



Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

Sistema operativo Renta Equipos del Valle S.A.S

Integrantes del grupo :

Samuel Guerrero Viveros - A00365567

Isabella Naranjo Vargas - A00365234

Enunciado :

Hemos sido contratados por la empresa Renta Equipos del Valle S.A.S, una empresa dedicada a la renta de maquinaria de todo tipo para la construcción, al igual que el transporte de maquinaria, la empresa lleva todos sus datos de forma manual, lo que genera grandes inconvenientes al requerir dicha información, como agilidad al tratar datos de: clientes, empleados, maquinaria, remisiones, devoluciones, y obras, al igual que la seguridad de los datos, la cual es mínima.

Por lo tanto, se hace indispensable el desarrollo de un sistema de software que permita llevar a cabo el manejo de la información, control de inventario y la gestión de remisiones, devoluciones, empleados, clientes, obras, máquinas y equipos. Esta empresa cuenta con un nombre, un NIT que lo identifica y un eslogan.

El sistema debe estar en la capacidad de gestionar (crear, actualizar, eliminar y deshabilitar) todos los productos que ofrece. Todo máquina tiene un número de guía, peso, número de serie y estado de disponibilidad.

Toda maquinaria en el sistema podrá ser eliminada cuando ésta ya no esté a la renta. Deshabilitar un objeto siempre será posible y esto implica que la maquinaria esté en mantenimiento o rentada.

Los clientes están clasificados en dos categorías, las cuales son natural o jurídico. Al ser natural, tiene nombre, apellido, un número de identificación, dirección y teléfono. Por otro lado, al ser



Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

juridicos se identifican con el nombre de la empresa, el NIT, dirección y teléfono. También se deben gestionar a los empleados que tienen nombres, apellidos, cargo y un número de identificación. Igualmente, se debe gestionar un usuario del sistema que serían los trabajadores con sus datos preliminares, un nombre de usuario y una contraseña para ingresar al sistema.

El programa debe incluir un inventario que se pueda actualizar manualmente por todo usuario y por medio de este inventario se llenan las remisiones y devoluciones. Una remisión o devolución tiene un código (autogenerado) de pedido, un listado de productos, la cantidad por cada uno, código de cada producto, el cliente que los solicita o regresa, el empleado que lo entrega o recibe, la fecha y hora de la solicitud (de tipo Date), y las observaciones del estado de entrega de la maquinaria o equipo. Al momento de registrarse el pedido, el programa debe tomar la fecha y hora de la solicitud, de la fecha y hora actual del sistema.

El programa debe permitir cambiar el estado de un pedido entre SOLICITADO, EN PROCESO, ENVIADO y ENTREGADO, o también CANCELADO. También es importante tener en cuenta que se puede cambiar el estado del pedido hacia adelante (por ejemplo, de SOLICITADO a EN PROCESO, o de SOLICITADO a ENVIADO, o de ENTREGADO a DEVUELTO) pero no hacia atrás.

El programa debe guardar toda su información a través de la serialización de sus objetos en archivos.

Este guardado debe ser transparente para el usuario del programa, es decir, cada vez que se registre o actualice información, esta se guardará en los archivos serializados. También se debe generar un archivo csv de pedidos, con una fila por cada pedido, con los datos del nombre, dirección y teléfono del cliente que lo solicitó, el nombre del empleado que lo autorizó, estado del pedido, debe tener tres columnas por cada pedido con el número de guía de la máquina, el valor unitario de esta y la cantidad.



Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

El reporte debe poder generarse en un rango de fechas y hora, es decir, al momento de generarlo se pregunta la fecha y hora inicial, y la fecha y hora final del reporte. Por defecto, el valor en la fecha y hora inicial son las 00:00 del día actual, y la fecha final debe ser por defecto 23:59 del día actual. El reporte debe estar ordenado por fecha y hora del pedido ascendente.

El programa debe tener una opción que permita buscar eficientemente un cliente dado su ID o una maquinaria dando su número de guía. Esta opción debe estar en el campo que permite realizar los pedidos.

El programa debe permitir importar datos de un archivo csv con información de clientes, productos y pedidos.

El programa debe poder cumplir con los siguientes requisitos :

Requerimientos Funcionales :

1. El programa debe estar montado y trabajado en GitHub.
2. El programa debe implementar una interfaz para iniciar sesión a través de un usuario y contraseña.
3. El programa debe guardar los datos del programa por persistencia persistente.
 - a. Una parte de la persistencia debe ser manejada a través de archivos de texto plano
 - b. Otra parte de la persistencia debe ser manejada a través de serialización de clases
 - c. Se deben identificar los principales casos de excepción y manejarlos pertinentemente en cada caso (pueden ser lanzadas en un método, propagados por otros y capturados posteriormente)
 - d. Algunas excepciones se manejan con las excepciones propias del api de java (al menos 4).
 - e. Se deben identificar al menos 3 excepciones personalizadas (con su propia clase).



Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

4. El proyecto debe implementar un inventario (con filtro) para toda la empresa, máquinas, equipos, oficina, etc.
5. Se debe generar un archivo CSV de pedidos, con una fila por cada remisión, con los siguientes datos: Razón social, contacto, dirección, obra, NIT, y teléfono de quien lo solicitó, si es un cliente ya creado se autogeneran los siguientes campos : Razón social, dirección, NIT, y teléfono igualmente con la opción de modificarlo manualmente, también debe ser incluido en el archivo CSV el nombre del empleado que hizo el pedido. debe tener cinco columnas por cada pedido con la cantidad de cada producto, la marca del producto (si aplica), el nombre del producto, la cantidad, y un campo de observaciones.
6. El proyecto debe incluir el diseño y la implementación de Pruebas Unitarias Automáticas para TODOS los métodos de las clases del mundo.
 - a. Diseño de los Casos de Prueba.
 - b. Implementación de las Pruebas Unitarias Automáticas.
7. El proyecto debe permitir la creación de remisiones o devoluciones, y poder seleccionar del inventario de la empresa los productos a agregar con su respectiva cantidad y Id al igual que dar opción de imprimirlas.
8. El programa debe permitir la creación de usuarios desde la misma pantalla de iniciar sesión
9. El programa debe permitir la creación de : empleados, clientes, máquinas, equipos, y obras.
10. La lista de clientes debe estar ordenada siempre alfabéticamente por apellido y nombre, por tanto, cada vez que se agrega un nuevo cliente, este debe insertarse de forma ordenada.
11. Debe incluir la gestión (Eliminar o despedir, suspender, editar, crear) del personal de la empresa, clientes, y obras.
12. El programa debe permitir gestionar (eliminar, editar, actualizar estado, actualizar ubicación, agregar, suspender) los equipos y máquinas de la empresa.



Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

13. El programa debe verificar antes de eliminar cualquier producto que ya esté dentro del programa, que no esté en estado de uso, o referenciado desde otro lado.
14. El programa debe tener medidas de seguridad para el trato de cierta información o toda.
15. Una remisión o devolución tiene un código (autogenerado) de pedido, un listado de productos, la cantidad por cada uno, marca(si aplica), nombre del producto, código INTERNO de cada producto, el cliente que los solicita o regresa, el empleado que lo entrega o recibe, la fecha y hora de la solicitud (de tipo Date), y las observaciones del estado de entrega de la maquinaria o equipo. Al momento de registrarse el pedido, el programa debe tomar la fecha y hora de la solicitud, de la fecha y hora actual del sistema, pero es posible cambiarla manualmente.
16. El programa debe permitir importar datos de un archivo csv con información de clientes, y pedidos.
17. El programa debe permitir generar una factura de cobro a partir de las remisiones creadas, y los precios podrán ser modificados por usuarios verificados.

Justificación :

Debido a un buen plazo para el desarrollo del proyecto, y la buena relación establecida con la empresa Renta Equipos del Valle podemos compensar la complejidad del proyecto y con la correcta planeación se decidió que dos programadores sería suficiente para llevar a cabo este proyecto y asegurándose de cumplir todos los requerimientos propuestos anteriormente.



Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

Diseño casos de prueba :

Nombre	Clase	Escenario
setupScenary1()	Client	Client(null, null, 0)
setupScenary1()	Person	Employee(null, null, 0)
setupScenary1()	User	User(null, null, 0, null, null)
setupScenary1()	Employee	Employee(null, null, 0)

Clase	Método	Escenario	Valores de entrada	Resultado
Client	getAddress()	setupScenary1() ()	“ “	Se obtiene el address del cliente
Client	setAddress()	setupScenary1() ()	“Address: Cra 1C”	El address del cliente se cambia
Client	getPhone()	setupScenary1() ()	“ “	Se obtiene el pone del cliente
Client	setPhone()	setupScenary1() ()	“Phone: 12345”	El phone del cliente cambia
Person	getName()	setupScenary1() ()	“ “	Se obtiene el name de la persona
Person	setName()	setupScenary1() ()	“Name: Narajo”	El name de la persona cambia
Person	getLastName()	setupScenary1() ()	“ “	Se obtiene Lastname de la persona
Person	setLastName()	setupScenary1() ()	“Lastname: Naranjo”	El Lastname <u>de</u> la persona cambia
Person	setId()	setupScenary1() ()	“ “	Se obtiene el ID de la persona



Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

Person	setId()	setupScenary1 ()	“Id: 123”	El ID de la persona cambia
User	setUsername()	setupScenary1 ()	“Username: Narajov”	El username del usuario cambia
User	getUsername()	setupScenary1 ()	“ “	Se obtiene el Username del usuario
User	setUserPassword()	setupScenary1 ()	“Password: 123”	Se cambia el password de la persona
User	getUserPassword()	setupScenary1 ()	“ “	Se obtiene el password de la persona
RentaEquiposDelV alle	setUsers()	setupScenary1 ()	“ArrayList con usuarios”	La Lista con usuarios cambia
RentaEquiposDelV alle	getUsers()	setupScenary1 ()	“ “	Se obtiene la lista de usuarios
RentaEquiposDelV alle	setClients()	setupScenary1 ()	“Arraylist con clientes”	La lista con clientes cambia
RentaEquiposDelV alle	getClients()	setupScenary1 ()	“ “	Se obtiene la lista con clientes
RentaEquiposDelV alle	setEmployee()	setupScenary1 ()	“ArrayList con Employees”	La lista con Employees cambia
RentaEquiposDelV alle	getEmployee()	setupScenary1 ()	“ “	Se obtiene la lista con Employees
RentaEquiposDelV alle	Load()	setupScenary1 ()	“ “	Se cargan los archivos serializados

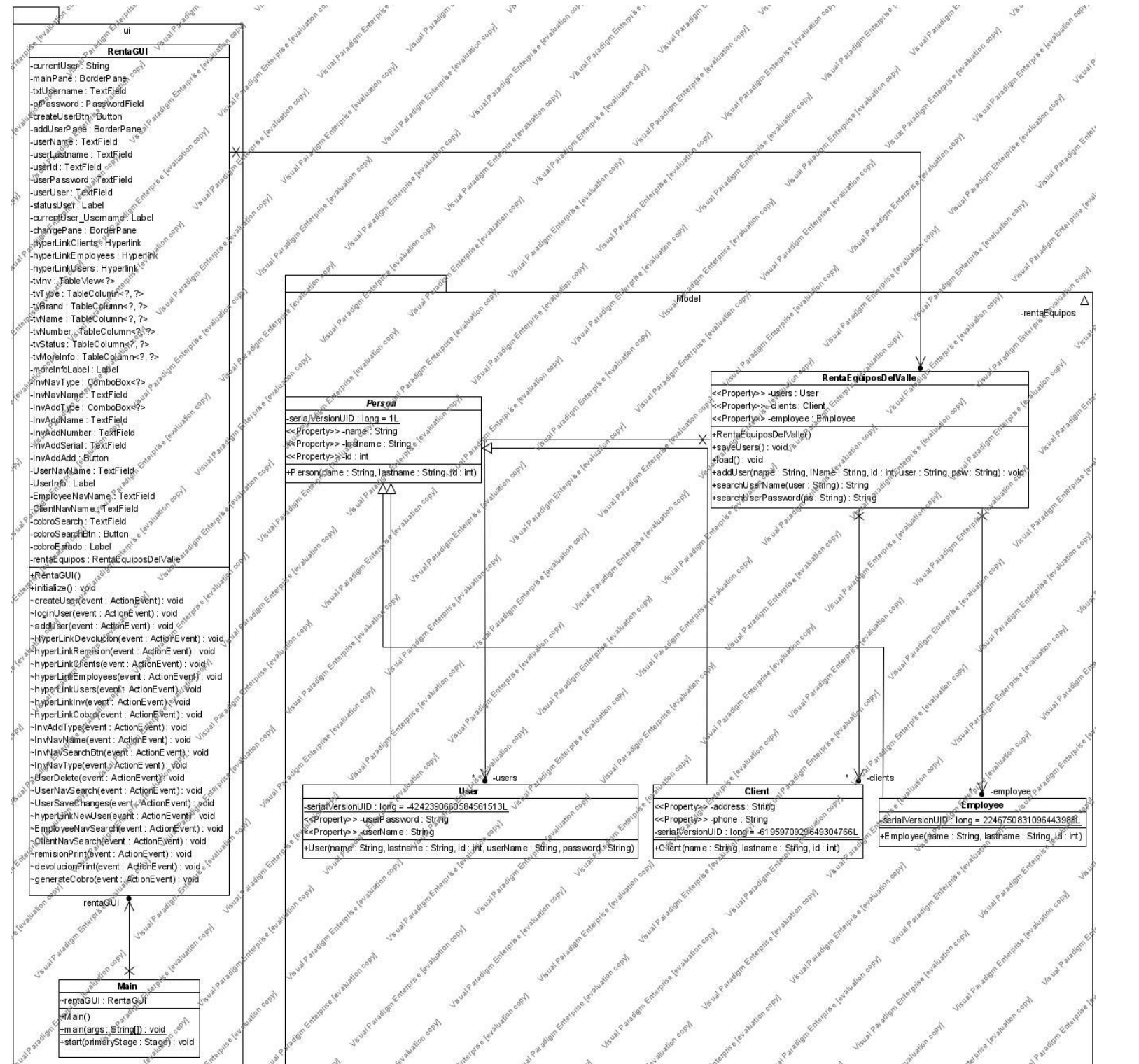


Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

				con usuarios
RentaEquiposDelValle	addUser()	setupScenary1 ()	“Isabella, Naranjo, 123, Narajov, 123”	Se añade un nuevo usuario a Clients
RentaEquiposDelValle	searchUsername()	setupScenary1 ()	“Narajov”	Se retorna el usuario encontrado
RentaEquiposDelValle	searchPassword()	setupScenary1 ()	“123”	Se retorna el usuario con la contraseña a encontrar



Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.





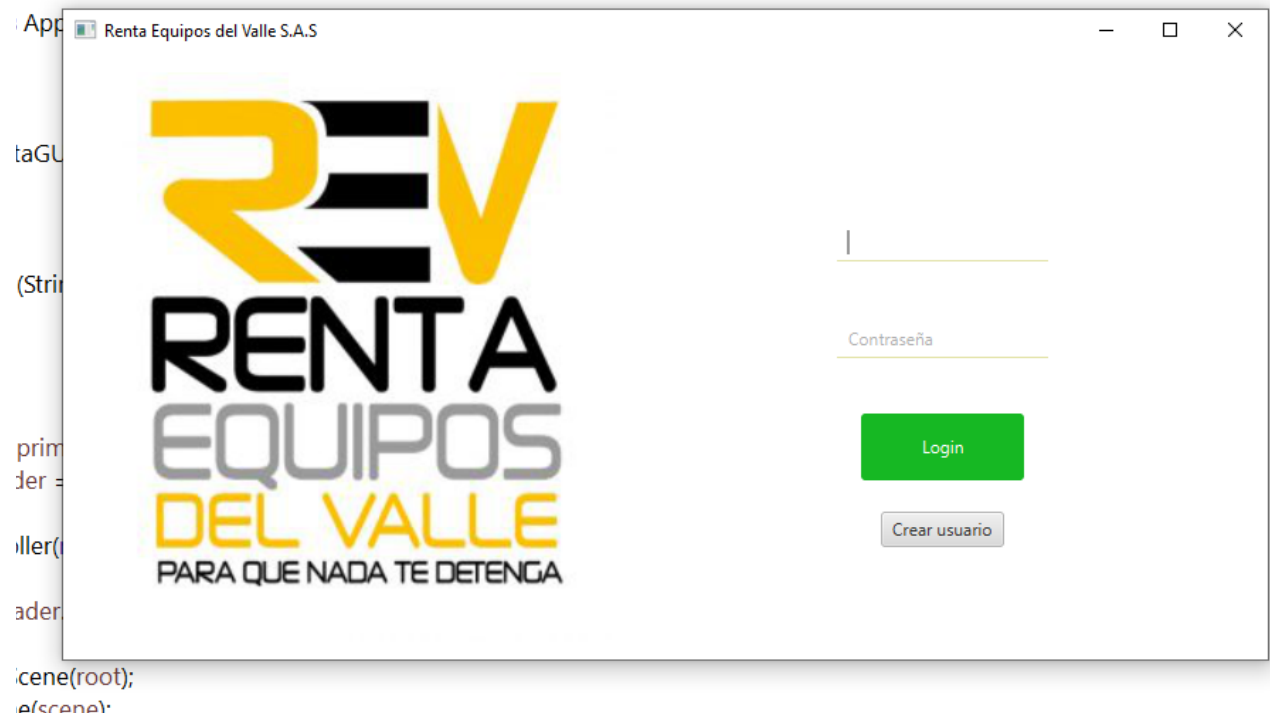
Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

Sketches :

Pantallazo de la interfaz para iniciar sesión:

Esta interfaz gráfica es la más importante del proyecto, esto se debe a que es el medio por el cual tendremos acceso a nuestro programa que contiene datos importantes y delicados de Renta equipos del Valle. Esta será la principal seguridad para la protección de los datos que se manejan en la aplicación.

Application,...





Proyecto de curso TI3 | Algoritmos y programación II - easyManagement.

Pantallazo de interfaz para crear usuarios:

Esta interfaz nos permite crear nuevos usuarios sin tener que entrar al programa

A screenshot of a web application window titled 'Crear usuario'. The window has a white background and a grey border. It contains five input fields for user creation: 'Nombre', 'Apellido', 'Identificación C.C', 'Usuario', and 'Contraseña'. Each field has a corresponding label to its left. Below the input fields is a grey button labeled 'Crear'. The window is displayed over a background that shows a ruler at the top and a code editor on the left with some text like 'ot);' and 'n'.