

Pontificia Universidad Javeriana



Asignatura:

Introducción a los Sistemas Distribuidos

Trabajo:

Proyecto 2 – Asignación
citas entrevistas de trabajo

Docente:

Enrique González

Bogotá D.C.

Noviembre, 20 de 2019

- **Introducción:**

El documento tiene como fin presentar la documentación pertinente al proyecto *2 Asignación citas entrevistas de trabajo*, mostrando los diagramas UML, con cada una de las clases implementadas; una descripción breve de los componentes principales del proyecto y por último un escenario de pruebas.

Cabe resaltar, que el equipo realizó una función adicional (bonus), la cual consiste en tener 2 servidores distribuidos que realizan de manera conjunta la tarea de procesar y registrar la información por parte de los candidatos y las ofertas de la empresa. Lo anterior se explicará con más detalle en la sección de escenario de pruebas.

El proyecto se realizó con Java RMI (Remote Method Invocation) que es un mecanismo que dispone el lenguaje de programación Java para hacer el llamado a objetos de manera remota, lo que permite tener un mecanismo simple de comunicación de servidores en ambientes distribuidos. También es importante mencionar que la topología de red que se utilizará en la sustentación se basa en 3 host, cada uno describe una entidad del proyecto (Candidato, Empresa o Servidores) y todos ellos están conectados a la misma red LAN, con el fin de facilitar la comunicación entre los procesos y que esta sea de manera ágil y flexible.

- **Esquema de procesos e interacción:**

El esquema general del proyecto se presenta en la figura 1, que es tomado de enunciado del proyecto y que evidencia cómo se realiza la comunicación y la interacción entre los procesos los cuales en este caso son los candidatos, empresas y servidor. Para el proyecto que presenta el equipo de trabajo, no solamente hay un servidor el cual recibe todas las peticiones de los procesos candidato y empresa, sino que puede haber n servidores que se comunican entre ellos y realizan toda la lógica relacionada con el procesamiento de la información de manera conjunto. En la sección de escenario de pruebas se explica con más detalle esquema de interacción entre cada uno de los componentes del proyecto, describiendo paso a paso cómo se realiza.

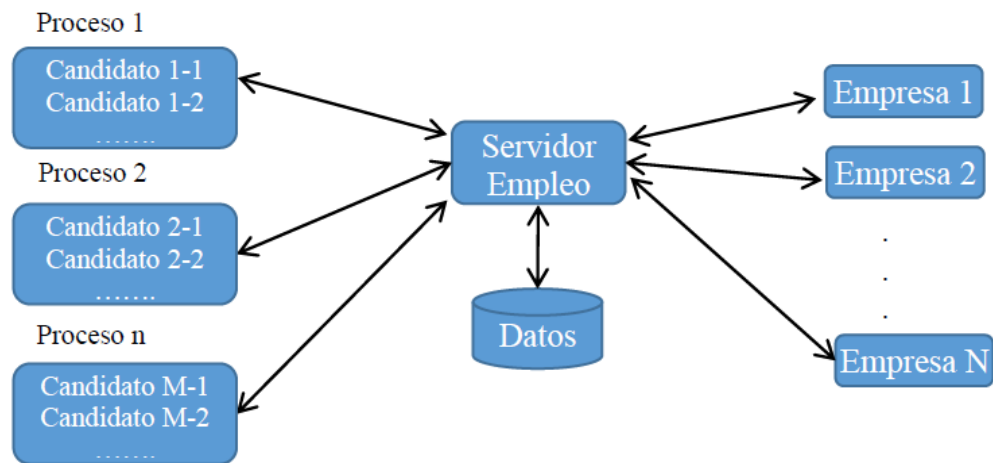


Figura 1. Esquema de procesos

- **Diagrama de clases:**

A grandes rasgos, el proyecto se compone de 3 subproyectos que se encuentran estructurados bajo la arquitectura de n-capas, en este caso, se tienen las capas (paquetes) de negocio, persistencia y presentación, en la cuales cada una tiene sus responsabilidades y se comunica con los otros paquetes y subproyectos mediante interfaces. El primer subproyecto es el de **Candidato** que es muy similar al subproyecto de **Empresa** en lo que respecta a la capa de negocio y se diferencian no más en la presentación y persistencia, ya que cada uno tiene su propia base de datos, donde para los candidatos, se tiene un archivo con toda la información de los candidatos y para el de empresas, se tiene toda la información relacionada con las ofertas de trabajo. El tercer subproyecto es el de **Servidor** que contiene las interfaces necesarias para conectarse con los otros componentes y la lógica para la asignación de entrevistas y procesamiento de la información.

En los diagramas de a continuación se presentación los diagramas de clase de los 3 proyectos que se describieron anteriormente, el diagrama 1 corresponde al proyecto **Candidatos**, el diagrama 2 corresponde al proyecto **Empresa** y por último el diagrama 3 corresponde con el proyecto de **Servidor**.

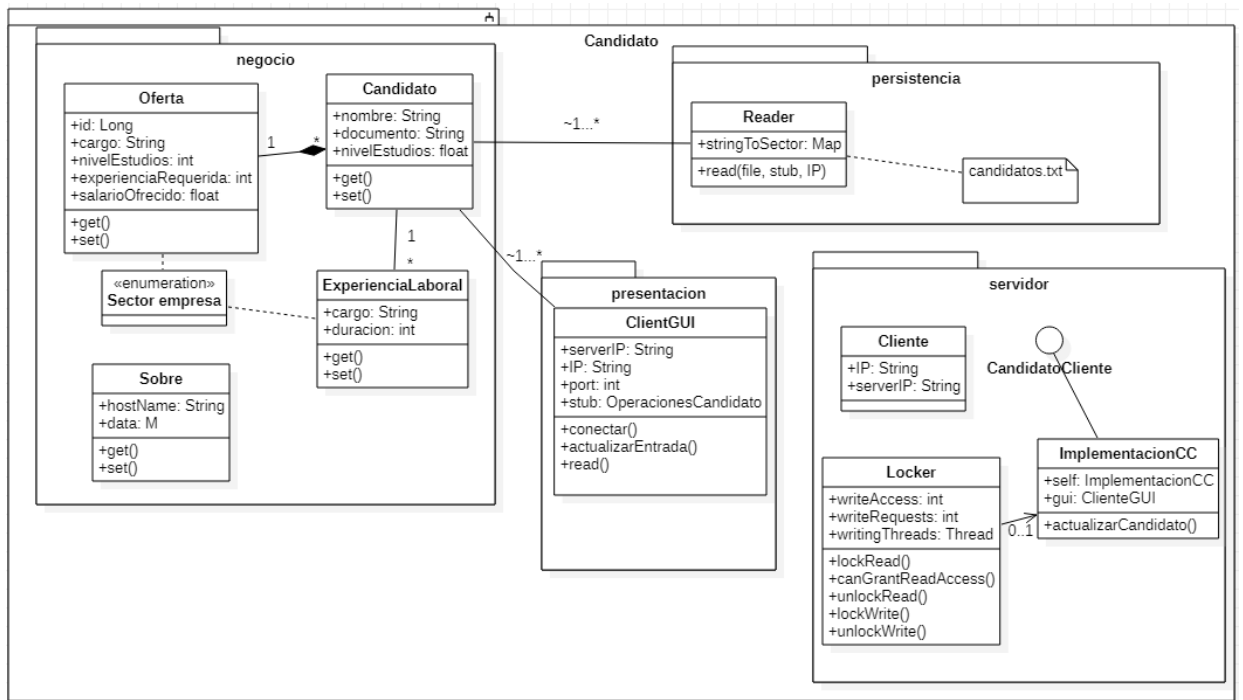


Diagrama 1. Proyecto Candidato

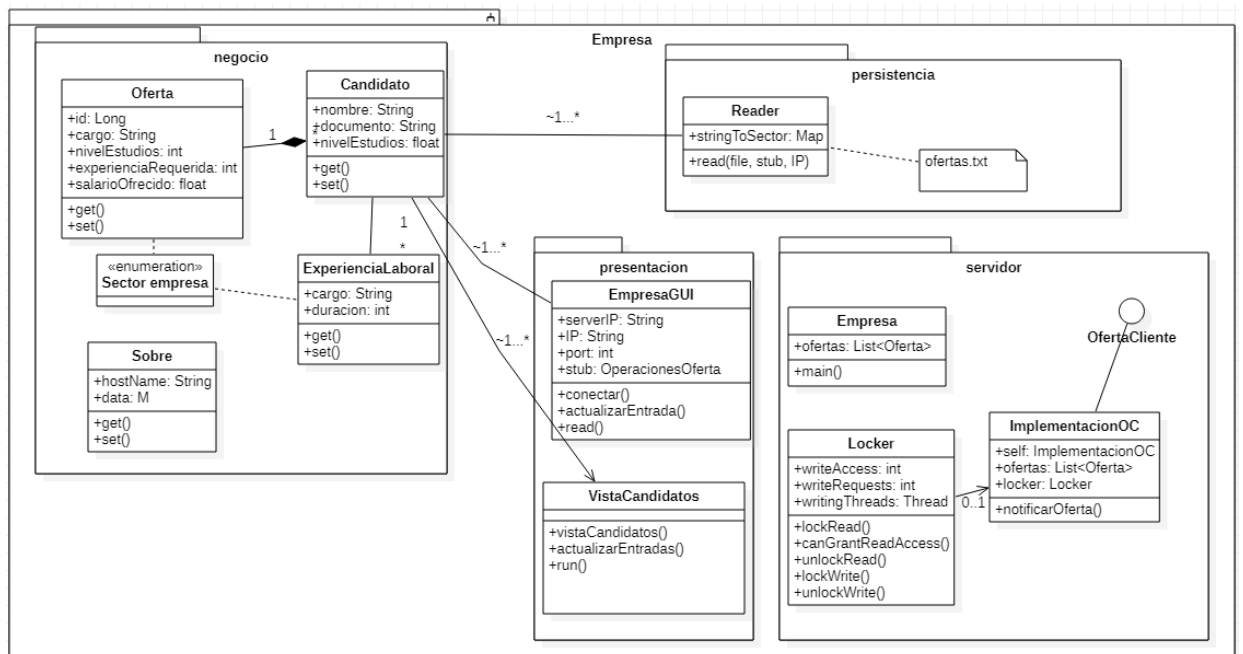


Diagrama 2. Proyecto Empresa

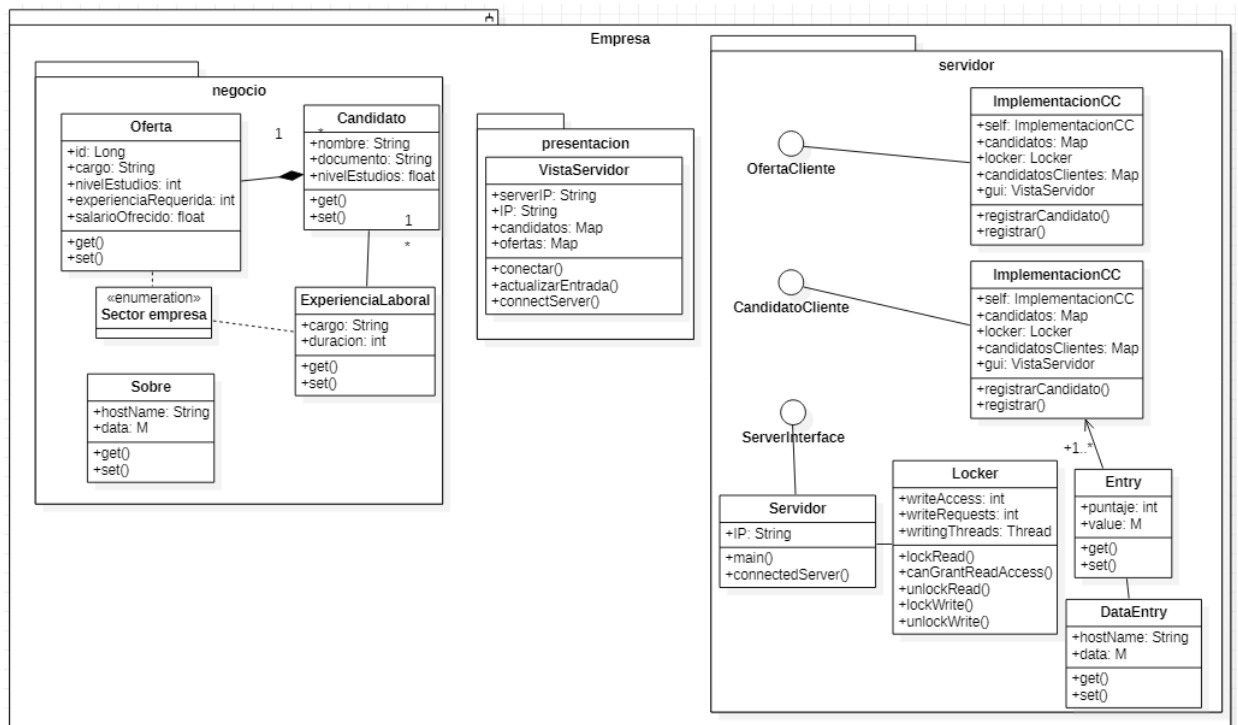


Diagrama 3. Proyecto Servidor

• Descripción general de los componentes:

Los componentes más importantes del proyecto se encuentran en cada uno de los 3 subproyectos (Candidato, Empresa y Servidor), dentro del paquete de negocio donde está toda la lógica y la forma en que se gestionan las conexiones y se realizan las mismas, entre las clases más importantes están:

📖 **ServerInterface:** solicita que se conecta con otro servidor, en caso de que se desee trabajar de manera distribuida con más de 1 servidor.

📖 **OperacionesCandidato:** permite a los clientes postear los candidatos, publicando toda la información personal de ellos.

📖 **OperacionesOferta:** permite a las empresas postear o publicar las ofertas de trabajo que estas tienen.

📖 **CandidatoCliente:** por medio de esta interfaz, se le notifica a un candidato que ya había posteado antes, que llegó una oferta nueva.

📖 **OfertaCliente:** por medio de esta interfaz, se le notifica a una empresa que ya había posteado antes, que llegó un candidato nuevo.

Finalmente, es importante mencionar que el control de concurrencia, básicamente se realiza bloqueando los recursos que se están utilizando, lo que es similar a los semáforos en sistemas operativos.

- **Escenario de prueba:**

Se realizaron 2 escenarios de prueba, en el primero, solo con un servidor, encargado de realizar encargado de recibir y procesas todas las solicitudes tanto de los candidatos como de las empresas y el otro escenario con dos servidores para que trabajaran y se comunicaran para recibir las solicitudes, por un lado, de los candidatos y por otro lado de las empresas.

En el primer caso, que se trabajó con un solo servidor, que lo que hace el servidor es recibir un nuevo, lo anota, lo introduce en la lista de candidatos y luego lo compara con todas las ofertas que tiene y si consigue un match (que el candidato coincida con los requisitos de la oferta de trabajo de la empresa) entonces le avisa a los candidatos que encontró un match para esa oferta en específico y luego el cliente actualiza sus datos, su interfaz gráfica. Del mismo modo, del lado de las empresas, cuando se publica una oferta, el servidor le busca match y si hace match, entonces se le avisa al candidato y este actualiza su información. Al momento de actualizar la información en los dos casos anteriores, el RMI se encarga de responder con el candidato ya actualizado, lo que facilita el proceso de comunicación con el servidor. En la figura 2, se muestra la interfaz gráfica de los 3 proyecto corriendo de manera conjunta, por un lado, el de candidatos, que muestra toda la información pertinente a cada uno de ellos; el de las empresas que muestra la información de las ofertas de trabajo y el del servidor, que tiene tanto las ofertas como los candidatos y va buscando la información para hacer match.

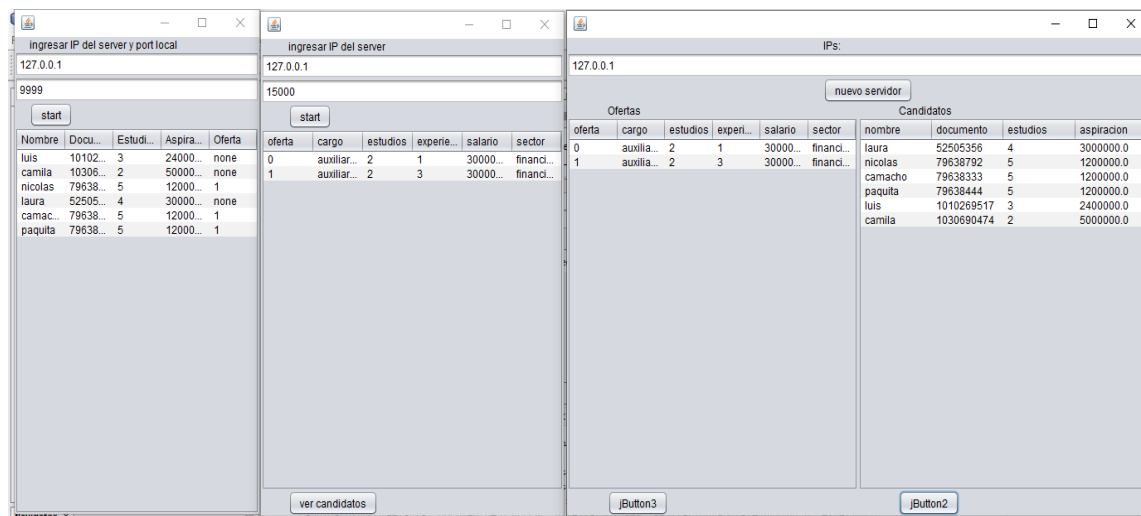


Figura 2. Interfaces visuales de los 3 componentes del proyecto

En la figura 3, se muestra lo que se comentó en el párrafo anterior de que cuando un candidato era indicado para la oferta de trabajo que se encontraba publicada, entonces se le notificaba y se actualizaba la información de este.

```
Candidato{nombre=luis, documento=1010269517, nivelEstudios=3, aspiracionLaboral=2400000.0, experiencia=[negocio.ExperienciaLaboral@2b  
Candidato{nombre=camila, documento=1030690474, nivelEstudios=2, aspiracionLaboral=5000000.0, experiencia=[negocio.ExperienciaLaboral@  
notificando:Candidato{nombre=nicolas, documento=79638792, nivelEstudios=5, aspiracionLaboral=1200000.0, experiencia=[negocio.Experier  
notificando:Candidato{nombre=nicolas, documento=79638792, nivelEstudios=5, aspiracionLaboral=1200000.0, experiencia=[negocio.Experier  
Candidato{nombre=nicolas, documento=79638792, nivelEstudios=5, aspiracionLaboral=1200000.0, experiencia=[negocio.ExperienciaLaboral@3  
Candidato{nombre=laura, documento=52505356, nivelEstudios=4, aspiracionLaboral=3000000.0, experiencia=[negocio.ExperienciaLaboral@4b8  
notificando:Candidato{nombre=camacho, documento=79638333, nivelEstudios=5, aspiracionLaboral=1200000.0, experiencia=[negocio.Experier  
notificando:Candidato{nombre=camacho, documento=79638333, nivelEstudios=5, aspiracionLaboral=1200000.0, experiencia=[negocio.Experier  
Candidato{nombre=camacho, documento=79638333, nivelEstudios=5, aspiracionLaboral=1200000.0, experiencia=[negocio.ExperienciaLaboral@1  
notificando:Candidato{nombre=paquita, documento=79638444, nivelEstudios=5, aspiracionLaboral=1200000.0, experiencia=[negocio.Experier  
notificando:Candidato{nombre=paquita, documento=79638444, nivelEstudios=5, aspiracionLaboral=1200000.0, experiencia=[negocio.Experier  
Candidato{nombre=paquita, documento=79638444, nivelEstudios=5, aspiracionLaboral=1200000.0, experiencia=[negocio.ExperienciaLaboral@7
```

Figura 3. Notificación cuando un candidato hace match

Como se comentó en la introducción, el equipo realizó un bonus, el cual consiste en trabajar con 2 o más servidores de manera distribuida para procesar y registrar toda la información que recibe por parte de los candidatos y de las empresas, para que posteriormente pueda realizar la asignación de citas para la entrevista. Con base en lo comentado anteriormente, el escenario de pruebas se realizó con 2 servidores para fines prácticos y mostrar la funcionalidad de manera completa, donde cada uno de ellos es dueño de una lista de candidatos de los que está encargado y una lista de ofertas de las que está encargado también. Cuando al servidor 1 uno le llega una solicitud de un candidato, este trata de llenar con toda la información que tiene en su lista y se puede seguir llenando ya que el candidato no tiene límite de ofertas que puede tener; y cuando este lo llena, la pasa al servidor 2 y lo que hace este es llenarlo con los candidatos que tiene, lo llena con sus ofertas y le avisa a todos sus clientes la actualización que acabó de suceder y luego le responde al servidor 1 con ese candidato y este la devuelve a la empresa que realizó la oferta. Es importante mencionar que el servidor 2 no guarda en su base de datos la información que el servidor 1 le está pasando, es decir, que cuando el servidor 1 le pasa un cliente, el servidor 2 no la añade, solo añade la información si pasa directamente desde un cliente.

Al igual que en la figura 1, en la figura 4 se muestra las interfaces que tiene el proyecto y en la consola se observa que se añadió un servidor, en este caso está en la misma máquina, y hace las funciones que se comentaron anteriormente.

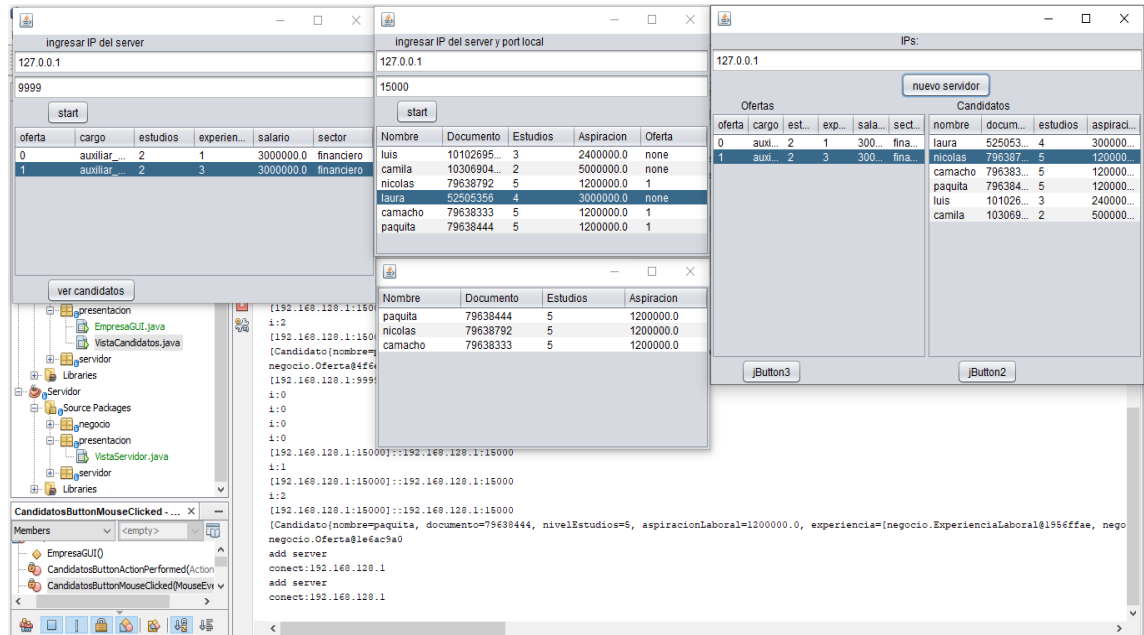


Figura 4. Proyecto corriendo con 2 servidores