



**Universidad ORT Uruguay**  
**Facultad de Ingeniería**  
**Escuela de Tecnología**

**OBLIGATORIO PROGRAMACIÓN 1**

		
<b>Felipe Caillabet – 280850</b>	<b>Luca Mirenda – 254413</b>	<b>Andrés Mayes – 230729</b>

**Clase M1C - Grupo 4**

**Docente: Bruno Díaz**

Analista Programador – Analista en Tecnologías de la Información

Fecha de entrega: 15-11-2021

# Índice

<b>Documento de análisis</b>	<b>4</b>
<b>Descripción general del problema a resolver</b>	<b>4</b>
Tipos de usuario del sistema	4
Listado de funcionalidades	4
<b>Detalle de Funcionalidades</b>	<b>5</b>
F01 – Ingresar a la aplicación	5
F02 – Habilitar y deshabilitar empresas	7
F03 – Crear tipos de vehículos	9
F04 – Visualizar información estadística	11
F05 – Registrarse en la aplicación	13
F06 – Realizar solicitud de envío	16
F07 – Ver listado de sus envíos	17
F08 – Ver listado de solicitudes pendientes específicas	18
F09 – Ver listado de solicitudes tomadas	19
<b>Documentación</b>	<b>20</b>
Precarga de datos	20
Vehículos	20
<b>ARRAY DE USUARIOS (length: 10)</b>	<b>20</b>
Usuarios de tipo PERSONA (4 PERSONAS)	20
Usuarios de tipo EMPRESA (5 EMPRESAS PRECARGADAS)	21
Usuario de tipo ADMINISTRADOR (1 ADMINISTRADOR PRECARGADO)	22
<b>ARRAY DE ENVÍOS (length: 5)</b>	<b>22</b>
<b>ESTRUCTURAS DE DATOS</b>	<b>23</b>



# **Documento de análisis**

## **Descripción general del problema a resolver**

Se debe realizar un programa gestor de envíos, contando con diferentes funcionalidades dependiendo el tipo de usuario. Dicha web permitirá al usuario crear solicitudes de envíos que desee realizar proporcionando los datos requeridos, siendo las cuentas de empresa las que acepten estos pedidos. Existiendo también la posibilidad de modificar pedidos desde una cuenta de administrador.

### **1.1. Tipos de usuario del sistema**

- Administrador
- Empresa
- Persona

### **1.2. Listado de funcionalidades**

F01 - Ingresar a la aplicación - Usuario/s: Administrador, Persona, Empresa

F02 - Habilitar y deshabilitar empresas - Usuario/s: Administrador

F03 - Crear tipos de vehículos - Usuario/s: Administrador

F04 - Visualizar información estadística - Usuario/s: Administrador, Persona, Empresa

F05 - Registrarse en la aplicación - Usuario/s: Persona, Empresa

F06 - Realizar solicitud de envío - Usuario/s: Persona

F07 - Ver listado de sus envíos - Usuario/s: Persona

F08 - Ver listado de solicitudes pendientes específicas - Usuario/s: Empresa

F09 - Ver listado de solicitudes tomadas - Usuario/s: Empresa

## **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 – Ingresar a la aplicación**

#### **2.1.1 Acceso**

Los tipos de usuario que tienen acceso a la funcionalidad son:

- Administrador
- Persona
- Empresa

#### **2.1.2 Descripción**

Esta funcionalidad tiene como objetivo permitir a aquellos usuarios que ya se hayan registrado, ingresar a la aplicación mediante su nombre de usuario único y su contraseña. Aquel usuario que no esté registrado o esté deshabilitado no tendrá acceso a la aplicación.

### 2.1.3 Interfaz de usuario

#### Ingresar a la aplicación

Usuario:

Contraseña:

### 2.1.4 Validaciones

Datos obligatorios:

- Usuario
- Contraseña

Mensajes de error:

- Usuario y contraseña - Al no ingresar datos: El botón [Log In] no estará habilitado.
- Usuario y contraseña - Al ingresar texto en ambos campos y estos no coincidan con algún usuario ya registrado: “El nombre de usuario y/o contraseña no son correctos”

## **2.2 F02 – Habilitar y deshabilitar empresas**

### **2.2.1 Acceso**

Los tipos de usuario que tienen acceso a la funcionalidad son:

- Administrador

### **2.2.2 Descripción**

Esta funcionalidad permitirá ver al administrador un listado de empresas con su respectiva información que están registradas, este podrá interactuar con esta lista, habilitando o deshabilitando el acceso y funcionalidades a las empresas.

A su vez, le permitirá al administrador realizar la búsqueda de una empresa en específico.

### 2.2.3 Interfaz de usuario

#### Habilitar/Deshabilitar empresa

Buscar por razón social:

Mostrar filtrado

Nombre empresa	RUT	Razon social	Estado	Habilitar/Deshabilitar
Nombre empresa 1	123	Razon social empresa 1	Habilitada	⊗
Nombre empresa 2	456	Razon social empresa 1	Deshabilitada	⊙
Nombre empresa 3	789	Razon social empresa 1	Habilitada	⊗

#### Muestra de busqueda

Buscar por razón social:

Mostrar todo

Nombre empresa	RUT	Razon social	Estado	Habilitar/Deshabilitar
Nombre empresa 3	789	Razon social empresa 1	Deshabilitada	⊗

### 2.2.4 Validaciones

Datos obligatorios:

- Razón Social

Mensajes de error:

- Razón Social - Al no ingresar un valor: “No se ingresaron datos”
- Razón Social - Al ingresar una razón social y/o nombre de fantasía no existente: “La razón social no existe en el sistema”



## 2.3 F03 – Crear tipos de vehículos

### 2.3.1 Acceso

Los tipos de usuarios que tienen acceso a la funcionalidad son:

- Admin

### 2.3.2 Descripción

Esta funcionalidad permite listar los vehículos existentes e ingresar nuevos tipos de vehículos proporcionando obligatoriamente el nombre, luego la misma aplicación le proporcionará un Id autogenerado para así poder identificarlo.

### 2.3.3 Interfaz de usuario

#### Crear tipos de vehiculos

Tipo	ID
Moto	837401
Camion	023764
Avion	262345

#### Agregar Vehiculos

Tipo de vehiculo:

### 2.3.4 Validaciones

Datos obligatorios

- Tipo de vehículo

Mensaje de error:

- Tipo de vehículo - Al no ingresar nada: “Se debe completar el campo”
- Tipo de vehículo - Al ingresar un vehículo existente: “El vehículo ingresado ya existe”

## 2.4 F04 – Visualizar información estadística

### 2.4.1 Acceso

Los tipos de usuarios que tienen acceso a la funcionalidad son:

- Admin
- Persona
- Empresa

### 2.4.2 Descripción

Esta funcionalidad permite ver los datos estadísticos. La información desplegada depende del tipo de usuario.

Para el administrador se mostrarán: total de km recorridos por una empresa, estado del pedido, estado de la solicitud y el nombre de la persona que ha realizado más entregas dentro de una empresa.

Las empresas podrán ver el cliente con más pedidos y la cantidad de pedidos pendientes, en tránsito y terminados.

Las personas podrán ver el estado de sus pedidos.

### 2.4.3 Interfaz de usuario

#### Informacion de empresas: Admin

Nombre empresa	Total de KM recorridos	Empleado con mayor cantidad de entregas	Pedidos aceptados
Empresa 1	8192 km	Lorem ipsum	3
Empresa 2	2341 km	Lorem ipsum	2
Empresa 3	8345 km	Lorem ipsum	3

### Informacion de pedidos: Admin

Estado de la solicitud	Estado del pedido	Nombre del usuario	Empresa asociada
Aceptada	En transito	Usuario 1	Empresa 1
No aceptada	N/A	Usuario 1	N/A
Aceptada	En deposito	Usuario 2	Empresa 2

## Informacion de pedidos: Empresa

### Ej: Empresa 1

Cliente con mas pedidos: Lorem, Ipsum

Traslados: 59

Pendiente ▼

Pendiente: 25 pedidos

### Informacion de pedidos: Usuario

#### Ej: Usuario 1

Pedidos pendientes: 3

Pedidos en transito: 1

Pedidos finalizados: 6

Pedidos aceptados: 70%

Estado de la solicitud	Estado del pedido	Empresa asociada
Aceptada	En transito	Empresa 1

## 2.4.4 Validaciones

No hay formularios para completar. No es necesario realizar validaciones.

## 2.5 F05 – Registrarse en la aplicación

### 2.5.1 Acceso

Los tipos de usuarios que tienen acceso a la funcionalidad son:

- Empresa
- Persona

### 2.5.2 Descripción

Esta funcionalidad permitirá registrarse a una persona que aún no está ingresada al sistema. Le permitirá al usuario diferenciarse de otros gracias a los datos proporcionados, que deberán cumplir con ciertas restricciones. Dependiendo cual es el tipo de usuario que se esté registrando se le exigirá que ingrese unos u otros datos. No habrá dos usuarios iguales.

### 2.5.3 Interfaz de usuario

#### Registrarse en la aplicacion

Seleccione el tipo de cuenta:

#### Registro Usuario

Nombre de Usuario:

Contraseña:

Cl:

Nombre:

Apellido:

## Registro Empresa

Nombre de usuario:

Nombre de fantasia:

RUT:

Razon Social:

Contraseña:

Tipo de vehiculo:

### 2.5.4 Validaciones

Datos obligatorios:

- Todos los campos

Para personas:

- Todos los campos - Si hay algún campo sin completar: No se habilita el botón [Registrar].
- Usuario - Se ingresa un nombre de usuario existente: “El nombre de usuario ya existe.”
- Contraseña - Se ingresa una contraseña sin una mayúscula, minúscula o un número: “La contraseña debe tener al menos una mayúscula, una minúscula y un número.”
- CI - Se ingresa un carácter invalido(letras, puntos, guiones,etc): “La CI se debe ingresar como un solo número. Sin puntos ni guiones. Ej: 1.234.567-8 se ingresa como 12345678”

- CI - El número ingresado no tiene 8 dígitos: “La CI debe tener exactamente 8 dígitos.”

Para empresas:

- Todos los campos - Si hay algún campo sin completar: No se habilita el botón [Registrar].
- Usuario - Se ingresa un nombre de usuario existente: “El nombre de usuario ya existe.”
- Contraseña - Se ingresa una contraseña sin una mayúscula, minúscula o un número: “La contraseña debe tener al menos una mayúscula, una minúscula y un número.”
- RUT - Se ingresa un RUT existente: “La empresa ya está registrada.”
- RUT - Se ingresa un carácter inválido: “Se ingresó un carácter no válido. Ingresar únicamente números”
- Razón Social - Se ingresa una razón social existente: “La empresa ya está registrada.”

## 2.6 F06 – Realizar solicitud de envío

### 2.6.1 Acceso

Los tipos de usuarios que tienen acceso a la funcionalidad son:

- Persona

### 2.6.2 Descripción

Esta funcionalidad permite realizar una solicitud de envío, para esto se deberá de seleccionar el tipo de vehículo necesario, la distancia en km, una descripción y una foto de la mercadería a trasladar.

### 2.6.3 Interfaz de usuario

#### Realizar solicitud de envío

Tipo de vehiculo:

Distancia:

Descripción:

Imagen:  Ningún archivo seleccionado

### 2.6.4 Validaciones

- Todos los campos - Si hay algún campo sin completar: No se habilita el botón [Ingresar solicitud].
- Distancia - Se ingresa un carácter inválido: “La distancia se debe ingresar solo con números”



## 2.7 F07 – Ver listado de sus envíos

### 2.7.1 Acceso

Los tipos de usuario que tienen acceso a la funcionalidad son:

- Persona

### 2.7.2 Descripción

Esta funcionalidad permite acceder al usuario a la lista de sus pedidos. Se mostrará foto, estado y empresa asociada al envío (en caso de existir) con el nombre de fantasía de la misma.

### 2.7.3 Interfaz de usuario

#### Ver listado de envíos

Foto	Estado	Empresa	Nombre de fantasía
	Pendiente	-	-
	Aceptado	Andy Inc.	Andy's
	En tránsito	Feli & associated	Feli's

### 2.7.4 Validaciones

No hay formularios para completar. No es necesario realizar validaciones.

## 2.8 F08 – Ver listado de solicitudes pendientes específicas

### 2.8.1 Acceso

Los tipos de usuario que tienen acceso a la funcionalidad son:

- Empresa

### 2.8.2 Descripción

Esta funcionalidad permite ver el listado de solicitudes de envío en estado pendiente que puede aceptar la empresa dependiendo del tipo de vehículo que ésta posee. Permite la asignación de envíos.

En el listado se mostrará la foto, distancia en kilómetros y nombre y apellido del usuario solicitante.

Una vez que la empresa asigna el envío, el mismo pasará automáticamente al estado “En tránsito”. Se actualizará el listado y el mismo desaparecerá.

### 2.8.3 Interfaz de usuario

#### Ver listado de solicitudes pendientes específicas

Estado	Vehículo	Distancia	Foto	Nombre del solicitante	Aceptar envío
No aceptado	Moto	5 km		Lorem Ipsum	<input type="checkbox"/>
No aceptado	Camion	45 km		Lorem Ipsum	<input type="checkbox"/>

### 2.8.4 Validaciones

No hay formularios para completar. No es necesario realizar validaciones.

## 2.9 F09 – Ver listado de solicitudes tomadas

### 2.9.1 Acceso

Los tipos de usuario que tienen acceso a la funcionalidad son:

- Empresa

### 2.9.2 Descripción

Esta funcionalidad permite ver el listado de solicitudes tomadas por la empresa.

En el listado se mostrará la foto, distancia en kilómetros, nombre y apellido del usuario solicitante y estado.

Las solicitudes que se encuentren en estado “En tránsito”, podrán ser marcadas como “Finalizadas”. El cambio de estado de la solicitud actualizará la lista para mostrar la información correcta.

### 2.9.3 Interfaz de usuario

#### Ver listado de solicitudes tomadas

Estado	Vehículo	Distancia	Foto	Nombre del solicitante	Aceptar envío
Aceptado	Moto	5 km		Lorem Ipsum	<input type="checkbox"/>
Aceptado	Camion	45 km		Lorem Ipsum	<input type="checkbox"/>

### 2.9.4 Validaciones

No hay formularios para completar. No es necesario realizar validaciones.

Precarga de datos

## Array Vehículos

Tipo	Id
“Moto”	1000
“Camioneta”	1001
“Camion”	1002

## Array de usuarios (length: 10)

### Usuarios de tipo Persona (4 usuarios)

Nombre	Apellido	Cedula	username	pass	pedidos	tipo
“Luca”	“Mirenda”	14725836	“lmirenda”	123	[]	“Persona”
“Felipe”	“Caillabet”	87654321	"fcaillabet"	789	[3,4]	“Persona”
“Andres”	“Mayes”	4928922	“amayes”	456	[1]	“Persona”
“Bruno”	“Diaz”	12345678	“bdiaz”	123	[0,2]	“Persona”

## Usuarios de tipo Empresa (5 usuarios)

<b>Fantasía</b>	“Empresa Uno”	“Empresa Dos”	“Empresa Tres”
<b>Habilitación</b>	True	True	True
<b>Password</b>	“Empresa01”	“Empresa02”	“Empresa03”
<b>Pedidos</b>	[2, 3, 4]	[1]	[]
<b>Razón social</b>	“Empresa Cero Uno”	“Empresa Cero Dos”	“Empresa Cero Tres”
<b>RUT</b>	101	102	103
<b>Tipo</b>	“Empresa”	“Empresa”	“Empresa”
<b>Username</b>	“Empresa1”	“Empresa2”	“Empresa3”
<b>Vehículo</b>	1000	1001	1001

<b>Fantasía</b>	“Empresa Cuatro”	“Empresa Cinco”
<b>Habilitación</b>	False	False
<b>Password</b>	“Empresa04”	“Empresa05”
<b>Pedidos</b>	[]	[]
<b>Razón social</b>	“Empresa Cero Cuatro”	“Empresa Cero Cinco”
<b>RUT</b>	104	105
<b>Tipo</b>	“Empresa”	“Empresa”
<b>Username</b>	“Empresa4”	“Empresa5”
<b>Vehículo</b>	1002	1002

### Usuario de tipo Administrador (1 usuario)

Username	Password	Tipo
Admin	Admin01	Administrador

### Array de envíos (length: 5)

Vehículo	1000	1001	1000	1000	1000
Distancia	5	10	5	5	5
Descripción	“Envío de supermercado”	“Motosierra”	“Mudanza de muebles”	“Mudanza de muebles”	“Mudanza de muebles”
Img	“envioSuper.jpg”	“motosierra.jpg”	“mudanza.jpg”	“mudanza.jpg”	“mudanza.jpg”
Empresa	null	“Empresa Cero Dos”	“Empresa Cero Uno”	“Empresa Cero Uno”	“Empresa Cero Uno”
Estado	“Pendiente”	“En tránsito”	“Finalizado”	“En tránsito”	“Finalizado”
Persona	“bdiaz”	“amayes”	“bdiaz”	“fcaillabet”	“fcaillabet”
Id	0	1	2	3	4

## Estructura de datos

Array asociativo	Atributos incluidos	Descripción del atributo
<b>UsuarioPersona</b>	<b>username</b>	String. Único entre todos los tipos de usuario existentes. Identifica al usuario.
	<b>pass</b>	String. Contraseña asociada al usuario. Contiene una mayúscula, una minúscula y un número.
	<b>cedula</b>	Numérica. La cédula de identidad del usuario.
	<b>nombre</b>	String. Nombre propio del usuario.
	<b>apellido</b>	String. Apellido del usuario.
	<b>pedidos</b>	Array indexado. Contiene los id de los envíos realizados por el usuario.
	<b>tipo</b>	String. Tipo de usuario. Valor: "Persona".
<b>UsuarioEmpresa</b>	<b>username</b>	String. Único entre todos los tipos de usuario existentes. Identifica al usuario.
	<b>pass</b>	String. Contraseña asociada al usuario. Contiene una mayúscula, una minúscula y un número.
	<b>fantasia</b>	String. Nombre de fantasía de la empresa.
	<b>rut</b>	Numérica. RUT asociado a la empresa. Único entre todos los usuarios.
	<b>razonSocial</b>	String. Razón Social asociada a la empresa. Único entre todos los usuarios.
	<b>vehiculo</b>	Numérica. Seleccionada desde un combo desplegable. Cada número está asociado a un tipo de vehículo distinto. (Ej: 1001 para Moto).
	<b>habilitacion</b>	Booleano. Determina si una empresa está habilitada (true) ó si está deshabilitada (false). Únicamente pueden acceder a la aplicación empresas habilitadas.
	<b>km</b>	Numérica. Cantidad de Km de pedidos finalizados.
	<b>pedidos</b>	Array indexado. Contiene los id de los envíos aceptados por la empresa.
	<b>tipo</b>	String. Tipo de usuario. Valor: "Empresa".

<b>UsuarioAdmin</b>	<b>username</b>	String. Único entre todos los tipos de usuario existentes. Valor: "Admin"
	<b>pass</b>	String. Contraseña asociada al usuario administrador. Contiene una mayúscula, una minúscula y un número. Valor: "Admin01".
	<b>tipo</b>	String. Tipo de usuario. Valor: "Administrador".
<b>Envio</b>	<b>vehiculo</b>	Numérica. ID del vehículo solicitado.
	<b>distancia</b>	Numérica. Distancia del envío en Km.
	<b>descripcion</b>	String. Breve descripción del envío.
	<b>img</b>	String. Ruta relativa a la imagen cargada. La imagen se carga desde un input tipo "file", desde una carpeta con imágenes precargadas.
	<b>empresa</b>	String. Razón Social de la empresa asociada al envío.
	<b>estado</b>	String. Corresponde al estado actual del envío. Valores posibles: "Pendiente", "En tránsito" y "Finalizado".
	<b>persona</b>	String. Nombre de usuario de la persona que realizó el envío.
	<b>id</b>	Numérica. ID del envío. Único entre todos los envíos. Autonumérico secuencial, se incrementa en 1 cada vez que se realiza una solicitud de envío.
<b>Vehiculo</b>	<b>tipo</b>	String. Único para cada vehículo.
	<b>id</b>	Numérica. Identificador del vehículo. Autonumérico secuencial, se incrementa en 1 automáticamente cada vez que se ingresa un nuevo vehículo al sistema.



## Variables globales

Nombre	Tipo	Descripción/Observaciones
<b>usuarios</b>	<b>Array indexado.</b>	Almacena todos los usuarios (usuarioPersona, usuarioEmpresa, usuarioAdmin) de la aplicación. Se inicia con los usuarios precargados. Cuando se registra un nuevo usuario se agrega en la última posición.
<b>vehiculos</b>	<b>Array indexado.</b>	Almacena todos los vehículos. Se inicia con los vehículos precargados. Cuando se genera un nuevo vehículo se agrega en la última posición.
<b>codigoVehiculo</b>	<b>Numérico. Entero.</b>	Corresponde al ID de cada vehículo. Se inicia con un valor de 1000 (correspondiente al código de moto). Ingresar un nuevo vehículo en el sistema aumenta el valor de esta variable en 1, y se asigna su valor como ID a su vehículo.
<b>envios</b>	<b>Array indexado.</b>	Almacena todos los envíos de la aplicación. Se inicia con los envíos precargados. Cuando se registra un nuevo envío se agrega en la última posición.
<b>idEnvio</b>	<b>Numérico. Entero.</b>	Corresponde al ID de cada envío. Se inicializa con un valor de 0. Ingresar una solicitud de pedido en el sistema aumenta el valor de esta variable en 1, y se asigna su valor como ID al envío.
<b>usuarioLoggeado</b>	<b>Array asociativo.</b>	Almacena el usuario logueado actualmente. Se inicia en "null". Cerrar sesión escribe el valor "null".

## Planilla de testing

Funcionalidad 01 - Ingresar a la aplicación				
	Prueba realizada	Resultado esperado	Resultado obtenido	Esperado y Obtenido
1	Ingresar con usuario no registrado	No se logra ingresar a la aplicación. Despliega mensaje de error: <i>"El nombre de usuario y/o contraseña no son correctos."</i>	No se logra ingresar a la aplicación. Despliega mensaje de error: "El nombre de usuario y/o contraseña no son correctos."	TRUE
2	Usuario existente, contraseña incorrecta	No se logra ingresar a la aplicación. Despliega mensaje de error: <i>"El nombre de usuario y/o contraseña no son correctos."</i>	No se logra ingresar a la aplicación. Despliega mensaje de error: "El nombre de usuario y/o contraseña no son correctos."	TRUE
3	Case sensitivity de la contraseña. Se intenta ingresar con usuario existente, y contraseña incorrecta.	No se logra ingresar a la aplicación. Despliega mensaje de error: <i>"El nombre de usuario y/o contraseña no son correctos."</i>	No se logra ingresar a la aplicación. Despliega mensaje de error: "El nombre de usuario y/o contraseña no son correctos."	TRUE
4	Ingresar con usuario precargado: Admin	Se ingresa a la aplicación. Se despliega el panel de administrador.	Se ingresa a la aplicación. Se despliega el panel de administrador.	TRUE
5	Ingresar con usuario precargado tipo empresa (habilitada): Empresa1	Se ingresa a la aplicación. Se despliega el panel de empresas.	Se ingresa a la aplicación. Se despliega el panel de empresas.	TRUE

6	Ingresar con usuario precargado tipo empresa (deshabilitada): Empresa4	No se logra ingresar a la aplicación. Despliega mensaje de error: " <i>Su cuenta aun no ha sido activada por el administrador.</i> "	No se logra ingresar a la aplicación. Despliega mensaje de error: " <i>Su cuenta aun no ha sido activada por el administrador.</i> "	TRUE
7	Ingresar con usuario precargado tipo persona: bdiaz	Se ingresa a la aplicación. Se despliega el panel de personas.	Se ingresa a la aplicación. Se despliega el panel de personas.	TRUE
8	Registrar un nuevo usuario tipo empresa desde la app. Intentar ingresar con dicho usuario.	No se logra ingresar a la aplicación. Despliega mensaje de error: " <i>Su cuenta aun no ha sido activada por el administrador.</i> "	No se logra ingresar a la aplicación. Despliega mensaje de error: " <i>Su cuenta aun no ha sido activada por el administrador.</i> "	TRUE
9	Registrar un nuevo usuario tipo persona desde la app. Intentar ingresar con dicho usuario.	Se ingresa a la aplicación. Se despliega el panel de personas.	Se ingresa a la aplicación. Se despliega el panel de personas.	TRUE
10	Case sensitivity del nombre de usuario. Intentar ingresar con un usuario existente (variando mayúsculas y minúsculas) y contraseña correctas. Usuario: BDIAZ	Se ingresa a la aplicación. Se despliega el panel de personas.	Se ingresa a la aplicación. Se despliega el panel de personas.	TRUE
11	Intentar ingresar a la app sin completar el campo de usuario	No se logra ingresar a la aplicación. Despliega mensaje de error: " <i>Se deben completar todos los campos.</i> "	No se logra ingresar a la aplicación. Despliega mensaje de error: " <i>Se deben completar todos los campos.</i> "	TRUE
12	Intentar ingresar a la app sin completar el campo de contraseña	No se logra ingresar a la aplicación. Despliega mensaje de error: "Se	No se logra ingresar a la aplicación.	TRUE

		<i>deben completar todos los campos."</i>	Despliega mensaje de error: <i>"Se deben completar todos los campos."</i>	
13	Click el botón [Iniciar sesión]	Se carga la pantalla de Ingresar a la aplicación	Se carga la pantalla de Ingresar a la aplicación	TRUE

Funcionalidad 03 - Crear tipos de vehículos				
	Prueba realizada	Resultado esperado	Resultado obtenido	Esperado vs Obtenido
1	Clickear el botón [Listado de vehículos] dentro del panel de Administrador	Se muestra el listado de vehículos y campo para agregar vehiculos	Se muestra el listado de vehículos y campo para agregar vehiculos	TRUE
2	Ingresar un vehículo no existente al sistema	Se ingresa el vehículo al sistema con ID autogenerado único. Se actualiza el listado de vehículos del panel de administrador, el combo desplegable de vehículos dentro del Registro y el combo desplegable de vehículos en Panel de usuario > Realizar solicitud de envío.	Se ingresa el vehículo al sistema con ID autogenerado único. Se actualiza el listado de vehículos del panel de administrador, el combo desplegable de vehículos dentro del Registro y el combo desplegable de vehículos en Panel de usuario > Realizar solicitud de envío.	TRUE
3	Ingresar un vehículo existente al sistema	No se ingresa el vehículo en el sistema. Se despliega el mensaje de error: <i>"El vehículo ya existe en el sistema."</i>	No se ingresa el vehículo en el sistema. Se despliega el mensaje de error: <i>"El vehículo ya existe en el sistema."</i>	TRUE
4	Case sensitivity - Ingresar un vehículo existente al sistema intercambiando mayúsculas y minúsculas	No se ingresa el vehículo en el sistema. Se despliega el mensaje de error: <i>"El vehículo ya existe en el sistema."</i>	No se ingresa el vehículo en el sistema. Se despliega el mensaje de error: <i>"El vehículo ya existe en el sistema."</i>	TRUE
5	Clickear el botón [Ingresar vehículo al sistema] con el campo vacío.	No se ingresa el vehículo en el sistema. Se despliega el mensaje de error: <i>"Se debe completar el campo."</i>	No se ingresa el vehículo en el sistema. Se despliega el mensaje de error: <i>"Se debe completar el campo."</i>	TRUE

<b>Funcionalidad 04 - Visualizar información estadística</b>				
	Prueba realizada	Resultado esperado	Resultado obtenido	Esperado vs Obtenido
1	Panel de Administrador: Click el botón [Visualizar información estadística]	Se muestra el panel de información estadística para administradores. (Listado de empresas con Km totales y pedidos finalizados).	Se muestra el panel de información estadística para administradores. (Listado de empresas con Km totales y pedidos finalizados).	TRUE
2	Panel de Administrador: Finalizar un envío con una empresa	Al volver al panel de administrador la información se muestra actualizada.	Al volver al panel de administrador la información se muestra actualizada.	TRUE
3	Panel de Empresa: Click el botón [Visualizar información estadística]	Se muestra el panel de información estadística para empresas. (Listado de personas con más pedidos finalizados y tabla con la cantidad de pedidos pendientes, en tránsito y finalizados).	Se muestra el panel de información estadística para empresas. (Listado de personas con más pedidos finalizados y tabla con la cantidad de pedidos pendientes, en tránsito y finalizados).	TRUE
4	Panel de Empresa: Aceptar un envío, luego volver al panel de información estadística	Se actualiza el listado de envíos pendientes, en tránsito y finalizados	Se actualiza el listado de envíos pendientes, en tránsito y finalizados	TRUE
5	Panel de Empresa: Finalizar un envío, luego volver al panel de información estadística	Se actualiza el listado de envíos pendientes, en tránsito y finalizados. Se actualiza el listado de personas con más envíos realizados en caso de ser apropiado.	Se actualiza el listado de envíos pendientes, en tránsito y finalizados. Se actualiza el listado de personas con más envíos realizados en caso de ser apropiado.	TRUE
6	Panel de Persona: Click [Listado de envíos]	Se muestra la información estadística del usuario junto con el listado de envíos	Se muestra la información estadística del usuario junto con el listado de envíos	TRUE

7	Panel de Persona: Realizar un envío, luego volver al panel de información estadística	Se actualiza la información estadística pertinente.	Se actualiza la información estadística pertinente.	TRUE
	Panel de Persona: Aceptar un envío con una empresa, luego volver al panel de información estadística	Se actualiza la información estadística pertinente.	Se actualiza la información estadística pertinente.	TRUE
8	Panel de Persona: Finalizar un envío con una empresa, luego volver al panel de información estadística	Se actualiza la información estadística pertinente.	Se actualiza la información estadística pertinente.	TRUE

Funcionalidad 05 - Registrarse en la aplicación				
	Prueba realizada	Resultado esperado	Resultado obtenido	Esperado vs Obtenido
1	Click en el botón [Registrarse]	Se carga la pantalla de Registrarse en la aplicación	Se carga la pantalla de Registrarse en la aplicación	TRUE
2	Seleccionar tipo de cuenta: Persona	Se carga la pantalla de Registro Persona con los campos correspondientes en blanco.	Se carga la pantalla de Registro Persona con los campos correspondientes en blanco.	TRUE
3	Seleccionar tipo de cuenta: Empresa	Se carga la pantalla de Registro Empresa con los campos correspondientes en blanco.	Se carga la pantalla de Registro Empresa con los campos correspondientes en blanco.	TRUE
4	Registro Persona: Click en el botón [Registrar] sin completar 1 o más campos	No se registra el usuario. Se despliega el error: <i>"Todos los campos son obligatorios."</i>	No se registra el usuario. Se despliega el error: <i>"Todos los campos son obligatorios."</i>	TRUE
5	Registro Persona: Se ingresa un nombre usuario de persona ya existente.	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese alias."</i>	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese alias."</i>	TRUE
6	Registro Persona: Se ingresa un nombre usuario de persona ya existente. Case sensitivity: Intercambiando mayúsculas y minúsculas	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese alias."</i>	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese alias."</i>	TRUE
7	Registro Persona: Se ingresan todos los datos correctamente, pero la contraseña no contiene un	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	TRUE



	número			
8	Registro Persona: Se ingresan todos los datos correctamente, pero la contraseña no contiene una mayúscula	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	TRUE
9	Registro Persona: Se ingresan todos los datos correctamente, pero la contraseña no contiene una minúscula	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	TRUE
10	Registro Persona: Se ingresan todos los datos correctamente, pero en la CI se ingresa un caracter no numérico	No se registra el usuario. Se despliega el error: <i>"La cédula debe contener solo números"</i>	No se registra el usuario. Se despliega el error: <i>"La cédula debe contener solo números"</i>	TRUE
11	Registro Persona: Se ingresan todos los datos correctamente y clickea [Registrar]	Se registra el usuario en el sistema. Se redirecciona a la pantalla de "Iniciar Sesión" para que el usuario pueda ingresar a la aplicación.	Se registra el usuario en el sistema. Se redirecciona a la pantalla de "Iniciar Sesión" para que el usuario pueda ingresar a la aplicación.	TRUE
12	Registro Empresa: Click en el botón [Registrar] sin completar 1 o más campos	No se registra el usuario. Se despliega el error: <i>"Todos los campos son obligatorios."</i>	No se registra el usuario. Se despliega el error: <i>"Todos los campos son obligatorios."</i>	TRUE
13	Registro Empresa: Se ingresa un nombre usuario ya existente.	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese alias."</i>	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese alias."</i>	TRUE
14	Registro Empresa: Se ingresa un nombre usuario de persona ya existente. Case sensitivity: Intercambiando mayúsculas y	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese alias."</i>	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese alias."</i>	TRUE

	minúsculas			
15	Registro Empresa: Se ingresa una Razón Social ya asociado a otro usuario.	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con esa razón social."</i>	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con esa razón social."</i>	TRUE
16	Registro Empresa: Se ingresa un RUT ya asociado a otro usuario.	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese RUT."</i>	No se registra el usuario. Se despliega el error: <i>"Ya existe un usuario con ese RUT."</i>	TRUE
17	Registro Empresa: Se ingresa un carácter no numérico para el RUT.	No se registra el usuario. Se despliega el error: <i>"El RUT debe ser un número."</i>	No se registra el usuario. Se despliega el error: <i>"El RUT debe ser un número."</i>	TRUE
18	Registro Empresa: No se selecciona un vehículo del combo desplegable.	No se registra el usuario. Se despliega el error: <i>"Todos los campos son obligatorios."</i>	No se registra el usuario. Se despliega el error: <i>"Todos los campos son obligatorios."</i>	TRUE
19	Registro Persona: Se ingresan todos los datos correctamente, pero la contraseña no contiene un número	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	TRUE
20	Registro Persona: Se ingresan todos los datos correctamente, pero la contraseña no contiene una mayúscula	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	TRUE
21	Registro Persona: Se ingresan todos los datos correctamente, pero la contraseña no contiene una minúscula	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	No se registra el usuario. Se despliega el error: <i>"La contraseña debe tener por lo menos una mayúscula, una minúscula y un número"</i>	TRUE

Funcionalidad 06 - Realizar solicitud de envío				
	Prueba realizada	Resultado esperado	Resultado obtenido	Espera do vs Obtenid o
1	Panel de Persona: Click el botón [Realizar solicitud de envío]	Se muestra panel con campos para rellenar del envío que se quiere ingresar (Tipo de vehículo (combo desplegable), Distancia (Número), Descripción, Imagen)	Se muestra panel con campos para rellenar del envío que se quiere ingresar (Tipo de vehículo (combo desplegable), Distancia (Número), Descripción, Imagen)	TRUE
2	Ingresa todos los datos de forma correcta y luego presionar el botón [Ingresa solicitud].	Se vacían los campos y se muestra mensaje	Se vacían los campos y se muestra mensaje	TRUE
	Ej: Vehículo: "Moto" Distancia: 3 Descripción: "Televisión 45 pulgadas" Imagen: "teleSamsung.jpg"	Vehículo: '' Distancia: '' Descripción: '' Imagen: ''  "Envío ingresado."	Vehículo: '' Distancia: '' Descripción: '' Imagen: ''  "Envío ingresado."	
3	Ingresa datos, pero dejar algún campo vacío.	"Completar todos los campos"	"Completar todos los campos"	TRUE
	Ej: Vehículo: "" Distancia: "" Descripción: "Heladera" Imagen: ""			
4	Ingresa al panel para realizar envíos, ingresa algunos datos pero sin presionar el botón de [Ingresa solicitud] apretar botón [Listado de envíos]. Y luego volver a presionar botón de [Realizar solicitud de envío]	Se vacían los campos antes ingresados	Se vacían los campos antes ingresados	TRUE

	<p>Ej: Vehículo: "Moto" Distancia: 4 Descripción: "Freezer" Imagen: "freezerFoto.jpg"</p> <p>Click [Listado de envíos]</p> <p>Click [Realizar solicitud de envío]</p>	<p>Vehículo: '' Distancia: '' Descripción: '' Imagen: ''</p>	<p>Vehículo: '' Distancia: '' Descripción: '' Imagen: ''</p>	
5	<p>Ingresar al panel para realizar envíos, ingresar algunos datos pero sin presionar el botón de [Ingresar solicitud], luego apretar botón [Cerrar sesión]. Después ingresar nuevamente al usuario y volver a presionar boton de [Realizar solicitud de envío]</p>	Se vacían los campos antes ingresados	Se vacían los campos antes ingresados	TRUE
	<p>Ej: Vehículo: "Camioneta" Distancia: 4 Descripción: "Freezer" Imagen: "freezerFoto.jpg"</p> <p>Click [Cerrar sesión]</p> <p>Usuario: "amayes" Contraseña: "456"</p> <p>Click [Log In]</p> <p>Click [Realizar solicitud de envío]</p>	<p>Vehículo: '' Distancia: '' Descripción: '' Imagen: ''</p>	<p>Vehículo: '' Distancia: '' Descripción: '' Imagen: ''</p>	
6	<p>Ingresar al panel para realizar envíos, ingresar todos los datos correctamente y luego presionar el botón de [Ingresar solicitud], luego apretar botón [Listado de envíos]. Y luego volver a presionar boton de [Realizar solicitud de envío]</p>	<p>Se muestra al ingresar los datos y presionar botón [Ingresar solicitud] un mensaje: "Envío ingresado." Luego al salir y volver a ingresar no se muestra ningún mensaje</p>	<p>Se muestra al ingresar los datos y presionar botón [Ingresar solicitud] un mensaje: "Envío ingresado." Luego al salir y volver a ingresar no se muestra ningún mensaje</p>	TRUE

	Vehículo: "Camioneta" Distancia: 4 Descripción: "Freezer" Imagen: "freezerFoto.jpg"  Click [Listado de envíos]  Click [Realizar solicitud de envío]	"Envío ingresado" al comienzo  Después de volver del panel de listado de envíos  "(Mensaje vacío)"	"Envío ingresado" al comienzo  Después de volver del panel de listado de envíos  "(Mensaje vacío)"	
	Ingresar al panel para realizar envíos, ingresar todos los datos correctamente y luego presionar el botón de [Ingresar solicitud], luego apretar botón [Cerrar sesión]. Después ingresar nuevamente al usuario y volver a presionar botón [Realizar solicitud de envío]	Se muestra al ingresar los datos y presionar botón [Ingresar solicitud] un mensaje: ""Envío ingresado."" Luego al salir y volver a ingresar no se muestra ningún mensaje	Se muestra al ingresar los datos y presionar botón [Ingresar solicitud] un mensaje: ""Envío ingresado."" Luego al salir y volver a ingresar no se muestra ningún mensaje	
7	Ej: Vehículo: "Camioneta" Distancia: 4 Descripción: "Freezer" Imagen: "freezerFoto.jpg"  Click [Cerrar sesión]  Usuario: "amayas" Contraseña: "456"  Click [Realizar solicitud de envío]	"Envío ingresado" al comienzo  Después de volver del panel de Log In  "(Mensaje vacío)"	"Envío ingresado" al comienzo  Después de volver del panel de Log In  "(Mensaje vacío)"	TRUE

**Código**

```

1 let usuarios = [];
2 let vehiculos = [];
3 let codigoVehiculo = 1000;
4 let envios = [];
5 let idEnvio = 0;
6 let usuarioLoggeado = null;
7
8 inicializar();
9
10 function inicializar() {
11     precargaDeDatos();
12     agregarEventoEnBotones();
13     ocultarPantallas();
14     toggleLogIn();
15     updateSelectVehiculos("#registroVehiculoEmpresa");
16 }
17
18 function agregarEventoEnBotones() {
19     document.querySelector("#btnRegistrarPersona").addEventListener("click",
registroPersona);
20     document.querySelector("#btnRegistrarEmpresa").addEventListener("click",
registroEmpresa);
21     document.querySelector("#btnTipoDeCuenta").addEventListener("click",
seleccionarTipoDeCuentaARegistrar);
22     document.querySelector("#btnLogIn").addEventListener("click", logIn);
23     document.querySelector("#btnIngresarVehiculo").addEventListener('click',
agregarVehiculoAlSistemaHandler);
24     document.querySelector("#btnEnviarSolicitudEnvio6").addEventListener('click',
btnSolicitudEnvioHandler);
25     document.querySelector("#btnPersonaListadoEnvios").addEventListener('click',
crearListadoEnviosPersona);
26     document.querySelector("#btnPersonaListadoEnvios").addEventListener('click',
calcularInfoEstadisticaPersona);
27     document.querySelector("#btnEmpresaEstadistica").addEventListener('click',
calcularInfoEstadisticaEmpresa);
28     document.querySelector("#btnAdminEstadistica").addEventListener("click",
calcularInfoEstadisticaAdmin);
29 }
30
31 // PRECARGA DE DATOS AL SISTEMA //
32
33
34 function precargaDeDatos() {
35     precargaDeVehiculos();
36     precargaDeUsuariosPersona();
37     precargaDeUsuariosEmpresa();
38     precargaDeUsuariosAdmin();
39     precargaDeEnvios();
40 }
41
42 function precargaDeVehiculos() {
43     vehiculos.push(new Vehiculo("Moto"));
44     vehiculos.push(new Vehiculo("Camioneta"));
45     vehiculos.push(new Vehiculo("Camion"));
46 }
47
48 function precargaDeUsuariosPersona() {
49     usuarios.push(new UsuarioPersona("bdiaz", 123, 12345678, "Bruno", "Diaz", []));
50     usuarios.push(new UsuarioPersona("amayes", 456, 4928922, "Andres", "Mayes",
[]));

```

```

51     usuarios.push(new UsuarioPersona("fcaillabet", 789, 87654321, "Felipe",
    "Caillabet", []));
52     usuarios.push(new UsuarioPersona("lmirenda", 123, 14725836, "Luca", "Mirenda",
    []));
53 }
54
55 function precargaDeUsuariosEmpresa() {
56     usuarios.push(new UsuarioEmpresa("Empresa1", "Empresa01", "Empresa Uno", 101,
    "Empresa Cero Uno", 1000, true));
57     usuarios.push(new UsuarioEmpresa("Empresa2", "Empresa02", "Empresa Dos", 102,
    "Empresa Cero Dos", 1001, true));
58     usuarios.push(new UsuarioEmpresa("Empresa3", "Empresa03", "Empresa Tres", 103,
    "Empresa Cero Tres", 1001, true));
59     usuarios.push(new UsuarioEmpresa("Empresa4", "Empresa04", "Empresa Cuatro", 104,
    "Empresa Cero Cuatro", 1002, false));
60     usuarios.push(new UsuarioEmpresa("Empresa5", "Empresa05", "Empresa Cinco", 105,
    "Empresa Cero Cinco", 1002, false));
61 }
62
63 function precargaDeUsuariosAdmin() {
64     usuarios.push(new UsuarioAdmin("Admin", "Admin01"));
65 }
66
67 function precargaDeEnvios() {
68     let pedido = new Envio(1000, 5, "Envio de supermercado", "envioSuper.jpg", null,
    "Pendiente", usuarios[0].username);
69     envios.push(pedido);
70     usuarios[0].pedidos.push(pedido.id);
71
72     pedido = new Envio(1001, 10, "Motosierra", "motosierra.jpg", "Empresa Cero Dos",
    "En tránsito", usuarios[1].username);
73     envios.push(pedido);
74     usuarios[1].pedidos.push(pedido.id);
75     usuarios[5].pedidos.push(pedido.id);
76
77     pedido = new Envio(1000, 5, "Mudanza de muebles", "mudanza.jpg", "Empresa Cero
    Uno", "Finalizado", usuarios[0].username);
78     envios.push(pedido);
79     usuarios[0].pedidos.push(pedido.id);
80     usuarios[4].pedidos.push(pedido.id);
81
82     pedido = new Envio(1000, 5, "Mudanza de muebles", "mudanza.jpg", "Empresa Cero
    Uno", "En tránsito", usuarios[2].username);
83     envios.push(pedido);
84     usuarios[2].pedidos.push(pedido.id);
85     usuarios[4].pedidos.push(pedido.id);
86
87     pedido = new Envio(1000, 5, "Mudanza de muebles", "mudanza.jpg", "Empresa Cero
    Uno", "Finalizado", usuarios[2].username);
88     envios.push(pedido);
89     usuarios[2].pedidos.push(pedido.id);
90     usuarios[4].pedidos.push(pedido.id);
91
92 }
93
94 function registroPersona() { // Registra un nuevo usuario en el sistema
95     // Ingreso de datos
96     let aliasIngresado =
    document.querySelector("#registroAliasUsuarioPersonaF5").value.trim();

```



```

97     let passIngresado =
document.querySelector("#registroPassUsuarioPersonaF5").value.trim();
98     let ciIngresado =
parseInt(document.querySelector("#registroCIUsuarioPersonaF5").value);
99     let nombreIngresado =
document.querySelector("#registroNombreUsuarioPersonaF5").value.trim();
100    let apellidoIngresado =
document.querySelector("#registroApellidoUsuarioPersonaF5").value.trim();
101
102    // Validacion de datos
103    let noEsNumero = isNaN(ciIngresado); // Booleano que devuelve true si la
cedula ingresada tiene algo distinto a un numero
104    if (aliasIngresado && passIngresado && ciIngresado != '' && nombreIngresado &&
apellidoIngresado) { // Revisa que se hayan ingresado datos en todos los
campos
105        if (!validarPassword(passIngresado)) {
106            document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "La
contraseña debe tener por lo menos una mayúscula, una minúscula y un número."
107        } else if (existeUsuarioPorUsuario(aliasIngresado)) {
108            document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "Ya
existe un usuario con ese alias.";
109        } else if (noEsNumero) {
110            document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "La
cédula debe contener solo números";
111        } else {
112            // Si paso todas las restricciones registra al usuario
113            registrarUsuarioPersona(aliasIngresado, passIngresado, ciIngresado,
nombreIngresado, apellidoIngresado);
114            displayBorradoCamposRegistroPersona();
115            toggleLogIn();
116            document.querySelector("#mensajeLogInNuevoUsuario").innerHTML =
`Bienviendo nuevo usuario ${aliasIngresado}. Puedes logearte en la app.`
117        }
118    } else {
119        document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "Todos
los campos son obligatorios.";
120    }
121 }
122 }
123
124
125 function validarPassword(password) { // Verifica que la Password
ingresada tenga como minimo 6 caracteres, al menos una mayuscula, una minuscula y un
numero.
126     let esValida = true;
127
128     if (!passwordTieneMayuscula(password) || !passwordTieneNumero(password) ||
password.length < 6 || !passwordTieneMinuscula(password)) {
129         esValida = false;
130     }
131
132     return esValida
133 }
134
135 function passwordTieneMayuscula(password) { //Verificar si un texto
contiene mayúsculas
136     let tieneMayuscula = false;
137
138     let i = 0;
139     while (!tieneMayuscula && i < password.length) {

```

```

140         if (password.charCodeAt(i) >= 65 && password.charCodeAt(i) <= 90) {
141             tieneMayuscula = true;
142         }
143         i++;
144     }
145
146     return tieneMayuscula
147 }
148
149 function passwordTieneMinuscula(password) {           //Verificar si un texto
contiene mayúsculas
150     let tieneMinuscula = false;
151
152     let i = 0;
153     while (!tieneMinuscula && i < password.length) {
154         if (password.charCodeAt(i) >= 97 && password.charCodeAt(i) <= 122) {
155             tieneMinuscula = true;
156         }
157         i++;
158     }
159
160     return tieneMinuscula
161 }
162
163 function passwordTieneNumero(password) {           //Verificar si un texto contiene
un número
164     let tieneNumero = false;
165
166     let i = 0;
167     while (!tieneNumero && i < password.length) {
168         if (!isNaN(password[i])) {
169             tieneNumero = true;
170         }
171         i++;
172     }
173
174     return tieneNumero
175 }
176
177 function existeUsuarioPorUsuario(alias) {           //Verificar si ya existe un
usuario con ese alias
178     let existe = false;
179     let i = 0;
180
181     while (!existe && i < usuarios.length) {
182         if ((alias.toUpperCase()) === (usuarios[i].username.toUpperCase())) {
183             existe = true;
184         }
185         i++;
186     }
187
188     return existe;
189 }
190
191 function objUsuarioPorUsuario(username){           // Recibe como parametro el
nombre de un usuario y en caso de existir retorna el objeto con ese nombre de
usuario
192     let encontrado = false;
193     let i = 0;
194     let objetoUsuario = null;

```

```

195     while (!encontrado && i<usuarios.length){
196         if (usuarios[i].username.toUpperCase() == username.toUpperCase()){
197             encontrado = true;
198             objetoUsuario = usuarios[i]
199         }
200         i++
201     }
202     return objetoUsuario
203 }
204 function posUsuarioPorUsuario(username){                // Recibe como parametro el
nombre de un usuario y en caso de existir retorna la posicion en el array
usuarios[]
205     let encontrado = false;
206     let i = 0;
207     while (!encontrado && i<usuarios.length){
208         if (usuarios[i].username.toUpperCase() == username.toUpperCase()){
209             encontrado = true;
210             return i
211         }
212         i++
213     }
214 }
215
216 function objUsuarioPorRazonSocial(razonSocial){        // Recibe como
parametro una razon social y en caso de existir retorna un AI con los objetos con
dicha RZ.
217     let arrayObjetos = [];
218     let i = 0;
219     while (i < usuarios.length){
220         if (usuarios[i].tipo == "Empresa"){
221             if (usuarios[i].razonSocial.toUpperCase() == razonSocial.toUpperCase()){
222                 arrayObjetos.push(usuarios[i]);
223             }
224         }
225         i++
226     }
227     if(arrayObjetos.length>0){
228         return arrayObjetos
229     }
230 }
231
232 function objUsuarioPorNombreFantasia(nombreFantasia){    // Recibe como
parametro un nombde de fantasia y en caso de existir retorna un AI con los objetos
con dicha NF.
233     let i = 0;
234     let arrayObjetos = [];
235
236     while(i<usuarios.length){
237         if(usuarios[i].tipo == "Empresa"){
238
239             if((usuarios[i].fantasia).toUpperCase() == nombreFantasia.toUpperCase())
{
240                 arrayObjetos.push(usuarios[i]);
241             }
242         }
243         i++
244     }
245     if(arrayObjetos.length>0){
246         return arrayObjetos
247     }

```

```

248 }
249
250 function registrarUsuarioPersona(alias, pass, ci, nombre, apellido) {
251     //Agregar una nueva persona al sistema
252     let nuevoUsuarioPersona = new UsuarioPersona(alias, pass, ci, nombre, apellido);
253     usuarios.push(nuevoUsuarioPersona);
254 }
255
256 function registroEmpresa() {
257     let aliasIngresado =
258 document.querySelector("#registroAliasEmpresaF5").value.trim();
259     let passIngresado =
260 document.querySelector("#registroPassEmpresaF5").value.trim();
261     let fantasiaIngresada =
262 document.querySelector("#registroNombreEmpresaF5").value.trim();
263     let rutIngresado =
264 parseInt(document.querySelector("#registroRUTEmpresaF5").value);
265     let razonSocialIngresada =
266 document.querySelector("#registroRazonSocialEmpresaF5").value.trim();
267     let vehiculoAsociado =
268 parseInt(document.querySelector("#registroVehiculoEmpresa").value);
269
270     // Validacion de datos
271     if (aliasIngresado && passIngresado && fantasiaIngresada && rutIngresado !== '' &&
272 razonSocialIngresada && vehiculoAsociado !== 0) { // Revisa que se hayan
273 ingresado datos en todos los campos
274         let contraseniaNoEsValida = !validarPassword(passIngresado);
275         let usuarioExistente = existeUsuarioPorUsuario(aliasIngresado);
276         let razonSocialUnica =
277 existeRazonSocialPorUsuarioEmpresa(razonSocialIngresada);
278         let rutUnico = existeRUTPorUsuarioEmpresa(rutIngresado);
279
280         if (contraseniaNoEsValida) {
281             document.querySelector("#pParaDesplegarErroresEmpresa").innerHTML = "La
282 contraseña debe tener por lo menos una mayúscula, una minúscula y un número."
283         } else if (usuarioExistente) {
284             document.querySelector("#pParaDesplegarErroresEmpresa").innerHTML = "Ya
285 existe un usuario con ese alias.";
286         } else if (razonSocialUnica) {
287             document.querySelector("#pParaDesplegarErroresEmpresa").innerHTML = "Ya
288 existe un usuario con esa razón social.";
289         } else if (rutUnico) {
290             document.querySelector("#pParaDesplegarErroresEmpresa").innerHTML = "Ya
291 existe un usuario con ese RUT.";
292         } else if (isNaN(rutIngresado)) {
293             document.querySelector("#pParaDesplegarErroresEmpresa").innerHTML = "El
294 RUT debe ser un número.";
295         } else {
296             registrarUsuarioEmpresa(aliasIngresado, passIngresado,
297 fantasiaIngresada, rutIngresado, razonSocialIngresada, vehiculoAsociado);
298             displayBorradoCamposRegistroEmpresa();
299             toggleLogIn();
300             document.querySelector("#mensajeLogInNuevoUsuario").innerHTML =
301 `Bienvenido nuevo usuario ${aliasIngresado}. Puedes logearte en la app.`
302         }
303     } else {
304         document.querySelector("#pParaDesplegarErroresEmpresa").innerHTML = "Todos
305 los campos son obligatorios.";
306     }
307 }

```

```

290 }
291
292 function existeRazonSocialPorUsuarioEmpresa(razonSocial) { // Checkear
entre las empresas registradas si ya existe la razón social ingresada. SI existe
devuelve 'true', sino existe 'false'
293     let existe = false;
294     let i = 0;
295     while (!existe && i < usuarios.length) {
296         if (razonSocial === usuarios[i].razonSocial) {
297             existe = true;
298         }
299         i++;
300     }
301     return existe;
302 }
303
304 function existeRUTPorUsuarioEmpresa(rut) { // Checkear entre las empresas
registradas si ya existe el rut ingresado. Si existe devuelve 'true', sino existe
'false'
305     let existe = false;
306     let i = 0;
307     while (!existe && i < usuarios.length) {
308         if (rut === usuarios[i].rut) {
309             existe = true;
310         }
311         i++;
312     }
313     return existe;
314 }
315
316
317 function registrarUsuarioEmpresa(alias, pass, fantasia, rut, razonSocial, vehiculo)
{
//Agregar una nueva empresa al sistema
318     let nuevoUsuarioEmpresa = new UsuarioEmpresa(alias, pass, fantasia, rut,
razonSocial, vehiculo, false, 0);
319     usuarios.push(nuevoUsuarioEmpresa);
320 }
321
322 function seleccionarTipoDeCuentaARegistrar() {
323     let tipoDeCuenta = document.querySelector("#tipoDeCuenta").value;
324     displayRegistrarEmpresaPersona(tipoDeCuenta);
325 }
326
327 function logIn() { // Controla los datos ingresados en el nombre de usuario
y la contraseña. En caso de que sean correctos entra a la aplicacion
328     let user = document.querySelector("#ingresoUsuario").value;
329     let pass = document.querySelector("#ingresoPassword").value
330     let i = 0;
331     let encontrado = false;
332     let mensaje = document.querySelector("#mensajeLogIn");
333
334
335     while (i < usuarios.length && !encontrado) { // Recorremos el array
de usuarios para buscar si los datos ingresados coinciden con algun usuario ya
registrado
336         if (usuarios[i].pass == pass && (usuarios[i].username).toUpperCase() ===
(user).toUpperCase()) {
337             if (usuarios[i].tipo == "Empresa" && usuarios[i].habilitacion == false){
// Si el usuario que desea ingresar es empresa y no esta habilitado por el
administrador

```

```

338         mensaje.innerHTML = "Su cuenta aun no ha sido activada por el
administrador."
339     }else {          // Si pasa el filtro ingresa a la aplicacion
340         let displayPanel = usuarios[i].tipo;
341         displayNavPanel(displayPanel);
342         usuarioLoggeado = usuarios[i];
343     }
344     encontrado = true;
345 } else if(pass =='' || user==''){
346     mensaje.innerHTML = 'Se deben completar todos los campos.'
347 } else {
348     mensaje.innerHTML = "El nombre de ususario y/o contraseña no son
correctos.";
349 }
350
351 i++;
352 document.querySelector("#ingresoUsuario").value = '';
353 document.querySelector("#ingresoPassword").value = '';
354 document.querySelector("#mensajeLogInNuevoUsuario").innerHTML = '';
355 }
356 }
357
358 function crearListaDeEmpresas() {
359     displayErrorBusquedaOFF();
360     let table = document.querySelector("#tableListadoEmpresas")
361     let tablaEmpresas = `
362         <thead>
363             <tr>
364                 <th>Nombre de fantasia</th>
365                 <th>RUT</th>
366                 <th>Razon social</th>
367                 <th>Estado</th>
368                 <th>Habilitar/Deshabilitar</th>
369             </tr>
370         <tbody>
371             `;
372
373     for (let i = 0; i < usuarios.length; i++) {
374         let usuarioActual = usuarios[i];
375         if (usuarioActual.tipo == "Empresa") {
376             let estadoParaModificar =
mostrarTextoHabilitarDeshabilitar(usuarioActual)          // Texto usado en la tabla
para mostrar si la empresa esta habilitada o no por el administrador
377             let estadoActual = leerEstado(usuarioActual);          // Lee el estado
de la empresa
378             tablaEmpresas += `
379                 <tr>
380                     <td>${usuarioActual.fantasia}</td>
381                     <td>${usuarioActual.rut}</td>
382                     <td>${usuarioActual.razonSocial}</td>
383                     <td>${estadoActual}</td>
384                     <td><button class="btn btn-warning btnCambioEstado"
btnempresaUsername="${usuarioActual.username}">${estadoParaModificar}</button></td>
385                 </tr>`
386             }
387         }
388     }
389     tablaEmpresas += `</tbody>`
390     table.innerHTML = tablaEmpresas;
391     activarBotonesCambioDeEstado();
392 }

```

```

392
393 function mostrarTextoHabilitarDeshabilitar(usuarioEmpresa) {           // Leer el
    estado (habilitado o deshabilitado) de la empresa. Devuelve el texto "habilitar" o
    "deshabilitar"
394     let texto = "";
395     if (usuarioEmpresa.habilitacion){
396         texto = "Deshabilitar";
397     } else {
398         texto = "Habilitar";
399     }
400     return texto
401 }
402 }
403
404 function leerEstado(usuarioEmpresa){           // Leer el estado de la empresa y
    devolverlo
405     let estadoEmpresa
406     if (usuarioEmpresa.habilitacion){
407         estadoEmpresa = "Habilitado";
408     } else {
409         estadoEmpresa = "Deshabilitado";
410     }
411     return estadoEmpresa
412 }
413
414 function cambiarEstadoDeEmpresa(){           // Recibe como parametro una empresa, cambia
    su estado de habilitado a deshabilitado o viceversa
415     let empresaUsernameClickeado = this.getAttribute("btnempresaUsername")
416     let empresa = objUsuarioPorUsuario(empresaUsernameClickeado);
417
418     if (empresa.habilitacion){
419         empresa.habilitacion = false;
420     }else{
421         empresa.habilitacion = true;
422     }
423     crearListaDeEmpresas();           // Actualizar el listado de empresas
424     crearListaDeEmpresasFiltrado();           // Actualizar el listado de empresas
425 }
426
427 function activarBotonesCambioDeEstado(){           //
    Activa todos los botones de "Habilitar/Deshabilitar" en la lista de empresas del
    panel de administrador
428     let listaBotones = document.querySelectorAll(".btnCambioEstado");           //
    Guardar todos los botones con el Tag indicado en un array
429     for(let i = 0; i<listaBotones.length;i++){
430         listaBotones[i].addEventListener('click',cambiarEstadoDeEmpresa);
431     }
432 }
433 }
434
435 function buscadorEmpresas(input){
    // Generar un array con la empresa que cumple con el criterio de
    busqueda
436     let criterioDeBusqueda = input.trim();
437     let empresasEncontradasPorRZ = objUsuarioPorRazonSocial(criterioDeBusqueda);
438     let empresasEncontradasPorNF = objUsuarioPorNombreFantasia(criterioDeBusqueda);
439
440     if(empresasEncontradasPorRZ){
        // Buscamos primero por Razon Social
    }

```



```

441         return empresasEncontradasPorRZ;                                // En caso
de encontrar generamos el array con la empresa encontrada
442     }
443     else if (empresasEncontradasPorNF){
        // En caso de no encontrar nada por RZ buscamos por NF
444         return empresasEncontradasPorNF;                                // En caso
de encontrar generamos el array con la empresa encontrada
445     } else {
        // En caso de no encontrar resultados por ningun metodo desplegar msj de
error
446         return "Error";
447     }
448 }
449 }
450
451 function crearListaDeEmpresasFiltrado() {
452     let criterioDeBusqueda =
document.querySelector("#textoBusquedaEmpresaF2").value;    // Leer del HTML
lo escrito en el campo de busqueda
453     let arrayEmpresas = buscadorEmpresas(criterioDeBusqueda);
454     if (arrayEmpresas != 'Error'){
455         let table = document.querySelector("#tableListadoEmpresasFiltrado");
456         table.innerHTML = ` <thead>
457             <tr>
458                 <th>Nombre de fantasia</th>
459                 <th>RUT</th>
460                 <th>Razon social</th>
461                 <th>Estado</th>
462                 <th>Habilitar/Deshabilitar</th>
463             </tr>
464             <tbody>
465
466                 for (let i = 0; i < arrayEmpresas.length; i++) {
467                     let estadoParaModificar =
mostrarTextoHabilitarDeshabilitar(arrayEmpresas[i])
468                     let estadoActual = leerEstado(arrayEmpresas[i]);
469
470                     table.innerHTML += ` <tr>
471                         <td>${arrayEmpresas[i].fantasia}</td>
472                         <td>${arrayEmpresas[i].rut}</td>
473                         <td>${arrayEmpresas[i].razonSocial}</td>
474                         <td>${estadoActual}</td>
475                         <td><button class="btn btn-warning btnCambioEstado"
btnempresaUsername="${arrayEmpresas[i].username}">${estadoParaModificar}</button>
476                         </td>
477                     </tr>`
478
479                 }
480                 table.innerHTML += `</tbody>`
481             }
482             activarBotonesCambioDeEstado();
483         }
484     }
485
486 function cortarPath(path) {          // Corta el fakepath de la ruta de la foto
487     let posDeArranquePath = posicionUltimoSlash(path) + 1;
488
489     let nuevoPath = path.slice(posDeArranquePath);
490

```



```

491     return nuevoPath
492 }
493
494 function posicionUltimoSlash(path) {          // Busca el último slash del path y
devuelve su posicion
495     let posUltimoBackSlash = -1;
496
497     for (let i = 0; i < path.length; i++) {
498         if (path[i] == "\\") {
499             posUltimoBackSlash = i;
500         }
501     }
502
503     return posUltimoBackSlash
504 }
505
506
507 function btnSolicitudEnvioHandler() {          // Verifica que todos los datos
ingresados sean correctos, y si lo son le envía a la función
"realizarSolicitudEnvio" los datos necesarios
508     let tipoVehiculoIngresado =
parseInt(document.querySelector("#solicitudEnvioVehiculo").value);
509     let distanciaIngresada =
document.querySelector("#solicitudEnvioDistancia").value;
510     let descripcionIngresada =
document.querySelector("#solicitudEnvioDescripcion").value;
511     let fotoIngresada = document.querySelector("#solicitudEnvioFoto").value;
512
513     if (tipoVehiculoIngresado && distanciaIngresada && descripcionIngresada &&
fotoIngresada) {
514         if (!isNaN(distanciaIngresada)) {
515             let distanciaNumerica = parseInt(distanciaIngresada);
516             let fotoIngresadaValidada = cortarPath(fotoIngresada);          // Se le
quita el fakepath a la foto ingresada
517             if (distanciaNumerica > 0) {
518                 realizarSolicitudEnvio(tipoVehiculoIngresado, distanciaNumerica,
descripcionIngresada, fotoIngresadaValidada);
519                 displaySuccessSolicitudEnvioON();
520                 displayErrorSolicitudEnvioOFF();
521             } else {
522                 displayErrorSolicitudEnvioON("La cantidad de km ingresada debe ser
mayor a cero");
523                 displaySuccessSolicitudEnvioOFF();
524             }
525         } else {
526             displayErrorSolicitudEnvioON("La distancia ingresada debe ser numerica");
527             displaySuccessSolicitudEnvioOFF();
528         }
529     } else {
530         displayErrorSolicitudEnvioON("Completar todos los datos");
531         displaySuccessSolicitudEnvioOFF();
532     }
533 }
534
535 function realizarSolicitudEnvio(tipoVehiculo, distancia, descripcion, foto) {
// Recibe los datos necesarios y crea el pedido, se le asigna al usuario
correspondiente
536     let pedido = new Envio(tipoVehiculo, distancia, descripcion, foto, null,
"Pendiente", usuarioLoggeado.username);
537     envios.push(pedido);

```

```

538     let i = posUsuarioPorUsuario(usuarioLoggeado.username);
539     usuarios[i].pedidos.push(pedido.id);
540 }
541
542 function crearListadoEnviosPersona() {      // Genera una tabla para listar los
envios del usuario
543     let table = document.querySelector("#listadoDeEnviosPersona");
544     let tableListadoEnvios = `
545     <header>
546         <tr>
547             <th>Foto</th>
548             <th>Descripción</th>
549             <th>Estado</th>
550             <th>Empresa</th>
551         </tr>
552     </header>
553     <body id="bodyListadoDeEnviosPersona"></tbody>`;
554
555     let hayEnvios = hayEnviosParaAgregar();
556     if (!hayEnvios) {
557         resultado = "No tiene pedidos asignados";
558     } else {
559         let bodyListado = armarBodyListadoEnviosPersona();      // Si tiene pedidos
llama a la función "armarBodyListadoEnviosPersona" y lo suma a la tabla
560         resultado = tableListadoEnvios + bodyListado
561     }
562
563     table.innerHTML = resultado
564 }
565
566 function hayEnviosParaAgregar() {      // Checkea si el usuario tiene pedidos
realizados
567     let hayEnvios = false;
568
569     let i = 0;
570     while (i < envios.length && !hayEnvios) {
571         let envioActual = envios[i];
572         if (usuarioLoggeado.username == envioActual.persona) {
573             hayEnvios = true;
574         }
575         i++
576     }
577
578     return hayEnvios
579 }
580
581 function armarBodyListadoEnviosPersona() {      // Arma un body con los pedidos
realizados por el usuario
582     let bodyListadoEnvios = ``;
583     for (let i = 0; i < envios.length; i++) {
584         let envioActual = envios[i];
585         let empresaAsignada = buscarEmpresaPorEnvio(envioActual);      // LLama a
la función "buscarEmpresaPorEnvio" para encontrar la empresa que realizó tal pedido
586         if (usuarioLoggeado.username == envioActual.persona) {
587             bodyListadoEnvios += `
588                 <tr>
589                     <td></td>
590                     <td>${envioActual.descripcion}</td>
591                     <td>${envioActual.estado}</td>
592                     <td>${empresaAsignada}</td>

```

```

593         </tr>`
594     }
595 }
596 return bodyListadoEnvios
597 }
598
599 function buscarEmpresaPorEnvio(objEnvio){ // Busca si el pedido tiene una
    empresa asignada y retorna true o false
600     if (objEnvio.empresa == null){
601         return "No hay empresa asignada"
602     } else {
603         return objEnvio.empresa
604     }
605 }
606
607
608 function calcularInfoEstadisticaPersona() { // Muestra la información
    estadística de los pedidos del usuario
609     let contadorPendientes = 0;
610     let contadorEnTransito = 0;
611     let contadorFinalizado = 0;
612
613     for (let i = 0; i < envios.length; i++) {
614         let envioActual = envios[i];
615         if (envioActual.persona == usuarioLoggeado.username) {
616             if (envioActual.estado == "Pendiente") {
617                 contadorPendientes++;
618             } else if (envioActual.estado == "En tránsito") {
619                 contadorEnTransito++;
620             } else if (envioActual.estado == "Finalizado") {
621                 contadorFinalizado++;
622             }
623         }
624     }
625
626     let pedidosTotales = contadorPendientes + contadorEnTransito +
    contadorFinalizado;
627     let porcentajeTomados = Math.round((contadorEnTransito + contadorFinalizado) /
    pedidosTotales * 100) + "%";
628
629     let resultado = `Envios totales: ${pedidosTotales}<br>
    Envios pendientes: ${contadorPendientes}<br>
    Envios en tránsito: ${contadorEnTransito}<br>
    Envios finalizados: ${contadorFinalizado}<br>
    Porcentaje de solicitud de envios tomados: ${porcentajeTomados}`
630
631     document.querySelector("#infoEstadistica").innerHTML = resultado;
632 }
633
634
635 function agregarVehiculo(tipo) { // Agregar un
    nuevo vehiculo al sistema. No agrega vehiculos ya existentes. Devuelve true si
    existe, false si no.
636     let vehiculoExistente = false;
637     let i = 0;
638     while (i < vehiculos.length && vehiculoExistente == false) { //
    Verificar que el vehiculo no existe en el sistema
639         if ((vehiculos[i].tipo).toUpperCase() == tipo.toUpperCase()) {
640             vehiculoExistente = true
641         }
642     }
643     i++

```

```

646     }
647     if (i == vehiculos.length) {
648         let nuevoVehiculo = new Vehiculo(tipo)
649         vehiculos.push(nuevoVehiculo)
650     }
651 }
652
653
654 function agregarVehiculoAlSistemaHandler(){ //Verifica que el nuevo vehículo no
esté vacío ni repetido, y si cumple lo agrega al sistema
655     let vehiculoLeido = document.querySelector("#tipoNuevoVehiculo").value;
656     if (vehiculoLeido == '') {
657         displayMensajeErrorVehiculoON("Se debe completar el campo.")
658     } else {
659         if(!checkVehiculoPorTipo(vehiculoLeido)){
660             agregarVehiculo(vehiculoLeido);
661             displayMensajeErrorVehiculoOFF();
662             crearListaDeVehiculos();
663         } else {
664             displayMensajeErrorVehiculoON("El vehiculo ya existe en el sistema.");
665         }
666     }
667     document.querySelector("#tipoNuevoVehiculo").value = ""
668 }
669
670 function checkVehiculoPorTipo(tipo){ // Verifica si existe un
vehiculo por nombre
671     let existe = false;
672     let i = 0;
673     while(!existe && i < vehiculos.length){
674         if((vehiculos[i].tipo).toUpperCase() == tipo.toUpperCase()){
675             existe = true;
676         }
677         i++
678     }
679     return existe
680 }
681
682 function tipoVehiculoPorId(id){ // Recibe como parametro
un id y busca a que vehiculo corresponde (ej 1000 corresponde a Moto)
683     let existe = false;
684     let i=0;
685     while(i<vehiculos.length && !existe) {
686         if(vehiculos[i].id == id){
687             existe = true;
688             return vehiculos[i].tipo
689         }
690         i++
691     }
692 }
693
694 function crearListaDeVehiculos(){ // Genera una tabla con los vehículos
existentes en el sistema
695     let table = document.querySelector("#listadoVehiculos");
696     table.innerHTML = ` <thead>
697         <tr>
698             <th>Tipo</th>
699             <th>ID</th>
700         </tr>
701         </thead>

```

```

702         <tbody>
703     `;
704     for(let i = 0; i < vehiculos.length; i++){
705         table.innerHTML += `<tr>
706             <td>${vehiculos[i].tipo}</td>
707             <td>${vehiculos[i].id}</td>
708         </tr>`
709     }
710     table.innerHTML += `</tbody>`
711 }
712
713 // -- Listado solicitudes pendientes para usuario tipo empresa -- //
714
715 function buscarPedidosDisponiblesPorVehiculo(objEmpresa){ // Recibe como
    parametro un objeto empresa y retorna un array con los objetos de pedidos que puede
    tomar (segun tipo vehiculo y disp)
716     let arrayPedidos = [];
717     for (let i = 0; i<envios.length; i++){
718         if(envios[i].estado == "Pendiente" && envios[i].vehiculo ==
    objEmpresa.vehiculo){
719             arrayPedidos.push(envios[i]);
720         }
721     }
722     return arrayPedidos
723 }
724
725
726 function crearListaDeSolicitudesPendientesEmpresa(){ // Genera la
    tabla con los pedidos que puede aceptar la empresa loggeada
727     let enviosPosibles = buscarPedidosDisponiblesPorVehiculo(usuarioLoggeado);
728     let tableBody = document.querySelector("#bodyListadoEnviosPendientesEmpresa");
729     tableBody.innerHTML = ``;
730
731     if(enviosPosibles.length > 0){
732         for(let i = 0; i< enviosPosibles.length; i++){
733             let j = buscarPosEnvioPorId(enviosPosibles[i].id);
734             let vehiculo = tipoVehiculoPorId(envios[j].vehiculo);
735             let usuarioPersona = objUsuarioPorUsuario(envios[j].persona);
736             tableBody.innerHTML += `<tr>
737                 <td></td>
739                 <td>${vehiculo}</td>
740                 <td>${envios[j].distancia}</td>
741                 <td>${envios[j].estado}</td>
742                 <td>${usuarioPersona.nombre}</td>
743                 <td>${usuarioPersona.apellido}</td>
744                 <td><button class="btn btn-success
    btnAceptarPedido" btnEnvioId="${envios[j].id}">Aceptar</button></td>
745             </tr>`
746         }
747     } else {
748         displayMensajeErrorSolicitudesPendientesON()
749     }
750     activarBotonesAceptarSolicitudesPendientes();
751 }
752
753 function activarBotonesAceptarSolicitudesPendientes(){
    // Activa todos los botones de "Aceptar" en la lista de solicitudes
    pendientes para empresa

```

```

754     let listaBotones = document.querySelectorAll(".btnAceptarPedido");
755         // guardar todos los botones con el Tag indicado en un array
756     for(let i = 0; i < listaBotones.length;i++){
757         listaBotones[i].addEventListener('click',aceptarPedido);
758     }
759 }
760 function activarBotonesFinalizarSolicitudesTransito(){
761     // Activa todos los botones de "Finalizar" en la lista de solicitudes
    aceptadas para empresa
762     let listaBotones = document.querySelectorAll(".btnFinalizarPedido");
763     // guardar todos los botones con el Tag indicado en un array
764     for(let i = 0; i < listaBotones.length;i++){
765         listaBotones[i].addEventListener('click',finalizarPedido);
766         // A todos los botones les asigna un event listener de click, que
    corre la funcion finalizarPedido()
767     }
768 }
769 function aceptarPedido(){
770     let envioIdClickeado = this.getAttribute("btnEnvioId")
771     let j = buscarPosEnvioPorId(envioIdClickeado);
772     envios[j].estado = "En tránsito";
773     envios[j].empresa = usuarioLoggeado.razonSocial;
774     usuarioLoggeado.pedidos.push(j);
775     crearListaDeSolicitudesPendientesEmpresa();
776     // Actualizar el listado de pedidos pendientes especificos
777     crearListadoDeSolicitudesTomadasEmpresa();
778     // Actualizar listado de solicitudes en transito + finalizadas
779 }
780 function finalizarPedido(){
781     // Cambiar el estado del envio de "En transito" a "Finalizado" y
    actualizar la tabla
782     let envioIdClickeado = this.getAttribute("btnEnvioId")
783     // Obtener el ID del envio
784     let j = buscarPosEnvioPorId(envioIdClickeado);
785     // Buscar el envio por ID
786     envios[j].estado = "Finalizado";
787     // Modificar su estado
788     crearListadoDeSolicitudesTomadasEmpresa()
789     // Actualizar el listado de solicitudes "En transito" y "Finalizadas"
790 }
791 function buscarPosEnvioPorId(id){ // Recibe como parametro el id de un envio,
    busca en el array de envios la posicion del obj con ese id
792     let i = 0;
793     let existe = false;
794     while(i<envios.length && !existe){
795         if(envios[i].id == id){
796             existe = true;
797             return i
798         }
799         i++
800     }
801 }

```

```

799
800 function buscarEnviosPorEmpresa(objEmpresa) {           // Recibe como parametro el
objeto empresa y busca sus pedidos asignados
801     let arrayPedidos = [[],[[]];
802     for (let i = 0; i < envios.length; i++){
803         if (envios[i].estado == "En tránsito" && envios[i].empresa ==
objEmpresa.razonSocial) {
804             arrayPedidos[0].push(envios[i]);
805         } else if ((envios[i].estado == "Finalizado" && envios[i].empresa ==
objEmpresa.razonSocial)) {
806             arrayPedidos[1].push(envios[i]);
807         }
808     }
809     return arrayPedidos
810 }
811
812 function crearListadoDeSolicitudesTomadasEmpresa(){           // Genera una tabla con
la información de todos los pedidos que la empresa loggeada tiene asignada, también
contiene botones para finalizar cualquier pedido
813     let table = document.querySelector("#tablaEnviosTomadosF9");
814     let enviosEnTransitoYFinalizados = buscarEnviosPorEmpresa(usuarioLoggeado);
815     let enviosEnTransito = enviosEnTransitoYFinalizados[0];
816     let enviosFinalizados = enviosEnTransitoYFinalizados[1];
817     let vehiculoEmpresa = tipoVehiculoPorId(usuarioLoggeado.vehiculo)
818
819     if (enviosEnTransito.length > 0 || enviosFinalizados.length > 0){
820         displayMensajeErrorSolicitudesEnTransitoYFinalizadosOFF();
821         let tablaPedidosTomados = `<header>
822                                     <tr>
823                                         <td><strong>Foto</strong></td>
824                                         <td><strong>Vehiculo</strong></td>
825                                         <td><strong>Distancia (Km)</strong></td>
826                                         <td><strong>Estado</strong></td>
827                                         <td><strong>Nombre</strong></td>
828                                         <td><strong>Apellido</strong></td>
829                                         <td><strong>Finalizar envio</strong>
</td>
830                                     </tr>
831                                     </header>
832                                     <body>`
833         for(let i = 0; i < enviosEnTransito.length; i++){
834             let j = buscarPosEnvioPorId(enviosEnTransito[i].id);
835             let usuarioPersona = objUsuarioPorUsuario(envios[j].persona)
836             let vehiculoEnvio = tipoVehiculoPorId(envios[j].vehiculo);
837             if (vehiculoEnvio == vehiculoEmpresa) {
838                 tablaPedidosTomados += `<tr>
839                                     <td></td>
840                                     <td>${vehiculoEnvio}</td>
841                                     <td>${envios[j].distancia}</td>
842                                     <td>${envios[j].estado}</td>
843                                     <td>${usuarioPersona.nombre}</td>
844                                     <td>${usuarioPersona.apellido}</td>
845                                     <td><button class="btn btn-success
btnFinalizarPedido" btnEnvioId="${envios[j].id}">Finalizar</button></td>
846                                     </tr>`
847             }
848         }
849
850         for(let i = 0; i < enviosFinalizados.length; i++) {

```



```

851     let j = buscarPosEnvioPorId(enviosFinalizados[i].id);
852     let usuarioPersona = objUsuarioPorUsuario(envios[j].persona);
853     let vehiculoEnvio = tipoVehiculoPorId(envios[j].vehiculo);
854     if (vehiculoEnvio == vehiculoEmpresa) {
855         tablaPedidosTomados += `<tr>
856                                     <td></td>
857                                     <td>${vehiculoEnvio}</td>
858                                     <td>${envios[j].distancia}</td>
859                                     <td>${envios[j].estado}</td>
860                                     <td>${usuarioPersona.nombre}</td>
861                                     <td>${usuarioPersona.apellido}</td>
862                                     <td><button class="btn btn-dark"
disabled>Finalizado</button></td>
863                                     </tr>`;
864     };
865     }
866     tablaPedidosTomados += `</body>`;
867     table.innerHTML = tablaPedidosTomados;
868 } else {
869     displayMensajeErrorSolicitudesEnTransitoYFinalizadosON("No hay envios
asociados a su empresa.");
870 }
871 activarBotonesFinalizarSolicitudesTransito();
872 }
873
874 function calcularInfoEstadisticaEmpresa() { // Muestra que persona realizó
más pedidos con la empresa loggeada
875     let infoEstadisticaMejoresUsuarios;
876
877     let usuariosConMasEnvios = encontrarUsuariosConMasEnvios();
878
879     if (usuariosConMasEnvios.length > 0) {
880         infoEstadisticaMejoresUsuarios = `La/s personas con mas envios finalizados
es/son: <br>`;
881         for (let i = 0; i < usuariosConMasEnvios.length; i++) {
882             let usuarioActual = usuariosConMasEnvios[i];
883             infoEstadisticaMejoresUsuarios += `${i + 1}. ${usuarioActual.nombre}
${usuarioActual.apellido} con ${encontrarNumMasGrande()} envíos.<br>`
884         }
885     } else {
886         infoEstadisticaMejoresUsuarios = "No hay pedidos finalizados asignados a
esta empresa.";
887     }
888
889     empresaDisplayEstadisticaMejoresUsuarios(infoEstadisticaMejoresUsuarios);
890 }
891
892 function encontrarUsuariosConMasEnvios() { // Crea un array con los
usuarios que tengan más pedidos finalizados
893     let cantPedidosFinalizadosPorUsuario = [];
894
895     for (let i = 0; i < usuarios.length; i++) {
896         let usuarioActual = usuarios[i];
897
898         if (usuarioActual.tipo == "Persona") {
899             let contEnviosFinalizados = 0;
900
901             for (let j = 0; j < usuarioActual.pedidos.length; j++) {
902                 let idPedidoActual = usuarioActual.pedidos[j];

```



```

903         let pedidoActual = buscarPedidoPorId(idPedidoActual);
904         if (pedidoActual.estado == "Finalizado" && pedidoActual.empresa ==
usuarioLoggeado.razonSocial) {
905             contEnviosFinalizados++
906         }
907     }
908
909     cantPedidosFinalizadosPorUsuario.push(contEnviosFinalizados)
910
911     } else {
912         cantPedidosFinalizadosPorUsuario.push(null)
913     }
914 }
915
916 let mayorCantEnvios = encontrarNumMasGrande(cantPedidosFinalizadosPorUsuario);
917
918 let mejoresUsuarios = [];
919
920 if (mayorCantEnvios != 0) {
921     mejoresUsuarios = armarArrayConMejoresUsuarios(mayorCantEnvios,
cantPedidosFinalizadosPorUsuario);
922 }
923
924 return mejoresUsuarios
925 }
926
927 function buscarPedidoPorId(idABuscar) {           // Recibe como parametro un id y
busca el pedido con ese id, luego devuelve el pedido
928     let pedido;
929     let pedidoEncontrado = false;
930
931     let i = 0
932     while (i < envios.length && !pedidoEncontrado) {
933         let pedidoActual = envios[i];
934         if (pedidoActual.id == idABuscar) {
935             pedido = pedidoActual;
936             pedidoEncontrado = true;
937         }
938         i++
939     }
940
941     return pedido
942 }
943
944 function encontrarNumMasGrande(arrayNumerico) {           // Recibe como parametro un
array n mero y busca el n mero m s grande del array y lo devuelve
945     let numMasGrande = 0;
946
947     for (let i = 0; i < envios.length; i++) {
948         let numActual = arrayNumerico[i];
949         if (numActual != null && numActual > numMasGrande ) {
950             numMasGrande = numActual
951         }
952     }
953
954     return numMasGrande
955 }
956
957 function armarArrayConMejoresUsuarios(numMasGrande, pedidosPorUsuario) {           //
Recibe como parametro un array (pedidosPorUsuario) con la cantidad de pedidos

```

finalizados por persona y el mayor número (numMasGrande) encontrado en él y devuelve un array con las personas con más pedidos

```
958     let mejoresUsuarios = [];
959
960     for (let i = 0; i < pedidosPorUsuario.length; i++) {
961         let numActual = pedidosPorUsuario[i];
962         if (numActual == numMasGrande) {
963             mejoresUsuarios.push(usuarios[i])
964         }
965     }
966
967     return mejoresUsuarios
968 }
969
970
971 function cantEnviosDeEstado(estadoElegido) { // Recibe como
parametro un estado ("Pendiente", "Finalizado" o "En Transito") y devuelve la
cantidad de pedidos con el estado del parametro para la empresa seleccionada
972     let contDeEnvios = 0;
973
974     if (usuarioLoggeado.pedidos.length > 0) {
975         let arrIdsDeEnviosDeEmpresa = usuarioLoggeado.pedidos;
976
977         for (let i = 0; i < arrIdsDeEnviosDeEmpresa.length; i++) {
978             let idActual = arrIdsDeEnviosDeEmpresa[i]
979             let pedidoActual = buscarPedidoPorId(idActual)
980
981             if (pedidoActual.estado == estadoElegido) {
982                 contDeEnvios++
983             }
984         }
985     }
986
987     return contDeEnvios
988 }
989
990 function crearTablaDeEnviosPorEstadoEmpresa(){ // Crea la tabla para la
empresa loggeada con los envios pendientes, transito y finalizados y sus cantidades.
991     let enviosPendientes = cantEnviosDeEstado("Pendiente");
992     let enviosEnTransito = cantEnviosDeEstado("En tránsito");
993     let enviosFinalizados = cantEnviosDeEstado("Finalizado");
994
995     document.querySelector("#infoEstadisticaEmpresaCantidadPorEstado").innerHTML =
996
997         <table class="table">
998             <header>
999                 <tr>
1000                     <td><strong>Estado</strong></td>
1001                     <td><strong>Cantidad</strong></td>
1002                 </tr>
1003             </header>
1004             <tbody>
1005                 <tr>
1006                     <td>Pendiente</td>
1007                     <td>${enviosPendientes}</td>
1008                 </tr>
1009                 <tr>
1010                     <td>En tránsito</td>
1011                     <td>${enviosEnTransito}</td>
1012                 </tr>
```

```

1012         <tr>
1013             <td>Finalizados</td>
1014             <td>${enviosFinalizados}</td>
1015         </tr>
1016     </tbody>
1017 </table>`
1018 }
1019
1020 function calcularInfoEstadisticaAdmin() {
1021     let tableBody = document.querySelector("#tablaInfoEstadisticaKMAAdminBody");
1022     tableBody.innerHTML = '';
1023
1024     for(let i = 0; i < usuarios.length; i++) {
1025         if (usuarios[i].tipo == "Empresa") {
1026             let nombreEmpresa = usuarios[i].razonSocial;
1027             let kmTotales = kmPorEmpresa(nombreEmpresa);
1028             let totalFinalizados = enviosFinalizadosPorEmpresa(nombreEmpresa)
1029             tableBody.innerHTML += `<tr>
1030                                     <td>${nombreEmpresa}</td>
1031                                     <td>${kmTotales}</td>
1032                                     <td>${totalFinalizados}</td>
1033                                 </tr>`
1034         }
1035     }
1036 }
1037
1038 function kmPorEmpresa(empresa){ // Recibe como parametro la razon
social y suma todos los km de los envios finalizados
1039     let total = 0;
1040
1041     for (let i=0; i < envios.length; i++) {
1042         if (envios[i].empresa == empresa && envios[i].estado == "Finalizado") {
1043             total += envios[i].distancia;
1044         }
1045     }
1046
1047     return total;
1048 }
1049
1050 function enviosFinalizadosPorEmpresa(empresa){ // Recibe como parametro
la razon social y suma la cantidad de envios finalizados
1051     let totalFinalizados = 0;
1052     for(let i=0; i<envios.length;i++){
1053         if(envios[i].empresa == empresa && envios[i].estado == "Finalizado"){
1054             totalFinalizados++;
1055         }
1056     }
1057     return totalFinalizados
1058 }

```

```

1 class UsuarioPersona{
2     constructor(_username, _pass, _CI, _nombre, _apellido, _pedidos){
3         this.username = _username;
4         this.pass = _pass;
5         this.cedula = _CI;
6         this.nombre = _nombre;
7         this.apellido = _apellido;
8         this.pedidos = [];
9         this.tipo = "Persona";
10    }
11 }
12
13 class UsuarioEmpresa {
14     constructor(_username, _pass, _fantasia, _rut, _razonSocial, _vehiculo,
15     _habilitacion){
16         this.username = _username;
17         this.pass = _pass;
18         this.fantasia = _fantasia;
19         this.rut = _rut;
20         this.razonSocial = _razonSocial;
21         this.vehiculo = _vehiculo;
22         this.habilitacion = _habilitacion; // True = empresa habilitada. False =
23         empresa deshabilitada.
24         this.pedidos = [];
25         this.tipo = "Empresa"
26     }
27 }
28
29 class UsuarioAdmin {
30     constructor(_username, _pass){
31         this.username = _username;
32         this.pass = _pass;
33         this.tipo = "Administrador"
34     }
35 }
36
37 class Envio {
38     constructor(_vehiculo, _distancia, _descripcion, _img, _empresa, _estado,
39     _persona){
40         this.vehiculo = _vehiculo;
41         this.distancia = _distancia;
42         this.descripcion = _descripcion;
43         this.img = _img;
44         this.empresa = _empresa; // Corresponde a la empresa que toma el
45         pedido
46         this.estado = _estado; // Estado corresponde a: "Pendiente", "En
47         tránsito", "Finalizado"
48         this.persona = _persona;
49         this.id = idEnvio;
50         idEnvio++;
51     }
52 }
53
54 class Vehiculo {
55     constructor(_tipo){
56         this.tipo = _tipo;
57         this.id = codigoVehiculo;
58         codigoVehiculo++
59     }
60 }

```



```

1  // SELECTORES //
2  //--Log In & Registro--//
3  const btnMostrarLogIn = document.querySelector("#btnDisplayLogIn");
4  const btnMostrarRegister = document.querySelector("#btnDisplayRegistrar");
5  const btnCerrarSesion = document.querySelector("#btnDisplayCerrarSesion");
6  const selectTipoCuenta = document.querySelector("#tipoDeCuenta");
7
8  //--Usuario: Adminitrador--//
9  const btnAdminEmpresas = document.querySelector("#btnAdminListarEmpresas");
10 const btnAdminVehiculo = document.querySelector("#btnAdminCrearVehiculos");
11 const btnAdminEstadistica = document.querySelector("#btnAdminEstadistica");
12 const btnMostrarEmpresasFiltradas = document.querySelector("#btnMostrarEmpresasF2");
13 const pMensajeErrorFiltrarEmpresas =
    document.querySelector("#mensajeErrorFiltrarEmpresas");
14 const pMensajeErrorIngresarVehiculo =
    document.querySelector("#mensajeErrorVehiculos");
15
16 //--Usuario: Persona--//
17 const btnPersonaSolicitudEnvio =
    document.querySelector("#btnPersonaRealizarSolicitudEnvio");
18 const btnPersonaListadoEnvios = document.querySelector("#btnPersonaListadoEnvios");
19
20 //--Usuario: Empresa--//
21 const btnEmpresaListadoEnvios = document.querySelector("#btnEmpresaListadoEnvios");
22 const btnEmpresaEstadistica = document.querySelector("#btnEmpresaEstadistica");
23
24 // EVENT LISTENERS //
25 btnMostrarLogIn.addEventListener('click', toggleLogIn);
26 btnMostrarRegister.addEventListener('click', toggleRegister);
27 btnCerrarSesion.addEventListener('click', cerrarSesion);
28 selectTipoCuenta.addEventListener('click', displayBtnSelect);
29
30 btnAdminEmpresas.addEventListener('click', adminDisplayEmpresas);
31 btnAdminVehiculo.addEventListener('click', adminDisplayVehiculos);
32 btnAdminEstadistica.addEventListener('click', adminDisplayEstadistica);
33 btnMostrarEmpresasFiltradas.addEventListener('click', adminDisplayEmpresasFiltradas);
34
35 btnPersonaSolicitudEnvio.addEventListener('click', personaDisplaySolicitudEnvio);
36 btnPersonaListadoEnvios.addEventListener('click', personaDisplayListadoEnvio);
37
38 btnEmpresaListadoEnvios.addEventListener('click', empresaDisplayListadoEnvio);
39 btnEmpresaEstadistica.addEventListener('click',
    empresaDisplayEstadisticaMejoresUsuarios)
40 btnEmpresaEstadistica.addEventListener('click',
    empresaDisplayEstadisticaEnviosPorEstado)
41
42
43
44 // FUNCIONES //
45 function cerrarSesion() {
46     usuarioLoggeado = null;
47     ocultarPantallas();
48     btnMostrarRegister.style.display = "inline-block"
49     btnCerrarSesion.style.display = "none";
50     btnMostrarLogIn.style.display = "inline-block"
51     document.querySelector("#panelLogInF1").style.display = "block"
52     document.querySelector("#mensajeLogIn").innerHTML = '';
53 }
54
55 function toggleLogIn() {

```

```

56     ocultarPantallas();
57     document.querySelector("#panelLogInF1").style.display = "block";
58     document.querySelector("#ingresoUsuario").value = "";
59     document.querySelector("#ingresoPassword").value = "";
60     document.querySelector("#mensajeLogInNuevoUsuario").innerHTML = '';
61     document.querySelector("#mensajeLogIn").innerHTML = '';
62 }
63
64 function toggleRegister() {
65     document.querySelector("#btnTipoDeCuenta").style.display = "none";
66     document.querySelector("#tipoDeCuenta").value = "0";
67     ocultarPantallas();
68     document.querySelector("#panelRegistroF5").style.display = "block";
69 }
70
71
72 function ocultarPantallas() { //
73     Ocultar todas las pantallas
74     document.querySelector("#panelLogInF1").style.display = "none"
75     document.querySelector("#listadoDeEmpresasF2").style.display = "none"
76     document.querySelector("#panelCreacionVehiculosF3").style.display = "none" //
77     Crear tipos de vehiculos
78     document.querySelector("#verInfoEstadisticaF4").style.display = "none" //
79     Visualizar info estadistica
80     document.querySelector("#panelRegistroF5").style.display = "none" //
81     Registrarse en la app
82     document.querySelector("#solicitudEnvioF6").style.display = "none" //
83     Realizar solicitud de envio
84     document.querySelector("#listadoEnviosF7").style.display = "none" //
85     Listado de envios
86     document.querySelector("#listadoEnviosPendientesF8").style.display = "none" //
87     Listado de sol pendientes (empresas)
88     document.querySelector("#listadoEnviosTomadosF9").style.display = "none" //
89     Listado de sol tommdas (empresas)
90     document.querySelector("#registroUsuario").style.display = "none"
91     document.querySelector("#registroEmpresa").style.display = "none"
92     document.querySelector("#panelAdmin").style.display = "none"
93     document.querySelector("#panelEmpresa").style.display = "none"
94     document.querySelector("#panelPersona").style.display = "none"
95     document.querySelector("#infoEstadisticaEmpresaMejoresUsuarios").style.display =
96     "none";
97     document.querySelector("#infoEstadisticaEmpresaCantidadPorEstado").style.display
98     = "none";
99     pMensajeErrorFiltrarEmpresas.style.display = "none";
100    pMensajeErrorIngresarVehiculo.style.display = "none";
101    btnCerrarSesion.style.display = "none";
102 }
103
104 function displayRegistrarEmpresaPersona(usuario) {
105     updateSelectVehiculos("#registroVehiculoEmpresa");
106     if (usuario == "Empresa") { //
107         Muestra unicamente las opciones de registro de Empresa
108         document.querySelector("#registroUsuario").style.display = "none";
109         document.querySelector("#registroEmpresa").style.display = "block";
110     } else if (usuario == "Persona") { //
111         Muestra unicamente las opciones de registro de Persona
112         document.querySelector("#registroUsuario").style.display = "block";
113         document.querySelector("#registroEmpresa").style.display = "none";
114     } else if (usuario == "") {

```

```

103     document.querySelector("#errorSelectTipocuenta").value = "Debe seleccionar un
tipo de dato.";
104 }
105 displayBorradoCamposRegistroEmpresa();
106 displayBorradoCamposRegistroPersona();
107 }
108
109 function displayBorradoCamposRegistroEmpresa() {
110     document.querySelector("#registroAliasEmpresaF5").value = "";
111     document.querySelector("#registroNombreEmpresaF5").value = "";
112     document.querySelector("#registroRUTEmpresaF5").value = "";
113     document.querySelector("#registroRazonSocialEmpresaF5").value = "";
114     document.querySelector("#registroPassEmpresaF5").value = "";
115     document.querySelector("#registroVehiculoEmpresa").value = 0;
116
117     //Borrar error
118     document.querySelector("#pParaDesplegarErroresEmpresa").innerHTML = "";
119 }
120
121 function displayBorradoCamposRegistroPersona() {
122     document.querySelector("#registroAliasUsuarioPersonaF5").value = "";
123     document.querySelector("#registroPassUsuarioPersonaF5").value = "";
124     document.querySelector("#registroCIUsuarioPersonaF5").value = "";
125     document.querySelector("#registroNombreUsuarioPersonaF5").value = "";
126     document.querySelector("#registroApellidoUsuarioPersonaF5").value = "";
127
128     //Borrar error
129     document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "";
130 }
131
132 function displayNavPanel(tipo) {
133     switch (tipo) {
134         case "Administrador":
135             document.querySelector("#panelAdmin").style.display = "block";
136             document.querySelector("#panelLogInF1").style.display = "none";
137             btnMostrarLogIn.style.display = "none";
138             btnMostrarRegister.style.display = "none";
139             btnCerrarSesion.style.display = "block";
140             break;
141
142         case "Empresa":
143             document.querySelector("#panelEmpresa").style.display = "block";
144             document.querySelector("#panelLogInF1").style.display = "none";
145             btnMostrarLogIn.style.display = "none";
146             btnMostrarRegister.style.display = "none";
147             btnCerrarSesion.style.display = "block";
148             break;
149
150         case "Persona":
151             document.querySelector("#panelPersona").style.display = "block";
152             document.querySelector("#panelLogInF1").style.display = "none";
153             btnMostrarLogIn.style.display = "none";
154             btnMostrarRegister.style.display = "none";
155             btnCerrarSesion.style.display = "block";
156             break;
157
158         default:
159             break;
160     }
161 }

```



```

162
163 function updateSelectVehiculos(id) {
164     document.querySelector(id).innerHTML = '';
165     for (let i = 0; i < vehiculos.length; i++) {
166         if (i == 0) {
167             document.querySelector(id).innerHTML += `<option selected disabled value
= 0>Ingrese una opcion</option>`
168         }
169         document.querySelector(id).innerHTML += `<option
value=${vehiculos[i].id}>${vehiculos[i].tipo}</option>`
170     }
171 }
172
173 function displayBtnSelect() {
174     if (selectTipoCuenta.value == "Empresa" || selectTipoCuenta.value == "Persona") {
175         document.querySelector("#btnTipoDeCuenta").style.display = "block";
176     }
177 }
178
179 function adminDisplayEmpresas() {
180     crearListaDeEmpresas();
181     document.querySelector("#listadoDeEmpresasF2").style.display = "block";
182     document.querySelector("#panelCreacionVehiculosF3").style.display = "none"
183     document.querySelector("#verInfoEstadisticaF4").style.display = "none"
184     document.querySelector("#tableListadoEmpresasFiltrado").style.display = "none"
185     // Ocultar las empresas filtradas
186     document.querySelector("#tableListadoEmpresas").style.display = "table"
187     // Mostrar todas las empresas
188     document.querySelector("#textoBusquedaEmpresaF2").value = "";
189     pMensajeErrorFiltrarEmpresas.style.display = "none"
190     // Ocultar msj de error
191     displayErrorBusquedaOFF();
192 }
193
194 function displayErrorBusquedaON() {
195     pMensajeErrorFiltrarEmpresas.style.display = "block"
196     document.querySelector("#tableListadoEmpresas").style.display = "none"
197     // Oculta la tabla con todas las empresas
198     document.querySelector("#tableListadoEmpresasFiltrado").style.display = "none"
199     // Muestra unicamente las empresas de la tabla filtrada
200 }
201
202 function displayErrorBusquedaOFF() {
203     pMensajeErrorFiltrarEmpresas.style.display = "none"
204 }
205
206 function adminDisplayVehiculos() {
207     displayMensajeErrorVehiculoOFF();
208     document.querySelector("#tipoNuevoVehiculo").value = "";
209     document.querySelector("#listadoDeEmpresasF2").style.display = "none";
210     document.querySelector("#panelCreacionVehiculosF3").style.display = "block";
211     document.querySelector("#verInfoEstadisticaF4").style.display = "none";
212     crearListaDeVehiculos();
213 }
214
215 function displayMensajeErrorVehiculoON(mensaje){
216     pMensajeErrorIngresarVehiculo.innerHTML = mensaje;
217     pMensajeErrorIngresarVehiculo.style.display = "block";
218 }
219
220 function displayMensajeErrorVehiculoOFF(){

```

```

215     pMensajeErrorIngresarVehiculo.style.display = "none";
216 }
217
218 function adminDisplayEstadistica() {
219     calcularInfoEstadisticaAdmin();
220     // Actualizar la tabla con informacion
221     document.querySelector("#listadoDeEmpresasF2").style.display = "none";
222     document.querySelector("#panelCreacionVehiculosF3").style.display = "none"
223     document.querySelector("#verInfoEstadisticaF4").style.display = "block"
224 }
225
226 function adminDisplayEmpresasFiltradas(){
227     let criterioDeBusqueda = document.querySelector("#textoBusquedaEmpresaF2").value;
228     // Leer del HTML lo escrito en el input field
229     let arrayEmpresas = buscadorEmpresas(criterioDeBusqueda)
230
231     if (arrayEmpresas == "Error"){
232         displayErrorBusquedaON();
233     } else {
234         displayErrorBusquedaOFF();
235         document.querySelector("#tableListadoEmpresas").style.display = "none"
236         // Oculta la tabla con todas las empresas
237         crearListaDeEmpresasFiltrado();
238         // Genera la tabla con los criterios de busqueda
239         document.querySelector("#tableListadoEmpresasFiltrado").style.display =
240 "table" // Muestra unicamente las empresas de la tabla filtrada
241     }
242 }
243
244 function personaDisplaySolicitudEnvio() {
245     displayErrorSolicitudEnvioOFF();
246     displaySuccessSolicitudEnvioOFF();
247     displayBorradoCamposSolicitudEnvioPersona();
248     updateSelectVehiculos("#solicitudEnvioVehiculo");
249     document.querySelector("#solicitudEnvioF6").style.display = "block";
250     document.querySelector("#listadoEnviosF7").style.display = "none";
251
252 }
253
254 function displayBorradoCamposSolicitudEnvioPersona() {
255     document.querySelector("#solicitudEnvioVehiculo").value = 0;
256     document.querySelector("#solicitudEnvioDistancia").value = "";
257     document.querySelector("#solicitudEnvioDescripcion").value = "";
258     document.querySelector("#solicitudEnvioFoto").value = "";
259 }
260
261 function displayErrorSolicitudEnvioON(mensaje){
262     // Display el error correspondiente en el panel de persona para ingresar
263     pedido
264     let mensajeError = document.querySelector("#pErrorSolicitudEnvio")
265     mensajeError.style.display = "block";
266     mensajeError.innerHTML = mensaje;
267 }
268
269 function displayErrorSolicitudEnvioOFF(){
270     document.querySelector("#pErrorSolicitudEnvio").style.display = "none";
271 }
272
273 function displaySuccessSolicitudEnvioON(){

```

```

267     displayBorradoCamposSolicitudEnvioPersona()
268     document.querySelector("#pSuccessSolicitudEnvio").style.display = "block"
269 }
270
271 function displaySuccessSolicitudEnvioOFF(){
272     document.querySelector("#pSuccessSolicitudEnvio").style.display = "none"
273 }
274
275 function personaDisplayListadoEnvio(){
276     document.querySelector("#solicitudEnvioF6").style.display = "none";
277     displayBorradoCamposSolicitudEnvioPersona()
278
279     document.querySelector("#listadoEnviosF7").style.display = "block";
280
281 }
282
283 function personaDisplayEstadistica(){
284     document.querySelector("#solicitudEnvioF6").style.display = "none";
285     document.querySelector("#solicitudEnvioVehiculo").value = 0;
286     document.querySelector("#solicitudEnvioDistancia").value = "";
287     document.querySelector("#solicitudEnvioDescripcion").value = "";
288     document.querySelector("#solicitudEnvioFoto").value = "";
289     document.querySelector("#pErroresSolicitudEnvio").innerHTML = "";
290     document.querySelector("#listadoEnviosF7").style.display = "none";
291     document.querySelector("#infoEstadisticaUsuario").style.display = "block";
292 }
293
294 function displayMensajeErrorSolicitudesPendientesON() {
295 }
296
297 function displayMensajeErrorSolicitudesEnTransitoYFinalizadosON(mensaje) {
298     document.querySelector("#pErrorListadoEnviosTomadosF9").style.display = "block";
299     document.querySelector("#pErrorListadoEnviosTomadosF9").innerHTML = mensaje;
300     document.querySelector("#tablaEnviosTomadosF9").style.display = "none";
301 }
302 function displayMensajeErrorSolicitudesEnTransitoYFinalizadosOFF() {
303     document.querySelector("#pErrorListadoEnviosTomadosF9").style.display = "block";
304     document.querySelector("#pErrorListadoEnviosTomadosF9").innerHTML = '';
305     document.querySelector("#tablaEnviosTomadosF9").style.display = "table";
306 }
307
308 function empresaDisplayListadoEnvio() {
309     document.querySelector("#listadoEnviosPendientesF8").style.display = "block";
310     document.querySelector("#listadoEnviosTomadosF9").style.display = "block";
311     document.querySelector("#infoEstadisticaEmpresaMejoresUsuarios").style.display =
"none";
312     document.querySelector("#infoEstadisticaEmpresaCantidadPorEstado").style.display
= "none";
313     crearListaDeSolicitudesPendientesEmpresa();
314     // Actualizar el listado de pedidos pendientes especificos
315     crearListadoDeSolicitudesTomadasEmpresa();
316     // Actualizar listado de solicitudes en transito + finalizadas
317     document.querySelector("#pErrorEstadisticaCantPorEstadoEmpresa").style.display =
"none";
318 }
319
320 function empresaDisplayEstadisticaMejoresUsuarios(mensaje) {
321     document.querySelector("#listadoEnviosPendientesF8").style.display = "none";
322     document.querySelector("#listadoEnviosTomadosF9").style.display = "none";

```

```
321     document.querySelector("#infoEstadisticaEmpresaMejoresUsuarios").style.display =
"block";
322     document.querySelector("#pInfoEstadisticaEmpresaMejoresUsuarios").innerHTML =
mensaje;
323 }
324
325 function empresaDisplayEstadisticaEnviosPorEstado() {
326     crearTablaDeEnviosPorEstadoEmpresa();
327     document.querySelector("#listadoEnviosPendientesF8").style.display = "none";
328     document.querySelector("#listadoEnviosTomadosF9").style.display = "none";
329     document.querySelector("#infoEstadisticaEmpresaCantidadPorEstado").style.display
= "block";
330 }
331
332 function displayErrorEnviosPorEstadoEmpresaON(mensaje) {
333     document.querySelector("#pErrorEstadisticaCantPorEstadoEmpresa").style.display =
"block";
334     document.querySelector("#pErrorEstadisticaCantPorEstadoEmpresa").innerHTML =
mensaje;
335     document.querySelector("#pSuccessEstadisticaCantPorEstadoEmpresa").style.display
= "none";
336 }
337
338 function displaySuccessEnviosPorEstadoEmpresaON(mensaje) {
339     document.querySelector("#pSuccessEstadisticaCantPorEstadoEmpresa").style.display
= "block";
340     document.querySelector("#pSuccessEstadisticaCantPorEstadoEmpresa").innerHTML =
mensaje;
341     document.querySelector("#pErrorEstadisticaCantPorEstadoEmpresa").style.display =
"none";
342 }
```

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8
9     <!-- ESTILOS -->
10    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
11        integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
12    <link rel="stylesheet" href="../css/style.css">
13    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.3.0/font/bootstrap-icons.css">
14
15    <!-- ICONO DE PESTAÑA -->
16    <link rel="icon" href="../img/icono2.png" type="image/x-icon">
17
18    <!-- TITULO -->
19    <title>Obligatorio Programación 1</title>
20 </head>
21
22 <body>
23     <nav class="navbar navbar-dark bg-dark">
24         <a class="navbar-brand" href="#">Obligatorio Programación 1</a>
25         <form class="form-inline">
26             <button class="btn btn-success boton" type="button"
id="btnDisplayLogIn">Iniciar Sesión</button>
27             <button class="btn btn-success boton" type="button"
id="btnDisplayRegistrar">Registrarse</button>
28             <button class="btn btn-danger boton" type="button"
id="btnDisplayCerrarSesion" display="hidden">Cerrar Sesion</button>
29         </form>
30     </nav>
31
32     <div id="panelAdmin" class="container">
33         <h2>Panel Administrador</h2>
34         <div id="adminPanelNav">
35             <button class="btn btn-primary" id="btnAdminListarEmpresas">Listado de
Empresas</button>
36             <button class="btn btn-primary" id="btnAdminCrearVehiculos">Listado de
vehiculos</button>
37             <button class="btn btn-primary" id="btnAdminEstadistica">Visualizar
informacion estadistica</button>
38         </div>
39     </div>
40
41     <div id="panelEmpresa" class="container">
42         <h2>Panel Empresa</h2>
43         <button class="btn btn-primary" id="btnEmpresaListadoEnvios">Listado de
envios</button>
44         <button class="btn btn-primary" id="btnEmpresaEstadistica">Visualizar
informacion estadistica</button>
45     </div>
46
47     <div id="panelPersona" class="container">

```

```

48     <h2>Panel Persona</h2>
49     <button class="btn btn-primary"
id="btnPersonaRealizarSolicitudEnvio">Realizar solicitud de envío</button>
50     <button class="btn btn-primary" id="btnPersonaListadoEnvios">Listado de
envíos</button>
51     </div>
52
53
54     <div id="panelLogInF1" class="container">
55         <h2>Ingresar a la aplicación</h2>
56         <label for="ingresoUsuario" class="control-label">Usuario: </label>
57         <input type="text" placeholder="Nombre de usuario" class="form-control"
id="ingresoUsuario">
58         <br><br>
59         <label for="ingresoPassword" class="control-label">Contraseña: </label>
60         <input type="password" placeholder="Ingresar contraseña..."
id="ingresoPassword" class="form-control">
61         <br><br>
62         <button id="btnLogIn" class="btn btn-secondary">Log In</button>
63         <p id="mensajeLogIn" class="errorMsj"></p>
64         <p id="mensajeLogInNuevoUsuario" class="welcomeMsj"></p>
65         <hr>
66     </div>
67
68
69
70     <!-- FUNCIONALIDAD 2 - HABILITAR DESHABILITAR EMPRESAS -->
71     <div id="listadoDeEmpresasF2" class="container">
72         <h2>Listado de empresas</h2>
73         <div id="listadoEmpresas">
74             <label for="textoBusquedaEmpresaF2" id="labBusquedaEmpresas"
class="control-label">Buscar por razón
75                 social:</label>
76             <label for="textoBusquedaEmpresaF2"> <i class="bi bi-search"></i>
</label>
77             <input type="text" id="textoBusquedaEmpresaF2" placeholder="Buscar..."
class="form-control-busqueda">
78             <br><br>
79             <input type="button" id="btnMostrarEmpresasF2" value="Mostrar filtrado"
class="btn btn-secondary">
80             <table class="table" id="tableListadoEmpresas">
81                 </table>
82             <table class="table" id="tableListadoEmpresasFiltrado">
83                 </table>
84             <p id="mensajeErrorFiltrarEmpresas" class="errorMsj">No hay resultados
que coincidan con su búsqueda.</p>
85         </div>
86         <hr>
87     </div>
88
89     <div id="panelCreacionVehiculosF3" class="container">
90         <h2>Listado de vehiculos</h2>
91         <div>
92             <table class="table" id="listadoVehiculos">
93
94             </table>
95         </div>
96         <div id="agregarVehiculos">
97             <h3>Agregar Vehiculos</h3>

```

```

98         <label for="tipoVehiculo" class="control-label">Tipo de vehiculo:
</label>
99         <input type="text" placeholder="Vehiculo" id="tipoNuevoVehiculo"
class="form-control">
100         <br><br>
101         <button class="btn btn-secondary" id="btnIngresarVehiculo">Ingresar
vehiculo al sistema</button>
102         <p id="mensajeErrorVehiculos" class="errorMsj"></p>
103     </div>
104     <hr>
105 </div>
106
107 <div id="verInfoEstadisticaF4" class="container">
108     <h2>Visualizar informacion estadistica</h2>
109     <br><br>
110     <h3>Informacion de empresas: Admin</h3>
111     <table class="table">
112         <thead>
113             <th>Razón Social</th>
114             <th>Km totales</th>
115             <th>Pedidos finalizados</th>
116         </thead>
117         <tbody id="tablaInfoEstadisticaKMAAdminBody">
118
119         </tbody>
120     </table>
121     <br><br><hr>
122 </div>
123
124 <div id="panelRegistroF5" class="container">
125     <h2>Registrarse en la aplicacion</h2>
126     <br>
127     <label for="tipoDeCuenta">Seleccione el tipo de cuenta: </label>
128     <select name="" id="tipoDeCuenta" class="form-select form-select-lg mb-3">
129         <option disabled selected value=0>Seleccione un tipo</option>
130         <option value="Empresa">Empresa</option>
131         <option value="Persona">Persona</option>
132     </select>
133     <button id="btnTipoDeCuenta" class="btn btn-secondary">Seleccionar</button>
134     <p class="errorMsj" id="pErrorSelectTipocuenta"></p>
135     <br><br>
136     <div id="registroUsuario">
137         <h3>Usuario: Persona</h3>
138         <br>
139         <label for="registroAliasUsuarioPersonaF5">Nombre de Usuario: </label>
140         <input type="text" id="registroAliasUsuarioPersonaF5"
placeholder="jfernandez">
141         <br><br>
142         <label for="registroPassUsuarioPersonaF5">Contraseña: </label>
143         <input type="password" id="registroPassUsuarioPersonaF5"
placeholder="Ab12345">
144         <br><br>
145         <label for="registroCIUsuarioPersonaF5">CI: </label>
146         <input type="text" id="registroCIUsuarioPersonaF5"
placeholder="12345678">
147         <br><br>
148         <label for="registroNombreUsuarioPersonaF5">Nombre: </label>
149         <input type="text" id="registroNombreUsuarioPersonaF5"
placeholder="Juan">
150         <br><br>

```



```

151         <label for="registroApellidoUsuarioPersonaF5">Apellido: </label>
152         <input type="text" id="registroApellidoUsuarioPersonaF5"
placeholder="Fernandez">
153         <br><br>
154         <button class=" btn btn-secondary"
id="btnRegistrarPersona">Registrar</button>
155         <p id="pParaDesplegarErroresPersona" class="errorMsj"></p>
156     </div>
157     <div id="registroEmpresa">
158         <h3>Usuario: Empresa</h3>
159         <br>
160         <label for="registroAliasEmpresaF5">Nombre de usuario: </label>
161         <input type="text" id="registroAliasEmpresaF5" placeholder="nombre123">
162         <br><br>
163         <label for="registroNombreEmpresaF5">Nombre de fantasia: </label>
164         <input type="text" id="registroNombreEmpresaF5" placeholder="Empresa SA">
165         <br><br>
166         <label for="registroRUTEmpresaF5">RUT:</label>
167         <input type="number" id="registroRUTEmpresaF5" placeholder="123456">
168         <br><br>
169         <label for="registroRazonSocialEmpresaF5">Razon Social:</label>
170         <input type="text" id="registroRazonSocialEmpresaF5" placeholder="Razon
social registrada">
171         <br><br>
172         <label for="registroPassEmpresaF5">Contraseña: </label>
173         <input type="password" id="registroPassEmpresaF5" placeholder="Ab12345">
174         <br><br>
175         <label for="registroVehiculoEmpresa">Tipo de vehiculo: </label>
176         <select id="registroVehiculoEmpresa">
177             </select>
178         <br><br>
179         <button class=" btn btn-secondary"
id="btnRegistrarEmpresa">Registrar</button>
180         <p id="pParaDesplegarErroresEmpresa" class="errorMsj"></p>
181     </div>
182     <hr>
183 </div>
184
185     <div id="solicitudEnvioF6" class="container">
186         <h2>Realizar solicitud de envío</h2>
187         <label for="solicitudEnvioVehiculo">Tipo de vehiculo: </label>
188         <select id="solicitudEnvioVehiculo">
189             </select>
190         <br><br>
191         <label for="solicitudEnvioDistancia">Distancia: </label>
192         <input type="number" id="solicitudEnvioDistancia" placeholder="Distancia en
Km">
193         <br><br>
194         <label for="solicitudEnvioDescripcion">Descripción: </label>
195         <input type="text" placeholder="Breve descripcion"
id="solicitudEnvioDescripcion">
196         <br><br>
197         <label for="solicitudEnvioFoto">Imagen:</label>
198         <input type="file" placeholder="URL" id="solicitudEnvioFoto">
199         <br><br>
200         <button id="btnEnviarSolicitudEnvio6" class="btn btn-secondary">Ingresar
solicitud</button>
201         <p id="pErrorSolicitudEnvio" class="errorMsj"></p>
202         <p id="pSuccessSolicitudEnvio" class="successMsj">Envio ingresado.</p>
203     </div>

```



```

204
205 <div id="listadoEnviosF7" class="container">
206     <h2>Información estadística de envíos del usuario</h2>
207     <div id="infoEstadistica"></div>
208     <h2>Listado de envíos del cliente</h2>
209     <table class="table" id="listadoDeEnviosPersona">
210         </table>
211 </div>
212
213 <div id="listadoEnviosPendientesF8" class="container">
214     <h2>Solicitudes pendientes</h2>
215     <table class="table" >
216         <thead>
217             <tr>
218                 <th>Foto</th>
219                 <th>Vehiculo</th>
220                 <th>Distancia (Km)</th>
221                 <th>Estado</th>
222                 <th>Nombre del solicitante</th>
223                 <th>Apellido del solicitante</th>
224                 <th>Aceptar envío</th>
225             </tr>
226         </thead>
227         <tbody id="bodyListadoEnviosPendientesEmpresa">
228             </tbody>
229     </table>
230 </div>
231
232 <div id="listadoEnviosTomadosF9" class="container">
233     <h2>Solicitudes tomadas</h2>
234     <table class="table" id="tablaEnviosTomadosF9">
235         </table>
236     <p id="pErrorListadoEnviosTomadosF9"></p>
237
238 </div>
239
240 <div id="infoEstadisticaEmpresaMejoresUsuarios" class="container">
241     <h2>Mejores usuarios</h2>
242     <p id="pInfoEstadisticaEmpresaMejoresUsuarios"></p>
243 </div>
244 <div id="infoEstadisticaEmpresaCantidadPorEstado" class="container">
245 </div>
246
247 <!-- ----- -->
248 <!-- LINKEO CON ARCHIVOS JS -->
249 <!-- CLASES -->
250 <script src="../js/constructores.js"></script>
251
252 <!-- ARCHIVO PRINCIPAL -->
253 <script src="../js/display.js"></script>
254 <script src="../js/codigo.js"></script>
255 </body>
256
257 </html>
258
259

```

```
1 * {
2     padding: 0;
3     margin: 0;
4     box-sizing: border-box;
5 }
6
7 div{
8     margin-bottom: 50px;
9 }
10
11 .control-label{
12     display: block;
13     padding-bottom: 5px;
14 }
15
16 .form-control {
17     font-size: 1rem;
18     line-height: 1.3;
19     display: block;
20     width: 20%;
21     padding: .375rem .75rem;
22     transition: border-color .15s ease-in-out,box-shadow .15s ease-in-out;
23     border: 1px solid #cfcfcf;
24     border-radius: 3px;
25 }
26
27 .form-control-busqueda {
28     font-size: 1rem;
29     line-height: 1.3;
30     display: inline-block;
31     width: 20%;
32     padding: .375rem .75rem;
33     transition: border-color .15s ease-in-out,box-shadow .15s ease-in-out;
34     border: 1px solid #cfcfcf;
35     border-radius: 3px;
36 }
37
38 select {
39     margin-bottom: 5px;
40 }
41
42 /* #estadoSolicitudEmpresaF4 {
43
44 } */
45
46 #labBusquedaEmpresas {
47     margin-left: 22px;
48     font-weight: bold;
49 }
50
51 h2 {
52     margin: 25px auto;
53 }
54
55
56 .form-inline{
57     margin-left:70%;
58 }
59
60 .form-inline{
```

```
61     padding-left: 15px;
62     padding-right: 15px;
63     margin-left: 15px;
64 }
65
66 .navbar-brand{
67     color: white;
68 }
69
70 .navbar {
71     justify-content: space-around;
72 }
73
74 a {
75     text-decoration: none;
76 }
77 .errorMsj {
78     margin-top: 25px;
79     color: red;
80     font-size: 21px;
81 }
82 .welcomeMsj {
83     margin-top: 25px;
84     color: rgb(0, 134, 29);
85     font-size: 21px;
86 }
87
88 .successMsj {
89     margin-top: 25px;
90     color: rgb(0, 134, 29);
91     font-size: 21px;
92 }
93
94 table img {
95     height: 120px;
96     width: 200px
97 }
98
99 .btn {
100     margin-bottom: 20px;
101 }
102
103 .boton {
104     margin-top: 20px;
105 }
```