Pontificia Universidad Javeriana

Introducción a los Sistemas Distribuidos

Mariela Josefina Curiel Huérfano

Sistema Distribuido de Bolsa de Empleo

Julián Ricardo Rizo Orjuela

Juan Sebastián Ruíz Bulla

Camilo Andrés Sandoval Guayambuco

27/09/2021

Contenido

Introducción 3
Diagrama de Clases 3
Diagrama de Secuencia 4
Protocolo de pruebas 5
Modelos del sistema 6
Bibliografía 8

Lista de tablas

Tabla 1 Resultado experimentos 5

Lista de figuras

Figura 1 Diagrama de clases [1] 3
Figura 2 Diagrama de secuencia [2] 4
Figura 3 Modelo publicador-subscriptor 6
Figura 4 Modelo Request-Reply 7

Introducción

En el presente informe se desarrollará la estructura de un sistema de empleo junto con sus respectivas características y funcionalidades principales, esto se hará mediante un diagrama de clases y un diagrama de secuencia respectivamente. Adicional a esto, se realizará un protocolo de pruebas en el que, si bien no será utilizado para esta entrega, nos establecerá las directrices al momento de desarrollar la entrega final. Finalmente, se presentará la descripción de la implementación inicial de una funcionalidad especifica la cual se verá a mayor detalle a continuación.

Diagrama de Clases

Para modelar el sistema de empleo se utilizó un despliegue de datos constituyente en el diagrama de clases, cada clase contiene las relaciones pertinentes al sistema junto con sus respectivos atributos y métodos correspondientes. Así como se muestra a continuación:

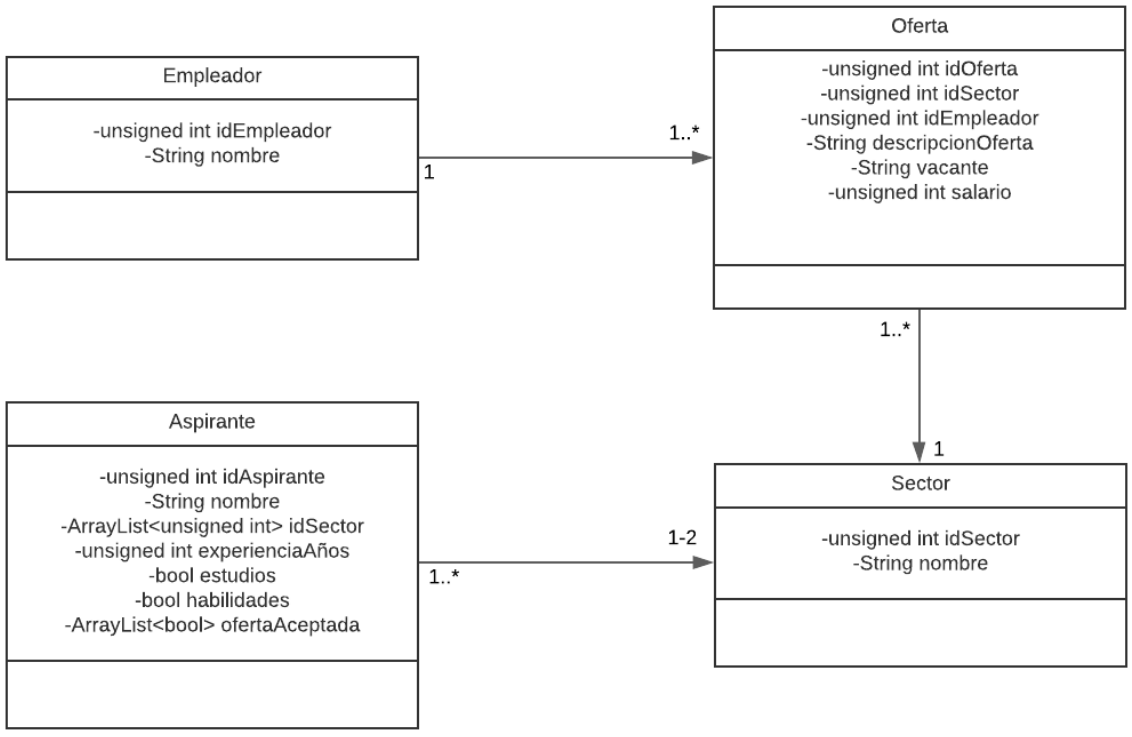

FIGURA 1 DIAGRAMA DE CLASES [1]

Diagrama de Secuencia

Se modeló un diagrama de secuencia del funcionamiento del sistema distribuido, para visualizar las interacciones entre los objetos y actores del mismo, viendo el orden de los eventos que ocurren cronológicamente. Sin embargo, cabe aclarar que eventos como lo son "*Envía oferta*" y "*Envía solicitud*" son asincrónicos

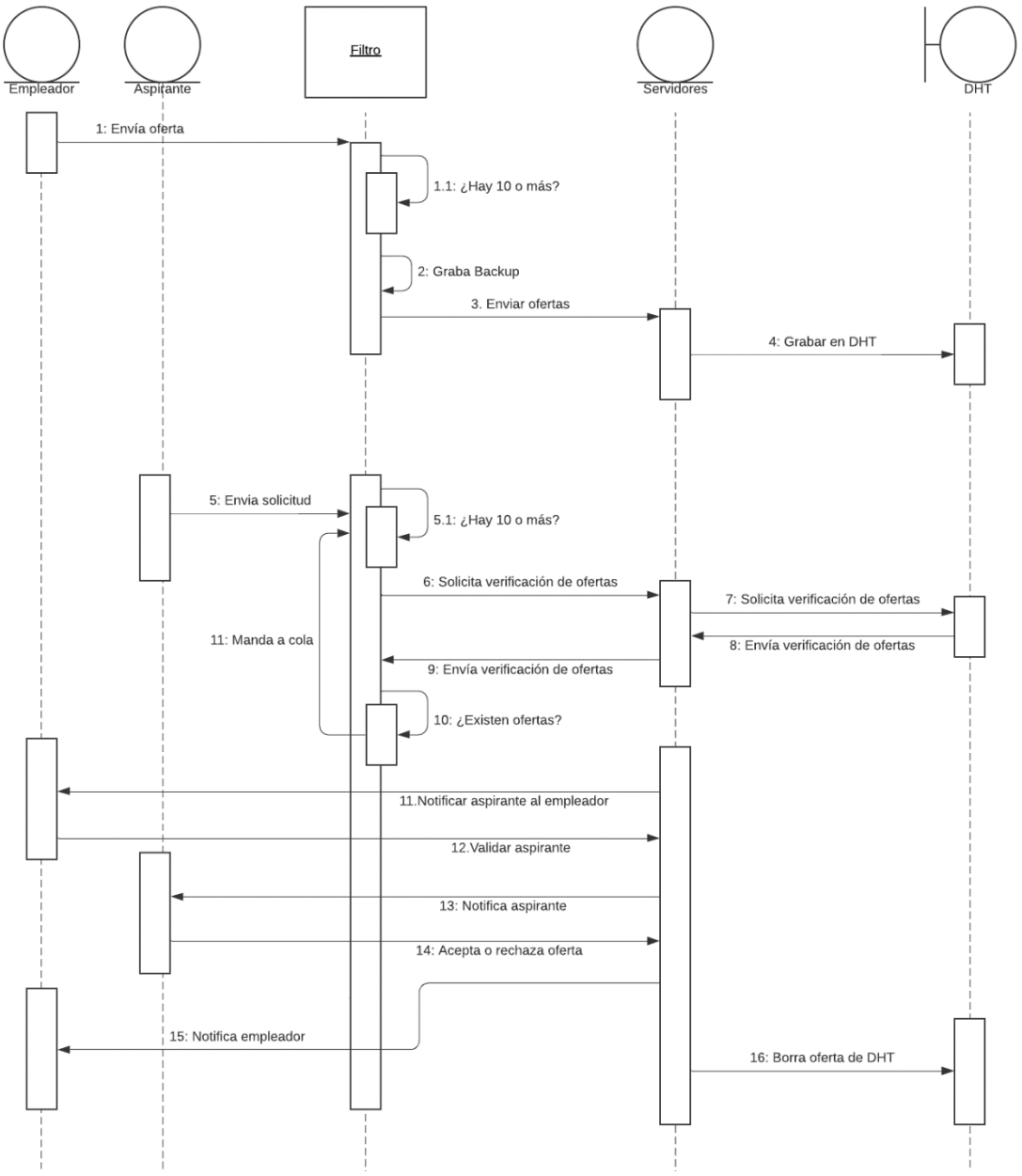

FIGURA 2 DIAGRAMA DE SECUENCIA [2]

Protocolo de pruebas

En este apartado se presentarán las diferentes pruebas que se realizarán para medir el funcionamiento de nuestro proyecto en diferentes ámbitos como lo puede ser el rendimiento en cuanto ciertas cantidades de solicitudes, detectar fallas en algún componente en particular, determinar tiempos de respuesta, etc.

Los protocolos de pruebas a realizar son los siguientes:

- **Funcionamiento satisfactorio:** En este protocolo se hará el funcionamiento normal del sistema en cuanto a la oferta y demanda en donde se asignan empleos de forma satisfactoria.
- **Varios aspirantes en un mismo cargo:** Así como lo dice el nombre de este protocolo, simularemos que dos aspirantes intenten obtener un cargo del mismo sector, esto lo haremos para los 5 sectores del trabajo escogidos.
- **Fallo en el filtro:** Este experimento consta de realizar los dos anteriores, pero con la diferencia de que en el T/2 alguno de los componentes del "Filtro" debe fallar, a pesar de ello se debe hacer un proceso idéntico, además, el sistema no debe parar su funcionamiento.

Para los experimentos previamente descritos se debe llenar la siguiente tabla:

Experimento I			**Experimento II**	
Número de solicitudes de empleo	T. de publicación de oferta laboral promedio	T. Respuesta promedio (Asignación de empleo)	T. Respuesta promedio (Filtro de datos por sector)	T. Promedio de búsqueda (consulta en la DHT)
5				
10				

TABLA 1 RESULTADO EXPERIMENTOS

- **Aspirante rechazado:** Dado el caso de que un aspirante no cumpla con los requisitos de una oferta determinada, el sistema lo pondrá en lista de espera de otra vacante.
- **Doble servidor:** Para esta prueba se hará la implementación en tres pcs, donde uno de ellos hará de empleador y aspirante, y los demás harán de filtro y servidor para verificar el funcionamiento del patrón Request-Reply.
- **Cantidad máxima de solicitudes:** Esta prueba de rendimiento, se hace con el propósito de determinar la cantidad máxima de solicitudes hasta que el servidor falle.

Modelos del sistema

Los modelos que se usaron para el desarrollo del proyecto fueron los siguientes:

- **Modelo publicador-subscriptor:** Este modelo es un patrón de distribución one-way-data y patrón de datos one-to-many, en el cual establece una conexión asíncrona entre un publicador y un conjunto de suscriptores, en este modelo el publicador transmite mensajes sin fin, sin importar existe algún suscriptor, simplemente deja los mensajes, caso contrario de lo que hace un suscriptor y es que si el suscriptor, al conectarse no hay un publicador el suscriptor debe esperar a que haya uno, esto se maneja a través de sockets para enviar o recibir datos. Adicional a esto, este patrón utiliza unos tópicos para recibir datos en específico.

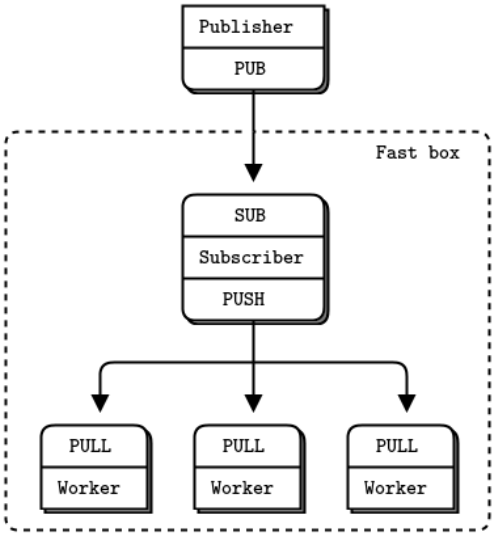

FIGURA 3 MODELO PUBLICADOR-SUBSCRIPTOR

- **Modelo Request - Reply:** Este modelo es una combinación de RPC con el modelo cliente – servidor. El cliente realiza una solicitud a un servidor, este procesa la solicitud y devuelve el resultado al cliente. REP, es el socket que usa el servidor. El cual solamente permite llamadas alternativas de recieve seguida de send; Por otro lado, el REQ es el que usa el cliente para hacer las solicitudes al servidor y recibir las correspondientes respuestas. Este socket solamente permite llamadas alternadas de send seguida de receive.

    - Fair – Queing:

Este tipo de algoritmo garantiza que el proceso recolector que recibe los mensajes de varios procesos, no se quede recibiendo simplemente de uno, haciendo la llegada de los mismos de manera equitativa, también formando una cola de mensajes en los procesos de salida.

- Round – Robin:

Este algoritmo consiste básicamente en que un planificador reparte el tiempo de CPU entre n procesos que se forman ordenadamente en una cola circular. El planificador le va asignando el tiempo a los procesos y estos van rotando cuando su tiempo de CPU se agota, en este momento deben esperar a que les vuelva a tocar el turno para que el planificador les vuelva asignar más tiempo de CPU. [3]

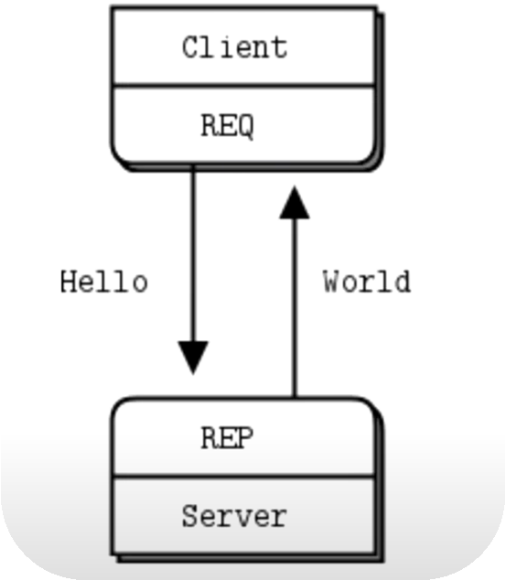

FIGURA 4 MODELO REQUEST-REPLY

Bibliografía

[1] S. R. C. S. Julian Rizo, «LucidChart,» 25 Septiembre 2021. [En línea]. Available: https://lucid.app/lucidchart/invitations/accept/inv_1a981def-9856-4cac-9060-035d3e57fd51?viewport_loc=192%2C38%2C1852%2C945%2C0_0.

[2] S. R. C. S. Julian Rizo, «Lucidchart,» 25 Septiembre 2021. [En línea]. Available: https://lucid.app/lucidchart/0b5a55fa-eed0-44ac-9b37-7c27f4d991e5/edit?shared=true&page=HWEp-vi-RSFO#.

[3] ZGuide, «Chapter 5 - Advanced Pub-Sub Patterns,» [En línea]. Available: https://zguide.zeromq.org/docs/chapter5/#High-Speed-Subscribers-Black-Box-Pattern.