



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

BÚSQUEDA AUTOMATIZADA EN
BOLSAS DE TRABAJO ONLINE CON
TÉCNICAS DE WEB SCRAPING.

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

LEONEL CRUZ NORBERTO

DIRECTOR:

ING. JOSÉ ANTONIO AYALA BARBOSA



Ciudad Universitaria, Cd. Mx., 2022

FACULTAD DE INGENIERÍA,
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

LEONEL CRUZ NORBERTO

Ingeniería en Computación
Lunes, 14 de noviembre de 2022

Agradecimientos

A mi madre Francisca, a mis hermanas Vanessa, Yisel y Adriana y a mi hermano Enrique, que estuvieron a mi lado en los días y en las noches durante mis horas de estudio.

A la máxima casa de estudios, la Universidad Nacional Autónoma de México, la cual me brindó todos los recursos para poder estudiar plenamente en la Facultad de Ingeniería y así desarrollarme académica, profesional y personalmente.

A mis incontables docentes, que compartieron su conocimiento conmigo a lo largo de mi formación académica.

A mi tutor José Antonio Ayala Barbosa por su paciencia, tolerancia, apoyo, motivación y tiempo que me dedicó a lo largo de la elaboración de este trabajo.

Finalmente a mis amigos de la vida y de la Facultad de Ingeniería, por darme fuerzas.

Índice general

1. Introducción	1
2. Marco Teórico	5
2.1. Bolsa de trabajo <i>online</i>	5
2.1.1. Historia	6
2.1.2. Características	6
2.1.3. Información concurrente	7
2.2. <i>Web scraping</i>	7
2.2.1. Componentes	8
2.2.2. Beneficios	9
2.2.3. Cuestiones legales	9
2.2.4. Aplicaciones y Trabajos Relacionados	10
2.3. Programación orientada a objetos (POO)	10
2.3.1. Lenguajes	10
2.3.2. Principios	11
2.3.3. Hilos	12
2.3.4. Beneficios	13
2.4. Patrones de diseño	14
2.4.1. Historia	14
2.4.2. Clasificación	15
2.4.2.1. CRUD	15
2.4.2.2. <i>DTO</i>	15
2.4.2.3. Modelo Vista Controlador (MVC)	16
2.5. Lenguaje de Modelado Unificado (<i>UML</i>)	18
2.5.1. Tipos de diagramas en <i>UML</i>	19
2.6. Tecnologías <i>Open Source</i>	22
2.6.1. Ventajas	22

2.6.2.	Historia	23
2.6.3.	Características	23
2.6.4.	Ejemplos	24
2.6.4.1.	Java SE	24
2.6.4.2.	Java Swing	24
2.6.4.3.	Apache Maven	25
2.6.4.4.	Selenium <i>WebDriver</i>	27
2.6.4.5.	JDBC driver	27
2.6.4.6.	Oracle <i>SQL</i>	28
2.6.4.7.	GitHub	29
2.6.4.8.	Hoja de cálculo	29
3.	Metodologías de trabajo	31
3.1.	Scrum	32
3.1.1.	Características	32
3.1.2.	Roles	32
3.1.3.	Hitos	34
3.1.4.	Herramientas	35
3.1.5.	Ventajas	35
3.2.	ISO/IEC 9126-1	36
3.2.1.	Objetivo	36
3.2.2.	Características	37
3.2.2.1.	Funcionalidad	37
3.2.2.2.	Características	37
3.2.3.	Ventajas	38
3.3.	ISO/IEC 29110	38
3.3.1.	Objetivo	39
3.3.2.	Características	39
3.3.3.	Ventajas	39
4.	Arquitectura de la aplicación	43
4.1.	Diagrama <i>UML</i> de la Arquitectura Conceptual	43
4.2.	Diagrama Entidad Relación de la Base de Datos	49
4.3.	Arquitectura Lógica MVC	52
4.3.1.	Controlador	52
4.3.2.	Vista	62
4.3.3.	Modelo	67

5. Ejecución de la aplicación	85
5.1. Ambiente de ejecución	85
5.1.1. Requisitos del sistema	85
5.1.1.1. Configuración de Base de Datos	90
5.1.2. Empaquetado de código y ejecución de la aplicación	95
5.2. Inicio de sesión	98
5.2.1. Acceso a la aplicación y gestión de credenciales	98
5.3. Búsqueda de perfiles	102
5.3.1. Parámetros de búsqueda	102
5.4. Generación de tablas	108
5.4.1. Resumen Jobomas	109
5.4.2. Resumen Superprof	110
5.5. Generación de reportes	111
5.5.1. Reporte Jobomas	112
5.5.2. Reporte Superprof	112
5.6. Terminar	113
5.6.1. Fin de ejecución	113
6. Pruebas y resultados	115
6.1. Revisión de registros en Base de Datos	115
6.1.1. Resultados	115
6.1.2. Tablas de base de datos de Jobomas	116
6.1.3. Tablas de base de datos de Superprof	121
6.2. Revisión de reportes	123
6.2.1. Hoja de cálculo Jobomas	123
6.2.2. Hoja de cálculo Superprof	128
7. Conclusiones y trabajo futuro	129

Índice de figuras

2.1.	Diagrama lógico de un <i>Scraper</i>	8
2.2.	Relación y dependencia entre hilos y procesos.	13
2.3.	Ejemplo de clase <i>DTO</i>	16
2.4.	Diagrama de secuencia de MVC.	17
2.5.	Interrelación entre los elementos del patrón MVC.	17
2.6.	Diagrama de clases.	20
2.7.	Diagrama de estado.	20
2.8.	Diagrama de actividad.	21
2.9.	Diagrama de casos de uso.	22
2.10.	Dependencias Maven en archivo pom.xml empleadas en el presente trabajo.	26
2.11.	Estructura de un proyecto Maven.	27
3.1.	Cerdos y gallinas en <i>Scrum</i> [1]	33
3.2.	Módulos del programa.	41
4.1.	Diagrama <i>UML</i> de la Arquitectura Conceptual completa.	44
4.2.	<i>UML</i> con los Hilos de ejecución.	45
4.3.	<i>UML</i> de los módulos de Superprof.	46
4.4.	<i>UML</i> de los módulos de Jobomas.	47
4.5.	Diagrama de flujo de la arquitectura completa.	48
4.6.	Modelo Entidad-Relación de tablas Jobomas en Base de Datos.	50
4.7.	Modelo Entidad-Relación de tablas Superprof en Base de Datos.	51
4.8.	Modelo Entidad-Relación de tabla Credenciales en Base de Datos.	51
4.9.	Distribución de los controladores dentro del proyecto java.	53
4.10.	Controlador que ejecuta la búsqueda de perfiles en diferentes bolsas de empleo virtuales.	54
4.11.	Controlador que ejecuta el hilo designado a Jobomas.	55

4.12. Controlador que ejecuta el hilo designado a Superprof.	55
4.13. Controlador que genera las hojas de cálculo con los registros obtenidos en el proceso de obtención de datos empleando la técnica de <i>web scraping</i>	56
4.14. Controlador que ejecuta la técnica de <i>web scraping</i> en Jobomas, se muestran las variables.	57
4.15. Controlador que ejecuta la técnica de <i>web scraping</i> en Jobomas, se muestran los métodos.	58
4.16. Controlador que ejecuta la creación de la tabla en pantalla con el resumen de resultados de Jobomas.	59
4.17. Controlador que valida las credenciales de acceso a la aplicación. . .	60
4.18. Controlador que ejecuta la técnica de <i>web scraping</i> en Superprof. . .	61
4.19. Controlador que ejecuta la creación de la tabla en pantalla con el resumen de resultados de Superprof.	62
4.20. Distribución de las vistas dentro del proyecto java.	63
4.21. Ventana que muestra la opción de generar los archivos de hojas de cálculo Jobomas y Superprof con la información obtenida en el proceso de obtención de datos empleando la técnica de <i>web scraping</i> y la opción de terminar la aplicación.	64
4.22. Ventana que solicita las credenciales de acceso a la aplicación.	65
4.23. Métodos de la vista que sobrescribe los la clase JPanel para insertar imágenes en las ventanas.	65
4.24. Ventana que solicita la los perfiles que se desean buscar.	66
4.25. Ventana que muestra un resumen de la información obtenida en el proceso de obtención de datos empleando la técnica de <i>web scraping</i> en las páginas de Jobomas y Superprof.	66
4.26. Distribución de los modelos dentro del proyecto java.	69
4.27. Modelo con los parámetros para realizar una conexión a la base de datos.	70
4.28. Modelo con las operaciones de lectura y escritura a la base de datos de Jobomas.	71
4.29. Modelo con las operaciones de lectura y escritura a la base de datos de Superprof.	72
4.30. Modelo con los parámetros que crean la hoja de cálculo con el resumen de los registro obtenidos del sitio <i>web</i> Superprof empleando la técnica de <i>web scraping</i>	73

4.31. Modelo con los parámetros que crean la hoja de cálculo con el resumen de los registros obtenidos del sitio <i>web</i> Jobomas empleando la técnica de <i>web scraping</i>	74
4.32. Modelo que contiene los parámetros para crear una instancia del <i>web-Driver Chrome</i> , este nos permitirá emplear el navegador de internet.	74
4.33. Modelo con los parámetros para almacenar la institución de educación de un candidato de Jobomas.	75
4.34. con los parámetros para almacenar la experiencia laboral de un candidato de Jobomas.	75
4.35. Modelo con los parámetros para almacenar datos personales de un candidato de Jobomas.	76
4.36. Modelo con los parámetros para almacenar los conocimientos en informática de un candidato de Jobomas.	77
4.37. Modelo con los parámetros para almacenar el resumen de un registro de la base de datos de Jobomas.	78
4.38. Modelo con los parámetros para almacenar los idiomas que maneja un candidato de Jobomas.	79
4.39. Modelo con los parámetros para almacenar otros conocimientos o habilidades de un candidato de Jobomas.	79
4.40. Modelo con los parámetros para almacenar un perfil de un candidato de Jobomas.	80
4.41. Modelo que contiene todos los parámetros recuperados del sitio <i>web</i> para un profesor.	81
4.42. Modelo con los parámetros para almacenar el resumen de un registro de la base de datos de Superprof.	82
4.43. Modelo con los parámetros que sobrescriben las celdas de las tablas que maneja la clase <i>DefaultTableCellRenderer</i> de java swing.	83
4.44. Modelo con los parámetros que sobrescriben los encabezados de las tablas que maneja la clase <i>TableCellRenderer</i> de java swing.	83
4.45. Modelo con los parámetros de creación de la tabla de resumen de los registros del sitio <i>web</i> Jobomas empleando la técnica de <i>web scraping</i> .	84
5.1. Ubicación del <i>ChromeDriver.exe</i>	87
5.2. Versión de <i>Java SE Runtime Enviroment</i>	87
5.3. Versión de Maven.	87
5.4. Versión de <i>SQL</i>	88
5.5. Versión del <i>SQL Developer</i> de Oracle.	88
5.6. Versión de Google Chrome.	89

5.7. Versión del <i>Chrome Driver</i>	89
5.8. Versión de <i>Open Office</i>	89
5.9. Tener acceso a Internet.	90
5.10. Acceso como sysdba.	91
5.11. Creación de tablespace.	91
5.12. Creación de usuario.	92
5.13. DDL de tabla ADMINISTRADOR.CREDENCIALES.	92
5.14. DDL de tablas que almacenan la información del sitio <i>web</i> Superprof	93
5.15. DDL de tablas que almacenan la información del sitio <i>web</i> Jobomas .	94
5.16. DDL de tablas que almacenan la información del sitio <i>web</i> Jobomas .	95
5.17. Archivos fuente.	96
5.18. Inicio de empaquetado	96
5.19. Fin de empaquetado.	97
5.20. Generación de carpeta target.	97
5.21. Generación de archivo jar.	97
5.22. Ejecución de aplicación.	98
5.23. Credenciales de acceso.	99
5.24. Ejecución de jar.	100
5.25. Credenciales incorrectas, se coloca al usuario incorrecto.	100
5.26. Mensaje en credenciales incorrectas.	101
5.27. Credenciales correctas.	101
5.28. Acceso correcto, vista de búsqueda de perfiles.	102
5.29. Habilitar buscador.	104
5.30. Buscador, rango de registros.	104
5.31. Buscador, caracteres válidos.	105
5.32. Buscador, búsqueda de perfiles.	105
5.33. Log, inicio.	106
5.34. Log, final.	107
5.35. Ejecución de procesos empleando Hilos.	108
5.36. Vista crear hoja.	109
5.37. Tabla Jobomas.	110
5.38. Tabla Superprof.	111
5.39. Crear hoja de cálculo y terminar.	111
5.40. Traza de fichero creado.	112
5.41. Archivo Jobomas.	112
5.42. Archivo Profesores.	113
5.43. Traza de termino de la ejecución de la aplicación.	113

6.1. Tiempos de obtención de datos.	116
6.2. Tabla Perfiles	117
6.3. Tabla Otros Conocimientos	117
6.4. Tabla Educación	118
6.5. Tabla Experiencia Laboral	119
6.6. Tabla Idiomas	120
6.7. Tabla Informática	120
6.8. Tabla Profesores	121
6.9. Tabla Clases	122
6.10. Tabla Nivel	122
6.11. Tabla Lugares de Clase	123
6.12. Hoja Candidatos	124
6.13. Hoja Idiomas	124
6.14. Hoja Educación	125
6.15. Hoja Experiencia Laboral	126
6.16. Hoja Informática	127
6.17. Hoja Otros	127
6.18. Hoja Superprof	128

Introducción

En el año 2020 la enfermedad COVID-19 proveniente del virus SARS-CoV-2 obligó al mundo a experimentar de forma masiva el trabajo en casa, esto trajo cambios en las prácticas de las empresas y en la forma de relacionarse con sus clientes y empleados. El trabajo virtual ha sido aceptado por diversas empresas, aumentó de 9% a 64% en tiempos de pandemia, los ejecutivos mencionan que el rendimiento de la empresa no se ha visto afectado en este cambio de esquema de trabajo a distancia sobre el presencial. Los perfiles tecnológicos son los más difíciles de cubrir, es una tarea difícil contratar a un empleado en este sector, siendo los centros urbanos los de mayor demanda aunque dada la pandemia se han contratado equipos remotos desde cualquier parte del planeta. El 35% de los empleados que renunciaron a su antiguo empleo aceptaron un trabajo remoto fuera de su país, ya que las barreras geográficas que impedían el reclutamiento internacional desaparecieron. La distribución geográfica de los empleados actuales en la misma ciudad bajó de un 59% a un 25% después de la pandemia.[2]

El ecosistema digital en América Latina de 2021 se centra en contenido, comunicación y *e-commerce*, los pagos digitales continúan avanzando en este año y los servicios de red son plataformas centrales de comunicación. Hay una revolución en el *e-commerce*, hubo un crecimiento real de 63.3% en 2020, la configuración de tiendas, envíos y pagos en su mayoría se han vuelto digitales. Los factores clave para sostener la expansión del uso de tecnologías e innovación en pagos de *e-commerce* de calidad son la logística, las plataformas en línea y la variación de los métodos de pago.[2]

En la actualidad los sistemas computacionales garantizan un óptimo funcionamiento en diversas tareas de empresas e instituciones, estas tareas van desde actividades simples hasta llegar a las más complejas que podemos imaginar. Los sistemas

1. INTRODUCCIÓN

computacionales están en constante evolución por lo que resulta importante mantenerse actualizado ya sea a nivel de software como de hardware cumpliendo con al menos con las condiciones de ambiente y configuración mínimas. Una óptima operación exige que las partes involucradas sean componentes funcionales o de lo contrario se pueden presentar fallos en cadena y exponenciales. El uso correcto de la tecnología da como resultado un ahorro en los costes de inversión y aprovechamiento de los recursos disponibles.[3]

El mundo se encuentra en la era del *Big Data* y el *web scraping* se ha vuelto una práctica común e importante porque permite la extracción de grandes cantidades de información de internet de forma automatizada. Esta técnica permite recopilar datos de interés, lo que posibilita a cualquier negocio obtener información abierta y estructurada para tomar mejores decisiones. Algunos ejemplos de lo que es posible llevar registro de forma automática son los cambios en las tendencias de precios de productos y servicios, la visualización de ofertas o actualización de divisas permitiendo así crear nuevos perfiles o enriqueciendo los ya existentes. En sus inicios esta tarea se realizaba de forma manual y repetitiva, pero hoy en día es obsoleta dada las grandes cantidades de información y los errores humanos que se pueden llegar a cometer. La automatización de tareas en el *web scraping* permite ahorrar recursos en tiempo y dinero mejorando la precisión, fiabilidad y calidad de los datos.

La publicación de perfiles y vacantes de empleo en sitios *web* está en pleno auge, esto aumentó debido a la situación actual ocasionada por la pandemia derivada del virus SARS-COV-2, por ello diversas bolsas de trabajo online y sitios de publicación de perfiles a través de internet cobraron mayor relevancia. Las empresas saben que el capital humano es de suma importancia, un buen equipo de trabajo cumplirá con los objetivos marcados por la empresa en el tiempo establecido y con la calidad esperada, tener responsables que hagan búsquedas de esta información en sitios *web* demanda un gran uso de recursos computacionales, económicos y, en especial, de tiempo. Esto se puede solventar por medio de técnicas de *web scraping* que permiten realizar el trabajo automático en un par de minutos, así se consiguen cantidades industriales de información sin tener un manejo prolongado del equipo de cómputo. A través de ciertos algoritmos de búsqueda automatizada se puede acceder a la *web* para extraer información relevante y actualizada de una o varias fuentes. Esta técnica permite el almacenamiento y la manipulación de los datos obtenidos como la generación de tablas, documentos o registros en base de datos.

Tener recursos humanos que hagan las búsquedas de perfiles laborales publicados en bolsas de trabajo online demanda un gran uso de recursos computacionales, económicos y de tiempo a las empresas, por ello, es necesario emplear técnicas automatizadas que faciliten esta actividad. Se propone la creación de una aplicación

de escritorio que utilice técnicas de *web scraping* para facilitarle al personal de la industria de Atracción de Talento la búsqueda de perfiles específicos en páginas online, este sistema ahorrará al cliente tiempo y recursos. Para llegar a esta solución se tendrá que indagar en la estructura de diferentes bolsas de trabajo online, definir las características principales de sus vacantes, realizar un modelo de base de datos acorde al análisis anterior, crear módulos para la obtención de datos online que se adapten al sitio *web* de interés y programar un orquestador de estos módulos, al final crear hojas de cálculo para presentar de manera amigable al usuario los resultados.

1. INTRODUCCIÓN

Marco Teórico

En este capítulo se presentan algunos conceptos que forman parte del marco teórico, y son importantes para entender el contexto en el que se desenvuelve el presente trabajo.

2.1 Bolsa de trabajo *online*

Una bolsa de trabajo *online* tiene como principal objetivo encontrar a través de un sitio *web* a un candidato adecuado para un puesto disponible de forma ágil y rápida, los publicados de vacantes y el acceso a los mismos mediante internet reduce los esfuerzos físicos y monetarios para quien realiza la publicación y para quien accede a esta; las bolsas de trabajo *online* son un mercado de trabajo grande y accesible. Una persona jurídica es la empresa que publica una vacante a un puesto disponible y una persona natural es aquella que accede a esta información a través del sitio *web*. Una bolsa de trabajo *online* es un intermediario entre la empresa reclutadora y el usuario que accede a esta información, la persona natural accede a esta información sin importar su región y al final es decisión de la persona natural si decide postular a la vacante o no.[4]

Las bolsas de trabajo *online* operan a través de un sitio *web* en internet, este lugar digital se vuelve el centro de convivencia entre la empresa y el buscador de empleo. Por un lado, la empresa publica de forma detallada la o las vacantes que tiene disponibles y, por otro lado, las personas que buscan una oportunidad laboral pueden subir su hoja de vida (*Currículum Vitae* o *CV*) directamente en el portal.

2. MARCO TEÓRICO

Dependiendo del sitio *web*, se cuentan con diversas formas de comunicación, una de ellas puede ser un sistema de chat o de correo propio del sitio.[4]

2.1.1 Historia

El los últimos años el desarrollo de tecnologías ha provocado que el internet tenga gran impacto en el modo de realizar las cosas en el día a día, la publicación de vacantes de empleo en portales de empleo virtuales no es la excepción. Se puede ver a esta forma de reclutamiento como una metodología novedosa, ya que el primer sitio *web* tiene lugar apenas hace 30 años. El desarrollo y mantenimiento de las bolsas de trabajo *online* impulsan a la economía, cultura, educación y al empleo en el mundo entero, por estas razones han surgido diversos sitios *web* a lo largo del planeta. Por mencionar a algunos sitios *web* tenemos[5, 6, 7, 8, 9]:

- Computrabajo: Fundada en 1999, tiene más de 20 años de experiencia. Hoy en día se pueden encontrar ofertas de trabajos remotos, trabajos desde casa.
- LinkedIn: Lanzada en 2003, comenzó como una red social y más tarde se convirtió en un portal de empleo en especial para quienes buscan un cargo ejecutivo. Esta plataforma permite mantener relación directa con muchas empresas.
- InfoJobs: Con 20 años en la *web*. Nacida en España y actualmente tiene presencia en América Latina. Destaca por permitir crear un blog con las ideas del usuario, haciendo énfasis en las habilidades profesionales.
- Superprof: Descrita como una comunidad nacida en 2013, nace en Biarritz, Francia. Se enfoca en agrupar a más de 17'009,375 profesores quienes ofrecen su servicio desde 28 países y en 14 idiomas.

Como podemos observar, las plataformas tienen cualidades que los distinguen y se enfocan a diferentes sectores de la población, sin embargo se puede decir que estos sitios no están enfocados al 100 % de la población, siendo así la rural la que menos presencia tiene[5].

2.1.2 Características

La principal característica es que las ofertas laborales al ser publicadas en la *web* permiten ser accedidas desde cualquier parte del mundo que tenga acceso a internet. La presencia de un portal de empleo puede extenderse a diferentes países a lo largo del planeta y el usuario puede obtener la información o contactar al reclutador de

forma *online* sin la necesidad de viajar o que ambas personas estén una frente a otra de manera física. Los *Currículums Vitae* son almacenados en base de datos y son agrupados en campos laborales. [4]

2.1.3 Información concurrente

Las bolsas de trabajo *online* tienen un módulo de aspirantes, en donde los candidatos registran información de interés para el reclutador. Normalmente se debe proporcionar la siguiente información[4] :

- Datos personales
- Formación académica
- Idiomas
- Nivel de computación
- Experiencia laboral
- Preguntas generales

2.2 Web scraping

En su mayoría, la única manera para visualizar y almacenar los datos en una página *web* es utilizando un explorador *web* y visitando diversos sitios específicos en internet para seleccionar los datos de interés, realizar este proceso de forma manual implica demasiado costo en tiempo, esfuerzo y recursos en recolectar grandes cantidades de datos. El *web scraping* realiza este proceso de forma automatizada disminuyendo así estos costos. Un *scraper*, figura 2.1, es una página *web* enfocada a copiar datos de terceros y plasmarlos dentro de su propia estructura dándoles la estructura que les convenga. [10]

Se llama *web scraping* al proceso codificado y automatizado encargado de obtener datos contenidos en páginas *web*. *Web scraping* consiste en la ejecución de un algoritmo escrito en algún lenguaje de programación que se conecta a un servidor *web* y que de la respuesta, generalmente una página *web*, obtiene los datos con información de interés para el analista de datos. El analista de datos es la persona encargada de identificar el patrón que siguen los datos en una página *web* para posteriormente codificar el algoritmo de extracción, procesamiento y almacenamiento de los mismos ya que en un principio estos parecen poco estructurados. [11]

2. MARCO TEÓRICO

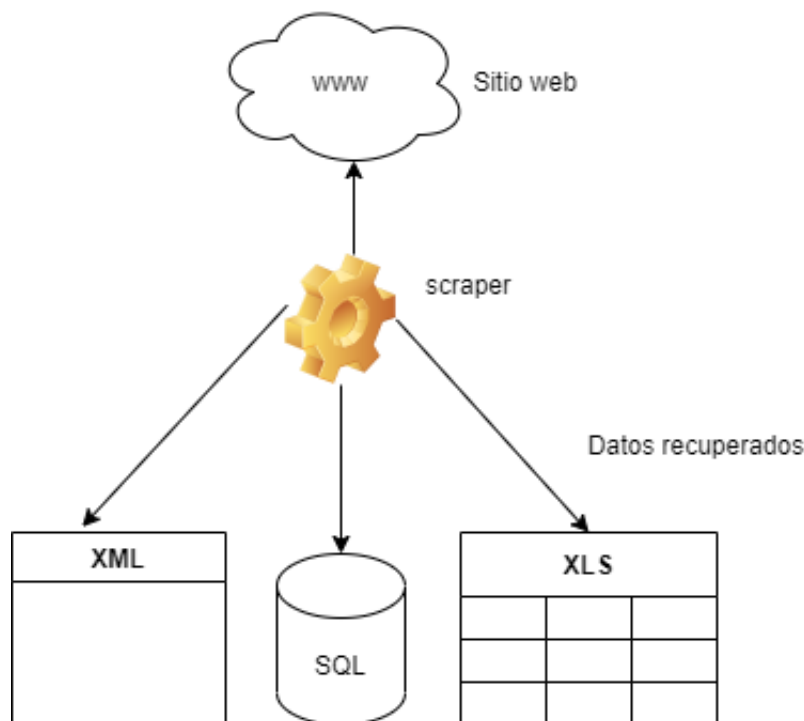


Figura 2.1: Diagrama lógico de un *Scraper*.

2.2.1 Componentes

Se identifican dos componentes principales en el *web scraping*: [10]

- **Web crawler:** También es llamado *spider* o *araña web*. Se trata de una aplicación de software que realiza una tarea repetitiva, un *bot*, programado para navegar en la página *web* deseada, haciendo uso de algoritmos la inspecciona de forma metódica y automatizada. Descarga una serie de direcciones *web* que le son proporcionadas, después analiza estas páginas y al final busca enlaces a páginas nuevas para así ser almacenadas en base de datos o en memoria.
- **Extractor de datos:** Es el encargado de extraer los datos de interés de una página *web*, es llamado *web scraper*. Se encarga de extraer los datos haciendo uso de selectores como *XPath* o *CSS Selectors*. Termina con el almacenamiento de los datos en el formato que el usuario requiera.

2.2.2 Beneficios

El *web scraping* optimiza los costos de recolección manual de grandes cantidades de datos cuando no se tienen protocolos de acceso (*APIs*) a los mismos, realiza la recolección de información de forma automática y aunque esta está publicada en Internet se considera inaccesible por no tener una estructura organizada. Al emplear *web scraping* se le da la estructura a los datos de uno o más sitios *web*, dándole sentido así a los datos. [11]. En primera instancia los beneficios se reflejan en tiempo, costes y esfuerzo humano. [10]

2.2.3 Cuestiones legales

Los límites del uso de *web scraping* sobre un sitio *web* no están muy bien definidos por un organismo o ley, por ello se analiza cada caso por separado.[12] Al emplear el *web scraping* se deben considerar una serie de aspectos legales. La extracción de datos es un privilegio proporcionado por los diferentes sitios *web* que se apegan a las leyes vigentes y a la seguridad cibernética aplicada a sus actividades, por ello la persona que implementa esta técnica de obtención de información debe apegarse a los términos de uso especificados por el sitio *web* de interés. Las leyes en el país en donde se realice la práctica de *web scraping* influyen, por ejemplo, en países en donde el *web scraping* es reconocido y aceptado el problema principal no es la extracción de datos si no el uso que se le da a los mismos.[10]

La legalidad de *web scraping* es un área gris en el campo legal, entendiendo como legal al cumplimiento de las leyes aplicables. Se puede caer en ilegalidades como infracción de derechos de autor, incumplimiento de contratos por términos de uso publicados en el sitio *web*, violación del derecho de privacidad, uso ilegal o fraudulento de datos, daños al sitio *web*, violaciones de privacidad de personas u organizaciones, por mencionar algunos.

La presente aplicación emplea los datos de los usuarios con los mismos fines que han sido publicados en el respectivo sitio *web*, no se hace uso fraudulento de los mismos, no se ve afectada la integridad de la información, se accede al sitio *web* y a los perfiles de manera legal, el sitio *web* no tiene afectaciones en su operativa por el uso del presente software y no se argumenta el origen de los datos como propios. Para trabajos futuros se recomienda seguir con los puntos antes mencionados. [13]

2. MARCO TEÓRICO

2.2.4 Aplicaciones y Trabajos Relacionados

Las aplicaciones son diversas, de forma general podemos pensar en un programa robot que explore el contenido de sitios *web* y que, sin importar el tamaño y estructura de los mismos, extraiga datos de interés para nosotros. Los métodos de obtención de datos son amplias, depende del analista de datos encontrar la solución a cada problema.[11]

Algunos ejemplos de problemas resueltos empleando la técnica de *web scraping* para extraer datos son:

- *Web scraping* de los Perfiles y Publicaciones de una Afiliación en *Google Scholar* utilizando Aplicaciones *web* e implementando un Algoritmo en R. El artículo trata el tema de extraer datos de *Google Scholar*, derivado de las pruebas comprueba la velocidad de extracción de datos, además de su eficiencia en el formato de salida.[14]
- Automatización de *web scraping* de los diarios de noticias para la empresa Isuri, San Martín de Porres. En esta investigación se accede a diarios de noticias *web* para la empresa Isuri. Concluye en que la empresa tuvo gran consumo de recursos monetarios y de personal, los mismos que fueron reducidos gracias al empleo de *web scraping*. [15]
- Herramientas *web scraping* para la Adquisición de Datos Turísticos. Este artículo plantea la recolección de datos como un problema dentro del turismo. La solución que propone es el uso de *web scraping*, resultando en ahorro de tiempo y empleando los datos en favor del sector turístico.[16]

2.3 Programación orientada a objetos (POO)

La programación orientada a objetos(POO) emplea objetos ligados mediante mensajes para la solución a cualquier problema, estos objetos y mensajes son organizados para emular al mundo real. Se entiende como objeto a una entidad del mundo real y consta de atributos(características) y métodos(acciones).[17]

2.3.1 Lenguajes

Estos son algunos ejemplos de lenguajes de programación orientados a objetos a lo largo de la historia: [18]

2.3 Programación orientada a objetos (POO)

- Simula-67: Nace en los años 60 de las manos de dos investigadores noruegos, Quisten Negar y Ole-Johan Del, se considera la base de la programación orientada a objetos. Este lenguaje hace uso de clases, herencia, métodos virtuales, entre otras definiciones de la programación orientada a objetos.
- Smalltalk: Creado en 1980, es el primer lenguaje orientado a objetos que dispone de un entorno de desarrollo gráfico integrado. Fue diseñado por el equipo Alan Kay de origen americano del centro de investigación informática californiana de XEROX. Hace uso de la noción de compilación dinámica del concepto "*Just In Time Compiler*", emplea un código intermedio.
- C++: Originalmente llamado *C with classes*, nace en 1980 a manos del danés Bjarne Stroustrup quien lo desarrolla para AT&T. C++ es una evolución del lenguaje C creado por el canadiense-americano Kernighan y Ritchie, hoy en día es muy utilizado.
- Oak: Desarrollado en 1990 por James Gosling para Sun Microsystems, a partir de 1995 será llamado Java. Es robusto y seguro, en la actualidad es muy utilizado y pertenece a Oracle.
- C#: Presentado en 2001 por Microsoft, es desarrollado por el danés Anders Hejlsberg. Se considera un lenguaje muy cercano a java.

2.3.2 Principios

La programación orientada a objetos nos permite construir sistemas complejos de una forma organizada y legible. Los 4 principios que se enumeran a continuación son herramientas que nos ayudan a escribir un código reutilizable y limpio.

- La encapsulación: Este principio consiste en restringir los efectos del cambio sobre los objetos, protege la integridad de los datos. El acceso a los datos está gestionado por procedimientos y no se puede acceder a ellos de forma directa.[19]

El encapsulamiento consiste en usar los objetos sin conocer el detalle de su implementación y utiliza métodos de acceso para disminuir el acoplamiento entre clases. Oculta la implementación de código, dejando al descubierto el qué puede hacer una clase y no el cómo se realiza, esta ventaja aísla una parte de código que puede ser reemplazable en cualquier momento sin tener afectaciones sobre el resto del programa.[20]

2. MARCO TEÓRICO

- La herencia: Este mecanismo ofrece una solución a la problemática de duplicación de código. Con frecuencia nos vemos en la necesidad de emplear copias de código, ya sea idénticas o muy similares, para ello en la herencia se define a una nueva clase(subclase) como extensión de una existente (superclase). Una ventaja se refiere al mantenimiento de clases, al realizar un cambio en código de una superclase que tiene afectación en n subclases, esta modificación solo se hace una vez en lugar de realizarlo n veces.[20]
- Polimorfismo: El polimorfismo consiste en ejecutar la misma operación pero con diferente implementación sobre distintos tipos de objetos, esta implementación depende del objeto sobre el que se hace el llamado del método.[21]
- La abstracción: Corresponde a las operaciones en un nivel más alto de un problema, se centra en funcionalidades generales. Es una definición que no se asocia a un caso concreto, se ve como una funcionalidad general sin tener en cuenta la complejidad interna.[20]

Una abstracción expresa las principales características de un objeto, es un conjunto generalizado de operaciones y características, es el modelo útil de una entidad sin ser la representación individual y concreta de la misma.[22]

2.3.3 Hilos

Un hilo representa un único flujo de ejecución en un programa que se ejecuta en un entorno de direcciones, es una secuencia de código o módulo específico de ejecución del programa y es supervisado por un proceso padre. Java soporta varios procesos que se ejecutan a la vez y cada uno en su propio espacio de direcciones, es decir, cada proceso tiene asignado un contexto. Ver figura 2.2.

Los hilos son llamados con frecuencia procesos ligeros dado que son un módulo que corre dentro de la ejecución de un programa y estos dependen de los recursos del programa. [23]

2.3 Programación orientada a objetos (POO)

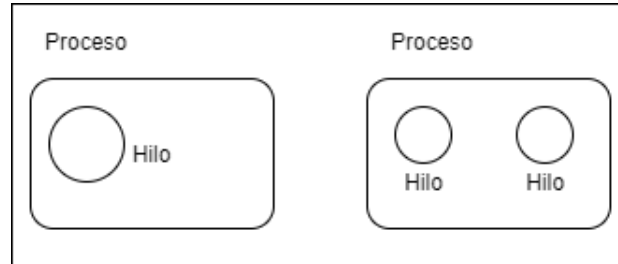


Figura 2.2: Relación y dependencia entre hilos y procesos.

La concurrencia en hilos es la cooperación de los procesos ligeros en el manejo de un único procesador. La velocidad en las operaciones de entrada y salida que ofrecen los procesadores permiten suspender de forma temporal la ejecución de un hilo para intercalar el uso del procesador con otro hilo. Este proceso ocurre de tal manera que parece que varios hilos se están ejecutando simultáneamente pero en realidad es el procesador el que se libera un momento para ejecutar un hilo u otro en intervalos de tiempo diferentes, este proceso se conoce como multiprogramación.

El soporte de multi-hilos consiste en la ejecución simultánea de diferentes procesos en un único procesador, estos procesos se ejecutan en diferentes instantes de tiempo. Este proceso no es perceptible para el usuario y le permite a las computadoras que tienen un único procesador hacer uso de la concurrencia.[24]

2.3.4 Beneficios

Los beneficios que trae el empleo de la programación orientada a objetos son[25, 26] :

- Descompone un problema en entidades u objetos modelados al mundo real, esto facilita al entendimiento del mismo ya que ajusta su estructura y modo de pensar al de hombre y no al de la máquina.
- Dado el enfoque realista que tiene, facilita el análisis y diseño del programa.
- Ayuda al rápido crecimiento de un programa y a su mantenimiento gracias a la reutilización y el polimorfismo.
- Gracias al encapsulamiento la identificación de errores, el entendimiento del código fuente y las modificaciones se vuelven rápidas.
- Es un paradigma robusto que sobrevive a la obsolescencia de cualquier sistema.

2. MARCO TEÓRICO

- Permite extraer con facilidad partes de un sistema antiguo, las interfaces son limpias y fiables.
- Los componentes construidos son reusables, por ello el software es extensible. Una extensión de código no significa recodificar desde cero.

2.4 Patrones de diseño

Desde el punto de vista de la computación, los patrones de diseño agilizan el desarrollo de software ya que ofrecen soluciones comprobadas a problemas comunes. Un patrón de diseño describe una solución aceptada y comprobada a un problema común, dándole una terminología a dicha solución para distinguirla de entre las demás.[27]

El conocimiento de patrones de diseño de software es fundamental en la formación básica de aquellas personas interesadas en la ingeniería o ciencias de la computación, es un conocimiento esencial que debe ser adquirido a principios de la carrera y aplicado a lo largo de la misma.[27]

2.4.1 Historia

En la actualidad el desarrollo de software se ha vuelto un reto en cuanto a patrones de diseño, en 1979 el arquitecto Christopher Alexander publica *The Timeless Way of Building* en donde propuso el uso de patrones para la construcción de edificios con mayor calidad y en donde menciona el utilizar una solución a un problema común en lugar de pensarlo cada vez que se requiera.[28]

En 1987 Ward Cunningham y Kent Beck se dan cuenta de las similitudes que existen entre la arquitectura y el desarrollo de software en cuanto a empleo de patrones de diseño, publicaron *Using Pattern Languages for OO Programs* luego de desarrollar cinco patrones de interacción hombre - máquina derivado de emplear varias ideas del arquitecto Christopher Alexander.[28]

A partir de 1994 y luego de que el grupo *Gang of Four* (GoF) compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides publicaran el libro *Design Patterns: Elements of Reusable Object Oriented Software* los patrones de diseño tuvieron gran éxito a lo largo del mundo de la computación. Este libro publica 23 patrones de diseño comunes a la época y aunque estos autores no inventaron los patrones de diseño sí difundieron la idea para ser utilizada en el desarrollo de software. [28]

2.4.2 Clasificación

El grupo *Gang of Four (GoF)* clasificó a los patrones de diseño en 3 grandes categorías de acuerdo a su propósito, a continuación se especifica la clasificación de los 23 patrones mencionados al principio de esta sección[28, 29, 30].

- Creacionales: Enfocados a la creación de nuevos objetos, en esta clasificación el proceso de creación se separa de la implementación. Independizan el ¿Qué?, ¿Quién?, ¿Cómo? y ¿Cuándo? en la creación del objeto. Tenemos los patrones *Factory method*, *Abstract factory*, *Builder*, *Singleton* y *Prototype*.
- Estructurales: Esta clasificación compone interfaces o implementaciones mediante el empleo de la herencia. Describe la forma en que se componen los objetos para así obtener la nueva funcionalidad, esta composición puede cambiar en tiempo de ejecución. Aquí se agrupan las clases y objetos que comparten características similares y de esta manera es como se forman estructuras más grandes. En este grupo encontramos los patrones de diseño *Adapter*, *Bridge*, *Composite*, *Decorator*, *Facade*, *Flyweight* y *Proxy*.
- Comportamiento: Este grupo de patrones de diseño facilitan el manejo de comportamientos complejos, describe cómo es que los objetos y las clases interactúan entre sí y cómo es que dividen las responsabilidades unos con otros. Aquí podemos encontrar el *Chain of responsibility*, *Command*, *Interpreter*, *Iterator*, *Mediator*, *Memento*, *Observer*, *State*, *Strategy*, *Template method* y *Visitor*.

2.4.2.1 CRUD

El paradigma CRUD es común en la construcción de aplicaciones *web* y está relacionado con la gestión y mantenibilidad de información digital, le indica a los desarrolladores cómo construir un modelo completo y utilizable. Las operaciones CRUD actúan sobre la capa de acceso a dato, estas operaciones son *create*(inserción), *read*(lectura), *update*(actualización) y *delete*(borrado) y permiten el mantenimiento de los datos almacenados en una Base de Datos. En HTTP estas operaciones se mapean a los verbos *GET* (lectura de registros), *PUT* (actualización múltiple o masiva de registros), *DELETE* (Borrado de registros) y *POST* (Creación de un nuevo registro)[31]. Se permite la administración de roles entre los usuarios.

2.4.2.2 DTO

Data Transfer Object por sus siglas en inglés, *DTO* es un patrón de diseño usado para el intercambio de datos en llamadas que tienen altos costos de recursos, estos

2. MARCO TEÓRICO

llamados pueden ser entre diferentes capas de una misma aplicación[32]. Este patrón de diseño permite la creación de objetos que sirven para transportar datos entre los procesos de un programa y así disminuir el número de llamados a métodos, en cada llamada se busca transmitir la mayor cantidad de datos necesarios. Un *DTO* debe tener métodos getter, métodos que proveen el valor de un atributo de un objeto, y setter, métodos que permiten modificar el valor de un atributo de un objeto[33]; como ejemplo de la estructura de un *DTO* podemos ver la figura 2.3. Permite la distribución de objetos ya que son objetos serializables que viajan a través del back-end y el front-end, de esta manera dan integridad a los datos[32].

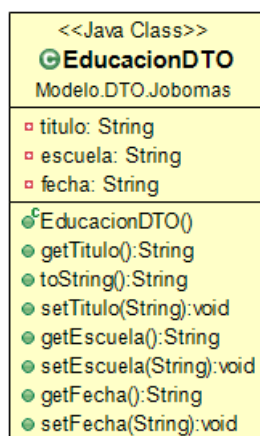


Figura 2.3: Ejemplo de clase *DTO*.

2.4.2.3 Modelo Vista Controlador (MVC)

En 1979 es descrito por Trygve Reenskaug el patrón MVC, Modelo-Vista-Controlador, e introducido en el Smalltalk-80 del lenguaje de programación Smalltalk, es un patrón de arquitectura de software aplicado a sistemas múltiples y sincronizados con los mismos datos, estos sistemas deben de ser de representación gráfica, en ventanas separadas y el diseño presenta diferentes escalas de aumento. Las partes de este patrón de diseño son el Modelo, la Vista y el Controlador, estas son entidades separadas que reducen el esfuerzo de programación a estas implementaciones. El diagrama de secuencia del patrón de diseño MVC se observa en la figura 2.4 y la interrelación entre estos elementos en la figura 2.5. Por ejemplo, un cambio en el Modelo aplica en automático a las Vistas aplicables. Un Modelo tiene la capacidad de definir múltiples Vistas.[34]

2.4 Patrones de diseño

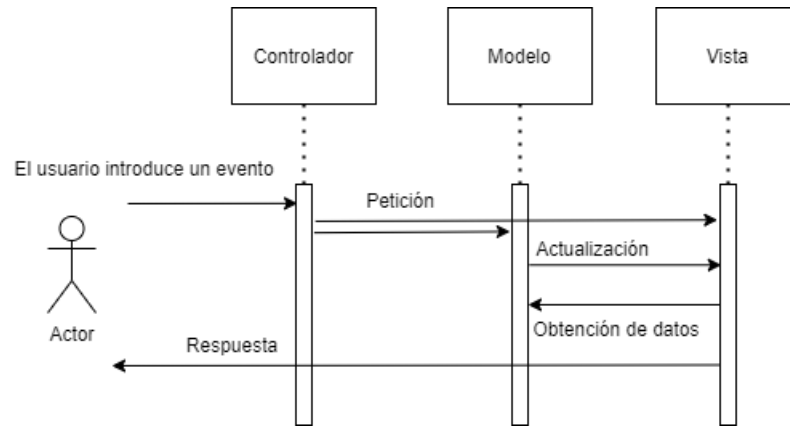


Figura 2.4: Diagrama de secuencia de MVC.

- Modelo: Objeto que representa, maneja y controla las transformaciones de los datos del programa. Son los datos y reglas de negocio.
- Vista: Objeto encargado de la representación visual de los datos representados por el Modelo hacia el usuario.
- Controlador: Actúa sobre los datos del Modelo, es llamado al realizar algún cambio sobre los mismos. Se encarga de dar sentido a las indicaciones que manda el usuario. Gestiona las entradas del usuario.

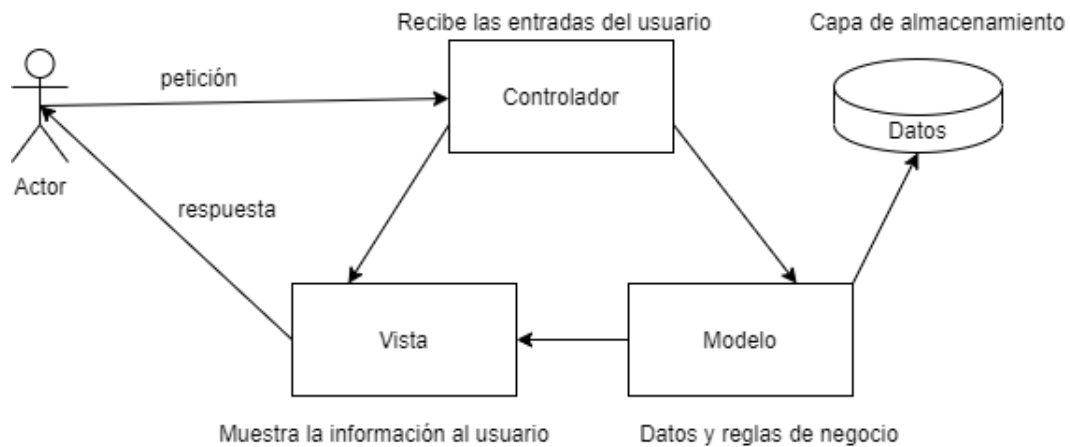


Figura 2.5: Interrelación entre los elementos del patrón MVC.

2. MARCO TEÓRICO

2.4.2.3.1 Ventajas

Entre las ventajas que presenta este modelo de arquitectura de software podemos encontrar[34, 35]

- Los componentes se pueden programar por separado para después ser incorporados en un solo programa.
- Implementación de cada componente por separado, su separación es clara
- *API*, Interfaz de Programación de Aplicación, fácil de entender. El reemplazo de cualquiera de los tres componentes no representa un problema y sin que los otros componentes se vean afectados.
- La actualización y el mantenimiento del software es sencillo, el tiempo para esta actividad es reducido.
- Conexión dinámica ente el Modelo y sus Vistas, se da en tiempo de ejecución.
- Produce componentes a partir de las interfaces
- Aplica con facilidad los cambios de estas interfaces

2.5 Lenguaje de Modelado Unificado (*UML*)

El Lenguaje Unificado de Modelado, *UML (Unified Modeling Language)*, permite el análisis y diseño orientado a objetos. En 1997 la estandarización de la *OMG (Object Management Group)* convierte al *UML* en un estándar. *UML* describe de una manera formal a los sistemas en los ciclos de desarrollo de software orientados a objetos, esto permite conectar los modelos con diversos lenguajes de programación como *Java*, *C++* o *Visual Basic*. El modelo resultante no es ambiguo, tiene una semántica bien definida, el lenguaje es semiformal ya que resulta fácil de entender y manipular, es flexible, detalla de manera muy completa y gráfica lo que está en código. Este modelo es rico en información, su interpretación respecto a la arquitectura de los sistemas de software es inequívoca. El meta modelo de *UML* consta de:[36]

- Estructurales o estáticos (conceptuales o lógicos: clases, colaboraciones, casos de uso y clases activas)
- Elementos conductuales o dinámicos (interacciones y máquinas de estado),

2.5 Lenguaje de Modelado Unificado (*UML*)

- Relaciones o conexiones entre los diferentes elementos (dependencia, asociación, generalización y realización)
- Elementos de agrupación
- Elementos para anotaciones.

El Lenguaje de Modelado Unificado(*UML*) es la representación gráfica de los elementos estáticos y dinámicos de una aplicación de software. Otras características que presenta son: [37]

- Los diagramas de secuencias representan la interacción entre las instancias de un sistema de software.
- Hace uso de condiciones mediante el empleo de banderas.
- Permite la identificación de un objeto.
- Identifica la clase del objeto.
- Modela la comunicación entre objetos.
- Modela el comportamiento alternativo u opcional.
- Emplea el operador negativo para comportamientos no esperados.
- Usa gates, estos modelan la entrada de mensajes externos o la salida de mensajes fuera de un marco.

Existen diferentes tipos de mensajes, estos pueden ser síncronos(el emisor espera por un mensaje de confirmación para determinar el éxito de la operación), asíncronos(el emisor no espera por un mensaje de confirmación para continuar con su operación), plano (sin distinción de sincronía, depende de un contexto o interpretación respecto a si es síncrono o no) y mensajes de retorno (luego de la recepción de mensajes asíncronos, es un mensaje de confirmación y su uso es opcional)[36].

2.5.1 Tipos de diagramas en *UML*

Los diagramas que no ofrece *UML* son:[36]

2. MARCO TEÓRICO

- Diagramas de Clases: Estos diagramas muestran a las clases y a los objetos de importancia en el sistema de software, también señala la visibilidad de cada una de las clases en el modelo. Captura la estructura estática de la información. Con frecuencia emplea la generalización, herencia y re-uso. Un ejemplo de diagrama de clases se observa en la figura 2.6.

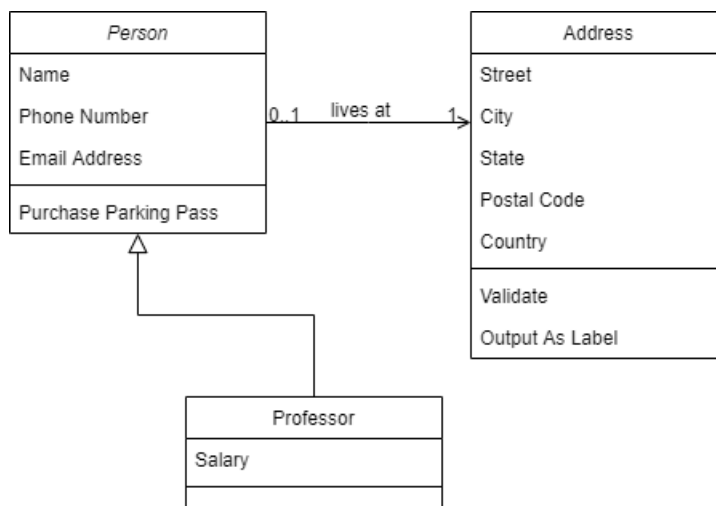


Figura 2.6: Diagrama de clases.

- Diagramas de Estado: También llamado carta de estado o diagrama de transición de estados. Estos diagramas modelan el comportamiento dinámico, muestran el conjunto de estados y todos los cambios por los que pasa un objeto durante su vida de ejecución. Es un modelo de comunicación. Son usados cuando el control es influenciado en gran medida por eventos externos. Un rectángulo con los vértices redondeados representa a un estado. Una línea recta que conecta a dos estados es la transición entre estados, ver figura 2.7.

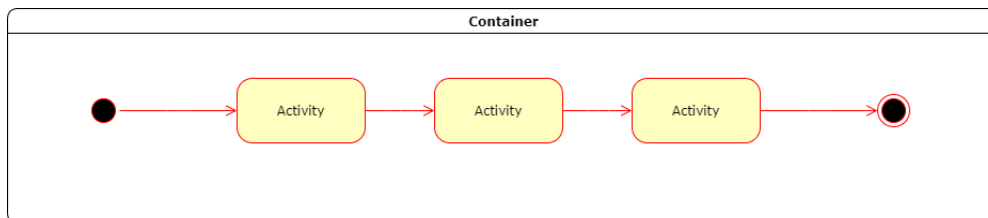


Figura 2.7: Diagrama de estado.

2.5 Lenguaje de Modelado Unificado (*UML*)

- Diagramas de Actividad: Es empleado cuando el control es principalmente síncrono, modela el flujo de control e información de un proceso. Derivado de los diagramas de estados. Tienen la característica de que todos los estados son estados de acción, se describe que acción se realiza en esta etapa del proceso, el envío de la mayoría de las transiciones dependen del término de la acción ejecutada en el estado anterior. Da a conocer los pasos y puntos de decisión ejecutados en el comportamiento de un objeto o en un proceso. Representa transiciones internas sin dar importancia a las transiciones externas. Un rectángulo de vértices redondeadas representa un paso, se diferencia de la representación de un estado al ser esta más ovalada. Los rombos representan puntos de decisión, ver figura 2.8.

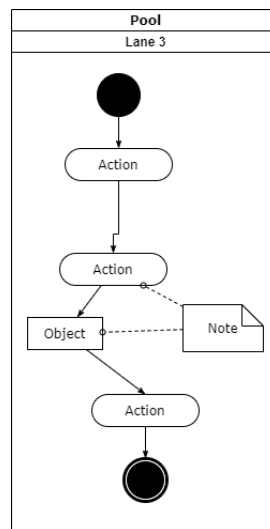


Figura 2.8: Diagrama de actividad.

- Diagramas de Casos de Uso: Se emplea en las primeras fases de análisis del sistema, muestra los servicios que se pueden recibir del mismo. Estos son las distintas operaciones que un sistema puede ejecutar así como a la relación que tienen con su entorno, entendiendo a entorno como al usuario o a otras aplicaciones. Estos diagramas proveen una muestra externa, desde el punto de vista del cliente, ver figura 2.9.

2. MARCO TEÓRICO

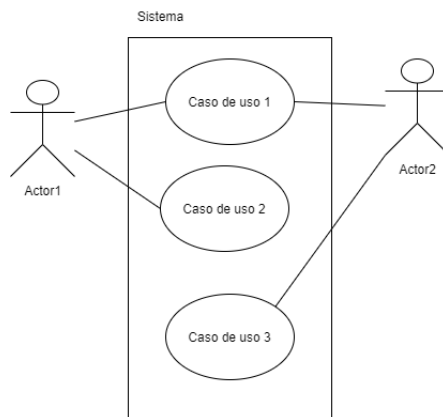


Figura 2.9: Diagrama de casos de uso.

2.6 Tecnologías *Open Source*

Las tecnologías *Open Source*, de uso libre o código abierto, ha incrementado su uso en los últimos años. Estas tecnologías son un soporte importante en los desarrollos debido a que ofrecen soluciones a diversos problemas en los desarrollos de software. Existe gran cantidad de herramientas *Open Source*, esto permite adecuar la que mejor se adapte a las necesidades del desarrollador o cliente y cumpliendo así a las exigencias del desarrollo de software.[38]

El software *Open Source* incluye el código fuente, esto le permite a las comunidades de desarrolladores que lo mejoren, lo corrijan, lo prueben y lo mantengan. Es un software gratuito, se ahorran costos de licencia.[39]

2.6.1 Ventajas

Este tipo de software le permite al desarrollador ser autónomo, no preocuparse por el pago de licencias para tener una parcialidad o totalidad del programa porque no hay funciones bloqueadas, la licencia no discrimina a ninguna persona ni limita los campos de aplicación, nadie puede dejarse fuera.[39]

La ideología que rige a este tipo de software es el compartir el código para que este sea modificado y que de esta manera tenga una calidad superior. Este software está en constante evolución, desarrollo, mantenimiento y mejora. Cada usuario lo usa de acuerdo a sus necesidades y dada la rápida aplicación los errores se corrigen a una velocidad impresionante, además de tener mejoras en tiempos muy cortos.[40]

2.6.2 Historia

Algunas fechas destacables en el movimiento *Open Source* son:[41]

- A finales de los años 50 y a principios de los 60 se desarrolla la comunidad a la que hoy en día se le conoce como hackers. En un principio el software era desarrollado por hackers.
- En los años 70 Richard Stallman se une al *Massachusetts Institute of Technology* (MIT), en donde se comenzó a reflexionar sobre el papel que desempeña la tecnología en la sociedad, durante esta época el software es accesible sin restricción alguna.
- En los años 80 las computadoras comenzaron a incorporar software de propietarios, un ejemplo es Microsoft, en este momento el código empezó a ser producido por empresas y se vuelve privado lo que antes era público.
- En 1983 Richard Stallman anunció el desarrollo de un sistema operativo libre, este sistema operativo es el GNU (*GNU's Not Unix*). junto al concepto de software libre nace el *Open Source* con las posibilidades y ventajas que este ofrece a la comunidad de desarrolladores.
- En 1991 Linus Torvalds acabó completando y desarrollando el sistema operativo GNU/Linux, estos son sistemas operativos tipo Unix de código abierto.

2.6.3 Características

Algunas características del software *Open Source* son:[40]

- Libre distribución.
- Ahorro en costos de licencia.
- Fácil obtención de código fuente.
- Permite el compartimiento de trabajos derivados.
- Integridad al acceder al código fuente del autor.
- No se permite la discriminación de personas o grupos.

2. MARCO TEÓRICO

2.6.4 Ejemplos

Estas tecnologías tienen a su propia comunidad de código abierto, las cuales en ocasiones agradecen las contribuciones a través de sus sitios *web* oficiales. Estas tecnologías son posible gracias a los colaboradores voluntarios, quienes han dedicado gran cantidad de horas de su propio tiempo para hacer que el código fuente esté disponible de manera gratuita para que cualquier persona pueda usarlo, disfrutarlo y mejorarlo.

2.6.4.1 Java SE

Java Standard Edition (Java SE) es una plataforma empleada en la programación de servidores de aplicaciones en el lenguaje de programación Java. Proporciona al usuario una base para la creación y despliegue de aplicaciones centradas en la red[42]. *Java Standard Edition* incluye un *API* con toda la funcionalidad del lenguaje de programación Java, se compone de una máquina virtual, herramientas de desarrollo, tecnologías de despliegue y librerías de clase propias de la tecnología Java.[43]

Java es un lenguaje de programación orientado a objetos de sintaxis sencilla. Las aplicaciones que se desarrollan son portables a numerosos sistemas operativos gracias a que tiene integrada su propia máquina virtual, estos programas solo tiene etapa de ejecución y compilación. El compilador genera *bytecode* que es independiente de cualquier arquitectura y al final el *bytecode* se traduce a código nativo. Java se encuentra en un campo de aplicación en la red de internet y redes locales, es un lenguaje en constante evolución y dada su riqueza funcional es de los primeros en ser enseñado en las escuelas y universidades. La comunidad Java está compuesta por millones de usuarios activos a nivel global.[44]

2.6.4.2 Java Swing

Es una librería de interfaz de usuario, su función es personalizar los componentes gráficos del software, dar forma al comportamiento de la interfaz de usuario. Permite la implementación de diversos objetos, estos son componentes visuales como botones, ventanas, ventanas emergentes, campos de texto, combo *box*, notificaciones, *check box*, tablas, barras de menú, entre otros. La portabilidad de esta librería está garantizada porque es independientes del sistema operativo donde se ejecuta la aplicación. Es fácil de emplear en un entorno de desarrollo integrado (IDE) como *JBuilder*, *Sun Forte for Java*, *Eclipse* o *Netbeans* por mencionar a algunos.[45]

Java Swing está fuertemente ligado al patrón de diseño MVC, esto tiene la ventaja de la fácil y rápida creación de interfaces de usuario, la interfaz de usuario se adapta

a la estructura de datos del usuario y dado que los modelos están separados de la vista la personalización de estos componentes es extensa.[45]

2.6.4.3 Apache Maven

Apache Maven es una herramienta que se basa en el concepto de un modelo de objetos de proyecto, denominado POM, facilita la comprensión y gestión de cualquier proyecto basado en Java. Maven se encarga de administrar la construcción, los informes y la documentación de un proyecto. Otras características de Maven son: [46]

- Maven es una forma estándar de construir proyectos, proporciona las mejores prácticas de desarrollo.
- La información que proporciona sobre el proyecto es de calidad
- Fácilmente se puede validar la consistencia del proyecto, informa sobre las dependencias utilizadas.
- Permite la fácil publicación de los proyectos así como la manera de compartir los JAR.
- Facilita el trabajo diario de los desarrolladores.
- El tiempo de comprensión del estado completo de un proyecto disminuye al emplear Maven.
- Ahorra tiempo de navegación entre los proyectos
- Crea un informe de pruebas unitarias que incluye la cobertura del código.

2. MARCO TEÓRICO

```
<dependencies>
  <!-- base de datos oracle -->
  <dependency>
    <groupId>com.oracle</groupId>
    <artifactId>ojdbc6</artifactId>
    <version>11.1.0.7.0</version>
  </dependency>

  <!-- hoja de cálculo -->
  <dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>3.16</version>
  </dependency>

  <!-- selenium -->
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.0.0-alpha-6</version>
  </dependency>
```

Figura 2.10: Dependencias Maven en archivo pom.xml empleadas en el presente trabajo.

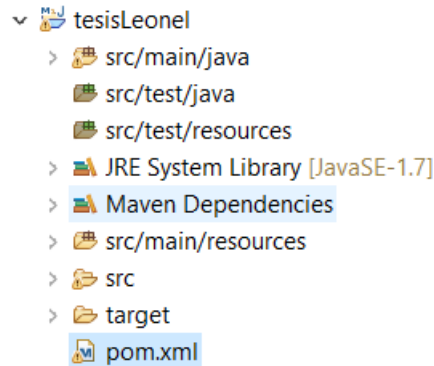


Figura 2.11: Estructura de un proyecto Maven.

2.6.4.4 Selenium *WebDriver*

Selenium es un proyecto formado por diversas herramientas y librerías que permiten la automatización de navegadores *web*. Selenium emula la interacción de un usuario con un navegador *web*, promueve la automatización de plataformas *web*. Selenium está disponible en los lenguajes de programación Java, Python, CSharp, Ruby, JavaScript y Kotlin. Por lo general es utilizado para la automatización de pruebas de sitios *web* de escritorio, este proyecto hace uso de una interfaz llamada *WebDriver*.^[47]

WebDriver permite la programación de diversos pasos o instrucciones que pueden ejecutar gran cantidad de navegadores *web*. *WebDriver* es diseñado como una interfaz de programación simple y concisa, es una *API* compacta orientada a objetos y controla al navegador de manera efectiva. *WebDriver* utiliza la *API* de automatización para controlar las instrucciones que el navegador *web* debe de ejecutar, lo hace de forma nativa tal cual lo haría un usuario. El control del navegador puede ser de manera local o de forma remota empleando el servidor de Selenium, esta navegación remota supone un adelanto en los procesos de automatización *web*.^[47]

2.6.4.5 JDBC driver

Un driver *Java Database Connectivity* (JDBC), controlador de conectividad a bases de datos en español, es un componente de software que le permite a los desarrolladores Java poder conectar eficazmente una aplicación a una base de datos Oracle. En el sitio oficial de Oracle se puede elegir el jar correcto y los artefactos adecuados en función de las necesidades de la aplicación y sus casos de uso que incluyen: producción, producción con instrumentación de depuración, observabilidad

2. MARCO TEÓRICO

y observabilidad con instrumentación de depuración, entre otros.[48]

El desarrollo de aplicaciones Java que hacen uso de base de datos Oracle resulta rápido y simple, puede ser en la nube o en las instalaciones. El driver JDBC otorga a los desarrolladores Java las funciones más actuales como base de datos autónomas Oracle, autoajuste de rendimiento, conexiones sencillas con acceso a las propiedades de conexión, alta disponibilidad, procesamiento en memoria y bases de datos conectadas entre sí para diseñar y desarrollar aplicaciones de alto rendimiento, seguras, escalables, confiables y con soporte a nuevos tipos de datos. Las constantes actualizaciones y características en este controlador simplifican la conexión al procesamiento autónomo de transacciones y al almacenamiento autónomo de datos, lo hace siempre apegándose a los estándares Java.[48]

2.6.4.6 Oracle *SQL*

Una Base de Datos en informática es un conjunto de datos o información estructurada de forma organizada, se almacena de forma digital en un sistema informático y son controladas por un DBMS (*DataBase Management System* o Sistema de Gestión de Bases de Datos). El conjunto de los datos y el DBMS es lo que conocemos como un sistema de Base de Datos o simplemente Base de Datos. Los datos normalmente se almacenan en una matriz de 2x2, siendo así accesibles mediante filas y columnas en tablas, el procesamiento y la consulta de datos resulta eficaz. Las Bases de Datos utilizan el lenguaje de consulta estructurada (*SQL, Structured Query Language*) para acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos.[49]

En la década de 1970 fue IBM y Oracle uno de los principales contribuyentes de desarrollo *SQL*, dieron lugar al estándar ANSI *SQL*. Una base de datos *SQL* Oracle está disponible en la nube y en las propias instalaciones, permite consultar, cargar, actualizar, eliminar y extraer datos. Mediante la ejecución de *scripts* (comandos) de *SQL* y DDL (lenguaje de definición de datos) es posible administrar el rendimiento, la seguridad, el almacenamiento, la configuración, entre otras funciones de una Base de Datos Oracle *SQL*. [49]

Oracle SQL Developer es un entorno de desarrollo integrado, este ayuda a simplificar los desarrollos y a administrar las Bases de Datos Oracle *SQL* ya sea en sus implementaciones tradicionales o en la nube. *Oracle SQL Developer* es la principal herramienta con tecnología Java para los sistemas operativos Windows, OS X y Linux que administra una base de datos Oracle *SQL*, proporciona tres interfaces a los usuarios: escritorio, navegador y línea de comandos. Proporciona el espacio de trabajo para ejecutar consultas y comandos, una consola de *DBA* (Administrador de Base de Datos), una interfaz de informes, una solución completa de modelado de

datos, una plataforma de migración de Base de Datos Oracle en las instalaciones a Oracle en la nube y una plataforma de migración para trasladar las Bases de Datos de terceros a Oracle.[50]

2.6.4.7 **GitHub**

Git fue desarrollado en 2005 por Linus Torvalds y actualmente está en constante mantenimiento, es el sistema de control de versiones más utilizado en el mundo, emplea una arquitectura distribuida en donde el repositorio es una copia de trabajo del código y alberga el historial completo de los cambios. Su principal característica es su modelo de ramificación, esto lleva a tener múltiples sucursales completamente independientes entre si y para crear, fusionar y eliminar estas líneas de desarrollo mediante las ramas lleva prácticamente solo algunos segundos. Gran cantidad de proyectos comerciales y de código abierto emplean Git para el control de versiones de su trabajo, este sistema se ha adaptado a gran cantidad de IDEs y sistemas operativos para que funcionen en sincronía y a la perfección. Git toma en cuenta aspectos como el rendimiento, la seguridad y la flexibilidad que un desarrollador requiere.[51, 52]

GitHub es un espacio en la *web* y un servicio en la nube destinado al desarrollo y almacenamiento de software, es una plataforma que aloja repositorios utilizando el sistema de registro y control de versiones Git, la comunidad está integrada por más de 65 millones de desarrolladores y cuenta con más de 200 millones de proyectos. En este sitio se permite registrar y sincronizar cualquier cambio en el código alojándolo en un repositorio público o privado. En esta plataforma se encuentran proyectos aprobados por la comunidad de desarrolladores de software. Se puede experimentar, eliminar errores, crear nuevas funciones, proponer cambios, revisar código, hacer sugerencias, solicitar información, integrar código en línea porque el código que se registra en GitHub puede llegar a una base de código o millones.[53]

2.6.4.8 **Hoja de cálculo**

Una hoja de cálculo es un software que permite realizar operaciones o cálculos con datos numéricos contenidos en una tabla con celdas verticales y horizontales que conforman filas y columnas, permite automatizar cálculos matemáticos, técnicos, estadísticos y financieros complejos. permite trabajar con gran cantidad de información, con diversos parámetros y con funciones propias del programa. Se pueden ejecutar macros, estos son una serie de comandos que automatizan instrucciones de manera secuencial y repetida. Las hojas de cálculo sirven de apoyo en el análisis de

2. MARCO TEÓRICO

los datos recopilados, ayudan a conocer el estado actual y entender que decisiones tomar en base a la información mostrada.[54]

Este software no requiere que el usuario tenga amplios conocimientos en informática ya que es muy intuitivo y gráfico, muestra la información de manera interesante y amigable al usuario mediante diferentes estilos, filtros y gráficas. Hay diversas empresas dedicadas al desarrollo de hojas de cálculo, algunas que son de código libre y gratuitas son KSpread incluida en el paquete KOffice de Linux y OpenCalc incluida en el paquete OpenOffice.[54]

Metodologías de trabajo

Los proyectos de trabajo son servicios que constan de un principio y de un final, tienen como finalidad un producto u objetivo final y de acuerdo a este marco temporal se tienen dos posibles finales: objetivo alcanzado o fracaso. Para evaluar la calidad en cuanto efectividad, productividad, seguridad y satisfacción de los sistemas de información es que se han desarrollado normas técnicas internacionales, estas normas buscan el mejoramiento de estos sistemas. Existen dos grandes enfoques en cuanto a metodologías de trabajo refiere, las tradicionales y las ágiles.[55]

- Tradicionales: La documentación es de gran importancia durante el ciclo de desarrollo del proyecto y la atención se centra en cumplir con un plan de proyecto, un cambio es sinónimo de un alto costo. El desarrollo está dividido de forma secuencial, sus partes importantes son la planificación, documentación y los procesos, como ejemplo tenemos la *Rational Unified Process*(RUP) y *Microsoft Solution Framework*(MSF).
- Ágiles: Tienen capacidad de respuesta a los cambios, es de importancia la relación con el cliente, está formado por un equipo auto organizado y multidisciplinario. En esta metodología se cuidan las relaciones entre los individuos, es preferible entregar al cliente activos funcionales sobre la documentación, la colaboración con el cliente está por encima de la negociación de los contratos y no hay gran apego al seguimiento de un estricto plan de trabajo. En este enfoque podemos encontrar *eXtreme Programming*(XP), *Scrum*, *Iconix*, *Cristal Methods*, *AUP* entre otras.

3. METODOLOGÍAS DE TRABAJO

3.1 Scrum

Creada en los años 80 por Ikujiro Nonaka e Hirotaka Takeuchi. En los años 90 Jeff Sutherland y Ken Schwaber formalizaron un marco de trabajo. En los años 2000 es que este enfoque toma fuerza con la Guía de Scrum y el Manifiesto ágil de desarrollo de *Software*, en este periodo se crearon programas y certificaciones que intensificaron al rededor del mundo el uso de *Scrum*. [56]

3.1.1 Características

Scrum es la metodología de desarrollo ágil de *software* más conocidas a nivel mundial, se basa en el trabajo de equipos autónomos para el desarrollo de productos. La metodología *Scrum* ofrece productos de buena calidad a bajos costes, generalmente está destinada al desarrollo de productos tecnológicos y está compuesta por diversos roles (*Product Owner*, *Scrum Master*, equipo de desarrollo, usuarios, *stakeholders* y *anagers*) y elementos (*product backlog*, *sprint backlog* e incremento). Durante el proceso de desarrollo del proyecto existen reuniones de planificación, seguimiento y revisión, por ejemplo, en los ciclos de desarrollo (*sprints*) hay reuniones diarias (*daily*) en donde se da seguimiento a los avances fijados y respondiendo a las preguntas ¿Qué hice desde la reunión anterior?, ¿Qué haré hasta una nueva reunión? y ¿Tengo algún inconveniente que hay que solucionar para poder continuar?. Otras características de la metodología *Scrum* son: [1]

- Equipo multidisciplinario.
- Equipos auto-organizados.
- Transmisión del conocimiento.

3.1.2 Roles

Los roles están formados por dos grupos, cerdos y gallinas (surge de un chiste de un cerdo y una gallina que intentan poner un restaurante, ver figura 3.1). [1]



Figura 3.1: Cerdos y gallinas en *Scrum*[1] .

Los cerdos está formado por aquellas personas comprometidas con el desarrollo del proyecto *Scrum*.

- *Product Owner*: Esta persona es el intermediario entre el cliente y el equipo *Scrum*, conoce el producto que el cliente solicita, escribe las ideas del cliente, las ordena por prioridad y toma las decisiones.
- *Scrum Master*: Es la persona encargada de emplear el modelo y metodología *Scrum* dentro del equipo *Scrum*, interactúa con el cliente, con áreas de apoyo y con el equipo para evitar inconvenientes o malos entendidos en las definiciones de los requerimientos del proyecto o necesidades del propio equipo *Scrum*.
- Equipo de Desarrollo: Equipo de entre 5 y 9 personas, auto-organizado y multidisciplinario, ellos mismos estiman el esfuerzo requerido para completar las tareas.

Las gallinas están fuera del proceso *Scrum*, pero son necesarios en la planificación de cada *sprint* y en la retroalimentación de los avances en el proyectos.

- Usuarios: Son las personas que harán uso del producto final.
- *Stakeholders*: Estas personas participan en las revisiones de los *sprints*, a ellos el producto les produce un beneficio. Son los clientes del *Scrum*.
- *Managers*: Estas personas participan en la toma de decisiones de selección de objetivos y requisitos finales.

3. METODOLOGÍAS DE TRABAJO

3.1.3 Hitos

Los elementos que forman *Scrum* son el *Product Backlog*, *Sprint Backlog* e Incremento. En estos elementos se almacena una lista priorizada con todos los requisitos o funcionalidades definidas, cada requisito aporta un valor al producto final. Durante cada *sprint* o iteración de la metodología *Scrum* tenemos:[1]

- Las historias de Usuario: Describe una funcionalidad del *software*, estimación de tiempo de realización, prioridad y dependencias. Esta debe ser completada con pruebas de aceptación.
- Formato de la Pila Del Producto (*Product Backlog*): Es la lista priorizada de requisitos que llevan al producto final, en esta lista se identifica y describe la funcionalidad así como su estimación de desarrollo.
- *Sprint Backlog*: Consiste en una lista de tareas que son asignadas al equipo de desarrollo del *Scrum* durante un *sprint*, en este listado el proyecto se descompone en unidades más pequeñas permitiendo detectar impedimentos en el mismo. Aquí podemos observar el listado de tareas con la persona responsable de su cumplimiento, el tiempo que lleva realizándose la tarea y el tiempo que falta por completarla.
- Incremento: Son los requisitos que se han completado en un *sprint* y aportan valor al producto final.

Existen 5 diferentes fases en formas de ciclos en un proyecto *Scrum* hasta completar las tareas del *backlog*.

1. Revisión de planes de versión: Se valida la situación actual del avance del proyecto y se determinan los requerimientos próximos a cubrir. Es una fase de validación de estándares que consiste en la distribución, revisión y ajuste de lo que se espera del producto.
2. *Sprint*: Es una iteración en días en la fase de desarrollo. Se maneja el empaquetado, revisión y ajuste del producto.
3. Revisión de *sprint*: Conocida como *sprint review*. Es una reunión del equipo *Scrum* en donde se revisan el progreso del producto de acuerdo a los alcanzables estimados, en esta fase se puede modificar el *backlog* y se planifica la próxima versión del *software* de acuerdo a los puntos descritos en el *backlog*.

4. Retroalimentación: Es una fase de análisis del requerimiento y su tiempo de realización, costo y calidad. Se valida si lo desarrollado concuerda con lo planificado.
5. Cierre: Es el fin del proyecto. Se plantea el hacer una versión del producto, testear o marketing.

3.1.4 Herramientas

Hay gran cantidad de herramientas para la ejecución de *Scrum*, desde hojas de cálculo, pizarras o herramientas *online* como:

- www.atlassian.com/es/software/jira: Herramienta que permite planificar, supervisar y publicar *software* de gran calidad.
- www.ntaskmanager.com: *Software* de gestión de proyectos que permite a sus equipos colaborar, planificar, analizar y gestionar las tareas diarias.
- www.quickscrum.com: Herramienta de gestión de proyectos para equipos ágiles.
- www.scrumdo.com: Es una potente plataforma de gestión del trabajo.
- www.vivifyscrum.com: Herramienta de gestión de proyectos basada en web para pequeños equipos ágiles y grandes organizaciones.
- monday.com: El *software* ágil para cualquier marco de *Scrum*.

3.1.5 Ventajas

Algunas ventajas que tiene *Scrum* sobre otras metodologías de trabajo son:[56]

- Satisfacción del cliente: El cliente se vuelve parte del proceso *Scrum* y se ve comprometido con el resultado final.
- Simplicidad: Están perfectamente identificados los eventos, así como quienes participan, su objetivo, el tiempo que debe tomar la realización de estos y cuál es el resultado esperado.
- Inspección: La reunión diaria, la revisión y la retrospectiva del *sprint* permiten dar seguimiento al avance del proyecto e identificar qué debe ser mejorado.

3. METODOLOGÍAS DE TRABAJO

- Adaptación: Es la disposición que se tiene al cambio de las características que se tienen del producto siempre que no se afecte la entrega antes pactada. Esta flexibilidad satisface al cliente.
- Trabajo en equipo: Durante el desarrollo del producto, el equipo se adapta para mejorar, cada integrante es parte fundamental del equipo.
- Tiempo: El marco de tiempo de iteración es de 2 a 4 semanas.
- El proceso: Es el equipo *Scrum* quien dicta que herramientas se van a emplear en el desarrollo de *software*.
- Orden: Hay una prioridad definida en cuanto a las funcionalidades del producto final.
- Retroalimentación: Hay retroalimentación por parte del equipo *Scrum* al final de cada iteración.

3.2 ISO/IEC 9126-1

En los últimos años ha aumentado el acceso a tecnologías que facilitan el manejo de información y la toma de decisiones en el sector empresarial, llegando así a la parte que influye de manera directa en el éxito o fracaso en el desarrollo de productos de *software*. Los sistemas de gestión de calidad permiten a las organizaciones mejorar su funcionamiento mediante el correcto manejo de sus recursos y logrando resultados económicos positivos.[57]

ISO (Organización Internacional de Normalización) es una organización a nivel mundial encargada de normalizar estándares, su trabajo consiste en preparar las normas internacionales mediante comités técnicos especializados en una materia.[58]

3.2.1 Objetivo

ISO /IEC 9126-1 es una norma que detalla las etapas para el proceso de evaluación de calidad, mediante métricas, de un sistema de información. Mide el grado de satisfacción por parte del cliente con respecto al producto.[57]

3.2.2 Características

3.2.2.1 Funcionalidad

La principal función de ISO/IEC 9126-1 consiste en evaluar las características de calidad de uso de un sistema de información a través de los usuarios, definiendo el ¿qué se va a medir?, ¿cómo se va a medir? y al final evaluar los resultados obtenidos basándose en los objetivos el alcance definido. Define las características que un *software* debe de cubrir para considerar que es de calidad[57]

3.2.2.2 Características

El estándar ISO9126-1 presenta dos modelos de calidad:[58]

1. Calidad interna y externa

- Calidad interna: Conjunto de atributos del *software* que permiten evaluar la capacidad que este tiene para satisfacer necesidades bajo un escenario específico. Consiste en la evaluación por especificación de requerimientos, arquitectura, diseño, segmentos de código fuente, entre otros.
- Calidad externa: Este modelo de calidad es evaluado en el momento en que un módulo o la totalidad del *software* se ejecuta en una computadora o en una red simulando un escenario lo más cercanamente posible al real.

2. Calidad en uso:

- Usabilidad: capacidad del *software* para ser entendido y manejado por el usuario bajo un escenario específico.
- Funcionalidad: capacidad del *software* de ser ejecutado correctamente bajo un escenario específico.
- Confiabilidad: capacidad del *software* de mantener un nivel óptimo de rendimiento bajo un escenario específico.
- Eficiencia: relación entre el grado de rendimiento y los recursos utilizados bajo un escenario específico.
- Mantenimiento: capacidad del *software* de ser modificado.
- Portabilidad: capacidad del *software* de cambiar entre ambientes.

Los aspectos que un evaluador toma en cuenta son: [57]

3. METODOLOGÍAS DE TRABAJO

1. Establecer los requerimientos de evaluación: En esta fase se define el propósito de la evaluación, el tipo de producto a evaluar y el modelo de calidad en que se basa la evaluación.
2. Especificar la evaluación: Se establece el contexto de uso de *software*, consiste en una relación de un grupo de usuarios y una o varias tareas de evaluación asignadas y el ambiente computacional en el que se efectuaron. El ambiente en el que se efectúan estas evaluaciones debe de ser un caso real productivo. De acuerdo a las métricas y la escala de Likert, los resultados obtenidos se pueden valorar como muy en desacuerdo, en desacuerdo, indiferente, de acuerdo y muy de acuerdo.

3.2.3 Ventajas

El estándar ISO/IEC 9126-1 le permite a un producto de *software* desarrollar su funcionalidad de acuerdo a las métricas y lineamientos en que se basa la calidad de uso, estas son efectividad, productividad, seguridad y satisfacción en un contexto específico de uso. El apego a este estándar durante el ciclo de desarrollo de *software* permite la ejecución de métodos que lleven a la producción de *software* de calidad, libre de defectos, que cumplen con su especificación y funcionalidad.[58]

3.3 ISO/IEC 29110

Las PyMEs(pequeñas y medianas empresas) dedicadas al desarrollo de *software* han tenido gran crecimiento en los últimos años, de aquí surgió la necesidad de contar con estándares o modelos apegados a su estructura y capacidad interna que les permitieran mejorar en el aspecto de implementación de perfiles. *ISO* se dio cuenta de esta necesidad y creo el estándar ISO/IEC 29110 dedicado a la descripción de perfiles de *software* de pequeño y mediano aporte.[59]

Un perfil se describe como un subconjunto de uno o más estándares que sirven para llevar a cabo una función, se componen por elementos extraídos de los estándares y cumplen con una funcionalidad específica. Un perfil se considera como un sistema de referencia significativo para los clientes, usuarios y proveedores, es una base reconocida internacionalmente. Un Perfil Internacional Estandarizado (*International Standardized Profile, ISP*) se refiere a un documento aprobado internacionalmente que especifica a uno o más perfiles.[60]

3.3.1 Objetivo

El estándar ISO/IEC 29110 consiste en un conjunto de buenas prácticas para asistir y evaluar los procesos de mejora a las PyMEs dedicadas al desarrollo de *software*. Consiste en un marco común para describir perfiles evaluables del ciclo de vida de *software* en el uso de PyMEs.[59]

ISO/IEC 29110 presenta una serie de guías enfocadas a la implementación de Perfiles de Ciclo de Vida para Empresas Muy Pequeñas (*Software Engineering. Lifecycle Profiles for Very Small Enterprises, VSE*).[60]

3.3.2 Características

El estándar ISO/IEC 29110 se compone del Proceso de Administración de Proyecto (AP) y del Proceso de Implementación de *Software* (IS). Tanto AP como IS constan de roles, actividades y documentos externos que se deben cumplir al evaluar el estado de los procesos. Esta norma cuenta con tres perfiles: Perfil Básico: Publicado en 2010 como ISO/IEC 29110 Perfil Básico, Perfil Intermedio y Perfil Avanzado se encuentran en desarrollo. El Perfil Básico se compone de:[59]

- Proceso de Administración de Proyecto (AP): Establece las tareas de los proyectos, lleva a cabo el cumplimiento de los objetivos de forma sistemática, valida la calidad del producto así como su tiempo y costos esperados.
- Proceso de Implementación *Software* (IS): De acuerdo a los requisitos especificados, asegura el cumplimiento sistemático de las actividades de análisis, diseño, construcción, integración y pruebas de *software*.

3.3.3 Ventajas

El estándar ISO/IEC 29110 le permite al Ciclo de Vida para Empresas Muy Pequeñas (*VSE*):[60]

- Generar un grupo de perfiles que permitan la construcción o mejoren procesos existentes.
- Otorga a las *VSE* un reconocimiento como generadoras de sistemas de *software* de calidad y a bajo costo.
- Se crean guías fáciles de entender, accesibles y utilizables por las *VSE*.

3. METODOLOGÍAS DE TRABAJO

- Atender las necesidades del mercado referentes a perfiles y niveles en dominios específicos para que las *VSE* logren un *software* de calidad.
- Le permite a las *VSE* considerar las necesidades, problemas y riesgos de sus dominios.
- Para proyectos complejos, otorga una guía para el trabajo conjunto de varias *VSE*.
- Las guías y perfiles creados son escalables.

El presente trabajo se enfoca en el uso de las metodologías ágiles, siendo más específicos en la *Scrum*. Se contará con un tablero virtual en donde se tendrá una lista de tareas que aporten valor al producto final. Las tareas pasarán por diferentes fases del desarrollo, estas son: Por hacer, En progreso, Hecho. Cuando se tenga un contratiempo y se requiera ayuda con alguna tarea se dará a conocer. Los cambios para mejorar el producto final se harán presentes durante la vida del proyecto. El estándar ISO9126-1 nos permitirá evaluar la calidad interna y externa del producto, es decir, conforme se van desarrollando ciertos módulos se irán probando y al finalizar el desarrollo se hará un test de la totalidad de la funcionalidad de la aplicación de *software* en un equipo de cómputo debidamente ambientado. El estándar ISO/IEC 29110 nos ayudará a describir los perfiles para cumplir con los objetivos de forma sistemática, con la calidad y los tiempos esperados. En la figura 3.2 vemos a manera de módulos la funcionalidad que se va a desarrollar.

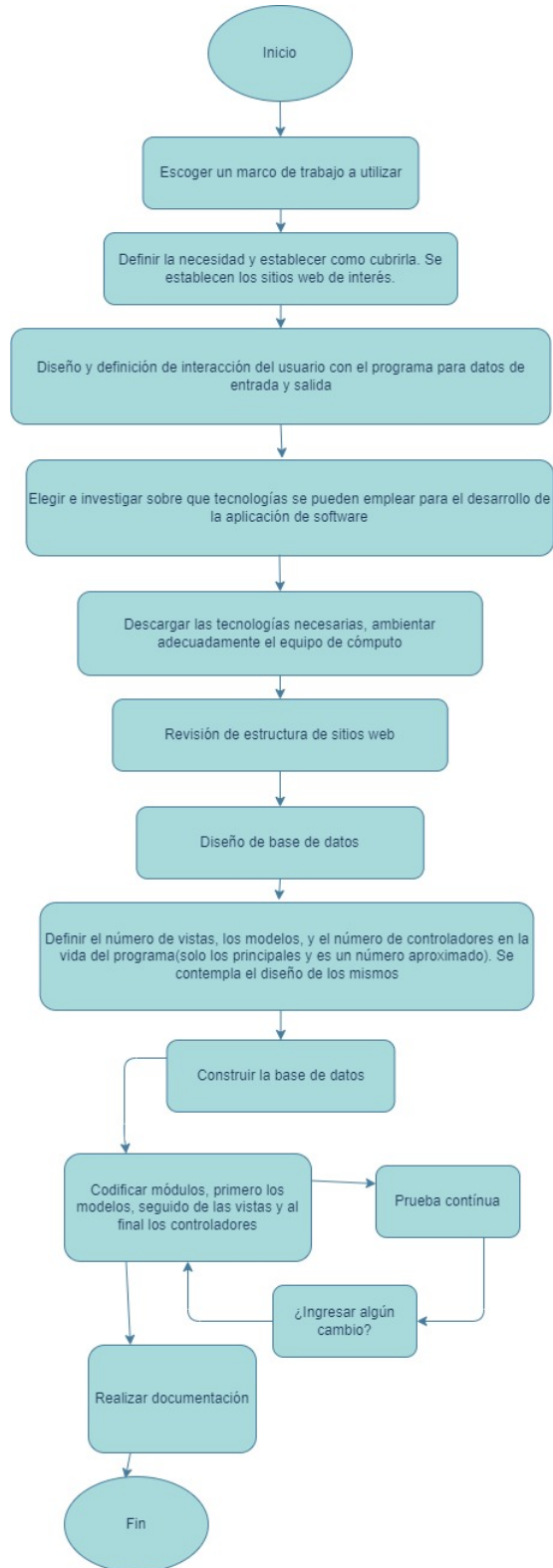


Figura 3.2: Módulos del programa.

3. METODOLOGÍAS DE TRABAJO

Arquitectura de la aplicación

En este capítulo se muestran los diagramas de diseño empleados para desarrollar los diferentes módulos de la aplicación. En cuanto al desarrollo de software se utilizó el patrón de diseño Modelo Vista Controlador y se modeló empleando las herramientas propias de los Entornos Integrados con los que se programó. Mediante diferentes diagramas se puede ver a grandes rasgos la lógica de programación, las propiedades de los atributos de los objetos y de las clases, la organización de las clases y los paquetes y la relación directa entre llamados de clases.

4.1 Diagrama *UML* de la Arquitectura Conceptual

A continuación se muestra el diagrama *UML* completo de la Arquitectura conceptual, aquí podemos ver las relaciones entre todas las clases que se invocan durante el ciclo de vida del programa. El diagrama respeta cierta simetría dado que se ha empleado el patrón de diseño Modelo Vista Controlador y a simple vista podemos ver que una diferencia notable es el número de clases *DTO* o Modelos empleados en el caso de Jobomas contra el caso de Superprof, esto se debe a que en cada sitio *web* la información se ha estructurado de manera diferente.

Se identifica el principio y el final de la ejecución del programa, así como resulta relativamente sencillo identificar a las clases que son controladores, modelos o vistas. Se observa como las clases se invocan unas a otras de manera secuencial y organizada. Un aspecto más que podemos destacar es el punto del programa en donde se realiza el lanzamiento de los Hilos de ejecución.

4. ARQUITECTURA DE LA APLICACIÓN

En la figura 4.1 se observa al diagrama *UML* de la Arquitectura Conceptual completa, se muestran todas las clases y las relaciones que estas tienen entre sí. Al seccionar el diagrama completo tenemos 3 partes:

- En la figura 4.2 se muestra la clase principal y el lanzamiento de los hilos en la ejecución del programa.
- En la figura 4.3 se muestran algunas clases encargadas del proceso sobre la página Superprof.
- En la figura 4.4 se muestran algunas clases encargadas del proceso sobre la página Jobomas.

Los diagramas de las imágenes 4.1, 4.2, 4.3 y 4.4 se han obtenido empleando la herramienta de modelado propia del *IDE NeatBeans*, estos se obtienen de manera automática y no hay manera de alterarlos.

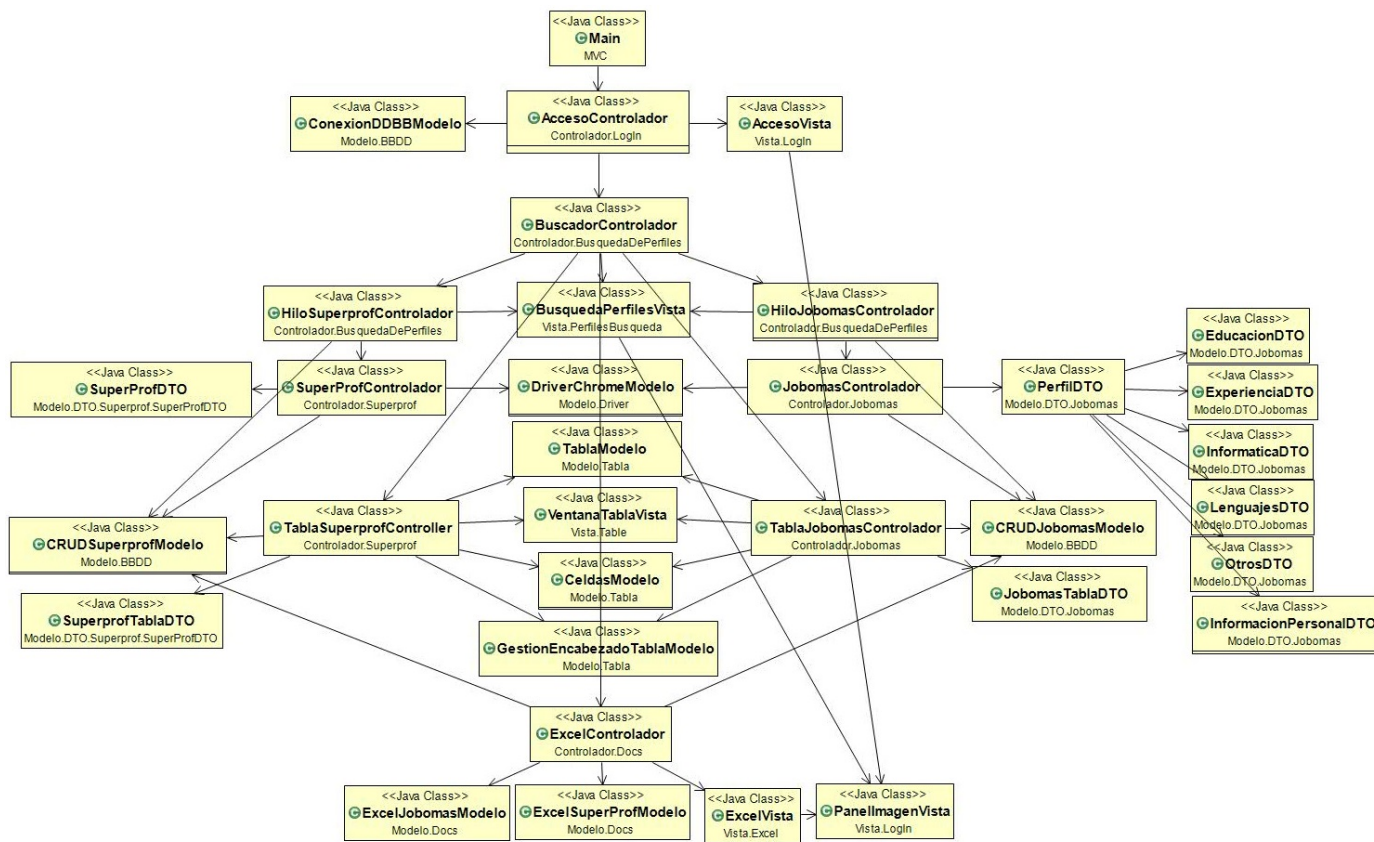


Figura 4.1: Diagrama *UML* de la Arquitectura Conceptual completa.

4.1 Diagrama UML de la Arquitectura Conceptual

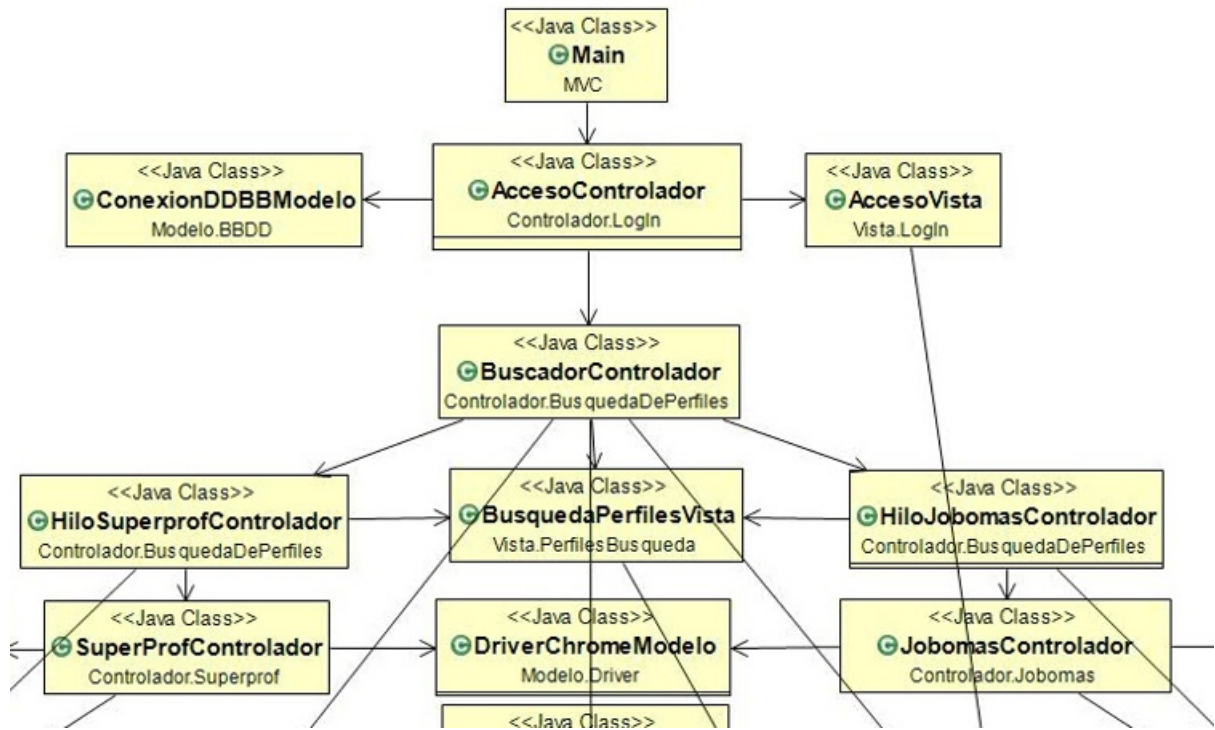


Figura 4.2: UML con los Hilos de ejecución.

4. ARQUITECTURA DE LA APLICACIÓN

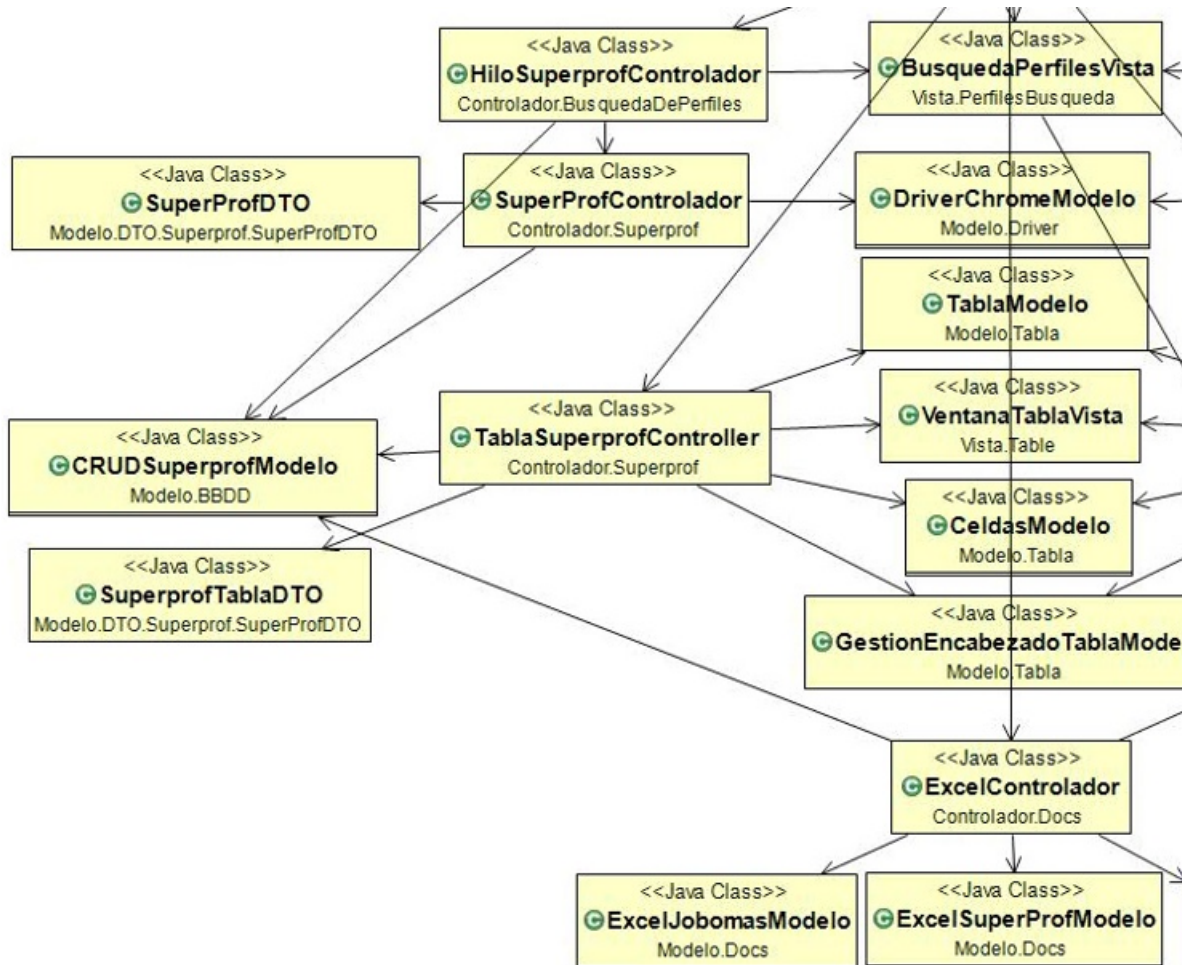


Figura 4.3: UML de los módulos de Superprof.

4.1 Diagrama UML de la Arquitectura Conceptual

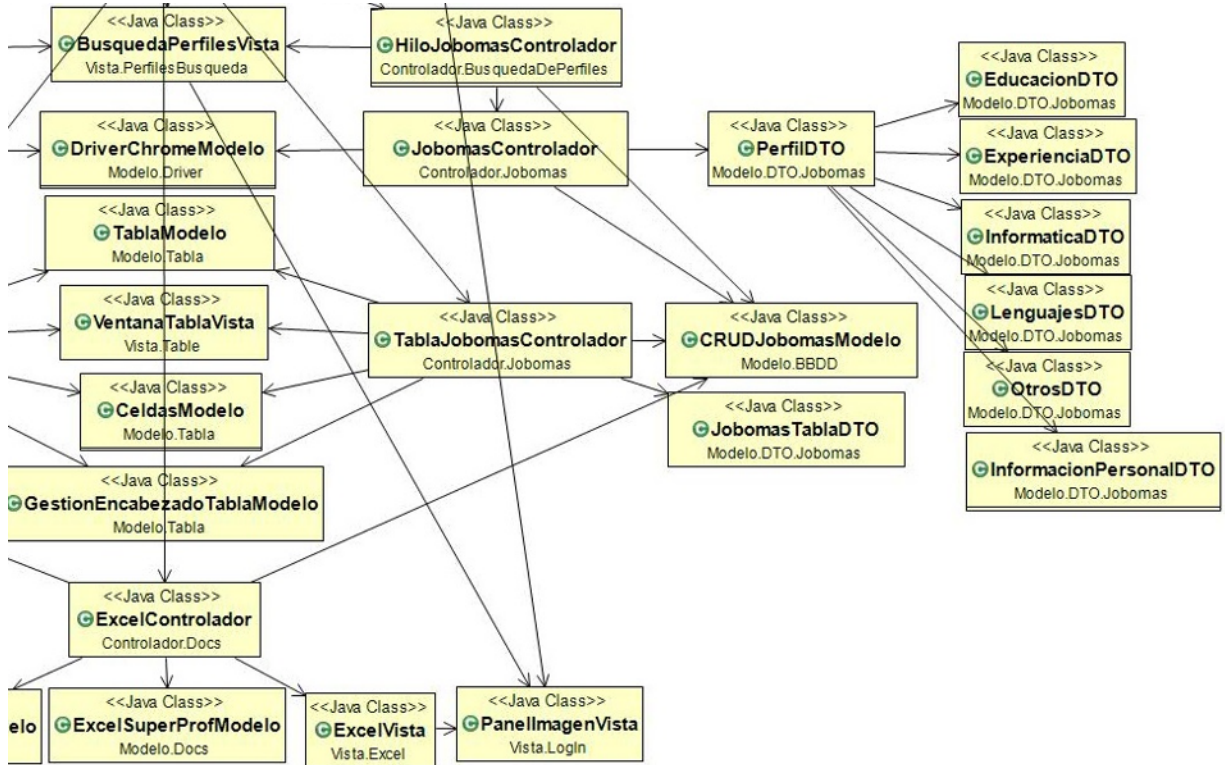


Figura 4.4: UML de los módulos de Jobomas.

El diagrama de flujo de la figura 4.5 expresa a grandes rasgos la lógica de programación en el desarrollo del presente trabajo de tesis. Podemos dar seguimiento a diferentes módulos programados como el correcto acceso a la aplicación, que al ejecutar el buscador se invocan a los dos Hilos programados (uno para cada sitio *web* consultado), algunos ciclos en toma de decisiones, la creación y el llamado a las instancias de JDBC *Driver* que se crean para acceder y manipular la base de datos, la invocación del ChromeDriver para navegar a través de la internet y con el uso de Selenium extraer la información de la *web*, la creación de las hojas de cálculo y el término de la ejecución del programa.

4. ARQUITECTURA DE LA APLICACIÓN

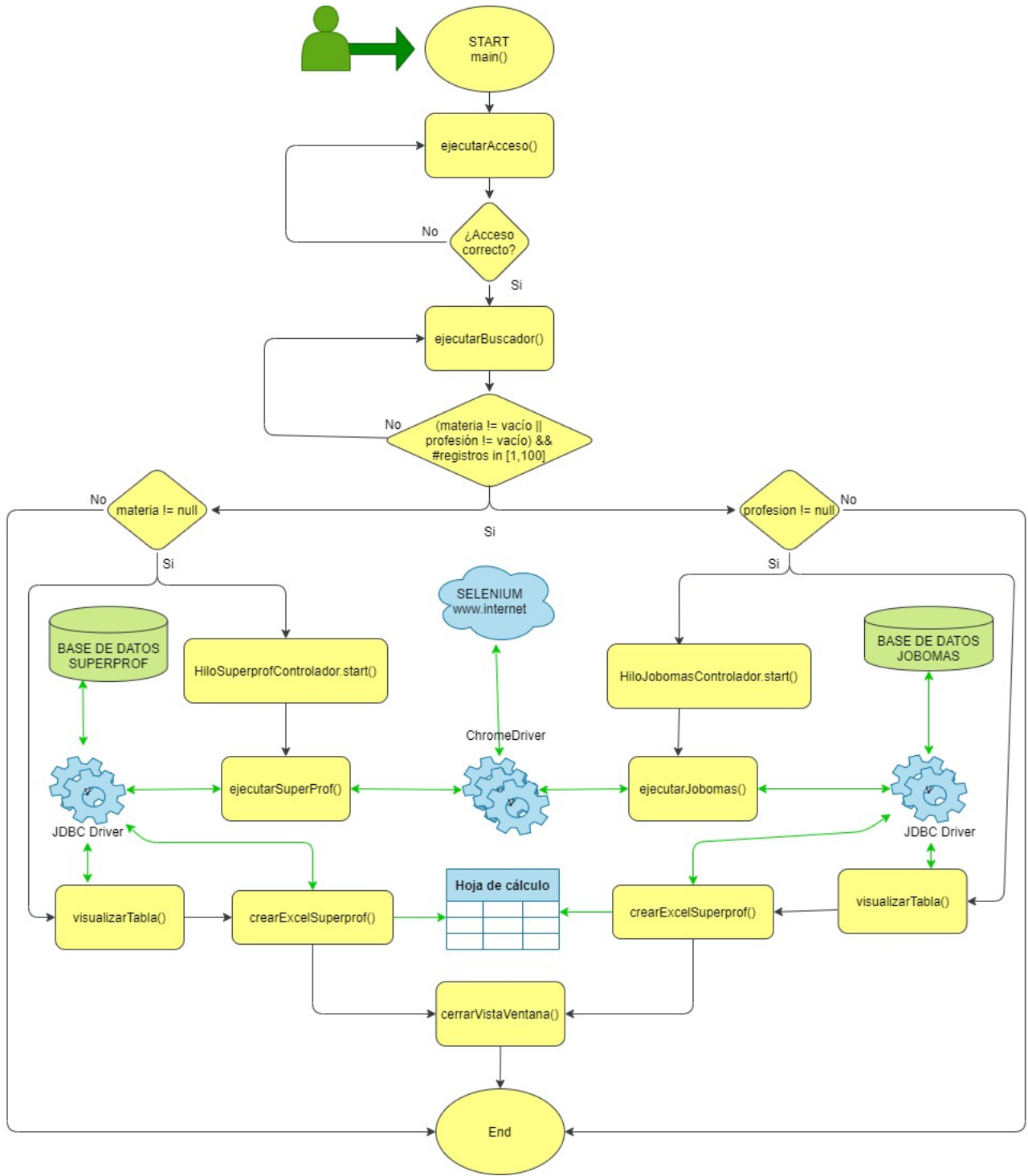


Figura 4.5: Diagrama de flujo de la arquitectura completa.

4.2 Diagrama Entidad Relación de la Base de Datos

Para poder agregar una nueva búsqueda en otro sitio *web*, es necesario que en la clase BuscadorControlador se lance un nuevo hilo de ejecución. Este nuevo hilo deberá contar con un controlador que se encargará de orquestar el ChromeDriver que instancia al navegador *web*, a los *DTOs* nuevos que van a almacenar y transferir la información del perfil y a las clases del modelo de base de datos que ejecutan las operaciones *CRUD* sobre las nuevas tablas. Se deberá agregar un controlador que tendrá la función de implementar la vista de la tabla que se muestra al usuario al final de la ejecución, este módulo creará los *DTOs* necesarios para el manejo de la información, a los modelos de la vista de la nueva ventana y a los modelos encargados de la lectura en las nuevas tablas de la base de datos. Por último, se agregará otro controlador dedicado a inicializar la hoja de cálculo con el resumen de la información obtenida, se emplearán clases *DTOs* que manejarán la transferencia de información, modelos de la hoja de cálculo y modelos que permitan la consulta a la base de datos.

4.2 Diagrama Entidad Relación de la Base de Datos

Se han generado varias tablas como producto del análisis de los sitios *web* Jobomas y Superprof, esto dado que cada página ofrece datos diferentes. Los modelos Entidad-Relación que resultan se enumeran de la siguiente manera:

- Tablas de sitio *web* Jobomas: La relación es uno a muchos, es decir, un perfil de Jobomas puede manejar diferentes idiomas, tecnologías de informática, títulos en escuelas, empleos en más de una empresa u otras habilidades. Ver figura 5.10.
- Tablas de sitio *web* Superprof: La relación también es uno a muchos, es decir, un profesor de Superprof puede manejar diferentes clases, lugares de clases y niveles de las mismas. Ver figura 5.11.
- Tabla Credenciales: Esta tabla a pesar de estar sola es igual de importante que las anteriores, aquí es donde van almacenados los usuarios y sus contraseñas para poder acceder a la aplicación. Ver figura 5.12.

Los diagramas 5.10, 5.11 y 5.12 se han modelado con el Gestor de Base de Datos de *SQLDeveloper* de Oracle y se generan de forma automática, nos permite ver el nombre de la tabla, los nombres, tipo y longitud de los atributos de la misma, su llave primaria y llave secundaria, entre otras cosas. Se detalla al campo ID como la llave primaria de la tabla padre y como se distribuye como llave secundaria en las otras tablas hijas, estos valores son de tipo Number dado que son auto incrementales

4. ARQUITECTURA DE LA APLICACIÓN

y los demás campos se dejaron abiertos a cadenas, por lo que son de tipo Varchar2 y solo varían en su tamaño.

Si se desea agregar un nuevo sitio *web* para la búsqueda de información, se deberán crear nuevas tablas acorde a los datos que ofrece la nueva página.

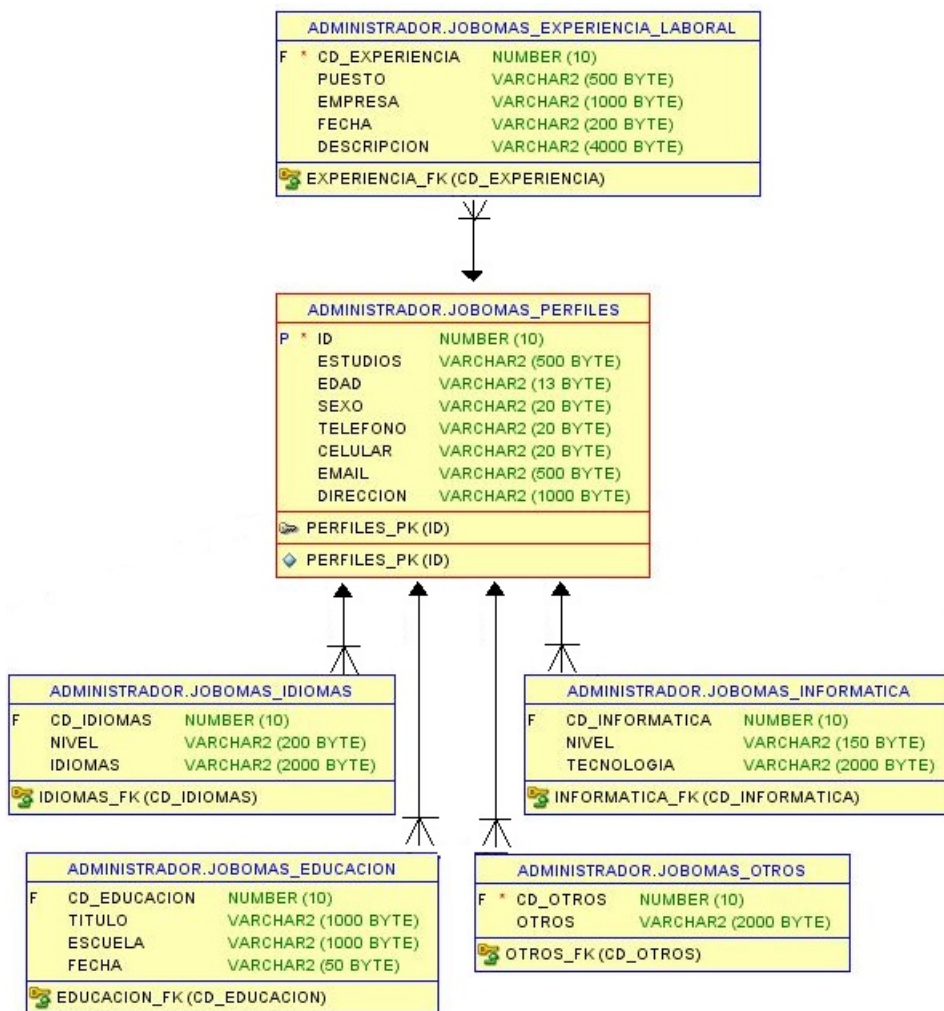


Figura 4.6: Modelo Entidad-Relación de tablas Jobomas en Base de Datos.

4.2 Diagrama Entidad Relación de la Base de Datos

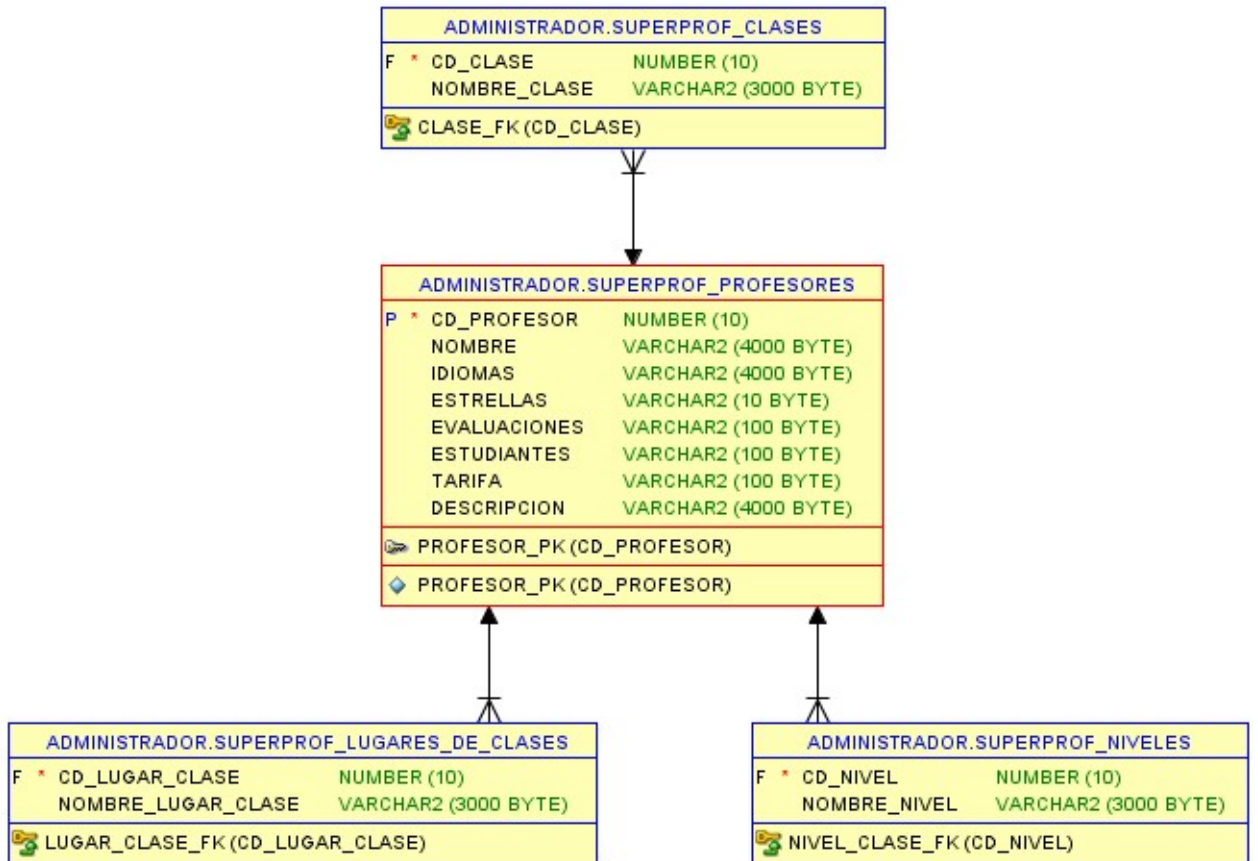


Figura 4.7: Modelo Entidad-Relación de tablas Superprof en Base de Datos.

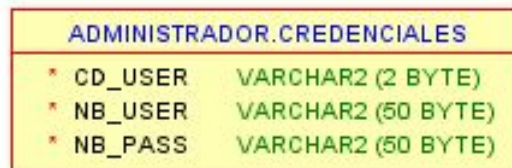


Figura 4.8: Modelo Entidad-Relación de tabla Credenciales en Base de Datos.

4. ARQUITECTURA DE LA APLICACIÓN

4.3 Arquitectura Lógica MVC

Como patrón de arquitectura de software se eligió al Modelo Vista Controlador(MVC) ya que permite programar los módulos por separado y esto nos ayuda a realizar test de funcionalidad independiente permitiendo identificar fallos de programación de manera más puntual, la codificación es ordenada y la lectura del código es transparente, además de que permitirá un evolutivo en tecnologías empleadas en las diferentes capas a futuro. A continuación se mostrará el empaquetado de las diferentes clases que conforman a los diferentes controladores, vistas y modelos. Las imágenes que representan a las clases en *UML* muestran el nombre de la clase, el nombre y tipo de sus atributos, los constructores que llegara a tener, el tipo de retorno y el nombre de los métodos que tiene, así como el paquete al que pertenece la clase. Estas representaciones *UML* y la distribución de los paquetes y clases se han generado con el propio *IDE NeatBeans*, por ello no han sido alteradas.

4.3.1 Controlador

Se crearon 5 paquetes para los controladores, como se muestra y siguiendo el orden de la figura 4.9 tenemos las siguientes clases:

- Controlador.BusquedaDePerfiles:
 - BuscadorControlador: Ver figura 4.10, Controlador orquestador del flujo del programa, contiene la lógica principal a los llamados de los principales controladores como lo son la ejecución de los Hilos, la instanciación de las tablas y la creación de las hojas de cálculo.
 - HiloJobomasControlador: Ver figura 4.11, Encargado de ejecutar el Hilo dedicado al sitio *web* Jobomas.
 - HiloSuperprofControlador: Ver figura 4.12, Encargado de ejecutar el Hilo dedicado al sitio *web* Superprof.
- Controlador.Docs:
 - ExcelControlador: Ver figura 4.13, se encarga de generar las hojas de cálculo con la información recuperada de uno o de ambos sitios *web*.
- Controlador.Jobomas:
 - JobomasControlador: Ver figura 4.14 y 4.15, accede al sitio *web* Jobomas, recupera los datos deseados, emplea los modelos para almacenar la información en base de datos.

4.3 Arquitectura Lógica MVC

- TablaJobomasControlador: Ver figura 4.16 , recupera el modelo de la vista con los parámetros del sitio *web* Jobomas, accede a base de datos, obtiene los valores para llenar el modelo de la vista y mostrar la información al usuario.
- Controlador.LogIn:
 - AccesoControlador 4.17, valida los datos de entrada(usuario y contraseña) con las credenciales autorizadas en base de datos, con ello permite o no el acceso a la aplicación programada.
- Controlador.Superprof:
 - SuperprofControlador: Ver figura 4.18, accede al sitio *web* Superprof, recupera los datos deseados, emplea los modelos para almacenar la información en base de datos.
 - TablaJobomasControlador: Ver figura 4.19, recupera el modelo de la vista con los parámetros del sitio *web* Superprof, accede a base de datos, obtiene los valores para llenar el modelo de la vista y mostrar la información al usuario.

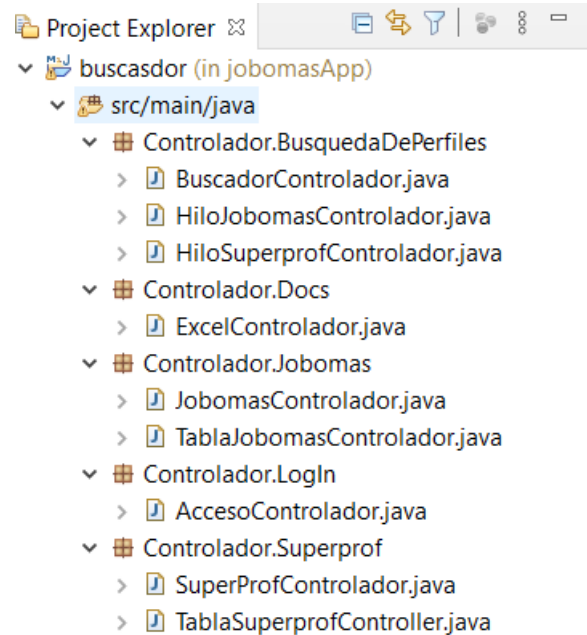


Figura 4.9: Distribución de los controladores dentro del proyecto java.

4. ARQUITECTURA DE LA APLICACIÓN

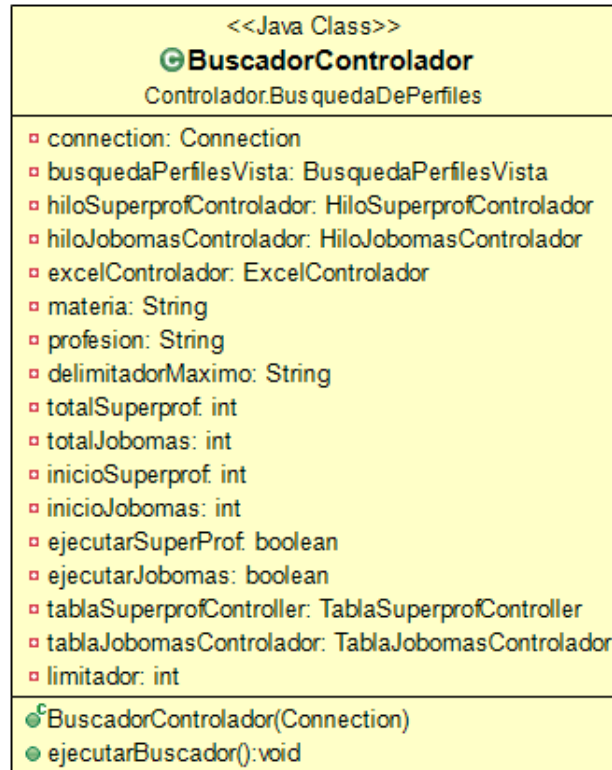


Figura 4.10: Controlador que ejecuta la búsqueda de perfiles en diferentes bolsas de empleo virtuales.

4.3 Arquitectura Lógica MVC

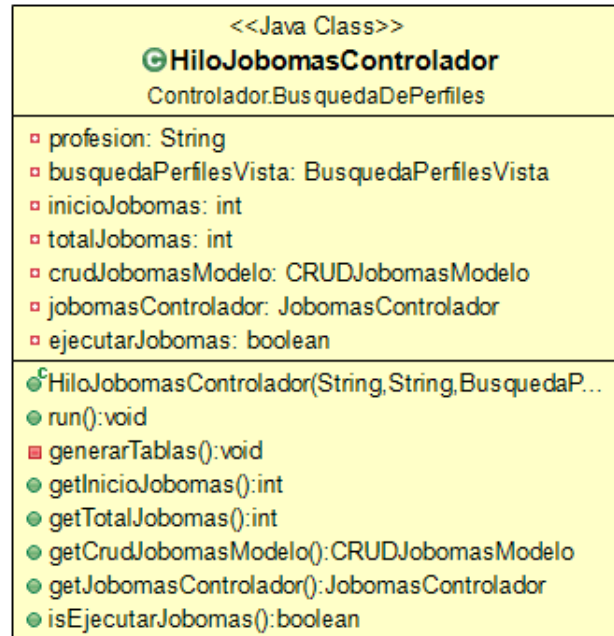


Figura 4.11: Controlador que ejecuta el hilo designado a Jobomas.

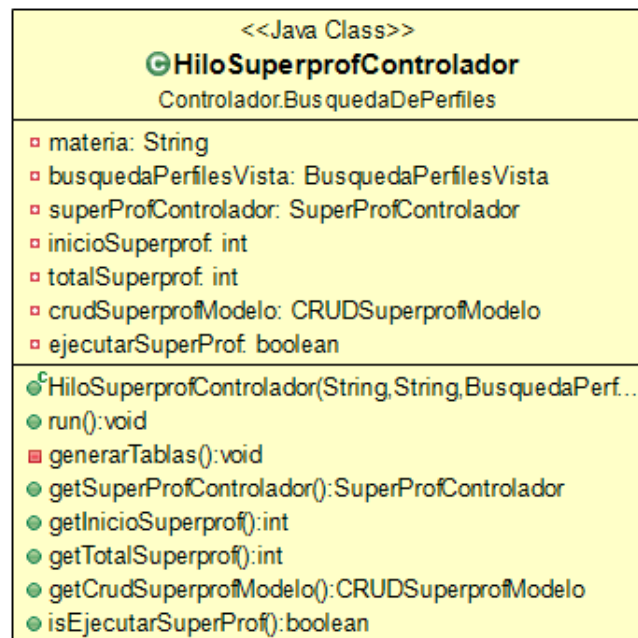


Figura 4.12: Controlador que ejecuta el hilo designado a Superprof.

4. ARQUITECTURA DE LA APLICACIÓN

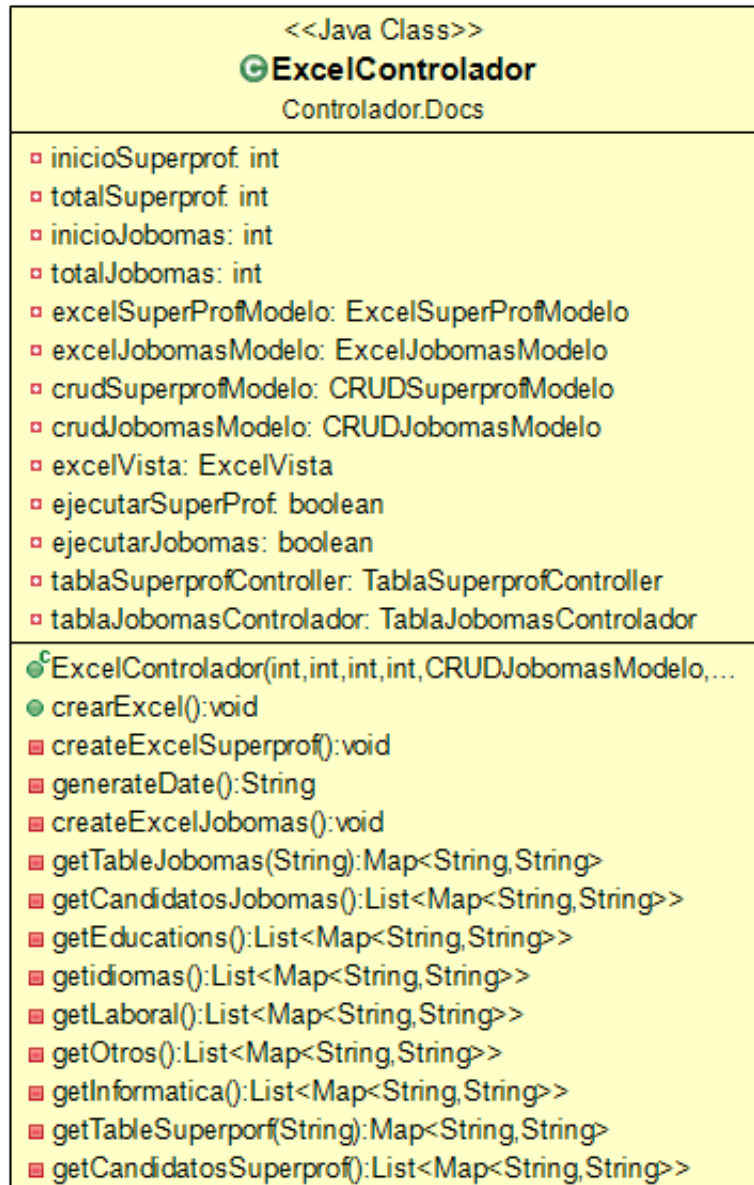


Figura 4.13: Controlador que genera las hojas de cálculo con los registros obtenidos en el proceso de obtención de datos empleando la técnica de *web scraping*.

4.3 Arquitectura Lógica MVC

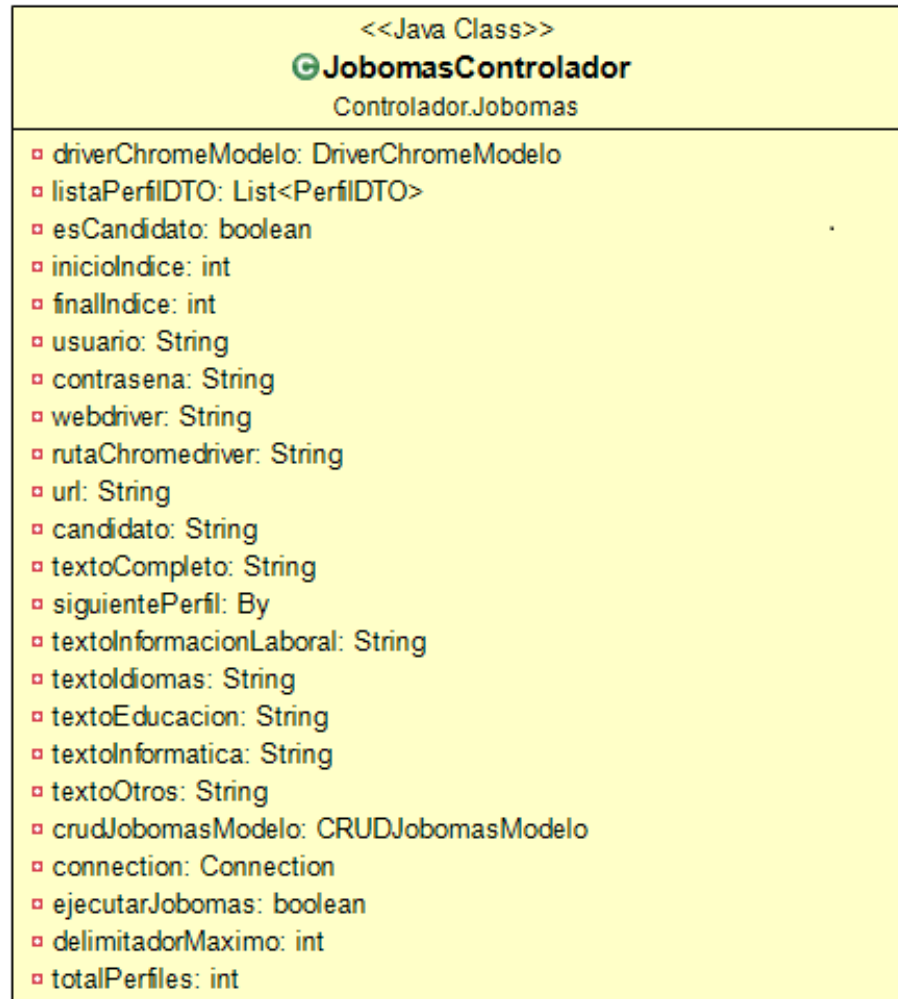


Figura 4.14: Controlador que ejecuta la técnica de *web scraping* en Jobomas, se muestran las variables.

4. ARQUITECTURA DE LA APLICACIÓN

```
● JobomasControlador(Connection,String,int)
● ejecutarJobomas():void
■ acceso():boolean
■ accesoFallido():boolean
■ buscarCandidatos():void
■ palabraClave():void
■ sinResultados():boolean
■ obtenerPerfiles():int
■ obtenerInformacionPerfiles():void
■ guardarTexto(String):void
■ verPrimerPerfilclick():void
■ obtenerTextoPerfil():String
■ verPrimerPerfil():PerfilDTO
■ existeSiguientePerfil(String):boolean
■ verPerfiles():void
■ verSiguientePerfil():void
■ obtenerSiguientePerfilDTO():PerfilDTO
■ obtenerPerfilDTO():PerfilDTO
■ obtenerListaInformacionPersonalDTO():List<InformacionPersonalDTO>
■ insertarBBDD(List<PerfilDTO>,int):void
■ insertarPerfil(PerfilDTO,String):void
■ insertarPerfiles(InformacionPersonalDTO,String):void
■ obtenerListaExperienciaDTO():List<ExperienciaDTO>
■ obtenerInformacionLaboral():String
■ obtenerListaEducacionDTO():List<EducacionDTO>
■ obtenerInformacionEducacion():String
■ obtenerListaLenguajesDTO():List<LenguajesDTO>
■ obtenerInformacionIdiomas():String
■ obtenerListaInformaticaDTO():List<InformaticaDTO>
■ obtenerInformacionInformatica():String
■ obtenerListaOtrosDTO():List<OtrosDTO>
■ obtenerInformacionOtros():String
● esCandidato():boolean
● getInicioIndice():int
● getFinalIndice():int
● getCrudJobomasModelo():CRUDJobomasModelo
● getEjecutarJobomas():boolean
```

Figura 4.15: Controlador que ejecuta la técnica de *web scraping* en Jobomas, se muestran los métodos.

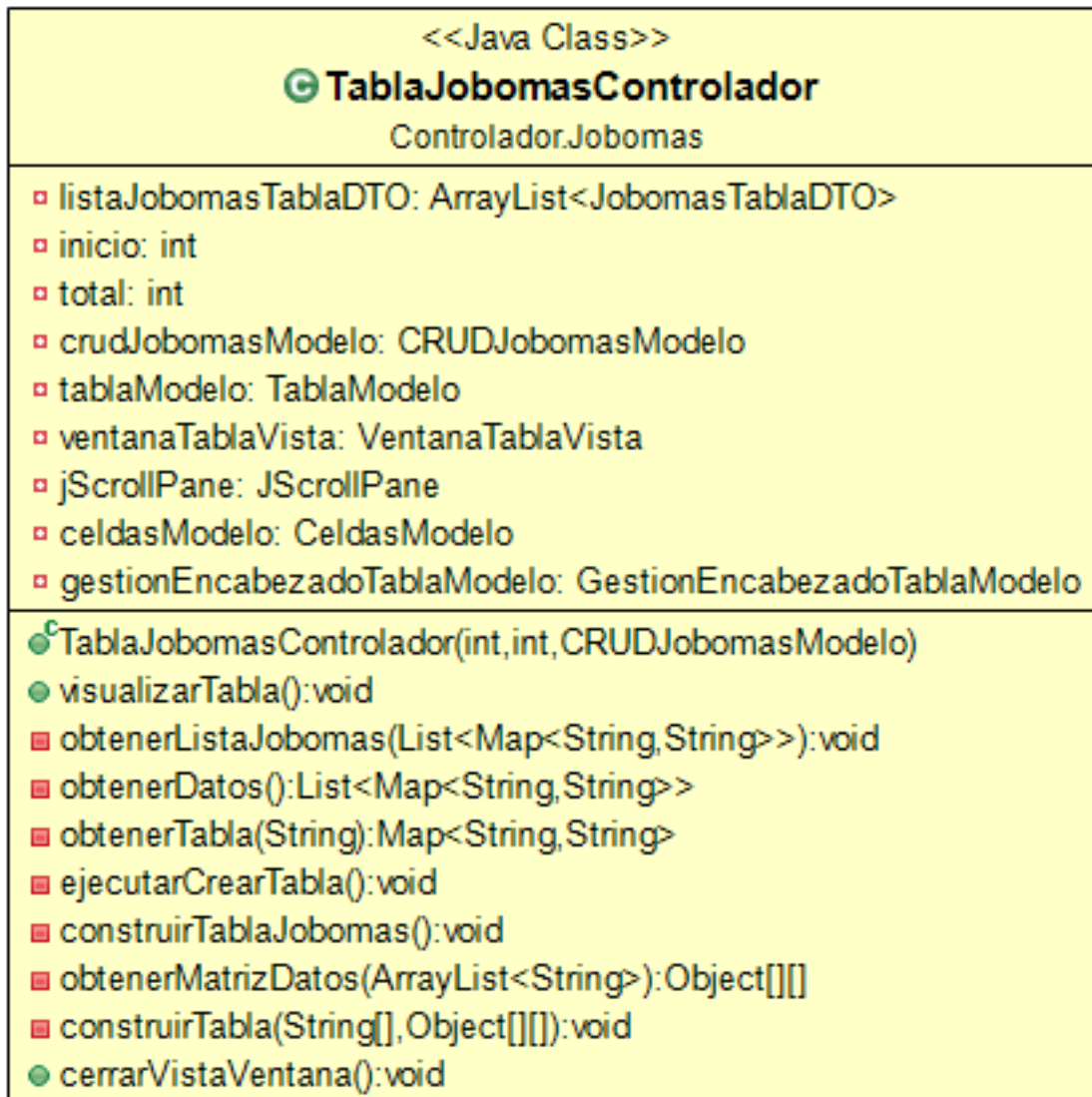


Figura 4.16: Controlador que ejecuta la creación de la tabla en pantalla con el resumen de resultados de Jobomas.

4. ARQUITECTURA DE LA APLICACIÓN

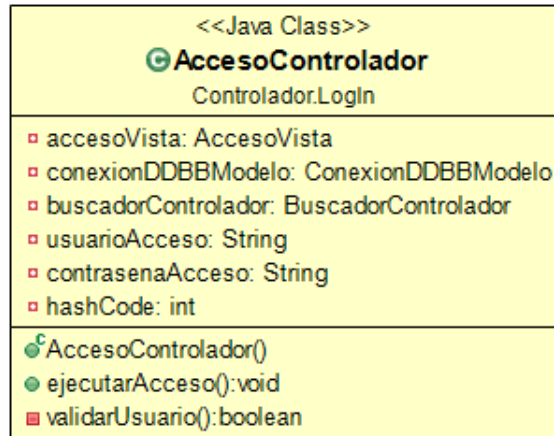


Figura 4.17: Controlador que valida las credenciales de acceso a la aplicación.

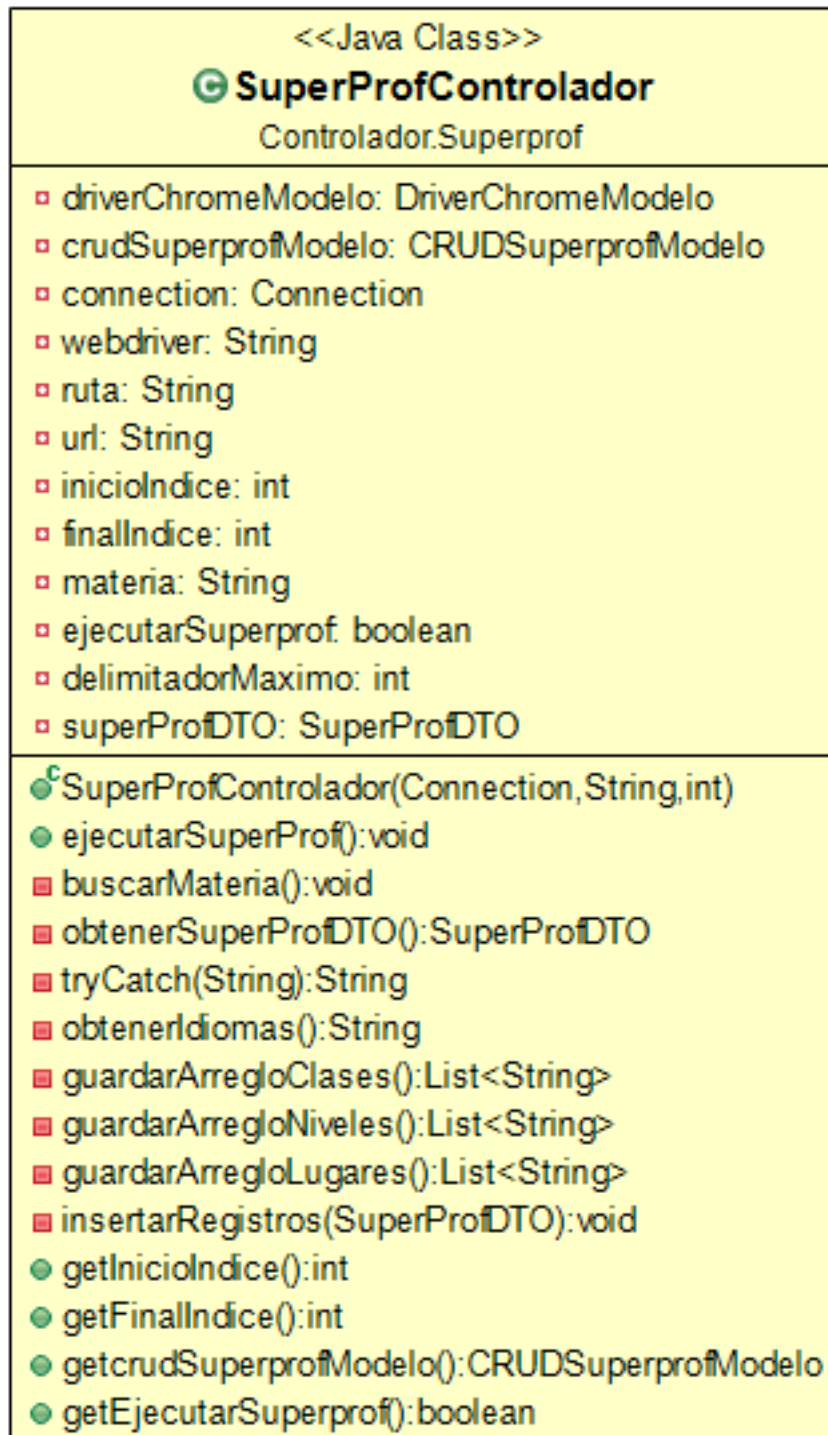


Figura 4.18: Controlador que ejecuta la técnica de *web scraping* en Superprof.

4. ARQUITECTURA DE LA APLICACIÓN

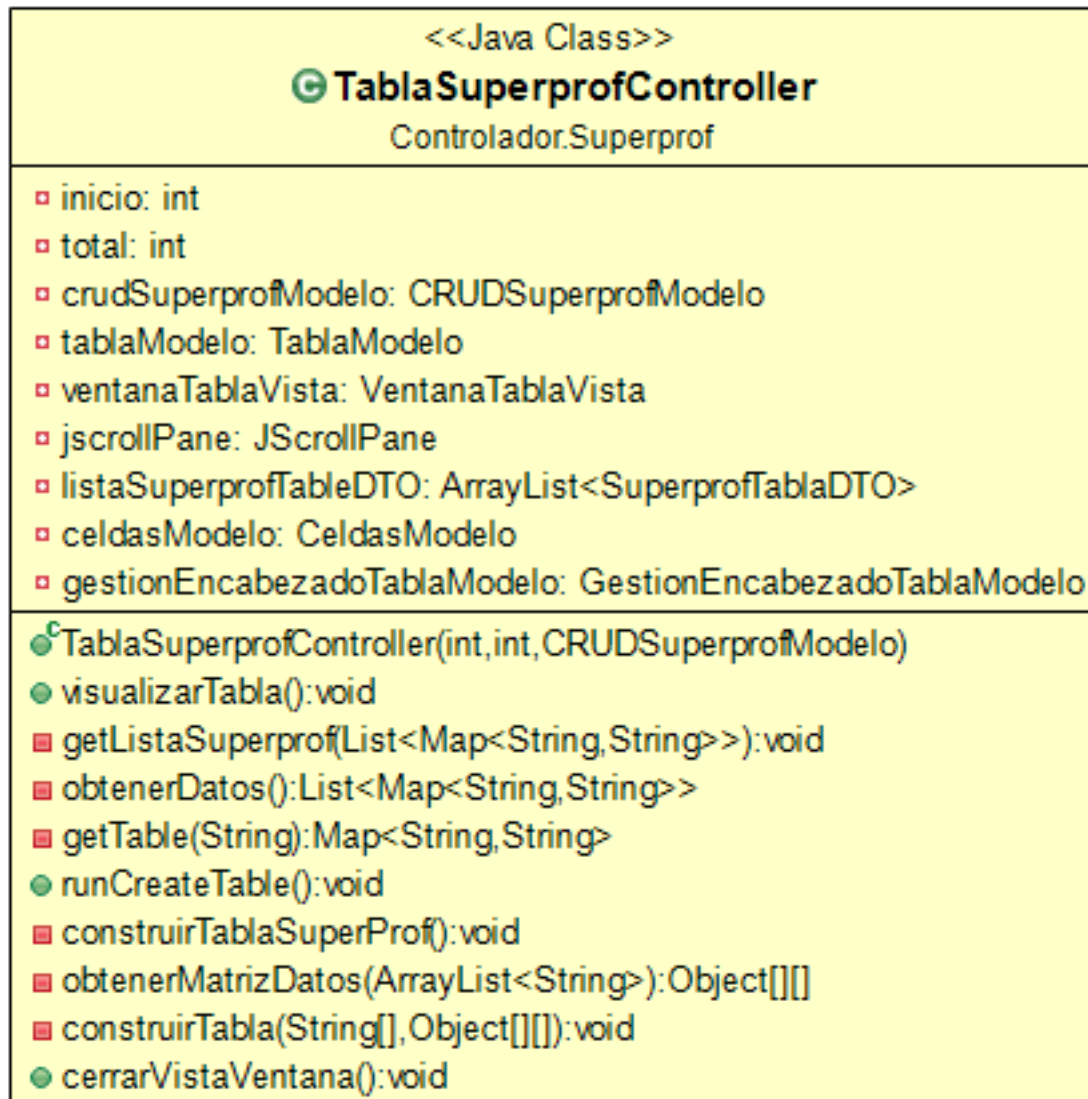


Figura 4.19: Controlador que ejecuta la creación de la tabla en pantalla con el resumen de resultados de Superprof.

4.3.2 Vista

Se crearon 4 paquetes para las Vistas, como se muestra y siguiendo el orden de la figura 4.20 tenemos las siguientes clases:

- Vista.Excel:

4.3 Arquitectura Lógica MVC

- ExcelVista: Ver figura 4.21, última vista, permite al usuario crear la hoja de cálculo o terminar la ejecución de la aplicación.
- Vista.Login:
 - AccesoVista: Ver figura 4.22, primera vista, es la que pide al usuario las credenciales de acceso a la aplicación.
 - PanellImageVista: Ver figura 4.23, plantilla que sirve a todas las vistas como base, contiene un constructor que recibe diversos parámetros de acuerdo a quien la invoca.
 - Vista.PerfilesBusqueda:
 - BusquedaPerfilesVista: Ver figura 4.24, ventana que pide al usuario ingresar que perfiles desea buscar y el número de los mismos.
 - Vista.Table:
 - VentanaTabalaVista: Ver figura 4.25, pantalla que muestra una o dos tablas con el resumen de los datos obtenidos en la búsqueda de perfiles en internet empleando la técnica de *web scraping*.



Figura 4.20: Distribución de las vistas dentro del proyecto java.

4. ARQUITECTURA DE LA APLICACIÓN

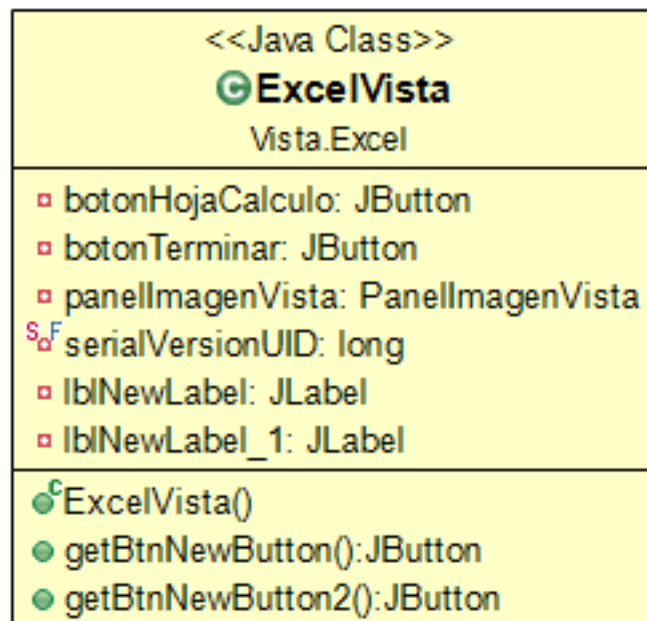


Figura 4.21: Ventana que muestra la opción de generar los archivos de hojas de cálculo Jobomas y Superprof con la información obtenida en el proceso de obtención de datos empleando la técnica de *web scraping* y la opción de terminar la aplicación.

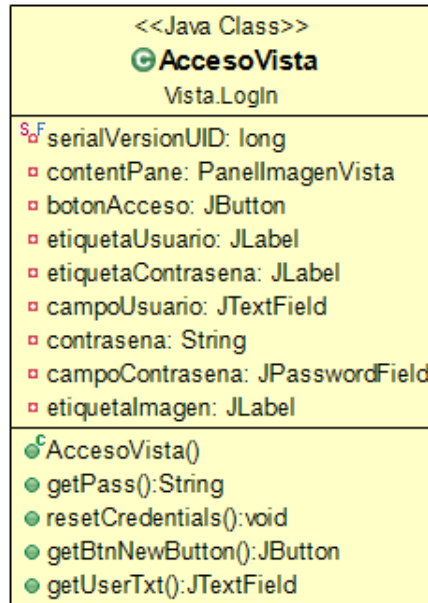


Figura 4.22: Ventana que solicita las credenciales de acceso a la aplicación.

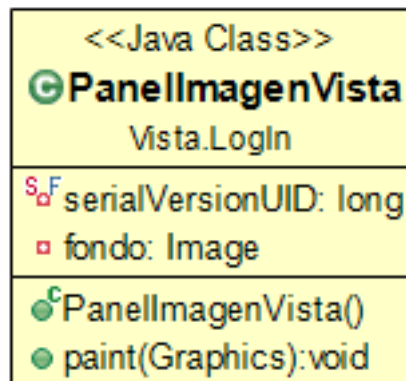


Figura 4.23: Métodos de la vista que sobrescribe los la clase JPanel para insertar imágenes en las ventanas.

4. ARQUITECTURA DE LA APLICACIÓN

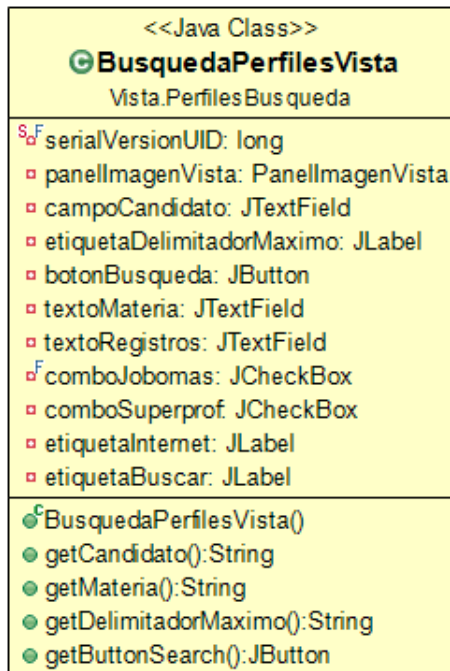


Figura 4.24: Ventana que solicita la los perfiles que se desean buscar.

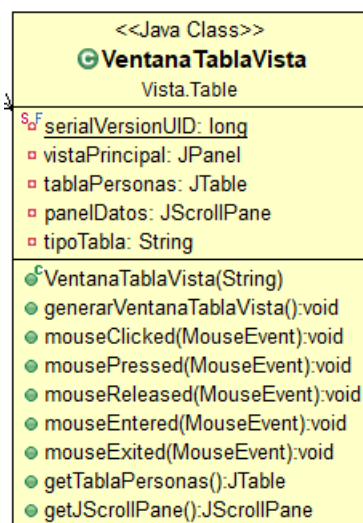


Figura 4.25: Ventana que muestra un resumen de la información obtenida en el proceso de obtención de datos empleando la técnica de *web scraping* en las páginas de Jobomas y Superprof.

4.3.3 Modelo

Se crearon 6 paquetes para los Modelos, como se muestra y siguiendo el orden de la figura 4.26 tenemos las siguientes clases:

- Modelo.BBDD:
 - ConexionDDBBModelo: Ver figura 4.27, modelo que contiene los parámetros para llevar a cabo la conexión a la base de datos Oracle.
 - CRUDJobomasModelo: Ver figura 4.28, contiene las instrucciones de inserción y obtención de información de las tablas dedicadas a almacenar datos de Jobomas en base de datos.
 - CRUDSuperprofModelo: Ver figura 4.29, contiene las instrucciones de inserción y obtención de información de las tablas dedicadas a almacenar datos de Superprof en base de datos.
- Modelo.Docs:
 - ExcelJobomasModelo: Ver figura 4.30, este modelo contiene los parámetros que permiten la creación de la hoja de cálculo con la información obtenida del sitio *web* Jobomas.
 - ExcelSuperprofModelo: Ver figura 4.31, este modelo contiene los parámetros que permiten la creación de la hoja de cálculo con la información obtenida del sitio *web* Superprof.
- Modelo.Driver:
 - DriverChromeModelo: Ver figura 4.32, contiene el constructor y los atributos necesarios instanciar el *Web-Driver* Chrome, este nos permitirá emplear el navegador de internet.
- Modelo.DTO.Jobomas: Son objetos de datos que sirven para almacenar datos del candidato de Jobomas.
 - EducacionDTO: Ver figura 4.33, maneja la institución de educación.
 - ExperienciaDTO: Ver figura 4.34, contiene la experiencia laboral.
 - InformacionPersonalDTO: Ver figura 4.35, almacena la información personal.
 - InformaticaDTO: Ver figura 4.36, distribuye entre las capas los conocimientos de informática.

4. ARQUITECTURA DE LA APLICACIÓN

- JobomasTablaDTO: Ver figura 4.37, contiene la información para llenar la tabla de la vista con el resumen de la información de un candidato.
 - LenguajesDTO: Ver figura 4.38, guarda el lenguaje que maneja.
 - OtrosDTO: Ver figura 4.39, hace referencia a datos extras que el candidato considera resaltar y no tuvo oportunidad de colocarlos en algún espacio dedicado.
 - PerfilDTO: Ver figura 4.40, objeto de datos principal, contiene una lista de cada uno de los *DTOs* anteriores, esto dado que puede tener uno o más de ellos, por ejemplo puede manejar uno o varios idiomas y el *DTO* dedicado a los idiomas solo almacena un idioma por instancia.
- Modelo.DTO.Superprof.SuperprofDTO: Son objetos de datos que sirven para almacenar datos del profesor de Superprof.
 - SuperProfDTO: Ver figura 4.41, contiene todos los parámetros recuperados del sitio *web* para un profesor.
 - SuperprofTablaDTO: Ver figura 4.42, contiene la información para llenar la tabla de la vista con el resumen de la información de un profesor.
- Modelo.Tabla:
 - CeldasModelo: Ver figura 4.43, modelo con los parámetros que sobrescriben las celdas de las tablas que maneja la clase `DefaultTableCellRenderer` de java swing, con esto se personalizan los parámetros que vienen por default al crear las celdas.
 - GestionEncabezadoTablaModelo: Ver figura 4.44, este modelo almacena los parámetros que sobrescriben los encabezados de las tablas que maneja la clase `TableCellRenderer` de java swing, con esto se personalizan los parámetros que vienen por default al crear los encabezados de las tablas mostradas al final de la ejecución del programa.
 - TablaModelo: Ver figura 4.45, modelo con los atributos de creación de la tabla de resumen de los registro del sitio *web* Jobomas y Superprof empleando la técnica de *web* scraping.

4.3 Arquitectura Lógica MVC



Figura 4.26: Distribución de los modelos dentro del proyecto java.

4. ARQUITECTURA DE LA APLICACIÓN

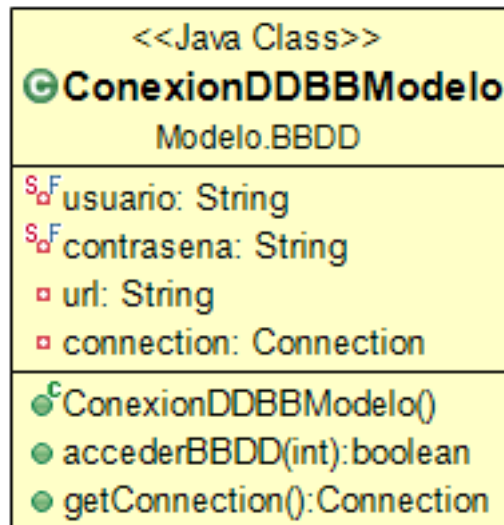


Figura 4.27: Modelo con los parámetros para realizar una conexión a la base de datos.

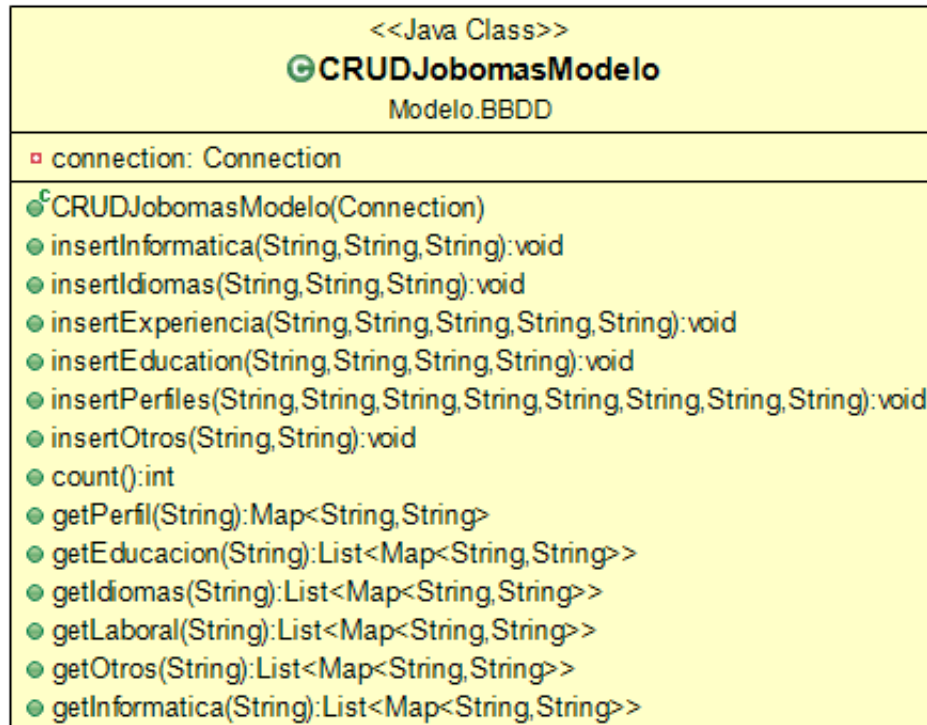


Figura 4.28: Modelo con las operaciones de lectura y escritura a la base de datos de Jobomas.

4. ARQUITECTURA DE LA APLICACIÓN

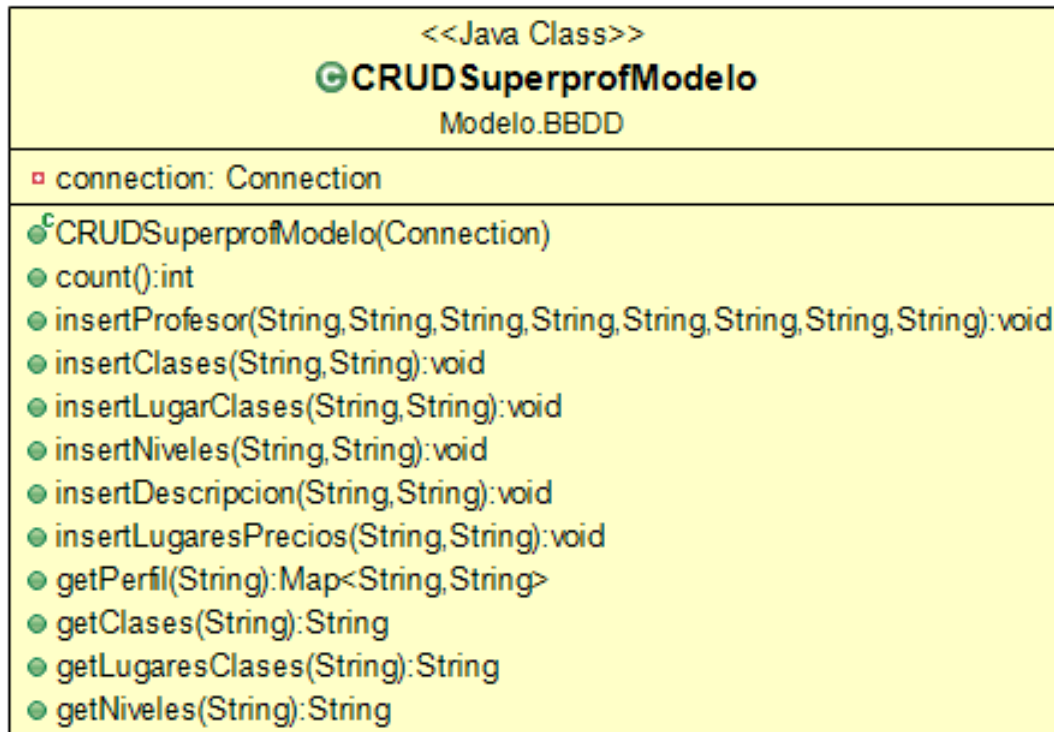


Figura 4.29: Modelo con las operaciones de lectura y escritura a la base de datos de Superprof.

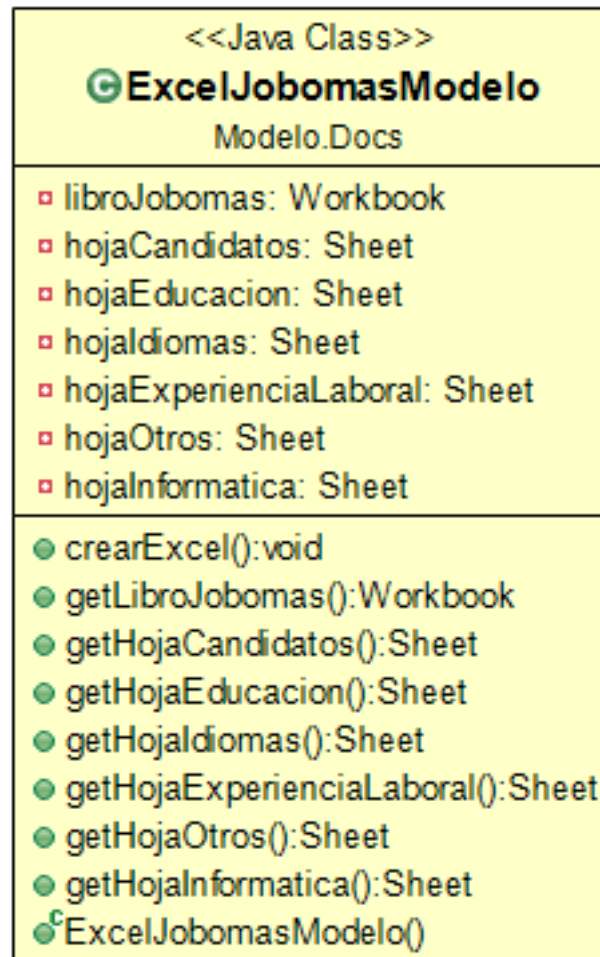


Figura 4.30: Modelo con los parámetros que crean la hoja de cálculo con el resumen de los registros obtenidos del sitio *web* Superprof empleando la técnica de *web scraping*.

4. ARQUITECTURA DE LA APLICACIÓN

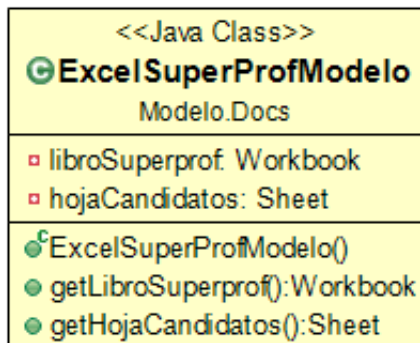


Figura 4.31: Modelo con los parámetros que crean la hoja de cálculo con el resumen de los registro obtenidos del sitio *web* Jobomas empleando la técnica de *web scraping*.

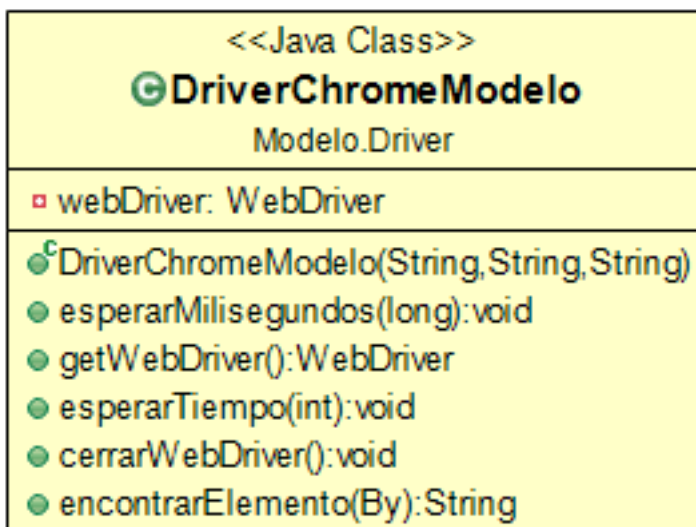


Figura 4.32: Modelo que contiene los parámetros para crear una instancia del *web-Driver Chrome*, este nos permitirá emplear el navegador de internet.

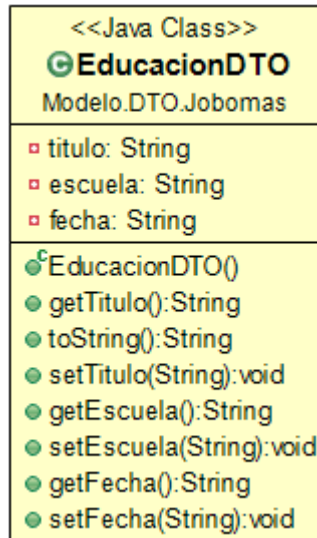


Figura 4.33: Modelo con los parámetros para almacenar la institución de educación de un candidato de Jobomas.

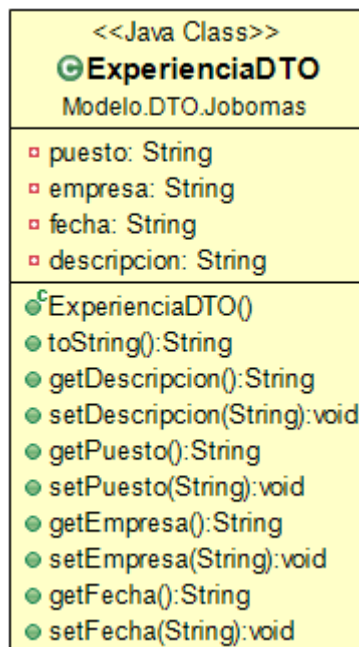


Figura 4.34: con los parámetros para almacenar la experiencia laboral de un candidato de Jobomas.

4. ARQUITECTURA DE LA APLICACIÓN

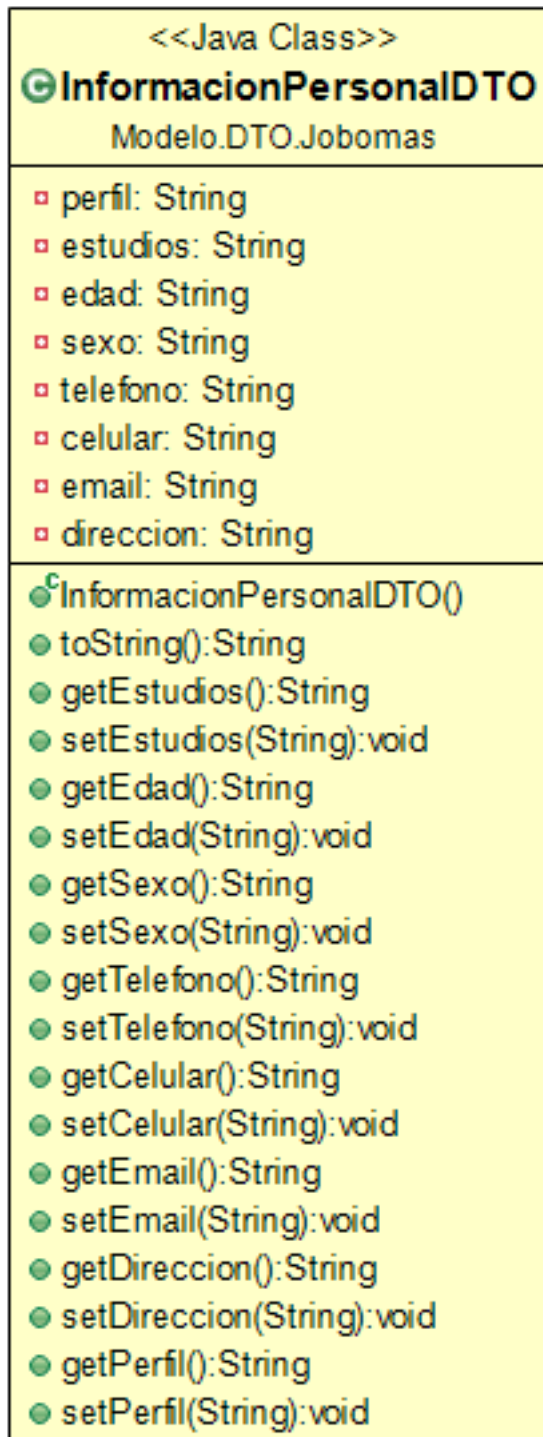


Figura 4.35: Modelo con los parámetros para almacenar datos personales de un candidato de Jobomas.

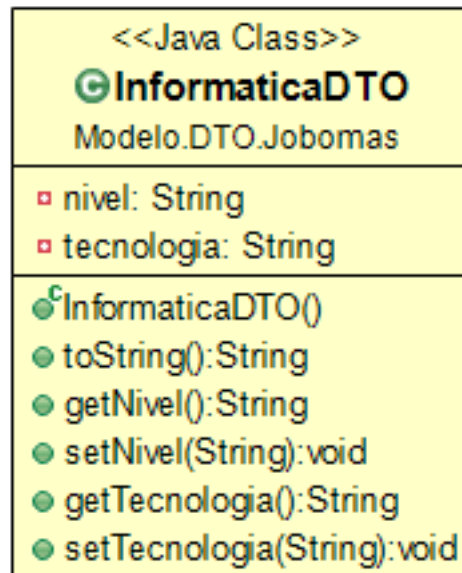


Figura 4.36: Modelo con los parámetros para almacenar los conocimientos en informática de un candidato de Jobomas.

4. ARQUITECTURA DE LA APLICACIÓN

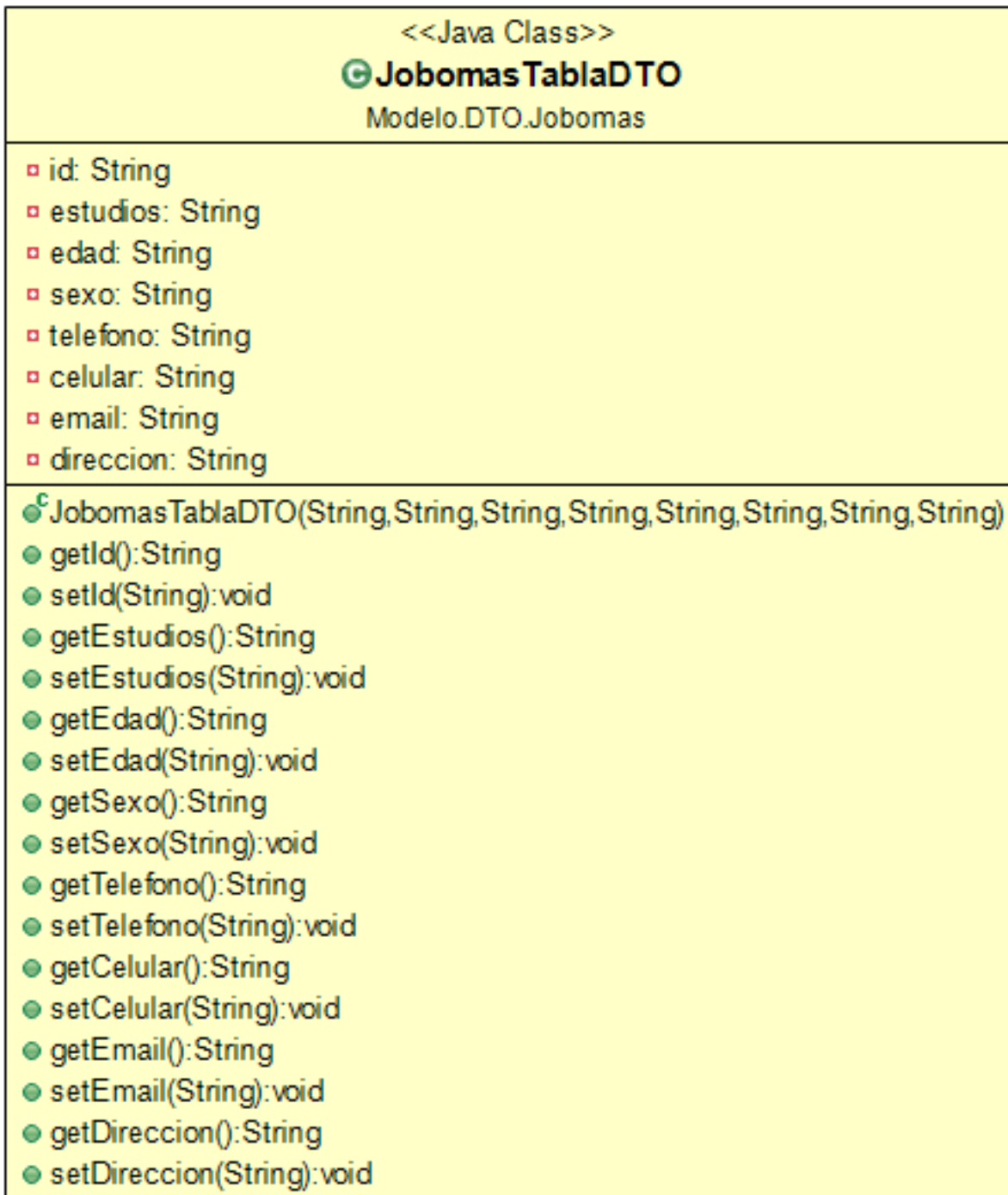


Figura 4.37: Modelo con los parámetros para almacenar el resumen de un registro de la base de datos de Jobomas.

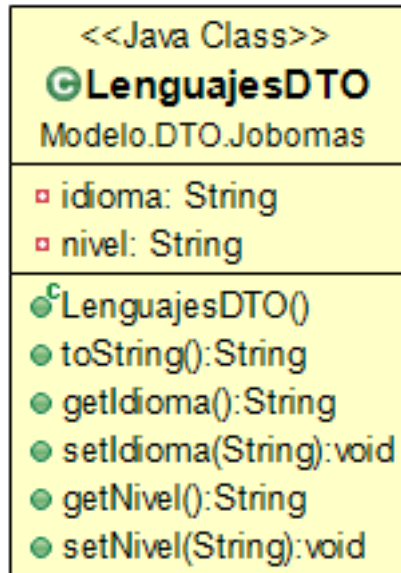


Figura 4.38: Modelo con los parámetros para almacenar los idiomas que maneja un candidato de Jobomas.

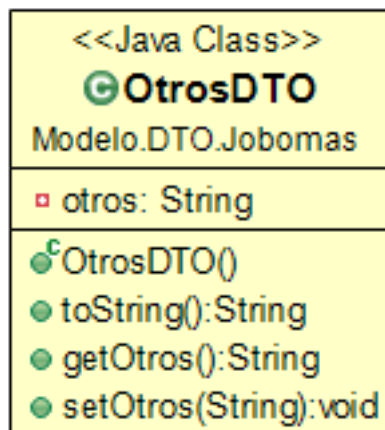


Figura 4.39: Modelo con los parámetros para almacenar otros conocimientos o habilidades de un candidato de Jobomas.

4. ARQUITECTURA DE LA APLICACIÓN

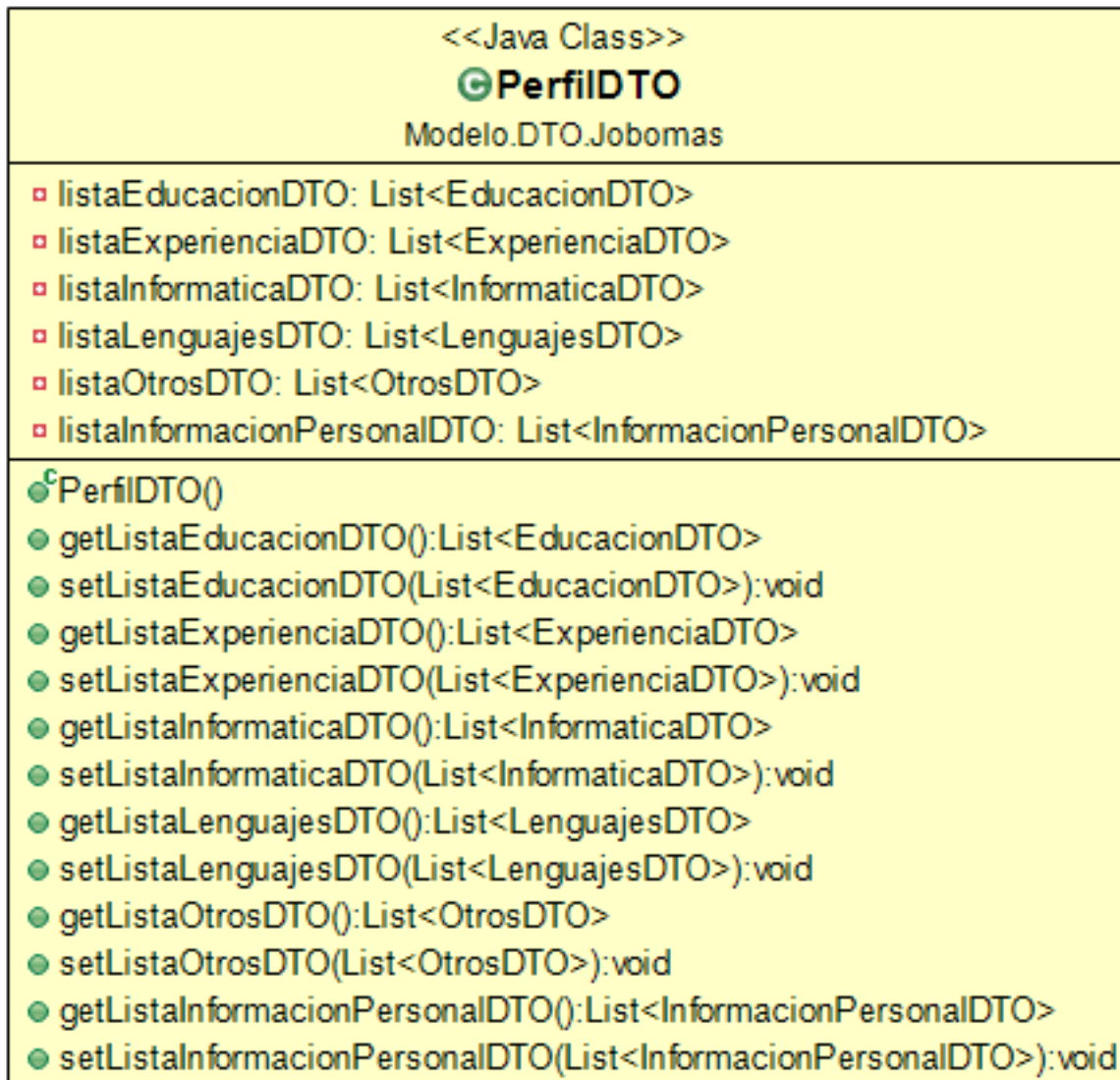


Figura 4.40: Modelo con los parámetros para almacenar un perfil de un candidato de Jobomas.

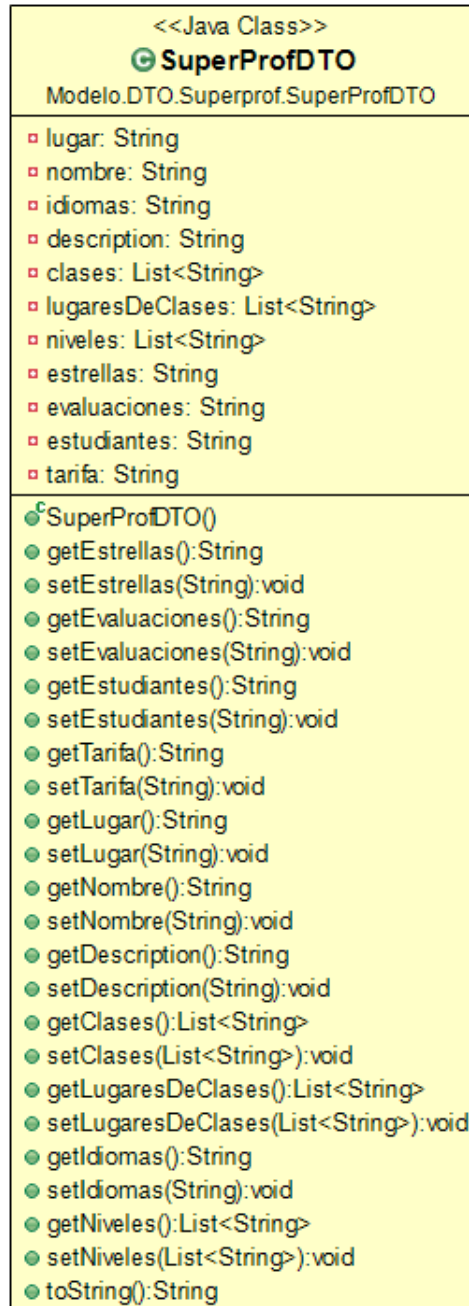


Figura 4.41: Modelo que contiene todos los parámetros recuperados del sitio *web* para un profesor.

4. ARQUITECTURA DE LA APLICACIÓN

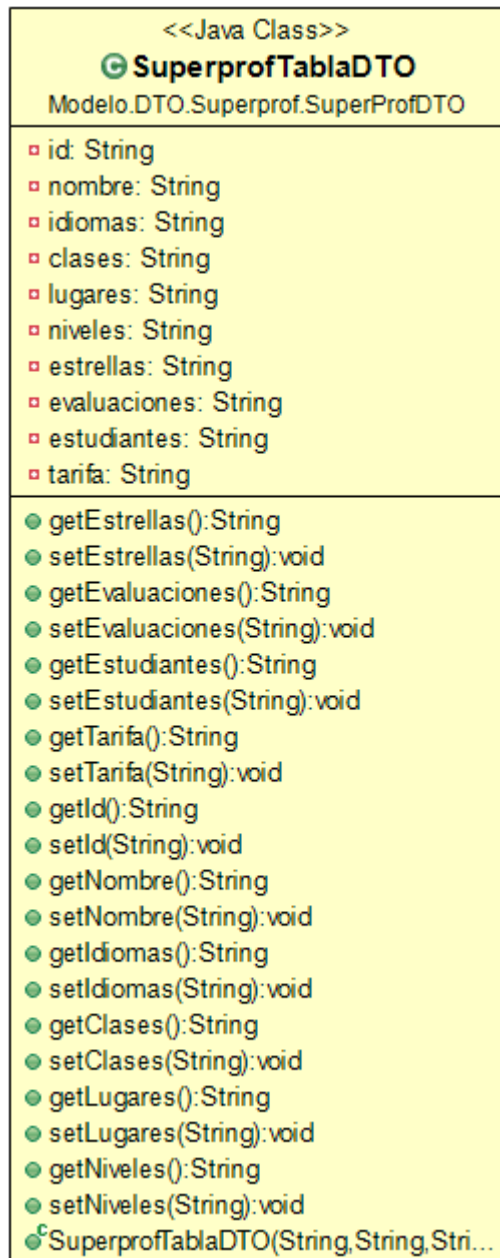


Figura 4.42: Modelo con los parámetros para almacenar el resumen de un registro de la base de datos de Superprof.

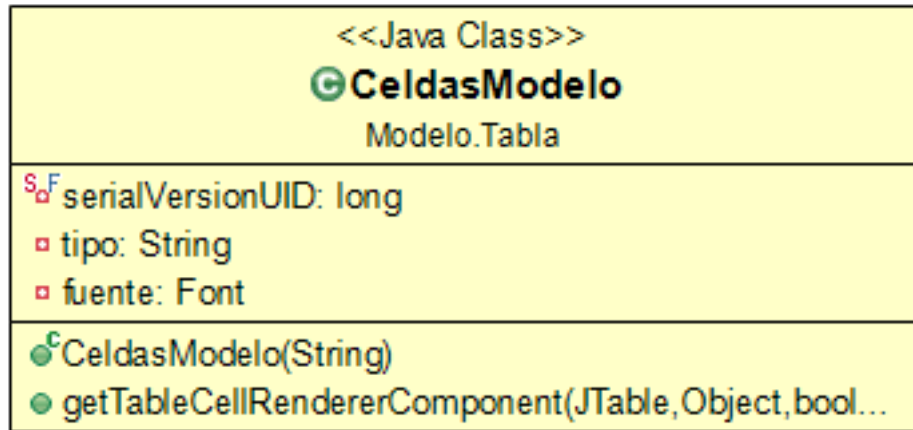


Figura 4.43: Modelo con los parámetros que sobrescriben las celdas de las tablas que maneja la clase DefaultTableCellRenderer de java swing.

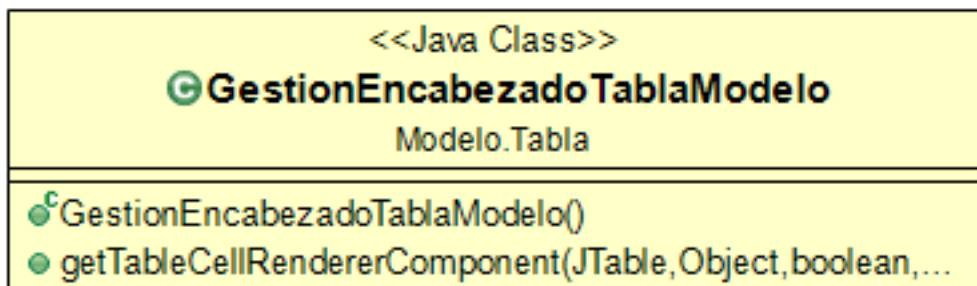


Figura 4.44: Modelo con los parámetros que sobrescriben los encabezados de las tablas que maneja la clase TableCellRenderer de java swing.

4. ARQUITECTURA DE LA APLICACIÓN

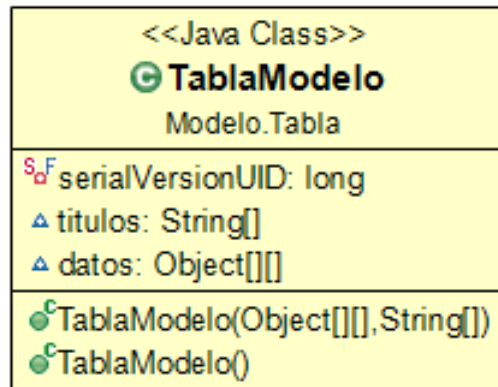


Figura 4.45: Modelo con los parámetros de creación de la tabla de resumen de los registro del sitio *web* Jobomas empleando la técnica de *web scraping*.

Ejecución de la aplicación

En el presente capítulo se describen las tecnologías requeridas y utilizadas para ejecutar la aplicación, cada paso y requerimiento es esencial para lograr una ejecución exitosa del programa.

5.1 Ambiente de ejecución

Para el correcto funcionamiento del *Web Scraper* se debe tener un equipo de cómputo ambientado de la manera adecuada, se debe de contar con un conjunto de programas y configuraciones las cuales serán mencionadas en este capítulo. Es importante mencionar que las pruebas realizadas se hacen en un sistema operativo Windows.

5.1.1 Requisitos del sistema

El *software* encargado de ejecutar el navegador *web* Chrome y ejecutar las tareas automatizadas se llama *Chrome Driver*, es un archivo de tipo aplicación, se debe encontrar en C:/Scraper/driver como lo muestra la figura 5.1.

Al ejecutar las tareas automatizadas este *driver* ejecutará las instrucciones programadas para obtener los datos deseados de los sitios *web*.

Dado que el lenguaje de programación es Java y como resultado de empaquetar el código fuente obtenemos un archivo con la extensión .jar es indispensable contar con una versión *Java SE Runtime Enviroment* superior a la 6. En este caso la versión instalada es 1.8.0.202, de esta manera podremos empaquetar nuestra codificación y ejecutar el resultado del mismo, ver figura 5.2.

5. EJECUCIÓN DE LA APLICACIÓN

El control de dependencias en nuestro proyecto será a través de Maven, esto nos permitirá usar clases ajenas a nuestro proyecto sin la necesidad de contar e instalar los archivos jar en nuestro equipo local, así como empaquetar nuestro código fuente, para el presente trabajo se debe contar con una versión Maven superior a la 3, en este caso la versión Maven es 3.8.4, ver figura 5.3.

El gestor de base de datos nos permitirá crear al menos un usuario con rol de administrador, un espacio en memoria dedicado a almacenar toda la configuración de la base de datos local para ello de debe contar con una versión Oracle *Database* superior a la 11, en este caso la versión *SQL* es 21.3.0.0.0, ver figura 5.4.

SQL Developer es una interfaz gráfica que nos va a permitir administrar nuestra base de datos relacional, así como manipular los datos de nuestros objetos de manera amigable, por ello debemos tener una versión del Oracle *SQL Developer* superior a la 11, en este caso se cuenta con la versión 21.2.1.204, ver figura 5.5.

La compatibilidad entre el navegador *web* y el *Chrome Driver* es muy específica, al momento de realizar la ejecución del programa se cuenta con la versión más reciente de Google Chrome y es la 96.0.4664.936, ver figura 5.6.

Esta compatibilidad se debe a que el *Chrome Driver* se acopla a la funcionalidad del navegador *web* en determinada versión, es quien accede a los atributos y acciones que el navegador acepta y permite en cierto momento.

El *Chrome Driver* es el 96.0.4664.45, en caso de contar con un navegador Chrome superior será necesario buscar en el sitio oficial de *Chrome Driver* la versión deseada y reemplazando en la ruta mencionada en la figura 5.7.

Después de que se generen las hojas de cálculo con los datos que hemos obtenido será necesario abrir los archivos resultantes con cualquier versión de *Open Office*, en este caso de cuenta con la versión de 4.1.11, ver figura 5.8.

Dado que los datos que se van a recopilar será de la internet, se debe contar con una conexión a esta red, en las pruebas se ejecutó el comando ping hacia el sitio *web* Google para obtener las estadísticas de envío y recepción de paquetes. Los resultados de la conexión a internet es de 38 ms mínimo, 42 ms máximo y una media de 39 ms en tiempos de ida y vuelta, ver figura 5.9.

5.1 Ambiente de ejecución

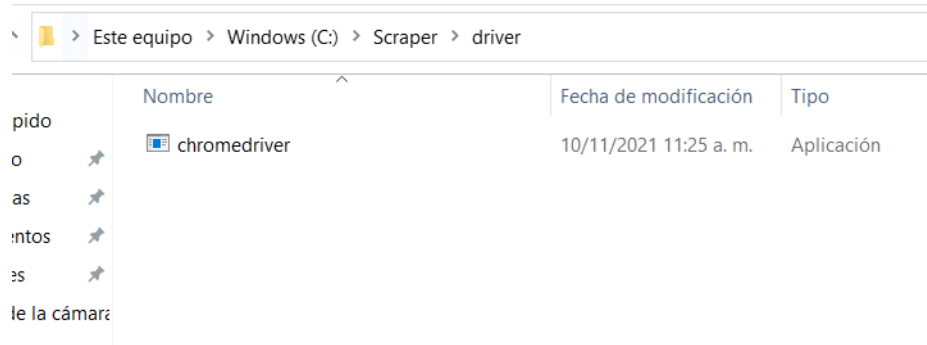


Figura 5.1: Ubicación del ChromeDriver.exe.

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\leone>java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)

C:\Users\leone>
```

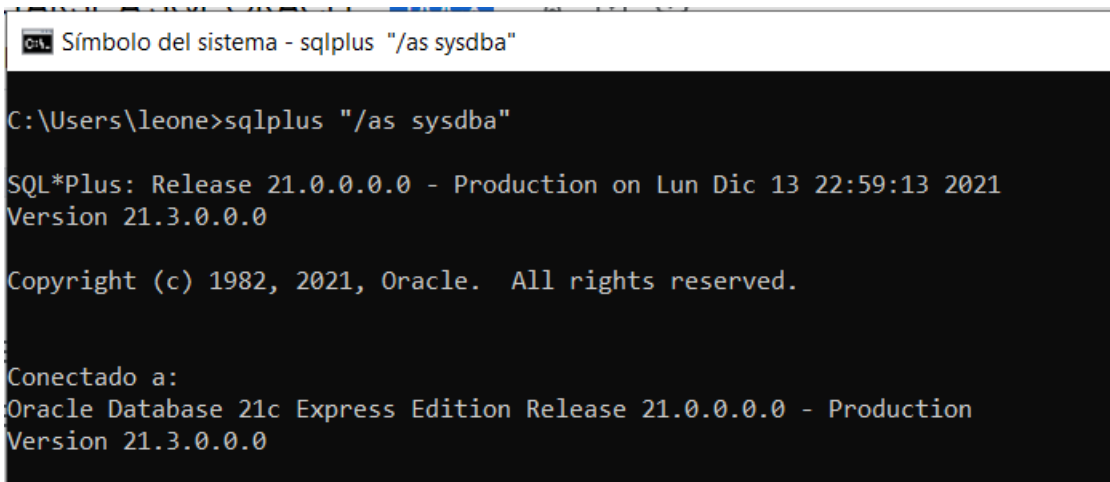
Figura 5.2: Versión de *Java SE Runtime Environment*.

```
Símbolo del sistema
C:\Users\leone>mvn -version
Apache Maven 3.8.4 (9b656c72d54e5baced989b64718c159fe39b537)
Maven home: C:\tools\apache-maven-3.8.4
Java version: 1.8.0_202, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_202\jre
Default locale: es_MX, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\leone>
```

Figura 5.3: Versión de Maven.

5. EJECUCIÓN DE LA APLICACIÓN



```
Símbolo del sistema - sqlplus "/as sysdba"

C:\Users\leone>sqlplus "/as sysdba"

SQL*Plus: Release 21.0.0.0.0 - Production on Lun Dic 13 22:59:13 2021
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
```

Figura 5.4: Versión de *SQL*.

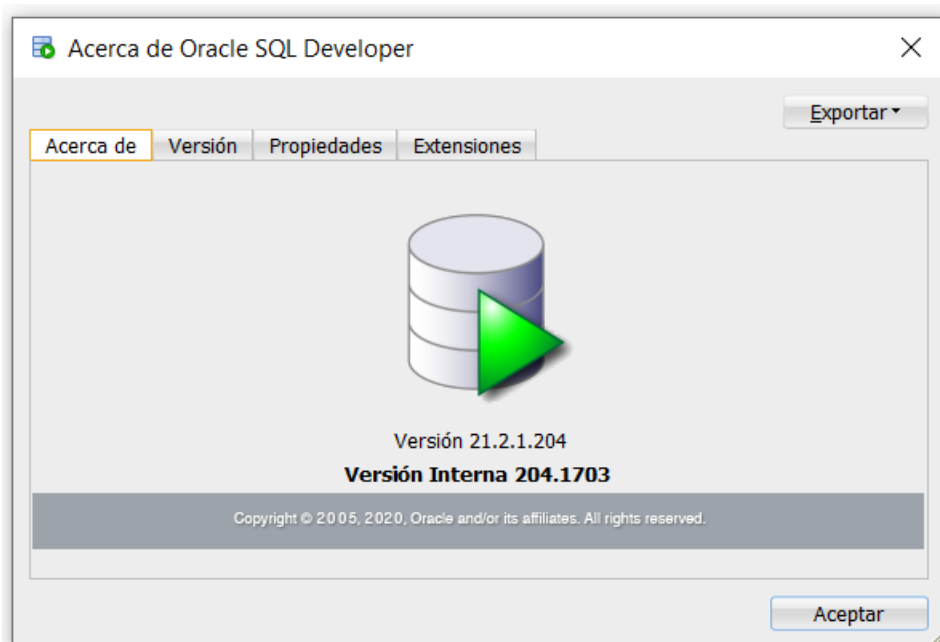


Figura 5.5: Versión del *SQL Developer* de Oracle.

5.1 Ambiente de ejecución

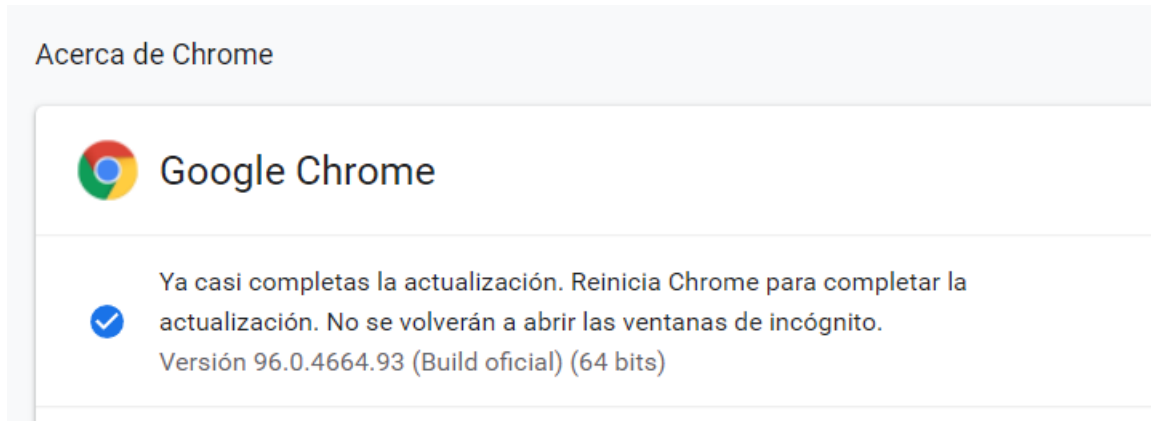


Figura 5.6: Versión de Google Chrome.

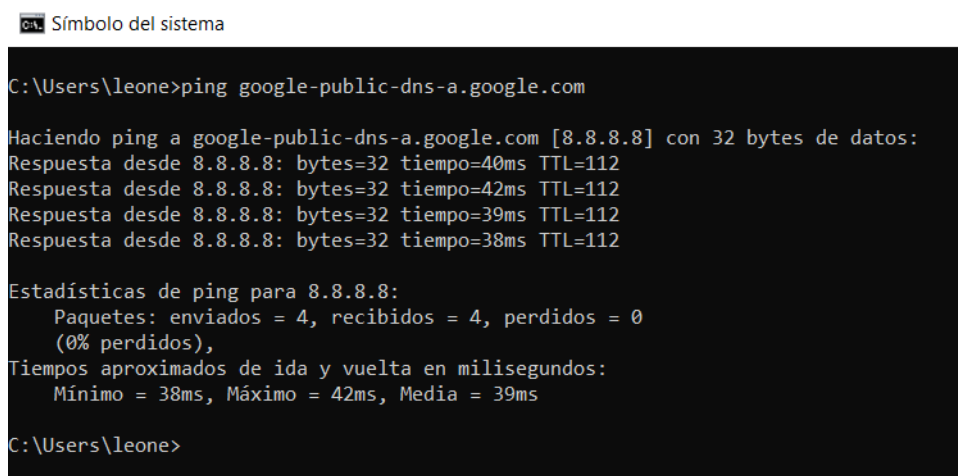
```
[SLF4J]: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.  
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444e61c0f17584b2fc-refs/branch-heads/4664@{c#947}) on port 35987
```

Figura 5.7: Versión del *Chrome Driver*



Figura 5.8: Versión de *Open Office*.

5. EJECUCIÓN DE LA APLICACIÓN



```

C:\Users\leone>ping google-public-dns-a.google.com

Haciendo ping a google-public-dns-a.google.com [8.8.8.8] con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=40ms TTL=112
Respuesta desde 8.8.8.8: bytes=32 tiempo=42ms TTL=112
Respuesta desde 8.8.8.8: bytes=32 tiempo=39ms TTL=112
Respuesta desde 8.8.8.8: bytes=32 tiempo=38ms TTL=112

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 38ms, Máximo = 42ms, Media = 39ms

C:\Users\leone>
```

Figura 5.9: Tener acceso a Internet.

5.1.1.1 Configuración de Base de Datos

En este apartado se describe como configurar al usuario y el tablespace de la Base de Datos, estos pasos son necesarios para poder ejecutar el archivo ddl que crea las tablas con sus respectivas columnas, llaves primarias y secundarias empleadas en el presente proyecto de tesis.

Se accede a la línea de comandos de *SQL* con los privilegios de administrador. Para esto basta con abrir una terminal cmd de windows y escribir el comando: sqlplus “/as sysdba”. Se mostrarán datos de la versión previamente instalada y un texto que nos indica que la conexión es correcta, ver figura 5.10.

Para destinar un espacio en la unidad de almacenamiento de nuestro equipo se debe crear un tablespace y asignarle un tamaño en memoria, aquí es donde se alojan los objetos y registros de nuestra Base de Datos. El comando ejecutado se muestra en la figura 5.11 y es:

```
CREATE TABLESPACE DATAFILE
'C:\oracle\app\oracle\oradata\XE\SCRAPER.dbf'
SIZE50MEXTENTMANAGEMENTLOCALAUTOLLOCATE;
```

Los siguientes comandos a ejecutar tienen la finalidad de crear al usuario y contraseña que permiten acceder a la Base de Datos local, después se asigna el usuario al tablespace como lo muestra la figura 5.12, se le otorgan los privilegios de administrador y se da un commit para guardar los cambios. Estos son los comandos:

```
alter session set“_ORACLE_SCRIPT”=true;
CREATE USER ADMINISTRADOR IDENTIFIED BY ADMINISTRADOR DE-
```


5.1 Ambiente de ejecución

```
FAULT TABLESPACE SCRPAER;  
GRANT ALL PRIVILEGES TO ADMINISTRADOR;  
COMMIT;
```

En las imágenes 5.13, 5.14, 5.15 y 5.16 se muestran las instrucciones para la creación de las tablas de Base de Datos Oracle. Es necesario para el almacenamiento de las credenciales de acceso a la aplicación (5.13), guardar la información obtenida del sitio *web* Superprof(5.14) y del sitio *web* Jobomas (5.15 y 5.16). En todos los casos se especifica el objeto creado(el nombre de la tabla), los atributos del mismo(nombre, tipo, longitud y capacidad de ser null de las columnas), el campo que es llave primaria y en caso de tener llave foránea se muestra la relación entre tablas padre e hija.

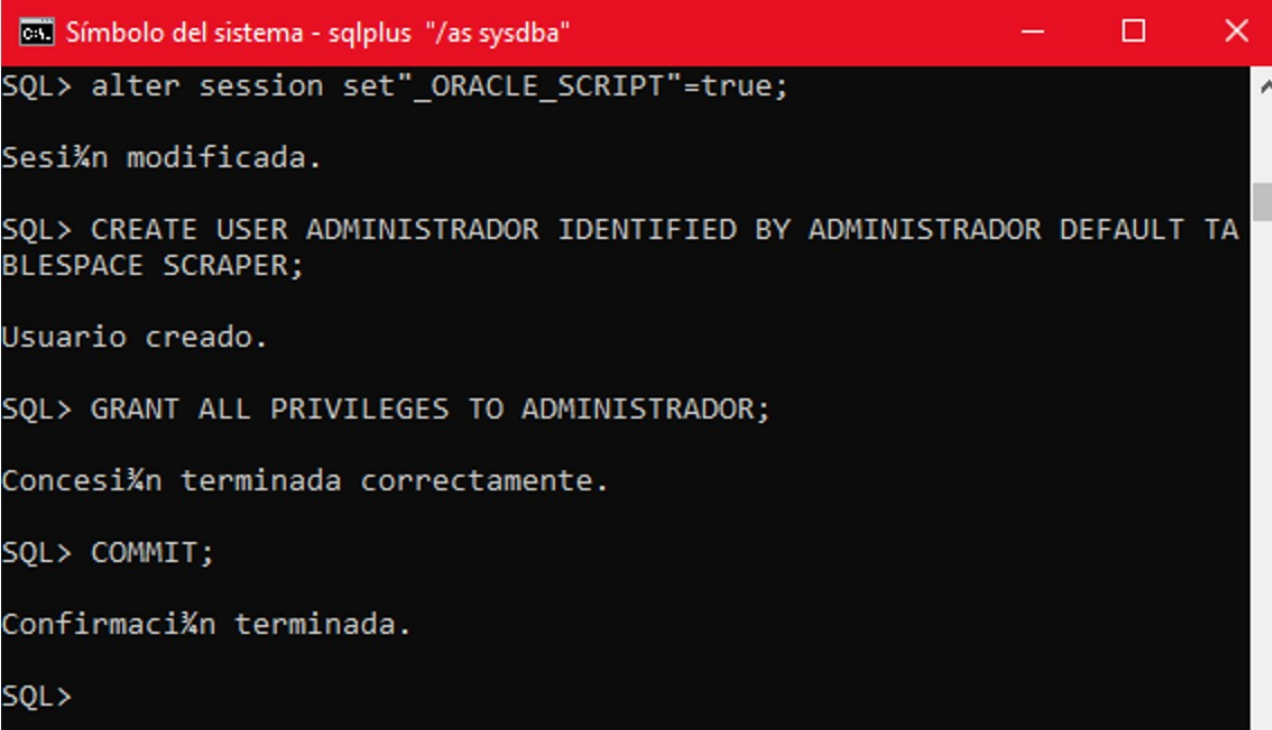
```
C:\Users\Leonel> sqlplus "/as sysdba"  
  
SQL*Plus: Release 18.0.0.0.0 - Production on Mar Nov 23 01:16:03 2021  
Version 18.4.0.0.0  
Copyright (c) 1982, 2018, Oracle. All rights reserved.  
  
Conectado a:  
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production  
Version 18.4.0.0.0
```

Figura 5.10: Acceso como sysdba.

```
SQL> CREATE TABLESPACE SCRAPER DATAFILE 'C:\oraclexe\app\oracle\oradata\XE\SCRAPER.dbf' SIZE 50M EXTENT MANAGEMENT LOCAL AUTOALLOCATE;  
  
Tablespace creado.
```

Figura 5.11: Creación de tablespace.

5. EJECUCIÓN DE LA APLICACIÓN



```
Símbolo del sistema - sqlplus "/as sysdba"
SQL> alter session set "_ORACLE_SCRIPT"=true;

Sesi n modificada.

SQL> CREATE USER ADMINISTRADOR IDENTIFIED BY ADMINISTRADOR DEFAULT TA
BLESPEACE SCRAPER;

Usuario creado.

SQL> GRANT ALL PRIVILEGES TO ADMINISTRADOR;

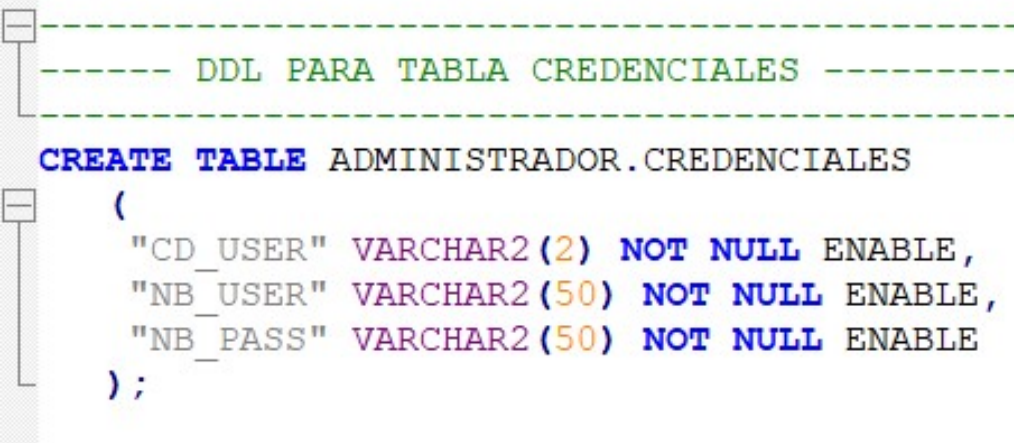
Concesi n terminada correctamente.

SQL> COMMIT;

Confirmaci n terminada.

SQL>
```

Figura 5.12: Creaci n de usuario.



```
-----
----- DDL PARA TABLA CREDENCIALES -----
-----
CREATE TABLE ADMINISTRADOR.CREDENCIALES
(
  "CD_USER" VARCHAR2(2) NOT NULL ENABLE,
  "NB_USER" VARCHAR2(50) NOT NULL ENABLE,
  "NB_PASS" VARCHAR2(50) NOT NULL ENABLE
);
```

Figura 5.13: DDL de tabla ADMINISTRADOR.CREDENCIALES.

5.1 Ambiente de ejecución

```
-----  
----- DDL DE TABLAS SUPERPROF -----  
-----  
  
CREATE TABLE administrador.superprof_profesores (  
    cd_profesor    NUMBER(10, 0)  
        NOT NULL ENABLE,  
    nombre         VARCHAR2(4000 BYTE),  
    idiomas       VARCHAR2(4000 BYTE),  
    estrellas     VARCHAR2(10 BYTE),  
    evaluaciones  VARCHAR2(100 BYTE),  
    estudiantes   VARCHAR2(100 BYTE),  
    tarifa        VARCHAR2(100 BYTE),  
    descripcion   VARCHAR2(4000 BYTE),  
    CONSTRAINT profesor_pk PRIMARY KEY ( cd_profesor )  
);  
  
CREATE TABLE administrador.superprof_clases (  
    cd_clase       NUMBER(10, 0)  
        NOT NULL ENABLE,  
    nombre_clase  VARCHAR2(3000 BYTE),  
    CONSTRAINT clase_fk FOREIGN KEY ( cd_clase )  
        REFERENCES administrador.superprof_profesores ( cd_profesor )  
);  
  
CREATE TABLE administrador.superprof_descripcion (  
    cd_descripcion NUMBER(10, 0)  
        NOT NULL ENABLE,  
    description   VARCHAR2(4000 BYTE),  
    CONSTRAINT descripcion_fk FOREIGN KEY ( cd_descripcion )  
        REFERENCES administrador.superprof_profesores ( cd_profesor )  
);  
  
CREATE TABLE administrador.superprof_lugares_de_clases (  
    cd_lugar_clase NUMBER(10, 0)  
        NOT NULL ENABLE,  
    nombre_lugar_clase VARCHAR2(3000 BYTE),  
    CONSTRAINT lugar_clase_fk FOREIGN KEY ( cd_lugar_clase )  
        REFERENCES administrador.superprof_profesores ( cd_profesor )  
);  
  
CREATE TABLE administrador.superprof_niveles (  
    cd_nivel       NUMBER(10, 0)  
        NOT NULL ENABLE,  
    nombre_nivel  VARCHAR2(3000 BYTE),  
    CONSTRAINT nivel_clase_fk FOREIGN KEY ( cd_nivel )  
        REFERENCES administrador.superprof_profesores ( cd_profesor )  
);
```

Figura 5.14: DDL de tablas que almacenan la información del sitio *web* Superprof

5. EJECUCIÓN DE LA APLICACIÓN

```
----- DDL PARA TABLAS JOBOMAS -----  
  
CREATE TABLE administrador.jobomas_perfiles (  
    id          NUMBER(10, 0)  
    |          NOT NULL ENABLE,  
    estudios    VARCHAR2(500 BYTE),  
    edad        VARCHAR2(13 BYTE),  
    sexo        VARCHAR2(20 BYTE),  
    telefono    VARCHAR2(20 BYTE),  
    celular     VARCHAR2(20 BYTE),  
    email       VARCHAR2(500 BYTE),  
    direccion   VARCHAR2(1000 BYTE),  
    CONSTRAINT perfiles_pk PRIMARY KEY ( id )  
);  
  
CREATE TABLE administrador.jobomas_experiencia_laboral (  
    cd_experiencia NUMBER(10, 0)  
    |          NOT NULL ENABLE,  
    puesto        VARCHAR2(500 BYTE),  
    empresa        VARCHAR2(1000 BYTE),  
    fecha          VARCHAR2(200 BYTE),  
    descripcion    VARCHAR2(4000 BYTE),  
    CONSTRAINT experiencia_fk FOREIGN KEY ( cd_experiencia )  
    |          REFERENCES administrador.jobomas_perfiles ( id )  
);  
  
CREATE TABLE administrador.jobomas_educacion (  
    cd_educacion  NUMBER(10, 0),  
    titulo        VARCHAR2(1000 BYTE),  
    escuela        VARCHAR2(1000 BYTE),  
    fecha          VARCHAR2(50 BYTE),  
    CONSTRAINT educacion_fk FOREIGN KEY ( cd_educacion )  
    |          REFERENCES administrador.jobomas_perfiles ( id )  
);  
  
CREATE TABLE administrador.jobomas_idiomas (  
    cd_idiomas    NUMBER(10, 0),  
    nivel          VARCHAR2(200 BYTE),  
    idiomas        VARCHAR2(2000 BYTE),  
    CONSTRAINT idiomas_fk FOREIGN KEY ( cd_idiomas )  
    |          REFERENCES administrador.jobomas_perfiles ( id )  
);
```

Figura 5.15: DDL de tablas que almacenan la información del sitio *web* Jobomas

```
CREATE TABLE administrador.jobomas_informatica (
  cd_informatica NUMBER(10, 0),
  nivel VARCHAR2(150 BYTE),
  tecnologia VARCHAR2(2000 BYTE),
  CONSTRAINT informatica_fk FOREIGN KEY ( cd_informatica )
    REFERENCES administrador.jobomas_perfiles ( id )
);

CREATE TABLE administrador.jobomas_otros (
  cd_otros NUMBER(10, 0)
    NOT NULL ENABLE,
  otros VARCHAR2(2000 BYTE),
  CONSTRAINT otros_fk FOREIGN KEY ( cd_otros )
    REFERENCES administrador.jobomas_perfiles ( id )
);
```

Figura 5.16: DDL de tablas que almacenan la información del sitio *web* Jobomas

5.1.2 Empaquetado de código y ejecución de la aplicación

El proceso de transformar el código fuente a un archivo con extensión `.jar` que nos permita ejecutar la aplicación desarrollada es conocida como “empaquetado de código”. Para empaquetar el código en un archivo `jar` se requiere la carpeta `scr` y el archivo `pom` del proyecto, estos archivos son los generados por el IDE al momento de codificar la aplicación, ver figura 5.17.

Situados en la raíz del proyecto, se abre una terminal `cmd` de windows y se ejecuta el comando `mvn package`, la tecnología de maven se encargará de descargar las dependencias, ejecutar los test y empaquetar nuestro código, ver figura 5.18 y 5.19.

Al termino del empaquetado se ve la leyenda `BUILD SUCCESS` en la consola, esto nos indica que el archivo `jar` ha sido creado de forma correcta. En la traza que muestra la terminal podemos visualizar más información como el tiempo total que tardó en el proceso o la fecha del mismo, ver figura 5.19.

Como resultado del empaquetado, en la raíz del proyecto se genera una nueva carpeta llamada `target`, ver figura 5.20.

Dentro de la carpeta `target` de encuentra el archivo `tesisLeonel-1.0.0.jar`, este archivo es nuestro archivo `jar`.El siguiente paso será ejecutarlo, ver figura 5.21.

Para ejecutarlo basta con dar doble clic sobre `tesisLeonel-1.0.0.jar`, de esta manera comienza con la ejecución de nuestra aplicación *Web Scraper* y comienza con abrir la ventana de acceso a la aplicación, ver figura 5.22.

5. EJECUCIÓN DE LA APLICACIÓN

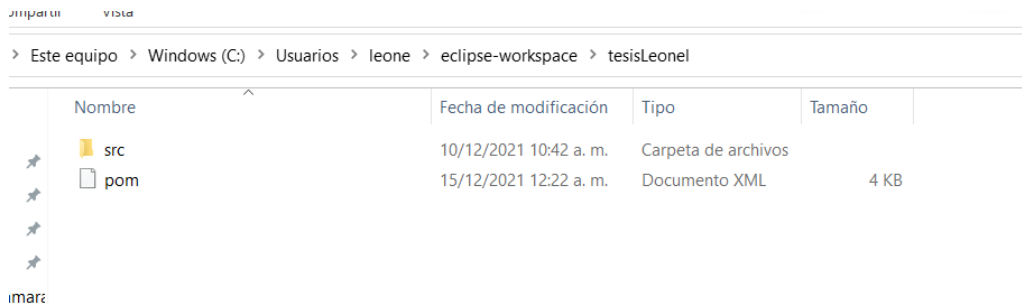


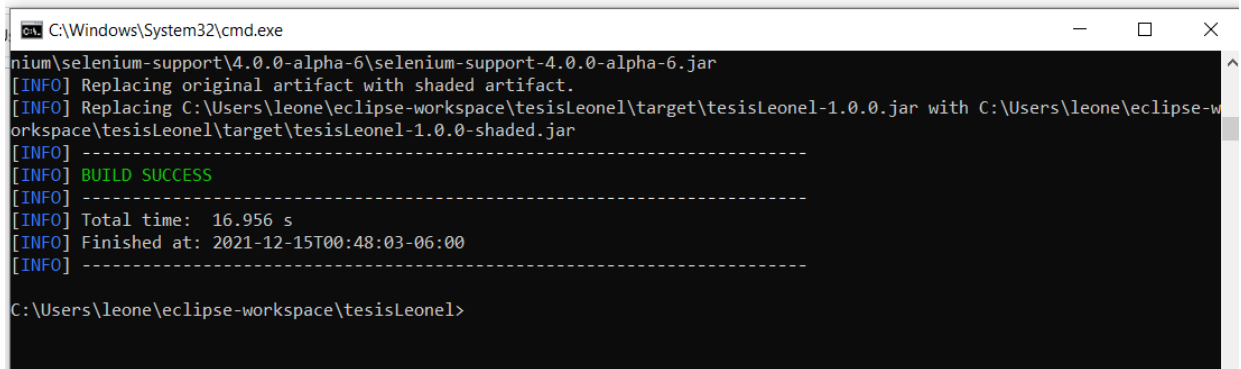
Figura 5.17: Archivos fuente.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\leone\eclipse-workspace\tesisLeonel>mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.tesis.leonel:tesisLeonel >-----
[INFO] Building tesisLeonel 1.0.0
[INFO]
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ tesisLeonel ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 10 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ tesisLeonel ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 34 source files to C:\Users\leone\eclipse-workspace\tesisLeonel\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ tesisLeonel ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ tesisLeonel ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ tesisLeonel ---
[INFO]
[INFO] --- maven-jar-plugin:3.1.0:jar (default-jar) @ tesisLeonel ---
[INFO] Building jar: C:\Users\leone\eclipse-workspace\tesisLeonel\target\tesisLeonel-1.0.0.jar
[INFO]
[INFO] --- maven-shade-plugin:1.3.1:shade (default) @ tesisLeonel ---
[INFO] Including com.oracle:ojdbc6:jar:11.1.0.7.0 in the shaded jar.
[INFO] Including org.apache.poi:poi:jar:3.16 in the shaded jar.
[INFO] Including commons-codec:commons-codec:jar:1.10 in the shaded jar.
[INFO] Including org.apache.poi:poi-ooxml-schemas:jar:3.16 in the shaded jar.
[INFO] Including org.apache.poi:poi-ooxml:jar:3.16 in the shaded jar.
[INFO] Including org.apache.commons:commons-collections4:jar:4.1 in the shaded jar.
[INFO] Including org.apache.xmlbeans:xmlbeans:jar:2.6.0 in the shaded jar.
[INFO] Including stax:stax-api:jar:1.0.1 in the shaded jar.
[INFO] Including com.jgoodies:jgoodies-forms:jar:1.8.0 in the shaded jar.
[INFO] Including com.jgoodies:jgoodies-common:jar:1.8.0 in the shaded jar.
[INFO] Including org.seleniumhq.selenium:selenium-java:jar:4.0.0-alpha-6 in the shaded jar.
[INFO] Including org.seleniumhq.selenium:selenium-api:jar:4.0.0-alpha-6 in the shaded jar.
[INFO] Including org.seleniumhq.selenium:selenium-chrome-driver:jar:4.0.0-alpha-6 in the shaded jar.
[INFO] Including com.google.auto.service:auto-service-annotations:jar:1.0-rc6 in the shaded jar.
[INFO] Including com.google.auto.service:auto-service:jar:1.0-rc6 in the shaded jar.
```

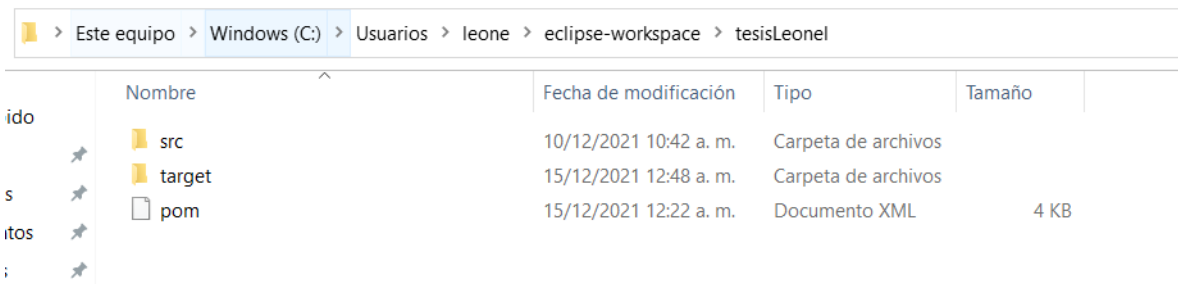
Figura 5.18: Inicio de empaquetado

5.1 Ambiente de ejecución



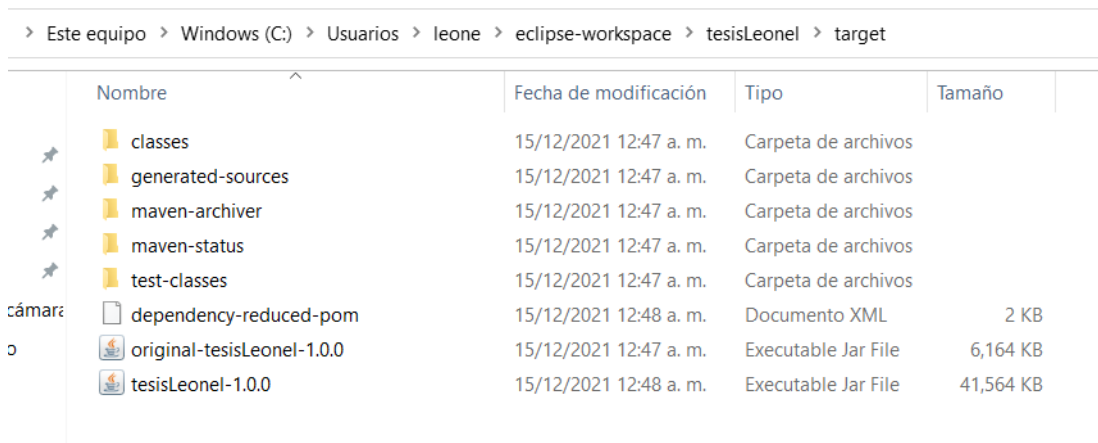
```
C:\Windows\System32\cmd.exe
nium\selenium-support\4.0.0-alpha-6\selenium-support-4.0.0-alpha-6.jar
[INFO] Replacing original artifact with shaded artifact.
[INFO] Replacing C:\Users\leone\eclipse-workspace\tesisLeone1\target\tesisLeone1-1.0.0.jar with C:\Users\leone\eclipse-workspace\tesisLeone1\target\tesisLeone1-1.0.0-shaded.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 16.956 s
[INFO] Finished at: 2021-12-15T00:48:03-06:00
[INFO] -----
C:\Users\leone\eclipse-workspace\tesisLeone1>
```

Figura 5.19: Fin de empaquetado.



Nombre	Fecha de modificación	Tipo	Tamaño
src	10/12/2021 10:42 a. m.	Carpeta de archivos	
target	15/12/2021 12:48 a. m.	Carpeta de archivos	
pom	15/12/2021 12:22 a. m.	Documento XML	4 KB

Figura 5.20: Generación de carpeta target.



Nombre	Fecha de modificación	Tipo	Tamaño
classes	15/12/2021 12:47 a. m.	Carpeta de archivos	
generated-sources	15/12/2021 12:47 a. m.	Carpeta de archivos	
maven-archiver	15/12/2021 12:47 a. m.	Carpeta de archivos	
maven-status	15/12/2021 12:47 a. m.	Carpeta de archivos	
test-classes	15/12/2021 12:47 a. m.	Carpeta de archivos	
dependency-reduced-pom	15/12/2021 12:48 a. m.	Documento XML	2 KB
original-tesisLeone1-1.0.0	15/12/2021 12:47 a. m.	Executable Jar File	6,164 KB
tesisLeone1-1.0.0	15/12/2021 12:48 a. m.	Executable Jar File	41,564 KB

Figura 5.21: Generación de archivo jar.

5. EJECUCIÓN DE LA APLICACIÓN

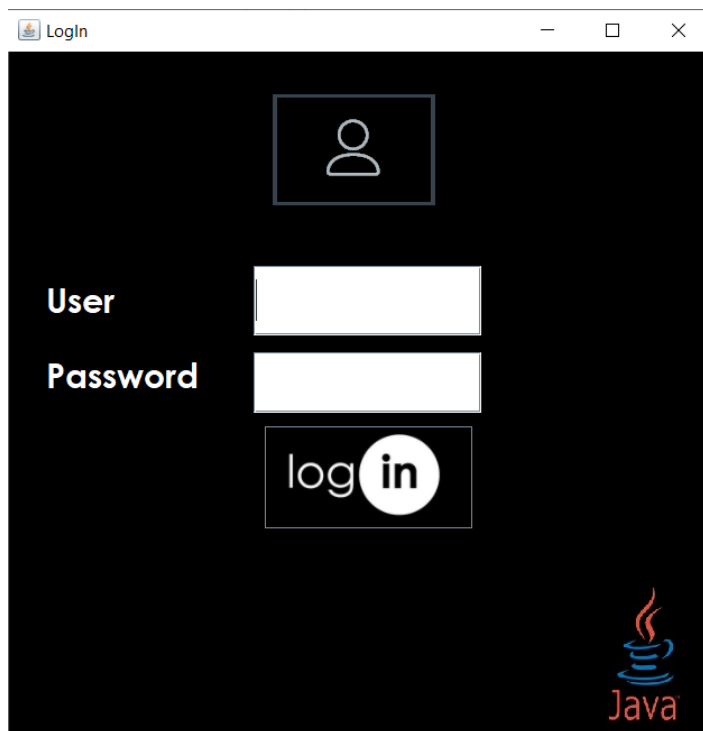


Figura 5.22: Ejecución de aplicación.

5.2 Inicio de sesión

Las credenciales otorgan permiso al usuario para ejecutar la aplicación, estas se almacenan en la tabla `ADMINISTRADOR.CREDENCIALES` de Base de Datos. Se pueden dar de alta tantos usuarios como se desee. Dentro del código fuente se valida el hashcode del usuario y la contraseña.

5.2.1 Acceso a la aplicación y gestión de credenciales

Se ha dado de alta al usuario *admin* con la contraseña *admin*, por lo que intentar acceder a la aplicación con datos diferentes a estos será considerado como un acceso fallido, ver figura 5.23.

La aplicación se puede ejecutar dando doble clic sobre el archivo *tesisLeonel-1.0.0.jar*, la máquina virtual de java es ejecutada en este momento, nuestro archivo jar es un conjunto de clases comprimidas, al instalar el *JDK* en automático se asocian los archivos .jar a la máquina virtual de java(*java.exe*). En este caso se usará la consola

5.2 Inicio de sesión

para ejecutar la aplicación y con esto dar seguimiento a la traza del proceso. Se debe abrir una terminal cmd en la misma ruta que el archivo .jar y ejecutar el comando `java -jar tesisLeonel-1.0.0.jar`, ver figura 5.24.

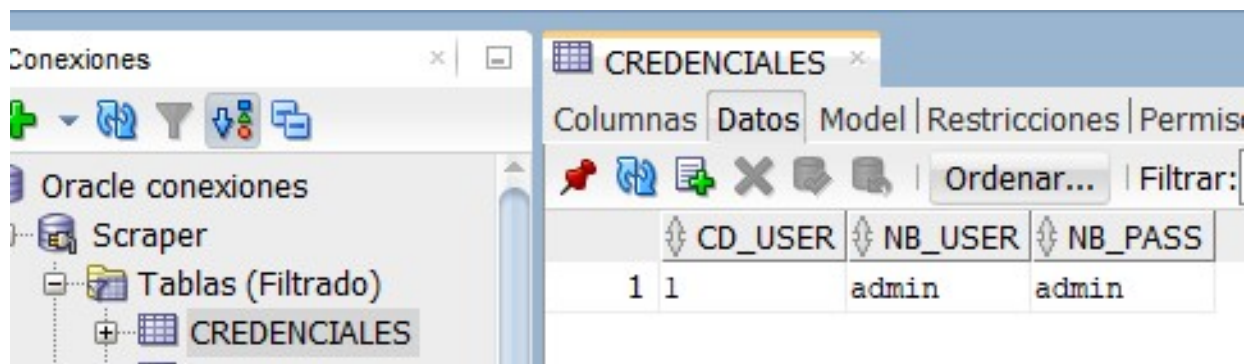
La única diferencia entre ejecutar el jar mediante comando en la terminal o dando doble clic sobre el jar es que se abrirá una ventana extra en donde se podrá ver la traza de ejecución.

Al ejecutar la aplicación nos vamos a encontrar con la primer ventana, esta es la de *LogIn*, se debe ingresar al usuario y contraseña de acceso en los recuadros *User* y *Password*. El campo *password* tiene una máscara de asteriscos para ocultar esta información. En esta figura se ha ingresado al usuario como *adminn*, con una *n* extra, al dar clic sobre el botón *log in* no permitirá el acceso. Las credenciales son consultadas en el motor de base de datos como lo indica la figura 5.25.

Cuando los datos almacenados en base de datos no son los mismos que los ingresados en el campo *User* y *Password* vuelven a estar vacíos, el nombre de la ventana ahora es *Reintento de acceso* y en la traza podemos ver los valores de *usuario* y *contraseña* ingresados además de la leyenda *Acceso fallido* en la terminal de la izquierda. Aquí tenemos a nivel código un ciclo que no permitirá el paso a la aplicación si no se ingresan los datos correctos, ver figura 5.26.

Para tener acceso a la aplicación, de acuerdo a las credenciales dadas de alta en Base de Datos, se debe escribir *admin* en *User* y *Pasword*, después dar clic en *LogIn*, esta acción rompe con la validación que impide el acceso al siguiente bloque de código y así continuamos con la siguiente ventana., ver figura 5.27.

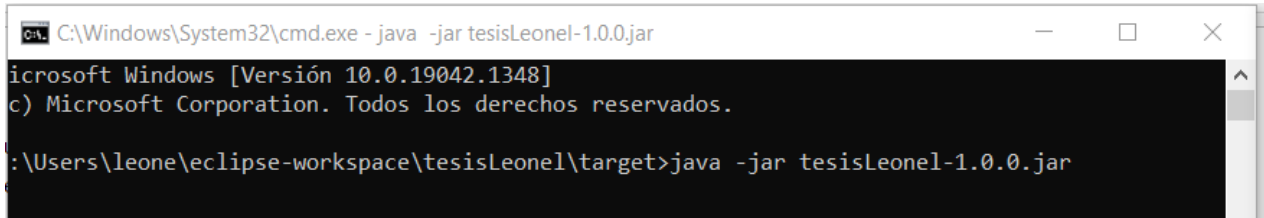
El acceso correcto permite pasar a la ventana 2, el Buscador. En la traza (terminal de la izquierda) se ve como el usuario y contraseña *admin* han sido introducidos, mostrando en esta ocasión un mensaje de *Acceso correcto*, ver figura 5.28.



Columnas	Datos	Model	Restricciones	Permis
1	1	admin	admin	

Figura 5.23: Credenciales de acceso.

5. EJECUCIÓN DE LA APLICACIÓN



```
C:\Windows\System32\cmd.exe - java -jar tesisLeonel-1.0.0.jar
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\leone\workspace\tesisLeonel\target>java -jar tesisLeonel-1.0.0.jar
```

Figura 5.24: Ejecución de jar.

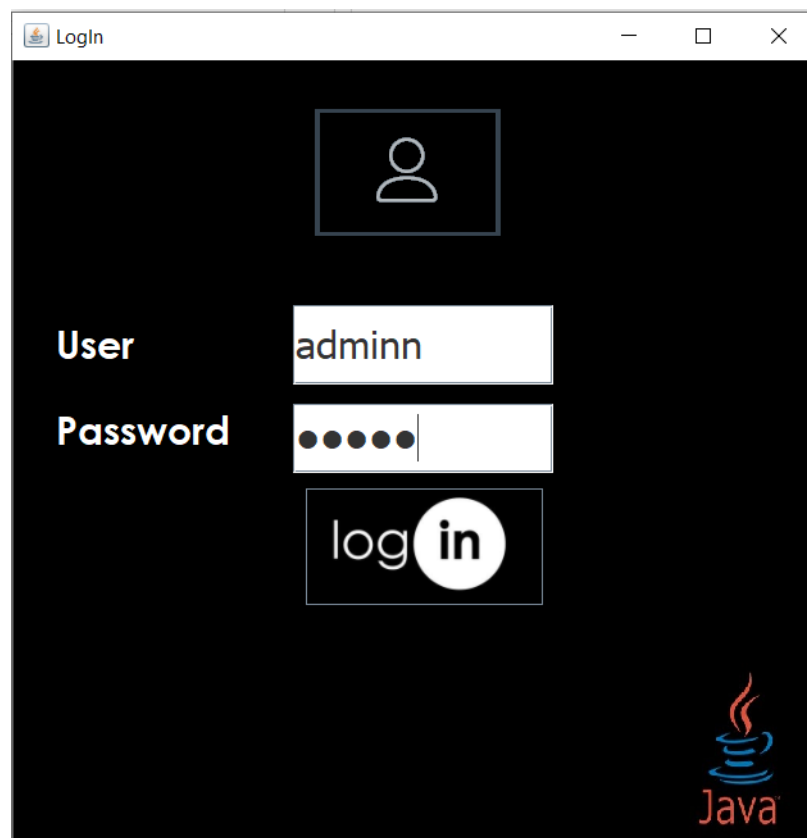


Figura 5.25: Credenciales incorrectas, se coloca al usuario incorrecto.

5.2 Inicio de sesión

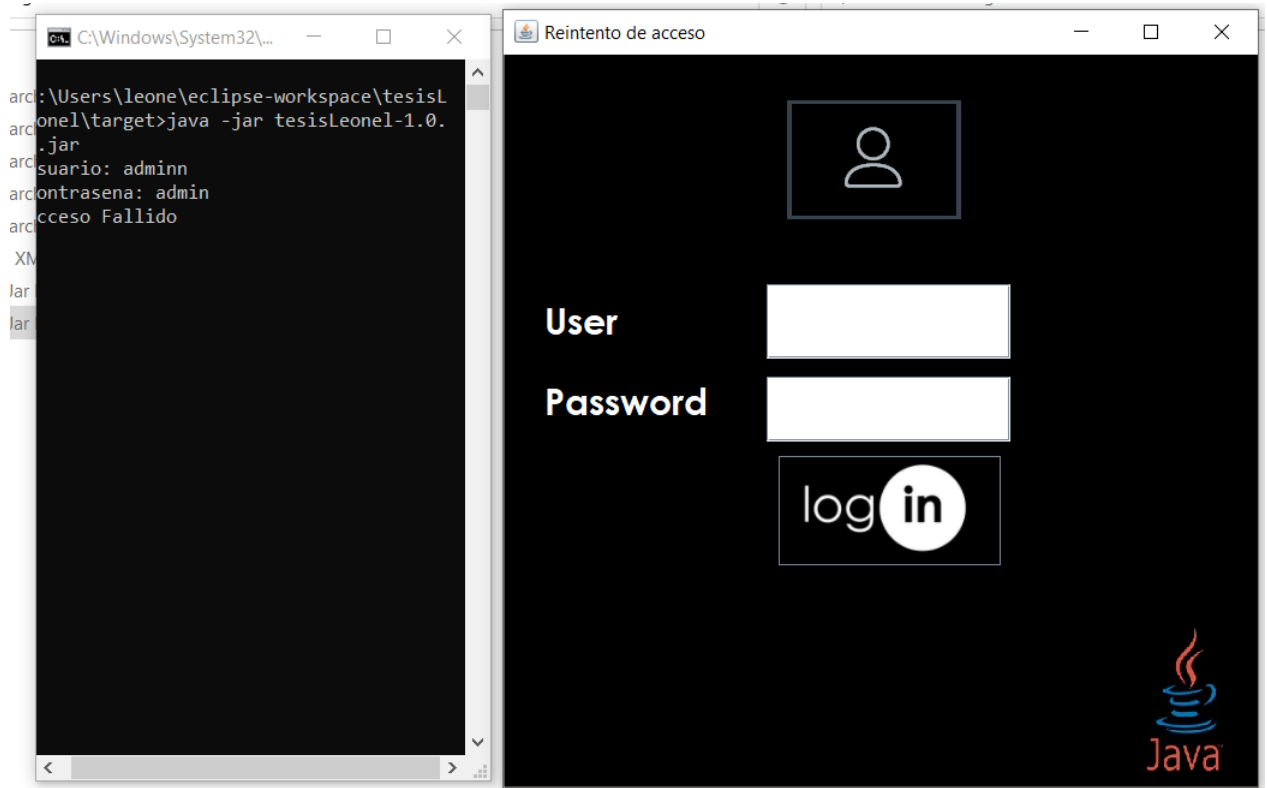


Figura 5.26: Mensaje en credenciales incorrectas.

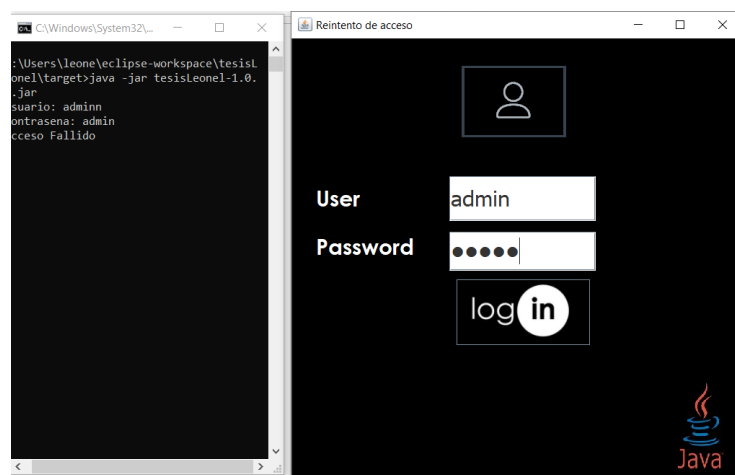


Figura 5.27: Credenciales correctas.

5. EJECUCIÓN DE LA APLICACIÓN

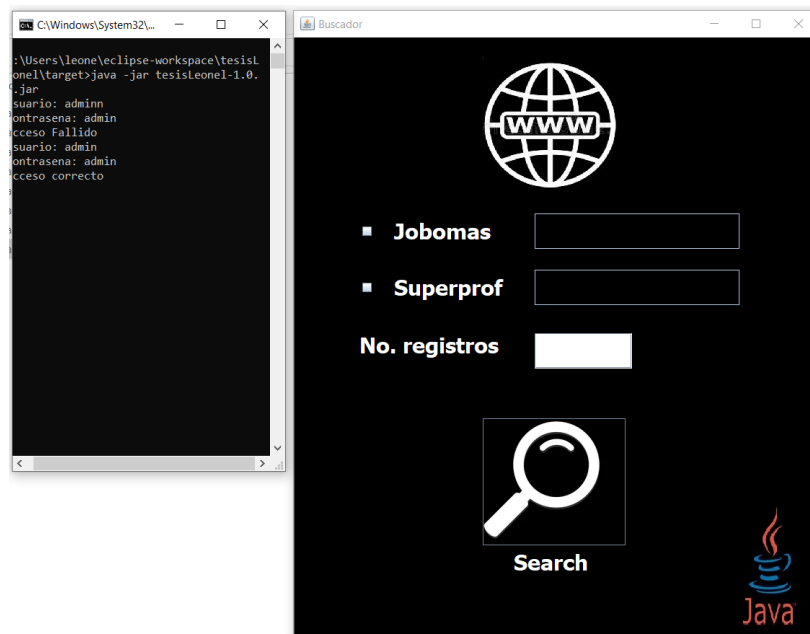


Figura 5.28: Acceso correcto, vista de búsqueda de perfiles.

5.3 Búsqueda de perfiles

En esta sección se muestra el proceso de búsqueda y recolección de datos de perfiles en internet empleando la técnica de *Web Scraper*. El proceso hace uso de hilos para ahorrar tiempos de procesamiento en las búsquedas, un hilo se ejecuta para el proceso de Jobomas y otro hilo en el proceso de Superprof. Al final los datos quedan almacenados en la Base de Datos *SQL*, se crean hojas de cálculo con la información obtenida y se muestra un resumen de la información en pantalla.

5.3.1 Parámetros de búsqueda

Se tiene un checkbox para habilitar la búsqueda de un perfil Jobomas y/o Superprof. Para realizar la búsqueda en un sitio *web* es necesario seleccionarlo dando clic sobre el recuadro en blanco, esto habilitará o deshabilitará la opción, al menos se debe de elegir una opción para poder realizar la búsqueda de perfiles o de lo contrario no se avanzará al siguiente módulo, a nivel código se delimitó el rango de búsqueda de registros entre 1 y 100 por lo que escribir un número fuera de estos límites bloqueará la continuidad de la aplicación como se muestra en la figura 5.30(el encabezado de

la ventana dice *Buscador, Escriba un número entre 1 y 100*).

Se programó una ventana emergente para el caso en donde el usuario escriba una letra en el campo designado al *Número de registros*, la alerta indica que se ingresen solo valores numéricos en el campo, ver figura 5.31.

Para esta demostración se ejecutará el *Web Scraper* en ambos sitios *web* habilitados en el checkbox, esto con la finalidad de ver la ejecución de los hilos durante el ciclo de vida del programa. El perfil de Jobomas que vamos a solicitar se busca que es *Ingeniero* y el profesor de Superprof es *matemáticas*, además se obtendrá la información de 10 registros para cada caso, ver figura 5.32.

En la figura 5.33 podemos ver la consola con diferente información, algunos datos que a simple vista se visualizan son *Número de registros solicitados, Perfil y Materia*, se observa como se inician dos instancias del navegador *web* mediante el *Web Driver* con la leyenda *Starting ChromeDriver ...*, el puerto que se le ha asignado, la fecha de la sesión creada de forma correcta y el mensaje *started successfully*. Se han iniciado dos instancias mediante los hilos, una instancia accederá a la página de Jobomas y otra a la página de Superprof.

En la traza de la figura 5.34 se observa como va alternando la ejecución entre Superprof y Jobomas, este es el comportamiento que toma el programa derivado de emplear Hilos en su programación, el segmento de código destinado a cada sitio *web* recupera la información e inserta en Base de Datos los registros de forma alternada. Primero termina de ejecutar el *Web Scraper* en Superprof y después en Jobomas, esto se debe al uso de Hilos y a que las páginas tienen una codificación interna diferente. La obtención de información de cada página de internet provoca los módulos de programación en cada Hilo entren en *race condition* o condición de carrera, por lo que compiten por la impresión en pantalla. Para el presente caso el acceso y recuperación de información en la página de Superprof lleva lapsos de tiempos más cortos que Jobomas. En ambos casos recupera los 10 perfiles de forma correcta, al termino de cada instancia cierra el navegador *web* correspondiente en diferentes tiempo como se muestra en el log.

En la figura 5.35 se observa el trabajo de los Hilos en Java, se crean 2 instancias del navegador *web* y en paralelo es que se recupera la información de los dos sitios *web*. Se abren de forma automática 2 exploradores *web*, comienzan a navegar a través de un sitio de internet cada navegador y al mismo tiempo.

5. EJECUCIÓN DE LA APLICACIÓN

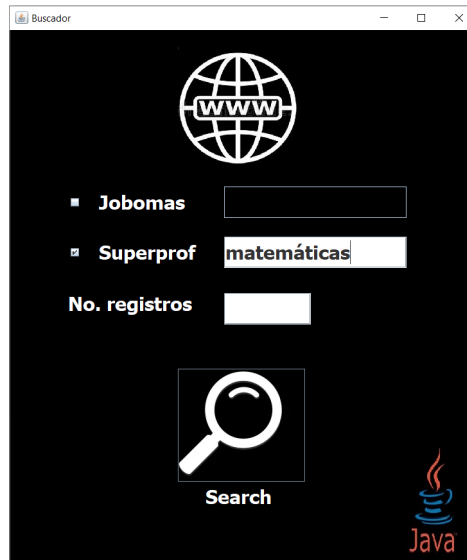


Figura 5.29: Habilitar buscador.



Figura 5.30: Buscador, rango de registros.

5.3 Búsqueda de perfiles

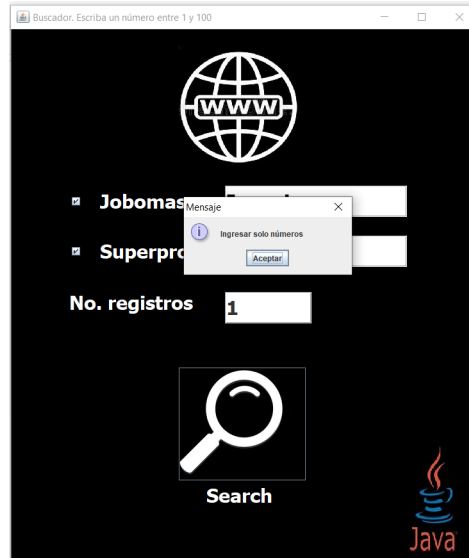


Figura 5.31: Buscador, caracteres válidos.



Figura 5.32: Buscador, búsqueda de perfiles.

5. EJECUCIÓN DE LA APLICACIÓN

```
C:\Windows\System32\cmd.exe - java -jar tesisLeonel-1.0.0.jar
C:\Users\leone\eclipse-workspace\tesisLeonel\target>java -jar tesisLeonel-1.0.0.jar
Usuario: adminn
Contraseña: admin
Acceso Fallido
Usuario: admin
Contraseña: admin
Acceso correcto
Número de Registros solicitados: 10
Perfil : Ingeniero
Materia : matemáticas
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444e61c0f17584b2fc-refs/branch-heads/4664@{
#947}) on port 35987
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping Chrom
eDriver safe.
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444e61c0f17584b2fc-refs/branch-heads/4664@{
#947}) on port 17069
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping Chrom
eDriver safe.
ChromeDriver was started successfully.
ChromeDriver was started successfully.
dic 15, 2021 1:21:11 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFORMACIËN: Detected dialect: W3C
dic 15, 2021 1:21:11 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFORMACIËN: Detected dialect: W3C
test insertando registros
Registrando perfil 1 Superprof
test insertando registros
Registrando perfil 2 Superprof
Registrando perfil 1 Jobomas
test insertando registros
Registrando perfil 3 Superprof
test insertando registros
Registrando perfil 4 Superprof
Registrando perfil 2 Jobomas
test insertando registros
Registrando perfil 5 Superprof
test insertando registros
Registrando perfil 6 Superprof
Registrando perfil 3 Jobomas
test insertando registros
Registrando perfil 7 Superprof
test insertando registros
Registrando perfil 8 Superprof
Registrando perfil 4 Jobomas
```


5.3 Búsqueda de perfiles

```
INFORMACION: Detected dialect: W3C
dic 15, 2021 1:21:11 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFORMACION: Detected dialect: W3C
test insertando registros
Registrando perfil 1 Superprof
test insertando registros
Registrando perfil 2 Superprof
Registrando perfil 1 Jobomas
test insertando registros
Registrando perfil 3 Superprof
test insertando registros
Registrando perfil 4 Superprof
Registrando perfil 2 Jobomas
test insertando registros
Registrando perfil 5 Superprof
test insertando registros
Registrando perfil 6 Superprof
Registrando perfil 3 Jobomas
test insertando registros
Registrando perfil 7 Superprof
test insertando registros
Registrando perfil 8 Superprof
Registrando perfil 4 Jobomas
test insertando registros
Registrando perfil 9 Superprof
test insertando registros
Registrando perfil 10 Superprof
Cerrando página Superprof
Registrando perfil 5 Jobomas
Registrando perfil 6 Jobomas
Registrando perfil 7 Jobomas
Registrando perfil 8 Jobomas
Registrando perfil 9 Jobomas
Registrando perfil 10 Jobomas
Cerrando página Jobomas
```

Figura 5.34: Log, final.

5. EJECUCIÓN DE LA APLICACIÓN

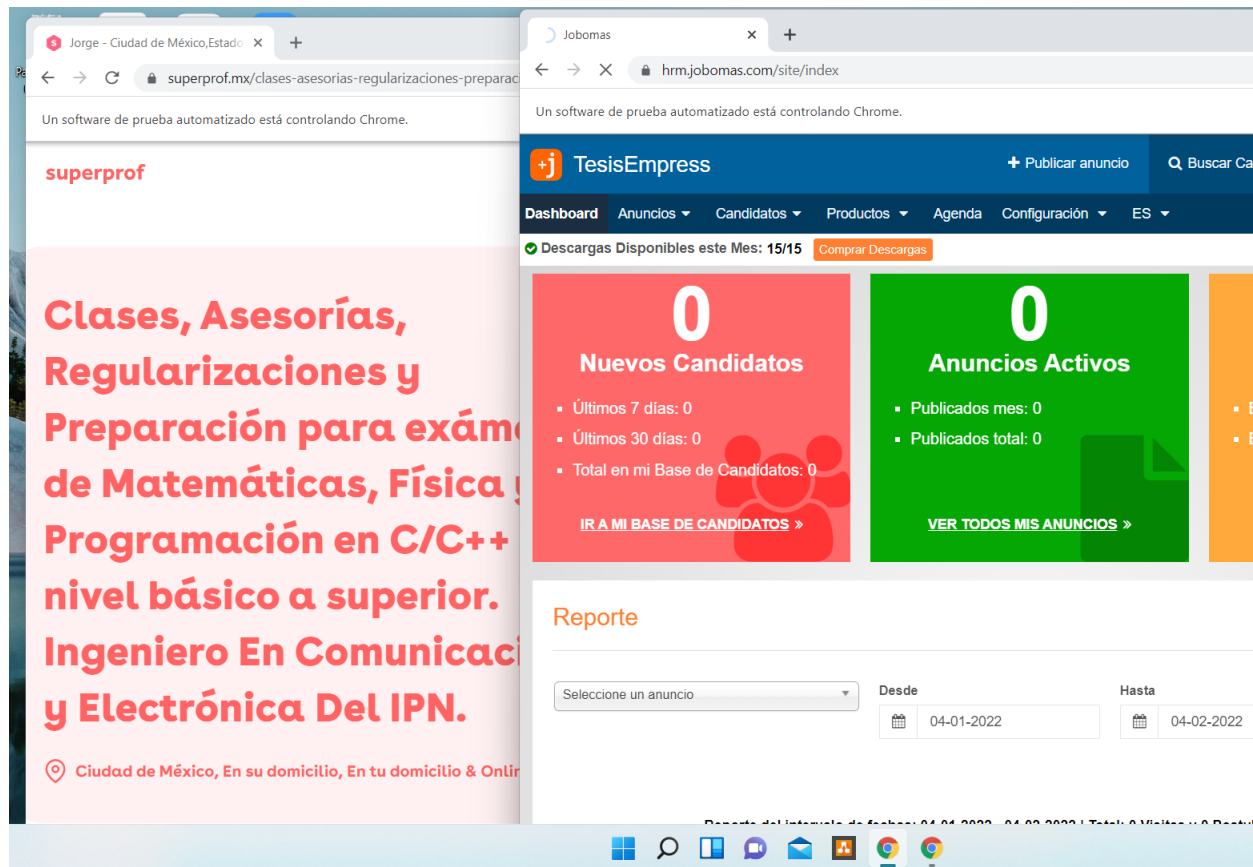


Figura 5.35: Ejecución de procesos empleando Hilos.

5.4 Generación de tablas

Al término de la búsqueda de perfiles en los sitios *web*, se genera una tabla con el resumen de los datos obtenidos de Superprof y otra tabla con los resultados del sitio *web* Jobomas, ver figura 5.36.

Se crea la vista que permite al usuario la opción de *Crear Hoja*: esta crea la hoja de cálculo con todos los datos que se almacenaron en base de datos de Jobomas y/o Superprof durante la ejecución de la aplicación, accede a la base de datos y llena el *DTO* correspondiente con los datos para que la misma viaje a través de las capas. La creación de estos archivos depende de qué sitio *web* se seleccionó al momento de ejecutar el programa y se genera un reporte por cada sitio *web* consultado. Cada reporte es guardado en la unidad de disco local y lleva como nombre un identificador de fecha y hora en el que es generado.

5.4 Generación de tablas

The image shows a Java Swing application with three windows. The 'Superprof resumen' window contains the following table:

ID	NOMBRE	IDIOMAS	ESTRELLAS	EVALUACIONES	ESTUDIANTES	TARIFA	CLASES	LUGARES DE CLASES	NIVELES
1	Luis	Inglés Español	5	24 evaluaciones	50+	\$150	Álgebra Mate...	En su domicilio Ciudad de M...	Todos l...
2	Raúl	Español	4.9	41 evaluaciones	50+	\$200	Álgebra Mate...	Online En tu domicilio Vera...	Todos l...
3	Luis Angel	Inglés Español	5	44 evaluaciones	50+	\$280	Álgebra Mate...	Online Ciudad de México	Prepar...
4	Ingeniero Noé	Español	4.9	68 evaluaciones	50+	\$200	Álgebra Mate...	Online En tu domicilio En s...	Todos l...
5	Luis	Español	4.9	33 evaluaciones	50+	\$300	Álgebra Mate...	Online Ciudad de México	Secun...
6	Erika	Español	5	34 evaluaciones	50+	\$200	Álgebra Mate...	Online Ciudad de México	Secun...
7	Melissa	Español	5	23 evaluaciones	50+	\$150	Álgebra Mate...	Online Ciudad de México	Todos l...
8	Carlos	Inglés Español	4.9	14 evaluaciones	50+	\$200	Álgebra Mate...	Online En tu domicilio En s...	Secun...
9	GERARDO	Español	4.9	32 evaluaciones	50+	\$250	Álgebra Mate...	Online En tu domicilio Ciud...	Prepar...
10	Fernando	Inglés Español	5	13 evaluaciones	48	\$110	Álgebra Mate...	Online Ciudad de México	Prepar...

The 'Jobomas resumen' window contains the following table:

ID	ESTUDIOS	EDAD	SEXO	TELEFONO	CELULAR	EMAIL	DIRECCION
1	Ingeniero en Sistemas Pro...	31 años	Hombre	123456789	null	xxxxxxxx@xxxx.xxx	San Luis Potosí, San Luis P...
2	INGENIERIA INDUSTRIAL Q...	57 años	No espe...	123456789	null	xxxxxxxx@xxxx.xxx	CD DE MEXICO, Estado de ...
3	Ingeniero mecánico	40 años	Hombre	123456789	null	xxxxxxxx@xxxx.xxx	México , Estado de México,...
4	Ingeniero Industrial Admini...	35 años	Mujer	123456789	null	xxxxxxxx@xxxx.xxx	Irapuato, Guanajuato, Méx...
5	Linux	48 años	Hombre	null	null	xxxxxxxx@xxxx.xxx	Tepeji del Rio de Ocampo ...
6	Ingeniero geólogo	29 años	Mujer	123456789	null	xxxxxxxx@xxxx.xxx	Chihuahua , Chihuahua, M...
7	Ingeniero Industrial y de Si...	30 años	Hombre	null	null	xxxxxxxx@xxxx.xxx	Toluca, México
8	Licenciatura en Bioquímica ...	29 años	Hombre	null	null	xxxxxxxx@xxxx.xxx	Ciudad de Mexico, Ciudad ...
9	Administración de Calidad ...	38 años	Hombre	123456789	null	xxxxxxxx@xxxx.xxx	San Luis Potosi, San Luis P...
10	Ingeniero industrial	33 años	Hombre	null	null	xxxxxxxx@xxxx.xxx	Tlaquepaque , Jalisco, Méxi...

The 'Crear Hoja' window contains a 'Terminar' button and a Java logo.

Figura 5.36: Vista crear hoja.

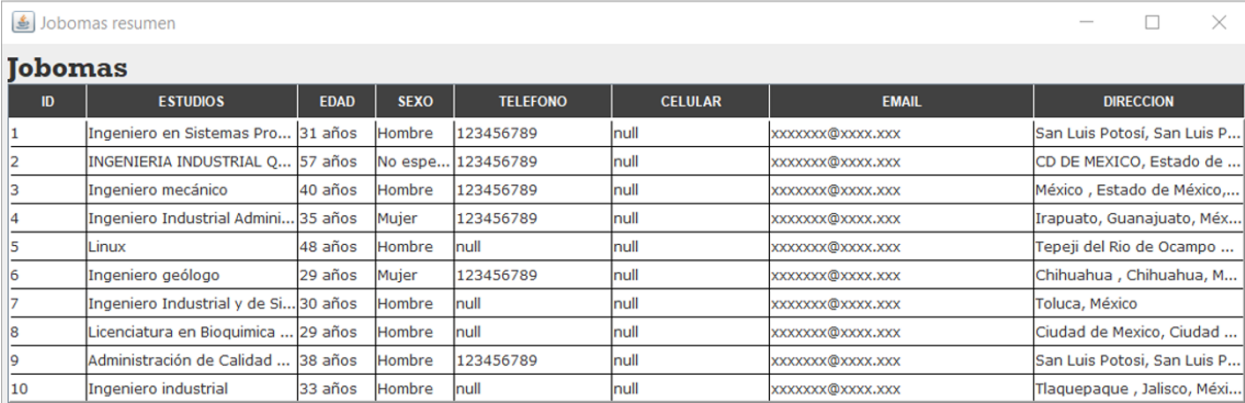
5.4.1 Resumen Jobomas

La ventana *Jobomas resumen*, figura 5.37, contiene información de los perfiles obtenidos en el sitio *web* Jobomas. Los datos que muestra corresponden a:

- Id: Número identificador del registro.
- Estudios: Último grado de estudios.
- Edad: Se expresa en años.
- Sexo: Hombre o mujer.
- Teléfono: Número telefónico.
- Celular: Número celular
- E-mail: Correo electrónico, tiene una máscara dado que se está usando una cuenta gratuita.

5. EJECUCIÓN DE LA APLICACIÓN

- Dirección: Ubicación geográfica donde vive el candidato.



ID	ESTUDIOS	EDAD	SEXO	TELEFONO	CELULAR	EMAIL	DIRECCION
1	Ingeniero en Sistemas Pro...	31 años	Hombre	123456789	null	xxxxxxx@xxxx.xxx	San Luis Potosí, San Luis P...
2	INGENIERIA INDUSTRIAL Q...	57 años	No espe...	123456789	null	xxxxxxx@xxxx.xxx	CD DE MEXICO, Estado de ...
3	Ingeniero mecánico	40 años	Hombre	123456789	null	xxxxxxx@xxxx.xxx	México , Estado de México,...
4	Ingeniero Industrial Admini...	35 años	Mujer	123456789	null	xxxxxxx@xxxx.xxx	Irapuato, Guanajuato, Méx...
5	Linux	48 años	Hombre	null	null	xxxxxxx@xxxx.xxx	Tepeji del Rio de Ocampo ...
6	Ingeniero geólogo	29 años	Mujer	123456789	null	xxxxxxx@xxxx.xxx	Chihuahua , Chihuahua, M...
7	Ingeniero Industrial y de Si...	30 años	Hombre	null	null	xxxxxxx@xxxx.xxx	Toluca, México
8	Licenciatura en Bioquímica ...	29 años	Hombre	null	null	xxxxxxx@xxxx.xxx	Ciudad de Mexico, Ciudad ...
9	Administración de Calidad ...	38 años	Hombre	123456789	null	xxxxxxx@xxxx.xxx	San Luis Potosi, San Luis P...
10	Ingeniero industrial	33 años	Hombre	null	null	xxxxxxx@xxxx.xxx	Tlaquepaque , Jalisco, Méxi...

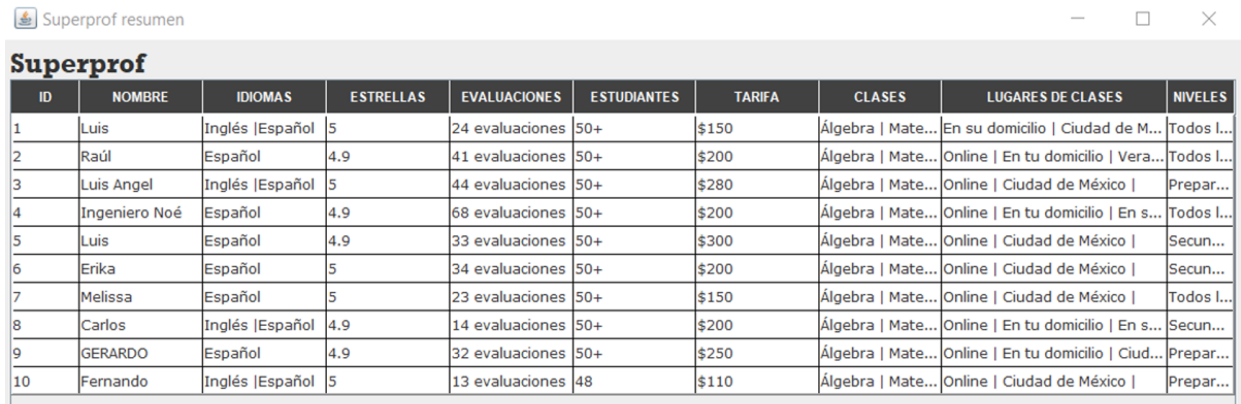
Figura 5.37: Tabla Jobomas.

5.4.2 Resumen Superprof

La ventana *Superprof resumen*, figura 5.38, contiene información de los perfiles obtenidos en el sitio *web* Superprof. Los datos que muestra corresponden a:

- Id: Identificador del registro.
- Nombre: Nombre del profesor.
- Idiomas: Idiomas que maneja el docente.
- Estrellas: Puntuación que tiene el profesor en el sitio *web*, 5 es la más alta.
- Evaluaciones: Número de evaluaciones realizadas por los alumnos hacia el profesor.
- Estudiantes: Número de estudiantes que tiene el profesor.
- Tarifa: Costo de la clase por hora, está en peso mexicano.
- Clases: Materias que imparte.
- Lugares de Clases: Forma en la que la clase es impartida
- Niveles: Perfiles de estudiantes a los que va dirigida la clase.

5.5 Generación de reportes



The screenshot shows a window titled "Superprof resumen" with a table of user profiles. The table has 10 rows and 10 columns. The columns are: ID, NOMBRE, IDIOMAS, ESTRELLAS, EVALUACIONES, ESTUDIANTES, TARIFA, CLASES, LUGARES DE CLASES, and NIVELES. The data is as follows:

ID	NOMBRE	IDIOMAS	ESTRELLAS	EVALUACIONES	ESTUDIANTES	TARIFA	CLASES	LUGARES DE CLASES	NIVELES
1	Luis	Inglés Español	5	24 evaluaciones	50+	\$150	Álgebra Mate...	En su domicilio Ciudad de M...	Todos I...
2	Raúl	Español	4.9	41 evaluaciones	50+	\$200	Álgebra Mate...	Online En tu domicilio Vera...	Todos I...
3	Luis Angel	Inglés Español	5	44 evaluaciones	50+	\$280	Álgebra Mate...	Online Ciudad de México	Prepar...
4	Ingeniero Noé	Español	4.9	68 evaluaciones	50+	\$200	Álgebra Mate...	Online En tu domicilio En s...	Todos I...
5	Luis	Español	4.9	33 evaluaciones	50+	\$300	Álgebra Mate...	Online Ciudad de México	Secun...
6	Erika	Español	5	34 evaluaciones	50+	\$200	Álgebra Mate...	Online Ciudad de México	Secun...
7	Melissa	Español	5	23 evaluaciones	50+	\$150	Álgebra Mate...	Online Ciudad de México	Todos I...
8	Carlos	Inglés Español	4.9	14 evaluaciones	50+	\$200	Álgebra Mate...	Online En tu domicilio En s...	Secun...
9	GERARDO	Español	4.9	32 evaluaciones	50+	\$250	Álgebra Mate...	Online En tu domicilio Ciud...	Prepar...
10	Fernando	Inglés Español	5	13 evaluaciones	48	\$110	Álgebra Mate...	Online Ciudad de México	Prepar...

Figura 5.38: Tabla Superprof.

5.5 Generación de reportes

La ventana mostrada en la figura 5.39 ofrece dos opciones al usuario:

- Crear Hoja: Generar las hojas de cálculo con los reportes de la información obtenida de internet. Al seleccionar esta opción se observa en la consola, figura 5.40, que se han creado dos ficheros, uno corresponde a Jobomas y otro a Superprof. El nombre del fichero está dado por un timestamp(conjunto de fecha y hora) del momento en el que el archivo es creado.
- Terminar: Termina el proceso y ejecución de la aplicación.

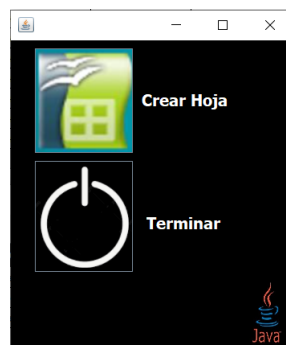


Figura 5.39: Crear hoja de cálculo y terminar.

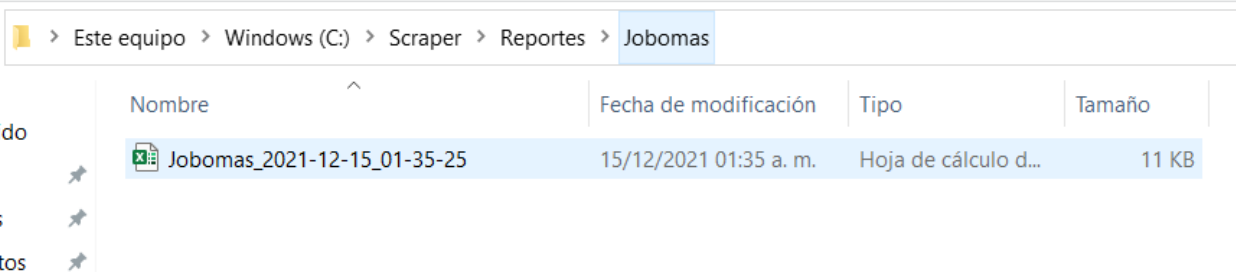
5. EJECUCIÓN DE LA APLICACIÓN

```
C:\Windows\System32\cmd.exe - java -jar tesisLeonel-1.0.0.jar
Registrando perfil 6 Jobomas
Registrando perfil 7 Jobomas
Registrando perfil 8 Jobomas
Registrando perfil 9 Jobomas
Registrando perfil 10 Jobomas
Cerrando página Jobomas
Fichero 2021-12-15_01-35-25 creado
Fichero 2021-12-15_01-35-25 creado
```

Figura 5.40: Traza de fichero creado.

5.5.1 Reporte Jobomas

La hoja de cálculo, figura 5.41, con la información del reporte de Jobomas se guarda en la ruta: *C>Scraper>Reportes>Jobomas* con el formato de nombre *Jobomas + timeStamp*. Al abrir este archivo podemos visualizar todos los datos obtenidos del sitio *web* Jobomas de forma ordenada.

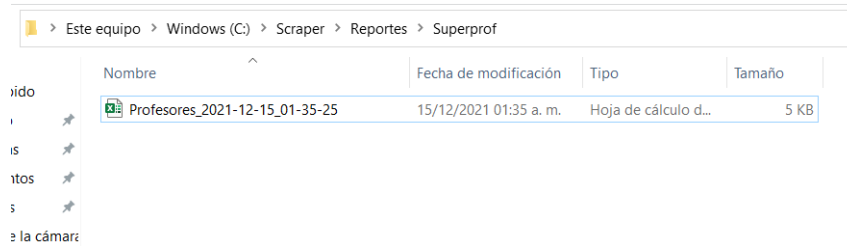


Nombre	Fecha de modificación	Tipo	Tamaño
Jobomas_2021-12-15_01-35-25	15/12/2021 01:35 a. m.	Hoja de cálculo d...	11 KB

Figura 5.41: Archivo Jobomas.

5.5.2 Reporte Superprof

La hoja de cálculo, figura 5.42, con la información del reporte de Superprof se guarda en la ruta *C>Scraper>Reportes>Superprof* con el formato de nombre *Profesores + timeStamp*. Al abrir este archivo podemos visualizar todos los datos obtenidos del sitio *web* Jobomas de forma ordenada.



Nombre	Fecha de modificación	Tipo	Tamaño
Profesores_2021-12-15_01-35-25	15/12/2021 01:35 a. m.	Hoja de cálculo d...	5 KB

Figura 5.42: Archivo Profesores.

5.6 Terminar

Al elegir la opción Terminar mostrada en la figura 5.39 accedemos al último módulo de nuestro programa en donde se indica el cierre de las ventanas de las imágenes 5.37 y 5.38 con el resumen de los perfiles obtenidos en los sitios *web* mediante el uso de la técnica de *Web Scraper*. Otra forma de terminar el programa es cerrando las ventanas de forma normal y manual.

5.6.1 Fin de ejecución

Al finalizar la aplicación mediante el botón *Terminar*, las ventanas se cierran en automático y en la traza de la ejecución de observa el mensaje *Ejecución terminada*, terminando así la ejecución de la aplicación de forma correcta.

```

C:\Windows\System32\cmd.exe - java -jar tesisLeonel-1.0.0.jar
Registrando perfil 5 Superprof
test insertando registros
Registrando perfil 6 Superprof
Registrando perfil 3 Jobomas
test insertando registros
Registrando perfil 7 Superprof
test insertando registros
Registrando perfil 8 Superprof
Registrando perfil 4 Jobomas
test insertando registros
Registrando perfil 9 Superprof
test insertando registros
Registrando perfil 10 Superprof
Cerrando página Superprof
Registrando perfil 5 Jobomas
Registrando perfil 6 Jobomas
Registrando perfil 7 Jobomas
Registrando perfil 8 Jobomas
Registrando perfil 9 Jobomas
Registrando perfil 10 Jobomas
Cerrando página Jobomas
Fichero 2021-12-15_01-35-25 creado
Fichero 2021-12-15_01-35-25 creado
Ejecución terminada

```

Figura 5.43: Traza de término de la ejecución de la aplicación.

5. EJECUCIÓN DE LA APLICACIÓN

Pruebas y resultados

6.1 Revisión de registros en Base de Datos

Después de terminar de ejecutar la aplicación búsqueda de perfiles en diferentes sitios *web*, la información recabada durante el proceso es almacenada en el motor de las tablas de la Base de Datos Oracle. En este capítulo se mostrarán los registros que quedaron en la base de datos relacional, así como los datos de las hojas de cálculo generadas.

6.1.1 Resultados

En la misma computadora en la que se llevó a cabo la ejecución de la presente aplicación *web* se realizaron varias ejecuciones de forma manual con un sujeto de pruebas. Recuperar información de un profesor en la página Superprof llevó en promedio un tiempo de 02:13 minutos, siendo que para 10 perfiles tomó 22:10 minutos. Obtener y almacenar la información de un perfil de Jobomas de forma manual varía demasiado dada la estructura de la página de internet y dado el volumen de información que el aspirante ingresa al sitio, siendo un promedio de 7 minutos por perfil en recopilar, dar orden y sentido a la información recabada. De esta manera obtener y ordenar la información de 10 perfiles lleva 70:00 minutos. El total de tiempo en las pruebas para el caso manual es de 92:10 minutos y al hacer uso de la aplicación *web* nos tomó 04:00 minutos. Cabe mencionar que, adicionalmente, se almacenaron los datos en base de datos utilizando el *scraper*, mientras que de forma manual solo se almacena la información en un reporte en una hoja de cálculo. En la figura 6.1 se muestran de forma gráfica los tiempos que duró la búsqueda de perfiles en Jobomas

6. PRUEBAS Y RESULTADOS

y en Superprof de forma manual, el total de tiempo en búsqueda manual y el tiempo de ejecución haciendo uso de la aplicación.

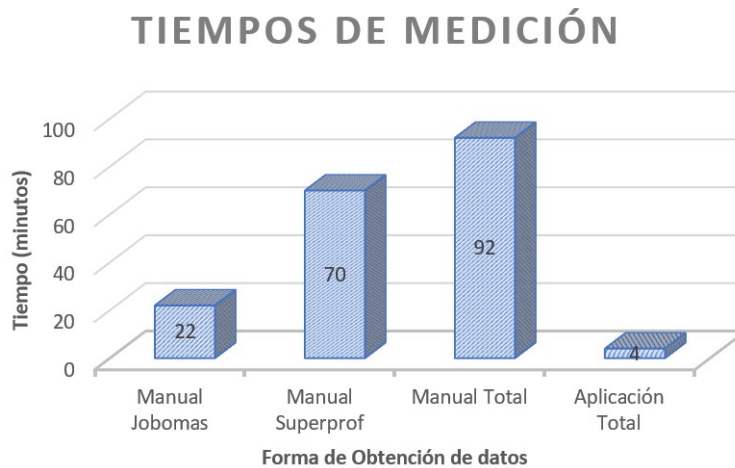


Figura 6.1: Tiempos de obtención de datos.

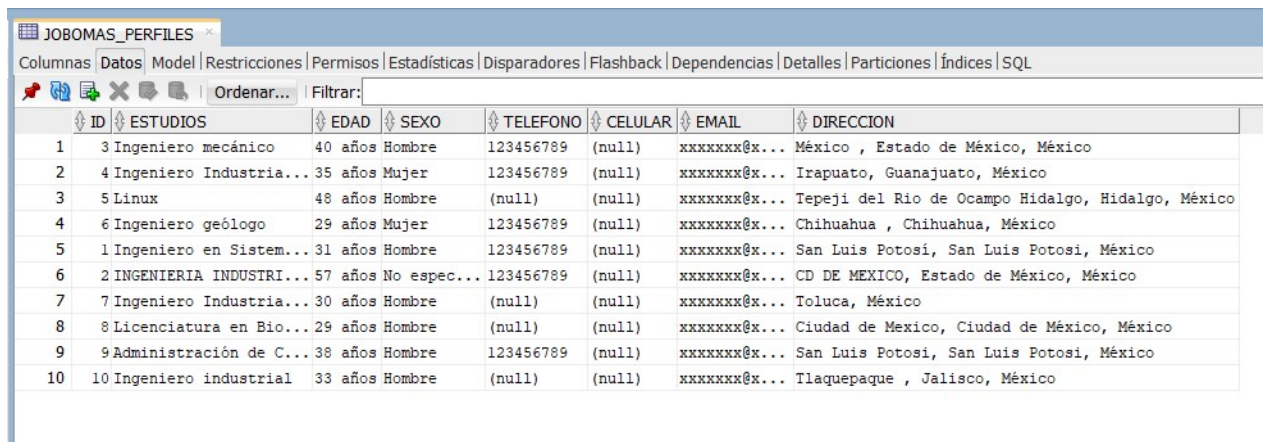
6.1.2 Tablas de base de datos de Jobomas

Las siguientes tablas contienen los datos de un perfil de Jobomas por ID generado, el campo ID relaciona todas las tablas, estas son:

- Tabla Perfiles, ver figura 6.2: Contiene información personal del candidato, sus campos son ESTUDIOS, EDAD, SEXO, TELEFONO, CELULAR, EMAIL y DIRECCION.
- Tabla Otros Conocimientos, ver figura 6.3: Lista la información adicional que el aspirante informa. Los campos que contiene son OTROS.
- Tabla Educación, ver figura 6.4: Almacena información académica del candidato. Los campos que contiene son TITULO, ESCUELA y FECHA.
- Tabla Experiencia Laboral, ver figura 6.5: Guarda información de antiguos empleos del candidato. Los campos que contiene son PUESTO, EMPRESA, FECHA y DESCRIPCION.
- Tabla Idiomas, ver figura 6.6: Corresponde a los datos de los idiomas que maneja el candidato. Los campos que la componen son NIVEL e IDIOMAS.

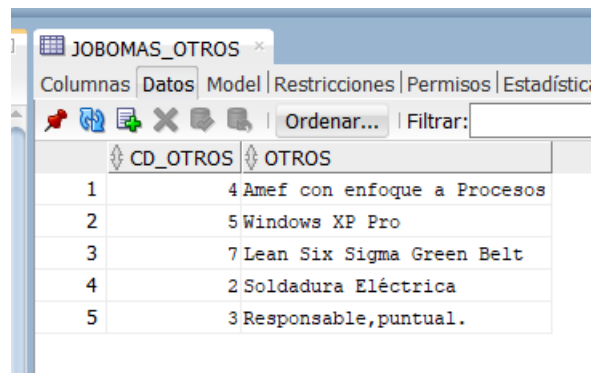
6.1 Revisión de registros en Base de Datos

- Tabla Informática, ver figura 6.7: Enumera los conocimientos del candidato en el uso de herramientas de software. Los campos que forman esta tabla son NIVEL y TECNOLOGÍA.



ID	ESTUDIOS	EDAD	SEXO	TELEFONO	CELULAR	EMAIL	DIRECCION
1	3 Ingeniero mecánico	40 años	Hombre	123456789	(null)	xxxxxxx@x...	México , Estado de México, México
2	4 Ingeniero Industria...	35 años	Mujer	123456789	(null)	xxxxxxx@x...	Irapuato, Guanajuato, México
3	5 Linux	48 años	Hombre	(null)	(null)	xxxxxxx@x...	Tepeji del Rio de Ocampo Hidalgo, Hidalgo, México
4	6 Ingeniero geólogo	29 años	Mujer	123456789	(null)	xxxxxxx@x...	Chihuahua , Chihuahua, México
5	1 Ingeniero en Sistem...	31 años	Hombre	123456789	(null)	xxxxxxx@x...	San Luis Potosí, San Luis Potosi, México
6	2 INGENIERIA INDUSTRI...	57 años	No espec...	123456789	(null)	xxxxxxx@x...	CD DE MEXICO, Estado de México, México
7	7 Ingeniero Industria...	30 años	Hombre	(null)	(null)	xxxxxxx@x...	Toluca, México
8	8 Licenciatura en Bio...	29 años	Hombre	(null)	(null)	xxxxxxx@x...	Ciudad de Mexico, Ciudad de México, México
9	9 Administración de C...	38 años	Hombre	123456789	(null)	xxxxxxx@x...	San Luis Potosi, San Luis Potosi, México
10	10 Ingeniero industrial	33 años	Hombre	(null)	(null)	xxxxxxx@x...	Tlaquepaque , Jalisco, México

Figura 6.2: Tabla Perfiles



CD_OTROS	OTROS
1	4 Amef con enfoque a Procesos
2	5 Windows XP Pro
3	7 Lean Six Sigma Green Belt
4	2 Soldadura Eléctrica
5	3 Responsable,puntual.

Figura 6.3: Tabla Otros Conocimientos

6. PRUEBAS Y RESULTADOS

CD...	TITULO	ESCUELA	FECHA
1	3 Ingeniero mecánico	Universidad Tecnológica de México	Actualmente estudiando
2	4 Ingeniero Industrial Administrativo	México	2009-09-01 - 2013-04-30
3	4 Tec. en contabilidad	CBTis 65	2001-08-01 - 2004-04-30
4	5 Linux	Instituto Linux	2020-01-01 - 2020-01-01
5	5 Certificación	Certificate Cybersecurity Essent...	2019-01-01 - 2019-01-01
6	5 Certificación Liderazgo en PMP	TenStep	2018-01-11 - 2018-02-07
7	5 Master in Marketing Digital	Instituto Internacional Español ...	2018-01-01 - 2019-01-01
8	5 Certificate The Key Elements to R...	TENSTEP	2017-04-01 - 2017-08-01
9	5 Certificación LdT	Cisco Systems	2017-01-04 - 2017-02-15
10	1 Ingeniero en Sistemas Productivos	Universidad Tecnológica de San L...	2011-01-01 - 2015-01-01
11	2 INGENIERIA INDUSTRIAL QUIMICA	INSTITUTO TECNOLOGICO DE MINATITLAN	1985-01-01 - 1988-01-01
12	5 Certificate Agile on Enterprise I...	TENSTEP	2017-01-01 - 2017-02-01
13	5 Certificación en IoT Linux Devnet	México	2017-01-01 - 2017-01-01
14	5 Certificación en Comercio Electro...	México	2017-01-01 - 2017-01-01
15	5 Certificación en PMO	TENSTEP-Chile	2017-01-01 - 2017-01-01
16	5 Certificación : Blandas	TENSTEP-Ecuador	2016-11-01 - 2016-12-01
17	5 Certificación en Lean Startup	Instituto de Economía Internacional	2016-11-01 - 2016-12-01
18	5 Certificación : Gestion del Crono...	TENSTEP	2016-09-01 - 2016-09-01
19	5 Certificación : Presentaciones Ef...	TENSTEP	2016-08-01 - 2016-08-01
20	5 Certificación en Marketing Digital	Instituto Internacional Español ...	2016-08-01 - 2016-11-01
21	5 Certificación PMP	TenStep	2016-07-11 - 2018-03-07
22	5 Certificación en PmP	PMI	2016-07-06 - 2018-05-31
23	5 Certificación CCNA CISCO	CISCO SYTEMS	2015-08-01 - 2015-10-01
24	5 Certificación : Community Manager	SCREENIE	2015-02-01 - 2015-03-01
25	5 Certificación en Practicas de Rec...	SCREENIE	2015-02-01 - 2015-03-01
26	5 Certificación NOM 020	STPS	2014-08-01 - 2014-10-01
27	5 ING EN REDES COMPUTACIONALES, TEL...	UNIVERSIDAD ETAC ALIAT UNIVERSID...	2013-08-01 - 2014-08-01
28	5 Certificación: Fundamentos de Red...	Cisco Sytems	2012-02-01 - 2012-02-01
29	5 DIPLOMADO FACTURACION ELECTRONICA	ESCUELA BANCARIA Y COMERCIAL (EBC)	2010-02-01 - 2011-02-01
30	5 LICENCIATURA EN REDES, TELECOMUNI...	UNIVERSIDAD ETAC ALIAT UNIVERSID...	2009-08-01 - 2012-12-01
31	5 ING ELECTRONICA AUTOMOTRIZ	GENERAL MOTORS	2001-01-01 - 2002-12-01
32	5 REPARACION DE MINICOMPONENTES Y S...	ESCUELA MEXICANA DE ELECTRICIDAD...	1993-01-01 - 1994-12-01
33	5 INGENIERIA ELECTRONICA	UNIVERSIDAD TECNOLOGICA DE MEXICO	1992-08-01 - 1996-12-01
34	5 TECNICO EN ELECTROMECHANICA	CBTIS No. 200	1989-09-01 - 1992-07-01
35	6 Ingeniero geólogo	Universidad autónoma de chihuahua	2011-01-20 - 2015-12-20
36	7 Ingeniero Industrial y de Sistemas	Universidad politécnica del vall...	2009-01-01 - 2015-01-01
37	8 Licenciatura en Bioquímica Diagno...	México	Actualmente estudiando
38	9 Administración de Calidad y manuf...	Instituto del Potosí	2016-01-01 - 2018-01-01
39	9 Ingeniero industrial en manufactura	Instituto del Potosí	2012-01-01 - 2016-01-01
40	10 Ingeniero industrial	UABC	2011-01-01 - 2015-01-01

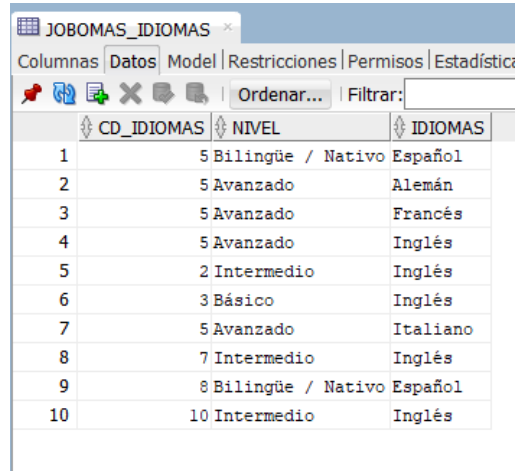
Figura 6.4: Tabla Educación

6.1 Revisión de registros en Base de Datos

CD..	PUESTO	EMPRESA	FECHA	DESCRIPCION
1	1 Inspector de calidad	Servicios Profesionales de Alt...	Jul-2019 Ene-2021 (1 año, 6 meses)	Dossier de calidad, reportes de
2	2 Asesor Técnico Comercial	Basser Solutions	Feb-2020 Actualmente trabajando	Asesor Técnico Comercial en la e
3	2 Asesor tecnico comercial	Soluciones Aplicadas al Manten...	May-2017 Feb-2020 (2 años, 9 meses)	Responsable del centro de servic
4	2 (null)	(null)	SOBRESTANTE MECANICO/CIVIL	Arendal S. De R.L. De C.V. May-2
5	2 Ejecutivo de zona (Zo...	Mantenimiento Quimico Industrial	Feb-2001 Dic-2008 (7 años, 10 meses)	Contratar distribuidores en la r
6	2 (null)	REPRESENTANTE TECNICO	HILTI MEXICANA, S, A. DE C.V	Dic-1998 Feb-2001 (2 años, 2 m
7	2 SUPERVISOR DE VENTAS ...	AGA DE MEXICO, S.A. DE C.V. (A...	Feb-1992 Jul-1997 (5 años, 5 meses)	DURANTE EL TIEMPO QUE ESTUVE LA
8	2	POR ULTIMO SE ME DIO LA OPORTU...	(null)	INGENIERO DE SERVICIOS. SUMINIST
9	3 Técnico e mantenimiento	Delfort	Nov-2019 Actualmente trabajando	Instalación y configuración de e
10	3 Encargado de mantenim...	Aceros de precision (A.P.S.A)	Oct-2016 Actualmente trabajando	Mantenimiento preventivo a inyec
11	3 Supervisor	British Federal México	Sep-2017 Ago-2019 (1 año, 11 meses)	Supervisión de personal mantenim
12	3 Facilitador de manten...	MAHLE, Componentes de Motores,...	Dic-2015 Jul-2016 (7 meses)	Planeación y administración de l
13	3 Técnico electromecanico	HILMEX SA. DE CV	May-2015 Dic-2015 (7 meses)	Mantenimiento preventivo y corre
14	3 Técnico de servicio	Middleby World Wide	Sep-2007 Jun-2014 (6 años, 9 meses)	Mantenimientos preventivos y cor
15	4 Team leader	Fukoku.	Mar-2018 Ago-2020 (2 años, 5 meses)	Evaluaciones y cambios de nivele
16	4 Supervisor	Grupo Maas.	Oct-2017 Mar-2018 (5 meses)	Manejo de personal. Elaboración
17	4 (null)	Analista de Relaciones Laborales	Continental	May-2013 Nov-2013 (6 meses) Cu
18	4 (null)	(null)	(null)	Auxiliar de Analista de Relacion
19	4 (null)	(null)	(null)	Auditor se calidad General Motor
20	5 Studies Markets and I...	Survey International	Feb-2021 Actualmente trabajando	Studies Markets and Investment
21	4 (null)	(null)	(null)	Supervisor-lider SRG Global Mar-
22	4 (null)	Encargada	Ferreteria Escobedo	Ago-2014 Dic-2015 (1 año, 4 me
23	5 TrustPilot	Studies Markets and Investment	Oct-2020 Actualmente trabajando	Studies Markets and Investment
24	5 Panelist studies Mark...	Ipsos Bimsa Investigación de M...	Jul-2020 Actualmente trabajando	Studies Markets and Investment
25	5 Presidente de Capacit...	Capain(Capacitacion Industrial...	Nov-2011 Actualmente trabajando	Implementar Estudios de Trabajo
26	5 Asesor Studies Markets	Livra.com	Ene-2007 Actualmente trabajando	Studies Markets and Investment
27	5 Studies Markets	TrustPilot	Ene-2007 Oct-2015 (8 años, 9 meses)	Studies Markets
28	5 Gerente de Operaciones	JK Tyre	Jun-2001 Dic-2011 (10 años, 6 meses)	Gerente de Operaciones JKTornel(
29	5 Ingeniero de Pruebas	GENERAL MOTORS	Ene-2000 Dic-2002 (2 años, 11 meses)	INGENIERIA AUTOMOTRIZ EN ELECTRO
30	5 Técnico de Laboratorio B	Topp Inc	Ene-1997 Abr-2001 (4 años, 3 meses)	Pruebas de RF, Audiometría, Repa
31	5 Electrico de Control	Articulos Higiénicos	Ene-1996 Ago-1998 (2 años, 7 meses)	Experiencia en Mantenimiento Pre
32	7 Ingeniero nuevos proy...	HITACHI AUTOMOTIVE SYSTEMS	Sep-2019 Actualmente trabajando	? Diseño de Lay out nuevas línea
33	8 Ingeniero de Calidad ...	SAFE centro de mezclas, PISA F...	Sep-2019 Actualmente trabajando	Dentro de las actividades desemp
34	8 Químico analista de m...	Probiomed, S.A. De C.V.	Sep-2017 Ago-2019 (1 año, 11 meses)	Químico analista encargado de re
35	9 Coordinador ERP	ALPURA SA DE CV	Ene-2019 Actualmente trabajando	Coordinador de procesos (recibo,
36	5 Electrónico y Coaching	Cementos Cruz Azul S.C.L	Feb-1993 Dic-1997 (4 años, 10 meses)	Control Electrónico , variadores
37	6 Profesionista en desa...	Grupo mexico	Mar-2017 Jun-2017 (3 meses)	? Mina San Antonio, en Aquiles S

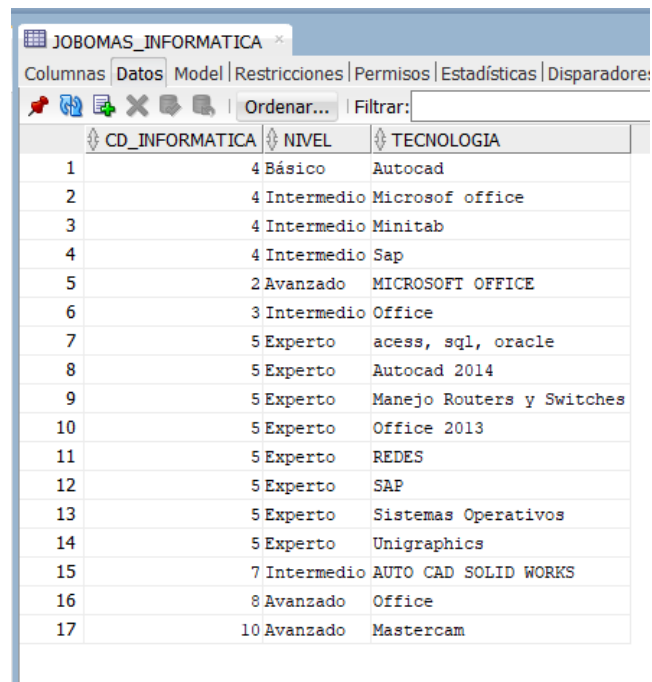
Figura 6.5: Tabla Experiencia Laboral

6. PRUEBAS Y RESULTADOS



CD_IDIOMAS	NIVEL	IDIOMAS
1	5 Bilingüe / Nativo	Español
2	5 Avanzado	Alemán
3	5 Avanzado	Francés
4	5 Avanzado	Inglés
5	2 Intermedio	Inglés
6	3 Básico	Inglés
7	5 Avanzado	Italiano
8	7 Intermedio	Inglés
9	8 Bilingüe / Nativo	Español
10	10 Intermedio	Inglés

Figura 6.6: Tabla Idiomas



CD_INFORMATICA	NIVEL	TECNOLOGIA
1	4 Básico	Autocad
2	4 Intermedio	Microsof office
3	4 Intermedio	Minitab
4	4 Intermedio	Sap
5	2 Avanzado	MICROSOFT OFFICE
6	3 Intermedio	Office
7	5 Experto	acess, sql, oracle
8	5 Experto	Autocad 2014
9	5 Experto	Manejo Routers y Switches
10	5 Experto	Office 2013
11	5 Experto	REDES
12	5 Experto	SAP
13	5 Experto	Sistemas Operativos
14	5 Experto	Unigraphics
15	7 Intermedio	AUTO CAD SOLID WORKS
16	8 Avanzado	Office
17	10 Avanzado	Mastercam

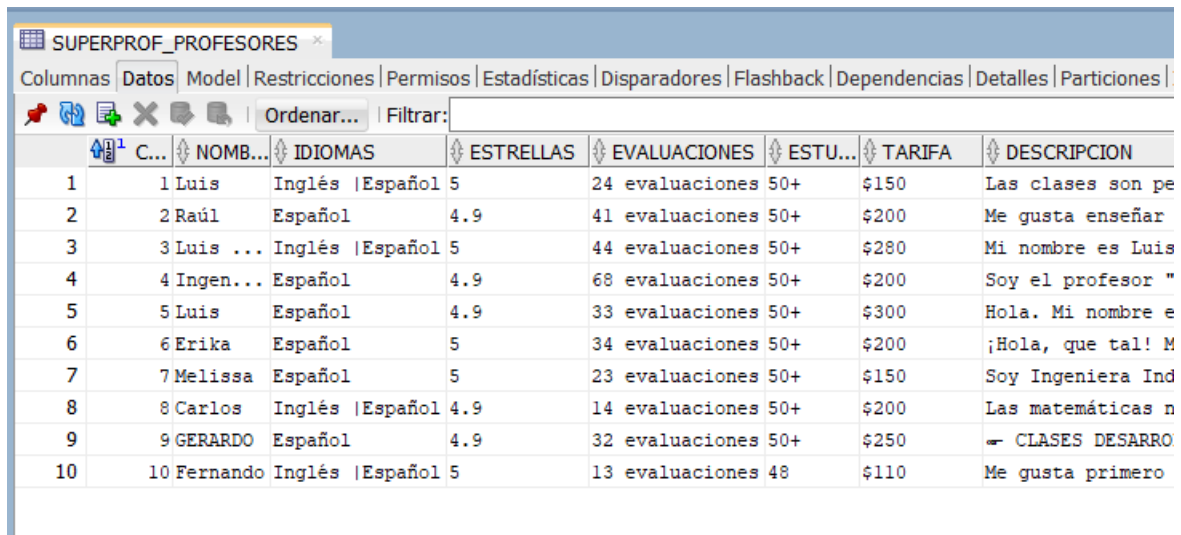
Figura 6.7: Tabla Informática

6.1 Revisión de registros en Base de Datos

6.1.3 Tablas de base de datos de Superprof

Las siguientes tablas contienen los datos de un perfil de profesor del sitio *web* Superprof por ID generado, el campo ID relaciona todas las tablas, estas son:

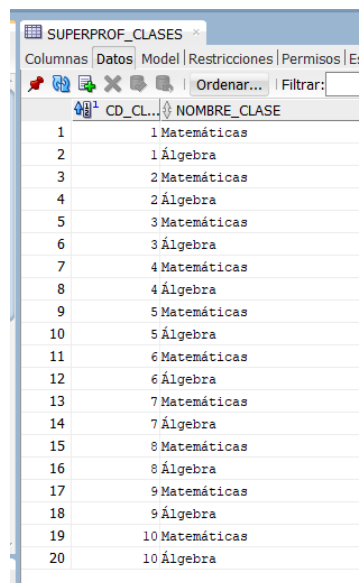
- Tabla Profesores, ver figura 6.8: Almacena la información básica de un profesor. Los campos que componen esta tabla son NOMBRE, IDIOMAS, ESTRELLAS, EVALUACIONES, ESTUDIANTES, TARIFA y DESCRIPCION.
- Tabla Clases, ver figura 6.9: Contiene las materias que imparte el profesor. La columna que contiene esta información es NOMBRE CLASE.
- Tabla Nivel, ver figura 6.10: Guarda el nivel al que está dirigida la materia del profesor. El campo que compone esta tabla es NOMBRE NIVEL.
- Tabla Lugar de Clase, ver figura 6.11: Corresponde a la información del medio por el cuál el profesor puede dar su clase, un dato interesante es que hay modalidad virtual y presencial. El campo que contiene esta información es NOMBRE LUGAR CLASE.



C...	NOMB...	IDIOMAS	ESTRELLAS	EVALUACIONES	ESTU...	TARIFA	DESCRIPCION
1	Luis	Inglés Español	5	24 evaluaciones	50+	\$150	Las clases son pe
2	Raúl	Español	4.9	41 evaluaciones	50+	\$200	Me gusta enseñar
3	Luis ...	Inglés Español	5	44 evaluaciones	50+	\$280	Mi nombre es Luis
4	Ingen...	Español	4.9	68 evaluaciones	50+	\$200	Soy el profesor "
5	Luis	Español	4.9	33 evaluaciones	50+	\$300	Hola. Mi nombre e
6	Erika	Español	5	34 evaluaciones	50+	\$200	¡Hola, que tal! M
7	Melissa	Español	5	23 evaluaciones	50+	\$150	Soy Ingeniera Ind
8	Carlos	Inglés Español	4.9	14 evaluaciones	50+	\$200	Las matemáticas n
9	GERARDO	Español	4.9	32 evaluaciones	50+	\$250	← CLASES DESARRO
10	Fernando	Inglés Español	5	13 evaluaciones	48	\$110	Me gusta primero

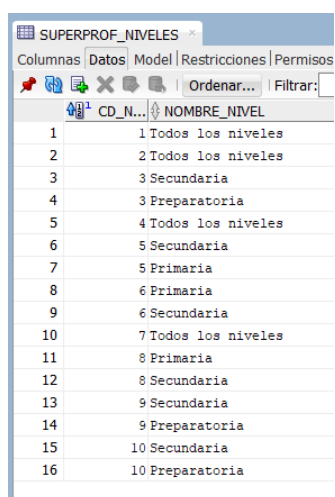
Figura 6.8: Tabla Profesores

6. PRUEBAS Y RESULTADOS



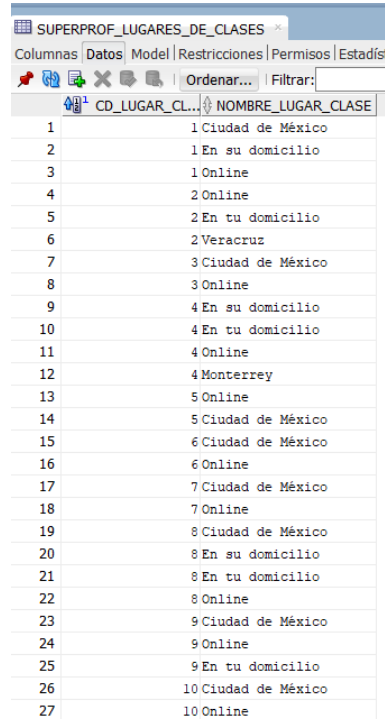
CD_CLASE	NOMBRE_CLASE
1	1 Matemáticas
2	1 Álgebra
3	2 Matemáticas
4	2 Álgebra
5	3 Matemáticas
6	3 Álgebra
7	4 Matemáticas
8	4 Álgebra
9	5 Matemáticas
10	5 Álgebra
11	6 Matemáticas
12	6 Álgebra
13	7 Matemáticas
14	7 Álgebra
15	8 Matemáticas
16	8 Álgebra
17	9 Matemáticas
18	9 Álgebra
19	10 Matemáticas
20	10 Álgebra

Figura 6.9: Tabla Clases



CD_NIVEL	NOMBRE_NIVEL
1	1 Todos los niveles
2	2 Todos los niveles
3	3 Secundaria
4	3 Preparatoria
5	4 Todos los niveles
6	5 Secundaria
7	5 Primaria
8	6 Primaria
9	6 Secundaria
10	7 Todos los niveles
11	8 Primaria
12	8 Secundaria
13	9 Secundaria
14	9 Preparatoria
15	10 Secundaria
16	10 Preparatoria

Figura 6.10: Tabla Nivel



	CD_LUGAR_CL...	NOMBRE_LUGAR_CLASE
1	1	Ciudad de México
2	1	En su domicilio
3	1	Online
4	2	Online
5	2	En tu domicilio
6	2	Veracruz
7	3	Ciudad de México
8	3	Online
9	4	En su domicilio
10	4	En tu domicilio
11	4	Online
12	4	Monterrey
13	5	Online
14	5	Ciudad de México
15	6	Ciudad de México
16	6	Online
17	7	Ciudad de México
18	7	Online
19	8	Ciudad de México
20	8	En su domicilio
21	8	En tu domicilio
22	8	Online
23	9	Ciudad de México
24	9	Online
25	9	En tu domicilio
26	10	Ciudad de México
27	10	Online

Figura 6.11: Tabla Lugares de Clase

6.2 Revisión de reportes

Con la finalidad de poder consultar los datos obtenidos en la ejecución del *web scraper*, los datos se guardan en hojas de cálculo. Por cada sitio *web* se crea un archivo, para leer estos archivos usaremos el software de uso libre *OpenOffice Calc*.

6.2.1 Hoja de cálculo Jobomas

Estas son las hojas con los datos almacenados en el archivo de hoja de cálculo correspondiente a Jobomas.

- Tabla Perfiles, ver figura 6.12: Estos datos corresponden con la tabla de la figura 6.2
- . Tabla Educación, ver figura 6.13: Estos datos corresponden con la tabla de la figura 6.4.

6. PRUEBAS Y RESULTADOS

- Tabla Experiencia Laboral, ver figura 6.14: Estos datos corresponden a la tabla de la figura 6.5.
- Tabla Idiomas, ver figura 6.15: Estos datos corresponden a la tabla de la figura 6.6.
- Tabla Informática, ver figura 6.16: Estos datos corresponden a la tabla de la figura 6.7.
- Tabla Otros Conocimientos, ver figura 6.17: Estos datos corresponden a la tabla de la figura 6.3.

1	ID	ESTUDIO	EDAD	SEXO	TELEFONO	MOBILE	EMAIL	LOCATION
2	1	Ingeniero	31 años	Hombre	123456789		xxxxxxx@	México , Estado de México, México
3	2	INGENIE	57 años	No espec	123456789		xxxxxxx@	Irapuato, Guanajuato, México
4	3	Ingeniero	40 años	Hombre	123456789		xxxxxxx@	Tepeji del Rio de Ocampo Hidalgo, Hidalgo, México
5	4	Ingeniero	35 años	Mujer	123456789		xxxxxxx@	Chihuahua , Chihuahua, México
6	5	Linux	48 años	Hombre			xxxxxxx@	San Luis Potosí, San Luis Potosí, México
7	6	Ingeniero	29 años	Mujer	123456789		xxxxxxx@	CD DE MEXICO, Estado de México, México
8	7	Ingeniero	30 años	Hombre			xxxxxxx@	Toluca, México
9	8	Licenciat	29 años	Hombre			xxxxxxx@	Ciudad de Mexico, Ciudad de México, México
10	9	Administ	38 años	Hombre	123456789		xxxxxxx@	San Luis Potosí, San Luis Potosí, México
11	10	Ingeniero	33 años	Hombre			xxxxxxx@	Tlaquepaque , Jalisco, México
12								

Figura 6.12: Hoja Candidatos

1	CD_IDIOMAS	NIVEL	IDIOMAS
2	5	Bilingüe	→Español
3	5	Avanzado	Alemán
4	5	Avanzado	Francés
5	5	Avanzado	Inglés
6	2	Intermedi	→Inglés
7	3	Básico	Inglés
8	5	Avanzado	Italiano
9	7	Intermedi	→Inglés
10	8	Bilingüe	→Español
11	10	Intermedi	→Inglés

Figura 6.13: Hoja Idiomas

6.2 Revisión de reportes

A	B	C	D	E	F	G	H
4	Tec. en c	CBTis 65	2001-08-01	2004-04-30			
5	Linux	Instituto I	2020-01-01	2020-01-01			
5	Certificac	Certificat	2019-01-01	2019-01-01			
5	Certificac	TenStep	2018-01-11	2018-02-07			
5	Master in	Instituto I	2018-01-01	2019-01-01			
5	Certificat	TESTEP	2017-04-01	2017-08-01			
5	Certificac	Cisco Sy	2017-01-04	2017-02-15			
1	Ingeniero	Universid	2011-01-01	2015-01-01			
2	INGENIE	INSTITUT	1985-01-01	1988-01-01			
5	Certificat	TENSTEP	2017-01-01	2017-02-01			
5	Certificac	México	2017-01-01	2017-01-01			
5	Certificac	México	2017-01-01	2017-01-01			
5	Certificac	TENSTEP	2017-01-01	2017-01-01			
5	Certificac	TENSTEP	2016-11-01	2016-12-01			
5	Certificac	Instituto	2016-11-01	2016-12-01			
5	Certificac	TENSTEP	2016-09-01	2016-09-01			
5	Certificac	TENSTEP	2016-08-01	2016-08-01			
5	Certificac	Instituto I	2016-08-01	2016-11-01			
5	Certificac	TenStep	2016-07-11	2018-03-07			
5	Certificac	PMI	2016-07-06	2018-05-31			
5	Certificac	CISCO S	2015-08-01	2015-10-01			
5	Certificac	SCREEN	2015-02-01	2015-03-01			
5	Certificac	SCREEN	2015-02-01	2015-03-01			
5	Certificac	STPS	2014-08-01	2014-10-01			
5	ING EN	UNIVERS	2013-08-01	2014-08-01			
5	Certificac	Cisco Sy	2012-02-01	2012-02-01			
5	DIPLOMA	ESCUELA	2010-02-01	2011-02-01			
5	LICENCIA	UNIVERS	2009-08-01	2012-12-01			
5	ING ELE	GENERA	2001-01-01	2002-12-01			
5	REPARA	ESCUELA	1993-01-01	1994-12-01			
5	INGENIE	UNIVERS	1992-08-01	1996-12-01			
5	TECNICO	CBTIS N	1989-09-01	1992-07-01			
6	Ingeniero	Universid	2011-01-20	2015-12-20			
7	Ingeniero	Universid	2009-01-01	2015-01-01			
8	Licenciat	México	Actualmente estudiando				
9	Administ	Instituto	2016-01-01	2018-01-01			
9	Ingeniero	Instituto	2012-01-01	2016-01-01			
10	Ingeniero	UABC	2011-01-01	2015-01-01			

Figura 6.14: Hoja Educación

6. PRUEBAS Y RESULTADOS

A	B	C	D	E	F	G	H	I	J	K	L	M	N
CD_EXP	PUESTO	EMPRESA	FECHA	DESCRIPTION									
1	Inspector	Servicios	Jul-2019	Dossier de calidad, reportes de pruebas de campo, realización de pruebas y manejo de instrumento									
2	Asesor T	Basser S	Feb-2020	Asesor Técnico Comercial en la especialidad de Torres de Enfriamiento y Tratamiento Químico para									
2	Asesor t	Solucion	May-2017	Responsable del centro de servicio para dar respuesta de mantenimiento preventivo, correctivo, capa									
2				SOBRES									
2	Ejecutivo	Mantenim	Feb-2001	Arendal S. De R.L. De C.V. May-2014 Dic-2016 (2 años, 7 meses) Coordinando equipos multidisci									
2				Contratar distribuidores en la región comprendida en los estados de Puebla, Veracruz, Tabasco, Oaxa									
2				REPRE									
2				HILTI ME									
2	SUPERV	AGA DE	Feb-1992	DURANTE EL TIEMPO QUE ESTUVE LABORANDO CON ESTA EMPRESA, EN MI PRIMERA ET									
2				POR ULTIMO SE M									
2				INGENIERO DE SERVICIOS. SUMINISTROS E INGRIA. DE CONTROL, S.A. DE C.V. Ene-1991 F									
3	Técnico	Delfort	Nov-2019	Instalación y configuración de elementos mecánicos y electrónicos. Diagnóstico de averías de los e									
3	Encargad	Aceros d	Oct-2016	Mantenimiento preventivo a inyectará, Caldera, autoclave, compresores granalladora, esmeriles, herra									
3	Supervis	British F	Sep-2017	Supervisión de personal mantenimiento preventivo y correctivo a robots, reparación de fallas mecánic									
3	Facilitad	MAHLE	Dic-2015	Planeación y administración de los mantenimientos predictivos, preventivos, correctivos, elaboració									
3	Técnico	HILMEX	May-2019	Mantenimiento preventivo y correctivos, de los equipos textiles, en hidráulica mecánica, neumático. l									
3	Técnico	Middleby	Sep-2007	Mantenimientos preventivos y correctivos, asesoramiento de los equipos al personal, instalación de l									
4	Team lea	Fukoku	Mar-2018	Evaluaciones y cambios de niveles. Aplicación de RIT. Generación de indicadores y reportes. Inducc									
4	Supervis	Grupo M	Oct-2017	Manejo de personal. Elaboración de reportes de incidentes e investigación de ellos. Registro de inci									
4		Analista	Continent	May-2013 Nov-2013 (6 meses) Cursos de capacitación para personal sindicalizado y de confianza.									
4				Auxiliar de Analista de Relaciones Laborales General Motors Feb-2012 May-2013 (1 año, 3 meses									
4				Auditor se calidad General Motors Feb-2009 Feb-2012 (3 años) Soporte en auditorías internas. Cor									
5	Studies	Survey In	Feb-2021	Studies Markets and Investment									
4				Supervisor-lider SRG Global Mar-2016 Sep-2017 (1 año, 6 meses) Manejo de personal. Aplicación									
4		Encargad	Ferreteria	Ago-2014 Dic-2015 (1 año, 4 meses) Control de inventarios. Manejo de caja. Atención a clientes. C									
5	TrustPilot	Studies	Oct-2020	Studies Markets and Investment									
5	Panelist	Ipsos Bin	Jul-2020	Studies Markets and Investment									
5	President	Capain(C	Nov-2011	Implementar Estudios de Trabajo a las Empresas que quieran actualizarse y continuar mejorando sus									
5	Asesor S	Livra.com	Ene-2007	Studies Markets and Investment									
5	Studies	TrustPilot	Ene-2007	Studies Markets									
5	Gerente	JK Tyre	Jun-2001	Gerente de Operaciones JKTorne(Llantas); Manejo de Supervision de Personal (sindicalizado y no si									
5	Ingeniero	GENERA	Ene-2000	INGENIERIA AUTOMOTRIZ EN ELECTRONICA AUTOMOTRIZ. PROYECTO APLICADO A LA INDU									
5	Técnico	Topp Inc	Ene-1997	Pruebas de RF, Audiometría, Reparación de Tarjetas de Teléfonos Alámbricos e Inalámbricos, Celul									
5	Electrico	Articulos	Ene-1996	Experiencia en Mantenimiento Preventivo y Correctivo, simbología Europea y Americana, hidráulica									
7	Ingeniero	HITACHI	Sep-2019	Diseño de Lay out nuevas líneas de Producción ? Contacto con Proveedores, para Nuevas instalac									
8	Ingeniero	SAFE ce	Sep-2019	Dentro de las actividades desempeñadas se encuentra la verificación de procesos para la preparaci									
8	Quimico	Probiome	Sep-2017	Químico analista encargado de realizar análisis de materias primas y producto terminado; realizandi									
9	Coordina	ALPURA	Ene-2019	Coordinador de procesos (recibo, almacén y embarques)									
5	Electróni	Cemento	Feb-1993	Control Electrónico, variadores de velocidad, encoders, motores paso a paso, sistemas de adquisic									
6	Profesion	Grupo m	Mar-2017	? Mina San Antonio, en Aquiles Serdán, Chihuahua me encontraba en el área de Geología en el pue									

Figura 6.15: Hoja Experiencia Laboral

6.2 Revisión de reportes

Jobomas_2021-12-15_01-35-25.xlsx - OpenOffice Calc

Archivo Editar Ver Insertar Formato Herramientas Datos

Arial 10 N C

A1 \sum = CD_INFORMATICA

	A	B	C	D	E
1	CD INFO	NIVEL	TECNOLOGIA		
2	4	Básico	Autocad		
3	4	Intermedi	Microsof office		
4	4	Intermedi	Minitab		
5	4	Intermedi	Sap		
6	2	Avanzado	MICROSOFT OFFICE		
7	3	Intermedi	Office		
8	5	Experto	access, sql, oracle		
9	5	Experto	Autocad 2014		
10	5	Experto	Manejo Routers y Switches		
11	5	Experto	Office 2013		
12	5	Experto	REDES		
13	5	Experto	SAP		
14	5	Experto	Sistemas Operativos		
15	5	Experto	Unigraphics		
16	7	Intermedi	AUTO CAD SOLID WORKS		
17	8	Avanzado	Office		
18	10	Avanzado	Mastercam		

Figura 6.16: Hoja Informática

Jobomas_2021-12-15_01-35-25.xlsx - OpenOffice Calc

Archivo Editar Ver Insertar Formato Herramientas Datos Vent

Arial 10 N C S

A1 \sum = CD_OTROS

	A	B	C	D	E	F
1	CD OTRO	OTROS				
2	4	Amaf con enfoque a Procesos				
3	5	Windows XP Pro				
4	7	Lean Six Sigma Green Belt				
5	2	Soldadura Eléctrica				
6	3	Responsable,puntual.				
7						

Figura 6.17: Hoja Otros

6. PRUEBAS Y RESULTADOS

6.2.2 Hoja de cálculo Superprof

Los datos almacenados en base de datos del sitio *web* Superprof se almacenan en una sola hoja, estos corresponden a los datos de nombre, idiomas, estrellas, evaluaciones, estudiantes, tarifa, clases, lugares de clases y niveles.

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID	NOMBRE	IDIOMAS	ESTRELLAS	EVALUACIONES	ESTUDIANTES	TARIFA	CLASES	LUGARES	NIVELES		
2	1	Luis	Inglés	5	24 evaluaciones	50+	\$150	Álgebra	En su domicilio	Todos los niveles		
3	2	Raúl	Español	4.9	41 evaluaciones	50+	\$200	Álgebra	Online	Todos los niveles		
4	3	Luis Angulo	Inglés	5	44 evaluaciones	50+	\$280	Álgebra	Online	Preparatoria	Secundaria	
5	4	Ingeniero	Español	4.9	68 evaluaciones	50+	\$200	Álgebra	Online	Todos los niveles		
6	5	Luis	Español	4.9	33 evaluaciones	50+	\$300	Álgebra	Online	Secundaria	Primaria	
7	6	Erika	Español	5	34 evaluaciones	50+	\$200	Álgebra	Online	Secundaria	Primaria	
8	7	Melissa	Español	5	23 evaluaciones	50+	\$150	Álgebra	Online	Todos los niveles		
9	8	Carlos	Inglés	4.9	14 evaluaciones	50+	\$200	Álgebra	Online	Secundaria	Primaria	
10	9	GERARDO	Español	4.9	32 evaluaciones	50+	\$250	Álgebra	Online	Preparatoria	Secundaria	
11	10	Fernando	Inglés	5	13 evaluaciones	48	\$110	Álgebra	Online	Preparatoria	Secundaria	
12												
13												

Figura 6.18: Hoja Superprof

Conclusiones y trabajo futuro

La internet aloja gran cantidad de sitios *web* que a su vez contienen grandes volúmenes de datos. *Web scraping* es una técnica automatizada de extracción de datos de sitios *web* y en conjunto con el uso de tecnologías *Open Source* se pueden crear programas que minimizan los costos de producción y la contratación de personal, ya que no existe el pago por licencias para el uso de tecnologías, entornos integrados o algún otro *software*; y, la ejecución de las tareas que demandan un uso prolongado de un equipo de cómputo por parte de capital humano se vuelven automáticas por parte de la ejecución de un programa.

El presente trabajo de ingeniería aplicada se trabajó con tecnologías y *softwares* de tipo *Open Source*, se desarrolló una aplicación *web* que buscó, presentó y almacenó datos de forma automática de perfiles de diferentes sitios de internet. La búsqueda en las páginas de internet se realizó de forma concurrente con la ayuda de los Hilos que ofrece Java y permitió minimizar el tiempo de ejecución y aprovechamiento del procesador. Se determina que el tiempo de ejecución de *Web scraping* en cada sitio *web* es diferente porque cada página tiene una estructura definida y sus datos en ella se encuentran estructurados o no estructurados. Esto influye en los tiempos de recolección de datos y depende del analista programador determinar el algoritmo de extracción de datos que permita la obtención de información. Este proceso de recolección de información ayuda a mitigar los errores que un humano pueda cometer al realizar esta tarea repetitiva y manual durante un tiempo prolongado.

Emular el comportamiento humano de forma automática mediante la técnica de *Web scraping* ahorra tiempo y dinero. Algunas tareas que se automatizaron en esta aplicación son el manejo de credenciales, ingresar al sitio *web*, buscar un perfil específico, obtener cada uno de los datos de interés, almacenar estos datos en Base

7. CONCLUSIONES Y TRABAJO FUTURO

de Datos y generar un reporte con el resumen de los datos en una hoja de cálculo. Los datos obtenidos son precisos y fiables debido a que son acorde a lo registrado en la *web*. Extraer la información de interés de internet no es suficiente, es importante agrupar y almacenar estos datos para que así puedan ser consultados y analizados en el momento que se requiera. El uso de hojas de cálculo le facilita al usuario la lectura de datos, ya que estas ofrecen una interfaz amigable y el uso de base de datos relacionales da persistencia y sentido a la relación de la información obtenida.

El programa automatiza la búsqueda, guardado y presentación de perfiles de profesionales alojados en diferentes sitios *web* enfocados en el almacenamiento y publicación de hojas de vida de diversos tipos. Aprovecha la gran cantidad de datos que se encuentran en internet, los acomoda y los expone de una manera amigable al usuario en reportes específicos por cada fuente consultada. Para ejecutar la aplicación del *Web Scraper* basta con lanzar la aplicación desde un equipo ambientado correctamente que cuente con conexión estable a internet sin necesidad de depender de grandes cantidades de capital humano. El uso de un *Web Scraper* ahorra tiempo, esfuerzo, recursos monetarios y de personal a una organización porque automatiza actividades que pueden ser programables y ejecutadas desde un ordenador.

En la misma computadora en la que se llevó a cabo la ejecución de la presente aplicación *web* se realizaron varias ejecuciones de forma manual con un sujeto de pruebas siendo siempre la búsqueda automatizada en bolsas de trabajo online con técnicas de *Web Scraping* la más rápida. Recuperar información de un profesor en la página Superprof llevó en promedio un tiempo de 02:13 minutos, siendo que para 10 perfiles tomó 22:10 minutos. Obtener y almacenar la información de un perfil de Jobomas de forma manual varía demasiado según la estructura de la página de internet y el volumen de información que el aspirante ingresa al sitio, siendo un promedio de 7 minutos por perfil en recopilar, dar orden y sentido a la información recabada. De esta manera, obtener y ordenar la información de 10 perfiles fue de 70:00 minutos. El total de tiempo en las pruebas para el caso manual es de 92:10 minutos y al hacer uso de la aplicación *web* nos tomó 04:00 minutos. Cabe mencionar que adicionalmente se almacenaron los datos en base de datos utilizando el *scraper* mientras que de forma manual no es así, solo se almacena la información en un reporte en una hoja de cálculo.

Este programa es una primera versión del *web scraper*, se ilustra el funcionamiento del mismo, por eso se programaron tareas repetitivas que una persona realiza de forma mecánica en 2 sitios *web* diferentes, Jobomas y Superprof; siendo el programa 23 veces más rápido que una persona en la búsqueda, recuperación y almacenamiento de información. También mitiga el error humano en cuanto a integridad de datos refiere y realiza otras actividades como el guardado de información en una base de

datos relacional. Se cubrió la necesidad de buscar perfiles de profesores y vacantes de empleo de forma automatizada en internet, de esta manera la aplicación tiene una funcionalidad específica y no existe una igual. Esta versión es un prototipo y, por ello, se piensa ampliar el número de sitios *web* de los cuales obtiene información. El patrón de diseño Modelo Vista Controlador le permite a la aplicación adaptarse a otro tipo de tecnologías.

Es importante mantener la compatibilidad entre el navegador web y el Chrome Driver, de otro modo, se pueden presentar errores de ejecución. Normalmente el navegador web se actualiza de manera automática y al mismo tiempo en el sitio oficial del Chrome Driver se libera una versión compatible o se puede desactivar esta actualización para mantener una versión estática en el navegador web. Una desventaja notoria en el uso de Web Scraping es que en caso de que la estructura del sitio web cambie, será necesario ajustar el algoritmo de extracción de información a esta nueva distribución porque el objeto o clase al que se desea acceder seguramente habrá cambiado y retornará un valor nulo. La estructura del código con la que está diseñada la aplicación facilita la identificación del error, así como también la corrección del código del programa.

Como trabajo futuro se plantea tener una aplicación con más funcionalidades y que ofrezca una interfaz que haga uso de una tecnología destinada a la capa de usuario. Se debe hacer un análisis de los sitios *web* con mayor publicación de perfiles profesionales para ser agregados al buscador, esto incrementará el número de sitios *web* que puedan ser consultados. Estas nuevas consultas demandan la creación de una tabla en base de datos dedicada a alojar credenciales y contraseñas de acceso a estas nuevas páginas. Gracias a que se trabajó con el Modelo Vista Controlador en la codificación del programa, se propone agregar un marco de trabajo que esté enfocado a la interfaz de usuario de la aplicación, el uso de esta nueva tecnología mejorará el aspecto visual que el programa ofrece al usuario sin afectaciones a los otros módulos del programa. Esta primera versión es un prototipo, por ello, hay funcionalidad que podría ir añadiéndose en versiones futuras. Un ejemplo es agregar algunos controles y herramientas de acceso al medio, como botones con la funcionalidad de parar y reanudar la búsqueda de perfiles, esto cuando el navegador *web* está en ejecución; así como una función que evite la información duplicada en base de datos.

7. CONCLUSIONES Y TRABAJO FUTURO

Bibliografía

- [1] M. Trigás Gallego, “Metodología scrum,” 2012.
- [2] Atlantico, “La transformación digital de américa latina - 2021,” 20.
- [3] F. Flores Díaz, *Virtualización de Servidores Sobre Sistemas Linux Mediante la Tecnología KVM*. [Informe de Actividades Profesionales, Facultad de Ingeniería, Universidad Nacional Autónoma de México] Repositorio Facultad de Ingeniería. <http://132.248.52.100:8080/xmlui/handle/132.248.52.100/17023>, 2019.
- [4] G. M. Medina Sanes, “Definición y evaluación de un modelo de calidad en uso para un portal de bolsa de trabajo utilizando la norma iso/iec 25000,” 2014.
- [5] C. F. ACOSTA ACOSTA and A. E. ORAMAS GUTIERREZ, “Diseño de una bolsa de trabajo virtual para la facultad de ciencias matematicas y fisicas de la universidad de guayaquil, ecuador,” Ph.D. dissertation, UNIVERSIDAD DE GUAYAQUIL FACULTAD CIENCIAS MATEMÁTICAS Y FÍSICAS CARRERA . . . , 2016.
- [6] LinkedIn, “Acerca de linkedin,” 2021.
- [7] Infojobs, “Sobre infojobs,” 2019.
- [8] Superprof, “¿quiénes somos?” 2021.
- [9] Computrabajo, “¿quiénes somos?” 20.
- [10] N. L. Vicente Stenhouse, “Habscraper: herramienta automatizada para la extracción de datos con web scraping.”
- [11] J. López, “Web scraping,” 2018.

BIBLIOGRAFÍA

- [12] P. Dolz del Castellar González-Cela, “Diseño y desarrollo de una aplicación web scraping de extracción de enlaces de tiendas web,” 2019.
- [13] V. Krotov and L. Silva, “Legality and ethics of web scraping,” 2018.
- [14] D. Murillo and D. Saavedra, “Web scraping de los perfiles y publicaciones de una afiliación en google scholar utilizando aplicaciones web e implementando un algoritmo en r,” in *Memorias de Congresos UTP*, 2017, pp. 8–15.
- [15] F. S. Campos-Sánchez, P. Chillón *et al.*, “Patrones espaciales de localización de alojamientos turísticos mediante análisis complementarios e integrados: Sig, sintaxis espacial y web-scraping,” *Boletín de la Asociación de Geógrafos Españoles*, no. 84, 2020.
- [16] A. M. Herrera, R. Aranda, and F. J. H. López, “Herramientas web scraping para la adquisición de datos turísticos,” *KIKAME*, vol. 9, no. 9, pp. 100–111, 2020.
- [17] F. J. C. Sierra, *Programación orientada a objetos con C++*. Grupo Editorial RA-MA, 2007, vol. 3.
- [18] L. Gervais, *Aprender la programación orientada a objetos con el lenguaje C*. Ediciones ENI, 2016.
- [19] A. E. C. González, “Programación orientada a objetos,” 2018.
- [20] D. J. Barnes, M. Kölling, and B. I. Brenta, *Programación orientada a objetos con Java*. Pearson Educación, 2007.
- [21] H. Mundo, “Programación orientada a objetos,” 1994.
- [22] G. Booch, “Análisis y diseño orientado a objetos,” *EU: Addison-Wesley*, 1996.
- [23] I. G. Pérez, A. J. S. Martín, and D. V. Hernández, “Java threads (hilos en java),” 2002.
- [24] F. D. Muñoz Escoí, E. Argente Villaplana, A. R. Espinosa Minguet, P. Galdamez Saiz, A. M. García Fornes, R. d. Juan Marín, and J. S. Sendra Roig, “Concurrencia y sistemas distribuidos,” *Colección Académica. Editorial UPV*, 2013.
- [25] B. E. Mazón Olivo, J. J. Cartuche Calva, V. L. Chimarro Chipantiza, and W. B. Rivas Asanza, “Fundamentos de programación orientada a objetos en java.” 2015.

- [26] J. A. Macías Guerrero, “Estudio comparativo de lenguajes de programación orientados a objetos basado en las facilidades para implantar el concepto de objeto.”
- [27] G. S. Bermúdez, I. M. Jiménez, and L. R. Rodríguez, “Uso de patrones de diseño: Un caso práctico,” *Ingeniería. Revista de la Universidad de Costa Rica*, vol. 22, no. 2, pp. 45–59, 2012.
- [28] J. A. I. Astorga, J. L. O. Tirado, R. U. R. Ramírez, J. C. L. Mendoza, and A. P. Garzón, “Clasificación de los patrones de diseño idóneos en programación android,” *Revista Digital de Tecnologías Informáticas y Sistemas*, vol. 3, no. 3, 2019.
- [29] A. Cortez and C. Naveda, “Especificación de patrones de diseño de comportamiento,” *Segundo Encuentro Investigadores de la RADU-2.014*, 2014.
- [30] C. A. Guerrero, J. M. Suárez, and L. E. Gutiérrez, “Patrones de diseño gof (the gang of four) en el contexto de procesos de desarrollo de aplicaciones orientadas a la web,” *Información tecnológica*, vol. 24, no. 3, pp. 103–114, 2013.
- [31] E. Amodeo, “Principios de diseño de apis rest,” *Canada: Leanpub*, 2013.
- [32] A. Arsaute, F. A. Zorzán, M. Daniele, and P. Martellotto, “Generación automática del modelo de diseño desde el modelo de análisis a través de reglas qvt,” in *XVIII Congreso Argentino de Ciencias de la Computación*, 2013.
- [33] G. Lopez Vasquez, “Desarrollo e implementacion de un sistema basado en patrones de diseno para optimizar los servicios de pesaje en la empresa balanzas vegasystems sac,” 2020.
- [34] Y. D. González and Y. F. Romero, “Patrón modelo-vista-controlador.” *Telemática*, vol. 11, no. 1, pp. 47–57, 2012.
- [35] J. G. C. Sagredo, A. T. Espinosa, M. M. Reyes, and M. d. L. L. García, “Automatización de la codificación del patrón modelo vista controlador (mvc) en proyectos orientados a la web,” *CIENCIA ergo-sum, Revista Científica Multidisciplinaria de Prospectiva*, vol. 19, no. 3, pp. 239–250, 2012.
- [36] P. S. Lagos, “Commonkads y el lenguaje de modelado unificado ce uml,” *Ingeniería informática*, no. 8, p. 4, 2002.

BIBLIOGRAFÍA

- [37] C. L. Vidal, R. F. Schmal, S. Rivero, and R. H. Villarroel, “Extensión del diagrama de secuencias uml (lenguaje de modelado unificado) para el modelado orientado a aspectos,” *Información tecnológica*, vol. 23, no. 6, pp. 51–62, 2012.
- [38] A. García Holgado and F. J. García Peñalvo, “Patrón arquitectónico para la definición de ecosistemas de elearning basados en desarrollos open source,” 2014.
- [39] D. Spano, “El open source como facilitador del open access,” in *II Encuentro Iberoamericano de editores científicos*, 2010, p. 16.
- [40] H. F. Chila Choque and J. Z. Mamani Arenas, “Sistema integrado basado en tecnologías open source hibernate, spring y jsf 2.0 aplicando patrones data access object y business object para la gestión de la información de coopain cabana-puno,” 2014.
- [41] R. Magallón Rosa, “Tecnologías cívicas y participación ciudadana,” 2014.
- [42] I. Corporation, “Aplicaciones java se y java ee,” 2021.
- [43] P. Corcuera, “Introducción a la tecnología java,” *Universidad de Cantabria*, pp. 7–8, 2017.
- [44] T. Groussard, *JAVA 7: Los fundamentos del lenguaje Java*. Ediciones Eni, 2012.
- [45] E. Bascón Pantoja, “El patrón de diseño modelo-vista-controlador (mvc) y su implementación en java swing,” *Acta Nova*, vol. 2, no. 4, pp. 493–507, 2004.
- [46] T. A. S. Foundation, “Welcome to apache maven,” 2021.
- [47] Selenium, “Documentación,” 2021.
- [48] Oracle, “Desarrolle aplicaciones java con oracle database,” 2021.
- [49] —, “¿qué es una base de datos?” 2021.
- [50] —, “Oracle sql developer,” 2021.
- [51] Atlassian, “¿por qué github?” 2021.
- [52] Git, “Sobre,” 2021.
- [53] GitHub, “¿por qué github?” 2021.

- [54] J. C. Santamarta Cerezal, R. Tomás, J. Rodríguez-Martín, L. E. Hernández Gutiérrez, M. Cano, A. Riquelme *et al.*, “Optimización y eficiencia en los cálculos de ingeniería mediante hojas de cálculo,” 2018.
- [55] R. G. Figueroa, C. J. Solís, and A. A. Cabrera, “Metodologías tradicionales vs. metodologías ágiles,” *Universidad Técnica Particular de Loja, Escuela de Ciencias de la Computación*, vol. 9, no. 1, pp. 1–10, 2008.
- [56] C. Rodríguez and R. Dorado, “¿ por qué implementar scrum?” *Revista Ontare*, vol. 3, no. 1, pp. 125–144, 2015.
- [57] H. A. G. Ávila, “Evaluación de las características de un sistema de información con base en la norma iso/iec 9126-1,” *Signos: Investigación en sistemas de gestión*, vol. 5, no. 2, pp. 33–44, 2013.
- [58] P. L. Alfonzo and S. I. Mariño, “Los estándares internacionales y su importancia para la industria del software,” 2013.
- [59] A. C. Pasini, S. Esponda, M. Boracchia, and P. M. Pesado, “Q-scrum: una fusión de scrum y el estándar iso/iec 29110,” in *XVIII Congreso Argentino de Ciencias de la Computación*, 2013.
- [60] J. A. Calvo-Manzano, J. Garzás, M. Piattini, F. J. Pino, J. Salillas, and J. L. Sánchez, “Perfiles del ciclo de vida del software para pequeñas empresas: los informes técnicos iso/iec 29110,” *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, vol. 4, no. 2, pp. 96–108, 2008.