

**TEAM-BE: PLUGIN DE ECLIPSE PARA LA CONFORMACIÓN DE GRUPOS EN
ESCENARIOS DE APRENDIZAJE COLABORATIVO**

JUAN PABLO BOTINA CARLOSAMA

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERIA
DEPARTAMENTO DE SISTEMAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2021**

**TEAM-BE: PLUGIN DE ECLIPSE PARA LA FORMACIÓN DE GRUPOS EN
ESCENARIOS DE APRENDIZAJE COLABORATIVO**

JUAN PABLO BOTINA CARLOSAMA

**Trabajo de grado presentado como requisito parcial para optar al título de
Ingeniero de Sistemas**

Director

Ing. OSCAR REVELO SANCHEZ, PhD(C)

Co-Director

Ing. JESÚS INSUASTI PORTILLA, PhD

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERIA
DEPARTAMENTO DE SISTEMAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2021**

NOTA DE RESPONSABILIDAD

“Las ideas y conclusiones aportadas en este Trabajo de Grado son responsabilidad de los autores.”

Artículo 1, Acuerdo No. 324 de octubre 11 de 1966, emanado del Honorable Consejo Directivo de la Universidad de Nariño.

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Artículo 13, Acuerdo No. 005 de 2010 emanado por el Honorable Consejo Académico de la Universidad de Nariño

Nota de aceptación:

Firma presidente del Jurado

Firma del Jurado

Firma del Jurado

San Juan de Pasto, octubre de 2021

AGRADECIMIENTOS

A mi madre por el apoyo constante e incondicional tanto material como emocional a lo largo de toda mi vida, especialmente en los momentos difíciles.

A mi director de tesis: Ing. Oscar revelo y mi codirector: Ing. Jesús Insuasti por compartir su experiencia para este proyecto.

A los docentes de Ingeniería de Sistemas por el conocimiento, enseñanzas, y apoyos brindados durante todo mi proceso como estudiante.

A la Universidad de Nariño por el conocimiento y las experiencias a las que tuve acceso al ser parte de esta gran institución.

A los docentes Marcela Guerrero y Javier Jiménez por su apoyo y colaboración para la etapa de pruebas en entorno real de este proyecto

A todos los amigos que directa o indirectamente me ayudaron a lograr mis objetivos durante mi etapa de formación y en este proyecto.

RESUMEN

La escritura colaborativa de código es una tendencia en el mundo del desarrollo actual y su aplicación en entornos educativos ayuda a motivar a nuevos programadores para adoptar una perspectiva de trabajo en equipo, desafortunadamente no es una técnica muy observada en ambientes académicos pues existen pocas herramientas completas que integren todas las funcionalidades necesarias para ello.

Collece 2.0 fue pensado para atender estas necesidades, es un *plugin* para el IDE Eclipse que proporciona un entorno de programación distribuido y colaborativo en tiempo real, escalable y preparado para incluir tecnologías de aprendizaje emergentes.

Por otra parte, la formación grupos es un paso clave para el desarrollo eficaz de actividades de trabajo colaborativo. La adecuada selección de individuos compatibles favorece la sinergia y la creación de ambientes que favorecen el compañerismo y el aprendizaje sólido tanto en escenarios profesionales como académicos. Lo anterior, acompañado estrategias de enseñanza adecuadas incide de forma positiva directamente en la calidad del código generado.

Siendo Collece 2.0 un escenario que potencia el trabajo colaborativo, un conjunto de grupos adecuadamente organizados permite un mejor aprovechamiento de sus funcionalidades. Una de las herramientas existentes que permite automatizar este proceso es la Librería TEAM-B desarrollada en la Universidad de Nariño la cual permite realizar una formación automática y homogénea de grupos basada en rasgos de personalidad mediante la aplicación de algoritmos genéticos para simular el proceso de selección del “más apto” teniendo en cuenta ciertos criterios de selección.

En el presente proyecto se integró exitosamente la librería en mención con Collece 2.0 para aprovechar los beneficios de la formación automática de grupos junto a todas las funcionalidades de un entorno de desarrollo de uso común entre estudiantes de programación como lo es el IDE Eclipse.

PALABRAS CLAVE: Formación de grupos, Trabajo Colaborativo, Escritura colaborativa de código, Entorno de desarrollo integrado, Desarrollo de *plugins*

ABSTRACT

Collaborative code writing is a trend in today's development world and its application in educational environments helps motivate new programmers to adopt a perspective of teamwork, unfortunately it is not a highly observed technique in academic environments as there are few complete tools that group all the required functionalities.

Collece 2.0 was designed to meet these needs, it is a plugin for the Eclipse IDE that provides a distributed and collaborative environment in real time, scalable and ready to include emerging learning technologies.

On the other hand, groups formation is a key step in effectively developing collaborative work activities. The appropriate selection of compatible individuals promotes synergy and creation of companionship environments in both professional and academic scenarios. This, together with appropriate teaching strategies, has a positive impact directly on the quality of the generated code.

With Collece 2.0 being a scenario that enhances collaborative work, a set of properly organized groups allows a better use of its functionalities. One of the existing tools that allows to automate this process is the TEAM-B Java library developed at University of Nariño which allows to perform an automatic and homogeneous group formation based on personality traits through the application of genetic algorithms to simulate the process of selecting the "fittest" considering certain selection criteria. The objective of this project is to integrate the mentioned library with Collece 2.0 to take advantage of the benefits of automatic group formation along with all the functionalities of a development environment known to most programming students.

In this project, the aforementioned library was successfully integrated with Collece 2.0 to take advantage of the benefits of automatic group formation in addition to all the existing functionalities of an Integrated Development Environment such as Eclipse.

KEYWORDS: Group formation, collaborative work, collaborative code writing, Integrated Development Environment (IDE), plugin development

TABLA DE CONTENIDO

pág.

INTRODUCCIÓN	16
1. PRESENTACIÓN DEL PROYECTO.....	18
1.1 TÍTULO	18
1.2 LÍNEA DE INVESTIGACIÓN.....	18
1.3 TEMA DE INVESTIGACIÓN	18
1.4 ÁREA DE INVESTIGACIÓN	18
1.5 PLANTEAMIENTO DEL PROBLEMA.....	19
1.6 FORMULACIÓN DEL PROBLEMA.....	22
1.6.1 Sistematización del problema.	22
1.7 OBJETIVOS DE LA INVESTIGACIÓN	22
1.7.1 Objetivo General.	22
1.7.2 Objetivos Específicos.....	22
1.8 JUSTIFICACIÓN.....	23
1.9 DELIMITACIÓN	25
1.10. ANTECEDENTES.....	25
1.10.1 Antecedentes a nivel internacional.	25
1.10.2 Antecedentes a nivel nacional.	27
1.10.3 Antecedentes a nivel regional.	28
1.11 SUPUESTOS TEÓRICOS DE LA INVESTIGACIÓN.....	28
1.11.1 Trabajo colaborativo.	28
1.11.2 Aprendizaje colaborativo.....	29

1.11.3 Formación de grupos.	30
1.11.4 Java.	31
1.11.5 IDE.	32
1.11.6 Eclipse IDE.	33
1.11.7 Collece 2.0.	34
1.11.8 GIT.	35
 2. METODOLOGÍA	37
2.1 MODALIDAD DEL PROYECTO.....	37
2.2 PARADIGMA DE INVESTIGACIÓN	37
2.3 TIPO DE INVESTIGACIÓN.....	37
2.4 DISEÑO DE LA INVESTIGACIÓN.....	37
2.4.1 Población.	39
2.4.2 Muestra.	40
2.4.3 Definición de tecnologías.	40
2.5 METODOLOGÍA DE INGENIERÍA DE SOFTWARE	41
2.5.1 Roles.....	44
2.5.2 Eventos.....	44
2.5.3 Artefactos.....	45
2.6 PRUEBAS DEL SOFTWARE.....	63
2.7 LENGUAJE DE MODELADO.....	63
 3. DESARROLLO DE LA INVESTIGACIÓN	64
3.1 ANÁLISIS, DISEÑO Y DESARROLLO	64
3.1.1 Objetivos del sistema.....	64

3.1.2 Requerimientos del sistema.....	65
3.1.3 Participantes.	74
3.1.4 Arquitectura del sistema.....	76
3.1.5 Casos de uso	84
3.2 DESCRIPCIÓN DE HERRAMIENTA DESARROLLADA	87
3.3 PRUEBAS DE SOFTWARE.....	92
4. RESULTADOS DE LA INVESTIGACIÓN	93
4.1 DISCUSIÓN DE RESULTADOS.....	94
4.2 VALIDACIÓN DE LA HERRAMIENTA.....	95
4.2.1 Detalles técnicos.....	95
4.2.2 Procedimiento.	96
4.2.3 Resultados encuesta posterior.....	97
5. CONCLUSIONES	103
6. RECOMENDACIONES.....	106
BIBLIOGRAFÍA.....	108
ANEXOS.....	113

LISTA DE FIGURAS

	pág.
Figura 1. Mapa mental del planteamiento del problema	21
Figura 2. Arquitectura cliente-servidor de Collece 2.0	35
Figura 3. Diseño de investigación	38
Figura 4. Proceso general de SCRUM.....	61
Figura 5. Tablero proceso SCRUM de TEAM-BE en Jira Software	62
Figura 6. Estructura general de Collece 2.0.....	76
Figura 7. Paquete networking de Collece 2.0	78
Figura 8. Paquete ui de Collece 2.0.....	79
Figura 9. Diagrama de clases del módulo TEAM-BE.....	82
Figura 10. Diagrama de secuencia de creación de sesiones.....	83
Figura 11. Diagrama de caso de uso general Collece 2.0	84
Figura 12. Diagrama caso de uso: CU-001 Creación de sesiones con TEAM-BE.....	86
Figura 13. Opciones de inicio de sesión de Collece 2.0	87
Figura 14. Asistente de gestión de sesiones de Collece 2.0.....	88
Figura 15. Ventana de creación de sesiones de Collece 2.0	89
Figura 16. Ejemplo de archivo de sesión yaml para Collece 2.0.....	89
Figura 17. Vista principal de herramienta TEAM-BE.....	90
Figura 18. Archivo CSV de ejemplo para TEAM-BE	91
Figura 19. Ejemplo de archivo PDF generado por TEAM-BE	91
Figura 20. Paquete de pruebas Collece 2.0.....	92
Figura 21. Ejecución de pruebas módulo TEAM-BE.....	92
Figura 22 . Sitio Web de TEAM-BE.....	93
Figura 23. Servidor remoto Collece 2.0	95
Figura 24. Análisis de encuesta - Gráfica de pregunta 1	97
Figura 25. Análisis de encuesta - Gráfica de pregunta 2	98
Figura 26. Análisis de encuesta - Gráfica de pregunta 3	99

Figura 27. Análisis de encuesta - Gráfica de pregunta 4	100
Figura 28. Análisis de encuesta - Gráfica de pregunta 5	102

LISTA DE TABLAS

pág.

Tabla 1. Distribución de participantes escenario de aplicación.....	40
Tabla 2. Personas y roles involucrados en el desarrollo del software.....	44
Tabla 3. Lista del producto (Product Backlog)	46
Tabla 4. Puntos de historia según su duración	47
Tabla 5. Sprint #1.....	48
Tabla 6. Sprint #2.....	49
Tabla 7. Sprint #3.....	50
Tabla 8. Sprint #4.....	51
Tabla 9. Historia de usuario #1	52
Tabla 10. Historia de usuario #2	53
Tabla 11. Historia de usuario #3	54
Tabla 12. Historia de usuario #4	56
Tabla 13. Historia de usuario #5	57
Tabla 14. Historia de usuario #6	58
Tabla 15. Historia de usuario #7	59
Tabla 16. Historia de usuario #8	60
Tabla 17. Objetivo OBJ-001 TEAM-BE	64
Tabla 18. Objetivo OBJ-002 TEAM-BE	65
Tabla 19. Requisito IRQ-001 TEAM-BE.....	66
Tabla 20. Requisito IRQ-002 TEAM-BE.....	67
Tabla 21. Requisito de interfaz INRQ-001 TEAM-BE	68
Tabla 22. Requisito de interfaz INRQ-002 TEAM-BE	69
Tabla 23. Requisito funcional TEAM-BE_RF_001	70
Tabla 24. Requisito funcional TEAM-BE_RF_002	71
Tabla 25. Requisito no funcional FNR-001	72
Tabla 26. Requisito no funcional FNR-002	73

Tabla 27. Requisito no funcional FNR-003	73
Tabla 28. Datos de participante 1	74
Tabla 29. Datos de participante 2	74
Tabla 30. Datos de participante 3	75
Tabla 31. Datos de participante 4	75
Tabla 32. Ejemplo de archivo CSV para la librería TEAM-B.....	81
Tabla 33. Descripción CU-001 Creación de sesiones con TEAM-BE	85

LISTA DE ANEXOS

pág.

Anexo 1. Formato de una historia de usuario en Jira Software	113
Anexo 2. Formato de un <i>Sprint</i> en Jira Software	114
Anexo 3. Plantilla para reuniones de retrospectiva	115
Anexo 4. Encuesta posterior a actividad de aplicación herramienta TEAM-BE ...	116
Anexo 5. Ejercicio de programación competitiva “ <i>Division of NLogonia</i> ”	118
Anexo 6. Manual de usuario herramienta TEAM-BE	120
Anexo 7. Resumen presentado en CACIED 2019	131
Anexo 8. Resumen presentado en EXPOPOSTER 2020	132
Anexo 9. Captura de pantalla registro DNDA	135
Anexo 10. Capturas de pantalla sitio web TEAM-BE	137

INTRODUCCIÓN

El ritmo acelerado de aparición y actualización de las tecnologías de desarrollo actuales requieren que su aprendizaje sea rápido y compatible con nuevos paradigmas de modo que los estudiantes puedan aprender y adaptarse a nuevos marcos de trabajo y reaccionar a los cambios de los mismos de forma oportuna, al mismo tiempo, las características de los proyectos de desarrollo actuales hacen necesario equipos más o menos numerosos cuyos miembros posean habilidades técnicas y sociales que se complementen con las de sus compañeros, aquí la academia juega un papel fundamental pues es donde el individuo adquiere la mayoría de estas habilidades y también las necesarias para adquirir otras junto con nuevo conocimiento.

En este contexto el aprendizaje colaborativo representa una alternativa interesante a la enseñanza tradicional, pues en este el aprendizaje pasa a ser un proceso social donde las interacciones entre estudiantes tienen el protagonismo y el profesor se convierte en un facilitador del aprendizaje, esto se apoya en la teoría del constructivismo social que sostiene que “las personas activamente construyen conocimiento mientras interactúan con su ambiente”¹. En este contexto se presenta Collece 2.0, una herramienta para apoyar el aprendizaje colaborativo de la programación de manera distribuida y en tiempo real, permite que varios participantes pueden trabajar en un problema determinado simultáneamente por medio de internet, además ofrece otras funcionalidades para apoyar el proceso tales como chat y bloqueo de regiones; por sus características Collece tiene potencial para usarse en entornos educativos y ayudar al desarrollo de las habilidades antes mencionadas.

Por otro lado, existe la librería de Java TEAM-B, desarrollada en la Universidad de Nariño, la cual permite formar grupos de forma automática por medio de algoritmos genéticos tomando como insumo diversas características cuantificables, por ejemplo, los rasgos de personalidad. El principal objetivo de este trabajo de grado fue integrar esta librería a Collece para favorecer su utilización en entornos de aprendizaje al automatizar un proceso que tradicionalmente se realiza de forma manual con la ventaja de tener grupos con individuos compatibles. Para lograrlo se definió un proceso de varias fases que empezó por la apropiación del conocimiento teórico y técnico necesario para el desarrollo y terminó en la creación de la herramienta y su documentación. Una vez terminado este proceso se aplicó la herramienta en un entorno educativo real con estudiantes de ingeniería de sistemas,

¹ MALDONADO PÉREZ, Marisabel. El Trabajo Colaborativo En El Aula Universitaria. En Laurus - Revista de Educación. Vol. 13 (2007); p. 263-278.

posteriormente se midió la pertinencia y aceptación de la misma por parte de los estudiantes participantes por medio de una encuesta.

Este documento ha sido organizado en capítulos que se describen a continuación: en el capítulo dos se abordan los conceptos teóricos involucrados en la investigación que se agrupan en los relacionados con el trabajo colaborativo y los conceptos técnicos como el lenguaje de programación y las herramientas de software utilizadas. En el siguiente capítulo se detalla la metodología utilizada con sus respectivas etapas, luego se describen los resultados obtenidos al aplicarla y por último se presentan las conclusiones del proceso y las recomendaciones para trabajos futuros.

1. PRESENTACIÓN DEL PROYECTO

1.1 TÍTULO

TEAM-BE: *plugin* de eclipse para la formación de grupos en escenarios de aprendizaje colaborativo

1.2 LÍNEA DE INVESTIGACIÓN

Las líneas de investigación donde se enfocará el trabajo de grado propuesto son:

- Procesos educativos apoyados por las nuevas tecnologías de la información y la comunicación
- Software y manejo de información (con enfoque en análisis y diseño de sistemas, ingeniería de software y programación)
- Análisis y diseño de sistemas, Programación e Ingeniería de software.

1.3 TEMA DE INVESTIGACIÓN

Formación de grupos en escenarios de aprendizaje colaborativo

1.4 ÁREA DE INVESTIGACIÓN

El desarrollo de este proyecto requirió apropiar y aplicar conocimientos relacionados con las áreas de Ingeniería de Software (IS) y programación de computadores.

1.5 PLANTEAMIENTO DEL PROBLEMA

Los avances tecnológicos actuales requieren profesionales capaces de afrontar los nuevos retos de la industria del desarrollo de software con habilidades técnicas específicas que deben ser fomentadas desde la formación académica, sin embargo, en la realidad existen pocos profesionales para suplir esa demanda. Para el caso de Latinoamérica se estima que se requerirán 1.2 millones de desarrolladores en distintas áreas para 2025, esto debido principalmente a rápida digitalización de las industrias y el incremento en las aplicaciones de uso masivo².

Uno de los retos en la formación de esos profesionales es fomentar formas de pensamiento y organización de las ideas que favorezcan competencias necesarias en todas las ciencias de la computación tales como el análisis, organización y representación lógica de las partes de un problema cuya solución implique la utilización de sistemas computacionales. Es decir, desarrollar lo que se conoce como pensamiento computacional.³ No obstante, el contar con esas habilidades técnicas no garantiza que un individuo pueda desenvolverse exitosamente en el mundo laboral, pues los proyectos reales requieren equipos de varias personas trabajando de forma coordinada para un mismo fin, por lo tanto también es necesario fomentar el desarrollo de habilidades para el trabajo en grupo especialmente para hacerlo de forma virtual, por esta razón la enseñanza debe entrenarlos en este tipo de escenarios de modo que estén preparados para hacerlo en el mundo real. Pues como lo plantean Ruíz Aguirre, Et al: “Trabajar en equipo en los ambientes virtuales no es solo una forma de organizar el trabajo y de aprender de manera colaborativa, también contribuye al desarrollo de una competencia transversal y necesaria a lo largo de la vida”.⁴ Para el caso de estudiantes de programación, una forma de fomentar esta competencia interpersonal en ellos es precisamente la realización de actividades académicas en el aula de clase que impliquen su participación activa en torno a un objetivo común como el resolver ejercicios de programación de forma colaborativa.

Dado lo anterior, es necesario contar con una herramienta software con un nivel adecuado de estabilidad que integre ciertas funcionalidades esenciales necesarias para realizar actividades de trabajo colaborativo en espacios académicos tales

² PORTAFOLIO. Programación, la carrera que más demandará el mercado laboral en 2021. Disponible en <https://www.portafolio.co/tendencias/programacion-la-carrera-que-mas-demandara-el-mercado-laboral-en-el-2021-548289>

³ ZAPATA-ROS, Miguel. Pensamiento computacional: Una nueva alfabetización digital. En Revista Educación a Distancia. (2015); p. 2-12.

⁴ RUÍZ AGUIRRE, Edith Inés, MARTÍNEZ DE LA CRUZ, Nadia Livier, y GALINDO GONZÁLEZ, Rosa María. El aprendizaje colaborativo en ambientes virtuales. 1 a ed. Guadalajara: Editorial Cenid, 2015

como la escritura de código distribuida en tiempo real con indicadores gráficos que indiquen quien se encuentra editando una determinada región de código (tele-punteros), comunicación y la posibilidad de organizar automáticamente los grupos con integrantes compatibles.

En la actualidad existen diversas herramientas para la programación colaborativa, muchas de ellas pueden utilizarse directamente desde un navegador web tales como Codeshare⁵, Collabedit⁶, Codechef⁷, Coderpad⁸ o Codecollab⁹, sin embargo, no todas proveen herramientas de comunicación o están pensadas principalmente realizar entrevistas de programación como en el caso de Coderpad, este último es de pago y limita algunas funciones para las cuentas gratuitas. Por lo anterior se hace necesario acudir una o varias herramientas externas para disponer de todas las funcionalidades necesarias antes mencionadas, además, en la mayoría de los casos no se cuenta con funcionalidades importantes para la productividad y eficiencia como el autocompletado, el subrayado de errores o la navegación en el código (por ejemplo, las funciones de “Ir a la definición” o “Encontrar usos” para una variable o función), funcionalidades comunes en la mayoría de los IDEs convencionales como Eclipse.

Una forma de aprovechar las ventajas de un IDE existente y agregarle funcionalidades específicas es hacerlo por medio de la adición de un *plugin*. Para el caso del IDE Eclipse existen diversos *frameworks* y herramientas de software gratuitas con las que se pueden crear, mantener y complementar *plugins* para múltiples propósitos. Uno de esos *plugins* es Collece 2.0, desarrollado en la Universidad de Castilla la Mancha en España, este satisface la mayoría de las funcionalidades necesarias para la programación colaborativa como la escritura de código remota en tiempo real, chat, bloqueo de regiones de código y tele-cursores, además de todas las funcionalidades presentes en el IDE Eclipse pero no dispone de una herramienta de formación automática de grupos, esta tarea debe realizarse manualmente de forma externa a la plataforma, la librería Java TEAM-B desarrollada en la Universidad de Nariño, satisface esto último, pero en el momento no es posible conectarla directamente con Eclipse. Además, las tareas necesarias para configurar una sesión colaborativa en Collece 2.0 tales como crear un repositorio GIT público, dar de alta a los usuarios en el servidor, y registrarlos en la sesión del grupo que les corresponda se deben realizar manualmente. El tener que realizar todos estos pasos previos para poder utilizar la herramienta sumado a que

⁵ Codeshare. [en línea]. Disponible en: <https://codeshare.io/>

⁶ Collabedit. [en línea]. Disponible en: <http://collabedit.com/>

⁷ Codechef. [en línea]. Disponible en: <https://www.codechef.com/>

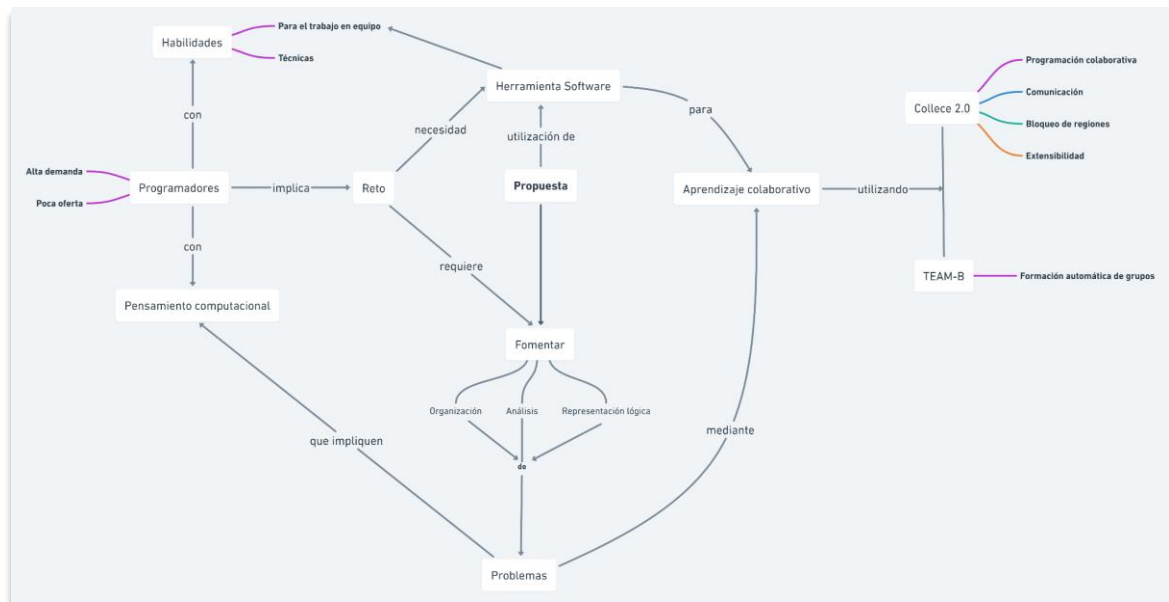
⁸ Coderpad. [en línea]. Disponible en: <https://coderpad.io/>

⁹ Codecollab. [en línea]. Disponible en: <https://codecollab.io/>

la documentación existente para hacerlo no es suficientemente completa puede desalentar a nuevos usuarios interesados en usar Collece 2.0 para sus actividades académicas.

El anterior planteamiento se ha resumido en el mapa mental mostrado en la figura 1.

Figura 1. Mapa mental del planteamiento del problema



Fuente: Creación propia

1.6 FORMULACIÓN DEL PROBLEMA

El presente proyecto se desarrolla para responder al siguiente interrogante principal:

¿Cómo integrar la formación automática de grupos en el entorno de trabajo colaborativo Collece 2.0 donde cada integrante sea compatible con los demás respecto a ciertas características?

1.6.1 Sistematización del problema.

- ¿Cómo es el proceso de creación de *plugins* para el IDE Eclipse?
- ¿Cuál es la arquitectura y funcionamiento del entorno College 2.0?
- ¿Cuál es el funcionamiento de la librería TEAM-B?
- ¿Cómo integrar la librería de formación automática de grupos TEAM-B al entorno de trabajo colaborativo College?
- ¿Cómo se puede validar el funcionamiento del módulo desarrollado?

1.7 OBJETIVOS DE LA INVESTIGACIÓN

1.7.1 Objetivo general. Incorporar al entorno de trabajo colaborativo Collece 2.0 la librería TEAM-B como un módulo para formación automática de equipos.

1.7.2 Objetivos específicos

- Estudiar el proceso de creación de *plugins* para el IDE Eclipse
- Conocer la arquitectura y funcionamiento general del Collece 2.0.
- Conocer el funcionamiento de la librería TEAM-B.
- Desarrollar un módulo para Collece 2.0 que incorpore la formación automática de grupos utilizando la librería TEAM-B.
- Validar el funcionamiento de la integración Collece2.0 – TEAM-B en cursos iniciales de programación.

1.8 JUSTIFICACIÓN

En mundo del desarrollo del software de hoy en día requiere programadores con ciertas habilidades mínimas para el trabajo en equipo como la comunicación clara y asertiva, la colaboración y el manejo adecuado de diferencias de pensamiento para lograr así un alto nivel de sinergia con los compañeros de trabajo, lo anterior debido que la complejidad del software requerido en la actualidad requiere equipos con una cantidad variable de personas en los que las interacciones sean valiosas de modo que generen productos de valor en tiempos razonables.

Estas habilidades pueden ser motivadas desde la enseñanza temprana mediante actividades prácticas de escritura de código que fomenten el trabajo cooperativo: “un proceso en el que cada individuo aprende más de lo que aprendería por sí solo, fruto de la interacción de los integrantes del equipo. Por lo tanto, un trabajo hecho con un grupo cooperativo tiene un resultado más enriquecedor al que tendría la suma del trabajo individual de cada miembro”¹⁰.

Una desventaja del trabajo colaborativo es que los integrantes del equipo pueden ser incompatibles en distintos ámbitos como el ritmo de trabajo y las características de la personalidad, lo que influye en la sinergia del equipo, por esto se hace necesario que la formación de estos grupos no sea arbitraria pues si bien entornos educativos es necesario que el individuo sepa relacionarse asertivamente con distintos tipos de personalidades (especialmente las que no son afines a la suya) el participar en proyectos grupales con personas de personalidades incompatibles puede desanimarlo y darle una perspectiva errada del trabajo colaborativo.

Los diversos avances en software y hardware incluyendo las redes informáticas permiten crear, modificar y utilizar herramientas de escritura colaborativa de código en tiempo real para apoyar e influir directamente en los procesos de aprendizaje. Sin embargo, para asegurar interacciones valiosas entre los miembros de un equipo estos deben tener un grado mínimo de compatibilidad, la cual puede ser lograda mediante el uso de herramientas computacionales, esto junto a estrategias de enseñanza adecuadas permite que los estudiantes adquieran habilidades sociales y técnicas al mismo tiempo. En particular, la librería TEAM-B cumple esta función por medio del uso de algoritmos genéticos que simulan el proceso de selección del más apto teniendo en cuenta para el caso específico, las características de personalidad de los individuos.

¹⁰ GUITERT, Montse y GIMÉNEZ, Ferran Giménez. Trabajo cooperativo en entornos virtuales de aprendizaje. Aprender en la virtualidad. 1 a ed. Barcelona: Editorial Gedisa, 2000. p. 13

Collece 2.0 es un *plugin* para el IDE Eclipse, familiar entre la mayoría de estudiantes de programación, una plataforma que integra la mayoría de funcionalidades necesarias para un ambiente de aprendizaje colaborativo en tiempo real y por su estructura modular permite integrar otras complementarias, para este caso la formación automática de equipos de trabajo homogéneos entre sí pero internamente heterogéneos¹¹.

Si bien en proyectos de desarrollo a mayor escala se utilizan herramientas más complejas como sistemas de control de versiones (git, subversion), la herramienta Collece aplicada en ambientes de aprendizaje ayudaría a introducir a los estudiantes al trabajo colaborativo y motivarlos a explorar otros enfoques para proyectos en equipo mediante la escritura colaborativa de código.

¹¹ SÁNCHEZ, Santiago, REDONDO, Miguel A., VALLEJO, David, GONZÁLEZ, Carlos, y BRAVO, Crescencio. COLLECE 2.0: A DISTRIBUTED REAL-TIME COLLABORATIVE PROGRAMMING ENVIRONMENT FOR THE ECLIPSE PLATFORM. En International Association for Development of the Information Society (AIDIS). (2017); p. 136-142.

1.9 DELIMITACIÓN

Este proyecto se enfocó en la integración de la librería TEAM-B de formación automática de grupos en Collece 2.0, un software desarrollado en el lenguaje de programación Java como *plugin* para el IDE Eclipse que integra varias herramientas para el trabajo colaborativo distribuido en tiempo real.

Collece es un proyecto del grupo de investigación *CHICO* de la Universidad de Castilla la Mancha en España, del cual se tuvo acceso al código fuente por medio de su repositorio, así como soporte para la integración de la librería.

1.10. ANTECEDENTES

Al ser este un trabajo orientado a acoplar una funcionalidad específica (formación automática de grupos) a un software más complejo (Collece 2.0) no se encontraron trabajos previos que busquen lograr un objetivo igual o similar, sin embargo, se encontraron trabajos que ayudarán a contextualizar el presente en áreas en las que este proyecto está inmerso, como la escritura colaborativa de código (programación colaborativa) y el desarrollo de *plugins* para el IDE Eclipse.

1.10.1 Antecedentes a nivel internacional. Se encontraron varios trabajos relacionados con la creación de *plugins* para Eclipse y otras plataformas que asisten o automatizan diversas tareas de programación.

*Desarrollo de un plug-in de eclipse para validar los nombres de los elementos de un esquema conceptual*¹², tesis de Maestría realizada en la Universitat Politècnica de Catalunya (España): donde se define, especifica y describe la implementación de una herramienta desarrollada como un *plugin* para Eclipse para indicar si los nombres de los elementos de un esquema UML cumplen o no con ciertas normas llamadas guías de nominación, esto con el objetivo de mejorar la legibilidad de esquemas conceptuales generados en procesos de ingeniería de requisitos de modo que los nombres de los elementos sean entendibles para todo tipo de personas involucradas en el documento incluso si no tienen conocimientos avanzados en programación.

¹² GARCÍA RANEA, Raúl. Desarrollo de un plug-in de Eclipse para validar los nombres de los elementos de un esquema conceptual. Barcelona, 2011, 438p. Tesis de maestría (Ingeniería Informática). Universitat Politècnica de Catalunya.

*Plugin en eclipse para automatizar el trabajo con el microkernel fiasco oc*¹³, tesis de pregrado realizada en la Universidad Central “Marta Abreu” de las Villas (UCLV) (Cuba): en la cual se describe el proceso de diseño e implementación de un *plugin* para el IDE Eclipse que facilita la tarea de edición del código fuente de Fiasco OC, un microkernel para sistemas embebidos, el principal objetivo de en este proyecto es facilitar la codificación agregando distintas herramientas ya existentes para otros lenguajes como autocompletado y la navegación por el código mediante hipervínculos.

*An eclipse plugin to support code smells detection*¹⁴, artículo realizado por estudiantes de la Facultad de Ciencias y Tecnología de la Universidad de Nueva Lisboa, y el Instituto Universitario de Lisboa (Portugal): donde se presenta una herramienta desarrollada como *plugin* para eclipse llamado *Smellchecker* que automatiza la detección objetiva de *code smells*, un término utilizado para referirse a malas prácticas de programación que pueden indicar la existencia de un problema más grande. El *plugin* funciona en base a un algoritmo de detección basado en estadísticas calibrado progresivamente por una comunidad de desarrolladores, en contraste con técnicas de detección tradicionales basadas en la opinión de expertos.

*Raccode: an eclipse plugin for assessment of programming exercises*¹⁵, artículo realizado por estudiantes de la Facultad de Ciencias de la Universidad de Porto (Portugal): en el cual presentan Raccode, un *plugin* para Eclipse que automatiza la evaluación de ejercicios de programación por medio de la integración con un sistema basado en la web llamado Mooshak a través de una *API REST* (Application Programming Interface). El *plugin* puede ser utilizado en competencias como maratones de programación o entornos de aprendizaje, incluye un sistema de puntaje (*rankings*), solicitudes de aclaraciones, gestión de impresiones y asignación de premios.

¹³ PÉREZ COSTA, Ernesto. *Plugin en Eclipse para automatizar el trabajo con el microkernel Fiasco OC*. Santa Clara. Cuba, 2016, 77p. Trabajo de grado (Licenciatura en Ciencias de la Computación). Universidad Central “Marta Abreu” de Las Villas.

¹⁴ PESSOA, Tiago, BRITO E ABREU, Fernando, PESSOA MONTEIRO, Miguel, y BRYTON, Sérgio. *An Eclipse Plugin to Support Code Smells Detection*. En Research Gate. (Abr.2012); p. 2-12.

¹⁵ SILVA, André, LEAL, José Paulo, y PAIVA, José Carlos. "Raccode: An eclipse plugin for assessment of programming exercises". [en línea]. 2018. {con acceso el 29 de agosto de 2019}. Disponible en <https://api.semanticscholar.org/CorpusID:51868020>

*Desarrollo e implementación de un plugin de google wallet para pagos online utilizando software open source*¹⁶, trabajo de grado de Ingeniería en Sistemas e Informática de la Universidad de las Fuerzas Armadas de Sangolquí (Quito, Ecuador): en cual el autor documenta el proceso de diseño y desarrollo de la billetera virtual *Google Wallet* por medio de un *plugin* en la solución para comercio electrónico *Virtuemart*. La metodología aplicada es OOHDM (Metodología de Desarrollo Hipermedia Orientada a Objetos) y el resultado final es un *plugin* instalable desde el gestor de extensiones de Joomla (un sistema gestor de contenidos de código abierto escrito en PHP).

1.10.2 Antecedentes a nivel nacional. Se encontraron pocos antecedentes referentes a la creación de *plugins* en general, especialmente los relacionados con la enseñanza de la programación, estos se detallan a continuación.

*Construcción de un plugin para el eclipse ide que le permita al estudiante adquirir las principales buenas prácticas de programación orientada a objetos en el lenguaje java*¹⁷, trabajo de grado de Ingeniería de Sistemas de la Universidad Piloto de Colombia: donde los autores documentan el proceso de creación de un *plugin* para Eclipse para apoyar la adopción de buenas prácticas de programación orientada a objetos en estudiantes de Java. El software detecta cuando no se cumple una determinada buena práctica y notifica al usuario mediante alertas que pueden visualizarse directamente en el editor de código, también provee un componente de estadísticas que permiten diagnosticar en que aspectos específicos está fallando el estudiante.

*Diseño y desarrollo de un plugin para la plataforma “moodle” que permite la realización de laboratorios remotos*¹⁸, trabajo de grado de Licenciatura en Electrónica donde se documenta la creación de un *plugin* para Moodle llamado RPL (Remote Practices Laboratory) que sirve como herramienta de apoyo para docentes y estudiantes en el desarrollo, construcción y realización de laboratorios de programación remotos haciendo uso de un robot manipulador y transmisión de video

¹⁶ PAZMIÑO PINTO, Santiago David. Desarrollo e implementación de un plugin de Google Wallet para pagos Online utilizando Software Open Source. Sangolquí, 2013, 97p. Trabajo de grado (Ingeniería en Sistemas e Informática). Universidad de las Fuerzas Armadas ESPE.

¹⁷ JIMÉNEZ BARRETO, Camilo Andrés y OLARTE GARZÓN, Guillermo. Construcción de un plugin para el eclipse IDE que le permita al estudiante adquirir las principales buenas prácticas de programación orientada a objetos en el lenguaje Java. Bogotá, 2015, 123p. Trabajo de grado (Ingeniería de Sistemas). Universidad Piloto de Colombia. Facultad de Ingeniería.

¹⁸ CORTES ROMERO, Henry. DISEÑO Y DESARROLLO DE UN PLUGIN PARA LA PLATAFORMA “MOODLE” QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS. Bogotá, 2016, 78p. Trabajo de grado (Licenciatura en Electrónica). Universidad Pedagógica Nacional. Facultad de Ciencia y Tecnología.

en tiempo real para generar un ambiente de aprendizaje virtual. Para el desarrollo se utiliza una metodología en cascada basándose en tres etapas principales: análisis, diseño - construcción y pruebas que pueden retroalimentarse entre sí si es necesario.

1.10.3 Antecedentes a nivel regional. Después de realizar una búsqueda en las principales fuentes académicas se encontró que existen algunos trabajos relacionados con desarrollar y probar software educativo en ambientes académicos, sin embargo, los softwares tratados no son *plugins* ni están relacionados con la formación de grupos por lo cual no se incluyeron.

No obstante, para la formación automática de grupos si existe un antecedente importante relacionado directamente con este proyecto:

*Cw-teams: software para la conformación de grupos de trabajo colaborativo basado en algoritmos genéticos*¹⁹, trabajo de grado de Ingeniería de Sistemas de la Universidad de Nariño, donde se documenta el proceso de creación de dos aplicativos web para apoyar el proceso de formación de grupos: BF-MANAGER, un gestor de cuestionarios para aplicar un test de aptitud a los individuos participantes que permite exportar los resultados a un archivo de texto plano, CW-TEAMS el cuál forma grupos de trabajo heterogéneos entre sí por medio de algoritmos genéticos partiendo de un archivo con las habilidades de cada integrante generado por el primer aplicativo.

1.11 SUPUESTOS TEÓRICOS DE LA INVESTIGACIÓN

1.11.1 Trabajo colaborativo. Es considerado una filosofía para la interacción y una forma personal de trabajo que implica diversos aspectos como el respeto a las contribuciones individuales y la construcción de consensos al presentarse diferencias entre los miembros²⁰.

Según Guitert & Giménez “es un proceso en el que cada individuo aprende más de lo que aprendería por sí solo, fruto de la interacción de los integrantes del equipo. Se da cuando existe una reciprocidad entre un conjunto de individuos que saben

¹⁹ CORDOBA, Franco Esteban. CW-TEAMS: SOFTWARE PARA LA CONFORMACIÓN DE GRUPOS DE TRABAJO COLABORATIVO BASADO EN ALGORITMOS GENÉTICOS. Pasto, 2019, 222p. Trabajo de grado (Ingeniería de Sistemas). Universidad de Nariño. Facultad de Ingeniería.

²⁰ MALDONADO PÉREZ, Marisabel. El Trabajo Colaborativo En El Aula Universitaria. En Laurus - Revista de Educación. Vol. 13 (2007); p. 263-278.

diferenciar y contrastar sus puntos de vista para llegar a generar un proceso de construcción del conocimiento”²¹. Los autores agregan que el resultado es más enriquecedor que la simple suma del trabajo individual de cada miembro.

Los miembros del grupo se enfrentan al desafío de argumentar y justificar puntos de vista para intentar convencer a los demás en contraposición a una estructura jerárquica de liderazgo en la que un líder puede imponer una visión por el hecho de tener autoridad, por lo tanto, la comunicación dentro del grupo puede ser compleja y las habilidades sociales son indispensables para desarrollar interacciones de calidad²².

Es importante subrayar las diferencias entre trabajo colaborativo y cooperativo, en primer lugar como lo explica Maldonado Pérez: con el trabajo cooperativo se abordan actividades que implican conocimiento básico o fundamental que puede requerir memorización como matemáticas o historia mientras que para las actividades que requieran razonamiento y discusión el trabajo colaborativo sería el apropiado²³. Además, como exponen Echazarreta, Prados, Poch, y Soler la formación de los grupos en el trabajo colaborativo es heterogénea en habilidades y características de los miembros; en el trabajo tradicional de grupos, estas son más homogéneas²⁴. Por último, en el ámbito educativo el trabajo colaborativo se caracteriza porque el profesor pasa de estructurar el trabajo (e incluso también a asignar partes de él a cada integrante) a acompañar el proceso como mediador, dejando que los estudiantes definan objetivos y tareas fomentando que realicen el trabajo juntos asumiendo responsabilidades individuales y grupales al tiempo.

1.11.2 Aprendizaje colaborativo. Es un tipo de trabajo colaborativo aplicado a la enseñanza donde se cambia la figura del docente como el que enseña y la del estudiante como el que aprende de forma pasiva. “En su lugar se presenta el aprendizaje como un proceso social que se construye en la interacción no sólo con el profesor, sino también con los compañeros, con el contexto y con el significado

²¹ GUILLERT, Montse y JIMÉNEZ, Ferran. Aprender a colaborar. Cooperar en clase: ideas e instrumentos para trabajar en el aula. Sevilla: MCEP, 1997. p. 196

²² MALDONADO PÉREZ, Marisabel, op. cit.

²³ MALDONADO PÉREZ, Marisabel, op. cit.

²⁴ ECHAZARRETA, Carmen, PRADOS, Ferran, POCH, Jordi, y SOLER, Josep. La competencia «El trabajo colaborativo»: Una oportunidad para incorporar las TIC en la didáctica universitaria. Descripción de la experiencia con la plataforma ACME (UdG). En UOC Papers - Revista sobre la sociedad del conocimiento. Vol. 8 (2009); p. 1-11.

que se le asigna a lo que se aprende”²⁵. Este tipo de enseñanza se centra en el estudiante y lo que pasa en su interior sin restarle atención a su desempeño en conjunto, esto se apoya en la teoría del constructivismo social que “sostiene que las personas activamente construyen conocimiento mientras interactúan con su ambiente”²⁶.

Se desarrolla a través de un proceso gradual en el que cada miembro y todos se sienten mutuamente comprometidos con el aprendizaje de los demás generando una interdependencia positiva que no implique competencia. Este tipo de aprendizaje se adquiere a través del empleo de métodos de trabajo grupal caracterizado por la interacción y el aporte de todos en la construcción del conocimiento. En el aprendizaje colaborativo el trabajo grupal apunta a compartir la autoridad, a aceptar la responsabilidad y el punto de vista del otro, y a construir consenso con los demás²⁷.

1.11.3 Formación de grupos. Es un área que ha ganado importancia en los últimos años dentro del trabajo colaborativo debido a la aparición de un subtipo de este denominado CSCW (*Computer Supported Collaborative Work*) pues hoy en día muchas actividades requieren realizarse utilizando herramientas tecnológicas en grupos que pueden estar separados físicamente y trabajar en distintos tiempos, esto requiere que los miembros sean compatibles y tengan cierto grado de complementariedad. Pero poner un grupo de personas a trabajar juntos no garantiza un buen resultado (o un buen aprendizaje en el ámbito educativo)²⁸.

Existen varios factores tenidos en cuenta en la formación de grupos, el principal de ellos son las características o cualidades individuales de los miembros pues cada uno puede ser, en determinadas situaciones, un estímulo para los demás miembros e influir en la forma de actuar, sentir o pensar de los mismos²⁹. Esto es especialmente notable en el caso de los defectos, pues un miembro conflictivo afecta el ambiente general del grupo de forma negativa lo cual influye directamente

²⁵ MALDONADO PÉREZ, Marisabel. El Trabajo Colaborativo En El Aula Universitaria. En Laurus - Revista de Educación. Vol. 13 (2007); p. 263-278.

²⁶ Ibíd.

²⁷ JOHNSON, David. W., JOHNSON, Roger T., y JOHNSON, Edythe. Los nuevos círculos de aprendizaje. La cooperación en el aula y la escuela. 1 a ed. Buenos Aires: Editorial Aique, 1999.

²⁸ MONTESERIN, Ariel, SCHIAFFINO, Silvia, GARCÍA, Patricio, y AMANDI, Analía. Análisis de la formación de grupos en Aprendizaje Colaborativo Soportado por Computadoras. En Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE). (Oct.2012); p. 2-4.

²⁹ MARÍN SANCHEZ, Manuel y TROYANO RIDRIGUEZ, Yolanda. Psicología social de los procesos grupales. 1 a ed. Madrid: Ediciones Pirámide, 2012.

en el desarrollo de las actividades asignadas. Otros factores importantes se detallan brevemente a continuación³⁰:

- **Competencia o habilidades de los miembros:** el grupo debe contar con miembros con los conocimientos, habilidades y experiencia necesarias para realizar las tareas grupales, también es necesario (sobre todo en el trabajo colaborativo) que los miembros pongan esos recursos al servicio del grupo y la consecución del objetivo en lugar reservárselas para sí con el fin de destacar o ejercer el liderazgo por el hecho de tenerlas
- **Tamaño del grupo:** si bien esta variable depende de la actividad a realizarse se recomienda una cantidad pequeña integrantes de modo que todos estén involucrados, también que la cantidad sea impar para evitar equilibrio en las posibles votaciones
- **Procesos de comunicación:** la adecuada comunicación dentro del grupo no sólo propicia un mejor ambiente de trabajo, también potencia la coordinación y sincronización al realizar las actividades, también potencia el flujo de información y conocimiento
- **Diversidad grupal:** se refiere a las características que hacen que un miembro se perciba distinto a otro. Con la diversidad se busca lograr complementariedad entre los miembros de modo que el grupo posee mayor cantidad de recursos como habilidades, conocimientos y experiencias en comparación a un grupo homogéneo; sin embargo, lo anterior no significa que deba forzarse la diversidad en todos los aspectos, por ejemplo, la similitud en edades puede facilitar la cohesión y los procesos de comunicación

1.11.4 Java. Lenguaje de programación de propósito general orientado a objetos publicado por primera vez en 1995 por Sun Microsystem (adquirida en 2010 por Oracle). Su objetivo es permitir que el código se escriba una vez y se ejecute en cualquier lugar, este es el mantra original de Java conocido como WORA por sus siglas en inglés: *Write Once Run Anywhere*³¹.

Es un lenguaje interpretado, lo que significa que su código no necesita ser preprocesado mediante un compilador, lo que maximiza la portabilidad a distintas

³⁰ Ibíd.

³¹ COMPUTERWEEKLY. "Write once, run anywhere?". [en línea]. 2018. Disponible en <https://www.computerweekly.com/feature/Write-once-run-anywhere>

arquitecturas³². Su sintaxis está basada en C++ para ser familiar fácil de migrar a él para desarrolladores antiguos.

Sus principales características se describen a continuación³³:

- **Simple, familiar y orientado a objetos:** Uno de los objetivos principales de la creación de Java fue que sea simple (en comparación a los lenguajes de la familia C predominantes en la época) que pueda ser aprendido (migrar) de forma sencilla por programadores acostumbrados a programar en C++, por eso se adoptó su sintaxis. Se cuenta con una extensa librería de objetos probados para proveer funcionalidades que van desde tipos básicos de datos hasta interfaces gráficas de usuario.
- **Robusto y seguro:** Posee un sistema de limpieza automática de memoria (Garbage Collector) que ayuda a evitar problemas relacionados con la gestión de memoria comunes en lenguajes como C++. Además, está diseñado para crear software altamente confiable al proporcionar una comprobación exhaustiva en tiempo de compilación junto a un segundo nivel de comprobación en tiempo de ejecución.
- **Portable y altamente independiente de la arquitectura:** Java está pensado para despliegues en entornos de red heterogéneos en software y hardware, para esto el compilador genera bytecodes, un formato intermedio independiente de la arquitectura. También define estrictamente los tipos básicos de datos y los operadores aritméticos de modo que sean iguales en todas las plataformas.
- **Alto rendimiento:** La plataforma Java logra un buen rendimiento adoptando un esquema en el cual el intérprete puede funcionar a velocidad máxima sin necesidad de verificar el entorno de ejecución constantemente, además el Garbage Collector funciona como un subproceso de baja prioridad en segundo plano que asegura una alta disponibilidad de memoria cuando se necesite.

1.11.5 IDE. Un IDE o Entorno de Desarrollo Integrado (*Integrated Development Environment* en inglés) es un software que integra diferentes herramientas necesarias en el desarrollo de software para permitirle a los desarrolladores incrementar la productividad al consolidar diferentes aspectos de crear un programa

³² ECURED. "Lenguaje interpretado". [en línea]. 2018. Disponible en https://www.ecured.cu/Lenguaje_interpretado

³³ ORACLE. "The Java Language Environment". [en línea]. 2018. Disponible en <https://www.oracle.com/java/technologies/introduction-to-Java.html>

de computador tales como editar código fuente, construir ejecutables, depurar o crear interfaces gráficas³⁴.

La principal utilidad que brindan los IDEs es reducir los pasos necesarios para configurar un entorno de desarrollo y ofrecer distintas herramientas en una misma interfaz gráfica. Por lo general se componen de cierta cantidad de funcionalidades base que se pueden extender con complementos denominados *plugin* desarrollados por los creadores del IDE o por la comunidad de internet.

1.11.6 Eclipse IDE. Es un entorno de desarrollo integrado escrito principalmente en Java usado mayoritariamente para crear aplicaciones en este lenguaje, también tiene soporte para muchos otros lenguajes como C, C++, Fortran y Javascript. Está construido en sobre un espacio de trabajo (*workspace*) base que puede ser extendido mediante un sistema de *plugins* con los que se puede agregar funcionalidades y soporte a otros lenguajes.

El IDE es sólo uno de los productos del proyecto Eclipse dedicado a producir y mantener el Eclipse SDK (Software Development Kit) compuesto por cuatro subproyectos principales: Platform (define un conjunto de marcos de trabajo y servicios comunes para crear un modelo basado en componentes), *Java Development Tools* (JDT), *Plug-In Development Tools* (PDE) y e4 (proyecto donde se estudian nuevas tecnologías para integrarlas al SDK)³⁵.

Eclipse hace uso de muchos componentes del SDK de Eclipse, principalmente las librerías para intercambio de datos como ECF (*Eclipse Communication Framework*) y para interfaces gráficas como JFace y SWT (*Standard Widget Toolkit*)³⁶.

Según la wiki oficial de Eclipse el proyecto inició por la necesidad de IMB de modularizar ciertas herramientas:

IMB quería reducir la gran cantidad de entornos de desarrollo incompatibles que se ofrecían a sus clientes e incrementar el reúso de componentes comunes en esos entornos. Usando el mismo marco común los equipos de desarrollo podrían aprovechar los componentes de los demás, integrarse fácilmente y permitía a cada desarrollador desplazarse entre proyectos. Eclipse surgió de la evolución de herramientas previas orientadas al mismo fin, como el IBM VisualAge Micro Edition

³⁴ CODECADEMY. "What Is an IDE?". [en línea]. 2018. Disponible en <https://www.codecademy.com/articles/what-is-an-ide>

³⁵ ECLIPSE FOUNDATION. "Eclipse Project". [en línea]. 2018. Disponible en <https://www.eclipse.org/eclipse/>

³⁶ ECLIPSE FOUNDATION. "Eclipse Platform Technical Overview". [en línea]. 2001. Disponible en <https://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.html>

que presentaba inconvenientes al tratar de ser ampliado por terceros por no estar diseñado de forma modular y ser un producto monolítico de código cerrado³⁷.

Hoy en día esa modularidad permite el surgimiento de nuevos proyectos dentro del proyecto Eclipse que aprovechen todas las herramientas existentes o las complementen.

1.11.7 Collece 2.0. COLLECE (COLLaborative Edition, Compilation and Execution of programs) es un *plugin* desarrollado para el IDE Eclipse en el Grupo *Computer Human Interaction and Collaboration* (CHICO) de la Universidad de Castilla La Mancha. Provee diversas funcionalidades para facilitar el aprendizaje de la programación por medio de la participación y colaboración en tiempo real de varios usuarios para resolver un problema concreto³⁸.

Su estructura se basa en un conjunto de vistas, editores y asistentes con código modular. La instalación se realiza por medio de un sitio web de modo que su instalación es sencilla para los usuarios, quienes pueden hacerlo directamente desde el editor de paquetes de Eclipse. El sistema distingue entre dos entidades: cliente y servidor. El servidor se encarga de manejar el sistema de sesiones, almacenar la información de los usuarios y la sincronizar los datos a través de los clientes. Las entidades cliente son los usuarios reales del sistema que se conectan a un servidor para colaborar remotamente en un proyecto colaborativo³⁹.

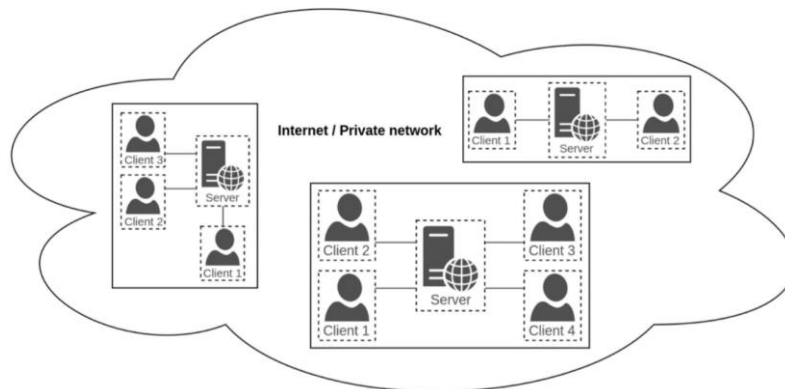
En la figura 2 se muestra la arquitectura general del entorno Collece 2.0. Como se puede observar, una instancia de servidor alberga a varias instancias cliente concurrentes conectados por medio de internet o una red local privada, pero no existe interconexión entre las instancias servidor.

³⁷ ECLIPSE FOUNDATION. "FAQ Where did Eclipse come from?". [en línea]. 2004. Disponible en https://wiki.eclipse.org/FAQ_Where_did_Eclipse_come_from%3F

³⁸ GRUPO COMPUTER HUMAN INTERACTION AND COLLABORATION (CHICO). "COLLECE-2.0". [en línea]. 2019. Disponible en <http://blog.uclm.es/grupochico/proyecto-iapro/collece-2-0/>

³⁹ SÁNCHEZ, Santiago, REDONDO, Miguel A., VALLEJO, David, GONZÁLEZ, Carlos, y BRAVO, Crescencio. COLLECE 2.0: A DISTRIBUTED REAL-TIME COLLABORATIVE PROGRAMMING ENVIRONMENT FOR THE ECLIPSE PLATFORM. En International Association for Development of the Information Society (AIDIS). (2017); p. 136-142.

Figura 2. Arquitectura cliente-servidor de Collece 2.0



Fuente: Collece 2.0: a distributed real-time collaborative programming environment for the eclipse platform ⁶

Entre las principales características de Collece 2.0 se destacan:

- Sesiones de trabajo: Un servidor puede alojar varias sesiones con múltiples usuarios conectados simultáneamente
- Edición multiusuario: Varios usuarios pueden editar un mismo archivo al mismo tiempo y ver los cambios de los demás, además cada uno puede saber quién está editando una determinada parte gracias a los telepunteros, indicadores de colores asociados a un usuario
- Bloqueo de regiones: Evita conflictos y mejora la coordinación entre los usuarios permitiendo que bloqueen y desbloqueen secciones concretas del código

1.11.8 GIT. Es un sistema de control de versiones distribuido, gratuito, de código abierto y fácil de aprender diseñado para gestionar proyectos de cualquier tamaño de forma rápida y eficiente⁴⁰. La principal ventaja de este sistema son las ramas (*Branch* en inglés) comparables a líneas de tiempo paralelas independientes entre sí, lo que permite que los miembros de un equipo de desarrollo trabajen en distintas áreas de un proyecto en distintos instantes de tiempo y unificar los cambios de forma sencilla permitiéndoles volver a versiones anteriores de ser necesario.

Al ser distribuido la mayoría de operaciones pueden realizarse de forma local, y replicarse en otros equipos o servidores cuando haya conexión contrario a los

⁴⁰ SOFTWARE FREEDOM CONSERVANCY. "Git". [en línea]. 2019. {con acceso el 14 de septiembre de 2020}. Disponible en <https://git-scm.com/>

sistemas centralizados pues estos suelen depender constantemente de la conexión con un servidor⁴¹.

Git puede usarse de forma local (en computadores personales), en redes internas o en internet por medio de diversas plataformas como Github, Gitlab o Bitbucket.

⁴¹ SOFTWARE FREEDOM CONSERVANCY. "About - Git". [en línea]. 2019. {con acceso el 14 de septiembre de 2019}. Disponible en <https://git-scm.com/about/branching-and-merging>

2. METODOLOGÍA

2.1 MODALIDAD DEL PROYECTO

El trabajo de grado propuesto se enmarca en la modalidad de Trabajo de Investigación

2.2 PARADIGMA DE INVESTIGACIÓN

Desde el punto de vista epistemológico y metodológico, el proceso investigativo se desarrolló bajo el paradigma positivista, dado que se fundamenta en el conocimiento científico. Este paradigma también suele conocerse como empírico-analítico o racionalista, supone que la realidad está dada y es única (con independencia de quien la estudie) y el medio para llegar a ella es la ciencia⁴².

2.3 TIPO DE INVESTIGACIÓN

En concordancia con la escuela de pensamiento del positivismo, esta investigación se enmarca en el enfoque cuantitativo, dado que se utilizan datos cuantitativos o medibles.

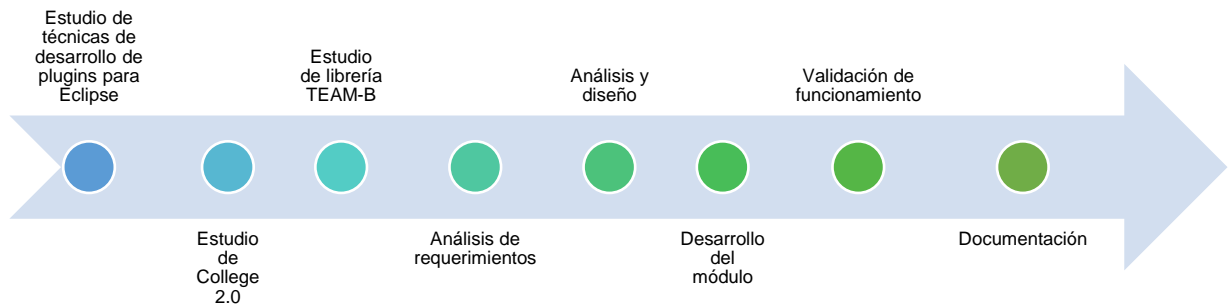
2.4 DISEÑO DE LA INVESTIGACIÓN

El proceso general de la investigación inició por la apropiación de conocimiento necesario sobre las tecnologías y productos sobre los cuales se trabajaría (principalmente el *plugin* de Eclipse Collece 2.0 y la librería Java TEAM-B) para de ese modo entender el problema a solucionar, plantear los objetivos del proyecto y trazar una ruta general para alcanzarlos.

La figura 3 muestra de forma gráfica el proceso seguido a lo largo de esta investigación.

⁴² FLORES, Erik. "Paradigma Positivista". [en línea]. 2018. {con acceso el 12 de diciembre de 2019}. Disponible en <https://sites.google.com/site/wikiminvestigacion2018/paradigma-positivista>

Figura 3. Diseño de investigación



Fuente: Creación propia

Una vez terminada la fase de desarrollo se procedió a validar el funcionamiento de Collece 2.0 junto con TEAM-BE en un entorno real pues ese es uno de los objetivos de este proyecto. Teniendo en cuenta el tipo y paradigma de investigación antes descritos se siguió un diseño empírico, basado en una prueba de campo aplicando un tipo de experimento en particular denominado pre-experimento con una sola medición. En este tipo de experimento se administra un estímulo o tratamiento a un grupo particular y después se aplica una medición de una o más variables para determinar si el estímulo tuvo algún nivel de influencia en esas variables.⁴³

Se utilizó este diseño experimental porque si bien es sencillo, permite verificar el correcto funcionamiento de la herramienta desarrollada desde el punto de vista técnico en un entorno real. Además, la medición permite conocer la percepción y aceptación de los participantes con respecto al uso de una herramienta de software (TEAM-BE) para la formación automática de grupos y Collece 2.0 como entorno de trabajo colaborativo. Lo anterior permite recibir retroalimentación de los participantes e identificar posibles mejoras para futuras versiones.

⁴³ FERNANDEZ COLLADO, Carlos y BAPTISTA LUCIO, Pilar. Metodología de la investigación. 6ta ed. México D.F: McGraw-Hill, 2014.

Para representar los elementos que el experimento se utiliza la siguiente tipología propuesta por Campbell y Stanley⁴⁴:

G X O

A continuación, se describe el significado de cada símbolo y su valor particular para esta investigación:

- *G*: Grupo de sujetos o casos. Para este caso se trabajó con un grupo de estudiantes de Ingeniería de Sistemas de la Universidad de Nariño, el grupo se describe en profundidad en los subcapítulos de población y muestra
- *X*: Tratamiento o estímulo experimental (la variable que se manipula). En este caso se trata de la utilización del conjunto de herramientas Collece 2.0 - TEAM-BE mediante el desarrollo de un ejercicio tipo programación competitiva que requiere que los participantes trabajen conjuntamente para su solución. La actividad se llevó a cabo con grupos de 3 estudiantes formados de forma automática por TEAM-BE usando como datos de entrada las características de personalidad de los participantes.
- *O*: Medición hecha sobre los sujetos o casos para medir el efecto del tratamiento o estímulo. En este tipo de experimento se realizó después, por esta razón el símbolo se ubica al final. En esta investigación se realizará una medición posterior por medio de una encuesta aplicada a todos los participantes para conocer los efectos del uso de la herramienta. El diseño de la encuesta, así como sus resultados se exponen en profundidad en el capítulo de validación de la herramienta.

2.4.1 Población. El conjunto de estudiantes que forman parte de comunidades académicas en formación relacionada con ciencias de la computación. En particular, la herramienta fue probada con estudiantes de Ingeniería de Sistemas de la Universidad de Nariño, de las sedes Ipiales y Pasto en espacios académicos facilitados por docentes. En ambos casos los participantes contaban con conocimientos previos suficientes en el lenguaje de programación usado (Java) y en el manejo del IDE Eclipse.

⁴⁴ CAMPBELL, Donald T. y STANLEY, Julián C. Diseños experimentales y cuasiexperimentales en la investigación social. 1 a ed. Buenos Aires: Rand McNally & Company, 1966.

2.4.2 Muestra. La muestra poblacional fue determinística. Se utilizó la participación voluntaria de grupos de estudiantes de semestres avanzados del programa de ingeniería de sistemas, distribuidos como se indica en la tabla 1.

Tabla 1. Distribución de participantes escenario de aplicación

Programa y Universidad	Materia	Docente a cargo	Número de participantes
Ingeniería de Sistemas. Universidad de Nariño sede Ipiales	Seminario de Computación e Informática 2	Sandra Marcela Guerrero Calvache	11
Ingeniería de Sistemas. Universidad de Nariño sede Pasto	Software gráfico	Javier Alejandro Jiménez Toledo	31

Fuente: creación propia

2.4.3 Definición de tecnologías. La herramienta se construyó con Java, pues es el lenguaje en el que se desarrollan los *plugins* para el IDE Eclipse y sobre el cual está construido Collece 2.0. A continuación se listan los *frameworks* específicos de Java utilizados:

Eclipse PDE: (*Plugin Development Environment*) es el macro conjunto de herramientas para crear, compilar, probar y desplegar *plugins* y otros complementos para el IDE Eclipse.

SWT: subconjunto de herramientas de PDE para la creación de interfaces de usuario, las cuales tienen el aspecto nativo del sistema operativo donde se ejecuten.

iTextPDF: conjunto gratuito de herramientas para la generación y manipulación de archivos PDF que provee un amplio conjunto de elementos gráficos para organizar información.

Retrofit: cliente HTTP que permite enviar y recibir información de servidores en internet de forma segura mediante API REST. En este caso se utilizó la API de Bitbucket para crear repositorios GIT de forma automática.

2.5 METODOLOGÍA DE INGENIERÍA DE SOFTWARE

Teniendo en cuenta que el desarrollo de la herramienta software fue realizado por una sola persona con asesoría de un docente y teniendo unos requisitos generales claros se optó por utilizar una metodología ágil de modo que se puedan realizar iteraciones que generen entregas periódicas que puedan ser evaluadas por el asesor.

En este contexto se hace necesario acudir al manifiesto ágil de la ingeniería de software, el cual define algunos principios y valores esenciales para cualquier metodología basada en el agilismo⁴⁵. En general con los valores se busca que el desarrollo esté enfocado en la satisfacción del cliente y los usuarios con una rápida adaptación a los cambios gestionando entregas periódicas del software funcionando, para esto se hace necesaria la participación constante de las partes interesadas. En cuanto a los principios destacan la inspección y la adaptación constante al cambio, pues además de la retroalimentación del cliente se hace una autoevaluación continua durante todo el proceso con el fin de encontrar oportunidades de mejora.

Dentro del amplio espectro de metodologías que comulguen con el manifiesto ágil se tienen los siguientes representantes:

- **Kanban**

Se apoya en la representación visual de los elementos de trabajo por medio de un tablero con distintas categorías denominado *Kanban Board* lo cual permite un monitoreo constante del avance del proyecto, esta metodología requiere una comunicación constante y en tiempo real de los miembros del equipo, así como la transparencia en cuanto a la capacidad de trabajo de los mismos⁴⁶.

- **Extreme Programming (XP)**

Está enfocada en producir software de alta calidad al mismo tiempo que se mantiene la calidad de vida de los integrantes del equipo, tiene un conjunto de prácticas bastante específicas lo cual hace que los escenarios para su aplicación sean algo restringidos pues está pensada para equipos pequeños

⁴⁵ PALACIO, Carlos. E-book Nociones básicas de metodologías ágiles. 1 a ed. Bogotá: Academia Pragma, 2020.

⁴⁶ ATLASSIAN AGILE COACH. "What is kanban?". [en línea]. 2019. {con acceso el 17 de noviembre de 2019}. Disponible en <https://www.atlassian.com/agile/kanban>

que comparten ubicación física y trabajan en proyectos con alto nivel de riesgo debido a tiempos de entrega fijos, requisitos cambiantes y el uso de tecnologías emergentes⁴⁷.

- **Agile Scrum Methodology**

Organiza y detalla el proyecto mediante historias de usuario, las cuales son descripciones breves de las características deseadas redactadas desde el punto de vista del interesado. Carlos Palacio de Academia Pragma presenta la siguiente definición:

Es un *framework* basado en los principios y valores del agilismo, parte de la base de lo que se llama un product backlog o pila de producto, que está priorizada. Los equipos entienden las necesidades con la ayuda del *product owner* (dueño del producto) y establecen un compromiso para desarrollar cierta porción de esa pila de producto en el *sprint* (tiempos de entrega), que suele ser un periodo entre 15 o 30 días. Lo ideal es que se realice en máximo 15 días, ya que se puede reducir el espectro de errores o cambios en ese tiempo⁴⁸.

Se realizan distintas reuniones en distintos periodos de tiempo y estados del proyecto con el fin de monitorear los avances y realizar cambios en caso de ser necesario, estas reuniones se caracterizan por ser breves y tener una agenda bien definida para evitar la pérdida de tiempo.

- **Lean Software Development**

“Es un *framework* ágil basado en optimizar el tiempo de desarrollo y los recursos, eliminar el desperdicio y entregar solo lo que el producto necesita”⁴⁹. En este los equipos entregan un mínimo producto viable al cliente del cual reciben retroalimentación e inician nuevas iteraciones para adaptar las recomendaciones.

- **Crystal**

⁴⁷ AGILE ALLIANCE. "Extreme Programming (Glossary)". [en línea]. 2019. {con acceso el 17 de noviembre de 2019}. Disponible en <https://www.agilealliance.org/glossary/xp>

⁴⁸ PALACIO, Carlos. E-book Nociones básicas de metodologías ágiles. 1 a ed. Bogotá: Academia Pragma, 2020.

⁴⁹ PRODUCTPLAN. "Lean Software Development". [en línea]. 2019. {con acceso el 17 de noviembre de 2019}. Disponible en <https://www.productplan.com/glossary/lean-software-development/>

“Es una metodología flexible centrada en las personas y sus interacciones en lugar de los procesos y herramientas. En otras palabras es una consecuencia directa de uno de los valores fundamentales del manifiesto ágil”⁵⁰. Tiene un enfoque adaptativo que confía en la capacidad de los equipos de optimizar sus procesos y responder a los cambios en los requisitos. También resta importancia a la documentación y los informes, lo cual puede ser una desventaja al momento de mostrar los avances a entidades externas al equipo.

- **Feature Driven Development (FDD)**

Organiza el proceso de desarrollo en torno a la entrega características, que son funciones pequeñas valoradas desde la perspectiva del cliente, similares al concepto de historias de usuario de SCRUM. Funciona mejor en equipos de desarrollo grandes y depende de la existencia de desarrolladores experimentados que lideren el proyecto. El ciclo de vida del proyecto empieza por desarrollar un modelo general y construir una lista de características para posteriormente realizar el proceso de planeación, diseño y construcción por cada una de ellas⁵¹.

Existen muchos más métodos ágiles; sin embargo, se ha considerado aquellos que son utilizados con más frecuencia.

Considerando las características del presente proyecto se ha optado por seguir la metodología SCRUM pues favorece la entrega constante e incremental de avances a través de iteraciones (en adelante *Sprints*), además, facilita adaptarse al cambio constante en los requisitos, un escenario muy probable pues los avances logrados se presentaban al asesor y al equipo de desarrollo de Collece 2.0 quienes aportaron sugerencias y solicitaron cambios en reuniones realizadas periódicamente.

Por otro lado, al ser un marco de trabajo ampliamente conocido y aplicado en proyectos de distinta naturaleza (no sólo de desarrollo de software) existe documentación amplia y diversas herramientas software que facilitan su aplicación. Finalmente, la metodología SCRUM es adaptativa, lo cual permitió que se realizaran algunos ajustes para su aplicación principalmente porque el equipo de desarrollo estuvo formado por una sola persona, estos ajustes fueron:

⁵⁰ PRODUCTPLAN. "Crystal Agile Framework". [en línea]. 2019. {con acceso el 17 de diciembre de 2019}. Disponible en <https://www.productplan.com/glossary/crystal-agile-framework/>

⁵¹ PRODUCTPLAN. "Feature Driven Development (FDD)". [en línea]. 2019. {con acceso el 17 de noviembre de 2019}. Disponible en <https://www.productplan.com/glossary/feature-driven-development/>

1. El desarrollador también cumple el rol de SCRUM Master
2. Solo se implementaron algunos de los eventos (reuniones) definidos en la guía de SCRUM

2.5.1 Roles. A continuación, en la tabla 2 se detallan las personas involucradas en el desarrollo del software, sus respectivos roles y las funciones que desempeñan según la literatura consultada.

Tabla 2. Personas y roles involucrados en el desarrollo del software

Persona	Rol	Funciones
Juan Pablo Botina Carlosama	<i>SCRUM Master</i>	Monitorear el cumplimiento de la metodología, facilitar la realización de los eventos, eliminar inconvenientes que impidan el avance global o del <i>sprint</i>
Juan Pablo Botina Carlosama	<i>Development Team</i>	Desarrollar el producto de forma incremental, entregando avances de forma periódica
<ul style="list-style-type: none"> - Oscar Revelo Sanchez (Creador de librería TEAM-B) - Jesús Insuasti (Co-Creador de librería TEAM-B) - Grupo de Investigación CHICO (Creador de <i>plugin</i> Collece 2.0) 	<i>Product Owner</i>	Gestionar la lista del producto (<i>product backlog</i>), definir prioridades, transmitir información entre equipo de desarrollo y los demás interesados (<i>stakeholders</i>)

Fuente: Guía funcional de SCRUM⁵²

2.5.2 Eventos. Como se indicó anteriormente no se implementaron todos los eventos especificados por SCRUM, en particular, en este proyecto no se implementó el evento *Daily SCRUM* pues como su nombre lo indica es una reunión diaria y está pensada principalmente para que el equipo de desarrollo discuta y planifique avances a corto plazo, en este caso al tratarse de una sola persona estas tareas de organización se llevaron a cabo de constantemente de forma individual.

A continuación se resumen los eventos que sí se implementaron con base en la guía de SCRUM⁵³:

Sprint planning: participa todo el equipo SCRUM, se define que historias de usuario formarán el *Sprint* y cuál será el entregable resultante

⁵² ACADEMIA PRAGMA. Guía funcional de SCRUM. 1 a ed. Bogotá: Academia Pragma, 2020.

⁵³ ACADEMIA PRAGMA. Guía funcional de SCRUM. 1 a ed. Bogotá: Academia Pragma, 2020.

Sprint review: reunión informal realizada al final del *Sprint* en la que participa todo el equipo SCRUM en esta se hacen cambios en la pila del producto si es necesario, se proyectan los posibles objetivos para los siguientes *Sprints* y al tiempo que se definen los posibles entregables de cada uno.

Sprint retrospective: se realiza después del *sprint review*, en esta el equipo realiza una autoinspección y un plan de mejoras (si fuese necesario) revisando lo que salió bien y mal en el *sprint* finalizado con lo cual se crea un plan para implementar las posibles mejoras en el corto y mediano plazo. Para este evento, al igual que para el *sprint review* se hizo uso constante de la herramienta Jira software para rastrear el progreso de las historias de usuario. Además, se creó un formato para llevar un registro de los temas tratados en este evento, el cual se puede observar en el Anexo 3.

2.5.3 Artefactos. A continuación, se detallan las herramientas de la metodología SCRUM utilizadas en el proceso de desarrollo de software de este proyecto. Para este proyecto se utilizó la herramienta gratuita online Jira Software, la cual está diseñada especialmente para metodologías de desarrollo ágil y cuenta con una plantilla especial para Scrum, allí se gestionó el proceso, desde la construcción del backlog hasta la gestión de las historias de usuario para cada Sprint.

2.5.3.1 Lista de producto (Product Backlog). Es una lista ordenada con todos los requisitos que se conocen del producto a desarrollar⁵⁴.

Para documentar este artefacto se tomó como base el formato propuesto en Project Management Docs⁵⁵. Se utilizó una escala discreta de tres valores (baja, media, y alta) para describir la prioridad y complejidad estimada de cada historia de usuario.

En la tabla 3 se muestran las funcionalidades o requisitos priorizados para la herramienta, se incluyen los que estuvieron presentes desde el inicio del proceso y los que aparecieron en iteraciones sucesivas.

Tabla 3. Lista del producto (Product Backlog)

No	Historia de usuario	Valor que agrega al proceso	Prioridad	Complejidad estimada
1	Como: Administrador Quiero: Lanzar la interfaz de formación de grupos desde Collece 2.0 Para: Formar grupos de trabajo sin salir del entorno Collece	10	Alta	Media
2	Como: Administrador Quiero: Cargar los datos de los individuos que formarán los grupos Para: Formar los grupos de forma automática con base en esos datos	10	Alta	Media
3	Como: Administrador Quiero: Formar los grupos de forma automática Para: Desarrollar actividades de programación colaborativa	10	Alta	Alta
4	Como: Administrador Quiero: Exportar los grupos formados a un archivo PDF Para: Visualizar la información de los grupos cuando lo requiera	8	Alta	Baja
5	Como: Administrador Quiero: Crear de forma automática los repositorios GIT necesarios para cada sesión Para: Que cada grupo pueda acceder al sistema	10	Alta	Media
6	Como: Administrador	10	Alta	Alta

⁵⁴ ACADEMIA PRAGMA. Guía funcional de SCRUM. 1 a ed. Bogotá: Academia Pragma, 2020.

⁵⁵ PROJECT MANAGEMENT DOCS. "Agile Product Backlog". [en línea]. 2020. Disponible en <https://www.projectmanagementdocs.com/template/agile-templates/agile-product-backlog/>

7	Quiero: Crear los usuarios y las sesiones en Collece de forma automática	10	Alta	Alta
	Para: que los participantes puedan acceder al sistema			
	Como: Usuario			
8	Quiero: Instalar el software de forma sencilla	8	Media	Alta
	Para: Desarrollar actividades de programación colaborativa			
	Como: Usuario			
8	Quiero: Acceder a un manual de instalación y uso del software	8	Media	Alta
	Para: Desarrollar actividades de programación colaborativa			
	Como: Usuario			

Fuente: Creación propia

2.5.3.2 Sprints. Son las unidades básicas de tiempo de Scrum, contienen una cantidad determinada de funcionalidades o historias de usuario con las que se espera lograr un incremento (entregable) que es presentado a los *Product Owner*. Su duración se establece en semanas al inicio del proceso y generalmente se mantiene hasta el final.

Para la estimación del tiempo necesario para cada historia de usuario se utilizaron los puntos de historia. Se utilizó una escala siguiendo la secuencia de números Fibonacci, estableciendo un rango de horas para cada valor como se muestra en la tabla 4.

Tabla 4. Puntos de historia según su duración

Puntos	Duración
1	1 a 8 horas
2	8 a 21 horas
3	21 a 34 horas
5	34 a 55 horas
8	55 o más horas

Fuente: Creación propia

Los sprints se registraron en Jira Software utilizando la interfaz que se muestra en el Anexo 2.

A continuación, se detallan los sprints de este proyecto, con sus fechas de inicio y finalización, así como las historias de usuario que las componen y su estimación en

horas. El formato utilizado se elaboró con base a la propuesta del sitio web ScrumManager⁵⁶.

En la tabla 5 se detalla el sprint 1, el cual contiene las historias de usuario iniciales necesarias para el resto de sprints.

Tabla 5. Sprint #1

ID Sprint		Sprint #1		
Fecha de inicio		08/01/2020		
Fecha de finalización		08/03/2020		
Número total de días		60		
Historias de usuario				
#	Nombre de la historia de usuario	Tareas	Responsable	Duración estimada (horas)
1	Lanzar la interfaz de formación de grupos desde Collece 2.0	Especificación de historias de usuario	Juan Botina	8
		Realizar el prototipo de interfaz gráfica de usuario	Juan Botina	8
		Codificación de la historia de usuario	Juan Botina	13
		Elaborar y ejecutar escenarios de pruebas	Juan Botina	8
2	Cargar los datos de los individuos que formarán los grupos	Especificación de historias de usuario	Juan Botina	8
		Realizar el prototipo de interfaz gráfica de usuario	Juan Botina	8
		Codificación de la historia de usuario	Juan Botina	13
		Elaborar y ejecutar escenarios de pruebas	Juan Botina	13

Fuente: Creación propia

⁵⁶ SCRUMMANAGER y NAVEGÁPOLIS. "Formato ejemplo para documentar el uso de scrum en un proyecto". [en línea]. 2012. Disponible en <https://www.safecreative.org/work/1206241856508-formato-ejemplo-para-documentar-el-uso-de-scrum-en-un-proyecto>

En la tabla 6 se detalla el sprint 2, que contiene las historias de usuario relacionadas a la formación de los grupos y la exportación de datos.

Tabla 6. Sprint #2

ID Sprint		Sprint #2		
Fecha de inicio		09/03/2020		
Fecha de finalización		08/05/2020		
Número total de días del sprint		60		
Historias de usuario				
#	Nombre de la historia de usuario	Tareas	Responsable	Duración estimada (horas)
1	Formar los grupos de forma automática	Especificación de historias de usuario	Juan Botina	8
		Codificación de la historia de usuario	Juan Botina	13
		Elaborar y ejecutar escenarios de pruebas	Juan Botina	13
2	Exportar los grupos formados a un archivo PDF	Especificación de historias de usuario	Juan Botina	8
		Realizar el diseño del documento	Juan Botina	13
		Codificación de la historia de usuario	Juan Botina	13
		Elaborar y ejecutar escenarios de pruebas	Juan Botina	13

Fuente: Creación propia

En la tabla 7 se detalla el sprint 3. Este contiene las historias de usuario relacionadas a la automatización de tareas necesarias en el proceso de creación de sesiones en Collece 2.0.

Tabla 7. Sprint #3

ID Sprint		Sprint #3		
Fecha de inicio		12/05/2020		
Fecha de finalización		11/07/2020		
Número total de días del sprint		60		
Historias de usuario				
#	Nombre de la historia de usuario	Tareas	Responsable	Duración estimada (horas)
1	Crear de forma automática los repositorios GIT necesarios para cada sesión	Especificación de historias de usuario	Juan Botina	8
		Codificación de la historia de usuario	Juan Botina	13
		Elaborar y ejecutar escenarios de pruebas	Juan Botina	13
2	Crear los usuarios y las sesiones en Collece de forma automática	Especificación de historias de usuario	Juan Botina	8
		Codificación de la historia de usuario	Juan Botina	13
		Elaborar y ejecutar escenarios de pruebas	Juan Botina	13

Fuente: Creación propia

En la tabla 8 se detalla el sprint 4. Este contiene historias de usuario relacionadas a facilitar la instalación y uso de la herramienta software desarrollada.

Tabla 8. Sprint #4

ID Sprint		Sprint #4		
Fecha de inicio		14/07/2020		
Fecha de finalización		12/07/2020		
Número total de días del sprint		60		
Historias de usuario				
#	Nombre de la historia de usuario	Tareas	Responsable	Duración estimada (horas)
1	Instalar el software de forma sencilla	Especificación de las historias de usuario	Juan Botina	8
		Preparación de los archivos de instalación del software	Juan Botina	13
		Publicación de los archivos de instalación en un sitio web	Juan Botina	13
2	Acceder a un manual de instalación y uso del software	Especificación de las historias de usuario	Juan Botina	8
		Diseño del manual de usuario	Juan Botina	13
		Publicación del manual de usuario en un sitio web	Juan Botina	13

Fuente: Creación propia

2.5.3.3 Historias de usuario. Son descripciones de una funcionalidad, vista desde el punto de vista de un usuario con un rol que va a interactuar con el sistema, como un usuario regular o un administrador. Las tablas 9 a 16 contienen la descripción de las funcionalidades requeridas para el software incluyendo la estimación en horas y su prioridad. Para cada una se definieron uno o más criterios de aceptación que permiten evaluar si el producto cumple con las necesidades del usuario. Para documentar estas historias de usuario se utilizó un formato elaborado con base en

la propuesta presentada por Villamizar Suaza⁵⁷. Las historias de usuario también se registraron en Jira Software utilizando la interfaz que se muestra en el Anexo 1.

Tabla 9. Historia de usuario #1

ID Historia	HU. #1			
Nombre	Lanzar la interfaz de formación de grupos desde Collece 2.0			
Actor	Administrador			
Descripción	Como Administrador Quiero lanzar la interfaz de formación de grupos desde Collece 2.0 Para formar grupos de trabajo sin salir del entorno Collece			
Estimación (horas)	34			
Prioridad	Must			
Criterios de aceptación				
ID. del escenario	Título del escenario	Contexto	Evento o condición	Resultado o comportamiento esperado
1	Campos vacíos	Si se dejan campos vacíos en la interfaz de creación de sesiones	Cuando se presione el botón 'OK'	El sistema mostrará un error diciendo 'No se permiten campos vacíos'
2	Datos erróneos	Si se ingresan campos erróneos en la interfaz de creación de sesiones	Cuando se presione el botón 'OK'	El sistema mostrará un mensaje indicando el error
3	Datos correctos	Si se ingresan correctamente los datos en la	Cuando se presione	El sistema lanzará la interfaz

⁵⁷ VILLAMIZAR SUAZA, Katherine. Definición de equivalencias entre historias de usuario y especificaciones en UN-LENCEP para el desarrollo ágil de software. 2013. Universidad Nacional de Colombia.

		interfaz de creación de sesiones	el botón 'OK'	de la herramienta TEAM-BE
--	--	----------------------------------	---------------	---------------------------

Fuente: Creación propia

Tabla 10. Historia de usuario #2

ID Historia	HU. #2			
Nombre	Cargar los datos de los individuos que formarán los grupos			
Actor	Administrador			
Descripción	Como Administrador Quiero cargar los datos de los individuos que formarán los grupos Para formar los grupos de forma automática con base en esos datos			
Estimación (horas)	42			
Prioridad	Must			
Criterios de aceptación				
ID. del escenario	Título del escenario	Contexto	Evento o condición	Resultado o comportamiento esperado
1	Datos erróneos	Si el archivo CSV seleccionado tiene un formato incorrecto	Cuando ser presione el botón “Cargar”	El sistema mostrará un error diciendo ‘Archivo CSV inválido’
2	Datos correctos	Si el archivo CSV seleccionado está correctamente formateado	Cuando ser presione el botón “Cargar”	El sistema cargará los datos de usuarios desde el archivo CSV y los mostrará en la interfaz gráfica

Fuente: Creación propia

Tabla 11. Historia de usuario #3

ID Historia	HU. #3			
Nombre	Formar los grupos de forma automática			
Actor	Administrador			
Descripción	Como Administrador Quiero formar los grupos de forma automática Para desarrollar actividades de programación colaborativa			
Estimación (horas)	34			
Prioridad	Must			
HU Relacionadas	Cargar datos de individuos (HU. #2)			
Criterios de aceptación				
ID. del escenario	Título del escenario	Contexto	Evento o condición	Resultado o comportamiento esperado
1	Datos incorrectos	Cuando los parámetros para la formación de grupos (número de individuos o tamaño del grupo) son incorrectos	Cuando se presione el botón 'formar grupos'	El sistema mostrará un mensaje indicando el error
2	Datos no cargados	Cuando no se hayan cargado los datos de los individuos que formaran los grupos	Cuando se presione el botón 'formar grupos'	El sistema mostrará un mensaje de error diciendo 'datos no cargados'

3	Datos correctos	Cuando los parámetros para la formación de grupos sean correctos	Cuando se presione el botón 'formar grupos'	El sistema empezará a formar los grupos
---	-----------------	--	---	---

Fuente: Creación propia

Tabla 12. Historia de usuario #4

ID Historia	HU. #4			
Nombre	Exportar los grupos formados a un archivo PDF			
Actor	Administrador			
Descripción	Como Administrador			
	Quiero exportar los grupos formados a un archivo PDF			
	Para visualizar la información de los grupos cuando lo requiera			
Estimación (horas)	47			
Prioridad	Must			
HU Relacionadas	Formar los grupos de forma automática (HU. #3)			
Criterios de aceptación				
ID. del escenario	Título del escenario	Contexto	Evento o condición	Resultado o comportamiento esperado
1	Grupos no formados	Cuando aún no se han formado los grupos	N/A	La interfaz gráfica no mostrará el botón 'Exportar a PDF'
2	Grupos formados	Cuando los grupos ya han sido formados	N/A	La interfaz gráfica mostrará el botón 'Exportar a PDF'
3	Grupos formados y botón 'Exportar a PDF' visible	Cuando los grupos han sido formados correctamente la interfaz muestra el botón 'Exportar a PDF'	Cuando se presione el botón 'Exportar a PDF'	Se generará un archivo PDF con la información de los grupos formados y sus integrantes

Fuente: Creación propia

Tabla 13. Historia de usuario #5

ID Historia	HU. #5			
Nombre	Crear de forma automática los repositorios GIT necesarios para cada sesión			
Actor	Administrador			
Descripción	Como Administrador Quiero crear de forma automática los repositorios GIT necesarios para cada sesión Para que cada grupo pueda acceder al sistema			
Estimación (horas)	34			
Prioridad	Must			
HU Relacionadas	Formar los grupos de forma automática (HU. #3)			
Criterios de aceptación				
ID. del escenario	Título del escenario	Contexto	Evento o condición	Resultado o comportamiento esperado
1	Problemas de conexión a internet	Cuando el equipo el computador donde se formen los grupos no pueda comunicarse correctamente con la plataforma de alojamiento de repositorios GIT	Cuando se presione el botón 'OK' de la interface de TEAM-BE	El sistema mostrará un mensaje de error indicando que los repositorios no pudieron ser creados
2	Conexión a internet disponible	Cuando el equipo el computador donde se formen los grupos tenga una conexión estable a internet	Cuando se presione el botón 'OK' de la interface de TEAM-BE	El sistema creará los repositorios GIT necesarios para cada la sesión de cada grupo

Fuente: Creación propia

Tabla 14. Historia de usuario #6

ID Historia	HU. #6			
Nombre	Crear los usuarios y las sesiones en Collece de forma automática			
Actor	Administrador			
Descripción	Como Administrador Quiero crear los usuarios y las sesiones en Collece de forma automática Para que los participantes puedan acceder al sistema			
Estimación (horas)	34			
Prioridad	Must			
HU Relacionadas	Crear repositorios GIT necesarios (HU. #5)			
Criterios de aceptación				
ID. del escenario	Título del escenario	Contexto	Evento o condición	Resultado o comportamiento esperado
1	Grupos no formados	Cuando aún no se han formado los grupos	Cuando se presione el botón 'OK' de la interface de TEAM-BE	El sistema mostrará un mensaje diciendo 'No se pueden crear las sesiones porque los grupos no están formados'
2	Grupos formados	Cuando ya se han formado los grupos	Cuando se presione el botón 'OK' de la interface de TEAM-BE	El sistema registrará los grupos y sus correspondientes usuarios

Fuente: Creación propia

Tabla 15. Historia de usuario #7

ID Historia	HU. #7			
Nombre	Instalar el software de forma sencilla			
Actor	Usuario			
Descripción	Como Usuario			
	Quiero instalar el software de forma sencilla			
	Para desarrollar actividades de programación colaborativa			
Estimación (horas)	34			
Prioridad	Must			
Criterios de aceptación				
ID. del escenario	Título del escenario	Contexto	Evento o condición	Resultado o comportamiento esperado
1	Instalación correcta	Cuando el usuario intente instalar el plugin Collece 2.0 con el módulo de formación automática de grupos TEAM-BE	N/A	El usuario podrá instalar el software por medio del gestor de plugins del IDE Eclipse o descargar un paquete preconfigurado con todo el software instalado

Fuente: Creación propia

Tabla 16. Historia de usuario #8

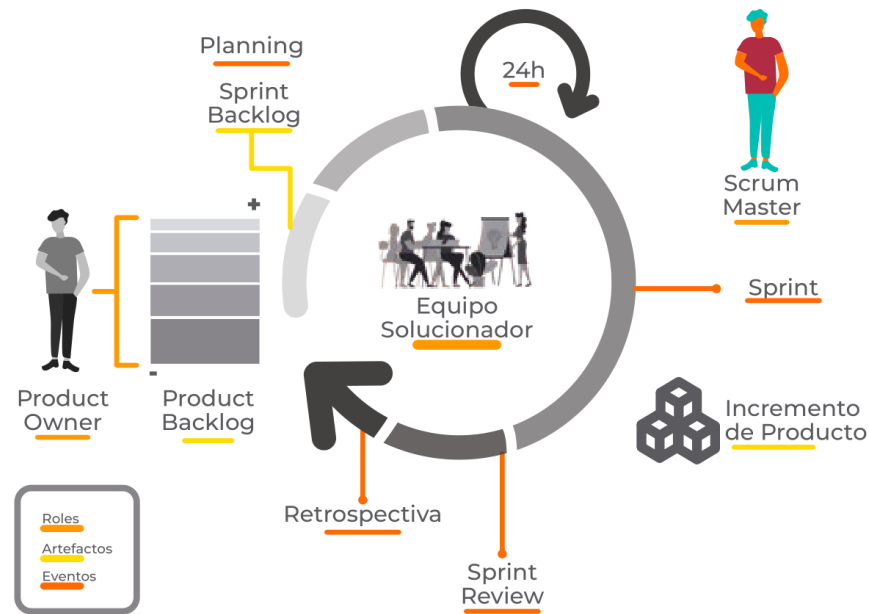
ID Historia	HU. #8			
Nombre	Acceder a un manual de instalación y uso del software			
Actor	Usuario			
Descripción	Como Usuario			
	Quiero acceder a un manual de instalación y uso del software			
	Para desarrollar actividades de programación colaborativa			
Estimación (horas)	34			
Prioridad	Must			
Criterios de aceptación				
ID. del escenario	Título del escenario	Contexto	Evento o condición	Resultado o comportamiento esperado
1	Manual disponible	Cuando el usuario necesite acceder a un manual de instrucciones	N/A	El usuario tendrá a disposición un manual con instrucciones de uso del plugin Collece 2.0 y el módulo TEAM-BE por medio de una página WEB

Fuente: Creación propia

2.5.3.4 Incremento o Entregable. Al final de cada *Sprint* se generó un avance (incremento) el cuál se socializó con los *Product Owner* en el *Sprint Review*, en estos eventos se determinó si el incremento cumplía con las expectativas y se hacían ajustes a la pila del producto (*Product backlog*) de ser necesario agregando requerimientos o funcionalidades.

Teniendo en cuenta lo anterior, el proceso SCRUM se implementó como se indica en la figura 4.

Figura 4. Proceso general de SCRUM

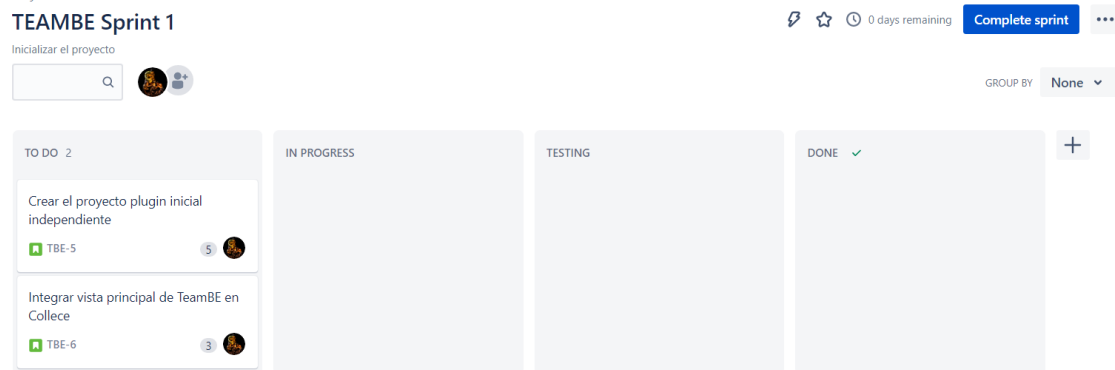


Fuente: Guía funcional de SCRUM⁵⁸

Cada *Sprint* inicia con el evento *Sprint Planning* en el cual se toman historias de usuario o funcionalidades de la pila del *backlog* que serán trabajadas en el transcurso del *sprint*, también pueden crearse nuevas historias de usuario de ser necesario. Una vez se inicial el *Sprint* se pasa a trabajar en el tablero que se muestra en la figura 5.

⁵⁸ ACADEMIA PRAGMA. Guía funcional de SCRUM. 1 a ed. Bogotá: Academia Pragma, 2020.

Figura 5. Tablero proceso SCRUM de TEAM-BE en Jira Software



Fuente: Creación propia

Las casillas de este tablero pueden cambiar en función a las características del proyecto a desarrollar, en este caso se utilizaron cuatro para describir el estado de una historia de usuario que se describen a continuación:

TO DO: ítems que no han sido iniciados aún

IN PROGRESS: ítems que se encuentran en desarrollo

TESTING: ítems que están terminados y se encuentran en pruebas, en caso de encontrarse errores deben volver al estado *IN PROGRESS*

DONE: ítems que han sido terminados y las probados con resultados satisfactorios

Cuando todos los ítems se encuentren en la casilla *DONE* el *Sprint* se considera finalizado y los avances logrados en este se discuten en el evento *Sprint Review*.

2.6 PRUEBAS DEL SOFTWARE

Para facilitar la evolución en código constante de la herramienta sin perjudicar los avances anteriores es necesario realizar pruebas constantemente de partes puntuales de TEAM-BE y su funcionamiento en conjunto con Collece 2.0.

En este caso durante el proceso se utilizó *JUnit*, un conjunto de bibliotecas especializadas para pruebas unitarias automatizadas en Java, su funcionamiento se basa principalmente en comparar datos esperados con los generados por el código. Estos procesos toman muy poco tiempo en comparación a su realización manual y pueden repetirse de forma indefinida lo cual facilita la detección de errores de forma temprana y optimiza el proceso de desarrollo en general.

2.7 LENGUAJE DE MODELADO

Para el desarrollo se utilizó UML, definido en la web de *Visual Paradigm* como “un lenguaje estandarizado que consiste en un conjunto de diagramas pensados para especificar, visualizar, construir y documentar artefactos de sistemas software”⁵⁹ . Es ampliamente utilizado al estar estrechamente relacionado con el paradigma orientado a objetos lo que facilita la comprensión del sistema tanto para el equipo de desarrollo como para personas externas al utilizar notaciones y conceptos generales conocidos incluso por personas con pocos conocimientos técnicos.

⁵⁹ VISUAL PARADIGM. "What is Unified Modeling Language (UML)?". [en línea]. 2019.

3. DESARROLLO DE LA INVESTIGACIÓN

En este capítulo se documenta el proceso de desarrollo de la herramienta software TEAM-BE, empezando por el análisis y diseño de la herramienta, posteriormente se describe el software desarrollado y las pruebas realizadas sobre él.

3.1 ANÁLISIS, DISEÑO Y DESARROLLO

A continuación, se modela el sistema a desarrollar por medio de descripción de sus objetivos, requerimientos, casos de uso, participantes y arquitectura para lo anterior se utilizó el lenguaje de modelado UML.

3.1.1 Objetivos del sistema. A continuación, se describen los objetivos del sistema teniendo en cuenta la funcionalidad que aporta TEAM-BE al entorno Collece 2.0.

En la tabla 17 se describe el objetivo principal de la herramienta software a desarrollar.

Tabla 17. Objetivo OBJ-001 TEAM-BE

OBJ-001	Permitir la formación de grupos dentro directamente desde Collece 2.0
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Descripción	La herramienta construida debe permitir formar grupos de trabajo directamente desde la interfaz de creación de sesiones de Collece 2.0 utilizando la librería Java TEAM-B
Sub-objetivos	Ninguno
Importancia	Vital
Urgencia	Inmediata
Estado	Implementado
Estabilidad	Alta
Comentarios	Ninguno

Fuente: Creación propia

La tabla 18 describe un objetivo secundario que apunta a agregar valor a la herramienta desarrollada.

Tabla 18. Objetivo OBJ-002 TEAM-BE

OBJ-002	Automatizar todas las tareas implicadas en la creación de sesiones de Collece 2.0
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Descripción	La herramienta debe automatizar al máximo el proceso de creación de sesiones en Collece 2.0 a partir de los grupos formados automáticamente, esto incluye crear los repositorios GIT remotos y registrar las sesiones y sus correspondientes usuarios en el servidor
Sub-objetivos	Ninguno
Importancia	Vital
Urgencia	Inmediata
Estado	Implementado
Estabilidad	Alta
Comentarios	Ninguno

Fuente: Creación propia

3.1.2 Requerimientos del sistema. Durante las reuniones iniciales y las celebradas posteriormente con el equipo de desarrollo de Collece 2.0 y el creador de la librería TEAM-B se recopilaron las necesidades que se especifican a continuación, también se tuvo en cuenta los objetivos y la delimitación del proyecto. Se utilizó como guía las recomendaciones basadas en el estándar IEEE 830-1998 de Arévalo, et al⁶⁰.

⁶⁰ ARÉVALO, Olga Esperanza, LINARES, Sonia Patricia, CORREA, Luisa Fernanda, OLARTE PARRA, Julieth, y GONZALEZ, Héctor Antonio. IEEE-STD-830-1998: PRÁCTICA RECOMENDADA PARA LAS ESPECIFICACIONES DE REQUISITOS DEL SOFTWARE. 2008, 41p. Universidad Nacional de Colombia.

3.1.2.1 Requisitos de información. A continuación, se describen los requisitos relacionados con las entradas y salidas de información del sistema.

La tabla 19 describe un requisito que apunta a facilitar la realización de actividades de programación en ambientes educativos al permitir que el organizador de la actividad tenga un reporte de los grupos formados con sus respectivos integrantes.

Tabla 19. Requisito IRQ-001 TEAM-BE

IRQ-001	Exportar información con los grupos formados automáticamente en formato PDF
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Dependencias	Diagrama CU general TEAM-BE
Datos específicos	Identificador del grupo Nombre y correo electrónico de los integrantes
Ocurrencias simultaneas	Baja 1 Cantidad media de ocurrencias simultaneas
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Estado	Implementado
Comentarios	Ninguno

Fuente: Creación propia

Debido a que el tiempo de procesamiento varía dependiendo de la capacidad de la máquina donde se ejecute TEAM-BE, es necesario mostrar al usuario el estado actual del proceso de formación de grupos. Este requisito se detalla en la tabla 20.

Tabla 20. Requisito IRQ-002 TEAM-BE

IRQ-002	Visualizar el progreso de la formación de grupos en tiempo real
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Dependencias	Diagrama CU-001: Creación de sesiones con TEAM-BE
Datos específicos	Porcentaje del progreso
Ocurrencias	Baja
simultaneas	1
	Cantidad media de ocurrencias simultaneas
Importancia	Baja
Urgencia	Baja
Estabilidad	Alta
Estado	Implementado
Comentarios	Ninguno

Fuente: Creación propia

3.1.2.1 Requisitos de interfaz. A continuación, se describen los requisitos referentes a la interfaz gráfica de usuario del módulo TEAM-BE.

En la tabla 21 para el requisito INRQ-001 se detallan los elementos visuales que van a componer la interfaz de usuario de la herramienta.

Tabla 21. Requisito de interfaz INRQ-001 TEAM-BE

INRQ-001	Interfaz de usuario
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Descripción	La interfaz de usuario permite ingresar toda la información necesaria para el proceso de formación automática de grupos, así como visualizar el resultado, lo anterior se consigue por medio de: <ul style="list-style-type: none"> • Botones • Cuadros de texto • Casillas de verificación • Mensajes de error • Listados • Vistas de árbol (<i>TreeView</i>)
Importancia	Vital
Urgencia	Inmediata
Estabilidad	Alta
Estado	Implementado
Comentarios	Ninguno

Fuente: Creación propia

En la tabla 22 para el requisito INRQ-002 se explica la necesidad de una barra de progreso para el proceso de formación automática de grupos.

Tabla 22. Requisito de interfaz INRQ-002 TEAM-BE

INRQ-002	Barra de progreso
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Descripción	El proceso de formación de grupos realizado por la librería TEAM-B puede durar algunos segundos o minutos dependiendo de la cantidad de individuos y el tamaño del grupo, por lo tanto, es necesario que la interfaz informe el progreso de la operación en tiempo real
Importancia	Vital
Urgencia	Alta
Estabilidad	Alta
Estado	Implementado
Comentarios	Ninguno

Fuente: Creación propia

3.1.2.2 Requisitos funcionales. En esta sección se muestran los requisitos esenciales para la herramienta software a desarrollar que se encontraron en el proceso de análisis.

La tabla 23 detalla el requisito TEAM-BE_RF_001 referente a mantener la característica de interfaz multilenguaje ya presente en el entorno Collece 2.0 para la herramienta software desarrollada.

Tabla 23. Requisito funcional TEAM-BE_RF_001

TEAM-BE_RF_001	Interfaz multilenguaje
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Dependencias	Diagrama CU general TEAM-BE
Descripción	La herramienta debe adaptarse a la compatibilidad para múltiples lenguajes de Collece 2.0, de modo que cuando se ejecute detecte todos los textos y etiquetas se muestren en el mismo lenguaje de Collece (el cual establece el lenguaje en base al del equipo donde se ejecute).
Ourrencias simultaneas	Baja 1
Importancia	Cantidad media de ourrencias simultaneas
Urgencia	Alta
Estabilidad	Inmediata
Estado	Alta
Comentarios	Implementado
	Ninguno

Fuente: Creación propia

La tabla 24 detalla el requisito TEAM-BE_RF_002 referente a la visualización por parte del usuario del resultado del proceso de formación automática de grupos.

Tabla 24. Requisito funcional TEAM-BE_RF_002

TEAM-BE_RF_002	Visualizar los grupos formados
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Dependencias	Diagrama CU-001: Creación de sesiones con TEAM-BE
Descripción	Al finalizar el proceso de formación de grupos la interfaz debe presentar el resultado visualmente al usuario de forma que sea fácilmente entendible, también debe permitir realizar la formación de grupos de nuevo si el usuario lo desea
Ocurrencias simultaneas	Baja 1 Cantidad media de ocurrencias simultaneas
Importancia	Alta
Urgencia	Inmediata
Estabilidad	Alta
Estado	Implementado
Comentarios	Ninguno

Fuente: Creación propia

3.1.2.3 Requisitos no funcionales. En esta sección se describen las propiedades o cualidades que debe tener la herramienta a desarrollar, lo cual se determinó que en la fase de análisis.

La tabla 25 describe el requerimiento no funcional FNR-001 referente a la portabilidad que debe tener la herramienta en distintos sistemas operativos.

Tabla 25. Requisito no funcional FNR-001

FNR-001	Portabilidad
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Dependencias	
Descripción	La herramienta desarrollada hace parte de Collece 2.0, el cual a su vez es parte de Eclipse IDE, todo lo anterior se encuentra desarrollado en diversos <i>frameworks</i> de Java, esto garantiza en gran medida la portabilidad, pues eclipse (y por tanto Collece y TEAM-BE) está disponible para los sistemas operativos más utilizados: Windows, GNU/Linux y MacOS.
Importancia	Alta
Urgencia	Inmediata
Estabilidad	Alta
Estado	Implementado
Comentarios	Ninguno

Fuente: Creación propia

La tabla 26 describe el requisito no funcional FNR-002 referente al grado de satisfacción de la herramienta software desarrollada.

Tabla 26. Requisito no funcional FNR-002

FNR-002	Correctitud
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Dependencias	
Descripción	La herramienta desarrollada cumple completamente con el grado de satisfacción del producto
Importancia	Vital
Urgencia	Alta
Estabilidad	Alta
Estado	Implementado
Comentarios	Ninguno

Fuente: Creación propia

La tabla 27 describe el requisito no funcional FNR-003 referente a la mantenibilidad y extensibilidad que debe tener la herramienta desarrollada.

Tabla 27. Requisito no funcional FNR-003

FNR-003	Mantenibilidad
Versión	1.0 (25/10/2020)
Autores	Juan Pablo Botina Carlosama
Fuentes	Grupo Galeras.NET
Dependencias	
Descripción	El código de la herramienta desarrollada al igual que Collece 2.0 es altamente modular e incluye documentación en las partes que lo requieren lo cual facilita su mantenibilidad y la adición de nuevas funcionalidades por parte de otros desarrolladores
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Estado	Implementado
Comentarios	Ninguno

Fuente: Creación propia

3.1.3 Participantes. En esta sección se describen todos las personas y entidades involucradas en la realización de este proyecto.

En la tabla se 28 detalla al participante 1, que desarrolla la herramienta software TEAM-BE como parte de su trabajo de grado.

Tabla 28. Datos de participante 1

Organización	Universidad de Nariño - Grupo Galeras.NET
Es Desarrollador	SI
Es Cliente	NO
Es Usuario	SI
Comentarios	<ul style="list-style-type: none"> • Lleva a cabo el desarrollo e implementación del sistema • Participa en la toma de decisiones • Provee soporte técnico a los usuarios finales

Fuente: Creación propia

En la tabla 29 se detalla al participante 2, quien es el principal asesor en este proyecto.

Tabla 29. Datos de participante 2

Organización	Universidad de Nariño - Grupo Galeras.NET
Es Desarrollador	NO
Es Cliente	NO
Es Usuario	SI
Comentarios	<ul style="list-style-type: none"> • Es propietario de TEAM-B • Provee apoyo técnico y asesoría al desarrollador • Participa en la toma de decisiones • Monitorea el progreso del desarrollo

Fuente: Creación propia

En la tabla 30 se detalla al participante 3 quien provee apoyo técnico en calidad de co-asesor de este proyecto.

Tabla 30. Datos de participante 3

Organización	Universidad de Nariño - Grupo Galeras.NET
Es Desarrollador	NO
Es Cliente	NO
Es Usuario	SI
Comentarios	<ul style="list-style-type: none"> • Provee apoyo técnico y asesoría al desarrollador

Fuente: Creación propia

En la tabla 31 se detalla al participante 4, quien representa al equipo de desarrollo de Collece 2.0.

Tabla 31. Datos de participante 4

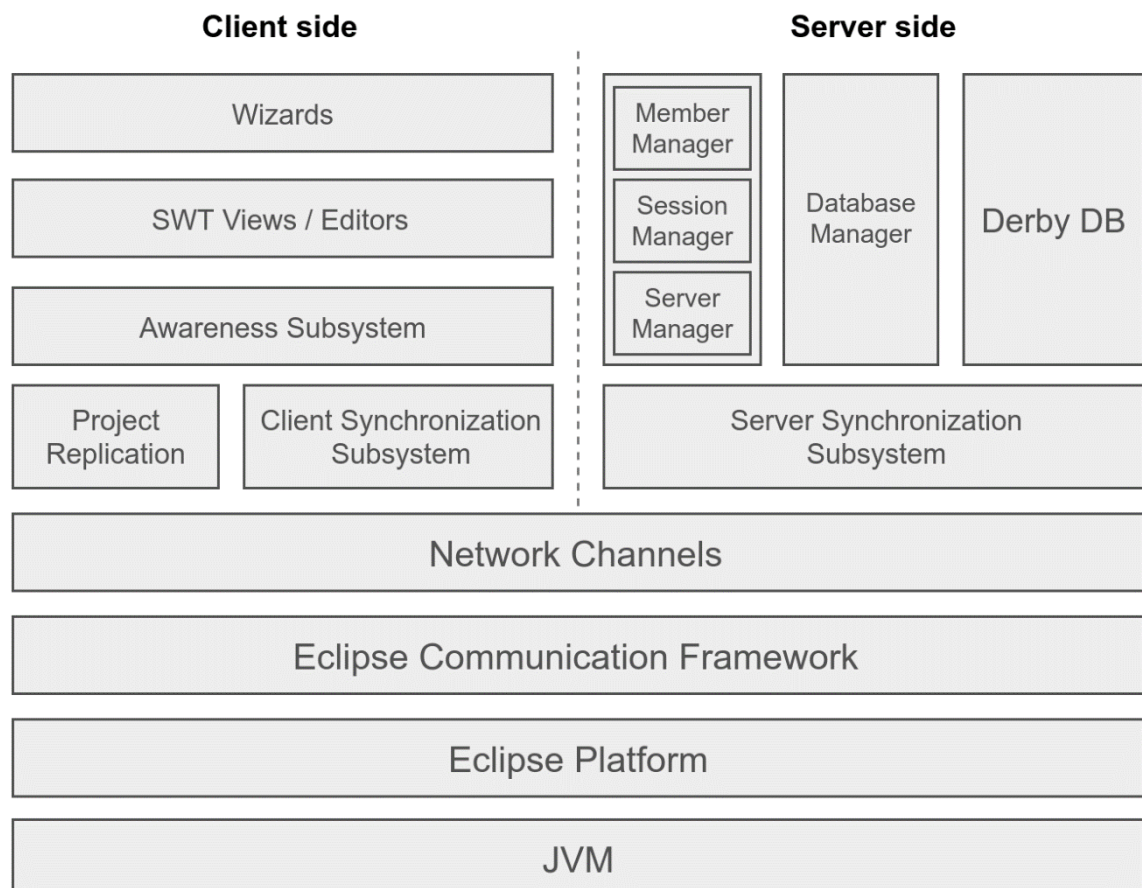
Organización	Universidad de Castilla la Mancha- Grupo de investigación CHICO
Es Desarrollador	NO
Es Cliente	SI
Es Usuario	SI
Comentarios	<ul style="list-style-type: none"> • Es propietario de <i>plugin</i> Collece 2.0 • Provee apoyo técnico y asesoría al desarrollador • Participa en la toma de decisiones

Fuente: Creación propia

3.1.4 Arquitectura del sistema. En este caso la herramienta TEAM-BE es parte de un software más complejo (Collece 2.0), por lo tanto, fue necesario estudiar la arquitectura de este por medio de la documentación existente para obtener una visión clara de la estructura de debe tener el software a desarrollar, en este sentido también es necesario presentar algunos aspectos generales del funcionamiento de la librería TEAM-BE.

3.1.4.1 Collece 2.0. El *plugin* funciona con una arquitectura cliente – servidor, una instancia de servidor puede alojar varias sesiones, cada una con sus correspondientes usuarios al mismo tiempo, las sesiones se crean y gestionan desde las instancias cliente y el servidor mantiene la información en una base de datos Derby local. La figura 6 muestra las distintas capas lógicas de Collece 2.0 y componentes de las instancias cliente y servidor.

Figura 6. Estructura general de Collece 2.0



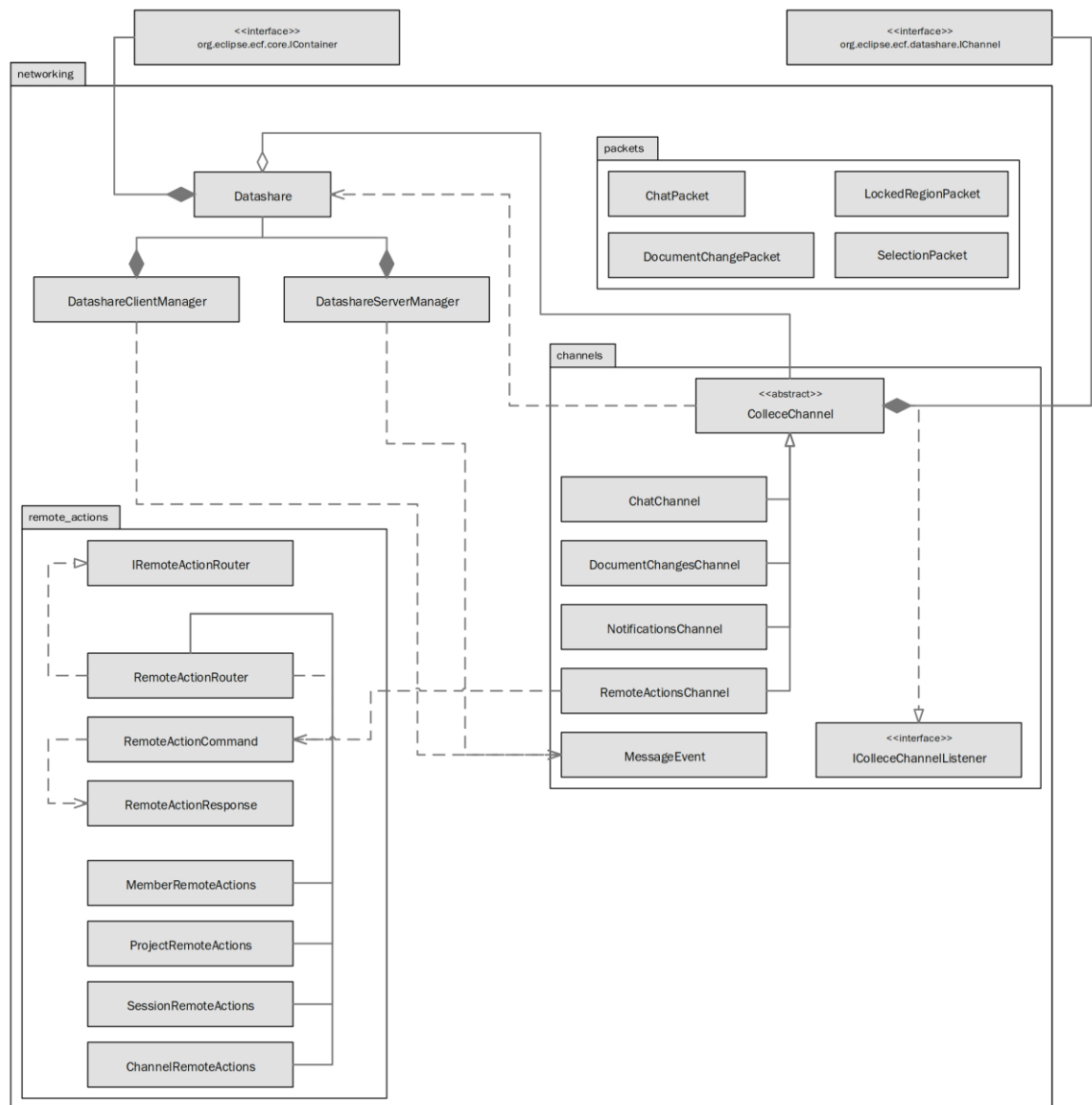
Fuente: Documentación Collece 2.0

La comunicación se realiza por medio del Marco de Comunicaciones de Eclipse (*Eclipse Communication Framework*), específicamente haciendo uso del paquete *DataShare*, un conjunto de clases que facilitan el intercambio de información por la red mediante mensajes con un formato específico que contienen la representación de un objeto Java, lo cual facilita la comunicación entre servidor y clientes. Estos mensajes se agrupan en varias categorías denominadas canales, los principales se listan a continuación:

- Canal de chat: transporta mensajes de texto enviados por los usuarios de una sesión
- Canal de actualizaciones de documento: transporta información sobre cambios hechos por un usuario a un archivo fuente
- Canal de acciones remotas: transporta mensajes con parámetros relacionados a un comando (por ejemplo, al registrar un usuario los parámetros serían el nombre, correo electrónico y contraseña). Este canal es utilizado por TEAM-BE en el proceso de registro automático de sesiones y usuarios.

Los canales descritos anteriormente son parte del paquete *networking* que puede observarse en la figura 7.

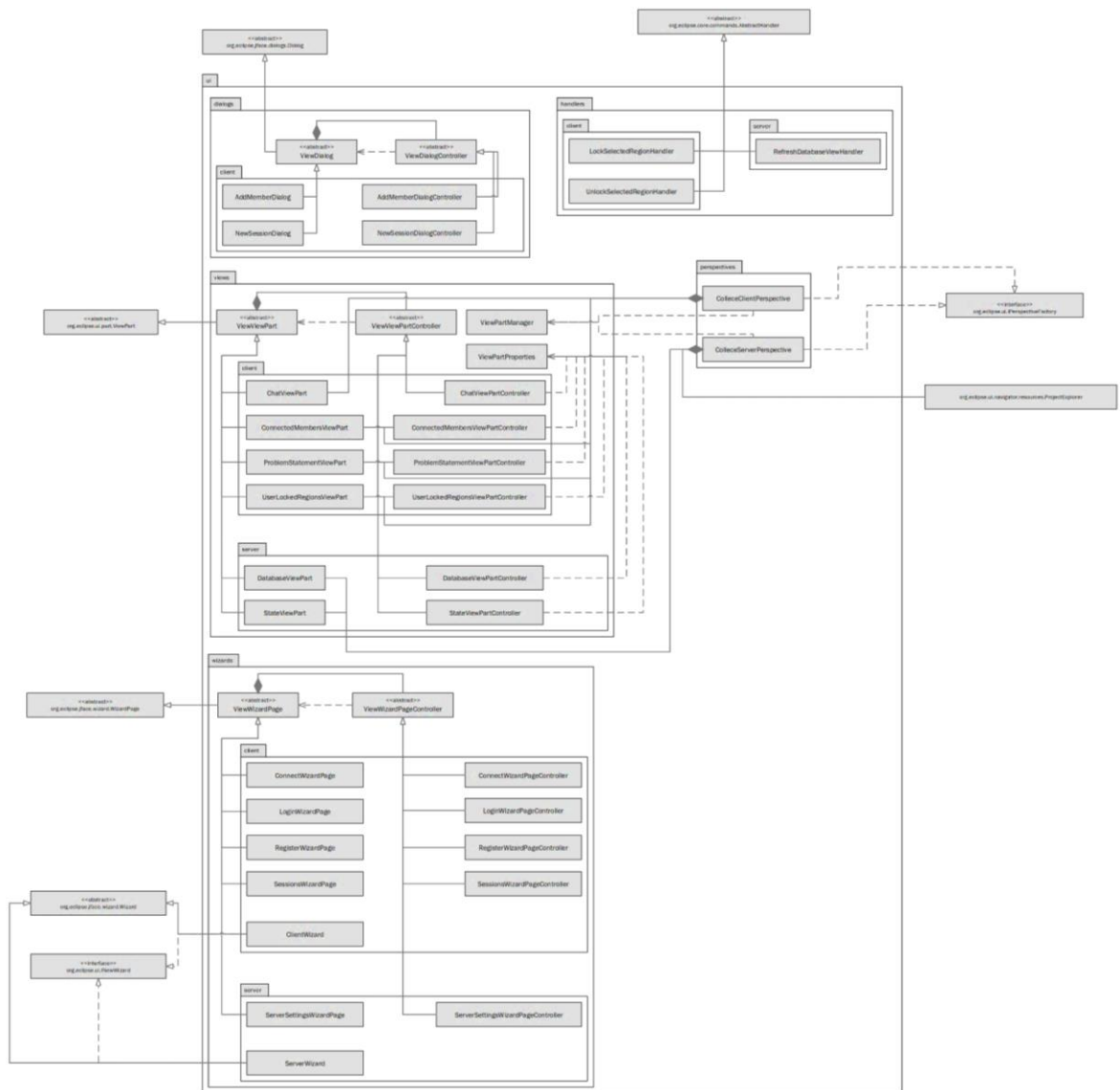
Figura 7. Paquete networking de Collece 2.0



Fuente: Diagrama de clases de Collece 2.0

En cuanto a la interfaz gráfica los componentes se agrupan principalmente en cuadros de dialogo (*dialogs*), vistas (*views*) y asistentes de configuración (*wizards*). Estos elementos y sus relaciones se muestran en el diagrama de clases de la figura 8.

Figura 8. Paquete ui de Collece 2.0



Fuente: Diagrama de clases de Collece 2.0

3.1.4.2 TEAM-B. Esta librería java forma grupos a partir de las características de los individuos involucrados utilizando un método de búsqueda heurística implementando un algoritmo genético como técnica de optimización para mantener tiempos de ejecución razonables⁶¹. Es importante aclarar que al implementar algoritmos genéticos el resultado del proceso no siempre será el mismo para los mismos datos de entrada. Las clases utilizadas en este proyecto se detallan a continuación:

- **Data:** contenedor para los parámetros para el proceso de formación de grupos, realiza validaciones de forma interna y comunica si existen errores por medio de excepciones
- **GA (Genetic Algorithm):** mediante esta clase se realiza el proceso principal, al finalizar entrega el resultado por medio de una matriz donde cada fila es un grupo y cada columna es el identificador de un usuario

La librería funciona en base a la información de los individuos junto con unos parámetros específicos que se detallan a continuación:

- **Tamaño del grupo:** número entero que indica cantidad de individuos que tendrá cada grupo
- **Tipo de agrupamiento:** número entero que indica el tipo de agrupamiento a utilizar (0: Homogéneo, 1: Heterogéneo, 2: Mixto)
- **Características homogéneas:** arreglo de números enteros, contiene la información de las características que se consideran homogéneas (para cada posición un 1 significa que la característica en consideración es homogénea, de lo contrario será un 0)
- **Características heterogéneas:** arreglo de números enteros, contiene la información de las características que se consideran heterogéneas, funciona bajo la misma lógica que el arreglo de características homogéneas
- **Archivo Información de los participantes:** ruta al archivo de texto plano separado por comas (CSV) que contiene los datos de los individuos que formarán los grupos incluyendo las características cuantificadas a considerar.

⁶¹ REVELO SÁNCHEZ, Oscar, INSUASTI PORTILLA, Jesús, y BARÓN SALAZA, Alexander. TEAM-B v2.0 Librería Java para la formación homogénea, heterogénea y mixta de grupos, basada en múltiples características. Pasto, 2019, Universidad de Nariño. Facultad de Ingeniería.

En la tabla 32 se incluye una fila de ejemplo con la estructura para cada línea del archivo.

Tabla 32. Ejemplo de archivo CSV para la librería TEAM-B

ID	NOMBRE	CORREO	C1	C2	C3	C4	C5
6	INDIVIDUO 6	i6@mail.com	4.000	3.000	3.7778	3.125	4.400

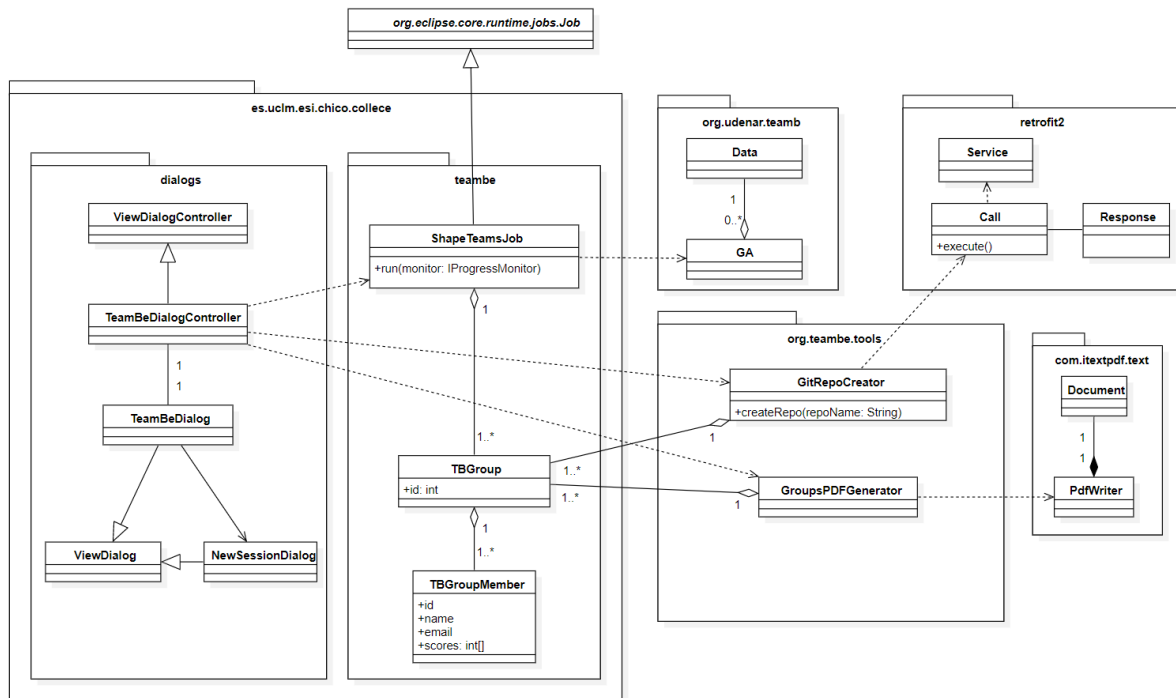
Fuente: Creación propia

3.1.4.3 Modulo TEAM-BE. El módulo consiste en varias clases agregadas a la estructura de Collece 2.0, principalmente a la interfaz de usuario de creación de sesiones del lado del cliente (paquete dialogs). A continuación, se resumen las clases principales:

- **TeamBeDialog:** se encarga de generar la vista principal de TEAM-BE (ver figura 14), este diálogo se lanza desde *NewSessionDialog* si el usuario habilita la formación automática de grupos
- **TeamBeDialogController:** Está asociada directamente a *TeamBeDialog*, se encarga de ejecutar las acciones iniciadas desde la interfaz de usuario tales como cargar los datos y establecer parámetros. En esta clase se realiza el proceso de formación de grupos por medio de un subproceso denominado *ShapeTeamsJob* que hace uso de la librería TEAM-B en segundo plano mientras informa el avance del proceso en la interfaz
- **GitRepoCreator:** Se encarga de conectarse a la API de Bitbucket para crear los repositorios remotos necesarios para cada sesión por medio de la librería *retrofit2*
- **GroupsPDFGenerator:** Permite generar un archivo PDF con el listado de grupos formados por TEAM-B, para ello utiliza la librería *itextpdf*
- **TBGroup:** representa un grupo formado automáticamente por la librería TEAM-B, contiene uno o varios individuos (clase *TBGroupMember*), sirve para facilitar el intercambio de información entre la librería TEAM-B, el controlador de la interfaz (*TeamBeDialogController*), la clase encargada de crear los repositorios GIT (*GitRepoCreator*) y la que genera el reporte en PDF (*GroupsPDFGenerator*), contiene un identificador único para facilitar su diferenciación en la interfaz y en el código
- **TBGroupMember:** representa un individuo perteneciente a un grupo, contiene la información personal (nombre y correo electrónico) y un arreglo con los valores numéricos

La figura 9 muestra el diagrama de clases del módulo TEAM-BE y su relación con los paquetes existentes de Collece 2.0.

Figura 9. Diagrama de clases del módulo TEAM-BE



Fuente: Creación propia

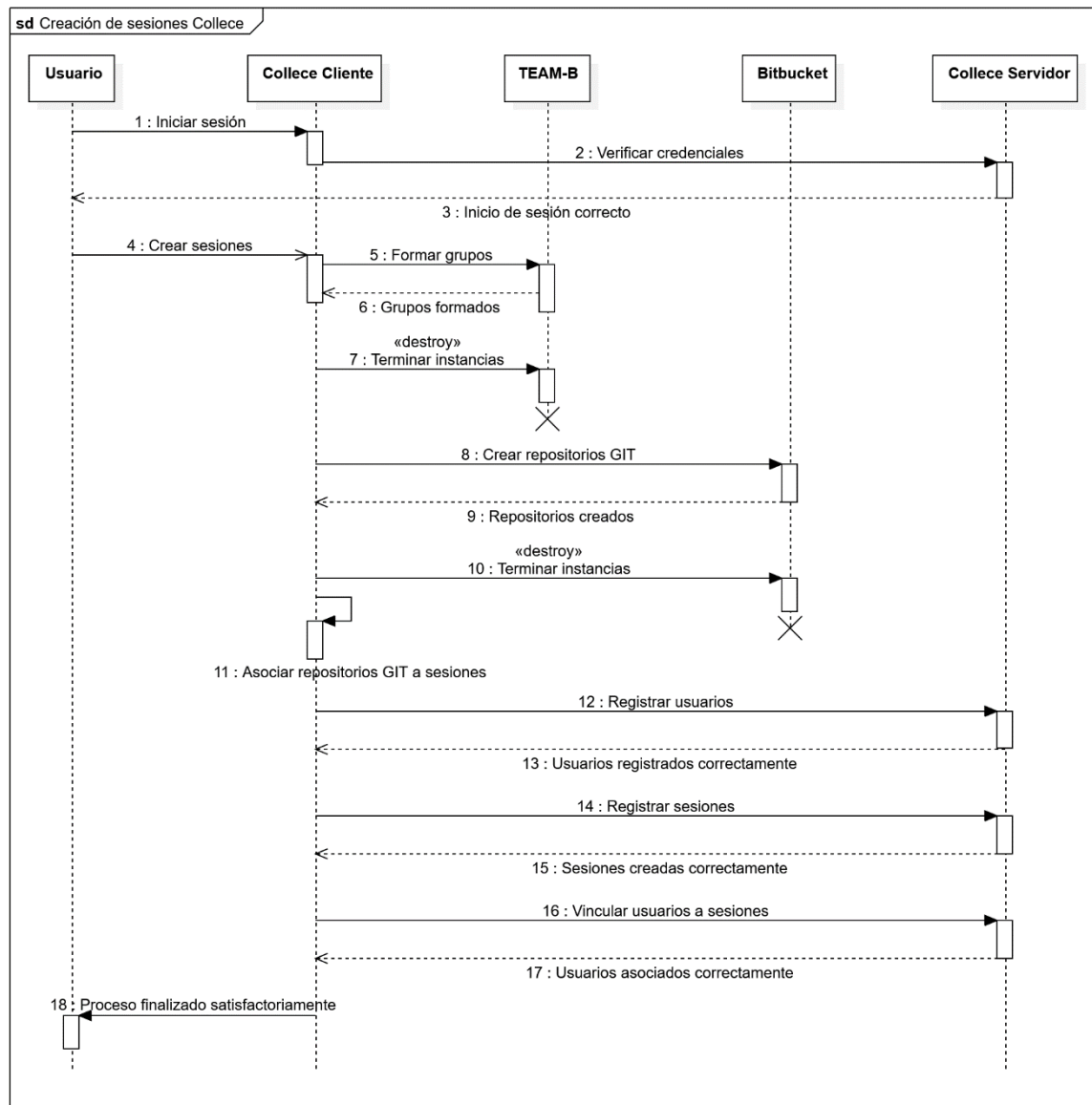
El módulo entra en funcionamiento en el proceso de creación de sesiones de Collece 2.0, donde el usuario, una vez ha iniciado sesión en una instancia de servidor, puede crear y unirse a sesiones de trabajo desde una instancia de cliente. Con TEAM-BE se agrega la opción de crear estas sesiones de forma automática tomando los grupos entregados por la librería TEAM-B.

En el caso de habilitar la formación automática de grupos se despliega la interfaz principal de TEAM-BE donde el usuario ingresa los parámetros necesarios para la formación automática de grupos, esos son:

- Tipo de agrupamiento
- Tamaño del grupo
- Especificación de tipo de agrupamiento (homogéneo o heterogéneo) por cada característica
- Archivo de texto plano (CSV) con los datos personales (nombre y correo electrónico) y características representadas en forma numérica de los participantes

Con estos datos se crean y registran los usuarios en el servidor de Collece 2.0, posteriormente las sesiones necesarias y por último se asocia cada usuario en la sesión que le corresponda, el proceso se resume en el diagrama de secuencia presentado en la figura 10.

Figura 10. Diagrama de secuencia de creación de sesiones



Fuente: Creación propia

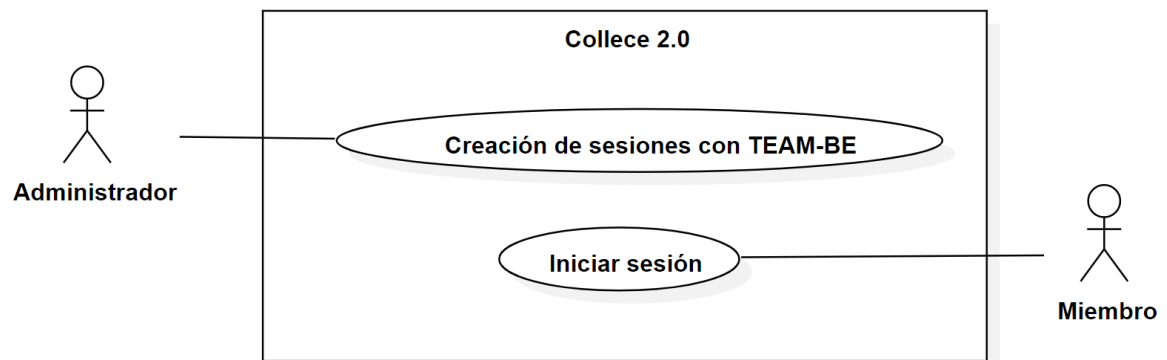
3.1.5 Casos de uso

3.1.5.1 Definición de usuarios. Collece permite que cualquier usuario registrado (haya sido creado por la herramienta TEAM-BE o no) en el servidor pueda crear sesiones, unirse a ellas y administrarlas (es decir eliminarlas, y gestionar sus usuarios) pero no puede modificar ni eliminar sesiones creadas por otros usuarios. Por lo tanto, para una sesión en particular existen dos tipos de usuario:

- **Administrador:** es el usuario que creó la sesión, puede eliminarla, añadir o eliminar usuarios de ella
- **Miembro:** pertenece a la sesión, pudo haber sido creado de forma automática por TEAM-BE, puede conectarse a la sesión a la cual pertenezca y editar código de forma colaborativa

El diagrama de caso de uso mostrado en la figura 11 muestra la relación de los distintos tipos de usuarios con el entorno Collece 2.0.

Figura 11. Diagrama de caso de uso general Collece 2.0



Fuente: Creación propia

En la tabla 33 se detalla el caso de uso CU-001 referente a la creación de sesiones para Collece 2.0 con la herramienta TEAM-BE.

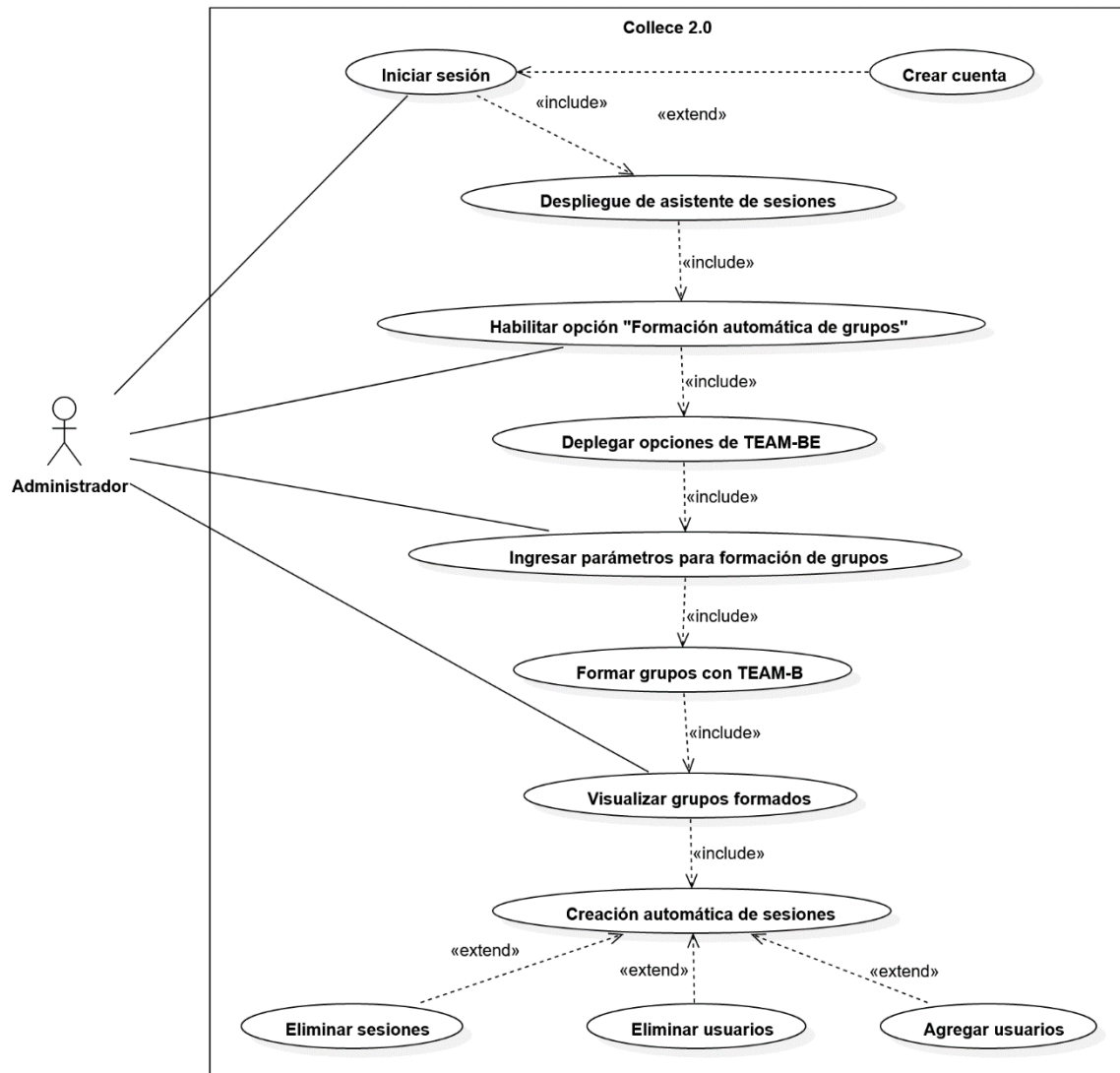
Tabla 33. Descripción CU-001 Creación de sesiones con TEAM-BE

CU-001	Creación de sesiones con TEAM-BE
Actores del sistema	Administrador
Descripción	La herramienta debe permitir la creación automatizada de sesiones en Collece con base en los grupos formados por la librería TEAM-B
Precondiciones	<ul style="list-style-type: none"> La instancia de servidor de Collece 2.0 debe estar ejecutándose en un equipo distinto al que será usado como cliente y ser accesible en red mediante una IP
Postcondiciones	Todas las sesiones han sido creadas
Flujo básico	<p>Paso Acción</p> <ol style="list-style-type: none"> El administrador inicia sesión en Collece desde una instancia cliente El administrador lanza el asistente de creación de sesiones El administrador habilita la opción “Formación automática de grupos” El sistema despliega la interfaz de la herramienta TEAM-BE El administrador ingresa los parámetros y los datos de los participantes para la formación de grupos El sistema forma los grupos por medio de la librería TEAM-B y muestra el resultado en la interfaz El administrador confirma que desea crear las sesiones El sistema registra las sesiones y los usuarios en el servidor de Collece
Flujo alternativo	<p>Paso Acción</p> <ol style="list-style-type: none"> Si el administrador no ha creado su cuenta deberá hacerlo para continuar con el proceso Si el administrador no habilita la formación automática de grupos el asistente de creación de sesiones lo guiará en el proceso de creación manual Si el administrador no desea crear las sesiones con los grupos que ha formado TEAM-B puede repetir el proceso para formar otros grupos o cancelarlo, en cuyo caso será devuelto a la interfaz de gestión de sesiones

Fuente: Creación propia

En la figura 12 se muestran los distintos eventos relacionados con el proceso de creación de sesiones en Collece 2.0 así como las acciones que puede realizar el usuario que administra las sesiones.

Figura 12. Diagrama caso de uso: CU-001 Creación de sesiones con TEAM-BE



Fuente: Creación propia

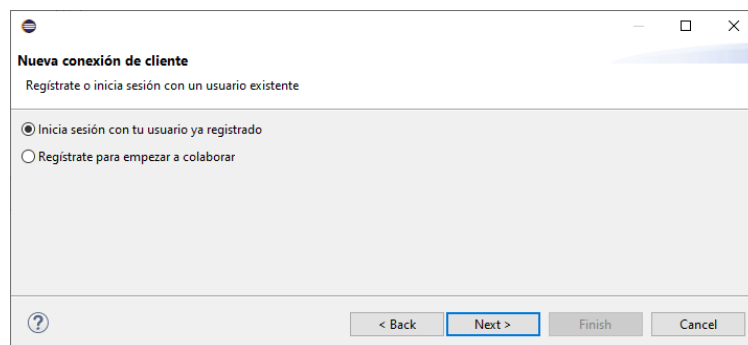
3.2 DESCRIPCIÓN DE HERRAMIENTA DESARROLLADA

Para el desarrollo de la herramienta TEAM-BE se utilizó el lenguaje Java, específicamente el SDK (kit de desarrollo de software) de Eclipse por medio del IDE *Eclipse Committers* (versión 2019-06). Gracias a que Eclipse es multiplataforma el desarrollo se realizó tanto en Windows como en GNU/Linux, así mismo la herramienta puede ser usada en cualquiera de estos sistemas operativos sin que sea necesaria alguna configuración especial.

La herramienta desarrollada es un módulo para el *plugin* Collece 2.0 que permite la formación automática de grupos y automatiza el proceso de registro de usuarios y la creación de sesiones por cada grupo en el sistema junto con los repositorios GIT remotos necesarios para cada una. La vista principal de TEAM-BE hace parte del asistente de creación de sesiones de Collece 2.0 del lado del cliente que se muestra a continuación:

1. El proceso inicia en la vista de inicio de sesión, donde el usuario puede registrarse o iniciar sesión en una cuenta creada previamente en un servidor determinado (los servidores se distinguen por su dirección IP). La figura 13 muestra la interfaz que permite elegir entre iniciar sesión con una cuenta existente o registrarse para crear una nueva.

Figura 13. Opciones de inicio de sesión de Collece 2.0



Fuente: Creación propia

2. Una vez el usuario ha iniciado sesión es dirigido al gestor de sesiones de Collece, donde puede visualizar las sesiones alojadas en el servidor, los usuarios que pertenezcan a cada una y cuáles de ellos se encuentran conectados. También puede crear nuevas sesiones y asignarles usuarios manualmente. En la figura 14 se muestra la interfaz de gestión de sesiones de Collece 2.0, donde un usuario puede crear, eliminar y unirse a las sesiones, así como y agregar o quitar usuarios de ellas.

Figura 14. Asistente de gestión de sesiones de Collece 2.0

Sesiones
Por favor, selecciona una sesión a la que unirte

Autenticado como
user1@uclm.es

¿Pública?	Nombre	Tipo	Creador	URL del repositorio	Fecha de ini...	Fecha de fi...
<input checked="" type="checkbox"/>	Test60...	pub...	user1...	https://bitbucket.org/team...	30/10/18 0...	30/12/22 0...
<input checked="" type="checkbox"/>	Test60...	pub...	user1...	https://bitbucket.org/team...	30/10/18 0...	30/12/22 0...

Nueva... Eliminar Actualizar Unirse

Miembros
INDIVIDUO 1 <individuo_1@mail.com>
INDIVIDUO 2 <individuo_2@mail.com>
INDIVIDUO 3 <individuo_3@mail.com>

Añadir... Eliminar

Miembros Conectados

Desconectar

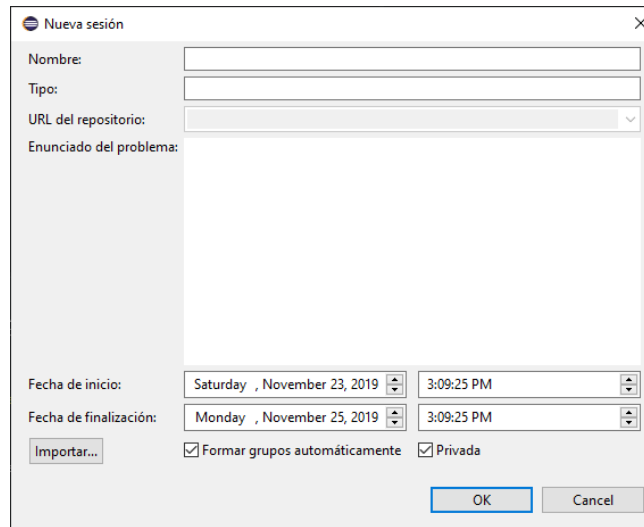
< Back Next > Finish Cancel

Fuente: Creación propia

3. Al hacer clic en la opción nueva sesión se despliega la vista observada en la figura 12. Aquí se ingresan las propiedades de la sesión: nombre, tipo (este campo no se utiliza y se removerá en futuras sesiones, puede estar en blanco), URL del repositorio GIT (al usar TEAM-BE no se habilita pues la herramienta crea los repositorios automáticamente), enunciado del problema (el texto de este campo es compatible con HTML), fechas de inicio y finalización (si la fecha actual está por fuera de ese rango los usuarios no podrán unirse a la sesión). Si la opción “Formar grupos automáticamente” está habilitada al pulsar el botón OK se despliega la vista principal de TEAM-BE.

En la figura 15 se muestra la interfaz de creación de sesiones de Collece 2.0, como se puede observar el campo “URL del repositorio” se deshabilita al marcar la opción “Formar grupos automáticamente” (módulo TEAM-BE) pues al utilizarlo se crearán varios repositorios, uno para cada sesión.

Figura 15. Ventana de creación de sesiones de Collece 2.0



Fuente: Creación propia

En la versión actual de Collece el campo de enunciado no puede editarse directamente, por lo que es necesario importar la sesión desde un archivo *yaml* cuyos campos se muestran en la figura 16, para usar la herramienta TEAM-BE el atributo *teambe-enabled* debe establecerse en *true*. Para importar el archivo se debe hacer clic en el botón importar en la ventana de creación de sesiones.

Figura 16. Ejemplo de archivo de sesión yaml para Collece 2.0

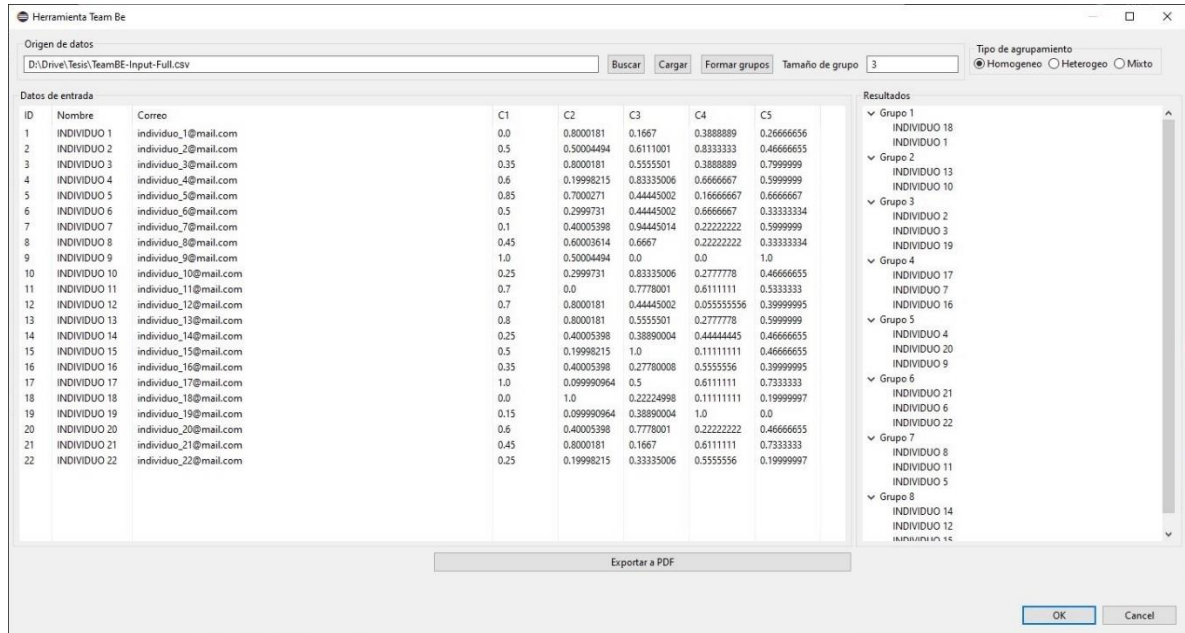
```
1 name: TeamBE Sesión 1
2 type: private
3 problem-statement: Este es un enunciado de prueba
4 date-start: 2020-09-01T00:00:00COT
5 date-end: 2021-03-30T00:00:00COT
6 teambe-enabled: true
7 private: true
```

Fuente: Creación propia

4. Cuando se ha importado correctamente el archivo de sesión y se ha habilitado la formación automática de grupos (campo *teambe-enabled* establecido en *true*) se despliega la vista principal de TEAM-BE observada en la figura 17. Para cargar los datos de los participantes se debe hacer clic en el botón Buscar y localizar el archivo CSV correspondiente, después se debe ingresar el tamaño de grupo y tipo de agrupamiento deseado, luego se debe hacer clic en el botón Cargar para visualizar los integrantes en la interfaz, posteriormente se hace clic en el botón formar grupos, esto iniciará

un subproceso que hace uso de la librería TEAM-B para formar los grupos con los datos proporcionados, el avance del progreso se refleja en una barra de progreso en la interfaz de TEAM-BE.

Figura 17. Vista principal de herramienta TEAM-BE



Fuente: Creación propia

El archivo de texto plano separado por comas (CSV) que contiene la información de los participantes debe tener las siguientes columnas:

1. Identificador (número entero)
2. Nombre (texto)
3. Correo electrónico (texto)
4. Característica 1 (número con decimales)
5. ...
6. Característica N (número con decimales)

En este caso se utilizaron 5 características porque es la cantidad que maneja el test de personalidad BigFive (Extraversión, Cordialidad, Responsabilidad, Estabilidad Emocional, Apertura a la experiencia) pero el código puede adaptarse fácilmente para manejar otra cantidad. En la figura 18 se muestra un ejemplo de archivo con los campos descritos anteriormente.

Figura 18. Archivo CSV de ejemplo para TEAM-BE

```
TeamBE-Input-Full1.csv
1,INDIVIDUO 1,individuo_1@mail.com,2.0000,4.0000,2.5556,2.5000,3.3000
2,INDIVIDUO 2,individuo_2@mail.com,3.2500,3.6667,3.4444,3.5000,3.6000
3,INDIVIDUO 3,individuo_3@mail.com,2.8750,4.0000,3.3333,2.5000,4.1000
4,INDIVIDUO 4,individuo_4@mail.com,3.5000,3.3333,3.8889,3.1250,3.8000
5,INDIVIDUO 5,individuo_5@mail.com,4.1250,3.8889,3.1111,2.0000,3.9000
6,INDIVIDUO 6,individuo_6@mail.com,3.2500,3.4444,3.1111,3.1250,3.4000
7,INDIVIDUO 7,individuo_7@mail.com,2.2500,3.5556,4.1111,2.1250,3.8000
8,INDIVIDUO 8,individuo_8@mail.com,3.1250,3.7778,3.5556,2.1250,3.4000
9,INDIVIDUO 9,individuo_9@mail.com,4.5000,3.6667,2.2222,1.6250,4.4000
```

Fuente: Creación propia

7. Cuando el proceso de formación de grupos ha finalizado es posible generar un archivo PDF con la información tal como se muestra en la figura 19.

Figura 19. Ejemplo de archivo PDF generado por TEAM-BE



Fuente: Creación propia

8. Cuando se han formado correctamente los grupos en la vista principal de TEAM-BE se debe hacer clic en el botón OK, entonces se inicia el proceso de creación automática de sesiones y usuarios, en el siguiente orden:
 - Se registran todos los usuarios en el servidor, independientemente del grupo que se les haya asignado
 - Se crean los repositorios GIT remotos para cada sesión en Bitbucket mediante la API pública

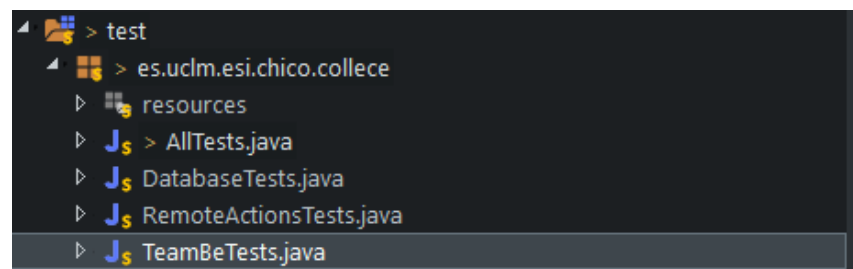
- Se registran una sesión por cada grupo formado
- Se registra cada usuario en la sesión correspondiente al grupo que le fue asignado

Cuando termine el proceso descrito los usuarios pueden unirse a la sesión que les corresponda desde una instancia cliente de Collece 2.0 usando el correo electrónico como usuario y contraseña.

3.3 PRUEBAS DE SOFTWARE

Se realizaron pruebas unitarias utilizando el *framework* JUnit integrado en el SDK de Eclipse, se creó la clase *TeamBeTests* en el paquete de pruebas existente de Collece (ver figura 20). La clase *AllTest* ejecuta todas las pruebas unitarias existentes incluyendo las del módulo TEAM-BE.

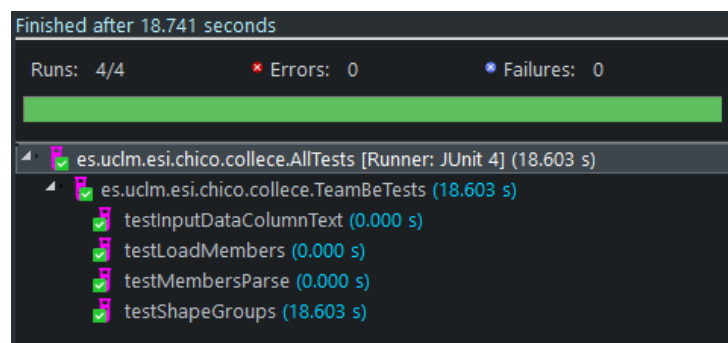
Figura 20. Paquete de pruebas Collece 2.0



Fuente: creación propia

Estas pruebas se ejecutaron continuamente durante el desarrollo para detectar si la adición de nuevas características afectaba las que habían sido terminadas, y corregirlo si fuera el caso, en la figura 21 se observa la última ejecución realizada de las pruebas mencionadas.

Figura 21. Ejecución de pruebas módulo TEAM-BE



Fuente: Creación propia

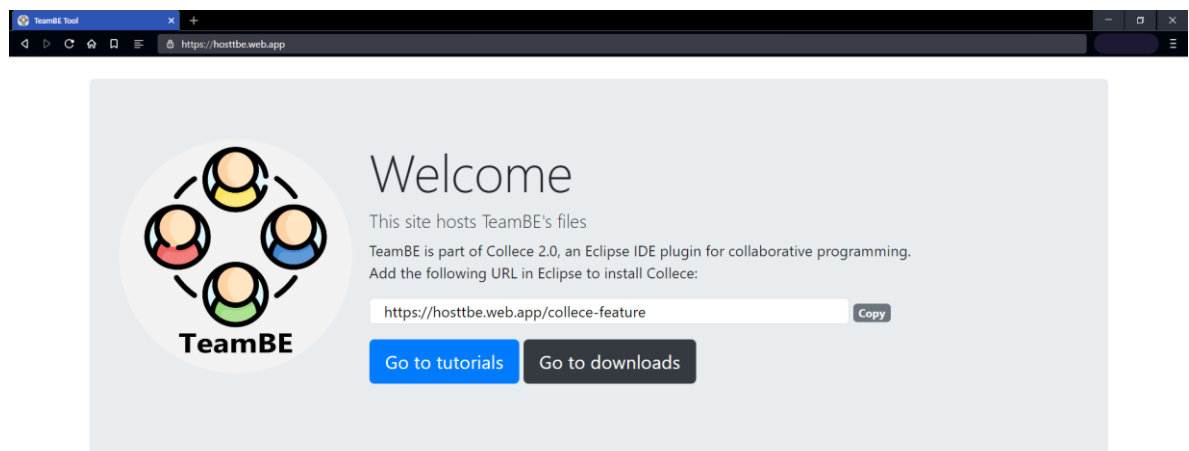
5. RESULTADOS DE LA INVESTIGACIÓN

La presente investigación permitió desarrollar una herramienta llamada TEAM-BE que permite realizar el proceso de formación automática de grupos desde la interfaz del *plugin* para Eclipse Collece 2.0, para ello se implementó la librería Java Team-B desarrollada en la Universidad de Nariño. TEAM-BE automatiza varias tareas como el registro de sesiones y usuarios en la instancia de servidor de Collece 2.0 así como la creación de repositorios GIT remotos necesarios para cada sesión de modo que al finalizar el proceso se tienen grupos formados en torno a ciertas características cuantificables (que podrían ser, por ejemplo, los rasgos de personalidad), cada uno con su correspondiente sesión privada.

Para facilitar la instalación y uso de la herramienta se elaboró un manual de usuario detallado (ver Anexo 6).

Se creó un sitio web (<https://hosttbe.web.app>) donde se encuentran los archivos necesarios para instalar Collece 2.0 con el módulo TEAM-BE desde el gestor de *plugins* de Eclipse IDE, así como descargar paquetes zip con el software portable preinstalado y listo para usar. En el sitio también se publicaron los manuales de instalación y utilización de la herramienta. La figura 22 muestra una captura de pantalla del sitio, donde se pueden observar los enlaces directos a las páginas de tutoriales y descargas. En el anexo 10 se incluyen más capturas de pantalla del sitio web.

Figura 22 . Sitio Web de TEAM-BE



Fuente: Creación propia

También se realizó el proceso de registro de esta herramienta software desarrollada en la Dirección Nacional de Derechos de Autor (DNDA). (Para visualizar el certificado de registro Ver Anexo 9)

Este trabajo fue presentado en la modalidad de poster en el Cuarto Congreso Andino en Computación, informática y Educación - CACIED 2019 (ver anexo 7) donde se obtuvo retroalimentación importante por parte de los asistentes y jurados, especialmente en cuanto a la metodología y la actividad de aplicación de la herramienta en un entorno real. También se presentó el proyecto en el evento virtual Expoposter 2020 bajo la modalidad de poster (ver anexo 8).

4.1 DISCUSIÓN DE RESULTADOS

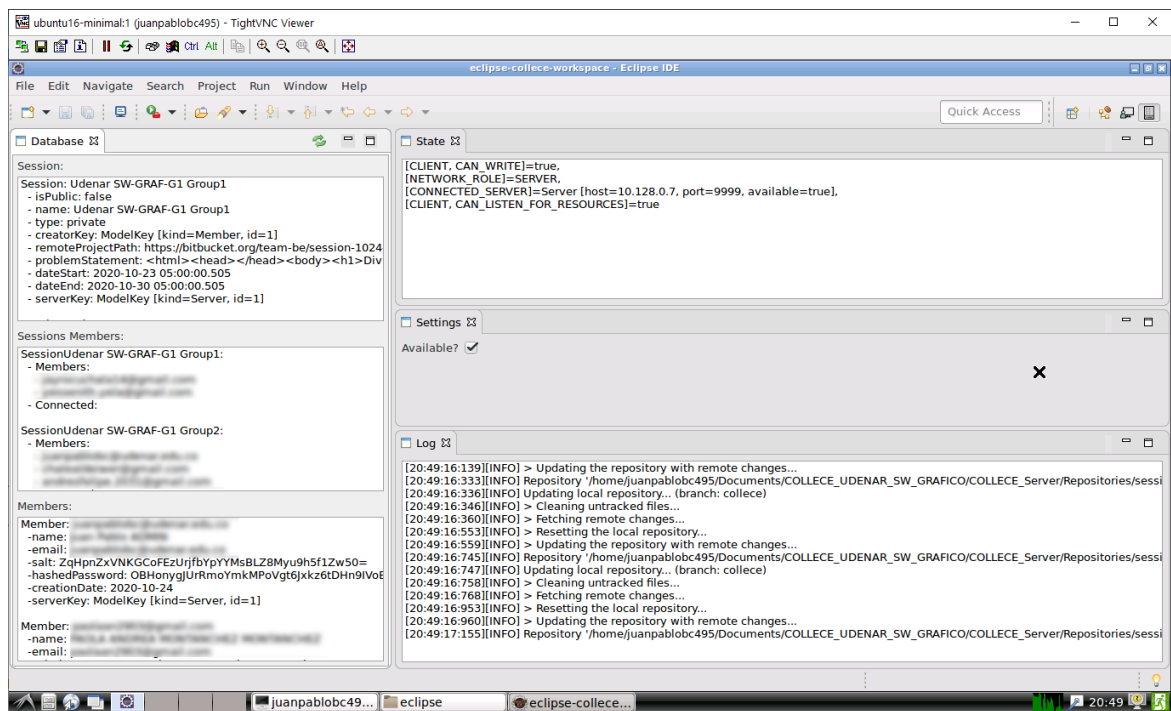
Inicialmente la herramienta TEAM-BE fue probada en la asignatura Seminario de Programación e Informática 2 con estudiantes de Ingeniería de Sistemas en la Universidad de Nariño sede Ipiales pero debido a diversos problemas técnicos relacionados con la instalación de la herramienta, la calidad de la conexión a internet de algunos participantes y un parámetro configurado de manera incorrecta en el software Collece 2.0 la actividad no pudo ser completada satisfactoriamente, estos inconvenientes fueron tenidos en cuenta para realizar el experimento en una segunda oportunidad con estudiantes de Software Gráfico en la sede Pasto donde pudo realizarse de forma satisfactoria, aquí la herramienta TEAM-BE mostró eficiencia y eficacia en el proceso de formación automática de grupos dentro de Collece 2.0 con un total de 30 participantes con los que se formaron grupos de 3 personas.

4.2 VALIDACIÓN DE LA HERRAMIENTA

4.2.1 Detalles técnicos. Para asegurar un buen rendimiento en la instancia de Servidor de Collece 2.0 y garantizar el acceso ininterrumpido por parte de los participantes se optó por ejecutar el servidor en una instancia de máquina virtual de la plataforma de servicios en la nube de Google (*Google Cloud Platform*), en particular, se utilizó una instancia de propósito general serie E2 (4 núcleos, 16GB de memoria RAM y 10GB de disco duro) el sistema operativo utilizado fue GNU/Linux - Ubuntu 16. La configuración del servidor consistió en instalar un entorno gráfico sencillo (LXDE) el kit de desarrollo de java (JDK 8) y un servidor VNC para acceder a la interfaz gráfica remotamente y finalmente instalar Eclipse IDE y dentro de él instalar Collece 2.0.

En la figura 23 se muestra el servidor funcionando visualizado con el programa TightVNC Viewer.

Figura 23. Servidor remoto Collece 2.0



Fuente: Creación propia

4.2.2 Procedimiento. A continuación, se describe el proceso seguido para validar el funcionamiento de la herramienta desarrollada en un entorno académico real, esta validación se realizó con estudiantes de Ingeniería de Sistemas de la Universidad de Nariño. Debido a que el semestre en que se llevó a cabo la actividad tuvo la modalidad virtual la actividad también se realizó de esa manera.

Para obtener datos cuantificables de los participantes se utilizó el test psicológico de BigFive que mediante un cuestionario de preguntas sencillas permite medir cinco dimensiones de la personalidad del individuo (extraversión, cordialidad, responsabilidad, estabilidad emocional y apertura a la experiencia). Los participantes diligenciaron el cuestionario mediante la plataforma Moodle, una vez todos los participantes realizaron este proceso se descargó un archivo CSV generado por la plataforma que puede ser usado directamente en TEAM-BE.

Inicialmente se realizó la actividad con estudiantes de la sede Ipiales, pero se presentaron algunos inconvenientes que impidieron la finalización satisfactoria de la actividad, para empezar a pesar de que se elaboró un manual para la instalación de Eclipse IDE y Collece 2.0 varios estudiantes reportaron dificultades para instalar las herramientas, también se presentaron problemas debidos a la configuración interna de Collece, específicamente en un parámetro que controlaba la cantidad de usuarios concurrentes conectados al servidor. A partir de esa experiencia se decidió crear un paquete zip portable con el software preinstalado y crear un sitio web para facilitar su descarga, además se corrigió el problema que dificultaba la conexión de los usuarios y se realizaron múltiples pruebas adicionales para detectar y solucionar problemas de ese estilo, gracias a esto en la segunda oportunidad no se presentaron problemas dificultades y la actividad se desarrolló satisfactoriamente.

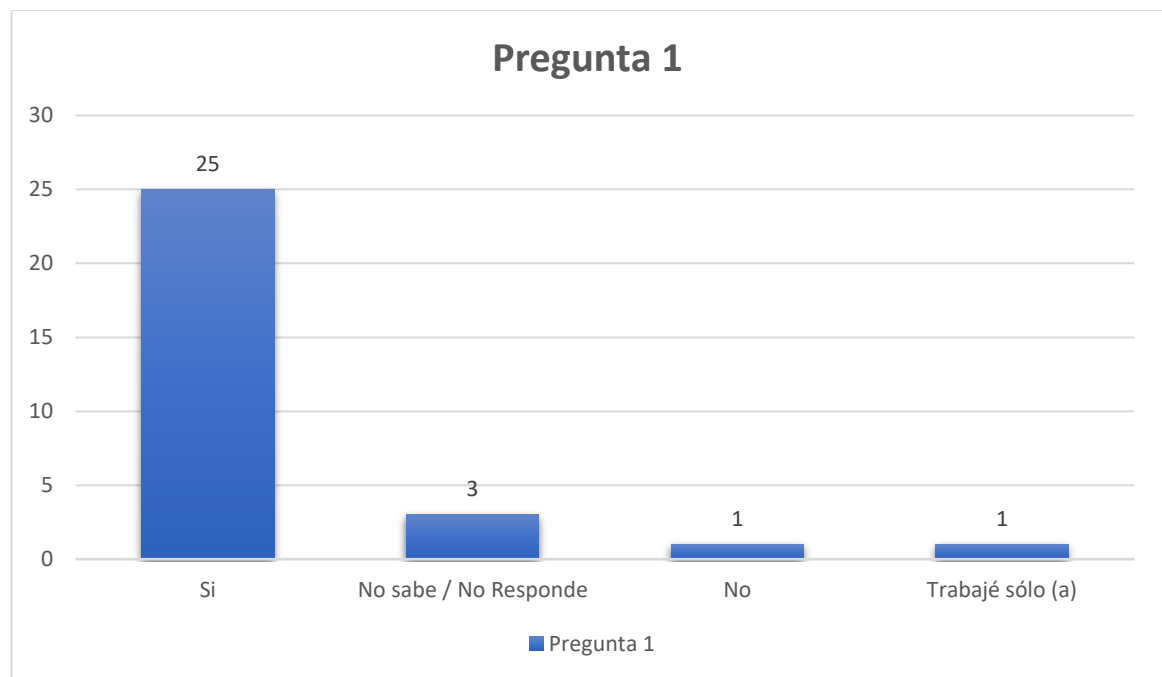
La actividad consistió en solucionar un problema sencillo de programación competitiva (ver anexo 5), el tamaño de los grupos fue de 3 y el tiempo límite fue de dos horas, la mayoría de los grupos solucionaron el problema antes de finalizar ese tiempo. Por otro lado, debido a que la actividad se realizó de forma remota con los participantes accediendo desde sus hogares algunos no pudieron conectarse por lo que 2 de los 8 grupos trabajaron con sólo uno o dos miembros.

Al finalizar la actividad los participantes contestaron una breve encuesta pensada para medir su percepción frente a la herramienta y la formación automática de grupos (para visualizar la encuesta realizada ver Anexo 4).

4.2.3 Resultados encuesta posterior. A continuación, se muestra el análisis de las respuestas dadas por los participantes de la actividad de aplicación de la herramienta TEAM-BE, la encuesta se realizó inmediatamente después de culminada la actividad. Como se mencionó anteriormente por problemas técnicos la actividad pudo ser desarrollada en su totalidad con el grupo de la sede Ipiales, por lo tanto, esas respuestas no se incluyen en este análisis. Se realizaron cinco preguntas abiertas relacionadas a la satisfacción en cuanto a los compañeros de trabajo asignados y el uso del entorno Collece. Todas fueron preguntas abiertas para tener una retroalimentación más amplia por parte de los participantes, para el análisis se realizó un agrupamiento de las respuestas para poder obtener estadísticas de ellas.

Pregunta 1: ¿Se encontró cómodo (a) trabajando con los compañeros (as) asignados (as) de forma automática? La figura 24 muestra la gráfica que consolida las respuestas a esta pregunta dadas por los participantes.

Figura 24. Análisis de encuesta - Gráfica de pregunta 1

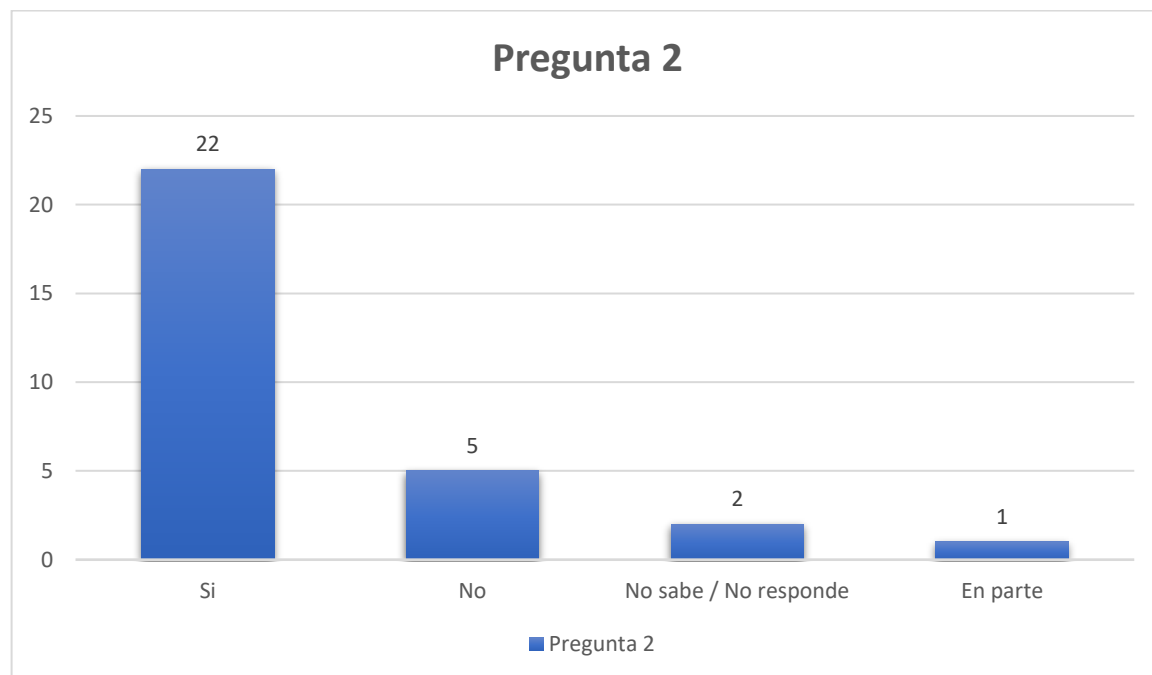


Fuente: Creación propia

Como se puede observar la mayoría de participantes (83.3%) se encontraron cómodos (as) trabajando con los compañeros asignados de forma automática. Algunos de ellos se conocían previamente mientras que en otros casos trabajaron por primera vez; esto no representó un problema en ninguno de los dos casos.

Pregunta 2: ¿Sintió complementariedad al trabajar con los compañeros asignados automáticamente? La figura 25 muestra la gráfica que consolida las respuestas a esta pregunta dadas por los participantes.

Figura 25. Análisis de encuesta - Gráfica de pregunta 2

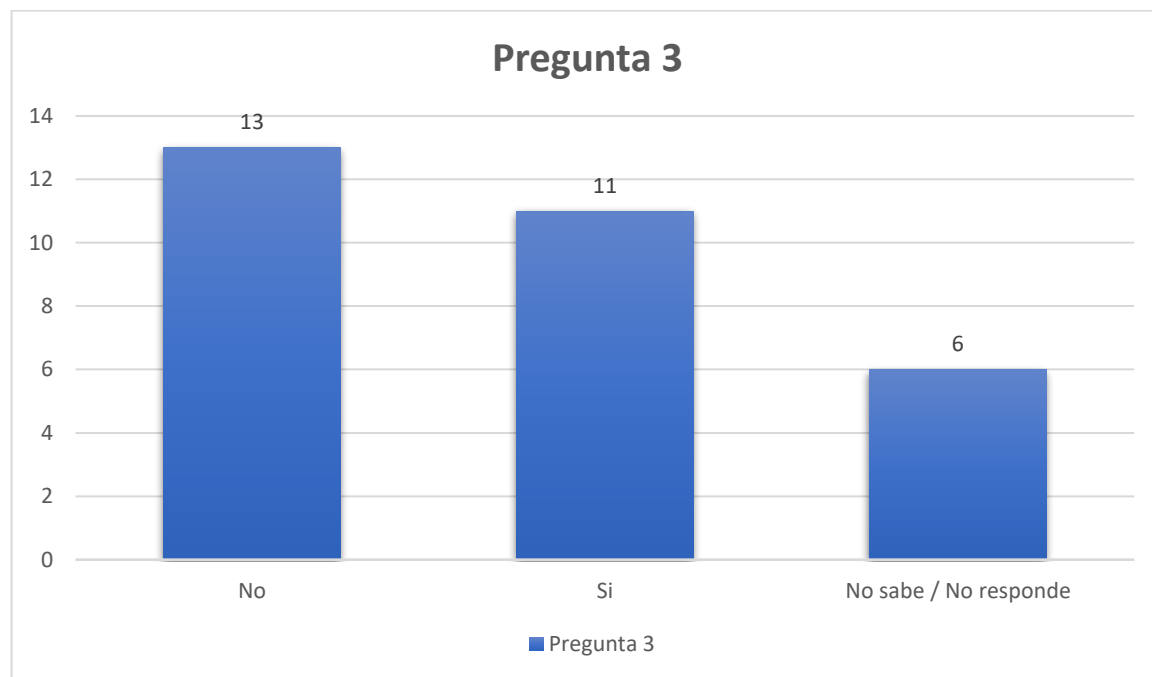


Fuente: Creación propia

El 73.3% de los participantes manifestaron sentir que sus habilidades y conocimientos se complementaban con los compañeros asignados de forma automática. Así mismo la mayoría manifestó que lograron organizarse rápidamente para encontrar la solución al problema, aunque no hayan trabajado previamente en otras actividades. Además, uno de ellos manifestó que esa configuración de grupo en particular permitió que las falencias o faltas de conocimiento sean compensadas por otros miembros del grupo.

Pregunta 3: *¿Considera que su equipo logró realizar la actividad de una mejor manera en comparación a si hubiera escogido sus compañeros (as) libremente?* La figura 26 muestra la gráfica que consolida las respuestas a esta pregunta dadas por los participantes.

Figura 26. Análisis de encuesta - Gráfica de pregunta 3



Fuente: Creación propia

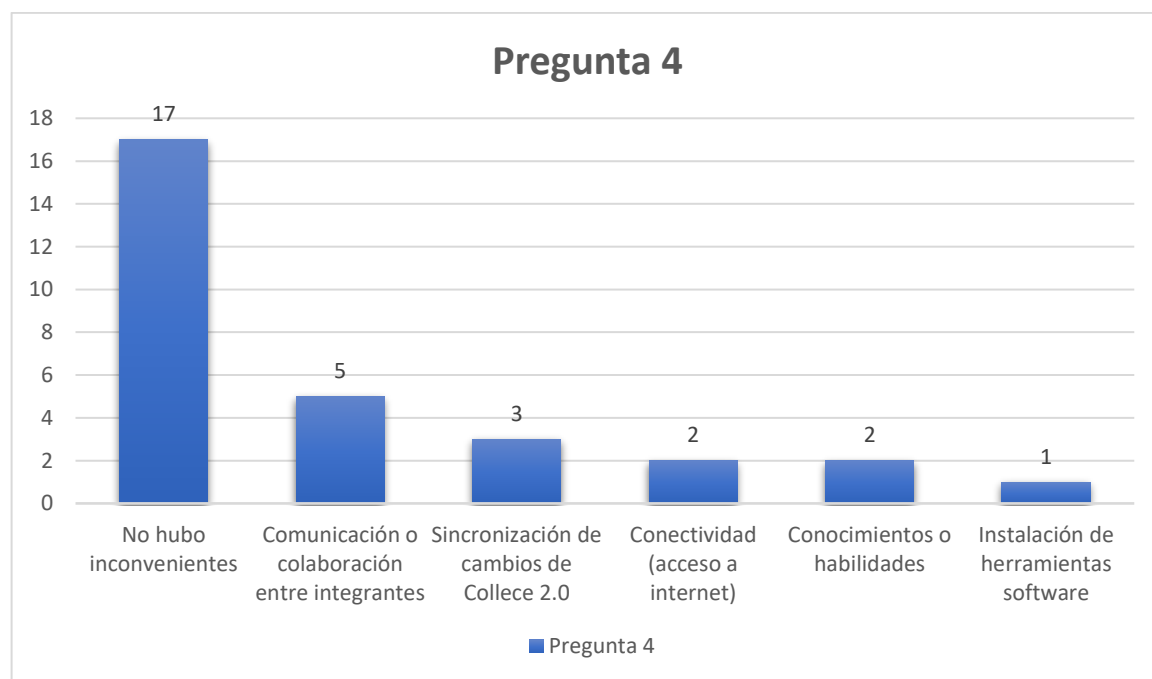
A pesar de que en otras partes de la encuesta la mayoría de participantes manifestaron haber sentido complementariedad al momento de trabajar con sus compañeros, cerca de la mitad de los participantes indicaron que hubieran realizado la actividad de mejor forma con compañeros escogidos libremente, las razones dadas fueron principalmente la falta de confianza con personas desconocidas, este inconveniente se vio amplificado debido a que por razones de fuerza mayor al momento de realizar la actividad (clases virtuales a consecuencia de la pandemia COVID-19) la comunicación entre los miembros de cada grupo se dio únicamente por medio del chat integrado en Colcece o alguna herramienta para llamadas de voz o video en internet, esto fue una barrera importante para el ambiente de trabajo, especialmente para participantes que no se conocían previamente.

Por otro lado, en cuanto a las personas que contestaron sí, algunos destacaron como ventaja el hecho de no conocerse previamente, pues de esto evitaba distracciones y facilitaba que los miembros del grupo aporten ideas diversas para la consecución del objetivo.

Pregunta 4: ¿Hubo inconvenientes en la realización del ejercicio? ¿Cuáles?

La figura 27 muestra la gráfica que consolida las respuestas a esta pregunta dadas por los participantes.

Figura 27. Análisis de encuesta - Gráfica de pregunta 4



Fuente: Creación propia

El 56.7% de los participantes no presentó ningún problema en el desarrollo de la actividad, el resto manifestó haber tenido al menos uno de los problemas categorizados a continuación:

- **Comunicación o colaboración entre integrantes (16.6%):** problemas externos a las herramientas software, tales como pobre comunicación entre los miembros, baja participación o ausencia de algunos provocando una distribución del trabajo inequitativa
- **Sincronización de cambios de Collece 2.0 (10%):** en algunas ocasiones (3) se presentó un bug en Collece que provocaba que la sincronización del código fuente no funcionara correctamente, esto causaba que los cambios algunos participantes no se reflejaran en las instancias cliente de sus compañeros o se borrarán líneas de código previamente escritas. Este incidente fue informado detalladamente al equipo de desarrollo de Collece 2.0 para su eventual corrección

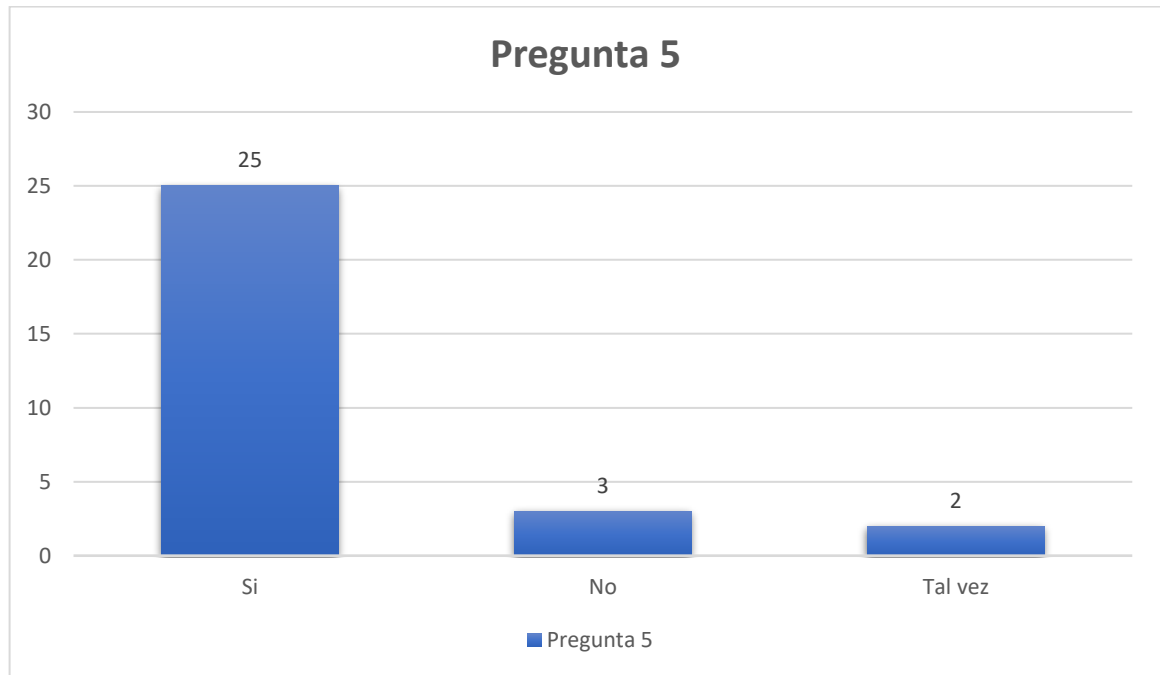
- **Conectividad (acceso a internet) (6.6%):** problemas técnicos relacionados a la velocidad, estabilidad y calidad de la conexión a internet de los participantes
- **Conocimientos o habilidades (6.6%):** los participantes en esta categoría expresaron dificultades en la comprensión del problema planteado o en la escritura del código por falta de conocimiento en el lenguaje Java
- **Instalación de herramientas software (3.3%):** dificultades en la instalación de las herramientas software necesarias para la actividad

Como se puede observar, algunos problemas surgieron (o fueron agravados) debido a que la actividad se realizó en modalidad virtual, si bien el entorno Collece puede usarse fácilmente de esta manera, existen factores difíciles de controlar como la calidad de la conexión a internet de los participantes o la facilidad de comunicación de los mismos.

En cuanto al fallo en la sincronización de cambios de Collece 2.0, si bien sucedió sólo en 3 casos en tiempos específicos y pudo haberse presentado por fallos en la red de internet de los participantes, se vio necesario informar sobre el problema a su equipo de desarrollo adjuntando el archivo log correspondiente para su revisión, pues podría afectar la realización de futuras actividades y entorpecer la implementación de otras funcionalidades.

Pregunta 5: ¿Utilizaría de nuevo la herramienta Collece con el complemento TEAM-BE? La figura 28 muestra la gráfica que consolida las respuestas a esta pregunta dadas por los participantes.

Figura 28. Análisis de encuesta - Gráfica de pregunta 5



Fuente: Creación propia

La gran mayoría de los participantes (83.3%) manifestaron que volverían a usar el conjunto de herramientas software (Eclipse, Collece 2.0, TEAM-BE) en otras actividades. En cuanto a los que respondieron no (10%) la principal razón expresada fue la preferencia por trabajar solos.

5. CONCLUSIONES

- En este proyecto se desarrolló el módulo TEAM-BE para el *plugin* de Eclipse Collece 2.0, el cuál integra correctamente la funcionalidad de formación de grupos mediante la implementación de la librería TEAM-B (desarrollada en la Universidad de Nariño). El módulo automatiza varias tareas necesarias en el proceso de creación de sesiones para cada grupo. Lo anterior, además de su interfaz intuitiva que facilita su uso hace que Collece 2.0 sea atractivo para su uso en actividades de enseñanza de la programación.
- Tras realizar un proceso de búsqueda sobre trabajos relacionados con este proyecto se encontraron algunos sobre construcción de *plugins* para el IDE Eclipse y otras plataformas como Moodle o Google Wallet, pero estos apuntan a objetivos diversos (tales como asistir la realización de laboratorios remotos o verificar la sintaxis de código existente) no relacionados directamente con los de este trabajo. Por tanto, hasta donde se conoce, este es el primer trabajo en su tipo, al menos a nivel regional, por este motivo se requirió de una profundización mayor en el tema. Afortunadamente se contó con soporte directo por parte del equipo de desarrollo de Collece 2.0.
- La estructura y código modular así como la documentación técnica del *plugin* Collece 2.0 para Eclipse permitió agregar la funcionalidad de formación automática de grupos aprovechando el código existente para satisfacer distintas necesidades como la creación de vistas y el soporte para varios idiomas. Además, el módulo agregado no afecta el funcionamiento ni la utilización de los ya existentes, dándole al usuario la opción de usar o no el módulo de formación automática de grupos TEAM-BE.
- La librería Java TEAM-B forma grupos homogéneos, heterogéneos y mixtos atendiendo a diferentes características cuantificables en tiempos de cómputo razonables. Para este caso se utilizaron como datos de entrada los rasgos de personalidad obtenidos con la prueba psicométrica “Big Five Inventory – BFI”, pero es posible utilizar otros instrumentos o test que permitan cuantificar diferentes características de los participantes. Para el experimento se utilizó un computador personal convencional, de modo que la librería puede utilizarse sin mayores inconvenientes sin necesidad de equipos especializados o capacidades técnicas excepcionales.
- La validación del funcionamiento de la herramienta en un entorno educativo real arrojó buenos resultados en cuanto a su funcionamiento y rendimiento.

Las distintas instancias de Collece 2.0 tanto cliente como servidor se ejecutaron en distintos sistemas operativos sin presentar problemas de compatibilidad. Además, la mayoría de los participantes manifestaron sentirse a gusto trabajando con los compañeros asignados y mostraron interés y curiosidad frente al trabajo colaborativo y la formación automática de grupos.

- Aunque la herramienta Collece 2.0 funciona correctamente tanto en redes locales como a través de internet es recomendable que las actividades académicas que involucren su uso se realicen de forma presencial. Al realizar la actividad de aplicación de forma remota surgieron algunos problemas técnicos relacionados con la calidad de la conexión a internet de los participantes y las especificaciones técnicas de sus equipos de cómputo. Esto sumado a otros factores particulares a cada estudiante como el individualismo o la comunicación poco asertiva entorpecieron el desarrollo de las actividades y dificultaron aprovechar el potencial de la formación automática de grupos. Por lo anterior se hace necesario acompañar estas actividades con metodologías pedagógicas que busquen subsanar esos problemas.
- La creación de *plugins* en Eclipse está apoyada por una amplia documentación en libros y artículos, así como por recursos digitales como blogs y foros apoyados por la comunidad de desarrolladores. Asimismo, existen diferentes *Frameworks* que suplen diversas necesidades según el *plugin* o módulo a desarrollar (como SWT para las interfaces o ECF para las comunicaciones). Esto gracias a la amplia utilización de Eclipse a lo largo de los años y su constante mantenimiento e implementación de nuevas tecnologías por parte de la Fundación Eclipse. Además, al ejecutarse sobre el lenguaje de programación Java existen recursos y comunidades externas de mucha utilidad a la hora de resolver dudas o problemas específicos.
- La utilización de la metodología ágil SCRUM facilitó la planeación y ejecución del proyecto, así como documentar el proceso de desarrollo de software mediante roles, eventos y artefactos. Lo anterior debido a que su flexibilidad permite implementar sus prácticas a distintos escenarios, en este caso un equipo de desarrollo pequeño con participantes distantes geográficamente.
- UML fue un lenguaje de modelado útil y adecuado para la descripción y especificación de los distintos aspectos técnicos de este proyecto. Este lenguaje provee los diagramas necesarios para modelar los distintos

componentes, escenarios y flujos de datos del software con un nivel adecuado de detalle.

6. RECOMENDACIONES

- Evaluar con mayor amplitud el funcionamiento y pertinencia del conjunto de herramientas software Eclipse IDE – Collece 2.0 – TEAM-BE en diferentes ambientes académicos, preferiblemente en cursos presenciales complementando las actividades con metodologías que favorezcan la colaboración entre los participantes.
- Realizar experimentos de aplicación de Collece 2.0 en entornos reales probando otro tipo de instrumentos para cuantificar las características de los participantes y usarlas como datos de entrada para TEAM-BE, por ejemplo, con test de inteligencia, de habilidades comunicativas o de capacidades de liderazgo.
- Agregar opciones de configuración a la interfaz de instancia servidor de Collece 2.0 para poder modificar algunos parámetros como el número máximo de usuarios concurrentes o las credenciales de la cuenta de la plataforma (Bitbucket) donde se crean los repositorios GIT necesarios para cada sesión, pues de momento todas esas opciones se encuentran en un archivo de texto que no es fácilmente editable una vez se ha compilado el *plugin*.
- Implementar la compatibilidad con otras plataformas de alojamiento de repositorios GIT tales como Gitlab o Github, pues actualmente sólo se realiza con Bitbucket, asimismo se recomienda permitir el uso de repositorios privados o en red local ya que de momento Collece 2.0 sólo funciona si estos son públicos en internet. Lo anterior puede representar un obstáculo para que los estudiantes utilicen la herramienta por cuenta propia pues limita el control de acceso que tendrán sobre el código que escriban.
- Permitir la personalización de las cuentas de usuario en Collece 2.0, puesto que de momento para un usuario registrado no es posible modificar ningún campo como el nombre, contraseña o correo electrónico, lo cual hace que la herramienta sea menos atractiva al no otorgar control sobre estos datos.
- Permitir el acceso a la herramienta Collece 2.0 de forma libre a la comunidad académica en general, pues actualmente es necesario solicitarlo al grupo de investigación CHICO (Computer Human Interaction and Collaboration) de la Universidad de Castilla la Mancha en España. Esto sumado a que gran parte

de documentación de usuario existente no es pública hace que la herramienta sea poco conocida y, por tanto, poco utilizada.

- Dar a conocer la herramienta TEAM-BE y su uso integrado con Collece 2.0 dentro de la Universidad de Nariño para que esta sea usada en la enseñanza de la programación.

BIBLIOGRAFÍA

- [1] MALDONADO PÉREZ, Marisabel. El Trabajo Colaborativo En El Aula Universitaria. En Laurus - Revista de Educación. Vol. 13 (2007); p. 263-278.
- [2] PORTAFOLIO. Programación, la carrera que más demandará el mercado laboral en 2021. En Portafolio. (2021); Disponible en <https://www.portafolio.co/tendencias/programacion-la-carrera-que-mas-demandara-el-mercado-laboral-en-el-2021-548289>
- [3] ZAPATA-ROS, Miguel. Pensamiento computacional: Una nueva alfabetización digital. En Revista Educación a Distancia. (2015); p. 2-12. Disponible en <https://revistas.um.es/red/article/view/240321/183001>
- [4] RUÍZ AGUIRRE, Edith Inés, MARTÍNEZ DE LA CRUZ, Nadia Livier, y GALINDO GONZÁLEZ, Rosa María. El aprendizaje colaborativo en ambientes virtuales. 1 a ed. Guadalajara: Editorial Cenid, 2015.
- [5] GUITERT, Montse y GIMÉNEZ, Ferran Giménez. Trabajo cooperativo en entornos virtuales de aprendizaje. Aprender en la virtualidad. 1 a ed. Barcelona: Editorial Gedisa, 2000. p. 13
- [6] SÁNCHEZ, Santiago, REDONDO, Miguel A., VALLEJO, David, GONZÁLEZ, Carlos, y BRAVO, Crescencio. COLLECE 2.0: A DISTRIBUTED REAL-TIME COLLABORATIVE PROGRAMMING ENVIRONMENT FOR THE ECLIPSE PLATFORM. En International Association for Development of the Information Society (AIDIS). (2017); p. 136-142. Disponible en <http://www.iadisportal.org/digital-library/collece-20-a-distributed-real-time-collaborative-programming-environment-for-the-eclipse-platform>
- [7] GARCÍA RANEA, Raúl. Desarrollo de un plug-in de Eclipse para validar los nombres de los elementos de un esquema conceptual. Barcelona, 2011, 438p. Tesis de maestría (Ingeniería Informática). Universitat Politècnica de Catalunya.
- [8] PÉREZ COSTA, Ernesto. Plugin en Eclipse para automatizar el trabajo con el microkernel Fiasco OC. Santa Clara. Cuba, 2016, 77p. Trabajo de grado (Licenciatura en Ciencias de la Computación). Universidad Central "Marta Abreu" de Las Villas.
- [9] PESSOA, Tiago, BRITO E ABREU, Fernando, PESSOA MONTEIRO, Miguel, y BRYTON, Sérgio. An Eclipse Plugin to Support Code Smells Detection. En Research Gate. (Abr.2012); p. 2-12. Disponible en https://www.researchgate.net/publication/224871887_An_Eclipse_Plugin_to_Support_Code_Smells_Detection
- [10] SILVA, André, LEAL, José Paulo, y PAIVA, José Carlos. "Raccocode: An eclipse plugin for assessment of programming exercises". [en línea]. 2018. {con

acceso el 29 de agosto de 2019}. Disponible en <https://api.semanticscholar.org/CorpusID:51868020>

- [11] PAZMIÑO PINTO, Santiago David. Desarrollo e implementación de un plugin de Google Wallet para pagos Online utilizando Software Open Souce. Sangolquí, 2013, 97p. Trabajo de grado (Ingeniería en Sistemas e Informática). Universidad de las Fuerzas Armadas ESPE.
- [12] JIMÉNEZ BARRETO, Camilo Andres y OLARTE GARZÓN, Guillermo. Construcción de un plugin para el eclipse IDE que le permita al estudiante adquirir las principales buenas prácticas de programación orientada a objetos en el lenguaje Java. Bogotá, 2015, 123p. Trabajo de grado (Ingeniería de Sistemas). Universidad Piloto de Colombia. Facultad de Ingeniería.
- [13] CORTES ROMERO, Henry. DISEÑO Y DESARROLLO DE UN PLUGIN PARA LA PLATAFORMA “MOODLE” QUE PERMITE LA REALIZACIÓN DE LABORATORIOS REMOTOS. Bogotá, 2016, 78p. Trabajo de grado (Licenciatura en Electrónica). Universidad Pedagógica Nacional. Facultad de Ciencia y Tecnología.
- [14] CORDOBA, Franco Esteban. CW-TEAMS: SOFTWARE PARA LA CONFORMACIÓN DE GRUPOS DE TRABAJO COLABORATIVO BASADO EN ALGORITMOS GENÉTICOS. Pasto, 2019, 222p. Trabajo de grado (Ingeniería de Sistemas). Universidad de Nariño. Facultad de Ingeniería.
- [15] GUILLERT, Montse y JIMÉNEZ, Ferran. Aprender a colaborar. Cooperar en clase : ideas e instrumentos para trabajar en el aula. Sevilla: MCEP, 1997. p. 196
- [16] ECHAZARRETA, Carmen, PRADOS, Ferran, POCH, Jordi, y SOLER, Josep. La competencia «El trabajo colaborativo»: Una oportunidad para incorporar las TIC en la didáctica universitaria. Descripción de la experiencia con la plataforma ACME (UdG). En UOC Papers - Revista sobre la sociedad del conocimiento. Vol. 8 (2009); p. 1-11.
- [17] JOHNSON, David. W., JOHNSON, Roger T., y JOHNSON, Edythe. Los nuevos círculos de aprendizaje. La cooperación en el aula y la escuela. 1 a ed. Buenos Aires: Editorial Aique, 1999.
- [18] MONTESERIN, Ariel, SCHIAFFINO, Silvia, GARCÍA, Patricio, y AMANDI, Analía. Analisis de la formación de grupos en Aprendizaje Colaborativo Soportado por Computadoras. En Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE). (Oct.2012); p. 2-4. Disponible en <https://www.br-ie.org/pub/index.php/sbie/article/view/1465/1230>
- [19] MARÍN SANCHEZ, Manuel y TROYANO RIDRIGUEZ, Yolanda. Psicología social de los procesos grupales. 1 a ed. Madrid: Ediciones Pirámide, 2012.

- [20] COMPUTERWEEKLY. "Write once, run anywhere?". [en línea]. 2018. {con acceso el 31 de agosto de 2019}. Disponible en <https://www.computerweekly.com/feature/Write-once-run-anywhere>
- [21] ECURED. "Lenguaje interpretado". [en línea]. 2018. {con acceso el 12 de febrero de 2019}. Disponible en https://www.ecured.cu/Lenguaje_interpretado
- [22] ORACLE. "The Java Language Environment". [en línea]. 2018. {con acceso el 31 de agosto de 2020}. Disponible en <https://www.oracle.com/java/technologies/introduction-to-Java.html>
- [23] CODECADEMY. "What Is an IDE?". [en línea]. 2018. {con acceso el 21 de diciembre de 2018}. Disponible en <https://www.codecademy.com/articles/what-is-an-ide>
- [24] ECLIPSE FOUNDATION. "Eclipse Project". [en línea]. 2018. {con acceso el 13 de septiembre de 2019}. Disponible en <https://www.eclipse.org/eclipse/>
- [25] ECLIPSE FOUNDATION. "Eclipse Platform Technical Overview". [en línea]. 2001. {con acceso el 13 de septiembre de 2019}. Disponible en <https://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.html>
- [26] ECLIPSE FOUNDATION. "FAQ Where did Eclipse come from?". [en línea]. 2004. {con acceso el 31 de agosto de 2019}. Disponible en https://wiki.eclipse.org/FAQ_Where_did_Eclipse_come_from%3F
- [27] GRUPO COMPUTER HUMAN INTERACTION AND COLLABORATION (CHICO). "COLLECE-2.0". [en línea]. 2019. {con acceso el 30 de noviembre de 2018}. Disponible en <http://blog.uclm.es/grupochico/proyecto-iapro/collece-2-0/>
- [28] SOFTWARE FREEDOM CONSERVANCY. "Git". [en línea]. 2019. {con acceso el 14 de septiembre de 2020}. Disponible en <https://git-scm.com/>
- [29] SOFTWARE FREEDOM CONSERVANCY. "About - Git". [en línea]. 2019. {con acceso el 14 de septiembre de 2019}. Disponible en <https://git-scm.com/about/branching-and-merging>
- [30] FLORES, Erik. "Paradigma Positivista". [en línea]. 2018. {con acceso el 12 de diciembre de 2019}. Disponible en <https://sites.google.com/site/wikiminvestigacion2018/paradigma-positivista>
- [31] FERNANDEZ COLLADO, Carlos y BAPTISTA LUCIO, Pilar. Metodología de la investigación. 6 a ed. México D.F: McGraw-Hill, 2014.
- [32] CAMPBELL, Donald T. y STANLEY, Julian C. Diseños experimentales y cuasiexperimentales en la investigación social. 1 a ed. Buenos Aires: Rand McNally & Company, 1966.

- [33] PALACIO, Carlos. E-book Nociones básicas de metodologías ágiles. 1 a ed. Bogotá: Academia Pragma, 2020.
- [34] ATlassian AGILE COACH. "What is kanban?". [en línea]. 2019. {con acceso el 17 de noviembre de 2019}. Disponible en <https://www.atlassian.com/agile/kanban>
- [35] AGILE ALLIANCE. "Extreme Programming (Glossary)". [en línea]. 2019. {con acceso el 17 de noviembre de 2019}. Disponible en <https://www.agilealliance.org/glossary/xp>
- [36] PRODUCTPLAN. "Lean Software Development". [en línea]. 2019. {con acceso el 17 de noviembre de 2019}. Disponible en <https://www.productplan.com/glossary/lean-software-development/>
- [37] PRODUCTPLAN. "Crystal Agile Framework". [en línea]. 2019. {con acceso el 17 de diciembre de 2019}. Disponible en <https://www.productplan.com/glossary/crystal-agile-framework/>
- [38] PRODUCTPLAN. "Feature Driven Development (FDD)". [en línea]. 2019. {con acceso el 17 de noviembre de 2019}. Disponible en <https://www.productplan.com/glossary/feature-driven-development/>
- [39] ACADEMIA PRAGMA. Guía funcional de SCRUM. 1 a ed. Bogotá: Academia Pragma, 2020.
- [40] PROJECT MANAGEMENT DOCS. "Agile Product Backlog". [en línea]. 2020. {con acceso el 5 de abril de 2020}. Disponible en <https://www.projectmanagementdocs.com/template/agile-templates/agile-product-backlog/>
- [41] SCRUMMANAGER y NAVEGÁPOLIS. "Formato ejemplo para documentar el uso de scrum en un proyecto". [en línea]. 2012. {con acceso el 30 de agosto de 2020}. Disponible en <https://www.safecreative.org/work/1206241856508-formato-ejemplo-para-documentar-el-uso-de-scrum-en-un-proyecto>
- [42] VILLAMIZAR SUAZA, Katerine. Definición de equivalencias entre historias de usuario y especificaciones en UN-LENCEP para el desarrollo ágil de software. 2013, 34-39p. Universidad Nacional de Colombia.
- [43] VISUAL PARADIGM. "What is Unified Modeling Language (UML)?". [en línea]. 2019. Disponible en <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>
- [44] ARÉVALO, Olga Esperanza, LINARES, Sonia Patricia, CORREA, Luisa Fernanda, OLARTE PARRA, Julieth, y GONZALEZ, Hector Antonio. IEEE-STD-830-1998: PRÁCTICA RECOMENDADA PARA LAS ESPECIFICACIONES DE REQUISITOS DEL SOFTWARE. 2008, 41p. Universidad Nacional de Colombia.

- [45] REVELO SÁNCHEZ, Oscar, INSUASTI PORTILLA, Jesús, y BARÓN SALAZA, Alexander. TEAM-B v2.0 Librería Java™ para la formación homogénea, heterogénea y mixta de grupos, basada en múltiples características. Pasto, 2019, Universidad de Nariño. Facultad de Ingeniería.
- [46] UVA ONLINE JUDGE. "11498 - Division of Nlogonia". [en línea]. 2018. {con acceso el 2 de junio de 2020}. Disponible en https://onlinejudge.org/index.php?option=onlinejudge&Itemid=8&page=show_problem&problem=2493

ANEXOS

Anexo 1. Formato de una historia de usuario en Jira Software

Crear los usuarios y las sesiones en Collece de forma automática



Description

Como Administrador

Quiero crear los usuarios y las sesiones en Collece de forma automática

Para que los participantes puedan acceder al sistema

Activity

Show: **Comments** ▾

Newest first ⌵



Add a comment...

To Do ▾

Details

Assignee



Juan Pablo BC

Labels

None

Sprint

[TB Sprint 3](#)

Story point estimate

3

Reporter



Juan Pablo BC

Fuente: Creación propia

Anexo 2. Formato de un *Sprint* en Jira Software

Edit sprint: TB Sprint 1

Sprint name *

TB Sprint 1

Duration

custom

Start date

1/8/2020 12:00 AM

End date

3/8/2020 12:00 AM

Sprint goal

Lanzar la interfaz gráfica principal de TEAM-BE desde el plugin Collece 2.0 y cargar los datos de los individuos que formaran los grupos.

Update

Cancel

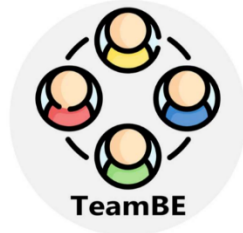
Fuente: Creación propia

Anexo 3. Plantilla para reuniones de retrospectiva

Información del proyecto		
Empresa u organización		
Nombre del proyecto		
Datos de la reunión		
Fecha y hora		
Número de <i>Sprint</i>		
Asistentes		
Tareas del <i>Sprint</i>		
Tarea	Estado	Responsable
Retrospectiva		
Aciertos	Errores	Por mejorar

Fuente: Creación propia

Anexo 4. Encuesta posterior a actividad de aplicación herramienta TEAM-BE



Herramienta de formación automática de grupos para Eclipse / Collece 2.0

Encuesta posterior a actividad TeamBE / Collece

Collece 2.0 es el software añadido a Eclipse como un plugin para agregar las funcionalidades para trabajo colaborativo como edición de código en tiempo real, chat y bloqueo de regiones de código.

TeamBE es una herramienta integrada directamente en Collece que se encarga de formar automáticamente los grupos, crear los usuarios y las sesiones necesarias.

Las siguientes preguntas ayudaran a medir la percepción de estas herramientas y aplicar posibles mejoras para versiones futuras.

Si usted contestó la encuesta previa a la realización de la actividad por favor utilice el mismo correo que utilizó para contestar esa encuesta.

***Obligatorio**

Dirección de correo electrónico *

Tu dirección de correo electrónico

Por favor ingrese su nombre *

Tu respuesta



¿Hubo inconvenientes en la realización del ejercicio? ¿Cuáles? *

Tu respuesta

¿Utilizaría de nuevo la herramienta Collece 2.0 con el complemento TeamBE? *

☐ Si

☐ No

☐ Otro: _____

Página 1 de 1

Enviar

Nunca envíes contraseñas a través de Formularios de Google.

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios

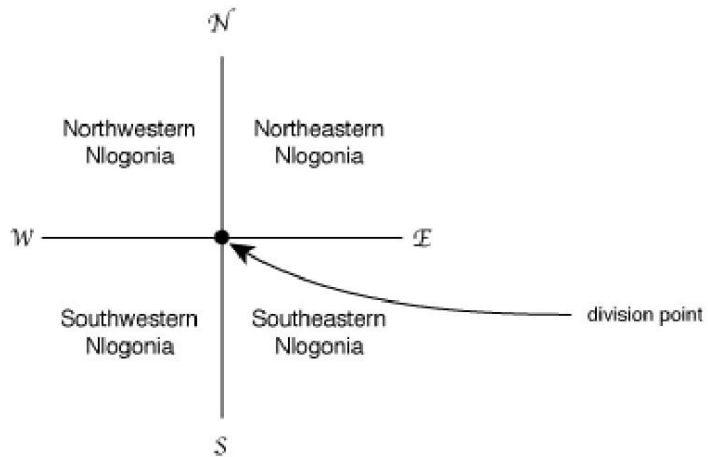


Fuente: Creación propia

Anexo 5. Ejercicio de programación competitiva “*Division of NLogonia*”

After centuries of hostilities and skirmishes between the four nations living in the land generally known as Nlogonia, and years of negotiations involving diplomats, politicians and the armed forces of all interested parties, with mediation by UN, NATO, G7 and SBC, it was at last agreed by all the way to end the dispute, dividing the land into four independent territories.

It was agreed that one point, called *division point*, with coordinates established in the negotiations, would define the country division, in the following way. Two lines, both containing the division point, one in the North-South direction and one in the East-West direction, would be drawn on the map, dividing the land into four new countries. Starting from the Western-most, Northern-most quadrant, in clockwise direction, the new countries will be called Northwestern Nlogonia, Northeastern Nlogonia, Southeastern Nlogonia and Southwestern Nlogonia.



The UN determined that a page in the Internet should exist so that the inhabitants could check in which of the countries their homes are. You have been hired to help implementing the system.

Input

The input contains several test cases. The first line of a test case contains one integer K indicating the number of queries that will be made ($0 < K \leq 10^3$). The second line of a test case contains two integers N and M representing the coordinates of the division point ($-10^4 < N, M < 10^4$). Each of the K following lines contains two integers X and Y representing the coordinates of a residence ($-10^4 \leq X, Y \leq 10^4$).

The end of input is indicated by a line containing only the number zero.

Output

For each test case in the input your program must print one line containing:

- the word 'divisa' (means border in Portuguese) if the residence is on one of the border lines (North-South or East-West);
- 'NO' (means NW in Portuguese) if the residence is in Northwestern Nlogonia;
- 'NE' if the residence is in Northeastern Nlogonia;
- 'SE' if the residence is in Southeastern Nlogonia;
- 'SO' (means SW in Portuguese) if the residence is in Southwestern Nlogonia.

Sample Input

```
3
2 1
10 10
-10 1
0 33
4
-1000 -1000
-1000 -1000
0 0
-2000 -10000
-999 -1001
0
```

Sample Output

```
NE
divisa
NO
divisa
NE
SO
SE
```

Fuente: UVA Online Judge⁶²

⁶² UVA ONLINE JUDGE. "11498 - Division of Nlogonia". [en línea]. 2018. Disponible en https://onlinejudge.org/index.php?option=onlinejudge&Itemid=8&page=show_problem&problem=2493

Anexo 6. Manual de usuario herramienta TEAM-BE



Universidad de Nariño – Departamento de Sistemas

TEAM-BE

**Módulo para la formación automática de grupos
colaborativos en el entorno Collece 2.0**

Manual de Usuario

Diciembre de 2020

Autores:

Juan Pablo Botina Carlosama

Oscar Revelo Sánchez, MSc.

Jesús Insuasti Portilla, MSc.



1. ¿QUE HACE TEAM-BE?

La herramienta desarrollada es un módulo para el plugin Collece 2.0 que permite la formación automática de grupos y automatiza el proceso de registro de usuarios y la creación de sesiones por cada grupo en el sistema junto con los repositorios GIT remotos en la plataforma Bitbucket necesarios para cada una. La vista principal de TEAM-BE hace parte del asistente de creación de sesiones.

A continuación, se describe todo el proceso para instalar y configurar el plugin para Eclipse Collece 2.0, el cual incluye el módulo para la formación automática de grupos TEAM-BE. También se detalla el proceso de formación de grupos y creación de sesiones, así como el proceso a seguir para unirse a ellas.

Como complemento a este manual se ha creado un video tutorial disponible en el siguiente enlace: <https://youtu.be/8g9X6sz4E7s>

2. INSTALACIÓN DE LA HERRAMIENTA

Eclipse IDE funciona sobre la máquina virtual de Java, por lo tanto, es necesario tener este lenguaje instalado en el equipo, en particular el Kit de Desarrollo de Java (JDK), la herramienta fue probada y desarrollada con la versión 8, se recomienda la versión 8 disponible en el siguiente link: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

2.1 PAQUETE PORTABLE

Se provee un archivo zip con Eclipse IDE en formato portable con Collece 2.0 y el módulo TEAM-BE previamente instalado, tan sólo es necesario tener instalado el SDK de Java. El paquete está disponible para los sistemas operativos Windows y GNU/Linux desde el link: <https://hosttbe.web.app/downloads>

Una vez descargado debe extraerse el archivo zip en una ubicación conocida y ejecutar el archivo binario de Eclipse.

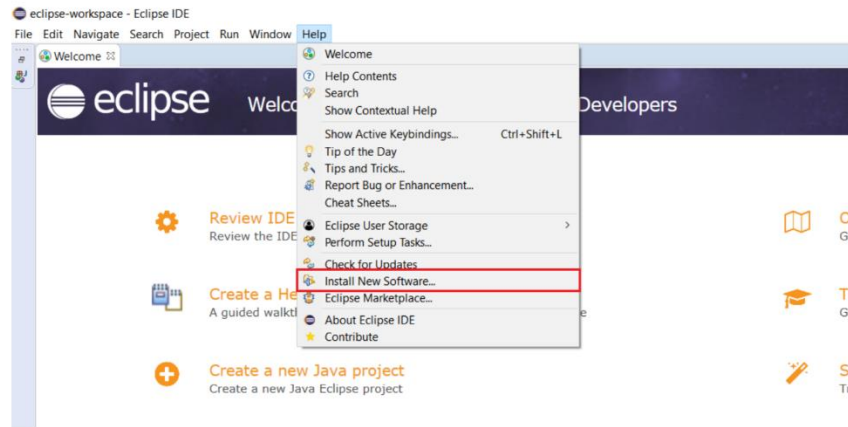
2.2 INSTALACIÓN MANUAL

Collece 2.0 es un plugin para el IDE Eclipse, por lo tanto, es necesario tenerlo instalado previamente (se recomienda la versión 2019-03 R disponible en el siguiente link: <https://www.eclipse.org/downloads/packages/release/2019-03/r>)

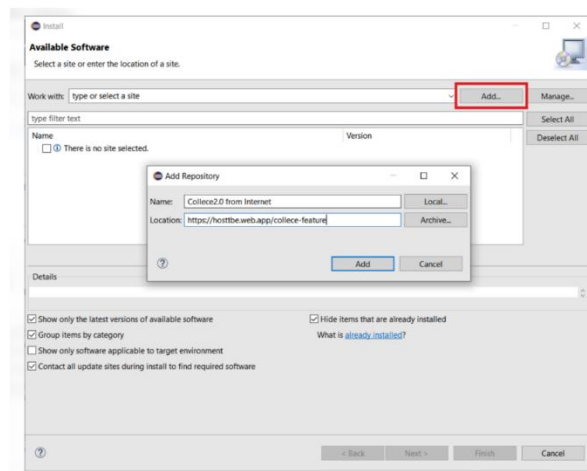
Una vez instalado Eclipse se procede a instalar el plugin Collece 2.0 como se indica a continuación:

Paso 1: Iniciar Eclipse IDE normalmente

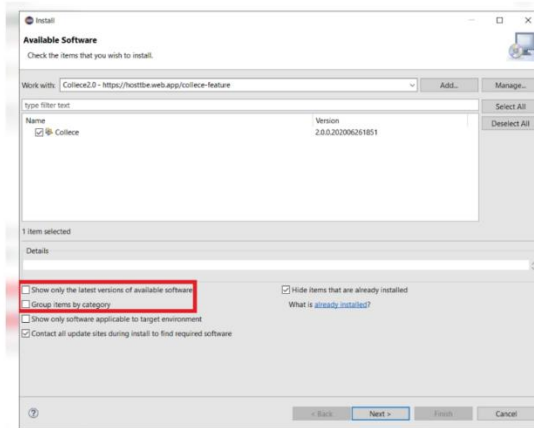
Paso 2: Dentro de Eclipse hacer clic en *Help* → *Add New Software*



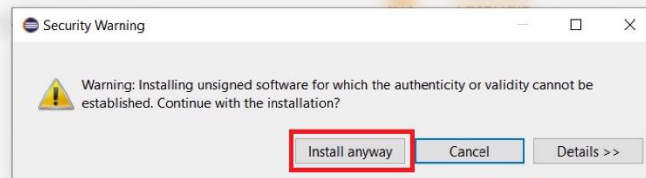
Paso 3: Se despliega la ventana de instalación, dentro de ella hacer clic en el botón *Add*. En el campo *name* ingresar "Collece 2.0". En el campo *location* ingresar <https://hosttbe.web.app/collece-feature>



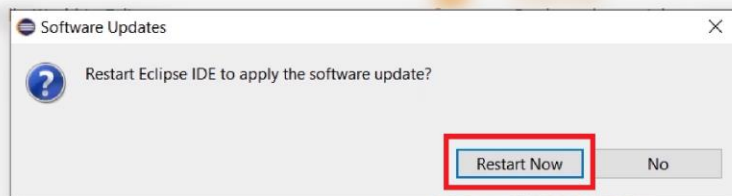
Después presionar el botón *Add*, posteriormente se deben desmarcar las opciones *Group items by category* y *Show only latest versions* y seleccionar *Collece* en la lista como se indica en la imagen:



Paso 4: Hacer clic en *Next* y continuar normalmente la instalación, al finalizar puede mostrarse el siguiente cuadro de diálogo, se debe hacer clic en ***Install anyway***



Paso 5: Cuando termine el proceso de instalación el asistente preguntará si desea reiniciar el IDE para aplicar los cambios, se debe hacer clic en el botón *Restart now*



Cuando el IDE se haya reiniciado la instalación de Collece 2.0 (con el módulo TEAM-BE) habrá terminado.

Se despliega la siguiente ventana donde se configuran los siguientes parámetros para el servidor:

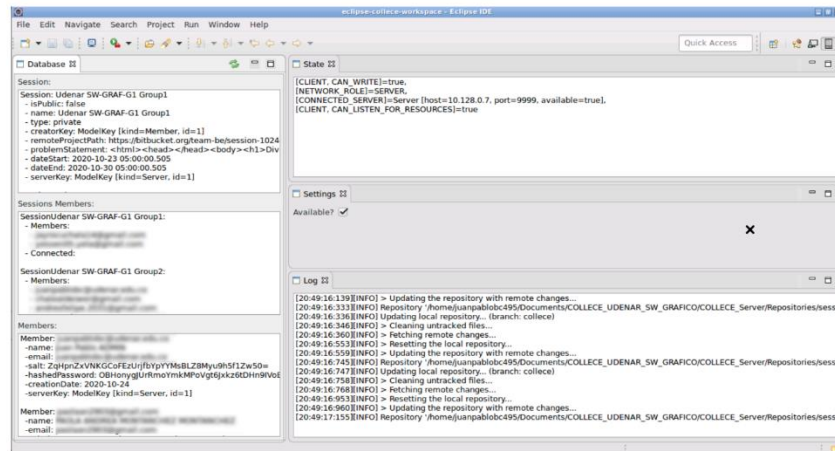
- Puerto: por medio del cual se realizará la conexión desde las instancias cliente
- Directorio del servidor: aquí se guardarán todos los datos relacionados con las sesiones incluyendo el código fuente
- Rama por defecto para git: es la rama (branch) que se utilizará para sincronizar los cambios en el código fuente para cada sesión
- Disponible: si está activado el servidor será accesible en la red y las instancias cliente podrán conectarse a él

Se recomienda dejar estos parámetros en sus valores por defecto y sólo cambiar el directorio del servidor si es necesario.

Una vez se han configurado estos parámetros se debe hacer clic en *Finish* y el servidor estará funcionando y será accesible en la red por medio de la IP del equipo donde se ejecute y el puerto configurado.

En la siguiente imagen se observa la interfaz del servidor ejecutándose, en caso de cerrar por error alguna de las vistas se puede recuperar la perspectiva haciendo clic en:

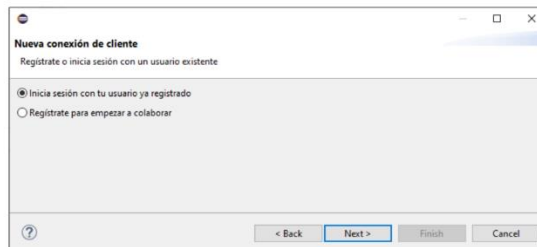
Window → Perspective → Open Perspective → Other → COLLECE-2.0 (server)



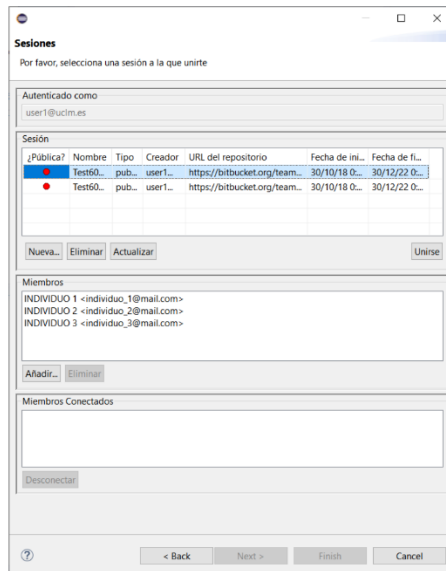
3.2 CREACIÓN DE SESIONES CON TEAM-BE

Para crear múltiples sesiones con base en los grupos formados por TEAM-B se debe seguir el siguiente proceso:

Paso 1: Se debe hacer clic en *File* → *New* → *Other* y en la categoría COLLECE-2.0 seleccionar *Client*, se despliega la ventana mostrada a continuación, aquí el usuario puede registrarse o iniciar sesión en una cuenta creada previamente en un servidor determinado (los servidores se distinguen por su dirección IP).



Paso 2: Una vez el usuario ha iniciado sesión es dirigido al gestor de sesiones de Collece, donde puede visualizar las sesiones alojadas en el servidor, los usuarios que pertenezcan a cada una y cuáles de ellos se encuentran conectados. También puede crear nuevas sesiones y asignarles usuarios manualmente.



Paso 3: Se debe hacer clic en la opción nueva sesión, entonces se despliega una nueva vista la vista donde se ingresan las propiedades de la sesión: nombre, tipo (este campo no se utiliza y se removerá en futuras sesiones, puede estar en blanco), URL del repositorio GIT (al usar TEAM-BE no se habilita pues la herramienta crea los repositorios automáticamente), enunciado del problema (el texto de este campo es compatible con HTML), fechas de inicio y finalización (si la fecha actual está por fuera de ese rango los usuarios no podrán unirse a la sesión). Si la opción "Formar grupos automáticamente" está habilitada al pulsar el botón OK se despliega la vista principal de TEAM-BE.

2. Nombre (texto)
3. Correo electrónico (texto)
4. Característica 1 (número con decimales)
5. ...
6. Característica N (número con decimales)

En este ejemplo se utilizaron 5 características porque es la cantidad que maneja el test de personalidad BigFive (Extraversión, Cordialidad, Responsabilidad, Estabilidad Emocional, Apertura a la experiencia).

```
1 1,INDIVIDUO 1,individuo_1@mail.com,2.0000,4.0000,2.5556,2.5000,3.3000
2 2,INDIVIDUO 2,individuo_2@mail.com,3.2500,3.6667,3.4444,3.5000,3.6000
3 3,INDIVIDUO 3,individuo_3@mail.com,2.8750,4.0000,3.3333,2.5000,4.1000
4 4,INDIVIDUO 4,individuo_4@mail.com,3.5000,3.3333,3.8889,3.1250,3.8000
5 5,INDIVIDUO 5,individuo_5@mail.com,4.1250,3.8889,3.1111,2.0000,3.9000
6 6,INDIVIDUO 6,individuo_6@mail.com,3.2500,3.4444,3.1111,3.1250,3.4000
7 7,INDIVIDUO 7,individuo_7@mail.com,2.2500,3.5556,4.1111,2.1250,3.8000
8 8,INDIVIDUO 8,individuo_8@mail.com,3.1250,3.7778,3.5556,2.1250,3.4000
9 9,INDIVIDUO 9,individuo_9@mail.com,4.5000,3.6667,2.2222,1.6250,4.4000
```

Cuando el proceso de formación de grupos ha finalizado es posible generar un archivo PDF con la información tal como se muestra en la siguiente figura.

Herramienta TeamBE

Listado de grupos formados automáticamente
Generado el 2020-11-25 18:21:10

Grupo #1

- INDIVIDUO 18 (individuo_18@mail.com)
- INDIVIDUO 1 (individuo_1@mail.com)

Grupo #2

- INDIVIDUO 13 (individuo_13@mail.com)
- INDIVIDUO 10 (individuo_10@mail.com)

Grupo #3

- INDIVIDUO 2 (individuo_2@mail.com)
- INDIVIDUO 3 (individuo_3@mail.com)
- INDIVIDUO 19 (individuo_19@mail.com)

Grupo #4

- INDIVIDUO 17 (individuo_17@mail.com)
- INDIVIDUO 7 (individuo_7@mail.com)
- INDIVIDUO 16 (individuo_16@mail.com)

Paso 5: Cuando se han formado correctamente los grupos en la vista principal de TEAM-BE se debe hacer clic en el botón OK, entonces se inicia el proceso de creación automática de sesiones y usuarios, en el siguiente orden:

- Se registran todos los usuarios en el servidor, independientemente del grupo que se les haya asignado
- Se crean los repositorios GIT remotos para cada sesión en Bitbucket mediante la API pública

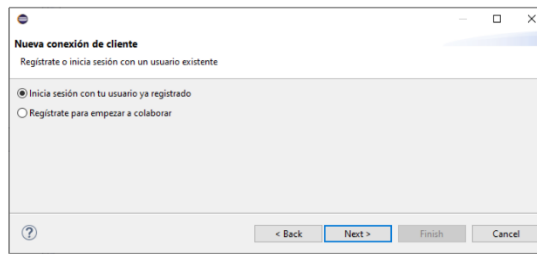
- Se registran una sesión por cada grupo formado
- Se registra cada usuario en la sesión correspondiente al grupo que le fue asignado

Cuando termine el proceso descrito los usuarios pueden unirse a la sesión que les corresponda desde una instancia cliente de Collece 2.0 usando el correo electrónico como usuario y contraseña.

3.3 INICIO DE SESIÓN COMO CLIENTE EN COLLECE

Para iniciar sesión como un usuario que ha sido registrado automáticamente por la herramienta TEAM-BE se debe realizar el siguiente proceso:

Paso 1: Se debe hacer clic en *File* → *New* → *Other* y en la categoría COLLECE-2.0 seleccionar *Client*, se despliega la siguiente ventana, se debe seleccionar la opción de iniciar sesión con un usuario registrado



Paso 2: Una vez el usuario ha iniciado sesión es dirigido al gestor de sesiones de Collece, donde puede visualizar las sesiones alojadas en el servidor, debe seleccionar la sesión a la que pertenezca y presionar el botón *Unirse*. La herramienta realizará todos los demás pasos necesarios, incluyendo clonar el repositorio remoto, este proceso puede tardar algunos segundos, una vez termine el usuario podrá editar los archivos fuente de forma colaborativa y tendrá acceso al chat grupal.

Si no se carga la perspectiva Collece Cliente automáticamente puede hacerse haciendo clic en:

Window → *Perspective* → *Open Perspective* → *Other* → *COLLECE-2.0 (client)*

Anexo 7. Resumen presentado en CACIED 2019

4º Congreso Andino en Computación, Informática y Educación – CACIED 2019

TEAM-BE: Plugin de Eclipse para la conformación de grupos en escenarios de aprendizaje colaborativo

Juan Pablo Botina Carlosama

Universidad de Nariño, Pasto, (Colombia)
juanpablobc@udenar.edu.co

Oscar Revelo Sanchez

Universidad de Nariño, Pasto, (Colombia)
orevelo@udenar.edu.co

Jesús Insuasti Portilla

Universidad de Nariño, Pasto, (Colombia)
insuasty@udenar.edu.co

Resumen

Introducción: La escritura colaborativa de código es una tendencia en el mundo del desarrollo actual y su aplicación en entornos educativos ayuda a motivar a nuevos programadores ayudando en la adopción de una perspectiva de trabajo en equipo. Collece 2.0 es un plugin para el IDE Eclipse que integra varias herramientas para el trabajo colaborativo distribuido, pero actualmente no posee una para la conformación automática de grupos. Team-B es una librería java que cumple este propósito, el objetivo del presente proyecto es integrar Team-B al entorno Collece 2.0

Objetivo: Se pretende incorporar al entorno de trabajo colaborativo Collece 2.0 la librería Team-B como un módulo para la conformación automática de equipos.

Metodología: Para el desarrollo del software se empleará la popular metodología ágil SCRUM, las iteraciones contempladas comprenden la recolección de información sobre la arquitectura y funcionamiento de Collece 2.0 y la librería Team-B, el análisis de requerimientos, diseño y desarrollo del módulo, así como su documentación.

Resultados: Al final del proceso se obtendrá una herramienta para la conformación automática de grupos que será de gran utilidad para los docentes de cursos iniciales de programación.

Conclusiones: Teniendo en cuenta que actualmente no existen herramientas que integren todas las funcionalidades que ofrece Collece 2.0 (incluyendo la conformación automática de grupos) y que el entorno Collece permite la adición de nuevos módulos se observa que existe viabilidad técnica y operativa para el desarrollo del proyecto, el cual pondrá a disposición de la comunidad académica una herramienta útil en una estrategia alternativa de enseñanza en los cursos de programación.

Palabras claves: Programación de computadores; Programación Colaborativa; Entorno de desarrollo integrado (IDE); Conformación de grupos.

San Juan de Pasto, noviembre 6, 7 y 8 de 2019

Anexo 8. Resumen presentado en EXPOPOSTER 2020

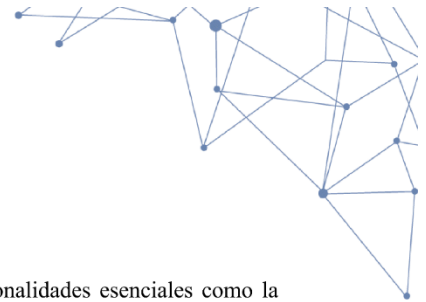


actuales requieren que su aprendizaje sea rápido y compatible con nuevos paradigmas de modo que los estudiantes puedan aprender y adaptarse a nuevos marcos de trabajo y reaccionar a los cambios de los mismos de forma oportuna, por otro lado las características de los proyectos de desarrollo en la actualidad hacen necesario equipos más o menos numerosos cuyos miembros posean habilidades técnicas y sociales que se complementen con las de sus compañeros. Para esto la academia juega un papel fundamental pues es allí donde el individuo adquiere la mayoría de estas habilidades y también las necesarias para adquirir otras junto con nuevo conocimiento. En este contexto el aprendizaje colaborativo representa una alternativa interesante a la enseñanza tradicional, pues en este el aprendizaje pasa a ser un proceso social donde las interacciones entre estudiantes pasan a tener el protagonismo y el profesor se convierte en un facilitador del aprendizaje, esto se apoya en la teoría del constructivismo social que sostiene que “las personas activamente construyen conocimiento mientras interactúan con su ambiente” [2]. En este contexto se presenta Collece 2.0, una herramienta para apoyar el aprendizaje colaborativo de la programación de manera distribuida y en tiempo real, de modo que varios participantes pueden trabajar en un problema determinado simultáneamente por medio de internet, además ofrece otras funcionalidades para apoyar el proceso tales como chat y bloqueo de regiones; por sus características Collece tiene potencial para usarse en entornos educativos y ayudar al desarrollo de las habilidades antes mencionadas. Por otro lado existe la librería de Java TeamB que permite formar grupos de forma automática por medio de algoritmos genéticos tomando como insumo diversas características, por ejemplo, los rasgos de personalidad. El principal objetivo de este trabajo de grado fue integrar esta librería a Collece para favorecer su utilización en entornos de aprendizaje al automatizar un proceso que tradicionalmente se realiza de forma manual con la ventaja de tener grupos con individuos compatibles. Para lograrlo se definió un proceso de varias fases que empezó por la apropiación del conocimiento teórico y técnico necesario para el desarrollo y terminó en la creación de la herramienta y su documentación, actualmente la mayoría de objetivos se encuentran cumplidos y se está realizando la aplicación de la herramienta en un entorno educativo real con estudiantes de ingeniería de sistemas de la Universidad de Nariño, una vez realizada se medirá la pertinencia y aceptación de la misma por parte de los estudiantes participantes por medio de una encuesta.

Palabras clave: Formación de grupos, Trabajo Colaborativo, IDE, Desarrollo de plugins

Problema de investigación: Los avances tecnológicos actuales permiten realizar tareas de programación en escenarios de trabajo colaborativo con diversas ventajas y estabilidad. Para ello





es necesario contar con una plataforma que integre ciertas funcionalidades esenciales como la escritura de código distribuida y en tiempo real, herramientas de comunicación y la organización automática de equipos con integrantes compatibles.

En el momento Collece 2.0 satisface la mayoría de estas necesidades como lo son la escritura de código remota en tiempo real, chat, y bloqueo de regiones de código, pero no dispone de una herramienta de formación automática de grupos. Esta tarea se realiza manualmente de forma externa a la plataforma. La librería Team-B desarrollada en la Universidad de Nariño, satisface esto último, pero en el momento no es posible conectarla directamente con Eclipse.

Referente teórico: Collece 2.0 (COLLaborative Edition, Compilation and Execution of programs) es un plugin de Eclipse que provee un conjunto de herramientas para facilitar el aprendizaje de la programación mediante la colaboración remota y asíncrona de varios usuarios para resolver un problema propuesto. Provee un chat integrado, bloqueo de regiones de código y por su naturaleza modular permite agregarle nuevas funcionalidades [1].

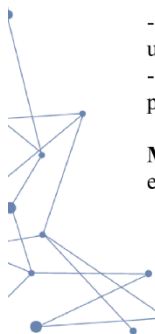
Team-B 2.0: es una librería Java™ para la formación homogénea, heterogénea y mixta de grupos, basada en múltiples características para la creación de aplicaciones que permitan formar grupos de manera automática para escenarios de aprendizaje colaborativo con individuos que presenten diferentes características cuantificables, para esto se hace uso de un algoritmo evolutivo como técnica de optimización [3]

Objetivo general: Incorporar la librería Team-B al entorno de trabajo colaborativo College 2.0 como un módulo para formación automática de grupos.

Objetivos específicos: - Caracterizar las diferentes técnicas de creación de plugins para el IDE Eclipse.

- Conocer la arquitectura general del College 2.0.
- Conocer el funcionamiento de la librería Team-B.
- Desarrollar un módulo para College 2.0 que incorpore la formación automática de equipos utilizando la librería Team-B.
- Validar el funcionamiento de la integración College2.0 – Team-B en cursos iniciales de programación.

Metodología: Al tratarse de un desarrollo de software para complementar un software ya existente primero fue necesario apropiarse el conocimiento teórico y técnico necesario para



posteriormente aplicarlo por medio de una metodología ágil, en este caso se implementó SCRUM, que si bien no es completamente aplicable al no estar presentes todos los roles necesarios contiene herramientas útiles como las iteraciones (sprints), historias de usuario y la planeación de entregables que sirvieron para mantener un flujo de trabajo ordenado y reaccionar oportunamente a los cambios en los requerimientos.

Resultados: 1. Módulo de formación automática de equipos basado en la librería Team-B para Collece 2.0
2. Documentación técnica del Know-How del módulo desarrollado
3. Los docentes de cursos iniciales de programación pueden desarrollar actividades colaborativas en sus espacios académicos utilizando el entorno Collece 2.0 y su módulo de formación automática de grupos de trabajo

Conclusiones: 1. En la actualidad del desarrollo de software las herramientas para el trabajo colaborativo a distancia son indispensables, si bien en ambientes profesionales se utilizan diversos IDEs y software especializado como Git o Mercurial la herramienta Collece2.0 / Team-BE constituye una herramienta didáctica útil para facilitar el aprendizaje realista de la programación.
2. Team-BE añade la funcionalidad de formación automática de grupos al entorno Collece 2.0 lo que permite organizar individuos según sus características de personalidad, esto favorece las interacciones valiosas entre los integrantes del grupo que podrán aprovechar las herramientas que provee Collece como edición colaborativa de código en tiempo real, bloqueo de regiones y chat.

Referencias: [1] Grupo Computer Human Interaction and Collaboration (CHICO), "COLLECE-2.0," 2019. <http://blog.uclm.es/grupochico/proyecto-iapro/collece-2-0/> (accedido Nov. 30, 2019).
[2] M. Maldonado Pérez, "El Trabajo Colaborativo En El Aula Universitaria," Laurus - Rev. Educ., vol. 13, no. 23, pp. 263–278, 2007.
[3] O. Revelo Sánchez, J. Insuasti Portilla, and A. Barón Salaza, "TEAM-B v2.0 Librería JavaTM para la formación homogénea, heterogénea y mixta de grupos, basada en múltiples características." Universidad de Nariño. Facultad de Ingeniería, Pasto, 2019.



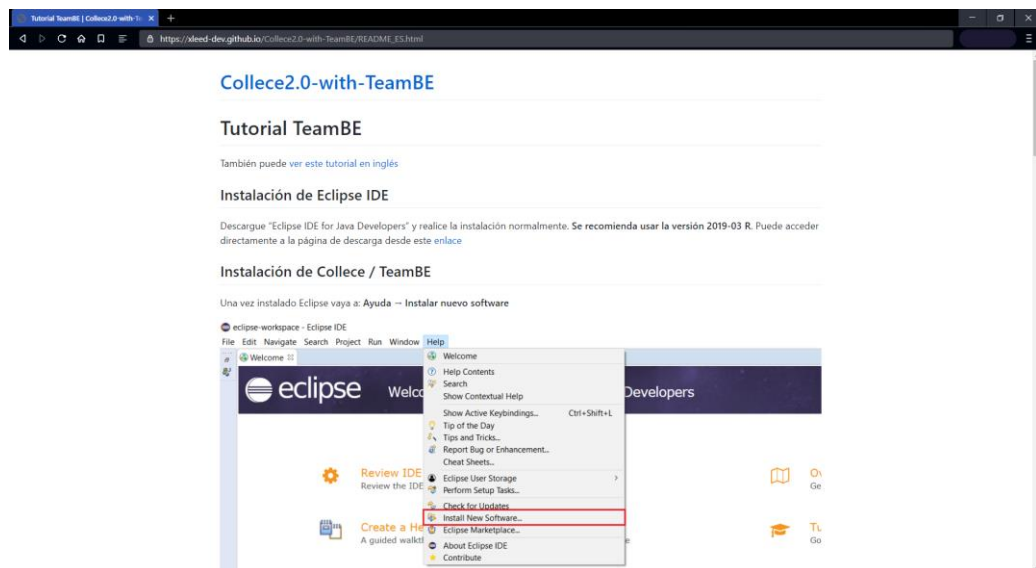
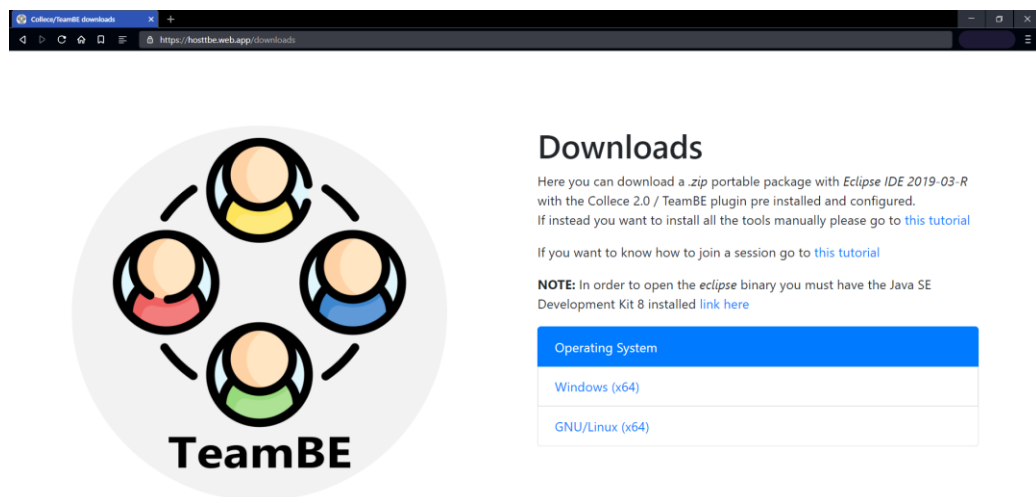
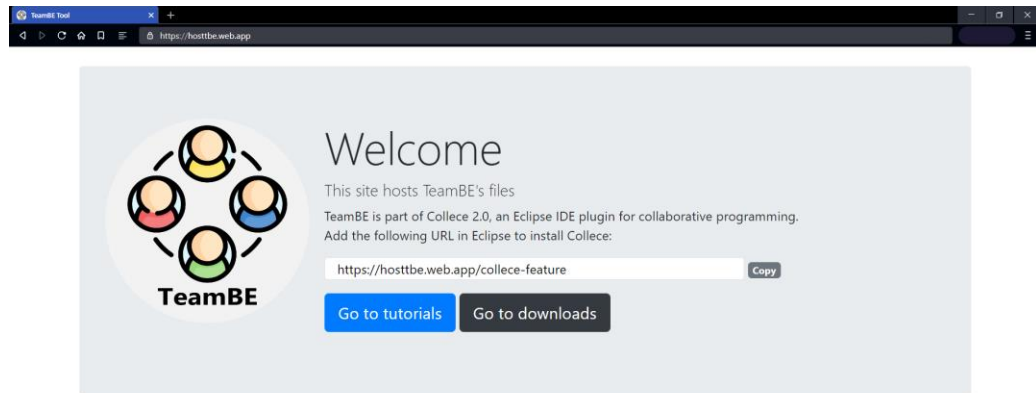
Anexo 9. Captura de pantalla registro DNDA

 MINISTERIO DEL INTERIOR DIRECCION NACIONAL DE DERECHO DE AUTOR UNIDAD ADMINISTRATIVA ESPECIAL OFICINA DE REGISTRO CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE		Libro - Tomo - Partida 13-83-413 Fecha Registro 17-feb.-2021
Page 1 of 2		
1. DATOS DE LAS PERSONAS		
AUTOR		
Nombres y Apellidos	OSCAR REVELO SANCHEZ	No de identificación CC 11438636
Nacional de	COLOMBIA	
Dirección	MZ B C24 SAUCES LA CAROLINA	Ciudad: PASTO
AUTOR		
Nombres y Apellidos	JESUS INSUASTI PORTILLA	No de identificación CC 98392695
Nacional de	COLOMBIA	
Dirección	CALLE 16 NO 40 16 B. SANTA ANA	Ciudad: PASTO
AUTOR		
Nombres y Apellidos	JUAN PABLO BOTINA CARLOSAMA	No de identificación CC 1085324872
Nacional de	COLOMBIA	
Dirección	CRA 4A # 19-73 B/BERNAL	Ciudad: PASTO
PRODUCTOR		
Razón Social	UNIVERSIDAD DE NARIÑO	Nit 800118954-1
Dirección	BRR CIUDAD UNIVERSITARIA TOROB	Ciudad: PASTO
2. DATOS DE LA OBRA		
Título Original	TEAM-BE MÓDULO PARA LA FORMACIÓN AUTOMÁTICA DE GRUPOS COLABORATIVOS EN EL ENTORNO COLLECE 2.0	
Año de Creación	2020	Pais de Origen COLOMBIA Año Edición
CLASE DE OBRA	INEDITA	
CARACTER DE LA OBRA	OBRA DERIVADA	
CARACTER DE LA OBRA	OBRA EN COLABORACION	
ELEMENTOS APORTADOS DE SOPORTE LOGICO	PROGRAMA DE COMPUTADOR	
ELEMENTOS APORTADOS DE SOPORTE LOGICO	DESCRIPCIÓN DEL PROGRAMA	
ELEMENTOS APORTADOS DE SOPORTE LOGICO	MATERIAL AUXILIAR	
3. DESCRIPCIÓN DE LA OBRA		
LA HERRAMIENTA DESARROLLADA ES UN MÓDULO PARA EL PLUGIN COLLECE 2.0 QUE PERMITE LA FORMACIÓN AUTOMÁTICA DE GRUPOS Y AUTOMATIZA EL PROCESO DE REGISTRO DE USUARIOS Y LA CREACIÓN DE SESIONES POR CADA GRUPO EN EL SISTEMA JUNTO CON LOS REPOSITORIOS GIT REMOTOS EN LA PLATAFORMA BITBUCKET NECESARIOS PARA CADA UNA. LA VISTA PRINCIPAL DE TEAM-BE HACE PARTE DEL ASISTENTE DE CREACIÓN DE SESIONES.		
4. OBSERVACIONES GENERALES DE LA OBRA		
5. DATOS DEL SOLICITANTE		
Nombres y Apellidos	JUAN PABLO BOTINA CARLOSAMA	No de Identificación 1085324872
Nacional de	COLOMBIA	Medio Radicación REGISTRO EN LINEA
Dirección	CRA 4A # 19-73 B/BERNAL	Ciudad PASTO
Correo electrónico	JUANPABLOBC495@GMAIL.COM	Teléfono 3147881953
En representación de	EN NOMBRE PROPIO	Radicación de entrada 1-2020-147602

	MINISTERIO DEL INTERIOR DIRECCION NACIONAL DE DERECHO DE AUTOR UNIDAD ADMINISTRATIVA ESPECIAL OFICINA DE REGISTRO	Libro - Tomo - Partida 13-83-413
	<u>CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE</u>	Fecha Registro 17-feb.-2021
Page 2 of 2		
<div style="text-align: center;"> _____ JULIAN DAVID RIATIGA IBÁÑEZ JEFE OFICINA DE REGISTRO (E)</div>		
MZIP		

Nota: El derecho de autor protege exclusivamente la forma mediante la cual las ideas del autor son descritas, explicadas, ilustradas o incorporadas a las obras. No son objeto de protección las ideas contenidas en las obras literarias y artísticas, o el contenido ideológico o técnico de las obras científicas, ni su aprovechamiento industrial o comercial (artículo 7o. de la Decisión 351 de 1993).

Anexo 10. Capturas de pantalla sitio web TEAM-BE



Fuente: Creación propia