

Universidad ORT Uruguay

Facultad de Ingeniería

Sistema de Gestión de Envíos

Entregado como requisito obligatorio para el curso de Programación I
Grupo 11



Jorge Sebastián Labadie Bentos (278119)

Noviembre 2021

RESUMEN

El sitio de gestión de envíos Online surge a partir de la necesidad de nuestro cliente, Curso de Programación I de las carreras Analista Programador/Analista en Tecnologías de la Información.

El proyecto, desarrollado en HTML 5/Javascript, se centra en mejorar la petición de envíos por parte de los usuarios finales y la gestión interna de los mismos por parte de las empresas/administradores.

Tabla de contenido

RESUMEN.....	2
ENUMERACIÓN DE LAS FUNCIONALIDADES DESARROLLADAS.....	4
DESCRIPCIÓN DE LAS ESTRUCTURAS DE DATOS UTILIZADAS	5
VARIABLES GLOBALES	6
DOCUMENTO DE ANALISIS	7
CODIGO COMPLETO.....	21
HTML.....	21
Main.js	22
Carga.js	23
Funciones.js	31
INFORME DE TESTING.....	52
INFORMACION PRECARGADA.....	53
Persona	53
TipoVehiculo	53
Empresa	53
Envio	53

ENUMERACIÓN DE LAS FUNCIONALIDADES DESARROLLADAS.

Para el correcto funcionamiento del sistema se desarrollaron las siguientes funcionalidades:

- Administrar empresas (habilitar y deshabilitar)
- Crear Tipos de Vehículos
- Ver estadísticas (administrador)
- Registrar Usuarios
- Ingresar Usuario
- Ver Solicitudes Pendientes
- Ver Solicitudes en Transito
- Ver Estadísticas (empresa)
- Realizar Solicitud Envío
- Ver Mis Solicitudes
- Ver Estadísticas (persona)

DESCRIPCIÓN DE LAS ESTRUCTURAS DE DATOS UTILIZADAS

Clase	Atributos incluidos	Descripción del atributo
Persona	Cedula	Númerica entera. Puede contener entre 7 y 9 dígitos. Identifica al funcionario.
	Nombre	String. Nombre propio del funcionario, debe contener al menos 3 caracteres. Solo puede contener caracteres alfabéticos, espacios y guiones.
	Apellido	String. Apellido del funcionario, debe contener al menos 3 caracteres. Solo puede contener caracteres alfabéticos, espacios y guiones.
	Usuario	String. Usuario único con el que podrá ingresar a la aplicación para gestionar sus envíos.
	Password	String. Contraseña para el acceso al sistema.
TipoVehiculo	Id	Númerica entera. Identificador único del tipo de vehículo. Auto numérico secuencial, se incrementa en 1 automáticamente cada vez que se ingresa un nuevo tipo de vehículo.
	Nombre	String. Nombre asignado al cargo, debe contener al menos 3 caracteres.
Empresa	Rut	String Rut empresarial.
	RazonSocial	String Razón social de la empresa.
	NombreFantasia	String Nombre de fantasía de empresa.
	Usuario	String Usuario de empresa con el que podrá ingresar a la aplicación.
	Password	String Contraseña de usuario.
	TipoVehiculo	TipoVehiculo Tipo de vehículo con el que operara la empresa.
	Estado	Boolean Estado de la empresa puede ser H (habilitada), D (deshabilitada).
Envio	Persona	Persona Persona asociada al envío.
	TipoVehiculo	TipoVehiculo Tipo de Vehículo asociado al envío.
	Distancia	Númerica Entera Distancia aproximada del envío (en KM).
	Foto	String Foto del envío (base 64).
	Estado	String Estado del envío, puede ser 'Pendiente', 'En Transito', 'Finalizada'.
	Empresa	Empresa Empresa asociada al envío (si no tiene el valor será "").
	Descripción	String Descripción breve sobre el producto.

VARIABLES GLOBALES

Nombre	Tipo	Descripción/Observaciones
personas	Array indexado. Cada elemento almacena un objeto correspondiente a una persona.	Almacena las personas registradas en la aplicación (incluye al administrador) Se inicializa con personas por defecto al iniciar la aplicación.
empresas	Array indexado. Cada elemento almacena un objeto correspondiente a una empresa.	Almacena las empresas registradas en la aplicación. Se inicializa con empresas por defecto al iniciar la aplicación.
tiposVehiculos	Array indexado. Cada elemento almacena un objeto correspondiente a un tipo de vehículo.	Almacena los tipos de vehículos existentes en la aplicación. Se inicializa con tipo de vehículos por defecto al iniciar la aplicación.
envios	Array indexado. Cada elemento almacena un objeto correspondiente a un envío.	Contador que almacena el último valor de id que se asignó a un cargo. Se utiliza para generar el Id que se asignará a un nuevo cargo ingresado.
tipoPerfilLogged	Numérico Entero	Tipo de perfil que ha iniciado sesión 1-Persona 2-Empresa 3-Administrador
isLogged	Boolean	Flag que se utiliza para saber si hay una persona o empresa que ha iniciado sesión. Se inicializa en false al iniciar la aplicación.
userLogged	Persona o Empresa	Datos del usuario que ha iniciado sesión.

DOCUMENTO DE ANALISIS

1. Descripción general del problema a resolver

Como usuario poder gestionar solicitudes de envíos de mercadería/artículos a distintas empresas, y como empresa contar con diferentes vehículos para hacerse cargo de determinadas entregas

1.1. Tipos de usuario del sistema

El sistema contará con 3 tipos de usuarios:

- Administrador ● Empresa
- Persona

1.2. Listado de funcionalidades

F01 – Administrar Empresas – Usuario/s: Administrador

F02 – Habilitar Empresa – Usuario/s: Administrador

F03 – Deshabilitar Empresa – Usuario/s: Administrador

F04 – Crear Tipos de Vehículos – Usuario/s: Administrador

F05 – Ver Estadísticas – Usuario/s: Administrador

F06 – Registrarse– Usuario/s: Persona, Empresa

F07 – Ingresar– Usuario/s: Persona, Empresa

F08 – Ver Solicitudes Pendientes – Usuario/s: Empresa

F09 – Ver Solicitudes En Tránsito – Usuario/s: Empresa

F10 – Ver Estadísticas – Usuario/s: Empresa

F11 – Realizar Solicitud Envío– Usuario/s: Persona

F12 – Ver Mis Solicitudes – Usuario/s: Persona

F13 – Ver Estadísticas – Usuario/s: Persona

2. Detalle de Funcionalidades

A continuación, se presenta el detalle de cada una de las funcionalidades a resolver en el obligatorio. Las funcionalidades están ordenadas por tipo de usuario.

2.1 F01 – Administrar Empresas

2.1.1 Acceso

Administrador

2.1.2 Descripción

Llevar al administrador a un formulario donde pueda habilitar o deshabilitar una empresa

2.1.3 Interfaz de usuario

The screenshot displays the 'ADMINISTRAR EMPRESAS' (Manage Companies) interface. At the top, there is a navigation bar with the title 'ADMINISTRADOR' and three menu items: 'ADMINISTRAR EMPRESAS', 'CREAR TIPOS DE VEHICULO', and 'VER ESTADISTICAS'. Below the navigation bar is a search bar labeled 'Buscar Empresa' with a magnifying glass icon. The main content area shows a grid of 15 company cards arranged in 5 rows and 3 columns. Each card represents a company 'Empresa S.A.' and contains the following information: 'Rut' (3559634), 'Razon Social' (3559634), 'Estado' (HABILITADA), and two status indicators: a green checkmark in a box and a red 'X' in a box. The interface is designed for an administrator to manage the status of various companies.

2.1.4 Validaciones

En el caso de presionar el botón de administrar empresas, se validara que se este logueado como administrador

2.2F02 – Habilitar Empresa

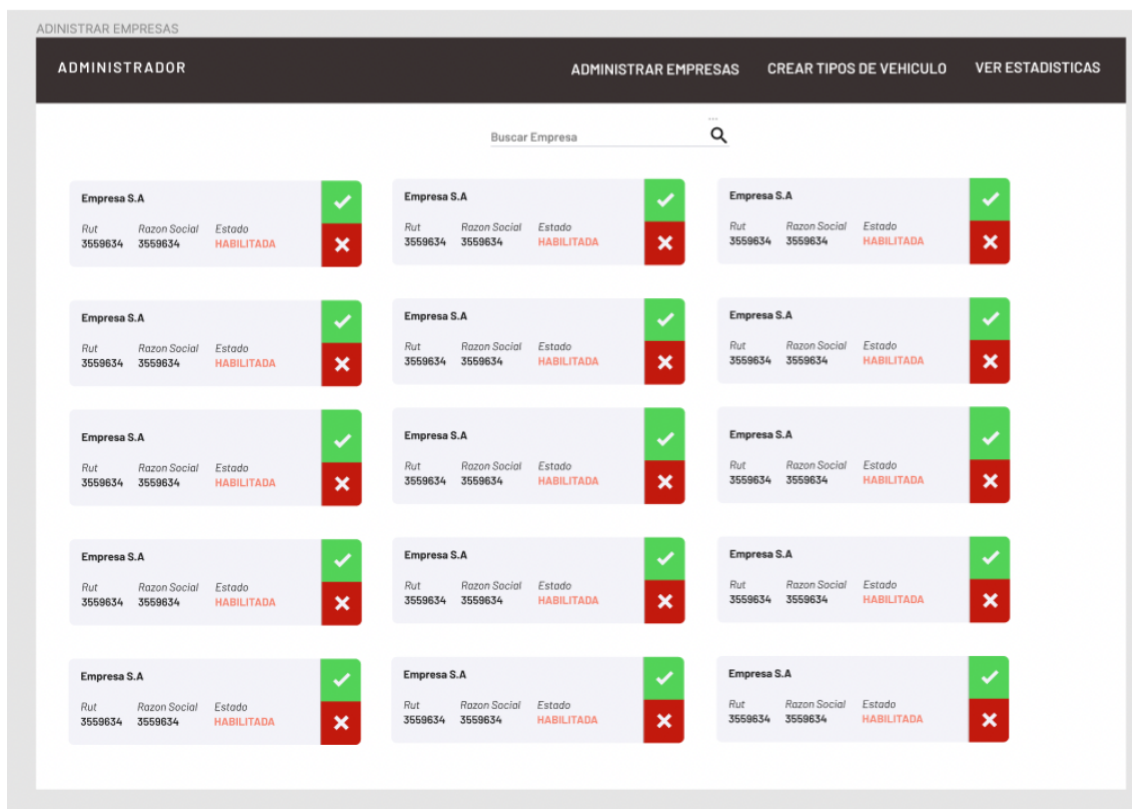
2.2.1 Acceso

Administrador

2.2.2 Descripción

Pasar una empresa a estado habilitado

2.2.3 Interfaz de usuario



2.2.4 Validaciones

Al presionar el botón de habilitar de una empresa se validará que se esté logueado como administrador y que dicha empresa se encuentre deshabilitada.

2.3F03 – Deshabilitar Empresa

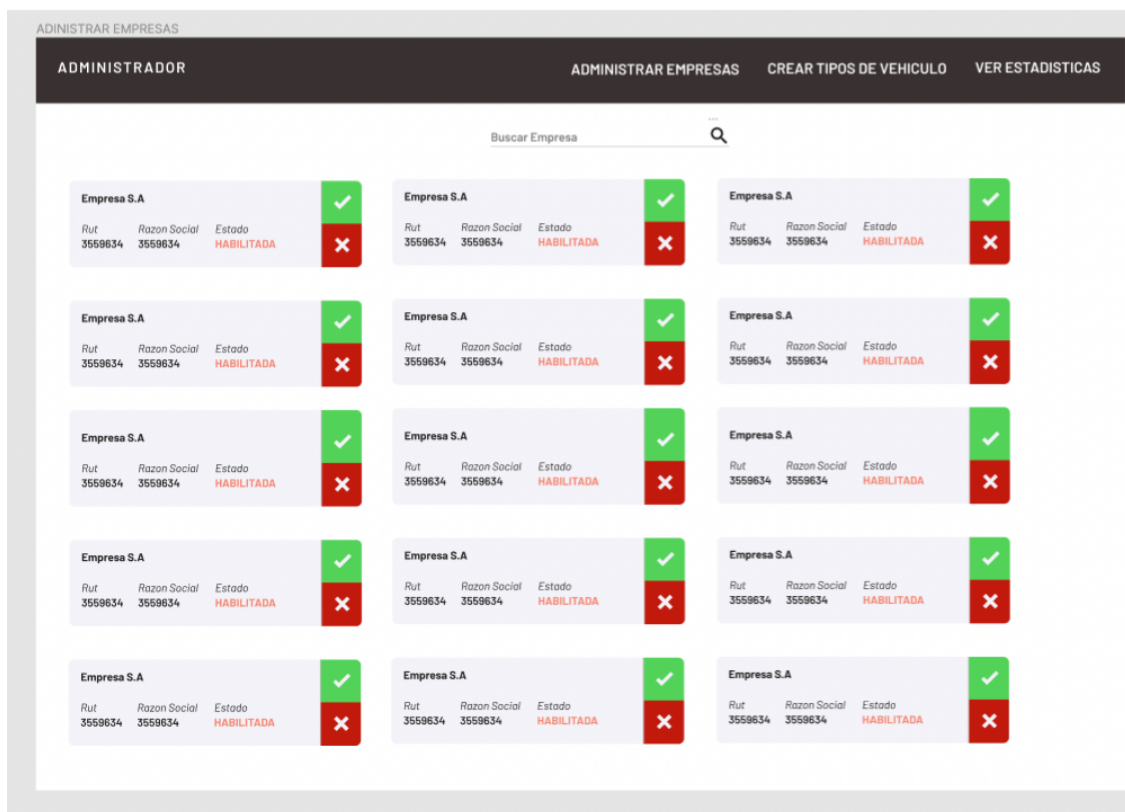
2.3.1 Acceso

Administrador

2.3.2 Descripción

Pasar una empresa a estado deshabilitado.

2.3.3 Interfaz de usuario



2.3.4 Validaciones

Al presionar el botón de deshabilitar de una empresa se validará que se esté logueado como administrador y que dicha empresa se encuentre habilitada.

2.4F04 – Crear Tipos de Vehículos

2.4.1 Acceso

Administrador

2.4.2 Descripción

Llevar al usuario a un formulario donde se listaran los tipos de vehículos existentes y donde se podrán dar de alta tipos de vehículos.

2.4.3 Interfaz de usuario

CREAR TIPOS DE VEHICULO

ADMINISTRADOR ADMINISTRAR EMPRESAS CREAR TIPOS DE VEHICULO VER ESTADISTICAS

Nombre +

12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta
12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta
12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta
12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta
12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta
12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta
12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta
12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta
12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta
12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta	12 - Camioneta

2.4.4 Validaciones

Al presionar el botón de crear tipo de vehículo se validará que se esté logueado como administrador, una vez dentro del formulario al presionar el botón alta de tipo de vehículo se validará que no haya otro tipo de vehículo con el mismo nombre.

2.5F05 – Ver Estadísticas

2.5.1 Acceso

Administrador

2.5.2 Descripción

Llevar al usuario a un formulario donde pueda ver la distancia recorrida por cada empresa, (solo de envíos finalizados)

2.5.3 Interfaz de usuario

VER ESTADÍSTICAS

ADMINISTRADOR

ADMINISTRAR EMPRESAS

CREAR TIPOS DE VEHICULO

VER ESTADÍSTICAS

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa S.A				K.M
Rut	Razon Social	Estado		
3559634	3559634	HABILITADA		32131231

Empresa			
---------	--	--	--

2.5.4 Validaciones

Al presionar el botón de ver estadísticas se validará que el tipo de perfil del usuario logeado actualmente sea administrador.

2.6F06 – Registrarse

2.6.1 Acceso

Persona y Empresa

2.6.2 Descripción

Llevar al usuario a un formulario que dependiendo del tipo de perfil que seleccione, le exigirá ciertos datos para poder registrarse en la aplicación.

2.6.3 Interfaz de usuario

The image displays two side-by-side registration forms. The left form is titled 'REGISTRO PERSONA' and the right form is titled 'REGISTRO EMPRESA'. Both forms have a 'BIENVENIDO' header. The 'REGISTRO PERSONA' form includes fields for 'CEDULA', 'NOMBRE USUARIO', 'NOMBRE', 'APELLIDO', and 'CONTRASEÑA', along with a 'PERSONA' dropdown menu. The 'REGISTRO EMPRESA' form includes fields for 'RUT', 'RAZÓN SOCIAL', 'NOMBRE DE FANTASIA', 'NOMBRE DE USUARIO', 'CONTRASEÑA', and 'TIPO DE VEHICULO' (a dropdown menu), along with an 'EMPRESA' dropdown menu. Both forms feature a blue 'REGISTRARSE' button and a blue '← VOLVER A INGRESAR' button.

2.6.4 Validaciones

Al presionar el botón de registrarse se validará que dependiendo del tipo de usuario elegido haya completado todos los datos obligatorios y que cada campo cumple con los criterios de aceptación. Por supuesto que también se validará que la empresa/usuario no haya sido registrada con anterioridad

2.7F07 – Ingresar

2.7.1 Acceso

Persona y Empresa

2.7.2 Descripción

Dado un tipo de perfil seleccionado por el usuario poder ingresar como él mismo.

2.7.3 Interfaz de usuario

LOGIN

BIENVENIDO

Nombre de usuario

Contraseña

PERSONA ▼

INGRESAR

REGISTRO

2.7.4 Validaciones

Al presionar el botón de ingresar se validará que dependiendo del perfil que haya sido elegido, que los campos obligatorios tengan contenido. En el caso del perfil empresa se validará la existencia de la misma conforme a los datos ingresados y en el caso del perfil de persona se validará la existencia de la misma conforme a los datos ingresados.

2.8F08 – Ver Solicitudes Pendientes

2.8.1 Acceso

Empresa

2.8.2 Descripción

Llevar al usuario a un formulario donde estarán listadas todas las solicitudes de envío pendientes de parte de las personas.

2.8.3 Interfaz de usuario

The screenshot displays a web interface titled "SOLICITUDES PENDIENTES". At the top, there is a dark navigation bar with the text "EMPRESA S.A" on the left and three menu items: "SOLICITUDES PENDIENTES", "SOLICITUDES EN TRANSITO", and "VER ESTADISTICAS". The main content area contains a 5x3 grid of 15 identical request cards. Each card has a light purple header with a "FOTO" label and a placeholder image. Below the header, the name "SEBASTIAN LABADIE" is displayed. Underneath, the word "DISTANCIA" is followed by the value "32131231 KM". At the bottom of each card is a blue button labeled "ASIGNAR".

2.8.4 Validaciones

Al presionar el botón de ver solicitudes pendientes se validará que el usuario logueado tenga el perfil empresa, que las solicitudes listadas están en estado "Pendiente" y que no contengan una empresa ya asociada, además de las solicitudes listadas coinciden en tipo de vehículo con la empresa logeada actualmente.

2.9F09 – Ver Solicitudes En Tránsito

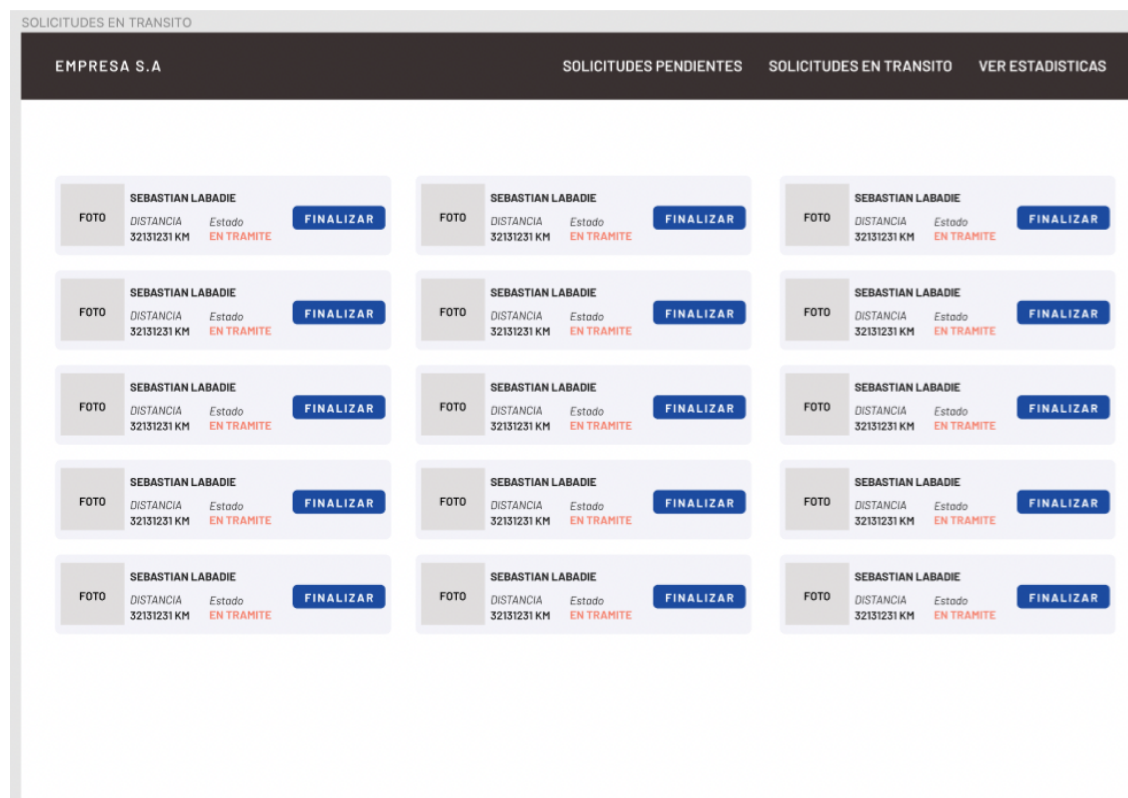
2.9.1 Acceso

Empresa

2.9.2 Descripción

Llevar al usuario a un formulario donde estarán listadas todas las solicitudes de envío que estén en tránsito de determinada empresa.

2.9.3 Interfaz de usuario



2.9.4 Validaciones

Al presionar el botón de ver solicitudes en tránsito se validará que el usuario logueado tenga el perfil empresa y que las solicitudes listadas están en estado “En Tránsito”, también se tendrá en cuenta que la empresa asociada a las solicitudes debe coincidir con la empresa logeada actualmente.

2.10 F10 – Ver Estadísticas

2.10.1 Acceso

Empresa

2.10.2 Descripción

Llevar al usuario a un formulario donde se listen las personas a las que se les han realizado más traslados junto con la cantidad de solicitudes realizadas por estado (“En Tránsito”, “Finalizada”)

2.10.3 Interfaz de usuario

The screenshot displays a web interface for viewing statistics. At the top, there is a navigation bar with the following items: 'VER ESTADÍSTICAS', 'EMPRESA S.A', 'SOLICITUDES PENDIENTES', 'SOLICITUDES EN TRANSITO', and 'VER ESTADÍSTICAS'. Below the navigation bar, there is a dropdown menu currently set to 'EN TRANSITO'. Underneath the dropdown, it states 'CANTIDAD DE SOLICITUDES EN TRANSITO: 33'. The main content area features a grid of 18 light blue rectangular buttons, each containing the text 'SEBASTIAN LABADIE', arranged in 6 rows and 3 columns.

2.10.4 Validaciones

Al presionar el botón de ver estadísticas se validará que el usuario logueado tenga el perfil empresa y la información que se presente en la lista de personas concuerde con la empresa logeada actualmente.

2.11 F11 – Realizar Solicitud Envío

2.11.1 Acceso

Persona

2.11.2 Descripción

Llevar al usuario a un formulario donde pueda dar de alta una solicitud de envío.

2.11.3 Interfaz de usuario

The screenshot shows a web interface for submitting a request. At the top, there is a header bar with the title 'REALIZAR SOLICITUD' on the left and three navigation links: 'REALIZAR SOLICITUD', 'VER MIS SOLICITUDES', and 'VER ESTADISTICAS'. Below the header, the user's name 'SEBASTIAN LABADIE' is displayed. The main content area contains a form with the following elements: three input fields labeled 'Distancia (KM)', 'Descripción', and 'Tipo de Vehículo' (which includes a dropdown arrow); a placeholder image box labeled 'FOTO'; and a prominent blue button labeled 'CREAR' at the bottom.

2.11.4 Validaciones

Al presionar el botón de realizar envío se validará que el usuario logueado tenga el perfil persona, que se hayan ingresado todos los campos obligatorios y que coincidan los tipos de datos con lo especificado.

2.12 F12 – Ver Mis Solicitudes

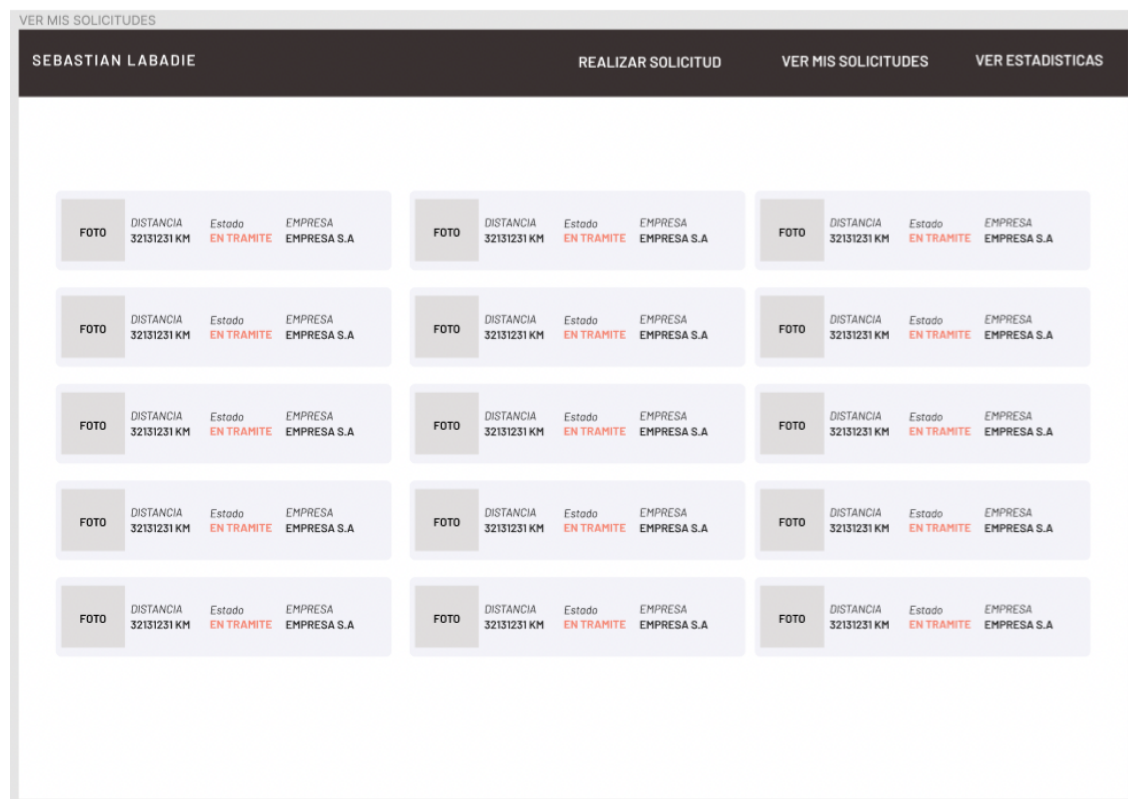
2.12.1 Acceso

Persona

2.12.2 Descripción

Llevar al usuario a un formulario donde se listen todas las solicitudes realizadas por el mismo.

2.12.3 Interfaz de usuario



2.12.4 Validaciones

Al presionar el botón de ver mis envíos se validará que el usuario logueado tenga el perfil persona, y que la información a listar de las solicitudes coincida con la persona logueada actualmente.

2.13 F13 – Ver Estadísticas

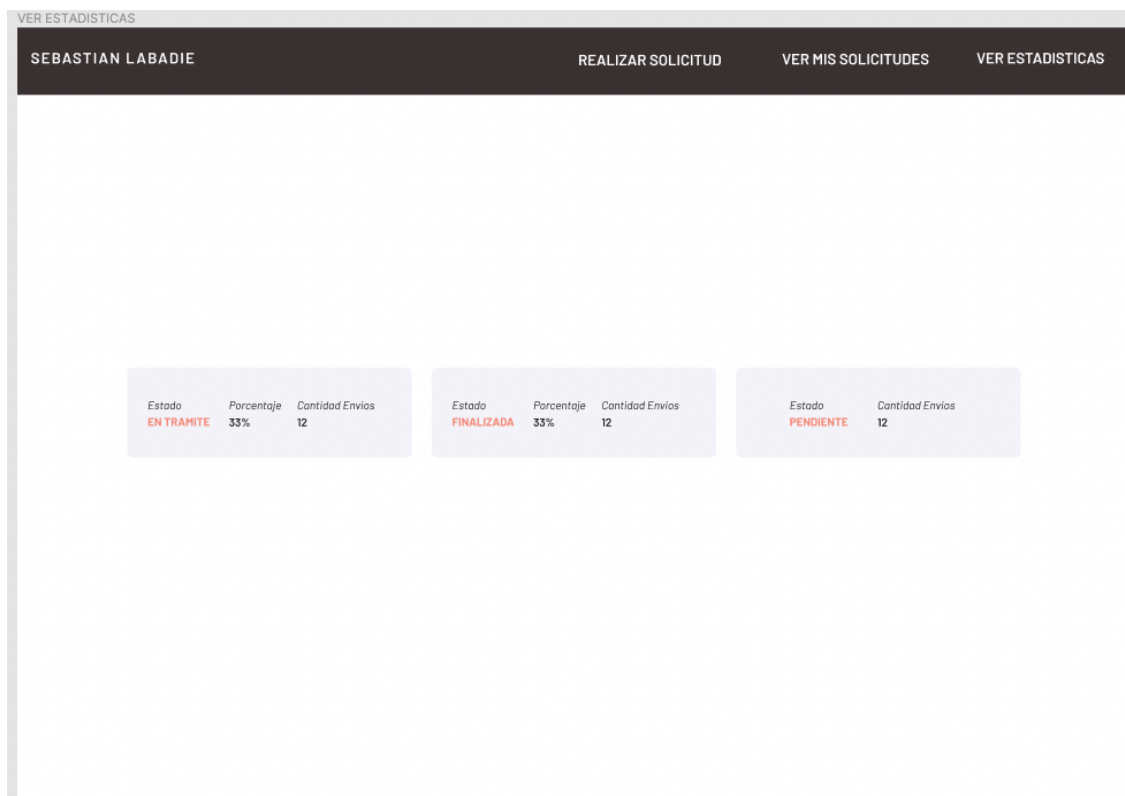
2.13.1 Acceso

Persona

2.13.2 Descripción

Llevar al usuario a un formulario donde se presentará el porcentaje de solicitudes en estado “En Tránsito” o “Finalizada”, además de cantidad de solicitudes discriminadas por estado “Pendiente”, “En Transito”, “Finalizada”

2.13.3 Interfaz de usuario



2.13.4 Validaciones

Al presionar el botón de ver estadísticas se validará que el usuario logueado tenga el perfil persona, y que la información a mostrar de las solicitudes coincida con la persona logueada actualmente.

CODIGO COMPLETO

HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Obligatorio - Sebastian Labadie 278119</title>

  <link rel="stylesheet" href="css/main.css">

</head>

<body>

  <div class="contenido">

    </div>

  <script src="./assets/icons.js" type="module" ></script>

  <script src="js/carga.js" ></script>

  <script src="js/funciones.js" type="module" ></script>

  <script src="js/main.js" type="module" ></script>

</body>

</html>
```

Main.js

```
"use strict";
```

```
import { crearFormularioLogin, handleLogin } from "../funciones.js";
```

```
window.isLogged = false
```

```
crearFormularioLogin()
```

```
document.querySelector('#btnLogin').addEventListener('click',handleLogin)
```

Carga.js

```
window.personas = []
```

```
window.empresas = []
```

```
window.tiposVehiculos=[]
```

```
window.envios=[]
```

```
const personas = window.personas
```

```
const empresas = window.empresas
```

```
const tiposVehiculos = window.tiposVehiculos
```

```
const envios = window.envios
```

```
class Persona{
```

```
  constructor(_cedula,_nombre,_apellido,_usuario,_password){
```

```
    this.Cedula = _cedula
```

```
    this.Nombre = _nombre
```

```
    this.Apellido = _apellido
```

```
    this.Usuario = _usuario
```

```
    this.Password = _password
```

```
  }
```

```
}
```

```
class TipoVehiculo{
```

```
  constructor(_id,_nombre){
```

```
    this.Id = _id
```

```
    this.Nombre = _nombre
```

```
  }
```

```
}
```

```
class Empresa{
    constructor(_rut,_razonSocial,_nombref,_usuario,_password,_tipoVehiculo,_estado){
        this.Rut = _rut
        this.RazonSocial = _razonSocial
        this.NombreFantasia = _nombref
        this.Usuario = _usuario
        this.Password = _password
        this.TipoVehiculo = _tipoVehiculo
        this.Estado = _estado
    }
}

class Envio{
    constructor(_persona,_tipoVehiculo,_distancia,_descripcion,_foto,_estado,_empresa){
        this.Persona = _persona
        this.TipoVehiculo = _tipoVehiculo
        this.Distancia = _distancia
        this.Foto = _foto
        this.Estado = _estado
        this.Empresa = _empresa
        this.Descripcion = _descripcion
    }
}

personas.push(new Persona('99999999','admin','admin','Admin','Admin01'))
personas.push(new Persona('51484225','Sebastian','Labadie','sebastianlb','1234'))
personas.push(new Persona('11111111','Isaias','Rodriguez','isaiasr','1234'))
personas.push(new Persona('22222222','Marcos','Moreno','macosm','1234'))
personas.push(new Persona('33333333','Mauro','Nevado','mauron','1234'))
personas.push(new Persona('44444444','Jose','Fernandez','josef','1234'))
```



```
personas.push(new Persona('5555555','Luis','Quiroz','luisq','1234'))
```

```
tiposVehiculos.push(new TipoVehiculo(1,'Moto'))
```

```
tiposVehiculos.push(new TipoVehiculo(2,'Camioneta'))
```

```
tiposVehiculos.push(new TipoVehiculo(3,'Camión'))
```

```
empresas.push(new Empresa('210128570016','razonsocial1','Deelivers','deelivers','1234',tiposVehiculos[0],'D'))
```

```
empresas.push(new Empresa('380351590216','razonsocial2','Happybox','happybox','1234',tiposVehiculos[1],'D'))
```

```
empresas.push(new Empresa('461347570117','razonsocial3','Instapack','instapack','1234',tiposVehiculos[2],'H'))
```

```
empresas.push(new Empresa('135215570218','razonsocial4','Kubbo','kubo','1234',tiposVehiculos[0],'H'))
```

```
empresas.push(new Empresa('756371570613','razonsocial5','Mox','mox','1234',tiposVehiculos[1],'H'))
```

[illegible]

1xWp0l2zko/mRM7ckRuyg0V1xZg9DP97VWXZS5fHY091x7RSatLKpPvqzUfyzKL6rDTmy+mmy34q7qDzG641xWtmqTo
UF+CGb97zNNd4tf3eauLy4qj/Z52l7loZb+p447YmWivp+Se6Yh65dYlZWif7Tt0KWW6lNZ+44VDiXB69X0cl+kpbLmzin4
vQ8ieeryWpGTaM8+qX36Vhoj06m3Wz3SnUhUgpU5RlF7OLzTLHh1nc3VpLmta1Si+2E3E7BYcY4vb5KrOlcxXSpDJ+9ZF+
P1Ok98bKL+nXjtnD6iDp9jx1bVMleWtWi/vQfOvkzf2WN4de5K3vKUPP7LlL3M2Y9TiydtmW+DJTuq2lAl1IAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADPJAAcG6xjD7XP9ovKEGunOm/cjT3XGmGUs1RvEu/wwwYXyKr5
8dO60La4cl+2suzA6Bdcd3Es1a2dKC7aknL8sjqmO4njWLOTjjV5Z5rLkt2oQ9y1+Jkv6jir29Wimgy256PY7q8trSHPd3FGH
D71WaividXxL6R+F7HNSxSnXmvs28ZVfill8TwbE+HsTc3UnL9uf3uduful+ppKtKdCbhWhOnNfZmmn8TDk9WyfbXb/1qp
6dX7rPef8A4wYHNuNK3vFLPJOTfQj/FN5Ez4+u7yDIYRtIU39pP0jXy+B4EWp1J0p89KcoT7YvJmS3qWe3M/4aK6LDXw9
mu8fxS6zVa/q5PRxpvkXwNXOTnLmm3N9snmzodpxFiFDJTqRrx7Ki1961N3Z8VWtTJXVKrRfbH1o/qUzmm/dLRWla9sb
Ow+wcipitL60u1nbV6dTuT1925yM0kN3SiTZPL3Dmlb7WAYs6gqmu8cy6lCc0hz6Fcnn0I5X1At6Roq6jfYOTtehKUV3kjm
2OMYhZP91vK1Nfd5s4+56HYLHjq/pZK7o0bhdWvUl+nwOqrbYa69C7HqMmPtpvgx37oel2PGuG18lCktbS688eaPvRvr
PELS9jnaXNGsvwTTfuPFFHPdsmPqy5otqXatGbcfqeSO+N2S/p9J7Z2e5g8isul8Ws8lRu6kor7NX118TsFhx3XWSvrKEI96jL
J+5/qbMfqOK3d0ZL6DLXjq76DQ2XFmE3OSdw6EuytHl+O3xN3RrU68OejUhUh2xaaNIMlL9s7stsdqd0bLgA7cAAAAAAA
AAAAAAAAAAAAAJmJNgWBXMAWGWZUAWzRGZ1jjTi+04WVpG5jz1bpy9GnLlj6uWeb8UdYq8dYhdQztf2alF7OMed/H9
DNm1ePDO1uWjFpsmWN68PTszj3N/a2qbubijS/jmkeT3OMYjdf299XkuxS5V7lkcLNZtvV9piv6pH21a6+nT91np1zxdhV
HPlrTrNdKUG/i8kam6460atbL2Sqz+S/U6Pze8jm01Mt/UM1uJ2aKaHFxNq7DdcXYtX5lGrToxfSnBaeLzNTc313dNu5uq1
XulNte44TkRzPwMt82S/daZaa4qU7YZMsumQzXcYvW7dCV3lSxftTjTRiTFXMsuugFtzHXpUq8HCvThVj2SjmviW9pHMgl
pbzhiwrZuip28vwPOPuZo7vhe9pZu3dO5j+F8svc/1O6N6kpNnM1iR5hXoVbefJcUp0p9k45GM9SqUo1IOFWKnDskk17
mam84aw+un6OMreXbTenuehzOP8DoaeTzTaa6o2VpjelWYShcSnBfZqesv1Obe8L3dHN21SncR6L6kvjp8TS3NtXtZctzR
qUn+OOXxONrVHZrTiuL0u7Zx/FSea9zNaYpZxElG4g5fdl6svczs7b7kxkkeokpl84tMSvLTLOfZUjH7rea9zN3Z8V1l6XdtCa
+9TfK/dsdxkjyO2rL2jQ1Inj2HXO1ZUpv7NVcvx2NmpRceaOTT6rXM7iYnhBvOzJ1fQjmk+hGU33EiyT2zGi7Cqpye7JUlpPN
5g2Spoc67SPVXQh9ybBstzaEqWXUhRb3RKjluQJTMtC4q28uehUnSl2wk0/gYkOhmTE7clmN+W/suLcVtsk60biC6VY5v3r
Jm9suO6MslfWk6b+9SlzL3PJnQ012JRGnHrM1OlF5UX0mK/MPWrHiHDLPuBykpv7NT1H8TaKWazWTXajxDrrLdHLs7+
8s9bS5q0V2Rk8vdsbcfqk/fX/DJf06Pst/l7LmMzziy4zxGikriNG5j2tcsvev0N7Zca2FXJXVKtby7cuePw1+Bsx67Dfzt/LJfR5a
eN/4dqzJzRwbLE7K9X7rdUar7ly1925zDXFotG8S2ZExO0rZgqCULAqALArmycwJBGYAgE5DICATkMgIBOQyA8a/pJ0s8Kw
OrkvVuKkPDP5HiljiV3ZPO2rzh3Z5r3Hvn9luhz8GWVbP+yvoae2E0fOy6nm6mPnO70NPPwdyw3jaccoX9HmX34fodqw
7GrG+inb14uX3XozyF7mSm3BqUW0+1Plw2wVnjo2VzWjl7WpJrPPP2E5l8uw7iTELLJek9LDsn+p2nDOMbOvlC6i6M+16
ozWwWqvrlrLtOXcPAX2tejcWUqFWM49zORyrsKlrHyjJal3Ejl7iBTNdEVzzzMnL3DkAw5LPJ69xOaXT4GVU/AnkQGJSW
WkSM5y2WRlyS00Dz7H4gYuWb7iVTfVmtJvdojdWBVQiuPewNBxmIKD3i1mmZEkuoyTA0t5w7htxm1SdCb+1SeXw2
NHeck3NPN2lanXX3ZepL9DuuyLzXT8iJrEp2eYXVnc2jyuqFSL3yjp79jB0PVXFNNPVPozV3fD+HXObdBUpv7VF8vw2+BXO
P8Dz4z2t1XtZ29apSf4ZZL3G/vOEq8M3Z3EKq+7UXK/ft+Ro7zD7uz/AL1bVacfvOOcfetDn2zA29nxVd0slcU6dePb9WX
w0N3acTWfFJVXO3l2VFmvejoa7iRF5geo0asK8OahUhUi+sJJovyPLU8uo1aGfPRnKnPtg2mbiz4mxChpVIC4j2VFr70dxkj
yO9Rj3E8pobLiyzqZK6p1KEu1evH4am8tLu1u4521enV65Rlr7tzuLRlUkOUy5ewJJbkjAo66lIxM2nRDL2IDCoMcveZuR9fi
Mkuwl3Y1EnLQvlpnlp2nHr3dvbr+trQj3Z5smBkyyzK8rb0zfsNTccQW0dKFOVR9r0RrbnHbuqnGm1Tj3HcUtLibxDs02oet
KUYZdZPUtT4zr4XpQxGvUy+w3zx+J0WrXqVXnVqSln2s48n6pbSkOneJ2V2tFukw+m+F8Sni+AWN/VgoTr01NxXta+RtDT
8HUvQ8KYPdstKX/Kmbg+kpv7Y3eBfb3TsAA6cgAAAACQAAAAAAaedfT5S9J9HlxLX+ruKMv5svmfM6Pqj6ZqKq/RtjWaz
clQqLuaqRZ8r9Tz9VHzbT2KP6zMKsnUvF7GZpX6FdExnoVe4HltLy4tJ81vWnTfcztGGcaXVHKN5BvOr7S0Z09Muupxalb
cw6raa8PVsN4lw++WUKqhU+7LRm3jV59YLNdx4jF5a9Ta2GPX9hy+hryaX2Z6oz2039Mrq5/6nrXrPpkSI95nTcN42p1co
31Nwl96OqO0WN9bXkVKhWhJdzM9qWrzC+t624crTUhMvyR3Yzy2SOHSFnloskHF9pHMyrbe7DqITypPVkZroVSz6luXX
Yg2M+wZt9MyVDLuLqOgTsx8rftIsqehdRy2J5c2Bj5F2kxiumpkUEs9SyS0AaxQLKGj85ls+wZ9oGqveH8OvM3O2jTm/t0
vUfw0fubNfbcHVI5uyuoZ/BWWT960+B3PXpqRk+qRExEo2h5be4Xe2WbubapGC+2lZr96OEtS1sev8ryepwLzA8OvM3Wt
YKb+3D1Je9bnM4/wh5iltxacW1JbNbnb73gxpOVjdeyFzF8AqX6Glnw5i0biFFWVScpvKLg04t+39Tn2W4g3LLiDEbXJRuH
Vgvs1VzL9Te2XGFKWSvLaUH96k+Ze5mtlw3CyXNjeK2NgutNT9LU/4UcepfDGHPq3tbzFaq+1Xl6Kn7lqXRivHdO38/65V
/Ujx1d5sMUsb98tpcQnP7m0vcc2bUE3Nngxu9nmMuNMRqThQw+jaYfbykouFvSSbWfWT1NhXuK1WT9LVnL2ssrSLcSj6
m3MO318Ws6GadXnfZE1dxxHLA2opfikaB7ZEMsjHEOJyS5Nzil3cJ+krSya2Whw22223m+1+0hLRjtLlil4cTMyp6vtHTz3k
N6IPWIBmOa/q3lVl8i72fntJhDnqQj96SxxJQ+oMGpOhhFjSaydOhTh7opHMKwjyQUVskkWPoYjaNnhSAAkAAAAAAEgAA

AAAAA6z9JIB3PAHEFNZZ/sdSWvcs/kfl+ep9lcVUVtCmYvSyZ57OtHbPeEj40h9RPuRh1cfKGzS8Slt2FSe4yNa3Qq9SdioEo
vHqYjLHYCF1zLSkrqWewEwM1CtUoT56M5Ql2xeRhiya7NhfV17a5Rr5V4Lt3O0YbxVh93IGcvRtFSWh5inoGU2w1strlt
V7bSlCrHmpSjKL2aZk9Fqen2GK3ljJO2rziuxvNHa8L46nFKF/RzW3PEzW09o46tFM9Z5d7VNDiDgjXWGOWV9DOhWi32
N6nPVTTtXcUzExyviYnhfISRX3kek7ERzSfaQLrbZEZ9/ulUZNdzhZQy1bAhrMla9C3L3tkqOmoN1EkWSXQtGLeyKVatGgm6
1aEfEIWyl9m5rbjHbOImqSIVfwNVdcRXM01RUaUdNtzuKWlxN4h2eS5VnNqC7ZPI4VfFbK3zUq3PLsidOq3devLOtVnLbd
nGTzfuLlxfxOT8Oz3HEqy/dqOXfl63xHi95Ww6vz1pR0WXX8uqKrZLsy/I4GPP/u6qn+H80deylhxNpl1nNttttt9WQwgzM
6ZrFZ3IBf7SP5nb3szqWHLO/t/40drbz0L8PDiyyeXvKt+qMyvTz2Fzlp4IVosyXv4lc/VJCT1CehWWrZK6gT2nNwOn6bHM
Opb89zSj/OjhlY3HBVJ1uL8Ggln+9034J5/I6pG9ohxadqzL6VBIPoHiBlyAgDIZAAmGBIAAAAAAMN5SveOr0ntUhKPvTR
8T8v13DrH1fcfbx8VYpSdvit9RksnTuKkGuzKbRj1ccNel8uL0JW46EZmJshbtKkshgQZlBGPqWg9ALZak5kllbgWiEx2Eewg
Wz0JbK9CeviQHQLb+JHTwJXzAtCcqucuaEnGXankbvDuJr+zeTqelp9ktzRMsczWLCpi014ekYTjxZV8oXSDGb7djs9tdW9xBS
oVYzXczxBGe0vbmzalb1pway2ehRbTxPavrqJ8vbkM9ieRpNyaiu2TyOlYPxNf1cJhGXo+fP6/L63vMFe+uK6brVpy20zKPpT
vtK+MkTG8O4XGJ2NtmqlwpNLBq7viaMOZWtDbP1pnWOMvY/zFR/W8TuMcQ5+pLY3WNXty2pVXFPPSPsOE5ym85y
cn3vPoY4fWfnoXWzz04il4cTMyvqlG9+5Is9ikno/PQlDHn7en5EJ7eehWT0yXnQZ9POwF47+ew4GP6WFT+KK+KOetFn
52NdxA/3GX8USLdsnl1tBhAyO3KwpZ4hQ/i+TO0M6zg/wDpCl4/kzsmepow8OLcjYTKZlluXOUvbz3IW9xJ6eBD389oDoS
tM/PaU6lI9UC3Q7N9GdP0nHOELTScp+6EjrK2O5fRBDn44t39yhVl8MvmW4Y3yV/IVlnak/w99WwAPdeMAAAAAA
AAAAAAAHx1xxS9Bxpj9LpG/rf8AO2fYrZ28k/SpRdv9luPwz3uef/iin8zLqo+MNOM7pdW6BEEowNsJZHQL7ERCUDWWiU
LICVvkX+0Ujuyc9QL9GQQ+obyTCITnqy0ShaJCU9PAsV6EpKAYzTbMsBal6BEgdkwB/8Ady/iOYl8jHYC8sPX8RzVos/YZ7c
y0V4Vlvp2fMrPVPxJk1m/PUiXXiStH6zLL7XnoVSyb8Qn9YCzfnwMcno/PQs+vnoY5bS89AKtavz0KZ/W89Cz6vzsUemfj+
QGaOsWaziGWVIFdtRfM2SfqvxNVxA87WH+8+Tit2yjj0KAQMIXzCFWeIQ7oyfwOWN5ZGgwP8Avv8AKfyN7nm/D9TTh7
XFuU9Myy0K9Citt9xa5Wl8iqf5/qS91kRlo8gjXzJ6lhbEvoEJ/Q9A+hKlZ8WXM8tKdnL4ziefRPTfoJp54ri1T7tCnH3yf6GjTR
vlhTnnbHL2QAHTPIAAAAAASAAAAAALH04U/R/SXijyeU4UZ//AMaXyPqc+af6QlH0fH8Z65VLKL3SmvkZ9THwaN
N3vM+hMWQxFnNny7KrTMnoQtmEnUnPcoTHUC6HVELYMIX7Sre5ZbMowQsjbYHhTxSF1y1PRzpxi4NrRt9H7jUo7Tw
5J2nDmJXazUkpcrW+ajkviyZBSL32lXmvNK7w6/d29W1rSo3EHCpHdP8/YYczslte2+LOJWuL8tOtCrUpULiPZDITcm+2Unp
s8ujNNimHXGG1/R3EfVefJNfVn7P0JzYJxvz4RizRk/lxC0dypK6lC5KRYoWQHYSf7jHs5jnJ+ovA4GB/3BL8RzuiXsKLcyvrwo
8+bw+Yl19rDeW/Z8yXs/E5dJ6N+0j72XeH9WXiOkvEA3kn4/kY3opex/kWl1XtKy2kBSWzy7/wAivRtd5afXxKvqvaSL9DU8
QP8AqKS/G/yZte3z1NTjqlONCEE5SIJ5JLNs5v2yiOWIlnhQwuvNSdTkltnq37jn0MKt4QSSqRdSe7bbXwKlxWl1NohwcD/
vcnrlyfNG6hqqZZRSJfJJaJLZEUTyVpC0Ur7Y2cTO63a3sUhpEtJlVsdC/Ujoxn594W4ExJk9SikPcQhbjPM9b+gamvQY1Vy1c
6UM/Yp5nkb2Z7P9BVNrAcSqNfXusvdCP6mrSR+rDNqZ/Tl6WAD2HlgAAAAAAJAAAAAAPnj+klRceK8LrZLKpZOOfa
41H/8A6R9Dng/9JmllD8P1llm4V4e5wfzKNRH6crsHfDxQiPQkiJ5z0GQqTmQQl6smJCC2CV8yP1HQdgQvOKL6yLmPH6w
IZUjtUEqHB1GMskq9SPN/C55v4ROqZ+rn3HZOMKjsuHrWID61KhOpl/DT5V8ZmrSR8plm1U/GldbtmqlrY5tNVIQnUi1uq
taVV/yGXBuM6c7uWfY7CvezquMY1WvWpTabeb3aTT13XeIl2lGlcVaCzUqClRi8tGoUlrXxmdRwuMr3ia35slzXE6rXJTJ
fNm6Yieksctt1h6DjWDVcPl6Wm/TWkmuWqumfSX67M1i2Ntg2NVbavWpV061lKNKLpvLWVWU5Z6/gjsZ8SvenK3/b8
Il6azkuZwWbcF2rq18V8Tz82mmvyrw24c/u6W5aVfKVRKMzS7DgP9y/zHOi9Pd+ZwMCf7l/mOctF7jPbmV9eES1894b38
SE8/PeMt/H8yHSZPfxJf1ZeJWWzy7yd0wKyzyl4lZaqRMt34lVqpECs9E/Ejd+e0mp18SNS/PUkT5+JHeM82vPUjPREuUT2f
noSir2eZbZ8QhjqpFz2kw6lxy6mSHn3hKJPciJL28CsXp57QhboSupWPeW6MJStyOoz3HalQnoe7fQpDI4Mcvv3VWX5L5H
hC6an0J9EtFUuA8OaWTqOpN+M5fobdFH6n9mTVz8P7u4AA9V5oAAAAAACQAAAAAADxj+kxSzwjA633bmpD3wz/
APSeznIH9JCjz8F2VVJvOd/DPuzhNFWaPhKzD3w+cmFuQOUh5j0YWRJVE56kJCMye4gCcyUV7S0dwLZIY7ksiOglcmzp+m
uaFJa89SMfzefcaONxillaN5RIOjCWXXWMqucv5YGisLj9kvKNfkU/RyUuVvLM7dcQhi9vK6w5U1PRzzhU+sqnl4w9jWcte
975m3STHWPLJqonpPh0S/rSdpWr1m2pw58++cp1P/64mh4Opp4jWqqLforZtPtILNr/AJkbvq0r4dhFVqCUZZRy35HyWS
jLslpN5djOLwVS5YxtWT5k68aXtUP/aBsZG2lUVKlc1vrw/aKs8vwUaUYR/mczkVMZueG6FOVsvSeigoum3pU5lqHL3Ny5
dd9TiW39ba2FOealUpU5TXaq1aVWX8kvga/im6SISLOWnNGpNfdacqjS/8AxL3gdzhTsel7WpfYfJKtGTVa2eSeabWi6Z5P
LozTNOMnGSaaETTWTRwfovoVYYnRSlyyhFRmovRrJzP9uql7zs3F04yxhXikpRpxUnlu9Xr4NGHU4q1j3Q2afLNvjLNgP9y/
zHOzz+BwMCf7m/4znLbTuPLty9KnCF189SW/yf5kPZrzuR5+JDpbPPPjWzK56vxJ6MDHPr4hbSE9n56kdH56hCs+vnqVz
16eWG83r51K9V56hC6a089pXPTz2BfoRLYIBLcs2VWstfOpDer89oFM9WZOHz2fnuLJ6ee8A9vaRHfz3kt6klkWWQxHbm
M9CABHRkakbomEL9UfSH0bU/R8DYLHtt1L3tv5nzbJ5QfsPqPhSl6HhJcKa+zaUl/IjfoY+Uyx6yfyjEnoAD0nngAAAAAAJA
AAAAAAPN/wCkBS9J9HNxL/V3FGf8+XzPSDpP000IV+jPHM1m4U4VF3ctSLOMkb0l3j6Wh8oIlgl8p6ULIIElghKXuT1K5k

kiHsy0epVbFokC3UhdQ9yOolZNzkYfe17C5Va3nlLaSe0l2NHGTBMTMTvBMRMbS7rSla8QW83lF80eWtbVFmtXHOXfp
HJdnczqN5hEMHwK6eHynKnGE5tS1nTnNZU1LtbzevvK2terbVY1qE3CpHZo7dY30MXpP0ThQxNLPXRS6Zp5Ppnp3m/D
ni/S3LDlwezrHDrFWEI4lOksnRpTnCDT2VOCopfzI6rxTN1rycJLTODy/FKFTvy9Kd8ucM9HWRXFKlOmGXPQybcU6ks5J
vfPli8u5vVHmuJT9NfOUZPNtZPr9SU8l381de40s70P6MLdv0laUcly5pLo5a5fzMX4tX/aMVu6meadRpPuWi/I3HBUY2XD
1xcKKgkm0vYm1+aOult6vd7mPVzxDXpY5lvsD/ucv4jnx28Ea7An+6yX4jYR0a8Dybcy9OvBJa+HzHb56kN+fEhvJvz1ldLdc
/O5L2ZXfz3k57hCs+vnqVb9V+epMlv56lG9/PUIV6+e0LfUjsfnchv5EgvPuleryl6eewnPUQhK8+8gQ2K5+fAkM9fEJ5lhayf
cFqmBbr57RFaENiOwFu0hiOzHRgQ3kRF6JlW9/PaStiYQlROLXasj6yw6kqFhb0UsITpxgl7EkfKvRD0l1Qp/eqwj75JH1qtdD
0NDHdLDrJ4gGQB6DCZEgECASMGlBOQAAAAAABDaW7Ak6v9KFH0/wBHvEMF/wDZVJbdiz+R2SVVIO/FMIX4bxej/rL
OtH305EWjeJTXpMPjXtEepSm/6texF1uzyXqQlAgjoQLJ3KlIsA6CJD3JWYAswt2OhEXuQLxZJWOxYCY7F6c5Qmpwk4yTz
TTyaZjS0LRCxbMNxKlikaNC9aheU5qpTnsPySaWffr+h1niLhrKvK7pr0NzS/tYperXfKpzqQeWjzills8tMnoY1odkwjF6NzG
Nri0YzyTjCrPv0yf6m3BqN/jdjzYNvIVlrX/YuDaVKOS5+Wky7M/0ijrC3OzcYyJst7K1p6QinJLuSSXzOrplWpne+y3TRtR2DAv
7rP+I2Ef0NBgH92qfxl2MHmvcefbtrwh+feQlInn56iTO89pGevntOXSy8+8Pr56lV9b2kvbz2hA/PvMb28fmW18+0r08US
Krp4FewnoI7COunnQlQ9l+ewp9trzuWe7l6v2/MkWj9Uq+1CL08CJvXIIQnv57SU/VZSP1dSz30CU7rwJT3KdCU9X57QLZ
k56ElZgUb9VlkVbJzJhDacL0/TcTYTS357yktvx+qEfMf0eUvT8cYJHXS5U9Pwpy+R9OLY9PRR8Zl5+sn5RAADaxgAAAAAA
AABDAAltixyqpFJy3OJVqZGcidxvqcedx3nBr1mupxJ3DewGxqXPeck8rKtb1qT2nCUH4pr5nGIOT3ZEdZ9qA+SOXkzi/s+r
7iY7mbEKfosRu6f3K1SPukzj56nkS9SFs9QiuehJDoexK2iYiBaRKky2JjsBZ7EJasl6EJ6gWWiLFEWWpAuvqsn9Sq2fcWXeEp
z2HVkDqBmq1qtaNNVJykqceWOfRZ7GPr57REjqBvcCf7tP+JGyh9U1eAPO3qfxl2cNjPbmV1eCXn3lXv57SW/PiVefn2kJE
9X56kt+fEhefeEwDeSIT08SONbsM9AhTsDeXnul6eewiWbTXnYkR2srHqca4xChSqyoxlKtXT/sqK55LXrlt45GW1rKvRVSM
ZRzzXLJapptNPxRO0x1RvDLHbz2EP6zJXtKye5AjoM9dfO4RDeSee4E7BbFWWj9UC0XuS9mUi9NCZPRKxAhklSSUO3fRJT
9L9IGFp/Z9JP3U5H0gfPv0JUVScRil/Z21WX/ACr5n0Eepo4/T/u87Vz8wAGtIAAAAAAABkABhqwbtYOFWps2ZjnTukBo
q9LfQ4E48rN/XobmtuaO4HAC0eZljk8iGtAPI3i+mqPFmNU1klG8q6L+Js1PU7H9JNL0PHmNxzzzuOf/iin8zrh5V42tMPSr
PSDMlMp1JOHaxMNYPsiGwEyJhsVm9i0NglZ7hdSHuT2gCxHtGehAuupZF17ssglZbAdCq3AuiJb5hbCXzA3XD7/AHep/EjZ
xaafgavAP7vU9qNhtF5lotzK2vDj2eepXPz4kvu2/wDchvz4nLozz895C+ZC2Xnqca5v7a2kqc581Z7Uqac5v/KtfeliZ4RM7O
SnovAxVasKVJ1Ks406a3lNpJeLONb3dapcxp1rWVCM4uUHKacnk1nmltv2sw2lCjcyld16UalZVJxi5+soKMnFcqei2Z0ortyj
ffhKv511+4W86yf+LP8Aq6e3a1m/BD9iq11nfXUpp/4NHOnDx+0/F+BdYhbyxKpYelX7XGkqzh15XpmaL6Q8aeEcPVFRny
3NznRptPjr70vBfmiylZtaKxHLi1orWbT4dko0aVCKp0KcKVNPMSlPc4uFT5rKMS95Tf88jgcH4hd4jgVrWxChVo3KyhJzjl6
TLaa9q+OZZMG/wBG0X2pv+Zs5tWa7xKYmJ2mHPzyzKPZsmT09rIWxCU9WUk9C2xXCJcdl4IUTjYTB6Fn1Kwlb1ALcnPcr
mF1Chp30B0ubibEKr+xaZf8U1+h7oeL/OfKWd5jdX7tOjD3uT+R7QevpY2xQ8vUz+plADQoAAAAAAAAAAAAAESipLU4dx
bZptHNAHWru3cW2kcM7TcW0akXluaC9tpUjvNaAfNf0v0/R8f4g/vwpT98F+h05HfvpwpKnxtGeWXpbSIJ9+TkvdAR5m
Tvl6GpThJKIjRWtStggnocAbLw2MZdbMCepZbFVuiiegSI9Qti00lgXjsXiY4Fo6BK62l7SSFuWJWxbtKJlWnvgOlGr7Uc+GmX
ga/An/VVfajYx6eBRblbXhaUowpuc5KMI6uUnkl7Wa9Ykqzyw+jUu39+Pq0lr997+GZEKfK8vbh3NONX9nlCFOM9YrOKk3
ltnm9/Ya3jfiSHD2FopDlleVfUoQe2fWTXyv0OqU90xWosubX9sTM8OZdN0qanjGUrWk/wDDp1FSi9es2+Z+GRzbCna0
7eLsI0VRnqpUsmPd+a3PnO/vbnELqdx1516095zef8A+l3HbvopxO6tsfjZw9JOzuE1UitYwkl6su7bLxNmTSTWkzvwY49V
Fr7bPW5/6Rtl/sp/nEwYXPkw5mm8p1XktW/XkZZa4rbN9KM/zgYsJ0w+L7ZVH/Plx/b/AN+7X5ee4HhuYhX9eXI5XlZXNv
y15U0lN8ktFT3y+rozsv0k29CXC19cSpQlWhGMITks3FopHPLszK4O8+PeluxUaEfgjld2lxf8MXNtZ0p1q9WdNKEFm366
NE3mctd/2Z4ptjtt+7dUPVoW6z2hD8jj4OuXDLZf7NP3mWnUpyhy0qkJ+j9WXLJPjbe0rhaaw62z39FH8kZ/Er3Jk9vaVX
US3HQHJLXz7SHoM9CstwESzeaKlZ8AnoErrcN6kEdvntCDRDMrnnmiejMD2v+j3Tyw/GquS9atThn7lt/M9aPM/oCo8vC
l7Wz/tLyXwhFHph7GnjbHDyc875JAAXKgAAAAAAAAAAAAAAX3FCFeDjUWefUyAD5w/pEYBd22L4fiMabnaOg6Mqk
deVqTaz7NJHkCPsfjShCtCgqklzg1KMoyWaaPEuLvo1p1ue6wHKIV3dvj+rL+F9DJmwzMzarViyxEe2Xk3QsjJeWteyul293
RnRrQeUoTWTMXQybbNUTulB7hdQtSEPRK7iCyIDqiewdUGCEodhCCCv4kxZEXolsgZUR3joEBCLICws2+Af2dX2o2K/Q1
2AaQreBsVp8Cm3KyvDiWP8AfMQ/3sP+nE8w4ww7FuKeLrinZW1R21s40IVJerBLmycs+vrZ7Z7Hp9i/3zEP97D/AKcTIQS
WSSyS6L2lmlPL9K3uiOuyu+P6kbS81wH6PLajfKni9V16kaUKzhTeUM/SSWTe7TUe7c7/h9haYdbxpWVvTo00ksorLPVvV9
d2Uj/pyt/5Wn/zzOWnmimUw1+6U48daR0hxs/8AvWgv9jJ/zQMvHKNLDYsNJRglKtCnkks28y2+MUI2W7/54nBcrWeA
QpX84xo16UoLeTmnostX4ERG8f9+6d+rrUsesLPiDe6+EKvi99e8kfrW8fUgorJet18DkTw3H8ZpyqY5iEMKsN5W9tLK
WX4pf+79hurC39BD0GEWFKxt/v1lZSffyLV/5mvYT+6qu8/S4ldwe2klB/CEPzLvFH2x/v/UKopP3T/wB/9cfCbe3w+y/ZMCt
pyoNuTrVZNQba1lm9ZeCy7zeUYKIqHTtbUEop+ZQ1txG8qW9arWrq3UYSkoUNXpHPWbXwSRsKE3KhTIJ+tKMW/aOV

WnfqtrG3RaT2Jb3IfQS+Zw6Qtisnq/PaWiY5PcCYvtJjORUtECxXPQkpLZEhHcl7IKZZVq+zVkj6J+g6nycB0Zf6y4qy/my+R38
6h9EIB2/0fYQnBxLOEqrzW/NOTT92R289nFG1Ij9nj5J3vIACxwAAAAAAAAAAAAAAAAAADR8Uw5qFJ9JOruJ2/iCPNBHVN
E5l1Dr3EfDmH49bunfUU5pepVjpOPsZ41xZwViOASIVjF3NI0rQWsf4l0PoJxMVSmpxcZxTi1k01mmV3x1vytpkmnD5aTLL
Y9f4u+ji3vfSXWC8ttcPV0n9Sb7uxnIN/Y3WHXM7a+oTo1o7xkvy7TffHanLVJTf+HGJTl6k7FaxOYlQz1AstvYQtifsIEQlkiyy
KRLlQqMnQLqRnoAHaWRXoTmEtvglZjW8DZrp4GrwF/23+X5mzz+RTblZXhw7H+94iv8Aaw/6cTkV6qoUKIWUZYUI8zjB
Zt5ZvRdpX7L++Yi/9rD/AKcSk8Rpyk4WUJ3dSLyfosuSL75vReGb7iZjeURO0MdjVhd39a7t5c9B04Uoz5WlJpybyz3WqWfa
ZK9/QoT9E5SqV+IKkuefuW3teSOPcKp6P0mJ3kbehovR0ZOKfc5v1n7FkUt5TVL0eGWcLei9fS1ouKem6h9aXteROOT1RE
7dHit1VnXlc16Soy5PRwhzczyzzzeWmeeWizOPD0NGtV/7NslOvm1Kq/UhF9VzPXwii9p6alfzpVbiddOiqj5opJPma0S2W
XTU4te9qYfwm7yIBValG1VRRk3rtudRHHdKuznWUpYhXc4bulT9SmtvV+Ly7jW33EuGWGF3daznTrxtOWDp0XlHmk3l
FNadHt2Hk+N8S4pjLavLmXoXtRp+rBeHXzM/DkLnFKH/YltCHLXrKrOfLm4JLWTFRJaELNcaXaN7yyzqd52pD1XCL+tinC7
vbilGIUrUqklCOeSjqlv3l3NFctKC6JJHCuKNO0wWrRoR5aVKg4QXYIHJHPWiy7DHbaesNdd46SMiT1b1Kt6HKV9mYZbm
R7FHUe066GRGJbGTNRTbaUV1bAlsxVJZt6Jatm6wbh3E8ZalaUPR27/8RWzjDwW8vDTvO/4DwJh1g41bxO+uVrzVorki/
wAMNvF5svx6e9+vEKL5616R1ed4Fw/iWMZSrdqg/8AxFbOPFw6y8F4noWBcB2Fm41b/wDfq61/rY5U4vuht78zuNOM
orRbaGWec2l2m3HgpRlvmftl3TCYuGG20W88qaXwOUYrRctrSXZFlymplAAAAAAAAAAAAAAAAAAAAHAxePNbs6tOO
TaO24is6LOsVY+syJdVcRkXOjYHEo4kJcdxNTjuBWGN2ro39CM/uzWko96Zu3Eo4kcpeDcWcC3+Cc9e2UruyWvPFetBfiX
zOn9D6lnBNNNZpnQuLvo9tMT57nDOW1u3q45epN966GbJp/NV9M3izxaJzMSwu9wm7lb4hQnRqrbNaS70+pwn1yM
sxt0lpid+sMnQx9S+fqlepylZMvHqY0ZlHky2EXoQtgtgLBmQmSmQNvgL/tvD5m0XTwNVgW9b2L5m0WxVPKyvDV4hSsl
V+a7c5yq5P9nUpSVRRTP0a30SWuhdO8rxUaUIWNBZJcyUqmXdFerHxz9hltHneXze6nCCfdyJ5ezNt+Jkr1YUaU6tWSjTpx
cpN9Es2yd/Cljy4tC1t6FdTzdS6az9JvlzVMu1Z7L2Zl5Oeifd8jyC9nivPDjCnJ8srLxhB/ZprSKf4XrH/9np+C4rbYvYqurRv0bW
TTWXLlJN8My3Lhmkb77q8WWLTtsy/wD1Wo8//DL/AJmRh1ONTBrWnUipQnbxjKL6pxSaD0v7iXZbxXxkZMPXLYWqf
SjD8kV+Fnl5xj/0f0rO3vb22vnG3o05VVSndN6ZvLPM7ZwNhlrh+AWla3pKNa5oxqVZvWUm1nln2LsOXxZ/8s4r/wCWn+
TOPw3iNo8Hw62pVlWrwtqalCinNw9VfWy0Xiy62S98fWfKmtKUv0bPFcv+zbrPZwa97SOXLTmXecPEngVu6HNFVKzUY
Rb1l6yzyXsOTN5tso8LvKSkMwjszHOSjFuTs2thK72895SUIFNyaSW7b2N9gnC2K4vyzpUf2a2f+NcJrNfhju/gu89BwDgn
DcMcatWDu7pa+lrpPlf4Y7R+L7y+mmtbrPSFN89a8dXnWcCMYpjHLK3oegtn/j104xf8Md5fBd56HgHA+HYc4Vbl09uY6
qpWS5Yv8MNI8X3nblQS2RkUTZTDSnDJfLa/LHGmorRF0i6iWUS1WpGJntoZ1YrvKxicqxhnXj7SUO10FIRiu4uVo/2a9hYs
VgAAAAAAAAAAAAAAAAAAAAADJxqzps67cRymzsl19RmhuY+uyJTDguJRxoQ4lGIHtjGNxOS4mNx47RRxOQ4lHEDUYx
hFli9rK3v6EasHs3vHvT6HkHF30f3uE89xh3Nd2a1a59eC711Pc3ExyjucXx1vy7peacPlvpl1RD6nuHFvAvlJHPcWfLaXr15or
1Z+1fM8gxvBr7Bbp0MQoSpyz9WW8ZexmHJImn8NePLFmujuZlIEZEVLPYdCHsBZkrqUWqZZbAbba361b2L5mzWy8D
VYH9ar7F8zaLZeBvblZXhXLP+93/wDvYf8ATidf4txqzoVrfB7ujWqK/IGnUIFuCjBvJv6vuXib6zdz3f/AO9j/wBOJ1C1vLbE
ONrm/uo9KFrYzWdqpy+vVejiXV7+9FmKu8zafEKr22jaPLs1bCreWDPC4J07b0So5JtR8evf2nV8ExBYHxOuHf2eVKwIH9
3lJ80pzevM32PbLpku8jF0K8SucTucO4bsFcTpSUHCNZxi0te7fq30OTw9wxeU8Thi2P3ru76Cfo4J5xp+PjslkWRX2Vn6k8+
PO7mbe60ft8OyVGv2u67rdP4zMlo/wBzof7uH5lWuPpe3qWmdGFPPsfrfjHSo3ISlTp1akbanCMY5UnzTllp9ZrJeCz7yh
bDh4pbwupKni1xL0Mvq2NvnnU/iy9aXs0XtOTb29aNBUBalTw61jtCnFOb/8ATH4s5ltbUbal9BBRb1lLeUva3q/EySIGMe
aTS6ak+7xBFFMsVta0bdylTh/WS+tUk+acvbJ6k1ZqMc5NRXazsmB8l4rimU5U/wBit3/iV4vma/DDf35HoOAcHYZhMo1Y
0ncXS/x6+UpL2LaPgi6mnvfrboptnrXpHV51gfCeK4tGMIS/ZLaWvpriLTa/DDd+OSPQsA4LwzCpQqum7m6j/jV8pNP8K2j
4a952iMEloi6Rxs4a04Zb5bX5UjBLYskXUSyiWOFVEskWSLxiBRRLqJZlSkBVI5mHx/rUcdROfh8fWzOoRlfUvqlsVp/VRY7V
gAAAAAAAAAAAAAAAAAAAAADfCfUZpbqPrM3db6rNVcx1ZCYa9xMbichoo4kOocdxKNGdoo4gcdxKtGdoo4gcdxKNHlaK
OIGBxOBimG2uJ2sre+oQrUpdJLb2dhs3Eo0EvFeLfo8usP57nCOa5tlq6f24L5nRcnFtSTTTyaejTPqFxoPcv8EWGNqVaklb
XvSpBaS/iRmyaeJ61X0zzHSzwt9Ssja4/gN/gdw6V/RcY5+rUjrGXsZqZMxzExO0tUTExvCU9GWiURaOxCW0wT61X2l2ifyN
TgryqVV3L82bVP5FVvUkcOHAR04xFZtZ1lrNf7uJ1LA/o/tLG9hc3l1UuqkKinTilyJNPNN9W9DsyuJULy8hGhWq1JzjKMYRy
TXJFZuT0S2T6hW9xcvO7rejg/wDBoNpP2z3fhkiyt7Vidp23VzWtuY32RKraWtxWjb0lO5qtTqU6EU5N5ZZY6L2vln0V3ca3
FX9np/6uhL1vGf6J05NKIToUITo04U6a+zFZLYtOSTiur0SWrb7EurON/w72/LHRo07en6OhCMIJt5L27vtZeUlFpyeWbSX
e8+nadlwTgrFMT5alzH9gt3rnVjnVku6HT/N7j0TAOFMNwdqpb0ee5yydeq+ao/Y+nhkX001rdbdFN89a9K9XnWB8HYrin
LORd9gt39utHOo13Q6f5svYeh4BwjhuENvAvJ1blf49b1p+HSPgjsMYJLRZfsjZTFWnDJfJa/KsYJblsolki6RY4VUSUi6RZRAo
kWUS6RKRIHlSkWUSyQ2FVEskWSJSJiEISnjYR1OFGJsrOOxKJbSH1SSlBEnTgAAAAAAAAAAAAAAAAAAAAAUqfVZrrmO5
sp7HBrrcghrpRMbRyJLVmNoOmFxmBnaKtEJcdxKNGdoo4BgaMbiZ3Eq0BgaMbichxMbQGBoq4mdxKNAC/sbe+t50L

ujCrSksnGSzPKeLvo4rW3pLrA861Hd0G/Wj7H1PY2jG4nF6ReNpd1vNZ6PI2cJU5ShUi4zi8nGSyali9D3zirc7D8ehKcoKhd
5aVoLXXU8c4i4ZxDAK7jeUm6L+pWgs4y/QxZMM06+GumWLdPLBg+ISp7EbNPbwNVhD/rJ+xfmbGMs5xhFSIUk0owin
KUvYlqzLMdWiOF+bReBj5kpQjq5SeUYpZuT7ElqzteBcD4nf8tS/f7BQeT5WIKs/DaPjm+49CwHhnDsGjnZ26VZrKVab5qkv
bJ/kskaMeltbrboovqKx0r1edYJwRiel8s7v9wt30muaq13R2j4+49CwHhfDcG9a1oJ18snXqPnqP/N09iyN9GKSySyLZGym
KtO2GW2S1+ZUJfJalsXS0JUTtwoWjEmUTJBaEoVSLqJKRdlhKqRZIsolkiRVrJSIJFkidhViskWLSJEoQkSkWSLKICC1NjarLI
4dOOpz7dCES5sdiSI7EnTgAAAAAAAAAAAAAAAAAAAAARLY4dZHMexx6q3BDgTiYmjkzWpialSwNFGjO0UaAwNFHEztF
GiHUSwtFHEzuJjaAwtfGjO0UaAwOJRoztFHEDC4IGjM0VcQMDice6taN1QnRuaUatKSycZLNM5jRRoDzy4+jS0liDq2V5W
treX16UYKT/yt7e5nbMC4cw7BoNWVvGFRrKVV+tUI7ZPXw2NzTjuZFERilazvEO/dNo2mVlxSWSWRZRLZEpHSFci0YgtBaj
ZCEi6iWSDWgGNI4bFcs2ZIR0OkbpSLpExRZlJZKEiUiYRZICqRZIskWSJQqkWSLKJZICEiyRKReMQJpR1OdQRxaSOZRQhEuR
HYKLYHTKAAAAAAAAAAAAAAAAAAAAAHsYKiM/QxzWgHDMtzC0cqaMLRCWBoo0Z2ijQGf00ZmijQGf00Z2ijQTEsD
RVozNFHEhLC4IGjO0UcQMLRRozNFWgMDRRxM7iUaArTW5kSEFqZEjmUwx5aMmKzMmREVkTARYloxyLJFooCqRLRdln
IDGol4xLJFoolCEi6iSkWSAhlIskWSAqol0iUiYQFUiYRZLJAQkWSJSLSXQFqaOVSRgpo5NNEwiWVALYEuQAAAAAAAAAAAA
AAAAAAAAAAKSRchgceSMTRyJlxNEJhgaKNGdoo0BgaKNGZoq0BhaKNGZoo0BhaKtGZoo0ExLC4IGjO0UaISwtFGjM0Va
AwtfHEzNFWgKQIzEHBGRi5IMKJDIpE5CBSKLpEpFIE6FUiUi6RKQqQolkiUiYQEJF0iUiYQFUiYRKRdICqRZIskSkBCRZILRp
AVSLpBlvFAWgjPBGKCMOSXK6ABKAAAAAAAAAAAAAAAAAAAAAADAxyMTABCrMbAISoyjAAo+pRgAVfUqwAQo+pV
gEOIHsyjAAqyj3AAmBdAESmEkoAQJiXWwBKEoAAWjuXQAFIsWW4AFkWQAEovHYACy2JQAFluXW4BKJZIGWIAhCwAJ
QAAAAAAAAAAD/9k=

```
envios.push(new Envio(personas[6],tiposVehiculos[0],20,'queda muy lejos',fotoEnvioDefault,'Pendiente',''))
```

```
envios.push(new Envio(personas[6],tiposVehiculos[1],10,'queda muy lejos',fotoEnvioDefault,'En  
Transito',empresas[4]))
```

```
envios.push(new Envio(personas[6],tiposVehiculos[0],15,'queda muy lejos',fotoEnvioDefault,'Pendiente',''))
```

```
envios.push(new Envio(personas[6],tiposVehiculos[0],25,'queda muy  
lejos',fotoEnvioDefault,'Finalizada',empresas[3]))
```

```
envios.push(new Envio(personas[5],tiposVehiculos[2],18,'queda muy lejos',fotoEnvioDefault,'En  
Transito',empresas[2]))
```

```
envios.push(new Envio(personas[4],tiposVehiculos[0],5,'queda muy lejos',fotoEnvioDefault,'En  
Transito',empresas[3]))
```

```
envios.push(new Envio(personas[4],tiposVehiculos[0],8,'queda muy lejos',fotoEnvioDefa4ult,'En  
Transito',empresas[0]))
```

```
envios.push(new Envio(personas[3],tiposVehiculos[0],14,'queda muy lejos',fotoEnvioDefault,'Pendiente',''))
```

```
envios.push(new Envio(personas[2],tiposVehiculos[1],22,'queda muy lejos',fotoEnvioDefault,'Pendiente',''))
```

```
envios.push(new Envio(personas[1],tiposVehiculos[2],40,'queda muy lejos',fotoEnvioDefault,'Pendiente',''))
```

Funciones.js

```
import { arrowLeft } from "../assets/icons.js"

let personas = window.personas
let empresas = window.empresas
let tiposVehiculos = window.tiposVehiculos
let envios = window.envios
let tipoPerfilLoged = window.tipoPerfilLoged
let isLogged = window.isLogged
let userLogged = window.userLogged

export const crearFormularioLogin = () =>{
  const divContenido = document.querySelector('.contenido')
  if (!isLogged){
    divContenido.innerHTML = `
    <div class='login'>
      <input type="text" id="txtUsuario" placeholder="usuario" >
      <input type="password" id="txtPassword" placeholder="password" >
      <select id="loginTipoUsuario">
        <option value="1">PERSONA</option>
        <option value="2">EMPRESA</option>
      </select>
      <p class='txtError' ></p>
      <div class='flex btnGroupLogin'>
        <input type="button" id="btnLogin" value="Ingresar" class="btnBlue001">
        <input type="button" id="btnRegistro" value="Registro" class="btnBlue001">
      </div>
    </div>
    `

    document.querySelector('#btnLogin').addEventListener('click',handleLogin)
    document.querySelector('#btnRegistro').addEventListener('click',crearFormularioRegistro)
  }else{
    divContenido.innerHTML = ``
  }
}

const crearMenu = () =>{
  const divContenido = document.querySelector('.contenido')
  const opcionesMenu = [
    {usuario:1,option:1,title:'Realizar Solicitud'},
    {usuario:1,option:2,title:'Ver Mis Solicitudes'},
    {usuario:1,option:3,title:'Ver Estadísticas'},
    {usuario:1,option:4,title:'Cerrar Sesión'},
```

```
{
  {usuario:2,option:1,title:'Solicitudes Pendientes'},
  {usuario:2,option:2,title:'Solicitudes En Transito'},
  {usuario:2,option:3,title:'Ver Estadísticas'},
  {usuario:2,option:4,title:'Cerrar Sesión'},
  {usuario:3,option:1,title:'Administrar Empresas'},
  {usuario:3,option:2,title:'Crear Tipos de Vehículos'},
  {usuario:3,option:3,title:'Ver Estadísticas'},
  {usuario:3,option:4,title:'Cerrar Sesión'},
}

divContenido.innerHTML = `
<div class='flex contenedor-menu'>
  ${opcionesMenu.filter(option=>option.usuario === tipoPerfilLogged).sort((a,b)=>a.option-
b.option).map(option=><input type="button" id="btnMenu${option.option}" value="${option.title}"
class="${option.option !== 4 ? 'btnBlue001': 'mt-3 btnRed001'}">`)}
</div>
`

document.querySelector('#btnMenu1').addEventListener('click',handleMenu1)
document.querySelector('#btnMenu2').addEventListener('click',handleMenu2)
document.querySelector('#btnMenu3').addEventListener('click',handleMenu3)
document.querySelector('#btnMenu4').addEventListener('click',handleMenu4)
}

export const handleLogin = () =>{
  const tipoUsuario = parseInt(document.querySelector('#loginTipoUsuario').value)
  const usuario = document.querySelector('#txtUsuario').value
  const password = document.querySelector('#txtPassword').value
  const txtError = document.querySelector('.txtError')

  if (!usuario || !password) {
    txtError.innerHTML=`Error usuario o contraseña vacío.`
    return
  }

  console.log(personas)
  if (tipoUsuario == 1) {
    const personasFiltered = personas.filter(persona => persona.Usuario === usuario && persona.Password ===
password)
    if (personasFiltered.length > 0){
      console.log(personasFiltered[0])
      isLogged = true
      userLogged = personasFiltered[0].Usuario

      crearFormularioLogin()
    }
  }
}
```



```
    if (personasFiltered[0].Usuario === 'Admin' ) {
        tipoPerfilLogged = 3
    }else{
        tipoPerfilLogged = 1
    }

    crearMenu()
}else{
    txtError.innerHTML=`Error usuario o contraseña incorrectos`
}
}else{
    const empresasFiltered = empresas.filter(empresa => empresa.Usuario === usuario && empresa.Password ===
password)
    if (empresasFiltered.length > 0){

        //Validar que empresa este habilitada
        if (empresasFiltered[0].Estado === 'D') {
            txtError.innerHTML=`Error empresa deshabilitada.`
            return
        }

        console.log(empresasFiltered[0])
        isLogged = true
        tipoPerfilLogged = 2
        userLogged = empresasFiltered[0].Usuario

        crearFormularioLogin()
        crearMenu()
    }else{
        txtError.innerHTML=`Error usuario o contraseña incorrectos`
    }
}
}

const handleMenu1 = () =>{
    switch (tipoPerfilLogged) {
        case 1:
            crearFormularioRealizarSolicitud()
            break;

        case 2:
            crearFormularioSolicitudesPendientes()
            break;
    }
}
```

```
    case 3:
        crearFormularioAdministrarEmpresas()
        break;

    }
}

const handleMenu2 = () =>{
    switch (tipoPerfilLogged) {
        case 1:
            crearFormularioVerMisSolicitudes()
            break;

        case 2:
            crearFormularioSolicitudesEnTransito()
            break;

        case 3:
            crearFormularioCrearTVehiculo()
            break;

    }
}

const handleMenu3 = () =>{
    switch (tipoPerfilLogged) {
        case 1:
            crearFormularioVerEstadisticasPersona()
            break;

        case 2:
            crearFormularioVerEstadisticasEmpresa()
            break;

        case 3:
            crearFormularioVerEstadisticasAdmin()
            break;

    }
}

const handleMenu4 = () =>{
    isLogged = false
    crearFormularioLogin()
}
```

```
const crearFormularioRegistro = () =>{
  const divContenido = document.querySelector('.contenido')
  let registroTipoUsuario = window.registroTipoUsuario

  if (!registroTipoUsuario) {
    registroTipoUsuario = '1'
  }

  const handleRegistrarse = () =>{
    const txtError = document.querySelector('#txtError')
    txtError.innerHTML=''

    if (registroTipoUsuario == '1'){
      const cedula = document.querySelector('#txtCedula').value
      const usuario = document.querySelector('#txtUsuario').value
      const nombre = document.querySelector('#txtNombre').value
      const apellido = document.querySelector('#txtApellido').value
      const password = document.querySelector('#txtPassword').value

      if (!cedula || !usuario || !nombre || !apellido || !password) {
        txtError.innerHTML=`Error hay datos vacios`
        return
      }

      if (personas.filter(persona => persona.Usuario === usuario).length > 0) {
        txtError.innerHTML=`Error usuario ya existe`
        return
      }

      personas.push(new Persona(cedula,nombre,apellido,usuario,password))
      alert('Usuario registrado correctamente')
    }else{
      const rut = document.querySelector('#txtRUT').value
      const razonSocial = document.querySelector('#txtRazonSocial').value
      const nombreFantasia = document.querySelector('#txtNombreFantasia').value
      const usuario = document.querySelector('#txtUsuario').value
      const password = document.querySelector('#txtPassword').value
      const tipoVehiculoSel = parseInt(document.querySelector('#solTipoVehiculo').value)
      const tipoVehiculoFiltered = tiposVehiculos.filter(tipoVehiculo=>tipoVehiculo.Id === tipoVehiculoSel)[0]

      if (!rut || !razonSocial || !nombreFantasia || !usuario || !password) {
        txtError.innerHTML=`Error hay datos vacios`
        return
      }
    }
  }
}
```

```
    }

    if (empresas.filter(empresa => empresa.Usuario === usuario).length > 0) {
        txtError.innerHTML=`Error usuario ya existe`
        return
    }

    empresas.push(new Empresa(rut,razonSocial,nombreFantasia,usuario,password,tipoVehiculoFiltered,'D'))
    alert('Usuario registrado correctamente')

}
}

const handleChangeTipoUsuario = (e) =>{
    registroTipoUsuario = e.target.value
    console.log(registroTipoUsuario)
    crearFormularioRegistroByType()
}

const crearFormularioRegistroByType = () =>{
    if (registroTipoUsuario == '1'){
        divContenido.innerHTML = `
        <div class='register'>
            <div class='flex'>
                <input type="text" id="txtCedula" placeholder="Cedula" >
                <input type="text" id="txtUsuario" placeholder="Nombre Usuario" >
            </div>
            <div class='flex'>
                <input type="text" id="txtNombre" placeholder="Nombre" >
                <input type="text" id="txtApellido" placeholder="Apellido" >
            </div>
            <div class='flex'>
                <input type="password" id="txtPassword" placeholder="Contraseña" >
            </div>
            <p id='txtError' class='txtError' ></p>
            <select id="registroTipoUsuario">
                <option value="1" ${ registroTipoUsuario === "1" ? 'selected' : null} >PERSONA</option>
                <option value="2" ${ registroTipoUsuario === "2" ? 'selected' : null} >EMPRESA</option>
            </select>
            <div>
                <input type="button" id="btnRegistrarse" value="Registrarse" class="btnBlue001">
                <input type="button" id="btnVolverLogin" value="Volver a Ingresar" class="btnBlue001">
            </div>
        </div>
        `
    },
```

```
}else{
  divContenido.innerHTML = `
    <div class='register'>
      <div class='flex'>
        <input type="text" id="txtRUT" placeholder="RUT" >
        <input type="text" id="txtRazonSocial" placeholder="Razon Social" >
      </div>
      <div class='flex'>
        <input type="text" id="txtNombreFantasia" placeholder="Nombre Fantasia" >
        <input type="text" id="txtUsuario" placeholder="Nombre Usuario" >
      </div>
      <div class='flex'>
        <input type="password" id="txtPassword" placeholder="Contraseña" >
        <select id="solTipoVehiculo">
          ${tiposVehiculos.map(tipoVehiculo=>
            `<option value="${tipoVehiculo.Id}">${tipoVehiculo.Nombre}</option>`).join("")
          }
        </select>
      </div>
      <p id='txtError' class='txtError' ></p>
      <select id="registroTipoUsuario" >
        <option value="1" ${ registroTipoUsuario === "1" ? 'selected' : null} >PERSONA</option>
        <option value="2" ${ registroTipoUsuario === "2" ? 'selected' : null} >EMPRESA</option>
      </select>
      <div>
        <input type="button" id="btnRegistrarse" value="Registrarse" class="btnBlue001">
        <input type="button" id="btnVolverLogin" value="Volver a Ingresar" class="btnBlue001">
      </div>
    </div>
  `
}

document.querySelector('#btnRegistrarse').addEventListener('click',handleRegistrarse)
document.querySelector('#btnVolverLogin').addEventListener('click',crearFormularioLogin)

//on change tipo usuario
document.querySelector('#registroTipoUsuario').addEventListener('change',(e)=>handleChangeTipoUsuario(e))
}

crearFormularioRegistroByType()

}
```

```
const crearFormularioRealizarSolicitud = () =>{
  const divContenido = document.querySelector('.contenido')

  divContenido.innerHTML = `
    <div>
      ${arrowLeft}
      <div>
        <div>
          <input type="number" id="txtDistancia" placeholder="Distancia (KM)" >
          <input type="text" id="txtDescripcion" placeholder="Descripción" >
          <select id="solTipoVehiculo">
            ${tiposVehiculos.map(tipoVehiculo=>
              `<option value="${tipoVehiculo.Id}">${tipoVehiculo.Nombre}</option>`).join("")
            }
          </select>
        </div>
        <div>
          <input type="file" id="file" placeholder="Ingrese Foto" class='mt-3' >
        </div>
      </div>
      <p class='txtError mt-3' ></p>
      <div>
        <input type="button" id="btnCrearSolicitud" value="Crear" class="btnBlue001 mt-3" >
      </div>
    </div>
  `

  const handleRealizarSolicitud = () =>{
    const distancia = parseInt(document.querySelector('#txtDistancia').value)
    const descripcion = document.querySelector('#txtDescripcion').value
    const txtError = document.querySelector('.txtError')
    const solTipoVehiculo = parseInt(document.querySelector('#solTipoVehiculo').value)
    const personaLogeada = obtenerPerOEmpLogeada('P')
    const vehiculoSeleccionado = tiposVehiculos.filter(tipoVehiculo=>tipoVehiculo.Id === solTipoVehiculo)[0]
    const foto = document.querySelector('#file').files[0]
    const reader = new FileReader();
    txtError.innerHTML = ``

    if (!foto) {
      txtError.innerHTML = `Error imagen vacia.`
      return
    }
  }
}
```

```
}

if (!distancia) {
  txtError.innerHTML = `Error distancia debe ser mayor a 0 y numerico.`
  return
}

reader.onload = () =>{
  const fotoB64 = reader.result.replace("data:", "").replace(/^./, "");
  console.log(fotoB64)

  envios.push(new Envio(personaLogeada,vehiculoSeleccionado,distancia,descripcion,fotoB64,'Pendiente',''))
  alert('Solicitud realizada con exito.')
  crearMenu()
}

reader.readAsDataURL(foto);

}

//click crear solicitud
document.querySelector('#btnCrearSolicitud').addEventListener('click',handleRealizarSolicitud)
//click volver
document.querySelector('#btnArrowLeft').addEventListener('click',crearMenu)
}

const crearFormularioVerMisSolicitudes = () =>{
  const divContenido = document.querySelector('.contenido')

  divContenido.innerHTML = `
    <div>
      ${arrowLeft}
      <div id='contenedorSolicitudes' class='mt-3 grid'>

    </div>
  </div>
  `

  const listarSolicitudes = () =>{
    const divContenedor = document.querySelector('#contenedorSolicitudes')
    const enviosFiltered = obtenerEnviosPersonaLogeada()
```

```
enviosFiltered.map(envio=>divContenedor.innerHTML+=`
<div class='flex cardContainer' >
  
  <div class='lable_value'>
    <p>Distancia</p>
    <p class='sbold'>${envio.Distancia} KM</p>
  </div>
  <div class='lable_value'>
    <p>Estado</p>
    <p class='txtEstado sbold'>${envio.Estado}</p>
  </div>
  <div class='lable_value'>
    <p>Empresa</p>
    <p class='sbold'>${envio.Empresa ? envio.Empresa.NombreFantasia : 'ninguna'}</p>
  </div>
</div>
`)
}

listarSolicitudes()

//click volver
document.querySelector('#btnArrowLeft').addEventListener('click',crearMenu)
}

const crearFormularioVerEstadisticasPersona = () =>{
  const divContenido = document.querySelector('.contenido')

  divContenido.innerHTML = `
    <div>
      ${arrowLeft}
      <div id='contenedorSolicitudes' class='mt-3 grid'>

      </div>
    </div>
  `

  const listarSolicitudes = () =>{
    const divContenedor = document.querySelector('#contenedorSolicitudes')
    const enviosFiltered = obtenerEnviosPersonaLogeada()

    let cantidadEnTransito = enviosFiltered.filter(envio=>envio.Estado === 'En Transito').length
    let cantidadFinalizada = enviosFiltered.filter(envio=>envio.Estado === 'Finalizada').length
    let cantidadPendiente = enviosFiltered.filter(envio=>envio.Estado === 'Pendiente').length
    let porcentajeEnTransito = (cantidadEnTransito * 100) / enviosFiltered.length
    let porcentajeFinalizada = (cantidadFinalizada * 100) / enviosFiltered.length
```



```
let porcentajePendiente = (cantidadPendiente * 100) / enviosFiltered.length

const estados = [
  {estado:'En Transito',cantidad:cantidadEnTransito,porcentaje: porcentajeEnTransito},
  {estado:'Finalizada',cantidad:cantidadFinalizada,porcentaje: porcentajeFinalizada},
  {estado:'Pendiente',cantidad:cantidadPendiente,porcentaje: porcentajePendiente},
]

estados.map(estado=>divContenedor.innerHTML+=`
<div class='flex cardContainer' >
  <div class='lable_value'>
    <p>Estado</p>
    <p class='txtEstado sbold'>${estado.estado}</p>
  </div>

  <div class='lable_value'>
    <p>Porcentaje </p>
    <p class='sbold'>${estado.porcentaje} %</p>
  </div>

  <div class='lable_value'>
    <p>Cantidad Envios</p>
    <p class='sbold'>${estado.cantidad}</p>
  </div>
</div>
`)
}

listarSolicitudes()

//click volver
document.querySelector('#btnArrowLeft').addEventListener('click',crearMenu)
}

const crearFormularioSolicitudesPendientes = () =>{
  const divContenido = document.querySelector('.contenido')

  divContenido.innerHTML = `
    <div>
      ${arrowLeft}
      <p id='txtError' class='txtError' ></p>
      <div id='contenedorSolicitudes' class='mt-3 grid'>

      </div>
    </div>
  `
}
```

```
,

const txtError = document.querySelector('#txtError')

const handleAsignarEnvio = (envio) =>{
  const empresa = obtenerPerOEmpLogeada('E')
  if (empresa.Estado === 'H') {
    envio.Empresa = empresa
    envio.Estado = 'En Transito'
  }else{
    txtError.innerHTML = `Error Empresa se Encuentra Deshabilitada.`
  }

  listarSolicitudes()
}

const listarSolicitudes = () =>{
  const divContenedor = document.querySelector('#contenedorSolicitudes')
  const enviosFiltered = obtenerEnviosPendientes()

  divContenedor.innerHTML = ``

  enviosFiltered.map((envio,i)=>{

    divContenedor.innerHTML+=`
    <div class='flex cardContainer' >
      
      <div class='lable_value'>
        <p class='bold mb-2'>${envio.Persona.Nombre} ${envio.Persona.Apellido}</p>
        <p>Distancia</p>
        <p class='sbold'>${envio.Distancia} KM</p>
      </div>
      <input type="button" id="btnAsignarEnvio${i}" value="Asignar" class="btnBlue001" >

    </div>
    `

  })

  enviosFiltered.map((envio,i)=>{
    //click asignar envio
    document.querySelector(`#btnAsignarEnvio${i}`).addEventListener('click',()=>handleAsignarEnvio(envio))
  })

}
```

```
listarSolicitudes()

//click volver
document.querySelector('#btnArrowLeft').addEventListener('click',crearMenu)

}

const crearFormularioSolicitudesEnTransito = () =>{
  const divContenido = document.querySelector('.contenido')

  divContenido.innerHTML = `
    <div>
      ${arrowLeft}
      <p id='txtError' class='txtError' ></p>
      <div id='contenedorSolicitudes' class='mt-3 grid'>

    </div>
  </div>
  `

  const txtError = document.querySelector('#txtError')

  const handleFinalizarEnvio = (envio) =>{
    const empresaLogged = obtenerPerOEmpLogeada('E')
    if (empresaLogged.Estado === 'H') {
      envio.Estado = 'Finalizada'
    }else{
      txtError.innerHTML = `Error Empresa se Encuentra Deshabilitada.`
    }
    listarSolicitudes()
  }

  const listarSolicitudes = () =>{
    const divContenedor = document.querySelector('#contenedorSolicitudes')
    const enviosFiltered = obtenerEnviosEnTransito()

    divContenedor.innerHTML = ``

    enviosFiltered.map((envio,i)=>{

      divContenedor.innerHTML+=`
      <div class='flex cardContainer' >
        
        <div class='lable_value'>
          <p class='bold mb-2'>${envio.Persona.Nombre} ${envio.Persona.Apellido}</p>

```

```
<div class='flex'>
  <div class='lable_value'>
    <p>Distancia</p>
    <p class='sbold'>${envio.Distancia} KM</p>
  </div>
  <div class='lable_value'>
    <p>Estado</p>
    <p class='txtEstado sbold'>${envio.Estado}</p>
  </div>
</div>
</div>

<input type="button" id="btnFinalizarEnvio${i}" value="Finalizar" class="btnBlue001" >

</div>
,

))

enviosFiltered.map((envio,i)=>{
  //click finalizar envio
  document.querySelector(`#btnFinalizarEnvio${i}`).addEventListener('click',()=>handleFinalizarEnvio(envio))
})

}

listarSolicitudes()

//click volver
document.querySelector('#btnArrowLeft').addEventListener('click',crearMenu)
}

const crearFormularioVerEstadisticasEmpresa = () =>{
  const divContenido = document.querySelector('.contenido')
  const estados = [ 'En Transito', 'Finalizada']
  let estadoSeleccionado = 'En Transito'

  divContenido.innerHTML = `
    <div>
      ${arrowLeft}
      <select id="estadisticasEstado">
        ${estados.map(estado=><option value="${estado}">${estado}</option>`)}
      </select>
      <p id='estadisticasCantidadSolicitudes' ></p>
      <div id='contenedorPersonas' class='mt-3 grid'>
```

```
        </div>
    </div>
    ,

    const changeEstado = (e) =>{
        estadoSeleccionado = e.target.value
        listarEstadisticaCantidadSolicitudes()
    }

    document.querySelector('#estadisticasEstado').addEventListener('change',(e)=>changeEstado(e))

    const listarEstadisticaCantidadSolicitudes = () =>{
        const cantidadSolicitudes = obtenerEnviosByEstadoEmpresa(estadoSeleccionado).length
        document.querySelector('#estadisticasCantidadSolicitudes').innerHTML=`Cantidad de solicitudes en estado
    ${estadoSeleccionado}:${cantidadSolicitudes}`
    }

    listarEstadisticaCantidadSolicitudes()

    const listarPersonas = () =>{
        const divContenedor = document.querySelector('#contenedorPersonas')

        const personasYCantidadFiltered = obtenerPersonasConMayoresEnviosByEmpresa()

        personasYCantidadFiltered.map(personaYCantidad=>divContenedor.innerHTML+=`
        <div class='flex cardContainer' >
            <p class='txtEstado sbold'>${personaYCantidad.persona.Nombre} ${personaYCantidad.persona.Apellido}</p>
            <p class='txtEstado sbold ml-2'>Cantidad: ${personaYCantidad.cantidadEnvios}</p>
        </div>
        `)
    }

    listarPersonas()

    //click volver
    document.querySelector('#btnArrowLeft').addEventListener('click',crearMenu)
}

const crearFormularioAdministrarEmpresas = () =>{
    const divContenido = document.querySelector('.contenido')
    let txtBuscarEmpresa = ""
    let empresasFiltered = []

    divContenido.innerHTML = `
```

```
<div>
  ${arrowLeft}
  <input type="text" id="txtBuscarEmpresa" placeholder="Buscar Empresa" >
  <p id='pError' class='txtError'></p>
  <div id='contenedorEmpresas' class='mt-3 grid'>
  </div>
</div>
```

```
const pError = document.querySelector('#pError')
```

```
const listarEmpresas = () =>{
  const divContenedor = document.querySelector('#contenedorEmpresas')
```

```
  empresasFiltered =
  empresas.filter(empresa=>empresa.RazonSocial.toLowerCase().includes(txtBuscarEmpresa.toLowerCase()))
```

```
  if (empresasFiltered.length === 0) {
    empresasFiltered =
    empresas.filter(empresa=>empresa.NombreFantasia.toLowerCase().includes(txtBuscarEmpresa.toLowerCase()))
  }
```

```
  if (empresasFiltered.length === 0) {
    pError.innerHTML = 'No hay resultados que coincidan con su búsqueda'
    divContenedor.innerHTML = ``
```

```
  }else{
    pError.innerHTML = ""
    divContenedor.innerHTML = ``
    empresasFiltered.map((empresa,i)=>divContenedor.innerHTML+=`
    <div class='flex cardContainer' >
      <div>
        <div class='mb-3'>
          <p class='sbold'>${empresa.NombreFantasia}</p>
        </div>
        <div class='flex' >
          <div class='lable_value'>
            <p>Rut</p>
            <p class='sbold'>${empresa.Rut}</p>
          </div>
          <div class='lable_value'>
            <p>Razón Social</p>
            <p class='sbold'>${empresa.RazonSocial}</p>
          </div>
        </div>
      </div>
```

```
        <div class='lable_value'>
          <p>Estado</p>
          <p class='txtEstado sbold'>${empresa.Estado}</p>
        </div>
      </div>
    </div>
  <div>
    ${empresa.Estado === 'D' ? `
      <input type="button" id="btnHabilitar${i}" value="Habilitar" class="btnBlue001" >`
    : `
      <input type="button" id="btnDeshabilitar${i}" value="Deshabilitar" class="btnBlue001" >
    `}
  </div>
</div>
`)
}
}

listarEmpresas()

const asignarClicks= () =>{
  //Click a los botones de habilitar y deshabilitar
  empresasFiltered.map((empresa,i)=>{
    if (empresa.Estado === 'D') {
      document.querySelector(`#btnHabilitar${i}`).addEventListener('click',()=>handleHabilitar(empresa))
    } else{
      document.querySelector(`#btnDeshabilitar${i}`).addEventListener('click',()=>handleDeshabilitar(empresa))
    }
  })
}

const changeEmpresa = (e) =>{
  txtBuscarEmpresa = e.target.value
  listarEmpresas()
  asignarClicks()
}

const handleHabilitar = (empresa) =>{
  empresa.Estado = 'H'
  listarEmpresas()
  asignarClicks()
}

const handleDeshabilitar = (empresa) =>{
```

```
    empresa.Estado = 'D'
    listarEmpresas()
    asignarClicks()
  }

  asignarClicks()

  //click volver
  document.querySelector('#btnArrowLeft').addEventListener('click',crearMenu)

  document.querySelector('#txtBuscarEmpresa').addEventListener('change',(e)=>changeEmpresa(e))
}

const crearFormularioCrearTVehiculo = () =>{
  const divContenido = document.querySelector('.contenido')

  divContenido.innerHTML = `
    <div>
      ${arrowLeft}
      <p id='txtError' class='txtError' ></p>
      <div class='flex'>
        <input type="text" id="txtNombreVehiculo" placeholder="Buscar Empresa" >
        <input type="button" id="btnCrearTVehiculo" value="Agregar" class='btnBlue001' >
      </div>
      <div id='contenedorTVehiculos' class='mt-3 grid'>

    </div>
  </div>
  `

  const listarTVehiculos = () =>{
    const divContenedor = document.querySelector('#contenedorTVehiculos')

    divContenedor.innerHTML = ``

    tiposVehiculos.map(tVehiculo=>{

      divContenedor.innerHTML+=`
        <div class='flex cardContainer' >
          <p>${tVehiculo.Id} - ${tVehiculo.Nombre}</p>
        </div>
      `
    })
  }
```



```
}

listarTVehiculos()

const handleCrearTVehiculo = () =>{
  const txtError = document.querySelector('#txtError')
  const nombreVehiculo = document.querySelector('#txtNombreVehiculo').value
  const nextId = tiposVehiculos.sort((a,b)=>a.Id-b.Id).reverse()[0].Id + 1
  txtError.innerHTML = ``

  if (!nombreVehiculo) {
    txtError.innerHTML = `Error nombre de vehiculo vacio`
  }else{
    tiposVehiculos.push(new TipoVehiculo(nextId,nombreVehiculo))
  }

  listarTVehiculos()
}

//click volver
document.querySelector('#btnArrowLeft').addEventListener('click',crearMenu)

document.querySelector('#btnCrearTVehiculo').addEventListener('click',handleCrearTVehiculo)
}

const crearFormularioVerEstadisticasAdmin = () =>{
  const divContenido = document.querySelector('.contenido')
  const empresasConKM = obtenerEmpresasConKMRecorridos()

  divContenido.innerHTML = `
    <div>
      ${arrowLeft}
      <div id='contenedorEmpresas' class='mt-3 grid'>
        </div>
      </div>
    `
  ,

  const listarEmpresas = () =>{
    const divContenedor = document.querySelector('#contenedorEmpresas')

    empresasConKM.map((empresaConKM,i)=>divContenedor.innerHTML+=`
    <div class='flex cardContainer' >
```

```
<div>
  <div class='mb-3'>
    <p class='sbold'>${empresaConKM.empresa.NombreFantasia}</p>
  </div>
  <div class='flex' >
    <div class='lable_value'>
      <p>Rut</p>
      <p class='sbold'>${empresaConKM.empresa.Rut}</p>
    </div>
    <div class='lable_value'>
      <p>Razón Social</p>
      <p class='sbold'>${empresaConKM.empresa.RazonSocial}</p>
    </div>
    <div class='lable_value'>
      <p>Estado</p>
      <p class='txtEstado sbold'>${empresaConKM.empresa.Estado}</p>
    </div>
    <div class='lable_value'>
      <p>KM</p>
      <p class='txtEstado sbold'>${empresaConKM.KM}</p>
    </div>
  </div>
</div>
`
}

listarEmpresas()

//click volver
document.querySelector('#btnArrowLeft').addEventListener('click',crearMenu)

}

const obtenerPerOEmpLogeada = (tipo) => {
  if (tipo === 'P') {
    return personas.filter(persona => persona.Usuario === userLogged)[0]
  }else{
    return empresas.filter(empresa => empresa.Usuario === userLogged)[0]
  }
}

const obtenerEnviosPersonaLogeada = () =>{
  const personaLogged = obtenerPerOEmpLogeada('P')
```

```
    return envios.filter(envio=>envio.Persona.Usuario === personaLogged.Usuario)
  }

const obtenerEnviosByEstadoEmpresa = (estado) =>{
  const empresaLogged = obtenerPerOEmpLogeada('E')
  return envios.filter(envio=>envio.Estado === estado && envio.Empresa.Usuario === empresaLogged.Usuario )
}

const obtenerEnviosPendientes = () =>{
  const empresaLogged = obtenerPerOEmpLogeada('E')
  return envios.filter(envio=>envio.Estado === 'Pendiente' && envio.TipoVehiculo.Id ===
empresaLogged.TipoVehiculo.Id)
}

const obtenerEnviosEnTransito = () =>{
  const empresaLogged = obtenerPerOEmpLogeada('E')
  return envios.filter(envio=>envio.Estado === 'En Transito' && envio.Empresa.Usuario === empresaLogged.Usuario)
}

const obtenerPersonasConMayoresEnviosByEmpresa = () =>{
  const empresaLogged = obtenerPerOEmpLogeada('E')
  const personasConEnvios = []
  personas.map(persona=>{
    if (persona.Usuario !== 'Admin') {
      const cantidadEnvios = envios.filter(envio=>envio.Persona.Usuario === persona.Usuario &&
envio.Empresa.Usuario === empresaLogged.Usuario).length
      personasConEnvios.push({persona,cantidadEnvios})
    }
  })

  return personasConEnvios.sort((a,b)=>a.cantidadEnvios - b.cantidadEnvios).reverse() //ordenar de mayor a menor
}

const obtenerEmpresasConKMRecorridos = () => {
  const empresasConKM = []

  empresas.map(empresa=>{
    const enviosEmpresa = envios.filter(envio=>envio.Empresa.Usuario === empresa.Usuario && envio.Estado ===
'Finalizada')
    const KM = enviosEmpresa.reduce((acc,el)=>acc+=el.Distancia,0)

    empresasConKM.push({empresa,KM})
  })

  return empresasConKM.sort((a,b)=>a.KM - b.KM).reverse() //ordenar de mayor a menor
}
```

INFORME DE TESTING

<u>Caso</u>	<u>Validación</u>	<u>Resultado</u>
Iniciar Sesión #1	Usuario Vacío	Error usuario o contraseña vacío.
Iniciar Sesión #2	Contraseña Vacía	Error usuario o contraseña vacío.
Iniciar Sesión #3	Usuario o Contraseña Incorrecto	Error usuario o contraseña incorrectos.
Iniciar Sesión #4	Usuario y Contraseña Correctos, pero mal tipo de usuario	Error usuario o contraseña incorrectos.
Registro #1	Cedula Vacía	Error hay datos vacíos.
Registro #2	Nombre Usuario Vacío	Error hay datos vacíos.
Registro #3	Nombre Vacío	Error hay datos vacíos.
Registro #4	Apellido Vacío	Error hay datos vacíos.
Registro #5	Contraseña Vacía	Error hay datos vacíos.
Registro #6	Rut Vacío	Error hay datos vacíos.
Registro #7	Razón Social Vacío	Error hay datos vacíos.
Registro #8	Nombre Fantasía Vacío	Error hay datos vacíos.
Ver Solicitudes Pendientes	Empresa Deshabilitada Asigna Envío	Error Empresa se Encuentra Deshabilitada.
Ver Solicitudes En Transito	Empresa Deshabilitada Finaliza Envío	Error Empresa se Encuentra Deshabilitada.
Crear TipoVehiculo	Nombre Vehículo Vacío	Error Nombre de Vehículo Vacío.
Realizar Solicitud Envio #1	Distancia Vacía	Error distancia debe ser mayor a 0 y numérico.
Realizar Solicitud Envio #2	Foto Vacía	Error Imagen Vacía.

INFORMACION PRECARGADA

Persona

<u>Cedula</u>	<u>Nombre</u>	<u>Apellido</u>	<u>Usuario</u>	<u>Password</u>
99999999	Admin	Admin	Admin	Admin01
51484225	Sebastian	Labadie	sebastianlb	1234
11111111	Isaias	Rodriguez	isaiastr	1234
22222222	Marcos	Moreno	marcosm	1234
33333333	Mauro	Nevado	mauron	1234
44444444	Jose	Fernandez	josef	1234
55555555	Luis	Quiroz	luisq	1234

TipoVehiculo

<u>Id</u>	<u>Nombre</u>
1	Moto
2	Camioneta
3	Camión

Empresa

<u>Rut</u>	<u>Razón Social</u>	<u>Nombre Fantasia</u>	<u>Usuario</u>	<u>Password</u>	<u>Tipo de Vehículo</u>	<u>Estado</u>
210128570016	RazonSocial1	Deelivers	deelivers	1234	0	D
380351590216	RazonSocial2	Happybox	happybox	1234	1	D
461347570117	RazonSocial3	Instapack	instapack	1234	2	H
135215570218	RazonSocial4	Kubbo	kubbo	1234	0	H
756371570613	RazonSocial5	Mox	mox	1234	1	H

Envio

<u>Persona</u>	<u>Tipo de Vehículo</u>	<u>Distancia</u>	<u>Descripción</u>	<u>Foto</u>	<u>Estado</u>	<u>Empresa</u>
6	0	20	Queda muy lejos	fotoEnvioDefault	Pendiente	"
6	1	10	Queda muy lejos	fotoEnvioDefault	En Transito	4
6	0	15	Queda muy lejos	fotoEnvioDefault	Pendiente	"
6	0	25	Queda muy lejos	fotoEnvioDefault	Finalizada	3
5	2	18	Queda muy lejos	fotoEnvioDefault	En Transito	2
4	0	5	Queda muy lejos	fotoEnvioDefault	En Transito	3
4	0	8	Queda muy lejos	fotoEnvioDefault	En Transito	0
3	0	14	Queda muy lejos	fotoEnvioDefault	Pendiente	"
2	1	22	Queda muy lejos	fotoEnvioDefault	Pendiente	"
1	2	40	Queda muy lejos	fotoEnvioDefault	Pendiente	"