

# Ejemplo de una aplicación de compra de productos usando una API de desarrollo

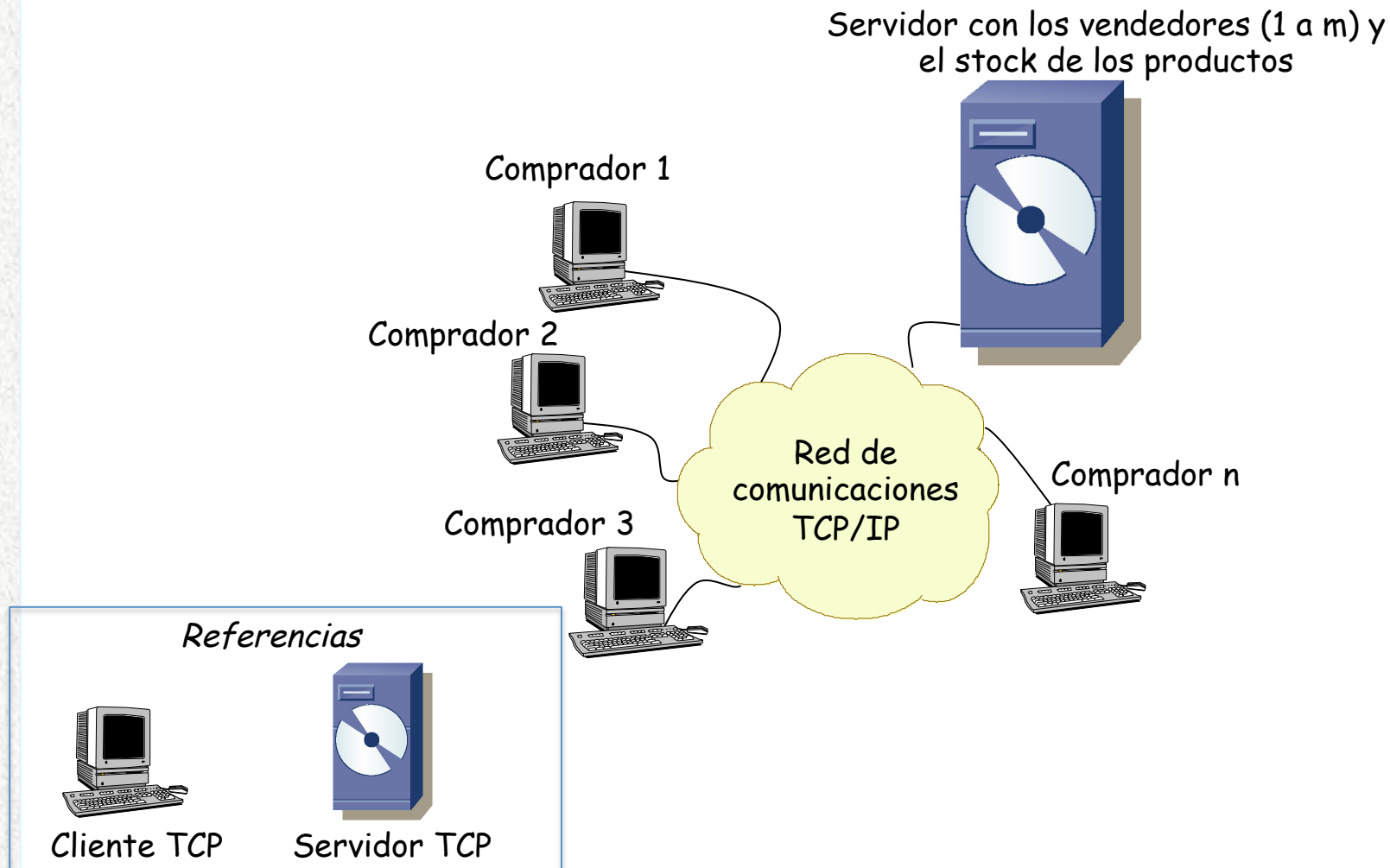
Protocolos de mensajes para las colas y las comunicaciones

Prof. María Feldgen

## Enunciado del Problema

- Un sistema con múltiples compradores distribuidos compra productos a una empresa con múltiples vendedores que atienden en un sitio centralizado.
- El comprador tiene un Id que es su número de cliente. El comprador inicia la compra enviando sus datos (nombre de la empresa, CUIT y tipo de factura) a algún vendedor que esté disponible.
- Cada vendedor tiene un id, que es el número de vendedor. Un vendedor libre atiende la recepción de los datos del cliente y de acuerdo a alguna acción interna (en nuestro caso, random), decide si aprueba o no la compra del cliente, respondiendo S o N al comprador.
- Si el comprador está autorizado a comprar,
  - envía cada producto y la cantidad solicitada, con una indicación si quiere comprar mas productos. El vendedor responde con la cantidad que puede vender según el stock disponible.
  - Cuando el comprador envió todos los productos, el vendedor arma la factura, calcula el total y el iva, según el tipo de factura (A: con iva discriminado, otro tipo con iva incluido en el total) y la envía al comprador.
  - El comprador muestra la factura y termina.
- caso contrario:
  - el comprador termina.
- En ambos casos el vendedor espera por un nuevo cliente.

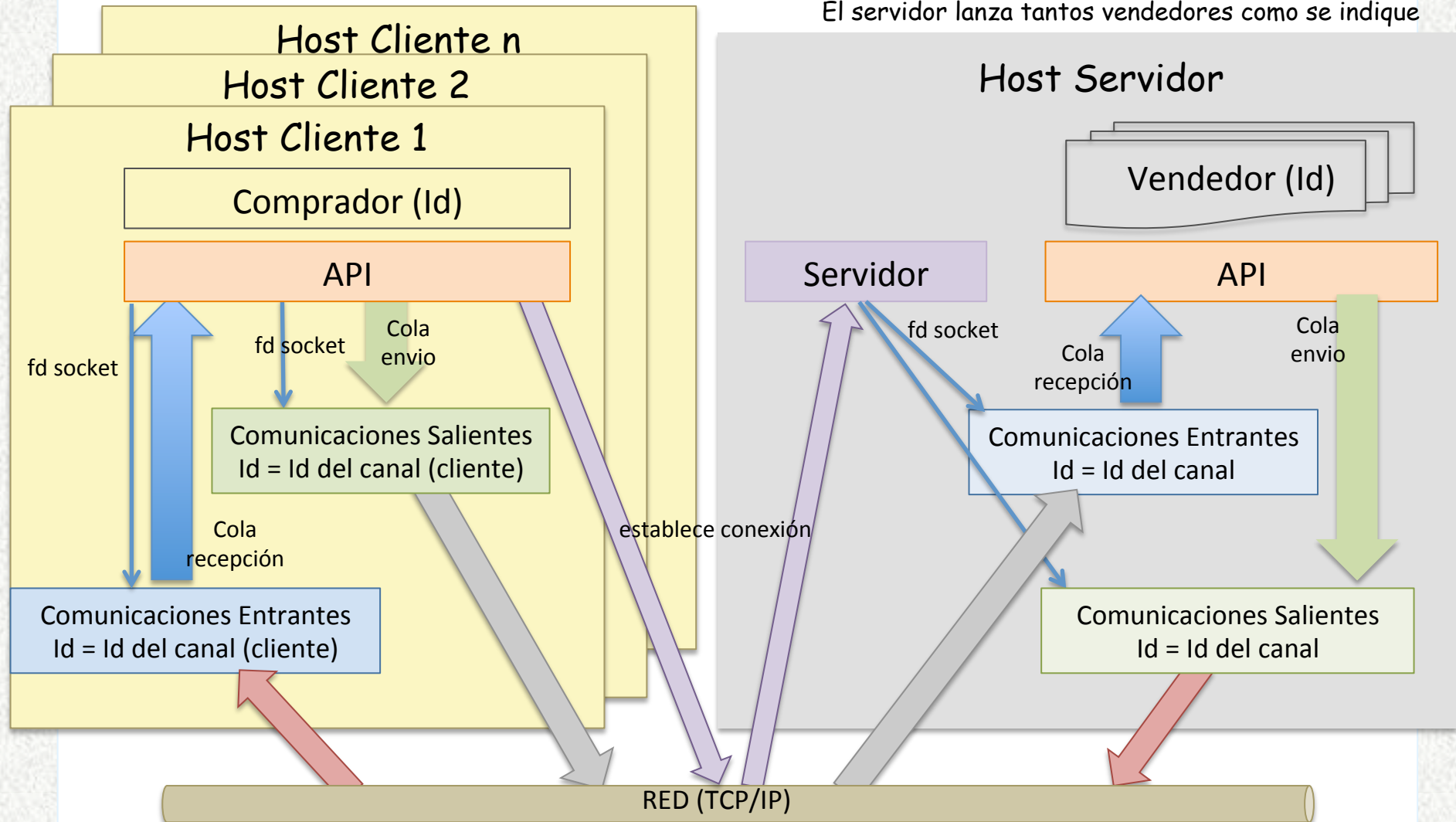
## Ambiente Cliente/Servidor



## Aplicación de Venta en un ambiente Cliente/Servidor

Corre un solo comprador por máquina y se le pasa el N° de Id

El servidor lanza tantos vendedores como se indique



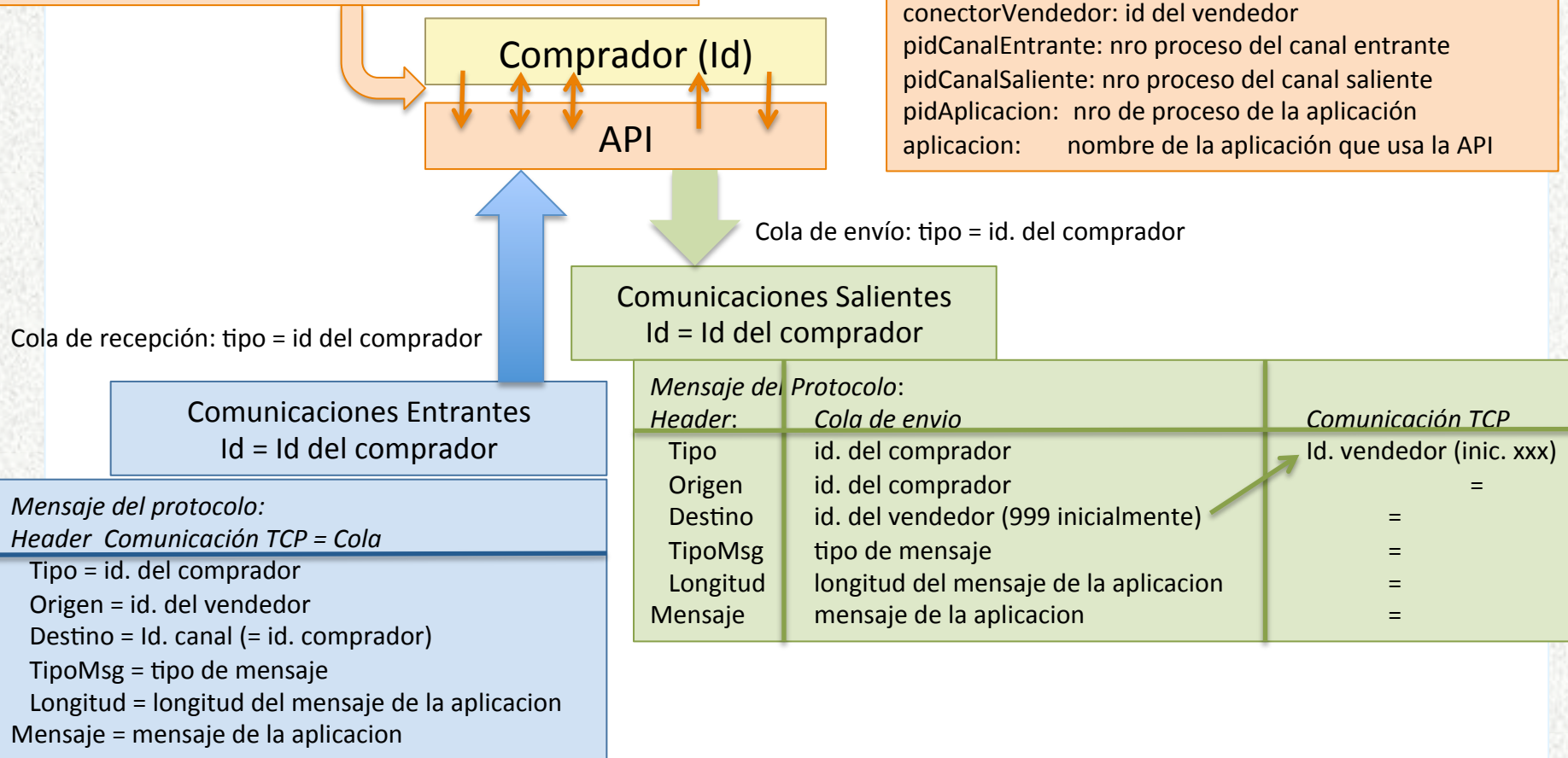
## Cliente (un socket por cliente)

### Primitivas:

```
int iniciarAplicacion(char *,char *,int, CONTROL *);
int enviarDatosCliente(void *, int, void *, int, CONTROL *);
int pedirProducto(void *, int, void *, int, CONTROL *);
int recibirFactura (void *, int, CONTROL *);
int cerrarAplicacion(CONTROL *);
```

### Información de Control:

```
recepcion:    fd cola de recepcion
envio:        fd cola de envio
socketfd:     fd del socket conectado
cliente:      Id. del comprador
conectorCliente: id del comprador
conectorVendedor: id del vendedor
pidCanalEntrante: nro proceso del canal entrante
pidCanalSaliente: nro proceso del canal saliente
pidAplicacion: nro de proceso de la aplicación
aplicacion:   nombre de la aplicación que usa la API
```





## Servidor (un socket por cliente)

### Primitivas:

```
int iniciarAplicacion(char *, char *,int, CONTROL *);
int aceptacionCliente(void *, int, CONTROL *);
int obtenerCliente(void *, int, CONTROL *);
int compraProducto(void *, int, CONTROL *);
int confirmacionProducto(void *, int, CONTROL *);
int entregarFactura(void *, int, CONTROL *);
int cerrarVenta(CONTROL *);
int actualizarStock( int, int, PRODUCTO *, CONTROL *)
```

Vendedor (Id)

API

Cola de envío:  
tipo = canal

### Información de Control:

mutex: fd del semaforo de Exclusion Mutua  
shmid: fd shared memory productos  
shmem: stock de productos en shared memory  
recepcion: fd cola de recepcion  
envio: fd cola de envio  
conectorVendedor: id del vendedor  
conectorVendedorGral: Id de vendedores libres  
conectorCliente: id del comprador  
conectorCanal: Id del canal de comunicaciones  
pidAplicacion: nro proceso de la aplicacion  
aplicacion: nombre de la aplicación que usa la API

Cola de recepción: tipo = id del vendedor  
inicialmente: 999 atiende cualquiera  
luego Id. del vendedor

Comunicaciones Salientes  
Id = Id del canal

uno por conexión TCP abierta

### Mensaje del Protocolo:

Header:	Cola de envío	Comunicación TCP
Tipo	id. de comunicaciones	Id. comprador
Origen	id. del vendedor	=
Destino	id. del comprador	=
TipoMsg	tipo de mensaje	=
Longitud	longitud del mensaje de la aplicacion	=
Mensaje	mensaje de la aplicacion	=

uno por conexión  
TCP abierta

Comunicaciones Entrantes  
Id = Id del canal

### Mensaje del Protocolo:

Header:	Comunicación TCP	Cola Recepcion
Tipo	id. del vendedor	=
Origen	id. del comprador	=
Destino	id. del vendedor (999 inicialmente)	id. del canal
TipoMsg	tipo de mensaje	=
Longitud	longitud del mensaje de la aplicacion	=
Mensaje	mensaje de la aplicacion	=