INDICE

[Información General. 3](#_Toc10889385)

[Propósito. 3](#_Toc10889386)

[Alcance. 3](#_Toc10889387)

[Personal involucrado. 3](#_Toc10889388)

[Básico. 3](#_Toc10889389)

[Convenciones, definiciones, acrónimos 3](#_Toc10889390)

[Referencias, 3](#_Toc10889391)

[Resumen 3](#_Toc10889392)

[Perspectiva del producto 3](#_Toc10889393)

[Funcionalidad del producto 3](#_Toc10889394)

[Características del producto 3](#_Toc10889395)

[Restricciones 3](#_Toc10889396)

[Restricciones 3](#_Toc10889397)

[Suposiciones 3](#_Toc10889398)

[Información funcional y no funcional. 3](#_Toc10889399)

[Proceso Funcional. 3](#_Toc10889400)

[Requerimientos Funcionales generales 3](#_Toc10889401)

[Requisitos comunes de las interfaces 4](#_Toc10889402)

[Interfaces de usuario 4](#_Toc10889403)

[Interfaces de hardware 4](#_Toc10889404)

[Interfaces de software 4](#_Toc10889405)

[Interfaces de comunicación 4](#_Toc10889406)

[Requerimientos Funcionales específicos 4](#_Toc10889407)

[Requisito funcional 1 4](#_Toc10889408)

[Requisito funcional 2 4](#_Toc10889409)

[Requerimientos No Funcionales 5](#_Toc10889410)

[Requisitos de rendimiento 5](#_Toc10889411)

[Seguridad 5](#_Toc10889412)

[Fiabilidad 5](#_Toc10889413)

[Disponibilidad 5](#_Toc10889414)

[Mantenibilidad 5](#_Toc10889415)

[Portabilidad 5](#_Toc10889416)

[Información técnica. 6](#_Toc10889417)

[Básico. 6](#_Toc10889418)

[Base Datos. 6](#_Toc10889419)

[Front End 6](#_Toc10889420)

[Vista “Login” 6](#_Toc10889421)

[Vista “registro”. 7](#_Toc10889422)

[Backend. 7](#_Toc10889423)

[Capa de la vista 7](#_Toc10889424)

[Capa de negocio 7](#_Toc10889425)

[Capa de datos 7](#_Toc10889426)

[Capa de servicio 7](#_Toc10889427)

[Seguridad. 8](#_Toc10889428)

[Complementos. 8](#_Toc10889429)

[Pendientes de definición. 8](#_Toc10889430)

[Pendientes Técnicos. 8](#_Toc10889431)

# Información General.

## Propósito.

El propósito de este documento es definir los requerimientos que la aplicación debe

tener, como una plataforma para la comunicación, control y seguimiento de

envíos terrestres de carga dentro del territorio colombiano. Esta especificación de requisitos de software está dirigida a ser leída y usada por usuarios o individuos que tengan interés en saber cómo funciona el sistema.

## Alcance.

**BuscaCarga** brindara información de primera mano a personas interesadas en el envío de cualquier tipo de carga con el propósito de obtener múltiples ofertas para los remitentes y transportadores mejorando el proceso de envíos y comunicación entre los actores dentro de territorio colombiano.

A fin de brindar la posibilidad de un servicio confiable y con ofertas competitivas para los remitentes.

## Personal involucrado.

|  |  |
| --- | --- |
| **Nombre** | Cristian Javier Ropero |
| **Rol** | Analista Programador |
| **Categoría Profesional** | Senior |
| **Responsabilidad** | Análisis de información, diseño y programación del SI |
| **Información de contacto** | [cjavieramaya@gmail.com](mailto:cjavieramaya@gmail.com) |

|  |  |
| --- | --- |
| **Nombre** | Pedro Alejandro Rojas |
| **Rol** | Analista Programador |
| **Categoría Profesional** | Senior |
| **Responsabilidad** | Análisis de información, diseño y programación del SI |
| **Información de contacto** | Parojas533@misena.edu.co |

## Básico.

## Convenciones, definiciones, acrónimos

|  |  |
| --- | --- |
| ***Nombre*** | ***Descripción*** |
| **Transportador** | Persona que utiliza vehículo para transportar mercancías a diferentes ciudades |
| **Remitente** | Persona que realiza envíos a diferentes ciudades utilizando transportadores |
| **RF** | Requerimiento Funcional |
| **RNF** | Requerimiento No Funcional |
| **API** | Una API es una interfaz de programación de aplicaciones (del inglés API: Application Programming Interface). Es un conjunto de rutinas que provee acceso a funciones de un determinado software |

## Referencias

|  |  |
| --- | --- |
| **Titulo del Documento** | **Referencia** |
| Standard IEEE 830 - 1998 | IEEE |

## Resumen

## Perspectiva del producto

Este ítem se basa en el árbol de problemas que realizaron con anticipación y de donde salió esa necesidad y que se piensa solucionar.

## Funcionalidad del producto

## Características del producto

## Restricciones

## Restricciones

## Suposiciones

# Información funcional y no funcional.

## Proceso Funcional.

|  |
| --- |
|  |

## Requerimientos Funcionales generales

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF01 |
| **Nombre del Requerimiento:** | Autentificación de Usuario. |
| **Características:** | Los usuarios deberán identificarse para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF05 * RNF08 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF02 |
| **Nombre del Requerimiento:** | Registrar Usuarios. |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario registrarse para los roles Transportador y Remitente, suministrando la información pertinente para poder acceder al sistema |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF03 |
| **Nombre del Requerimiento:** | Cargar soporte Pago Plan |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Transportador realizar el cargue del soporte de pago del pan para que este sea habilitado y hacer uso del sistema de ofertas |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |
| **Identificación del requerimiento:** | RF04 |
| **Nombre del Requerimiento:** | Ver solicitudes de Remitentes |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Transportador visualizar todas la solicitudes realizadas en el sistema por parte del Remitente |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF05 |
| **Nombre del Requerimiento:** | Ver solicitud Remitente |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Transportador visualizar la solicitud realizada por parte del Remitente para ver detalles de la solicitud |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF06 |
| **Nombre del Requerimiento:** | Generar Oferta |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Transportador realizar oferta de acuerdo al requerimiento de casa solicitud por parte del Remitente |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF07 |
| **Nombre del Requerimiento:** | Crear Solicitud |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Remitente crear una solicitud de carga con las características necesarias para detallar información de la carga |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF08 |
| **Nombre del Requerimiento:** | Ver ofertas realizadas |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Remitente visualizar las ofertas realizadas por parte de los Transportadores para que estas sean aprobadas |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |
| **Identificación del requerimiento:** | RF10 |
| **Nombre del Requerimiento:** | Aprobar Oferta |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Remitente aprobar oferta realizada por parte del Transportador permitiéndole elegir la mejor opción |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF11 |
| **Nombre del Requerimiento:** | Visualizar datos de Transportador |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Remitente visualizar los datos del transportador luego de aprobar la oferta sobre sus solicitud |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF12 |
| **Nombre del Requerimiento:** | Visualizar datos Remitente |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Transportador visualizar información de Remitente cuando este apruebe la oferta realizada |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF13 |
| **Nombre del Requerimiento:** | Consultar Perfil de usuario |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Transportador y Remitente consultar información de su perfil |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF14 |
| **Nombre del Requerimiento:** | Actualizar Perfil Usuario |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario Transportador y Remitente actualizar información respectiva de su perfil |
| **Requerimiento NO funcional:** | * RNF01 |
| **Prioridad del requerimiento:**  Alta | |

## Requisitos comunes de las interfaces

### Interfaces de usuario

Las interfaces de usuario son implementadas aplicando la experiencia de usuarios (UX) las cuales solo funcionaran en sistema operativo Android.

### Interfaces de hardware

Es necesario disponer de dispositivo móvil con periféricos cámara, GPS.

### Interfaces de software

Sistema Operativo Android superior a la versión 7

### Interfaces de comunicación

L a interfaz de comunicación que se utilizaran son FTP, http para el consumo de información de las bases de datos

## Requerimientos Funcionales específicos

### Requisito funcional 1

* Autentificación de Usuarios: los usuarios deberán identificarse para acceder a cualquier parte del sistema.
* El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad

### Requisito funcional 2

* **Consultar Información:** El sistema ofrecerá al usuario información general acerca de la Instrucción Académica, materias, Calendario de eventos.
* **Consultar Instrucción Académica:** Muestra información general sobre los cursos que la institución ofrece, en qué consiste, duración por período académico.
* **Consultar Materias:** Permite a los estudiantes ver las asignaciones que los docentes han publicado, descargar guía e información relevante a la materia y ver notas obtenidas en evaluaciones.
* **Consultar Calendario de Eventos:** Muestra a los usuarios información relevante a noticias u otros eventos planificados.

## Requerimientos No Funcionales

### Requisitos de rendimiento

* Garantizar que el diseño de las consultas u otro proceso no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red.

### Seguridad

* Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
* Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas.
* Facilidades y controles para permitir el acceso a la información al personal autorizado a través de Internet, con la intención de consultar y subir información pertinente para cada una de ellas.

### Fiabilidad

* El sistema debe tener una interfaz de uso intuitiva y sencilla
* La interfaz de usuario debe ajustarse a las características de la web de la institución, dentro de la cual estará incorporado el sistema de gestión de procesos y el inventario

### Disponibilidad

* La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado que permita la posible falla en cualquiera de sus componentes, contar con una contingencia, generación de alarmas.

### Mantenibilidad

* El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible
* La interfaz debe estar complementada con un buen sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones informáticas).

### Portabilidad

# Información técnica.

## Básico.

1. Las hojas tienen columnas escondidas. Las hojas con columnas escondidas tienen un objetivo de proceso y de formulación que ayuda al control del modelo. Es importante que cuando se requiera adicionar líneas o columnas se haga un análisis de impacto sobre el modelo y sobre las rutinas o macros que utilizan información o referencias de celdas y rangos fijos.

## Base Datos.

1. Esta base de datos es el repositorio donde se almacenan los valores comunes a los usuarios del Libro Business Case.
2. El concepto general de tener una base de datos con la información centralizada es tener el control de adición y modificación de datos utilizados en el cálculo del modelo de Business Case.
3. Modelo MER:

|  |
| --- |
|  |

1. Path: **localhost:3000**
2. Los perfiles administradores pueden modificar cualquier campo o tabla que reside en esta base datos.

Los perfiles usuarios pueden leer cualquier campo o tabla que reside en esta base datos.

1. Script

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema bc123

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema bc123

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `bc123` DEFAULT CHARACTER SET utf8 ;

USE `bc123` ;

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_pais`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_pais` (

`idpais` INT(11) NOT NULL AUTO\_INCREMENT,

`Cod\_pais` VARCHAR(45) NULL,

`nombre` VARCHAR(45) NULL DEFAULT 'Nombre de pais',

PRIMARY KEY (`idpais`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_departamento`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_departamento` (

`iddepartamento` INT(11) NOT NULL AUTO\_INCREMENT,

`Cod\_Departamento` INT NULL,

`nombre` VARCHAR(45) NULL DEFAULT NULL,

`idpais` INT(11) NOT NULL,

PRIMARY KEY (`iddepartamento`),

INDEX `fk\_tbl\_departamento\_tbl\_pais1\_idx` (`idpais` ASC) VISIBLE,

CONSTRAINT `fk\_tbl\_departamento\_tbl\_pais1`

FOREIGN KEY (`idpais`)

REFERENCES `bc123`.`tbl\_pais` (`idpais`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bc123`.`estados`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`estados` (

`idestados` INT NOT NULL AUTO\_INCREMENT,

`descripcion` VARCHAR(45) NULL,

`tipoEstado` VARCHAR(45) NULL,

PRIMARY KEY (`idestados`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_membresia`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_membresia` (

`id\_membresia` INT(11) NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(32) NOT NULL,

`valor` DOUBLE NOT NULL,

`soportePago` VARCHAR(128) NULL,

`estados\_idestados` INT NOT NULL,

PRIMARY KEY (`id\_membresia`),

INDEX `fk\_tbl\_membresia\_estados1\_idx` (`estados\_idestados` ASC) VISIBLE,

CONSTRAINT `fk\_tbl\_membresia\_estados1`

FOREIGN KEY (`estados\_idestados`)

REFERENCES `bc123`.`estados` (`idestados`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8

COMMENT = ' ';

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_municipio`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_municipio` (

`idmunicipio` INT(11) NOT NULL,

`nombre` VARCHAR(45) NULL DEFAULT NULL,

`cod\_municipio` VARCHAR(45) NULL,

`tbl\_departamento\_iddepartamento` INT(11) NOT NULL,

PRIMARY KEY (`idmunicipio`),

INDEX `fk\_tbl\_municipio\_tbl\_departamento1\_idx` (`tbl\_departamento\_iddepartamento` ASC) VISIBLE,

CONSTRAINT `fk\_tbl\_municipio\_tbl\_departamento1`

FOREIGN KEY (`tbl\_departamento\_iddepartamento`)

REFERENCES `bc123`.`tbl\_departamento` (`iddepartamento`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_permisos`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_permisos` (

`id\_permisos` INT(11) NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(32) NULL DEFAULT NULL,

PRIMARY KEY (`id\_permisos`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_rol`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_rol` (

`codigoRol` INT(11) NOT NULL AUTO\_INCREMENT,

`nombrerol` VARCHAR(32) NOT NULL,

PRIMARY KEY (`codigoRol`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_rol\_permisos`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_rol\_permisos` (

`idrolxpermiso` INT NOT NULL AUTO\_INCREMENT,

`fecha\_conexion` DATE NULL,

`tbl\_permisos\_id\_permisos` INT(11) NOT NULL,

`tbl\_rol\_codigoRol` INT(11) NOT NULL,

INDEX `fk\_tbl\_rol\_permisos\_tbl\_permisos1\_idx` (`tbl\_permisos\_id\_permisos` ASC) VISIBLE,

PRIMARY KEY (`idrolxpermiso`),

INDEX `fk\_tbl\_rol\_permisos\_tbl\_rol1\_idx` (`tbl\_rol\_codigoRol` ASC) VISIBLE,

CONSTRAINT `fk\_tbl\_rol\_permisos\_tbl\_permisos1`

FOREIGN KEY (`tbl\_permisos\_id\_permisos`)

REFERENCES `bc123`.`tbl\_permisos` (`id\_permisos`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_tbl\_rol\_permisos\_tbl\_rol1`

FOREIGN KEY (`tbl\_rol\_codigoRol`)

REFERENCES `bc123`.`tbl\_rol` (`codigoRol`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bc123`.`Usuario`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`Usuario` (

`idUsuario` INT NOT NULL AUTO\_INCREMENT,

`cedula` INT NULL,

`nombres` VARCHAR(64) NULL,

`contraseña` VARCHAR(45) NULL,

`correoElectronico` VARCHAR(64) NULL,

PRIMARY KEY (`idUsuario`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_usuario\_rol`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_usuario\_rol` (

`idUsuarioRol` INT NOT NULL AUTO\_INCREMENT,

`fecha\_conexion` DATE NULL,

`tbl\_rol\_codigoRol` INT(11) NOT NULL,

`Usuario\_idUsuario` INT NOT NULL,

PRIMARY KEY (`idUsuarioRol`),

INDEX `fk\_tbl\_usuario\_rol\_tbl\_rol1\_idx` (`tbl\_rol\_codigoRol` ASC) VISIBLE,

INDEX `fk\_tbl\_usuario\_rol\_Usuario1\_idx` (`Usuario\_idUsuario` ASC) VISIBLE,

CONSTRAINT `fk\_tbl\_usuario\_rol\_tbl\_rol1`

FOREIGN KEY (`tbl\_rol\_codigoRol`)

REFERENCES `bc123`.`tbl\_rol` (`codigoRol`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_tbl\_usuario\_rol\_Usuario1`

FOREIGN KEY (`Usuario\_idUsuario`)

REFERENCES `bc123`.`Usuario` (`idUsuario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bc123`.`tbl \_remitente`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl \_remitente` (

`id\_remitente` INT NOT NULL AUTO\_INCREMENT,

`telefonoFijo` VARCHAR(45) NULL,

`ciudadResidencia` VARCHAR(45) NULL,

`Usuario\_idUsuario` INT NOT NULL,

PRIMARY KEY (`id\_remitente`),

INDEX `fk\_tbl \_remitente\_Usuario1\_idx` (`Usuario\_idUsuario` ASC) VISIBLE,

CONSTRAINT `fk\_tbl \_remitente\_Usuario1`

FOREIGN KEY (`Usuario\_idUsuario`)

REFERENCES `bc123`.`Usuario` (`idUsuario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_transportador`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_transportador` (

`id\_transportador` INT NOT NULL AUTO\_INCREMENT,

`experiencia` INT NULL,

`tbl\_membresia\_id\_membresia` INT(11) NOT NULL,

`Usuario\_idUsuario` INT NOT NULL,

PRIMARY KEY (`id\_transportador`),

INDEX `fk\_tbl\_transportador\_tbl\_membresia1\_idx` (`tbl\_membresia\_id\_membresia` ASC) VISIBLE,

INDEX `fk\_tbl\_transportador\_Usuario1\_idx` (`Usuario\_idUsuario` ASC) VISIBLE,

CONSTRAINT `fk\_tbl\_transportador\_tbl\_membresia1`

FOREIGN KEY (`tbl\_membresia\_id\_membresia`)

REFERENCES `bc123`.`tbl\_membresia` (`id\_membresia`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_tbl\_transportador\_Usuario1`

FOREIGN KEY (`Usuario\_idUsuario`)

REFERENCES `bc123`.`Usuario` (`idUsuario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`tbl\_calificacion`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`tbl\_calificacion` (

`id\_calificacion` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(16) NULL,

`estado` TINYINT(1) NULL,

PRIMARY KEY (`id\_calificacion`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`Calificacion\_Remitente`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`Calificacion\_Remitente` (

`idCalificacionRemitente` INT NOT NULL AUTO\_INCREMENT,

`fecha` DATE NULL,

`comentario` TEXT(512) NULL,

`tbl\_transportador\_id\_transportador` INT NOT NULL,

`tbl \_remitente\_id\_remitente` INT NOT NULL,

`tbl\_calificacion\_id\_calificacion` INT NOT NULL,

PRIMARY KEY (`idCalificacionRemitente`),

INDEX `fk\_Calificacion\_Remitente\_tbl\_transportador1\_idx` (`tbl\_transportador\_id\_transportador` ASC) VISIBLE,

INDEX `fk\_Calificacion\_Remitente\_tbl \_remitente1\_idx` (`tbl \_remitente\_id\_remitente` ASC) VISIBLE,

INDEX `fk\_Calificacion\_Remitente\_tbl\_calificacion1\_idx` (`tbl\_calificacion\_id\_calificacion` ASC) VISIBLE,

CONSTRAINT `fk\_Calificacion\_Remitente\_tbl\_transportador1`

FOREIGN KEY (`tbl\_transportador\_id\_transportador`)

REFERENCES `bc123`.`tbl\_transportador` (`id\_transportador`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Calificacion\_Remitente\_tbl \_remitente1`

FOREIGN KEY (`tbl \_remitente\_id\_remitente`)

REFERENCES `bc123`.`tbl \_remitente` (`id\_remitente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Calificacion\_Remitente\_tbl\_calificacion1`

FOREIGN KEY (`tbl\_calificacion\_id\_calificacion`)

REFERENCES `bc123`.`tbl\_calificacion` (`id\_calificacion`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`Calificacion\_Transportador`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`Calificacion\_Transportador` (

`idCalificacionTransportador` INT NOT NULL AUTO\_INCREMENT,

`fecha` DATE NULL,

`comentario` TEXT(512) NULL,

`tbl \_remitente\_id\_remitente` INT NOT NULL,

`tbl\_transportador\_id\_transportador` INT NOT NULL,

`tbl\_calificacion\_id\_calificacion` INT NOT NULL,

PRIMARY KEY (`idCalificacionTransportador`),

INDEX `fk\_Calificacion\_Transportador\_tbl \_remitente1\_idx` (`tbl \_remitente\_id\_remitente` ASC) VISIBLE,

INDEX `fk\_Calificacion\_Transportador\_tbl\_transportador1\_idx` (`tbl\_transportador\_id\_transportador` ASC) VISIBLE,

INDEX `fk\_Calificacion\_Transportador\_tbl\_calificacion1\_idx` (`tbl\_calificacion\_id\_calificacion` ASC) VISIBLE,

CONSTRAINT `fk\_Calificacion\_Transportador\_tbl \_remitente1`

FOREIGN KEY (`tbl \_remitente\_id\_remitente`)

REFERENCES `bc123`.`tbl \_remitente` (`id\_remitente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Calificacion\_Transportador\_tbl\_transportador1`

FOREIGN KEY (`tbl\_transportador\_id\_transportador`)

REFERENCES `bc123`.`tbl\_transportador` (`id\_transportador`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Calificacion\_Transportador\_tbl\_calificacion1`

FOREIGN KEY (`tbl\_calificacion\_id\_calificacion`)

REFERENCES `bc123`.`tbl\_calificacion` (`id\_calificacion`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`paquete`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`paquete` (

`idpaquete` INT NOT NULL AUTO\_INCREMENT,

`ancho` INT NULL,

`alto` INT NULL,

`peso` INT NULL,

`descripcion` TEXT NULL,

`fechaPublicacion` DATETIME NULL,

PRIMARY KEY (`idpaquete`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`vehiculo`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`vehiculo` (

`idvehiculo` INT NOT NULL AUTO\_INCREMENT,

`placa` VARCHAR(6) NULL,

`modelo` YEAR NULL,

PRIMARY KEY (`idvehiculo`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`destino`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`destino` (

`iddestino` INT NOT NULL AUTO\_INCREMENT,

`tbl\_municipio\_idmunicipio` INT(11) NOT NULL,

PRIMARY KEY (`iddestino`),

INDEX `fk\_destino\_tbl\_municipio1\_idx` (`tbl\_municipio\_idmunicipio` ASC) VISIBLE,

CONSTRAINT `fk\_destino\_tbl\_municipio1`

FOREIGN KEY (`tbl\_municipio\_idmunicipio`)

REFERENCES `bc123`.`tbl\_municipio` (`idmunicipio`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`origen`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`origen` (

`idorigen` INT NOT NULL AUTO\_INCREMENT,

`tbl\_municipio\_idmunicipio` INT(11) NOT NULL,

PRIMARY KEY (`idorigen`),

INDEX `fk\_origen\_tbl\_municipio1\_idx` (`tbl\_municipio\_idmunicipio` ASC) VISIBLE,

CONSTRAINT `fk\_origen\_tbl\_municipio1`

FOREIGN KEY (`tbl\_municipio\_idmunicipio`)

REFERENCES `bc123`.`tbl\_municipio` (`idmunicipio`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`envio`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`envio` (

`idenvio` INT NOT NULL AUTO\_INCREMENT,

`detalle` TEXT(256) NULL,

`fechaenvio` DATE NULL,

`fechaentrega` DATE NULL,

`fecharecogida` DATE NULL,

`paquete\_idpaquete` INT NOT NULL,

`vehiculo\_idvehiculo` INT NOT NULL,

`destino\_iddestino` INT NOT NULL,

`origen\_idorigen` INT NOT NULL,

`tbl\_transportador\_id\_transportador` INT NOT NULL,

PRIMARY KEY (`idenvio`),

INDEX `fk\_envio\_paquete1\_idx` (`paquete\_idpaquete` ASC) VISIBLE,

INDEX `fk\_envio\_vehiculo1\_idx` (`vehiculo\_idvehiculo` ASC) VISIBLE,

INDEX `fk\_envio\_destino1\_idx` (`destino\_iddestino` ASC) VISIBLE,

INDEX `fk\_envio\_origen1\_idx` (`origen\_idorigen` ASC) VISIBLE,

INDEX `fk\_envio\_tbl\_transportador1\_idx` (`tbl\_transportador\_id\_transportador` ASC) VISIBLE,

CONSTRAINT `fk\_envio\_paquete1`

FOREIGN KEY (`paquete\_idpaquete`)

REFERENCES `bc123`.`paquete` (`idpaquete`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_envio\_vehiculo1`

FOREIGN KEY (`vehiculo\_idvehiculo`)

REFERENCES `bc123`.`vehiculo` (`idvehiculo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_envio\_destino1`

FOREIGN KEY (`destino\_iddestino`)

REFERENCES `bc123`.`destino` (`iddestino`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_envio\_origen1`

FOREIGN KEY (`origen\_idorigen`)

REFERENCES `bc123`.`origen` (`idorigen`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_envio\_tbl\_transportador1`

FOREIGN KEY (`tbl\_transportador\_id\_transportador`)

REFERENCES `bc123`.`tbl\_transportador` (`id\_transportador`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `bc123`.`oferta`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bc123`.`oferta` (

`idoferta` INT NOT NULL AUTO\_INCREMENT,

`valor` VARCHAR(45) NULL,

`decripcion` TEXT(256) NULL,

`envio\_idenvio` INT NOT NULL,

`tbl \_remitente\_id\_remitente` INT NOT NULL,

PRIMARY KEY (`idoferta`),

INDEX `fk\_oferta\_envio1\_idx` (`envio\_idenvio` ASC) VISIBLE,

INDEX `fk\_oferta\_tbl \_remitente1\_idx` (`tbl \_remitente\_id\_remitente` ASC) VISIBLE,

CONSTRAINT `fk\_oferta\_envio1`

FOREIGN KEY (`envio\_idenvio`)

REFERENCES `bc123`.`envio` (`idenvio`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_oferta\_tbl \_remitente1`

FOREIGN KEY (`tbl \_remitente\_id\_remitente`)

REFERENCES `bc123`.`tbl \_remitente` (`id\_remitente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

## Front End

El fornt end se realizo en Ionic 4

### Vista “Login”

Esta vista permite el ingreso al sistema de usuarios q están registrados es la base de datos, sino están registrados, no pueden acceder alsistema. El registro se hace en la empresa cliente

Se definen los valores que impactan la funcionalidad de la vista.

1. Se definen las celdas que deben ser obligatorias al diligenciar un business case.
2. Se definen los centros de costo.
3. Se define que centros de costo componen una ayuda para celdas de selección.

### Vista “registro”.

Cada una de las tablas diseñadas en esta hoja está separada por una columna en BLANCO, esta columna no debe ser borrada, modificada.

Se encuentra toda información base de operación del libro. El objetivo de esta hoja es encontrar centralizada la adición, modificación de valores utilizados en el proceso de fabricación de un business case.

1. Se encuentra las descripciones que son utilizadas para asignar un grupo de ayuda a una celda definida por parámetros.
2. Se encuentran valores asociados a la descripción de ayuda. Este valor es utilizado para afectar una celda y a su vez comprometida en los cálculos o formulación del Business Case.

## Backend.

El backend se realizo en asp .net core 3.0 con una arquitectura de 4 capas

### Capa de la vista

En la capa de la vista se codifico la vista user para acceder a registrar los usuarios por medio de usuarios administradores y se encuentra en la ruta 10.100.100.200/user

En la capa de la vista se codifico la vista login para acceder a registrar los usuarios en la vista user por medio de usuarios administradores y se encuentra en la ruta 10.100.100.200/login

En la capa de la vista se codifico la vista user para acceder a registrar los usuarios por medio de usuarios administradores y se encuentra en la ruta 10.100.100.200/user

### Capa de negocio

Se manejó la lógica de negocio de la app por medio de las clases A, B, y la clase C solo funciona como modelo de nuestro proyecto

### Capa de datos

Se creó un modelo que se conecta a nuestra base de datos que esta en el servidor 10.100.100.255:3306 por medio del usuario aaa y el password bbb.

### Capa de servicio

Se crearon las clases A, b, c, d y e que permite el uso de la api de nuestro servidor 10.100.100.255:3306 por medio del usuario aaa y el password bbb.

La clase A nos realiza los métodos get, post, put y delete de la tabla mama

La clase B nos realiza los métodos get, post y delete de la tabla papa

La clase B nos realiza los métodos get, de la tabla tio

## Seguridad.

1. El código tiene una clase que se encarga de encriptar las contraseñas del usuario por medio del siguiente método de la clase encrip:
2. La conexión con las api maneja un nivel de autenticación basado en auth 2.0 .

## Complementos.

Para compilar el programa se debe instalar el framework Tequendama en el servidor XXX.xxx.YYY.yyy en modo administrador.

El servidor de base de datos debe tener 1 Tb de RAM para realizar el procesamiento.

# Pendientes de definición.

* Participación perfiles Comercial
* Participación perfiles Cuenta
* Participación perfiles Diferenciación
* Valor flujo de caja
* Resumen para BC Servicio
* Resumen para BC Producto
* Escenarios
* Plantilla
* Informes de análisis

# Pendientes Técnicos.

* Ocultar hojas
* Documentar Solución
* Actualizar base de datos ABC.bak