

Proyecto – Fase I

Objetivos:

➤ Objetivo General:

Desarrollar una solución de software ERP para la empresa Ninty destinado para trabajar en el sector de las PYMES.

➤ Objetivos Específicos:

- Diseñar la función de usuarios para permitirles la gestión del sistema.
- Diseñar la función de la tienda de módulos para agregarlos a la aplicación.
- Diseñar los módulos que estarán disponibles en la tienda.
- Plantear la propuesta del módulo extra que será implementado en la tienda virtual.
- Definir los requerimientos del sistema.
- Crear la base de datos del proyecto.
- Conocer más sobre los sistemas ERP.

Alcance del proyecto:

Se pretende desarrollar una aplicación basada en los fundamentos de los sistemas ERP destinada para uso de las PYMES que busca ser una solución para agilizar y automatizar los procesos de las empresas contratantes, también busca incorporar nuevas tecnologías a las empresas ayudando en la gestión de tareas y facilitando su realización ahorrando tiempo y dándole mejor control sobre sus operaciones. Será una aplicación en la cual los propios clientes podrán elegir los módulos que desean cuando sea necesario y pudiendo descartar aquellos que no les son de utilidad.

Panorama general:

Este proyecto tiene por objeto diseñar una herramienta de gestión que se adecúe a cada negocio por medio de módulos que el mismo cliente elegirá para formar su propio sistema. Más adelante el cliente podrá reemplazar los módulos que desee, quitar algunos y agregar nuevos, según las necesidades que la empresa presente.

La aplicación estará compuesta por múltiples módulos donde cada uno ejecutará determinadas funciones para la gestión de la empresa que los haya seleccionado. Los módulos principales de la aplicación son:

- Módulo usuarios: donde se encuentran los distintos usuarios que poseerá el sistema, como lo son los administradores del sistema, del servicio y los usuarios que harán uso del ERP.
- Tienda de módulos: tienda virtual donde se podrán agregar nuevos módulos y eliminar aquellos que quedaron obsoletos, dentro de los módulos que se tienen se encuentran: gestión de inventario, gestión de ventas, gestión de compras, facturación, reclutamiento, gestión de flotas, blog, eventos, CRM y el módulo propuesto: ‘Marketing’, para la gestión de acciones promocionales.

Requerimientos iniciales:

Requerimientos funcionales:

➤ Funciones del módulo usuario:

Ref #	Función	Categoría
R1.1	Habilita o deshabilita los módulos del sistema.	Evidente
R1.2	Registra el control de los clientes que han contratado los servicios.	Oculto
R1.3	Registrar tarjeta de crédito del contratante.	Evidente
R1.4	Registrar los pagos de los servicios prestados.	Oculto
R1.5	Agregar o eliminar módulos al ERP contratado.	Evidente
R1.6	Crear nuevos usuarios del sistema.	Evidente
R1.7	Gestionar las funciones de los módulos.	Evidente

➤ Funciones de la tienda de módulos:

Ref #	Función	Categoría
R2.1	Agregar módulos a la tienda virtual.	Evidente
R2.2	Eliminar módulos no funcionales.	Evidente

➤ Funciones de gestión de inventario:

Ref #	Función	Categoría
R3.1	Debitar los productos luego de una venta.	Oculto
R3.2	Registrar creación, eliminación o modificación de los productos.	Evidente

➤ Funciones de gestión de ventas:

Ref #	Función	Categoría
R4.1	Gestionar las ventas realizadas por la empresa.	Ocultas

➤ Funciones de gestión de compras:

Ref #	Función	Categoría
R5.1	Gestionar las compras realizadas por la empresa.	Evidente
R5.2	Realizar un análisis de ingresos.	Evidente

➤ Funciones de facturación:

Ref #	Función	Categoría
R6.1	Realizar la facturación luego de una venta.	Evidente

➤ Funciones de reclutamiento:

Ref #	Función	Categoría
R7.1	Mostrar información del personal.	Evidente
R7.2	Agendar y calendarizar entrevistas de trabajo	Ocultas

➤ Funciones de gestión de flotas:

Ref #	Función	Categoría
R8.1	Asignación de viajes	Ocultas

➤ Funciones del blog:

Ref #	Función	Categoría
R9.1	Publicación de anuncios.	Evidente
R9.2	Recibir comentarios.	Evidente

➤ Funciones de eventos:

Ref #	Función	Categoría
R10.1	Crear eventos en el calendario.	Evidente
R10.2	Manejar estados.	Ocultas

➤ Funciones de CRM:

Ref #	Función	Categoría
-------	---------	-----------

R11.1	Proyectar oportunidades de negocio.	Evidente
R11.2	Mover oportunidades.	Evidente

➤ Funciones de Marketing:

Ref #	Función	Categoría
R12.1	Crear campañas de promoción.	Evidente
R12.2	Realizar seguimiento de clientes.	Oculto

Atributos del sistema:

Ref #	Función	Cat.	Atributo	Detalles y restricciones	Cat.
R1.1	Habilita o deshabilita los módulos del sistema.	Evidente	Tiempo de respuesta.	5 segundos como máximo.	Obligatorio
R1.4	Registrar los pagos de los servicios prestados.	Oculto	Tolerancia a fallas.	Debe registrar los pagos en un plazo máximo de 24 horas.	Obligatorio
R1.6	Crear nuevos usuarios del sistema.	Evidente	Tiempo de guardado.	Se debe de guardar el usuario y activar en menos de 15 segundos.	Obligatorio
R3.1	Debitar los productos luego de una venta.	Oculto	Conteo de inventario.	Se debe notificar al administrador en menos de 5 minutos cuando se agote un producto.	Obligatorio
R4.1	Gestionar las ventas realizadas por la empresa.	Oculto	Notificación de venta exitosa.	5 segundos como máximo.	Obligatorio
R5.1	Gestionar las compras realizadas por la empresa.	Evidente	Notificación de compra.	5 segundos como máximo.	Obligatorio
R7.1	Mostrar información del personal.	Evidente	Tiempo de respuesta	Se debe mostrar la información en menos de 10 segundos.	Obligatorio

R9.1	Publicación de anuncios.	Evidente	Actualización de contenido.	Se debe de actualizar el blog en menos de 10 segundos.	Obligatorio
R9.2	Recibir comentarios.	Evidente	Despliegue de datos.	Muestra el comentario en un máximo de 5 segundos.	Obligatorio
R11.2	Mover oportunidades.	Evidente	Visualización	Permite mover anuncios de un lugar a otro actualizando la pantalla cada segundo.	Opcional
			Cambio de interfaz	Se realiza en un máximo de 5 segundos	Opcional
			Actualización de datos	Los datos ingresados se introducen en menos de 5 segundos.	Obligatorio
			Ventanas salientes	Los cuadros de diálogo emergentes se muestran en menos de 3 segundos.	Opcional

Cientes:

Cliente	Descripción	Módulos
Administrador del sistema	Es quien gestiona los módulos que se presentan y además quien lleva un control sobre los clientes que han contratado el servicio de ERP además de llevar el control sobre los pagos de la aplicación.	Módulo Usuario, Tienda de módulos, sub-módulos presentes en la tienda.
Administrador de servicios	Es quien contrata los servicios del ERP, podrá agregar o eliminar módulos al sistema contratado y además puede crear nuevos usuarios para que utilicen los demás módulos contratados.	Módulo Usuario, tienda de módulos, sub-módulos presentes en la tienda.
Usuario registrado	Es creado por el administrador de servicios y quien gestiona las funciones que ofrece cada módulo en específico.	Sub-módulos presentes en la tienda.

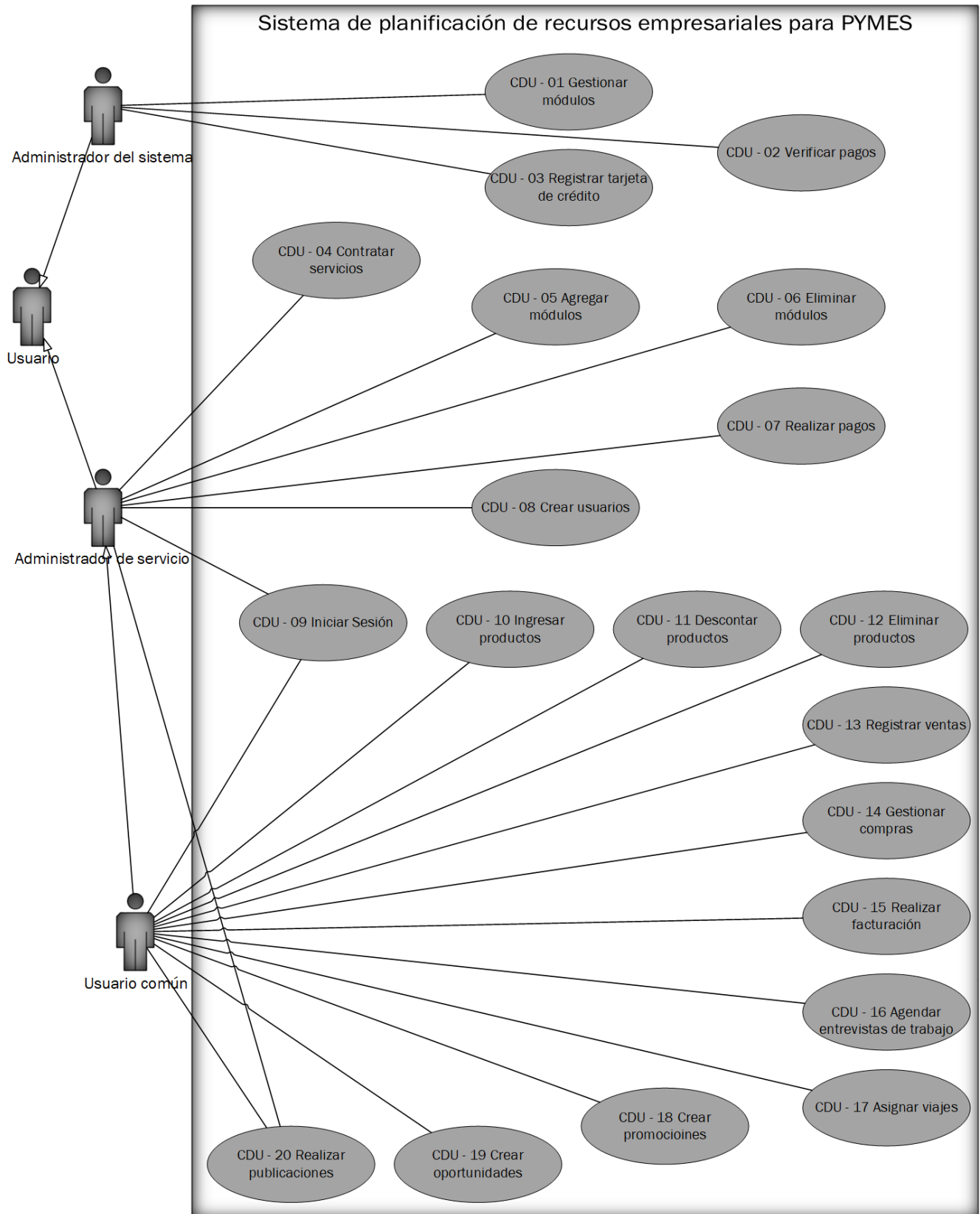
Glosario:

Para una mejor comprensión del proyecto se ha elaborado una lista con algunos términos que pueden llegar a provocar una confusión a los usuarios y que por ello se les ha colocado en la siguiente tabla junto a su significado:

Término	Categoría/Modelo	Descripción
Blog	Requerimientos	Página web que se actualiza regularmente y que suele tratar contenido de interés.
Boolean	Entidad-relación	Tipo de dato que representa valores que pueden ser verdadero o falso.
Credencial	Entidad-relación	Prueba de autenticación que se le da a una persona.
CRM	Requerimientos	‘Customer Relationship Management’, gestión de relaciones con clientes.
Date	Conceptual	Tipo de dato que almacena valores de fechas.
Double	Entidad-relación	Tipo de dato que representa valores decimales.
ERP	Requerimientos	‘Enterprise Resource Planning’, sistema de planificación de recursos empresariales.
INT	Entidad-relación	Tipo de dato que representa valores enteros.
Interfaz	Requerimientos	Conexión funcional que permite la visualización de información en la pantalla de la computadora.
Marketing	Requerimientos	Técnicas que tienen como objeto mejorar la comercialización de un producto.
Módulo	Requerimientos	Es una porción de un programa que realiza determinadas tareas.
String	Conceptual	Tipo de dato que representa una serie de caracteres
PYME	Requerimientos	Siglas de Pequeña Y Mediana Empresa
Usuario	Entidad-relación	Persona que utiliza un servicio.
Varchar	Entidad-relación	Tipo de dato que almacena una serie de caracteres.

Diagrama de casos de uso:

➤ Alto nivel:



Nombre del caso de uso:	CDU-01 Gestionar módulos
Actores:	Administrador del sistema
Tipo:	Primario
Descripción:	Permite que el administrador del ERP pueda habilitar o deshabilitar los módulos.

Nombre del caso de uso:	CDU-02 Verificar pagos
Actores:	Administrador del sistema
Tipo:	Primario
Descripción:	Permite que el administrador del ERP verifique los pagos recibidos por los servicios prestados.

Nombre del caso de uso:	CDU-03 Registrar tarjeta de crédito
Actores:	Administrador del sistema
Tipo:	Primario
Descripción:	Al efectuar una contratación con una tarjeta nueva, el administrador deberá registrarla anteriormente para que así el cliente pueda realizar los demás pagos.

Nombre del caso de uso:	CDU-04 Contratar servicios
Actores:	Administrador de servicios
Tipo:	Primario
Descripción:	Por medio de ello nuevas empresas podrán contratar los servicios que ofrece Ninty.

Nombre del caso de uso:	CDU-05 Agregar módulos
Actores:	Administrador de servicios
Tipo:	Primario
Descripción:	Permite que el administrador cliente agregue módulos para la gestión de su empresa.

Nombre del caso de uso:	CDU-06 Eliminar módulos
Actores:	Administrador de servicios

Tipo:	Primario
Descripción:	Permite que el administrador cliente elimine aquellos módulos que ya no son de su utilidad.

Nombre del caso de uso:	CDU-07 Realizar pagos
Actores:	Administrador de servicios
Tipo:	Primario
Descripción:	Por este módulo se realiza el pago por los servicios prestados.

Nombre del caso de uso:	CDU-08 Crear usuarios
Actores:	Administrador de servicios
Tipo:	Primario
Descripción:	Por este módulo es que el administrador podrá crear nuevos usuarios que podrán gestionar los demás módulos adquiridos.

Nombre del caso de uso:	CDU-09 Iniciar sesión
Actores:	Administrador de servicios y usuario común
Tipo:	Primario
Descripción:	Permite que los usuarios que compraron la aplicación puedan acceder al portal del sistema y así gestionar los módulos presenta.

Nombre del caso de uso:	CDU-10 Ingresar productos
Actores:	Usuario común
Tipo:	Primario
Descripción:	Permite que los usuarios ingresen productos al inventario.

Nombre del caso de uso:	CDU-11 Descontar productos
Actores:	Usuario común
Tipo:	Primario
Descripción:	Por este módulo el usuario podrá descontar los productos que se encuentren en el inventario.

Nombre del caso de uso:	CDU-12 Eliminar productos
--------------------------------	---------------------------

Actores:	Usuario común
Tipo:	Primario
Descripción:	Módulo con el cual el usuario eliminará los productos del inventario.

Nombre del caso de uso:	CDU-13 Registrar ventas
Actores:	Usuario común
Tipo:	Primario
Descripción:	Permite que los usuarios registren las ventas realizadas.

Nombre del caso de uso:	CDU-14 Gestionar compras
Actores:	Usuario común
Tipo:	Primario
Descripción:	Permite que los usuarios administren la información de compras realizadas.

Nombre del caso de uso:	CDU-15 Realizar facturación
Actores:	Usuario común
Tipo:	Primario
Descripción:	Módulo que recoge la información de las ventas realizadas y la ordena para ser impresa y entregar una copia al cliente, además envía la información recopilada al módulo de ventas.

Nombre del caso de uso:	CDU-16 Agendar entrevistas de trabajo
Actores:	Usuario común
Tipo:	Primario
Descripción:	Módulo con el cual se agendan y calendarizan las entrevistas de trabajo para personal que se necesite contratar.

Nombre del caso de uso:	CDU-17 Asignar viajes
Actores:	Usuario común
Tipo:	Primario

Descripción:	Con este módulo se tendrá más control sobre el personal que realice viajes para entrega de producto.
---------------------	--

Nombre del caso de uso:	CDU-18 Crear promociones
Actores:	Usuario común
Tipo:	Primario
Descripción:	Función del módulo extra con el cual se podrán gestionar las promociones con el motivo de atraer más clientela.

Nombre del caso de uso:	CDU-19 Crear oportunidades
Actores:	Usuario común
Tipo:	Primario
Descripción:	Con esta funcionalidad se crearán las oportunidades destinadas a los clientes para buscar hacer negocio.

Nombre del caso de uso:	CDU-20 Realizar publicaciones
Actores:	Usuario común y Administrador de servicios
Tipo:	Primario
Descripción:	Función con la cual los usuarios podrán publicar anuncios en el blog de la empresa.

Modelo conceptual:

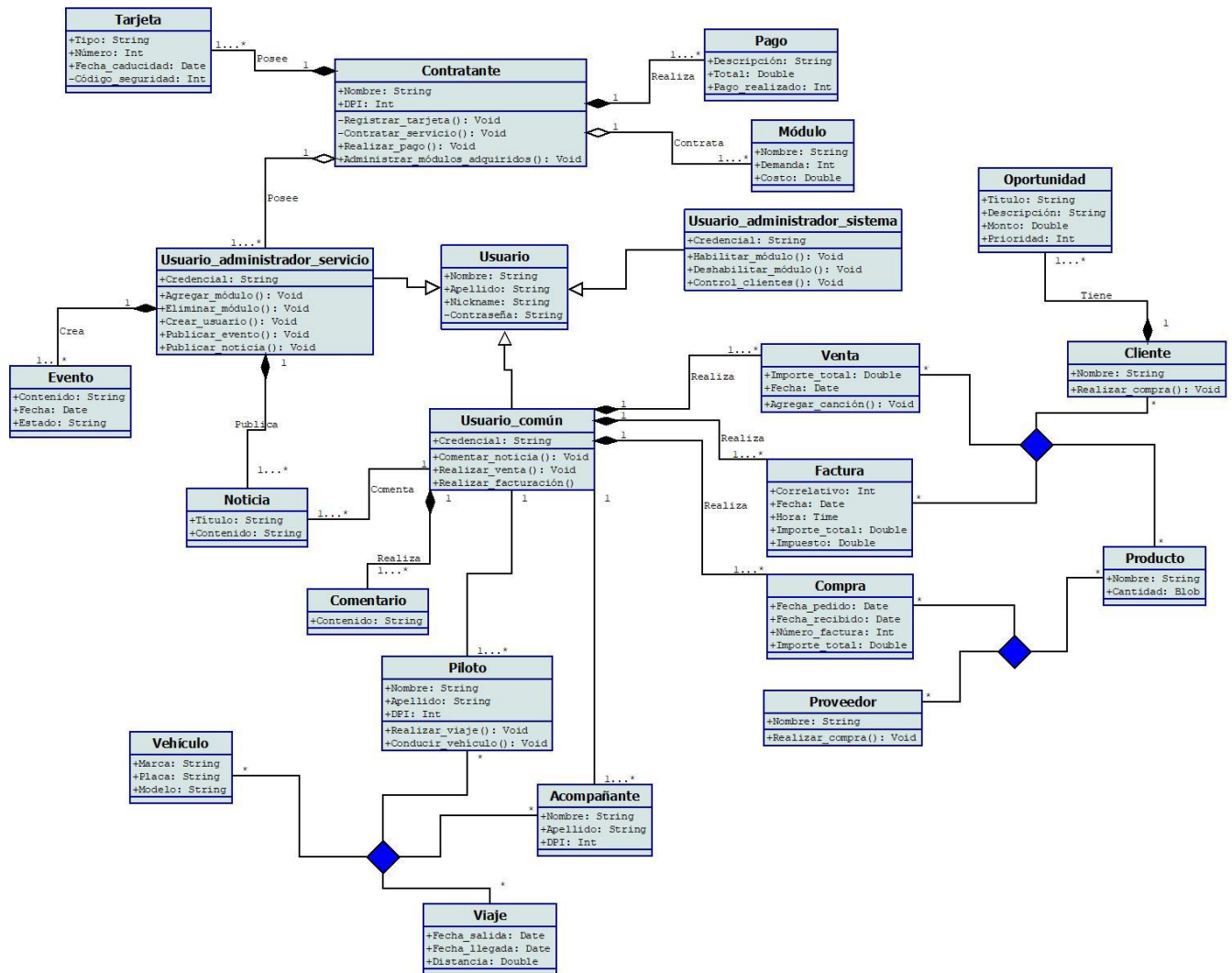
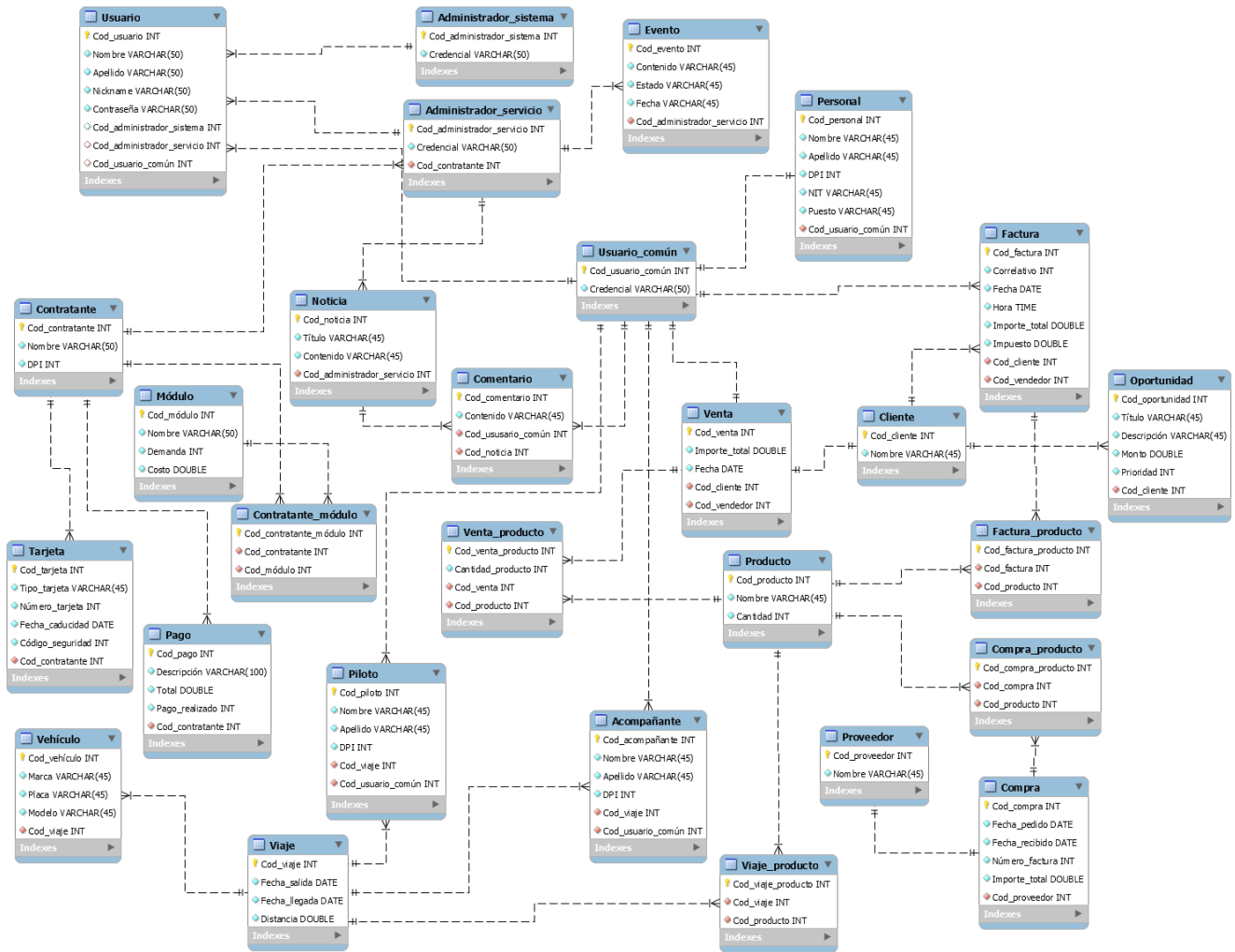
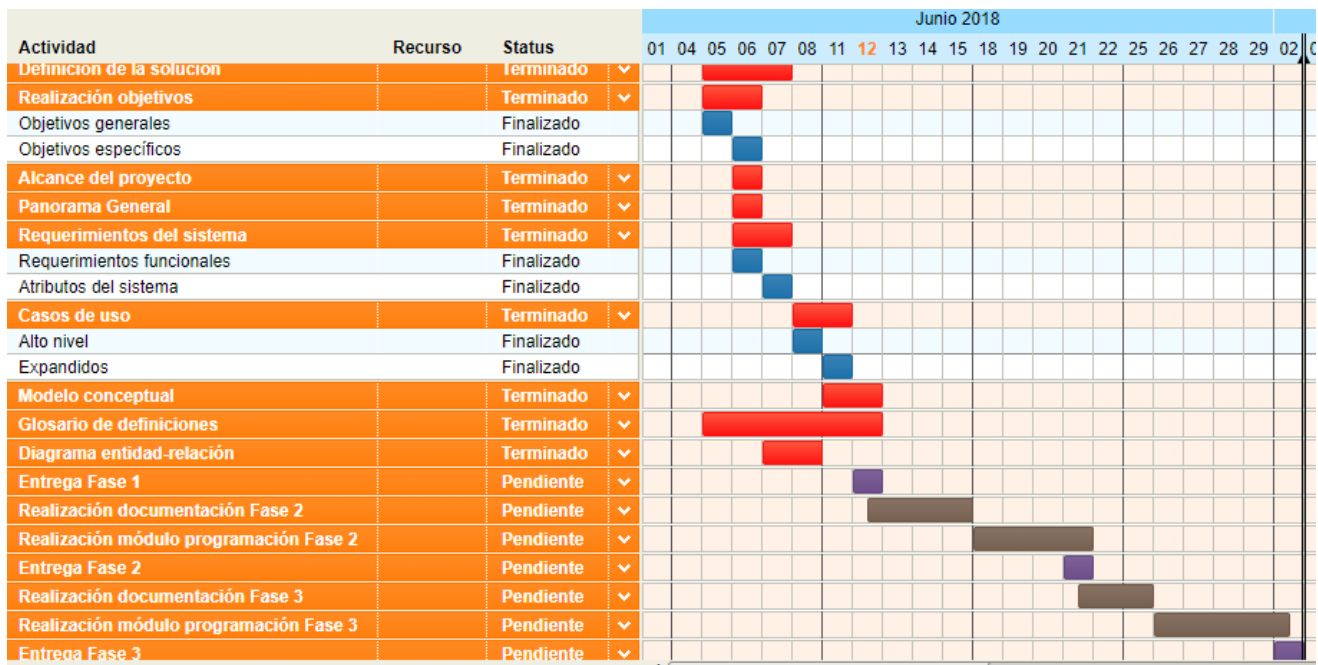


Diagrama entidad-relación:



Planificación de proyecto



Script:

```
CREATE DATABASE SistemaERP
```

```
GO
```

```
USE SistemaERP
```

```
CREATE TABLE IF NOT EXISTS `SistemaERP`.`Usuario_común` (
  `Cod_usuario_común` INT NOT NULL,
  `Credencial` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`Cod_usuario_común`))
```

```
CREATE TABLE IF NOT EXISTS `SistemaERP`.`Administrador_sistema` (
  `Cod_administrador_sistema` INT NOT NULL,
  `Credencial` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`Cod_administrador_sistema`))
```

```
CREATE TABLE IF NOT EXISTS `SistemaERP`.`Contratante` (
  `Cod_contratante` INT NOT NULL,
  `Nombre` VARCHAR(50) NOT NULL,
  `DPI` INT NOT NULL,
  PRIMARY KEY (`Cod_contratante`))
```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Administrador_servicio` (
  `Cod_administrador_servicio` INT NOT NULL,
  `Credencial` VARCHAR(50) NOT NULL,
  `Cod_contratante` INT NOT NULL,
  PRIMARY KEY (`Cod_administrador_servicio`),
  INDEX `Cod_contratante_idx` (`Cod_contratante` ASC),
  CONSTRAINT `Cod_contratante`
    FOREIGN KEY (`Cod_contratante`)
    REFERENCES `SistemaERP`.`Contratante` (`Cod_contratante`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Usuario` (
  `Cod_usuario` INT NOT NULL,
  `Nombre` VARCHAR(50) NOT NULL,
  `Apellido` VARCHAR(50) NOT NULL,
  `Nickname` VARCHAR(50) NOT NULL,
  `Contraseña` VARCHAR(50) NOT NULL,
  `Cod_administrador_sistema` INT NULL,
  `Cod_administrador_servicio` INT NULL,
  `Cod_usuario_común` INT NULL,
  PRIMARY KEY (`Cod_usuario`),
  INDEX `Cod_usuario_común_idx` (`Cod_usuario_común` ASC),
  INDEX `Cod_administrador_sistema_idx` (`Cod_administrador_servicio` ASC),
  CONSTRAINT `Cod_usuario_común`
    FOREIGN KEY (`Cod_usuario_común`)
    REFERENCES `SistemaERP`.`Usuario_común` (`Cod_usuario_común`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Cod_administrador_sistema`
    FOREIGN KEY (`Cod_administrador_servicio`)
    REFERENCES `SistemaERP`.`Administrador_sistema` (`Cod_administrador_sistema`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Cod_administrador_servicio`
    FOREIGN KEY (`Cod_administrador_servicio`)
    REFERENCES `SistemaERP`.`Administrador_servicio` (`Cod_administrador_servicio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Módulo` (
  `Cod_módulo` INT NOT NULL,
  `Nombre` VARCHAR(50) NOT NULL,
  `Demanda` INT NOT NULL,
  `Costo` DOUBLE NOT NULL,
  PRIMARY KEY (`Cod_módulo`))

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Contratante_módulo` (
  `Cod_contratante_módulo` INT NOT NULL,
  `Cod_contratante` INT NOT NULL,
  `Cod_módulo` INT NOT NULL,
  PRIMARY KEY (`Cod_contratante_módulo`),
  INDEX `Cod_contratante_idx` (`Cod_contratante` ASC),
  INDEX `Cod_módulo_idx` (`Cod_módulo` ASC),
  CONSTRAINT `Cod_contratante`
    FOREIGN KEY (`Cod_contratante`)
      REFERENCES `SistemaERP`.`Contratante` (`Cod_contratante`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Cod_módulo`
    FOREIGN KEY (`Cod_módulo`)
      REFERENCES `SistemaERP`.`Módulo` (`Cod_módulo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Pago` (
  `Cod_pago` INT NOT NULL,
  `Descripción` VARCHAR(100) NOT NULL,
  `Total` DOUBLE NOT NULL,
  `Pago_realizado` INT NOT NULL,
  `Cod_contratante` INT NOT NULL,
  PRIMARY KEY (`Cod_pago`),
  INDEX `Cod_contratante_idx` (`Cod_contratante` ASC),
  CONSTRAINT `Cod_contratante`
    FOREIGN KEY (`Cod_contratante`)
      REFERENCES `SistemaERP`.`Contratante` (`Cod_contratante`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Tarjeta` (
  `Cod_tarjeta` INT NOT NULL,

```



```

`Tipo_tarjeta` VARCHAR(45) NOT NULL,
`Número_tarjeta` INT NOT NULL,
`Fecha_caducidad` DATE NOT NULL,
`Código_seguridad` INT NOT NULL,
`Cod_contratante` INT NOT NULL,
PRIMARY KEY (`Cod_tarjeta`),
INDEX `Cod_contratante_idx` (`Cod_contratante` ASC),
CONSTRAINT `Cod_contratante`
FOREIGN KEY (`Cod_contratante`)
REFERENCES `SistemaERP`.`Contratante` (`Cod_contratante`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Producto` (
  `Cod_producto` INT NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `Cantidad` INT NOT NULL,
  PRIMARY KEY (`Cod_producto`))

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Cliente` (
  `Cod_cliente` INT NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Cod_cliente`))

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Venta` (
  `Cod_venta` INT NOT NULL,
  `Importe_total` DOUBLE NOT NULL,
  `Fecha` DATE NOT NULL,
  `Cod_cliente` INT NOT NULL,
  `Cod_vendedor` INT NOT NULL,
  PRIMARY KEY (`Cod_venta`),
  INDEX `Cod_cliente_idx` (`Cod_cliente` ASC),
  INDEX `Cod_vendedor_idx` (`Cod_vendedor` ASC),
  CONSTRAINT `Cod_cliente`
  FOREIGN KEY (`Cod_cliente`)
  REFERENCES `SistemaERP`.`Cliente` (`Cod_cliente`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `Cod_vendedor`
  FOREIGN KEY (`Cod_vendedor`)
  REFERENCES `SistemaERP`.`Usuario_común` (`Cod_usuario_común`)

```

ON DELETE NO ACTION
ON UPDATE NO ACTION)

```
CREATE TABLE IF NOT EXISTS `SistemaERP`.`Venta_producto` (  
  `Cod_venta_producto` INT NOT NULL,  
  `Cantidad_producto` INT NOT NULL,  
  `Cod_venta` INT NOT NULL,  
  `Cod_producto` INT NOT NULL,  
  PRIMARY KEY (`Cod_venta_producto`),  
  INDEX `Cod_venta_idx` (`Cod_venta` ASC),  
  INDEX `Cod_producto_idx` (`Cod_producto` ASC),  
  CONSTRAINT `Cod_venta`  
    FOREIGN KEY (`Cod_venta`)  
      REFERENCES `SistemaERP`.`Venta` (`Cod_venta`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `Cod_producto`  
    FOREIGN KEY (`Cod_producto`)  
      REFERENCES `SistemaERP`.`Producto` (`Cod_producto`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)
```

```
CREATE TABLE IF NOT EXISTS `SistemaERP`.`Proveedor` (  
  `Cod_proveedor` INT NOT NULL,  
  `Nombre` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Cod_proveedor`))
```

```
CREATE TABLE IF NOT EXISTS `SistemaERP`.`Compra` (  
  `Cod_compra` INT NOT NULL,  
  `Fecha_pedido` DATE NOT NULL,  
  `Fecha_recibido` DATE NOT NULL,  
  `Número_factura` INT NOT NULL,  
  `Importe_total` DOUBLE NOT NULL,  
  `Cod_proveedor` INT NOT NULL,  
  PRIMARY KEY (`Cod_compra`),  
  INDEX `Cod_proveedor_idx` (`Cod_proveedor` ASC),  
  CONSTRAINT `Cod_proveedor`  
    FOREIGN KEY (`Cod_proveedor`)  
      REFERENCES `SistemaERP`.`Proveedor` (`Cod_proveedor`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)
```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Factura` (
  `Cod_factura` INT NOT NULL,
  `Correlativo` INT NOT NULL,
  `Fecha` DATE NOT NULL,
  `Hora` TIME NOT NULL,
  `Importe_total` DOUBLE NOT NULL,
  `Impuesto` DOUBLE NOT NULL,
  `Cod_cliente` INT NOT NULL,
  `Cod_vendedor` INT NOT NULL,
  PRIMARY KEY (`Cod_factura`),
  INDEX `Cod_cliente_idx` (`Cod_cliente` ASC),
  INDEX `Cod_vendedor_idx` (`Cod_vendedor` ASC),
  CONSTRAINT `Cod_cliente`
    FOREIGN KEY (`Cod_cliente`)
      REFERENCES `SistemaERP`.`Cliente` (`Cod_cliente`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `Cod_vendedor`
    FOREIGN KEY (`Cod_vendedor`)
      REFERENCES `SistemaERP`.`Usuario_común` (`Cod_usuario_común`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Factura_producto` (
  `Cod_factura_producto` INT NOT NULL,
  `Cod_factura` INT NOT NULL,
  `Cod_producto` INT NOT NULL,
  PRIMARY KEY (`Cod_factura_producto`),
  INDEX `Cod_factura_idx` (`Cod_factura` ASC),
  INDEX `Cod_producto_idx` (`Cod_producto` ASC),
  CONSTRAINT `Cod_factura`
    FOREIGN KEY (`Cod_factura`)
      REFERENCES `SistemaERP`.`Factura` (`Cod_factura`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `Cod_producto`
    FOREIGN KEY (`Cod_producto`)
      REFERENCES `SistemaERP`.`Producto` (`Cod_producto`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Compra_producto` (
  `Cod_compra_producto` INT NOT NULL,
  `Cod_compra` INT NOT NULL,
  `Cod_producto` INT NOT NULL,
  PRIMARY KEY (`Cod_compra_producto`),
  INDEX `Cod_compra_idx` (`Cod_compra` ASC),
  INDEX `Cod_producto_idx` (`Cod_producto` ASC),
  CONSTRAINT `Cod_compra`
    FOREIGN KEY (`Cod_compra`)
    REFERENCES `SistemaERP`.`Compra` (`Cod_compra`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Cod_producto`
    FOREIGN KEY (`Cod_producto`)
    REFERENCES `SistemaERP`.`Producto` (`Cod_producto`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Personal` (
  `Cod_personal` INT NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `Apellido` VARCHAR(45) NOT NULL,
  `DPT` INT NOT NULL,
  `NIT` VARCHAR(45) NOT NULL,
  `Puesto` VARCHAR(45) NOT NULL,
  `Cod_usuario_común` INT NOT NULL,
  PRIMARY KEY (`Cod_personal`),
  INDEX `Cod_usuario_común_idx` (`Cod_usuario_común` ASC),
  CONSTRAINT `Cod_usuario_común`
    FOREIGN KEY (`Cod_usuario_común`)
    REFERENCES `SistemaERP`.`Usuario_común` (`Cod_usuario_común`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Viaje` (
  `Cod_viaje` INT NOT NULL,
  `Fecha_salida` DATE NOT NULL,
  `Fecha_llegada` DATE NOT NULL,
  `Distancia` DOUBLE NOT NULL,
  PRIMARY KEY (`Cod_viaje`))

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Vehículo` (
  `Cod_vehículo` INT NOT NULL,
  `Marca` VARCHAR(45) NOT NULL,
  `Placa` VARCHAR(45) NOT NULL,
  `Modelo` VARCHAR(45) NOT NULL,
  `Cod_viaje` INT NOT NULL,
  PRIMARY KEY (`Cod_vehículo`),
  INDEX `Cod_piloto_idx` (`Cod_viaje` ASC),
  CONSTRAINT `Cod_viaje`
    FOREIGN KEY (`Cod_viaje`)
      REFERENCES `SistemaERP`.`Viaje` (`Cod_viaje`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Piloto` (
  `Cod_piloto` INT NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `Apellido` VARCHAR(45) NOT NULL,
  `DPI` INT NOT NULL,
  `Cod_viaje` INT NOT NULL,
  `Cod_usuario_común` INT NOT NULL,
  PRIMARY KEY (`Cod_piloto`),
  INDEX `Cod_viaje_idx` (`Cod_viaje` ASC),
  INDEX `Cod_usuario_común_idx` (`Cod_usuario_común` ASC),
  CONSTRAINT `Cod_viaje`
    FOREIGN KEY (`Cod_viaje`)
      REFERENCES `SistemaERP`.`Viaje` (`Cod_viaje`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Cod_usuario_común`
    FOREIGN KEY (`Cod_usuario_común`)
      REFERENCES `SistemaERP`.`Usuario_común` (`Cod_usuario_común`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Acompañante` (
  `Cod_acompañante` INT NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `Apellido` VARCHAR(45) NOT NULL,
  `DPI` INT NOT NULL,
  `Cod_viaje` INT NOT NULL,

```

```

`Cod_usuario_común` INT NOT NULL,
PRIMARY KEY (`Cod_acompañante`),
INDEX `Cod_viaje_idx` (`Cod_viaje` ASC),
INDEX `Cod_usuario_común_idx` (`Cod_usuario_común` ASC),
CONSTRAINT `Cod_viaje`
  FOREIGN KEY (`Cod_viaje`)
  REFERENCES `SistemaERP`.`Viaje` (`Cod_viaje`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `Cod_usuario_común`
  FOREIGN KEY (`Cod_usuario_común`)
  REFERENCES `SistemaERP`.`Usuario_común` (`Cod_usuario_común`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Viaje_producto` (
  `Cod_viaje_producto` INT NOT NULL,
  `Cod_viaje` INT NOT NULL,
  `Cod_producto` INT NOT NULL,
  PRIMARY KEY (`Cod_viaje_producto`),
  INDEX `Cod_viaje_idx` (`Cod_viaje` ASC),
  INDEX `Cod_producto_idx` (`Cod_producto` ASC),
  CONSTRAINT `Cod_viaje`
    FOREIGN KEY (`Cod_viaje`)
    REFERENCES `SistemaERP`.`Viaje` (`Cod_viaje`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Cod_producto`
    FOREIGN KEY (`Cod_producto`)
    REFERENCES `SistemaERP`.`Producto` (`Cod_producto`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS `SistemaERP`.`Noticia` (
  `Cod_noticia` INT NOT NULL,
  `Título` VARCHAR(45) NOT NULL,
  `Contenido` VARCHAR(45) NOT NULL,
  `Cod_administrador_servicio` INT NOT NULL,
  PRIMARY KEY (`Cod_noticia`),
  INDEX `Cod_administrador_servicio_idx` (`Cod_administrador_servicio` ASC),
  CONSTRAINT `Cod_administrador_servicio`
    FOREIGN KEY (`Cod_administrador_servicio`)

```

```
REFERENCES `SistemaERP`.`Administrador_servicio` (`Cod_administrador_servicio`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```

```
CREATE TABLE IF NOT EXISTS `SistemaERP`.`Comentario` (
  `Cod_comentario` INT NOT NULL,
  `Contenido` VARCHAR(45) NOT NULL,
  `Cod_usuario_común` INT NOT NULL,
  `Cod_noticia` INT NOT NULL,
  PRIMARY KEY (`Cod_comentario`),
  INDEX `Cod_usuario_común_idx` (`Cod_usuario_común` ASC),
  INDEX `Cod_noticia_idx` (`Cod_noticia` ASC),
  CONSTRAINT `Cod_usuario_común`
    FOREIGN KEY (`Cod_usuario_común`)
    REFERENCES `SistemaERP`.`Usuario_común` (`Cod_usuario_común`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Cod_noticia`
    FOREIGN KEY (`Cod_noticia`)
    REFERENCES `SistemaERP`.`Noticia` (`Cod_noticia`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

```
CREATE TABLE IF NOT EXISTS `SistemaERP`.`Evento` (
  `Cod_evento` INT NOT NULL,
  `Contenido` VARCHAR(45) NOT NULL,
  `Estado` VARCHAR(45) NOT NULL,
  `Fecha` VARCHAR(45) NOT NULL,
  `Cod_administrador_servicio` INT NOT NULL,
  PRIMARY KEY (`Cod_evento`),
  INDEX `Cod_administrador_servicio_idx` (`Cod_administrador_servicio` ASC),
  CONSTRAINT `Cod_administrador_servicio`
    FOREIGN KEY (`Cod_administrador_servicio`)
    REFERENCES `SistemaERP`.`Administrador_servicio` (`Cod_administrador_servicio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

```
CREATE TABLE IF NOT EXISTS `SistemaERP`.`Oportunidad` (
  `Cod_oportunidad` INT NOT NULL,
  `Título` VARCHAR(45) NOT NULL,
```

```
`Descripción` VARCHAR(45) NOT NULL,  
`Monto` DOUBLE NOT NULL,  
`Prioridad` INT NOT NULL,  
`Cod_cliente` INT NOT NULL,  
PRIMARY KEY (`Cod_oportunidad`),  
INDEX `Cod_cliente_idx` (`Cod_cliente` ASC),  
CONSTRAINT `Cod_cliente`  
FOREIGN KEY (`Cod_cliente`)  
REFERENCES `SistemaERP`.`Cliente` (`Cod_cliente`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)
```