

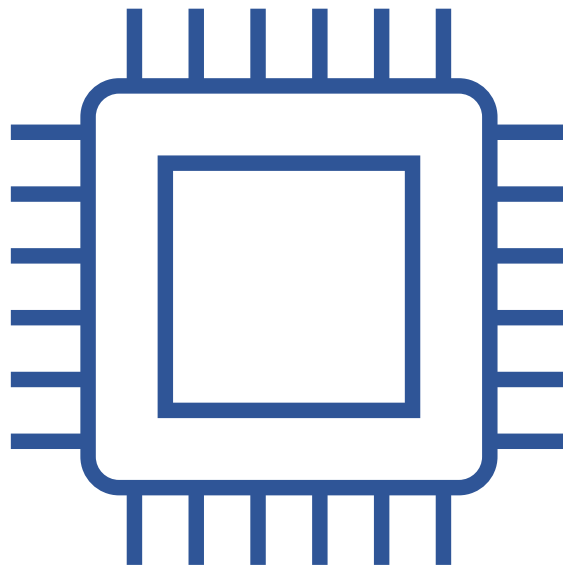
Francisco Joaquín Murcia Gómez

48734281H

Grado en ingeniería informática UA

Sistemas distribuidos

FASE 1. Fundamentos de sistemas distribuidos



Índice

Sistemas distribuidos	0
Diseño de la arquitectura conceptual.....	2
Características del SD	2
Esquema grafico.....	4
Explicación esquema	5
Diseño de la arquitectura Técnica.....	6
Esquema grafico.....	6
Explicación esquema	7

Diseño de la arquitectura conceptual

Características del SD

Escalable:

- Puede haber un numero variable de usuarios según se registren
- Puede haber un numero variable de reservas

Seguridad:

- Haremos uso de mecanismos de autenticación puesto que los datos de los usuarios han de ser confidenciales
- Se ha de encriptar todos los datos relacionados con el pago de una reserva (Tarjeta, IBAN...) para evitar posibles robos hacia nuestros clientes
- La transición de dinero ha de estar encriptada

Heterogeneidad

- Ha de ser compatible con los sistemas de reserva de los proveedores
- El sistema de la agencia debe de ser compatible con los sistemas des las diferentes entidades barias

Extensibilidad

- La empresa en un futuro puede ampliar su modelo de negocio, por eso ha de estar preparada para implementar nuevas funcionalidades

Concurrencia y sincronización

- Varios usuarios pueden registrarse, reservar o entrar al sistema a la vez, pero no pueden reservar el mismo asiento de vuelo, habitación y vehículo, en cuyo caso al segundo usuario le aparecerá un mensaje de error explicativo
- Cuando un usuario hace una reserva, al proveedor le aparecerá dicha reserva

Tolerancia a fallos

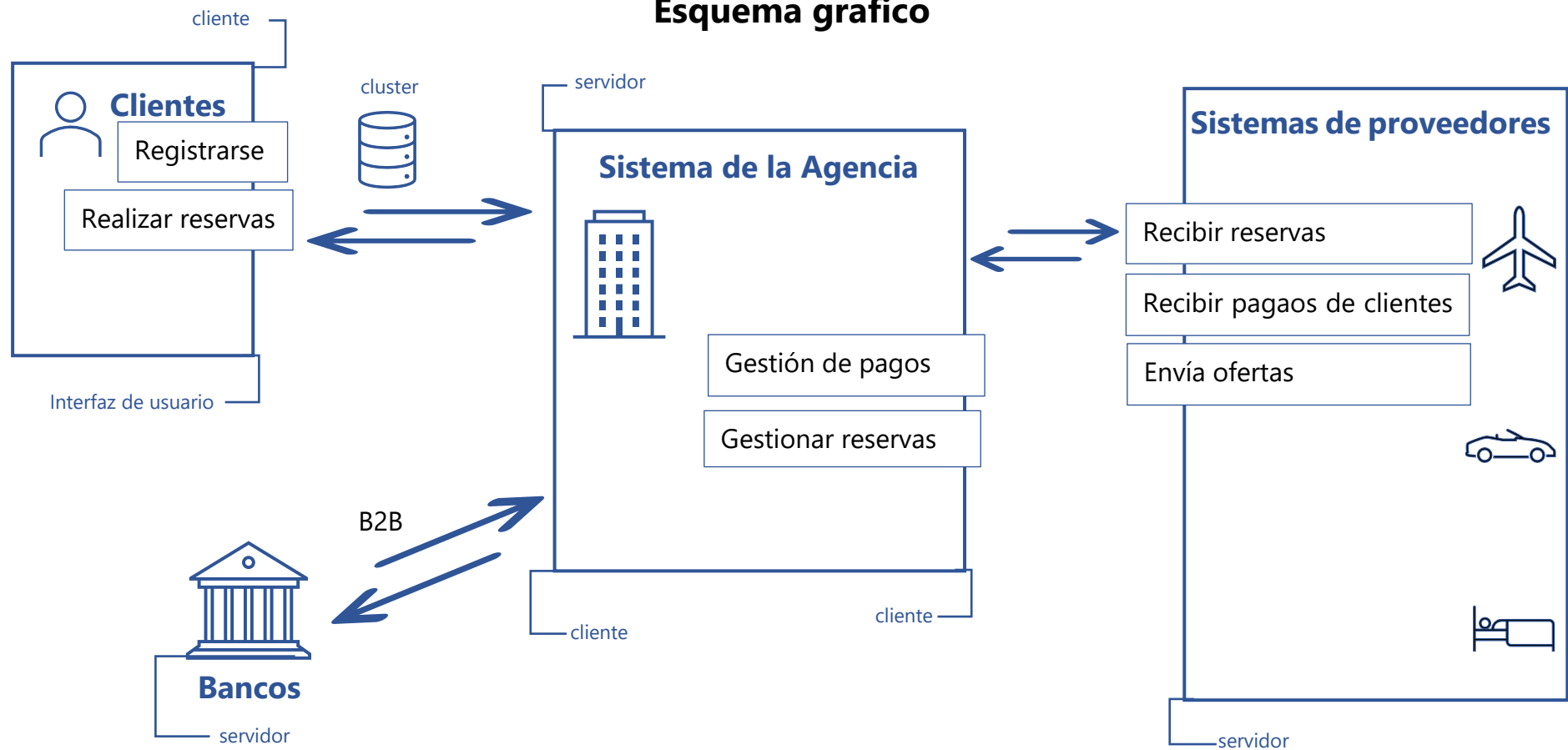
- En caso de que el usuario no se autentifique bien se le denegara el acceso hasta que se autentifique bien

- Si al realizar una reserva no cumple con los requisitos establecidos, la reserva no se realiza y aparece un mensaje de error explicándole al usuario lo ocurrido
- En caso de no haber fondos o de caducidad de la tarjeta, el pago no se realizará y por lo tanto no se realizará la reserva, se cancelará el proceso de reserva y aparecerá un mensaje explicativo de error

Transparencia

- Los clientes no saben cuándo ha ocurrido un error ajeno a ellos (HW/SW)
- Los clientes no saben cuándo un usuario se registra o reserve algo

Esquema grafico



Explicación esquema

En este sistema distribuido tenemos tres conexiones, los clientes con la agencia, la agencia con los bancos y la agencia con los proveedores

Clientes con la agencia

En clientes con agencia he decidido aplicar una arquitectura de cliente servidor, los clientes solicitan a la agencia registrarse o realizar reservas y la agencia se las resuelve.

Para la gestión de las peticiones utilizamos un cluster.

Agencia con los bancos

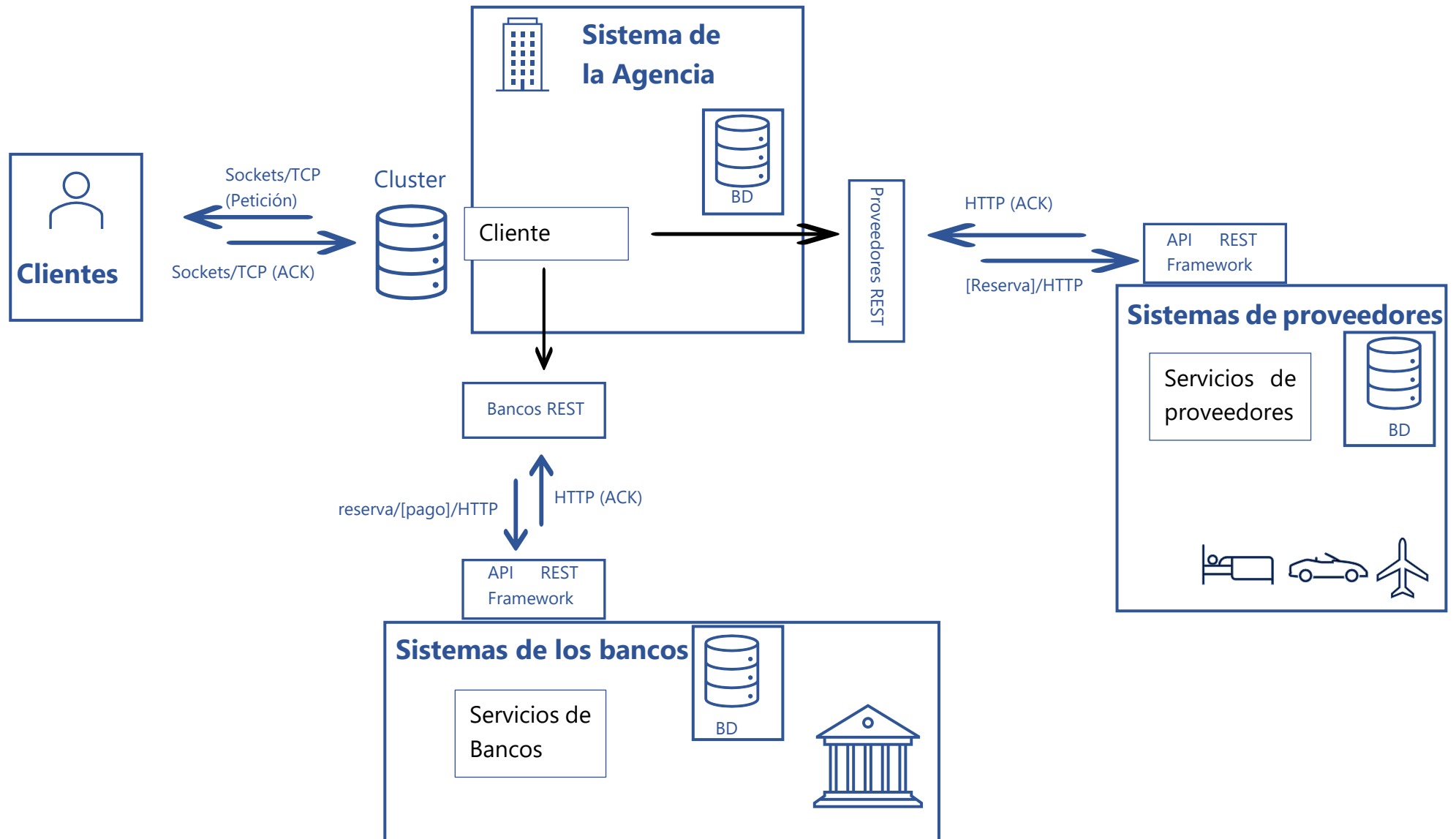
En este caso es una interfaz "Business to Business" que usaría la arquitectura cliente servidor, ya que la agencia genera peticiones para efectuar los pagos de los clientes y las entidades le contestan si se ha realizado o si se efectuado correctamente

Agencia con los proveedores

He usado una arquitectura cliente servidor ya que los proveedores envían las ofertas, la agencia envía las reservas que va gestionando.

Diseño de la arquitectura Técnica

Esquema grafico



Explicación esquema

Como ya se explico en el apartado de la arquitectura conceptual, la arquitectura es de SOA, pero en este caso el registro está implícito, usamos una arquitectura de cliente/servidor con API RESTFul.

Cientes – Agencia

En esta conexión se ha empleado sockets con el protocolo TCP, cuando un cliente manda una petición a la agencia, esta devuelve un ACK para empezar la comunicación, se comprueba en la BD si la autenticación es correcta, si tiene reservas..., esta base de datos almacena todos los datos de los clientes, así como las reservas etc. Cuando la agencia resuelve la petición se la envía al cliente.

Para evitar la saturación de peticiones por parte de los clientes, se hará uso de un cluster.

Agencia - Proveedores

En esta conexión usamos protocolo HTTP con REST, la agencia le envía a los proveedores la reserva, conectándose usando framework de API REST, el proveedor implementará la reserva en la base de datos que tiene cada proveedor, un vez guardado, devolverá un mensaje de respuesta a la agencia.

Agencia – Bancos

En esta conexión usamos protocolo HTTP con REST, la agencia le envía a las entidades el dinero de la reserva, conectándose usando framework de API REST, dicha entidad guardara el registro del pago en en la base de datos que tiene cada banco, un vez guardado, devolverá un mensaje de respuesta a la agencia.