Proyecto IPC2

Fase 1

Shubert Alexander Alonzo Agustin
201503971
12/6/2018

Definición de la solución

Objetivos

General

Crear un software ERP que cumpla con los requisitos expuestos por el cliente(Ninty), creando módulos para que el sistema pueda ser comercializado.

Específicos

1. Crear un sistema de gestión de plataforma para poder cambiar precios.
2. Crear un sistema de gestión para el cliente que adquiera la plataforma pueda manejar usuarios y módulos.
3. Crear módulos relacionados entre sí para generar un sistema automatizado.
4. Crear un sistema para gestionar inventario y productos.
5. Crear un sistema para gestionar compras, ventas y facturación.
6. Crear un sistema para control de empleados y manejo de flotas.
7. Crear un sistema CRM.
8. Crear un sistema para gestionar eventos y anuncios.

Alcance del proyecto

Será un software modular capaz de satisfacer las necesidades de la mayoría de PYMES que buscan una solución más barata que un software a la medida, y por lo tanto podrán contratar solo los módulos que les sean necesarios.

Este software es escalable, por lo tanto da la oportunidad que la empresa (Ninty) en un futuro agregue más módulos a una tienda virtual que tendrá el software, teniendo así la posibilidad de ofrecer a sus clientes una solución a necesidades futuras.

Panorama general de la aplicación

El software da solución a una necesidad que tienen las PYMES, que es encontrar una herramienta para su control a un precio accesible, ya que normalmente el software que estas necesitan se realiza a la medida, teniendo un precio más elevado.

Es por eso que este software debe de ser modular para darle la opción a los contratantes de elegir solo los módulos necesarios para su negocio. Esto ofrece diferentes ventajas tanto a los contratantes como a los vendedores del software.

Los principales módulos serán:
Un módulo para gestión de inventario, este módulo permitirá llevar un control de los productos que maneja la empresa, como agregarlos, eliminarlos, o editar la información de estos.

Un módulo para gestionar las ventas, este módulo permitirá llevar un control de las ventas que la empresa realiza guardando toda la información de las mismas.

Un módulo para gestionar las compras, este módulo permitirá llevar un control de las compras realizadas a proveedores y los ingresos que estas producirán.

Un módulo de facturación, este módulo permitirá facturar todas las ventas que la empresa realizó, si el usuario tiene tambien el modulo de ventas e inventario, se podrá generar una factura automática por cada venta, y de igual forma el inventario se actualiza.

Un módulo de reclutamiento, este módulo le será de utilidad al área de recursos humanos de la empresa, automatizando los procesos de selección de personal y teniendo acceso a la información de los empleados actuales.

Un módulo de gestión de flotas, este módulo será de utilidad a empresas que hacen entregas y requieren llevar un control de sus pilotos, vehículos y paquetes enviados.

Un módulo de blog, este módulo será de utilidad para hacer anuncios o avisos para los trabajadores de la empresa, teniendo estos la capacidad de comentarlos.

Un módulo de eventos, este módulo será de utilidad para llevar un control de los eventos que la empresa realiza.

Un módulo CRM, este módulo permitirá proyectar oportunidades de negocios de acuerdo a cotizaciones que se hayan realizado, con esto la empresa podrá mejorar sus ofertas para cerrar una venta.

Un módulo para registro de asistencia, este módulo permitirá llevar un control de los días y horas que el empleado trabaja, y se podrá comparar con el horario en el que el empleado está contratado.

Los usuarios serán empresas, en especial PYMES, que necesiten un software para llevar un control interno y acelerar su crecimiento.

Los usuarios serán los siguientes:

- un administrador de parte de Ninty para llevar un control de las empresas que contrataron el servicio y los módulos que adquirieron.
- un administrador de parte de la empresa que contrató el servicio para poder crear usuarios y adquirir más módulos.
- usuarios creados por el administrador de la empresa para acceder a las funciones de los módulos.

Requerimientos iniciales del sistema

Requerimientos funcionales

Ref. #	Funcion	Categoría
1	El sistema debe ser capaz de gestionar usuarios, es decir, crear, modificar y eliminar estos.	Evidente
2	El sistema debe permitir utilizar el módulo tienda a los clientes que adquirieron el software.	Evidente
3	Una vez ya adquiridos el sistema debe permitir cancelar la suscripción a los módulos adquiridos por el cliente si éste así lo desea.	Evidente
4	Una vez adquirido el sistema debe permitir gestionar módulo Inventario	Evidente
5	Una vez adquirido el sistema debe permitir Gestionar módulo Ventas	Evidente
6	Una vez adquirido el sistema debe permitir Gestionar módulo Compras	Evidente
7	Una vez adquirido el sistema debe permitir Gestionar módulo Facturar	Evidente

8	Una vez adquirido el sistema debe permitir Gestionar módulo Reclutar Personal	Evidente
9	Una vez adquirido el sistema debe permitir Administrar Personal	Evidente
10	Una vez adquirido el sistema debe permitir Gestionar módulo Flotas	Evidente
11	Una vez adquirido el sistema debe permitir Gestionar módulo Blog	Evidente
12	Una vez adquirido el sistema debe permitir Gestionar módulo Eventos	Evidente
13	Una vez adquirido el sistema debe permitir Controlar módulo CRM	Evidente
14	Una vez adquirido el sistema debe permitir Gestionar módulo Asistencia	Evidente
15	El sistema debe debitar automáticamente el cargo mensual al que el cliente está suscrito.	Oculto

Atributos del sistema

Usabilidad
● El sistema se desarrollara con una arquitectura de 3 capas.

- Será una aplicación web responsive con el fin de garantizar una buena visualización en los diferentes dispositivos.
- El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
- El sistema debe ser modular.

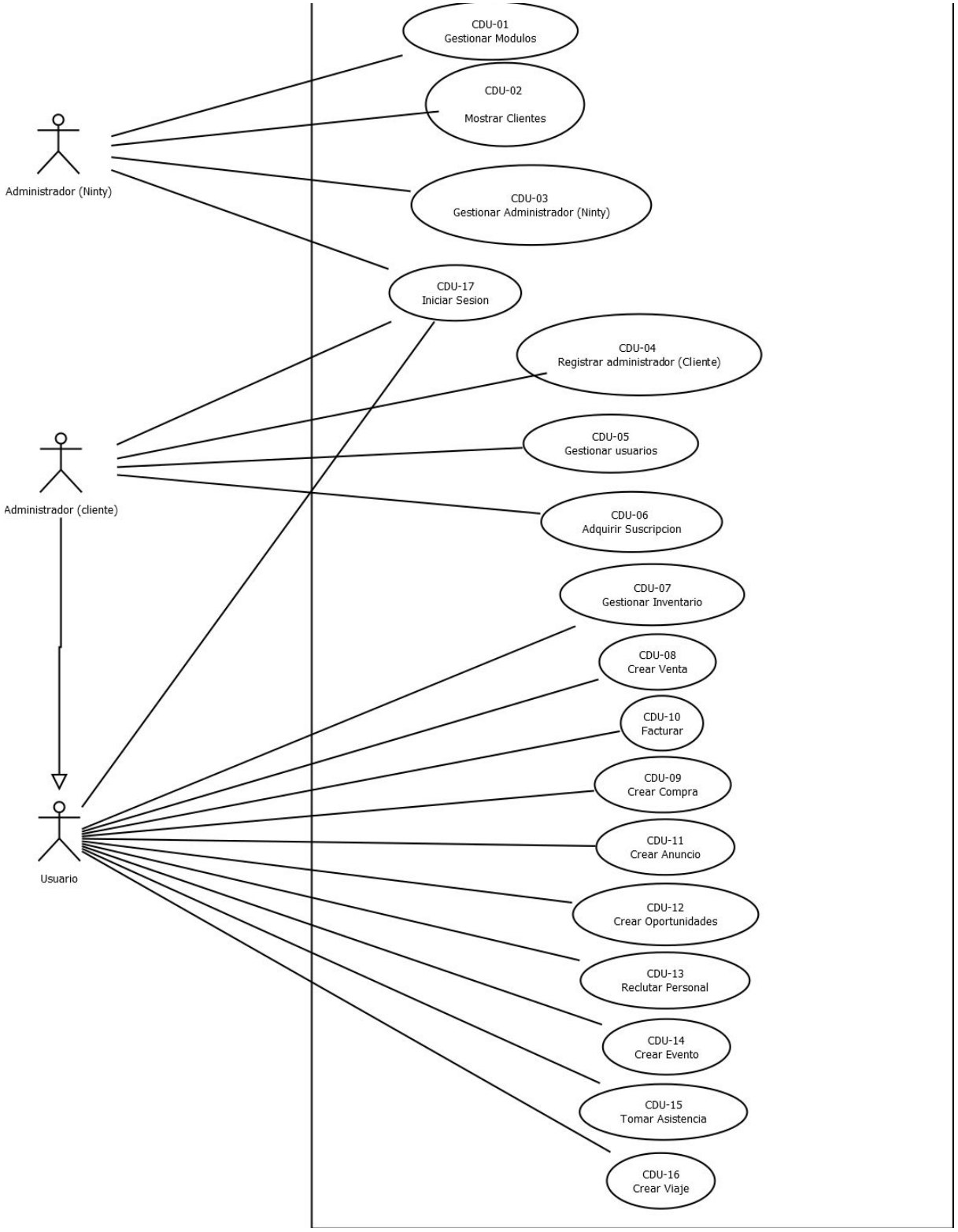
Definición de Clientes de la aplicación

Los clientes serán empresas en especial PYMES que necesiten un software para llevar un control interno y acelerar su crecimiento.

Cliente	Descripción	Módulos
Super Administrador	Este será un usuario para administrar precios, y llevar un control de las entidades que adquirieron el software.	Consulta de Datos, Manejo de Precios
Administrador	Este usuario podrá crear más usuarios que tengan acceso a los módulos que este adquiera.	Creación de usuarios, Gestion de Modulos
Usuario	Este usuario podrá tener acceso a los módulos que el administrador que lo creó tenga acceso.	Ingresar información al sistema.

Casos de uso

Alto Nivel



Caso de uso	CDU-01 Gestionar Módulos
Actores	Administrador(Ninty)
Tipo	Primario
Descripción	El administrador podrá habilitar y deshabilitar los módulos de la plataforma, también podrá modificar los precios de estos.

Caso de uso	CDU-02 Gestionar Clientes
Actores	Administrador(Ninty)
Tipo	Primario
Descripción	El administrador podrá tener un control sobre los clientes que adquirieron el software y los pagos que realizan.

Caso de uso	CDU-03 Gestionar Administrador(Ninty)
Actores	Administrador(Ninty)
Tipo	Opcional
Descripción	El administrador puede crear otros administradores si fuera necesario.

Caso de uso	CDU-04 Registrarse a la plataforma
Actores	Administrador(cliente)
Tipo	Primario
Descripción	Un cliente procede a registrarse a la plataforma para tener acceso al software, convirtiéndose en un administrador.

Caso de uso	CDU-05 Gestionar Usuarios
Actores	Administrador(cliente)

Tipo	Primario
Descripción	Un administrador podrá crear, modificar y eliminar cuentas que podrán usar los módulos que este adquirió.

Caso de uso	CDU-06 Gestionar Suscripción
Actores	Administrador(Ninty)
Tipo	Primario
Descripción	Un administrador podrá adquirir más módulos, o cancelar su suscripción.

Caso de uso	CDU-07 Gestionar Inventario
Actores	Administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver o modificar inventarios.

Caso de uso	CDU-08 Gestionar Ventas
Actores	Administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver o registrar ventas.

Caso de uso	CDU-09 Gestionar Compras
Actores	Administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver o registrar compras.

Caso de uso	CDU-10 Facturar
Actores	Administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver o crear facturas en el sistema.

Caso de uso	CDU-11 Gestionar Blog
Actores	Administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver, crear y comentar publicaciones en el blog.

Caso de uso	CDU-12 Gestionar CRM
Actores	Administrador(cliente), usuario
Tipo	Opcional
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver, modificar o crear las oportunidades que se gestionan en este módulo.

Caso de uso	CDU-13 Reclutar
Actores	Administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede tener acceso a la información de los empleados de la empresa y agendar entrevistas de trabajo.

Caso de uso	CDU-14 Gestionar Eventos

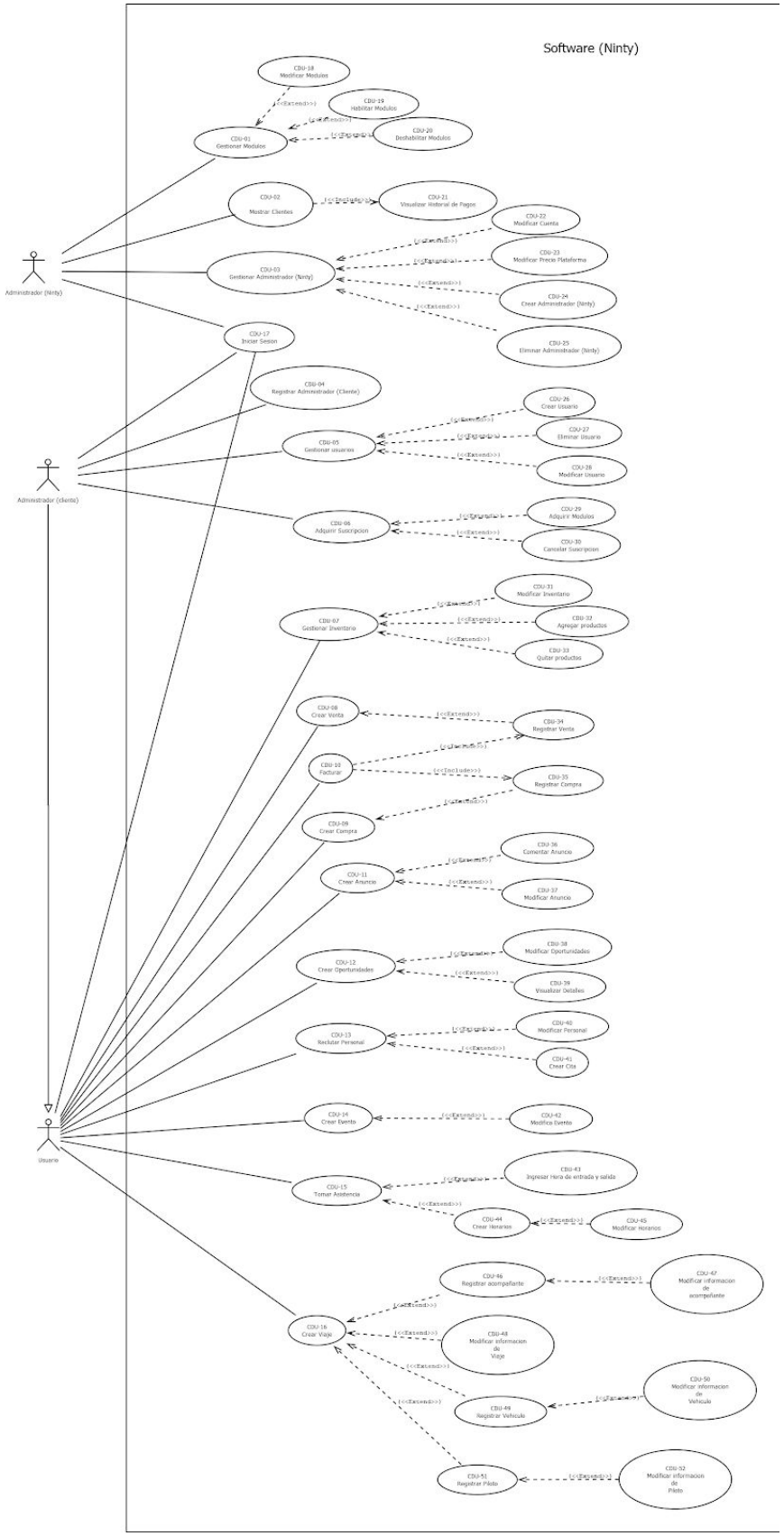
Actores	Administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver o agendar eventos.

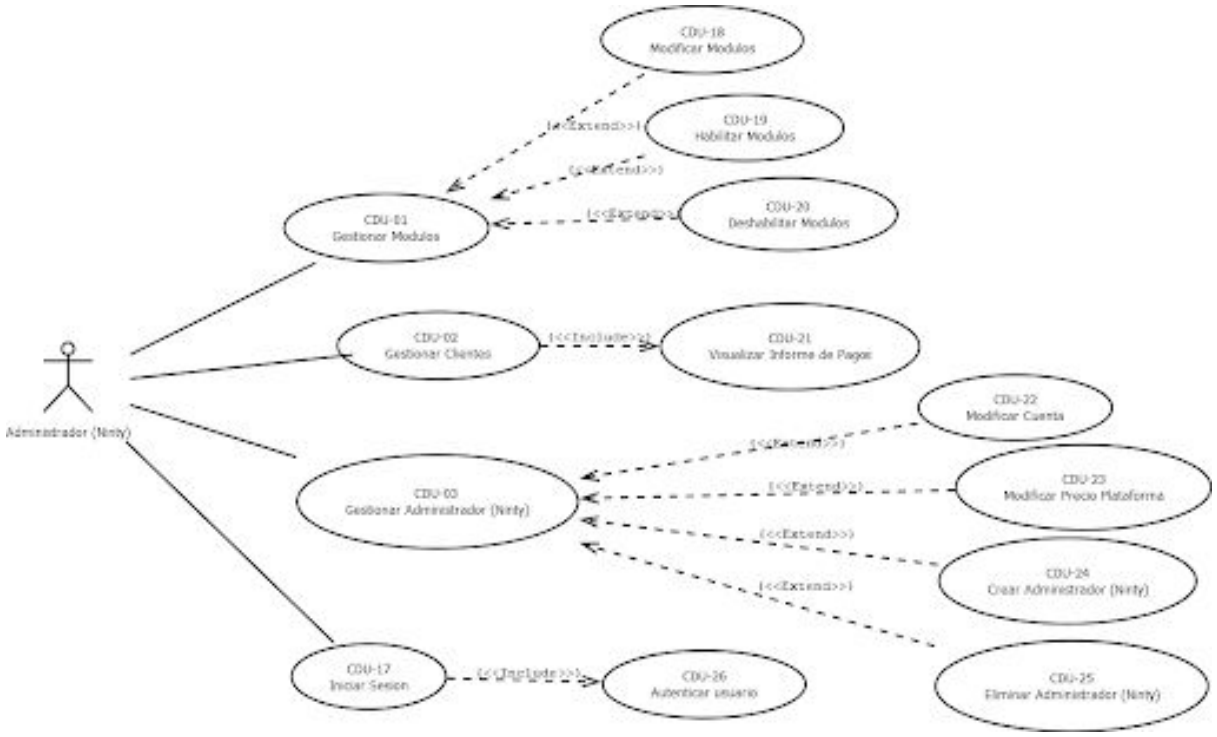
Caso de uso	CDU-15 Gestionar Flota
Actores	Administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver, modificar y agregar información de la flota de vehículos que la empresa maneja.

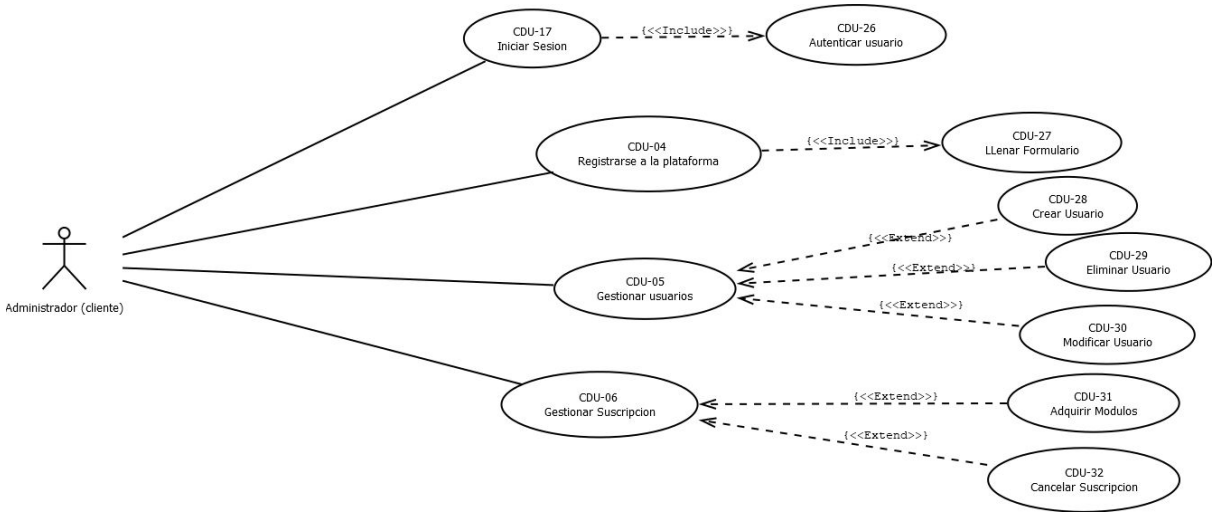
Caso de uso	CDU-16 Gestionar Asistencia
Actores	Administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver la asistencia de cada empleado así como generar horarios de trabajo.

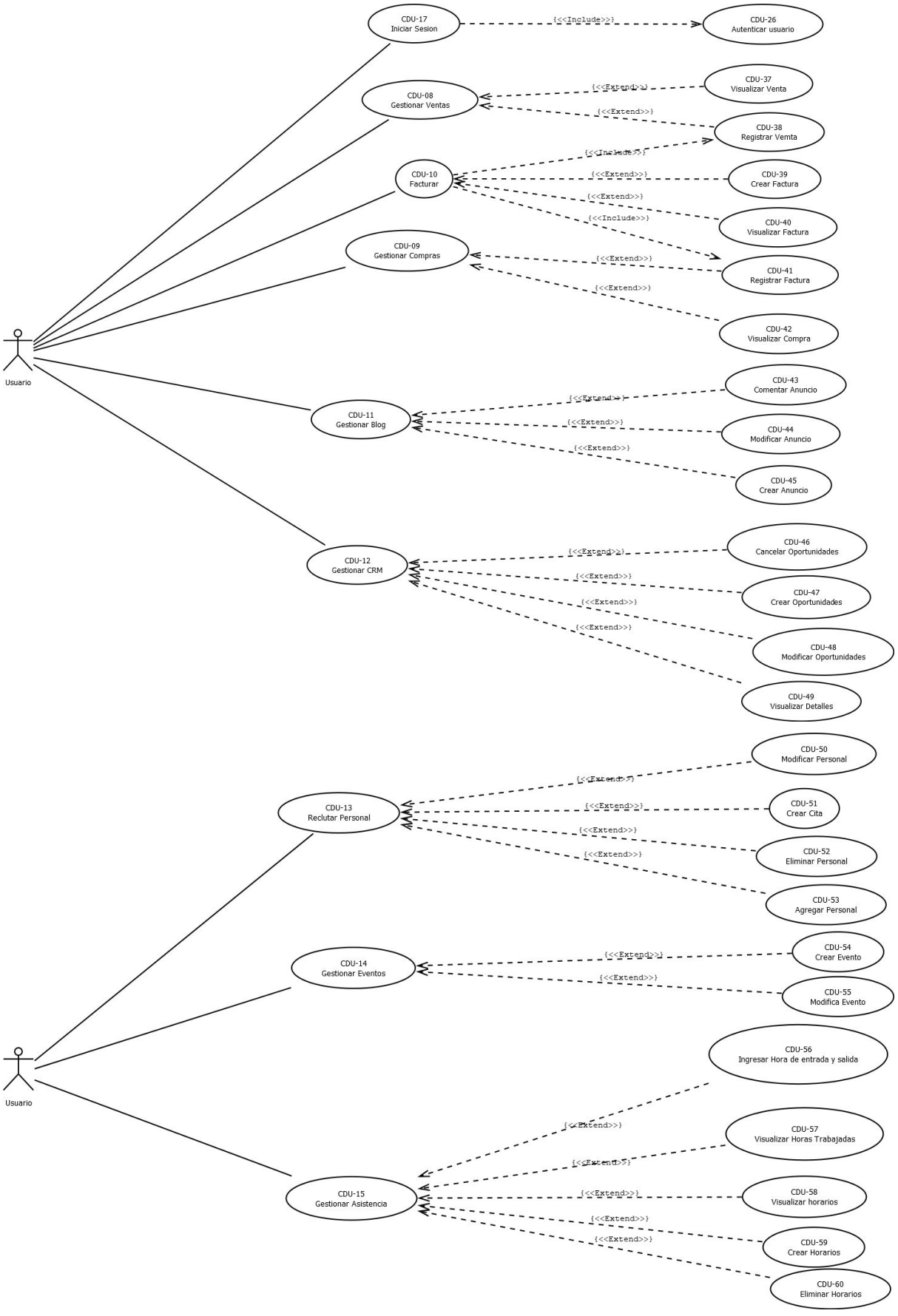
Caso de uso	CDU-17 Iniciar Sesión
Actores	Administrador(Ninty, administrador(cliente), usuario
Tipo	Primario
Descripción	Si el administrador está suscrito a este módulo entonces cualquier usuario que este haya creado puede ver la asistencia de cada empleado así como generar horarios de trabajo.

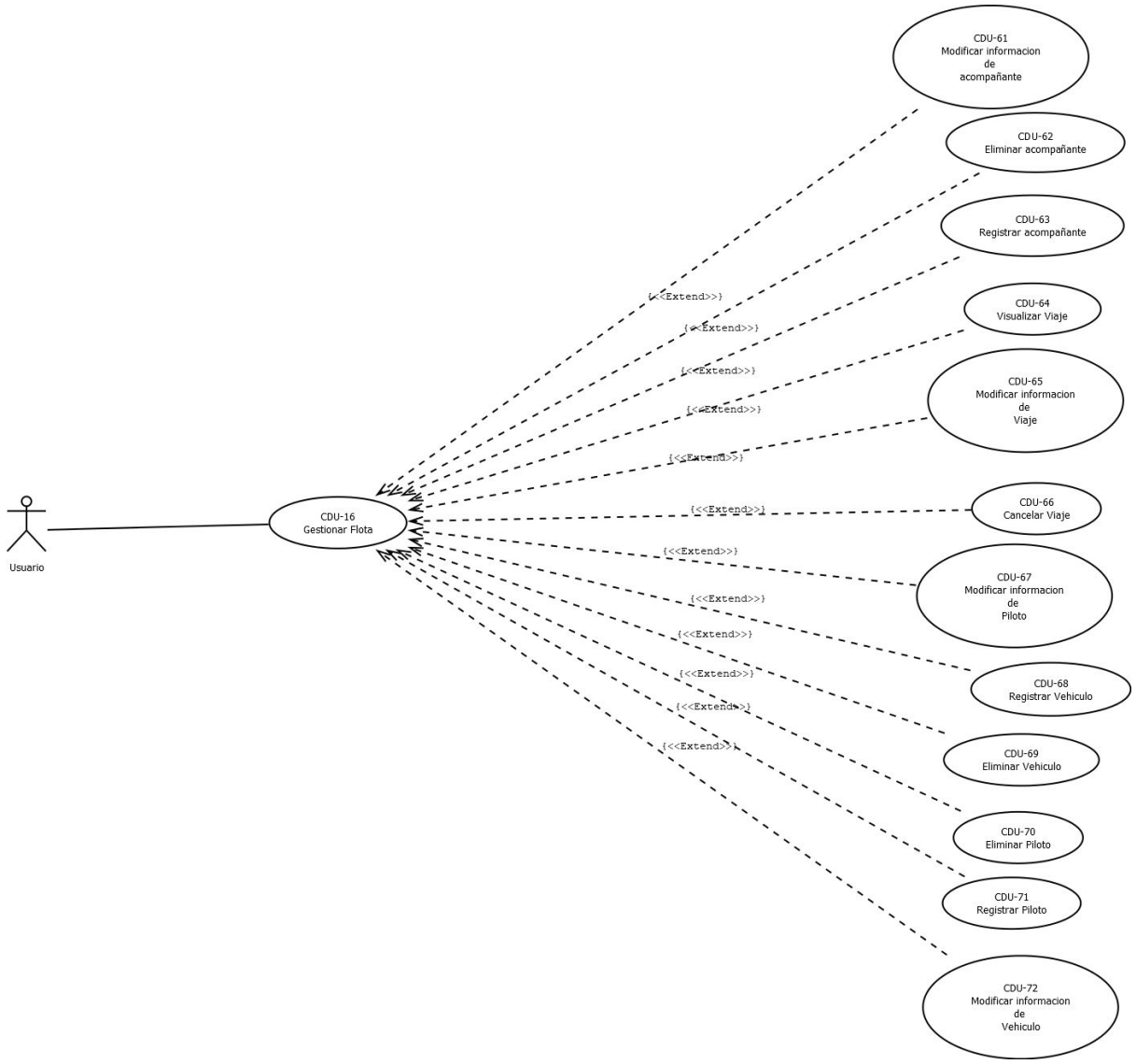
Expandidos


Software (Ninty)
CDU-18 Modificar Modulos
CDU-19 Habilitar Modulos
CDU-01 Gestionar Modulos
CDU-20 Deshabilitar Modulos
CDU-02 Mostrar Clientes
CDU-21 Visualizar Historial de Pagos
CDU-22 Modificar Cuenta
CDU-23 Modificar Precio Plataforma
CDU-03 Gestionar Administrador (Ninty)
CDU-24 Crear Administrador (Ninty)
CDU-25 Eliminar Administrador (Ninty)
Administrador (Ninty)
CDU-17 Iniciar Sesion
CDU-04 Registrar Administrador (Cliente)
CDU-26 Crear Usuario
CDU-27 Eliminar Usuario
CDU-05 Gestionar usuarios
CDU-28 Modificar Usuario
Administrador (cliente)
CDU-06 Adquirir Suscripcion
CDU-29 Adquirir Modulos
CDU-30 Cancelar Suscripcion
CDU-31 Modificar Inventario
CDU-07 Gestionar Inventario
CDU-32 Agregar productos
CDU-33 Quitar productos
CDU-08 Crear Venta
CDU-34 Registrar Venta
CDU-10 Facturar
CDU-35 Registrar Compra
CDU-09 Crear Compra
CDU-36 Comentar Anuncio
CDU-11 Crear Anuncio
CDU-37 Modificar Anuncio
CDU-38 Modificar Oportunidades
CDU-12 Crear Oportunidades
CDU-39 Visualizar Detalles
CDU-40 Modificar Personal
CDU-13 Reclutar Personal
CDU-41 Crear Cita
Usuario
CDU-14 Crear Evento
CDU-42 Modifica Evento
CDU-43 Ingresar Hora de entrada y salida
CDU-15 Tomar Asistencia
CDU-44 Crear Horarios
CDU-45 Modificar Horarios
CDU-46 Registrar acompañante
CDU-47 Modificar informacion de acompañante
CDU-16 Crear Viaje
CDU-48 Modificar informacion de Viaje
CDU-50 Modificar informacion de Vehiculo
CDU-49 Registrar Vehiculo
CDU-51 Registrar Piloto
CDU-52 Modificar informacion de Piloto


CDU-18
Modificar Modulos
CDU-19
Habilitar Modulos
CDU-20
Deshabilitar Modulos
CDU-01
Gestionar Modulos
CDU-02
Gestionar Clientes
CDU-21
Visualizar Informe de Pagos
Administrador (Ninty)
CDU-22
Modificar Cuenta
CDU-03
Gestionar Administrador (Ninty)
CDU-23
Modificar Precio Plataforma
CDU-24
Crear Administrador (Ninty)
CDU-25
Eliminar Administrador (Ninty)
CDU-17
Iniciar Sesion
CDU-26
Autenticar usuario


CDU-17
Iniciar Sesion
{<<Include>>}
CDU-26
Autenticar usuario
CDU-04
Registrarse a la plataforma
{<<Include>>}
CDU-27
LLenar Formulario
CDU-28
Crear Usuario
{<<Extend>>}
CDU-29
Eliminar Usuario
{<<Extend>>}
Administrador (cliente)
CDU-05
Gestionar usuarios
{<<Extend>>}
CDU-30
Modificar Usuario
CDU-06
Gestionar Suscripcion
{<<Extend>>}
CDU-31
Adquirir Modulos
{<<Extend>>}
CDU-32
Cancelar Suscripcion


Usuario
CDU-17
Iniciar Sesion
{<<Include>>}
CDU-26
Autenticar usuario
CDU-08
Gestionar Ventas
{<<Extend>>}
CDU-37
Visualizar Venta
{<<Extend>>}
CDU-38
Registrar Vemta
CDU-10
Facturar
{<<Include>>}
{<<Extend>>}
CDU-39
Crear Factura
{<<Extend>>}
CDU-40
Visualizar Factura
{<<Include>>}
CDU-09
Gestionar Compras
{<<Extend>>}
CDU-41
Registrar Factura
{<<Extend>>}
CDU-42
Visualizar Compra
CDU-11
Gestionar Blog
{<<Extend>>}
CDU-43
Comentar Anuncio
{<<Extend>>}
CDU-44
Modificar Anuncio
{<<Extend>>}
CDU-45
Crear Anuncio
CDU-12
Gestionar CRM
{<<Extend>>}
CDU-46
Cancelar Oportunidades
{<<Extend>>}
CDU-47
Crear Oportunidades
{<<Extend>>}
CDU-48
Modificar Oportunidades
{<<Extend>>}
CDU-49
Visualizar Detalles
Usuario
CDU-13
Reclutar Personal
{<<Extend>>}
CDU-50
Modificar Personal
{<<Extend>>}
CDU-51
Crear Cita
{<<Extend>>}
CDU-52
Eliminar Personal
{<<Extend>>}
CDU-53
Agregar Personal
CDU-14
Gestionar Eventos
{<<Extend>>}
CDU-54
Crear Evento
{<<Extend>>}
CDU-55
Modifica Evento
CDU-15
Gestionar Asistencia
{<<Extend>>}
CDU-56
Ingresar Hora de entrada y salida
{<<Extend>>}
CDU-57
Visualizar Horas Trabajadas
{<<Extend>>}
CDU-58
Visualizar horarios
{<<Extend>>}
CDU-59
Crear Horarios
{<<Extend>>}
CDU-60
Eliminar Horarios



Caso de uso	CDU-18 Modificar Módulos
Actores	Administrador (Ninty)
Propósito	Cambiar precio y nombre de los módulos
Resumen	El administrador entra al manejo de la plataforma y selecciona el módulo que desea modificar.
Tipo	opcional y esencial
Referencia Cruzadas	CDU-01
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El administrador selecciona la	1. El sistema muestra la información

opción Modificar Módulos 2. El administrador selecciona el módulo que desea modificar 3. Cambia la información correspondiente y presiona “guardar cambios”	de los módulos disponibles en el sistema 2. El sistema permite modificar el nombre y precio del módulo 3. El sistema guarda los cambios en la base de datos
Cursos Alternos	
3. La información ingresada no es válida, se muestra un mensaje indicando los errores y se repite el proceso hasta que los datos sean válidos.	

Caso de uso	CDU-19 Habilitar Módulos
Actores	Administrador (Ninty)
Propósito	Cambiar el estado de los módulos a activo
Resumen	El administrador entra al manejo de la plataforma y selecciona el módulo que desea habilitar
Tipo	Secundario y esencial
Referencia Cruzadas	CDU-01, CDU-20
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario selecciona mostrar módulos 2. El usuario selecciona el módulo que desea habilitar	1. El sistema muestra los módulos 2. El sistema cambia el estado del módulo a habilitado
Cursos Alternos	
2. El modulo ya esta habilitado, el sistema conduce al usuario a CDU-20 deshabilitar módulos.	

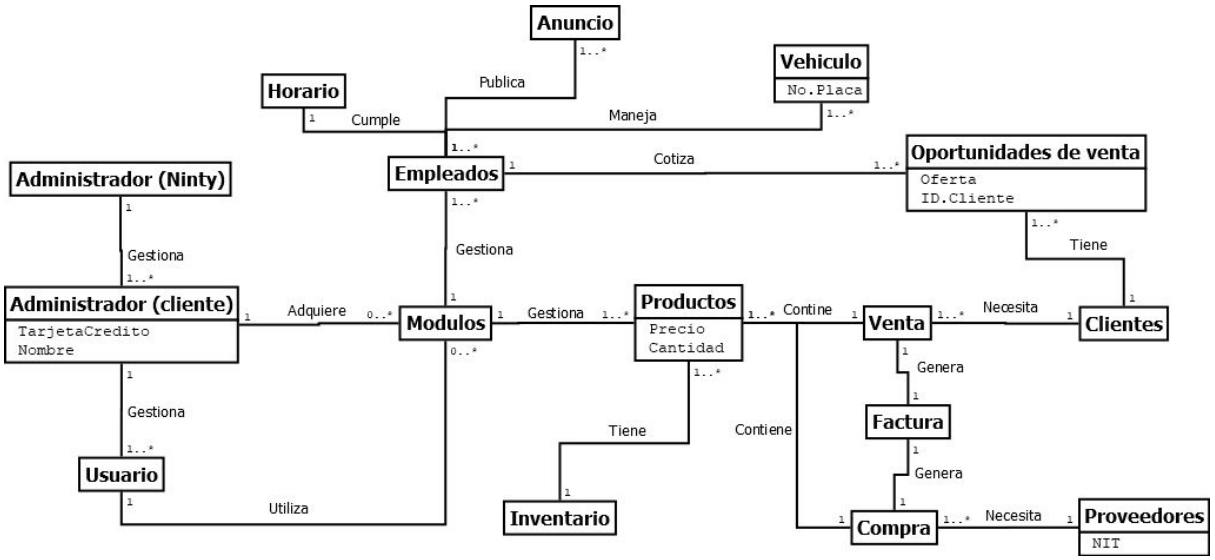
Caso de uso	CDU-20 Deshabilitar Módulos
Actores	Administrador (Ninty)
Propósito	Cambiar el estado de los módulos a inactivo

Resumen	El administrador entra al manejo de la plataforma y selecciona el módulo que desea deshabilitar
Tipo	Secundario y esencial
Referencia Cruzadas	CDU-01, CDU-19
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El usuario selecciona mostrar módulos 2. El usuario selecciona el módulo que desea deshabilitar	1. El sistema muestra los módulos 2. El sistema cambia el estado del módulo a deshabilitado
Cursos Alternos	
2. El modulo ya esta deshabilitado, el sistema conduce al usuario a CDU-19 habilitar módulos.	

Caso de uso	CDU-21 Visualizar informe de pagos
Actores	Administrador (Ninty)
Propósito	Visualizar el historial de transacciones de los clientes
Resumen	El administrador entra a la gestión de clientes y selecciona al cliente sobre el cual quiere obtener información
Tipo	Primario y esencial
Referencia Cruzadas	CDU-02
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El administrador de la plataforma selecciona gestionar clientes 2. El administrador de la plataforma selecciona el cliente sobre el que desea obtener información	1. El sistema muestra una lista de clientes 2. El sistema muestra el historial de pagos del cliente
Cursos Alternos	
1. La lista no se carga completamente, la página se vuelve a cargar.	

Caso de uso	CDU-22 Modificar Cuenta
Actores	Administrador (Cliente)
Propósito	Cambiar la información de las cuentas que el cliente creo
Resumen	El administrador selecciona la cuenta de usuario(empleado) que creó para cambiar la información de esta.
Tipo	Secundario y esencial
Referencia Cruzadas	CDU-03
Curso normal de los eventos	
Acción de los actores	Respuesta del sistema
1. El cliente con usuario de administrador entra a la opción de gestionar usuarios 2. Selecciona el usuario que desea modificar 3. Cambia la información y presiona aceptar	1. El sistema muestra los usuarios que el administrador ha creado 2. El sistema muestra la información del usuario 3. El sistema actualiza la información en la base de datos
Cursos Alternos	
3. La información es inválida, el sistema muestra un mensaje de error y regresa al paso 3.	

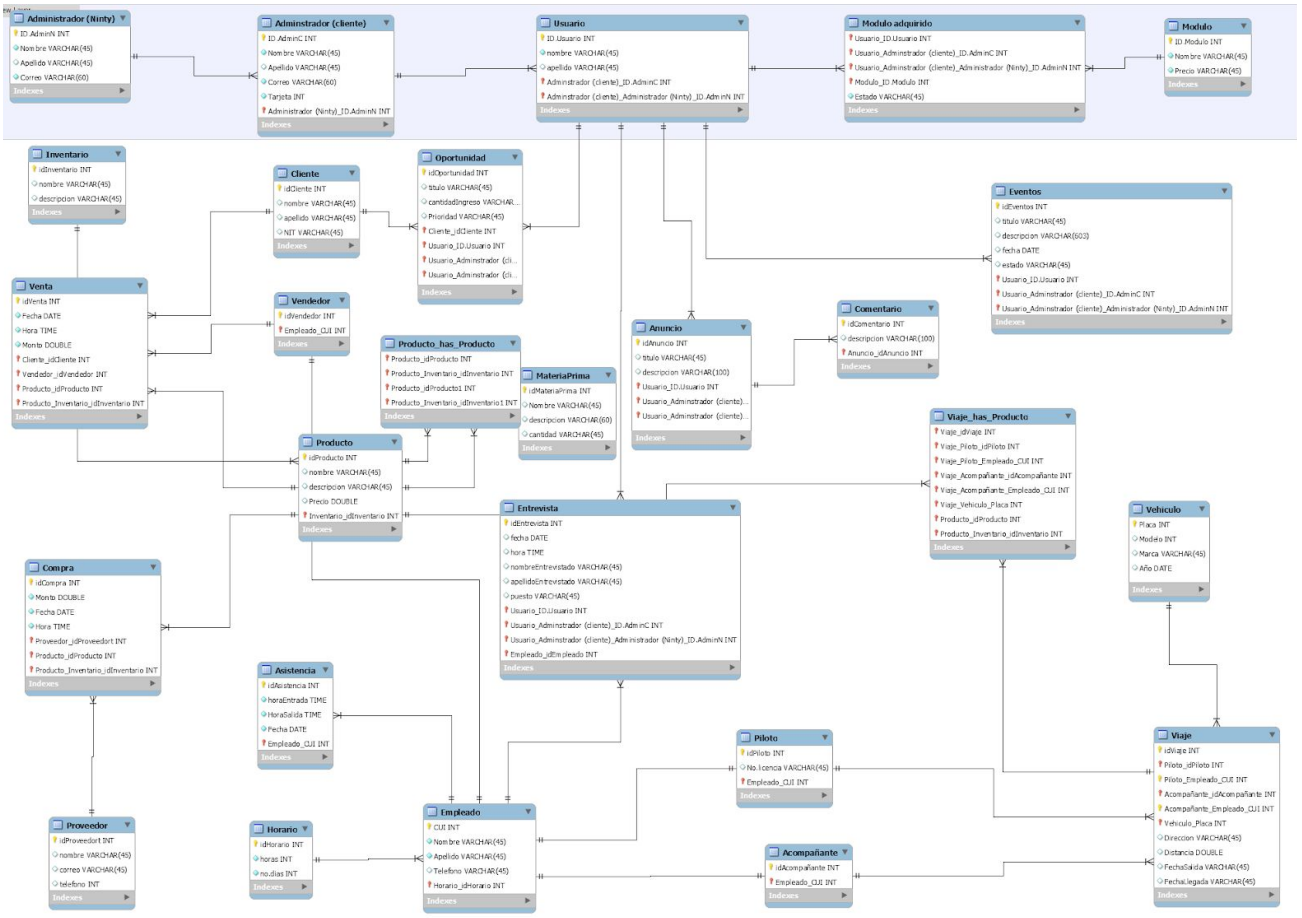
Modelo Conceptual



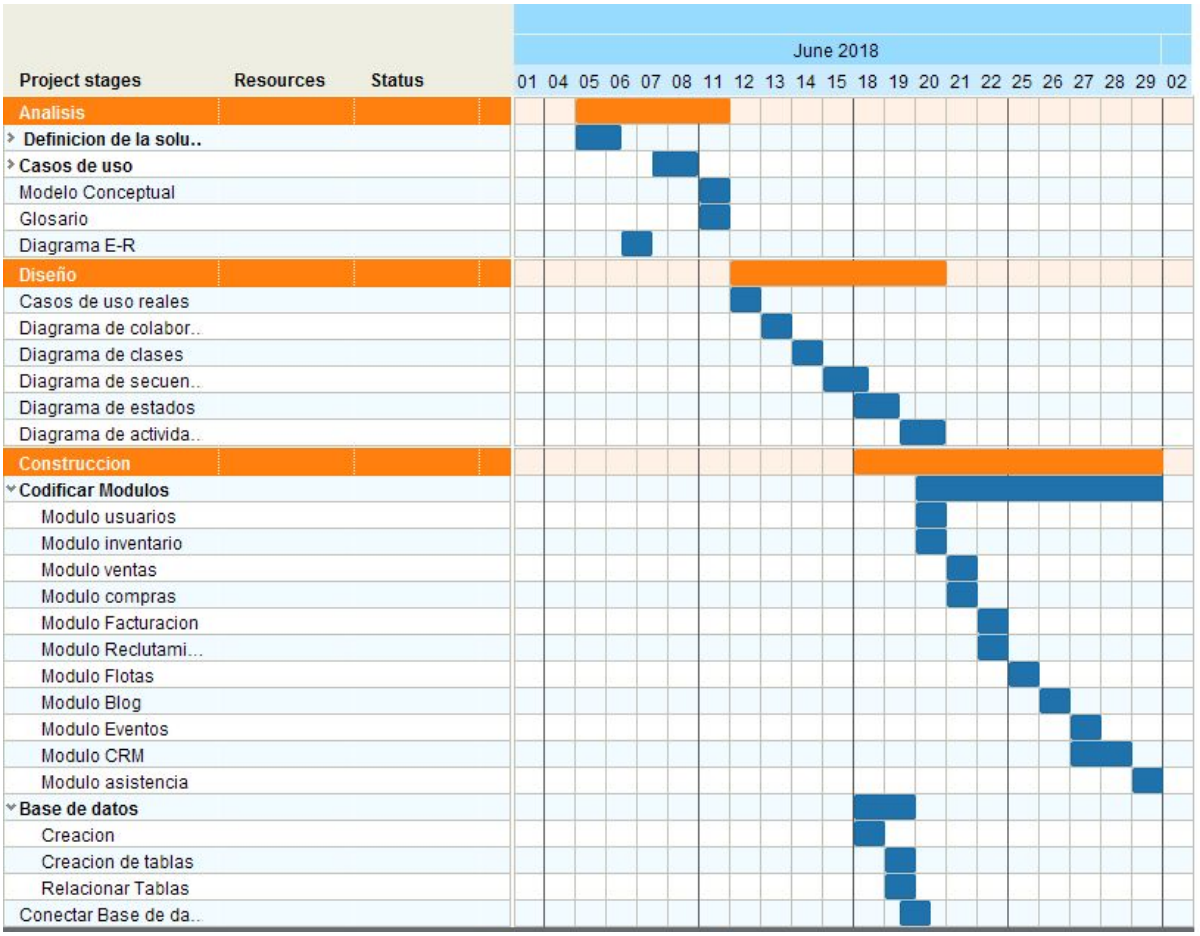
Glosario de definiciones técnicas

Acción: Ejercicio de la posibilidad de hacer.
Administrador: Persona que administra bienes ajenos.
CRM:Gestión de relaciones con los clientes) es un término de la industria de la información que se aplica a metodologías, software y, en general, a las capacidades de Internet que ayudan a una empresa a gestionar las relaciones con sus clientes de una manera organizada.
ERP: (Enterprise Resource Planning – Planificación de Recursos Empresariales) es un conjunto de sistemas de información que permite la integración de ciertas operaciones de una empresa, especialmente las que tienen que ver con la producción, la logística, el inventario, los envíos y la contabilidad.
Gestionar: Ocuparse de la administración, organización y funcionamiento de una empresa, actividad económica u organismo.
Módulo: Elemento con función propia concebido para poder ser agrupadas de distintas maneras con otros elementos constituyendo una unidad mayor.
PYME: Empresa pequeña o mediana en cuanto a volumen de ingresos, valor del patrimonio y número y trabajadores.
Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.
Usuario: Dicho de una persona: Que tiene derecho de usar de una cosa ajena con cierta limitación.

Diagrama Entidad-Relación



Planificación



Script creación de base de datos

```
CREATE TABLE IF NOT EXISTS `mydb`.`Administrador (Ninty)` (
  `ID.AdminN` INT NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `Apellido` VARCHAR(45) NULL,
  `Correo` VARCHAR(60) NOT NULL,
  PRIMARY KEY (`ID.AdminN`),
  UNIQUE INDEX `ID.AdminN_UNIQUE` (`ID.AdminN` ASC))
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Adminstrador (cliente)` (
  `ID.AdminC` INT NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `Apellido` VARCHAR(45) NULL,
```

```
  `Correo` VARCHAR(60) NOT NULL,
  `Tarjeta` INT NOT NULL,
  `Administrador (Ninty)_ID.AdminN` INT NOT NULL,
  PRIMARY KEY (`ID.AdminC`, `Administrador (Ninty)_ID.AdminN`),
  UNIQUE INDEX `ID.AdminC_UNIQUE` (`ID.AdminC` ASC),
  INDEX `fk_Adminstrador (cliente)_Administrador (Ninty)_idx` (`Administrador
(Ninty)_ID.AdminN` ASC),
  CONSTRAINT `fk_Adminstrador (cliente)_Administrador (Ninty)`
    FOREIGN KEY (`Administrador (Ninty)_ID.AdminN`)
    REFERENCES `mydb`.`Administrador (Ninty)` (`ID.AdminN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Usuario` (
  `ID.Usuario` INT NOT NULL,
  `nombre` VARCHAR(45) NOT NULL,
  `apellido` VARCHAR(45) NULL,
  `Adminstrador (cliente)_ID.AdminC` INT NOT NULL,
  `Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN` INT NOT NULL,
  PRIMARY KEY (`ID.Usuario`, `Adminstrador (cliente)_ID.AdminC`, `Adminstrador
(cliente)_Administrador (Ninty)_ID.AdminN`),
  UNIQUE INDEX `ID.Usuario_UNIQUE` (`ID.Usuario` ASC),
  INDEX `fk_Usuario_Adminstrador (cliente)1_idx` (`Adminstrador (cliente)_ID.AdminC`
ASC, `Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN` ASC),
  CONSTRAINT `fk_Usuario_Adminstrador (cliente)1`
    FOREIGN KEY (`Adminstrador (cliente)_ID.AdminC` , `Adminstrador
(cliente)_Administrador (Ninty)_ID.AdminN`)
    REFERENCES `mydb`.`Adminstrador (cliente)` (`ID.AdminC` , `Administrador
(Ninty)_ID.AdminN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Modulo` (
  `ID.Modulo` INT NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `Precio` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID.Modulo`),
  UNIQUE INDEX `ID.Modulo_UNIQUE` (`ID.Modulo` ASC))



CREATE TABLE IF NOT EXISTS `mydb`.`Modulo adquirido` (
  `Usuario_ID.Usuario` INT NOT NULL,
  `Usuario_Adminstrador (cliente)_ID.AdminC` INT NOT NULL,
```

```
  `Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN` INT NOT NULL,
  `Modulo_ID.Modulo` INT NOT NULL,
  `Estado` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Usuario_ID.Usuario`, `Usuario_Adminstrador (cliente)_ID.AdminC`,
`Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`, `Modulo_ID.Modulo`),
  INDEX `fk_Usuario_has_Modulo_Modulo1_idx` (`Modulo_ID.Modulo` ASC),
  INDEX `fk_Usuario_has_Modulo_Usuario1_idx` (`Usuario_ID.Usuario` ASC,
`Usuario_Adminstrador (cliente)_ID.AdminC` ASC, `Usuario_Adminstrador
(cliente)_Administrador (Ninty)_ID.AdminN` ASC),
  CONSTRAINT `fk_Usuario_has_Modulo_Usuario1`
    FOREIGN KEY (`Usuario_ID.Usuario` , `Usuario_Adminstrador (cliente)_ID.AdminC` ,
`Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
    REFERENCES `mydb`.`Usuario` (`ID.Usuario` , `Adminstrador (cliente)_ID.AdminC` ,
`Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Usuario_has_Modulo_Modulo1`
    FOREIGN KEY (`Modulo_ID.Modulo`)
    REFERENCES `mydb`.`Modulo` (`ID.Modulo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Inventario` (
  `idInventario` INT NOT NULL,
  `nombre` VARCHAR(45) NULL,
  `descripcion` VARCHAR(45) NULL,
  PRIMARY KEY (`idInventario`),
  UNIQUE INDEX `idInventario_UNIQUE` (`idInventario` ASC))


CREATE TABLE IF NOT EXISTS `mydb`.`Producto` (
  `idProducto` INT NOT NULL,
  `nombre` VARCHAR(45) NULL,
  `descripcion` VARCHAR(45) NULL,
  `Precio` DOUBLE NULL,
  `Inventario_idInventario` INT NOT NULL,
  PRIMARY KEY (`idProducto`, `Inventario_idInventario`),
  UNIQUE INDEX `idProducto_UNIQUE` (`idProducto` ASC),
  INDEX `fk_Producto_Inventario1_idx` (`Inventario_idInventario` ASC),
  CONSTRAINT `fk_Producto_Inventario1`
    FOREIGN KEY (`Inventario_idInventario`)
    REFERENCES `mydb`.`Inventario` (`idInventario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Cliente` (
  `idCliente` INT NOT NULL,
  `nombre` VARCHAR(45) NULL,
  `apellido` VARCHAR(45) NULL,
  `NIT` VARCHAR(45) NULL,
  PRIMARY KEY (`idCliente`),
  UNIQUE INDEX `idCliente_UNIQUE` (`idCliente` ASC))


CREATE TABLE IF NOT EXISTS `mydb`.`Horario` (
  `idHorario` INT NOT NULL,
  `horas` INT NOT NULL,
  `no.dias` INT NOT NULL,
  PRIMARY KEY (`idHorario`),
  UNIQUE INDEX `idHorario_UNIQUE` (`idHorario` ASC))


CREATE TABLE IF NOT EXISTS `mydb`.`Empleado` (
  `CUI` INT NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `Apellido` VARCHAR(45) NOT NULL,
  `Telefono` VARCHAR(45) NULL,
  `Horario_idHorario` INT NOT NULL,
  PRIMARY KEY (`CUI`, `Horario_idHorario`),
  UNIQUE INDEX `CUI_UNIQUE` (`CUI` ASC),
  INDEX `fk_Empleado_Horario1_idx` (`Horario_idHorario` ASC),
  CONSTRAINT `fk_Empleado_Horario1`
    FOREIGN KEY (`Horario_idHorario`)
    REFERENCES `mydb`.`Horario` (`idHorario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Vendedor` (
  `idVendedor` INT NOT NULL,
  `Empleado_CUI` INT NOT NULL,
  PRIMARY KEY (`idVendedor`, `Empleado_CUI`),
  UNIQUE INDEX `idVendedor_UNIQUE` (`idVendedor` ASC),
  INDEX `fk_Vendedor_Empleado1_idx` (`Empleado_CUI` ASC),
  CONSTRAINT `fk_Vendedor_Empleado1`
    FOREIGN KEY (`Empleado_CUI`)
    REFERENCES `mydb`.`Empleado` (`CUI`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Venta` (
  `idVenta` INT NOT NULL,
  `Fecha` DATE NOT NULL,
  `Hora` TIME NOT NULL,
  `Monto` DOUBLE NOT NULL,
  `Cliente_idCliente` INT NOT NULL,
  `Vendedor_idVendedor` INT NOT NULL,
  `Producto_idProducto` INT NOT NULL,
  `Producto_Inventario_idInventario` INT NOT NULL,
  PRIMARY KEY (`idVenta`, `Cliente_idCliente`, `Vendedor_idVendedor`,
`Producto_idProducto`, `Producto_Inventario_idInventario`),
  INDEX `fk_Venta_Cliente1_idx` (`Cliente_idCliente` ASC),
  INDEX `fk_Venta_Vendedor1_idx` (`Vendedor_idVendedor` ASC),
  INDEX `fk_Venta_Producto1_idx` (`Producto_idProducto` ASC,
`Producto_Inventario_idInventario` ASC),
  CONSTRAINT `fk_Venta_Cliente1`
    FOREIGN KEY (`Cliente_idCliente`)
    REFERENCES `mydb`.`Cliente` (`idCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Venta_Vendedor1`
    FOREIGN KEY (`Vendedor_idVendedor`)
    REFERENCES `mydb`.`Vendedor` (`idVendedor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Venta_Producto1`
    FOREIGN KEY (`Producto_idProducto` , `Producto_Inventario_idInventario`)
    REFERENCES `mydb`.`Producto` (`idProducto` , `Inventario_idInventario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Proveedor` (
  `idProveedort` INT NOT NULL,
  `nombre` VARCHAR(45) NULL,
  `correo` VARCHAR(45) NULL,
  `telefono` INT NULL,
  PRIMARY KEY (`idProveedort`),
  UNIQUE INDEX `idProveedort_UNIQUE` (`idProveedort` ASC))


CREATE TABLE IF NOT EXISTS `mydb`.`Compra` (
  `idCompra` INT NOT NULL,
  `Monto` DOUBLE NOT NULL,
  `Fecha` DATE NOT NULL,
  `Hora` TIME NOT NULL,
```

```
  `Proveedor_idProveedort` INT NOT NULL,
  `Producto_idProducto` INT NOT NULL,
  `Producto_Inventario_idInventario` INT NOT NULL,
  PRIMARY KEY (`idCompra`, `Proveedor_idProveedort`, `Producto_idProducto`,
`Producto_Inventario_idInventario`),
  UNIQUE INDEX `idCompra_UNIQUE` (`idCompra` ASC),
  INDEX `fk_Compra_Proveedor1_idx` (`Proveedor_idProveedort` ASC),
  INDEX `fk_Compra_Producto1_idx` (`Producto_idProducto` ASC,
`Producto_Inventario_idInventario` ASC),
  CONSTRAINT `fk_Compra_Proveedor1`
    FOREIGN KEY (`Proveedor_idProveedort`)
    REFERENCES `mydb`.`Proveedor` (`idProveedort`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Compra_Producto1`
    FOREIGN KEY (`Producto_idProducto` , `Producto_Inventario_idInventario`)
    REFERENCES `mydb`.`Producto` (`idProducto` , `Inventario_idInventario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Entrevista` (
  `idEntrevista` INT NOT NULL,
  `fecha` DATE NULL,
  `hora` TIME NULL,
  `nombreEntrevistado` VARCHAR(45) NULL,
  `apellidoEntrevistado` VARCHAR(45) NULL,
  `puesto` VARCHAR(45) NULL,
  `Usuario_ID.Usuario` INT NOT NULL,
  `Usuario_Adminstrador (cliente)_ID.AdminC` INT NOT NULL,
  `Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN` INT NOT NULL,
  `Empleado_idEmpleado` INT NOT NULL,
  PRIMARY KEY (`idEntrevista`, `Usuario_ID.Usuario`, `Usuario_Adminstrador
(cliente)_ID.AdminC`, `Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`,
`Empleado_idEmpleado`),
  INDEX `fk_Entrevista_Usuario1_idx` (`Usuario_ID.Usuario` ASC, `Usuario_Adminstrador
(cliente)_ID.AdminC` ASC, `Usuario_Adminstrador (cliente)_Administrador
(Ninty)_ID.AdminN` ASC),
  INDEX `fk_Entrevista_Empleado1_idx` (`Empleado_idEmpleado` ASC),
  CONSTRAINT `fk_Entrevista_Usuario1`
    FOREIGN KEY (`Usuario_ID.Usuario` , `Usuario_Adminstrador (cliente)_ID.AdminC` ,
`Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
    REFERENCES `mydb`.`Usuario` (`ID.Usuario` , `Adminstrador (cliente)_ID.AdminC` ,
`Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
```

```
  CONSTRAINT `fk_Entrevista_Empleado1`
    FOREIGN KEY (`Empleado_idEmpleado`)
    REFERENCES `mydb`.`Empleado` (`CUI`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Vehiculo` (
  `Placa` INT NOT NULL,
  `Modelo` INT NULL,
  `Marca` VARCHAR(45) NULL,
  `Año` DATE NULL,
  PRIMARY KEY (`Placa`),
  UNIQUE INDEX `Placa_UNIQUE` (`Placa` ASC))


CREATE TABLE IF NOT EXISTS `mydb`.`Piloto` (
  `idPiloto` INT NOT NULL,
  `No.licencia` VARCHAR(45) NULL,
  `Empleado_CUI` INT NOT NULL,
  PRIMARY KEY (`idPiloto`, `Empleado_CUI`),
  UNIQUE INDEX `idPiloto_UNIQUE` (`idPiloto` ASC),
  INDEX `fk_Piloto_Empleado1_idx` (`Empleado_CUI` ASC),
  CONSTRAINT `fk_Piloto_Empleado1`
    FOREIGN KEY (`Empleado_CUI`)
    REFERENCES `mydb`.`Empleado` (`CUI`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Acompañante` (
  `idAcompañante` INT NOT NULL,
  `Empleado_CUI` INT NOT NULL,
  PRIMARY KEY (`idAcompañante`, `Empleado_CUI`),
  INDEX `fk_Acompañante_Empleado1_idx` (`Empleado_CUI` ASC),
  CONSTRAINT `fk_Acompañante_Empleado1`
    FOREIGN KEY (`Empleado_CUI`)
    REFERENCES `mydb`.`Empleado` (`CUI`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Viaje` (
  `idViaje` INT NOT NULL,
  `Piloto_idPiloto` INT NOT NULL,
  `Piloto_Empleado_CUI` INT NOT NULL,
```

```
  `Acompañante_idAcompañante` INT NOT NULL,
  `Acompañante_Empleado_CUI` INT NOT NULL,
  `Vehiculo_Placa` INT NOT NULL,
  `Direccion` VARCHAR(45) NULL,
  `Distancia` DOUBLE NULL,
  `FechaSalida` VARCHAR(45) NULL,
  `FechaLlegada` VARCHAR(45) NULL,
  PRIMARY KEY (`idViaje`, `Piloto_idPiloto`, `Piloto_Empleado_CUI`,
`Acompañante_idAcompañante`, `Acompañante_Empleado_CUI`, `Vehiculo_Placa`),
  INDEX `fk_Viaje_Piloto1_idx` (`Piloto_idPiloto` ASC, `Piloto_Empleado_CUI` ASC),
  INDEX `fk_Viaje_Acompañante1_idx` (`Acompañante_idAcompañante` ASC,
`Acompañante_Empleado_CUI` ASC),
  INDEX `fk_Viaje_Vehiculo1_idx` (`Vehiculo_Placa` ASC),
  CONSTRAINT `fk_Viaje_Piloto1`
    FOREIGN KEY (`Piloto_idPiloto`)
    REFERENCES `mydb`.`Piloto` (`idPiloto`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Viaje_Acompañante1`
    FOREIGN KEY (`Acompañante_idAcompañante`)
    REFERENCES `mydb`.`Acompañante` (`idAcompañante`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Viaje_Vehiculo1`
    FOREIGN KEY (`Vehiculo_Placa`)
    REFERENCES `mydb`.`Vehiculo` (`Placa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Anuncio` (
  `idAnuncio` INT NOT NULL,
  `titulo` VARCHAR(45) NULL,
  `descripcion` VARCHAR(100) NULL,
  `Usuario_ID.Usuario` INT NOT NULL,
  `Usuario_Adminstrador (cliente)_ID.AdminC` INT NOT NULL,
  `Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN` INT NOT NULL,
  PRIMARY KEY (`idAnuncio`, `Usuario_ID.Usuario`, `Usuario_Adminstrador
(cliente)_ID.AdminC`, `Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`),
  INDEX `fk_Anuncio_Usuario1_idx` (`Usuario_ID.Usuario` ASC, `Usuario_Adminstrador
(cliente)_ID.AdminC` ASC, `Usuario_Adminstrador (cliente)_Administrador
(Ninty)_ID.AdminN` ASC),
  CONSTRAINT `fk_Anuncio_Usuario1`
    FOREIGN KEY (`Usuario_ID.Usuario` , `Usuario_Adminstrador (cliente)_ID.AdminC` ,
`Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
```

```
    REFERENCES `mydb`.`Usuario` (`ID.Usuario` , `Adminstrador (cliente)_ID.AdminC` ,
`Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Oportunidad` (
  `idOportunidad` INT NOT NULL,
  `titulo` VARCHAR(45) NULL,
  `cantidadIngreso` VARCHAR(45) NULL,
  `Prioridad` VARCHAR(45) NULL,
  `Cliente_idCliente` INT NOT NULL,
  `Usuario_ID.Usuario` INT NOT NULL,
  `Usuario_Adminstrador (cliente)_ID.AdminC` INT NOT NULL,
  `Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN` INT NOT NULL,
  PRIMARY KEY (`idOportunidad`, `Cliente_idCliente`, `Usuario_ID.Usuario`,
`Usuario_Adminstrador (cliente)_ID.AdminC`, `Usuario_Adminstrador
(cliente)_Administrador (Ninty)_ID.AdminN`),
  INDEX `fk_Oportunidad_Cliente1_idx` (`Cliente_idCliente` ASC),
  INDEX `fk_Oportunidad_Usuario1_idx` (`Usuario_ID.Usuario` ASC, `Usuario_Adminstrador
(cliente)_ID.AdminC` ASC, `Usuario_Adminstrador (cliente)_Administrador
(Ninty)_ID.AdminN` ASC),
  CONSTRAINT `fk_Oportunidad_Cliente1`
    FOREIGN KEY (`Cliente_idCliente`)
    REFERENCES `mydb`.`Cliente` (`idCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Oportunidad_Usuario1`
    FOREIGN KEY (`Usuario_ID.Usuario` , `Usuario_Adminstrador (cliente)_ID.AdminC` ,
`Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
    REFERENCES `mydb`.`Usuario` (`ID.Usuario` , `Adminstrador (cliente)_ID.AdminC` ,
`Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Eventos` (
  `idEventos` INT NOT NULL,
  `titulo` VARCHAR(45) NULL,
  `descripcion` VARCHAR(603) NULL,
  `fecha` DATE NULL,
  `estado` VARCHAR(45) NULL,
  `Usuario_ID.Usuario` INT NOT NULL,
  `Usuario_Adminstrador (cliente)_ID.AdminC` INT NOT NULL,
  `Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN` INT NOT NULL,
```

```
  PRIMARY KEY (`idEventos`, `Usuario_ID.Usuario`, `Usuario_Adminstrador
(cliente)_ID.AdminC`, `Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`),
  INDEX `fk_Eventos_Usuario1_idx` (`Usuario_ID.Usuario` ASC, `Usuario_Adminstrador
(cliente)_ID.AdminC` ASC, `Usuario_Adminstrador (cliente)_Administrador
(Ninty)_ID.AdminN` ASC),
  CONSTRAINT `fk_Eventos_Usuario1`
    FOREIGN KEY (`Usuario_ID.Usuario` , `Usuario_Adminstrador (cliente)_ID.AdminC` ,
`Usuario_Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
    REFERENCES `mydb`.`Usuario` (`ID.Usuario` , `Adminstrador (cliente)_ID.AdminC` ,
`Adminstrador (cliente)_Administrador (Ninty)_ID.AdminN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Asistencia` (
  `idAsistencia` INT NOT NULL,
  `horaEntrada` TIME NOT NULL,
  `HoraSalida` TIME NOT NULL,
  `Fecha` DATE NOT NULL,
  `Empleado_CUI` INT NOT NULL,
  PRIMARY KEY (`idAsistencia`, `Empleado_CUI`),
  INDEX `fk_Asistencia_Empleado1_idx` (`Empleado_CUI` ASC),
  CONSTRAINT `fk_Asistencia_Empleado1`
    FOREIGN KEY (`Empleado_CUI`)
    REFERENCES `mydb`.`Empleado` (`CUI`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Viaje_has_Producto` (
  `Viaje_idViaje` INT NOT NULL,
  `Viaje_Piloto_idPiloto` INT NOT NULL,
  `Viaje_Piloto_Empleado_CUI` INT NOT NULL,
  `Viaje_Acompañante_idAcompañante` INT NOT NULL,
  `Viaje_Acompañante_Empleado_CUI` INT NOT NULL,
  `Viaje_Vehiculo_Placa` INT NOT NULL,
  `Producto_idProducto` INT NOT NULL,
  `Producto_Inventario_idInventario` INT NOT NULL,
  PRIMARY KEY (`Viaje_idViaje`, `Viaje_Piloto_idPiloto`, `Viaje_Piloto_Empleado_CUI`,
`Viaje_Acompañante_idAcompañante`, `Viaje_Acompañante_Empleado_CUI`,
`Viaje_Vehiculo_Placa`, `Producto_idProducto`, `Producto_Inventario_idInventario`),
  INDEX `fk_Viaje_has_Producto_Producto1_idx` (`Producto_idProducto` ASC,
`Producto_Inventario_idInventario` ASC),
  INDEX `fk_Viaje_has_Producto_Viaje1_idx` (`Viaje_idViaje` ASC, `Viaje_Piloto_idPiloto`
ASC, `Viaje_Piloto_Empleado_CUI` ASC, `Viaje_Acompañante_idAcompañante` ASC,
`Viaje_Acompañante_Empleado_CUI` ASC, `Viaje_Vehiculo_Placa` ASC),
```

```
  CONSTRAINT `fk_Viaje_has_Producto_Viaje1`
    FOREIGN KEY (`Viaje_idViaje` , `Viaje_Piloto_idPiloto` , `Viaje_Piloto_Empleado_CUI` ,
`Viaje_Acompañante_idAcompañante` , `Viaje_Acompañante_Empleado_CUI` ,
`Viaje_Vehiculo_Placa`)
    REFERENCES `mydb`.`Viaje` (`idViaje` , `Piloto_idPiloto` , `Piloto_Empleado_CUI` ,
`Acompañante_idAcompañante` , `Acompañante_Empleado_CUI` , `Vehiculo_Placa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Viaje_has_Producto_Producto1`
    FOREIGN KEY (`Producto_idProducto` , `Producto_Inventario_idInventario`)
    REFERENCES `mydb`.`Producto` (`idProducto` , `Inventario_idInventario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`Comentario` (
  `idComentario` INT NOT NULL,
  `descripcion` VARCHAR(100) NULL,
  `Anuncio_idAnuncio` INT NOT NULL,
  PRIMARY KEY (`idComentario`, `Anuncio_idAnuncio`),
  UNIQUE INDEX `idComentario_UNIQUE` (`idComentario` ASC),
  INDEX `fk_Comentario_Anuncio1_idx` (`Anuncio_idAnuncio` ASC),
  CONSTRAINT `fk_Comentario_Anuncio1`
    FOREIGN KEY (`Anuncio_idAnuncio`)
    REFERENCES `mydb`.`Anuncio` (`idAnuncio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


CREATE TABLE IF NOT EXISTS `mydb`.`` (
)


CREATE TABLE IF NOT EXISTS `mydb`.`MateriaPrima` (
  `idMateriaPrima` INT NOT NULL,
  `Nombre` VARCHAR(45) NULL,
  `descripcion` VARCHAR(60) NULL,
  `cantidad` VARCHAR(45) NULL,
  PRIMARY KEY (`idMateriaPrima`),
  UNIQUE INDEX `idMateriaPrima_UNIQUE` (`idMateriaPrima` ASC))


CREATE TABLE IF NOT EXISTS `mydb`.`Producto_has_Producto` (
  `Producto_idProducto` INT NOT NULL,
  `Producto_Inventario_idInventario` INT NOT NULL,
  `Producto_idProducto1` INT NOT NULL,
```

```
  `Producto_Inventario_idInventario1` INT NOT NULL,
  PRIMARY KEY (`Producto_idProducto`, `Producto_Inventario_idInventario`,
`Producto_idProducto1`, `Producto_Inventario_idInventario1`),
  INDEX `fk_Producto_has_Producto_Producto2_idx` (`Producto_idProducto1` ASC,
`Producto_Inventario_idInventario1` ASC),
  INDEX `fk_Producto_has_Producto_Producto1_idx` (`Producto_idProducto` ASC,
`Producto_Inventario_idInventario` ASC),
  CONSTRAINT `fk_Producto_has_Producto_Producto1`
    FOREIGN KEY (`Producto_idProducto` , `Producto_Inventario_idInventario`)
    REFERENCES `mydb`.`Producto` (`idProducto` , `Inventario_idInventario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Producto_has_Producto_Producto2`
    FOREIGN KEY (`Producto_idProducto1` , `Producto_Inventario_idInventario1`)
    REFERENCES `mydb`.`Producto` (`idProducto` , `Inventario_idInventario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```