



Universidad Veracruzana

Facultad de Estadística e Informática

---

---

Síntesis Narrativa del trabajo de investigación intitulado:  
“Análisis de la investigación en el uso de Bots  
para el desarrollo de software”

Presentan:  
Ricardo Moguel Sánchez  
César Sergio Martínez Palacios

Directores:  
Dr. Jorge Octavio Ocharán Hernández  
Dr. Héctor Xavier Limón Riaño

Xalapa, Ver. a 12 de Enero del 2022

## 1. Resumen

Dentro de la ingeniería de software se han aplicado diferentes herramientas a cada una de la etapa que existen en el desarrollo para así lograr que el software cumpla con una alta calidad. Una de estas herramientas que se ha adaptado para tener aplicaciones a las actividades del desarrollo de software son los bots. Aun cuando existen diferentes trabajos sobre bots, no se conoce de alguno que describa y analice la investigación en el uso de *bots* para el desarrollo de software. En este documento se reporta un mapeo sistemático por medio de una síntesis narrativa donde se tiene como objetivo el análisis del estado actual de las propuestas para el uso de bots en el desarrollo de software. Se obtuvieron 30 estudios primarios para esta revisión dentro del periodo 2011-2020, reportando 32 diferentes bots que se aplican a las actividades del desarrollo de software. En conclusión, se abarco un reducido numero de estudios en este mapeo, identificando que el uso de bots en el desarrollo de software aun es un tema que aún se encuentra en pleno despegue.

Palabras clave: Bots, development bots, desarrollo de software, mapeo sistemático de la literatura, síntesis narrativa

## 2. Introducción

Dentro de la ingeniería de software se ha buscado utilizar diferentes técnicas para poder aplicar a cada una de la etapa que existen en el desarrollo para así lograr que el software salga con una calidad esperada y cumpliendo los requisitos solicitados. Esto ha ocasionado una tendencia de la evolución del software a sistemas cada vez más complejos y por ende requieren de mucho más trabajo. Debido a esto la industria está en constante búsqueda de nuevas técnicas y mejoras para aplicarlos a la ingeniería de software con el fin de agilizar el proceso de desarrollo de software y reducir los tiempos de desarrollo desde elicitación de requisitos para una aplicación nueva hasta la documentación de un sistema final.

Con estos antecedentes es que han surgido múltiples herramientas tecnológicas con el objetivo de apoyar a un desarrollador en su trabajo. Y además se han adaptado otro tipo de softwares para que su objetivo principal ahora sea el anteriormente mencionado. Han nacido tecnologías utilizadas en el desarrollo de software bajo el termino Inteligencia Artificial, las cuales puede incluir Machine Learning, Procesadores de Lenguaje Natural y Redes Neuronales. Similar a Inteligencia Artificial, una tecnología emergente que nació con un objetivo diferente al del cual hablamos y se ha logrado adaptar para aplicar al desarrollo de software con el fin de optimizar el proceso de desarrollo son los Bots.

La definición de un bot puede llegar a ser muy ambigua como menciona Lebeuf et al. (2018):

“El término bot varía desde describir scripts simples que automatizan una tarea en segundo plano, hasta aplicaciones complejas que interactúan con uno o más humanos y se adaptan de manera autónoma a las actividades que realizan las personas y otros sistemas, e incluso hasta aplicaciones de software que utilizan la inteligencia artificial para imitar el comportamiento y la inteligencia humanos.”

Analizando las palabras de Lebeuf se puede notar que los bots tienen múltiples objetivos y claramente diferentes usos y aplicaciones para estos bots. Basado en esto es que nacen múltiples tipos de bots y etiquetas como lo son: Bots Informativos, Game Bots, Spam Bots, ChatBots, y DevBots.

Los DevBots tienen como enfoque solucionar o apoyar en problemas que existen en el desarrollo del software. Como puede serlo las malas prácticas de codificación, la falta de productividad en el desarrollador, la poca o nula comunicación entre equipos de trabajo, entre otros. Cabe aclarar que de un DevBot se busca un apoyo al desarrollador mas no reemplazarlo, ya que muchas personas logran confundir estos objetivos y pierden el enfoque a la hora de crear un DevBot.

### 3. Trabajo Relacionado

Dentro de este capítulo se hablará de los diferentes trabajos identificados que se relacionen a los bots y el trabajo que se desea realizar. Así para identificar el nivel de literatura que hay sobre los bots y motivar la realización del trabajo.

#### 3.1. Revisiones y mapeos sistemáticos relacionados

Para realizar una búsqueda exhaustiva y completa de trabajos relacionados se elaboraron cadenas de búsquedas con los conceptos importantes de bots en desarrollo y sus sinónimos. La cadena de búsqueda fue adaptada para su uso en Web Of Science y Scopus, indexadores de alta credibilidad que agrupan trabajos de investigación revisados por pares. Para identificar trabajos relacionados se introdujo la siguiente cadena de búsqueda en Scopus:

```
(("bot" OR "DevBot" OR "bots" OR "devbots" ) AND ("systematic literature review" OR "systematic review" OR "secondary study" OR "mapping study" OR "systematic mapping study" OR "literature review"))
```

Para Web Of Science se introdujo una cadena similar ajustada:

```
(("bot" OR "devbot" OR "bots" OR "devbots") AND ("systematic literature review" OR "systematic review" OR "secondary study" OR "mapping study" OR "systematic mapping study" OR "literature review"))
```

De la búsqueda de trabajos, un total de once trabajos se enfocan en la detección o descripción de bots sociales o maliciosos. En estos trabajos podemos identificar que se tiene la definición de un bot como una identidad falsa de redes sociales que solo infla números en estas. Esto es algo crítico por que el término bot se gana una mala fama y ocasiona cierto rechazo a la sociedad general.

A continuación, en la Tabla 1 se muestran los 11 trabajos relacionados:

Tabla 1 Trabajos relacionados

Numero:	Estudio
RW1	Kotkov, D., Pandey, G., & Semenov, A. (2018, December). Gaming bot detection: a systematic literature review. In <i>International Conference on Computational Social Networks</i> (pp. 247-258).
2	Stieglitz, S., Brachten, F., Ross, B., & Jung, A. K. (2017). Do social bots dream of electric sheep? A categorisation of social media bot accounts.
3	Orabi, M., Mouheb, D., Al Aghbari, Z., & Kamel, I. (2020). Detection of bots in social media: A systematic review.
4	de Morais, D. M. G., & Digiampietri, L. A. (2021, June). Methods and Challenges in Social Bots Detection: A Systematic Review.
5	Alothali, E., Zaki, N., Mohamed, E. A., & Alashwal, H. (2018, November). Detecting social bots on Twitter: A literature review.
6	Wiesenberg, M., & Tench, R. (2020). Deep strategic mediatization: Organizational leaders' knowledge and usage of social bots in an era of disinformation.
7	Collins, B., Hoang, D. T., Dang, D. T., & Hwang, D. (2020, November). Method of Detecting Bots on Social Media. A Literature Review.
8	Providel, E., & Mendoza, M. (2021). Misleading information in Spanish: a survey. <i>Social Network Analysis and Mining</i> .
9	Paul, H., & Nikolaev, A. (2021). Fake review detection on online E-commerce platforms: a systematic literature review.
10	Wang, P., Wu, L., Aslam, B., & Zou, C. C. (2009, August). A systematic study on peer-to-peer botnets.
11	Ben Sassi, I., & Ben Yahia, S. (2021). Malicious accounts detection from online social networks: a systematic review of literature.

Se encontraron también trabajos sobre la aplicación de bots, aunque no en el área de ingeniería de software: encontrados incluyen cinco estudios. Estos cinco estudios investigan los chatbots en particular y se empieza a ver cómo, aunque sea mínima, hay una tendencia hacia el trabajo con la ayuda de bots en el futuro. Por medio de investigaciones bien realizadas y explicando diferentes puntos de vista. De estos, dos hablan de chatbots en general, uno se enfoca en la educación, uno en reuniones empresariales y uno en el dominio de medicina.

A continuación, en la Tabla 2 de muestran los 5 trabajos relacionados a la aplicación de bots:

Tabla 2 Trabajos relacionados a la aplicación de bots

Numero:	Enfoque	Referencia
RW12	Educación	Pérez, J. Q., Daradoumis, T., & Puig, J. M. M. (2020). Rediscovering the use of chatbots in education: A systematic literature review.
13	Chatbots	Jung, H., Lee, J., & Park, C. (2020). Deriving Design Principles for Educational Chatbots from Empirical Studies on Human–Chatbot Interaction.
14	Medicina	Bhirud, N., Tataale, S., Randive, S., & Nahar, S. (2019). A Literature Review On Chatbots In Healthcare Domain.
15	Empresarial	Syvänen, S., & Valentini, C. (2020). Conversational agents in online organization–stakeholder interactions: a state-of-the-art analysis and implications for further research.
16	Chatbot	Borsci, S., Malizia, A., Schmettow, M., Van Der Velde, F., Tariverdiyeva, G., Balaji, D., & Chamberlain, A. (2021). The Chatbot Usability Scale: the Design and Pilot of a Usability Scale for Interaction with AI-Based Conversational Agents.

Aparte de estudios enfocándose a bots, existen estudios secundarios sobre la Inteligencia Artificial (IA). En donde se mencionan bots aplicados a la ingeniería de software, es un poco parecido al anterior punto, pero este ya no está enfocado en bots si no que tiene un discurso más amplio a la ciencia de la inteligencia artificial. Entre estos se incluyen cinco estudios sobre usos de IA fuera del área de Ingeniería de Software (IS). Finalmente, se identificaron dos estudios sobre la aplicación de bots para la automatización de procesos de negocio en empresas. A continuación, en la Tabla 3 se muestran los 5 trabajos relacionados sobre la IA

Tabla 3 Trabajos relacionados sobre la inteligencia artificial

Numero:	Referencia
RW1 7	Kaartemo, V., & Helkkula, A. (2018). A systematic review of artificial intelligence and robots in value co-creation: current status and future research avenues.
18	Ciupe, A., Meza, S., & Orza, B. (2020, September). Systematic Assessment of Interactive Instructional Technologies in Higher Engineering Education.
19	Orgeolet, L., Foulquier, N., Misery, L., Redou, P., Pers, J. O., Devauchelle-Pensec, V., & Saraux, A. (2020). Can artificial intelligence replace manual search for systematic literature? Review on cutaneous manifestations in primary Sjögren’s syndrome.
20	Skinner, G., & Walmsley, T. (2019, February). Artificial intelligence and deep learning in video games a brief review.
21	Blitch, J. G. (1996). Artificial intelligence technologies for robot assisted urban search and rescue.

Revisiones o Mapeos Sistemáticos específicamente sobre bots en el desarrollo de software, de lo que investigamos, no existen. Aun así, existen investigaciones del uso de bots en otras disciplinas fuera de la IS y existen revisiones sistemáticas del uso de inteligencia artificial en el desarrollo de software. Las revisiones posteriores son importantes debido a que los bots son una especie de inteligencia artificial como menciona Lebeuf *et al.* (2019) “El término bot varía desde describir scripts simples que automatizan una tarea en segundo plano, hasta aplicaciones complejas que interactúan con uno o más humanos y se adaptan de manera autónoma a las actividades que realizan las personas y otros sistemas”.

Algunas revisiones sistemáticas de literatura (RSLs) con algunas similitudes se encontraron. Los trabajos con mayor semejanza son mostrados a continuación. El RSL “Systematic literature review: Teaching novices programming using robots” en donde se investigan bots utilizados para asistir a programadores novatos. (Major *et al.*, 2012)

Aunque es parecido, el enfoque de este trabajo es preciso y en este trabajo se busca ampliar el conocimiento en el uso de bots en diferentes tareas de desarrollo y no limitarnos a un caso en particular. Un trabajo de RSL con un área de cobertura más cercano al nuestro es el trabajo “Rediscovering the use of chatbots in education: A systematic literature review” en donde se estudia diversos usos de chatbots en el área de educación. (Perez *et al.*, 2020) Cabe mencionar que chatbots cae dentro de los bots de desarrollo utilizados en la industria de informática. Afortunadamente, un estudio secundario con un alcance parecido al área de desarrollo “Ways of Applying Artificial Intelligence in Software Engineering” en la forma de un RSL que analiza literatura detallando usos de IA en la ingeniería de software. (Feldt *et al.*, 2018)

A pesar de ser similar, este trabajo es diferente en que se especificó un solo aspecto de la ingeniería de software, el desarrollo, y una sola rama de la IA, las tecnologías bot. Por ende, de acuerdo con lo investigado, podemos concluir que no existe ningún trabajo de investigación dedicado a la recopilación de información sobre el uso práctico de bots en actividades de desarrollo de software.

## 4. Método

Para alcanzar el objetivo de este trabajo de investigación - identificar investigaciones que reporten el uso de *bots* para el desarrollo de software - se elaboró un Mapeo Sistemático de la Literatura (MSL) basado en el proceso de producción de revisiones sistemáticas de Kitchenham *et al.* (2015). Para este trabajo, primero se elaboraron preguntas de investigación y después se seleccionó una estrategia de búsqueda de estudios primarios. Posteriormente, se ejecutó la búsqueda de estudios, se condujo una evaluación de calidad de los estudios y se extrajeron los datos de los estudios. Finalmente, se realizó una síntesis narrativa con los resultados de la extracción de datos para responder a las preguntas de investigación plasmadas al inicio del proceso de producción del MSL.

### 4.1 Preguntas de investigación

Primero, se especificaron preguntas que se pretenden responder en el trabajo por medio de una investigación de posibles estudios primarios. Las preguntas de investigación fueron refinadas conforme se realizó el proceso de búsqueda para asegurar que puedan ser respondidas adecuadamente con la literatura obtenida.

Las preguntas iniciales fueron elaboradas durante el 30 de septiembre del 2021, pero fueron gradualmente ajustadas durante el curso de la selección de estudios primarios debido a que una pregunta no estaba lo suficientemente asociada a los datos extraídos de los estudios seleccionados para ser satisfecha. Entonces, armados con unas preguntas de investigación, se puede identificar el tipo de estudio secundario a realizar. En la siguiente lista se plasman las preguntas de investigaciones del MSL elaborado junto con breves justificaciones de su necesidad de ser respondidas.

#### **PI-1 ¿Cuáles son las actividades de desarrollo en donde se usan bots?**

Consideramos necesario primero identificar en cuales actividades dentro del desarrollo de software los bots están siendo usados.

##### **PI-1.1 ¿Cuál es el objetivo del bot en la actividad de desarrollo usada?**

Adicionalmente, se debe considerar porque se está utilizando en una etapa y como es de ayuda para un desarrollador.

##### **PI-1.2 ¿Cuáles son las actividades de desarrollo en donde más se reporta el uso de bots?**

También, se piensa agrupar a los bots por el tipo de actividad en el que se utilizan para saber si existen actividades de desarrollo en donde se aprovecha más de los bots. Al responder esta pregunta se podrá identificar áreas del desarrollo en donde no se utilizan bots para trabajos futuros.



## **PI-2 ¿Cuáles son los beneficios reportados del bot aplicado a una actividad de desarrollo?**

Se busca identificar Para poder informar a desarrolladores de lo que se puede esperar de provecho con el uso de un bot.

## **PI-3 ¿Que problemas se presentan en las actividades del desarrollo que el bot contribuye a su solución?**

Se sabe que nada es perfecto, y que la inclusión de alguna nueva herramienta al equipo de desarrollo incluye nuevos retos. Se requiere saber cuáles son los nuevos problemas o retos que implica el uso del bot en una etapa del desarrollo de software.

## **PI-4 ¿Cuál es el nivel de inteligencia del bot?**

Debido a que el futuro de DevBots está en la integración de Inteligencia Artificial a los Bot para crear asistentes artificiales en vez de herramientas de desarrollo. Para medir si un bot es un asistente artificial se debe categorizar el nivel de inteligencia de un DevBot (Erlenhov *et al.*, 2019). De esta forma se logra identificar el nivel de apoyo que brinda el DevBot dentro de la actividad de la Ingeniería de Software y su nivel de utilidad para la actividad especificada, las características de inteligencia se identifican con tres aspectos. Estos son Autonomía, Razonamiento y Adaptación (C. Lebeuf *et al.*, 2018).

- Adaptación. Un bot es adaptable cuando puede percibir un contexto de proyecto de desarrollo y puede usar ese contexto para cambiar la forma en que interactúan con usuarios.
- Razonamiento. Un bot tiene la capacidad de razonamiento cuando exhibe un comportamiento guiado ya sea por simples reglas lógicas simples u técnicas avanzadas de la IA que impulsan su comportamiento.
- Autonomía. Un bot es autónomo cuando no depende de la participación humana antes de actuar.

## 4.2 Proceso de identificación de estudios

Para la identificación de estudios se ejecutó una búsqueda automatizada siguiendo el método Proceso Sistemático de Búsqueda basado en un Estándar Pseudo-Oro (Zhang et al., 2011). A continuación, se detallan los aspectos de las actividades realizadas durante la selección de estudios primarias.

### 4.2.1 Determinación de las fuentes de datos

El proceso de búsqueda de Zhang comienza con la selección de fuentes. Las fuentes fueron seleccionadas el 10 de octubre del 2021 en donde se realizó una evaluación de motores posibles. Además de los motores seleccionados originalmente, durante el proceso de búsqueda se amplió la búsqueda con la inclusión de adicionales motores de búsqueda para asegurar una búsqueda rigurosa y exhaustiva. Para la búsqueda manual se revisarán las actas de congreso de International Conference on Software Engineering (ICSE) y sus subsidiarios incluyendo desde 2011 a 2020 relacionadas a Bots. Para la búsqueda automatizada se implementará la cadena de búsqueda en las bibliotecas digitales ACM, IEEE Xplore, Springer Link, Science Direct y EBSCO Host. Se seleccionaron los motores de búsqueda principalmente porque son los cuales se otorga acceso completo por medio de CONRICYT disponible a nosotros por medio de la Universidad Veracruzana, además de que la ICSE declara en su sitio web que sus acuerdos de conferencias se encontraran por completo en ACM e IEEE Xplore en particular, lo cual brinda una mayor confianza y estatus de que se encontraran artículos necesarios para la investigación

#### 4.2.2 Definición de la estrategia de búsqueda.

Basándonos en el procedimiento de búsqueda sistemática de H. Zhang et al. (2009), mencionado anteriormente, se realizó un procedimiento de construcción de un Estándar Pseudo-Oro por medio de una búsqueda manual en un conjunto limitado de revistas de

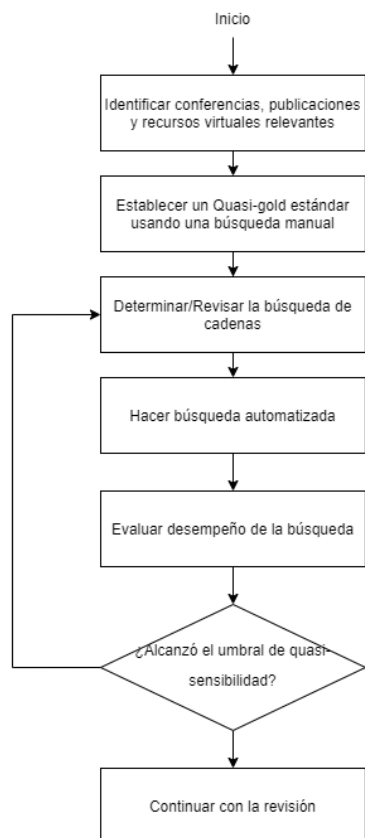


Figura 1 Proceso de obtención de estándar pseudo-oro. Adaptado de Zhang, H., Babar, M. A., & Tell, P. (2011)

temas específicos, el conjunto de artículos relevantes encontrados se utilizó luego para evaluar la integridad de la consecuente búsqueda automatizada, las actividades realizadas durante la búsqueda de estudios primarios se pueden ver en la siguiente Figura:

El Estándar Pseudo-Oro utiliza dos criterios clave para evaluar la completitud de una búsqueda automatizada los cuales son el retorno (Dieste, Grimán & Juristo 2009) y la precisión (Zhang, Babar & Tell 2011). El retorno de una búsqueda es la proporción (o porcentaje) de todos los estudios relevantes que se encuentran en la búsqueda y la precisión de una búsqueda es la proporción (o porcentaje) de los estudios encontrados que son relevantes para las preguntas de investigación que aborda una revisión.

Se resume en las siguientes formulas:

Retorno = Número de estudios relevantes encontrados / Total de estudios relevantes encontrados.

Precisión= Número de estudios relevantes encontrados / Total de estudios encontrados.

### 4.2.3 Desarrollo de la búsqueda de la literatura.

El proceso de búsqueda inicio con la búsqueda manual para como primer paso en el desarrollo de un estándar pseudo-oro. La búsqueda manual fue realizada el 4 y 5 de octubre del 2021. Durante la búsqueda manual se investigaron estudios del congreso de *International Conference on Software Engineering (ICSE)* y sus conferencias hijas desde 2011 a 2020 relacionadas a bots. En la Tabla 4 se resaltan los veintiún artículos relevantes identificados.

Tabla 4 Estudios de la búsqueda manual

Id	Estudio
BM-1	Balachandran, V. (2013, May). Reducing human effort and improving quality in peer code reviews using automatic static analysis and reviewer recommendation.
BM-2	Kim, D., Nam, J., Song, J., & Kim, S. (2013, May). Automatic patch generation learned from human-written patches.
BM-3	Urli, S., Yu, Z., Seinturier, L., & Monperrus, M. (2018, May). How to design a program repair bot? insights from the repairnator project.
BM-4	Bradley, N., Fritz, T., & Holmes, R. (2018, May). Context-aware conversational developer assistants.
BM-5	Brown, C., & Parnin, C. (2019, May). Sorry to bother you: designing bots for effective recommendations.
BM-6	Moreno, D., Duenas, S., Cosentino, V., Fernandez, M. A., Zerouali, A., Robles, G., & Gonzalez-Barahona, J. M. (2019, May). SortingHat: wizardry on software project members.
BM-7	van Tonder, R., & Le Goues, C. (2019, May). Towards s/engineer/bot: Principles for program repair bots.
BM-8	Wyrich, M., & Bogner, J. (2019, May). Towards an autonomous bot for automatic source code refactoring.
BM-9	Erlenhov, L., de Oliveira Neto, F. G., Scandariato, R., & Leitner, P. (2019, May). Current and future bots in software development.
BM-10	Matthies, C., Dobrigkeit, F., & Hesse, G. (2019, May). An additional set of (automated) eyes: Chatbots for agile retrospectives.
BM-11	Subramanian, V., Ramachandra, N., & Dubash, N. (2019, May). TutorBot: contextual learning guide for software engineers.
BM-12	Kumar, R., Bansal, C., Maddila, C., Sharma, N., Martelock, S., & Bhargava, R. (2019, May). Building sankie: An ai platform for devops.
BM-13	Sharma, V. S., Mehra, R., Kaulgud, V., & Podder, S. (2019, May). A smart advisor for software delivery-a bot framework for awareness, alerts and advice.
BM-14	Marginean, A., Bader, J., Chandra, S., Harman, M., Jia, Y., Mao, K., ... & Scott, A. (2019, May). SapFix: Automated end-to-end repair at scale.
BM-15	Paikari, E., Choi, J., Kim, S., Baek, S., Kim, M., Lee, S., ... & Cheong, C. (2019, May). A chatbot for conflict detection and resolution.
BM-16	Wessel, M., Steinmacher, I., Wiese, I., & Gerosa, M. A. (2019, May). Should i stale or should i close? an analysis of a bot that closes abandoned issues and pull requests.
BM-17	Lin, C. T., Ma, S. P., & Huang, Y. W. (2020, June). MSABot: A chatbot framework for assisting in the development and operation of microservice-based systems.

BM-17	Dominic, J., Houser, J., Steinmacher, I., Ritter, C., & Rodeghero, P. (2020, June). Conversational bot for newcomers onboarding to open source projects.
BM-19	Dominic, J., Ritter, C., & Rodeghero, P. (2020, June). Onboarding Bot for Newcomers to Software Engineering.
BM-20	Dey, T., Mousavi, S., Ponce, E., Fry, T., Vasilescu, B., Filippova, A., & Mockus, A. (2020, June). Detecting and characterizing bots that commit code.

De acuerdo con la Etapa 3 del Proceso Sistemático de Búsqueda de Zhang, se elaboraron cadenas de búsqueda en base a los términos de búsqueda previamente definidos. Las tablas de conceptos y términos se encuentran en el Apéndice B: Conceptos y Términos de bots relevantes a bots. Las cadenas de búsqueda fueron ingresadas a los motores de búsqueda IEEE Xplore y ACM Digital Library para evaluar su rendimiento conforme al Quasi-Gold Standard de Zhang. La evaluación del rendimiento se determinó en base a los criterios de Retorno y Precisión previamente definidos conforme a los treinta estudios relevantes resultantes de la búsqueda manual. El Quasi-Gold Standard de Zhang dicta que una cadena de búsqueda es ideal para un Mapeo Sistemático de la Literatura con una precisión de 0.8 o mayor. La cadena con mejor rendimiento fue la cadena con Id-6 con un retorno de 95% y una precisión de 3.6% debido a que se encontraron diecinueve de los veinte relevantes entre 528 estudios identificados por la cadena.

En la Tabla 5 se muestran en detalle los resultados de la evaluación de las cadenas de búsqueda elaboradas conforme el Estándar Pseudo-oro de Zhang ordenadas por Retorno incremental. Se puede ver a detalle todos los resultados de todas las cadenas.

Tabla 5 Cadenas de búsqueda candidatas

Id	Cadena	Estudios encontrados	Estudios relevantes encontrados	Retorno	Precisión
1	("software development" OR "software project") AND (bot OR agent OR assistant)	263	10	0.5	0.0380
2	"software development" AND (bot OR devbot)	107	11	0.55	0.0934
3	"Software development" AND (*bot* OR agent OR assistant) AND (anal* OR cod* OR automati* OR use)	9607	9	0.45	0.0001
4	(Analysis bot OR Chatbot OR Conversational bot OR Conversational developer assistant OR Developer assistant OR Program repair bot OR Repair bot OR Review bot OR Software bot)	1601	13	0.65	0.0081
5	((Analysis bot OR Chatbot OR Conversational bot OR Conversational developer assistant OR Developer assistant OR Repair bot OR automated repair OR Review bot OR Software bot ))	2067	13	0.65	0.0063

6	(Analysis bot OR chatbot OR "conversational bot" OR "conversational developer assistant" OR "developer assistant" OR Repair bot OR automated repair OR review bot OR software bot) AND ("software development" OR "software project" OR "open source project" OR "software engineering" OR "repair")	528	19	0.95	0.036
---	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----	----	------	-------

#### 4.2.4 Selección de estudios relevantes

Basados en los resultados de la búsqueda manual, se seleccionó la cadena:

(Analysis bot OR chatbot OR "conversational bot" OR "conversational developer assistant" OR "developer assistant" OR Repair bot OR automated repair OR review bot OR software bot) AND ("software development" OR "software project" OR "open source project" OR "software engineering" OR "repair")

Con un retorno de 0.8 y una precisión de 0.04. De acuerdo con la metodología de Zhang para un MSL se necesita por lo menos un retorno de 0.8 como en la cadena de búsqueda seleccionada.

Una vez terminada la selección de una cadena de búsqueda, se elaboraron criterios de inclusión y exclusión para identificar estudios para la búsqueda automatizada. En la siguiente Tabla 6 se muestra los criterios de inclusión y exclusión.

Tabla 6 Criterios de selección de estudios

Criterios de Inclusión	Criterios de Exclusión
CI-1 La fecha de publicación es de 2011 a 2020.	CE-1 Estudios duplicados
CI-2 La publicación debe estar en el idioma inglés.	CE-2 La publicación es de un libro.
CI-3 La publicación hace referencia a bots en el desarrollo de software	CE-3 La publicación es una reseña de un estudio.
CI-4 La publicación debe responder al menos una pregunta de investigación.	CE-4 La publicación no pasa la evaluación de calidad

El número de participantes en la selección son dos. Estos dos son los mismos autores del MSL. De acuerdo con la metodología de Kitchenham *et al.* (2015) se descartó literatura en etapas. Para iniciar se ingresó una cadena de búsqueda en cada uno de los motores de búsqueda. En la Tabla 7 se detallan las cadenas ingresadas para la búsqueda automatizada en base a la cadena de búsqueda elaborada con el mejor retorno y precisión resultante de la búsqueda manual.

Tabla 7 Cadenas de búsqueda ingresadas a diferentes motores de búsqueda

Motor de búsqueda	Cadena de búsqueda
IEEE Xplore	(Analysis bot OR chatbot OR "conversational bot" OR "conversational developer assistant" OR "developer assistant" OR Repair bot OR automated repair OR review bot OR software bot) AND ("software development" OR "software project" OR "open-source project" OR "software engineering" OR "repair")
ACM Digital Library	[[Title: analysis bot] OR [Title: chatbot] OR [Title: "conversational bot"] OR [Title: "conversational developer assistant"] OR [Title: "developer assistant"] OR [Title: repair bot] OR [Title: automated repair] OR [Title: review bot] OR [Title: software bot]] AND [[Title: "software development"] OR [Title: "software project"] OR [Title: "open source project"] OR [Title: "software engineering"] OR [Title: "repair"]] AND [[Abstract: analysis bot] OR [Abstract: chatbot] OR [Abstract: "conversational bot"] OR [Abstract: "conversational developer assistant"] OR [Abstract: "developer assistant"] OR [Abstract: repair bot] OR [Abstract: automated repair] OR [Abstract: review bot] OR [Abstract: software bot]] AND [[Abstract: "software development"] OR [Abstract: "software project"] OR [Abstract: "open source project"] OR [Abstract: "software engineering"] OR [Abstract: "repair"]]
Springer Link	"software development" AND (bot OR "development bot" OR devbot) AND (use OR application OR operation OR research)
Science Direct	(Analysis bot OR chatbot OR "conversational bot" OR "developer assistant" OR Repair bot OR review bot OR software bot) AND ("software development" OR "software engineering")
EBSCO Host	(Analysis bot OR chatbot OR "conversational bot" OR "conversational developer assistant" OR "developer assistant" OR Repair bot OR automated repair OR review bot OR software bot) AND ("software development")

El proceso de búsqueda automatizada siguió las siguientes etapas.:

- 1) Eliminar literatura por idioma y año.  
Primero que nada, los investigadores aplicaron los filtros básicos basados en los criterios CE-1 y CI-2 los cuales hablan del idioma de la literatura deseada y el año de publicación.
- 2) Eliminar literatura que no sea artículo de investigación sobre software.  
Basándose en el CI-3 se eliminó literatura irrelevante para la investigación.
- 3) Eliminar literatura secundaria y terciaria.  
Después, en los casos que la literatura resultó haciendo referencia a el desarrollo de software, basándose en el CE-2, se analizó si la literatura no corresponde a literatura secundaria o terciaria con ayuda de los motores de búsqueda donde se tienen filtros por estudios primarios y secundarios.
- 4) Eliminar estudios por título con base en términos de búsqueda.  
En esta etapa se excluyó de la literatura aquellos estudios que no incluyeron por lo menos dos términos de búsqueda haciendo referencia al criterio CI-4. Cabe

mencionar que estas primero 4 etapas fueron realizadas el mismo día en el 18 de octubre del 2021.

- 5) Eliminar estudios por título que no hablan de bots en el desarrollo de software. En esta etapa se excluyó de la literatura aquellos estudios no relevantes para la investigación hablando de temas ajenos a los bots en el desarrollo de software con base al título propio del estudio.
- 6) Eliminar estudios por resumen que no hablan de bots en el desarrollo de software. En esta etapa se excluyó de la literatura aquellos estudios que no fueron relevantes para la investigación debido a que al leer su resumen se enfocaba a temas ajenos a los bots en el desarrollo de software. La etapa 5 y 6 fueron realizadas ambas en el 18 de octubre del 2021.
- 7) Examinar estudios relevantes por medio de revisión crítica. En esta etapa se leyeron completamente desde introducción hasta conclusión la literatura restante con el fin de decidir si son de suficiente calidad con respecto a sus capacidades para responder una o varias de las preguntas de investigación con base al criterio CI-5. Este proceso de lectura ocurrió junto con la extracción de datos durante el mes de noviembre del 2021.
- 8) Evaluar la calidad de los estudios. Finalmente, el estudio pasó por un proceso de evaluación de calidad. El proceso de evaluación es detallado a profundidad en la sección 4.3.3. Evaluación de calidad

En la Tabla 8 se pueden ver los resultados de cada etapa del proceso de búsqueda automatizada.

Tabla 8 Resultados de la búsqueda automatizada

Base de datos	Inicio	Etapas 1	Etapas 2	Etapas 3	Etapas 4	Etapas 5	Etapas 6	Etapas 7
ACM Digital Library	3725	2029	1277	1184	239	25	12	2
IEEE Xplore	919	530	253	248	182	55	38	23
Springer Link	774	532	231	172	56	17	2	1
Science Direct	993	478	330	221	58	21	4	0
EBSCO Host	634	292	85	82	36	22	7	4
Totales	7045	3861	2176	1907	571	140	63	30



### 4.3 Extracción de datos y clasificación

En esta sección se detallan las actividades para la extracción de datos y evaluación de calidad posterior a la selección de estudios.

#### 4.3.1 Formato de extracción

Para contestar las preguntas de investigación se decidió obtener información de cada estudio restante del proceso de la selección de búsqueda. En la Tabla 9 se muestra el formato concentrador elaborada para la extracción de datos de estudios seleccionados.

Tabla 9 Formato de concentrador de datos

<b>DATOS DE LA PUBLICACIÓN</b>	
Título	
Autores	
Año	
Publicador	
Tipo de publicación (memorias, <i>journal</i> , etc.)	[artículo de congreso, journal científico, artículo de revista]
DOI	
Palabras clave	
<i>Abstract</i> o resumen	
Objetivos	
Pregunta/s de investigación relacionada/s	
Base de datos fuente	
Tema	
Tecnologías	
Instrumentos de investigación	
<b>DATOS PARA LA EXTRACCIÓN</b>	
Tipo de Bot	
Etapas de desarrollo (PI-1)	
Objetivo del Bot (PI-2)	
Retos o problemas enfrentados con el uso del bot (PI-3)	
Nivel de inteligencia del Bot (PI4)	

La tabla esta dividida en dos secciones. Estos son Datos de publicación y Datos para la extracción. La primera sección contiene información para la identificación de cada estudio individual como el título propio del estudio, los autores del estudio, el año de publicación del estudio y el tipo de publicación del estudio ya sea un artículo de congreso, un artículo de un *journal* científico o un artículo de una revista. Adicionalmente se recolecta el

hipervínculo del DOI de cada estudio en donde se puede observar en línea el artículo completo y la base de datos de donde fue obtenida el estudio en su completitud. Aparte de los datos de identificación, se extrajeron los Abstractos, objetivos y palabras claves de cada estudio para identificar durante la etapa final del proceso de selección de estudios si un estudio puede responder por lo menos una pregunta de investigación. Para apoyar en ese esfuerzo, se recolecta también cuales de las preguntas de investigación establecidas pueden ser respondidas satisfactoriamente después de leer completamente el estudio. Información para identificar cuales preguntas pueden ser respondidas incluye el tema del estudio, tecnologías investigadas en el estudio e instrumentos de investigación mencionados en el estudio.

La sección dos, Datos para la extracción se refieren a datos de un estudio directamente asociados a las preguntas de investigación. Primero se recolecta el tipo de bot identificado en el estudio dependiendo de objetivo del bot dividido en las categorías bot de productividad, información o colaboración (C. Lebeuf et al., 2018) y si es un tipo de bot definido como chatbot, analysis bot, voice bot, repairbot o devbot de acuerdo con los Antecedentes de este trabajo. Los datos restantes por extraer se refieren cada uno a una pregunta de investigación. La etapa de desarrollo se identifica para responder a la PI-1, el objetivo del bot para responder a la PI-2, los retos enfrentados con el uso de bots para PI-3 y finalmente el nivel de inteligencia de acuerdo con los aspectos de la inteligencia de un bot Contexto, Razonamiento y Autonomía. (C. Lebeuf et al., 2018) Estos aspectos se refieren a la capacidad de un bot para reconocer el dominio y contexto de su área de trabajo, razonar y producir soluciones, recomendaciones o acciones que asistan a los desarrolladores que emplean su uso y su habilidad de trabajar de manera autónoma sin comandos explícitos de un humano respectivamente.

#### 4.3.2 Extracción de datos

El proceso de extracción de datos fue ejecutado en etapas siguiendo el método de Kitchenham *et al.* (2015) durante el mes de noviembre del 2021. Los datos fueron gradualmente agregados en un archivo de Microsoft Excel. En Excel, los resultados fueron separados en hojas divididas entre los motores de búsqueda EBSCO Host, IEEE Xplore, ACM DL y Springer Link. Primero, se dividió el trabajo de extracción entre los dos investigadores. Después, cada investigador fue llenando en Excel una fila de la tabla del concentrador de datos al leer cada estudio por completo. Después de leer el estudio y llenar su entrada en el concentrador de datos, se realizó una evaluación de calidad del estudio individual en otro archivo de Excel siguiendo el formato del Criterios para la evaluación de calidad. Este proceso fue repetido hasta extraer los datos y evaluar la calidad de todos los estudios seleccionados. En el caso que había dudas al extraer datos o evaluar la calidad se realizaba una discusión entre los investigadores para resolver desacuerdos. Las discusiones realizadas normalmente se enfocaban en la categorización del nivel de inteligencia de un bot. Este proceso transcurrió desde el 8 al 22 de noviembre.

### 4.3.3. Evaluación de calidad

El proceso para la evaluación de calidad ocurrió en paralelo que la extracción de datos. El proceso para la evaluación de calidad de estudios primarios se basó del método de Kitchenham *et al.* (2015). Las etapas para esta fase del trabajo incluyeron:

- Seleccionar lista de cotejo
- Asignar calificaciones de calidad
- Validar calificaciones
- Evaluar el impacto de las calificaciones

Se realizó este proceso de evaluación de calidad para asignar calificaciones a los trabajos primarios seleccionados para medir la importancia de los resultados de extracción de datos de los estudios individuales. La lista de cotejo usada fue originalmente propuesta por Dybå, T. y Dingsøyr, T. (2008a) con su lista de verificación de calidad que se utiliza en varios tipos de estudios.

Adicionalmente se le agrego un sistema de puntos para calificar la calidad en base a la lista de cotejo el cual se basó en una puntuación de 10 puntos divididos entre las 11 preguntas siguientes:

- P1. ¿El documento se basa en una investigación (o es simplemente un informe de "lecciones aprendidas" basado en la opinión de expertos)? (1 pt.)
- P2. ¿Existe una declaración clara de los objetivos de la investigación? (1 pt.)
- P3. ¿Existe una descripción adecuada del contexto en el que se realizó la investigación? (1 pt.)
- P4. ¿El diseño de la investigación fue apropiado para abordar los objetivos de la investigación? (1 pt.)
- P5. ¿La estrategia de contratación fue adecuada a los objetivos de la investigación? (0.5 pt.)
- P6. ¿Había un grupo de control con el que comparar los tratamientos? (0.5 pt.)
- P7. ¿Se recopilaron los datos de una manera que abordará el tema de la investigación? (1 pt.)
- P8. ¿Fue el análisis de datos suficientemente riguroso? (1 pt.)
- P9. ¿Se ha considerado adecuadamente la relación entre el investigador y los participantes? (0.5 pt.)
- P10. ¿Existe una declaración clara de los hallazgos? (1 pt.)
- P11. ¿Tiene valor el estudio para la investigación o la práctica? (1.5 pt.)

Conociendo las puntuaciones se asignaron las calificaciones de puntaje de la lista de cotejo como se muestra en la Tabla 10.

Tabla 10 Puntaje de evaluación de calidad

Nivel de calidad	Rango
Alto	10-8.5 pts.
Medio	8-6 pts.
Bajo	5-1 pts.

Que los documentos cumplan con la calidad del trabajo fue una parte importante del desarrollo. Antes de empezar con el proceso de evaluación de calidad, se hizo una prueba piloto con cinco estudios relevantes para verificar que la lista de cotejo sea funcional. Los cinco trabajos son los siguientes:

Trabajo 1: Balachandran, V. (2013, May). Reducing human effort and improving quality in peer code reviews using automatic static analysis and reviewer recommendation

Trabajo 2: Erlenhov, L., de Oliveira Neto, F. G., Scandariato, R., & Leitner, P. (2019, May). Current and future bots in software development.

Trabajo 3: Lin, C. T., Ma, S. P., & Huang, Y. W. (2020, June). MSABot: A chatbot framework for assisting in the development and operation of microservice-based systems.

Trabajo 4: Paikari, E., & Van Der Hoek, A. (2018, May). A framework for understanding chatbots and their future.

Trabajo 5: Wyrich, M., & Bogner, J. (2019, May). Towards an autonomous bot for automatic source code refactoring.

En la Tabla 11 se pueden ver los resultados en el formato de concentrador de evaluación de calidad empleada en Excel para la evaluación de estudios posterior a la extracción de datos. Se encuentra la lista completa resultante de la ejecución del proceso de evaluación de calidad en el Apéndice C: Resultados de evaluación de calidad.

Tabla 11 Formato de concentrador de evaluación de calidad

Pregunta	Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4	Trabajo 5
¿El documento se basa en una investigación (o es simplemente un informe de "lecciones aprendidas" basado en la opinión de expertos)? (1 pt.)	Se basa en una investigación	Se basa en una investigación	Se basa en una investigación	Se basa en una investigación	Se basa en una investigación
¿Existe una declaración clara de los objetivos de la investigación? (1 pt.)	Sí	Sí	Sí	Sí	Sí
¿Existe una descripción adecuada del contexto en el que se realizó la investigación? (1 pt.)	Sí	Sí	Sí	Sí	Sí
¿El diseño de la investigación fue apropiado para abordar los objetivos de la investigación? (1 pt.)	Sí	Sí	Sí	Sí	Sí
¿La estrategia fue adecuada a los objetivos de la investigación? (0.5 pt.)	Sí	Sí	Sí	No	Sí
¿Había un grupo de control con el que comparar los tratamientos? (0.5 pt.)	Sí	No	Sí	No	No
¿Se recopilaron los datos de una manera que abordará el tema de la investigación? (1 pt.)	Sí	Sí	Sí	Sí	Sí
¿Fue el análisis de datos suficientemente riguroso? (1 pt.)	Sí	No	Sí	Sí	Sí
¿Se ha considerado adecuadamente la relación entre el investigador y los participantes? (0.5 pt.)	Sí	No	Sí	No	Sí
¿Existe una declaración clara de los hallazgos? (1 pt.)	Sí	Si	No	Si	Sí
¿Tiene valor el estudio para la investigación o la práctica? (1.5 pt.)	Sí	Sí	Sí	No	Sí
Total	10 pts.	8 pts.	9 pts.	6.5 pts.	9.5 pts.

### 4.3.3. Síntesis de datos

Con los datos extraídos, se realizó un análisis mediante una síntesis narrativa en el periodo entre 15 de diciembre del 2021 y 10 de enero del 2022 con el objetivo de responder a las preguntas de investigación planteadas. La síntesis narrativa se basó en los 32 bots detallados en los 30 estudios restantes de la evaluación de calidad. Debido a la naturaleza flexible de una síntesis narrativa no hay una sola metodología formal para seguir el proceso. Sin embargo, la síntesis narrativa elaborada tomo como guía los casos de estudio de Cruzes, Dybå y Höst (2014) para adaptar los datos extraídos para una síntesis narrativa en vez de una síntesis temática de Creswell orientada a la ingeniería de software de Cruzes y Dybå (2011) como fue planeado originalmente. De acuerdo con Cruzes, Dybå y Höst (2014), una síntesis narrativa se puede definir como:

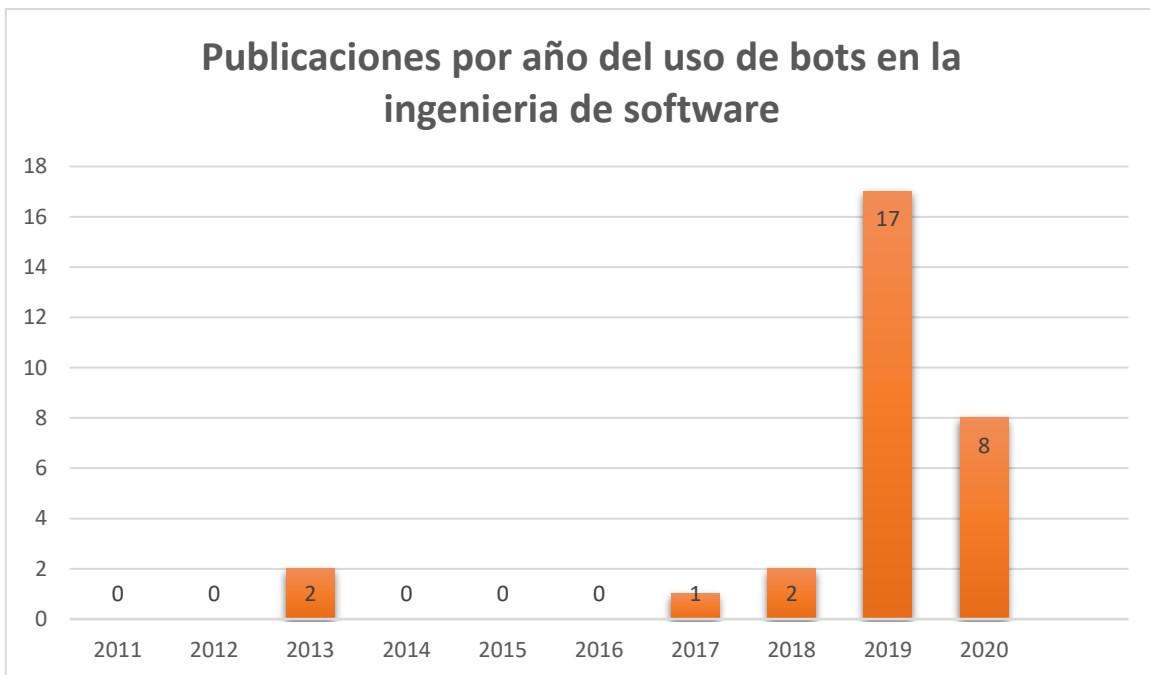
“El uso de palabras y texto para condensar y explicar los encuentros de la síntesis. Aunque se puede incluir la manipulación de datos estadísticos, la característica definida (de la síntesis narrativa) es la adopción de un enfoque textual al proceso de síntesis para ‘relatar la historia’ de los datos extraídos de los estudios seleccionados.”

La síntesis narrativa elaborada para este MSL no sigue una metodología establecida formal como en el proceso de identificación de estudios, evaluación de calidad y extracción de datos debido a que se planea en el futuro cercano realizar la totalidad de una síntesis temática en base a los datos extraídos en este trabajo. En la sección Resultados se puede ver la narración de los datos analizados. Aparte de una narración que relata el análisis de los datos, se encuentran graficas que ilustran las respuestas a las preguntas de investigación realizadas.

## 5. Resultados

Dentro de esta sección se detallan los resultados que se han obtenido del proceso ejecutado con el método anteriormente mencionado. En los siguientes puntos se profundiza los resultados sobre las actividades de desarrollo encontradas en donde se aplican bots, los beneficios que puede brindar un bot y consecuentemente, los retos asociados con el uso de bots.

Se encontraron diferentes bots repartidos entre los años 2011 a 2020 y se encuentra un detalle muy peculiar, cómo se ve en la figura NUMERO, que hubo un auge del uso de bots en la ingeniería de software durante el año 2019 y no es mera casualidad, justo durante este año es que se celebra el primer workshop internacional sobre el uso de bots en la ingeniería de software BotSE, después de ese año los resultados de documentos sobre bots aumento de una manera muy exponencial aunque en 2020 se hayan encontrado un poco menos pero eso es debido a la pandemia de COVID que sufrió el mundo.

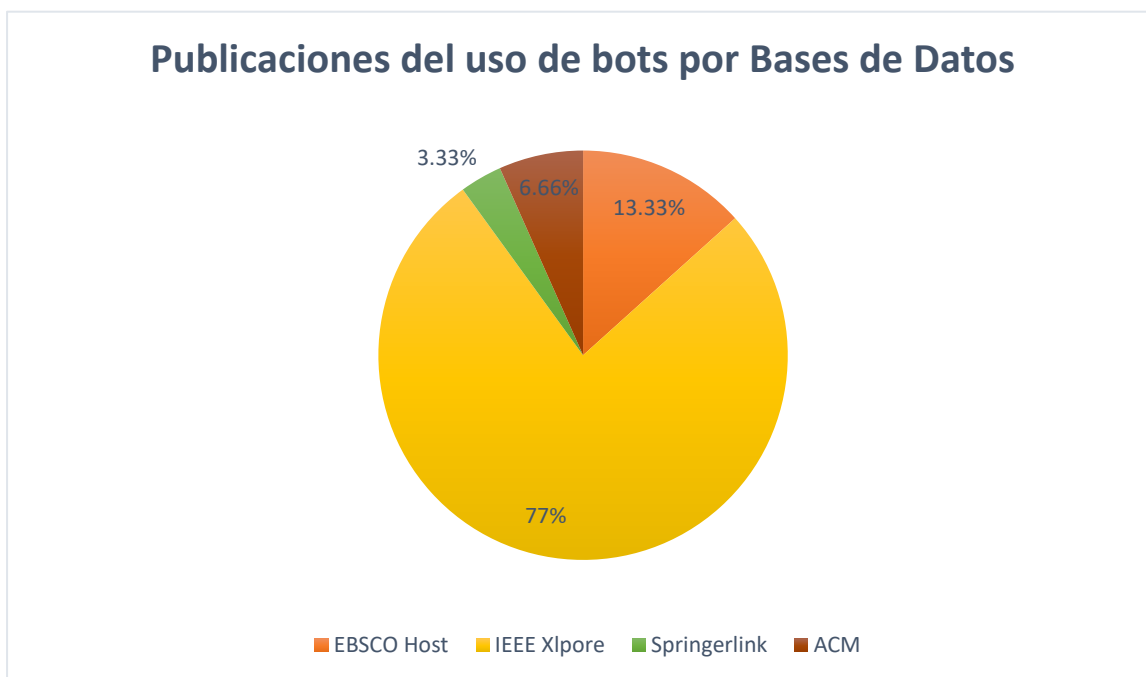


Sabiendo esto podemos declarar que este tema logro tener alta relevancia internacional gracias al workshop de BotSE al que los investigadores y directores asistieron con anterioridad.



Inclusive en la figura NUMERO podemos ver que, entre los distintos tipos de artículos, en su mayoría todos provienen de un congreso reafirmando lo anteriormente mencionado de que el workshop de BotSE genero un interés internacional en el tema del uso de bots en la ingeniería de software, los demás artículos provienen de journals y una última excepción proviene de una revista.

Como ya se ha mencionado anteriormente, gracias al workshop internacional de bots en la ingeniería de software el tema tuvo una gran alza a nivel internacional sobre el tema, aunque también no por eso debemos ignorar todo el trabajo existente que se encuentra detrás del año 2019, que, aunque solo representan un 20% de los trabajos totales, demuestran que el tema ya se encontraba en existencia y solo necesitaba un empujón internacional para empezar a tomar un auge.



## 5.1 Actividades de desarrollo en donde se utilizan bots

En la figura 2 se detalla específicamente las actividades de desarrollo de software que se encontraron entre los 30 artículos primarios, de estos resultados podemos resaltar que la actividad con más apoyo de parte de los bots es la gestión y trazabilidad del proyecto de software.

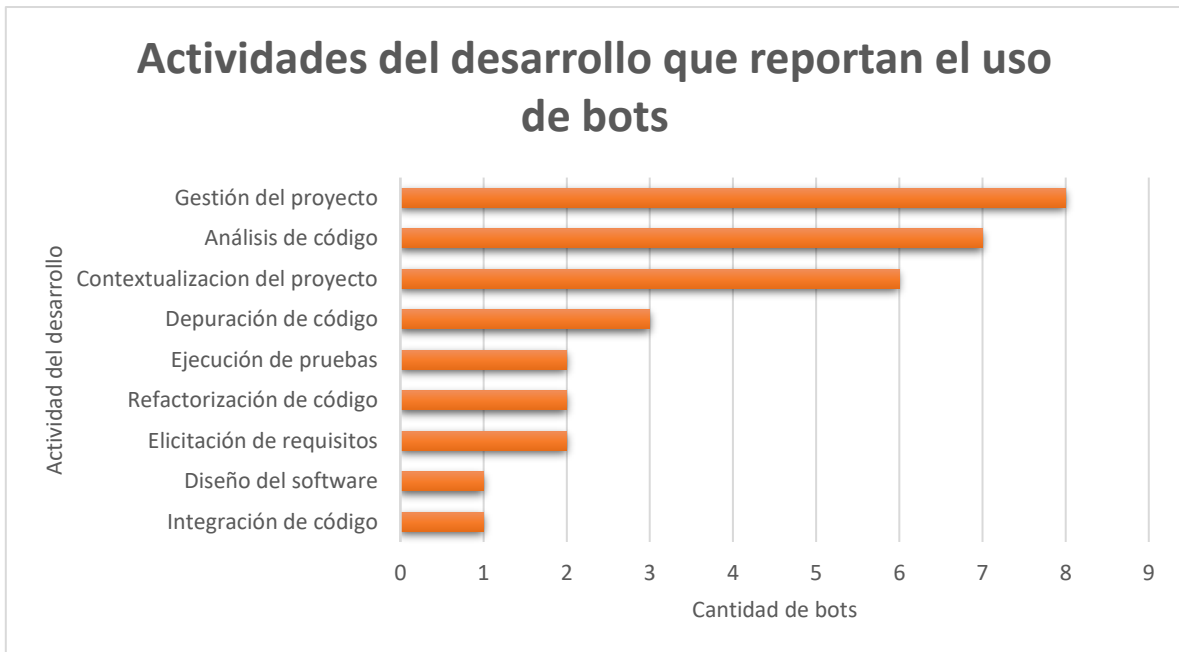


Figura 2 Actividades del desarrollo que reportan el uso de bots

En la tabla 12 *Actividades del desarrollo que reportan el uso de bot* se muestra el desglose de los bots específicos a cada actividad del desarrollo reportada.

Actividad	Id de estudio y nombre	Cantidad de bots	Porcentaje del total de bots (%)
Gestión del proyecto	PS-2 Bot de Hukal P et al PS-8 Devy Bot PS-13 Bot de Matthies C. <i>et al.</i> PS-17 Stale Bot PS-21 TA Bot PS-23 Code Climate Bot PS-29 JIT Bot PS-30 Bot de Georgios A. <i>et al.</i>	8	25%

Análisis de código	PS-10 Repairnator PS-11 SapFix PS-12 Sankie PS-23 Danger Bot PS-24 Review Bot PS-25 CCBot PS-27 C-3PR Bot	7	22%
Contextualización del proyecto	PS-5 Bot de Dominic J., Ritter C. y Rodeghero P. PS-7 MSR Bot PS-16 Smart Advisor PS-19 Ptracer PS-26 API Bot PS-28 Bot de Ilić A. <i>et al.</i>	6	19%
Depuración de código	PS-1 Bot de Belskii A. e Itsykson V. M PS-6 Reparnaitor PS-14 Bot de Monperrus M	3	10%
Ejecución de pruebas	PS-3 Bot de Padgham L. <i>et al.</i> PS-23 Travis CI Bot	2	6%
Refactorización de código	PS-18 Refactoring Bot PS-20 RefBot	2	6%
Elicitación de requisitos	PS-4 Bot de Dwitam F. y Rusli A. PS-22 Bot de Surana C. <i>et al.</i>	2	6%
Diseño de software	PS-9 Bot de Gilson F. y Weyns D.	1	3%
Integración de código	PS-15 Sayme	1	3%

## Bots sobre gestión del proyecto

Como dice Matthies, C. *et al.* (2019) " Los equipos de desarrollo de software ya dedican una cantidad significativa de tiempo a comunicarse en soluciones de chat y, a veces, también las usan durante las retrospectivas." Se han intentado buscar diferentes alternativas para poder gestionar el proyecto de desarrollo. Esto se refleja en que justamente es la actividad de desarrollo con más uso de bots. Con el bot de Matthies, C. *et al.* (2019) en PS-13 proponiendo el uso de chatbots en el dominio del desarrollo de software con un enfoque en respaldar los análisis y las mediciones de los datos del proyecto de los equipos, como el TA Bot en el PS-21 que intenta resolver retrasos de trabajo durante el desarrollo, con gráficas del impacto que ocasiona y recomendaciones para mitigar los efectos negativos.

El Stale Bot en PS-17 tiene propuestas similares buscando disminuir el número de pull requests e issues que tienen que enfocarse los desarrolladores al etiquetar automáticamente como "Stale" o expirado las issues y pull requests, los cuales no han sido atendidos por el equipo de desarrollo después de cierto tiempo. Parecido a este se encuentra el JITbot en PS-29 el cual se encarga de priorizar las pull request que los revisores deben checar primero. Además, un bot con propuestas similares se encuentran el Code Climate Bot en PS-23 que se encarga de recordar al desarrollador de arreglos de códigos que se encuentren pendientes en el código.

Incluso la propuesta del Devy Bot en PS-18 enfocado en la creación de un asistente de desarrollo conversacional que reduce la cantidad de comandos manuales de bajo nivel que los desarrolladores necesitan realizar, liberándolos para que se concentren en sus tareas de alto nivel.

También existen propuestas más enfocadas a la comunicación entre equipos como la propuesta Georgios A. *et al.* (2020) en PS-30 que busca construir un clasificador de intenciones entre los mensajes del equipo de desarrollo, para un asistente de conversación de gestión integrado en las plataformas de comunicación modernas; o la

## **Bots de análisis de código**

También se aprecia que entre las principales actividades identificadas se encuentra el análisis estático del código, el cual se ejecuta en la etapa de codificación y mantenimiento del software, y que hoy en día es una actividad esencial en el desarrollo de software ya que se trabaja directamente con el código y puede llegar a presentar demasiadas problemáticas pues como menciona Balachandran, V. (2013, May) "El principal desafío en la revisión de códigos es la gran cantidad de esfuerzo humano involucrado; esto es cierto incluso para las revisiones de código asistidas por herramientas."

Y esto sucede debido a todo lo que llega a involucrar un análisis estático de código, como menciona en su libro McConnell, S. (2004). "Esto podría deberse a una cantidad excesiva de información involucrada en la aplicación del estándar de codificación o a que los revisores priorizan estas comprobaciones a favor de la verificación lógica."

Existen propuestas de bots muy útiles para esta actividad como el Repairnaitor Bot en PS-10 el cual identifica errores para solucionar compilados fallidos y propone la solución con un parche generado. O el bot Sankie en PS-12 el cual realiza análisis de códigos y se apoya con la herramienta de java Error Prone. Incluso el CCBot en PS-25 analiza el código y en base a sus resultados puede insertar de manera automática contratos inferidos en el código.

Pero en soluciones más generales podemos encontrar al Danger Bot en PS-23 el cual realiza un análisis general del código y te recomienda las soluciones al mismo. Incluso el C-3PR Bot en PS-27 proponiendo un bot que al analizar el código detalla las violaciones que se encuentran en el mismo o el Review Bot el cual realiza una misma actividad que el C-3PR Bot pero más enfocado a un estándar de codificación.

Y ya para un uso más específico se encuentra el SapFix Bot en PS-11 el cual ejecuta un análisis de código y posteriormente recomienda diferentes actividades en DevOps a realizar después de un pull request.

## **Bots de Contextualización del proyecto**

Y la tercera actividad con más uso de bots reportados es la actividad de contextualización del proyecto con las propuestas de Dominic, J. et al. (2020) en PS-20 el cual utiliza un bot para guiar a los nuevos programadores a través del código, lo que permite que el programador experimentado se concentre en su trabajo o el MSR Bot en PS-7 el cual documenta un bot enfocado a responder preguntas sobre repositorios de github para contextualizar a los desarrolladores nuevos en el equipo de trabajo.

Inclusive con el APIBot en PS-26 el cual se encarga de contestar preguntas acerca del api que se utiliza en el desarrollo, para aquellos nuevos integrantes en un equipo de desarrollo, inclusive la propuesta del Smart Advisor en PS-16 el cual y responder preguntas de desarrolladores, dar alertas y recomendaciones apropiadas al contexto y el flujo de trabajo. También encontramos el Bot de Ilic A. *et al.* el cual responde a las preguntas de los usuarios sobre programación.

O incluso a algo más específico con el PTracer en PS-19 que monitorea el repositorio del kernel de Linux y reporta los parches nuevos que resuelven bugs relevantes, contextualizando a los desarrolladores de sistemas operativos comerciales basados en el kernel.

## **Bots de Depuración de código**

A partir de estas actividades encontramos una cantidad de bots más escasa, pero aun así se encuentran propuestas interesantes como el bot de Belskii A. e Itsykson V. M en PS-1 con un bot que depura el código con sugerencias de parches para la depuración de este o en el Bot de Monperrus M. en PS-14 con un bot que automáticamente realiza parches de depuración de código y reporta al desarrollador el por qué.

Y ya un bot mas específico con Reparnaitor en PS-6 el cual proponer parches a fallas aun no detectadas por desarrolladores al monitorear compilados del proyecto en DevOps.

## **Bots de Ejecución de Pruebas**

Con tan solo 2 propuestas de bots muy similares encontramos el Travis CI Bot en PS-23 el cual ejecutaba pruebas independientes, brindando al usuario un pequeño reporte de las pruebas o el Bot de Padgham L. *et al.* el cual de manera general ejecuta y depura pruebas unitarias anteriormente creadas.

## **Bots de Refactorización de código**

Encontrando solo 2 bots cuyo objetivo es similar se encuentra el RefBot en PS-20 el cual busca crear recomendaciones de refactorización para eliminar code smells encontrados en el código y el Refactoring Bot en PS-18 el cual busca mejorar la calidad de un código refactorizándolo por su propia cuenta.

### **Bots de elicitación de requisitos**

Aquí, aunque se encuentran solo 2 bots, tienen propuestas muy interesantes y útiles como el Bot de Dwitam F. y Rusli A. en PS-4 el cual realiza entrevistas a stakeholders en proyectos de software y en base a la entrevista crea historias de usuario o el Bot de Surana C. *et al.* en PS-22 que realiza una conversación más natural con los Stakeholders y obtiene los requisitos en base a la conversación, clasificándolos en diferentes tipos de requisitos.

### **Bots de Diseño de software**

Teniendo solo 1 propuesta se tiene al Bot de Gilson F. y Weyns D. que monitorea las diferentes conversaciones y discusiones del equipo de desarrollo y genera artefactos de diseño en base a esto.

### **Bots de integración de código**

Encontrando solo a Sayme Bot en PS-15 proporciona avisos de potenciales conflictos directos e indirectos al subir cambios a un repositorio de GitHub cuando un desarrollador manipula un archivo

## 5.2 Beneficios del uso de bots en actividades de desarrollo

En la Figura 3 Gráfica de beneficios del uso de bots, se muestran los beneficios resultantes de la aplicación de bots en actividades de desarrollo reportados en los estudios seleccionados. Como se puede ver, de los 30 artículos finales, de estos resultados se puede observar que el beneficio más reportado de la aplicación de un bot es la generación de reportes, avisos o advertencias para informar a un desarrollador durante sus actividades de desarrollo y ayudarlo en su toma de decisiones con la información proporcionada por el bot.

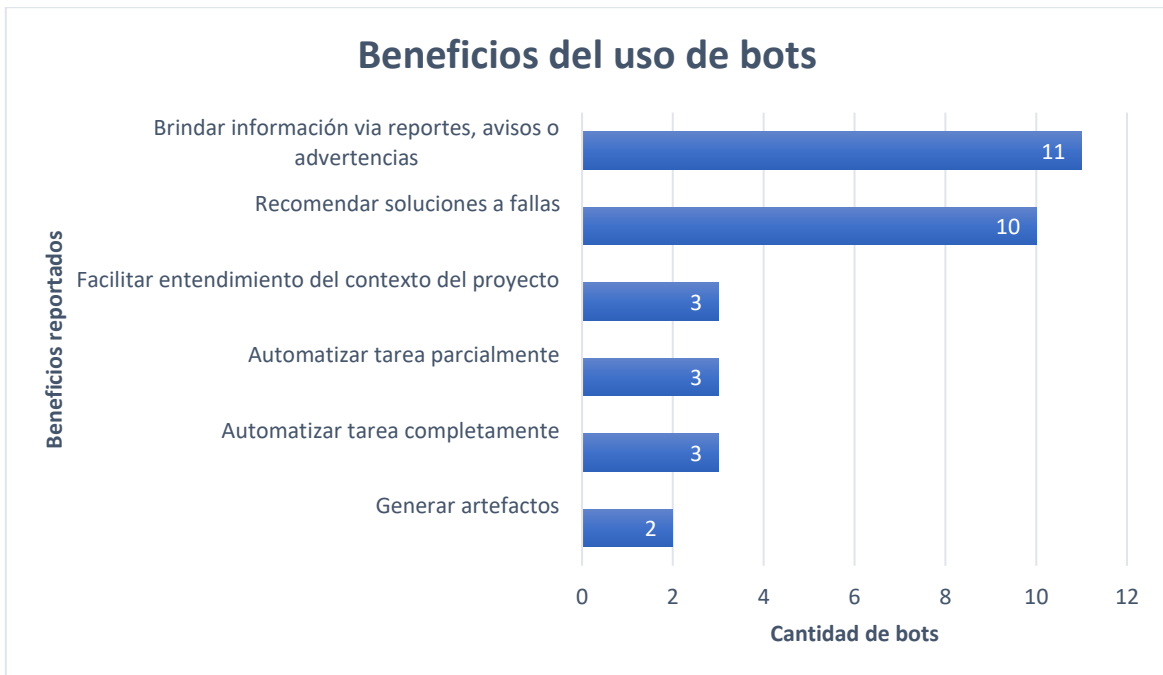


Figura 3 Gráfica de beneficios del uso de bots



En la Tabla 13 Categorías de beneficios de bots se muestra el desglose de los bots específicos a cada categoría de beneficios reportados.

Tabla 13 Categorías de beneficios de bots

Beneficio	Id de estudio y Nombre del bot	Cantidad de bots	Porcentaje del total de bots (%)
Brindar información vía reportes, avisos o advertencias	PS-3 Bot de Padgham L. et al., PS-13 Bot de Matthies C. et al., PS-15 Sayme, PS-16 Smart Advisor, PS-19 PTracer, PS-21 TA Bot, PS-23 Code Climate Bot, PS-23 Danger Bot, PS-28 Bot de Ilić A. et al., PS-29 JIT Bot, PS-30 Bot de Georgios A. et al.	11	34.375
Recomendar soluciones a fallas	PS-1 Bot de Belskii A. e Itsykson V. M., PS-2 Bot de Hukal P. et al., PS-6 Repairnator, PS-10 SapFix, PS-11 Sankie, PS-12 Bot de Brown C. y Parnin C., PS-14 Bot de Monperrus M. PS-20 RefBot, PS-25 CCBot, PS-27 C-3PR Bot	10	31.25
Automatizar tarea parcialmente	PS-17 StaleBot, PS-22 Bot de Surana C. et al., PS-23 Travis CI Bot	3	9.375
Automatizar tarea completamente	PS-8 Devy, PS-18 Refactoring Bot, PS-24 Review Bot	3	9.375
Facilitar entendimiento del contexto del proyecto	PS-5 Bot de Dominic J., Ritter C. y Rodeghero P., PS-7 MSR Bot, PS-26 API Bot	3	9.375
Generar artefactos	PS-4 Bot de Dwitam F. y Rusli A., PS-9 Bot de Gilson F. y Weyns D.	2	6.25

De los 32 bots identificados en la búsqueda que son aplicados al desarrollo de software, se reporta que 11 tienen el beneficio de proporcionar información útil ya sea de forma proactiva sin ser solicitada directamente por el desarrollador. Este es el caso en el chatbot Sayme en PS-15 que proporciona avisos de potenciales conflictos directos e indirectos al subir cambios a un repositorio de GitHub cuando un desarrollador manipula un archivo. Alternativamente, se puede usar un chatbot que brinda información útil cuando se solicite para apoyar en la toma de decisiones de desarrolladores como en el caso del bot Smart Advisor en PS-16 propuesto con la capacidad de responder preguntas de desarrolladores,

dar alertas y recomendaciones apropiadas al contexto y el flujo de trabajo durante codificación y mantenimiento.

El segundo beneficio más reportado con 10 bots es la recomendación de soluciones a fallas identificadas durante el desarrollo. Al obtener soluciones posibles a un problema enfrentado con el apoyo de un bot, el desarrollador pudo tomar una decisión adecuada sobre los parches generados por el bot en vez de invertir tiempo en analizar el problema por su cuenta. Estos bots son predominantemente repair bots que ayudan al desarrollador al identificar errores y dar posibles soluciones que le ahorra el tiempo al desarrollador y mejora la calidad del producto final. Por ejemplo, el bot Repairnator en Ep-6 genera parches a fallas aun no detectadas por desarrolladores al monitorear compilados del proyecto. En el caso que se encuentre una falla, Repairnator intenta replicar y reparar la falla y reporta al desarrollador el resultado del intento. Similarmente en el caso C-3PR Bot en PS-27, el bot sugiere correcciones a fallas identificadas por medio de un análisis estático a través por medio de pull requests al repositorio de un proyecto de software.

Por otra parte, se reporta que un bot ayuda al desarrollador a ahorrar tiempo al automatizar parte de una actividad en 3 casos o al automatizar completamente una tarea como es reportado en otros 3 casos. Los bots que brindan soporte al automatizar parcialmente una actividad incluyen los bots: el chatbot de Surana C. *et al.* en PS-22, el Devbot Travis CI en PS-23, y el bot de análisis Stale Bot. en PS-17. Estos bots automatizan parcialmente las actividades de entrevistar usuarios, integración continua y trazar pull requests respectivamente para ahorrar tiempo. En contraste, el bot Devy en PS-8 automatiza tareas de Git y GitHub al hacer commits a un repositorio, asignar revisiones de código y gestionar un repositorio via comandos de voz. Refactoring Bot creado en PS-18 ahorra tiempo al automatizar completamente la refactorización de código para resolver code smells identificados por la herramienta de análisis estático SonarQube. Posteriormente presenta los cambios al desarrollador para su revisión vía pull requests en GitHub. Finalmente, Review Bot de PS-24 automatiza el análisis estático al generar las comprobaciones de violaciones de estándares de codificación y patrones de defectos comunes, y publicar los resultados de sus analices.

También, se reporta de beneficio la mejora en el entendimiento del contexto de un proyecto de software en 3 casos. Estos son chatbots de información utilizados para la capacitación de nuevos miembros de un equipo de desarrollo. El primero, fue el chatbot API Bot en PS-26 brinda la facilidad de introducir al desarrollador al uso de un API documentada, ahorrándole tiempo de lectura de la documentación y facilitar la familiarización de un desarrollador con una API. Después en el 2020 se crearon bots para resolver preguntas para capacitar desarrolladores nuevos a un proyecto. El chatbot MSR Bot de PS-7 facilita el entendimiento del contexto de un repositorio para entender el software y los cambios realizados por el equipo. Al igual, el chatbot de Dominic J., Ritter C. y Rodeghero P. en PS-5 responde a preguntas de miembros para que se capaciten.

Finalmente, en 2 casos, los bots se usan para generar automáticamente modelos y por ende ahorrar tiempo al manualmente crear dichos artefactos. El primero, el chatbot propuesto por Gilson F. y Weyns D. en PS-9 se visiona como un chatbot que serviría para generar modelos de diseño automáticamente en base a las discusiones entre el equipo de desarrollo en el canal de comunicación Slack. El segundo, propuesto por Dwitam F. y Rusli A. en PS-4 es un chatbot para crear historias de usuario para los desarrolladores al conversar con interesados cuando entreguen requisitos preliminares o retroalimentación de un producto de software.

### 5.3 Retos con el uso de bots

En la Figura 4 Gráfica de retos enfrentados con el uso de bots, se muestran los retos reportados con la aplicación de bots en actividades de desarrollo. De los 32 bots extraídos de los estudios primarios, en 11 casos no se mencionaban retos. De los restantes 21 casos, el reto más grande fue la baja efectividad de recomendaciones en la forma de avisos, advertencias, parches o soluciones. Esto se refiere a que al usar el bot, un desarrollador no consideró útil el bot porque no fue de apoyo sus recomendaciones resultando en una pérdida de tiempo al emplear el bot y analizar sus recomendaciones.

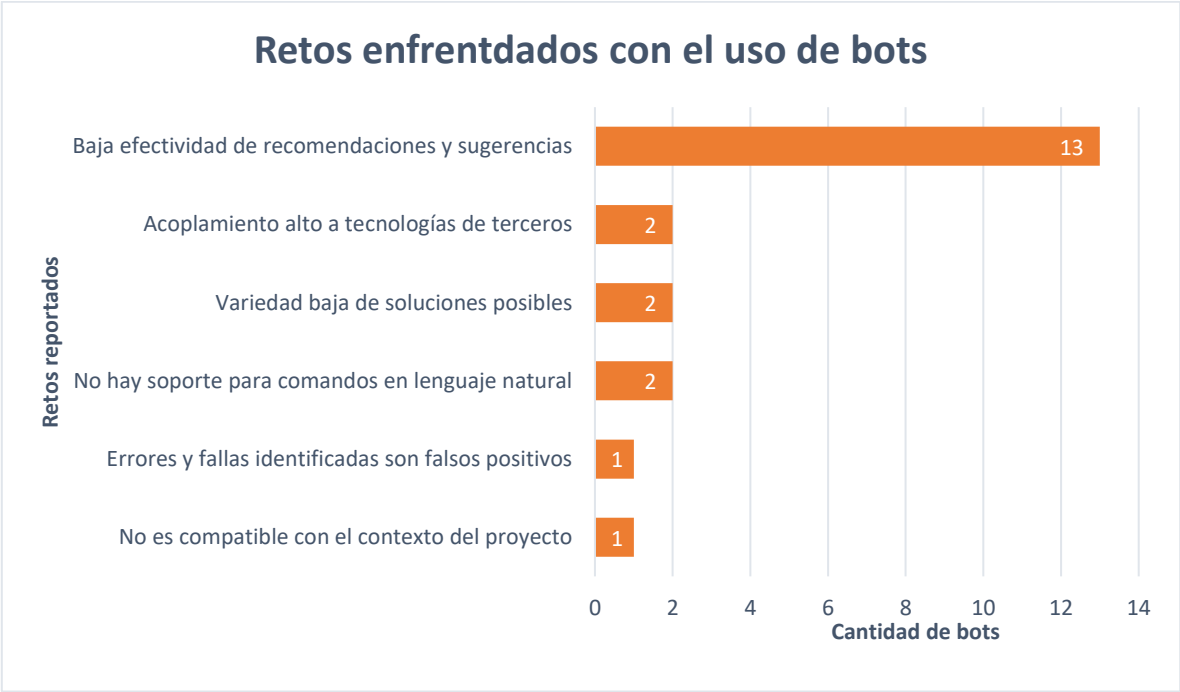


Figura 4 Gráfica de retos enfrentados con el uso de bots

En la Tabla 14 Categorías de retos con el uso de bots se muestra el desglose de los bots específicos a cada categoría de retos reportados.

Tabla 14 Categorías de retos con el uso de bots

Reto	Id de estudio y Nombre del bot	Cantidad de bots	Porcentaje del total de bots (%)
Baja efectividad de recomendaciones y sugerencias	PS-1 Bot de Belskii A. e Itsykson V. M., PS-5 Bot de Dominic J., Ritter C. y Rodeghero P., PS-6 Repairnator, PS-10 SapFix, PS-12 Bot de Brown C. y Parnin C., PS-14 Bot de Monperrus M., PS-16 Smart Advisor, PS-19 PTracer PS-23 Code Climate Bot, PS-23 Travis CI Bot, PS-23 Danger Bot PS-26 API Bot, PS-27 C-3PR Bot	13	52.38
Acoplamiento alto a tecnologías de terceros	PS-7 MSR Bot, PS-11 Sankie	2	9.52
Variedad baja de soluciones posibles	PS-20 RefBot, PS-21 TA Bot	2	9.52
No hay soporte para comandos en lenguaje natural	PS-8 Devy, PS-18 Refactoring Bot	2	9.52
Errores y fallas identificadas son falsos positivos	PS-13 Bot de Padgham L. et al.	1	4.76
No es compatible con el contexto del proyecto	PS-13 Bot de Matthies C. et al.	1	4.76

En los 13 casos en donde se reporta que un bot no fue efectivo en sus soluciones recomendadas se encuentra el C-3PR Bot en PS-27 que sufre problemas ya que un desarrollador puede determinar que una corrección sugerida no es útil y rechazarla. Por ejemplo, cuando el bot eliminó un fragmento de código comentado que los desarrolladores realmente querían conservar. Similarmente, PTracer en PS-19 tuvo una baja efectividad porque 49 de 151 o el 32.45% de los parches recomendados por el bot fueron rechazados principalmente porque los parches no funcionaban o no estaban relacionados a resolver bugs. Aún peor, el bot de Bot de Brown C. y Parnin C. en PS-12 se reporta que sus recomendaciones de instalación de una herramienta de análisis estático fueron inefectivas. Solo 2 de 52 o el 3.8% de las recomendaciones de su bot fueron aceptadas debido a que el

bot no reconocía el contexto de trabajo del desarrollador. Esto es debido a que al intentar instalar la herramienta el compilado ya no funcionaba o violaba el formato seguido por el desarrollador en los archivos manipulados por el bot para instalar la herramienta. Por otro extremo, en PS-23 se reportó que Travis CI Bot no fue útil para los desarrolladores que lo utilizaron. Se reporta que tuvieron dificultades en la ejecución de pruebas y preferían depurar manualmente.

Un reto adicional para bots que proponen soluciones y parches para la depuración de fallas fue la variedad limitada de soluciones. En dos casos se reporta que los bots estaban limitados en su cantidad de soluciones posibles. En el caso de RefBot en PS-20, se tenía una variedad baja de soluciones porque el bot solo se enfoca en Pull Requests recientes y no en el contexto actual del desarrollador. Aún más, el bot se enfocaba solo en atributos de calidad basados en el modelo QMOOD que puede ser detrimento en el caso que los desarrolladores no necesitan ese enfoque. Otro caso que reporto falta de variedad en sus recomendaciones fue el TA Bot en PS-21 porque el chatbot no tenía suficientes tipos de retrasos para dar recomendaciones efectivas. El bot era deficiente debido a que la recomendación que proporcionaba no resolvía la causa de los retrasos sino solo los síntomas.

De la misma manera, las demás dificultades reportadas ocurren muy pocas veces en la literatura seleccionada. En dos casos se reporta que se tuvieron dificultades con el uso del bot debido a que no hay soporte para ciertos comandos en lenguaje natural. El problema principal de Refactoring Bot reportado en PS-18 era que aún no se tenía soporte comandos en lenguaje natural y por lo tanto era alta la curva de aprendizaje para utilizar el bot ya que un desarrollador debe aprenderse los comandos específicos del chatbot. En el caso del chatbot Devy de PS-8, el bot no lograba interpretar comandos de ciertas frases. Por ejemplo, no interpretaba "who has made changes" como comando para identificar quien ha editado un archivo. Los participantes no consideraban que el bot puede interpretar comando de manera más ambigua en vez de ser específicos como los comandos en Git. Tampoco se tiene transparencia para conocer los pasos que hizo el bot para completar una tarea. Otro problema reportado en el estudio fue la falta de posibilidad de interactuar con el bot en una situación con varias personas hablando en un contexto social.

También se reporta en otros dos casos que la dependencia alta en tecnologías de terceros. Por ejemplo, se reportaba que se requería una conexión de internet constante. Por lo tanto, las respuestas eran poco útiles sin una conexión de internet como en el caso del chatbot MSR Bot en PS-7. El bot Sankie de PS-11 igualmente tenía una dependencia alta a tecnologías externas. En este caso, los análisis de Sankie estaban acoplados al pipeline de integración y despliegue continuo de los servicios de DevOps de la plataforma Microsoft Azure. Adicionalmente se reportó que puede haber falsos negativos en las recomendaciones de Sankie ya que se puede dar el caso que el bot genera una recomendación efectiva, pero es ignorada.

Los últimos retos fueron reportados con una sola instancia cada una. En el caso de la propuesta del bot de Matthies C. et al. en PS-13, se plantea como un reto que la mayoría no son compatibles con tareas específicas del desarrollo ágil y no reconocen contexto y aportan perspectivas como los demás miembros del equipo de desarrollo en la actualidad. Finalmente, se reportó en un caso que las fallas y errores que identificaba el bot de Padgham *et al.* en PS-3 tenía una tendencia de generar falsos positivos en las pruebas que generaba el bot y no se podía determinar con una sola prueba si la implementación era consistente con el diseño del proyecto.

#### 5.4 Niveles de inteligencia de bots

Las características de inteligencia que identificamos en este trabajo son la Autonomía, Razonamiento y Adaptación (C. Lebeuf et al., 2018).

En la Figura 5 Capacidad de adaptación de bots, se muestra el contado de nivel de inteligencia de un bot en base a la su capacidad de adaptación al interactuar con el desarrollador. Para poder adaptarse un bot requiere reconocer el contexto de los en actividades de desarrollo para poder adaptarse a las necesidades del desarrollador o tienen la capacidad de personalizar y refinar sus acciones para el beneficio del desarrollador conforme más interactúa con él. De los 32 bots extraídos de los estudios primarios, en 16 o un 50 % de los casos se reporta que un bot puede reconocer contexto. En contraste, en 13 casos se reporta que el bot no tiene la capacidad y en 3 casos no se menciona el nivel de inteligencia del bot.

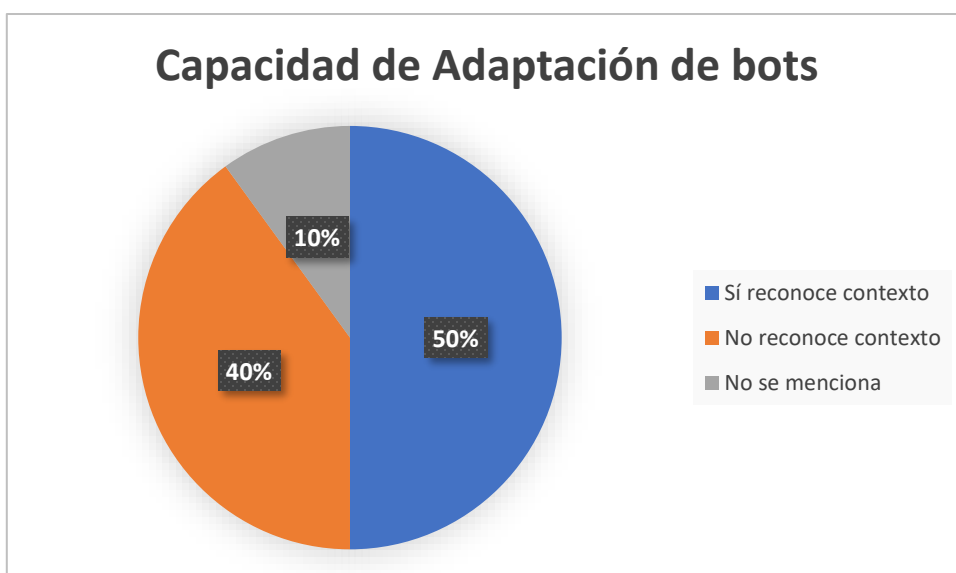


Figura 5 Capacidad de adaptación de bots

En la Tabla 15 Desglose de capacidad de adaptación de bots se muestra el desglose de los bots específicos a cada categoría de Adaptación reportados.

Tabla 15 Desglose de capacidad de adaptación de bots

Capacidad de adaptación	Id de estudio y Nombre del bot	Cantidad de bots	Porcentaje del total de bots (%)
Sí tiene	PS-3 Bot de Padgham L. et al., PS-5 Bot de Dominic J., Ritter C. y Rodeghero P., PS-6 Repairnator, PS-8 Devy, PS-11 Sankie PS-15 Sayme, PS-16 Smart Advisor, PS-22 Bot de Surana C. et al., PS-23 Code Climate Bot, PS-23 Danger Bot, PS-24 Review Bot, PS-25 CCBot, PS-26 API Bot, PS-27 C-3PR Bot, PS-29 JIT Bot, PS-30 Bot de Georgios A. et al.	16	50.0
No tiene	PS-1 Bot de Belskii A. e Itsykson V. M., PS-4 Bot de Dwitam F. y Rusli A., PS-7 MSR Bot, PS-9 Bot de Gilson F. y Weyns D. PS-10 SapFix, PS-12 Bot de Brown C. y Parnin C. PS-17 StaleBot, PS-18 Refactoring Bot PS-19 PTracer, PS-20 RefBot, PS-21 TA Bot PS-23 Travis CI Bot, PS-28 Bot de Ilić A. et al.	13	40.625
No se menciona	PS-2 Bot de Hukal P. et al., PS-13 Bot de Matthies C. et al., PS-14 Bot de Monperrus M.	3	9.375

La mitad de los bots clasificados demuestran Adaptabilidad. Bots como el chatbot para capacitación de Dominic J., Ritter C. y Rodeghero P. en PS-5 es propuesto con la capacidad de entender todo el contexto del área de un proyecto y brindar su conocimiento a los nuevos miembros del equipo dependiendo del proyecto al que se asignaran. Aparte de chatbots, bots de reparación como el Padgham *et al.* en PS-3 genera un oráculo para gestionar pruebas y categorizar fallas detectadas en base a un modelo de fallas configurable. También pueden adaptarse los bots de análisis. En el caso de JIT Bot en PS-29, el bot analiza el repositorio del desarrollador y sus commits realizados para adaptarse al contexto del proyecto y priorizar los pull requests que deben pasar por revisiones de código.

En la Figura 6 Capacidad de Autonomía de bots, se muestra el contado de nivel de inteligencia de un bot en base a la su capacidad de autonomía. La autonomía se refiere a la falta de necesidad de datos de entrada por parte de un humano para que el bot pueda realizar su propósito. De los 32 bots identificados, en la mayoría de los casos, 17 casos o un 53% se reportó que un bot es autónomo y puede ejecutar sus tareas y brindar beneficios sin comandos explícitos de un humano. Solo en 12 casos o 38% los bots no tienen esta capacidad y en 3 casos o 9% de los casos reportados no se menciona un nivel de inteligencia.

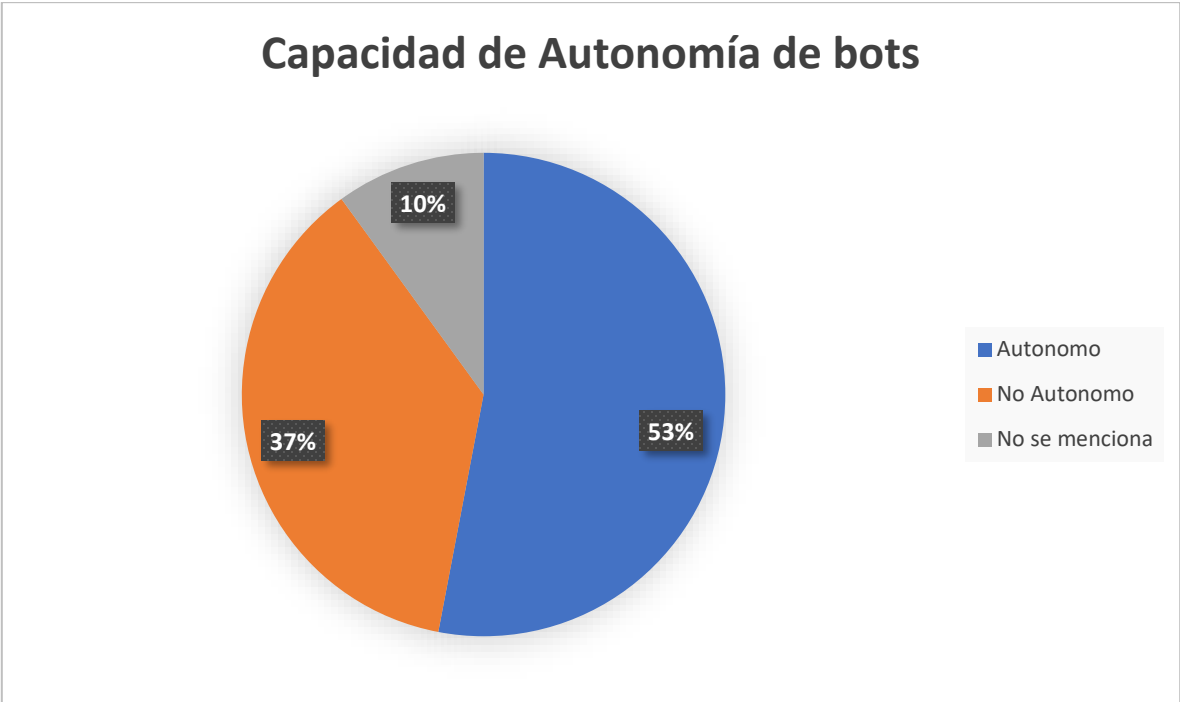


Figura 6 Capacidad de Autonomía de bots



En la Tabla 16 Desglose de capacidad de autonomía de bots se muestra el desglose de los bots específicos a cada categoría de Autonomía reportados.

Tabla 16 Desglose de capacidad de autonomía de bots

Capacidad de autonomía	Id de estudio y Nombre del bot	Cantidad de bots	Porcentaje del total de bots (%)
Sí tiene	PS-6 Repairnator, PS-9 Bot de Gilson F. y Weyns D., PS-10 SapFix, PS-11 Sankie, PS-12 Bot de Brown C. y Parnin C., PS-15 Sayme, PS-16 Smart Advisor, PS-17 StaleBot, PS-19 PTracer, PS-20 RefBot, PS-21 TA Bot, PS-23 Code Climate Bot, PS-23 Travis CI Bot, PS-23 Danger Bot, PS-27 C-3PR Bot, PS-29 JIT Bot, PS-30 Bot de Georgios A. et al.	17	53.125
No tiene	PS-1 Bot de Belskii A. e Itsykson V. M., PS-3 Bot de Padgham L. et al., PS-4 Bot de Dwitam F. y Rusli A., PS-5 Bot de Dominic J., Ritter C. y Rodeghero P., PS-7 MSR Bot, PS-8 Devy, PS-18 Refactoring Bot, PS-22 Bot de Surana C. et al., PS-24 Review Bot, PS-25 CCBot, PS-26 API Bot, PS-28 Bot de Ilić A. et al.	12	37.5
No se menciona	PS-2 Bot de Hukal P. et al., PS-13 Bot de Matthies C. et al., PS-14 Bot de Monperrus M.	3	9.375

Los bots que demuestran Autonomía tienen un grado de independencia y realizan sus tareas por sí mismos y contactan al desarrollador por mensajes y notificaciones para proporcionar información, avisos, recomendaciones o reportes de estatus ya sea periódicamente por su cuenta o cuando un humano lo solicite. Un caso que ilustra alta independencia es el bot JIT Bot de PS-29. Después de realizar su análisis inicial se queda al pendiente del repositorio y sus cambios sin requerir intervención humana. Los bots de productividad también pueden ser autónomos como el bot C-3PR en PS-27. Por su propia cuenta C-3PR sugiere correcciones a violaciones de análisis estáticos por medio de pull requests que elabora después de por su propia cuenta ejecutar un análisis estático.

En la Figura 7 Capacidad de razonamiento de bots, se muestra el contado de nivel de inteligencia de un bot en base a la su capacidad de Razonamiento. El razonamiento se

refiere a la existencia de un mecanismo interno al bot para analizar datos de entrada, realizar una tarea para ayudar al desarrollador y proporcionar datos de salida para avisarle a un desarrollador de los resultados de las acciones del bot. De los 32 bots identificados, en la mayoría de los casos, 20 o un 57% de los casos se logró extraer un mecanismo de razonamiento para un bot. De los casos restantes, en 9 o 26% de los casos, los bots no tienen esta capacidad y en 6 o 17% de los casos no se menciona un mecanismo de razonamiento.

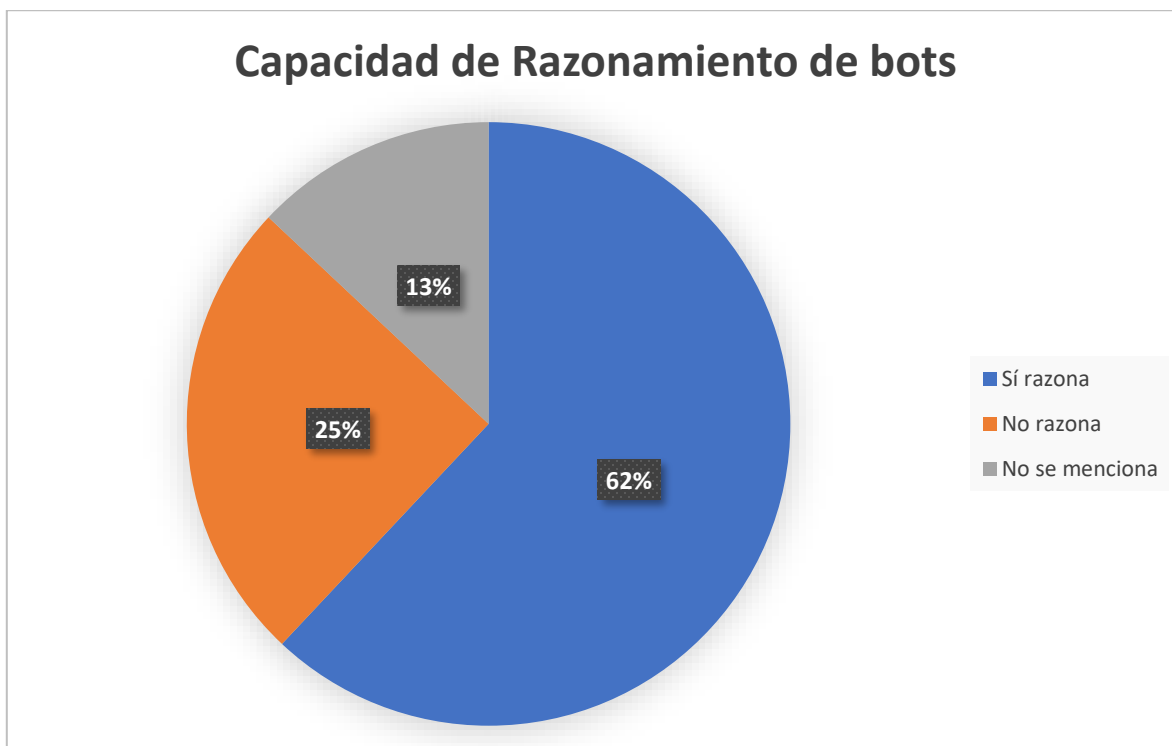


Figura 7 Capacidad de razonamiento de bots

En la Tabla 17 Desglose de capacidad de razonamiento de bots se muestra el desglose de los bots específicos a cada categoría de Razonamiento.

Tabla 17 Desglose de capacidad de razonamiento de bots

Capacidad de razonar	Id de estudio y Nombre del bot	Cantidad de bots	Porcentaje del total de bots (%)
Sí tiene	PS-1 Bot de Belskii A. e Itsykson V. M., PS-3 Bot de Padgham L. et al., PS-4 Bot de Dwitam F. y Rusli A., PS-5 Bot de Dominic J., Ritter C. y Rodeghero P., PS-6 Repairnator, PS-7 MSR Bot, PS-8 Devy, PS-9 Bot de Gilson F. y Weyns D., PS-10 SapFix, PS-11 Sankie, PS-15 Sayme, PS-16 Smart Advisor, PS-19 PTracer, PS-20 RefBot, PS-21 TA Bot, PS-22 Bot de Surana C. et al. PS-26 API Bot, PS-27 C-3PR Bot, PS-29 JIT Bot, PS-30 Bot de Georgios A. et al.	20	62.5
No tiene	PS-12 Bot de Brown C. y Parnin C., PS-17 StaleBot, PS-23 Code Climate Bot, PS-23 Travis CI Bot, PS-23 Danger Bot, PS-24 Review Bot, PS-25 CCBot, PS-28 Bot de Ilić A. et al.	8	25
No se menciona	PS-2 Bot de Hukal P. et al., PS-13 Bot de Matthies C. et al., PS-14 Bot de Monperrus M., PS-18 Refactoring Bot	4	12.5

En la Figura 8 Gráfica de mecanismos de razonamiento de bots, se muestra los tipos de mecanismos de razonamiento diferenciados de los datos extraídos. Se puede ver claramente que el Procesamiento Natural de Lenguaje (NLP), es el mecanismo de razonamiento más común con 8 casos identificados de un total de 20. Esto es debido a que la gran mayoría de los chatbots y voicebots identificados deben procesar lenguaje hablado o escrito por un desarrollador humano por medio de conversación local en vez de comandos designados.

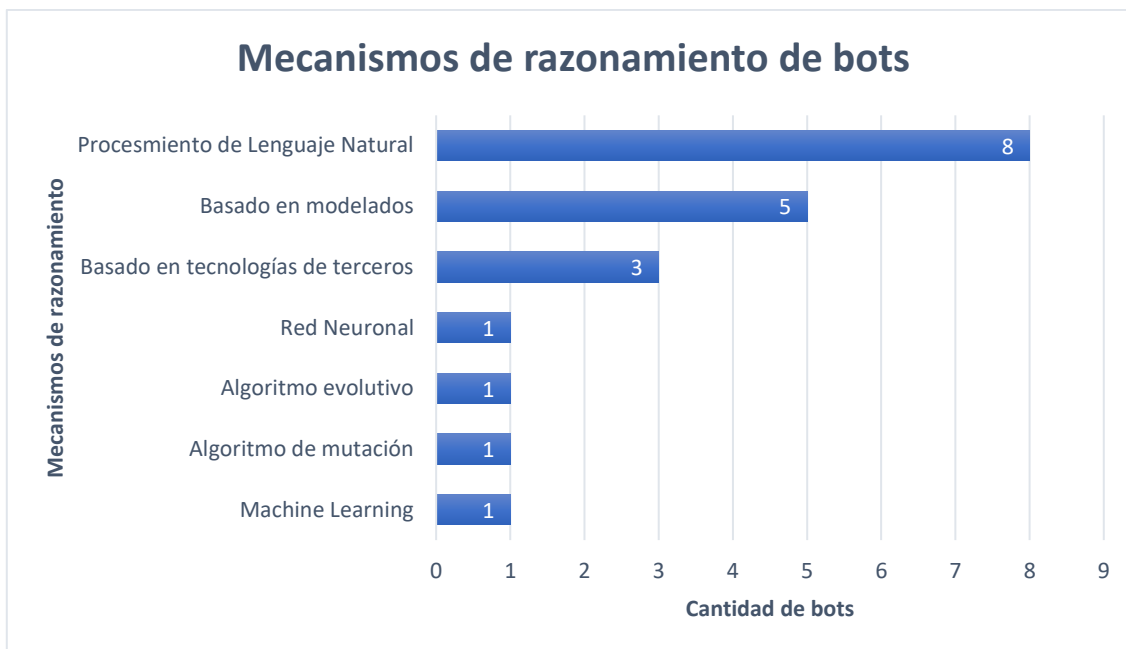


Figura 8 Gráfica de mecanismos de razonamiento de bots

En la Tabla 18 Categorías de mecanismos de razonamiento de bots se muestra las categorías identificadas de los bots que exhiben razonamiento.

Tabla 18 Categorías de mecanismos de razonamiento de bots

Mecanismo de razonamiento	Id de estudio y Nombre del bot	Cantidad de bots	Porcentaje del total de bots (%)
Procesamiento natural de lenguaje (NLP)	PS-5 Bot de Dominic J., Ritter C. y Rodeghero P., PS-7 MSR Bot, PS-9 Bot de Gilson F. y Weyns D., PS-15 Sayme, PS-16 Smart Advisor, PS-22 Bot de Surana C. et al., PS-26 API Bot, PS-30 Bot de Georgios A. et al.	8	40.0
Basado en modelos	PS-3 Bot de Padgham L. et al., PS-4 Bot de Dwitam F. y Rusli A., PS-8 Devy, PS-21 TA Bot, PS-27 C-3PR Bot	5	25.0
Basado en tecnologías de terceros	PS-6 Repairnator, PS-11 Sankie, PS-29 JIT Bot	3	15.0
Red Neuronal	PS-19 Ptracer	1	5.0
Algoritmo evolutivo	PS-20 RefBot	1	5.0

Algoritmo de mutación	PS-10 SapFix	1	5.0
Machine Learning	PS-1 Bot de Belskii A. e Itsykson V. M.	1	5.0

El segundo mecanismo más prevalente es basado en modelos con 5 casos. Estos bots generan modelos especializados para realizar su trabajo. Por instancia, el bot de reparación de Padgham et al. en PS-3 usa un modelo de fallas para categorizar fallas en excepciones, errores y advertencias. Otra instancia significativa es el chatbot de Dwitam F. y Rusli A. en PS-4 que utiliza un modelo basado en el lenguaje de marcado artificial AIML para su base de conocimientos que utiliza para crear historias de usuario para los desarrolladores al conversar con interesados.

Un mecanismo de razonamiento no tan común identificado fue basado en tecnologías de terceros como por ejemplo el bot Sankie en PS-11. Se utiliza Azure Batch para entrenar a Sankie y crear una base de conocimiento. Adicionalmente, usa Azure Cloud Services como motor de inferencia del bot. Otro caso es Repairnator de PS-6. Repairnator ejecuta pruebas a través de la API de Maven y JUnit y observa los resultados de prueba. El último caso es Devy de PS-8 que utiliza un servicio de intención basado en una máquina de estado finito para interpretar comandos y actualizar su modelo de contexto para determinar tareas a realizar.

Los demás mecanismos de razonamiento identificados solo tienen un caso cada uno. El bot de PTracer de PS-19 razona por medio de una red neuronal que toma como entrada parches del repositorio git del kernel de Linux para entrenar la red neuronal del bot con el fin de identificar si se resuelven o no bugs. Otro bot que utiliza una tecnología de inteligencia artificial es el bot de Belskii, A. e Itsykson, V. M. en PS-1 que usa Machine Learning para entrenar un motor de razonamiento. El bot SapFix de PS-10 utiliza un algoritmo de mutación para solucionar excepciones de apuntador nulo. Similarmente, el bot RefBot de PS-20 emplea el algoritmo evolutivo NSGA-II en donde identifica y evalúa atributos de calidad con ecuaciones predeterminadas para crear recomendaciones de refactorización para el desarrollador.

## 6. Discusión

Se nota claramente que el estado de este arte se encontraba muy escaso hasta llegar al año 2019 donde justamente se celebra el primer workshop internacional sobre el uso de Bots en la ingeniería de software o BotSE. Y gracias al mismo es que este tema comenzó a tomar relevancia entre los Ingenieros de Software, se aprecia también que en el siguiente año los trabajos primarios se redujeron significativamente, pero aun así fueron altos. Esto puede ser debido a que durante ese año fue que afecto globalmente la pandemia de COVID-19 causando una cuarentena en muchísimos países y llevando a una celebración virtual del evento.

Como pudimos ver, se tuvo una alta variedad de resultados y sorprendentemente superaron las expectativas de los investigadores ya que se creía que los bots más comunes llegarían a estar enfocados plenamente en la reparación de código y además de enfocarse en la ejecución de pruebas automatizadas, pero no resulto ser así.

### **Actividades de desarrollo donde se utilizan bots**

Las actividades más cargadas son aquellas donde se realiza una gestión, trazabilidad y colaboración con el proyecto, es decir, aquellas actividades donde se realiza comunicación directa con el equipo de desarrollo, seguimiento al trabajo de software realizado y manteniendo control en los repositorios virtuales donde se colabora y guarda el trabajo. También las actividades donde se contextualiza a los miembros del equipo ya sea para un nuevo miembro en el equipo de trabajo que necesita capacitarse para entender el contexto del trabajo o incluso para analizar el trabajo que realizan los otros miembros del equipo y que suben a repositorios virtuales donde almacenan el trabajo.

### **Beneficios del uso de bots en actividades de desarrollo**

Y derivando de estas actividades es que se muestra claramente que los mayores beneficios que brindan los bots encontrados en esta investigación son reportes que generan los mismos y además recomendar soluciones a los desarrolladores de su trabajo realizado en los repositorios virtuales, y de una manera obvia también uno de los beneficios a resaltar es la contextualización de área de trabajo.

### **Retos con el uso de bots**

En contraste con todos estos puntos positivos también analizamos que los retos que se encuentran con los bots donde el principal problema que se presenta, y claramente en relación a los puntos positivos que mencionamos, es que las recomendaciones o avisos que brinda el bot llegan a veces a ser un poco útiles y en comparación a lo que se busca realizar

con el mismo, es decir, a pesar de que muchas veces se busca gestionar y controlar procesos sencillos dentro del ámbito del trabajo, el bot no logra su objetivo completamente en algunos momentos de su funcionamiento ya sea porque se usa mal el bot o por que el bot aún no se encuentra en su 100%.

### **Niveles de inteligencia de bots**

Y en cuanto a los niveles de inteligencia de los bots obtenidos, se observa que por lo menos la mitad de los 32 identificados cumplen con los 3 puntos de inteligencia definidos por Lebeuf C. *et al.* (2020) que son:

- Autonomía
- Razonamiento
- Análisis de contexto

En los demás resultados algunos bots no cumplían con estos puntos de inteligencia o simplemente no se llegaba a mencionar en el documento.

## 7. Validez del estudio y limitantes

### 7.1 Confiabilidad de la búsqueda de literatura

La búsqueda de literatura fue realizada en base a la metodología de búsqueda automática Proceso Sistemático de Búsqueda basado en un Estándar Pseudo-Oro de Zhang *et al.* (2011) De acuerdo con Zhang *et al.*, para que la búsqueda automática sea confiable se debe elaborar una cadena de búsqueda con una precisión mínima del 80% vía una búsqueda manual limitada. Como se mencionó previamente en la sección 3 Método, la precisión de una búsqueda es la proporción de los estudios encontrados que son relevantes para las preguntas de investigación que aborda una revisión. Para alcanzar esta precisión, se fue realizando una búsqueda manual de la cual sus resultados se puede ver en la sección 4.2.3. Desarrollo de la búsqueda de la literatura de este documento. Después de realizar la búsqueda manual, se realizaron pruebas preliminares de varias cadenas de búsqueda en los motores de búsqueda IEEE Xplore en donde se calculaban dos métricas. La primera fue la cantidad de estudios filtrados por la cadena de búsqueda cuantos estudios seleccionados de la búsqueda manual localizaba la cadena de búsqueda bajo prueba y cuantos estudios en total y cuantos los estudios iterando por cadenas de búsquedas después de generar una cadena de búsqueda inicial basada en términos derivados de conceptos que investigamos del área de bots. Se pueden encontrar la tabla de conceptos y términos en el Apéndice B: Conceptos y Términos de bots. Por otra parte, se puede encontrar los resultados de las pruebas de las cadenas generadas en la [Tabla 5 Cadenas de búsqueda candidatas](#). Conforme la selección de fuentes, para la búsqueda manual se revisaron las actas de congreso de ICSE y sus subsidiarios incluyendo desde 2011 a 2020 relacionadas a bots. Adicionalmente, se amplió la búsqueda automática para abordar más bases de datos de lo originalmente planteado para incrementar la completitud y rigor de la MSL. Las bases de datos iniciales eran las bibliotecas digitales del área técnica IEEE Xplore, ACM Digital Library y Science Direct. Después se incluyó la biblioteca virtual del área general EBSCO Host y Springer Link. El motivo de selección de estos motores de búsqueda fue principalmente porque son las a las cuales tuvimos acceso completo gratuito, además de que la ICSE tiene declarado en su sitio web que sus acuerdos de conferencias se encuentran por completo en ACM Digital Library e IEEE Xplore en particular, lo cual brinda una mayor confianza a la búsqueda de artículos necesarios para continuar la investigación.

### 7.2 Confiabilidad de la selección de estudios

Para asegurar que se tenga una selección de estudios ordenada y de calidad se siguió la metodología para elaboración de estudios secundarios en la ingeniería de software de Kitchenham *et al.* (2015) como se detalla en **4.2.4 Selección de estudios relevantes**.

Aparte de seguir una metodología reconocida para la selección de estudios, se realizó una búsqueda independiente entre los dos investigadores del equipo de investigación César Sergio Martínez Palacios y Ricardo Moguel Sánchez. Cuando se encontraban desacuerdos se debatían hasta llegar a un consenso. En los casos que no se encontraba un consenso, para asegurar imparcialidad, se discutieron los resultados del progreso de la selección de



estudios con el director del equipo Dr. Jorge Octavio Ocharán Hernández y Co director: Dr. Xavier Limón Riaño. Un mecanismo final fue la contemplación e integración de una etapa de evaluación de la calidad a realizar después de la extracción de datos para asegurar una calidad adecuada para la investigación de estudios seleccionados detallado **4.3.3. Evaluación de calidad**. Con esta etapa del proceso de selección de estudios se clasificó los estudios seleccionados en base a un puntaje de calidad para establecer la importancia de los datos extraídos de cada estudio en base a su capacidad de responder a las preguntas de investigación de este trabajo. De esta manera se puede asegurar que las preguntas de investigación sean respondidas de una medida adecuada. Con este paso, se eliminó un estudio de baja calidad conforme a su utilidad en el contexto de nuestra investigación.

### 7.3 Confiabilidad de la extracción de datos y clasificación

El proceso de extracción de datos siguió el método de Kitchenham *et al.* (2015) como se detalla en la sección 4.3.2 Extracción de datos. De acuerdo con la metodología se elaboró un Formato de concentrador de datos situado en la Tabla 9 Formato de concentrador de datos. Gracias a esta la sección de extracción de datos de la plantilla, pudimos asegurar que los datos extraídos eran relevantes y a las preguntas de investigación. Al extraer los datos se pudo establecer una trazabilidad desde los resultados extraídos y las preguntas de investigación para garantizar que las preguntas de investigación fueron respondidas utilizando información clara, concisa y trazable a las preguntas de investigación. se comparará y contrastará resultados en una discusión posterior. Para disminuir el sesgo de confirmación, se comparó y contrastó los resultados de la extracción entre los dos investigadores. Además, cuando se tenían desacuerdos, se debatían hasta llegar a un consenso y en los casos en donde no se tenía un consenso se consultaba la opinión profesional de los directores para resolver los desacuerdos.

#### 7.4 Confiabilidad de la síntesis de datos

Originalmente, se contempló realizar una síntesis temática de Creswell adaptada para la ingeniería de Software conforme la metodología de Cruzes y Dybå (2011) pero, debido a restricciones de tiempo, se tuvo que posponer su elaboración para marzo 2022. En vez, se realizó una síntesis narrativa que tiene el beneficio de requerir menos tiempo para su elaboración, ser más flexible y útil para construir, aunque no existe una metodología formal establecida por la naturaleza de la síntesis. Aun así, se elaboró la síntesis narrativa en base al ejemplo de trabajo de Cruzes *et al.* (2014) .

La síntesis narrativa elaborada en este trabajo de reporte de MSL fue limitada en su rigor y aseguramiento de calidad al seguir una metodología establecida debido a que se planea en el futuro cercano realizar la totalidad de una síntesis temática en base a los datos extraídos en este trabajo.

## 8. Conclusión

Este informe aporta un mapeo sistemático realizado entre septiembre a diciembre 2021. Se identificaron hasta el momento, 30 trabajos primarios repartidos en 4 diferentes bases de datos, a pesar de que se usaron 5, y con esos se respondieron a las 4 preguntas de investigación definidas, las cuales eran:

- PI-1 ¿Cuáles son las actividades de desarrollo en donde se usan bots?
- PI-2 ¿Cuáles son los beneficios reportados del bot aplicado a una actividad de desarrollo?
- PI-3 ¿Que problemas se presentan en las actividades del desarrollo que el bot contribuye a su solución?
- PI-4 ¿Cuál es el nivel de inteligencia del bot aplicado?

Y respondiendo a estas actividades, podemos concluir de manera general que los resultados obtenidos del mapeo sistemático son:

30 trabajos primarios identificados en 4 bases de datos diferentes.

Las actividades con más uso de bots son aquellas donde se gestiona, traza y colabora el proyecto. (PI-1)

Los mayores beneficios reportados en los bots eran los reportes que generaban. (PI-2)

El mayor problema presentado fue la poca utilidad de los reportes generados. (PI-3)

La mitad de los bots reconocían contexto, razonaban y lograban tener autonomía. (PI-4)

El workshop de BotSE ha creado un auge en el arte.

Estos resultados son benefactores a los practicantes para comenzar a adentrarse en el tema y poder probar nuevas tecnologías para su desarrollo académico, además de que a los investigadores les sirve para poder aumentar su alcance a la hora de seleccionar estudios y poder partir de este punto para investigar a más profundidad y no partir de un punto desconocido. Concluimos que el tema de investigación sobre bots aplicados al desarrollo de software aún se encuentra en plena iniciación. Por lo tanto, es bueno aportar al mismo con esta investigación para servir de punto de partida para el trabajo futuro de investigadores y desarrolladores.

## 9. Referencias

1. Babar, M. A., & Zhang, H. (2009). Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. IEEE.
2. IBM. (1 de Julio de 2014). Software Development. Obtenido de IBM Research: [https://researcher.watson.ibm.com/researcher/view\\_group.php?id=5227](https://researcher.watson.ibm.com/researcher/view_group.php?id=5227)
3. Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). Evidence-based software engineering and systematic reviews. CRC Press.
4. Lebeuf C., A. Z. (2019). Defining and Classifying Software Bots:A Faceted Taxonomy. 1st International Workshop on Bots in Software Engineering (BotSE).
5. Lebeuf, C. (2018). A Taxonomy of Software Bots: Towards a Deeper Understanding of Software Bot Characteristics. University of Victoria, Greater Victoria, BC, Canada, 16-18.
6. Wohlin, C. (2014). Guidelines for Snowballing in Systematic Literature. New York.
7. Zhang, H., Babar, M. A., & Tell, P. (2011). Identifying relevant studies in software engineering. *Information and Software Technology*, 53(6), 625-637.
8. Kotkov, D., Pandey, G., & Semenov, A. (2018, December). Gaming bot detection: a systematic literature review. In *International Conference on Computational Social Networks* (pp. 247-258).
9. Stieglitz, S., Brachten, F., Ross, B., & Jung, A. K. (2017). Do social bots dream of electric sheep? A categorisation of social media bot accounts.
10. Orabi, M., Mouheb, D., Al Aghbari, Z., & Kamel, I. (2020). Detection of bots in social media: A systematic review.
11. de Moraes, D. M. G., & Digiampietri, L. A. (2021, June). Methods and Challenges in Social Bots Detection: A Systematic Review.
12. Alothali, E., Zaki, N., Mohamed, E. A., & Alashwal, H. (2018, November). Detecting social bots on Twitter: A literature review.
13. Wiesenber, M., & Tench, R. (2020). Deep strategic mediatization: Organizational leaders' knowledge and usage of social bots in an era of disinformation.
14. Collins, B., Hoang, D. T., Dang, D. T., & Hwang, D. (2020, November). Method of Detecting Bots on Social Media. A Literature Review.
15. Providel, E., & Mendoza, M. (2021). Misleading information in Spanish: a survey. *Social Network Analysis and Mining*.
16. Paul, H., & Nikolaev, A. (2021). Fake review detection on online E-commerce platforms: a systematic literature review.
17. Wang, P., Wu, L., Aslam, B., & Zou, C. C. (2009, August). A systematic study on peer-to-peer botnets.
18. Ben Sassi, I., & Ben Yahia, S. (2021). Malicious accounts detection from online social networks: a systematic review of literature.
19. Pérez, J. Q., Daradoumis, T., & Puig, J. M. M. (2020). Rediscovering the use of chatbots in education: A systematic literature review.

20. Jung, H., Lee, J., & Park, C. (2020). Deriving Design Principles for Educational Chatbots from Empirical Studies on Human–Chatbot Interaction.
21. Bhirud, N., Tataale, S., Randive, S., & Nahar, S. (2019). A Literature Review On Chatbots In Healthcare Domain.
22. Syvänen, S., & Valentini, C. (2020). Conversational agents in online organization–stakeholder interactions: a state-of-the-art analysis and implications for further research.
23. Borsci, S., Malizia, A., Schmettow, M., Van Der Velde, F., Tariverdiyeva, G., Balaji, D., & Chamberlain, A. (2021). The Chatbot Usability Scale: the Design and Pilot of a Usability Scale for Interaction with AI-Based Conversational Agents.
24. Kaartemo, V., & Helkkula, A. (2018). A systematic review of artificial intelligence and robots in value co-creation: current status and future research avenues.
25. Ciupe, A., Meza, S., & Orza, B. (2020, September). Systematic Assessment of Interactive Instructional Technologies in Higher Engineering Education.
26. Orgeolet, L., Foulquier, N., Misery, L., Redou, P., Pers, J. O., Devauchelle-Pensec, V., & Saraux, A. (2020). Can artificial intelligence replace manual search for systematic literature? Review on cutaneous manifestations in primary Sjögren's syndrome.
27. Skinner, G., & Walmsley, T. (2019, February). Artificial intelligence and deep learning in video games a brief review.
28. Blitch, J. G. (1996). Artificial intelligence technologies for robot assisted urban search and rescue.
29. Balachandran, V. (2013, May). Reducing human effort and improving quality in peer code reviews using automatic static analysis and reviewer recommendation.
30. Kim, D., Nam, J., Song, J., & Kim, S. (2013, May). Automatic patch generation learned from human-written patches.
31. Thongtanunam, P., Kula, R. G., Cruz, A. E. C., Yoshida, N., & Iida, H. (2014, June). Improving code review effectiveness through reviewer recommendations
32. Beschastnikh, I., Lungu, M. F., & Zhuang, Y. (2017, May). Accelerating software engineering research adoption with analysis bots.
33. Guzman, E., Ibrahim, M., & Glinz, M. (2017, May). Mining twitter messages for software evolution.
34. Avellis, G., Harty, J., & Yu, Y. (2017, May). Towards mobile twin peaks for app development. In 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft) (pp. 189-193). IEEE.
35. Halder, R., Proença, J., Macedo, N., & Santos, A. (2017, May). Formal verification of ROSbased robotic applications using timed automata.
36. Urli, S., Yu, Z., Seinturier, L., & Monperrus, M. (2018, May). How to design a program repair bot? insights from the repairnator project.
37. Paikari, E., & Van Der Hoek, A. (2018, May). A framework for understanding chatbots and their future.
38. Bradley, N., Fritz, T., & Holmes, R. (2018, May). Context-aware conversational developer assistants.

39. Brown, C. (2019, May). Digital nudges for encouraging developer actions.
40. Moreno, D., Duenas, S., Cosentino, V., Fernandez, M. A., Zerouali, A., Robles, G., & Gonzalez-Barahona, J. M. (2019, May). SortingHat: wizardry on software project members.
41. Kavaler, D., Trockman, A., Vasilescu, B., & Filkov, V. (2019, May). Tool choice matters: JavaScript quality assurance tools and usage outcomes in GitHub projects.
42. van Tonder, R., & Le Goues, C. (2019, May). Towards s/engineer/bot: Principles for program repair bots.
43. Wyrich, M., & Bogner, J. (2019, May). Towards an autonomous bot for automatic source code refactoring.
44. Erlenhov, L., de Oliveira Neto, F. G., Scandariato, R., & Leitner, P. (2019, May). Current and future bots in software development.
45. Matthies, C., Dobrigkeit, F., & Hesse, G. (2019, May). An additional set of (automated) eyes: Chatbots for agile retrospectives.
46. Subramanian, V., Ramachandra, N., & Dubash, N. (2019, May). TutorBot: contextual learning guide for software engineers.
47. Kumar, R., Bansal, C., Maddila, C., Sharma, N., Martelock, S., & Bhargava, R. (2019, May). Building sankie: An ai platform for devops.
48. Sharma, V. S., Mehra, R., Kaulgud, V., & Podder, S. (2019, May). A smart advisor for software delivery-a bot framework for awareness, alerts and advice.
49. Marginean, A., Bader, J., Chandra, S., Harman, M., Jia, Y., Mao, K., ... & Scott, A. (2019, May). SapFix: Automated end-to-end repair at scale.
50. Pinheiro, A. M., Rabello, C. S., Furtado, L. B., Pinto, G., & de Souza, C. R. (2019, May). Expecting the unexpected: distilling bot development, challenges, and motivations.
51. Paikari, E., Choi, J., Kim, S., Baek, S., Kim, M., Lee, S., ... & Cheong, C. (2019, May). A chatbot for conflict detection and resolution.
52. Monperrus, M. (2019, May). Explainable software bot contributions: Case study of automated bug fixes.
53. Lin, C. T., Ma, S. P., & Huang, Y. W. (2020, June). MSABot: A chatbot framework for assisting in the development and operation of microservice-based systems.
54. Dominic, J., Houser, J., Steinmacher, I., Ritter, C., & Rodeghero, P. (2020, June). Conversational bot for newcomers onboarding to open source projects.
55. Dominic, J., Ritter, C., & Rodeghero, P. (2020, June). Onboarding Bot for Newcomers to Software Engineering.
56. Abdellatif, A., Costa, D., Badran, K., Abdalkareem, R., & Shihab, E. (2020, June). Challenges in chatbot development: A study of stack overflow posts.
57. Dey, T., Mousavi, S., Ponce, E., Fry, T., Vasilescu, B., Filippova, A., & Mockus, A. (2020, June). Detecting and characterizing bots that commit code
58. Cruzes, D.S., Dybå, T., Runeson, P., & Höst, M. (2014). Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example. *Empirical Software Engineering*, 20, 1634-1665.

59. D. S. Cruzes and T. Dyba, "Recommended Steps for Thematic Synthesis in Software Engineering," 2011 International Symposium on Empirical Software Engineering and Measurement, 2011, pp. 275-284, doi: 10.1109/ESEM.2011.36.

## 10. Apéndice A: Estudios Primarios

Identificador	Título	Autores	Año	DOI
PS-1	Automatic generation of software bug fixes based on analysis of software repositories	Belskii, A.; Itsykson, V. M.	2020	10.18721/JCSTCS.13204
PS-2	Bots Coordinating Work in Open Source Software Projects.	Hukal, Philipp; Berente, Nicholas; Germonprez, Matt; Schecter, Aaron	2019	10.1109/MC.2018.2885970
PS-3	Model-Based Test Oracle Generation for Automated Unit Testing of Agent Systems.	Padgham, Lin; Zhang, Zhiyong; Thangarajah, John; Miller, Tim	2013	10.1109/TSE.2013.10
PS-4	User stories collection via interactive chatbot to support requirements gathering.	Dwitam, Ferliana; Rusli, Andre	2020	10.12928/TELKOMNIKA.v18i2.14866
PS-5	Onboarding Bot for Newcomers to Software Engineering	James Dominic, Charles Ritter, Paige Rodeghero	2020	10.1145/3379177.3388901
PS-6	How to Design a Program Repair Bot? Insights from the Repairator Project	S. Urli; Z. Yu; L. Seinturier; M. Monperrus	2018	10.1145/3183519.3183540
PS-7	MSRBot: Using Bots to Answer Questions from Software Repositories	Ahmad Abdellatif, Khaled Badran, Emad Shihab	2020	10.1007/s10664-019-09788-5
PS-8	Context-Aware Conversational Developer Assistants	N. Bradley; T. Fritz; R. Holmes	2018	10.1145/3180155.3180238
PS-9	When Natural Language Processing Jumps into Collaborative Software Engineering	F. Gilson; D. Weyns	2019	10.1109/ICSA-C.2019.00049
PS-10	SapFix: Automated End-to-End Repair at Scale	A. Marginean; J. Bader; S. Chandra; M. Harman; Y. Jia; K. Mao; A. Mols; A. Scott	2019	10.1109/ICSE-SEIP.2019.00039

PS-11	Building Sankie: An AI Platform for DevOps	R. Kumar; C. Bansal; C. Maddila; N. Sharma; S. Martelock; R. Bhargava	2019	10.1109/ BotSE.2019.00020
PS-12	Sorry to Bother You: Designing Bots for Effective Recommendations	C. Brown; C. Parnin	2019	10.1109/ BotSE.2019.00021
PS-13	An Additional Set of (Automated) Eyes: Chatbots for Agile Retrospectives	C. Matthies; F. Dobrigkeit; G. Hesse	2019	10.1109/ BotSE.2019.00017
PS-14	Explainable Software Bot Contributions: Case Study of Automated Bug Fixes	M. Monperrus	2019	10.1109/ BotSE.2019.00010
PS-15	A Chatbot for Conflict Detection and Resolution	E. Paikari; J. Choi; S. Kim; S. Baek; M. Kim; S. Lee; C. Han; Y. Kim; K. Ahn; C. Cheong; A. van der hoek	2019	10.1109/ BotSE.2019.00016
PS-16	A Smart Advisor for Software Delivery - A Bot Framework for Awareness, Alerts and Advice	V. S. Sharma; R. Mehra; V. Kaulgud; S. Podder	2019	10.1109/ BotSE.2019.00014
PS-17	Should I Stale or Should I Close? An Analysis of a Bot That Closes Abandoned Issues and Pull Requests	M. Wessel; I. Steinmacher; I. Wiese; M. A. Gerosa	2019	10.1109/ BotSE.2019.00018
PS-18	Towards an Autonomous Bot for Automatic Source Code Refactoring	M. Wyrich; J. Bogner	2019	10.1109/ BotSE.2019.00015
PS-19	PTracer: A Linux Kernel Patch Trace Bot	Y. Wen; J. Cao; S. Cheng	2019	10.1109/ ASE.2019.00140
PS-20	RefBot: Intelligent Software Refactoring Bot	V. Alizadeh; M. A. Ouali; M. Kessentini; M. Chater	2019	10.1109/ ASE.2019.00081



PS-21	Progress Management Method for Software Development Project-based Learning Using Automated Teaching Assistants	M. Yamamoto; K. Fukuoka; R. Kiyohara; Y. Terashima	2019	10.23919/ ICMU48249. 2019.9006667
PS-22	Intelligent Chatbot for Requirements Elicitation and Classification	Chetan Surana Rajender Kumar Surana; Shriya; Dipesh B. Gupta; Sahana P. Shankar	2019	10.1109/ RTEICT46194. 2019.9016907
PS-23	Improving Feedback on GitHub Pull Requests: A Bots Approach	Zhewei Hu; Edward F. Gehringer	2019	10.1109/ FIE43999. 2019.9028685
PS-24	Reducing human effort and improving quality in peer code reviews using automatic static analysis and reviewer recommendation	Vipin Balachandran	2019	10.1109/ ICSE.2013.6606642
PS-25	Automatic Contract Insertion with CCBot	Scott A. Carr, Francesco Logozzo, and Mathias Payer	2017	10.1109/ TSE.2016.2625248
PS-26	APIBot: Question Answering Bot for API Documentation	Yuan Tian, Ferdian Thung , Abhishek Sharma, and David Lo	2013	10.1109/ ICSE-NIER.2017.17
PS-27	C-3PR: A Bot for Fixing Static Analysis Violations via Pull Requests	Antonio Carvalho, Welder Luz, Diego Marc'ilio, Rodrigo Bonifacio, Gustavo Pinto, and Edna Dias Canedo	2020	10.1109/ SANER48275. 2020.9054842
PS-28	Chatbot development using Java tools and libraries	Anita Ilić, Ana Ličina, Dušan Savić	2020	10.1109/ IT48810. 2020.9070294
PS-29	JITBot: An Explainable Just-In-Time Defect Prediction Bot	Chaiyakarn Khanan , Worawit Luewichana ,Krissakorn Pruktharathikoon, Jirayus Jiarpakdee , Chakkrit Tantithamthavorn Morakot Choetkiertikul , Chaiyong Ragkhitwetsagul , Thanwadee Sunetnanta	2020	10.13039/ 100000163-ARC
PS-30	Intent Classification for a Management Conversational Assistant	Georgios A. Dafoulas, Manal A. Ismail y Abdelrahman H. Hefny	2020	10.1109/ ICCES51560. 2020.9334685

## 11. Apéndice B: Conceptos y Términos de bots

### 11.1 Conceptos de bots

En la Tabla 19 Conceptos de bots se muestran los diferentes conceptos clave del dominio de bots de los cual se basa este MSL.

Tabla 19 Conceptos de bots

Concepto	Definición
Software Development	En español conocido como “desarrollo de software” se refiere a un conjunto de actividades de la ciencia computacional dedicadas al proceso de crear, diseñar, desplegar y mantener software de acuerdo con (IBM, 2014).
Bot	También conocidos como chatbots o software bots, en el ámbito de la ingeniería de software, se refiere a un grupo de tecnologías que conectan a un usuario con un servicio de software. (Lebeuf, 2018). Sinónimos incluyen asistentes y agentes de software
DevBot	Un Bot de desarrollo es definido como Bot que brinda una interfaz que conecta al desarrollador con servicios de software y adicionalmente aporta servicios adicionales. (Lebeuf, 2018)
Automation	Aplicación de tecnologías para producir y desplegar un servicio o producto con mínima intervención humana. (Techopedia, 2021)
Software Engineering Tool	Una herramienta de software también conocidos como herramienta de programación, se refiere a un producto de software que proporciona soporte automático o semiautomático para un desarrollador. (Kuhn, 1989)
Software Development Life Cycle	El modelo “Software Development Life Cycle” (SDLC) describe de manera genérica el desarrollo de software independiente de la metodología específica al desarrollo. De acuerdo con este modelo, el desarrollo de software está dividido entre siete etapas. Estas son: Planeación, Análisis de requisitos, Diseño, Programación, Prueba, Despliegue, Operación y Mantenimiento. (Jevtic, 2019)
Software Development Methodology	La metodología de desarrollo de software especifica las etapas y actividades a realizar para un proyecto de software dependiendo de las necesidades y restricciones específicas de una empresa de desarrollo y el producto de software a elaborar. Metodologías reconocidas en la industria incluyen Cascada, Desarrollo orientado a funcionalidad, Metodología Ágil, SCRUM y Programación Extrema. (Majewski, 2019)
Development Activities	Las actividades de desarrollo son tareas realizadas por un equipo de desarrollo durante el ciclo de vida de desarrollo de software. Cada etapa incluye actividades específicas como la creación de documentación, manuales de uso, diseño arquitectural, ejecución de pruebas unitarias, etc. Las actividades de desarrollo a realizar dependen de la metodología de desarrollo seleccionada a seguir durante el desarrollo.

## 11.2 Términos de bots

En la Tabla 20 Términos de bots se muestran los términos derivados de los conceptos clave del dominio de bots para la elaboración de cadenas de búsqueda para la identificación de estudios.

Tabla 20 Términos de bots

Concepto Relacionado	Término	Definición
Software Development, Software Development Life Cycle	Software Development	En español conocido como “desarrollo de software” se refiere a un conjunto de actividades de la ciencia computacional dedicadas al proceso de crear, diseñar, desplegar y mantener software de acuerdo con (IBM, 2014).
Bot DevBot	Bot	También conocidos como chatbots o software bots, en el ámbito de la ingeniería de software, se refiere a un grupo de tecnologías que conectan a un usuario con un servicio de software. (Lebeuf, 2018)
DevBot Software Engineering Tool	DevBot	Un Bot de desarrollo es definido como Bot que brinda una interfaz que conecta al desarrollador con servicios de software y adicionalmente aporta servicios adicionales (Lebeuf, 2018)
Bot DevBot	Assitant	Sinónimo de bot. También conocidos como “Smart Assitant” asistente inteligente o “Smart Agent”
Automation DevBot	Automation	Aplicación de tecnologías para producir y desplegar un servicio o producto con mínima intervención humana. (Techopedia, 2021)
Software Engineering Tool Development Activities	Software Analysis	Actividad de desarrollo donde se examinan artefactos de software para evaluar la calidad en base a criterios como: robustez, seguridad, rendimiento entre otros. (Freescale Semiconductor, 2005)
Development Activities Software Development Life Cycle	Software Coding	Actividad de desarrollo en donde se transforma el diseño del sistema a código fuente en un lenguaje de programación. Se debe tomar énfasis en que codificación no es programación sino una actividad dentro de Programación. (Kingsley, 2021)
Development Activities Software Development Life Cycle	Software Onboarding	Proceso de integración de un nuevo miembro a un equipo de desarrollo para adaptarlo a las actividades, proyectos y flujos de trabajo establecidos por equipo. (Digital Adoption, 2019)

