|  |  |  |
| --- | --- | --- |
| logo_blanco_negro | **UNIVERSIDAD CATÓLICA ANDRÉS BELLO**  Urb. Montalbán – La Vega- Apartado 29086  Teléfono: +58212 - 4074407 Fax: +58212 - 4074447  Caracas, 1021 –Venezuela  **Facultad de Ingeniería**  **Escuela de Ingeniería Informática** | logo_alta_res |

Prof. Christofer Castillo

Integrantes:

Mary Carmen Carrascal CI:

Angélica Omaña CI:17.514.377

Emmanuel Perez CI:

Juan Carlos Escalante CI:

Luis Figueras CI:

Gustavo Briceño CI:

Caracas, 29 de abril de 2010

Resumen

En el presente informe se señalan los métodos y recursos necesarios para la creación de un sistema que logre el manejo de gestión de vuelos, gestión de aeronaves, gestión de clientes, reserva y compra de boletos, donde existen dos oficinas, la oficina administrativa la cual se encarga del modulo de vuelos y aviones, y la oficina comercial que maneja los clientes, compras y reservas de boletos, adicionalmente en la oficina administrativa se manejan los reportes, los cuales serán información almacenada para las consultas que sean necesarias.

Para resolver problema expuesto se decidió crear un sistema automatizado capaz de manejar la concurrencia de clientes en la agencia de viaje, y el envió de información entre ambas oficinas, información que fue manejada con archivos XML, además de esto se tiene un servidor central el cual está capacitado para actualizar la información que llega, de tal manera que la información sea la misma para todas las oficinas.

Con esta implementación se dio solución a la concurrencia entre los clientes y las oficinas, utilizando herramientas como son sockets y JavaRMI, además de prevenir las fallas que pudieran ocasionarse, se usaron servidores espejos los cuales tomarían el mando en caso que el servidor principal presentara algún problema, con esta manera se garantiza la integridad de la información y seguridad en el sistema.

Introducción

Se desea realizar un sistema para el manejo de una agencia de viaje, donde los clientes podrán reservar y comprar boletos, además se podrá gestionar los vuelos y aviones, así como también se tendrán los reportes más importantes.

Se tendrá la oficina comercial la cual recibe datos de los clientes y las reservas y compras que se realicen, esta información va a un servidor principal, el cual en caso de falla habrá otro que tomara el control, esta información llega a la oficina administrativa donde estará un mainframe que tendrá todos estos datos y el cual se va a encargar de mantener toda la información centralizada y actualizada en dicha oficina, y en caso de fallas tendrá su servidor respaldo.

Problema

Con lo explicado anteriormente se puede observar los problemas del sistema de la agencia de viaje, lo cuales son:

* Entablar la comunicación entre la oficina comercial y la oficina administrativa.
* Asegurar que la comunicación sea persistente, sin perdidas de datos.
* Tener reportes más importantes necesarios para conocer la información que ayude en la toma de decisiones del negocio, como por ejemplo “dar descuento al cliente que haya hecho más compras”

Solución

Para solucionar estos problemas mencionados usamos las herramientas como JavaRMI para la comunicación entre el cliente/servidor y la oficina administrativa, y los sockets para la comunicación entre la oficina comercial y el cliente/servidor, todo esto fue a través del lenguaje de programación java con el que se tienen los conocimientos para ejecutar el sistema. Además una arquitectura de sistema distribuido que garantice la fiabilidad y seguridad de la información y el sistema.

Los sockets permiten la iteración remota entre dos o más computadoras donde entre ellos habrán clientes y servidores, utilizando ips y puertos específicos, nosotros implementamos sockets para lograr la comunicación entre la oficina comercial y el servidor/cliente principal.

Por otra parte los JavaRMI logra la invocación de métodos remotos entre un cliente y un servidor, o podrían ser varios servidores como es en nuestro caso, entonces creamos una interfaz tanto para la oficina comercial y para la oficina administrativa, estas interfaces llaman a los métodos que se encuentran en el servidor.

Utilizamos un mainframe que tendrá toda la información centralizada por medio de xml, con esto se logran los reportes y que la información sea persistente e igual en todos los servidores. Y asi en ambas oficinas tendrán las consultas iguales. (Ver imagen 1).

Resultados Obtenidos

El sistema desarrollado alcanza los objetivos que se querían cumplir, de forma sencilla y completa, estableciendo la solución a los problemas que había desde un principio y los posibles problemas encontrados durante la realización de éste. Es importante aclarar que como fue un proyecto con un límite de tiempo y que además no se cuenta con el conocimiento de todas las herramientas posibles, es probable que no sea la implementación más optima a las necesidades.

Conclusión

Es evidente que para lograr los resultados obtenidos fue necesario la investigación y el desarrollo de todos los conocimientos aprendidos durante la carrera, además de esto asegurar que todos teníamos las herramientas básicas y que éstas sean compatibles para todos, ya que al tener problemas de compatibilidad no sería exitoso las pruebas fundamentales de lo que se está desarrollando.

Por otra parte, es importante la organización de un equipo, como en nuestro caso que somos 6 personas en este sistema, debimos tener un buen control de lo que hacíamos y repartir de manera igual el trabajo, además de reunirnos semanalmente, como mínimo una vez para unir lo que llevábamos y conversar de cómo vamos y que nos falta.

Sugerencias

Cuando se desee desarrollar un sistema como este en un futuro, o simplemente mejorar este proyecto, es necesario entender el porqué se utilizo la arquitectura de la solución planteada, al igual de las herramientas que fueron implementadas.

Es posible buscar una mejoría en la persistencia de los datos, como por ejemplo que las peticiones que han sido enviadas nunca se pierdan y se lleve una bitácora de éstas.

Debilidades

El sistema no maneja la persistencia de los datos en el siguiente caso:

Cuando se envía una petición y durante ese envío el servidor principal se cae, el servidor espejo no logra recuperar esa data, por ende ésta se pierde, sin embargo si el servidor principal cae sin ninguna petición en el camino, al enviar una petición nueva el servidor respaldo si la maneja sin inconvenientes.

Reglas del negocio

-No se eliminaran las compras, ya que si es una compra significa que fue cancelada (pagada) por ende no se puede eliminar del sistema.

-No se eliminan los clientes que están en el sistema, bien sea que están porque hicieron una reserva o una compra permanecerán en el sistema hasta que este cambie en un futuro, ya que nos parece lo mas adecuado mantener todos los clientes porque no existe la seguridad de que no volverán nuevamente a comprar.

-Se podrán eliminar los aviones y los vuelos, ya que un avión pudo haberse dañado a ultima hora y estar éste relacionado a un vuelo.





