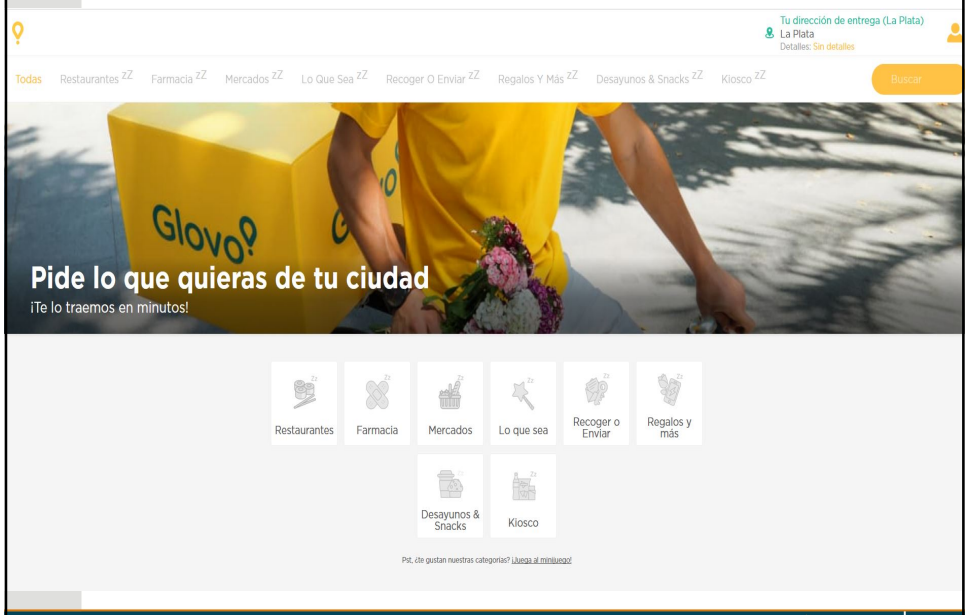
Ejemplo: Gloovo



Lifia

Objetivos del Ejemplo

- Repasar los conceptos basicos sobre un diseño realista
- Vamos a analizar el diseño sin preocuparnos por COMO llegamos a el (por ahora).

Se trata de describir con objetos la funcionalidad mas importante de un sistema del tipo Glovo o PedidosYa, que permite que cualquier usuario registrado (cliente) pida un producto en un comercio registrado y alguien (otros usuarios especiales que se postulan como “Gloovers") le lleve el producto a la casa.

Con un conjunto de opciones desplegables el "cliente" elige producto (o producto y negocio en el que lo quiere comprar) y registra un pedido. En ese momento sabe el precio que pagará tanto por el producto como por el envío. El sistema todavía no puede estimar un tiempo de entrega. El “sistema” recibe el pedido y lo postea entre los usuarios “Gloovers". Cuando alguno de ellos lo “toma”, el pedido se asigna al "acarreador“ (un gloover) y se comunica al negocio que provee el producto para que lo prepare. El Gloover va al negocio y retira el producto. Utiliza la aplicación para confirmar que retiro el producto. Se dirige a la dirección de entrega y entrega el producto. Utiliza la aplicación para confirmar que lo entregó.

Cuando el cliente recibe el pedido, utiliza la aplicación para confirmar la recepción. En ese momento, se carga el costo a su medio de pago, se para al negocio, y se paga al "acarreador". Adicionalmente el cliente puede calificar al gloover con un puntaje y un comentario

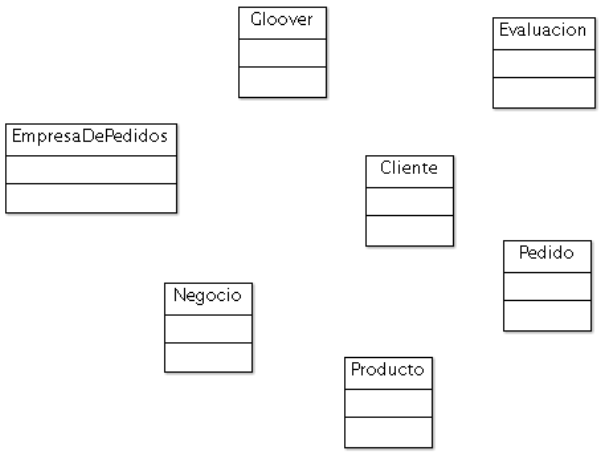
Lifia

Contexto del Ejemplo y restricciones

- No vamos a preocuparnos (ahora) por el diseño de las interfases o los formularios.
- No vamos a preocuparnos por la comunicacion entre los usuarios (desde telefonos, tabletas, computadoras, etc) y la aplicacion
- Vamos a ignorar el "almacenamiento" de los datos (en archivos, bases de datos, centralizado, distribuido, etc)

- Sin embargo podemos decir:
  - Es correcto asumir que esos aspectos no implicaran cambio alguno en el sistema (nuevas interfases por ejemplo, nuevos dispositivos, etc)
  - La "comunicacion" es transparente al software (casi siempre)

"Diseño" inicial



Casos de Uso del sistema Gloovo

Lifia

Suposiciones iniciales

- Inicialmente no tenemos gloovers, ni clientes, ni negocios, ni productos.

Lifia

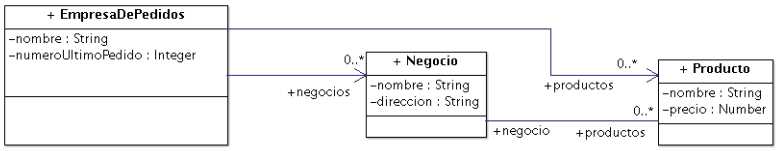
Como entender/diseñar Gloovo

-El sistema “arranca” con la creacion de los objetos de interfaz apropiados y un objeto de la clase EmpresaDePedidos (llamemosle glovo).

-Para simplificar asumimos que todos los eventos generados por la interfaz son atendidos por este objeto, aunque un diseño mas realista podria incluir otros objetos (para hacer de “fachada” del sistema)

-Asumimos tambien que los negocios y productos se agregaron al sistema (quizas con interfaces adaptadas a los dueños de los negocios

(**Observar la notacion**)



Lifia

Creacion del objeto gloovo

gloovo:=EmpresaDePedidos new
gloovo initialize (“Gloovo S.A”)

-Esto nos crea un objeto gloovo cuya variable de instancia nombre sera “Gloovo S.A”

El metodo

initialize (label)
nombre:=label
productos:=Collection new
negocios:= Collection new
clientes:= Collection new
gloovers:= Collection new

Lifia

Algo mas modular y preparado para cambios

initialize (label)
nombre:=label
self initializeCollections

initializeCollections
productos:=SomeCollection new
negocios:= SomeCollection new
clientes:= SomeCollection new
gloovers:= SomeCollection new

(QUE PEDIREMOS DE ESTAS COLECCIONES?)

Lifia

Agregar un Negocio

- En Clase Empresa de Pedidos

agregarNegocio (n, d)
negocio:=Negocio new
negocio inicializar (n,d)
negocios add (n)

Lifia

Inicializar Negocio

- En Clase Negocio

```
inicializar (n, d)
 nombre:=n
 direccion:= d
 productos:= Collection new
```


Agregar un Producto a Gloovo

- Este proceso depende de nuestro esquema de negocios.
- En Empresa de Pedidos

```
agregarProducto (n,p, negocio)
   produ:= Producto new
   produ inicializar (n,p,negocio)
   productos add (produ)
   negocio agregarProducto (produ)
```

Agregar un producto a un Negocio

- En Clase Negocio

```
agregarProducto (p)
 productos add (p)
```

Lifia

Es realista?

- Da lo mismo agregar los productos de la pizzeria del barrio que los productos de Garbarino?
- Problemas?: Cantidad de productos, Quien lo hace?
- Opciones?: Hacerlo “por programa”

Lifia

Implementando casos de uso del sistema

- **Registrarse como Cliente**
  - El usuario interactua con un formulario, llena ciertos campos y envia el formulario que es recibido por el objeto gloovo
  - gloovo (instancia de EmpresaDePedidos) ejecuta el metodo de dicha clase que debe:
    - Crear el objeto Cliente
    - Agregarlo a la "coleccion" de clientes del Negocio

registrarCliente (nombre, direccion, email, password)

c:= Cliente new
c inicializar (nombre, direccion, email, password)
clientes add (c)

DONDE CHEQUEAMOS CONSISTENCIA DE ESTA INFORMACION?

Lifia

Clase Cliente-Inicializacion

Inicializar (nomb, dir, mail, pass)

nombre:= nomb
direccion:= dir
email:= mail
password:= pass

Lifia

Temas de discusion en este metodo

- Que es “direccion”
  - String? “Calle 9 Nro 1354 La Plata Buenos Aires”
  - Un objeto de otra clase?
- Una suposicion razonable es que la interfaz crea objetos de clase “Direccion” con atributos estandar (calle, numero, ciudad, codigo postal,provincia, pais) y que el Cliente tiene una variable de instancia “direccion” que es una referencia a dicho objeto

Lifia

Otros casos de Uso

- **Registrarse como Gloover**
  - Este proceso puede ser mas “largo” y requerir chequeos no atómicos por seguridad. El candidato llena un formulario y lo submite. El objeto gloovo (en realidad el objeto referenciado por la variable gloovo) recibe el mensaje y el metodo correspondiente deberia:
    - Pedir validar los datos basicos del Gloover
    - Disparar un proceso de validacion mas elaborado (offline) y luego el responsable (con otro form) disparara el metodo agregarGloover

registrarGloover (nombre, direccion, email,…..)

g:= Gloover new

g inicializar (…..)

self chequearDatos (g)

……En otro momento, y disparado desde otra interfaz…

aceptarGloover (g)

gloovers add: g

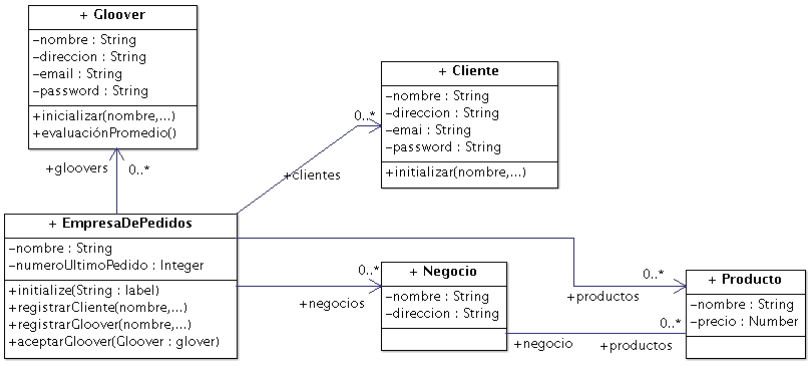
Lifia

Temas en este metodo y otros

- Notificacion de los gloovers? Donde? Cuando?
- Validacion “profunda”: otro sistema? El mismo que se comunica con otros?
- Variables de instancia de EmpresaDePedidos?

Lifia

Diseño hasta ahora



Lifia

El usuario compra....

- Para comprar el cliente debe:
  - Loguearse
  - Poner cosas en el carrito
  - Hacer el pedido: con todo lo que hay en el carrito

- Mensaje login (email, password) Enviado a gloovo

```
login (email, password)
  cliente:= clientes findAndValidate (email, password)
  return (cliente)
```

Tenemos que ver como tratar el caso de error!!!

Lifia

Manejar el carrito de compras

- Necesitamos una clase Carrito. Un objeto carrito (por ahora) tiene una coleccion de productos (en esta version un producto tiene un negocio asociado).
- Quien conoce al carrito? (**ver inicializaciones**)

- Mensaje agregarProducto (p) en Clase Cliente

```
agregarProducto (p)
  carrito agregar (p)
```

Lifia

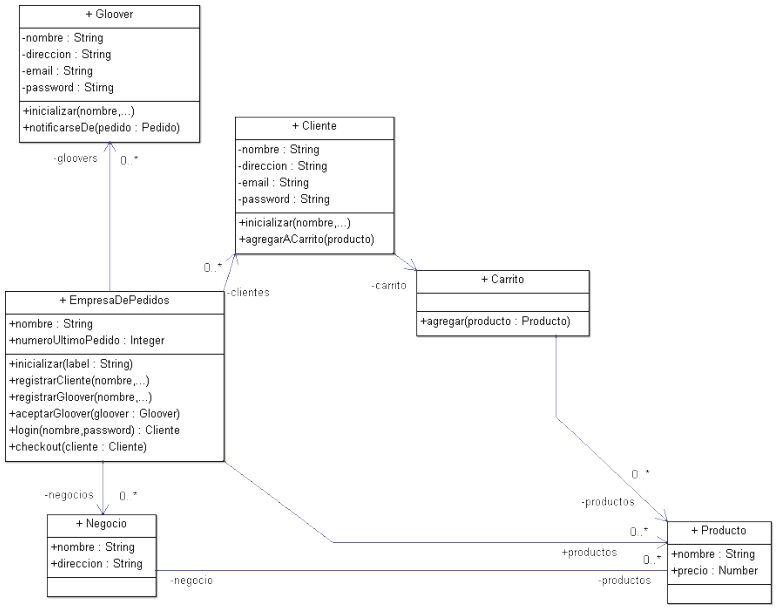
Inicializacion Cliente corregida

Inicializar (nomb, dir, mail, pass)

nombre:= nomb
direccion:= dir
email:= mail
password:= pass
carrito:= Carrito new

Lifia

ClienteConCarrito



Lifia

Generar un pedido

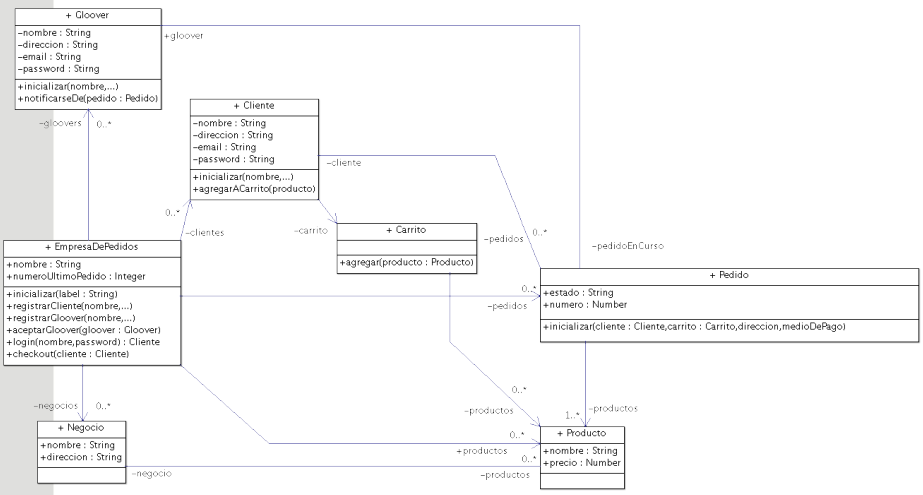
- El cliente decide “cerrar” su pedido y comprar
- Mensaje checkout (cliente) en objeto gloovo
- Por que en gloovo y no en cliente? (ver proceso checkout)

```
checkout (cliente, direccionEnvio,medioDePago)
  p:= Pedido new
  p inicializarCon (cliente, cliente carrito, direccionEnvio,
                    medioDePago)
  gloovers notificarseDe (p)
  pedidos add (p)
```

Observar relacion Cliente-Pedido, Metodo carrito en Cliente?

Lifia

Diseño actual



Lifia

Notificacion de un pedido nuevo

- En la clase Gloover tenemos un metodo
  notificarseDePedido (p)
- Supongamos que la notificacion es via Whatsapp.
- En la clase Gloover tendriamos codigo especifico necesario para enviar un Whatsapp desde nuestro sistema.
- No parece ser una incumbencia de esa clase.
- Una solucion mejor es delegar esta tarea en un objeto de una clase especifica (ComunicacionWhatsapp)

Lifia

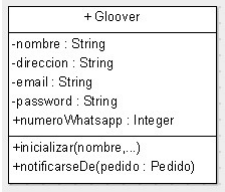
Notificacion…

- Entonces en Clase Gloover

```
notificarseDePedido (p)
  c:=ConexionConWhatsapp new
  c enviarMensaje (numeroWhatsapp, p)
```

Entonces:
un gloover debe conocer su numeroWhatsapp

Que pasa con p (el pedido actual)?
Que mandamos por whatsapp? UN TEXTO
Puede recibir el objeto c un pedido?



Lifia

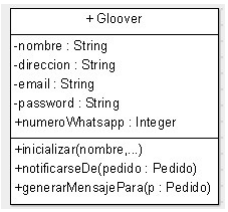
Notificacion….

notificarseDePedido (p)

c:=ConexionConWhatsapp new

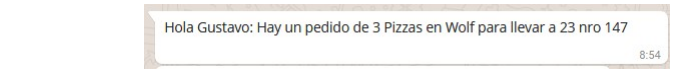
texto:= self generarMensajePara (p)

c enviarMensaje (numeroWhatsapp, texto)



generarMensajePara (p)

-que necesitamos aca para retornar un mensaje util para el gloover?



Lifia

Un gloover se postula…

- El gloover recibe una notificacion de pedido y se postula
- Donde esta el metodo correspondiente?
- Quien lo “dispara”?

- Opciones:
  - En la clase Gloover

    postularsePara (p:Pedido) Y Tendria que avisarle a gloovo (lo conoce?)

  - En la clase EmpresaDePedidos

    postularGlover (g:Gloover, p: Pedido)

Lifia

Un gloover se postula….y se asigna

- Este donde este el “primer” metodo (dependera de la configuracion de la interfaz entre otras cosas), en EmpresaDePedidos se procesa el pedido

postularGloover (g:Gloover, p:Pedido)

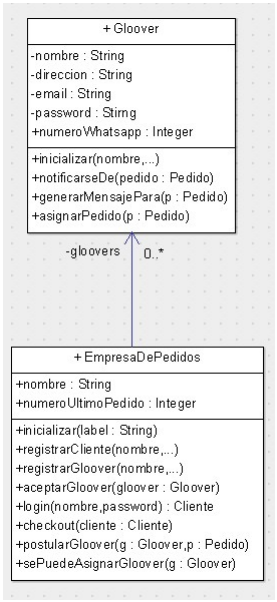
self sePuedeAsignarGloover (g) Mirar si el tipo tiene denuncias, etc

g asignarPedido (p)

LE TENEMOS QUE AVISAR AL CLIENTE? Como hacemos?

Lifia

EmpresaDePedidos y Gloover modificados



Lifia

Asignar un pedido a un gloover, avisarle al usuario

- En la Clase Gloover tenemos el metodo

```
asignarPedido (pedido)
  pedidoEnCurso:=pedido
  pedido asignarGloover (self)
```

- En la clase Pedido

```
asignarGloover (g)
 gloover:=g
 estado:= “glooverAsignado”
```

CUAL ERA EL ESTADO INICIAL?

Lifia