| **Objetivos:**   1. Analizar principales requerimientos. 2. Diseñar diagrama de clases de la aplicación**.** 3. Realizar un ejercicio con interfaz gráfica de usuario (GUI) en Java. 4. Almacenar información en la memoria temporal. 5. Serializar la información de un requerimiento. 6. Implementar excepciones para manejo de archivos. | **Recursos:**   * Internet * Eclipse o Netbeans. * Libro: Introducción a la estructura de datos - Jorge Villalobos. * GitHub ejercicios: <https://github.com/ThiagoAndres/EstructuraDatos1-IV-UNIMAR> |
| --- | --- |
| **Proyecto:**  -Proyecto sobre Tienda de Zapatos  **Autoría:**   | Proyecto Curso: | Estructuras de datos I | | --- | --- | | Ejercicio: | Aplicación de Tienda de Zapatos | | Autores: | 1.Juan Diego Botina Realpe | | 2.Cristian David Landázuri Mambuscay | | 3.Janier David Acosta Morales | | Fecha realización: | 24 de abril 2024 |   **Requerimientos Funcionales:**   | Nombre | R1- Mostrar la información de un par de Zapatos | | --- | --- | | Resumen | Muestra al usuario el precio, talla, material, estilo y unidades vendidas de un par de zapatos del catálogo. | | Entradas | * Nombre del modelo de zapatos * Marca | | Resultados | Se muestra al usuario la información del par de zapatos. |  | Nombre | R2- Agregar un modelo de zapatos al catálogo | | --- | --- | | Resumen | Agrega un modelo de zapatos al catálogo sin zapatos | | Entradas | * Nombre del modelo de zapatos * Tipo de zapatos (deportivos, casuales, formales, etc.) * Nombre del archivo de imagen del modelo de zapatos en jpg o png | | Resultados | -Si toda la información es correcta, el modelo de zapatos se agrega al catálogo con la información del nombre, marca, tipo y nombre de la imagen.  -Si los datos proporcionados no son válidos (nombre del modelo, marca, tipo y/o imagen están vacíos), se muestra un mensaje de error y no se agrega. |  | Nombre | R3-Agregar un par de Zapatos a un modelo | | --- | --- | | Resumen | Agrega un nuevo par de zapatos con su información completa a un modelo. | | Entradas | * Marca * Talla * Precio * Color * Cantidad | | Resultados | -Si la información es correcta, el par de zapatos se agrega al modelo..  -Si los datos proporcionados no son válidos (el precio no es un valor numérico o la talla es vacía), se muestra un mensaje de error y no se agrega. |  | Nombre | R4-Vender un par de Zapatos | | --- | --- | | Resumen | Vender un par de zapatos a un usuario dado su correo electrónico. | | Entradas | * Correo electrónico del usuario * Nombre del modelo de zapatos * Talla * Cantidad a comprar | | Resultados | -Si toda la información es correcta, la cantidad de unidades vendidas del par de zapatos se incrementa.  -Si toda la información es correcta, se genera una factura con el formato correspondiente.  -Si hay un error de entrada/salida al escribir la factura, se muestra un mensaje al usuario con el error.  -Si la dirección electrónica no es válida, se muestra un mensaje de error y la venta no se realiza. |  | Nombre | R5- Historial de Pedidos | | --- | --- | | Resumen | Almacenar los pedidos de los usuarios con su respectivo zapato y modelo | | Entradas | Zapato y Modelo Actual | | Resultados | * Se genera los registros de venta |  | Nombre | R6- Informe Zapato más caro | | --- | --- | | Resumen | Procesa el Array de zapatos y encuentra el más caro | | Entradas | * ArrayList zapatos | | Resultados | El zapato más caro es: —---------------------------. |  | Nombre | R7- Informe de acuerdo a las unidades vendidas | | --- | --- | | Resumen | Compara el número de unidades vendidas y dependiendo de este genera un mensaje | | Entradas | * Contador de unidades vendidas | | Resultados | Si son menos de 30 pares de zapatos vendidos: regular de lo contrario si son más: excelente |   **Requerimiento No Funcional:**     | Tipo | Persistencia | | --- | --- | | Descripción | * La información del modelo del mundo debe ser persistente. * El proceso de hacer persistente la información debe ser transparente para el usuario (no   participa en el proceso).   * Se deben usar archivos secuenciales para almacenar la información una vez el   programa haya terminado su ejecución. |   **Diagrama de Clases** | | |