**TruckGestion**

**Universidad del Valle – Tecnología en Desarrollo de Software**

|  |  |
| --- | --- |
| Alejandro Cuenca | 2264475-2724 |
| Camilo García | 2264484-2724 |
| Manuel Builes | 2264325-2724 |
| Oscar Hernández | 2264488-2724 |

**1. Introducción**

Decidimos hacer un proyecto referente a el transporte de cargas pesadas ya que lo más probable es que no haya mucha gente que elija este tema y además es una aplicación interesante para hacer teniendo en cuenta que estamos hablando de programación orientada a objetos.

Este proyecto puede ser útil para el sector del transporte de mercancías y para la empresa que sea que pueda utilizar este programa, va a ser algo pequeño obviamente, quizás se pueda hacer algo más grande, pero por temas de tiempo no va a ser así.

Volviendo al tema, el proyecto que planteamos es básicamente una aplicación que puede servir para identificar:

• la carga y tonelaje

• ciudad origen

• ciudad destino

• camión

• hora de salida

• hora llegada

• estado de la carga al salir y al llegar

• empresa de la que sale la carga

• lugar/empresa a la que llega la carga

Esto es lo que hemos planteado hasta el momento.

La utilidad de este proyecto es básicamente tener la posibilidad de ver las entregas y cargas que se están llevando a cabo por parte de la empresa de transporte, pero también mirar la data que se genera de las entregas: los conductores, empresas, rutas que más se usan, los conductores que están empleados, los camiones, horas en las que más se viaja, las ciudades que más se visitan, etc.

**2. Problema de investigación**

• Problema a tratar: control de la data de una empresa de transporte, tanto en el momento como a posteriori para análisis de datos, en este caso referente a una empresa logística. En este caso es una aplicación pequeña que tratara lo básico del problema, datos, encargos, recorridos y entregas, lo básico de una empresa de logística. En caso de que esta aplicación fuera más compleja serviría para en ver tiempo real (en caso de que se conectara a un servidor en internet y los camiones tuvieran sensores) monitorear los encargos que se estuvieran haciendo, obviamente esto es una aplicación pequeña y no se va a conectar a ningún server, pero la idea general es esta.

• Importancia: este tipo de sistemas ya existen, no es algo innovador lo que vamos a hacer, pero puede ser útil para una empresa de logística tener control sobre sus rutas y sobre su data.

• Preguntas relacionadas al problema:

¿Como esta aplicación permitirá manejar los envíos de una empresa de transporte?

¿Cómo se gestionará la data?

**3. Objetivos**

**3.1 Objetivo General**

Diseñar un programa para una empresa de transporte que lleve los registros, ubicaciones y datos de sus camiones en todo el valle del cauca

**3.2 Objetivos Específicos**

* Conseguir una recolección de datos Útiles y no datos basura o irrelevantes
* Hacer código legible
* Crear tablas para poder ver los datos de los diversos objetos:
  + Administradores
  + Moderadores
  + Mecánicos
  + Clientes
    - Activos o Inactivos
    - Tipo de empresa (a que se dedica o que fabrica, etc.)
    - Ubicación
    - Si tiene o no un envió pendiente
  + Camioneros
    - Camión asignado
    - Cedula/ID
    - Edad
    - Nombre
  + Camiones
    - Marca
    - Tonelaje
    - Placa
    - Tipo de camión (europeo o americano) (pequeño o grande)
    - Si esta quieto o en movimiento
  + Cargas
    - Camión asignado
    - Camionero
    - Ruta
    - Origen de salida y Destino de llegada
    - Horas de salida y llegada
    - Tamaño y tonelaje
    - Carga
  + Lista
    - Aquí se conectan todos los datos
* Implementar un sistema de entrada por códigos para los distintos tipos de usuarios •Admins •Camioneros •Clientes •Mecánicos •Empleados
* Construir una forma de guardar los datos de todos los objetivos anteriores

**4. Sistema propuesto**

Descripción de la aplicación a desarrollar:

Se Desarrollará una aplicación para gestionar el sistema de una empresa de Transportes y envíos en el Departamento del Valle del Cauca.

•La aplicación permitirá: Registrar todos los transportes, ubicaciones y estados de los camiones, rutas, camioneros, empresa cliente, lugar origen y lugar destino

•La aplicación contara con opciones para: Registrar nuevos camioneros y camiones, además obviamente de que estos podrán colocar la ruta a la que se están dirigiendo y la carga que están llevando

•La aplicación se implementará en lenguaje: C++

**4.1 Usuarios**

• Admin: Es el que puede controlar y modificar todo lo que está en la aplicación como y cuando quiera, se tiene que tener cuidado con a quien se le dan estos permisos ya que podría causar bastantes daños si los usa mal.

• Empleado: Puede modificar la mayoría de cosas (excepto las de vital importancia como por ejemplo eliminar cosas de las bases de datos) que hay en la aplicación y gestionarla con permisos del Admin, en caso de que se le den permisos a una persona para que pueda moderar hay que tener cuidado con el uso que le da a estos permisos.  
• Mecánico: repara los camiones en caso de averías

• Camionero: Puede ver cuáles son sus cargas y hacia donde tiene que ir.

• Cliente: Puede ver él está de su envió (en metáfora porque esta aplicación no va tener sistemas de GPS ya os lo digo) y obviamente que fue lo que encargo

**4.2 Entidades**

Descripción de cada una de las entidades (clases) que hacen parte de la aplicación y sus atributos

Main: aquí va todo el programa conectado con las demás clases y el log-in

Admin: la persona que se puede encargar de todo incluso de modificar cosas de las demás clases

Empleado: puede ejercer ciertas acciones que aún no están decididas

Mecánico: Reparan camiones en caso de que estén dañados

Camionero: se le asigna un camión y una carga y un cliente o destino

Cliente: es el que encarga y recibe la mercancía, o devolverla dado el caso

Camiones: es este objeto se ejecutarán los sistemas, listas y adiciones de vehículos

Cargas: Es el producto que se llevara al destino concretado

**4.3 Requerimientos Funcionales**

Descripción de cada una de las funcionalidades del sistema. Se deben especificar mínimo 15 requerimientos funcionales. Usar la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| **No. Req** | **Descripción** | **Usuarios** |
| 1 | La aplicación permite crear usuarios tipo Mod, Camionero | Admin |
| 2 | La aplicación permite crear Clientes | Admin |
| 3 | Modificar usuarios o eliminar usuarios | Admin |
| 4 | Ver el origen y destino de una carga | Todos |
| 5 | Ordenar por números (Me niego a hacer ordenamiento alfabético de manera total y absolutamente rotunda) | Todos |
| 6 | Ver las cargas disponibles y las rutas que estas tienen | Camioneros |
| 7 | Ver los envíos realizados | Clientes |
| 8 | Guardar data | Admin |
| 9 | Ver el estado de un camión, si esta estacionado o en movimiento | Admin |
| 10 | Modificar Usuarios | Admin |
| 11 | Eliminar Usuarios | Admin |
| 12 | Ingresar a la aplicación como un tipo de usuario dependiendo de una selección de una categoría y un código de 5 dígitos | Todos |
| 13 | Dependiendo de que usuario entre se verán distintas cosas y opciones adicionales | Depende |
| 14 | Reportar daño de camión | Camionero |
| 15 | Los clientes no ingresan a la app si no que ingresan el código de la carga para ver el estado de esta |  |
| 16 | Los clientes pueden hacer devoluciones | Clientes |
| 17 | Reparación de camiones | Mecánicos |
| 18 |  |  |
| 19 |  |  |
| 20 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**4.4 Requerimientos No Funcionales**

Descripción de las características del sistema. Usar la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| **No. Req** | **Atributo** | **Descripción** |
| 1 | Plataforma de Software | Windows 7 en adelante... |
| 2 | Plataforma de Hardware | Equipo de Cómputo:   * ….. * ….. * …... * …… |
| 3 | Interfaz | Como es la interfaz del usuario |

**4.5 Diagrama de caso de uso del cliente**

Diagrama de casos de uso representa las acciones (funciones) a las cuales tiene acceso el Actor (cliente). En otras palabras, es que puede hacer el Actor en el sistema.

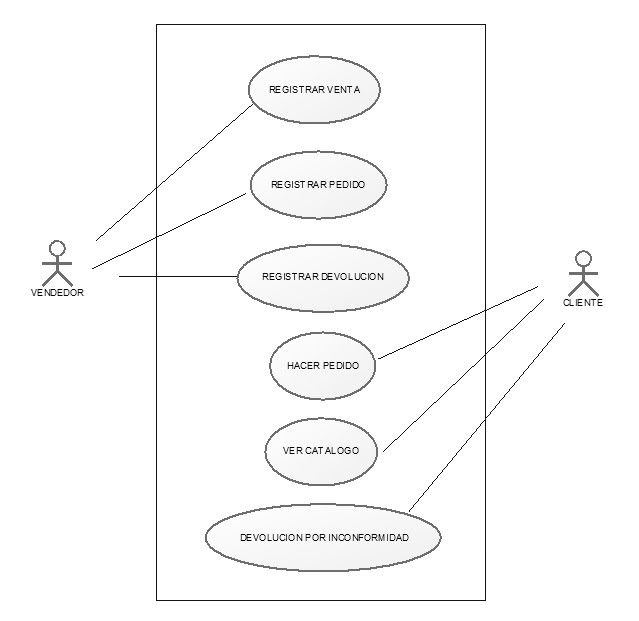
Las acciones se representan mediante un ovalo, el cual tiene una breve descripción de la acción que el Actor puede realizar en el sistema. La descripción comienza con un verbo.

Los casos de uso se basan en la descripción de los Requerimientos Funcionales.

Por ejemplo: el siguiente diagrama de casos de uso del cliente representa a dos actores: Vendedor y Cliente.

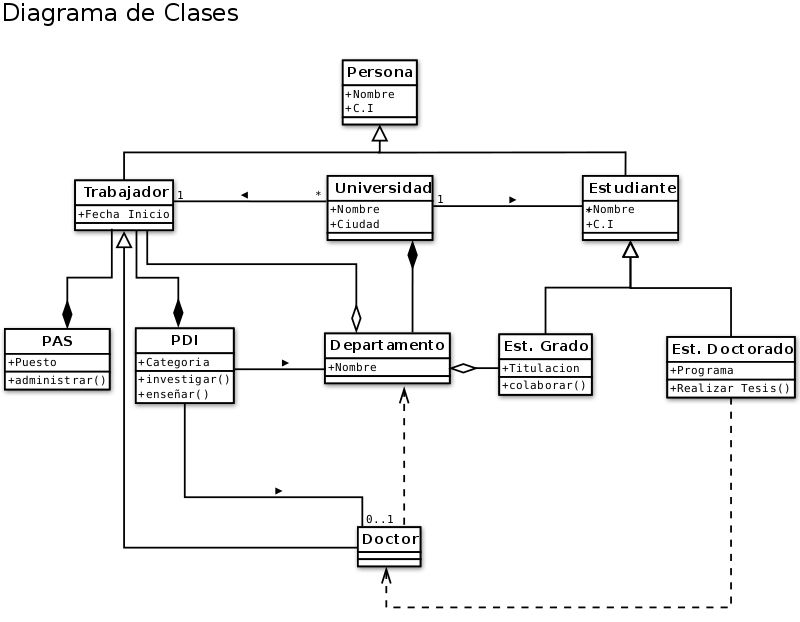
El Actor Vendedor puede realizar las funciones (caso de uso) de: Registrar venta, Registrar pedido, Registrar devolución.

El Actor Cliente puede realizar las funciones (caso de uso) de: Hacer pedido, Ver catálogo, Devolver por inconformidad.



**5. Diagrama de clases**

Diagrama de clases de la aplicación donde se muestre las clases implementadas, atributos, métodos con su correspondiente tipo de acceso (public, private o protected) y las relaciones entre ellas (herencia, composición).

****

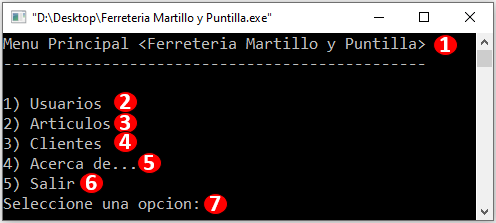
**6. Sistema Implementado**

Capturas de pantalla de todas las ventanas implementadas para la aplicación con la descripción de su funcionalidad y las opciones.

La descripción de cada ventana se hace en una nueva hoja, por ejemplo;

**6.1 Ventana – Menú Principal**

Pantalla 1. Menú principal aplicación “Ferretería Martillo y Puntilla”



**Descripción de la ventana**

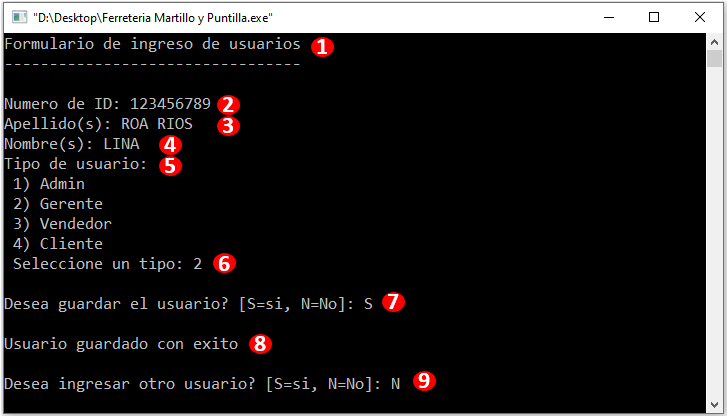
Menú con opciones para ingresar a submenús y funciones principales de la aplicación

**Descripción de las opciones**

1. Título de la ventana.
2. Opción para ingresar al submenú de gestión de usuarios.
3. Opción para ingresar al submenú de gestión de artículos.
4. Opción para ingresar al submenú de gestión de clientes.
5. Visualiza en pantalla los datos personales de los programadores de la aplicación.
6. Opción para salir de la aplicación.
7. Entrada de datos: permite ingresar la opción que desea ejecutar el usuario.

**6.2 Ventana – Ingreso de usuarios**

Pantalla 2. Formulario ingreso de usuarios



**Descripción de la ventana**

Formulario que permite el ingreso de datos de un nuevo usuario a la aplicación.

**Descripción de los componentes**

1. Título de la ventana.
2. Entrada de datos: permite ingresar el número de identificación del nuevo usuario.
3. Entrada de datos: permite ingresar el apellido(s) del nuevo usuario.
4. Entrada de datos: permite ingresar el nombre(s) del nuevo usuario.
5. Menú para seleccionar el tipo de usuario y su rol.
6. Entrada de datos: permite seleccionar el tipo de usuario y su rol.
7. Entrada de datos: Mensaje para confirmar el almacenamiento del usuario.
8. Mensaje de confirmación del almacenamiento del usuario.
9. Entrada de datos: Mensaje para preguntar si desea ingresar otro usuario. En caso afirmativo se muestra nuevamente la ventana <Ingreso de usuarios>, de lo contrario vuelve al submenú de <Gestión de Usuarios>

**6.n Ventana – XXXXX XXXXX XXXXX**

Pantalla n. XXXXX XXXXXXX XXXXXXX XXXXXXX

Captura de pantalla

**Descripción de la ventana**

XXXXX XXXXXXX XXXXXXX XXXXXXX

**Descripción de los componentes**

1. XXXXX XXXXXXX XXXXXXX XXXXXXX.
2. XXXXX XXXXXXX XXXXXXX XXXXXXX.
3. XXXXX XXXXXXX XXXXXXX XXXXXXX.
4. ……
5. .

**7. Conclusiones**

Deben ser:

Desde el punto de vista del sector analizado

Desde el punto de vista personal

Desde el punto de vista del perfil profesional (como futuro Ingeniero de Sistemas)

**Bibliografía**

(usar formato APA)