

Universidad Autónoma de Baja California

Facultad de Ciencias Ensenada



**Soporte para el inicio de interacciones de ayuda en el DSD
con base en el nivel de involucramiento de los colaboradores
potenciales**

TESIS

que presenta para obtener el grado de MAESTRO EN CIENCIAS

GENOVEVA VIRIDIANA SILVA RODRÍGUEZ

DIRECTOR DE TESIS:

DR. ALBERTO LEOPOLDO MORÁN Y SOLARES

CO-DIRECTOR DE TESIS:

DR. VÍCTOR MANUEL GONZÁLEZ Y GONZÁLEZ

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS
UNIDAD ENSENADA

**Soporte para el inicio de interacciones de ayuda en el DSD con base
en el nivel de involucramiento de los colaboradores potenciales**

TESIS

Que para obtener el grado de maestría en ciencias presenta:

Genoveva Viridiana Silva Rodríguez

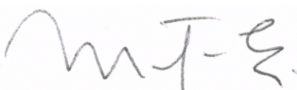
Aprobada por:



Dr. Alberto L. Morán y Solares
Director de Tesis



Dr. Víctor M. González y González
Codirector de Tesis



Dra. Mónica E. Tentori Espinosa
Miembro del comité



Dr. Ramón R. Palacio Cinco
Miembro del comité

Ensenada Baja California, México. Enero del 2013

RESUMEN de la tesis de **Genoveva Viridiana Silva Rodríguez**, presentada como requisito parcial para la obtención del grado de **MAESTRO EN CIENCIAS**. Ensenada Baja California, México, Enero del 2013.

Soporte para el inicio de interacciones de ayuda en el DSD con base en el nivel de involucramiento de los colaboradores potenciales

Resumen aprobado por:



Dr. Alberto L. Morán y Solares
Director de Tesis



Dr. Víctor M. González y González
Co-director de Tesis

Durante el proceso de desarrollo de software, los desarrolladores comparten e intercambian conocimientos e información relacionados con su trabajo interactuando constantemente unos con otros. Esto se vuelve complicado en entornos distribuidos, como los de Desarrollo Distribuido de Software (DSD, por sus siglas en inglés), debido a la distancia física y temporal que impide el acceso expedito a los miembros del equipo. Un desarrollador a menudo se enfrenta a la necesidad de identificar a quién contactar, con el fin de resolver un problema. Esta problemática, es un reto que exige identificar elementos de información que apoyen a contrarrestar la carencia de conciencia debida a la distancia geográfica.

En esta tesis se aborda de manera particular lo referente a cómo los trabajadores en DSD pueden ser apoyados por la tecnología para decidir a quién contactar en el momento en el que enfrentan algún problema o requieren de información que les impide continuar desempeñando su trabajo. Con esto se propone que el desarrollador que

requiere la ayuda cuente con información relevante de los colaboradores potenciales, de acuerdo a lo que se requiere para resolver el problema que enfrenta. Con dicha información, el desarrollador solicitante podría inferir quién es el colaborador potencial que más le conviene contactar para iniciar una colaboración de ayuda.

Para lograr esto, se introduce y define con base en la literatura un modelo de métricas para la identificación de colaboradores potenciales, mediante la conciencia del Involucramiento, Experiencia y Disponibilidad de los colegas. Se argumenta que estos tres conceptos pueden utilizarse para aumentar la conciencia acerca del nivel de participación, habilidades, experiencias y conocimiento entre los miembros de una organización. Lo cual permite crear el conocimiento compartido de la tarea y del equipo, que permita a los ingenieros de software alcanzar la familiaridad entre los miembros de una organización.

Se valida este modelo con el diseño y desarrollo de un sistema interactivo con dos formas de visualización de información. Con la finalidad de medir el impacto del modelo propuesto y la facilidad de uso del prototipo del sistema interactivo de visualización de información, se obtuvo la percepción de 26 desarrolladores profesionales a través de entrevistas semi-estructuradas, cuestionario SUS y 7 preguntas adicionales. Los resultados obtenidos indicaron que los desarrolladores consideran uno o más de los elementos de conciencia del modelo propuesto, durante el proceso de toma de decisión respecto a quién acudir en situaciones de ayuda. Además, percibieron que después de usar el sistema por un tiempo, la presentación de esta información mediante los sistemas de visualización propuestos resulta más útil que la presentación de información en una visualización tabular simple.

Palabras Clave: *Colaboradores potenciales, Sistemas colaborativos, Desarrollo Distribuido de Software, Interacciones de ayuda, Involucramiento, Expertos.*

ABSTRACT of the thesis, presented by **Genoveva Viridiana Silva Rodríguez**, in order to obtain the MASTER of SCIENCE. Ensenada Baja California, México, Enero del 2013.

Soporte para el inicio de interacciones de ayuda en el DSD con base en el nivel de involucramiento de los colaboradores potenciales

Approved by:



Dr. Alberto L. Morán y Solares
Thesis Advisor



Dr. Víctor M. González y González
Thesis Co-advisor

During the software development process, developers share and exchange knowledge and information related to their work by constantly interacting among them. This becomes difficult in distributed environments, such as in Distributed Software Development (DSD), due to the physical and temporal distance that hampers readily access to team members. A developer is often faced with the need to identify whom to contact in order to solve a problem. This problem is a challenge that requires identifying information elements to cope with the lack of awareness due to geographical distance.

In this thesis we particularly address how DSD workers can be supported by technology to decide whom to contact at the moment at which they face a problem or require further information that prevents them from performing their work. Here we propose that a developer who requires this support could acquire relevant information about the potential collaborators, according to what is required to solve the problem faced.

With this information, the developer in need could infer who is the most convenient potential collaborator to contact to initiate a help interaction.

To achieve this, based on a literature review we introduce and define a model of metrics for identifying potential collaborators through an awareness of involvement, experience and availability of their colleagues. It is argued that these three concepts can be used to raise awareness about the level of participation, skills, experiences and knowledge among the members of an organization. This allows the creation of a shared understanding of the task and the team, which allows software developers to achieve familiarity between the members of an organization.

This model was validated through the design and development of an interactive system with two modalities of visualizing information. In order to measure the impact of the proposed model and the usability of the interactive information visualization modalities, the perception about the system was obtained from 26 professional developers through semi-structured interviews, the SUS questionnaire and 7 additional questions. The results indicate that developers consider one or more of the awareness elements of the proposed model during their process to decide to whom to turn for help in those situations. Furthermore, they perceived that after using the system for a while, the presentation of this information using the proposed visualization modalities is more useful than the presentation of the information using a simple tabular visualization.

Keywords: *Potential collaborators, collaborative systems, Distributed Software Development, Help interactions, Involvement, Experts.*

Dedicada

A Dios, por darme la fortaleza para poder concluir este trabajo de investigación y por cuidar de mi familia y de mí en estos años de estar lejos de ellos.

A mis padres Gerardo y Josefina con todo mi amor, por siempre creer en mí y motivarme en todos estos años de vida a dar lo mejor, a ponerme metas, luchar por conseguirlas y darme la herencia más preciada que son mis estudios.

A mi hermoso hermano Gerardo, por confiar en mí y creer que puedo lograr lo que me propongo. Recuerda flakito: "Querer es poder".

A mi amado Betito, por apoyarme en esta etapa de mi vida y ser paciente en esos momentos difíciles.

A mi abuelita Genoveva, por consentirme cada visita a casa y esperar siempre lo mejor de mí. A mis seres queridos que me acompañan desde el cielo, mis abuelitos (Lola y Jaime), mi tía Juanita y mi tía Gela.

Agradecimientos

A mis dos directores de tesis: Dr. Alberto Leopoldo Morán y Solares y el Dr. Víctor Manuel González y González. Por sus consejos, tiempo, dedicación y el gran apoyo brindado para realizar este trabajo de investigación.

A la Dra. Aurora Vizcaíno Barceló por su disposición, comentarios y consejos que contribuyeron de gran forma al logro de este trabajo de investigación.

A los miembros de mi comité Dra. Mónica Tentori y Dr. Ramón Palacio Cinco, cuyos comentarios y consejos contribuyeron a este trabajo de investigación.

A la familia Martínez Nava (Anita, Sandra y Paco) por su amistad y apoyo que me han brindaron desde mi llegada a Ensenada.

Al Instituto Tecnológico Autónomo de México (ITAM) por haberme recibido y apoyado durante mi estancia académica.

A Ismael Hernández, Daniel Miranda y Javier Villagrana por su valiosa amistad y apoyo durante esta etapa.

A la Facultad de Ciencias de la UABC y al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico y todos los recursos brindados durante mis estudios de maestría.

A mis compañeros del MYDCI, especialmente a Claudia, Mary, Arturo, Vicky, Raymundo, Everardo, Ana, Julián y Sixto, con quienes conviví y compartí buenas experiencias durante mi estancia en UABC.

Índice general

Resumen	III
Dedicatoria	VII
Agradecimientos	VIII
Índice de figuras	XII
Índice de tablas	XV
1. Introducción.	1
1.1. Antecedentes.	3
1.2. Planteamiento del problema	7
1.2.1. Preguntas de Investigación	8
1.3. Objetivos de la investigación	8
1.3.1. Objetivo general	8
1.3.2. Objetivos específicos	8
1.4. Metodología de la investigación	9
1.5. Contribuciones principales de esta tesis	11
1.6. Contenido del documento	12
2. Trabajo relacionado.	15

2.1. Sistemas de conciencia.	18
2.2. Sistemas de localización de expertos	24
2.3. Visualización de Información	31
2.4. Ideas de diseño	58
2.5. Resumen del capítulo	59
 3. Involucramiento, Experiencia y Disponibilidad: apoyo a inicio de interacciones de ayuda en desarrollo distribuido de software.	60
3.1. Situaciones en las que se requiere identificar colaboradores potenciales.	61
3.2. Modelo para identificación de colaboradores potenciales para inicio de interacciones de ayuda.	66
3.2.1. Involucramiento.	68
3.2.2. Experiencia.	70
3.2.3. Disponibilidad.	73
3.3. Proceso del modelo propuesto.	74
3.3.1. Flujo de trabajo del modelo.	75
3.3.2. Identificar a colaboradores potenciales por su Involucramiento. .	76
3.3.3. Identificar a colaboradores potenciales por su Experiencia. . . .	78
3.3.4. Identificar a colaboradores potenciales por su Disponibilidad. . .	80
3.3.5. Identificar colaboradores potenciales mediante un Espacio compartido.	81
3.4. Resumen del capítulo.	83
 4. Sistema Interactivo de Visualización de Información.	85
4.1. Visualizaciones.	86
4.1.1. Involucramiento.	88
4.1.2. Experiencia.	89
4.1.3. Disponibilidad.	91

4.2. Sistemas Interactivos.	92
4.2.1. Sistema Interactivo en forma horizontal.	93
4.2.2. Sistema Interactivo en forma de campana.	96
4.3. Funcionamiento del sistema.	99
4.4. Resumen del capítulo.	101
5. Resultados.	102
5.1. Objetivos.	103
5.2. Hipótesis.	103
5.3. Diseño del estudio.	104
5.4. Participante.	105
5.5. Procedimiento.	105
5.6. Resultados.	107
5.7. Discusión.	114
5.8. Resumen del capítulo.	119
6. Conclusiones, contribución y trabajo futuro.	121
6.1. Conclusiones.	122
6.2. Contribución.	124
6.3. Trabajo futuro.	126
Referencias	128
Lista de Apéndices	140
A. Cuestionario escala 7-Likert	140
B. SUS, System Usability Scale	142

Índice de figuras

1.1. Matriz de espacio y tiempo (tomado de [1]).	5
1.2. Transiciones entre trabajo individual y colaborativo (tomado de [2]). . .	6
1.3. Metodología seguida durante el proceso de investigación.	10
2.1. Perspectivas de estudio para abordar problemas de coordinación en DSD.	17
2.2. Interfaz web del sistema TeamSpace (tomado de [3])	33
2.3. Interfaz del sistema Rear View Mirror (tomado de [4])	35
2.4. Interfaz del sistema Expertise Browser (tomado de [5])	36
2.5. Interfaz del sistema Palantír (tomado de [6])	37
2.6. Interfaz del sistema ProjectWatcher (tomado de [7])	38
2.7. Interfaz de los complementos de Caise (tomado de [8])	39
2.8. Interfaz de Project View (tomado de [9])	40
2.9. Interfaz de CommunityNet (tomado de [10])	41
2.10. Interfaz de Celine (tomado de [11])	41
2.11. Interfaz de ExpertiseNet (tomado de [12])	42
2.12. Interfaz de Constellation (tomado de [13])	42
2.13. Interfaz del sistema FastDash (tomado de [14])	44
2.14. Interfaz del sistema Expertise Cloud (tomado de [15])	45
2.15. Interfaz del sistema Ensemble (tomado de [16])	45
2.16. Interfaz del sistema Impromptu (tomado de [17])	46
2.17. Visualizaciones de SmallBlue (tomado de [18])	47

2.18. Interfaz del sistema Tesseract (tomado de [19])	49
2.19. Interfaz del sistema WIPDash (tomado de [20])	50
2.20. Interfaz del sistema CWS-IM (tomado de [21])	52
2.21. Visualizaciones de Scamp (tomado de [22])	53
2.22. Interfaz del sistema Saros (tomado de [23])	54
2.23. Visualización de CodeBook (tomado de [24])	55
2.24. Visualización de Hoozizat (tomado de [25])	56
2.25. Interfaces de Trusty (tomado de [26])	57
3.1. Elementos de información y métricas que contribuyen a responder la pregunta ¿Quiénes son las personas que podrían tener la información para resolver el problema?.	67
3.2. Flujo de trabajo para identificar colaboradores potenciales que apoyen a trabajadores en DSD en la resolución de un problema o duda.	75
3.3. Entradas y salidas para determinar colaboradores potenciales por Invo- lucramiento.	77
3.4. Entradas y salidas para determinar colaboradores potenciales por Expe- riencia.	79
3.5. Entradas y salidas para determinar colaboradores potenciales por Dis- ponibilidad.	80
3.6. Métricas que contribuyen a la identificación de colaboradores potenciales obtenidas a partir de un espacio compartido.	82
4.1. Visualización general de un colaborador potencial.	87
4.2. Visualización del involucramiento de un colaborador potencial.	89
4.3. Visualización de la experiencia de un colaborador potencial.	90
4.4. Interfaz del sistema interactivo de visualización en forma horizontal. . .	93

4.5. Información adicional mostrada como porcentajes en la visualización general en el sistema en forma horizontal.	96
4.6. Interfaz del sistema interactivo de visualización en forma de campana. .	97
4.7. Detalles de la visualización general a través de la técnica FishEye en el sistema en forma de campana.	99
4.8. Representación del funcionamiento de un sistema interactivo de visualización para la identificación de colaboradores potenciales.	100
5.1. Tareas correctas y parcialmente correctas.	109
5.2. Tiempos obtenidos en la identificación de colaboradores potenciales. . .	111
5.3. Puntuaciones del SUS obtenidas por forma de visualización.	112
5.4. Puntuaciones del cuestionario escala 7-Likert obtenidas por forma de visualización.	113

Índice de tablas

2.1. Elementos de conciencia de literatura de trabajo colaborativo.	19
2.2. Heurísticas para identificación de expertos utilizadas por sistemas de localización de expertos.	26
2.3. Técnicas de visualización de información utilizadas por los trabajos de la literatura revisada.	32
4.1. Visualización de la disponibilidad de un colaborador potencial por op- ciones de filtrado.	91
4.2. Opciones de filtrado en el sistema en forma horizontal.	95
4.3. Opciones de filtrado en el sistema en forma de campana.	98
5.1. Definición de los escenarios utilizados en la evaluación.	106

Capítulo 1

Introducción.

Diariamente las personas se enfrentan a situaciones en las que tienen que decidir a quién contactar con el objetivo de resolver una duda o problema. De lo mundano a lo especializado, identificar a los colaboradores potenciales es un reto que implica un análisis mental que integra una variedad de criterios para apoyar nuestras decisiones. Estos desafíos son particularmente arduos cuando se requiere de información y conocimiento especializado. En el contexto del desarrollo de software, se presentan mayormente escenarios en los cuales se requiere identificar a colaboradores potenciales (p.ej. falta de entendimiento en algoritmos específicos, carecer de conocimiento en cuestiones técnicas, no entender el funcionamiento interno de las estructuras de datos o malinterpretar el flujo de trabajo en el cual se basa el software). Algunos problemas podrían llegar a tener efectos colaterales, tales como la incorrecta modificación de un bloque de código o el provocar retardos en las actividades del grupo. En muchos de estos casos, la mejor manera de encontrar una respuesta es a través de iniciar colaboraciones potenciales, para lo cual se requiere identificar a colaboradores potenciales que puedan proporcionar una orientación o información al respecto.

En el desarrollo distribuido de software (DSD, por sus siglas en inglés), en donde se lleva a cabo la separación de los procesos de desarrollo y sus colaboradores en dife-

rentes localizaciones geográficas, se presentan barreras que dificultan la capacidad de los desarrolladores para comunicarse y coordinar adecuadamente su trabajo, así como tener conciencia de información relevante de los demás desarrolladores. Esto, a su vez afecta la naturaleza de la interacción dentro de los equipos de trabajo al existir pocas oportunidades para la interacción espontánea y para la adquisición del conocimiento del equipo [27]. Por esta razón, generalmente se requiere de más tiempo para que los miembros de un equipo distribuido obtengan el conocimiento, reciban una respuesta, o corrijan un problema de comunicación.

Este cambio de paradigma exige nuevos mecanismos que permitan a los miembros de los equipos de DSD localizar a colaboradores que tengan información relevante y útil, y que puedan contactarse de una manera sencilla y eficaz, para resolver un problema, independientemente de que no los conozcan, o se encuentren físicamente lejos.

En esta tesis de manera particular se aborda lo referente a cómo los trabajadores en DSD pueden ser apoyados por la tecnología para decidir a quién contactar en el momento en el que tienen que enfrentar algún problema o requieren de información que les impide continuar desempeñando su trabajo, y cuando se encuentran en un contexto reducido de conocimiento del equipo y una comunicación cara a cara reducida, debido a la dispersión geográfica. Con esto se propone que el desarrollador solicitante cuente con información relevante de los colaboradores potenciales, de acuerdo a lo que se requiere para resolver el problema que enfrenta. Con dicha información, el desarrollador solicitante podría inferir quién es el colaborador potencial que más le conviene contactar para iniciar una colaboración. Para ello se propone un modelo de métricas para la identificación de colaboradores potenciales y un sistema interactivo de visualización de información. La usabilidad de dicho sistema prototipo se evaluó a través de escenarios, en los cuáles se ejemplificaron situaciones en las que los trabajadores en DSD enfrentan problemas que requieren de la localización de colaboradores potenciales. Los resultados de la evaluación proporcionan evidencia que muestra la aceptación favorable del sistema

interactivo de visualización por parte de los participantes en las evaluaciones en términos de usabilidad y apoyo en el proceso cognitivo de la identificación de colaboradores potenciales.

En este capítulo introductorio se describen antecedentes y el planteamiento del problema, luego se presentan las preguntas de investigación que guiaron el trabajo, para enseguida definir los objetivos a alcanzar. Posteriormente se detalla la metodología de trabajo utilizada en la realización de esta tesis, las principales contribuciones y se finaliza con la descripción del contenido del resto del documento.

1.1. Antecedentes.

El desarrollo profesional de software es una actividad compleja, por lo general es realizado por un grupo de personas que se especializan en diferentes áreas y que individualmente producen componentes de software que son integrados en un solo producto de software. Estos componentes son generados siguiendo una agenda a priori, la cual establece *qué* es lo que se debe de producir, *quién* lo debe de producir, *cómo* se debe de producir y *cuándo* se debe de producir. Esto es necesario para que cada una de las partes resultantes del esfuerzo individual se integren e inter-operen adecuadamente.

Este enfoque crea dependencias entre las tareas realizadas, y a su vez entre las personas que las realizan. Para desempeñar efectivamente las tareas, los trabajadores deben de coordinar sus esfuerzos, no sólo los relacionados específicamente con la tarea (“taskwork”), sino también en relación al trabajo grupal (“teamwork”).

En los esfuerzos del desarrollo profesional de software, las tareas tienen un nivel de complejidad inherente debido a su tamaño y estructura. Cuando las tareas son pocas, sencillas y bien definidas, es más fácil identificar y comprender qué partes de la tarea afectan a otras partes. Desde el punto de vista del “taskwork”, cuando el número de tareas y el número de relaciones entre ellos aumenta, la realización de esas tareas se

hace difícil porque los individuos tienen mayores dificultades para integrar las partes de una tarea e identificar cómo afectan a otras tareas (“task complexity”) [28]. Desde el punto de vista de “teamwork”, la complejidad de la tarea puede variar dependiendo del número de individuos que trabajan en ella y la forma en que se encuentran organizados y distribuidos geográficamente. Cuando las condiciones son difíciles para que los miembros se comuniquen y coordinen entre sí, con el fin de gestionar eficazmente las dependencias entre tareas (p. ej. el incremento en la distancia geográfica entre ellos), se dice que aumenta la complejidad debido al trabajo grupal (“Team coordination complexity”) [28].

De acuerdo con Espinosa et al. [29], una manera de lidiar con los problemas de coordinación es a través de la comunicación y el conocimiento del equipo, no sólo acerca de la tarea (p. ej. el conocimiento compartido de la tarea y la conciencia de la tarea), sino también sobre los miembros del equipo (p. ej. el conocimiento compartido del equipo y la conciencia de la presencia). Este conocimiento compartido se puede generar a través de lo que se denomina “familiaridad” [28], que se define como el conocimiento que se genera conforme los miembros de un equipo trabajan juntos. Esto permite que se conozcan unos a otros y que los miembros se habitúen a los principales problemas, lo que lleva a facilitar la interacción entre los miembros del equipo e identificar la ubicación de las fuentes de conocimiento y experiencia dentro del equipo.

La creación de este conocimiento (alcanzar la familiaridad) entre los miembros de un equipo distribuido geográficamente es más complejo que cuando los miembros están co-localizados. Esto se debe a que la distancia disminuye las oportunidades de interacción cara a cara, tanto de manera formal como informalmente [28, 29].

De acuerdo con las dimensiones de espacio-tiempo propuestas por Ellis et al. [1], el trabajo en grupo puede realizarse de manera co-localizada (mismo lugar) o distribuido (diferentes lugares) (ver Figura 1.1).

	Mismo tiempo	Diferentes tiempos
Mismo lugar	Interacción cara a cara	Interacción asíncrona
Diferentes lugares	Interacción distribuida síncrona	Interacción distribuida asíncrona

Figura 1.1: Matriz de espacio y tiempo (tomado de [1]).

En el trabajo en grupo de manera co-localizada, los miembros se encuentran a la vista unos de otros, esto es, si un miembro del grupo trata de saber si un colega de trabajo está disponible para solicitarle ayuda, sólo basta con mirar en ese momento hacia su lugar de trabajo para conocer sus actividades y disponibilidad en el momento. De esta manera, si un colega se encuentra en su lugar de trabajo (mismo lugar – mismo tiempo) es posible realizar interacciones cara a cara con él, en caso contrario (mismo lugar – diferentes tiempos) se le podría dejar un mensaje para cuando regrese pueda enterarse del motivo de la interacción.

Por otro lado, en el trabajo en grupo de manera distribuida los miembros no se encuentran a la vista, lo que impide poder obtener información de lo que sucede en su espacio de trabajo en determinado momento (diferentes lugares – mismo tiempo), por lo que generalmente se recurre a herramientas de comunicación síncronas como el teléfono o la videoconferencia para interactuar. De otra forma (diferentes lugares – diferentes tiempos) son posibles las interacciones asíncronas, con los miembros del grupo.

En los equipos de desarrollo de software co-localizados, Kraut y Streeter [27] encontraron que la proximidad física entre los miembros del equipo permite interacciones informales cara a cara. Esto a su vez, permite algunas funciones tales como el segui-

miento de personas, tomar o dejar mensajes, hacer arreglos para una reunión, entregar documentos, dar o recibir ayuda, informar sobre el progreso y comunicar noticias [30], se lleven acabo de una manera sencilla y ligera.

Existe una relación entre el nivel de complejidad de una tarea y el grado de formalidad de la comunicación, por lo que las tareas más complejas requieren mayor comunicación [30]. Al inicio de un proyecto, como parte de la organización y control a través de metodologías preestablecidas, se definen los roles y actividades a realizar por parte de cada uno de los miembros de manera individual, pero a su vez son parte de actividades que conforman un trabajo colaborativo. Para ello, se pueden realizar transiciones del trabajo individual al trabajo colaborativo sin ningún problema y viceversa [2] (ver Figura 1.2).

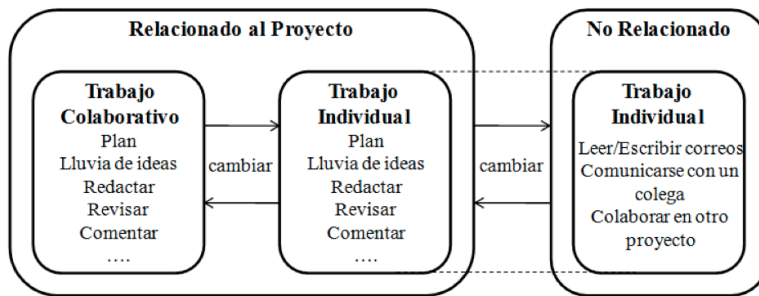


Figura 1.2: Transiciones entre trabajo individual y colaborativo (tomado de [2]).

A lo largo de esta línea de investigación, también se encontró que en los equipos co-localizados, las personas son la fuente más utilizada y valorada en los proyectos de desarrollo de software, comparado con los documentos implicados [28]. También se encontró que los ingenieros de software obtienen la información que necesitan primordialmente de las personas con las que tienen proximidad física, y que la facilidad con la que la información se puede obtener es un factor determinante.

Estudios de observación de las prácticas de desarrollo de software de un equipo colocalizado, revelan que la comunicación informal es una actividad importante entre

este tipo de trabajadores. Kuroki et al. [31] indican que, los desarrolladores invierten alrededor del 33% de su tiempo de trabajo comunicándose cara a cara con sus colegas para poder de esta manera, i) entender las actividades desarrolladas por terceros, ii) identificar la siguiente actividad que deben desarrollar, e iii) identificar con quien necesitan interactuar para encontrar una solución a un problema en particular.

Por lo tanto, el comportamiento típico cuando un desarrollador enfrenta un problema que no puede resolver por sí mismo (p. ej. que no entiende el flujo de trabajo que el software debe de soportar, no conoce cuál de los módulos cambiar para corregir el defecto, no cuenta con la información necesaria para aclarar un documento de especificación funcional, etc.) es contactar a un colega cercano [30], es decir, a un miembro del equipo que le sea “familiar” y que se encuentre físicamente cerca.

Esto sin embargo, en equipos de desarrolladores en DSD resulta más complejo, dado que la familiaridad se reduce y el concepto de “cercanía” cambia debido a la dispersión geográfica de los miembros.

1.2. Planteamiento del problema

De manera particular, este trabajo será enfocado en investigar cómo los desarrolladores en DSD pueden ser apoyados por la tecnología y les ayude a decidir a quién contactar cuando se les presenta un problema o requieren de información que les impide continuar desempeñando su trabajo, y cuando se encuentran en un contexto reducido (o inexistente) de familiaridad (conocimiento compartido sobre el equipo y la tarea) y una comunicación cara a cara reducida (formal e informal), debido a la dispersión geográfica.

1.2.1. Preguntas de Investigación

Con base a la problemática descrita en los antecedentes mencionados previamente, se han definido las siguientes preguntas que guían este trabajo de investigación:

- En primer lugar se requiere dar respuesta a la pregunta: ¿Cómo determinar el momento y bajo qué condiciones un ingeniero de software requiere identificar un colaborador potencial entre los miembros de una organización?
- Posteriormente será necesario responder a la pregunta: ¿Cuáles son los criterios para identificar un colaborador potencial entre los miembros de una organización?
- Con lo que podremos dar respuesta a la pregunta: ¿Cuáles son las formas de representar y mostrar a un ingeniero de software la información de los criterios que identifican a un colaborador potencial entre los miembros de una organización?
- Finalmente es importante conocer ¿Cuál es el impacto de identificar a colaboradores potenciales en el trabajo de un ingeniero de software?

1.3. Objetivos de la investigación

1.3.1. Objetivo general

Modelar, caracterizar y evaluar los inicios de interacciones informadas bajo el supuesto de que se requiere la identificación de colaboradores potenciales entre los miembros de una organización para maximizar los beneficios de una interacción.

1.3.2. Objetivos específicos

Con el fin de lograr el objetivo general, se desprenden los siguientes objetivos específicos:

- Caracterizar en el desarrollo distribuido de software, las situaciones en las que un ingeniero de software tiene una necesidad o problema que requiera identificar a un colaborador potencial para iniciar una interacción de ayuda.
- Modelar cómo los miembros de una organización identifican a colaboradores potenciales dentro de este, en caso de un problema o necesidad.
- Determinar las métricas para identificar a colaboradores potenciales entre los miembros de una organización.
- Identificar la información y mecanismos que proporcionen conciencia de los criterios que identifican a un colaborador potencial entre los miembros de una organización.
- Validar el modelo de identificación de colaboradores potenciales a través del diseño de dos sistemas interactivos de visualización de información.
- Medir el impacto del modelo propuesto para la identificación de colaboradores potenciales sobre un ambiente controlado con miembros de varios equipos de trabajo en el desarrollo de software.

1.4. Metodología de la investigación

La metodología de investigación utilizada para el desarrollo de este trabajo consta de seis etapas (ver Figura 1.3). Enseguida se describen las actividades llevadas a cabo en las diferentes etapas de esta metodología.

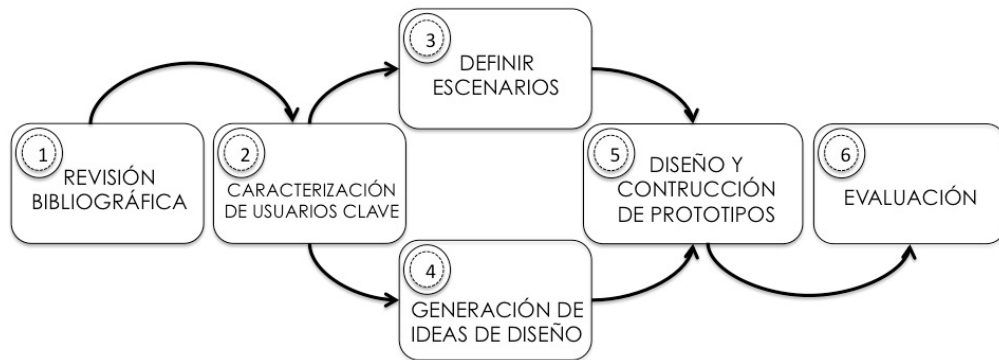


Figura 1.3: Metodología seguida durante el proceso de investigación.

1. Revisión bibliográfica. En esta etapa se realizó un análisis bibliográfico con la finalidad de identificar los trabajos previos realizados para el tema propuesto y fundamentar el marco teórico para el presente trabajo. Con esta etapa se obtuvo un modelo de trabajo para DSD y un conjunto de hallazgos que sustentan la creación de una nueva propuesta para la identificación de colaboradores potenciales cuando se presenta una duda o problema.
2. Caracterización de usuarios clave. Esta etapa consistió en realizar un análisis de las situaciones en las que los trabajadores del DSD enfrentan dudas o problemas. La finalidad de esta etapa, fue la de comprender e identificar las necesidades de identificación de colaboradores potenciales entre los desarrolladores de una organización durante el desarrollo de software. Con esta etapa se obtuvo la caracterización de actividades de DSD, en donde se presentan frecuentemente situaciones en las que requieren de apoyo para resolver un problema o duda.
3. Definición de escenarios. Se estudiaron distintos escenarios, los cuales representaban situaciones en las que los desarrolladores enfrentaban la necesidad de localizar a un colega para resolver una duda o problemas. Con base a esto, se visualizó y contextualizó la problemática referente a las necesidades de identificación de co-

laboradores potenciales en DSD.

4. Generación de ideas de diseño. Con base a la literatura revisada, la caracterización de usuarios clave y la definición de escenarios, se determinaron las características que se toman en cuenta para la identificación de colaboradores potenciales en DSD. A través de estas características se propuso un modelo de métricas basado en un conjunto de elementos de información identificados.
5. Diseño y construcción de prototipos. Con base en las ideas de diseño de la etapa anterior y consultando literatura respecto a visualización de información, se definió las formas de presentación de cada una de las métricas definidas. A partir de estas visualizaciones individuales, se diseñaron dos prototipos de sistemas interactivos de visualización.
6. Evaluación. Teniendo dos formas de visualización de información, se procedió a realizar una evaluación basada en escenarios, con el objetivo de obtener la utilidad y facilidad de uso del funcionamiento de un sistema interactivo de visualización para la identificación de colaboradores potenciales por parte de los trabajadores de DSD. Además, se realizó un análisis detallado de los datos obtenidos en el caso de estudio, con los cuales también se pudieron generar las conclusiones de esta investigación.

Resumidas las seis etapas que conforman la metodología utilizada en esta investigación, se pasará a definir las partes por las que está constituida la presente tesis.

1.5. Contribuciones principales de esta tesis

Las principales contribuciones derivadas de este trabajo de tesis se presentan a continuación:

1. *Un modelo que obtenga y represente la información de Involucramiento, Experiencia y Disponibilidad del desarrollador, la cual es requerida para identificar colaboradores potenciales en situaciones de ayuda.* Este modelo está basado en un análisis de literatura, tanto de sistemas de conciencia y sistemas de localización de expertos. Éste fue creado dada la variedad de instancias contempladas para la identificación de expertos en el desarrollo de software. Por lo cual, con la finalidad de proveer al trabajo de DSD tanto de conocimiento de la tarea como del equipo, se creó un modelo de métricas que sirva para la identificación de colaboradores potenciales. Dicho modelo de métricas presentado en el Capítulo 3, fue base para comprender el conjunto de métricas necesarias para la identificación de las personas que pueden apoyar en resolver una duda o problema en DSD.

Para validar el modelo de métricas para la identificación de colaboradores potenciales, se diseñó con base en la literatura de visualización de información dos sistemas interactivos de visualización de información. Los cuales a través de una visualización general que incorporaba los tres conceptos (Involucramiento, Experiencia y Disponibilidad), permitía a los desarrolladores explorar y obtener conciencia en menor tiempo, de la identidad de un grupo de colaboradores potenciales de acuerdo al contexto de su duda o problema. Dichos prototipos interactivos se presentan en el Capítulo 4, y fueron base para posteriormente realizar una evaluación con usuarios potenciales.

1.6. Contenido del documento

El resto de este documento está dividido en 6 capítulos y 2 apéndices, el contenido de cada uno de ellos se describe brevemente a continuación:

En el capítulo 2 se presenta un resumen general de trabajos de investigación que

aportan algunos elementos que intervienen en la identificación de colaboradores potenciales en DSD. Estos trabajos son parte de las áreas de investigación: sistemas de conciencia, sistemas de localización de expertos y visualización de información.

En el capítulo 3 se describe un modelo para la identificación de colaboradores potenciales creado a partir del análisis de literatura de coordinación de actividades de DSD, identificación de fuentes de conocimiento en DSD y recomendación de expertos. Además, se describe cómo son obtenidos los elementos de información del modelo propuesto.

En el capítulo 4 se presentan el diseño de dos sistemas interactivos de visualización de información, que permitan validar el modelo de identificación de colaboradores potenciales propuesto. Además, se describe el funcionamiento de un sistema interactivo de información al momento de que un desarrollador requiere de identificar colaboradores potenciales, para atender una situación de ayuda.

En el capítulo 5 se describe la evaluación basada en escenarios, la cual consistió en presentar información de colaboradores potenciales a través de un sistema interactivo de visualización a desarrolladores profesionales. La finalidad de esta evaluación es contrastar una presentación tabular simple de información de colaboradores potenciales, con las dos modalidades de visualización de información propuestas. La comparación se realizará en términos de facilidad de uso y apoyo en la identificación de colaboradores potenciales en situaciones de ayuda.

En el capítulo 6 se muestran las conclusiones y reflexiones derivadas de este trabajo; las principales aportaciones, así como algunas oportunidades de trabajo futuro identificadas durante el desarrollo de la misma.

El apéndice A presenta el cuestionario de utilidad empleado para el caso de estudio con los desarrolladores profesionales de DSD.

El apéndice B presenta el cuestionario SUS aplicado en el caso de estudio con los desarrolladores profesionales de DSD.

Capítulo 2

Trabajo relacionado.

El desarrollo de software es una actividad colaborativa en donde grupos de desarrolladores trabajan juntos en un tiempo global. Durante el desempeño de las tareas, los trabajadores necesitan mantener conciencia de cómo determinadas tareas o artefactos del proyecto están progresando, lo que los demás trabajadores realizaron o se encuentran realizando y el estado actual de los recursos asociados al proyecto.

Es posible analizar la colaboración a través de la comunicación, la coordinación y la cooperación, lo cual Fuks llamó modelo de colaboración 3C [32]. En este modelo se observa que i) durante la comunicación, las personas hacen negociaciones y toman decisiones, para poder generar compromisos; ii) durante la coordinación, se organizan y gestionan las tareas y las personas involucradas; y iii) durante la cooperación, se ejecutan las tareas por parte de los miembros del grupo a través de una operación conjunta en un espacio compartido, en la cual manipulan un conjunto de artefactos compartidos.

La coordinación puede ser vista como el eslabón de conexión entre la comunicación y la colaboración dada la necesidad que surge de coordinar personas, recursos y tareas. La coordinación de personas, por ejemplo, está profundamente relacionada con la comunicación y el contexto. La coordinación de los recursos, por otra parte, está relacionada

con el espacio compartido (es decir, cooperación). Finalmente, la coordinación de tareas consiste en la gestión de interdependencias entre las tareas que se llevan a cabo para lograr un objetivo [32].

La separación física de los trabajadores en el desarrollo distribuido de software (DSD por sus siglas en inglés), impone desafíos relacionados con el apoyo de la conciencia durante la colaboración en este paradigma de desarrollo. Esto es porque la información de conciencia necesaria en tales escenarios es tácita, inherente, dinámica y contextual. Es tácita ya que la mayoría de lo que los trabajadores de DSD hacen en los espacios de colaboración se basa en la experiencia, habilidades, heurísticas e interacciones que no siempre se encuentran documentadas [33]. Por otro lado, cuando un software es producido desde múltiples zonas geográficas, resulta más complicado gestionar y coordinar las dependencias de tareas, incrementando el tiempo de desarrollo [34].

En este capítulo se presenta una revisión de la literatura que recopila, analiza y clasifica estudios sobre los problemas de coordinación causados por la naturaleza de los equipos en DSD. Se presenta un análisis de algunos trabajos, clasificados con base en los esfuerzos realizados para enfrentar los problemas de coordinación en DSD, los cuales han sido abordados por los investigadores desde tres perspectivas: *sistemas de conciencia*, *sistemas de localización de expertos* y *sistemas de visualización de información* (ver Figura 2.1).



Figura 2.1: Perspectivas de estudio para abordar problemas de coordinación en DSD.

En cuanto a los sistemas de conciencia, esta perspectiva ha sido abordada por algunos investigadores con la finalidad de apoyar a los trabajadores a tener a su disposición información referente al control de versiones del código fuente [8, 11, 20, 22], a las tareas de un proyecto [3, 4, 7, 9, 17, 21], al seguimiento de los artefactos y las relaciones entre ellos [6, 14, 23, 24], y al análisis de las dependencias socio-técnicas [19, 26], entre otras, que a su vez les permite crear supuestos realistas acerca del proyecto.

El interés de investigadores pertenecientes a la comunidad del trabajo colaborativo asistido por computadora (CSCW, por sus siglas en inglés) ha aumentado en lo referente a la creación y gestión del conocimiento en entornos de colaboración. Así se puede observar como los sistemas de localización de expertos consisten en la gestión de conocimiento y sistemas de distribución, esto con la finalidad de apoyar a los usuarios a rastrear información humana y fuentes de conocimiento en lugar de los documentos [35].

Finalmente, algunos de los sistemas de conciencia y de recomendación identificados, se han apoyado en diversas herramientas y técnicas de sistemas de visualización. Así se puede ver como algunos de los trabajos de investigación se basan en técnicas de visualización para aumentar cualquiera de las vistas existentes en los entornos de desarrollo o para proporcionar puntos de vista especializados de las actividades humanas en

combinación con la información del artefacto del software.

2.1. Sistemas de conciencia.

Para coordinarse, los miembros del equipo de trabajo deben de ser conscientes de sus tareas y las interdependencias. Dourish y Bellotti [36] describen la conciencia como la comprensión de las actividades de otros, lo que proporciona un contexto de su propia actividad. Para ello, se han destinado los sistemas de conciencia para apoyar a las personas a construir y mantener conciencia de las actividades que realizan los demás, el contexto o situación, sobre todo cuando los participantes se encuentran distribuidos. Este apoyo resulta importante en el incremento del trabajo colaborativo, fomentando de esta forma las relaciones sociales y la mejora del bienestar general de los individuos [37,38].

El concepto de conciencia es aplicable en una variedad de actividades y áreas, sin embargo dentro de la disciplina de las ciencias de la computación una de éstas es CSCW. En este contexto, el concepto de colaboración potencial [39] identifica una serie de elementos para la colaboración en grupo. Con esto es posible descubrir o crear oportunidades para colaborar en momentos apropiados. Investigaciones en el campo de CSCW se han centrado en el uso de estos elementos de conciencia de las actividades de desarrollo, con la finalidad de facilitar la coordinación en el desarrollo de software (ver Tabla 2.1).

Tabla 2.1: Elementos de conciencia de literatura de trabajo colaborativo.

Categoría	Elementos de conciencia	TeamSpace (2001)	Project Watcher (2004)	IMPROMPTU (2008)	CWV-IM (2010)	Rear View Mirror (2002)	ProjectView-IM (2004)	Caise (2004)	Celine (2005)	Syde (2010)	WIPDash (2009)	Palantir (2003)	FASTDash (2007)	Saros (2010)	Codebook (2010)	Tesseract (2009)	Trusty (2011)
Quién	Presencia	X			X	X	X										X
	Identidad	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	Autoría	X			X	X		X	X		X	X	X	X	X		X
	Agrupado	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
Dónde	Localización	X	X		X	X											X
	Visión	X	X		X				X			X				X	X
	Vista	X		X				X		X	X	X	X	X	X	X	X
	Alcance	X								X	X	X	X	X	X		X
Qué	Acciones	X	X	X				X	X	X	X	X	X	X	X		
	Propósito	X					X			X	X	X					
	Artefactos	X	X	X	X	X		X	X	X	X	X	X	X	X		
	Habilidades				X												X
Por qué	Motivación										X						X
Cuándo	Expectativa										X					X	X
Cómo	Historial	X	X		X			X	X	X	X						X
	Nivel de actividad		X		X			X	X	X	X	X	X	X			X

Para proporcionar y distribuir información de conciencia, algunos trabajos de investigación han sido propuestos para la presentación de información referente a las tareas de un proyecto, tales como:

TeamSpace [3] apoya a equipos en DSD mediante la administración de procesos de trabajo compartidos y el mantenimiento de artefactos compartidos en un proyecto. Este sistema gestiona las reuniones asincrónicas y da acceso a los encuentros sincrónicos de forma distribuida. Estos datos se integran a la información referente a la gestión de documentos, tareas, personas y proyectos, de manera que los miembros de un equipo de trabajo tomen conciencia de las tareas relacionadas y del seguimiento de las mismas.

ProjectWatcher [7] observa y registra información puntual sobre modificaciones realizadas por el usuario. Además, proporciona una visión general de trabajadores involucrados en el proyecto, qué artefactos han estado trabajando y en qué proyecto han estado trabajando.

IMPROMPTU (IMPROving MDE's Potential to support Tasks that are genUine) es un marco de trabajo [17] que permite a los usuarios compartir información de las tareas a través de despliegues de ambiente. Este sistema proporciona una representación visual de los miembros del grupo de trabajo y sus actividades. De esta manera se permite interacción conjunta con la información del problema a resolver, la cual se comparte en los despliegues ambientales para su discusión y reflexión.

CWS-IM (Collaborative Working Sphere – Instant Messaging, por sus siglas en inglés) [21], por un lado tiene como objetivo permitir el manejo de una colaboración potencial en situaciones síncronas, tales como: i) conocimiento de la disponibilidad de los desarrolladores y el rol que desempeñan en el momento, ii) conocimiento de las unidades de trabajo asignadas de manera individual y en grupos, iii) notificación a los miembros del equipo de trabajo cuando una unidad de trabajo ha finalizado, y iv) recepción de notificaciones implícitas respecto a las unidades de trabajo en las cuales el trabajador participa. La herramienta presenta información sobre situaciones pasadas o

futuras para poder mantener colaboraciones potenciales en situaciones asíncronas.

Rear View Mirror (RVM) [4] es una aplicación que combina la mensajería instantánea y características de la conciencia de presencia. Para apoyar a equipos de trabajo distribuidos a mantener la conciencia de los acontecimientos del entorno de trabajo, se proporciona información de la identidad y presencia de los miembros que integran un equipo de trabajo. RVM está construido bajo el modelo cliente-servidor, ejecutando en la máquina del usuario cada cliente.

ProjectView-IM [9] sugiere modificaciones en el diseño a la interfaz de mensajería instantánea para enfrentar problemas de memoria y administración de trayectorias. De acuerdo al estudio realizado, los investigadores sugieren tres características a incluir en los sistemas de mensajería para facilitar la gestión de trayectorias de trabajo. Dichas características consisten en: i) un componente de conciencia de actividades generales de los colaboradores y su disponibilidad, ii) un componente de información del nivel de compromiso de un colaborador en una tarea conjunta, y iii) un componente que proporcione recordatorios referente a las actividades que el usuario requiere atender.

Otro método utilizado para proporcionar conciencia para apoyo en la coordinación, es la recopilación de información del repositorio de control de versiones del código fuente. En este caso, algunos trabajos relacionados son:

Caise [8] es una arquitectura de software que con base en comentarios proporciona a los desarrolladores conciencia de la localización del resto de los desarrolladores, de los posibles conflictos y de la necesidad de una colaboración más estrecha. Esta arquitectura proporciona soporte en tiempo real, para la colaboración entre desarrolladores y herramientas colaborativas.

Celine [11] es un sistema que proporciona información referente a los conflictos de artefactos, soportado por diferentes topologías (modelo caótico, estrella y estrella extendida). Este sistema soporta la gestión de conflictos a través de un mecanismo de

conciencia, que permite conocer la identidad de los desarrolladores que han realizado modificaciones en los artefactos del proyecto, así como las diferencias entre la copia del artefacto y el artefacto modificado.

Syde [22] es una herramienta de colaboración que proporciona conciencia de las acciones de los miembros del equipo de trabajo, por medio de la propagación de los cambios que ocurren en el momento, y la notificación de posibles conflictos a causa de cambios realizados. Syde recopila datos de una manera transparente y no intrusiva para el usuario.

WIPDash (Work Item and People Dashboard, por sus siglas en inglés) [20] es un sistema que tiene como objetivo apoyar a los equipos de software para obtener conciencia de la situación general de un proyecto, y entender las actividades en curso. Dicha información se presenta en una pantalla compartida ubicada en un espacio en común para el equipo de desarrollo de software.

El seguimiento de los artefactos y las relaciones entre ellos también se utilizan para apoyo en la coordinación en los proyectos de DSD. Herramientas que consideran este enfoque incluyen:

Palantír [6] es una herramienta que proporciona información de conciencia de espacio de trabajo como complemento de los sistemas de gestión de configuración existente. De manera particular, esta herramienta informa al desarrollador la identidad de los colegas que realizan modificaciones a ciertos artefactos, calcula una medida de severidad simple relacionada a los cambios y despliega dicha información de una manera configurable y no-invasiva.

FASTDash [14] es una herramienta que utiliza una representación del código fuente compartido, extiende esta conciencia y resalta las actividades actuales de los miembros del equipo. Esta información le permite a un desarrollador determinar cuáles miembros del equipo tienen acceso a archivos desprotegidos, y cuáles archivos están siendo

visualizados, así como qué métodos y clases están siendo modificados, proporcionando conciencia inmediata de posibles situaciones de conflicto.

Saros [23] es un plug-in para el entorno de desarrollo integrado (IDE, por sus siglas en inglés) Eclipse, el cual permite a los desarrolladores acceso recurrente a los artefactos compartidos. Además permite tener conocimiento de las modificaciones realizadas a los artefactos y los responsables de cada uno de estos cambios.

El framework Codebook [24] maneja una amplia variedad de problemáticas de los trabajos realizados anteriormente, analiza varios tipos de artefactos, y es personalizable por y para los usuarios finales. Además permite identificar y presentar las relaciones transitivas entre las personas, código, errores, casos de prueba, entre otros artefactos relacionados.

También la aplicación de análisis de las dependencias socio-técnicas apoya en la detección de problemas de coordinación. Herramientas que toman este enfoque incluyen:

Teserract [19] muestra las múltiples relaciones que existen entre artefactos, desarrolladores, errores y comunicaciones. Además, provee de la conciencia de las actividades de los desarrolladores con otros miembros de la organización, lo cual permite la identificación de los miembros de los equipos de trabajo involucrados, así como el poder predecir defectos en la etapa de desarrollo.

Trusty [26], es una red social basada en web (WBSN por sus siglas en inglés) diseñada para soportar el proceso de DSD, con el objetivo de mejorar la comunicación y la coordinación en los equipos virtuales, así como de facilitar la creación de confianza entre los miembros para beneficio de los proyectos de DSD. Los elementos de información que proporciona son: información referente a los miembros de la organización, reportes y análisis estadístico relacionados con la red social de la misma. Además, provee de diferentes mecanismos para compartir información informal, para comunicación y para la administración de conocimiento.

Como se puede percibir, los trabajos anteriores distribuyen más de un elemento de conciencia entre miembros de una organización de DSD como apoyo a la coordinación. Sin embargo, cuando los trabajadores de DSD se enfrentan a la necesidad de resolver problemas para cumplir con las tareas asignadas, estos requieren de invertir tiempo para determinar la información de su interés. Esto significa realizar un esfuerzo extra para encontrar una solución o localizar a una persona que cuente con la experiencia necesaria para apoyar. Dentro de los trabajos anteriores resulta recurrente el uso de elementos como la identidad y presencia, como parte de la visión general de las actividades relacionadas al proceso de desarrollo de software. La coordinación de las actividades es posible al conocer quiénes se encuentran participando en qué actividades, así como la disponibilidad actual de cada uno de los miembros.

Para apoyar al proceso natural que realizan los trabajadores de software para la localización de expertos, se requiere identificar determinados elementos de información. Por ello, en el siguiente apartado se presentan contribuciones relacionadas con las heurísticas utilizadas para la localización de expertos dentro de un grupo de trabajo.

2.2. Sistemas de localización de expertos

Todos los días, las personas en las organizaciones deben de resolver problemas para cumplir con el trabajo asignado. A menudo, una gran cantidad de esfuerzos se realizan para encontrar una solución o una persona que tiene la experiencia necesaria para apoyar. El esfuerzo para entender quién cuenta con la experiencia, los aspectos sociales y cognitivos específicos de la persona indicada, incluye la localización de expertos.

La localización de expertos anteriormente se ha abordado a través de estudios de desempeño de expertos, la localización de fuentes de información y el intercambio interpersonal. Sin embargo, pocos trabajos se han enfocado en aumentar el comportamiento

natural de las personas para la localización de expertos.

Los sistemas de recomendación han resultado en una solución técnica para apoyar a los usuarios en la localización de expertos, representando una serie de intentos para la resolución de problemas con la entrega y la sobrecarga de información. Estos sistemas son conocidos también como sistemas de filtrado social y filtrado de colaboración, ya que mantienen un perfil de cada usuario en función del histórico de la utilización del sistema o información proporcionada por el usuario [40].

Como parte del análisis de la literatura referente a sistemas de localización de expertos, se distinguen diferentes fuentes de datos técnicos y sociales utilizados para proporcionar recomendaciones. Algunas fuentes utilizadas son: las contribuciones del trabajador al código fuente, archivos y comunicaciones del trabajador, información de redes sociales y combinación de criterios técnicos y sociales (ver tabla 2.2).

Tabla 2.2: Heurísticas para identificación de expertos utilizadas por sistemas de localización de expertos.

Trabajos	Recomendación por otros usuarios	Historial de control de cambios	Relaciones con otros miembros	Relaciones con artefactos	Historial de conversaciones y correo electrónico	Número de personas que se han apoyado	Fuentes explícitas proporcionadas por el usuario
Expertise Recommender (2001)	X	X		X			
Expertise Browser (2002)		X		X			
Expert Finder (2004)			X	X			
CommunityNet (2005)					X		
ExpertiseNet (2005)			X		X		
Step_IN (2007)				X			
Constellation (2007)					X		
QuMe (2007)			X			X	
Presley (2007)		X	X	X	X		
Emergent Expertise Locator (2007)		X		X			
Expertise Cloud (2008)			X	X			
Ensemble (2008)		X	X	X			
SmallBlue (2009)					X		
Hoozizat (2010)				X			
Conclus (2010)			X	X	X		
IM-an-Expert (2011)				X			X

El uso de la información del repositorio de control de versiones y el análisis de redes sociales (SNA, por sus siglas en inglés) para buscar expertos (o identificar la propiedad de artefactos) son mecanismos que se han utilizado para proporcionar conciencia en apoyo a la coordinación. Algunos trabajos que consideran esta aproximación incluyen:

Expertise Recommender (ER, por sus siglas en inglés) de McDonald [40, 41] se implementó con base en dos heurísticas: apoyo técnico e historial de cambios. Esta última heurística utiliza la “Línea 10”, norma que establece que los autores de la revisión son los expertos del archivo. En ER los expertos se clasifican de acuerdo con el tiempo de

revisión, de manera que el último desarrollador en modificar el archivo tiene un rango más alto. ER genera recomendaciones y devuelve una lista de posibles personas de contacto, esta lista no está estrictamente ordenada por la experiencia de los individuos, sino que ER se basa tanto en la probable experiencia como en aspectos sociales para crear y ordenar la lista de recomendaciones.

Expertise Browser [5], es una herramienta que busca resolver el problema de búsqueda de expertos. Para realizarlo utiliza los datos del sistema de administración de cambios para localizar a la gente con el nivel de experiencia deseado. Para lograrlo la herramienta cuantifica la experiencia y presenta evidencia para validar esta cuantificación como una medida del nivel de experiencia. De manera fácil es posible encontrar personas que trabajaron en determinadas partes de un proyecto, y se puede visualizar cómo su experiencia se distribuye sobre el producto (módulo o subsistema).

Presley [42] es una herramienta que tiene como objetivo facilitar la identificación y recomendación de expertos en un proyecto. La herramienta perfila a los expertos con base en la información del código fuente, relaciones de los artefactos codificados, histórico del control de versiones y registros de comunicaciones entre los miembros de los equipos.

ExpertFinder [43] es una herramienta para localizar a un experto a través del uso de información de la organización. En la práctica, esta información es la documentación generada durante un proyecto o actividad similar. Se generan modelos de experiencia utilizando algoritmos de modelado de texto basado en el modelo de espacio vectorial para analizar los documentos. Además cuenta con filtros que reciben los resultados del análisis y selección de las personas más relacionadas con un documento en específico.

El sistema ExpertiseNet [12] tiene como objetivo el describir y actualizar el perfil de una persona experta. A este perfil de experiencia se le llama un ExpertNet y se representa a través de grafos que describen las relaciones y la evolución de la experiencia personal. Tanto el contenido del texto como las citas de los documentos se analizan para

extraer la relación y la información de la experiencia de la persona.

El sistema CommunityNet [10] modela y predice el comportamiento humano cuando se requiere recibir y difundir información, esto a través del análisis de contactos y contenidos de las comunicaciones personales. El sistema crea un perfil personal llamado CommunityNet, que integra automáticamente el contenido de correo electrónico y el análisis de redes sociales.

La herramienta Step-IN (socio-technical platform for in-situ networking) [44], soporta colaboración de conocimiento situado (situated knowledge collaboration). Este sistema auxilia a un desarrollador de software a encontrar a colaboradores que primero tengan el nivel de conocimientos (experto) sobre la información de interés y segundo que tengan cierta voluntad de ofrecer ayuda de manera oportuna. La identificación de estos expertos se basa en un perfil técnico, el cual se crea con base en el conteo del número de ocasiones en que ha interactuado con el método sobre el cual se tiene interés.

DiMicco et al. propone la red social Constellation [13] que a través de una visualización de redes sociales, permite a los usuarios descubrir, explorar las conexiones sociales entre los expertos y poder trazar el camino más corto entre ellos. Constellation combina información de tres mecanismos de obtención: red social de las empresas, base de datos de patentes y base de datos de publicaciones. Con apoyo de una búsqueda de expertos a través de palabras clave es posible identificar, localizar y conectarse con los expertos a distancia.

El sistema Question Matching Engine (QuMe) [45] es una red de pregunta-respuesta que atiende problemas de comunidades tales como foros. QuMe infiere el nivel de experiencia de los participantes de un foro a través del uso de datos de las redes sociales de la comunidad en línea. A partir de esta información, se genera una clasificación de experiencia con base al número de personas que se ha apoyado.

Minto y Murphy [46] presenta Emergent Expertise Locator, que utiliza los clientes de mensajería instantánea incrustados en los entornos de trabajo (IDE) para crear

conciencia de los expertos. Minto y Murphy utilizan un método propuesto por Cataldo et al. [47], el cual mide las dependencias de tareas entre los trabajadores, a través de la comprobación de la “congruencia” entre estas dependencias de tareas y las actividades de coordinación realizadas por los trabajadores.

En el entorno del software de código abierto (OSS, por sus siglas en inglés), Alonso et al. propone Expertise Cloud [15] que consiste en un enfoque que utiliza la clasificación del árbol de código fuente como un camino para contactar a los expertos. El mecanismo de detección de experiencia utiliza un método de clasificación basado en reglas, que examina la relación entre expertos y desarrolladores, así como la cantidad de apoyo que realmente ofrecen los expertos. Para obtener esta información, se extrae una fuente de datos CVS para identificar a los expertos y sus contribuciones. Los investigadores argumentan que con el registro CVS y una buena descripción de alto nivel del árbol del código fuente, es posible identificar automáticamente la experiencia.

Ensemble [16] es un conjunto de herramientas ligeras construidas sobre IBM Rational Team Concert, que analiza el trabajo actual de un desarrollador para recomendar a otras personas que se encuentren trabajando en artefactos relacionados (código fuente, elementos de trabajo y conjunto de cambios). De igual manera, los desarrolladores pueden utilizar esta herramienta para mantener una pequeña lista de vigilancia de otras personas. Se utilizan dos heurísticas para sugerir a los desarrolladores las personas con las que se deben de comunicar: arco reflejo y lazo de nodos. La heurística de arco reflejo se basa en la intuición de que si el desarrollador P_1 trabaja en el artefacto A_1 , y el desarrollador P_2 trabaja en el artefacto A_2 , y A_2 depende de A_1 , entonces P_1 debe comunicarse con P_2 . La heurística lazo de nodos se basa en la intuición de si que dos desarrolladores trabajar en un mismo artefacto, deben comunicarse entre sí.

SmallBlue [18] es una aplicación basada en el concepto de red social que tiene como objetivo analizar las relaciones sociales de la organización para proveer de respuestas a preguntas importantes tales como “¿Quién sabe qué?” así como, “¿Quién sabe quién?”.

Se basa en una pequeña aplicación cliente que localmente analiza la bandeja de salida del correo electrónico y las salidas de las transcripciones de las conversaciones de la mensajería instantánea. A través de esta información el sistema determina la red social e infiere la experiencia de los miembros relacionados.

Begel et al. [25] presenta Hoozizat, una herramienta desarrollada para localizar personas que poseen y son responsables de una funcionalidad, API, producto o servicio, creado como búsqueda en un portal web sobre Codebook.

La herramienta Conscius [48] propone la localización de un experto a través del código fuente, historial de control de cambios, documentación del proyecto y la lista de correos del desarrollador. La herramienta analiza el contenido de cada correo electrónico para identificar el tema y el código fuente relacionado. A partir de la información recolectada se analizan las siguientes relaciones: desarrollador-código fuente, código fuente-código fuente, desarrollador-desarrollador y documentación-código fuente.

IM-an-Expert [49] es un servicio automatizado que recibe preguntas a través de mensajería instantánea (IM, por sus siglas en inglés), localiza y contacta a posibles usuarios que contesten o se interesen en el tema de la pregunta. El grupo de expertos se localiza con base en su disponibilidad actual, esto es, que no se encuentre ocupado, ausente, en una reunión, en una llamada, etc. Esta información se obtiene a través de la información de presencia del cliente de IM. Por el mismo medio, el usuario se pone en contacto para el seguimiento de la pregunta planteada.

Como parte del proceso de desarrollo de software, existe un flujo de información constante a través de diferentes canales que permiten determinar expertos. Actualmente existe una rica exploración de sistemas de recomendación de expertos, siendo las heurísticas más recurrentes las basadas en el historial del control de cambios y las relaciones entre artefactos de un proyecto. Esto debido a que permite diferenciar con facilidad a una persona que ha trabajado por un período breve en un área en particular

de código de alguien que tiene más experiencia, así como localizar a una persona con experiencia en la mayoría del proceso de desarrollo del producto.

Los sistemas de localización de expertos anteriores permiten al usuario, de manera general, introducir uno o más términos de búsqueda proporcionando como resultado una lista de personas cuyos atributos coinciden con los especificados por el usuario. Sin embargo, estudios realizados demuestran que los sistemas de recomendación no soportan bien la tarea de localización de expertos, ya que a menudo resulta difícil de explicar por qué se sugiere una persona en particular, que es clave para que el usuario pueda evaluar dicha recomendación [50,51].

Como se ha notado, los trabajos previamente presentados se apoyan en diferentes técnicas de visualización para la presentación de información. Card, Mackinlay y Shneiderman proponen la visualización de información (InfoVis por sus siglas en inglés) como las representaciones de datos abstractos que son soportadas por computadora, interactivas y visuales, con el objetivo de ampliar el conocimiento [52]. Por ello en el siguiente apartado se describen los elementos y técnicas de visualización utilizadas por estos trabajos.

2.3. Visualización de Información

La conciencia de las actividades de los colegas facilita la coordinación de los esfuerzos de colaboración. Factores como la distancia geográfica entre equipos, las diferencias culturales, y las reglas y normas de una organización, pueden hacer más difícil la tarea de determinar las actividades de los colegas de un desarrollador. En las secciones anteriores se describieron trabajos que ofrecen y aumentan la conciencia de los desarrolladores de proyectos de software. Algunas de estos trabajos se basan en técnicas de visualización como un medio para proporcionar conciencia (ver tabla 2.3).

Tabla 2.3: Técnicas de visualización de información utilizadas por los trabajos de la literatura revisada.

Trabajos	Código de color	Gráficos	Líneas de tiempo	Representaciones gráficas de espacios de trabajo	Imágenes de video	Iconos e imágenes	Detalles al pasar por encima el cursor	Vista de árbol	Vista Jerárquica	Vista de red social	Tamaño de la fuente	Tamaño del ícono o imagen
TeamSpace (2001)				X	X	X						
Rear View Mirror (2002)	X					X	X					
Expertise Browser (2002)	X							X				
Palantir (2003)	X							X	X			
Project Watcher (2004)	X	X					X	X				
Caise (2004)		X						X				
ProjectView-IM (2004)	X							X				
CommunityNet (2005)										X		
Celine (2005)	X					X						
ExpertiseNet (2005)										X		
Constellation (2007)	X									X		
FASTDash (2007)	X					X		X				
Expertise Cloud (2008)											X	
Ensemble (2008)						X			X			
IMPROMPTU (2008)						X						
SmallBlue (2009)						X	X			X		
Tesseract (2009)	X	X	X							X		
WIPDash (2009)	X					X	X	X				X
CVW-IM (2010)	X							X				
Syde (2010)	X					X		X			X	
Saros (2010)	X					X		X				
Codebook (2010)						X				X		
Hooziat (2010)						X						
Trusty (2011)	X					X				X		

La visualización es generalmente aceptada como una buena opción para la transmisión de información de conciencia debido a su capacidad de aprovechar los efectos de percepción que son familiares para los usuarios, tales como los efectos de primer plano y de fondo, así como el uso de color que permite que la información de interés sobresalga. Sin embargo, la visualización debe estar diseñada para soportar los roles particulares de los usuarios y las tareas específicas que necesitan realizar [53].

Enfocándonos sólo en las herramientas que proporcionan específicamente conciencia

de las actividades humanas, presentamos una descripción de las técnicas de visualización utilizadas por algunos de los trabajos presentados anteriormente. Estos incluyen:

TeamSpace [3] es una herramienta que ofrece una interfaz web, la cual presenta un área de control principal organizada por pestañas, lo que permite que los usuarios puedan gestionar sus actividades de proyectos en entornos centrados en el trabajo (ver Figura 2.2a). La pestaña de reuniones (meetings) proporciona acceso a toda la información relacionada con la reunión: las reuniones se pueden crear, editar, programar, revisar y unir a otras. La pestaña de documentos (documents) permite a los usuarios crear, compartir y administrar documentos semi-estructurados, como documentos de texto. La pestaña de tareas (tasks) proporciona acceso a los elementos de acción del usuario, y permite el seguimiento y la asignación de puntos de acción. La pestaña de personas (People) contiene todas las tareas relacionadas con la administración de usuarios.

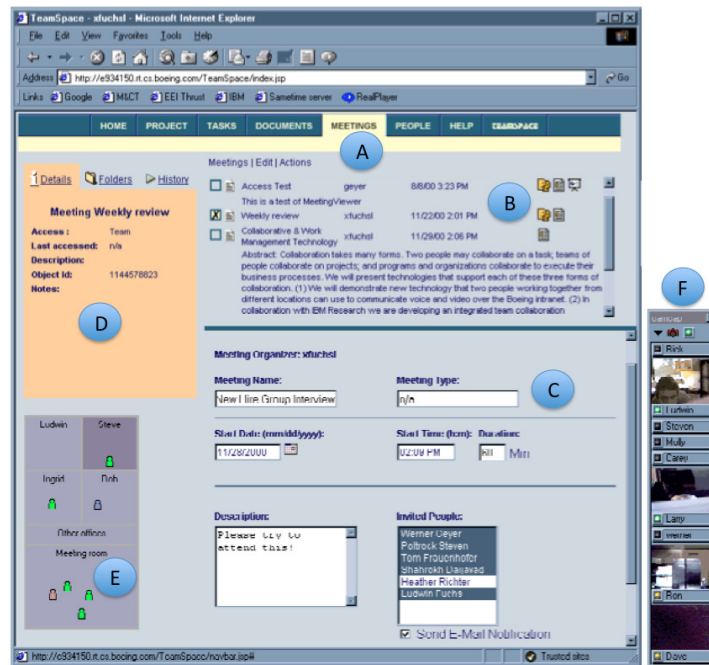


Figura 2.2: Interfaz web del sistema TeamSpace (tomado de [3])

Como parte de la pestaña de reuniones, se presenta un área de visualización general de tamaño variable que se divide entre una vista de lista que se utiliza para buscar y recuperar listas de objetos (ver Figura 2.2b). Una vista de documento que proporciona una visualización y área de entrada de documentos o contenido arbitrario (ver Figura 2.2c).

La información sobre los objetos seleccionados se muestran en la Figura 2.2d. Debajo de esta información se presenta una representación gráfica de las actividades del usuario (ver Figura 2.2e), las cuales se muestran en un mapa de dos dimensiones (o habitación). El mapa de habitación muestra el estado actual de los usuarios (activo, ocupado, inactivo, conectado, desconectado). Además, las habitaciones establecen automáticamente un canal de comunicación de chat y se integran con el comunicador teamspace (ver Figura 2.2f). Cada vez que un usuario entra en una habitación, puede ver las imágenes en directo de otros usuarios que se encuentren en la misma habitación a través de teamspace.

Rear View Mirror [4] es una implementación de un mensajero instantáneo y de presencia. La identidad de otros usuarios se muestra por medio de una imagen en ícono. El estado de presencia de cada persona es indicado por el color del borde de la imagen, y se obtienen más detalles al pasar el cursor por encima de la imagen (ver Figura 2.3a). Los usuarios pueden cambiar su estado de forma explícita a través de la selección de un estado en el menú, o por medio de la configuración del salvapantallas como los tiempos de espera que cambian de estado después de un período configurable de inactividad del mouse y teclado.

El sistema proporciona soporte para grupos similar a las “listas de amigos” que ofrecen otras aplicaciones de mensajería instantánea. Con la diferencia que estas listas se definen por el usuario e incluyen a los miembros de su grupo de trabajo. Por otro lado, los usuarios dispersos geográficamente pueden estar enterados de los acontecimientos en otros sitios antes o después de las horas de trabajo, a través de un chat persistente

que proporciona el sistema (ver Figura 2.3b).

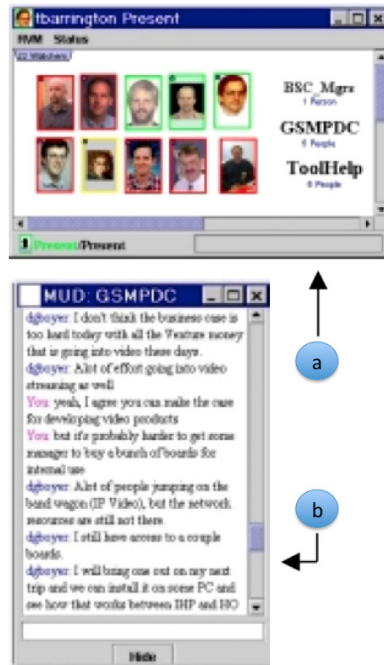


Figura 2.3: Interfaz del sistema Rear View Mirror (tomado de [4])

Expertise Browser [5] propone una interfaz sencilla en la que el usuario pueda solicitar la búsqueda de un experto. La vista de código muestra parte del árbol del código fuente (ver Figura 2.4a). Al expandir el nodo del módulo principal se presentan los archivos individuales. Por otro lado, se tiene acceso a dos vistas organizacionales (reporte de fallos y comentarios de código) las cuales representan dos tipos de experiencia (ver Figura 2.4b).

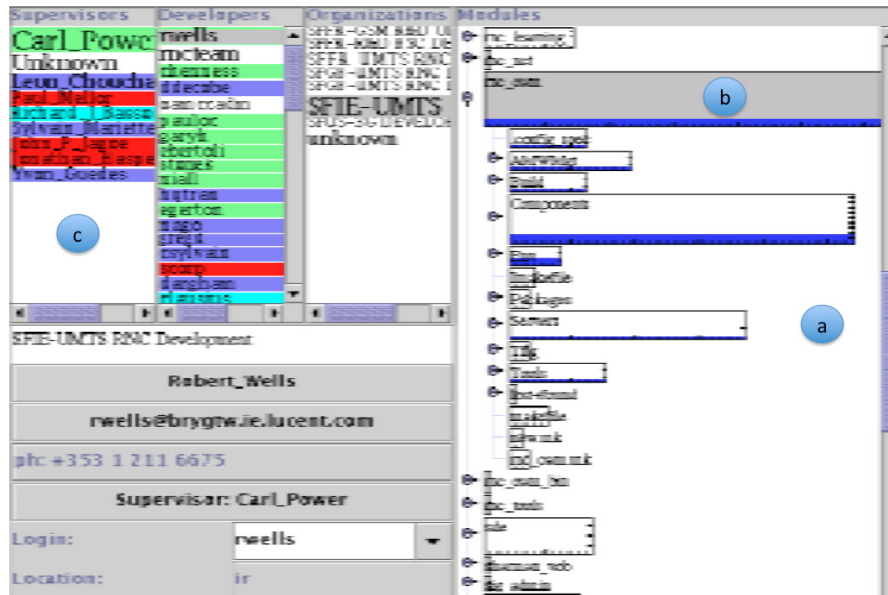


Figura 2.4: Interfaz del sistema Expertise Browser (tomado de [5])

El tamaño vertical de una unidad (una persona o un módulo de código) representa el número de átomos de experiencia (EA) obtenidas por persona o como parte del módulo. El tamaño horizontal representa el número de personas que han contribuido deltas al módulo o archivo. De esta manera, al determinar el listado de las personas expertas se destaca con el color amarillo la de mayor proporción (ver Figura 2.4c).

La herramienta presenta varias vistas de detalle, las cuales proporcionan:

- Listado de detalle de las unidades del producto u organización seleccionados.
- Búsqueda textual y selección de funciones para la identificación del producto deseado o unidades de la organización.
- Medios para contactar a una persona por correo electrónico.

Palantír [6] proporciona conciencia a los equipos de desarrollo de software distribuido de las actividades ocurridas en los diferentes espacios de trabajo. La herramienta

presenta dos visualizaciones, las cuales pueden ser utilizadas en paralelo. La primera consiste en una simple notificación que presenta información relacionada con el componente seleccionado (ver Figura 2.5a).

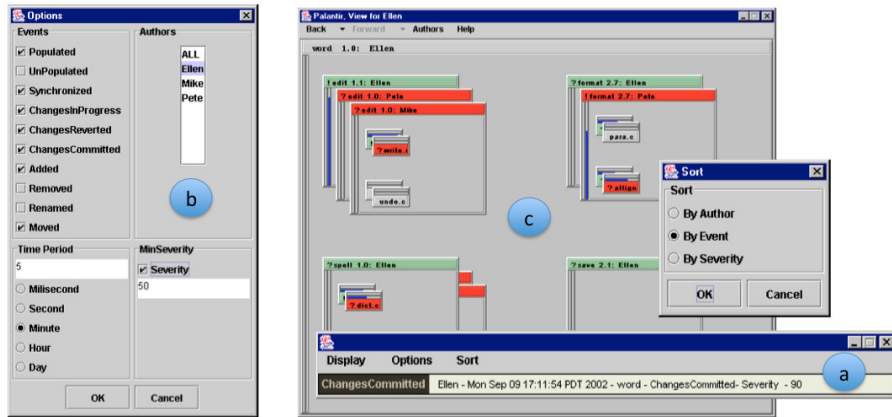


Figura 2.5: Interfaz del sistema Palantír (tomado de [6])

En la segunda visualización el usuario puede configurar los elementos de información que desea visualizar, a través de un panel de configuración (ver Figura 2.5b). De acuerdo con los datos solicitados, se representa la información en la vista de recursos soportada por algunas técnicas de visualización. Con el apoyo de anotaciones de color en los nombres de los archivos se muestra la edición de archivos actuales. La vista de árbol de recursos utiliza barras horizontales para indicar la gravedad de los cambios en curso y comentados. Otra vista es la vista jerárquica que muestra las diferencias por pares de acuerdo con los conflictos entre cada posible par de autores (ver Figura 2.5c).

ProjectWatcher [7] permite visualizar las actividades pasadas y actuales en los artefactos de un proyecto. Cuenta con un complemento de la vista general, donde los artefactos del proyecto se muestran a través de una simple pila que muestra paquetes, archivos, clases y métodos. Los artefactos son siempre apilados por fecha de creación, cada desarrollador tiene asignado un color único, y este color se puede añadir a los bloques de la vista general con base en un conjunto de filtros. Además, se muestra un

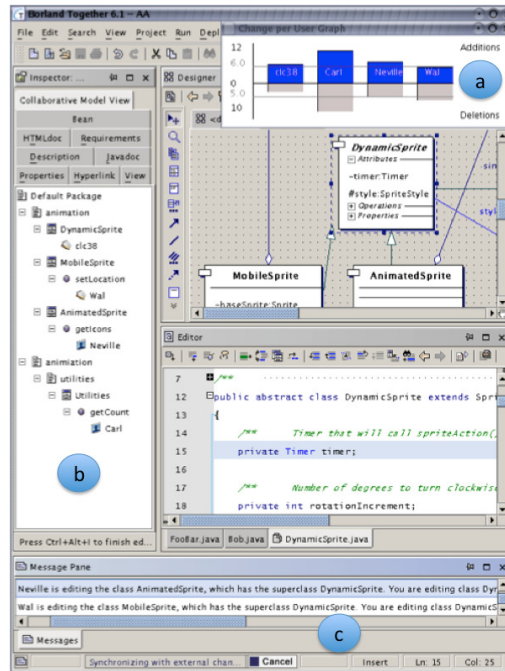


Figura 2.7: Interfaz de los complementos de Caise (tomado de [8])

Se maneja un widget de gestión de proyectos de colaboración llamado gráfico de cambios. Este panel proporciona gráficos que representan actualizaciones en tiempo real de las acciones sobre la semántica del proyecto (ver Figura 2.7a). Así mismo, se muestra un árbol de usuarios que presenta información en tiempo real acerca de la localización de cada usuario activo en el proyecto, y muestra un ícono de visión o edición de acuerdo a su estado (ver Figura 2.7b). Finalmente se presenta un panel de mensajes que muestra mensajes desde un módulo de retroalimentación que analiza las relaciones entre los recursos y desarrolladores de un proyecto (ver Figura 2.7c).

Project View [9] presenta una interfaz similar a la de un mensajero instantáneo tradicional, a través de la que proporciona conciencia global de los colaboradores en línea y disponibles. Así mismo, presenta una lista de todos los proyectos del grupo de trabajo (ver Figura 2.8).

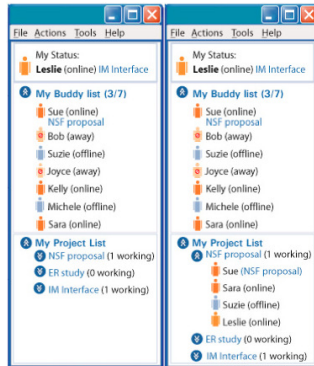


Figura 2.8: Interfaz de Project View (tomado de [9])

La interfaz de la vista de proyectos (Project View) facilita la división de atención para varios proyectos sin la interrupción de la tarea principal. Para ello, se propone un código de colores; que agrupa información relacionada y agrega información sobre el proyecto actual de una persona a la lista de amigos. Además, permite a los usuarios expandir o contraer las listas de proyectos tal como se realiza en la mayoría de los clientes de mensajería instantánea.

CommunityNet [10] presenta una interfaz de un sistema de visualización y consulta. La distancia entre los nodos representa la cercanía (medida por las frecuencias de comunicación) de una persona a la persona en el nodo central. Los usuarios pueden hacer clic en el nodo de ligar con CommunityNet de otra persona. Asimismo, este sistema muestra redes sociales personales, que incluyen los contactos de un usuario durante un período de tiempo determinado (ver Figura 2.9).

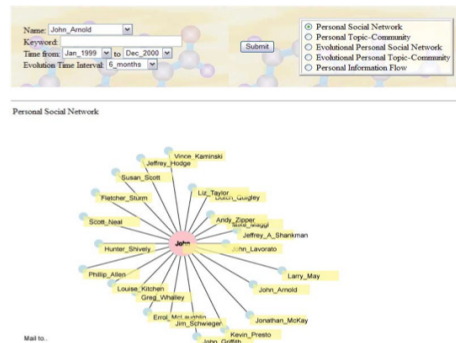


Figura 2.9: Interfaz de CommunityNet (tomado de [10])

Celine [11] proporciona información del estado de cada artefacto del proyecto a través de un único icono, esto aún cuando esté presente en más de un área de trabajo. Para reducir la cantidad de información de conciencia proporcionada, el sistema agrupa los espacios de trabajo de un usuario en una sola fuente de divergencias. La interfaz de Celine proporciona la manera de visualizar con precisión quiénes son los participantes que han realizado modificaciones en un artefacto, así como las diferencias entre los artefactos y la copia local que posee el usuario (ver Figura 2.10).

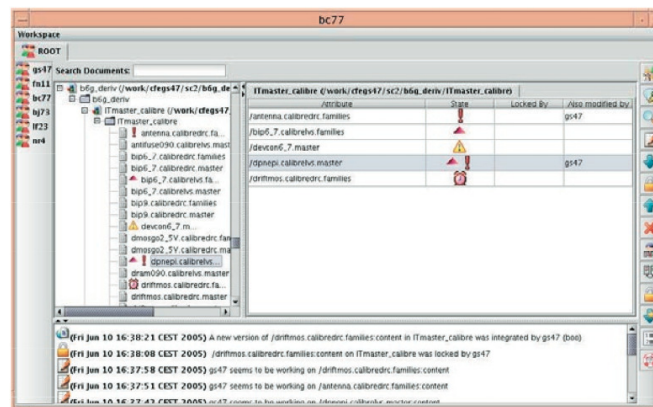


Figura 2.10: Interfaz de Celine (tomado de [11])

ExpertiseNet [12] se compone de un componente de hipertexto y un componente

gráfico. El componente de hipertexto permite al usuario navegar, buscar e inspeccionar información de los expertos. El componente gráfico proporciona las relaciones del experto con determinados tópicos (ver Figura 2.11).

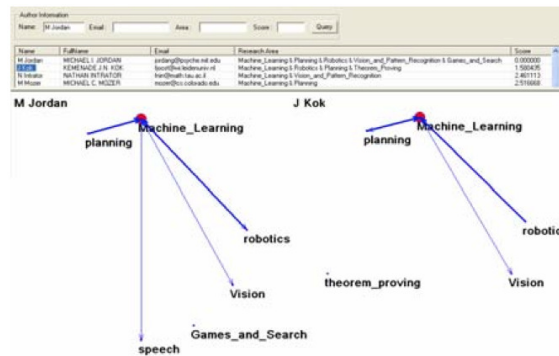


Figura 2.11: Interfaz de ExpertiseNet (tomado de [12])

Constellation [13] es una aplicación que puede ejecutarse como una aplicación de escritorio o como un applet en un navegador web. Esta aplicación soporta la búsqueda por nombre del empleado y palabra clave; muestra a través de un grafo las relaciones de las personas con los trabajos de un grupo de trabajo (ver Figura 2.12).

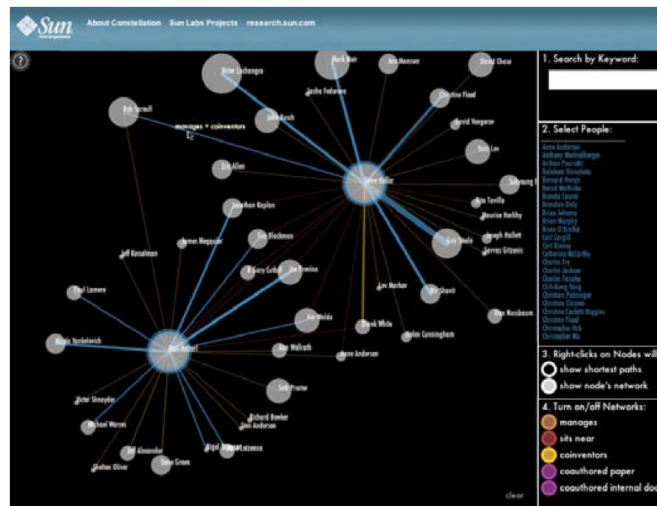


Figura 2.12: Interfaz de Constellation (tomado de [13])

FASTDash [14] proporciona conciencia de las actividades de un grupo de programadores. Como parte de la interfaz del sistema, las carpetas se visualizan como rectángulos anidados, el tamaño de cada uno se basa en el número de archivos que contiene la carpeta. Los archivos son representados en la visualización por una etiqueta textual del nombre del archivo y un ícono. Para poder identificar las acciones sobre los archivos se manejan diferentes indicadores, incluyendo:

- Un borde dorado que indica que un programador tiene un archivo enfocado a Visual Studio (ver Figura 2.13a).
- Un sombreado a rayas indica que el archivo está actualmente siendo revisado por más de un programador (ver Figura 2.13b).
- Un borde rojo indica cuando el programador está enfocado en un archivo y está en modo de depuración en Visual Studio (ver Figura 2.13d).
- Un sombreado en azul indica que un archivo fue abierto por un programador. La foto del programador se muestra como presencia para archivos abiertos (ver Figura 2.13e).
- Un sombreado amarillo indica que un archivo fue abierto y está siendo editado por un programador (ver Figura 2.13f).

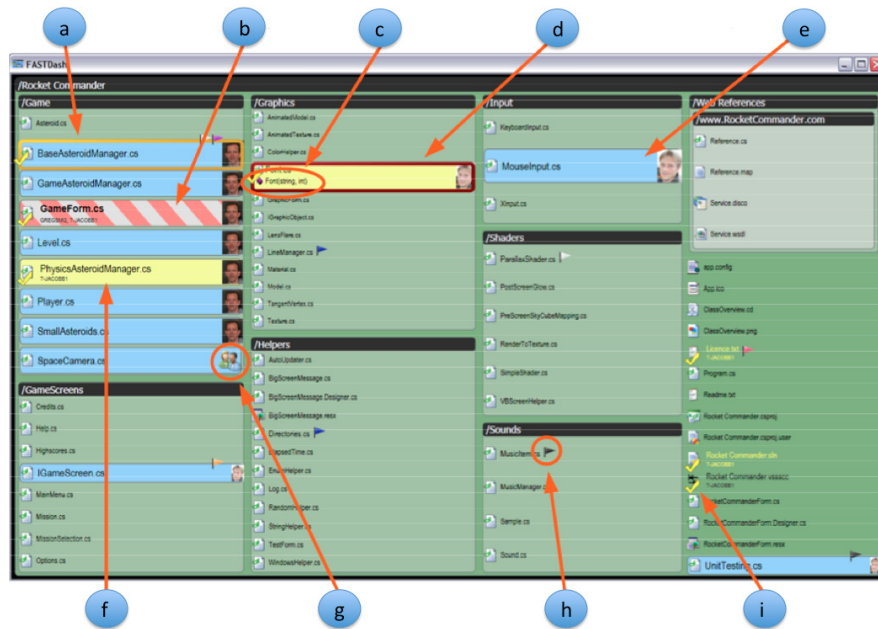


Figura 2.13: Interfaz del sistema FastDash (tomado de [14])

Cuando un archivo se encuentra disponible es posible visualizar información de la clase o método (ver Figura 2.13c). Así como visualizar la imagen del desarrollador o un ícono de múltiples programadores si varios de estos lo están utilizando (ver Figura 2.13g). Otros íconos utilizados son: una bandera que indica un comentario en el archivo (ver Figura 2.13h) y una palomita amarilla que indica que el archivo fue revisado, pero si además el texto está de color amarillo indicará que durante la revisión se realizaron modificaciones (ver Figura 2.13i).

Expertise Cloud [15] propone la idea de una “nube de etiquetas” de sitios web. El sistema presenta una interfaz que resume las actividades de las personas que han realizado comentarios en un archivo de registro. Una nube de etiquetas es una visualización de las etiquetas generadas por los usuarios y que describen el contenido de un sitio web. En este contexto, una etiqueta es una palabra clave relevante o etiqueta de tema que el usuario ha introducido. La importancia de una etiqueta es proporcional al tamaño de

la fuente con la que está escrita. El usuario puede hacer clic en un nombre y la nube actual presenta todas las categorías en las que el usuario ha hecho contribuciones. Estas son las áreas de especialización para una persona en particular (Figura 2.14).

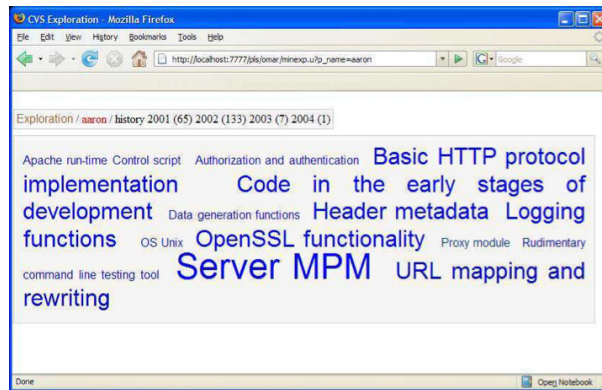


Figura 2.14: Interfaz del sistema Expertise Cloud (tomado de [15])

Ensemble [16] consta de dos herramientas: una de recomendación que analiza el trabajo actual de un desarrollador para recomendar a otras personas; y una lista de vigilancia que apoya a los desarrolladores a determinar los momentos para colaborar (ver Figura 2.15).

El listado de recomendaciones con apoyo de íconos representa la identidad de los desarrolladores. Se muestra el contexto de cada recomendación cuando se expande el triángulo bajo el nombre de la persona recomendada. Este contexto incluye un listado de los artefactos modificados por la persona recomendada y que tiene en común con el usuario.

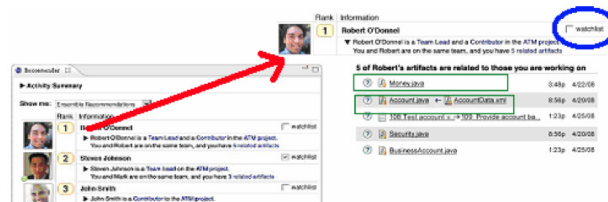


Figura 2.15: Interfaz del sistema Ensemble (tomado de [16])

La interfaz de Impromptu [17] proporciona una representación visual de los miembros del grupo de trabajo y sus actividades. Está disponible para pantallas grandes y aplicaciones individuales a disposición del grupo. Esta última está compuesta por un control de colaboración, barra del colaborador y una barra emergente de pantalla compartida.

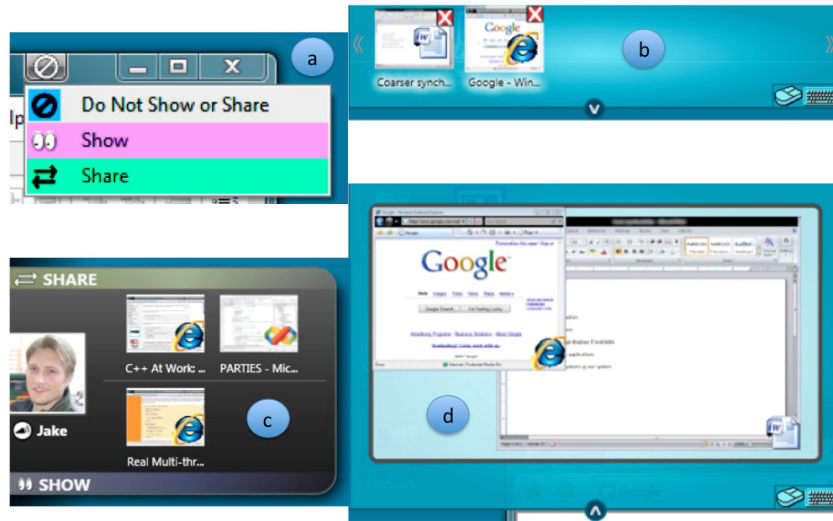


Figura 2.16: Interfaz del sistema Impromptu (tomado de [17])

En el control de colaboración se visualizan las ventanas de aplicación. Así mismo, se utiliza para configurar la disposición de las ventanas para los miembros del grupo, esto es, si sólo pueden ver (Show) o interactuar con ellas (Share) (ver Figura 2.16a). En la barra emergente de la pantalla compartida se presentan como íconos las ventanas que son compartidas. Al pasar por encima presenta un resumen de las ventanas (ver Figura 2.16b). Al seleccionar las flechas se expande la vista, que permite cambiar la posición en la pantalla compartida (ver Figura 2.16d). En la barra del colaborador se representa con apoyo de íconos, cada participante en la sesión de colaboración y las ventanas de aplicación que cada uno de ellos puso a disposición del grupo (ver Figura 2.16c).

La suite SmallBlue [18] apoya a los usuarios a administrar sus redes personales, y

poder generar una red extendida (amigos de los amigos), con el objetivo de localizar y tener acceso a la experiencia e información. Esta suite consiste en cinco componentes: SmallBlue y cuatro aplicaciones de usuario basadas en web. Estas cuatro aplicaciones se describen a continuación.

SmallBlue Ego es una herramienta de gestión personal que crea una visualización de red social personal. En esta visualización se representa la cercanía social del usuario hacia un contacto, a través de la distancia del centro de la red hacia la imagen que representa al contacto. Además, el color del fondo representa las diferentes divisiones de la compañía (ver Figura 2.17a).

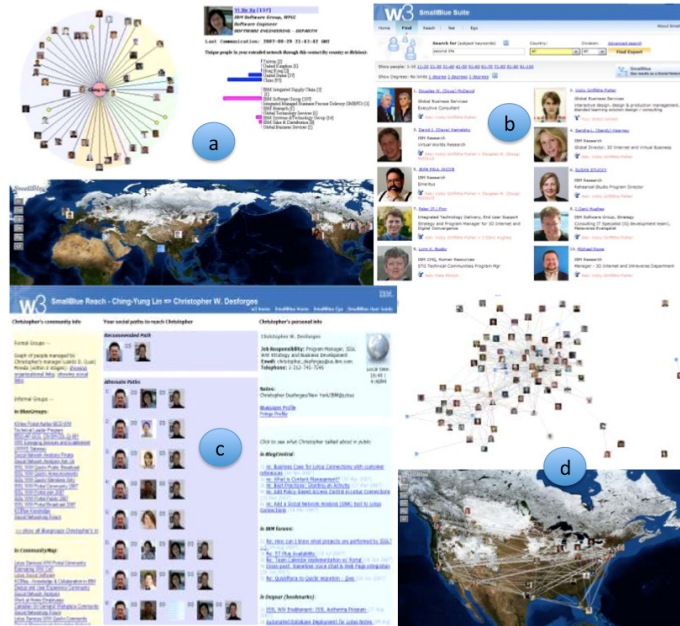


Figura 2.17: Visualizaciones de SmallBlue (tomado de [18])

El motor de búsqueda llamado SmallBlue Find, proporciona una lista de personas que coinciden con la cadena de búsqueda y palabras claves. En la lista se presentan imágenes de las personas destacadas, además del rol, la función y el estado de la conexión (ver Figura 2.17b). Al hacer clic en la imagen o nombre de una persona se redirecciona

a SmallBlue Find que presenta información actualizada de esta persona.

El motor de análisis de red llamado SmallBlue Reach muestra las rutas más cortas para llegar a una persona, se presentan hasta 6 grados. Cada una de estas rutas se representa con íconos de imágenes de las personas relacionadas. Asimismo, presenta los grupos formales e informales de la organización a los cuales pertenece una persona y sus actividades públicas en blogs, foros, perfiles, etc. (ver Figura 2.17c).

SmallBlue Net presenta y analiza la red social de los expertos asociados a un tema, esto a través de una visualización de red social. Al pasar el cursor sobre la imagen del contacto presenta la información del rol y la unidad de negocio a la que pertenece. El sistema proporciona varias formas de visualización de la red social de expertos asociados a un tema, la primera a través de una visualización de red social con divisiones en diferentes colores que representan cada división de la compañía. Además de la tradicional vista de agrupación de redes, también se proporciona una vista de mapa que muestra la ubicación de las personas y la manera en la que están conectadas (ver Figura 2.17d).

Tesseract [19] analiza diferentes recursos del proyecto con la finalidad de determinar las relaciones socio-técnicas de un proyecto. A través de cuatro visualizaciones se proporciona una visión de estas relaciones, un par de estas visualizaciones consisten en redes que representan las relaciones que existe entre los archivos del proyecto (ver Figura 2.18b) y las relaciones de comunicación entre los desarrolladores (ver Figura 2.18c).

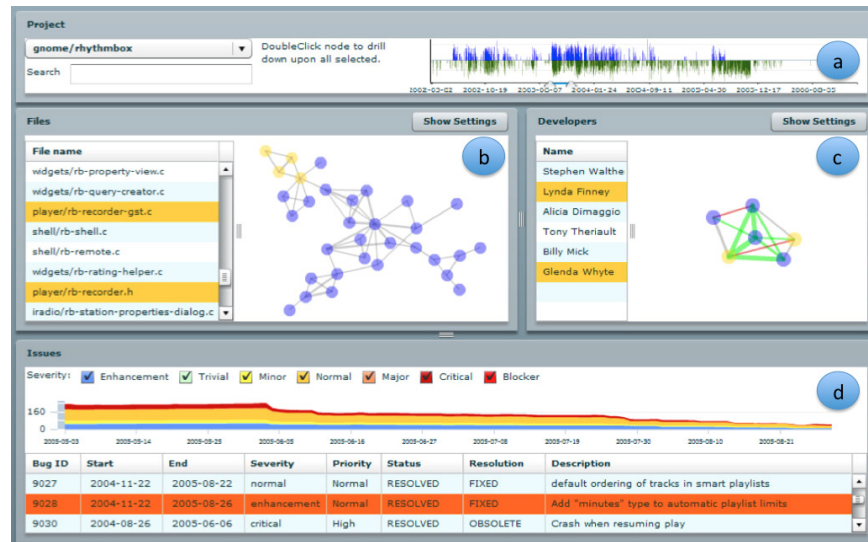


Figura 2.18: Interfaz del sistema Tesseract (tomado de [19])

Un panel de actividad del proyecto con comentarios de código, con apoyo de un gráfico de barras permite una visión general de la actividad del proyecto (ver Figura 2.18a). Asimismo, otro panel presenta con apoyo de un código de colores los temas recurrentes durante el desempeño de las tareas de desarrollo (ver Figura 2.18d).

WIPDash [20] comparte una visualización general para ser vista por los miembros del equipo, así mismo tienen acceso a una visualización individual donde pueden estar intercalando las diferentes vistas, filtros y solicitar detalles a demanda. La ventana de WIPDash consiste en dos secciones: la primera contiene una representación espacial de las áreas del proyecto y sus elementos de trabajo (ver Figura 2.19a), mientras que la segunda consiste de una lista de modos de vistas, un panel de equipo y listas desplegables de iteraciones, elementos de trabajo y tipos (ver Figura 2.19b).

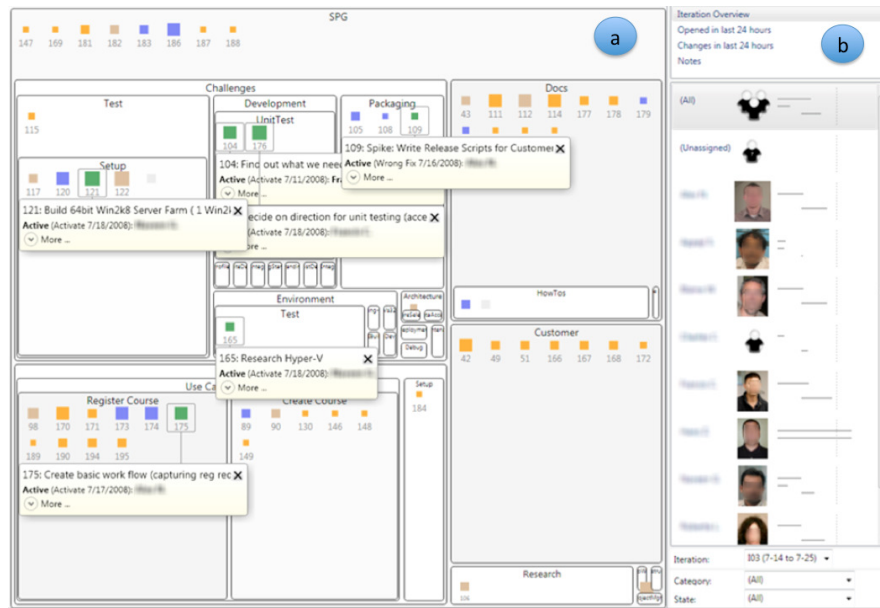


Figura 2.19: Interfaz del sistema WIPDash (tomado de [20])

El sistema utiliza un mapa de árbol (tree map) encuadrado para la representación de las áreas del proyecto y cada uno de estos se muestra en forma de rectángulos. Cada uno de los rectángulos es de tamaño proporcional al número de elementos de trabajo abiertos en el área, es de un color más oscuro equivalente a los elementos de trabajo abiertos asociados a la zona y se etiquetan con el nombre del área.

Cada rectángulo en el mapa de árbol contiene íconos que representan los elementos de trabajo asociados con la zona. Los iconos de un rectángulo se colocan en el orden en que fueron creados a partir de la esquina superior izquierda. El color de un ícono indica el estado del elemento de trabajo (por ejemplo, propuesto, activo, resuelto o cerrado) y la forma de un icono indica el tipo de elemento de trabajo (por ejemplo, función, error o tarea). El tamaño del icono representa ya sea el nivel de prioridad o las horas estimadas del elemento de trabajo, con iconos de tamaño más grandes que representan los elementos de mayor prioridad o mayor estimación de las horas de trabajo, según lo

señalado por el usuario o el equipo.

Al mover el cursor del ratón sobre el icono de un elemento de trabajo muestra una descripción de los detalles del artículo. Al hacer clic en un icono muestra una ventana emergente con información sobre el elemento de trabajo. Un pequeño símbolo amarillo de nota adhesiva se muestra en el elemento de trabajo para indicar que tiene una nota adjunta.

El panel del equipo contiene los nombres y fotos de los miembros del equipo. Al hacer clic en un miembro del equipo se muestran los iconos de todos los elementos de trabajo asignados o cerrado por el miembro del equipo seleccionado. Para cada persona, WIPDash muestra líneas horizontales que representan la cantidad total de trabajo, la cantidad de trabajo realizado, y el trabajo que queda en la iteración que se asigna a esa persona.

El mensajero instantáneo CWS-IM (Collaborative Working Sphere – Instant Messaging, por sus siglas en inglés) [21] al igual que Project View muestra información de forma tradicional de un mensajero instantáneo, presentando una lista de posibles colaboradores y sus tareas asignadas en un proyecto en particular (ver Figura 2.20).

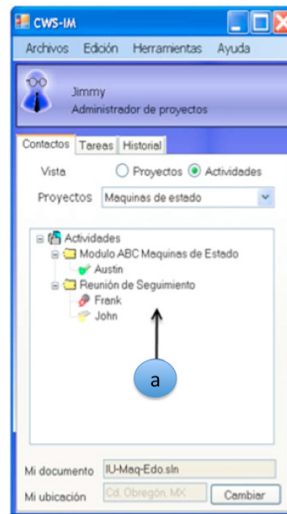


Figura 2.20: Interfaz del sistema CWS-IM (tomado de [21])

El estado de un usuario se representa a través del uso de un código de colores para denotar Disponibilidad Selectiva: si el estado del usuario se presenta en verde, esto significa que los usuarios (emisor y receptor posible) están trabajando en el mismo proyecto y en la misma actividad (Disponible), si la situación se presenta en amarillo, lo que significa que los usuarios están trabajando en el mismo proyecto, pero en diferentes actividades (alcanzable, pero ocupado), y si la situación se presenta en rojo, esto significa que se está trabajando en diferentes proyectos (Ocupado) (ver Figura 2.20a).

Syde [22] tiene una extensión para una vista al cliente llamada Scamp que permite a los desarrolladores obtener información en tiempo real sobre quién cambio qué y cuándo. Para mostrar esta información Scamp utiliza tres visualizaciones: una vista de nube de palabras, una vista de contenedores y un explorador de paquetes (ver Figura 2.21).

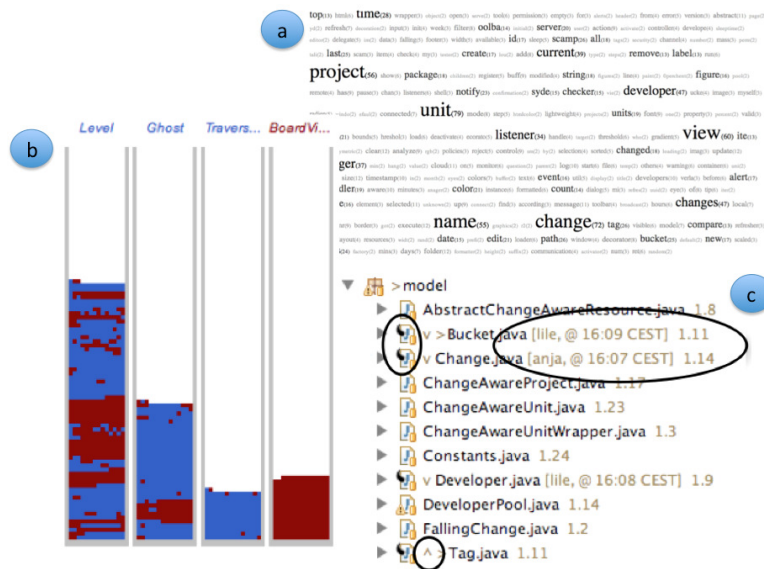


Figura 2.21: Visualizaciones de Scamp (tomado de [22])

La nube de palabras consiste de una lista de palabras donde se asigna un tamaño de acuerdo a la frecuencia de las palabras en el código fuente y un color relacionado al desarrollador que realizó el cambio más reciente en la clase en cuestión. Al hacer clic en una palabra llevará al usuario al código fuente (ver Figura 2.21a).

La vista de contenedores que se van llenando progresivamente con los cambios individuales representados como pequeños cuadrados. El color de cada cambio representa al desarrollador responsable de ello. Dichos cambios muestran un orden cronológico, por lo tanto los cambios en el fondo del contenedor son más antiguos, mientras que los más recientes aparecen en la parte superior. La ventana de tiempo puede ser ajustada a demanda del desarrollador y los patrones de llenado de cada contenedor reflejan el esfuerzo que cada desarrollador ha invertido en él (ver Figura 2.21b). Scamp ofrece una decoración en forma de pequeñas anotaciones en el explorador de paquetes en Eclipse, donde los archivos del proyecto se muestran. Si un desarrollador que está utilizando Syde y Scamp cambia un archivo, su representación en el explorador de paquetes se puede

representar de tres maneras: un icono de superposición, una flecha, y una anotación textual (ver Figura 2.21c).

Saros [23] representa varios elementos para crear conciencia de las actividades de los desarrolladores a través del uso de colores. Un rastro de color permite rastrear las contribuciones de los autores individuales (ver Figura 2.22a) y las últimas veinte ediciones de texto se muestran en cada editor (ver Figura 2.22b). Como parte de la barra de desplazamiento se muestra una barra de color que indica la posición del usuario dentro del archivo (ver Figura 2.22c).

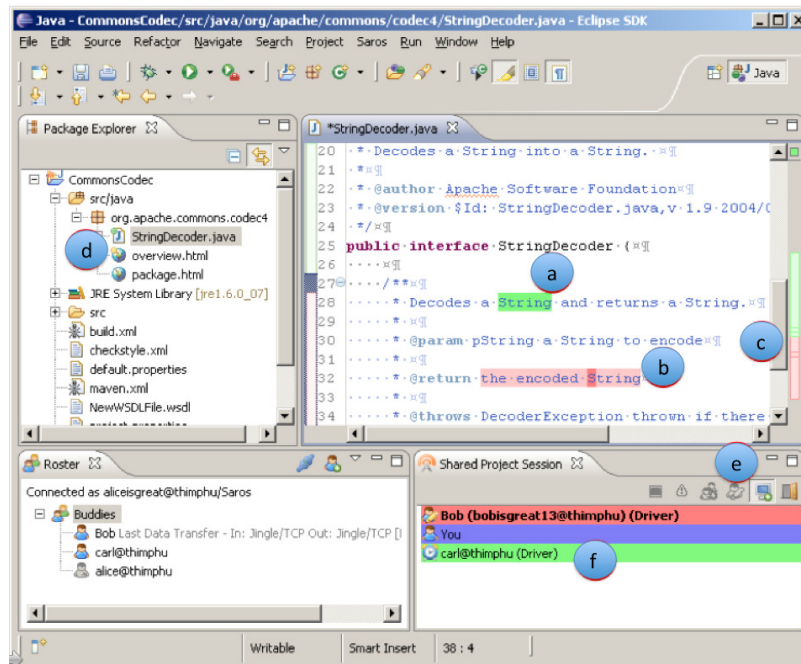


Figura 2.22: Interfaz del sistema Saros (tomado de [23])

A través de una vista de árbol se muestran los archivos abiertos o vistos actualmente por los usuarios (ver Figura 2.22d). La herramienta permite mantener conciencia de las actividades de un determinado usuario, combinando la información de conciencia de los archivos y entre los archivos (ver Figura 2.22e). También se presenta información

acerca de las aplicaciones actuales en uso por otros desarrolladores utilizando listados con uso de colores e íconos (ver Figura 2.22f).

CodeBook [24] presenta un grafo canónico, que muestra las posibles relaciones entre los objetos de CodeBook. Las etiquetas de los bordes llamadas *menciones* están escritas en letra cursiva para indicar que se puede derivar de una inferencia incorrecta. Las rutas en negrita ayudan a explicar un escenario (ver Figura 2.23).

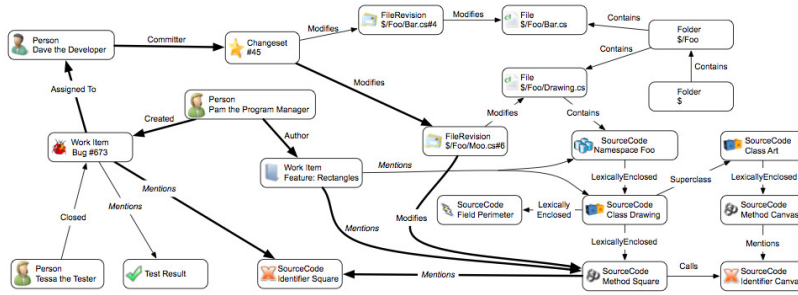


Figura 2.23: Visualización de CodeBook (tomado de [24])

Hoozizat [25] es una herramienta que presenta un área de control principal organizada por pestañas, de izquierda a derecha: personas, elementos de trabajo, código fuente (parcialmente oculto) y archivos (no se muestran). En la pestaña de información general se presenta un listado como resultado de la identificación de expertos. Dicho listado se agrupa de acuerdo al rol que desempeña, así como un ícono con la imagen de la persona identificada. Al situar el cursor sobre las imágenes, se muestra un cuadro de texto con información de contacto de esa persona. Además, cerca de la imagen se resaltan todas las apariciones de esta persona en el resultado de la búsqueda, para que el usuario pueda identificar a otras personas u objetos relacionados a la persona (ver Figura 2.24).

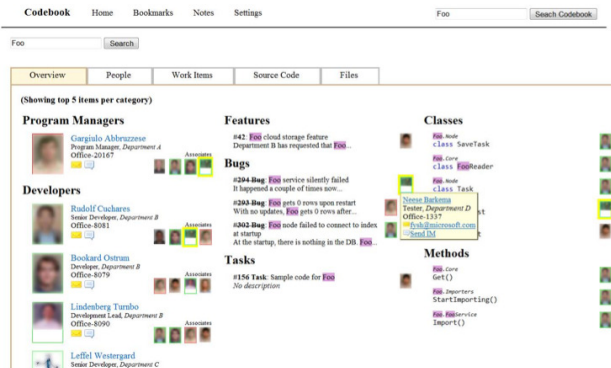


Figura 2.24: Visualización de Hoozizat (tomado de [25])

Trusty [26] representa el estado del usuario con un código de colores de manera similar a la de CWS-IM. Este código de color también es guiado por criterios de disponibilidad selectiva. Otra aplicación de colores la hace en el marco de la foto del desarrollador para indicar si es o no el momento adecuado para comenzar una interacción sincrónica con la otra persona. Existen cinco posibles colores (azul, verde, amarillo, naranja y rojo). El ajuste se basa en la situación actual y el tiempo con respecto a las horas en que la persona prefiere ser contactado (ver Figura 2.25a).

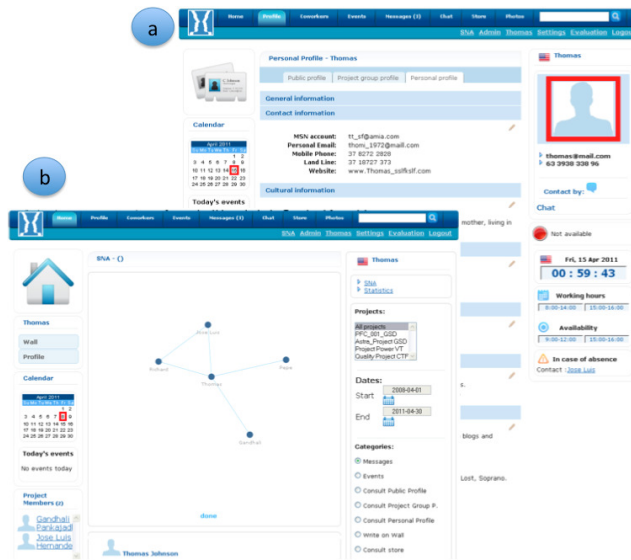


Figura 2.25: Interfaces de Trusty (tomado de [26])

Recurriendo al uso de representaciones de redes sociales, en Trusty se muestra información en la que los nodos representan a los usuarios de confianza. La herramienta puede ser utilizada para analizar diversos aspectos de la interacción en la red social, incluyendo el tráfico de mensajes, publicación de eventos, uso de la pared, las visitas de perfil, y la utilización de repositorio de conocimiento (ver Figura 2.25b).

Las herramientas aquí descritas para proporcionar conciencia de elementos de colaboración, así como la localización de expertos están diseñadas para presentar diferentes elementos de información de conciencia. Esto nos ofrece una serie de técnicas de visualización interesantes para presentar información a trabajadores de DSD.

2.4. Ideas de diseño

Después de haber analizado la literatura se han encontrado una serie de elementos que pueden ser útiles para el desarrollo de un modelo para la identificación de colaboradores potenciales en DSD.

Para ello, los trabajos citados aportan las siguientes ideas:

- La conciencia de cambios en artefactos (p.ej. código) resulta de interés para los desarrolladores. Debido a que estos consideran a las personas que realizan acciones sobre el código como personas con el conocimiento necesario para poder apoyar en tareas relacionadas. Esto surge de las secciones 2.1 y 2.2.
- Las relaciones entre los desarrolladores y los distintos artefactos involucrados en el proyecto resultan útiles al momento de decidir a quién contactar en situaciones de ayuda. Esto surge de las secciones 2.1 y 2.2.
- Los sistemas de recomendación deben de mostrar de manera transparente el por qué una recomendación en particular es adecuada para ellos. Esto debido a que los usuarios se pueden llegar a sentir frustrados al tratar de entender por qué fue recomendada una persona en particular. Esto surge de la sección 2.2.
- La forma de visualización para proporcionar conciencia a trabajadores de DSD deberá determinarse de acuerdo al tipo de elementos de información, asegurándose de que los usuarios puedan interpretarlos correctamente. Esto se deriva de la sección 2.3.

Como se puede apreciar, los trabajos descritos en este capítulo permiten entre otras cosas coordinar actividades a través de proporcionar diferentes elementos de información, desempeñando un papel importante en la colaboración en DSD. Estas herramientas están diseñadas para diferentes propósitos, no exclusivamente para la identificación

de colaboradores potenciales, presentando diferentes elementos de información. Es por ello que con estas ideas de diseño, se sustenta el diseño de un modelo para identificar colaboradores potenciales en situaciones de ayuda en DSD y el cual se aborda en el siguiente capítulo. Los elementos de conciencia de este modelo llevarán a selecciones más rápidas y mejor informadas de los trabajadores de DSD.

2.5. Resumen del capítulo

En este capítulo se han descrito una serie de trabajos relacionados con las áreas de sistemas de conciencia, sistemas de localización de expertos y visualización de información, de los cuales se han obtenido ideas de diseño para guiar una propuesta para la identificación de colaboradores potenciales en situaciones de ayuda en DSD. Como parte de los resultados de este análisis se obtuvo lo siguiente:

- Se identificaron elementos de conciencia que se han propuesto para proporcionar conciencia en entornos de desarrollo de software distribuido.
- Se identificaron diferentes heurísticas que se han utilizado para la localización de expertos.
- Se identificaron diferentes técnicas de visualización de información empleadas para proporcionar conciencia a los trabajadores de DSD.
- Se enlistaron una serie de ideas de diseño para la identificación de colaboradores potenciales en DSD.

En el siguiente capítulo se presenta un modelo para identificación de colaboradores potenciales en situaciones de ayuda en DSD, el cual fue definido con base en las ideas de diseño obtenidas de este capítulo.

Capítulo 3

Involucramiento, Experiencia y Disponibilidad: apoyo a inicio de interacciones de ayuda en desarrollo distribuido de software.

En los capítulos anteriores se presentó un análisis de trabajos de investigación que aportan elementos para los inicios de colaboración en el desarrollo distribuido de software (DSD, por sus siglas en inglés). Con base a esto, se establecieron un conjunto de ideas de diseño que motivan la necesidad de contar con un modelo de identificación de colaboradores potenciales, que apoye a los desarrolladores en DSD para decidir a quién contactar en el momento en que tienen que enfrentar algún problema o requieren de información que les impide continuar desempeñando su trabajo (ver sección 2.4). Por ello, en este capítulo se presenta una caracterización de las situaciones en las que un ingeniero de software tiene necesidad de identificar a un colaborador potencial. Posteriormente se presenta y describe un modelo para la identificación de colaboradores potenciales creado a partir del análisis de la literatura de trabajo colaborativo y localización de

expertos.

3.1. Situaciones en las que se requiere identificar colaboradores potenciales.

Los escenarios suelen ser relatos que cuentan una historia en la que se describe una o más tareas de un proceso específico. Con el desarrollo de escenarios se suelen identificar aspectos importantes que afectan en el mundo real, por lo que son útiles a lo largo del proceso de diseño, especialmente al desarrollar descripciones de tareas para la determinación de requerimientos y decisiones de diseño [54].

Para entender en qué momentos y bajo qué condiciones se presentan situaciones de ayuda (p. ej. resolver un problema que impide continuar con la tarea asignada, ser nuevo miembro en el proyecto y desconocer aspectos importantes del desarrollo, una duda técnica de la plataforma de desarrollo), se analizaron trabajos referentes a la coordinación de actividades de DSD [27, 55–60], identificación de fuentes de conocimiento en DSD [61–63] y recomendación de expertos [51, 64]. Con esto, de manera particular se detectaron y definieron escenarios que sirvieron de apoyo durante el entendimiento de las situaciones de ayuda que se presentan a trabajadores del DSD. A partir de estos escenarios, fue posible identificar los momentos en los que, los desarrolladores requieren identificar colaboradores potenciales entre los miembros de una organización.

Para generar este entendimiento, se definieron cinco escenarios con personajes ficticios y con base en criterios identificados en la literatura para la identificación de colaboradores potenciales. A continuación se describe cada uno de ellos:

Áreas de experiencia: Marcos Fuentes de 29 años de edad, egresó de la carrera de Ingeniero en Computación del Tecnológico de Monterrey (ITESM) hace cinco

años, y desde hace tres años trabaja para DSI en Querétaro. Actualmente desempeña el rol de programador senior en el proyecto que dio inicio esta semana en la compañía manufacturera.

El día de ayer fue el primer día de reuniones para el proyecto. Esta es la primera vez que Marcos colabora con este grupo de trabajo, por lo que, la reunión inicial de “rompe hielo” fue muy útil, ya que pudo conocer un poco respecto al perfil y experiencia previa de cada uno de sus colegas.

Durante la asignación de tareas del plan de trabajo del proyecto el día de ayer, a Marcos le fue designada la codificación de la capa de datos, la cual es parte del módulo de validación de datos del proveedor y se estableció que esta capa debe desarrollarse en la plataforma javascript.

El día de hoy Marcos inició el trabajo con el proyecto. Después de codificar un bloque de código, Marcos lo ejecuta para verificarlo, obteniendo como resultado una serie de errores en la lectura de la base de datos. Marcos realiza unos ajustes, sin embargo no logra resolver todos los errores, lo cuál le genera una gran confusión, debido a que no sabe si la implementación de la lectura de la base de datos está siendo ejecutada correctamente. Este incidente resulta en un problema, ya que le impide avanzar en el cumplimiento de la tarea asignada.

Para poder completar la actividad, Marcos requiere solicitar apoyo de algún colega que le ayude a identificar cómo resolver los errores de conexión a la base de datos.

Disponibilidad inmediata: Cristian Robles de 32 años de edad, egresó de la carrera de Ingeniero en Sistemas Computacionales de la Universidad de Arizona (Estados Unidos) hace siete años y desde hace dos años, trabaja para DSI en Arizona (Estados Unidos). Actualmente desempeña el rol de Arquitecto de Software en el

proyecto de la compañía manufacturera.

Esta mañana el Administrador de proyecto, que está al mando del equipo de trabajo de Cristian, recibe una llamada del Administrador de proyecto del equipo en Querétaro (México); el cual, le solicita su apoyo de manera urgente para atender unos errores detectados en la interfaz del módulo de envío de facturación de proveedores.

Dada la planeación de las tareas, el equipo de trabajo en Querétaro solicita dichas modificaciones con un tiempo de respuesta de 24 horas o menos de ser posible. Algunos de estos errores le han sido asignados a Cristian, quien al revisar el código se enfrenta a unas consultas a la base de datos muy complejas y con un tiempo de ejecución alto. Cristian debe de resolverlo en el tiempo establecido, y para poder cumplir con el tiempo de respuesta debe solicitar apoyo de algún colega.

Ubicación geográfica: Javier Soto de 35 años de edad, egresó de la carrera de Licenciado en Informática de la Universidad Autónoma de Querétaro (México) hace 10 años y desde hace cuatro años trabaja para DSI en Querétaro. Actualmente desempeña el rol de Programador junior en el proyecto de la compañía manufacturera. Javier ya tiene experiencia colaborando con algunos de los integrantes del equipo de trabajo de Sonora debido a su participación en proyectos anteriores, por lo que conoce muy bien su experiencia y forma de trabajar.

En esta ocasión a Javier se le asignaron algunas actividades de apoyo, en la generación de reportes de cuentas por pagar del módulo de proveedores. Para realizar esta tarea se le ha especificado, que la generación de reportes deberá ser con la aplicación de Crystal Reports, además de cumplir con los estándares de formatos utilizados por la compañía manufacturera.

Javier ejecuta la aplicación para verificar si lo que codificó en la aplicación para

mandar llamar al reporte funciona bien. Sin embargo, no logra visualizar el reporte, por lo que realiza unos ajustes y vuelve a ejecutar. Sin embargo, no obtiene ningún resultado, lo que le empieza a generar algo de frustración. Para poder cumplir con su tarea, Javier requiere solicitar apoyo de algún colega a su alcance que le apoye a entender el por qué no puede visualizar el reporte.

Involucramiento en el proyecto: Erika Sotomayor de 27 años de edad, egresó de la carrera de Ingeniero en Sistemas Computacionales de la Universidad Autónoma de San Luis Potosí (México) hace cinco años, y desde hace dos años y medio trabaja para DSI en Querétaro (México). Erika actualmente desempeña el rol de Control de Calidad en el proyecto de la compañía manufacturera.

Como parte de sus actividades asignadas, Erika le da seguimiento a la actividad de pruebas al módulo de validación de los archivos de la facturación electrónica. Estos archivos deben cumplir ciertos lineamientos establecidos de acuerdo a la Secretaría de Hacienda y especificadas en el diseño por los Arquitectos de Software en Arizona (Estados Unidos).

Al realizar las pruebas correspondientes, Erika detectó varios errores y aspectos que incumplían el diseño original y que al consultar la documentación, no se indicaban los motivos, generándole confusión al respecto. Erika requiere solicitar el apoyo de algún colega que le pueda explicar si los cambios realizados fueron solicitados o si deberá de proceder a notificar los errores encontrados.

Contribución en el proyecto: Sandra Figueroa de 33 años de edad, egresó de la carrera de Ingeniero en Computación de la Universidad Autónoma de Sonora (México) hace once años y desde hace cuatro años trabaja para DSI en Querétaro. Sandra actualmente desempeña el rol de programador senior en el proyecto de la

compañía manufacturera.

A Sandra le fueron asignadas tareas y tiempos para el proyecto por el Administrador del mismo. Actualmente, ha cumplido con dos entregables y se encuentra esperando la liberación de ambos por parte de los de Control de Calidad en Arizona (Estados Unidos).

Esa tarde, Sandra recibió una notificación a través del sistema de administración de proyectos, respecto a algunos errores identificados; además se le señalaron las fechas límite en las cuales debían de estar resueltos los mismos. De los errores identificados, uno está relacionado con el archivo de validación de facturación. Sin embargo, en éste Sandra solamente colaboró en la codificación de un submódulo de envío a cuentas por pagar.

Como detalle del error, se indica que se trata de problemas con la conexión del submódulo de envío de cuentas por pagar con el submódulo de validación de facturación, lo cual corresponde a la primera fase de codificación del artefacto `blayer_connection.js`, en el cual estuvieron trabajando integrantes del equipo de trabajo de Querétaro. Para que Sandra pueda realizar las modificaciones pertinentes requiere el apoyo de un colega que le indique si las modificaciones que requiere, no afectan al resto de las conexiones codificadas en el artefacto.

Como se puede notar en los escenarios, durante el desempeño de las actividades de desarrollo de software se presentan situaciones en las que se requiere de interacciones de ayuda. Los cambios que se realizan frecuentemente en los procesos, la falta de comunicación y ausencia de integración de herramientas colaborativas pueden generar que los trabajadores dispersos no se percaten de aspectos importantes del desarrollo, debido a la falta de conciencia de las actividades de los colegas. Además de traer como consecuencia el no poder localizar a la persona indicada, que les permita trabajar en colaboración de manera eficiente, evitando rediseños, retrabajos y replanificaciones.

Los desarrolladores pasan entre el 50 % y 70 % de su tiempo dependiendo de la fase del ciclo de desarrollo de software, comunicándose con otros [65]. Por lo tanto, conocer con quién comunicarse puede ayudar a incrementar la efectividad de un desarrollador. En la siguiente sección se presenta un modelo para la identificación de colaboradores potenciales basado en la literatura.

3.2. Modelo para identificación de colaboradores potenciales para inicio de interacciones de ayuda.

Basados en la caracterización de las situaciones en las que un ingeniero de software tiene necesidad de identificar a un colaborador potencial previamente descritas, la literatura de sistemas de conciencia y sistemas de localización de expertos (ver secciones 2.1 y 2.2), se identificaron un conjunto de ideas de diseño para la identificación de colaboradores potenciales en DSD (ver sección 2.4). Los elementos de conciencia identificados se conjuntaron en un modelo de métricas, que pueda aportar a un ingeniero de software conciencia de a quién contactar cuando enfrenta diferentes situaciones de ayuda.

La colaboración es parte fundamental en la construcción y evolución de un producto de software a gran escala. Los ingenieros de software suelen pasar el 80 % de su tiempo buscando fuentes de conocimiento en las diferentes fases del desarrollo [66]. En entornos distribuidos la localización de personas con los conocimientos necesarios resulta más complicada, donde debido a la distancia geográfica, a diferencias culturales y de tiempo, y a la falta de interacciones cara a cara se crean barreras de comunicación [55–57].

Las recientes investigaciones han propuesto herramientas que utilizan un método específico para la identificación de expertos y presentan esta información mediante un listado de expertos. Los métodos se centran en evaluar las relaciones entre el desarrollador y los artefactos del proyecto. Además de que estos métodos se aplican en

contextos específicos y no proporcionan manera de cambiar o ampliar los mecanismos de clasificación, ni fuentes de información. Por ello, cuando los desarrolladores requieran identificar expertos en diferentes contextos con diferentes necesidades, tendrán que utilizar herramientas adecuadas para esta tarea.

Con base en lo anterior, presentamos un modelo de métricas para la identificación de colaboradores potenciales, mediante la conciencia del Involucramiento, Experiencia y Disponibilidad de los colegas. Este modelo tiene como objetivo, proporcionar al desarrollador varios elementos de conciencia que apoyen a su toma de decisión respecto a quién contactar en diferentes situaciones de ayuda (ver Figura 3.1).



Figura 3.1: Elementos de información y métricas que contribuyen a responder la pregunta ¿Quiénes son las personas que podrían tener la información para resolver el problema?.

Los tres conceptos del modelo (Involucramiento, Experiencia y Disponibilidad) pretenden aumentar la conciencia acerca de las habilidades, experiencias y conocimiento entre miembros de una organización. Esto nos permite crear el conocimiento compartido de la tarea y del equipo, que permita a los ingenieros de software alcanzar la familiari-

dad entre los miembros de una organización.

En las siguientes subsecciones se describen a detalle los tres principales conceptos (p. ej. Involucramiento, Experiencia y Disponibilidad) y las métricas que los componen. De esta forma, se podrá tener una visión más clara de cómo estos elementos de conciencia apoyan en la identificación de colaboradores potenciales.

3.2.1. Involucramiento.

Tomando en cuenta el análisis de la literatura, se propone la siguiente definición para este concepto:

“Es la proporción de tiempo, talento y conocimiento con la que un desarrollador de software contribuye para la creación de un producto.”

Está dado por el conocimiento (p. ej. sobre el dominio del sistema, del proceso de desarrollo que se está siguiendo para la construcción del mismo, del diseño y la implementación del sistema entre otros aspectos) que se adquiere al desempeñar las actividades del ciclo de desarrollo de software. Sin embargo, el trabajo y la experiencia de un trabajador difieren de los de otros trabajadores del mismo proyecto, ya que cada uno obtiene un conjunto único de conocimientos [63]. Por naturaleza, los seres humanos adquirimos conocimiento cada vez que interactuamos con diversos objetos, esto es, a través de la auto-implicación y el compromiso [67].

Para poder determinar el nivel de involucramiento de un desarrollador, se propone un conjunto de métricas, las cuales son descritas a continuación:

- *Porcentaje de participación (tiempo) en el proyecto.* Se define como el porcentaje de tiempo que invierte un desarrollador en realizar las tareas asignadas de determinado proyecto. La necesidad de contactar a los propietarios de un proceso no resulta un problema para los desarrolladores que han estado colaborando en el proyecto durante un tiempo considerable, conociendo también en que partes del código intervienen los desarrolladores. Sin embargo, los desarrolladores que recientemente se unieron a un proyecto se enfrentan a la situación contraria [62].
- *Porcentaje de participación (artefactos) en el proyecto.* Se define como el porcentaje de artefactos que un desarrollador ha generado o editado como parte de su participación en un proyecto. Esta métrica se basa en la experiencia de un desarrollador y su desempeño en general, los cuales pueden ser inferidos a través del conjunto de cambios y la autoría de código fuente. Un desarrollador puede ser percibido como el experto en un área de código base, si su código es utilizado por otros desarrolladores [68].
- *Porcentaje de contribución en el artefacto.* Se define como el porcentaje en que un desarrollador ha generado o editado el artefacto relacionado con la duda o problema actual. Esta métrica se aborda desde la perspectiva de que los trabajadores de desarrollo de software, consideran que si un desarrollador realiza repetidamente revisiones en un proceso en específico, es muy probable que se trate de un experto en dicho proceso [38,69]. Por lo tanto, si se requiere de apoyo en este proceso se podrá conocer a quién contactar a través del análisis del historial de control de cambios.

La información de las métricas presentadas nos permite ser conscientes del esfuerzo de un desarrollador en un determinado artefacto, la frecuencia con la que ha trabajado en un proyecto, el interés en un artefacto en particular. Con esto, se podrá determinar un perfil técnico, que contribuirá como información de entrada para la creación de un

perfil general de colaboradores potenciales. De esta manera, podremos presentar un listado de personas que sean capaces de apoyar en diferentes situaciones de ayuda. Así mismo, otros elementos de información aportan conciencia para este perfil general, en la siguiente subsección se describen algunos de ellos.

3.2.2. Experiencia.

El adquirir conocimiento de un determinado artefacto, a través de la actividad de un desarrollador con el contenido de los artefactos de un proyecto, coincide con cómo los seres humanos aprendemos y adquirimos conocimiento. Sin embargo, esta información sólo es suficiente para crear un perfil con base en la capacidad para responder a preguntas generales o proporcionar soporte técnico [5]. Por ello, se propone el concepto de Experiencia para ampliar el perfil general de un colaborador potencial. De esta manera, estos podrán ser identificados con base en la capacidad de apoyar en situaciones de ayuda de un tema en específico.

Tomando en cuenta el análisis de la literatura, se propone la siguiente definición para este concepto:

“Es la capacidad de un desarrollador para desempeñar tareas específicas.”

El comunicar lo que se desea puede resultar difícil en el entorno de DSD, debido a que cada desarrollador se encuentra inmerso en sus actividades asignadas que no logra darse cuenta de la cantidad de conocimiento que se da por hecho, pero que debería de hacerse explícito. Herbsleb et al. [56] consideran que a través de examinar los curriculums de los desarrolladores que serán asignados al proyecto, se puede identificar a los desarrolladores con conocimiento del dominio y experiencia que harán que resulte mucho más fácil transmitir lo que se desea.

Para poder determinar la experiencia de un desarrollador, se propone un conjunto

de métricas, las cuales son descritas a continuación:

- *Confiabilidad.* Se define como el porcentaje en que un desarrollador ha cumplido con los tiempos de entrega definidos. La confianza es un aspecto importante para colaboraciones exitosas y productivas, además de que se define en términos de la expectativa de una persona a otra. La falta de información acerca de los colegas impide la construcción de lazos de confianza entre los miembros de los equipos distribuidos y por lo tanto, invertir más tiempo en comparar el trabajo de otros [70, 71].

Dada la dependencia entre tareas de un proyecto, si una persona tiende a esperar hasta el último día para entregar el código fuente y del cuál dependen otras tareas, resultará difícil que otro desarrollador confíe en sus habilidades [68]. En DSD, las colaboraciones implican la necesidad de trabajar con personas de diferentes culturas, por lo que es necesario que exista confianza para que los desarrolladores sientan que se pueden expresar [71].

- *Calidad.* Se define como el porcentaje en que el trabajo realizado hasta el momento por el desarrollador ha sido entregado con el menor número de errores. Al inicio de un proyecto resulta complicado estimar las habilidades de cada trabajador y los riesgos que puede implicar esto en el desarrollo. Para ello resulta importante poner a prueba, a largo plazo, la experiencia y habilidades del trabajador en el desarrollo de software. Así mismo, resulta importante también comparar el trabajo previo para poder reducir un poco los problemas de calidad [61].

Esta métrica se basa en la suposición de que cada vez que una persona edita un código fuente lo hace sin errores, y que por lo tanto construirá lazos de confianza con los demás desarrolladores [68].

- Áreas de experiencia. Se define como el conjunto de áreas de desarrollo de software (administración, calidad, programación, análisis y diseño) en las que el desarrollador tiene conocimiento. La información sobre las habilidades de los desarrolladores es considerada útil en este contexto para identificar quién es un experto en un tema en particular, lo cual puede ser aprovechado para tratar de evitar “reinventar la rueda”, situación que ocurre frecuentemente en el desarrollo de software [26].
- Rol. Se define como el rol que el desarrollador se encuentra desempeñando actualmente. La conciencia de la estructura del grupo se relaciona con la experiencia de los trabajadores, con base en las funciones y responsabilidades que asumen. La conciencia del rol es importante, al momento de decidir con quién iniciar una interacción durante el desempeño de las actividades del proyecto. Sobre cada rol se impone una referencia del nivel de conocimiento y las capacidades del trabajador que lo desempeña. Por ejemplo considere el caso de un arquitecto de proyecto, que es un trabajador con amplio conocimiento acerca del producto y de las personas, lo que lo convierte en una fuente de conocimiento recurrente [5, 37].

Dentro de las organizaciones de desarrollo de software, se manejan diferentes roles y responsabilidades. Sin embargo, existen roles claves (administrador del proyecto, arquitectos de software, líder técnico, líder de proyecto, etc.) que facilitan la comunicación entre sitios distribuidos y que pueden representar la adaptación efectiva de la comunicación o pueden significar un cuello de botella importante [69, 72].

Con la información de estas métricas se representan los intereses y experiencia de un desarrollador, lo cual representa una fuente más concreta para determinar su capacidad para poder atender problemas o dudas de un tema en particular. Con esto, se podrá determinar un perfil en temas en particular, que al igual que el perfil técnico del Involucramiento contribuirá en la creación de un perfil general de colaboradores potenciales. Finalmente, elementos de información acerca de la disponibilidad de los posibles

contactos completan este perfil general de colaboradores potenciales. Dichas métricas se describen en la siguiente subsección.

3.2.3. Disponibilidad.

Como se ha mencionado anteriormente, uno de los factores a los que se asocia el retraso debido a la interdependencia de las actividades entre los sitios de trabajo en DSD es la carencia de proximidad física. Para lidiar con esto, se requiere de una comunicación efectiva, identificando a los colaboradores potenciales, y facilitando el contacto a través de un medio de comunicación informal, que permita obtener conciencia de la disponibilidad de los posibles contactos [72].

Tomando en cuenta el análisis de la literatura, se propone la siguiente definición para este concepto:

“Es la información actual de un desarrollador, en términos de ubicación geográfica y disponibilidad.”

La disponibilidad resulta importante para los desarrolladores dispersos geográficamente, ya que deben coordinar sus actividades con otros. Sin embargo, debido a las diferencias horarias resulta complicado encontrar el momento ideal para contactarse [26,70]. Los trabajadores de desarrollo de software co-localizados pueden iniciar interacciones fácilmente, debido a que conocen, o pueden obtener información del entorno y la disponibilidad de los colegas. Por ejemplo, si la oficina de algún colega se encuentra a unos cuantos metros de distancia, resultará fácil de contactar a dicho colega. En cambio, para los trabajadores ubicados en diferentes zonas geográficas resulta mucho más complicado el conocer la disponibilidad de alguien. Para lo cual, por ejemplo, las llamadas sin respuesta podrían significar el estado actual de un colega [56].

Para poder determinar la disponibilidad de un desarrollador, se propone un conjunto de métricas, las cuales son descritas a continuación:

- *Ubicación geográfica.* Se define como la ubicación actual del desarrollador. Esta métrica se considera de relevancia para el DSD, puesto que la conciencia de esta información puede prevenir problemas derivados de las diferencias horarias (p.ej. iniciar la colaboración en horarios no adecuados para los colegas) [72].
- *Disponibilidad actual.* Se define como el estado actual del desarrollador, el cual puede ser ocupado o disponible. La disponibilidad resulta importante para los desarrolladores dispersos geográficamente que deben coordinar sus actividades con otros, pero que debido a la distancia y a las diferencias horarias resulta complicado encontrar el momento ideal para contactarse [26, 70].

Con los elementos del modelo para la identificación de colaboradores potenciales definidos, es posible determinar las fuentes de información necesarias para obtener la información para cada métrica propuesta.

3.3. Proceso del modelo propuesto.

Los conceptos de Involucramiento, Experiencia y Disponibilidad, conjuntan una serie de métricas que son posibles de determinar a partir de datos recabados durante el proceso de desarrollo. En este apartado se describe el proceso del modelo propuesto para identificar a un grupo de colaboradores potenciales que proporcione ayuda a trabajadores en DSD en situaciones de ayuda.

3.3.1. Flujo de trabajo del modelo.

El conjunto de métricas del Involucramiento y de la Experiencia se obtiene a partir de la información que resulta del desempeño de las actividades de desarrollo de software y que se coordina a través de la gestión de proyectos. Mientras que la Experiencia, dependerá de la información recabada por las organizaciones, a partir de los curriculums de los trabajadores. Finalmente, la disponibilidad se obtiene por medio de mecanismos de comunicación de las organizaciones. Con base en esta información, el trabajador podrá decidir a quién acudir para resolver su duda o problema (ver Figura 3.2).

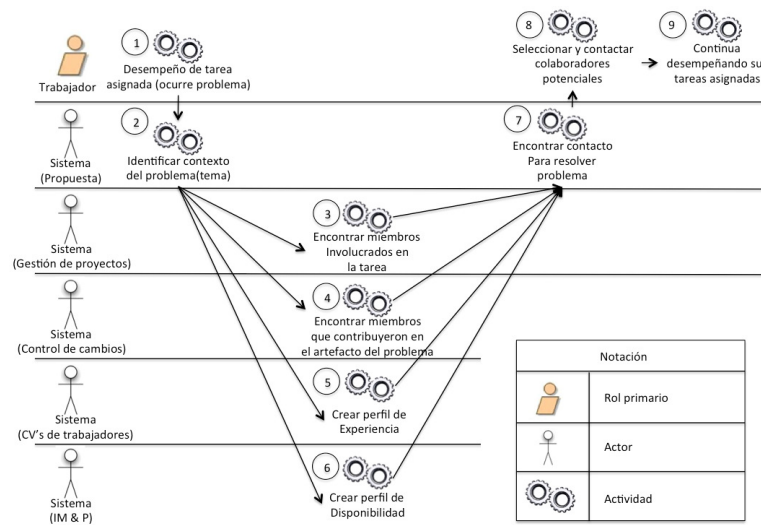


Figura 3.2: Flujo de trabajo para identificar colaboradores potenciales que apoyen a trabajadores en DSD en la resolución de un problema o duda.

En primer lugar, en un momento dado mientras el trabajador realiza sus tareas asignadas, se le presenta un problema que para obtener ayuda requiere que se comunique con un tercero (ver Figura 3.21). Al no conocer a quién contactar, el trabajador activa la vista de colaboradores potenciales en su ambiente de trabajo, a través del cual envía una petición de “ayuda para la identificación de colaboradores potenciales”.

El sistema propuesto recolecta información relacionada al contexto del problema

(p. ej. trabajador, tarea, artefacto – Figura 3.22) y envía solicitud a otros sistemas para trabajar en colaboración con ellos e inician procesos para obtener la siguiente información: i) el sistema de administración de proyecto localiza los trabajadores que se encuentran trabajando en la tarea asociada al problema (ver Figura 3.23), ii) el sistema de gestión de cambios identifica a los trabajadores que han contribuido en la creación, edición y modificación del artefacto asociado con el problema, en conjunto con el anterior determina el Involucramiento de cada miembro (ver Figura 3.24), iii) el sistema de administración de curriculums de los miembros de la organización, contribuye en la creación de un perfil de Experiencia de cada miembro (ver Figura 3.25), y iv) los sistemas de mensajería instantánea y presencia (IM & P) localizan a los miembros del proyecto que actualmente se encuentran disponibles y al alcance del miembro que requiere de ayuda, determinando el perfil de disponibilidad de cada miembro (ver Figura 3.26).

El sistema propuesto crea una lista de colaboradores potenciales que tienen información relevante para resolver el problema (ver Figura 3.27), con base en la información de Involucramiento, Experiencia y Disponibilidad de los miembros del proyecto. Finalmente, se proporciona esta información al trabajador (ver Figura 3.28), para que pueda seleccionar y contactar al colaborador potencial que le apoye a resolver su problema y poder continuar desempeñando sus actividades asignadas (ver Figura 3.29).

3.3.2. Identificar a colaboradores potenciales por su Involucramiento.

Para poder calcular las métricas que determinan el Involucramiento de un trabajador, se requiere de algunos elementos de información de entrada. En la figura 3.3 podemos ver que la información de entrada consiste en una lista actualizada de los trabajadores que han participado en el proyecto, una lista de las actividades asignadas

a cada trabajador, el reporte de las actividades de cada trabajador, una lista de artefactos relacionados a la tarea, el historial de cambios realizados al artefacto relacionado al problema y el contexto el problema.

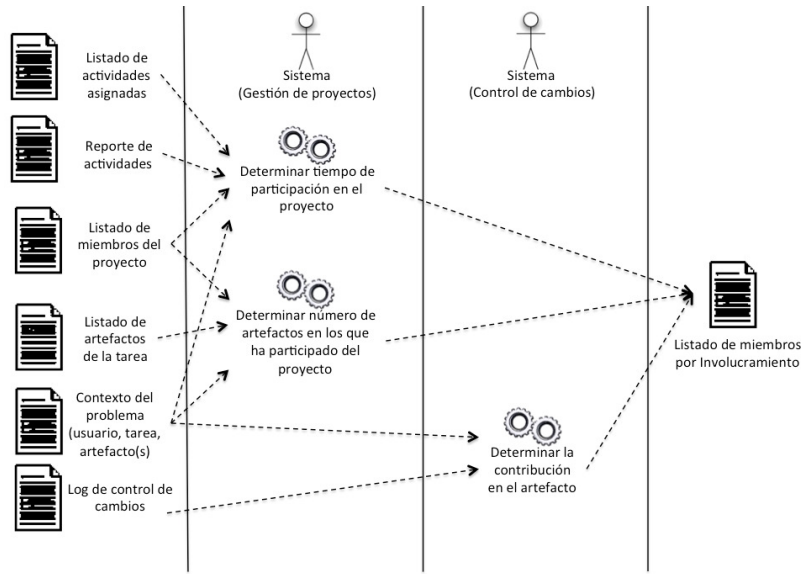


Figura 3.3: Entradas y salidas para determinar colaboradores potenciales por Involucramiento.

Para determinar el porcentaje de participación (tiempo) en el proyecto se propone determinar mediante la siguiente operación:

$$\%T_P = (Tiempofinal_tarea - Tiempoinicial_tarea)(100)/Tiempototal_proyecto \quad (3.1)$$

A partir del tiempo en que el trabajador inició su participación en el proyecto, se calcula la diferencia con el tiempo en que finalizó la misma y se calcula la proporción con base al tiempo total que tomó el desarrollo del proyecto.

Por otro lado, el porcentaje de participación (artefactos) en el proyecto se propone determinar a partir de la siguiente operación:

$$\%A_P = (\#Artefactos_asignados)(100)/Total_artefactos_proyecto \quad (3.2)$$

Con base en el número de artefactos que le fueron asignados a cada trabajador, se calcula la proporción con respecto al total de artefactos que pertenecen al proyecto.

Finalmente, el porcentaje de contribución en el artefacto, se propone determinar mediante de la siguiente operación:

$$\%A_C = (TamañoInicial - TamañoFinal)(100)/TamañoArchivo_actual \quad (3.3)$$

Tomando como referencia el tamaño inicial del artefacto del contexto del problema, se calcula la diferencia con el tamaño final del mismo y se calcula la proporción con base al tamaño actual del artefacto.

Como resultado de este subproceso de Involucramiento, obtenemos un listado de colaboradores potenciales involucrados en el contexto del problema.

3.3.3. Identificar a colaboradores potenciales por su Experiencia.

Para poder calcular las métricas que determinan la Experiencia de un trabajador, se requiere de algunos elementos de información de entrada. En la figura 3.4 podemos ver que la información de entrada consiste en una lista actualizada de las actividades asignadas a cada trabajador, el reporte de las actividades de cada trabajador, una lista

de artefactos relacionados a la tarea, los curriculums de los trabajadores y el contexto del problema.

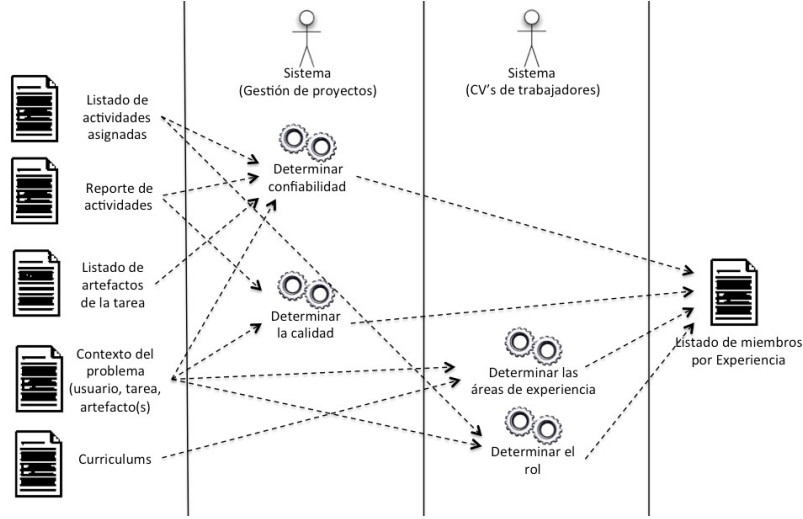


Figura 3.4: Entradas y salidas para determinar colaboradores potenciales por Experiencia.

Para determinar la confiabilidad del trabajo de los trabajadores, se propone la siguiente operación:

$$\%R_b = (TiempoEntrega_real)(100)/TiempoEntrega_definido \quad (3.4)$$

Con base tiempo real en que el trabajador liberó el entregable de la tarea relacionada, se calcula la proporción con respecto al tiempo de entrega que se determinó inicialmente.

La calidad del trabajo del trabajador se propone determinar con base a una escala de tres niveles (Alto-A, Medio-B o Bajo-C), la cual será dada por el líder de proyecto. En cuanto a las áreas de experiencia, estas podrán ser identificadas a través del análisis de la información del curriculum de cada uno de los trabajadores. Finalmente el rol de

cada trabajador se determinará de acuerdo a las actividades que se encuentra desempeñando en el contexto del problema.

Como resultado de este subproceso de Experiencia, obtenemos un listado de colaboradores potenciales de acuerdo a su experiencia en los tópicos relacionados al contexto del problema.

3.3.4. Identificar a colaboradores potenciales por su Disponibilidad.

Para poder calcular las métricas que determinan la disponibilidad de un trabajador, se requiere de algunos elementos de información de entrada. En la figura 3.5 podemos ver que la información de entrada consiste en una lista actualizada de los trabajadores que están participado en el proyecto, una lista de trabajadores que se encuentran actualmente disponibles y un listado de los trabajadores que se encuentran al alcance (ubicación geográfica) del trabajador que requiere ayuda.

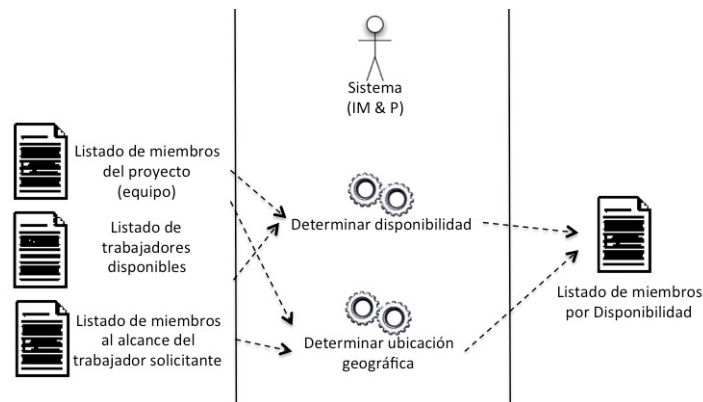


Figura 3.5: Entradas y salidas para determinar colaboradores potenciales por Disponibilidad.

Por medio de los mecanismos de mensajería y presencia de la organización, es posible

obtener el estado actual y ubicación geográfica de los miembros del proyecto [72]. Con esto, se obtiene un listado de miembros ordenados por su disponibilidad y de acuerdo al alcance con respecto al trabajador solicitante.

Con los elementos y fuentes de información identificados será posible obtener como resultado final un conjunto de colaboradores potenciales, que puedan apoyar en la situación de ayuda definida en el contexto del problema. Sin embargo, cada una de estas fuentes de información deben de determinarse acorde a la actividad actual del colaborador actual, para poder tener inicios de colaboración en momentos más adecuados. Por ello, en el siguiente apartado se propone incluir el concepto de espacio compartido.

3.3.5. Identificar colaboradores potenciales mediante un Espacio compartido.

Basados en la caracterización de las actividades de desarrollo de software, Palacio [72] determina un espacio compartido a partir de conjuntar un grupo de trabajo virtual (ver Figura 3.6a), esferas de trabajo colaborativas (ver Figura 3.6b) y repositorio de proyectos (ver Figura 3.6c).

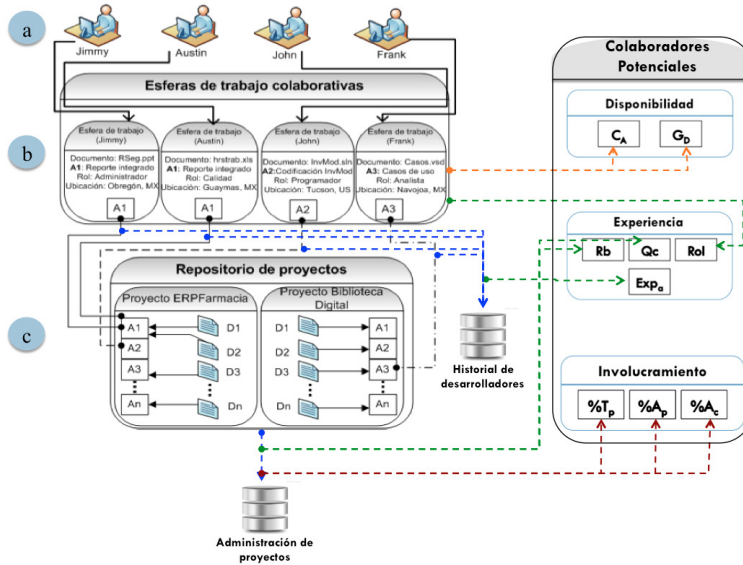


Figura 3.6: Métricas que contribuyen a la identificación de colaboradores potenciales obtenidas a partir de un espacio compartido.

El *grupo de trabajo virtual* se compone de un grupo de desarrolladores que trabajan en diferentes ubicaciones geográficas. Los miembros del grupo centran sus esfuerzos en llevar a cabo una determinada actividad con un conjunto de recursos asociados y colaboradores en un proyecto en particular. Cada conjunto representa una Esfera de Trabajo [73].

Las *esferas de trabajo colaborativas* son el conjunto de esferas de trabajo que son categorizados para identificar quiénes trabajan en la misma actividad/mismo proyecto, en diferente actividad/mismo proyecto y en diferente actividad/diferente proyecto.

El *repositorio de proyectos* contiene información referente a las actividades de los proyectos, en los que la organización está comprometida. Las actividades tienen asociados documentos.

De acuerdo a los elementos de información que componen estas tres partes del es-

pacio compartido, es posible determinar las métricas definidas anteriormente. En la Figura 3.6 se identifican mediante líneas punteadas a partir de que elemento del espacio compartido obtiene cada una de las métricas que componen al Involucramiento, Experiencia y Disponibilidad.

El conocimiento de la actividad de los miembros de un grupo de trabajo es conjunto en *esferas de trabajo colaborativas* [72]. Este modelo provee a los trabajadores de desarrollo de software información suficiente y pertinente no sólo de la situación actual (individual o grupal) sino también de las situaciones que suceden a su alrededor (periférica). Soporta el concepto de colaboración potencial [39], para lo que incluye un conjunto de mecanismos mediante los cuales es posible compartir información resultante de una unidad de trabajo. Además, vincula el trabajo realizado de manera individual mediante la gestión personal de actividades, para lo cual utiliza los artefactos digitales o recursos resultantes del trabajo individual. Estos mecanismos que representan los diferentes niveles de conciencia, son acorde a la actividad actual del colaborador potencial.

3.4. Resumen del capítulo.

En este capítulo se ha proporcionado la definición del modelo propuesto para la identificación de colaboradores potenciales. Así mismo, se definieron cada una de las métricas propuestas del modelo. Dicho modelo surgió a partir del análisis de la literatura e ideas de diseño identificadas en el capítulo anterior. Los principales resultados de este capítulo incluyen:

- Se caracterizaron las situaciones de ayuda que se presentan en el desarrollo de software con base en la literatura.

- Se definió un modelo para la identificación de colaboradores potenciales en situaciones de ayuda.
- Se describieron las métricas de los tres conceptos que componen el modelo propuesto.
- Se identificó la información y mecanismos a partir de los cuales se definen las métricas del Involucramiento, Experiencia y Disponibilidad.
- Se describió como mediante la asociación con el modelo de espacio compartido es posible definir las métricas del modelo propuesto.

En el siguiente capítulo se describen los detalles de diseño de dos propuestas de sistemas interactivos de visualización información. A través de los cuales, se presentarán las métricas del modelo propuesto a trabajadores de DSD.

Capítulo 4

Sistema Interactivo de Visualización de Información.

En el capítulo anterior se describió un modelo para la identificación de colaboradores potenciales en situaciones de ayuda en el desarrollo distribuido de software (DSD, por sus siglas en inglés). Con base en esto, en este capítulo se ha diseñado un prototipo de sistema interactivo de visualización de información, que apoye a los trabajadores de DSD a tomar la decisión de a quién contactar para resolver situaciones de ayuda.

La comunidad de ingeniería de software ha puesto atención en investigar los procesos de gestión y flujo de trabajo con el objetivo de planear las actividades de coordinación [37]. El desarrollo de software es un proceso intensivo en información, por lo que información de conciencia adicional puede distraer a los desarrolladores. Sin embargo, poco ha sido el interés en desarrollar herramientas más flexibles que no saturen el espacio de trabajo con información con diferentes grados de granularidad, diferentes representaciones visuales e interactividad [74, 75].

La visualización de información, es un medio a través del cual se puede dar sentido a los datos. Algunas propiedades visuales, tales como: la posición, el tamaño, la forma

y el color son aprovechadas por los diseñadores de visualizaciones para apoyar a los usuarios a interpretar los datos que se les presentan. De esta manera, resulta un apoyo en los procesos de búsqueda de manera eficiente [76, 77].

Por otro lado, una sobrecarga de información trae consigo la posibilidad de perderse en los datos. Esto nos lleva a que estos datos resulten en información irrelevante para la tarea relacionada, que sean malinterpretados o que sean presentados de manera inapropiada [75, 77].

Por lo anterior, en este capítulo se describe de manera detallada el diseño y desarrollo de sistemas de visualización de información para identificar a colaboradores potenciales a través de los conceptos de Involucramiento, Experiencia y Disponibilidad. Además, se describe un conjunto de mecanismos de cómo la información puede ser obtenida mediante espacios compartidos [72] y cómo esta información es presentada al usuario final por medio de un sistema interactivo de visualización de información.

4.1. Visualizaciones.

En este apartado, se definen las técnicas de visualización utilizadas para presentar las métricas que conforman el modelo propuesto en el Capítulo 3. Para que las métricas de los conceptos: Involucramiento, Experiencia y Disponibilidad, permitan ofrecer soporte adecuado para la identificación de colaboradores potenciales en situaciones de ayuda, debe de presentar éstas métricas de manera que le permita al usuario integrarlas a su conocimiento. A partir de este conocimiento, el usuario podrá tomar la decisión de a quién contactar para iniciar una interacción de ayuda.

El sistema visual humano se considera el sentido con mayor ancho de banda (cantidad de datos en un intervalo de tiempo). Por ello, los seres humanos poseemos la capacidad de procesar rápidamente escenas visuales, texto y símbolos [75, 77]. Con base

en esto, se propone el diseño de una visualización que incorpore aspectos visuales, tales como color y forma que permitan una fácil interacción y entendimiento de la información que se presenta (ver Figura 4.1a).

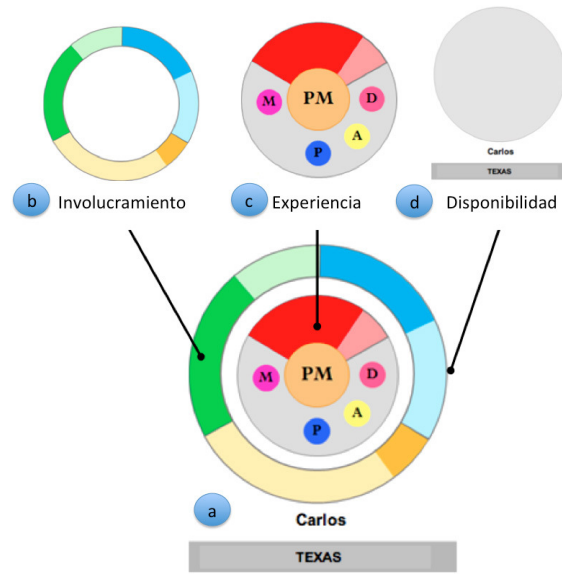


Figura 4.1: Visualización general de un colaborador potencial.

A través de una visualización general con patrones concéntricos, con cada anillo dividido en múltiples sectores, se representan los tres conceptos: Involucramiento (ver Figura 4.1b), Experiencia (ver Figura 4.1c) y Disponibilidad (ver Figura 4.1d). La forma radial nos permite agrupar [78, 79] las métricas en una sola figura. La literatura indica que los patrones concéntricos han sido aplicados para visualizar relaciones dentro de datos multidimensionales. En la técnica del anillo dividido en sectores, cada elemento de información se asigna a un sector. Para identificar estos elementos, se asigna a cada sector un color. Similar a la codificación de color que presenta el sistema radial Traffic Analyzer [78], que aplica la técnica del anillo dividido en segmentos para visualizar el tráfico de una red.

En el ojo humano hay tres tipos de conos que detectan los colores, sobretodo el rojo,

verde y azul a los cuales son más sensibles. Lo que sugiere que nuestra percepción de los colores es similar a una cámara de video y monitores, que proyecta o detecta una amplia variedad de colores a través de combinar los píxeles rojos, verdes y azules [80]. Por ello, en nuestra visualización se hace uso de un código de colores primarios, que resulten fácil de percibir para la mayoría de los usuarios.

Cuando se utiliza el color se debe tener en cuenta el probable significado que tendrá por parte de la persona que lo observa. Los colores pueden tener diferentes significados, esto es en parte cultural, sobretodo para la cultura occidental. Sin embargo, actualmente esto resulta en un problema menor, ya que los significados se pueden aprender debido a la internacionalización de las personas [81].

Las visualizaciones de cada concepto del modelo propuesto (Involucramiento, Experiencia y Disponibilidad) a su vez, se compone de varios elementos de visualización, que se describen en la siguiente sección.

4.1.1. Involucramiento.

Este concepto se representa mediante un anillo dividido en segmentos, cada uno de ellos corresponde a las tres métricas relacionadas: porcentaje de participación (tiempo) en el proyecto (ver Figura 4.2a), porcentaje de participación (artefectos) en el proyecto (ver Figura 4.2b) y el porcentaje de contribución en el artefacto (ver Figura 4.2c).

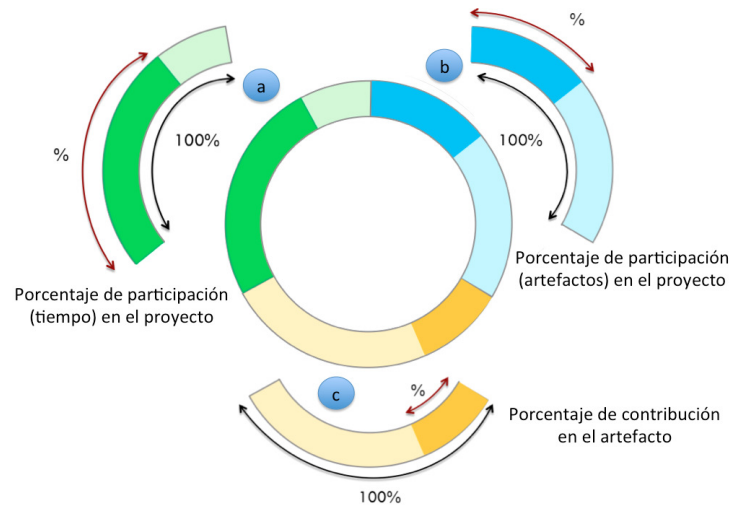


Figura 4.2: Visualización del involucramiento de un colaborador potencial.

Cada sector representa el 100% de cada una de las métricas que representa y las cuales se identifican con un color asignado. El color más tenue se utiliza como fondo de cada sector, el color más fuerte representa el porcentaje actual de cada métrica.

4.1.2. Experiencia.

Siendo consistentes en el uso de formas radiales, este concepto se representa mediante una forma circular que conjunta varias formas en su interior, para presentar las métricas relacionadas (ver Figura 4.3).

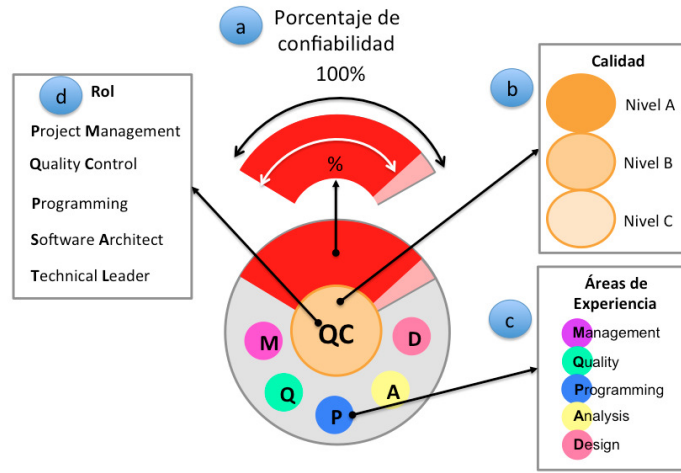


Figura 4.3: Visualización de la experiencia de un colaborador potencial.

La *confiabilidad* consiste en un segmento del círculo, identificado con el color rojo. Al igual que las métricas del involucramiento, el sector representa el 100% y el color más tenue se utiliza como fondo de cada sector, mientras que el color más fuerte es el que representa el porcentaje de la métrica (ver Figura 4.3a).

La *calidad* consiste en una forma circular al centro de la visualización individual. El color varía de acuerdo al nivel de calidad asignado, el cual puede ser un nivel alto (A), nivel medio (B) o nivel bajo (C) (ver Figura 4.3b).

Las *áreas de experiencia* consisten en varias formas circulares con diferentes colores asignados, estas aparecerán iluminadas siempre y cuando el colaborador potencial cuente con conocimiento de dichas áreas (ver Figura 4.3c).

El *rol* consiste en un texto que abrevia el nombre completo del rol que se está desempeñando (ver Figura 4.3d). A través de palabras y expresiones familiares para los desarrolladores, es posible que se transmitan conceptos con mayor claridad [81].

4.1.3. Disponibilidad.

Este concepto se representa de acuerdo con la combinación de filtrado en la que se asocie. En la Tabla 4.1 se muestran las figuras que representan la Disponibilidad para cada combinación posible con los dos conceptos restantes (Involucramiento y Experiencia).

Tabla 4.1: Visualización de la disponibilidad de un colaborador potencial por opciones de filtrado.

Operación de filtrado	Disponible	No disponible
Involucramiento Disponibilidad	 Romina QUERÉTARO	 Israel QUERÉTARO
Experiencia Disponibilidad	 Joel QUERÉTARO	 Leonardo QUERÉTARO
Involucramiento Experiencia Disponibilidad	 Romina QUERÉTARO	 Israel QUERÉTARO
Disponibilidad	 Alejandra MONTERREY	 Susana BOSTON

La *disponibilidad* puede tener dos valores (disponible o no disponible). Si se encuentra disponible el colaborador potencial, la visualización se encontrará iluminada, en caso contrario, si se encuentra no disponible la visualización se encontrará con nivel medio de transparencia. Por ello, podemos ver en la Tabla 4.1 cómo la figura muestra diferente nivel de iluminación de acuerdo al conjunto de conceptos seleccionados y la métrica disponibilidad.

Por otro lado, la *ubicación geográfica* se representa de manera textual. Un cuadro de texto al pie de la visualización muestra el nombre del lugar geográfico en dónde se encuentra ubicado actualmente el trabajador en DSD.

La psicología de Gestalt supone que la percepción visual es un proceso global evolutivo, en el que los seres humanos tienden a percibir formas geométricas simples [82]. Nuestra visualización conjunta formas y colores, para lograr proporcionar conciencia de quién se considera un colaborador potencial para apoyar en situaciones de ayuda. El resultado de ésta búsqueda puede significar un conjunto amplio de colaboradores potenciales. Por ello, resulta importante estudiar las diferentes formas de interacción y determinar las más apropiadas para proveer de un sistema fácil de utilizar. En la siguiente sección se abordan aspectos de diseño de sistemas interactivos que apoyaron en el diseño de un prototipo.

4.2. Sistemas Interactivos.

La interacción permite a un usuario explorar un conjunto de datos, a través de controles adecuados en los que se especifiquen los datos y el interés sobre ellos. Al permitir seleccionar los datos que el usuario desea visualizar, filtrar información no relacionada que les impida centrarse en los temas pertinentes, y ordenar la información, se logra exponer patrones para el análisis de información [76].

El filtrado es intrínseco al proceso de visualización, para lo cual se deben de construir una variedad de visualizaciones para cada conjunto de datos seleccionados por el usuario. Los datos se pueden filtrar utilizando un conjunto amplio de componentes, tales como un simple botón de radio, casillas de verificación, listas de desplazamiento, y cuadros de búsqueda, entre otros [76].

El ordenar es otra de las operaciones básicas dentro de una visualización, que al realizarse de manera adecuada se podrá percibir una tendencia en los datos o agrupar en función de una unidad de análisis [76]. Con base en estas operaciones se diseñaron dos sistemas interactivos: uno en forma horizontal y otro en forma de campana. Cada uno de ellos es descrito en las subsecciones siguientes.

4.2.1. Sistema Interactivo en forma horizontal.

En este sistema se presentan vistas de cinco visualizaciones generales, iniciando con el conjunto de los colaboradores potenciales que se consideren más apropiados para apoyar en la situación de ayuda (ver Figura 4.4).

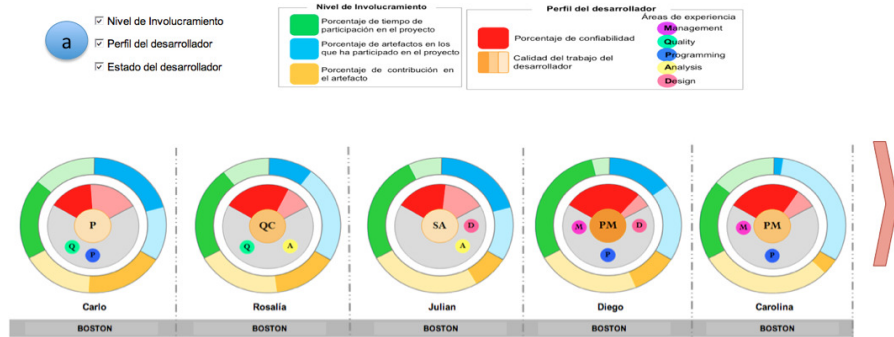
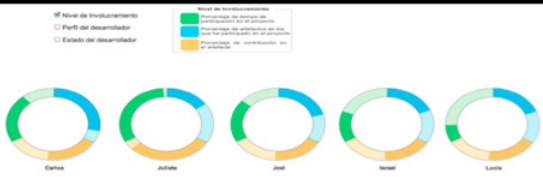
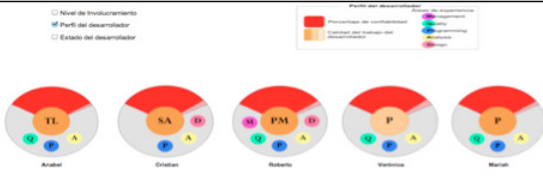
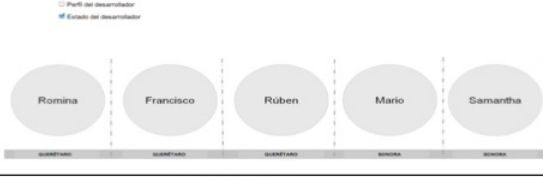
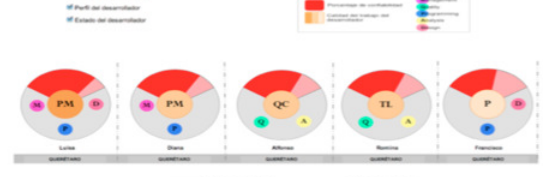
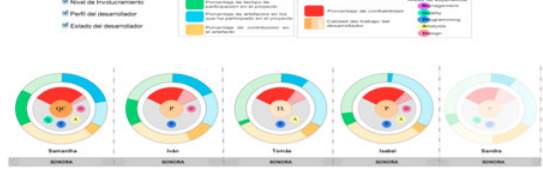


Figura 4.4: Interfaz del sistema interactivo de visualización en forma horizontal.

El filtrado de los datos se lleva a cabo a través de las casillas de verificación, para cada uno de los conceptos del modelo de identificación de colaboradores potenciales: Involucramiento, Experiencia y Disponibilidad (ver Figura 4.4a). En cada opción de

filtrado, se ordenan los datos con base en la unión de cada una de las métricas que conforman el concepto relacionado. En la Tabla 4.2 se presenta la visualización resultante de cada una de las combinaciones posibles de las opciones de filtrado. Así podemos observar que en cada visualización resultado, la primera visualización general corresponde al colaborador potencial más apropiado para poder iniciar una interacción de ayuda.

Tabla 4.2: Opciones de filtrado en el sistema en forma horizontal.

Opción de filtrado	Visualización resultado
Involucramiento	
Experiencia	
Involucramiento Experiencia	
Disponibilidad	
Involucramiento Disponibilidad	
Experiencia Disponibilidad	
Involucramiento Experiencia Disponibilidad	

La familiarización con los diferentes componentes del sistema será progresiva conforme el usuario interactúe con el mismo. Sin embargo, para apoyar al usuario en recordar las asociaciones a los diferentes componentes, se presenta un cuadro de guía que muestra información de ayuda para la interacción.

Para poder obtener mayor información respecto a los porcentajes presentados, se deberá poner el cursor sobre la visualización general del colaborador potencial de interés (ver Figura 4.5).

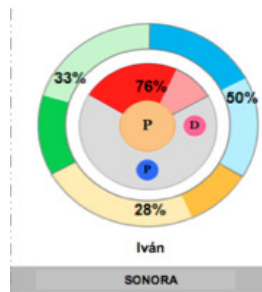


Figura 4.5: Información adicional mostrada como porcentajes en la visualización general en el sistema en forma horizontal.

4.2.2. Sistema Interactivo en forma de campana.

En este sistema se presentan 26 visualizaciones generales de los colaboradores potenciales identificados. Con la característica de que en la parte central de la vista, se encuentran los colaboradores potenciales que se consideran más apropiados para apoyar en la situación de ayuda (ver Figura 4.6).

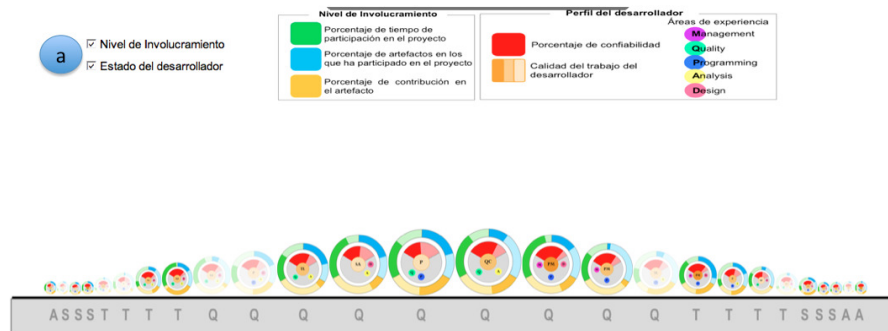


Figura 4.6: Interfaz del sistema interactivo de visualización en forma de campana.

El filtrado de los datos se lleva a cabo a través de las casillas de verificación, para dos de los tres conceptos del modelo de identificación de colaboradores potenciales: Involucramiento y Experiencia (ver Figura 4.6a). En la Tabla 4.3 se presenta la visualización resultante de cada una de las combinaciones posibles de las opciones de filtrado. Al igual que el sistema horizontal, en cada opción de filtrado se ordenan los datos con base en la unión de cada una de las métricas que conforman el concepto relacionado; con la característica de que las visualizaciones generales ubicadas al centro, corresponden a los colaboradores potenciales que se consideran más apropiados para poder iniciar una interacción de ayuda.

Tabla 4.3: Opciones de filtrado en el sistema en forma de campana.

Opción de filtrado	Visualización resultado
Involucramiento	☒ Nivel de Involucramiento ☐ Estado del desarrollador Nivel de involucramiento - Verde: Porcentaje de tiempo de participación en el proyecto - Azul: Porcentaje de artefactos en los que ha participado en el proyecto - Naranja: Porcentaje de contribución en el artefacto
Disponibilidad	☐ Nivel de Involucramiento ☒ Estado del desarrollador ATQQQA A A A S T T Q Q T S S A A QQTAA
Involucramiento Disponibilidad	☒ Nivel de Involucramiento ☒ Estado del desarrollador Nivel de involucramiento - Verde: Porcentaje de tiempo de participación en el proyecto - Azul: Porcentaje de artefactos en los que ha participado en el proyecto - Naranja: Porcentaje de contribución en el artefacto Perfil del desarrollador - Rojo: Porcentaje de confiabilidad - Naranja: Calidad del trabajo del desarrollador Áreas de experiencia - Engagement - Quality - Programming - Analysis - Design ASSST T T T Q Q Q Q Q Q Q Q Q T T T TSSAA

De manera similar al sistema horizontal, se presenta el apoyo visual para el usuario para recordar las asociaciones a los diferentes componentes. Además, al pasar el cursor sobre las visualizaciones generales, se obtendrá una vista ampliada de cada una de ellas (ver Figura 4.7). Esta técnica es llamada FishEye, y proporciona detalle local y del contexto en general dentro de un amplio conjunto de datos [83].

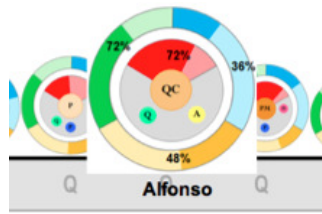


Figura 4.7: Detalles de la visualización general a través de la técnica FishEye en el sistema en forma de campana.

Para entender de manera más clara el funcionamiento de estos sistemas para la identificación de colaboradores potenciales, en la siguiente sección se explica de manera detallada como la visualización apoya a los trabajadores del DSD en situaciones de ayuda.

4.3. Funcionamiento del sistema.

Para ilustrar de manera clara el funcionamiento de un sistema interactivo de visualización, se presenta la Figura 4.8, la cual representa los componentes que intervienen cuando a un trabajador en DSD se le presentan situaciones de ayuda y que permite mostrar un escenario de uso del sistema interactivo de visualización.

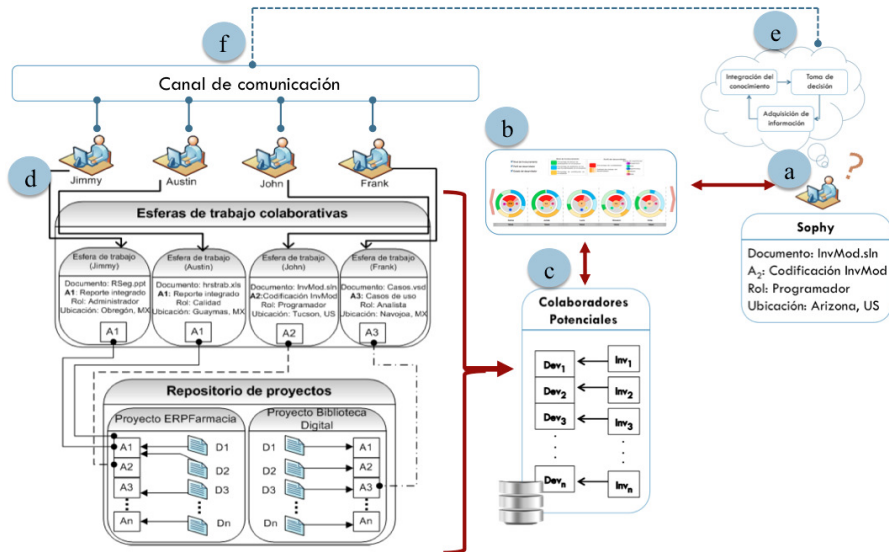


Figura 4.8: Representación del funcionamiento de un sistema interactivo de visualización para la identificación de colaboradores potenciales.

En este caso, Sophy (programador) está trabajando en el documento `InvMod.sln`, el cual es un documento asociado a la actividad A_2 : Codificación del `InvMod`. Cuando se le presenta una situación de ayuda, Sophy requiere identificar al colega con el que se pueda comunicar para consultar algunos detalles del diseño de la base de datos (ver Figura 4.8a). Sophy solicita información a través del sistema interactivo de visualización (ver Figura 4.8b), que realiza una búsqueda en su repositorio de datos sobre el Involucramiento, Experiencia y Disponibilidad de los colaboradores potenciales (ver Figura 4.8c). Esta información a su vez, se recolecta de las esferas de trabajo colaborativas propuestas por Palacio et al. [72] (ver Figura 4.8d). A Sophy se le presenta el resultado de la búsqueda a través del mismo sistema interactivo. Con esta información proporcionada por el sistema, Sophy crea conocimiento a través de su proceso cognitivo, que la lleva a tomar la decisión de contactar a Jimmy por que ha estado más involucrado en ese proyecto (ver Figura 4.8e). Sophy podrá contactar a Jimmy mediante los diferentes

canales de comunicación con los que cuenta la organización y podrá continuar con sus tareas asignadas (ver Figura 4.8f).

4.4. Resumen del capítulo.

En este capítulo se ha proporcionado una descripción del diseño de un conjunto de visualizaciones, que representan los conceptos y métricas del modelo propuesto en el Capítulo 3. Además se ha proporcionado la descripción del diseño de un sistema interactivo con dos formas de visualización de información, que presentan las visualizaciones antes propuestas. Como parte de los resultados de los resultados de este capítulo:

- Se presentó una descripción de los elementos de visualización que utilizan las visualizaciones propuestas.
- Se presentó una descripción de las técnicas de visualización de información que utilizan los dos sistemas interactivos de visualización de información (Horizontal, Campana).
- Se presentó una representación y escenario del funcionamiento del sistema interactivo de visualización de información, cuando se presentan situaciones de ayuda en DSD.

Como parte del área de interacción humano-computador (HCI, por sus siglas en inglés) se consideran las evaluaciones de usuario, debido a la necesidad de realizar pruebas con usuarios reales y de ser posible, en situaciones reales. Por ello, en el siguiente capítulo se presenta la evaluación del modelo y del sistema interactivo de visualización de información por un grupo de trabajadores de la industria del desarrollo de software.

Capítulo 5

Resultados.

En el capítulo anterior se describió el diseño de un sistema interactivo de visualización de información, el cual presenta tres grupos de métricas (Involucramiento, Experiencia y Disponibilidad) para apoyo en la identificación de colaboradores potenciales en desarrollo distribuido de software (DSD). A través de este sistema de visualización de información, los desarrolladores pueden decidir a quién contactar cuando se les presenta un problema o requieren de información que les permita continuar desempeñando su trabajo.

En este capítulo se presenta la evaluación de dicho sistema interactivo de visualización, el cual consistió en presentar estos tres grupos de métricas para la identificación de colaboradores potenciales en situaciones de ayuda. El objetivo de esta evaluación es captar la percepción de utilidad y facilidad de uso de dicho sistema interactivo de visualización por parte de un grupo de trabajadores de DSD.

Además, con la evaluación se pretende demostrar que a través de los tres grupos de métricas propuestos (Involucramiento, Experiencia y Disponibilidad) se puede apoyar a los trabajadores en la identificación de colaboradores potenciales entre los miembros de una organización, en caso de un problema o necesidad.

Para recolectar la información percibida por los participantes, se generó un cues-

tionario escala 7-Likert de siete preguntas en términos de facilidad de uso en la identificación de colaboradores potenciales. Además, se utilizó la escala de usabilidad del sistema (SUS, System Usability Scale) propuesta en [84].

Este caso de estudio se dividió en cuatro pasos: la delimitación de los objetivos de la evaluación, el establecimiento de las hipótesis, la validación de las hipótesis y el análisis de los resultados. Las siguientes secciones describen las cuatro fases realizadas para este caso de estudio.

5.1. Objetivos.

Los objetivos de esta evaluación consistieron en evaluar:

1. el impacto de un sistema interactivo de visualización de información en términos de utilidad y facilidad para la identificación de colaboradores potenciales en DSD; y
2. conocer el impacto de proporcionar información que apoya en la toma de decisión de a quién acudir en situaciones de ayuda.

5.2. Hipótesis.

Para guiar esta evaluación se estableció un conjunto de hipótesis de trabajo, en términos de usabilidad (satisfacción, efectividad y eficiencia).

En cuanto a efectividad se establecen:

HO: No existen diferencias en la efectividad para identificar colaboradores potenciales a través del sistema interactivo de visualización con respecto a la de realizarlo con la visualización tabular.

H1a: Los sujetos identificarán en más ocasiones y de forma correcta a colaboradores potenciales en situaciones de ayuda a través del sistema interactivo de visualización, que con la visualización tabular.

En cuanto a eficiencia se establece que:

HO: No existen diferencias en el tiempo que le toma a los sujetos para identificar colaboradores potenciales a través del sistema interactivo de visualización respecto a que se realice a través de la visualización tabular.

H2a: Los sujetos identificarán en menor tiempo a colaboradores potenciales en situaciones de ayuda a través del sistema interactivo de visualización, que con la visualización tabular.

En cuanto a la satisfacción se establecen:

HO: No existen diferencias en la percepción de los sujetos en términos de utilidad y facilidad del sistema interactivo de visualización y la visualización tabular.

H3a: Los sujetos percibirán que el sistema interactivo de visualización es más útil que la visualización tabular.

H3b: Los sujetos percibirán que el sistema interactivo de visualización es más fácil de utilizar que la visualización tabular.

5.3. Diseño del estudio.

El paradigma de diseño para este estudio fue WithIn Subjects. Para evitar el efecto de aprendizaje, se asignaron aleatoriamente los participantes del estudio a cada forma de visualización. El 50% de los participantes iniciaron interactuando con el sistema interactivo de visualización y posteriormente con la visualización tabular. Mientras

que el otro 50 % iniciaron con la visualización tabular y continuaron con el sistema interactivo de visualización.

5.4. Participante.

La evaluación se realizó con 26 trabajadores profesionales de desarrollo distribuido de software, de los cuales 18 fueron de sexo masculino y 8 de sexo femenino.

El promedio de edad de los participantes fue de 28 años y en cuanto a la experiencia laboral, los participantes promediaban 5 años de experiencia.

5.5. Procedimiento.

En total se llevaron a cabo 9 actividades para realizar la evaluación, las cuales se describen a continuación:

1. Preparación de material para evaluación.
2. Presentación con los trabajadores. Se les informó a los participantes la intención del estudio y se les explicó cuál sería su participación y la forma en que se llevaría a cabo.
3. Solicitud de datos demográficos. Se les pidió que respondieran algunas preguntas respecto a datos demográficos, tales como: nombre, edad, último grado de estudios, experiencia en el desarrollo de software y rol/es que ha desempeñado en el desarrollo de software.
4. Presentación de métricas. Se les mostró a los sujetos una presentación en la cual se les explicaba cada una de las métricas para la identificación de colaboradores potenciales, así como su representación visual.

5. Percepción de utilidad de las métricas propuestas. Se les presentó a los sujetos un escenario que representaban situaciones que se pueden presentar en el desarrollo distribuido de software. Al final se les realizó una entrevista a cada participante, para captar la forma en que ellos identifican a colaboradores potenciales en situaciones de ayuda y su percepción con respecto a las métricas propuestas.
6. Entrenamiento en formas de visualización. Se les capacitó a los sujetos para el uso de sistemas interactivos de visualización y el filtrado de información en la forma tabular.
7. Interacción con el sistema de visualización. Se introdujo a los participantes a dos escenarios, los cuales representaban situaciones de ayuda en la que se requiere identificar a ciertos colaboradores potenciales en específico (ver Tabla 5.1).

Tabla 5.1: Definición de los escenarios utilizados en la evaluación.

Escenario	Meta	Elementos
Áreas de experiencia	Identificar a colegas con experiencia en programación	Áreas de experiencia-Perfil del desarrollador
Involucramiento	Identificar a colegas involucrados en el proyecto	Porcentaje de participación en el proyecto, porcentaje de artefactos en los que participa en el proyecto y porcentaje de contribución en el artefacto-Involucramiento

Después de leer el escenario de cada una de las dos tareas a realizar, se les pidió que utilizaran la forma de visualización correspondiente y seleccionaran con base a su

propio juicio el (/los) colega(s) que podría(n) apoyarlo en resolver el problema planteado. Posteriormente cambiaron de forma de visualización.

8. Evaluación de usabilidad y facilidad de uso. Se les pidió al final de las dos tareas que llenaran tanto un cuestionario escala 7-Likert referente a la facilidad de uso (ver Apéndice A) como el cuestionario SUS (System Usability Scale) (ver Apéndice B) [84].
9. Entrevista final. Se realizó una entrevista de manera personal a cada participante, para captar sus preferencias con respecto a las dos formas de visualización de información y la percepción de facilidad de uso y utilidad de la visualización que se propone.

5.6. Resultados.

En este apartado se presentan los resultados del estudio para determinar la percepción de trabajadores de DSD, en términos de usabilidad (efectividad, eficiencia y satisfacción) del sistema interactivo de visualización de información para la identificación de colaboradores potenciales en DSD.

Efectividad:

Puesto que se les pide a los sujetos identificar con base a su propio juicio el (los) colega(s) que podría(n) apoyarlo en resolver el problema planteado, consideramos las siguientes hipótesis en cuanto a efectividad:

HO: No existen diferencias en la efectividad para identificar colaboradores potenciales a través del sistema interactivo de visualización con respecto a la de realizarlo con la visualización tabular.

H1a: Los sujetos identificarán en más ocasiones y de forma correcta a colaboradores potenciales en situaciones de ayuda a través del sistema interactivo de visualización, que con la visualización tabular.

Para validar esta hipótesis se verificó que el colega seleccionado cumpliera con las condiciones planteadas. Sin embargo, se podría dar el caso de que seleccionaran a uno de los colegas correctos sin considerar las métricas adecuadas de acuerdo al escenario. Esta situación derivada de que dichas métricas no eran presentadas de forma explícita al participante, sino que estos realizaban la selección bajo su propio juicio. Por lo tanto, para interpretar los datos de la acción tomada por el participante, ésta fue codificada de la siguiente forma:

- Si el sujeto identificaba al colega correcto y consideraba la métrica correspondiente al escenario, se consideraba como tarea correcta y se asignaba un 1.
- Si el sujeto identificaba al colega correcto pero no consideraba la métrica correspondiente al escenario, se consideraba parcialmente correcta y se le asignaba un 0.5.
- Si el sujeto no identificaba al colega correcto se consideraba incorrecta y se le asignaba un 0.

Para el procesamiento de los datos obtenidos, se utilizó el paquete estadístico R-project. Para llevar a cabo la comparación entre la efectividad de las dos formas de visualización de información, se realizó un análisis mediante la prueba t de Student para muestras independientes. Dicho análisis fue para comparar las medias con respecto a la efectividad de los participantes para identificar colaboradores potenciales en dos situaciones de ayuda, a través de las dos formas de visualización de información y determinar si existen diferencias significativas entre ellas.

Para decidir si los participantes identificaban efectivamente a colaboradores potenciales a través de una forma de visualización en relación con la otra en un escenario determinado, la prueba t calcula un valor p basado en el desempeño de los datos. Si el valor p es lo suficientemente pequeño (valor de $p < 0.05$), se concluye que la diferencia es significativa.

Por ello, en la Figura 5.1 presentamos los datos de las tareas correctas y parcialmente correctas obtenidas en cada una de las dos tareas realizadas.

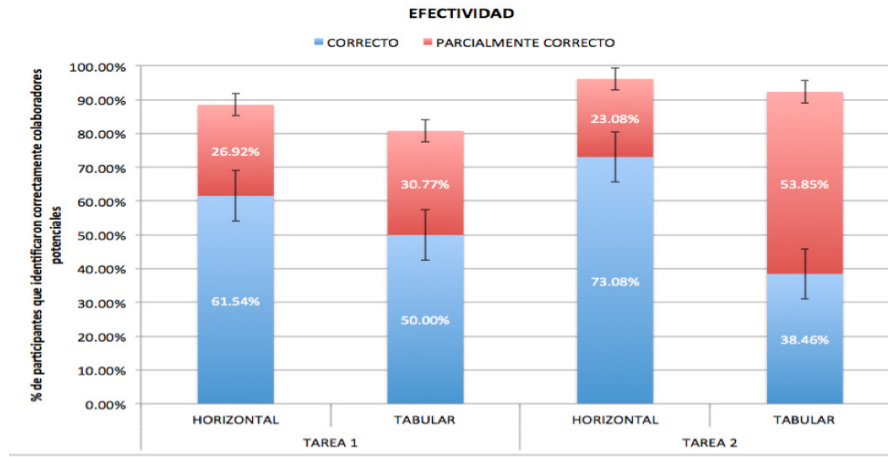


Figura 5.1: Tareas correctas y parcialmente correctas.

El análisis de las diferencias entre el sistema interactivo de visualización ($media = 0.75, desv.est. = 0.354$) y la forma tabular ($media = 0.65, desv.est. = 0.394$) en la tarea 1 no fue significativo ($t_{0.05} = 1.308, GL = 25, p = 0.1013$). El análisis de diferencias entre el sistema interactivo de visualización ($media = 0.85, desv.est. = 0.275$) y la forma tabular ($media = 0.65, desv.est. = 0.309$) en la tarea 2, arrojó una diferencia significativa ($t_{0.05} = 2.606, GL = 25, p = 0.0076$).

Los resultados del análisis de efectividad en la identificación de colaboradores potenciales (con un rango de 0 a 1) proporcionan evidencia para rechazar la hipótesis H1a y aceptar la hipótesis nula (H0) para esta hipótesis en particular, la cual establece

que no existen diferencias en la efectividad para identificar colaboradores potenciales a través del sistema interactivo de visualización respecto a que se realice a través de la visualización tabular.

Eficiencia:

Puesto que se les pide a los sujetos identificar con base a su propio juicio el/los colegas que podrían apoyarlo en resolver el problema planteado, consideramos las siguientes hipótesis en cuanto a eficiencia:

HO: No existen diferencias en el tiempo que le toma a los sujetos para identificar colaboradores potenciales a través del sistema interactivo de visualización respecto a que se realice a través de la visualización tabular.

H2a: Los sujetos identificarán en menor tiempo a colaboradores potenciales en situaciones de ayuda a través del sistema interactivo de visualización, que con la visualización tabular.

Para el procesamiento de los datos obtenidos, se utilizó al igual que el caso anterior el paquete estadístico R-project. Para llevar a cabo la comparación entre la eficiencia entre las dos formas de visualización de información en las dos situaciones de ayuda planteadas, se realizó un análisis mediante la prueba t de Student para muestras independientes. Dicho análisis fue para comparar la medias con respecto a la eficiencia que mostraron los participantes al identificar colaboradores potenciales de forma correcta y parcialmente correcta, en las situaciones de ayuda correspondientes a cada tarea.

Para obtener datos y probar esta hipótesis se guardaron los tiempos que le tomaba identificar al colaborador potencial a cada participante por tarea. En la Figura 5.2 se presentan las medias para los tiempos en las respuestas correctas y parcialmente correctas de las dos tareas realizadas.

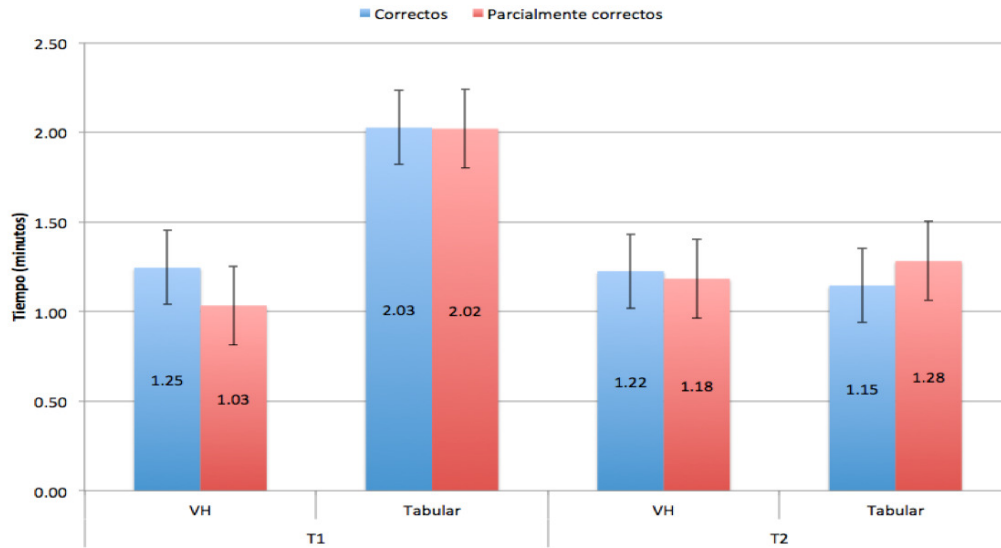


Figura 5.2: Tiempos obtenidos en la identificación de colaboradores potenciales.

En la tarea 1 se obtuvieron tiempos menores en la interacción con el sistema interactivo de visualización ($media = 1.18, desv.est. = 0.698$) en comparación con la visualización tabular ($media = 2.02, desv.est. = 0.799$). Mientras que en la tarea 2, se registraron tiempos muy similares entre el sistema interactivo de visualización ($media = 1.21, desv.est. = 0.786$) y la visualización tabular ($media = 1.22, desv.est. = 0.775$).

En la tarea 1 se obtuvo que $t_{0.05} = 1.6819, GL = 42, p = 0.00028$ por lo tanto se establece que existen diferencias significativas, lo que nos indica que en la interacción con el sistema interactivo de visualización, los sujetos identificaron a colaboradores potenciales en menos tiempo en comparación con la visualización tabular. Mientras que en la tarea 2 se obtuvo que $t_{0.05} = 1.6779, GL = 47, p = 0.4826$ por lo tanto se establece que no existen diferencias significativas entre el tiempo que les tomó a los participantes en identificar colaboradores potenciales a través del sistema interactivo de visualización y la visualización tabular. En este caso la H2a es rechazada.

Satisfacción:

Puesto que partimos de que los participantes contaran con información para la identificación de colaboradores potenciales, consideramos las siguientes hipótesis en cuanto a la utilidad y facilidad de uso:

HO: No existen diferencias en la percepción de los sujetos en términos de utilidad del sistema interactivo de visualización y la visualización tabular.

H3a: Los sujetos percibirán que el sistema interactivo de visualización es más útil que la visualización tabular.

Para abordar esta hipótesis se utilizó el cuestionario SUS. Los resultados arrojados por este cuestionario (con rango de 0 a 100) y que se presentan en la Figura 5.3, indican que con el sistema interactivo de visualización en forma horizontal ($media = 74.42, desv.est. = 12.734$) es más útil que la visualización tabular ($media = 65.67, desv.est. = 17.657$).

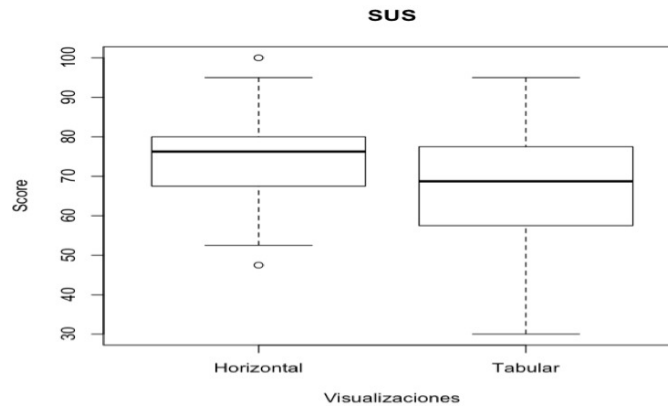


Figura 5.3: Puntuaciones del SUS obtenidas por forma de visualización.

Para confirmar esta información se realizó la prueba t de Student a nivel de significancia de $p < 0.05$. Dado que $t_{0.05} = 1.986$, $GL = 25$, $p = 0.029$ se comprueba que esta diferencia es significativa. Por lo que los resultados arrojados del SUS indican que la H3a es aceptada.

En cuanto a percepción de facilidad de uso en la interacción con el sistema interactivo de visualización y la visualización tabular.

HO: No existen diferencias en la percepción de los sujetos en términos de facilidad de uso del sistema interactivo de visualización y la visualización tabular.

H3b: Los sujetos percibirán que el sistema interactivo de visualización es más fácil de utilizar que la visualización tabular.

Para abordar esta hipótesis se generó un cuestionario de siete preguntas en términos de facilidad de uso en la interacción con el sistema de visualización. Este cuestionario utiliza una escala Likert de 1 (completamente en desacuerdo) a 7 (completamente de acuerdo). Los resultados arrojados por este cuestionario y presentados en la Figura 5.4, indican que con el sistema interactivo de visualización en forma horizontal ($media = 5.91, desv.est. = 0.978$) es más satisfactorio interactuar para identificar colaboradores potenciales, que con la visualización tabular ($media = 5.16, desv.est. = 1.473$).

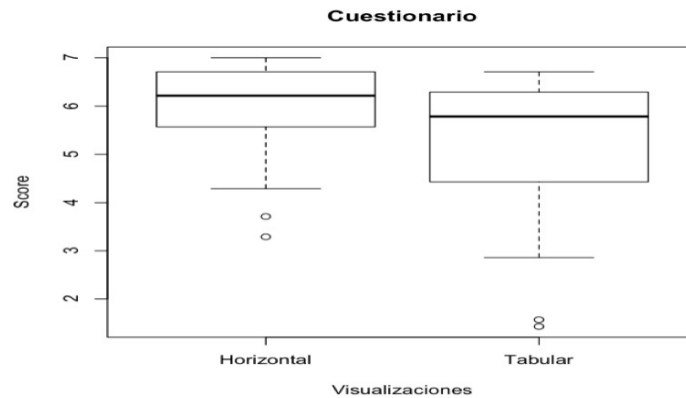


Figura 5.4: Puntuaciones del cuestionario escala 7-Likert obtenidas por forma de visualización.

Para confirmar esta información se realizó la prueba t de Student a nivel de significancia de $p < 0.05$. Dado que $t_{0.05} = 1.842$, $GL = 25$, $p = 0.0386$ se valida que si existen

diferencias significativas, que nos confirman que el sistema interactivo de visualización es percibido como más fácil de usar. Lo cual permite aceptar H3b.

5.7. Discusión.

A continuación se presenta una discusión respecto al análisis de los resultados obtenidos, reflexionado sobre las razones que podrían justificarlos.

En cuanto a la *efectividad*, en el caso de H1a se considera que los participantes no mostraron mayor efectividad para identificar colaboradores potenciales a través del sistema interactivo de visualización respecto a que se realice a través de la visualización tabular, debido a que en ambas formas de visualización se presentó la misma información. Sin embargo, se obtuvo un porcentaje alto de efectividad tanto en el sistema interactivo de visualización (Tarea 1 – 88.46 %, Tarea 2 – 96.16 %) como en la visualización tabular (Tarea 1 – 80.77 %, Tarea 2 – 92.31 %). Estos resultados dan indicios para pensar que las métricas del modelo propuesto, ofrecen información suficiente para la identificación de colaboradores potenciales, e influenciar en la decisión de los trabajadores de contactar a una persona en particular en situaciones de ayuda. La creación de lazos de confianza en DSD resulta complicada debido a la dispersión geográfica de los miembros. Esto ha sido abordado por trabajos de investigación como [61, 68, 72], quién considera que la falta de colaboración puede implicar una falta de confianza. Por lo que algunos elementos de información influyen en el desarrollo de esta confianza, tales como la conciencia de que cada vez que una persona edita un código fuente lo realiza sin errores, el cumplimiento de los tiempos de entrega cuando existe una dependencia entre tareas de un proyecto, así como si un código es utilizado frecuentemente por otros desarrolladores. Esta explicación es por comentarios referentes a algunas de las métricas propuestas, como los siguientes:

... si es importante [métrica de calidad] por el hecho de que si está persona es... digamos es una persona que por lo regular sus entregables no tienen tantos defectos, o son trabajos bien presentados. Por lo regular tienen más facilidad para expresarte las cosas, o lo hacen de una manera más sencilla... [Participante 7].

... si es muy importante [métrica de confiabilidad], porque si surge un problema en producción y corres con el que estás más seguro que te lo va a sacar más... que es más seguro que te lo saque sin problemas en el tiempo que te dan... [Participante 3]

... ésta persona tiene un conocimiento más técnico del módulo de ese artefacto [métrica de contribución], cuando se desarrolló y lo que a mí me puede tomar comprender, que será un día, tres horas, cuatro horas. Está persona si la identificó a tiempo, ésta persona me puede explicar cómo funciona en cuestión ya sea de minutos o una hora... [Participante 7]

... una persona entre más haya pasado tiempo con el proyecto [métrica de tiempo de participación], usualmente va a hacer más cosas, más componentes y pues pues te puede, digamos dar una mejor solución o tienes más confianza de que te puedan ayudar... [Participante 6]

... una persona aunque haya contribuido mucho en el proyecto, o que sea muy confiable, si no tiene tiempo vas a tener que estar esperando a que se desocupe [métrica de disponibilidad]; y a lo mejor en ese tiempo que estas ahí esperando te puede ayudar a resolverlo otra persona que a lo mejor no te lo va a resolver rápidamente pero no vas a tener ese tiempo muerto... [Participante 5]

El mantenimiento de software es una tarea en la que se requiere comprender el

funcionamiento de los sistemas de software y los procesos que en ocasiones fueron desarrollados por terceras personas. Considerando los trabajos de [85], el proporcionar información relevante que apoye, fomente y mejore la colaboración. Esta explicación se presenta con base en los comentarios realizados por algunos participantes del estudio. Estos comentarios incluyen:

... después es muy difícil cuando tienes que hacer un mantenimiento donde un programa que tu no hiciste, que tu no analizaste. Al momento de tu querer hacer una modificación, prácticamente es iniciar de ceros. Entonces si tu sabes quién inicio esto, o quién colaboró [métrica de contribución], ya te ahorras mucho tiempo al dar mantenimientos o en hacer modificaciones también... [Participante 4]

... cómo esta más involucrado en todo el código es más factible que sepa dónde puede estar el error, sabe manejar bien el código, sabe dónde tener que ir a buscar cierto tipo de cosas, en dónde puede estar fallando y ese tipo de cosas... [Participante 10]

Para apoyo en inicios de colaboración, trabajos como el de [72] consideran que se requiere proporcionar información suficiente al iniciador de la interacción, para que esté pueda determinar el momento adecuado para contactarlo. Para lo cual, elementos de información de ubicación geográfica y disponibilidad actual de los desarrolladores proporcionan conciencia que puede influenciar en la decisión de cuando es apropiado contactar un colaborador potencial. Esta explicación se soporta en base a los comentarios referentes a las métricas del concepto de Disponibilidad:

... la ubicación geográfica también impacta en el lenguaje, que tengamos facilidad de comunicarnos... [Participante 2]

...para saber si esta al alcance de una llamada [métrica de ubicación geográfica], si puedo contar con él en los próximos cinco minutos o si tengo que esperarme todavía unas 24 horas... [Participante 21]

... es importante [métrica de ubicación geográfica] por el horario, porque si es que no va a estar disponible, pues ni pierdes tiempo en intentar contactarlo... [Participante 24]

En lo que respecta a la *eficiencia*, los resultados para H2a indican que tanto en el sistema interactivo de visualización como en la visualización tabular se registraron tiempos similares en la identificación de colaboradores potenciales. Sin embargo, en el caso de la visualización tabular los participantes contaban con experiencia previa en la utilización de filtros (en general) en una hoja de cálculo, la cual influyó en los resultados obtenidos mientras que no tenían experiencia similar en la utilización del sistema de visualización propuesto. En este sentido, el sistema interactivo de visualización presenta la información a través de elementos gráficos (formas y colores), para lo cual requieren que los participantes se familiaricen con sus significados previo a su interacción. La visualización de información, es un medio a través del cual se puede dar sentido a los datos. Los diseñadores de visualizaciones aprovechan algunas propiedades visuales, tales como: la posición, el tamaño, la forma y el color, para apoyar a los usuarios a interpretar los datos que se les presentan. De esta manera, resulta un apoyo en los procesos de búsqueda de manera eficiente [76,77]. Esta explicación se presenta con base en los comentarios realizados por algunos participantes del estudio. Estos comentarios incluyen:

...creo que es más complicado aquí [sistema interactivo de visualización]. Porque aquí tienes que seguir el código de colores para poder entender el porcentaje y saber de que te están hablando cada color... [Participante 4].

... me tarde más en el otro [sistema interactivo de visualización] porque no entendía las siglas, entonces tenía que estar buscando arriba y abajo... [Participante 20].

... en ésta herramienta [sistema interactivo de visualización] no puedo filtrar. Por ejemplo, como te digo, yo eliminaría primero a este, este y este que son QA's y dejaría a los puros programadores, diseñadores y arquitectos... [Participante 6].

... el rol me costo por el hecho de estar en inglés y abreviado, acordarme de que significaba [el acrónimo]... [Participante 9].

Finalmente en cuanto a *satisfacción* a través de la utilidad y facilidad de uso, para el caso de H3a y H3b se cree que los participantes percibieron al sistema interactivo de visualización más útil que la visualización tabular, porque el sistema interactivo permite a los trabajadores conocer de manera rápida y con poco esfuerzo a qué colega acudir para poder resolver una duda o problema. Trabajos como el de [53], consideran que la visualización es generalmente aceptada como una buena opción para la transmisión de información de conciencia debido a su capacidad de aprovechar los efectos de percepción que son familiares para los usuarios, tales como los efectos de primer plano y de fondo, así como el uso de color que permite que la información de interés sobresalga. Esta explicación surge de los comentarios realizados por algunos participantes del estudio en su entrevista final, tales como:

... en el gráfico es más fácil ubicar a todas las personas, aunque llegues al mismo resultado pero lo haces de manera más rápida y más fácil, ya después, cuando llevas un período de tiempo utilizándolo... [Participante 1]

...porque es más visual y porque te ahorra mucho tiempo, porque en excel pues hay que estarle ahora si que todas...todas las opciones que te interesan ir seleccionando, seleccionando y seleccionando... [Participante 5]

...no necesito analizar tanto los números, sino que simplemente miro de primer instancia, miro simplemente las gráficas...me muestra rápidamente quiénes son los más adecuados... [Participante 7]

...te da completamente directo a la persona, el sistema prácticamente se encarga de hacer lo que tu haces con la mente... [Participante 10]

...a primera vista alcanza uno a identificar las diferencias de las personas... [Participante 19]

5.8. Resumen del capítulo.

En este capítulo se han presentado los resultados de un caso de estudio realizado con trabajadores profesionales de desarrollo distribuido de software. El estudio fue realizado para evaluar dos formas de visualización de información (sistema interactivo de visualización y visualización tabular) en términos de:

- Utilidad y facilidad de uso.
- Apoyo en la identificación de colaboradores potenciales en situaciones de ayuda.

Como parte de los resultados de este caso de estudio, se identificaron algunos puntos que resultan de importancia para mejorar y apoyar a los inicios de interacción en situaciones de ayuda a través de un sistema interactivo de visualización. Este tipo de

observaciones permiten plantear en el siguiente capítulo las principales conclusiones, aportaciones y trabajo a futuro derivados de la presente tesis.

Capítulo 6

Conclusiones, contribución y trabajo futuro.

En este trabajo de tesis se atiende la problemática de identificar a colaboradores potenciales para atender situaciones de ayuda en el desarrollo distribuido de software (DSD, por sus siglas en inglés). Esto fue abordado mediante la propuesta de tres conceptos: Involucramiento, Experiencia y Disponibilidad, los cuales se componen de métricas descritas en el capítulo 3. Para validar este modelo, se diseñaron dos sistemas interactivos de visualización de información, buscando optimizar la presentación e integración de diferentes métricas, a través de las cuales sea posible identificar a colaboradores potenciales dentro de una organización.

Con base en lo anterior, fue posible evaluar el modelo y los sistemas interactivos de visualización de información, mediante una evaluación basada en escenarios con trabajadores de la industria del desarrollo de software.

En las siguientes secciones se presentan las conclusiones de este trabajo, las principales aportaciones del mismo, y las principales líneas de trabajo futuro.

6.1. Conclusiones.

La industria del desarrollo de software con el cambio a un paradigma distribuido, ha ganado ventaja competitiva en términos de costos y calidad. Sin embargo, aunado a estas ventajas, el DSD conlleva nuevos desafíos de tipo contextual, cultural y geográfico entre los desarrolladores.

Durante el desarrollo de software, los desarrolladores suelen compartir e intercambiar conocimientos relacionados a su trabajo interactuando constantemente unos con otros. Esto se vuelve complicado en entornos distribuidos, debido a la distancia temporal y física que impide el acceso expedito a los miembros del equipo. Un desarrollador a menudo se enfrenta a la necesidad de identificar a quién contactar, con el fin de resolver un problema.

Algunos de los problemas que los desarrolladores enfrentan en este contexto ocurren durante la codificación de un bloque de código, al realizar cambios o actividades de mantenimiento, al atender errores, al reproducir fallos, al tratar de entender el comportamiento de un ejecutable, o durante el razonamiento sobre el diseño, entre otros. En estos casos, las limitaciones por la distancia geográfica retardan inicios de interacción entre desarrolladores, lo que hace difícil la identificación de expertos. Esta problemática, es un reto que exige identificar elementos de información que aporten a la conciencia pérdida por la distancia geográfica. Como parte de ello, este trabajo se enfocó en la necesidad de proporcionar información de conciencia a los trabajadores de DSD, que los apoye a identificar a colaboradores potenciales cuando enfrentan situaciones en las que requieren ayuda.

Para lo anterior, en este trabajo se inició con un proceso de entendimiento basado en la literatura de los momentos en los que un ingeniero de software enfrenta la necesidad de identificar a un colaborador inicial. Con esto, se procedió a definir escenarios que determinaran los criterios que toman en cuenta los desarrolladores para

identificar colaboradores potenciales entre los miembros de una organización. Considerando estos criterios, se definió un modelo de métricas, que mediante los conceptos de Involucramiento, Experiencia y Disponibilidad brindan elementos de conciencia a los desarrolladores, para que puedan tomar una mejor decisión respecto a quién contactar en situaciones de ayuda.

Continuando con el trabajo de investigación, se estableció la necesidad de validar los elementos de conciencia que componen el modelo propuesto. Se revisó literatura de visualización de información para determinar la forma de representar y mostrar a los ingenieros de software esta información. Con base en estas ideas de diseño de visualizaciones, se introdujo una visualización general para representar el Involucramiento, Experiencia y Disponibilidad de un colaborador potencial. De lo anterior se derivó el desarrollo de dos prototipos de sistemas interactivos de visualización de información, los cuales presentan información de los colaboradores potenciales identificados a los ingenieros de software.

Con la finalidad de medir el impacto del modelo propuesto y facilidad de uso del prototipo del sistema interactivo de visualización de información, se llevó a cabo una evaluación mediante escenarios. En dicha evaluación, se obtuvo la percepción de 26 desarrolladores profesionales en términos de utilidad a través de entrevistas semi-estructuradas y facilidad de uso mediante cuestionarios escala 7-Likert de siete preguntas y SUS (System Usability Scale) [84]. Los resultados obtenidos indicaron que los elementos de conciencia del modelo propuesto, apoyan a los desarrolladores a identificar colaboradores potenciales en situaciones de ayuda. Cabe mencionar que la evidencia muestra que los participantes fueron eficientes tanto en el sistema interactivo de visualización como en la visualización tabular, ya que se registraron tiempos similares en la identificación de colaboradores potenciales. Sin embargo, los participantes contaban con experiencia en el manejo de la visualización tabular, mientras que era la primera ocasión que interactuaban con el sistema interactivo de visualización de información. La evaluación

también mostró evidencia de que los participantes percibieron al sistema interactivo de visualización más útil que la visualización tabular.

6.2. Contribución.

La principal contribución de este proyecto de investigación se centra en un modelo de métricas, que mediante los conceptos de conciencia del Involucramiento, Experiencia y Disponibilidad de los desarrolladores se identifica a un grupo de colaboradores potenciales. Este trabajo de investigación se ubica en el área de investigación de CSCW (Trabajo Cooperativo Asistido por Computadora, por sus siglas en inglés) y particularmente en la coordinación de trabajo colaborativo. Esto debido a que a través de la conciencia de quiénes son colaboradores potenciales, a los trabajadores les tomará menos tiempo en identificar con quién contactarse para resolver dudas o problemas que les impiden continuar con sus actividades. Este modelo fue construido con base en literatura de sistemas de conciencia y sistemas de localización de expertos. Estos trabajos describen las principales aproximaciones disponibles en la literatura y revelan cómo las investigaciones en estas áreas aportan ideas de diseño para la identificación de colaboradores potenciales.

Los sistemas de conciencia proporcionan elementos de información referente a los espacios de trabajo. Así vemos que los trabajos citados en la sección 2.1, se enfocan en la conciencia requerida durante la colaboración (es decir, la colaboración actual, p. ej. cómo progresan las tareas particulares o los artefactos del proyecto, las actividades actuales de los compañeros desarrolladores y cual es el estado actual de los recursos relacionados al proyecto). Para ello, proporcionan una amplia variedad de elementos de información para que los desarrolladores puedan identificar y enfrentar problemas de coordinación en DSD. Sin embargo, en los momentos en los que requieren identificar a quién contactar para resolver dudas o problemas, los desarrolladores deben de

identificar por separado algunos elementos que los apoyen para dicha tarea. Por esta razón, este modelo agrupa elementos de conciencia específicos para la identificación de colaboradores potenciales en situaciones de ayuda.

En cuanto a los sistemas de localización de expertos, éstos se enfocan en proporcionar un listado puntual de las personas consideradas expertos de acuerdo a una determinada heurística. Los trabajos citados en la sección 2.2, contienen diferentes mecanismos para la identificación de expertos y cada uno fue desarrollado para desempeñarse en contextos específicos. Por lo tanto, resultan de un uso reducido al momento de requerir localizar expertos en diferentes contextos, con diferentes necesidades. Para ello, este modelo proporciona información basado en la actividad que realiza cada individuo desde su espacio de trabajo al momento que se le presenta una situación de ayuda.

Tomando como base los trabajos antes mencionados, se contribuyó a la determinación de elementos de conciencia para la identificación de colaboradores potenciales para inicios informados de colaboración de ayuda, los cuales apoyan a los desarrolladores en la toma de decisión de a quién contactar para resolver una duda o problema y además proporcionan una base para la creación de ideas de diseño para desarrollar sistemas extensibles que identifiquen colaboradores potenciales.

Para poder validar este modelo, se diseñaron dos prototipos interactivos de visualización soportados en la literatura de visualización de información. Los trabajos citados en la sección 2.3 presentan diferentes técnicas de visualización de información para los diversos elementos de información que se presentan. El desarrollo de software es un proceso intensivo en información, por lo que la información de conciencia adicional pueden distraer a los desarrolladores [76, 77]. Por ello, se propone una visualización más flexible, que integra elementos de información mediante los cuales los desarrolladores serán capaces de identificar colaboradores potenciales.

6.3. Trabajo futuro.

Durante la realización de este trabajo de tesis surgieron algunas observaciones o ideas que podrían haber ampliado algunos aspectos de la tesis. Dichas ideas resultaron interesantes para profundizarse, sin embargo no pudieron ser atendidas debido al alcance de esta tesis. Algunas de estas ideas se describen a continuación como líneas de trabajo futuro:

- *Evaluación del prototipo en un caso de estudio en el ambiente de DSD.* Se plantea realizar un estudio que permita evaluar el prototipo en un ambiente real de DSD con distintos roles de trabajo. Con base en el conjunto inicial de resultados del estudio realizado por Palacio [72], se pudiera proponer un estudio que permitiría conocer el valor de las recomendaciones de colaboradores potenciales identificados mediante el modelo de métricas propuesto. Además se pudieran generar resultados para conocer los efectos de las recomendaciones en diferentes situaciones de ayuda.
- *Integrar el modelo de métricas para identificación de colaboradores potenciales al prototipo de Esferas de Trabajo Colaborativas (CWS-IM).* El prototipo CWS-IM [72] ofrece elementos de información relacionados con la actividad de los desarrolladores en un tiempo dado. Por ello, se plantea la idea de integrar a éste las métricas del modelo propuesto. De esta manera, incorporando y presentando información de conciencia sobre el Involucramiento, la Experiencia y la Disponibilidad de los desarrolladores, se podrá identificar un grupo de colaboradores potenciales que no sólo beneficie a la persona que desea iniciar colaboración, sino que también beneficie al permitir escoger a la persona más indicada para colaborar.
- *Evaluación del prototipo en un caso de estudio por desarrolladores co-localizados y*

distribuidos en términos de la “sorpresa cultural”. Se plantea realizar una evaluación, que permita investigar la influencia de las sorpresas derivadas de las diferencias culturales sobre los criterios de identificación de colaboradores potenciales. Con esto, se pudieran generar resultados para conocer el efecto de las “sorpresas culturales” [86] en el comportamiento para la toma de decisiones de a quién contactar en situaciones de ayuda. Esto daría una mejor perspectiva para determinar si la información proporcionada es suficiente para determinar colaboradores potenciales en DSD.

Bibliografía

- [1] Clarence A Ellis, Simon J Gibbs, and Gail Rein. Groupware: some issues and experiences. *Commun. ACM*, 34(1):39–58, January 1991.
- [2] Alberto L Morán. *Potential and Actual Collaboration Support in the design of distributed Groupware*. Phd thesis, Institut National Polytechnique de Grenoble, 2005.
- [3] Werner Geyer, Heather Richter, Ludwin Fuchs, Tom Fraunhofer, Shahrokh Daijavad, and Steven Poltrock. A team collaboration space supporting capture and access of virtual meetings. *Managing*, pages 188–196, 2001.
- [4] Mark Handel and James D. Herbsleb. What is chat doing in the workplace? *Proceedings of the 2002 ACM conference on Computer supported cooperative work - CSCW '02*, page 1, 2002.
- [5] A Mockus and J D Herbsleb. Expertise Browser: a quantitative approach to identifying expertise. *Proceedings of the 24th International Conference on Software Engineering ICSE 2002*, pages 503–512, 2002.
- [6] Anita Sarma. Palantír : Increasing Awareness in Distributed Software Development. *International Workshop in Global Software Development at ICSE2002*, 2002.
- [7] Kevin A Schneider, Carl Gutwin, Reagan Penner, and David Paquette. Mining a Software Developer 's Local Interaction History. *Development*, pages 5–9.

- [8] D Sundmark, A Moller, and M Nolin. Towards synchronous collaborative software engineering, 2004.
- [9] Susan R Fussell, Sara Kiesler, Leslie D Setlock, Peter Scupelli, and Suzanne Weisband. Effects of Instant Messaging on the Management of Multiple Project Trajectories. *Management Information Systems*, 2004.
- [10] Xiaodan Song, Ching-Yung Lin, Belle L Tseng, and Ming-Ting Sun. Modeling and predicting personal information dissemination behavior. *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining KDD 05*, page 479, 2005.
- [11] Jacky Estublier and Sergio Garcia. Process model and awareness in SCM. In *Proceedings of the 12th international workshop on Software configuration management SCM 05*, volume 12 of *SCM '05*, pages 59–74. ACM Press, 2005.
- [12] Xiaodan Song, Belle L Tseng, Ching-yung Lin, and Ming-ting Sun. Expertisenet: Relational and evolutionary expert modeling. *User Modeling 2005*, 3538:99–108, 2005.
- [13] N Yankelovich JM DiMicco. Constellation: Using Visualization to Find the Path to Experts. 2007.
- [14] Jacob T Biehl, Mary Czerwinski, Greg Smith, and George G Robertson. FAST-Dash: a visual dashboard for fostering awareness in software teams. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, volume San Jose, of *CHI '07*, pages 1313–1322. ACM, 2007.
- [15] Omar Alonso, Premkumar T Devanbu, and Michael Gertz. Expertise identification and visualization from CVS. *Proceedings of the 2008 international workshop on Mining software repositories MSR 08*, page 125, 2008.

- [16] Pei Feng Xiang, Annie T T Ying, Ping Cheng, Ya Bin Dang, Kate Ehrlich, Mary E Helander, Paul M Matchen, Andrew Sempere, Peri L Tarr, Clay Williams, and Shun Xiang Yang. Ensemble: a recommendation tool for promoting communication in software teams. *Proceedings of the 2008 international workshop on Recommendation systems for software engineering*, page 2, 2008.
- [17] Jacob T Biehl, William T Baker, Brian P Bailey, Desney S Tan, Kori M Inkpen, and Mary Czerwinski. IMPROMPTU : A New Interaction Framework for Supporting Collaboration in Multiple Display Environments and Its Field Evaluation for Co-located Software Development. *Interfaces*, April 5-10:939–948, 2008.
- [18] Ching-Yung Lin, Kate Ehrlich, Vicky Griffiths-Fisher, and Christopher Desforges. SmallBlue: People Mining for Expertise Search. *Ieee Multimedia*, 15(1):78–84, 2008.
- [19] Anita Sarma, Larry Maccherone, Patrick Wagstrom, and James Herbsleb. Tesseract: Interactive visual exploration of socio-technical relationships in software development. In *2009 IEEE 31st International Conference on Software Engineering*, ICSE '09, pages 23–33. IEEE Computer Society, 2009.
- [20] Mikkel R Jakobsen, Roland Fernandez, Mary Czerwinski, Kori Inkpen, Olga Kulyk, and George G Robertson. WIPDash : Work Item and People Dashboard for Software Development Teams. *Ifip International Federation For Information Processing*, (1):791–804, 2009.
- [21] Ramón R Palacio, Alberto L Morán, Aurora Vizcaíno, Víctor M González, D E S Navojoa, Facultad De Ciencias, and División De Ingeniería. Knowledge Flow as Facilitator for Getting into Collaboration in Distributed Software Development. *2011 44th Hawaii International Conference on System Sciences*, pages 1–10, 2011.
- [22] Michele Lanza, Lile Hattori, and Anja Guzzi. Supporting Collaboration Awareness with Real-Time Visualization of Development Activity, 2010.

- [23] Stephan Salinger, Christopher Oezbek, Karl Beecher, and Julia Schenk. Saros: An eclipse plug-in for distributed party programming. In *Proceedings International Conference on Software Engineering*, pages 48–55, 2010.
- [24] Andrew Begel and Thomas Zimmermann. Codebook : Discovering and Exploiting Relationships in Software Repositories. *Architecture*, 1(April):125–134, 2010.
- [25] Andrew Begel and Thomas Zimmermann. Keeping up with your friends: function Foo, library Bar.DLL, and work item 24. *International Conference on Software Engineering*, pages 20–23, 2010.
- [26] Gabriela Noemi Aranda, Aurora Vizcaíno, José Luís Hernández, Ramón R Palacio, and Alberto L Morán. Trusty: a tool to improve communication and collaboration in DSD. In *Proceedings of the 17th international conference on Collaboration and technology*, CRIWG’11, pages 224–231, Berlin, Heidelberg, 2011. Springer-Verlag.
- [27] Robert E Kraut and Lynn A Streeter. Coordination in software development. *Commun. ACM*, 38(3):69–81, March 1995.
- [28] J. a. Espinosa, S. a. Slaughter, R. E. Kraut, and J. D. Herbsleb. Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development. *Organization Science*, 18(4):613–630, July 2007.
- [29] J Espinosa, Sandra Slaughter, Robert Kraut, and James Herbsleb. Team Knowledge and Coordination in Geographically Distributed Software Development. *Journal of Management Information Systems*, 24(1):135–169, 2007.
- [30] E A Isaacs, S Whittaker, D Frohlich, and B O’Conaill. Informal communication re-examined: New functions for video in supporting opportunistic encounters. In K E Finn, A J Sellen, and S B Wilbur, editors, *VideoMediated Communication*, pages 459–485. Lawrence Erlbaum, 1997.

- [31] Márcio Kuroki Gonçalves, Leidson R. B. de Souza, and Víctor M González. Collaboration, Information Seeking and Communication: An Observational Study of Software Developers' Work Practices. *Journal of Universal Computer Science*, 17(14):1913–1930, 2011.
- [32] Hugo Fuks, Alberto Raposo, Marco A. Gerosa, Mariano Pimental, and Carlos J.P. Lucena. The 3C Collaboration Model, 2007.
- [33] Peter Busch and Debbie Richards. Graphically defining articulable tacit knowledge, 2003.
- [34] J D Herbsleb and R E Grinter. Architectures, coordination, and distance: Conway's law and beyond. *IEEE Software*, 16(5):63–70, 1999.
- [35] Patricia Ruma Spence and Madhu Reddy. Beyond Expertise Seeking: A Field Study of the Informal Knowledge Practices of Healthcare IT Teams. *Comput. Supported Coop. Work*, 21(2-3):283–315, June 2012.
- [36] Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. *Proceedings of the 1992 ACM conference on Computersupported cooperative work CSCW 92*, 3(November):107–114, 1992.
- [37] Igor Steinmacher, Ana Paula Chaves, and Marco Aurélio Gerosa. Awareness Support in Distributed Software Development : A Systematic Review and Mapping of the Literature. *Computer Supported Cooperative Work*, 2012.
- [38] Cleidson R B De Souza and David F Redmiles. The Awareness Network, To Whom Should I Display My Actions? And, Whose Actions Should I Monitor? *IEEE Transactions on Software Engineering*, 37(3):325–340, 2011.

- [39] O. Robles A.L. Morán, J. Favela, R. Romero, H. Natsu, C. Perez and A. Martínez. Potential and Actual Collaboration Support for Distributed Pair-Programming. *Computación y Sistemas Vol. II, No. 3*, pages 211–229, 2008.
- [40] David W McDonald. Evaluating expertise recommendations. *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work GROUP 01*, page 214, 2001.
- [41] David W McDonald and Mark S Ackerman. Expertise recommender: a flexible recommendation system and architecture. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, volume Philadelph of CSCW '00, pages 231–240. ACM New York, NY, USA, 2000.
- [42] C C Da Trindade, Y A M Barbosa, A K O Moraes, J O De Albuquerque, and S R De Lemos Meira. An Expert Recommender System to Distributed Software Development: Requirements, Project and Preliminary Results, 2009.
- [43] David Mattox, Mark T Maybury, and Daryl Morey. Enterprise expert and knowledge discovery. In *Proceedings of the HCI International '99 (the 8th International Conference on Human-Computer Interaction) on Human-Computer Interaction: Communication, Cooperation, and Application Design-Volume 2 - Volume 2*, pages 303–307, Hillsdale, NJ, USA, 1999. L. Erlbaum Associates Inc.
- [44] Yunwen Ye, Yasuhiro Yamamoto, and Kumiyo Nakakoji. A socio-technical framework for supporting programmers. *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering ESECFSE 07*, 3(1):351–360, 2007.
- [45] Jun Zhang, Mark S Ackerman, Lada Adamic, and Kevin Kyung Nam. QuME: a mechanism to support expertise finding in online help-seeking communities. In

- UIST 07 Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 111–114. ACM, 2007.
- [46] Shawn Minto and Gail C Murphy. Recommending Emergent Teams. *Fourth International Workshop on Mining Software Repositories MSR07ICSE Workshops 2007*, (Section 5):5–5, 2007.
- [47] Marcelo Cataldo, Patrick A Wagstrom, James D Herbsleb, and Kathleen M Carley. Identification of coordination requirements: implications for the Design of collaboration and awareness tools. *Evolution*, pages 353–362, 2006.
- [48] Alan Moraes, Eduardo Silva, Cleyton Da Trindade, Yuri Barbosa, and Silvio Meira. Recommending experts using communication history. *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering RSSE 10*, pages 41–45, 2010.
- [49] Ryen W White, Matthew Richardson, and Yandong Liu. Effects of community size and contact rate in synchronous social q&a. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’11, pages 2837–2846, New York, NY, USA, 2011. ACM.
- [50] Rashmi Sinha and Kirsten Swearingen. The role of transparency in recommender systems. *CHI 02 extended abstracts on Human factors in computing systems CHI 02*, page 830, 2002.
- [51] Svetlana Yarosh, Tara Matthews, and Michelle Zhou. Asking the Right Person : Supporting Expertise Selection in the Enterprise. *Search*, (2):2247–2256, 2012.
- [52] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*, volume 1st of *The Morgan Kaufmann Series in Interactive Technologies*. Morgan Kaufmann, 1999.

- [53] Cleidson R De Souza, Stephen Quirk, Erik Trainer, and David F Redmiles. Supporting collaborative software development through the visualization of socio-technical dependencies. *Proceedings of the 2007 international ACM conference on Conference on supporting group work GROUP 07*, page 147, 2007.
- [54] John M Carroll, editor. *Scenario-based design: envisioning work and technology in system development*. John Wiley & Sons, Inc., New York, NY, USA, 1995.
- [55] J D Herbsleb. Global Software Engineering: The Future of Socio-technical Coordination, 2007.
- [56] J D Herbsleb, D J Paulish, and M Bass. Global software development at siemens: experience from nine projects. *Proceedings 27th International Conference on Software Engineering 2005 ICSE 2005*, pages 524–533, 2005.
- [57] James D Herbsleb and Deependra Moitra. Guest Editors’ Introduction: Global Software Development. *IEEE Softw.*, 18(2):16–20, March 2001.
- [58] C R B De Souza and D F Redmiles. An empirical study of software developers’ management of dependencies and changes, 2008.
- [59] C R B De Souza, D Redmiles, G Mark, J Penix, and M Sierhuis. Management of interdependencies in collaborative software development. In *2003 International Symposium on Empirical Software Engineering 2003 ISESE 2003 Proceedings*, pages 294–303. IEEE Comput. Soc, 2003.
- [60] Marcelo Cataldo, James D Herbsleb, and Kathleen M Carley. Socio-Technical Congruence : A Framework for Assessing the Impact of Technical and Work Dependencies on Software Development Productivity. *Building*, pages 2–11, 2008.

- [61] Andrew J Ko, Robert DeLine, and Gina Venolia. Information Needs in Collocated Software Development Teams. *29th International Conference on Software Engineering ICSE07*, 355(9197):344–353, 2007.
- [62] Jianguo Wang and Anita Sarma. Which Bug Should I Fix : Helping New Developers Onboard a New Project. *Search*, 11:76–79, 2011.
- [63] Thomas Fritz, Gail C Murphy, and Emily Hill. Does a programmer’s activity indicate knowledge of code? In *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering ESECFSE 07*, ESEC-FSE ’07, pages 341–350. ACM, 2007.
- [64] N Sadat Shami, Kate Ehrlich, Geri Gay, and Jeffrey T Hancock. Making sense of strangers’ expertise from signals in digital artifacts. *Proceedings of the 27th international conference on Human factors in computing systems CHI 09*, page 69, 2009.
- [65] John Anvik and Gail C Murphy. Determining Implementation Expertise from Bug Reports. *Fourth International Workshop on Mining Software Repositories MSR07ICSE Workshops 2007*, pages 2–2, 2007.
- [66] Miryung Kim. An exploratory study of awareness interests about software modifications. In *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE ’11, pages 80–83, New York, NY, USA, 2011. ACM.
- [67] Kjeld Schmidt. The Trouble with ‘Tacit Knowledge. *Computer Supported Cooperative Work*, 21(2-3):163–225, 2012.

- [68] Erik H Trainer, Ban Al-Ani, and David F Redmiles. Impact of collaborative traces on trustworthiness. *Proceeding of the 4th international workshop on Cooperative and human aspects of software engineering CHASE 11*, 17(4):40, 2011.
- [69] David W McDonald and Mark S Ackerman. Just Talk to Me: A Field Study of Expertise Location. In *Proceedings Of The ACM Conference On ComputerSupported Cooperative Work CSCW 98*, volume 98, pages 315–324. ACM Press, ACM, 1998.
- [70] Erik H Trainer and David F Redmiles. Foundations for the design of visualizations that support trust in distributed teams. In *Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '12*, pages 34–41, New York, NY, USA, 2012. ACM.
- [71] Ban Al-Ani, Erik Trainer, David Redmiles, and Erik Simmons. Trust and surprise in distributed teams: towards an understanding of expectations and adaptations. In *Proceedings of the 4th international conference on Intercultural Collaboration, ICIC '12*, pages 97–106, New York, NY, USA, 2012. ACM.
- [72] Ramón R Palacio. *Soporte a la entrada en colaboración en el desarrollo distribuido de software mediante esferas de trabajo colaborativas*. Phd thesis, Universidad Autónoma de Baja California, 2010.
- [73] Victor M González, Leonardo Galicia, and Jesús Favela. Supporting the Planning and Organization of Multiple Activities in the Workplace. *Ifip International Federation For Information Processing*, 266(1415):235–238, 2007.
- [74] Erik H Trainer and David F Redmiles. A Survey of Visualization Tools that Promote Awareness of Software Development Activities. Technical report, University of California, Irvine, 2009.

- [75] Jean-Daniel Fekete, Jarke J Van Wijk, John T Stasko, and C North. The Value of Information Visualization. *Information Visualization*, 4950:1–18, 2008.
- [76] Jeffrey Heer and Ben Shneiderman. Interactive dynamics for visual analysis. *Commun. ACM*, 55(4):45–54, April 2012.
- [77] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering the Information Age Solving Problems with Visual Analytics*. 2010.
- [78] Rafael Veras Guimares, Anderson Gregorio Marques Soares, Nikolas Jorge Santiago Carneiro, Aruanda Sim X F Es Meiguins, and Bianchi Serique Meiguins. Design Considerations for Drill-down Charts, 2011.
- [79] Geoffrey M Draper, Yarden Livnat, and Richard F Riesenfeld. A survey of radial methods for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):759–776, 2009.
- [80] User Interface and Design Rules. *Designing with the Mind in Mind*, volume 36. Morgan Kaufmann, 2010.
- [81] Debbie Stone, Caroline Jarrett, Mark Woodroffe, and Shailey Minocha. *User Interface Design and Evaluation (The Morgan Kaufmann Series in Interactive Technologies) (The Morgan Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- [82] Kayur Patel, Naomi Bancroft, Steven M Drucker, James Fogarty, Andrew J Ko, and James A Landay. Gestalt : Integrated Support for Implementation and Analysis in Machine Learning. In *October*, UIST 2010, pages 37–46. ACM Press, 2010.
- [83] Mikkel Rønne Jakobsen and Kasper Hornbæk. Fisheye interface: research problems and practical challenges. In *Proceedings of the Second IFIP WG 13.7 con-*

- ference on Human-computer interaction and visualization*, HCIV'09, pages 76–91, Berlin, Heidelberg, 2011. Springer-Verlag.
- [84] John Brooke. SUS - A quick and dirty usability scale. *Usability evaluation in industry*, 189:189–194, 1996.
- [85] M A Storey, C Bennett, R I Bull, and D M German. Remixing visualization to support collaboration in software maintenance, 2008.
- [86] Judith S Olson and Gary M Olson. Culture Surprises in Remote Software Development Teams. *Queue*, 1(9):52–59, 2003.

Apéndice A

Cuestionario escala 7-Likert



Entrevista guiada por cuestionario

1. Pienso que fue sencillo el poder identificar el nivel de involucramiento de los colegas

[]	[]	[]	[]	[]	[]	[]
Completamente en desacuerdo	En desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	De acuerdo	Completamente de acuerdo

2. Pienso que fue sencillo el poder identificar el perfil de los colegas

[]	[]	[]	[]	[]	[]	[]
Completamente en desacuerdo	En desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	De acuerdo	Completamente de acuerdo

3. Pienso que fue claro el identificar el estado actual y ubicación geográfica de los colegas

[]	[]	[]	[]	[]	[]	[]
Completamente en desacuerdo	En desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	De acuerdo	Completamente de acuerdo

4. Pienso que pude generar rápidamente la combinación de criterios para la identificación de colaboradores potenciales

[]	[]	[]	[]	[]	[]	[]
Completamente en desacuerdo	En desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	De acuerdo	Completamente de acuerdo

5. Pienso que puedo identificar fácilmente al colega que me puede ayudar a resolver una duda o problema, realizándolo a través de este sistema.

[]	[]	[]	[]	[]	[]	[]
Completamente en desacuerdo	En desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	De acuerdo	Completamente de acuerdo

6. Pienso que puedo identificar correctamente al colega que me puede ayudar a resolver una duda o problema, realizándolo a través de este sistema.

[]	[]	[]	[]	[]	[]	[]
Completamente en desacuerdo	En desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	De acuerdo	Completamente de acuerdo

7. Considero satisfactorio el resultado final, y mi selección del colega que me puede apoyar en resolver una duda o problema.

[]	[]	[]	[]	[]	[]	[]
Completamente en desacuerdo	En desacuerdo	Ligeramente en desacuerdo	Neutral	Ligeramente de acuerdo	De acuerdo	Completamente de acuerdo

Comentarios y sugerencias:

Apéndice B

SUS, System Usability Scale



CUESTIONARIO SUS

Escala de facilidad de uso del sistema Instrucciones: Para cada uno de los enunciados marca con una cruz el recuadro que mejor describa tu reacción al Sistema que utilizaste. Asume que dicho Sistema (prototipo) será desarrollado como un producto funcional.

	Completamente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Completamente de acuerdo
1. Creo que me gustaría usar este sistema a menudo.	☐	☐	☐	☐	☐
2. El sistema me pareció innecesariamente complejo.	☐	☐	☐	☐	☐
3. Pensaba que el sistema era sencillo de utilizar.	☐	☐	☐	☐	☐
4. Creo que necesitaría la ayuda de un técnico para poder usar este sistema.	☐	☐	☐	☐	☐
5. Me pareció que las distintas funciones de este sistema estaban bien integradas.	☐	☐	☐	☐	☐
6. Me pareció que había demasiadas incoherencias en este sistema.	☐	☐	☐	☐	☐
7. Creo que casi todo el mundo aprendería a usar este sistema rápidamente.	☐	☐	☐	☐	☐
8. Me pareció que el sistema es muy complicado de usar.	☐	☐	☐	☐	☐
9. Me encontré muy a gusto utilizando el sistema.	☐	☐	☐	☐	☐
10. Tuve que aprender muchas cosas antes de poder hacerme con este sistema.	☐	☐	☐	☐	☐