

**Universidad de los Andes  
Facultad de Ingeniería  
Base de datos  
Semestre A-2017**

**Proyecto Base de datos A-2017  
1ra Fase  
Cadena de restaurantes**

**Alumnos:  
Gustavo Meza CI: 18.244.429  
Victor Albornoz CI: 25.151.526**

## Descripción del problema

Se requiere de una aplicación web donde el cliente pueda pedir su plato de comida, para que posteriormente le llegue a domicilio. La aplicación web también tendrá un apartado para las personas encargadas de gestionar los pedidos.

La empresa tiene varios locales, los cuales cada uno contiene varios empleados. Cada local podrá tener varios menú asociados.

Un mismo menú puede repetirse para varios locales. Al menú se le pueden agregar varios platos y los platos pueden asociarse a varios menú.

Cada plato puede tener o no extras. Cada plato forma parte de varios pedidos, y los pedidos estarán conformados por uno o más platos.

Un pedido opcionalmente puede tener bebidas asociadas. Y el pedido pertenecerá a una venta realizada cuando su estado cambie a “Entregado”. El pedido mantendrá los siguientes estados: En espera, procesando, enviado, entregado o cancelado.

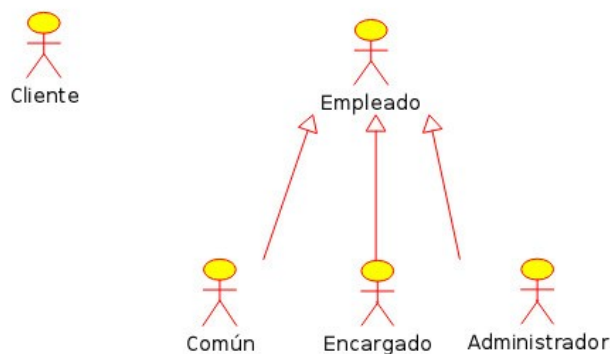
Un pedido también contiene uno de los dos tipos de pago disponibles: efectivo o transferencia.

Los usuarios pueden ser empleado, clientes o ambos. Ya que un cliente puede convertirse también en un empleado y un empleado puede realizar pedidos convirtiéndose en un cliente. Cabe resaltar que el usuario al menos debe ser uno los dos. Un cliente podrá estar asociado a varios pedidos.

Cada local tendrá varios empleados. El cargo del empleado podrá ser: Común, Encargado o administrador. Los que sean de tipo “encargado” serán los responsables de gestionar los pedidos..

Los datos bancarios solamente servirán para mostrarlos al cliente que realice un pedido y seleccione como método de pago “Transferencia bancaria”. Cada local podrá tener varias cuentas bancarias asociadas y las cuentas pueden pertenecer a varios locales. Se le mostrará al cliente las cuentas bancarias según el local que haya seleccionado.

## Modelo de Actores



**Cliente:**

- Cliente podrá iniciar sesión en la aplicación, una vez hecho podrá escoger entre los diferentes locales.
- Dispondrá de un menú donde observará los distintos platos disponibles con sus respectivas descripciones y el precio.
- Luego de escoger el plato podrá escoger una bebida de manera opcional.
- Podrá escoger un extra de manera opcional.
- Podrá escoger si pagar mediante transferencia bancaria o cobro a destino (efectivo).
- Luego de escoger el método de pago tendrá que rellenar la dirección de envío. Podrá escoger si utilizar su dirección personal, guardar una dirección de envío o reutilizar la dirección de envío almacena por ultima vez.
- Se le mostrará un resumen de su pedido y si está de acuerdo realizará el envío del pedido.
- Si el cliente escogió transferencia bancaria el cliente deberá de realizar el pago y luego introducir el número de confirmación, para que pueda ser enviada la comida.
- Si el cliente escogió “pago cobro destino” solamente tendrá que esperar a que llegue su pedido y podrá pagar en efectivo.
- El cliente podrá verificar el estado de su pedido (En espera, procesando, enviado, entregado o cancelado).
- El cliente podrá cancelar o editar su pedido siempre y cuando sea por transferencia y aún no haya confirmado la misma.
- Si el cliente escogió cobro a destino y quiere cancelar o editar su pedido deberá llamar al restaurante para que el administrador realice el cambio en el sistema.

**Empleado tipo “Encargado”:**

- El encargado podrá primeramente gestionar los locales, agregar y editarlos así como también podrá gestionar el menú asociado.
- Al gestionar el menú podrá editarlo para agregar o quitar platos. Podrá asociar un mismo menú a varios locales y viceversa.
- Podrá también agregar nuevos platos, cambiar le precio, eliminar, entre otras acciones.
- Podrá visualizar los pedidos entrantes, que serán ordenados por llegada (fecha).
- Podrá visualizar el estado del pedido, y también podrá cambiarlo.
- Verificará si el pedido tiene método de pago “Transferencia” o “Cobro a destino”.
- Si tiene transferencia bancaria deberá esperar que el cliente envíe el número de confirmación para poder procesar el pedido.
- Si tiene como método de pago cobro a destino deberá mandar el aviso para que se haga inmediatamente la comida y colocar el pedido en el estado “procesando”.
- Una vez que el pedido esté listo, podrá enviarlo y cambiar el estado a “Enviado”.
- Igual pasará cuando el cliente pague con transferencia bancaria, luego de confirmar la misma se preparará la comida y se enviará.
- Podrá realizar diferentes filtros sobre el listado de pedidos, por fecha, por estado del pedido, por cliente, combinación entre varias opciones, etc.

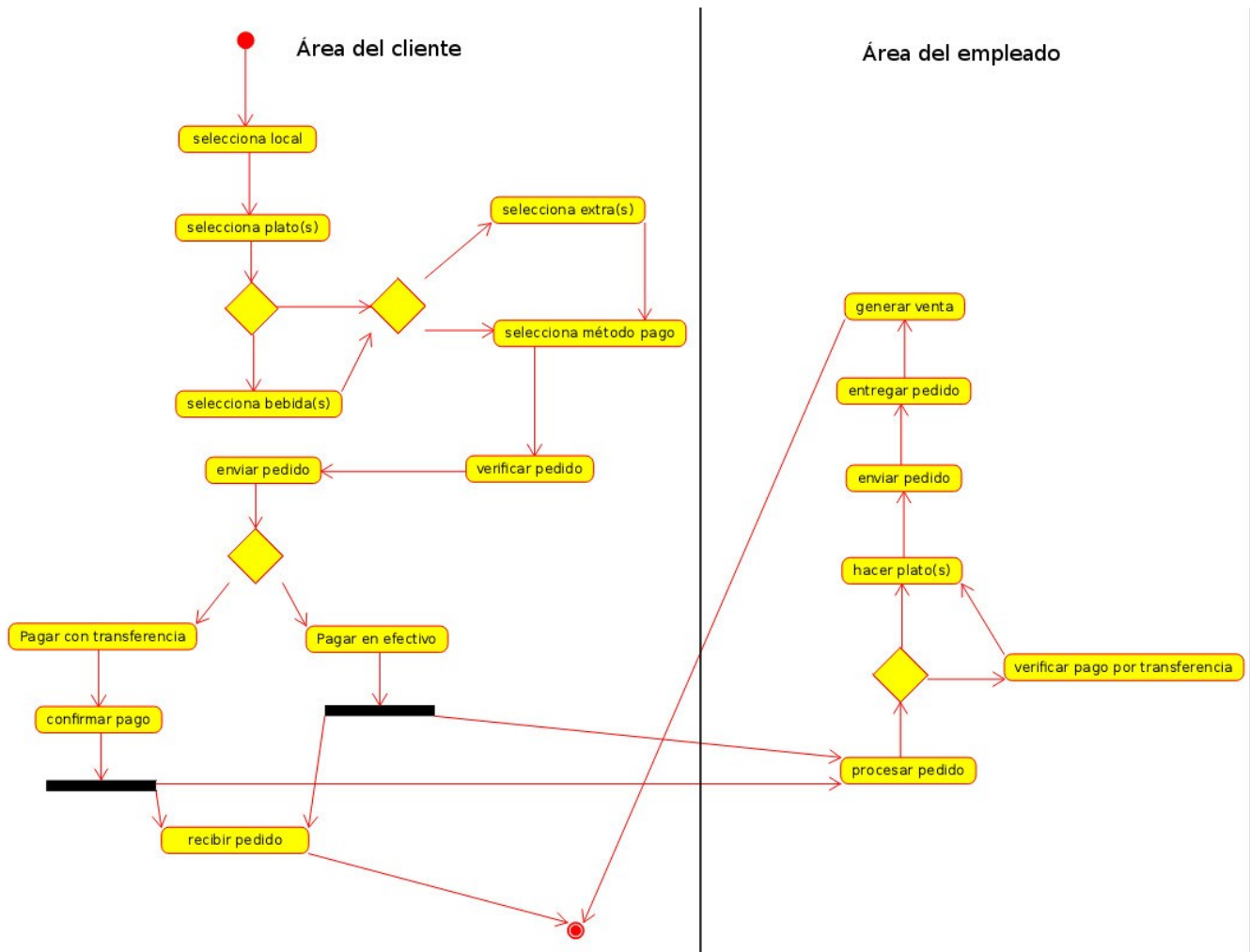
**Empleado tipo “Administrador”:**

- El administrador podrá realizar las mismas acciones que el encargado.
- Además podrá gestionar los datos bancarios que se le mostrarán al cliente para las transferencias y también gestionará las ventas que se realicen.

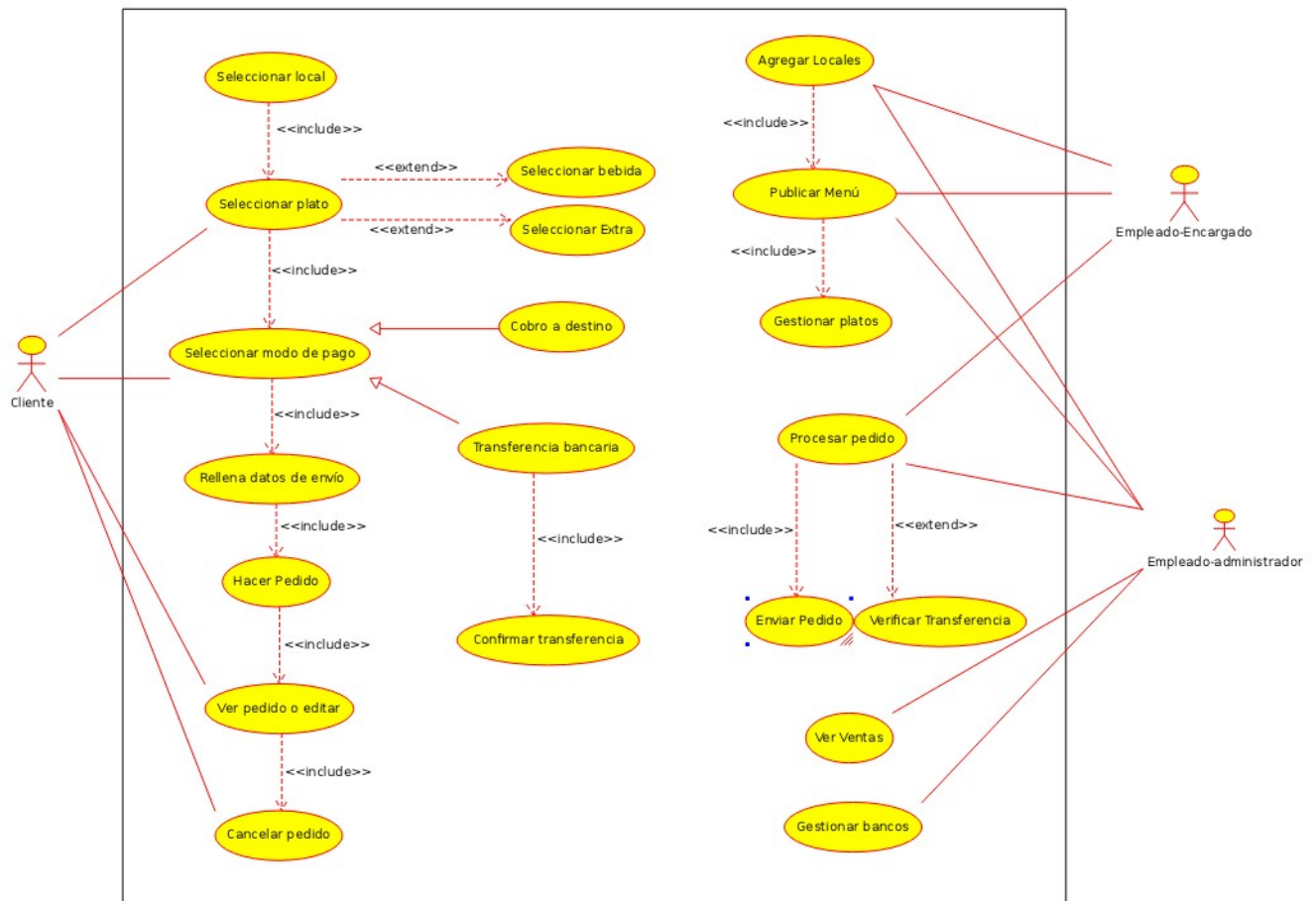
### Empleado tipo “Común”:

- Éste tipo de empleado no tendrá acceso para gestionar los pedidos.
- Solo podrá acceder cuando el administrador le ordene y le cambie el tipo a “encargado” o “administrador”.

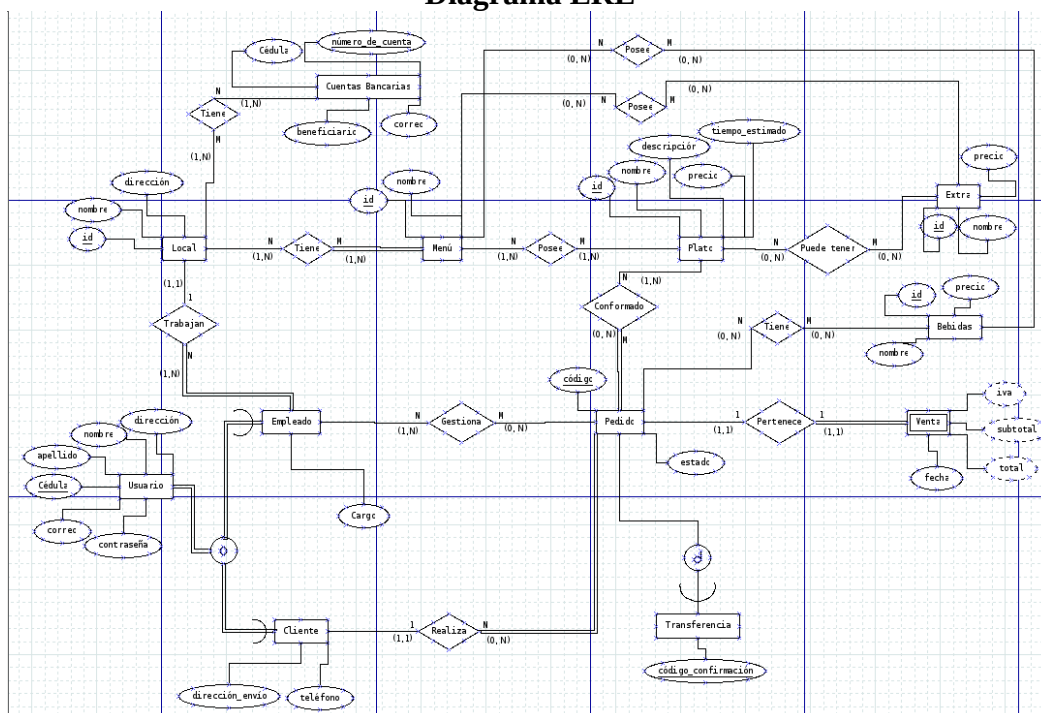
### Diagrama de actividades



## Caso de uso



## Diagrama ERE



## Modelo relacional

**Usuario** (cedula, nombre, apellido, dirección, correo, contraseña)  
**Empleado** (cedula, cargo, *idlocal*)  
**Cliente** (cedula, dirección\_de\_envio, telefono)  
**GestionaPedidos** (cedulaempleado, codigopedido)  
**Local** (id, nombre, direccion)  
**Cuentas\_bancarias** (numero\_de\_cuenta, cedula, beneficiario, correo)  
**Tiene\_cuenta\_bancaria** (*idlocal*, numero\_de\_cuenta)  
**Menú** (id, nombre)  
**LocalMenu** (*idlocal*, *idmenu*)  
**Plato** (id, nombre, descripción, precio, tiempo\_estimado)  
**MenuPlato** (*idmenu*, *idplato*)  
**TenerExtra** (*idplato*, *idextra*)  
**Extra** (id, nombre, precio)  
**MenuExtra** (*idmenu*, *idextra*)  
**PedidosyPlatos** (codigopedido, *idplato*)  
**Pedido** (código, estado, *cedulacliente*)  
**Transferencia** (código, codigo\_confirmacion)  
**Venta** (codigopedido, iva, subtotal, total, fecha)  
**Tienebebida** (codigopedido, *idbebida*)  
**Bebidas** (id, nombre, precio)  
**MenuBebida** (*idmenu*, *idBebida*)

### Restricciones:

- a) cedula en **Empleado** hace referencia a **Usuario**(cedula).
- b) idlocal en **Empleado** es clave foránea en **Local**(id).
- c) cedula en **Cliente** hace referencia a **Usuario**(cedula).
- d) cedulaempleado y codigopedido en **GestionaPedidos** son clave foránea de **Empleado**(cedula) y **Pedido**(codigo), respectivamente.
- e) idlocal y numero\_de\_cuenta en **Tiene\_cuenta\_bancaria** son claves foráneas de **Local**(id) y **Cuentas\_bancarias**(numero\_de\_cuenta) respectivamente.
- f) idlocal e idmenu en **LocalMenu** son claves foráneas de **Local**(id) y **Menu**(id), respectivamente.
- g) idmenu e idplato en **MenuPlato** son claves foráneas de **Menu**(id) y **Plato**(id), respectivamente.
- h) idplato e idextra en **TenerExtra** son claves foráneas de **Plato**(id) y **Extra**(id), respectivamente.
- i) codigopedido e idplato en **PedidosyPlatos** son clave foráneas de **Pedido**(codigo) y **Plato**(id), respectivamente.
- j) cedulacliente en **Pedido** es clave foránea de **Cliente**(cedula).
- k) codigopedido en **Venta** es clave foránea de **Pedido**(codigo).
- l) codigopedido e idbebida en **Tienebebida** son clave foráneas de **Pedido**(codigo) y **Bebida**(id), respectivamente.
- m) idmenu y idextra en **MenuExtra** son clave foráneas de **Menu**(id) y **Extra**(id), respectivamente.
- n) idmenu y idBebida en **MenuBebida** son clave foráneas de **Menu**(id) y **Bebida**(id), respectivamente.

## Normalización

**1FN:** ninguna relación posee atributos multivaluados, por lo tanto ya está en 1FN.

**2FN:** para nuestro modelo no existe ninguna relación que tenga algún atributo que dependa funcionalmente de una clave parcial, por lo tanto ya está en 2FN.

**3FN:** **Cuentas\_bancarias** (numero de cuenta, cedula, beneficiario, correo), en ésta relación existe una dependencia funcional de doble implicación entre cedula y beneficiario (cedula <--> beneficiario), por lo tanto se normaliza.

**Cuentas\_bancarias** (numero de cuenta, *cedulaBeneficiario*, correo),

**Beneficiario** (cedula, nombre)

**El modelo relacional normalizado a 3FN quedaría:**

**Usuario** (cedula, nombre, apellido, dirección, correo, contraseña)

**Empleado** (cedula, cargo, *idlocal*)

**Cliente** (cedula, dirección\_de\_envio, telefono)

**GestionaPedidos** (*cedulaempleado*, *codigopedido*)

**Local** (id, nombre, direccion)

**Cuentas\_bancarias** (numero de cuenta, *cedulaBeneficiario*, correo)

**Beneficiario** (cedula, nombre)

**Tiene\_cuenta\_bancaria** (*idlocal*, *numero de cuenta*)

**Menú** (id, nombre)

**LocalMenu** (*idlocal*, *idmenu*)

**Plato** (id, nombre, descripción, precio, tiempo\_estimado)

**MenuPlato** (*idmenu*, *idplato*)

**TenerExtra** (*idplato*, *idextra*)

**Extra** (id, nombre, precio)

**MenuExtra** (*idmenu*, *idextra*)

**PedidosyPlatos** (*codigopedido*, *idplato*)

**Pedido** (código, estado, *cedulacliente*)

**Transferencia** (código, codigo\_confirmacion)

**Venta** (*codigopedido*, iva, subtotal, total, fecha)

**Tienebebida** (*codigopedido*, *idbebida*)

**Bebidas** (id, nombre, precio)

**MenuBebida** (*idmenu*, *idBebida*)

**El modelo relacional en SQL puede verlo en el archivo proyectoDB.sql**