



Universidad Católica Andrés Bello

Ingeniería Informática

Desarrollo del Software

Informe Final

Courier UCAB

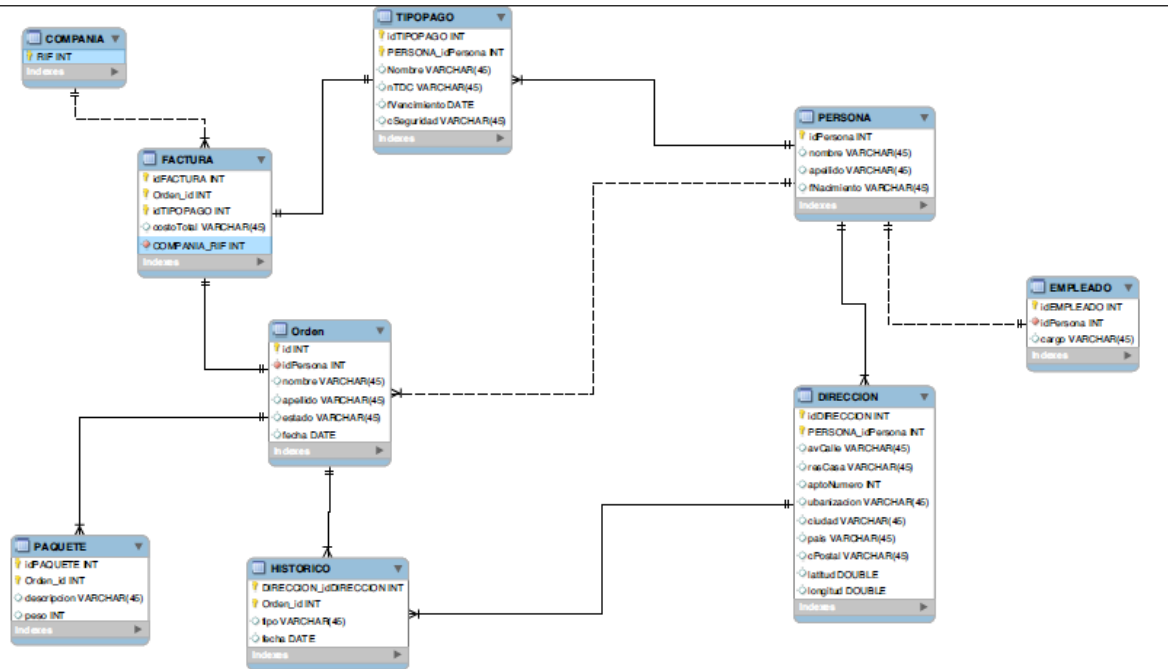
Drey Laya

Ricardo Sampayo

Alberto Grespan

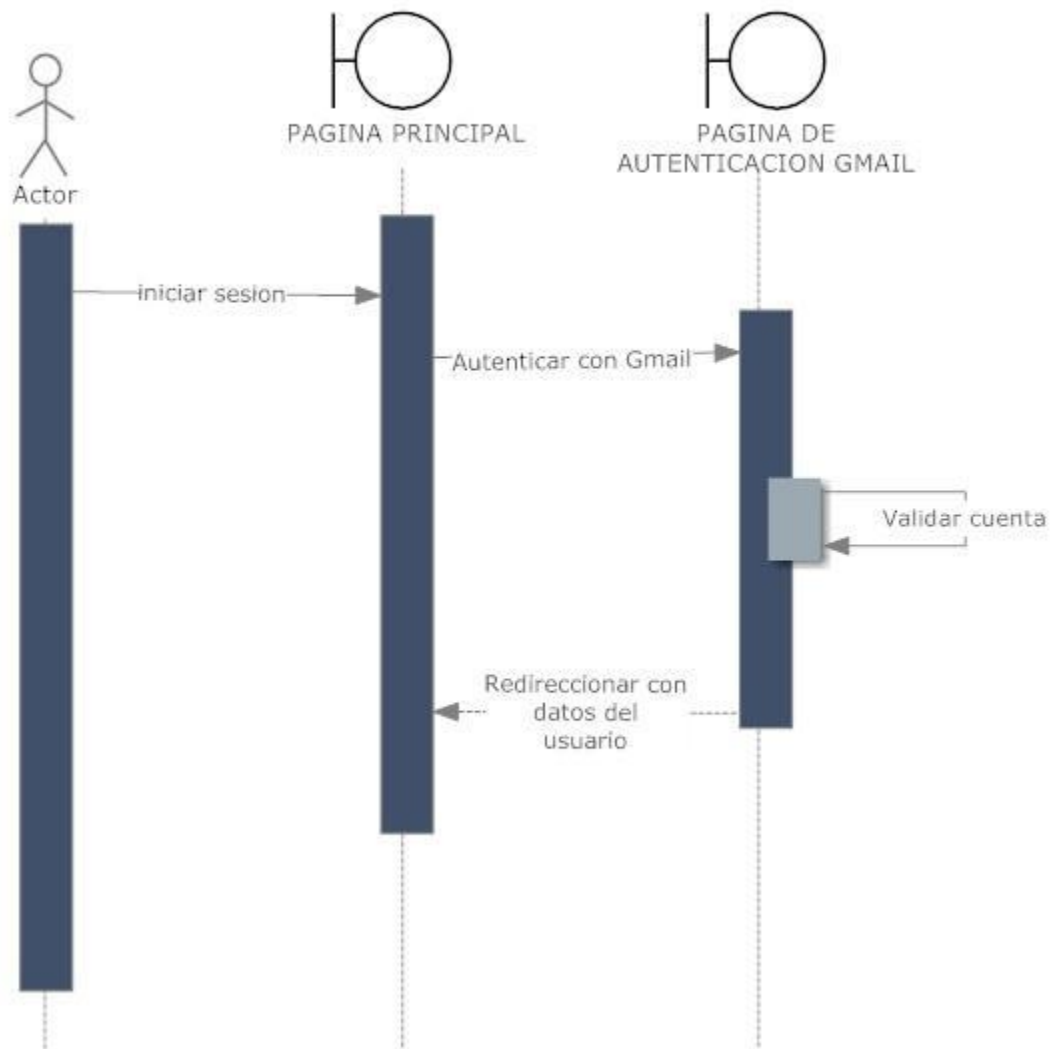
Caracas, Julio de 2011

MODELO ENTIDAD RELACION

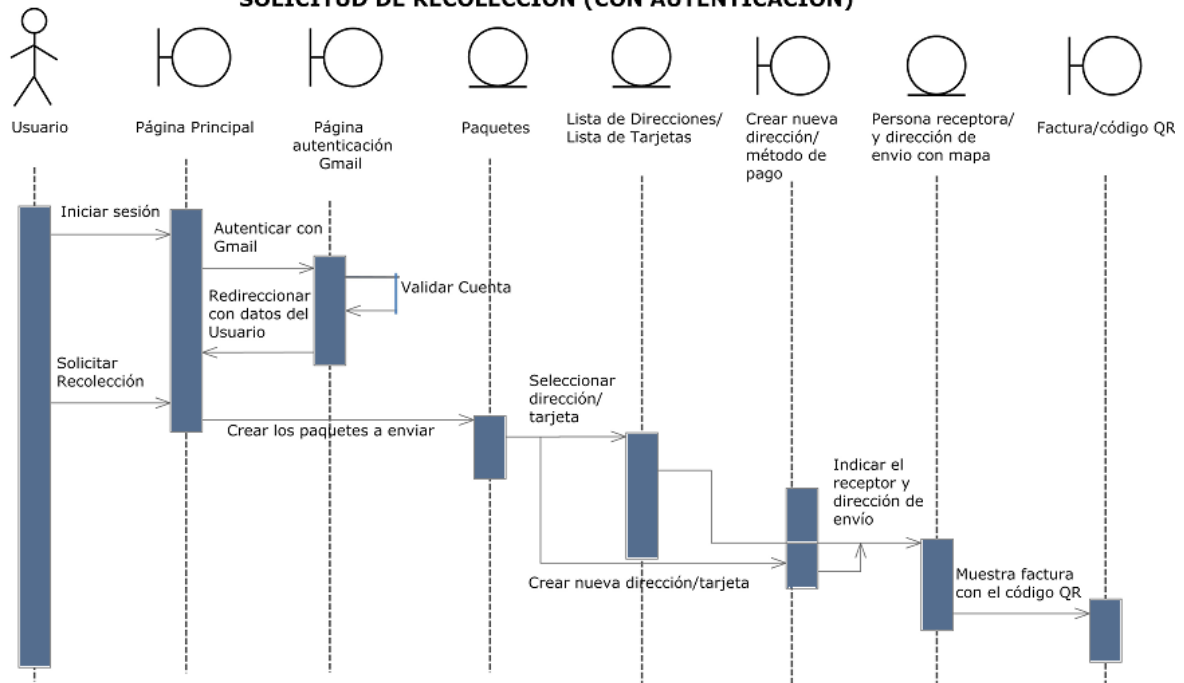


DIAGRAMAS DE SECUENCIA

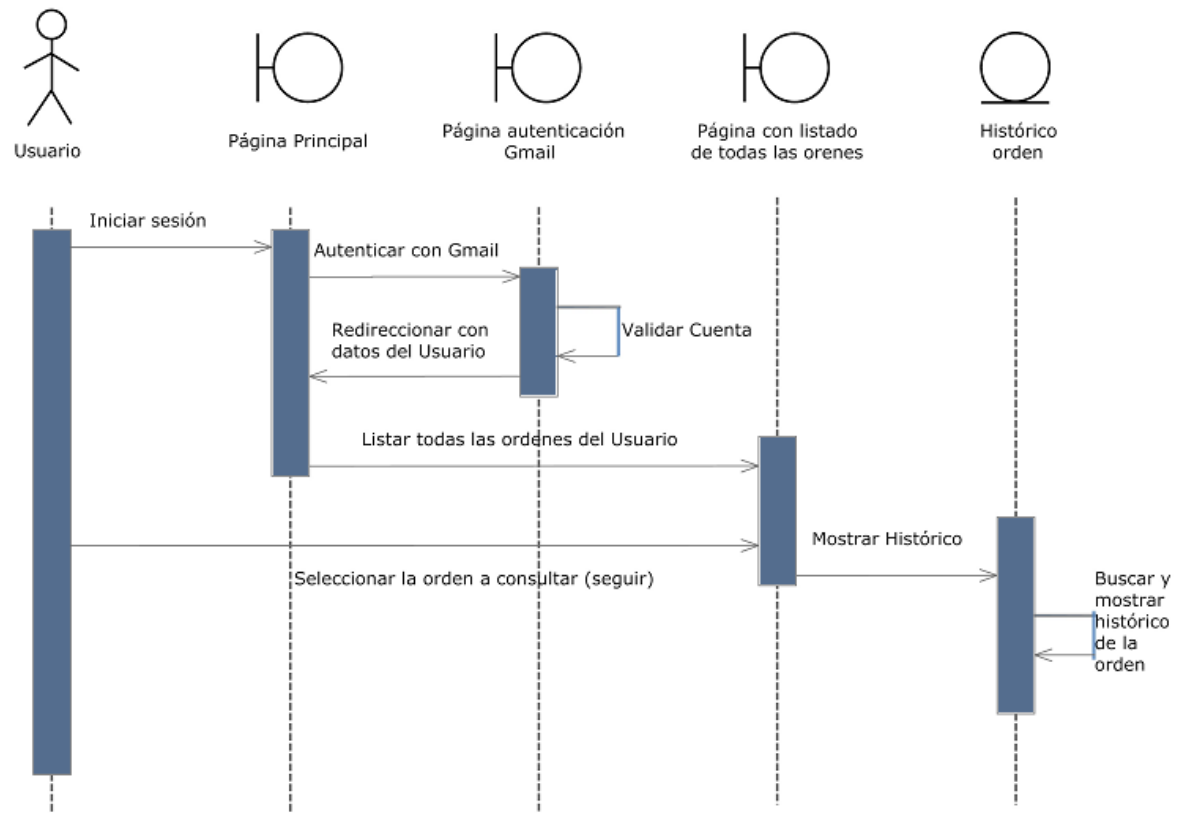
Escenario Login con Autenticación



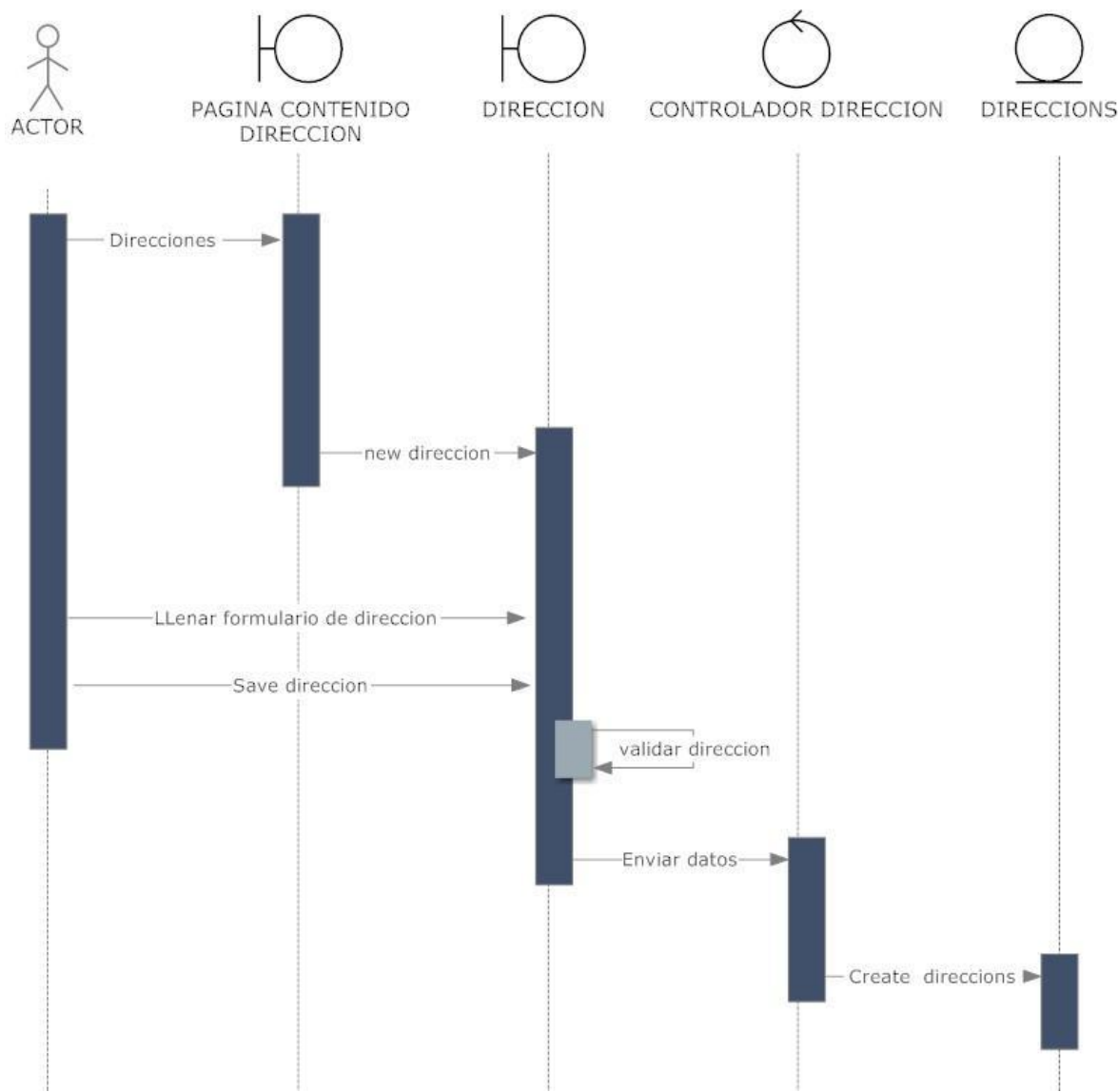
SOLICITUD DE RECOLECCIÓN (CON AUTENTICACIÓN)



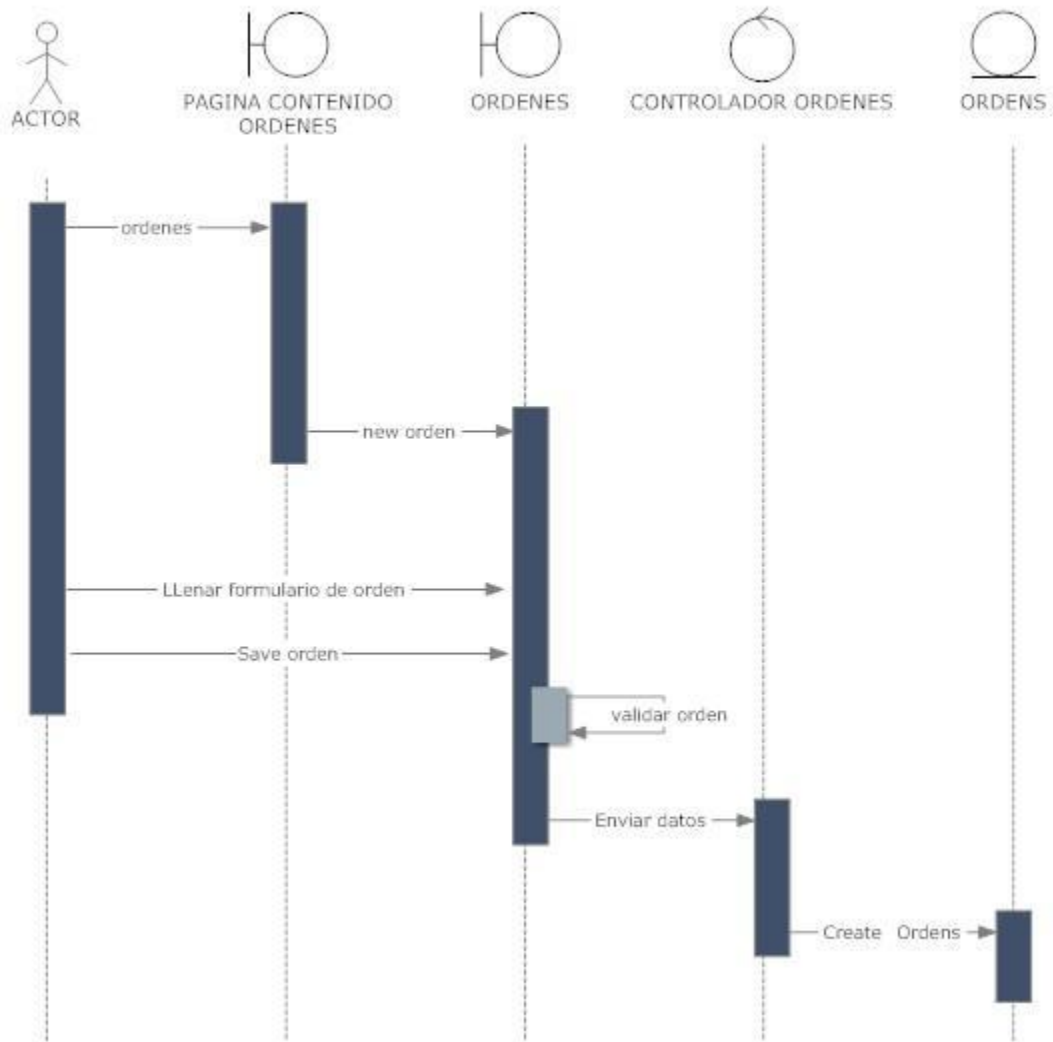
SOLICITUD DE RASTREO (SEGUIMIENTO DE PAQUETE INICIANDO SESIÓN)



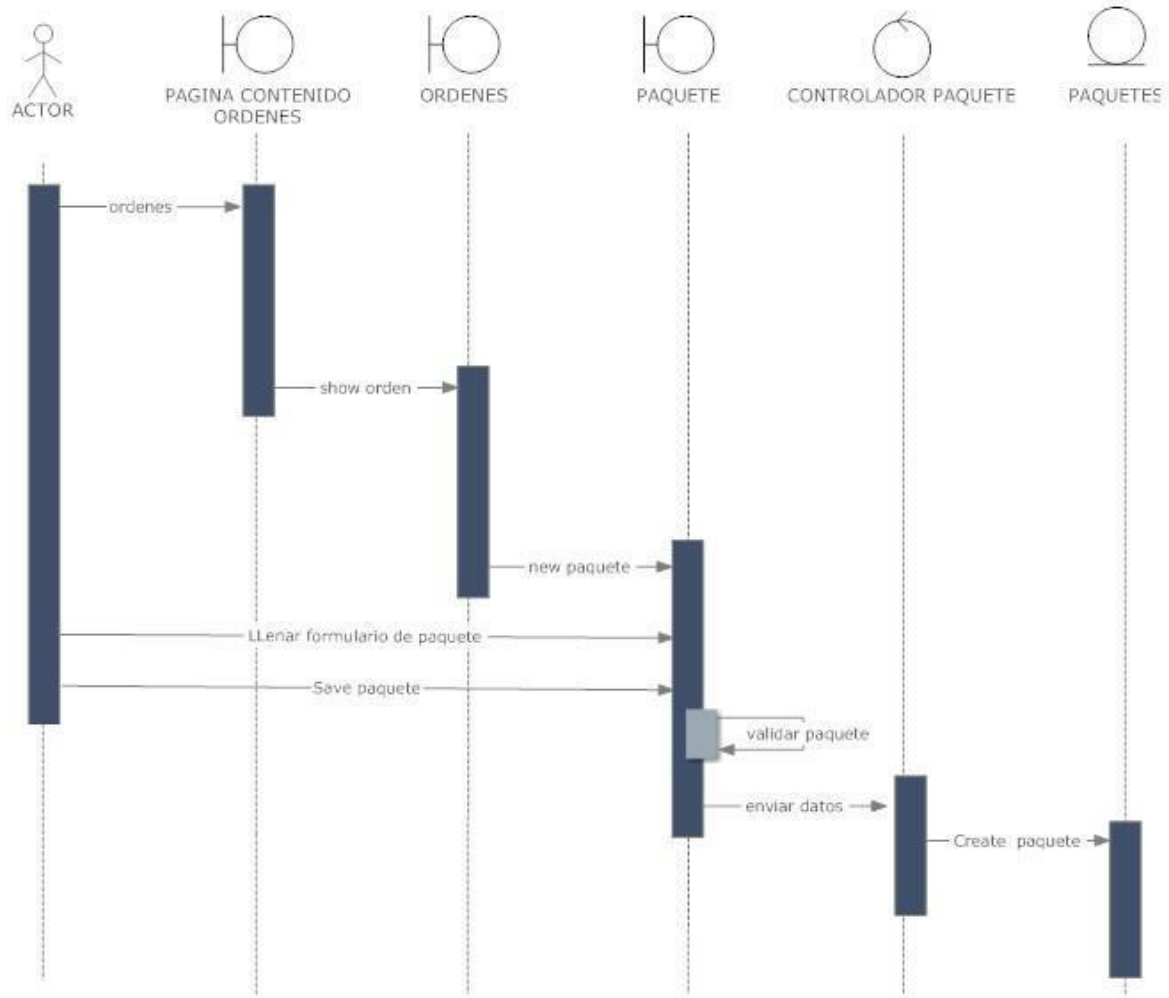
ESCENARIO CREAR DIRECCION



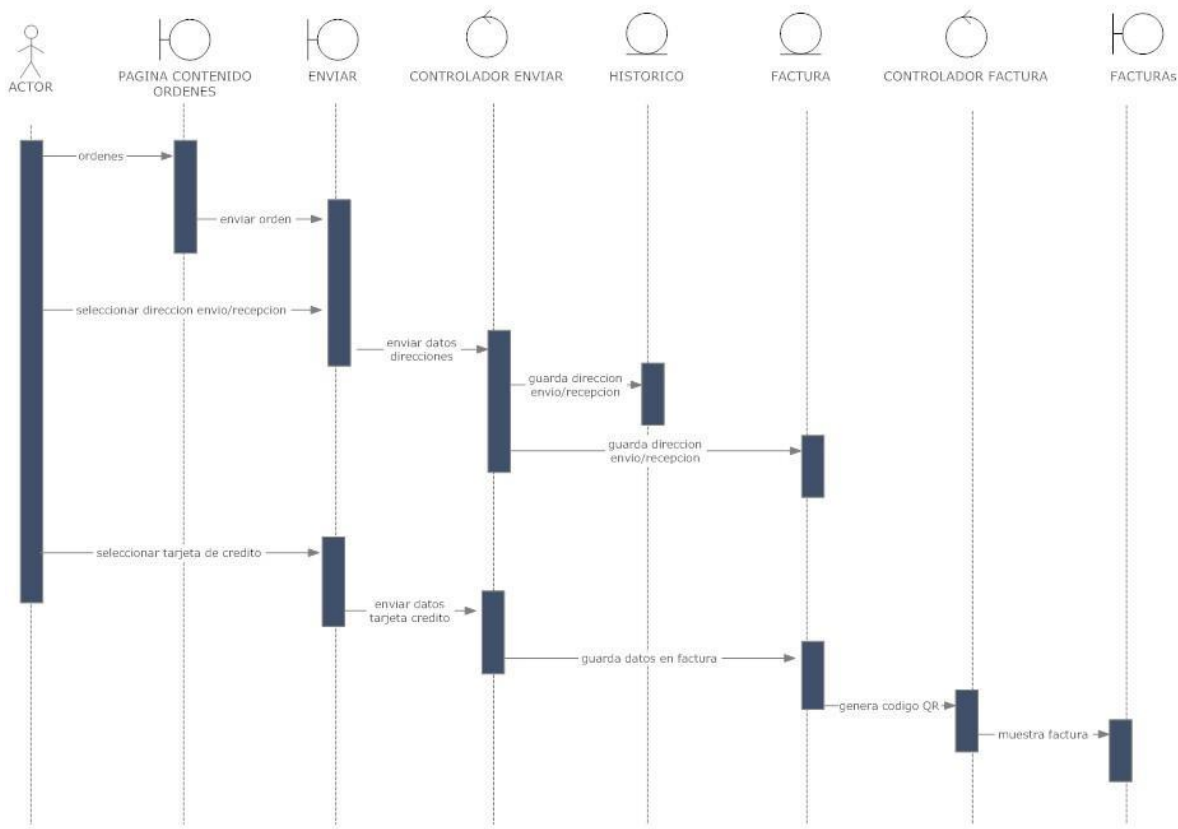
ESCENARIO CREAR ORDENES



ESCENARIO AGREGAR PAQUETE A ORDEN



ESCENARIO ENVIAR ORDEN



PATRON DE DISEÑO

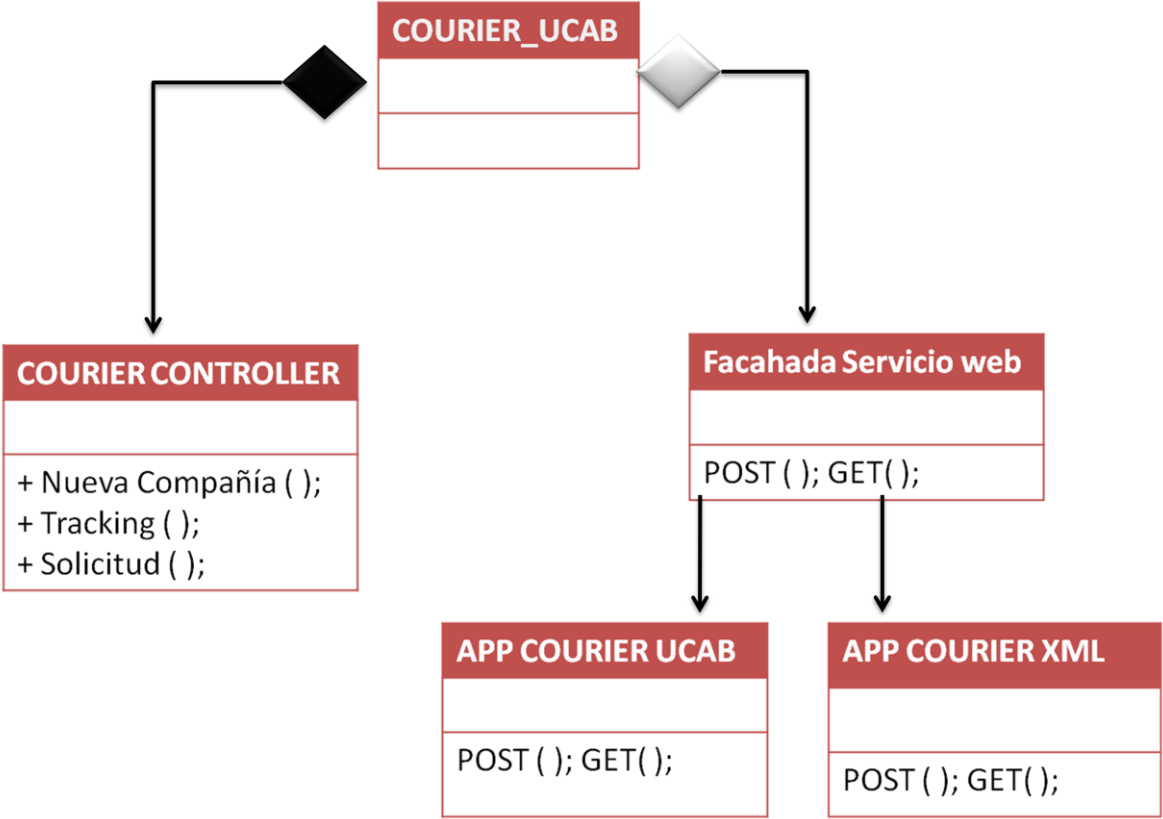
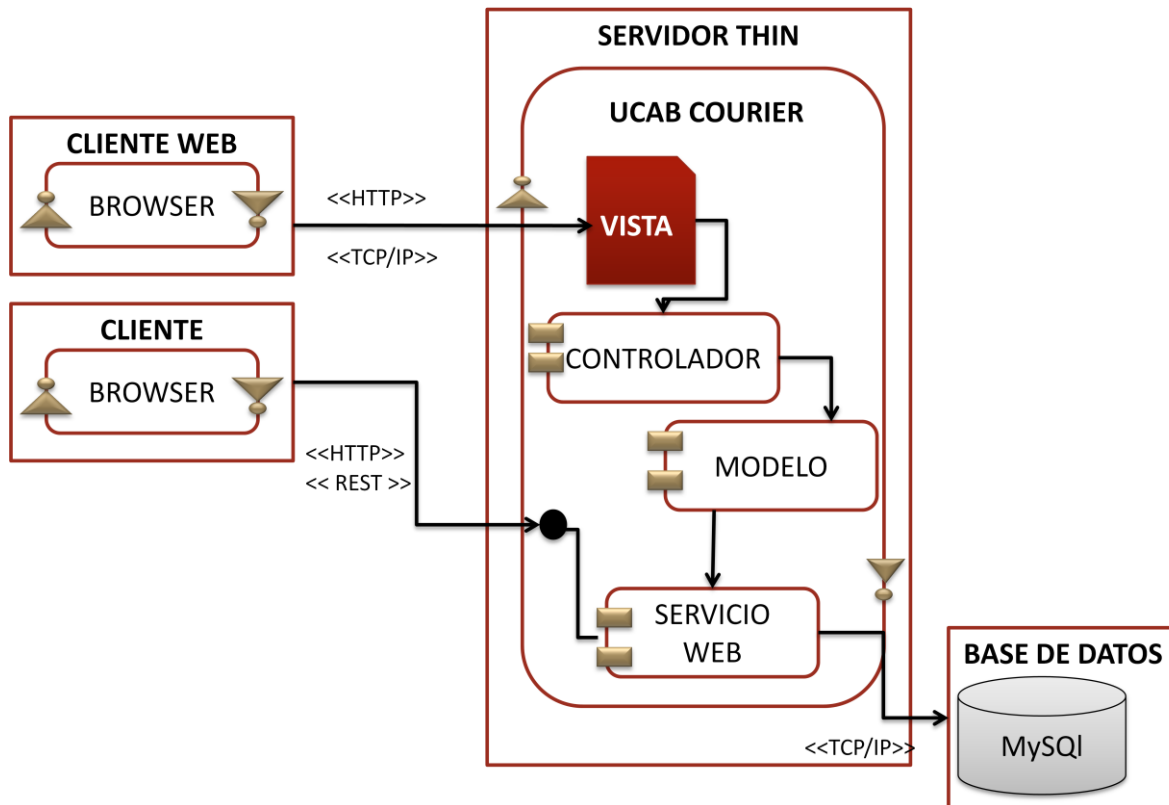


DIAGRAMA DE DESPLIEGUE



Podemos identificar dos tipos de clientes que por medio de BROWSER pueden conectarse a la aplicación directamente a la interfaz (VISTAS) del servicio web o conectarse a los métodos ofrecidos por COURIER UCAB , el protocolo usado para la conexión de VISTAS es el de HTTP TCP/IP y para el de servicio web HTTP REST y el recurso usado es el xml.

Dentro del SERVIDOR THIN encontramos nuestra aplicación y en su contenido la VISTA conectada al CONTROLADOR, éste conectado hacia el MODELO y esta ultima al SERVICIO WEB, que a su vez mediante QUERYS accede a la BASE DE DATOS.

Como se puede notar se trabajó bajo el esquema Modelo vista Controlador. En el modelo de diseño, se maneja la definición de tablas de la base de datos y sus relaciones; y se usó modelos también para poder ejecutar pruebas unitarias con la lógica de negocio.

IMPLEMENTACION

Para el desarrollo de esta aplicación se utilizo el Framework de Rails 3 (3.0.7), basado en el lenguaje de Ruby (1.9.2). Para el almacenamiento de la información se esta utilizando Mysql, como servidor utilizamos THIN server, y como manejador de versiones Git.

Instalación.

Primero se requiere instalar el Ruby y el Mysql.

Para el Mysql descargamos la ultima versión del Mysql community server 5.5 de la pagina de mysql.

Posteriormente instalamos Git, siguiendo los pasos de <http://help.github.com/mac-set-up-git/> siguiendo los pasos del sistema operativo correspondiente.

Posteriormente, bajamos RVM (Ruby versión manager), el cual es un versionador de ruby, que nos permite tener múltiples versiones de Ruby independientes en el sistema.

Para instalar RVM es necesario instalar Git antes.

Para Linux/mac, ejecutamos este comando por Terminal.

```
bash < <(curl -s https://rvm.beginrescueend.com/install/rvm)
```

Para mac ejecutamos el siguiente comando.

```
user$ echo '[[ -s "$HOME/.rvm/scripts/rvm" ]] && .  
"$HOME/.rvm/scripts/rvm" # Load RVM function' >> ~/.bash_profile
```

Para Linux el siguiente comando .

```
user$ echo '[[ -s "$HOME/.rvm/scripts/rvm" ]] && .  
"$HOME/.rvm/scripts/rvm" # Load RVM function' >> ~/.bashrc
```

Una vez instalado el RVM procedemos a instalar el Ruby en su ultima versión ejecutando el siguiente comando.

```
user$ rvm install 1.9.2
```

Una vez que tenemos instalado el ultimo Ruby lo ponemos por defecto en el sistema

```
user$ rvm use 1.9.2 --default
```

Una vez finalizado de instalar el Ruby en el sistema instalamos el Framework de Rails ejecutando el siguiente comando.

```
user$ gem install rails
```

Ahora nos bajamos el proyecto del servidor de **github**.

```
user$ git pull git@github.com:sampayo/courier.git
```

Nos metemos en la carpeta del proyecto

```
user$ cd ~/rutaalacarpeta
```

Posterior a esto instalamos Bundler, que es una gema la cual nos permite instalar todas las gemas que contenga el proyecto automáticamente.

```
user$ gem install bundler
```

Luego de instalar el Bundler podremos instalar todas las gemas del proyecto con la ejecución de este comando.

```
user$ bundler install
```

Ahora deberíamos tener todas las gemas como la del servidor “thin”, “pawl” para el pdf, y todas las que trae Rails, para verlas ejecutamos el siguiente comando.

```
user$ gem list
```

Prendemos el servidor, con el siguiente comando, debemos estar dentro de la carpeta del proyecto.

```
user$ cd ~/carpetadelproyecto
```

```
user$ rails server thin
```

Creamos la base de datos, en Mysql.

```
user$ rake db:create
```

Migramos la información de nuestra aplicación para la base de datos

```
user$ rake db:migrate
```

Le precargamos una información

```
user$ rake db:seed
```

Ahora procedemos al Browser y nos metemos en <http://localhost:3000> y podremos utilizar la aplicación.