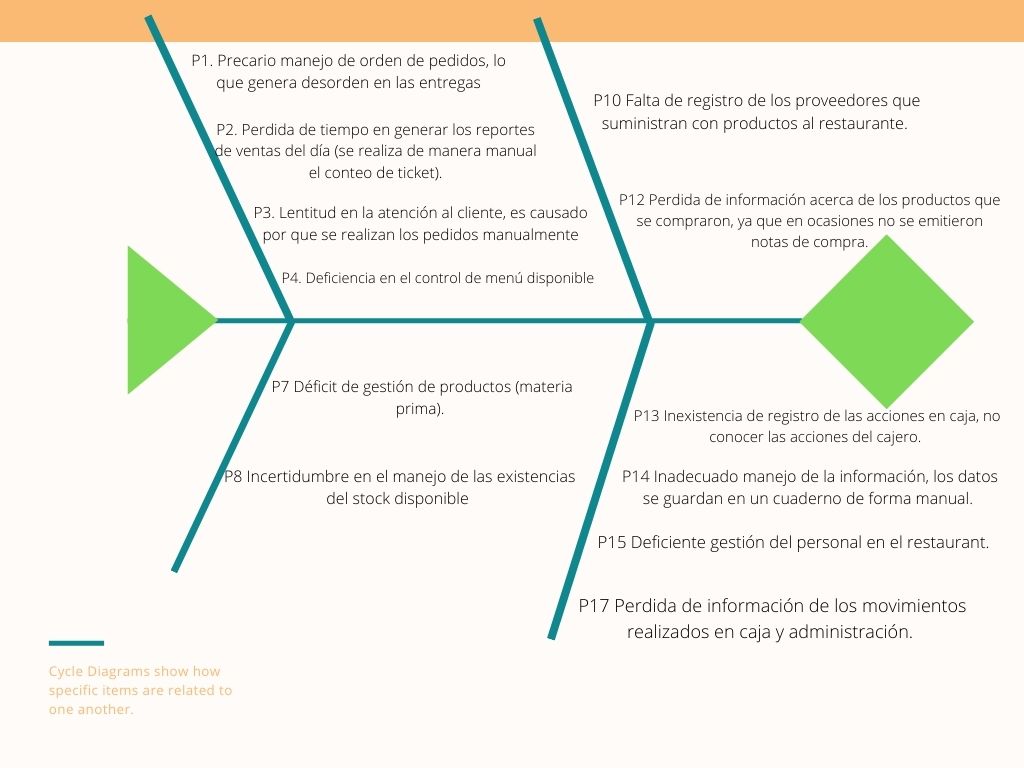
**Compra:** Los problemas **P10, P12,** pertenecen a la categoría compra, dichos problemas serán resueltos en su totalidad con el uso de nuestro sistema.

**Venta:** Los problemas **P1, P2, P3, P4,** pertenecen a la categoría venta, dichos problemas serán resueltos en su totalidad con el debido uso del sistema.

**Inventario:** Los problemas **P7, P8,** son problemas de la categoría inventario, la totalidad de estos problemas serán abarcados por nuestro sistema de modo que serán resueltos.

**Información:** Los problemas **P13, P14, P15, P17,** pertenecen a los problemas de registro y pérdida de datos.

## Identificar las causas



## Analizar y discutir del Diagrama

**Compra.:** Aquí podemos analizar los diferentes problemas que surgen en los pedidos y compras que se hacen, tales como: inadecuado manejo de los registros en la nota de compra, mal control de los egresos por compras realizadas.

**Venta:** En este campo se puede ver el mal seguimiento de las ventas realizadas, la demora al dar un ticket y un mal control de los registros de este campo.

**Inventario:** En este caso de tiene una pérdida de tiempo al buscar un producto que uno no sabe si le quedan o no al no llevar un buen registro de estos.

**Información:** En este campo al tener un mal manejo de información puede producirse perdidas de información valiosa tal como datos del proveedor, personal, demora al buscarlos, etc.

# MODELO DE NEGOCIO

## Diagrama de actividad proceso de compra de producto



## Diagrama de actividad proceso de venta de producto



## Diagrama de actividad proceso de inventariar stock



# ANEXOS

