

UNIVERSIDAD DEL VALLE DE GUATEMALA
CAMPUS CENTRAL
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
PROGRAMACIÓN ORIENTADA A OBJETOS



ENTREGA 2:

Identificación de requisitos funcionales, diagrama de clases y planificación

POR:

Julio Orellana: 24207

Ginebra Estrada: 24079

Sofia Batz: 241281

Ricardo Rodríguez: 18907

9 de agosto de 2024, Guatemala, Guatemala.



Requisitos funcionales

- Gestionar productos/materia prima. (Prioridad 1)
- Gestionar compras. (Prioridad 1)
- Gestionar ventas. (Prioridad 1)
- Gestionar empleados.
- Gestionar clientes.
- Gestionar almacén.
- Gestionar ubicación.
- Gestionar categorías de productos.
- Generar reporte de ventas.
- Gestionar Datos de manera persistente.

- Categoría
- Ubicación
- Producto
- Producto final
- Materia prima
- Empleado
- Vendedor
- Bodeguero
- Comprador
- Cliente
- Compra
- Venta
- Reporte



Identificación y Descripción de Clases necesarias:

class RawMaterial:

Esta clase se encarga de gestionar objetos de materia prima con los que cuenta la empresa.

abstract class Product:

Es una plantilla para crear productos, materiales, entre otros.

class FinalProduct:

Esta clase se encarga de gestionar los productos finales que tiene y distribuye la empresa, cada uno de estos objetos tendrá sus propias acciones y propiedades según corresponda.

class Storage:

Esta clase se va a encargar de gestionar el almacenamiento de los productos en sus respectivas ubicaciones, es decir, será la plantilla de almacén para la empresa. Esta clase tendrá acciones únicas de ella misma como buscar objetos (productos) en el almacén, filtrar productos, entre otras acciones relacionadas.

class Proveedor:

Esta clase servirá de plantilla para que la empresa pueda gestionar (crear) sus proveedores si los necesita para mantener una persistencia en el stock de los productos del inventario.

class Location:

Esta clase se encarga de gestionar las ubicaciones de los productos en el almacén de la empresa

class Store:

Esta clase se encargará de gestionar la compra y venta de los productos de la empresa, también esta clase estará encargada de gestionar los productos en la clase Storage permitiendo realizar actualizaciones en la misma. Esta clase contará con una lista de empleados y clientes ya que es la que administra el control general de la microempresa.

class Employee:

Esta clase servirá de plantilla para crear empleados y se encargará de gestionar la creación de empleados otorgándoles los privilegios correspondientes en la empresa para permitir realizar gestiones administrativas sin embargo estará restringida a otras funcionalidades para evitar fraude administrativo en la empresa, es decir sus atributos y métodos serán implementados según los requerimientos funcionales y necesidades de el usuario final.

UNIVERSIDAD DEL VALLE DE GUATEMALA

CAMPUS CENTRAL

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

PROGRAMACIÓN ORIENTADA A OBJETOS



class Customers:

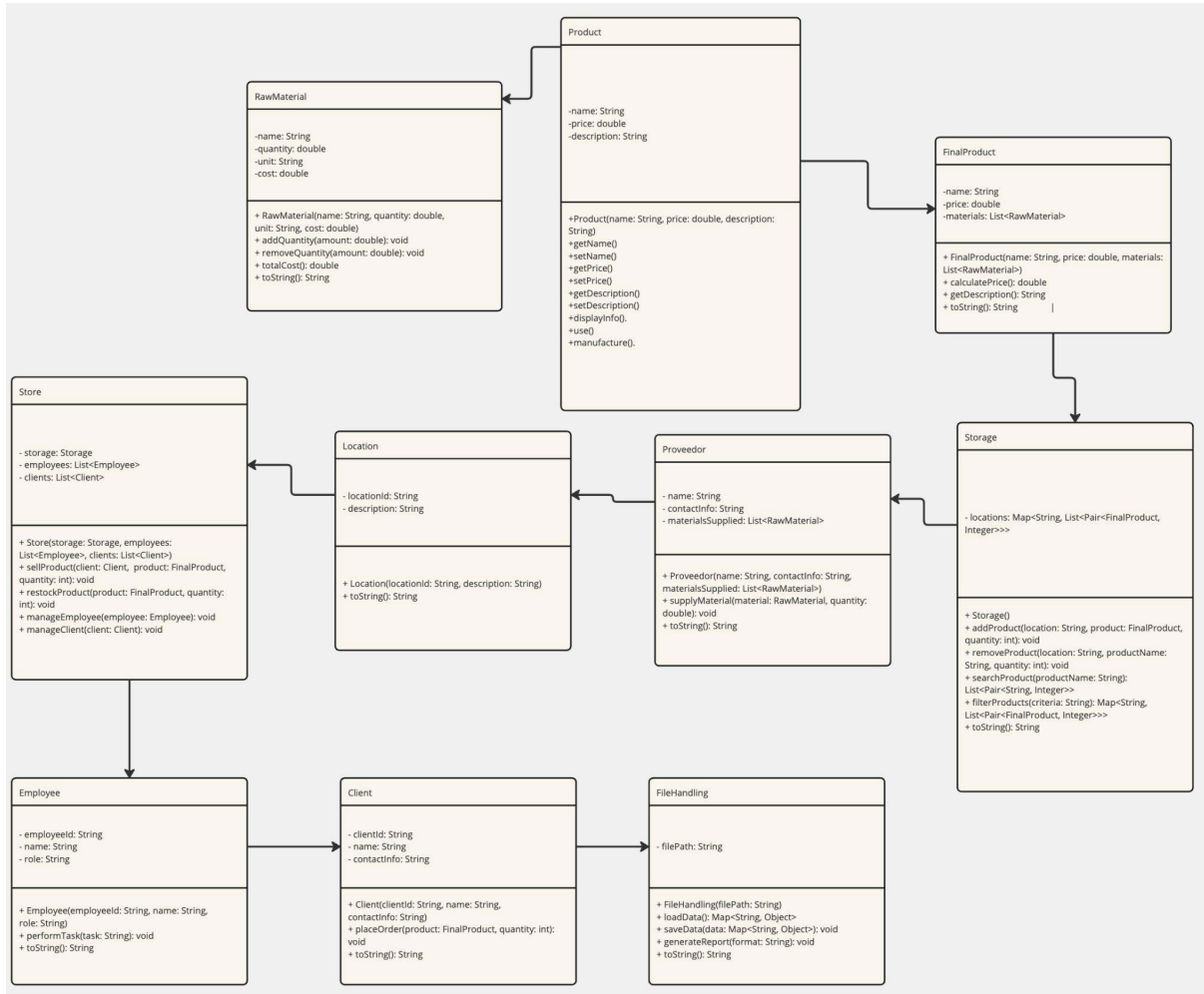
Esta clase servirá de plantilla para la gestión y creación de clientes en el sistema de la empresa, esta clase prácticamente se encargará de generar usuarios para poder realizar compra venta de productos llevando un control administrativo del inventario.

class FileHandling:

Esta clase se encargará de gestionar todas las interacciones con archivos del sistema, es decir, se encargará de la persistencia de datos, Esta clase tendrá acciones como cargar y sobrescribir la base de datos de la empresa (pendiente de definir si se usara una BD relacional o simplemente archivos csv con la información) esto sera definido según la crítica a recibir de parte de los usuarios finales. Esta clase también se encargará de generar reportes de la compra, venta o simplemente de el inventario mismo en archivos csv, pdf, txt según lo solicite el usuario final.



Diseño del Sistema:





Tareas de requisitos funcionales

Gestionar productos/materia prima

- Diseñar, hacer y probar la clase de productos.
- Diseñar, hacer y probar clases hijas de materia prima y de producto final.
- Hacer y probar un método para cambiar el nombre de un producto.
- Hacer y probar un atributo que permita diferenciar entre dimensionales.
- Hacer y probar un método que permita ver las existencias del producto.
- Hacer y probar un método que permita cambiar las existencias del producto.
- Hacer y probar un método que permita cambiar la ubicación del producto.
- Hacer y probar un método que verifique si es posible realizar una disminución de inventario.

Gestionar compras

- Diseñar, hacer y probar una clase de compras.
- Hacer y probar un método que permita aumentar las existencias de un producto.
- Hacer y probar un método que permita seleccionar varias materias primas.

Gestionar ventas

- Diseñar, hacer y probar una clase de ventas.
- Hacer y probar un método que permita disminuir las existencias de un producto.
- Hacer y probar un método que permita seleccionar varios productos finales.

Gestionar empleados

- Diseñar, hacer y probar una clase de empleados.
- Diseñar, hacer y probar clases hijas de vendedor, comprador y bodeguero.
- Hacer y probar un método para cambiar el nombre del empleado.
- Hacer y probar un método para validar la identidad del usuario.
- Hacer métodos para las clases hijas para que solo puedan cambiar ciertos tipos de productos según el tipo de empleado.

Gestionar clientes

- Diseñar, hacer y probar una clase de clientes.
- Hacer y probar un atributo que permita conocer qué productos compró, cuántos y la fecha.
- Hacer y probar un método que permita cambiar el nombre del cliente.
- Hacer y probar un método que permita agregar elementos al atributo que contiene el historial de compras.
- Hacer y probar un método que permita eliminar elementos del atributo que contiene el historial de compras.

UNIVERSIDAD DEL VALLE DE GUATEMALA
CAMPUS CENTRAL
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
PROGRAMACIÓN ORIENTADA A OBJETOS



Gestionar almacén

- Diseñar, hacer y probar una clase de almacén.
- Hacer y probar un atributo que permita guardar ubicaciones.
- Hacer y probar un método que permita cambiar el nombre del almacén.
- Hacer y probar un método que permita agregar una ubicación al almacén.
- Hacer y probar un método que permita quitar una ubicación al almacén.

Gestionar ubicación

- Diseñar, hacer y probar una clase de ubicación.
- Hacer y probar un atributo que permita el almacenamiento de productos.
- Hacer y probar un método que permita cambiar el nombre de la ubicación.
- Hacer y probar un método que permita agregar un producto a la ubicación.
- Hacer y probar un método que permita quitar un producto a la ubicación.

Gestionar categorías de productos

- Hacer y probar un atributo para la clase producto para la categoría.
- Hacer y probar un método que permita asignar la categoría a un producto.

Generar reporte de ventas

- Diseñar, hacer y probar una clase reporte.
- Hacer y probar atributos que permita conocer qué productos se han comprado, cuántos y la fecha.
- Hacer y probar un método que permita filtrar un producto específico.
- Hacer y probar un método que permita filtrar un rango de fechas específicas.
- Hacer y probar un método que genere una suma por cada producto seleccionado en las fechas seleccionadas.



Investigación de la tecnología disponible

Al momento de realizar sistemas de inventario debemos de tomar en cuenta que se necesitan diferentes tecnologías para realizar las funciones requeridas. Investigando se encontró que la herramienta Dolibarr está enfocada a pequeñas y medianas empresas. Es una herramienta que permite controlar las facturas, el inventario, las existencias y también la gestión de proyectos y las ventas. Con base a esta herramienta se analizó que se utiliza para resolver problemas similares a nuestro proyecto que es un sistema de gestión de inventarios para microempresas.

Tecnologías y herramientas que utilizaron en Dolibarr

PHP: Dolibarr está creada principalmente en PHP el cuál es un lenguaje de scripting muy popular para aplicaciones web porque es fácil de usar y tiene una gran comunidad de soporte

JavaScript: Se utiliza para crear interfaces de usuario dinámicas y mejorar la interacción del usuario en las aplicaciones. Para crear interfaces más complejas se pueden usar frameworks.

Python: No es el lenguaje principal de Dolibarr, este solo es utilizado para las soluciones de manejo de inventarios debido a su facilidad de uso y a la presencia de bibliotecas como Django que permiten un desarrollo de web rápido.

Además de lenguajes de programación, este sistema necesita bases de datos ya que por el tema de inventarios si se necesita un orden en los datos que se ingresan.

MySQL: Este tipo de aplicaciones se utilizan para manejar grandes volúmenes de datos relacionados a la microempresa

Herramientas de gestión y colaboración

Git: Sistema de control de versiones utilizado para gestionar el código fuente y colaborar entre desarrolladores.

Tecnologías Web

HTML/CSS: Son de las tecnologías base para construir interfaces de usuario en la web. CSS frameworks como Bootstrap pueden ayudar a desarrollar rápidamente interfaces responsivas.

AJAX: Ayuda para solicitudes asincrónicas al servidor y actualizar la interfaz de usuario sin necesidad de recargar la página

UNIVERSIDAD DEL VALLE DE GUATEMALA
CAMPUS CENTRAL
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
PROGRAMACIÓN ORIENTADA A OBJETOS



Dolibarr es una excelente opción para la gestión de inventarios y otras operaciones comerciales. El modelo de software libre permite la personalización y extensión de la comunidad. La interfaz modular permite agregar o remover funcionalidades de acuerdo con las necesidades de la empresa. Dolibarr puede adaptarse a medida que crece la empresa, este conjunto de herramientas y tecnologías es esencial para el desarrollo y el mantenimiento de una solución de manejo de inventarios que garantice que sea escalable, eficiente y adaptable a las necesidades cambiantes de las microempresas.

¿Cómo se utilizará de referencia en nuestro proyecto?

Este programa servirá de referencia para nuestro proyecto y nos ayudará a crear un sistema que permita que los usuarios ajusten las diferentes funcionalidades según las necesidades, debido a su arquitectura y capacidad de personalización. Se buscará una interfaz intuitiva para que los propietarios de las microempresas puedan administrar sus inventarios sin ningún problema. Para mejorar la experiencia se podría implementar funciones clave como el seguimiento de existencias en tiempo real y la gestión de múltiples ubicaciones, entre otras funcionalidades que hará nuestro proyecto.

Planificación y gestión

[Link de la tabla con la Planificación](#)



FORMULARIO DE GESTIÓN DE TIEMPO

Nombre: José Rodríguez
 Carné: 18907

| Fecha | Inicio | Fin | Tiempo Interrupción (min) | Tiempo trabajado (min) | Tarea | Comentarios |
|------------|--------|-------|---------------------------|------------------------|---|--------------------|
| 07/08/2024 | 11:30 | 12:10 | 0 | 40 | Definir tareas y posibles clases a usar | N/A |
| 07/08/2024 | 19:00 | 19:45 | 0 | 45 | Requisitos funcionales | N/A |
| 07/08/2024 | 21:50 | 22:30 | 5 | 40 | Requisitos funcionales | Vi un video |
| 09/08/2024 | 11:10 | 12:50 | 25 | 75 | Requisitos funcionales y planeación | Responder mensajes |
| 09/08/2024 | 14:00 | 14:30 | 0 | 30 | Planeación | N/A |

Nombre: Julio Orellana
 Carné: 24207

| Fecha | Inicio | Fin | Tiempo Interrupción (min) | Tiempo trabajado (min) | Tarea | Comentarios |
|----------|----------|----------|---------------------------|------------------------|--|---|
| 09/08/24 | 11:20 am | 12:50 am | 35 | 55 | Plantear las clases a usar en base a los requisitos funcionales. | Debate sobre cómo implementar cada clase. |
| 09/08/24 | 8:15 pm | 9:30 am | N/A | 75 | Identificación y descripción de las clases necesarias para el sistema. | N/A |

UNIVERSIDAD DEL VALLE DE GUATEMALA
CAMPUS CENTRAL
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
PROGRAMACIÓN ORIENTADA A OBJETOS



Nombre: Ginebra Estrada
 Carné: 24079

| Fecha | Inicio | Fin | Tiempo Interrupción (min) | Tiempo trabajado (min) | Tarea | Comentarios |
|----------|----------|----------|---------------------------|------------------------|---------------------------------------|---------------|
| 09/08/24 | 11:20 am | 12:30 am | 15 | 55 | Hacer la investigación con Sofia Batz | Fui al baño. |
| 09/08/24 | 9:00 pm | 9:50 am | 5 | 50 | UML | Vi el celular |

Nombre: Sofía Batz
 Carné: 241281

| Fecha | Inicio | Fin | Tiempo Interrupción (min) | Tiempo trabajado (min) | Tarea | Comentarios |
|------------|-----------|-----------|---------------------------|------------------------|--|-------------|
| 09/08/2024 | 11:20 a.m | 12:30 p.m | 0 | 50 | Realizar la investigación de las herramientas que podrían ser utilizadas | N/A |
| 09/08/2014 | 9:30 p.m | 9:45 | 0 | 15 | Definir mis tareas seleccionadas del proyecto con las horas que se usarán. | N/A |