



UNAH
UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

Facultad de Ingeniería
Departamento de Ingeniería en Sistemas
Sistemas Expertos (IS-912)

Primera Entrega de Proyecto

Presentado por:

Allan Noel López Cruz 20131003133
Jesús Daniel Arias 20141001564
Oscar Andrés Vasquez 20131009602
Saúl Alberto Guzmán 20121902121

Ciudad Universitaria, Tegucigalpa MDC, Francisco Morazán
16 de noviembre 2017

Índice

Sección	Pág.
Introducción	3
I Título y descripción del proyecto	4
¿A quién va dirigido? ¿Cómo funciona?	
II Modelo de negocio canvas	6
III Módulos o formularios	7
IV Tipos de usuario y niveles de acceso	9
V Mockups de formularios	10
VI Modelo relacional de base de datos	13
VII Diccionario de datos	14
VIII Herramientas de desarrollo utilizadas	21
IX Funcionalidades y tecnologías investigadas	25
X Glosario de términos:	34
XI Evaluación de equipo	35

Introducción

El siguiente documento contiene la información general del proyecto de la clase Sistemas expertos (Desarrollo Web). En las páginas que siguen se describe detalladamente la propuesta del nuestro proyecto: carry, junto con su logotipo y slogan.

Además, hemos desglosado la lista de módulos, páginas, subpáginas y formularios que conforma la estructura de nuestro aplicativo web. Posteriormente se explica brevemente las funciones de cada módulo.

En este documento se presenta un modelo canvas de forma vertical para que pueda ser legible sin problemas, adjunta al entregable se presenta un archivo .pdf con el formato convencional del modelo canvas.

Presentamos capturas de un conjunto de formularios que han sido completados para esta entrega, así como las imágenes de mockups de los formularios que faltan por maquetar.

Para completar todas las tareas de esta entrega se hizo uso de diferentes herramientas: bootstrap (requerido), la herramienta Balsamiq para diseño de mockups. Las imágenes utilizadas son en su mayoría descargadas del sitio Pixabay, todas cedidas por sus autores y publicadas por ellos en dicho sitio. Junto con esas herramientas, se deja documentado un conjunto de otras herramientas utilizadas para mejorar el funcionamiento del proyecto.

I Título y descripción del proyecto



Carry será un sitio en internet por medio del cual se puedan encontrar negocios y productos, ordenarlos y recibirlos en determinado lugar. El sitio estará orientado a tres servicios:

1. Gestión de entregas
2. Compras por catálogo electrónico
3. Envío de paquetes.

¿A quién va dirigido?

1. MiPyMes: Gestión de entregas

Existen muchas MiPyMes que se dedican a la fabricación o producción de productos a pequeña escala. Es común que estos pequeños negocios no puedan ofrecer un servicio de entrega o distribución automatizado de sus productos, pues en muchos casos no cuentan con recursos económicos para poder pagar personal que haga ese trabajo, o para cubrir el costo operativo que eso representa.

→ Las ordenes son ingresadas al sitio por un usuario autorizado en la cuenta del negocio.

2. Usuarios convencionales: Compras por catálogo:

Si un negocio no sólo quiere gestionar sus entregas, sino que además quiere alcanzar a más clientes, entonces publica su catálogo para que cualquier usuario vea sus productos y pueda ordenarlos, y entonces uno de los agentes de carry se encarga de llevarlo al lugar especificado.

→ Un usuario cualquiera ingresa al sitio, busca productos que necesita, los ordena, indica el destino y un agente Carry se lo entrega.

3. MiPyMes y otros usuarios: Envío de paquetes en la ciudad

En muchas ocasiones las personas necesitan recoger o entregar un paquete, pero su tiempo no se los permite. El usuario indica donde se recogerá el paquete y donde debe entregarlo, un agente de entrega carry, hará la diligencia.



Un usuario solicita el servicio de entrega, indica el origen donde se recogerá la entrega y el destino final de la misma.

¿Cómo funciona?

La idea es contar con un equipo de personas que trabajarán para carry realizando las entregas por toda la ciudad. Cuando un usuario realice un pedido, el sitio debe identificar cuáles de nuestros agentes están más cerca del lugar donde se debe recoger el pedido, sin que interfiera con la ruta que dicho agente tiene previamente definida.

Carry no tendrá participación en la fijación de precios, el servicio consiste únicamente en realizar la entrega a cambio de una tarifa por acarreo. En casos comunes, el pago de la tarifa de entrega le corresponde al cliente que solicita el producto, salvo que se llegue a un acuerdo entre el vendedor y carry.

¿Quiénes harán la entrega?

Lo agentes carry serán personas mayores de dieciocho años, que trabajarán con la empresa en calidad de empleado. Cuando un usuario realice un pedido, recibirá un numero de orden que solo él y el agente conocerán, y es el que servirá para hacer efectiva la entrega. El usuario deberá brindar su nombre y número de teléfono, puede suceder que, en algunos casos, dependiendo del servicio, el cliente tenga que presentar algún tipo de documentación específica.

¿Cómo se realizará la entrega?

Los agentes de carry tendrán diferentes formas de transportarse, principalmente lo harán en bicicleta y motocicleta, todo puede cambiar dependiendo la evolución del negocio y al tamaño de las entregas.

¿Formas de pago?

La principal forma de pago será el efectivo. Cada entrega tendrá un tiempo y precio mínimo, que puede variar en casos muy específicos (peso, tamaño, distancia, recorrido). Carry se reservará el derecho de fijar si el pago se haga previo o en el momento de la entrega.

Para los usuarios empresariales existirán planes de pago por suscripción. Dichas suscripciones podrán ser semanales, mensuales, trimestrales, etc. Nuestros usuarios deberán realizar un depósito a una cuenta bancaria y nuestro sitio podrá revisar el depósito y verificar la validez de la suscripción. Lo anterior mientras se configuran otras opciones de pago en línea.

II Modelo de negocio canvas

Socios clave				1
Empresas o personas que deseen contribuir o invertir su capital.	Clientes empresariales	Agencias de ventas de bicicletas o motocicletas	Talleres mecánicos de reparación	4
Segmentos de clientes				2
MiPymes sin capacidad de distribución de sus productos, y que desean hacerlo a un costo flexible.	Cualquier usuario que desee ordenar algo sin moverse del lugar donde se encuentra			2
Propuesta de valor				3
Plataforma de gestión de entregas.	"carry.com" será un sitio en internet por medio del cual se puedan encontrar negocios y productos, ordenarlos y recibirlos en determinado lugar, de forma rápida y segura.	Además se podrán realizar envíos de paquetes.		3
Actividades clave				4
Desarrollar sitio web y montarlo en un servidor de internet.	Contratar agentes de entrega.	Promocionar el sitio		3
Canales de distribución				5
Atención al cliente vía teléfono, correo electrónico, o redes sociales.	Presentación del servicio en el sitio web.			2
Flujos de ingreso				6
Ingreso por comisión de venta en el catalogo electrónico.	Ingresos por publicidad mostrada en el sitio web.	Ingreso por pago de suscripción de cliente empresarial.	Ingreso por cada tarifa cobrada de cada entrega realizada.	4
Estructura de costo				7
Costos de hosting y mantenimiento del sitio web	Costos de publicidad y administración.	Pago de impuestos y otros gastos.	Pago de planilla de empleados, alquiler, préstamos.	4
Recursos Clave				8
Computadoras, establecimiento físico, servidor. Desarrolladores web. Personal para realizar las entregas. Capital Inicial.				1
Relaciones con los clientes				9
Asistencia personalizada	Motivación a través de ofertas, promociones o descuentos según las épocas del año.			2

III Módulos o formularios

Desglose:

1. Módulo de administración AU

1.1 Administración de usuarios y empleados

- 1.1.1 Activar/suspender usuarios y aspirantes
- 1.1.2 Aprobar registro empleados
- 1.1.3 habilitar usuario designado

1.2 Administración de catálogo electrónico

- 1.2.1 *Registro de artículos*

2. Módulo de catálogo electrónico

2.1 Rejilla de artículos

- 2.1.1 *Registro de artículos* AU, DU
- 2.1.2 Suspender artículos DU
- 2.1.3 Eliminar artículos DU
- 2.1.4 *Editar artículos* DU
- 2.1.5 Buscar artículos AU, DU, U
- 2.1.6 Agregar al carrito AU, DU, U

2.2 Carrito de ordenes

-> facturar

2.3 Detalles de oferta AU, DU, U

3. Módulo de entregas

3.3 Panel de entregas

- 3.3.1 Entregas realizadas AU, DU, U, E
- 3.3.2 Entregas pendientes AU, DU, U, E
- 3.3.3 Entregas canceladas AU, DU, U, E
- 3.3.4 Cancelar entrega AU, DU, U, E
- 3.3.5 Notificar entrega AU, E

3.4 *Registro de nueva entrega* AU, DU, U,

4. Modulo facturación

4.1 Panel de facturas

- 4.1.1 Facturas pagadas AU, DU, U
- 4.1.2 Facturas pendientes AU, DU, U

4.2 Registro de factura AU, DU, U

5. Módulo de usuarios

4.3 *Formulario de registro de usuario (Usuario)*

4.4 *Formulario de registro de empleado (aspirante)*

4.5 *Formulario de acceso (tipificado, usuario, empleado)*

4.6 Panel configuración de perfil (usuario, empleado)

Niveles de acceso:

U: Usuario convencional
 DU: Usuario designado
 AU: Usuario administrador
 E: empleado

Tipografía:

Módulo

Subpágina

Formulario de registro o modificación.

Funciones de módulo

Detalle:**Módulo de administración**

El módulo de administración permitirá habilitar, deshabilitar e incluso registrar nuevos usuarios en el sistema de carry. Podrá realizar consultas en el registro de actividad de los usuarios del sistema. En este módulo se deberá ofrecer una pequeña panorámica de la actividad general del sitio. Únicamente el administrador tendrá acceso a este módulo.

Módulo de catálogo electrónico

En el catálogo electrónico, se podrá visualizar y consultar los artículos públicos que han sido registrados para su venta. Desde el catálogo electrónico se podrán registrar ordenes de envío de productos. Los empleados son los únicos que no pueden acceder al catálogo electrónico.

Módulo de entregas

El módulo de entregas debe mostrar un historial de las entregas solicitadas o realizadas, dependiendo de cómo se haya autenticado el usuario al momento de ingresar. Desde este módulo se maneja el registro de las ordenes que los usuarios solicitan. En este, como en todos los módulos, las opciones o funcionalidades que se muestren dependerán del usuario que haya ingresado.

Módulo de facturación

El módulo de facturación debe mostrar un historial de las facturas pagadas por el usuario y los respectivos artículos sujetos a cada factura. El módulo de factura supone ser exclusivo para los usuarios convencionales que hacen las compras, como los usuarios empresariales.

Módulo de usuarios

El módulo de usuarios lo componen los formularios de acceso y registro de usuario, junto al panel de usuario donde cada usuario podrá realizar cambios a la información de su cuenta.

IV Tipos de usuario y niveles de acceso

Usuario administrador

Es el usuario de mayor jerarquía. Tiene acceso a todos los módulos y puede consultar toda la información de la actividad del sitio.

Usuario designado (cliente empresarial)

El usuario designado o cliente empresarial, es un usuario que está vinculado a una empresa, es decir, es el responsable de manejar la cuenta de una empresa en el sistema de carry. El usuario designado puede inscribir y comprar artículos en el catálogo electrónico. También puede hacer uso del servicio de envío de paquetes.

Empleado

Este tipo de usuario o acceso es para el personal que realiza las entregas. Cuando un usuario empleado ingresa al panel de entregas del sitio, podrá observar todas las ordenes que se deben entregar, y un historial de las entregas que ha realizado. Además, tendrá las opciones de notificar una entrega, cuando esta haya sido completada. El empleado no podrá registrar nuevas entregas, pues es el sistema quien asigna las entregas de acuerdo a la localización y distancias.

Usuario convencional

El usuario de cliente convencional es cualquier persona que desee utilizar cualquiera de los servicios disponibles en "carry.com". Cuando un usuario convencional ingresa al sistema, tiene acceso a los siguientes módulos:

En el catálogo electrónico, el usuario convencional podrá observar las ofertas disponibles que han sido publicadas por los usuarios designados y ordenar con el carrito de compras. Cuando decida pagar, accederá al módulo de facturación para realizar el pago del o los productos que ha solicitado, una vez que el pago de los productos, el cliente indica en el módulo de entregas los detalles de la entrega.

V Mockups de formularios

**Estos fueron los mockups iniciales, al final del desarrollo, algunos fueron cambiados en algunos aspectos o incluso en su totalidad, para un mejor funcionamiento.*

Formulario de ajustes de cuenta: ajustes generales

The mockup shows a web browser window with a sidebar menu on the left containing 'General', 'Seguridad y Login', 'Métodos de Pago', and 'Lista de Roles'. The main content area is titled 'Ajustes Generales de Cuenta' with a gear icon. It contains three input fields: 'Nombre', 'Apellido', and 'Correo', each followed by an 'Editar' button.

Formulario de ajustes de cuenta: cambio de contraseña

The mockup shows the same web browser window, but the sidebar menu now highlights 'Seguridad y Login'. The main content area is titled 'Seguridad y Login' with a padlock icon. It contains a section titled 'Cambiar Contraseña' with three input fields: 'Actual', 'Nuevo', and 'Confirmar Nuevo', followed by a 'Guardar' button.

Formulario de ajustes de cuenta: método de pago

General
Seguridad y Login
Método de Pago
Lista de deseos

Método de Pago

Agregar método de pago

Número de tarjeta

Nombre en la tarjeta

Fecha vencimiento: Enero | Diciembre

Código seguridad

Cancelar

Agregar método

Método de pago 1

Editar

Método de pago 2

Editar

Editar

Coloquear esta sección cuando se da click en Cancelar

Esconda esta sección cuando se da click en Editar

Panel de entregas:

Ordenar por:

Destino Final ☐

Destinatario ☐

Fecha ☐

Estado ☐

Artículo	Destino Final	Destino Origen	Distancia	Estado	Fecha	Costo

← Page 1 of 3 →

Formulario de solicitud de orden de envío (entrega):

The screenshot shows a web browser window titled 'A Web Page'. Inside the browser, there is a form titled 'Entrega'. The form contains two text input fields: 'Direccion Origen' and 'Direccion Final'. Below these fields are two buttons: 'Aceptar' and 'Cancelar'.

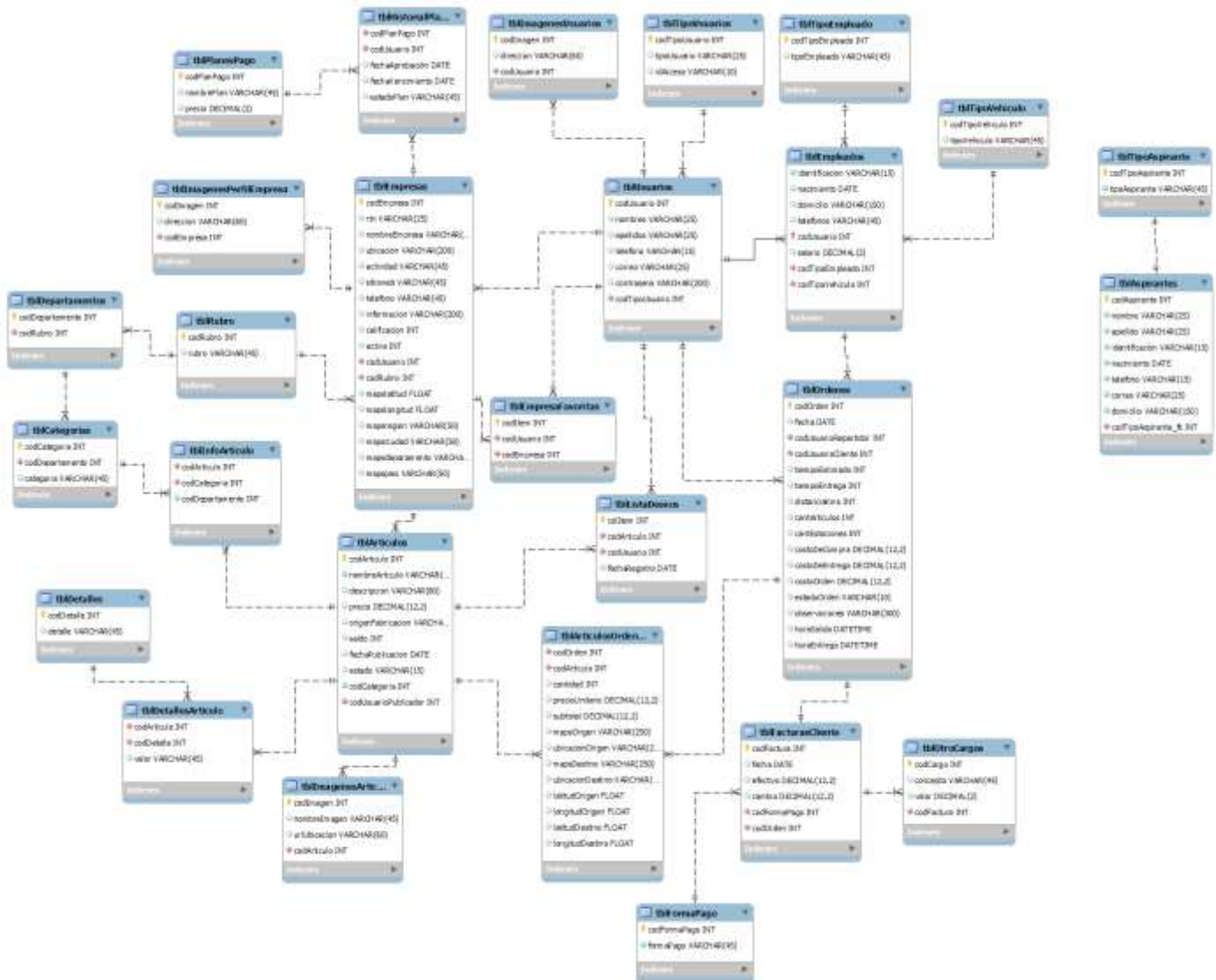
Panel de facturas:

The screenshot shows a web browser window titled 'A Web Page'. Inside the browser, there is a panel titled 'Panel de facturas'. The panel contains a table with the following data:

Fecha\	orden	Total	entregado
25/06/2016i	4120	200	<input checked="" type="checkbox"/>
25/06/2012i	1882	8000	<input type="checkbox"/>
12/04/2017	2089	100	<input checked="" type="checkbox"/>
25/06/2016i	40	200	<input checked="" type="checkbox"/>

Below the table, there is a button labeled 'Cancelar seleccionada'. To the left of the table, there are two buttons: 'Métodos de pago' and 'Lista de deseos'. To the right of the table, there is a dropdown menu labeled 'Ordenar por'.

VI Modelo relacional de base de datos



VII Diccionario de datos

1	tblArticulos				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codArticulo	int(11)	Codigo del Articulo	AUTO_INCREMENT	Primaria
2	nombreArticulo	varchar(45)	Nombre del articulo		
3	descripcion	varchar(80)	Descripcion del articulo		
4	precio	decimal(12,2)	Precio por unidad del articulo		
5	origenFabricacion	varchar(45)	Empresa donde fue fabricado		
6	saldo	int(11)	Cantidad de articulos disponibles		
7	fechaPublicacion	date	Fecha de publicación del articulo		
8	estado	varchar(15)	Estado de venta del articulo		
9	codCategoria	int(11)	Codigo de la categoria		foránea
10	codUsuarioPublicador	int(11)	Codigo del usuario que publico el articulo		foránea

2	tblarticulosordenados				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codOrden	int(11)	Codigo de orden a la que pertenece		primaria
2	codArticulo	int(11)	Codigo del Articulo que se hace referencia		
3	cantidad	int(11)	Cantidad de productos		
4	precioUnitario	decimal(12,2)	Precio del articulo		
5	subtotal	decimal(12,2)	Subtotal del costo del articulo		
6	mapsOrigen	varchar(250)	Mapa donde esta ubicado		
7	ubicacionOrigen	varchar(200)	direccion de la ubicacion		
8	mapsDestino	varchar(250)	Mapa de destino		
9	ubicacionDestino	varchar(250)	Ubicacion del destino		
10	latitudOrigen	float	Latitud de la ubicacion del articulo		
11	longitudOrigen	float	Longitud ubicacion del articulo		

3	tblaspirantes				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codAspirante	int(11)	Codigo del Aspirante	AUTO_INCREMENT	primaria
2	nombre	varchar(25)	Nombre del aspirante		
3	apellido	varchar(25)	Apellido del aspirante		
4	identificacion	varchar(15)	Numero de indentidad		
5	nacimiento	date	Fecha de nacimiento		
6	telefono	varchar(15)	Telefono del aspirante		
7	correo	varchar(25)	Correo del aspirante		
8	domicilio	varchar(150)	Domicilio del aspirante		
9	codTipoAspirante_fk	int(11)	Tipo de aspirante		foránea

4	tblcategorias				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codCategoria	int(11)	Codigo de la categoria	AUTO_INCREMENT	Primaria
2	codDepartamento	int(11)	Codigo del Departamento		Foranea
3	categoria	varchar(45)	Nombre de la categoria		

5	tblDepartamentos				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codDepartamento	int(11)	Codigo del departamento	AUTO_INCREMENT	Primaria
2	codRubro	int(11)	Codigo al rubro		Foránea

6	tbldetallesarticulo				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codArticulo	int(11)	Codigo del Articulo		Foranea
2	codDetalle	int(11)	Codigo del detalle del articulo		Foranea
3	valor	varchar(45)	Valor del articulo		

7	tbldetalles				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codDetalle	int(11)	Codigo del detalle	AUTO_INCREMENT	Primaria
2	detalle	varchar(45)	Descripcion del detalle		

8	tblEmpleados				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	identificacion	varchar(15)	numero de indentificaicion		
2	nacimiento	date	Fecha de nacimiento		
3	domicilio	varchar(150)	Direccion del domicilio		
4	telefonos	varchar(45)	Numero telefonico		
5	codUsuario	int(11)	Codigo de usuario		Foranea
6	salario	decimal(2,0)	Salario que gana		
7	codTipoEmpleado	int(11)	Codigo del tipo de empleado que pertenece		Foranea
8	codTipoVehiculo	int(11)	Codigo del tipo de vehiculo		Foranea
9	identificacion	varchar(15)	numero de indentificaicion		

9	tblEmpresasFavoritas				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codItem	int(11)	Codigo de empresa favorita	AUTO_INCREMENT	Primaria
2	codUsuario	int(11)	Codigo de usuario que seleccion dicha empresa como su favorita		

3	codEmpresa	int(11)	Codigo de empresa que selecciono como favorita		
---	------------	---------	--	--	--

10	tblEmpresas				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codEmpresa	int(11)	Codigo de la empresa	AUTO_INCREMENT	Primaria
2	rtn	varchar(25)	Registro tributario nacional de la empresa		
3	nombreEmpresa	varchar(45)	Nombre de la empresa		
4	ubicacion	varchar(200)	Domicilio de la empresa		
5	actividad	varchar(45)	Actividad a la que se dedica		
6	sitioweb	varchar(45)	Direccion o pagina web de la empresa		
7	telefono	varchar(45)	Numero telefonico de la empresa		
8	informacion	varchar(200)	Informacion o descripcion sobre la empresa		
9	calificacion	int(11)	Calificacion de los servicios de la empresa en general		
10	activa	int(11)	Estado de operaciones de la empresa		
11	codUsuario	int(11)	Usuario que administra la empresa (codigo)		Foránea
12	codRubro	int(11)	Codigo de rubro a la que la empresa se dedica		Foránea
13	mapslatitud	float	Latitud, coordenada geografica		
14	mapslongitud	float	Longitud, coordenada geografica		
15	mapsregion	varchar(50)	Region donde esta ubicada la empresa		
16	mapsciudad	varchar(50)	Ciudad en donde esta la empresa		
17	mapsdepartamento	varchar(50)	Departamento donde se ubica la empresa		
18	mapspais	varchar(50)	Pais donde se encuentra la empresa		

11	tblFacturasCliente				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codFactura	int(11)	Codigo de la factura	AUTO_INCREMENT	Primaria
2	fecha	date	Fecha de emision de la factura		
3	efectivo	decimal(12,2)	Monto que se recibio en efectivo		
4	cambio	decimal(12,2)	Cambio del monto total		
5	codFormaPago	int(11)	Codigo de forma de pago (efectivo, debito o credito)		foranea
6	codOrden	int(11)	Numero de orden		foranea

12	tblFormaPago				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codFormaPago	int(11)	Codigo de forma de pago	AUTO_INCREMENT	Primaria
2	formaPago	varchar(45)	Debito, credito o efectivo		

13	tblImágenesArtículo				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codImagen	int(11)	Codigo de la imagen	AUTO_INCREMENT	Primaria
2	nombrelimagen	varchar(45)	Nombre de la imagen		
3	urlUbicacion	varchar(60)	Url donde esta ubicada la imagen		
4	codArticulo	int(11)	Codigo del articulo que se muestra		Foránea

14	tblImágenesPerfilEmpresa				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codImagen	int(11)	Codigo de la imagen de perfil de la empresa	AUTO_INCREMENT	Primaria
2	direccion	varchar(60)	Url de la imagen		
3	codEmpresa	int(11)	Codigo de la empresa a que pertenece la foto		Foránea

15	tblInfoArticulos				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codArticulo	int(11)	Codigo del articulo	AUTO_INCREMENT	Primaria
2	codCategoria	int(11)	Codigo de categoria a la que pertenece el articulo		Foránea
3	codDepartamento	int(11)	Codigo de departamento al que pertenece el articulo		Foránea

16	tblImágenesUsuarios				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codImagen	int(11)	Codigo de la imagen de perfil del usuario	AUTO_INCREMENT	Primaria
2	direccion	varchar(60)	Url de la imagen de perfil del usuario		
3	codUsuario	int(11)	Codigo de usuario a quien pertenece la foto		Foránea

17	tblHistorialPlanes				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codPlanPago	int(11)	Codigo de plan de suscripcion	AUTO_INCREMENT	Primaria

2	codUsuario	int(11)	Codigo del usuario que compro el plan		Foránea
3	fechaAprobación	date	Fecha de aprobacion del plan		
4	fechaVencimiento	date	Fecha de vencimiento de la suscripcion		
5	estadoPlan	varchar(45)	Estado del plan (activo o vencido)		

18	tbllistadeseos: tabla de lista de deseos de usuarios.				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	colItem	int(11)	Codigo de item en la lista de deseo.	AUTO_INCREMENT	Llave primaria.
2	codArticulo	int(11)	Llave del codigo de artículo.		Llave foránea.
3	codUsuario	int(11)	Llave identificadora del usuario.		Llave foránea.
4	fechaRegistro	date	Fecha en que se agrega el artículo a la lista.		

19	tblordenes				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codOrden	int(11)	Codigo correlativo a la orden.	AUTO_INCREMENT	Llave primaria.
2	fecha	date	Fecha de la orden.		
3	codUsuarioRepartidor	int(11)	Identificador del repartidor.		Llave foránea.
4	codUsuarioCliente	int(11)	Identificador del cliente.		Llave foránea.
5	tiempoEstimado	int(11)	Tiempo estimado de la orden.		
6	tiempoEntrega	int(11)	Tiempo de entrega.		
7	distanciaKms	int(11)	Distancia para la entrega.		
8	cantArticulos	int(11)	Cantidad de artículos.		
9	cantEstaciones	int(11)	Cantidad de estaciones.		
10	costoDeCompra	decimal(12,0)	Costos de compra.		
11	costoDeEntrega	decimal(12,0)	Costos de la entrega.		
12	costoOrden	decimal(12,0)	Costo total de la orden.		
13	estadoOrden	varchar(10)	Estado actual de la orden.		
14	observaciones	varchar(300)	Observaciones.		
15	horaSalida	datetime	Hora de salida del repartidor.		
16	horaEntrega	datetime	Hora de entrega.		

20	tbltrocargos				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codCargo	int(11)	Codigo correlativo al cargo.	AUTO_INCREMENT	Llave primaria.
2	concepto	varchar(45)	Concepto del cargo.		
3	valor	decimal(2,0)	Valor del cargo.		
4	codFactura	int(11)	Factura a la que pertenece.		Llave foránea.

21	tblplanespago				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave

22	codPlanPago	int(11)	Codigo correlativo al plan de pago.	AUTO_INCREMENT	Llave primaria.
2	nombrePlan	varchar(45)	Nombre del plan.		
3	precio	decimal(2,0)	Precio del plan.		

23	tblrubro				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codRubro	int(11)	Codigo correlativo al rubro.	AUTO_INCREMENT	Llave primaria.
2	rubro	varchar(45)	Nombre del rubro.		

24	tblaspirantes				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codAspirante	int(11)	Código correlativo al aspirante.	AUTO_INCREMENT	Llave primaria.
2	nombre	varchar(25)	Nombre del aspirante.		
3	apellido	varchar(25)	Apellido del aspirante.		
4	identificacion	varchar(15)	Identificación del aspirante.		
5	nacimiento	date	Fecha nacimiento del aspirante.		
6	telefono	varchar(15)	Telefono del aspirante.		
7	correo	varchar(25)	Correo del aspirante.		
8	domicilio	varchar(150)	Domicilio del aspirante.		
9	codTipoAspirante_fk	int(11)	Id tipo de aspirante.		Llave foránea.

25	tbltipoempleado				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codTipoEmpleado	int(11)	Código correlativo al tipo de empleado.	AUTO_INCREMENT	Llave primaria.
2	tipoEmpleado	varchar(45)	Tipo de empleado		

26	tbltipoempresa				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codTipoEmpresa	int(11)	Código correlativo al tipo de empresa.	AUTO_INCREMENT	Llave primaria.
2	tipoEmpresa	varchar(45)	Tipo de empresa.		

27	tbltipousuarios				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codTipoUsuario	int(11)	Código correlativo al tipo de usuario.	AUTO_INCREMENT	Llave primaria.
2	tipoUsuario	varchar(25)	Tipo de usuario.		
3	idAcceso	varchar(10)	Id de acceso.		

28	tbltipovehiculo				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codTipoVehiculoPrimaria	int(11)	Numero correlativo de identificacion unico	AUTO_INCREMENT	Primaria
2	tipoVehiculo	varchar(45)	El tipo de vehiculo que un usuario utiliza		

29	tblusuarios				
#	Nombre	Tipo	Descripción	Extras	Tipo de Llave
1	codUsuario	int(11)	Codigo correlativo del usuario	UTO_INCREMENT	Primaria
2	nombres	varchar(25)	Nombres del usuario		
3	apellidos	varchar(25)	Apellidos del usuario		
4	telefono	varchar(15)	telefono del usuario		
5	correo	varchar(25)	correo electronico de usuario		
6	contrasena	varchar(200)	contraseña de usuario		
7	codTipoUsuario	int(11)	codigo de acceso al sistema		Foránea

VIII Herramientas de desarrollo utilizadas

Bootstrap

Bootstrap es un framework de front end gratuito. Este framework trae consigo diseños de CSS y HTML basados en plantillas. Bootstrap fue desarrollado por Mark Otto en Twitter y liberado al público como proyecto open source mediante GitHub.

Para este proyecto se utilizó la versión 3, ya se puede descargar la versión 4 en el siguiente link:

<https://getbootstrap.com/>

A continuación, exponemos las razones de por qué utilizamos Bootstrap en nuestro proyecto:

- Se implementa de una manera ágil, basta con descargar los archivos, descomprimirlos e incluirlos en la etiqueta head de nuestro documento HTML. Además, ofrece la posibilidad de utilizar css pre-procesado.
- Su sistema de rejilla (grid) contiene 12 columnas, componentes y layouts full responsive. Éstas características no permiten como programadores diseñar layouts que varíen según el tamaño del dispositivo en el que se está desplegando el aplicativo sin necesidad de utilizar medias queries y pensar mucho en los pixeles de cada dispositivo.
- Un aplicativo web cuenta con elementos como títulos, listas, tablas, botones, formularios etc. Todos estos elementos fundamentales de HTML han sido estilizados con clases pre-establecidas.
- Además de proveernos con clases, Bootstrap trae consigo un plugins de Javascript empaquetados, lo que facilita la creación de componentes complejos como un menú dropdown o sliders, tabs y acordeones.
- Contiene una documentación extensiva y muy bien ejemplificada.

WampServer

Es un entorno de desarrollo web para Windows con el que podrás crear aplicaciones web con Apache, PHP y bases de datos MySQL database. También incluye PHPMyAdmin y SQLiteManager para manejar tus bases de datos.

Provee a los desarrolladores con los cuatro elementos necesarios para un servidor web: un Sistema Operativo (Windows), un manejador de base de datos (MySQL), un software para servidor web (Apache) y un software de programación script Web (PHP (generalmente), Python o PERL), debiendo su nombre a dichas herramientas. Lo mejor de todo es que WAMP5 es completamente gratuito. WAMP incluye, además de las últimas versiones de Apache, PHP y MySQL, versiones anteriores de las mismas, para el caso de que se quiera testear en un entorno de desarrollo particular. El uso de WAMP permite servir páginas HTML a Internet, además de poder gestionar datos en ellas, al mismo tiempo WAMP, proporciona lenguajes de programación para desarrollar aplicaciones Web.

phpMyAdmin

Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet, está disponible bajo la licencia GPL (General Public License) y en más de 50 idiomas este proyecto se encuentra vigente desde el año 1998. Con esta herramienta puedes crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos.

NodeJS

Node.js es una librería y entorno de ejecución de E/S dirigida por eventos y por lo tanto asíncrona que se ejecuta sobre el intérprete de JavaScript creado por Google V8. Lo cierto es que está muy de moda aunque no es algo nuevo puesto que existen librerías como Twisted que hacen exactamente lo mismo pero sí es cierto que es la primera basada en JavaScript y que tiene un gran rendimiento.

Además de la alta velocidad de ejecución de Javascript, la verdadera magia detrás de Node.js es algo que se llama Bucle de Eventos (Event Loop). Para escalar grandes volúmenes de clientes, todas las operaciones intensivas I/O en Node.js se llevan a cabo de forma asíncrona. El enfoque tradicional para generar código asíncrono es engorroso y crea un espacio en memoria no trivial para un gran número de clientes (cada cliente genera un hilo, y el uso de memoria de cada uno se suma). Para evitar esta ineficiencia, así como la dificultad conocida de las aplicaciones basadas en hilos, (programming threaded applications), Node.js mantiene un event loop que gestiona todas las operaciones asíncronas.

Es rápido, muy rápido. Esto es importante por varias razones:

1. El desarrollo es más rápido.
2. ejecución de tests de unidad se puede hacer más rápido.
3. Las aplicaciones son La
4. más rápidas y por tanto la experiencia de usuario es mejor.
5. Menor coste de infraestructura.




Además de sus aplicaciones innatas, Node.js tiene muchas aplicaciones que se han escrito en módulos, excelente para agregar capacidades adicionales a las aplicaciones Node.js. Uno de los más famosos es Socket.io, un módulo para gestionar las conexiones persistentes entre el cliente y el servidor, permitiendo al servidor enviar actualizaciones en tiempo real a los clientes. (que es la magia que mantiene a nuestra plataforma de GPS y seguimiento en línea).

Para este proyecto se trabajó con la versión 6.11, esta y otras versiones se pueden descargar en el sitio oficial: <https://nodejs.org/es/>

Editores de texto

Los editores de texto son programas informáticos que crean y editan archivos digitales en un formato de texto básico o texto plano, es decir, archivos que no contengan formato de texto específico.

Para este proyecto, cada uno de los integrantes del grupo utilizó el de su preferencia:

<p>Sublime text</p>  <p>https://www.sublimetext.com/</p>	<p>Atom</p>  <p>https://atom.io/</p>	<p>vscode</p>  <p>https://code.visualstudio.com/</p>
--	--	--

Gestor MySQL Workbench

MySQL Workbench es una herramienta diseñada para arquitectos de bases de datos, desarrolladores y administradores de bases de datos (DBA). También nos proporciona modelado de datos (con lo que podemos generar un script para crear una base de datos), desarrollo de SQL y herramientas de administración para la configuración del servidor, administración de usuarios, respaldos de bases de datos, etc.

Este programa nos facilita el diseño y mantenimiento de la base de datos, automatiza ciertas tareas que requieren mucho tiempo y que son propensas a errores. Permite el diseño de bases de datos basadas en modelos, que es la metodología más eficiente para crear bases de datos válidas y que funcionan bien, minimizando el factor de error humano al crear un script desde cero. Las utilidades de Validación de Modelos y Esquemas hacen cumplir los estándares de mejores prácticas para el modelado de datos, también hacen cumplir los estándares de diseño físico específicos de MySQL para que no se cometan errores al construir nuevos diagramas ER o generar bases de datos físicas MySQL.

MySQL Workbench es una herramienta que facilitó el diseño de la base de datos del proyecto en cuestión (Carry). Es programa muy interactivo y de fácil uso, nos permitió el diseño de una base de datos relativamente grande partiendo desde un modelo relacional para tener una mejor perspectiva de cómo se iban a modelar y manipular los datos para el correcto funcionamiento de la página web; luego a través de un proceso de Forward Engineering (opción que nos permite MySQL Workbench), generamos el script de la base de datos a partir del modelo, listo para crear y montar la base de datos en el servidor.

GitHub

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.

GitHub aloja un repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en otra cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas nuevas modificaciones se puede enviar un pull al dueño del proyecto. Éste podrá analizar los cambios realizados fácilmente, y si considera interesante la contribución, adjuntarlo con el repositorio original.

El código fuente de este proyecto se puede encontrar en el siguiente link:
<https://github.com/AllanNLopez/carry>

IX Funcionalidades y tecnologías investigadas

Express

Express es un framework de Node.js. Está construido en torno a la configuración simple de conexiones de middleware. Express provee un conjunto robusto de características para desarrollar aplicaciones web. Sin duda el éxito de Express radica en lo sencillo que es usarlo, y además abarca un sin número de aspectos que muchos desconocen, pero son necesarios.

Con el siguiente ejemplo podemos notar lo sencillo que es comenzar a instalar Express. Para instalar Express basta con que corramos el siguiente comando en nuestra terminal:

```
$ npm install express --save
```

Express nos permite construir middlewares a la medida de nuestra aplicación para responder a requests HTTP. Express define una tabla de enrutamiento, la cual es utilizada para realizar diferentes acciones basadas en métodos HTTP y URL's. También nos permite renderizar páginas HTML basadas en argumentos recibidos y plantillas, como es el caso de Jade, Pug y EJS, siendo éste último más simple y con una menor sintaxis por aprender.

La ventaja de utilizar Express en nuestro proyecto es que no tenemos que repetir el mismo código una y otra vez. Node.js es un mecanismo de entrada y salida que contiene un módulo HTTP. Si hubiésemos utilizado éste módulo, mucho trabajo como parsing, cookies, almacenamiento de sesiones, seleccionando rutas basadas en expresiones reguladas, tendrían que ser re-implementadas.

Datatables

DataTables es un plug-in para la biblioteca jQuery Javascript. Es una herramienta altamente flexible, basada en los fundamentos de la mejora progresiva, y agregará controles avanzados de interacción a cualquier tabla HTML. Datatables permite mostrar y operar datos de manera "dinámica" con las tablas de las paginas web de los sitios.

Datatables es un software de código libre muy completo, que ofrece funcionalidades todavía más potentes en su versión de paga. El código fuente se puede descargar de su sitio web <https://datatables.net>

El objetivo declarado de DataTables es "Mejorar la accesibilidad de los datos en tablas HTML". Para lograr esto, reconocemos que DataTables tiene dos categorías de usuarios que interactúan con las interfaces que tiene el software:

Usuarios finales: aquellos que usan la interfaz en el navegador que creas con DataTables.
Desarrolladores: los desarrolladores que trabajan con DataTables para crear sus aplicaciones, sitios, servicios, etc.

Los usuarios finales deben poder obtener información útil de la tabla lo más rápido posible y para esto, DataTables tiene características incorporadas como ordenar, buscar y paginar.

Los desarrolladores que usan DataTables, pueden usar DataTables en sus aplicaciones, personalizarla para que coincida con sus requisitos exactos, para permitir que sus usuarios finales se beneficien de las características que DataTables puede presentar.

La documentación de la herramienta cubre detalles de cómo instalar DataTables en su sitio en la primera instancia, explicando cómo puede obtener datos para las tablas que crea y detallando cómo se puede personalizar su interfaz de usuario final, hasta llegar a temas más avanzados. como crear sus propios complementos para DataTables para mejorar aún más sus capacidades y la interfaz sin problemas con su aplicación.

Principales funciones utilizadas en el proyecto:

Inicialización: Este tipo de inicialización hace que la tabla del carrito de compras se muestre sin paginación y mover de lugar la caja de texto para buscar.

```
table= $('#carrito').DataTable({
  "dom": '<"toolbar">frtip',
  "paging": false
});
```

Inicialización múltiple: Para inicializar más de una tabla a la vez, se puede hacer a través de la implementación de un selector de clase, como es en este ejemplo: la clase display.

```
$('#table.display').DataTable( );
```

Captura de eventos: De esta forma se captura el id de la fila seleccionada con el evento click del mouse.

```
$('#carrito').on( 'click', 'tr', function () {
  id = table.row( this ).id();
  actualizarCuadroResumen();
} );
```

Google maps

Es sólo HTML, CSS y JavaScript trabajando junto. Los mapas son solo imágenes que se cargan en el fondo a través de peticiones ejecutadas por la tecnología de AJAX, y se insertan en un <div> en la página HTML. Mientras navegas en el mapa, el API envía información acerca de las nuevas coordenadas y los niveles de "zoom" del mapa a través de AJAX y esto retorna las imágenes.

El API consiste de archivos JavaScript que contienen las clases, métodos y propiedades que se usan para el comportamiento de los mapas. Las coordenadas están expresadas usando números decimales separados por coma. La latitud siempre precede la longitud. La latitud es positiva si va después del punto mostrado en el mapa y negativo si va antes. La longitud es positiva si va arriba del punto y negativa si va debajo.

Estas son las funciones mas básicas necesarias e implementadas en el proyecto:

La obtención de las coordenadas se obtiene a través de una función oyente que trae la aplicación el cual le mandamos el mapa que creamos y esta nos devuelve un evento en una función anónima como parámetro un variable de tipo objeto que tiene como un parámetro un json que es latLng que es la longitud y latitud juntas

```
google.maps.event.addListener(map, 'click', function(event) {
    geocodeLatLng(geocoder, map, infoWindow, event.latLng);
});
```

Para obtener la localización primero comprobamos si el navegador tiene este soporte y mandamos a llamar la variable navigator.geolocation que nos devolverá un true si está disponible o un false en caso contrario la función anónima dentro de getCurrentPosition nos devuelve un variable que contiene como json la latitud y longitud

```
if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(function(position) {
        var pos = {
            lat: position.coords.latitude,
            lng: position.coords.longitude
        };
    });
}
```

Para calcular la distancia ente 2 coordenada en el mapa de google se utiliza el servicio de DistanceMatrixService que nos marcara con una ruta conveniente la unión de los 2 puntos

respetando las vías existentes y si se puede escoger que tipo de medio se utilizara para llegar del destino a él origen como en este caso en un auto se usa DRIVING.

```
var service = new google.maps.DistanceMatrixService();
    service.getDistanceMatrix(
    {
        origins: [puntoA],
        destinations: [puntoB],
        travelMode: 'DRIVING',
        unitSystem: google.maps.UnitSystem.METRIC,
        avoidHighways: false,
        avoidTolls: false,
    }, callback);
```

Facebook Login API

El inicio de sesión con Facebook para aplicaciones es un modo rápido y cómodo de crear cuentas e iniciar sesión en tu aplicación en varias plataformas. Está disponible para iOS, Android, sitios web, Windows Phone, aplicaciones para computadoras y dispositivos, como Smart TV y objetos de Internet de las cosas.

El Facebook login API posee 3 casos de uso que veremos a continuación:

- Creación de cuentas: El inicio de sesión con Facebook permite a las personas crear rápida y fácilmente una cuenta en tu aplicación sin tener que establecer (y posiblemente olvidar más tarde) una contraseña.
- Personalización: Las experiencias personalizadas son más atractivas y aumentan la retención. El inicio de sesión con Facebook te permite acceder a información que sería complicado o difícil obtener mediante tu propio formulario de registro; por ejemplo, las cosas que a una persona le gustan, su cumpleaños, su ciudad de origen, su ubicación actual o sus antecedentes laborales.
- Interacción social: Muchas aplicaciones con gran capacidad de retención permiten a las personas conectar con sus amigos para compartir sus experiencias dentro de la aplicación.

El proceso de revisión del inicio de sesión está diseñado para ser rápido y ligero. Una aplicación puede solicitar a las personas los tres permisos siguientes sin que sea necesario que Facebook la revise:

- public profile
- email
- user_friends

Si la aplicación solicita otro permiso, deberá someterse a la revisión de Facebook para que dicho permiso se muestre en el cuadro de diálogo de inicio de sesión con Facebook de tu aplicación.

Inicio de sesión con Facebook para web con el SDK para JavaScript

El inicio de sesión con Facebook con el SDK de Facebook para JavaScript permite a las personas iniciar sesión en tu página web con sus credenciales de Facebook.

Para implementar el inicio de sesión con Facebook con el SDK para JavaScript, necesitas un identificador de la aplicación de Facebook antes de empezar, que puedes crear y recuperar en el panel de aplicaciones. También necesitas algún sitio donde alojar archivos HTML.

Implementa el inicio de sesión con los siguientes pasos:

1. Elegir la configuración de tu aplicación

En el Panel de aplicaciones, selecciona tu aplicación. A continuación, elige si quieres permitir el inicio de sesión con SMS o correo electrónico y selecciona la configuración de seguridad para tu aplicación. Para obtener más información sobre cómo elegir el token de acceso, consulta Tokens de acceso y sobre cómo elegir la configuración de la clave secreta de tu aplicación, consulta Uso de la API Graph.

2. Ingresar la URL de redireccionamiento

En el panel de aplicaciones, selecciona tu aplicación y ve a Productos > Inicio de sesión con Facebook > Configuración. Ingresa la URL de redireccionamiento para obtener la autorización.

3. Comprobar el estado del inicio de sesión

El primer paso al cargar tu página web es determinar si una persona ya inició sesión en tu aplicación con el inicio de sesión con Facebook. Este proceso comienza con una llamada a `FB.getLoginStatus`. Esta función activa una llamada a Facebook para obtener el estado de inicio de sesión y llamar a tu función de devolución de llamada con el resultado.

4. Solicitar el inicio de sesión

Si la persona no inició sesión en Facebook, primero se le pedirá que lo haga y después se pasará al inicio de sesión en la aplicación. El SDK para JavaScript lo detecta automáticamente, de modo que no tienes que hacer nada para activar este comportamiento.

5. Cerrar las sesiones

Puedes cerrar la sesión de las personas en tu aplicación adjuntando la función `FB.logout` del SDK para JavaScript a un botón o enlace.

```

<script>

  // Esto es llamado con los resultados desde FB.getLoginStatus().

  function statusChangeCallback(response) {

    console.log('statusChangeCallback');

    console.log(response);

    // El objeto respuesta es retornado con un campo de estatus que permite
    // a la aplicacion el estado de login del usuario.

    if (response.status === 'connected') {

      // Has iniciado session en tu app y facebook

      testAPI();

    } else {

      // La persona no esta logueada en nuestra app o somos incapaces de
      decirlo.

      document.getElementById('status').innerHTML = 'Por favor logueate '
+
      'en esta app.';

    }

  }

  // Esta funcion es llamada cuando alguien presiona el botón de
  // login. Mira el manejador onlogin adjunto en el codigo
  // que te presentamos a continuacion.

  function checkLoginState() {

    FB.getLoginStatus(function(response) {

      statusChangeCallback(response);

    });

  }

```

```

window.fbAsyncInit = function() {

  FB.init({

    appId      : '{TU-ID-DE-APP}',

    cookie     : true,  // Activa las cookies para que el servidor pueda
acceder a la sesion

    xfbml      : true,  // parsea los plugins sociales en esta
                        pagina

    version    : 'v2.8' // Usa la graph api version 2.8

  });


  // Ahora que hemos inicializado el JavaScript SDK, llamamos

  // FB.getLoginStatus(). Esta funcion obtiene el estado de

  // la persona visitando esta pagina y puede retornar uno de tres estados
hacia

  // el callback que proporcionamos. Pueden ser:

  //

  // 1. Logueado en tu app ('connected')

  // 2. Logueado en facebook, pero no en tu app ('not_authorized')

  // 3. No esta logueado en facebook y tampoco se puede decir

  //    si en la app o no

  //

  // Estos tres casos son manejados en la funcion callback


  FB.getLoginStatus(function(response) {

    statusChangeCallback(response);

  });

};

```

```

// Carga el sdk de forma asincrona

(function(d, s, id) {

    var js, fjs = d.getElementsByTagName(s)[0];

    if (d.getElementById(id)) return;

    js = d.createElement(s); js.id = id;

    js.src = "//connect.facebook.net/en_US/sdk.js";

    fjs.parentNode.insertBefore(js, fjs);

})(document, 'script', 'facebook-jssdk');

// Aqui ejecutamos una pequeña prueba de Graph API después de que el
// logueo

// fue exitoso. Mira en statusChangeCallback() cuando esta llamada es
// hecha.

function testAPI() {

    console.log('Welcome! Fetching your information.... ');

    FB.api('/me', function(response) {

        console.log('Successful login for: ' + response.name);

        document.getElementById('status').innerHTML =

            'Thanks for logging in, ' + response.name + '!';

    });

}

</script>

```


Observaciones:

Esta API solo funciona cuando se utiliza php de backend, ya que es una API en el cual al configurar la app en Facebook developers necesita de una URL para saber dónde está la app. Con nodeJS como backend esta API resulta inútil ya que la app en Facebook por alguna razón no puede interpretar la URL y por consiguiente el SDK no puede cargar.

Para nodeJS es necesario utilizar PassportJS, es un middleware que nos permite manejar la sesión de Facebook desde el servidor, la implementación es más compleja que en el SDK de Javascript pero es más segura. Para una correcta implementación en necesario AngularJS con su modulo Factory que nos facilita el manejo de los tokens generados por los métodos de autenticación de Facebook, También nos permite autorizar a un usuario ya existente en nuestra base de datos para utilizar la autenticación por Facebook. A diferencia del SDK de javascript, no necesitamos una URL, solo necesitamos saber la versión de la Graph API, la ID de nuestra app y la secret-key que la proporciona facebook con nuestra app.

X Glosario de términos:

- **Cross docking:** Su intención es también acordar con los proveedores una preparación de la mercancía para reducir costes de recepción, almacenaje y preparación.
- **delivery:** es una actividad parte de la función logística que tiene por finalidad colocar bienes, servicios e información directo en el lugar de consumo (Al cliente final).
- **Disrupción:** la configuración de una nueva oferta de valor que es brindada por un oferente que no es el líder del momento, pero que presenta prestaciones sustantivamente mejores para el consumidor final.
- **e-commerce** consiste en la distribución, venta, compra, marketing y suministro de información de productos o servicios a través de Internet.
- **empresa virtual** es una nueva forma de organización funcional, que se describe entorno a la evolución dinámica de la tecnología y de la globalización. Su estructura organizativa se apoya en el uso de Internet para ampliar sus transacciones y operaciones en el mercado del ciberespacio.
- **Intermediario:** Que media entre dos o más partes para que lleguen a un acuerdo en un negocio o problema.
- **marketing digital** es la aplicación de las estrategias de comercialización llevadas a cabo en los medios digitales. Todas las técnicas del mundo off-line son imitadas y traducidas a un nuevo mundo, el mundo online. En el ámbito digital aparecen nuevas herramientas como la inmediatez, las nuevas redes que surgen día a día, y la posibilidad de mediciones reales de cada una de las estrategias empleadas.
- **MiPyMe:** La pequeña y mediana empresa es una empresa con características distintivas, y tiene dimensiones con ciertos límites ocupacionales y financieros prefijados por los Estados o regiones. Las pymes son agentes con lógicas, culturas, intereses y un espíritu emprendedor específicos. También existe el término MiPyME (acrónimo de "micro, pequeña y mediana empresa"), que es una expansión del término original, en donde se incluye a la microempresa.
- **retail** es un término de la lengua inglesa que se emplea para nombrar a la venta minorista. La comercialización de productos al por menor, por lo tanto, constituye el retail.
- **startup** compañía de nueva creación que tienen unas grandes posibilidades de crecimiento y, en muchas ocasiones, cuenta con un modelo de negocio escalable. A pesar de que el término de startup pueda hacer referencia a compañías de cualquier ámbito o sector, lo más habitual es que haga alusión a las empresas que poseen un fuerte componente tecnológico y que están vinculadas al mundo de Internet y las TICs.

XI Evaluación de equipo

Nombre	Autoevaluación (100%)
Allan Noel López	95%
Jesus Daniel Arias	95%
Oscar Andrés Vasquez	95%
Saúl Guzmán	88%