


Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO PROGRAMACIÓN 1

		
Felipe Caillabet – 280850	Luca Mirenda – 254413	Andrés Mayes – 230729

Clase M1C - Grupo 4

Docente: Bruno Díaz

Analista Programador – Analista en Tecnologías de la Información

Fecha de entrega: 15-11-2021

Índice

Documento de análisis	4
Descripción general del problema a resolver	4
Tipos de usuario del sistema	4
Listado de funcionalidades	4
Detalle de Funcionalidades	5
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
Precarga de datos	20
Array Vehículos	20
Array de usuarios (length: 10)	20
Usuarios de tipo Persona (4 usuarios)	20
Usuarios de tipo Empresa (5 usuarios)	21
Usuario de tipo Administrador (1 usuario)	22
Array de envíos (length: 5)	22
Estructura de datos	23
Variables globales	25
Planilla de testing	26

Funcionalidad 01 - Ingresar a la aplicación	26
Funcionalidad 03 - Crear tipos de vehículos	29
Funcionalidad 04 - Visualizar información estadística	30
Funcionalidad 05 - Registrarse en la aplicación	32
Funcionalidad 06 - Realizar solicitud de envío	35
Código	38

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:

Q Buscar...

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 búsqueda

Buscar por razón social:

Q Nombre empresa 3

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:

Ingresar vehiculo al sistema

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 fantasía:

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 invalido: “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)

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

Fantasía	“Empresa Cuatro”	“Empresa Cinco”
Habilitación	False	False
Password	“Empresa04”	“Empresa05”
Pedidos	[]	[]
Razón social	“Empresa Cero Cuatro”	“Empresa Cero Cinco”
RUT	104	105
Tipo	“Empresa”	“Empresa”
Username	“Empresa4”	“Empresa5”
Vehículo	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
UsuarioPersona	username	String. Único entre todos los tipos de usuario existentes. Identifica al usuario.
	pass	String. Contraseña asociada al usuario. Contiene una mayúscula, una minúscula y un número.
	cedula	Numérica. La cédula de identidad del usuario.
	nombre	String. Nombre propio del usuario.
	apellido	String. Apellido del usuario.
	pedidos	Array indexado. Contiene los id de los envíos realizados por el usuario.
	tipo	String. Tipo de usuario. Valor: "Persona".
UsuarioEmpresa	username	String. Único entre todos los tipos de usuario existentes. Identifica al usuario.
	pass	String. Contraseña asociada al usuario. Contiene una mayúscula, una minúscula y un número.
	fantasia	String. Nombre de fantasía de la empresa.
	rut	Numérica. RUT asociado a la empresa. Único entre todos los usuarios.
	razonSocial	String. Razón Social asociada a la empresa. Único entre todos los usuarios.
	vehiculo	Numérica. Seleccionada desde un combo desplegable. Cada número está asociado a un tipo de vehículo distinto. (Ej: 1001 para Moto).
	habilitacion	Booleano. Determina si una empresa está habilitada (true) ó si está deshabilitada (false). Únicamente pueden acceder a la aplicación empresas habilitadas.
	km	Numérica. Cantidad de Km de pedidos finalizados.
	pedidos	Array indexado. Contiene los id de los envíos aceptados por la empresa.
	tipo	String. Tipo de usuario. Valor: "Empresa".

UsuarioAdmin	username	String. Único entre todos los tipos de usuario existentes. Valor: "Admin"
	pass	String. Contraseña asociada al usuario administrador. Contiene una mayúscula, una minúscula y un número. Valor: "Admin01".
	tipo	String. Tipo de usuario. Valor: "Administrador".
Envio	vehiculo	Numérica. ID del vehículo solicitado.
	distancia	Numérica. Distancia del envío en Km.
	descripcion	String. Breve descripción del envío.
	img	String. Ruta relativa a la imagen cargada. La imagen se carga desde un input tipo "file", desde una carpeta con imágenes precargadas.
	empresa	String. Razón Social de la empresa asociada al envío.
	estado	String. Corresponde al estado actual del envío. Valores posibles: "Pendiente", "En tránsito" y "Finalizado".
	persona	String. Nombre de usuario de la persona que realizó el envío.
	id	Numérica. ID del envío. Único entre todos los envíos. Autonumérico secuencial, se incrementa en 1 cada vez que se realiza una solicitud de envío.
Vehiculo	tipo	String. Único para cada vehículo.
	id	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
usuarios	Array indexado.	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.
vehiculos	Array indexado.	Almacena todos los vehículos. Se inicia con los vehículos precargados. Cuando se genera un nuevo vehículo se agrega en la última posición.
codigoVehiculo	Numérico. Entero.	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.
envios	Array indexado.	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.
idEnvio	Numérico. Entero.	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.
usuarioLoggeado	Array asociativo.	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

	Funcionalidad 04 - Visualizar información estadística			
	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	Esperado vs Obtenido
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: "amayes" 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 <!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="tablaInfoEstadisticaKAdminBody">
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 // Archivo: constructores.js//
2 class UsuarioPersona{
3     constructor(_username, _pass, _CI, _nombre, _apellido, _pedidos){
4         this.username = _username;
5         this.pass = _pass;
6         this.cedula = _CI;
7         this.nombre = _nombre;
8         this.apellido = _apellido;
9         this.pedidos = [];
10        this.tipo = "Persona";
11    }
12 }
13
14 class UsuarioEmpresa {
15     constructor(_username, _pass, _fantasia, _rut, _razonSocial, _vehiculo,
16     _habilitacion){
17         this.username = _username;
18         this.pass = _pass;
19         this.fantasia = _fantasia;
20         this.rut = _rut;
21         this.razonSocial = _razonSocial;
22         this.vehiculo = _vehiculo;
23         this.habilitacion = _habilitacion; // True = empresa habilitada. False =
24         empresa deshabilitada.
25         this.pedidos = [];
26         this.tipo = "Empresa"
27     }
28 }
29
30 class UsuarioAdmin {
31     constructor(_username, _pass){
32         this.username = _username;
33         this.pass = _pass;
34         this.tipo = "Administrador"
35     }
36 }
37
38 class Envio {
39     constructor(_vehiculo, _distancia, _descripcion, _img, _empresa, _estado,
40     _persona){
41         this.vehiculo = _vehiculo;
42         this.distancia = _distancia;
43         this.descripcion = _descripcion;
44         this.img = _img;
45         this.empresa = _empresa; // Corresponde a la empresa que toma el
46         pedido
47         this.estado = _estado; // Estado corresponde a: "Pendiente", "En
48         tránsito", "Finalizado"
49         this.persona = _persona;
50         this.id = idEnvio;
51         idEnvio++;
52     }
53 }
54
55 class Vehiculo {
56     constructor(_tipo){
57         this.tipo = _tipo;
58         this.id = codigoVehiculo;
59         codigoVehiculo++
60     }
61 }

```



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

```

51     usuarios.push(new UsuarioPersona("amayes", 456, 4928922, "Andres", "Mayes",
52     []));
53     usuarios.push(new UsuarioPersona("fcaillabet", 789, 87654321, "Felipe",
54     "Caillabet", []));
55     usuarios.push(new UsuarioPersona("lmirenda", 123, 14725836, "Luca", "Mirenda",
56     []));
57 }
58
59 function precargaDeUsuariosEmpresa() {
60     usuarios.push(new UsuarioEmpresa("Empresa1", "Empresa01", "Empresa Uno", 101,
61     "Empresa Cero Uno", 1000, true));
62     usuarios.push(new UsuarioEmpresa("Empresa2", "Empresa02", "Empresa Dos", 102,
63     "Empresa Cero Dos", 1001, true));
64     usuarios.push(new UsuarioEmpresa("Empresa3", "Empresa03", "Empresa Tres", 103,
65     "Empresa Cero Tres", 1001, true));
66     usuarios.push(new UsuarioEmpresa("Empresa4", "Empresa04", "Empresa Cuatro", 104,
67     "Empresa Cero Cuatro", 1002, false));
68     usuarios.push(new UsuarioEmpresa("Empresa5", "Empresa05", "Empresa Cinco", 105,
69     "Empresa Cero Cinco", 1002, false));
70 }
71
72 function precargaDeUsuariosAdmin() {
73     usuarios.push(new UsuarioAdmin("Admin", "Admin01"));
74 }
75
76 function precargaDeEnvios() {
77     let pedido = new Envio(1000, 5, "Envio de supermercado", "envioSuper.jpg", null,
78     "Pendiente", usuarios[0].username);
79     envios.push(pedido);
80     usuarios[0].pedidos.push(pedido.id);
81
82     pedido = new Envio(1001, 10, "Motosierra", "motosierra.jpg", "Empresa Cero Dos",
83     "En tránsito", usuarios[1].username);
84     envios.push(pedido);
85     usuarios[1].pedidos.push(pedido.id);
86     usuarios[5].pedidos.push(pedido.id);
87
88     pedido = new Envio(1000, 5, "Mudanza de muebles", "mudanza.jpg", "Empresa Cero
89     Uno", "Finalizado", usuarios[0].username);
90     envios.push(pedido);
91     usuarios[0].pedidos.push(pedido.id);
92     usuarios[4].pedidos.push(pedido.id);
93
94     pedido = new Envio(1000, 5, "Mudanza de muebles", "mudanza.jpg", "Empresa Cero
95     Uno", "En tránsito", usuarios[2].username);
96     envios.push(pedido);
97     usuarios[2].pedidos.push(pedido.id);
98     usuarios[4].pedidos.push(pedido.id);
99
100    pedido = new Envio(1000, 5, "Mudanza de muebles", "mudanza.jpg", "Empresa Cero
101    Uno", "Finalizado", usuarios[2].username);
102    envios.push(pedido);
103    usuarios[2].pedidos.push(pedido.id);
104    usuarios[4].pedidos.push(pedido.id);
105 }
106
107 function registroPersona() { // Registra un nuevo usuario en el sistema
108     // Ingreso de datos

```

```

97     let aliasIngresado =
document.querySelector("#registroAliasUsuarioPersonaF5").value.trim();
98     let passIngresado =
document.querySelector("#registroPassUsuarioPersonaF5").value.trim();
99     let ciIngresado =
parseInt(document.querySelector("#registroCIUsuarioPersonaF5").value);
100     let nombreIngresado =
document.querySelector("#registroNombreUsuarioPersonaF5").value.trim();
101     let apellidoIngresado =
document.querySelector("#registroApellidoUsuarioPersonaF5").value.trim();
102
103     // Validacion de datos
104     let noEsNumero = isNaN(ciIngresado); // Booleano que devuelve true si la
cedula ingresada tiene algo distinto a un numero
105     if (aliasIngresado && passIngresado && ciIngresado != '' && nombreIngresado &&
apellidoIngresado) { // Revisa que se hayan ingresado datos en todos los
campos
106         if (!validarPassword(passIngresado)) {
107             document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "La
contraseña debe tener por lo menos una mayúscula, una minúscula y un número."
108         } else if (existeUsuarioPorUsuario(aliasIngresado)) {
109             document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "Ya
existe un usuario con ese alias.";
110         } else if (noEsNumero) {
111             document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "La
cédula debe contener solo números";
112         } else {
113             // Si paso todas las restricciones registra al usuario
114             registrarUsuarioPersona(aliasIngresado, passIngresado, ciIngresado,
nombreIngresado, apellidoIngresado);
115             displayBorradoCamposRegistroPersona();
116             toggleLogIn();
117             document.querySelector("#mensajeLogInNuevoUsuario").innerHTML =
`Bienviendo nuevo usuario ${aliasIngresado}. Puedes logearte en la app.`
118         }
119     } else {
120         document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "Todos
los campos son obligatorios.";
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
133 return esValida
134 }
135
136 function passwordTieneMayuscula(password) { //Verificar si un texto
contiene mayúsculas
137     let tieneMayuscula = false;
138

```



```

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

```

```

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

```

```

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

```

```

289     document.querySelector("#pParaDesplegarErroresEmpresa").innerHTML = "Todos
los campos son obligatorios.";
290 }
291 }
292
293 function existeRazonSocialPorUsuarioEmpresa(razonSocial) {           // Checkear
entre las empresas registradas si ya existe la razón social ingresada. SI existe
devuelve 'true', sino existe 'false'
294     let existe = false;
295     let i = 0;
296     while (!existe && i < usuarios.length) {
297         if (razonSocial === usuarios[i].razonSocial) {
298             existe = true;
299         }
300         i++;
301     }
302     return existe;
303 }
304
305 function existeRUTPorUsuarioEmpresa(rut) {           // Checkear entre las empresas
registradas si ya existe el rut ingresado. Si existe devuelve 'true', sino existe
'false'
306     let existe = false;
307     let i = 0;
308     while (!existe && i < usuarios.length) {
309         if (rut === usuarios[i].rut) {
310             existe = true;
311         }
312         i++;
313     }
314     return existe;
315 }
316
317
318 function registrarUsuarioEmpresa(alias, pass, fantasia, rut, razonSocial, vehiculo)
{
    //Agregar una nueva empresa al sistema
319     let nuevoUsuarioEmpresa = new UsuarioEmpresa(alias, pass, fantasia, rut,
razonSocial, vehiculo, false, 0);
320     usuarios.push(nuevoUsuarioEmpresa);
321 }
322
323 function seleccionarTipoDeCuentaARegistrar() {
324     let tipoDeCuenta = document.querySelector("#tipoDeCuenta").value;
325     displayRegistrarEmpresaPersona(tipoDeCuenta);
326 }
327
328 function logIn() {           // Controla los datos ingresados en el nombre de usuario
y la contraseña. En caso de que sean correctos entra a la aplicacion
329     let user = document.querySelector("#ingresoUsuario").value;
330     let pass = document.querySelector("#ingresoPassword").value
331     let i = 0;
332     let encontrado = false;
333     let mensaje = document.querySelector("#mensajeLogIn");
334
335
336     while (i < usuarios.length && !encontrado) {           // Recorremos el array
de usuarios para buscar si los datos ingresados coinciden con algun usuario ya
registrado
337         if (usuarios[i].pass == pass && (usuarios[i].username).toUpperCase() ===
(user).toUpperCase()) {

```

```

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

```

```

390     table.innerHTML = tablaEmpresas;
391     activarBotonesCambioDeEstado();
392 }
393
394 function mostrarTextoHabilitarDeshabilitar(usuarioEmpresa) {          // Leer el
estado (habilitado o deshabilitado) de la empresa. Devuelve el texto "habilitar" o
"deshabilitar"
395     let texto = "";
396     if (usuarioEmpresa.habilitacion){
397         texto = "Deshabilitar";
398     } else {
399         texto = "Habilitar";
400     }
401     return texto
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 function buscadorEmpresas(input){
435     // 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

```

```

441     if(empresasEncontradasPorRZ){
442         // Buscamos primero por Razon Social
443         return empresasEncontradasPorRZ; // En caso
444     } de encontrar generamos el array con la empresa encontrada
445     else if (empresasEncontradasPorNF){
446         // En caso de no encontrar nada por RZ buscamos por NF
447         return empresasEncontradasPorNF; // En caso
448     } de encontrar generamos el array con la empresa encontrada
449     } else {
450         // En caso de no encontrar resultados por ningun metodo desplegar msj de
451         error
452         return "Error";
453     }
454 }
455
456 function crearListaDeEmpresasFiltrado() {
457     let criterioDeBusqueda =
458     document.querySelector("#textoBusquedaEmpresaF2").value; // Leer del HTML
459     lo escrito en el campo de busqueda
460     let arrayEmpresas = buscadorEmpresas(criterioDeBusqueda);
461     if (arrayEmpresas != 'Error'){
462         let table = document.querySelector("#tableListadoEmpresasFiltrado");
463         table.innerHTML = ` <thead>
464             <tr>
465                 <th>Nombre de fantasia</th>
466                 <th>RUT</th>
467                 <th>Razon social</th>
468                 <th>Estado</th>
469                 <th>Habilitar/Deshabilitar</th>
470             </tr>
471             <tbody>
472
473         for (let i = 0; i < arrayEmpresas.length; i++) {
474             let estadoParaModificar =
475             mostrarTextoHabilitarDeshabilitar(arrayEmpresas[i])
476             let estadoActual = leerEstado(arrayEmpresas[i]);
477
478             table.innerHTML += ` <tr>
479                 <td>${arrayEmpresas[i].fantasia}</td>
480                 <td>${arrayEmpresas[i].rut}</td>
481                 <td>${arrayEmpresas[i].razonSocial}</td>
482                 <td>${estadoActual}</td>
483                 <td><button class="btn btn-warning btnCambioEstado"
484                 btnempresaUsername="${arrayEmpresas[i].username}">${estadoParaModificar}</button>
485                 </td>
486             </tr>`
487
488         }
489         table.innerHTML += `</tbody>`
490     }
491     activarBotonesCambioDeEstado();
492 }
493
494 function cortarPath(path) { // Corta el fakepath de la ruta de la foto
495     let posDeArranquePath = posicionUltimoSlash(path) + 1;
496 }

```



```

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

```



```

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

```

```

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

```



```

700         <th>ID</th>
701     </tr>
702 </thead>
703 <tbody>
704     `;
705     for(let i = 0; i < vehiculos.length; i++){
706         table.innerHTML += `<tr>
707             <td>${vehiculos[i].tipo}</td>
708             <td>${vehiculos[i].id}</td>
709             </tr>`
710     }
711     table.innerHTML += `</tbody>`
712 }
713
714 // -- Listado solicitudes pendientes para usuario tipo empresa -- //
715
716 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)
717     let arrayPedidos = [];
718     for (let i = 0; i<envios.length; i++){
719         if(envios[i].estado == "Pendiente" && envios[i].vehiculo ==
objEmpresa.vehiculo){
720             arrayPedidos.push(envios[i]);
721         }
722     }
723     return arrayPedidos
724 }
725
726
727 function crearListaDeSolicitudesPendientesEmpresa(){ // Genera la
    tabla con los pedidos que puede aceptar la empresa loggeada
728     let enviosPosibles = buscarPedidosDisponiblesPorVehiculo(usuarioLoggeado);
729     let tableBody = document.querySelector("#bodyListadoEnviosPendientesEmpresa");
730     tableBody.innerHTML = ``;
731
732     if(enviosPosibles.length > 0){
733         for(let i = 0; i< enviosPosibles.length; i++){
734             let j = buscarPosEnvioPorId(enviosPosibles[i].id);
735             let vehiculo = tipoVehiculoPorId(envios[j].vehiculo);
736             let usuarioPersona = objUsuarioPorUsuario(envios[j].persona);
737             tableBody.innerHTML += `<tr>
738                 <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
748         } else {
749             displayMensajeErrorSolicitudesPendientesON()
750         }
751         activarBotonesAceptarSolicitudesPendientes();
752     }
753

```

```

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

```

```

797     i++;
798   }
799 }
800
801 function buscarEnviosPorEmpresa(objEmpresa) { // Recibe como parametro el
objeto empresa y busca sus pedidos asignados
802   let arrayPedidos = [[],[]];
803   for (let i = 0; i < envios.length; i++){
804     if (envios[i].estado == "En tránsito" && envios[i].empresa ==
objEmpresa.razonSocial) {
805       arrayPedidos[0].push(envios[i]);
806     } else if ((envios[i].estado == "Finalizado" && envios[i].empresa ==
objEmpresa.razonSocial)) {
807       arrayPedidos[1].push(envios[i]);
808     }
809   }
810   return arrayPedidos
811 }
812
813 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
814   let table = document.querySelector("#tablaEnviosTomadosF9");
815   let enviosEnTransitoYFinalizados = buscarEnviosPorEmpresa(usuarioLoggeado);
816   let enviosEnTransito = enviosEnTransitoYFinalizados[0];
817   let enviosFinalizados = enviosEnTransitoYFinalizados[1];
818   let vehiculoEmpresa = tipoVehiculoPorId(usuarioLoggeado.vehiculo)
819
820   if (enviosEnTransito.length > 0 || enviosFinalizados.length > 0){
821     displayMensajeErrorSolicitudesEnTransitoYFinalizadosOFF();
822     let tablaPedidosTomados = `<header>
823       <tr>
824         <td><strong>Foto</strong></td>
825         <td><strong>Vehiculo</strong></td>
826         <td><strong>Distancia (Km)</strong></td>
827         <td><strong>Estado</strong></td>
828         <td><strong>Nombre</strong></td>
829         <td><strong>Apellido</strong></td>
830         <td><strong>Finalizar envio</strong>
831       </tr>
832     </header>
833     <body>`
834     for(let i = 0; i < enviosEnTransito.length; i++){
835       let j = buscarPosEnvioPorId(enviosEnTransito[i].id);
836       let usuarioPersona = objUsuarioPorUsuario(envios[j].persona)
837       let vehiculoEnvio = tipoVehiculoPorId(envios[j].vehiculo);
838       if (vehiculoEnvio == vehiculoEmpresa) {
839         tablaPedidosTomados += `<tr>
840           <td></td>
842           <td>${vehiculoEnvio}</td>
843           <td>${envios[j].distancia}</td>
844           <td>${envios[j].estado}</td>
845           <td>${usuarioPersona.nombre}</td>
846           <td>${usuarioPersona.apellido}</td>
847           <td><button class="btn btn-success
848             btnFinalizarPedido" btnEnvioId="${envios[j].id}">Finalizar</button></td>
849         </tr>`
850       }
851     }
852   }
853 }

```

```

849     }
850
851     for(let i = 0; i < enviosFinalizados.length; i++) {
852         let j = buscarPosEnvioPorId(enviosFinalizados[i].id);
853         let usuarioPersona = objUsuarioPorUsuario(envios[j].persona);
854         let vehiculoEnvio = tipoVehiculoPorId(envios[j].vehiculo);
855         if (vehiculoEnvio == vehiculoEmpresa) {
856             tablaPedidosTomados += `<tr>
857                                     <td></td>
858                                     <td>${vehiculoEnvio}</td>
859                                     <td>${envios[j].distancia}</td>
860                                     <td>${envios[j].estado}</td>
861                                     <td>${usuarioPersona.nombre}</td>
862                                     <td>${usuarioPersona.apellido}</td>
863                                     <td><button class="btn btn-dark"
disabled>Finalizado</button></td>
864                                     </tr>`;
865         };
866     }
867     tablaPedidosTomados += `</body>`;
868     table.innerHTML = tablaPedidosTomados;
869 } else {
870     displayMensajeErrorSolicitudesEnTransitoYFinalizadosON("No hay envios
asociados a su empresa.");
871 }
872 activarBotonesFinalizarSolicitudesTransito();
873 }
874
875 function calcularInfoEstadisticaEmpresa() { // Muestra que persona realizó
más pedidos con la empresa loggeada
876     let infoEstadisticaMejoresUsuarios;
877
878     let usuariosConMasEnvios = encontrarUsuariosConMasEnvios();
879
880     if (usuariosConMasEnvios.length > 0) {
881         infoEstadisticaMejoresUsuarios = `La/s personas con mas envios finalizados
es/son: <br>`;
882         for (let i = 0; i < usuariosConMasEnvios.length; i++) {
883             let usuarioActual = usuariosConMasEnvios[i];
884             infoEstadisticaMejoresUsuarios += `${i + 1}. ${usuarioActual.nombre}
${usuarioActual.apellido}<br>`;
885         }
886     } else {
887         infoEstadisticaMejoresUsuarios = "No hay pedidos finalizados asignados a
esta empresa.";
888     }
889
890     empresaDisplayEstadisticaMejoresUsuarios(infoEstadisticaMejoresUsuarios);
891 }
892
893 function encontrarUsuariosConMasEnvios() { // Crea un array con los
usuarios que tengan más pedidos finalizados
894     let cantPedidosFinalizadosPorUsuario = [];
895
896     for (let i = 0; i < usuarios.length; i++) {
897         let usuarioActual = usuarios[i];
898
899         if (usuarioActual.tipo == "Persona") {
900             let contEnviosFinalizados = 0;

```



```

901
902     for (let j = 0; j < usuarioActual.pedidos.length; j++) {
903         let idPedidoActual = usuarioActual.pedidos[j];
904         let pedidoActual = buscarPedidoPorId(idPedidoActual);
905         if (pedidoActual.estado == "Finalizado" && pedidoActual.empresa ==
usuarioLoggeado.razonSocial) {
906             contEnviosFinalizados++
907         }
908     }
909
910     cantPedidosFinalizadosPorUsuario.push(contEnviosFinalizados)
911
912     } else {
913         cantPedidosFinalizadosPorUsuario.push(null)
914     }
915 }
916
917 let mayorCantEnvios = encontrarNumMasGrande(cantPedidosFinalizadosPorUsuario);
918
919 let mejoresUsuarios = [];
920
921 if (mayorCantEnvios != 0) {
922     mejoresUsuarios = armarArrayConMejoresUsuarios(mayorCantEnvios,
cantPedidosFinalizadosPorUsuario);
923 }
924
925 return mejoresUsuarios
926 }
927
928 function buscarPedidoPorId(idABuscar) {           // Recibe como parametro un id y
busca el pedido con ese id, luego devuelve el pedido
929     let pedido;
930     let pedidoEncontrado = false;
931
932     let i = 0
933     while (i < envios.length && !pedidoEncontrado) {
934         let pedidoActual = envios[i];
935         if (pedidoActual.id == idABuscar) {
936             pedido = pedidoActual;
937             pedidoEncontrado = true;
938         }
939         i++
940     }
941
942     return pedido
943 }
944
945 function encontrarNumMasGrande(arrayNumerico) {           // Recibe como parametro un
array n merico y busca el n mero m s grande del array y lo devuelve
946     let numMasGrande = 0;
947
948     for (let i = 0; i < arrayNumerico.length; i++) {
949         let numActual = arrayNumerico[i];
950         if (numActual != null && numActual > numMasGrande ) {
951             numMasGrande = numActual
952         }
953     }
954
955     return numMasGrande
956 }

```



```

957
958 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
959     let mejoresUsuarios = [];
960
961     for (let i = 0; i < pedidosPorUsuario.length; i++) {
962         let numActual = pedidosPorUsuario[i];
963         if (numActual == numMasGrande) {
964             mejoresUsuarios.push(usuarios[i])
965         }
966     }
967
968     return mejoresUsuarios
969 }
970
971
972 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
973     let contDeEnvios = 0;
974
975     if (usuarioLoggeado.pedidos.length > 0) {
976         let arrIdsDeEnviosDeEmpresa = usuarioLoggeado.pedidos;
977
978         for (let i = 0; i < arrIdsDeEnviosDeEmpresa.length; i++) {
979             let idActual = arrIdsDeEnviosDeEmpresa[i]
980             let pedidoActual = buscarPedidoPorId(idActual)
981
982             if (pedidoActual.estado == estadoElegido) {
983                 contDeEnvios++
984             }
985         }
986     }
987
988     return contDeEnvios
989 }
990
991 function crearTablaDeEnviosPorEstadoEmpresa(){           // Crea la tabla para la
empresa loggeada con los envios pendientes, transito y finalizados y sus cantidades.
992     let enviosPendientes = cantEnviosDeEstado("Pendiente");
993     let enviosEnTransito = cantEnviosDeEstado("En tránsito");
994     let enviosFinalizados = cantEnviosDeEstado("Finalizado");
995
996     document.querySelector("#infoEstadisticaEmpresaCantidadPorEstado").innerHTML =
997
998         <table class="table">
999             <header>
1000                 <tr>
1001                     <td><strong>Estado</strong></td>
1002                     <td><strong>Cantidad</strong></td>
1003                 </tr>
1004             </header>
1005             <tbody>
1006                 <tr>
1007                     <td>Pendiente</td>
1008                     <td>${enviosPendientes}</td>
1009                 </tr>
1010             </tbody>

```

```

1010         <td>En tránsito</td>
1011         <td>${enviosEnTransito}</td>
1012     </tr>
1013     <tr>
1014         <td>Finalizados</td>
1015         <td>${enviosFinalizados}</td>
1016     </tr>
1017 </tbody>
1018 </table>`
1019 }
1020
1021 function calcularInfoEstadisticaAdmin() {
1022     let tableBody = document.querySelector("#tablaInfoEstadisticaKMAAdminBody");
1023     tableBody.innerHTML = '';
1024
1025     for(let i = 0; i < usuarios.length; i++) {
1026         if (usuarios[i].tipo == "Empresa") {
1027             let nombreEmpresa = usuarios[i].razonSocial;
1028             let kmTotales = kmPorEmpresa(nombreEmpresa);
1029             let totalFinalizados = enviosFinalizadosPorEmpresa(nombreEmpresa)
1030             tableBody.innerHTML += `<tr>
1031                                     <td>${nombreEmpresa}</td>
1032                                     <td>${kmTotales}</td>
1033                                     <td>${totalFinalizados}</td>
1034                                 </tr>`
1035         }
1036     }
1037 }
1038
1039 function kmPorEmpresa(empresa){ // Recibe como parametro la razon
social y suma todos los km de los envios finalizados
1040     let total = 0;
1041
1042     for (let i=0; i < envios.length; i++) {
1043         if (envios[i].empresa == empresa && envios[i].estado == "Finalizado") {
1044             total += envios[i].distancia;
1045         }
1046     }
1047
1048     return total;
1049 }
1050
1051 function enviosFinalizadosPorEmpresa(empresa){ // Recibe como parametro
la razon social y suma la cantidad de envios finalizados
1052     let totalFinalizados = 0;
1053     for(let i=0; i<envios.length;i++){
1054         if(envios[i].empresa == empresa && envios[i].estado == "Finalizado"){
1055             totalFinalizados++;
1056         }
1057     }
1058     return totalFinalizados
1059 }

```

```

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

```

```

56
57 function toggleLogIn() {
58     ocultarPantallas();
59     document.querySelector("#panelLogInF1").style.display = "block";
60     document.querySelector("#ingresoUsuario").value = "";
61     document.querySelector("#ingresoPassword").value = "";
62     document.querySelector("#mensajeLogInNuevoUsuario").innerHTML = '';
63     document.querySelector("#mensajeLogIn").innerHTML = '';
64 }
65
66 function toggleRegister() {
67     document.querySelector("#btnTipoDeCuenta").style.display = "none";
68     document.querySelector("#tipoDeCuenta").value = "0";
69     ocultarPantallas();
70     document.querySelector("#panelRegistroF5").style.display = "block";
71 }
72 }
73
74 function ocultarPantallas() { //
75     Ocultar todas las pantallas
76     document.querySelector("#panelLogInF1").style.display = "none"
77     document.querySelector("#listadoDeEmpresasF2").style.display = "none"
78     document.querySelector("#panelCreacionVehiculosF3").style.display = "none" //
79     Crear tipos de vehiculos
80     document.querySelector("#verInfoEstadisticaF4").style.display = "none" //
81     Visualizar info estadistica
82     document.querySelector("#panelRegistroF5").style.display = "none" //
83     Registrarse en la app
84     document.querySelector("#solicitudEnvioF6").style.display = "none" //
85     Realizar solicitud de envio
86     document.querySelector("#listadoEnviosF7").style.display = "none" //
87     Listado de envios
88     document.querySelector("#listadoEnviosPendientesF8").style.display = "none" //
89     Listado de sol pendientes (empresas)
90     document.querySelector("#listadoEnviosTomadosF9").style.display = "none" //
91     Listado de sol tommdas (empresas)
92     document.querySelector("#registroUsuario").style.display = "none"
93     document.querySelector("#registroEmpresa").style.display = "none"
94     document.querySelector("#panelAdmin").style.display = "none"
95     document.querySelector("#panelEmpresa").style.display = "none"
96     document.querySelector("#panelPersona").style.display = "none"
97     document.querySelector("#infoEstadisticaEmpresaMejoresUsuarios").style.display =
98     "none";
99     document.querySelector("#infoEstadisticaEmpresaCantidadPorEstado").style.display
100     = "none";
101     pMensajeErrorFiltrarEmpresas.style.display = "none";
102     pMensajeErrorIngresarVehiculo.style.display = "none";
103     btnCerrarSesion.style.display = "none";
104 }
105
106 function displayRegistrarEmpresaPersona(usuario) {
107     updateSelectVehiculos("#registroVehiculoEmpresa");
108     if (usuario == "Empresa") { //
109         Muestra unicamente las opciones de registro de Empresa
110         document.querySelector("#registroUsuario").style.display = "none";
111         document.querySelector("#registroEmpresa").style.display = "block";
112     } else if (usuario == "Persona") { //
113         Muestra unicamente las opciones de registro de Persona
114         document.querySelector("#registroUsuario").style.display = "block";
115         document.querySelector("#registroEmpresa").style.display = "none";
116     }
117 }

```

```

104     } else if (usuario == "") {
105         document.querySelector("#errorSelectTipocuenta").value = "Debe seleccionar un
tipo de dato.";
106     }
107     displayBorradoCamposRegistroEmpresa();
108     displayBorradoCamposRegistroPersona();
109 }
110
111 function displayBorradoCamposRegistroEmpresa() {
112     document.querySelector("#registroAliasEmpresaF5").value = "";
113     document.querySelector("#registroNombreEmpresaF5").value = "";
114     document.querySelector("#registroRUTEmpresaF5").value = "";
115     document.querySelector("#registroRazonSocialEmpresaF5").value = "";
116     document.querySelector("#registroPassEmpresaF5").value = "";
117     document.querySelector("#registroVehiculoEmpresa").value = 0;
118
119     //Borrar error
120     document.querySelector("#pParaDesplegarErroresEmpresa").innerHTML = "";
121 }
122
123 function displayBorradoCamposRegistroPersona() {
124     document.querySelector("#registroAliasUsuarioPersonaF5").value = "";
125     document.querySelector("#registroPassUsuarioPersonaF5").value = "";
126     document.querySelector("#registroCIUsuarioPersonaF5").value = "";
127     document.querySelector("#registroNombreUsuarioPersonaF5").value = "";
128     document.querySelector("#registroApellidoUsuarioPersonaF5").value = "";
129
130     //Borrar error
131     document.querySelector("#pParaDesplegarErroresPersona").innerHTML = "";
132 }
133
134 function displayNavPanel(tipo) {
135     switch (tipo) {
136         case "Administrador":
137             document.querySelector("#panelAdmin").style.display = "block";
138             document.querySelector("#panelLogInF1").style.display = "none";
139             btnMostrarLogIn.style.display = "none";
140             btnMostrarRegistrar.style.display = "none";
141             btnCerrarSesion.style.display = "block";
142             break;
143
144         case "Empresa":
145             document.querySelector("#panelEmpresa").style.display = "block";
146             document.querySelector("#panelLogInF1").style.display = "none";
147             btnMostrarLogIn.style.display = "none";
148             btnMostrarRegistrar.style.display = "none";
149             btnCerrarSesion.style.display = "block";
150             break;
151
152         case "Persona":
153             document.querySelector("#panelPersona").style.display = "block";
154             document.querySelector("#panelLogInF1").style.display = "none";
155             btnMostrarLogIn.style.display = "none";
156             btnMostrarRegistrar.style.display = "none";
157             btnCerrarSesion.style.display = "block";
158             break;
159
160         default:
161             break;
162     }

```

```

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

```



```

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

```

```

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

```



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

```
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 }
```