DOCUMENTO DEL SISTEMA DE CAMIONCITOS S.A

**Autores:**

Bonifaz Chacon Jose Xavier

Calderon Serrano Kleidy Anallely

Cedeño Plúa Isaac Alexander

Coloma Guzmán John Steven

Molina Aguilera Katherine Daniela

Tomalá Magallanes Kevin Saúl

**Tabla de contenido**

[Resumen 3](#_Toc136202109)

[Abstract 3](#_Toc136202110)

[Introducción 4](#_Toc136202111)

[Estimación Cocomo 5](#_Toc136202112)

[Cocomo Básico – proyecto orgánico 5](#_Toc136202113)

[Cocomo Intermedio – Proyecto Orgánico 6](#_Toc136202114)

[COCOMO II 9](#_Toc136202115)

[Metodología de desarrollo 12](#_Toc136202116)

[Herramientas para el desarrollo del sistema 12](#_Toc136202117)

[Diagrama de clases 17](#_Toc136202118)

[Diagrama de componentes 17](#_Toc136202119)

[Diagrama de paquetes 18](#_Toc136202120)

[Resultado 18](#_Toc136202121)

[Conclusión 21](#_Toc136202122)

# Resumen

En la empresa Camioncitos S.A la falta de un sistema adecuado para la distribución de choferes y planificación de rutas ha generado asignaciones ineficientes de recursos, retrasos en los tiempos de entrega y una mayor carga de trabajo para la secretaria encargada. Estos problemas negativamente en la satisfacción del cliente y en la pérdida de ventas, lo que pone en riesgo a la empresa.

Por lo tanto, se propone la creación de un software que automatice y agilice la distribución y coordinación de choferes a vehículos disponibles, trazando la ruta más eficiente para la entrega de productos. Este sistema optimizará la asignación de recursos, reducirá los tiempos de entrega y mejorará la satisfacción del cliente.

Para lograr estos objetivos, se utilizarán los modelos COCOMO I y COCOMO II adaptados a las necesidades de Camioncitos S.A para estimar los costos, el esfuerzo, el tiempo y los recursos requeridos para el proyecto. Asimismo, se empleará una metodología de desarrollo SCRUM donde se dará la participación activa de los usuarios y el equipo de desarrollo para obtener un resultado más realista.

Se diseñará una base de datos en SQL Developer y la codificación del proyecto se realizará en el IDE Microsoft Visual Studio en lenguaje C#. También se utilizarán herramientas específicas para garantizar la organización y el cumplimiento de los plazos establecidos como los diagramas de clase, de componentes y el modelo de casos de uso.

**Palabras Clave:** Camioncitos S.A, Metodología SCRUM, COCOMO, Software, Distribución.

# Abstract

In Camioncitos S.A. the lack of an adequate system for the distribution of drivers and route planning has generated inefficient resource allocation, delays in delivery times and an increased workload for the secretary in charge. These problems negatively affect customer satisfaction and lost sales, putting the company at risk.

Therefore, we propose the creation of a software that automates and streamlines the distribution and coordination of drivers to available vehicles, tracing the most efficient route for product delivery. This system will optimize resource allocation, reduce delivery times, and improve customer satisfaction.

To achieve these objectives, the COCOMO I and COCOMO II models adapted to the needs of Camioncitos S.A. will be used to estimate the costs, effort, time, and resources required for the project. Likewise, a SCRUM development methodology will be used with the active participation of the users and the development team to obtain a more realistic result.

A database will be designed in SQL Developer and the coding of the project will be done in Microsoft Visual Studio IDE in C# language. Specific tools will also be used to ensure the organization and compliance with the established deadlines such as class diagrams, component diagrams and the use case model.

**Keywords:** Camioncitos S.A., SCRUM Methodology, COCOMO, Software, Distribution.

# Introducción

Actualmente, el uso de sistemas informáticos por parte de las empresas está en constante crecimiento. Estas organizaciones buscan automatizar y mejorar sus servicios a través de la implementación de sistemas que les permitan gestionar de manera eficiente diversas actividades. Esto incluye a las empresas de envíos, que necesitan un sistema para manejar el envío de sus productos, asignar choferes a vehículos específicos y planificar rutas particulares para cada entrega.

En este sentido, la gerencia de Camioncitos S.A se ha visto interesada en hacer uso de un software que permita optimizar el sistema de la gestión de envíos que posee la empresa, y así aminorar el trabajo de secretaría.

El presente proyecto tiene como objetivo principal desarrollar un software que permita optimizar el sistema de gestión de envíos de la empresa Camioncitos S.A. El software buscará automatizar y agilizar la distribución y coordinación de choferes a vehículos disponibles, trazando la ruta más eficiente para la entrega de productos.

Cabe destacar que este proyecto es realizado por estudiantes de sexto semestre de la Universidad de Guayaquil que forman parte de la Facultad de Ciencias, Matemáticas y Física, bajo la guía del Ph.D. Franklin Parrales docente de la asignatura construcción de software.

Se espera lograr una mejora significativa en la distribución de la gestión de envíos de Camioncitos S.A, brindando a la empresa una solución informática que optimice sus servicios y contribuya a su crecimiento y eficiencia en el mercado.

# Estimación Cocomo

## Cocomo Básico – proyecto orgánico

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Submodelos básicos** | **a** | **b** | **c** | **d** |
| Orgánico | 2,4 | 1,05 | 2,5 | 0,38 |
| Semi-acoplado | 3,0 | 1,12 | 2,5 | 0,35 |
| Empotrado | 3,6 | 1,20 | 2,5 | 0,32 |

* **SUBMODELOS DE COCOMO I**

**SIZE Estimado =** 19 KLOC

* **ORGÁNICO**

**Esfuerzo** = =2.4 \* (19)1.05 = 52 PM (Personas-Mes)

**DevTime** = = 2.5 \* (52)0.38 = 11.2 M (Meses)

=> **Personas**: 52/11.2 = 4.64 personas

=> **Productividad**: 19000/52 = 365.3 LOC/PM

* **SEMI-ACOPLADO**

**Esfuerzo** = =3.0 \* (19)1.12 = 81 PM (Personas-Mes)

**DevTime** = = 2.5 \* (81)0.35 = 11.6 M (Meses)

=> **Personas**: 81/11.6 = 6.98 personas

=> **Productividad**: 19000/81 = 234.5 LOC/PM

* **EMPOTRADO**

**Esfuerzo** = =3.6 \* (19)1.20 = 123 PM (Personas-Mes)

**DevTime** = = 2.5 \* (123)0.32 = 11.6 M (Meses)

=> **Personas**: 123/11.6 = 10.6 personas

=> **Productividad**: 19000/123 = 154.4 LOC/PM

## Cocomo Intermedio – Proyecto Orgánico

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Submodelos intermedios** | **a** | **b** | **c** | **d** |
| Orgánico | 3,2 | 1,05 | 2,5 | 0,38 |
| Semi-acoplado | 3,0 | 1,12 | 2,5 | 0,35 |
| Empotrado | 2,8 | 1,20 | 2,5 | 0,32 |

**Tabla Factores de Costo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributos | Valor | | | | | |
| Muy bajo | Bajo | Normal | Alto | Muy Alto | Extra alto |
| Atributos de Software | | | | | | |
| Fiabilidad | 0,75 | 0,88 | 1 | 1,15 | 1,4 |  |
| Tamaño de Base de datos |  | 0,94 | 1 | 1,08 | 1,16 |  |
| Complejidad | 0,7 | 0,85 | 1 | 1,15 | 1,3 | 1,65 |
| Atributos de hardware | | | | | | |
| Restricciones de tiempo de ejecución |  |  | 1 | 1,11 | 1,3 | 1,66 |
| Restricciones de memoria virtual |  |  | 1 | 1,06 | 1,21 | 1,56 |
| Volatilidad de la máquina virtual |  | 0,87 | 1 | 1,15 | 1,3 |  |
| Tiempo de respuesta |  | 0,87 | 1 | 1,15 | 1,3 |  |
| Atributo de personal | | | | | | |
| Capacidad de análisis | 1,46 | 1,19 | 1 | 0,86 | 0,71 |  |
| Experiencia en la aplicación | 1,29 | 1,13 | 1 | 0,91 | 0,82 |  |
| Calidad de los programadores | 1,42 | 1,17 | 1 | 0,86 | 0,7 |  |
| Experiencia en la máquina virtual | 1,21 | 1,1 | 1 | 0,9 |  |  |
| Experiencia en lenguaje | 1,14 | 1,07 | 1 | 0,95 |  |  |
| Atributos del proyecto | | | | | | |
| Técnicas actualizadas de programación | 1,24 | 1,1 | 1 | 0,91 | 0,82 |  |
| Utilización de herramientas de software | 1,24 | 1,1 | 1 | 0,91 | 0,83 |  |
| Restricciones de tiempo de desarrollo | 1,23 | 1,08 | 1 | 1,04 | 1,1 |  |

**Cálculo de SIZE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Factores de Peso** | | | | |
| Factores Funcionales de Peso | Parámetros de Medida (1) | | | Contador (2) | Total Multiplicación (1) \*(2) |
| Simple | Media | Compleja |
| N.º Entrada de usuario | 7 | 10 | 15 | 20 | 200 |
| N.º Salida usuario | 5 | 7 | 10 | 10 | 70 |
| N.º Consulta usuario | 3 | 4 | 6 | 8 | 32 |
| N.º Archivos Lógicos Internos (tablas) | 4 | 5 | 7 | 4 | 20 |
| N.º Interfaces externas | 3 | 4 | 6 | 6 | 24 |
|  |  |  |  | | 346 |

**Cálculo fórmula LOC**

**Tabla de conversión de: Correlación código fuente a PF**

|  |  |
| --- | --- |
| Lenguaje | Correlación Código Fuente por PF (aprox) |
| Assembler | 320 |
| C | 128 |
| ALGOL | 105 |
| FORTRAN | 105 |
| PASCAL | 91 |
| RPG | 80 |
| PL/1 | 80 |
| Modula-2 | 80 |
| Prolog | 64 |
| LISP | 64 |
| BASIC | 64 |
| 4GL para BD | 40 |
| APL | 32 |
| Smaltallk | 29 |
| Query | 13 |
| Spreadsheet | 6 |
| Sql | 13 |
| VB | 24 |
| Java | 46 |
| Html | 14 |
| Delphi | 118 |
| C++ | 53 |
| COBOL | 107 |
| C# | 58 |

**Cálculo de la Variable FAE (multiplicador)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributos | Valor | | | | | |
| Muy bajo | Bajo | Normal | Alto | Muy Alto | Extra alto |
| Atributos de Software | | | | | | |
| Fiabilidad | 0,75 | 0,88 | 1 | 1,15 | 1,4 |  |
| Tamaño de Base de datos |  | 0,94 | 1 | 1,08 | 1,16 |  |
| Complejidad | 0,7 | 0,85 | 1 | 1,15 | 1,3 | 1,65 |
| Atributos de hardware | | | | | | |
| Restricciones de tiempo de ejecución |  |  | 1 | 1,11 | 1,3 | 1,66 |
| Restricciones de memoria virtual |  |  | 1 | 1,06 | 1,21 | 1,56 |
| Volatilidad de la máquina virtual |  | 0,87 | 1 | 1,15 | 1,3 |  |
| Tiempo de respuesta |  | 0,87 | 1 | 1,15 | 1,3 |  |
| Atributo de personal | | | | | | |
| Capacidad de análisis | 1,46 | 1,19 | 1 | 0,86 | 0,71 |  |
| Experiencia en la aplicación | 1,29 | 1,13 | 1 | 0,91 | 0,82 |  |
| Calidad de los programadores | 1,42 | 1,17 | 1 | 0,86 | 0,7 |  |
| Experiencia en la máquina virtual | 1,21 | 1,1 | 1 | 0,9 |  |  |
| Experiencia en lenguaje | 1,14 | 1,07 | 1 | 0,95 |  |  |
| Atributos del proyecto | | | | | | |
| Técnicas actualizadas de programación | 1,24 | 1,1 | 1 | 0,91 | 0,82 |  |
| Utilización de herramientas de software | 1,24 | 1,1 | 1 | 0,91 | 0,83 |  |
| Restricciones de tiempo de desarrollo | 1,23 | 1,08 | 1 | 1,04 | 1,1 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Submodelos intermedios | a | b | c | d |
| Orgánico | 3,2 | 1,05 | 2,5 | 0,38 |
| Semi-acoplado | 3,0 | 1,12 | 2,5 | 0,35 |
| Empotrado | 2,8 | 1,20 | 2,5 | 0,32 |

**ENTONCES SE TIENE:**

# COCOMO II

* **MODELO DE DISEÑO INICIAL**

**A =** 2.5

**SIZE Estimado =** 19 KLOC

**B =** 0.91 + 0.01 \* (Sum of rating on scaling factors for the project)

**Factores de Escala B:**

PREC = nominal (3.72)

FLEX = very high (1.01)

RESL = high (2.83)

TEAM = extra high (0.00)

PMAT = nominal (4.68)

**Multiplicadores M**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cost Drivers** | **Extra Low** | **Very Low** | **Low** | **Nominal** | **High** | **Very High** | **Extra High** |
| **RCPX** | 0.73 | 0.81 | 0.98 | 1.0 | 1.30 | 1.74 | 2.38 |
| **RUSE** | - | - | 0.95 | 1.0 | 1.07 | 1.15 | 1.24 |
| **PDIF** | - | - | 0.87 | 1.0 | 1.29 | 1.81 | 2.61 |
| **PERS** | 2.12 | 1.62 | 1.26 | 1.0 | 0.83 | 0.63 | 0.50 |
| **PREX** | 1.59 | 1.33 | 1.12 | 1.0 | 0.87 | 0.71 | 0.62 |
| **FCIL** | 1.43 | 1.30 | 1.10 | 1.0 | 0.97 | 0.73 | 0.62 |
| **SCED** | - | 1.43 | 1.14 | 1.0 | 1.0 | 1.0 | - |

**Solución**

**Person Months**

**Person Months**

* **MODELO POST ARQUITECTURA**

**Cost Drivers:**

RELY = Low (0.75)

TIME = Very high (1.31)

ACAP = Very low (1.50)

PCAP = Very low (1.37)

TOOL = Low (1.12)

SCED = Very low (1.29)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cost Drivers** | **Very Low** | **Low** | **Nominal** | **High** | **Very High** | **Extra High** |
| RELY | 0.75 | 0.88 | 1.00 | 2.48 | 1.24 | 0.00 |
| DATA |  | 0.93 | 1.00 | 2.03 | 2.03 | 0.00 |
| CPLX | 0.75 | 0.88 | 1.00 | 2.83 | 1.41 | 0.00 |
| RUSE |  | 0.91 | 1.00 | 2.19 | 1.10 | 0.00 |
| DOCU | 0.89 | 0.95 | 1.00 | 3.12 | 1.56 | 0.00 |
| TIME |  |  | 1.00 | 1.11 | 1.31 | 1.67 |
| STOR |  |  | 1.00 | 1.06 | 1.21 | 1.57 |
| PVOL |  | 0.87 | 1.00 | 1.15 | 1.30 |  |
| ACAP | 1.50 | 1.22 | 1.00 | 0.83 | 0.67 |  |
| PCAP | 1.37 | 1.16 | 1.00 | 0.87 | 0.74 |  |
| PCON | 1.24 | 1.10 | 1.00 | 0.92 | 0.84 |  |
| AEXP | 1.22 | 1.10 | 1.00 | 0.89 | 0.81 |  |
| PEXP | 1.25 | 1.12 | 1.00 | 0.88 | 0.81 |  |
| LTEX | 1.22 | 1.10 | 1.00 | 0.91 | 0.84 |  |
| TOOL | 1.24 | 1.12 | 1.00 | 0.86 | 0.72 |  |
| SITE | 1.25 | 1.10 | 1.00 | 0.82 | 0.84 | 0.78 |
| SCED | 1.29 | 1.10 | 1.00 | 1.00 | 1.00 |  |

**Solución**

# Metodología de desarrollo

# Herramientas para el desarrollo del sistema

**Visual Studio**

Visual Studio es un entorno de desarrollo integrado (IDE) creado por Microsoft. Proporciona un conjunto completo de herramientas y servicios para el desarrollo de software, incluyendo la escritura de código, la depuración, el diseño de interfaces de usuario y la administración de proyectos.

Visual Studio admite múltiples lenguajes de programación, como C#, C++, Visual Basic, F# y otros. Al utilizar Visual Studio en el proyecto, hemos obtenido varios beneficios y ventajas que han contribuido al éxito y eficiencia del desarrollo:

* Amplia compatibilidad de lenguajes: Visual Studio ofrece soporte para una amplia gama de lenguajes de programación, lo cual me permitió seleccionar el lenguaje más adecuado para el proyecto.
* Entorno de desarrollo integrado completo: Visual Studio proporciona un conjunto completo de herramientas y características que agilizaron mi flujo de trabajo. El entorno integrado incluye un editor de código altamente funcional, herramientas de depuración avanzadas y capacidades de prueba integradas. Esto nos permitió realizar todas las tareas de desarrollo desde un único lugar, lo que mejoró la productividad y la eficiencia general.
* Depuración y análisis avanzados: La funcionalidad de depuración y análisis de Visual Studio resultó invaluable durante el desarrollo del proyecto. Nos permitió identificar y solucionar rápidamente errores y problemas en el código, lo que aceleró el proceso de depuración y garantizó un software más estable y libre de errores.

**C#**

Como lenguaje de programación, escogimos C#, pues la sintaxis es clara, legible, y nos facilita la comprensión y escritura del código. Además, cuenta con un amplio conjunto de bibliotecas y frameworks, un entorno de desarrollo integrado poderoso y una comunidad activa de desarrolladores que brindan soporte y recursos adicionales. Al utilizar una estructura de código bien definida y fácilmente comprensible, nos reduce la posibilidad de cometer errores.

Además de su sintaxis, C# nos ofrece otras características:

* Amplio conjunto de bibliotecas y frameworks: C# cuenta con un vasto conjunto de bibliotecas y frameworks en la plataforma .NET, que proporcionan una funcionalidad predefinida y reutilizable para una amplia gama de tareas comunes en el desarrollo de software. Estas bibliotecas y frameworks permiten ahorrar tiempo y esfuerzo al aprovechar soluciones ya existentes en lugar de tener que desarrollar todo desde cero.
* Entorno de desarrollo integrado (IDE): Visual Studio es el entorno de desarrollo integrado (IDE) principal para el desarrollo en C#. Proporciona herramientas poderosas que facilitan la escritura, depuración y prueba del código. La interfaz intuitiva de Visual Studio, junto con las numerosas características de productividad, como la finalización automática de código, la refactorización y la depuración visual, contribuyen a una experiencia de programación más eficiente y cómoda.

**Metodología Scrum**

1. Visión del proyecto:

* Desarrollar un sistema automatizado de gestión de envíos de paquetes para organizaciones de transporte, que permita asignar eficientemente vehículos y choferes, controlar los costos y el rendimiento, y proporcionar información útil sobre los movimientos y gastos de los vehículos.

1. Planificación del proyecto:

* Crear un backlog del producto que incluya los requerimientos del sistema, las funcionalidades principales y los objetivos del proyecto.
* Identificar los stakeholders y sus roles en el proyecto.
* Establecer un equipo Scrum que incluya al Product Owner, Scrum Master y miembros del equipo de desarrollo.

1. Sprint 0 (Preparación):

* Definir el objetivo del Sprint 0: establecer el entorno de trabajo, definir la arquitectura del sistema y crear una lista inicial de historias de usuario.
* Realizar reuniones de planificación y estimación para determinar las tareas necesarias para cumplir con el objetivo del Sprint 0.
* Crear un plan de proyecto y establecer las fechas y duraciones de los sprints.

1. Sprint 1:

* Realizar reuniones diarias de seguimiento (daily scrums) para que el equipo se mantenga sincronizado y actualizado sobre el progreso y los obstáculos.
* Desarrollar las historias de usuario prioritarias del backlog del producto.
* Realizar pruebas y asegurar la calidad del software desarrollado.
* Realizar una revisión del sprint al finalizar el periodo de desarrollo, demostrando el trabajo completado al Product Owner y otros stakeholders.
* Realizar una retrospectiva del sprint para identificar mejoras y lecciones aprendidas.

1. Sprints subsiguientes:

* Repetir los pasos del Sprint 1 para cada sprint subsiguiente, desarrollando nuevas historias de usuario y realizando las reuniones diarias, revisiones y retrospectivas correspondientes.
* Ajustar el backlog del producto según las necesidades y prioridades emergentes.
* Mantener una comunicación constante con los stakeholders para obtener retroalimentación y realizar ajustes en el proyecto según sea necesario.

1. Entrega del producto:

* Al finalizar los sprints planificados, realizar una revisión final del producto completo.
* Realizar pruebas exhaustivas y asegurar la calidad del software.
* Realizar una entrega final del producto al cliente o usuario final.

1. Mantenimiento y mejora continua:

* Realizar seguimiento y mantenimiento del sistema después de la entrega, solucionando problemas y realizando mejoras según sea necesario.
* Recopilar retroalimentación de los usuarios y stakeholders para identificar oportunidades de mejora.
* Utilizar la retroalimentación y los aprendizajes obtenidos para mejorar futuros proyectos y aplicarlos en el ciclo de desarrollo iterativo de Scrum.

Distribución de roles entre los integrantes:

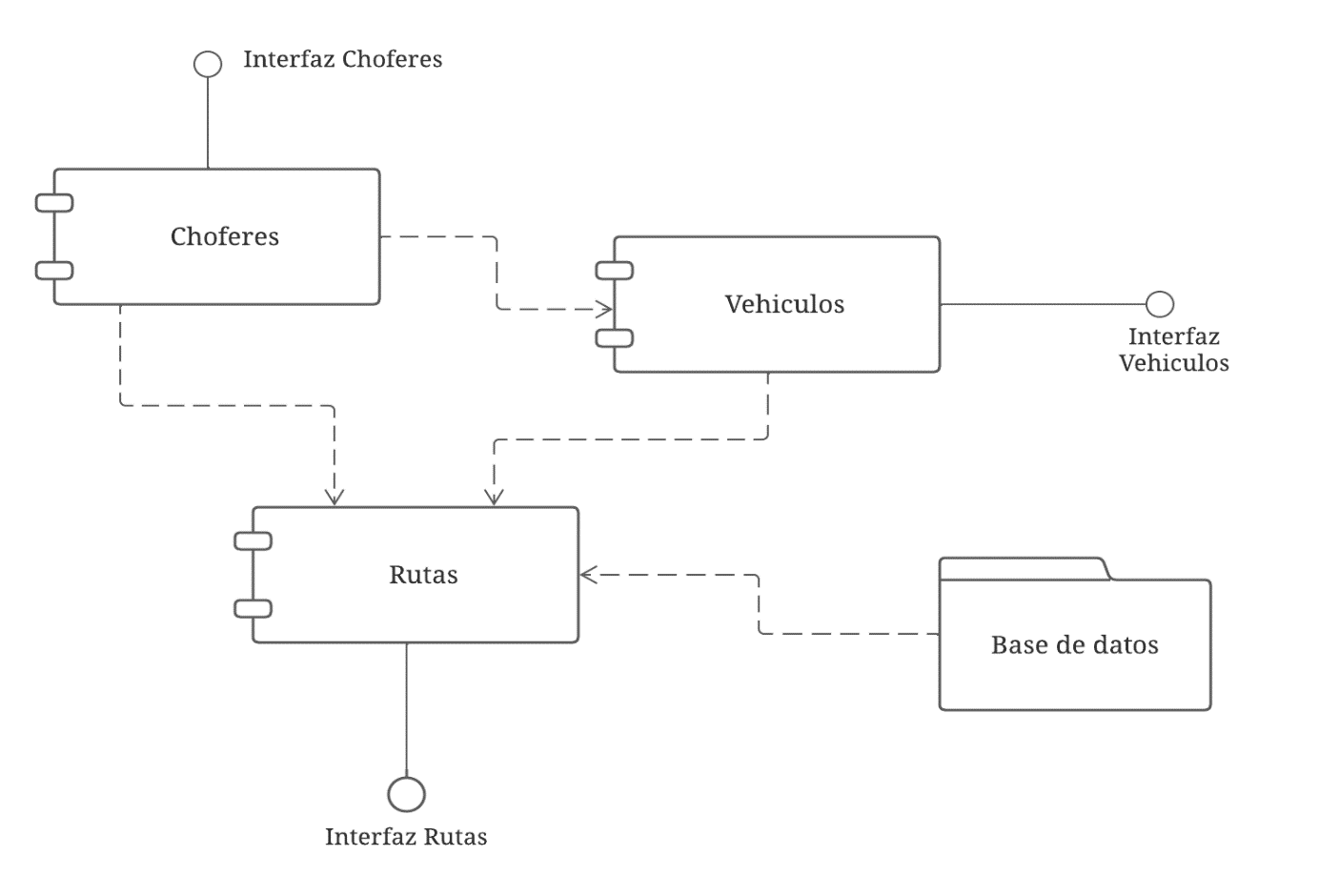
|  |  |  |
| --- | --- | --- |
| **Nombres** | **Contacto** | **Rol** |
| Bonifaz Chacón José | jose.bonifazc@ug.edu.ec | Líder de diseño arquitectónico |
| Calderón Serrano Kleidy | kleidy.calderons@ug.edu.ec | Líder de gestión y mejora |
| Cedeño Plúa Isaac | isaac.cedenop@ug.edu.ec | Líder conciliador |
| Coloma Guzmán John | John.colomag@ug.edu.ec | Líderes de requerimientos |
| Molina Aguilera Katherine | katherine.molinaa@ug.edu.ec | Líderes de requerimientos |
| Tomalá Magallanes Kevin | kevin.tomalam@ug.edu.ec | Secretario |

## Diagrama de clases

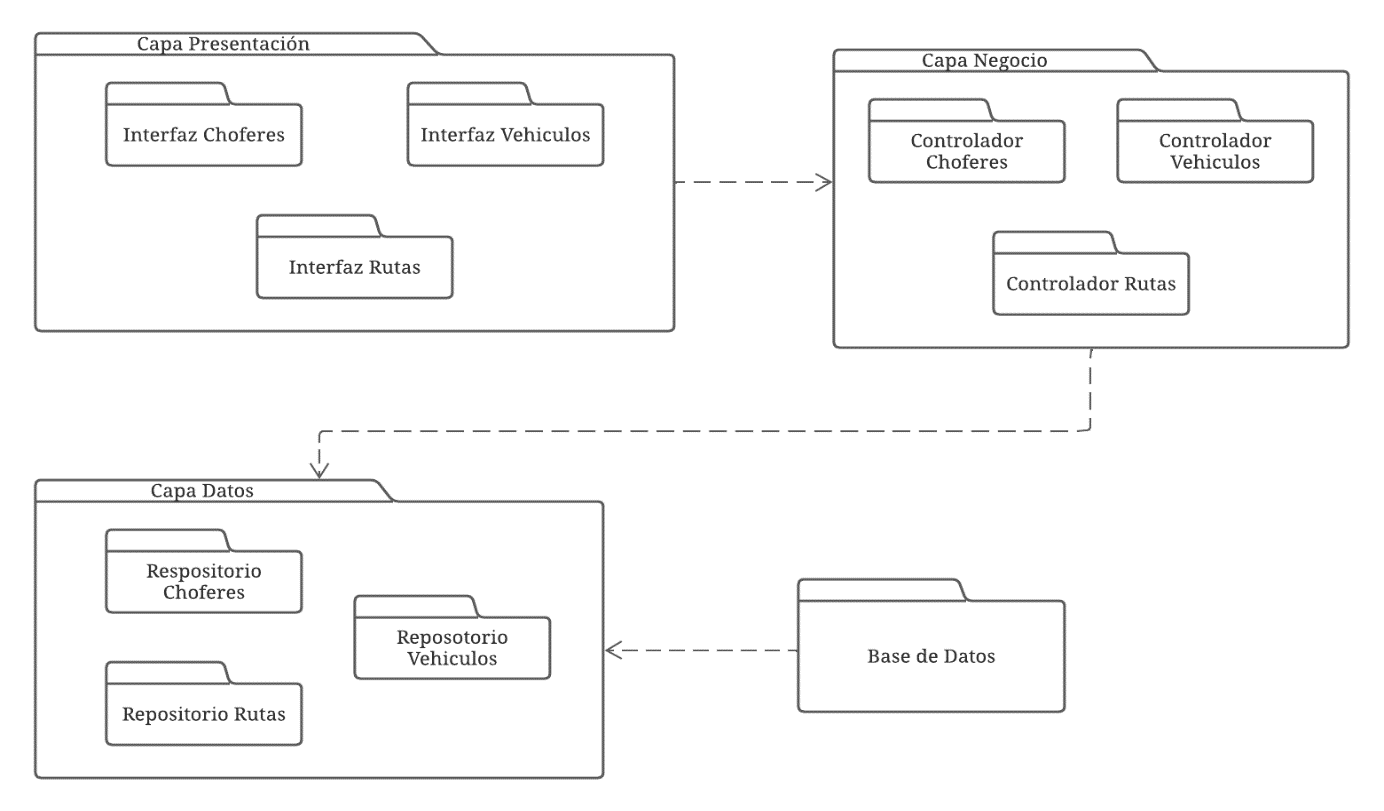
**Diagrama

Descripción generada automáticamente**

## Diagrama de componentes

****

## Diagrama de paquetes

****

# Resultado

El resultado del proyecto es un sistema de gestión para una organización dedicada a los servicios de envío de paquetes. El sistema tiene como objetivo principal optimizar las operaciones de la empresa, permitiendo una asignación eficiente de vehículos y choferes, un seguimiento y control adecuado de las rutas y entregas, y el registro de información relevante como el kilometraje y los gastos de combustible.

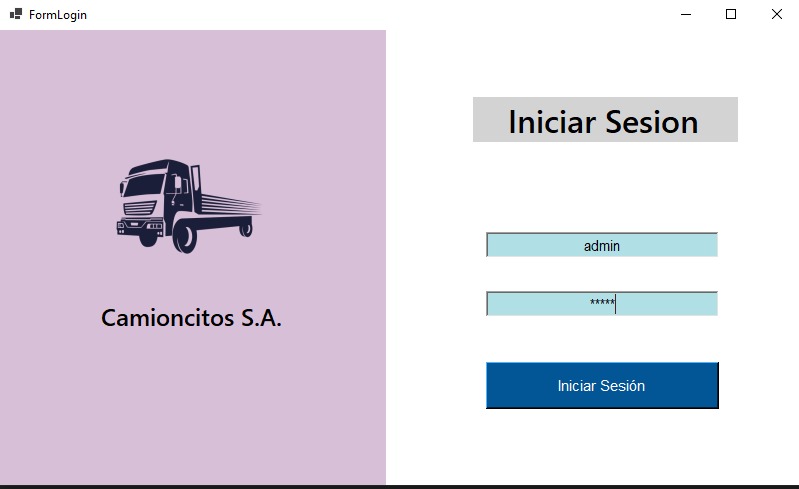
El proyecto documenta las decisiones de arquitectura más significativas tomadas en cuenta para el desarrollo del sistema, utilizando diferentes vistas de arquitectura para describir los aspectos del mismo.

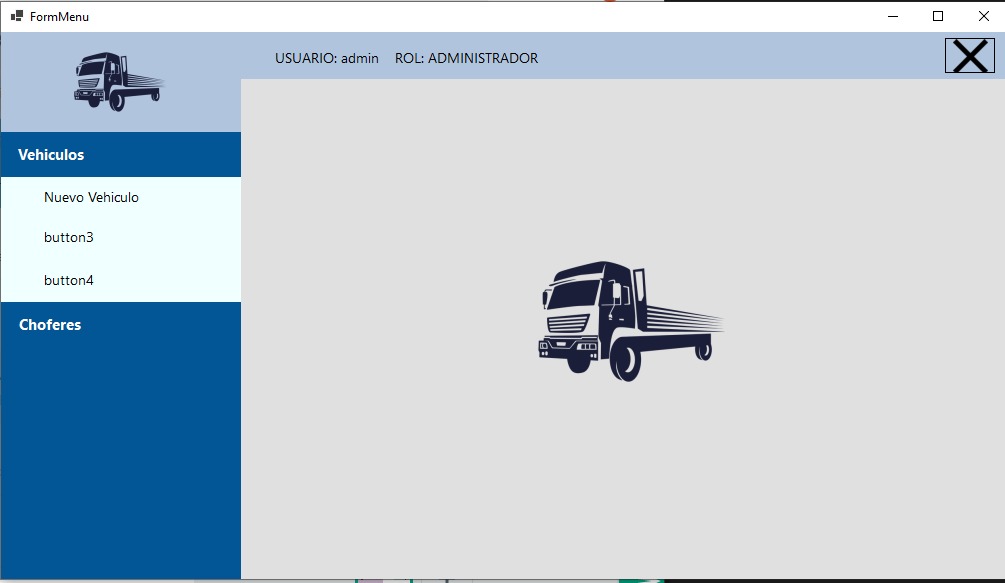
En cuanto al alcance del sistema, se espera que cumpla con las siguientes funcionalidades:

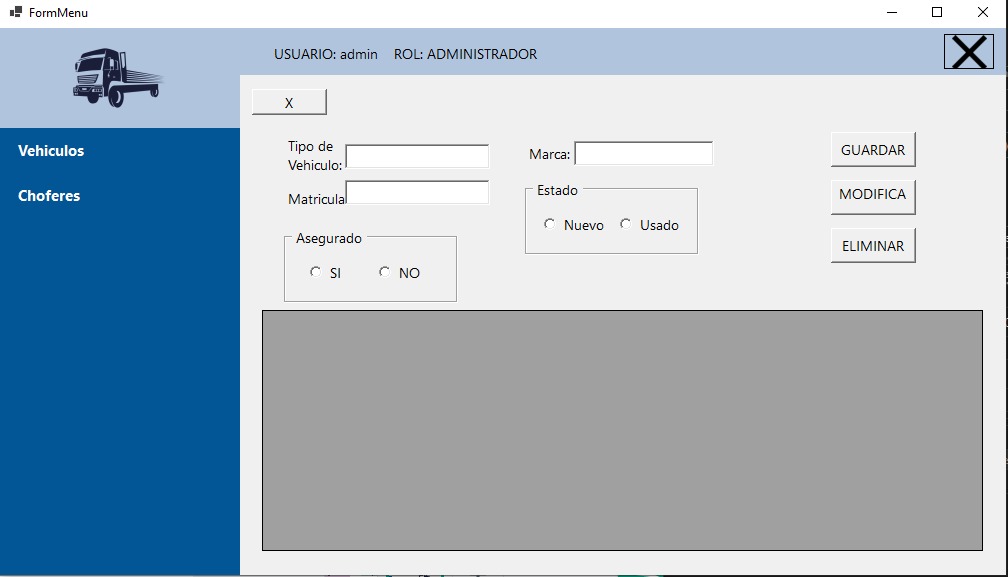
* Distribución eficiente y justa de tareas entre conductores y vehículos.
* Almacenamiento de registros de choferes, vehículos y rutas en una base de datos accesible en cualquier momento.
* Mejor control de las tareas realizadas por cada conductor y vehículo, así como los costos y rendimiento de cada uno.
* Seguimiento del movimiento de los vehículos en cada ruta.
* Registro del kilometraje recorrido por cada vehículo y los gastos de combustible generados.
* Estabilidad y funcionamiento sin interrupciones.
* Provisión de datos actualizados y precisos para una toma de decisiones más rápida y eficaz.

El sistema implementa una arquitectura Model-View-Controller (MVC), donde la capa de modelo se encarga de la gestión de datos y lógica de negocio, la capa de controlador maneja las solicitudes del usuario y coordina las interacciones, y la capa de vista se encarga de la presentación de la información al usuario.

En términos de calidad, el uso de la arquitectura MVC contribuye a la extensibilidad, confiabilidad y portabilidad del sistema. También se mencionan requisitos de hardware y software para la ejecución del sistema.







# Conclusión

Se ha desarrollado una solución de software basada en el patrón de arquitectura Model-View-Controller (MVC) para mejorar la gestión. El objetivo principal del proyecto es de automatizar los procesos de asignación de vehículos y choferes, optimizando así la eficiencia y maximizando las ganancias de la empresa.

La solución implementada permite distribuir de manera eficiente y justa las tareas entre los conductores y vehículos, evitando errores y asignaciones manuales. Además, registra y almacena los datos de choferes, vehículos, rutas y pagos relacionados con los servicios de entrega. El sistema proporciona un seguimiento del movimiento de los vehículos en cada ruta, registra el kilometraje recorrido y los gastos de combustible, brindando a la empresa una visión clara del rendimiento y los costos de cada vehículo.

Se ha tenido en cuenta la audiencia y las necesidades de los diferentes usuarios involucrados, como gerentes, secretarias, choferes, proveedores, empleados y clientes. El sistema ha sido diseñado para ser fácil de usar, confiable y estable, evitando interrupciones que puedan afectar las ventas y el cumplimiento de las tareas asignadas.