



Universidad Veracruzana

Facultad de Estadística e Informática

---

---

Protocolo del trabajo de investigación intitulado:  
"Análisis de la investigación en el uso de Bots  
para el desarrollo de software"

Modalidad:  
Monografía

Como requisito parcial para obtener el grado de:  
Licenciado en ingeniería de software

Presentan:  
Ricardo Moguel Sánchez  
César Sergio Martínez Palacios

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

Héctor Xavier Limón Riaño  
14/12/21

Xalapa, Ver. a 15 de Diciembre del 2021

**APROBADO**

Por Jorge Octavio Ocharán Hernández fecha 11:02 , 15/12/2021

# Contenido

1	ANTECEDENTES.....	2
2	DEFINICIÓN DEL PROBLEMA.....	5
3	MOTIVO.....	6
4	JUSTIFICACIÓN.....	7
5	DELIMITACIÓN DEL PROYECTO .....	8
6	ESQUEMA DE FUNDAMENTOS .....	9
6.1	INGENIERÍA DE SOFTWARE.....	9
6.1.1	<i>DESARROLLO DE SOFTWARE .....</i>	<i>9</i>
6.1.2	<i>CICLO DE VIDA DEL DESARROLLO DE SOFTWARE .....</i>	<i>10</i>
6.2	HERRAMIENTAS DE DESARROLLO DE SOFTWARE.....	11
6.3	TÉCNICAS DE I.A. APLICADAS AL DESARROLLO DE SOFTWARE .....	12
6.4	BOTS.....	13
6.4.1	<i>DEVBOTS .....</i>	<i>14</i>
6.4.2	<i>CHAT BOTS .....</i>	<i>14</i>
6.4.3	<i>NIVEL DE INTELIGENCIA DE BOTS .....</i>	<i>15</i>
7	MÉTODO .....	16
8	CONTENIDO DEL TRABAJO DE INVESTIGACIÓN.....	28
9	CRONOGRAMA .....	30
10	BIBLIOGRAFÍA .....	31

## 1 Antecedentes

Hoy en día el software se encuentra a niveles muy altos de exigencia en cuanto a calidad y funcionalidades, los software se vuelven cada vez más complejos y multidisciplinares, con todos estos retos en el camino de un desarrollador es que se han creado diferentes metodologías, disciplinas formales e incluso software nuevos para lidiar con los problemas causados por alta complejidad, según la investigación de Nagaria y Hall (2015) “Con muchos errores informados por los desarrolladores y son causados por el complejo entorno de desarrollo en el que se produce el desarrollo de software” (p7.).

Esta situación, impulsa la creación de tecnologías dirigidas a manejar y disminuir estos problemas que enfrentan los ingenieros y desarrolladores de software como la falta de comunicación entre equipos de desarrollo, las malas prácticas de programación, los cortos tiempos de desarrollo, entre otro; como mencionan Nagaria y Hall (2015) “En el corazón de muchos defectos de software está el error humano. Muchos de esos errores humanos los cometen los desarrolladores” (p.1).

Dentro del proceso de desarrollo de software se utilizan diferentes técnicas para poder aplicar a cada etapa del desarrollo y que el software resulte con una calidad esperada y cumpliendo los requisitos solicitados.

Con la meta de controlar complejidad en los procesos de desarrollo de software, muchos desarrollos parten con la meta de crear nuevas herramientas para apoyar a los programadores y desarrolladores como es el caso con Entornos de Desarrollo Integrados o IDEs por sus siglas en inglés y repositorios virtuales.

Gracias a la facilidad de estas herramientas los desarrolladores se han logrado enfocar en otros caminos, como por ejemplo han empezado a buscar lograr aplicaciones con un comportamiento inteligente, o sea que busque simular la inteligencia de una misma persona pero que este elaborada artificialmente por desarrolladores. De este tipo de aplicaciones inteligentes se deriva el término Inteligencia Artificial la cual define John MCarty (2007) como “La ciencia y la ingeniería de la fabricación de máquinas inteligentes que planean simular a la inteligencia humana”.

Con base en la inteligencia artificial es que se derivan diferentes tecnologías que se pueden clasificar, según Stuart J. Russell y Peter Norvig (2010), como: “Sistemas que piensan como humanos, sistemas que piensan racionalmente, sistemas que actúan racionalmente y sistemas que actúan como humanos” (p.2). Hablando de este último, es donde podemos clasificar a un tipo de aplicación inteligente llamada bot.

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 utilizar la inteligencia artificial para imitar el comportamiento y la inteligencia humanos.”

Analizando las palabras de Lebeuf sabemos 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 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 en su totalidad, ya que muchas personas logran confundir estos objetivos y pierden el enfoque a la hora de crear un DevBot.

Las actividades conocidas que pueden realizar un bot incluyen, pero no están limitadas a:

- Derivar y documentar requisitos por contratos (I. Qasse, 2021).
- Trazar atributos de calidad, hitos e información específica a un proyecto de software. (V. S. Sharma, 2019)
- Apoyar y mejorar mantenimiento de los sistemas de software por detección y corrección automática de bugs. (Monperrus, 2019)
- Apoyar y mejorar el proceso de refactorización de código. (V. Alizadeh, 2019)
- Apoyar y responder preguntas de los desarrolladores sobre un sistema. (V. Subramanian, 2019)
- Mejorar los procesos del desarrollo de software al apoyar el aprendizaje de estudiantes de ingeniería de software. (Parnin, 2021)

Identificando estas actividades consideramos la definición de un Devbot según Lebeuf (2013), “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”.

## 2 Definición del Problema

La ingeniería de software ya ha visto una adopción de Bots a un ritmo muy constante. Y según Margaret-Anne Storey y Alexey Zagalsky (2016) “Los bots ayudan a los desarrolladores a ser más productivos automatizando tareas tediosas, ayudando a los desarrolladores a estar al tanto de las actividades importantes de la comunidad o del proyecto y reduciendo las interrupciones” (p.928).

Distintos tipos de bots han sido generados con la intención de apoyar a un desarrollador de software y debido a esto es que se han logrado posicionar en el ojo de muchos desarrolladores, pero a pesar de que el término Bot se encuentra en plena culminación no existen muchos estudios y literatura sobre sus aplicaciones a diferentes áreas de trabajo en la ingeniería de software, hay una escasa cantidad de trabajos de investigación y de compilación de fuentes académicas que citen el uso práctico de bots a la ingeniería de software.

### **3 Motivo**

#### **Objetivo General**

El objetivo general de esta monografía es analizar la investigación actual en el uso de bots para el desarrollo de software con el fin de contar con una descripción y organización del estado del arte a través de una síntesis temática. Derivado de esto se desea identificar diferentes DevBots que tengan algún tipo de aplicación dentro de la ingeniería de software.

#### **Objetivos Específicos**

A continuación, se encuentran los objetivos específicos en orden secuencial para cumplir con el objetivo general de este trabajo.

- Identificar las investigaciones que reporte el uso de bots para el desarrollo de software a través de una selección y búsqueda utilizando bases de datos científicas.
- Sintetizar los hallazgos en las investigaciones seleccionadas para la identificación de patrones a través de la extracción y codificación con un mapeo sistemático de la literatura.
- Evaluar la confiabilidad de la síntesis para una correcta interpretación de los resultados a través de una valoración de la calidad y cantidad de los hallazgos.
- Identificar líneas de investigación que apoyen en la toma de decisiones de investigadores interesados en DevBots.

## 4 Justificación

El objetivo de este trabajo de investigación es la consolidación de información académica que reporta usos de bots en el proceso de desarrollo. De acuerdo con C. Lebeuf et al. (2019) es importante identificar las similitudes y diferencias entre software bots para establecer pautas sobre el uso de bots dirigidos a desarrolladores. Basándose en esto, esperamos que desarrolladores podrán usar esta literatura para ayudarlos a tomar una decisión sobre que tecnología Bot utilizar para sus futuros proyectos de software. Adicionalmente, nuestro trabajo servirá de guía para futuros investigadores de herramientas DevBots. Las líneas de investigación surgidas del análisis de los usos de tecnologías bot en el ciclo de desarrollo de software servirán de guía para futuros trabajos necesarios para combatir la carencia de artículos académicos.

Erlendov et. al. (2019) concluye que el futuro de DevBots está en la integración de Inteligencia Artificial a los Bot para crear desarrolladores artificiales en vez de herramientas de desarrollo. Con este enfoque en mente, un análisis de la aplicación de Bots en el desarrollo de software es necesario para medir cuánto progreso hay a esta idealidad.

Según nuestro conocimiento, no existe una fuente de literatura sobre bots enfocados al desarrollo de software. Una cantidad significativa de artículos en academia vienen del taller International Workshop on Bots in Software Engineering (BotSE) inaugurado recientemente en 2019 con la meta de describir nuevos resultados y hacer contribuciones al cuerpo de conocimiento en el área de bots para la ingeniería de software. Lebeuf (2013) concluye que se requieren investigaciones futuras que consideren ambas la complejidad y diversidad de software bots. Por estas razones, la elaboración de una monografía se ve necesaria debido a la carencia de fuentes sobre patrones de uso de tecnologías bot enfocados al desarrollo.



## 5 Delimitación del proyecto

### **Alcance**

Para el alcance de esta investigación de bots aplicados al desarrollo de software se el alcance esta abarcado en los siguientes puntos.

- Únicamente se considerarán trabajos de investigación en literatura blanca en donde se reportan el uso de bots para el desarrollo de software.
- Únicamente se contemplarán literatura blanca en bases de datos científicas disponibles a la Universidad Veracruzana a través de CONRICYT.
- Únicamente se considerarán estudios que cumplan con un mínimo de calidad determinado a través de un proceso de evaluación de calidad.
- Únicamente se considerarán estudios publicados entre 2011 y 2020 para poder evaluar estudios recientes.
- Únicamente se recopilarán líneas de investigación que se mencionan por autores de estudios primarios.

### **Limitaciones**

En los siguientes puntos se mencionan las restricciones debido a la naturaleza del trabajo de investigación a realizar.

- El tema del uso de bots en el desarrollo de software es un tema recientemente investigado a profundidad por lo cual es probable que se encuentre menos información de la esperada.
- No se puede controlar la calidad de estudios primarios identificados durante el proceso de búsqueda.
- Solo se está contemplando dos procesos de búsqueda. El Proceso Sistemático de Búsqueda desarrollado por H. Zhang et al. (2009) y el proceso de Snowballing adaptado para Ingeniería de Software de Wohlin (2014) debido a que solo tenemos dos miembros en el equipo de investigadores.

## 6 Esquema de fundamentos

Existen conceptos fundamentales que se deben contextualizar antes de la investigación sobre bots aplicados al desarrollo de software. Por iniciar se remarcan fundamentos relevantes a este trabajo de la ingeniería de software.

### 6.1 Ingeniería de software

La Ingeniería de software ha tenido un impacto muy grande a nivel mundial en los últimos años, se sabe que la ingeniería de software no es una rama de la informática; es una disciplina de la ingeniería que se basa en parte en la informática como menciona (L. Briand,2012). “Esta disciplina se encarga de asegurar una alta calidad en el software que se desarrolle por medio de un preciso y arduo proceso de desarrollo de software”.

Además de que maneja múltiples técnicas y herramientas para la construcción de software y se requiere de un equipo de alto nivel para llevar a cabo todo el proceso por el que pasa un software, todo un procedimiento sistemático previamente establecido como menciona Leach, R. J. (2018) en su libro “El término ingeniería de software se refiere a un procedimiento sistemático que se utiliza en el contexto de un conjunto de objetivos generalmente aceptados para el análisis, diseño, implementación, prueba y mantenimiento de software” y a este proceso riguroso se le puede conocer como Desarrollo de software

#### 6.1.1 Desarrollo de software

El principal objetivo de la ingeniería de software es desarrollar software de alta calidad estableciendo un proceso a seguir aplicando diferentes técnicas existentes y aplicando el uso de múltiples herramientas para poder facilitar su desarrollo. Y según IBM (2014) “Un 90 por ciento de la tecnología orientada a los servicios depende del uso de habilidades de desarrollo de software. Aproximadamente el 80 por ciento de la tecnología orientada a sistemas y el 5 por ciento de los fabricantes de hardware también utilizan estas habilidades para crear modelos y desarrollar ideas”.

Además de que este proceso se elabora comúnmente por un equipo completo de desarrolladores, como menciona Leach, R. J. (2018) en su libro, “Incluso si una sola persona entendiera todo el código fuente y los archivos de datos relacionados necesarios para una nueva versión del software, no habría tiempo suficiente para realizar los cambios necesarios de manera oportuna. Por lo tanto, la naturaleza competitiva del mercado de software y la complejidad de los productos en sí obligan esencialmente al empleo de equipos de desarrollo de software”. Todo esto se hace llevando un ciclo de vida para el software, así se definen etapas y se trabaja en un orden específico para desarrollar el software

### **6.1.2 Ciclo de vida del desarrollo de software**

Debido a que el desarrollo de software es una actividad demasiado grande se ha dividido este trabajo en varias tareas, que incluyen realizar cambios en la funcionalidad, cambiar el entorno, corregir errores y realizar mejoras para evitar problemas futuros (V. T. Rajlich and K. H. Bennett, 2000). A lo largo de los años han existido muchas propuestas para definir un ciclo de vida del desarrollo de software, esto sucede porque el software logra ser un producto de múltiples objetivos y con muchos enfoques, pero hoy en día el modelo más popular y utilizado por los ingenieros de software es el siguiente:

- Elicitación de requerimientos
- Diseño del software
- Codificación del software
- Pruebas del software
- Publicación del software
- Mantenimiento de software

Cada etapa se ejecuta en el orden mencionado, es decir, se empieza en la Elicitación de requerimientos y termina en el Mantenimiento de software, aplicando diferentes técnicas y utilizando múltiples herramientas para lograr una ejecución perfecta de la fase del ciclo de vida de software.

## 6.2 Herramientas de desarrollo de software

Conociendo todo el proceso por el que pasa un ingeniero de software es que se busca la creación de herramientas que faciliten el proceso, como menciona Robbins, J. E. (1999) “La mayoría de las herramientas de diseño de software brindan soporte para editar, ver, almacenar, compartir y transformar”.Partiendo de esto es que se ha estado buscando herramientas, que en su mayoría es más software, para apoyar a un ingeniero de software en cada etapa del ciclo de vida de software. Como menciona C. Prasad and W. Schulte (2013), “Los ingenieros de software de hoy tienen acceso a una amplia gama de herramientas de desarrollo comerciales y caseras, y continuamente surgen nuevas.”

Entre todas estas Herramientas se encuentra algunas que aplican diferentes técnicas de inteligencia artificial con diferentes propósitos y en múltiples etapas, además de que estas pueden clasificarse según Riddle, W. E., & Fairley, R. E. (2012) en su libro, “Las herramientas para el desarrollo de software se pueden clasificar como herramientas cognitivas, herramientas aumentativas y herramientas de notación”. Basado en la clasificación anteriormente mencionada tenemos las siguientes definiciones:

- Las herramientas de notación proporcionan los medios en los que las ideas pueden expresarse y comunicarse.
- Las herramientas cognitivas son las técnicas de resolución de problemas que se utilizan para desarrollar sistemas de software; incluyen técnicas como la descomposición jerárquica, el ocultamiento de información, la codificación estructurada y las pruebas sistemáticas.
- Las herramientas de aumento son sistemas de software e incluyen herramientas orientadas a la implementación, como compiladores y paquetes de depuración y en estas últimas se pueden encontrar herramientas que aplican técnicas de inteligencia artificial

### **6.3 Técnicas de Inteligencia Artificial aplicadas al desarrollo de software**

Como mencionan Krishnamoorthy, C. S., & Rajeev, S. (2018) en su libro “La adaptación de la resolución de problemas a diferentes entornos y requisitos se puede lograr fácilmente utilizando técnicas proporcionadas por IA y sistemas expertos”.

La inteligencia Artificial se ha encontrado un papel importante en el desarrollo de software, brindando múltiples herramientas para facilitar el proceso del ciclo de vida de desarrollo. Según M. Harman (2012) “Se ha utilizado tres áreas amplias de técnicas de IA:

- 1) Técnicas de optimización y búsqueda computacional (campo conocido como Ingeniería de software basada en búsquedas (SBSE)).
- 2) Técnicas de métodos borrosos y probabilísticos de razonamiento en presencia de incertidumbre.
- 3) Técnicas de clasificación, aprendizaje y predicción.

Existen muchos derivados de la Inteligencia artificial debido a que es una ciencia muy extensa que hoy en día sigue expandiéndose más, de esta ciencia nacen los bots que según Lebeuf (2018) “Los bots han demostrado ser un gran banco de pruebas para que los investigadores experimenten con muchos de los avances derivados de la investigación de la IA.”

## 6.4 Bots

Como se ha mencionado al inicio de este trabajo el concepto de bots se sigue teniendo muy ambiguo actualmente, y en palabras de C. Lebeuf (2018) “La investigación sobre tecnologías de bots de software abarca múltiples áreas de la informática, así como otras disciplinas, en particular ciencia de decisiones, biología, sociología y muchas otras. Esto sirve para aumentar aún más la dificultad de identificar qué constituye exactamente un bot.”

Existen muchas definiciones de bots propuestas y ya hemos mencionado la definición propuesta por C. Lebeuf (2018) anteriormente por lo que se identifica como los bots llegan a tener múltiples objetivos y propósitos causando que estos se clasifiquen en diferentes tipos y hasta se encuentran diferentes tipos de clasificaciones de bots ya sea se clasificando en base a su conjunto de comportamientos especializados o también se ha intentado clasificar los bots de software según su propósito o función.

Basándose en la clasificación en base a su conjunto de comportamientos especializados, debido a que es una clasificación de alto nivel según C. Lebeuf (2018), podemos clasificarlos en:

- Agentes de Software
- DevBots
- ChatBots
- Web bots
- Agentes encarnados
- Agentes inteligentes

Estos bots, inclusive, no llegan a ser la principal herramienta para apoyar al ciclo de vida del desarrollo de software, sino que también usualmente integran con frecuencia servicios secundarios en los canales de comunicación, proporcionando un conducto entre los usuarios y otras herramientas de apoyo. Además de que

existen múltiples propuestas para clasificar su inteligencia basándose en diferentes parámetros y utilidades que brinda el bot.

#### **6.4.1 Devbots**

Esta clasificación de bots se les conoce así principalmente por que apoyan al desarrollo de software y se enfocan en diferentes etapas del ciclo de vida de este. Este término es realmente nuevo, pero a pesar de su reciente nacimiento han logrado generar un gran interés por parte de investigadores y desarrolladores, en palabras de L. Erlenhov, F. Gomes de Oliveira Neto, R. Scandariato and P. Leitner (2019) "Los bots que apoyan el desarrollo de software ("DevBots") se consideran un enfoque prometedor para hacer frente a la complejidad cada vez mayor de la ingeniería y el desarrollo de software moderno."

Cabe aclarar que es posible que otros tipos de bots, como chatbots, encajen en esta clasificación debido a que si se apoya al desarrollo de software se puede llegar a clasificar como un DevBot.

#### **6.4.2 Chat bots**

Estos datan su origen desde 1950 con Alan Turing con su "Test de Turing" el cual consistía en simular una persona por medio de una maquina inteligente. Y tuvo que pasar un tiempo para llegar a lograr encontrar un término a este tipo de máquinas inteligentes definiéndolo como un chatbot y según las palabras de T. Zemčík (2019), "Un chatbot es un programa con un cierto nivel de inteligencia artificial, que se comunica con una persona u otro chatbot para dar al observador de la conversación la impresión de que se trata de una conversación con una persona real".

Los chatbots pueden tomar un papel importante como herramienta de desarrollo de software y su impacto dependerá de cómo es que se utiliza el chatbot y cuál es su objetivo.

### 6.4.3 Nivel de inteligencia de bots

Es bien sabido que los bots logran tener una inteligencia para funcionar y ejecutar sus funciones para el apoyo del desarrollador, pero algunos desarrolladores se han llegado a preguntar o han intentado resaltar la siguiente pregunta ¿Cuál es el nivel de inteligencia que tiene el bot?

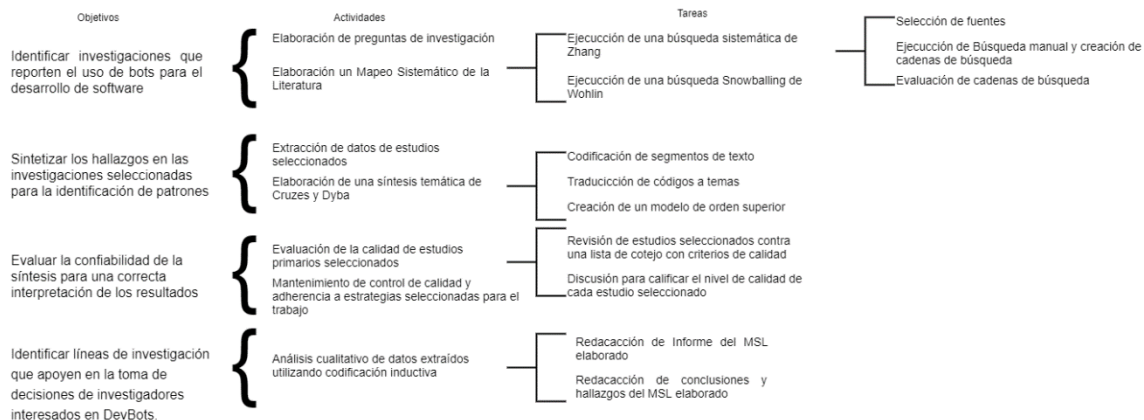
Si ya es bien conocido que el termino bot se encuentra en plena novedad ahora encontrar una clasificación de la inteligencia de estos es complicado, existen varias propuestas para clasificar a un bot y para este trabajo contemplamos principalmente la propuesta de Carlene Lebeuf, Margaret-Anne Storey, y Alexey Zagalsky (2018) la se basa en lo siguiente:

- Adaptación: Algunos bots son sensibles al contexto y pueden usar ese contexto para cambiar la forma en que interactúan con los usuarios.
- Razonamiento: Algunos bots siguen reglas lógicas simples; otros usan IA más avanzada para impulsar su comportamiento.
- Autonomía: Algunos bots son completamente autónomos, algunos dependen de la participación humana antes de actuar y otros utilizan un enfoque mixto.



## 7 Método

En la Figura 1 se detallan las actividades de acuerdo con los objetivos establecidos en Motivo. Como se puede observar, para el primer objetivo: “Identificar las investigaciones que reporten el uso de bots para el desarrollo de software” se contienen dos actividades.



**Figura 1 Método de investigación**

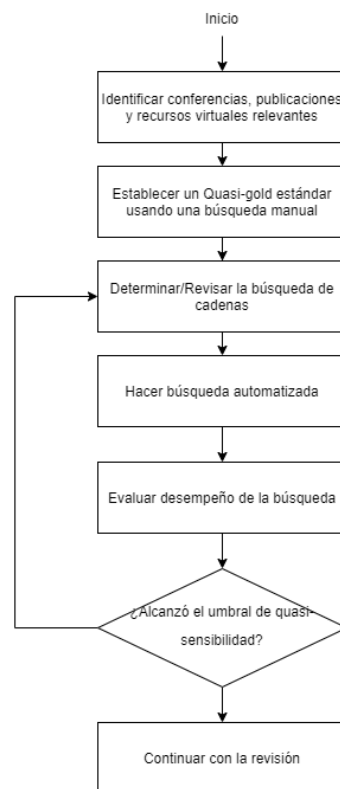
Estas actividades son “Elaboración de preguntas de investigación y Elaboración de un Mapeo Sistemático de la Literatura.

El paso primordial de este estudio secundario es la elaboración de preguntas de investigación. Antes de identificar posibles estudios primarios a analizar, es necesario redactar preguntas que pretende responder este trabajo por medio de una investigación. Armados con unas preguntas de investigación, se puede identificar el tipo de estudio secundario a realizar.

En el caso de este trabajo, se decidió realizar un Mapeo Sistemático de la Literatura (MSL) para la monografía por las siguientes razones. En primera, no existen tanto estudios de los cuales se puede realizar una investigación profunda como es demandado en una Revisión Sistemática de la Literatura (RSL). En segunda, el objetivo general de este trabajo es identificar y analizar de manera general los distintos usos de bots en actividades del desarrollo de software. Acorde al objetivo general no se requiere de una profundización rigurosa de trabajos

específicos como en un RSL sino un modelado de temas relacionados en el área de bots de software posible por medio de un MSL. Finalmente, para realizar el mapeo sistemático se ejecutarán dos fases. En la primera Fase se realizará una búsqueda automatizada siguiendo el Proceso Sistemático de Búsqueda basado en un Estándar Pseudo-Oro. (Zhang et al., 2011) En la segunda Fase se realizará una búsqueda por medio de una búsqueda Snowballing para estudios secundarios adaptada a la Ingeniería de Software. (Wohlin, 2014)

En la Figura 2, se describe el proceso de obtención del Estándar Quasi-gold para el Proceso Sistemática de Búsqueda previamente mencionado de Zhang et. al. (2011)



**Figura 2 Proceso de obtención de Estándar Quasi-gold adaptado de Zhang H., Babar M., & Tell P. (2011)**

El proceso de búsqueda de Zhang comienza con la selección de fuentes. 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 las 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.

De acuerdo con la construcción de un Estándar Quasi-gold se debe ejecutar una búsqueda manual en un conjunto limitado de revistas de temas específicos, el conjunto de artículos relevantes encontrados se utilizará luego para crear cadenas de búsqueda. Después, las cadenas de búsqueda elaboradas se ingresan a un motor de búsqueda para evaluar su rendimiento conforme al Estándar Quasi-Gold de Zhang. La evaluación del rendimiento se determina en base a los criterios de Retorno y Precisión.

El retorno de una búsqueda es la proporción (o porcentaje) de todos los estudios relevantes que se encuentran en la búsqueda, mientras que 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.

Entonces, conforme a los criterios de Retorno y Precisión definidos se evalúa el rendimiento de las cadenas de búsqueda contra los estudios relevantes obtenidos de la búsqueda manual. Cabe resaltar que el Estándar Quasi-Gold 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.

Una vez concluida la primera Etapa de la búsqueda, para la segunda Fase del procedimiento de búsqueda, se realizará una búsqueda Snowballing siguiendo los pasos de Wohlin (2014) resumidos en la siguiente figura.

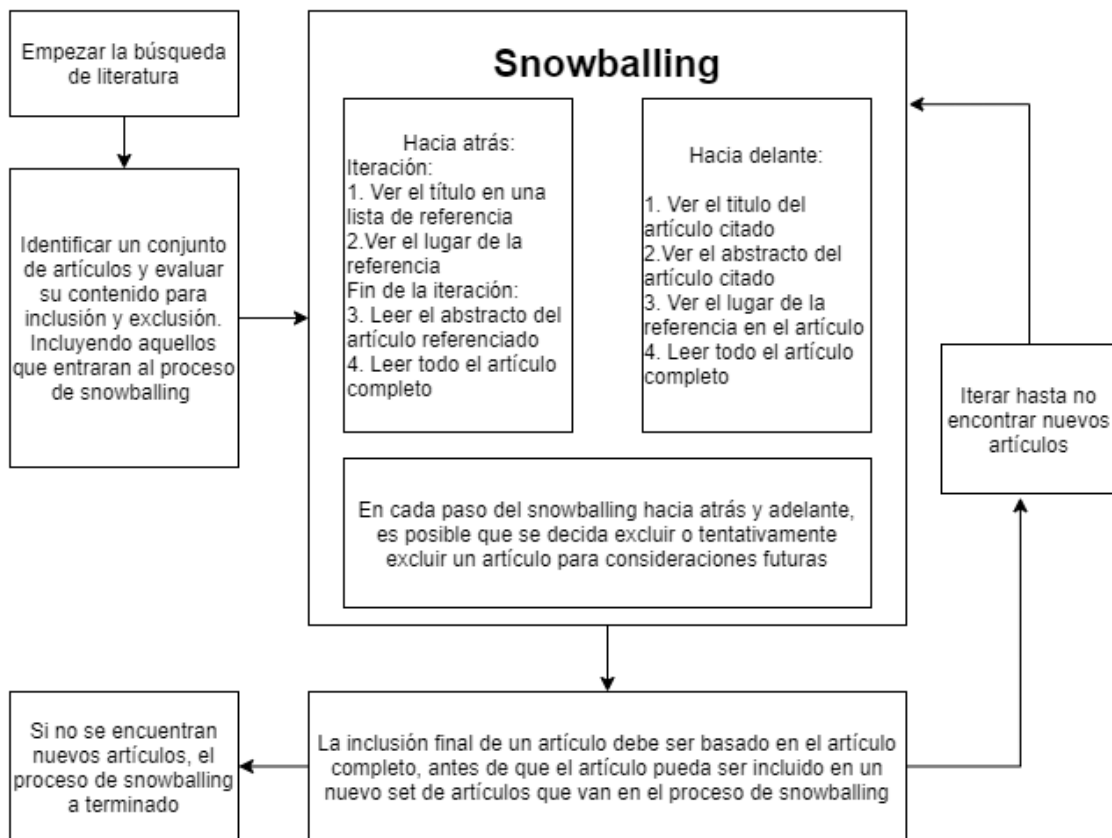


Figura 3 Proceso de búsqueda Snowballing de Wohlin C. (2014)

Primero, se hace un proceso de Snowballing hacia atrás para extraer la mayor cantidad de información posible del documento, aplicando los criterios de exclusión e inclusión, que se está examinando y no ir al nuevo documento hasta que no haya más información disponible en el documento que se está examinando mientras nos hacemos las siguientes preguntas:

- Título: ¿Es provisionalmente un artículo para incluir?
- Lugar de publicación: ¿Se publica en un lugar donde sea relevante?
- Autores: ¿Los autores han publicado artículos relevantes en el área estudiada anteriormente?

Después a el documento se le aplicara Snowballing hacia delante, identificación de nuevos artículos basados en aquellos artículos que citan el artículo examinando. Se examinará el abstracto y en caso de que sea insuficiente se examina el lugar que cita al documento, y en un último caso que esto no brinde información suficiente se termina analizando el artículo completo. Todo esto aplicando los mismos criterios que en el Snowballing hacia atrás.

Para el segundo objetivo: “Sintetizar los hallazgos en las investigaciones seleccionadas para la identificación de patrones” se contienen dos actividades. Estas actividades son la extracción de datos y elaboración de síntesis temática. Para cumplir con este objetivo primero y siguiendo los pasos establecidos por el método para estudios secundarios en la ingeniería de software de Kitchenham et. al. (2015)” Siempre que sea posible, la extracción de datos debe ser realizada de forma independiente por dos o más investigadores. Los datos de los investigadores deben compararse y los desacuerdos se deben resolver por consenso entre los investigadores o mediante el arbitraje de un investigador independiente adicional.”

Para la extracción de datos se realizará un proceso independiente con un equipo conformado por:

- Director: Dr. Jorge Octavio Ocharán Hernández
- Co director: Dr. Xavier Limón Riaño
- Investigadores: César Sergio Martínez Palacios y Ricardo Moguel Sánchez

Después de completar el proceso de extracción de datos se comparará y contrastará resultados en una discusión posterior. Desacuerdos no resueltos serán debatidos hasta llegar a un consenso. En el caso de no llegar a un consenso se consultará a los directores para resolver el desacuerdo. Al realizar la discusión se disminuyera los riesgos asociados con la calidad de la literatura seleccionada. Estos riesgos incluyen el sesgo de confirmación que al tener dos investigadores es reducido. Al extraer los datos de la búsqueda automatizada se usarán hojas de cálculo de Microsoft Excel en formato CSV descargadas de los motores de búsquedas. En la siguiente tabla se muestra el formato en el que se extraerá la información necesaria de cada estudio primario durante el procedimiento de búsqueda:

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

Siguiendo el proceso de ingeniería de software de Kitchenham et. al. (2015), después de la extracción de datos se elabora una síntesis de datos. Para el MSL planeado se ha seleccionado realizar una Síntesis Temática. De acuerdo con el proceso de Síntesis Temática de Creswell adaptada para la Ingeniería de Software (Cruzes & Dybå, 2011) los pasos a seguir para la síntesis de datos son los siguientes:

1. Identificar segmentos específicos

Una vez concluida la extracción de datos, se identificarán los segmentos de texto en donde se encuentran los datos para la extracción. Estos datos estarán concentrados dentro del formato de extracción de datos producido de cada estudio restante después de la extracción de datos. Entonces los segmentos consistirán en los siguientes datos: Tipo de Bot, Nivel de inteligencia del Bot, Objetivo del Bot, Etapa de desarrollo y Beneficios reportados del uso del Bot.

2. Etiquetar segmentos

Una vez identificados los segmentos, se asignarán nombres únicos bajo un esquema dependiendo del nombre de cada tipo de dato. Cada segmento tendrá una etiqueta única que consiste en dos partes, el tipo de dato y un número asignado de manera secuencial. Por ejemplo, las etiquetas seguirán el patrón: Tipo-01, Tipo-02, ... Tipo-N.

3. Traducir etiquetados a temas

Después de la etiquetación, se agruparán los segmentos bajo temas principales bajo los cuales segmentos tienen áreas en común. Estas áreas en común son conocidas como patrones o temas.

#### 4. Categorizar temas

Al tener un conjunto de temas, se realizará un análisis y consecuente categorización en donde los temas serán agrupados en los temas de mayor importancia y relevancia con nuestras preguntas de investigación. En este paso se incrementa el nivel de abstracción de los temas.

#### 5. Elaborar síntesis temática

Al obtener temas de un alto nivel de abstracción, se pueden establecer relaciones entre temas dentro de un modelo. En nuestro caso el modelo es un mapa utilizado para responder las preguntas de investigación.

Aparte del proceso de búsqueda y síntesis de datos, se contempla incluir mecanismos para asegurar la calidad de estudios identificados y del proceso de MSL a ejecutar para cumplir el tercer objetivo: "Evaluar la confiabilidad de la síntesis para una correcta interpretación de los resultados". Con este objetivo en mente se realizará una evaluación de calidad de estudios primarios con el fin de descubrir el nivel de importancia que tienen los estudios seleccionados durante el procedimiento de búsqueda de la literatura. Para la evaluación de calidad de estudios primarios se seguirán los pasos del método de Kitchenham et. al. (2015). Las etapas para esta fase del trabajo incluyen:

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

Se realiza 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 verificación a usar en este proceso fue propuesta por Dybå, T. y Dingsøy, T. (2008a) con su lista de verificación de calidad que se puede utilizar en varios tipos de estudios.



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

1. ¿El documento se basa en una investigación (o es simplemente un informe de "lecciones aprendidas" basado en la opinión de expertos)? (1 pt.)
2. ¿Existe una declaración clara de los objetivos de la investigación? (1 pt.)
3. ¿Existe una descripción adecuada del contexto en el que se realizó la investigación? (1 pt.)
4. ¿El diseño de la investigación fue apropiado para abordar los objetivos de la investigación? (1 pt.)
5. ¿La estrategia de contratación fue adecuada a los objetivos de la investigación? (0.5 pt.)
6. ¿Había un grupo de control con el que comparar los tratamientos? (0.5 pt.)
7. ¿Se recopilaron los datos de una manera que abordará el tema de la investigación? (1 pt.)
8. ¿Fue el análisis de datos suficientemente riguroso? (1 pt.)
9. ¿Se ha considerado adecuadamente la relación entre el investigador y los participantes? (0.5 pt.)
10. ¿Existe una declaración clara de los hallazgos? (1 pt.)
11. ¿Tiene valor el estudio para la investigación o la práctica? (1.5 pt.)

Conociendo las puntuaciones podemos hacer la siguiente clasificación de puntaje con la lista de verificación:

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 es parte importante para el desarrollo del trabajo de investigación. Junto con un proceso de evaluación de calidad, para asegurarse que se están siguiendo todos los pasos de las estrategias seleccionadas y de manera correcta se realizaran discusiones junto con los directores del trabajo de investigación conforme se elaboren las actividades de este estudio establecidas en Método.

Considerando el último objetivo: “Identificar líneas de investigación que apoyen en la toma de decisiones de investigadores interesados en DevBots”, está planeado una actividad principal con dos tareas. Para cumplir con este objetivo se realizará un análisis cualitativo de datos extraídos utilizando una codificación inductiva. Los temas extraídos de la síntesis de datos de estudios seleccionados serán utilizados por medio de una codificación inductiva para identificar líneas de investigación para trabajo futuro.

La codificación cualitativa es un proceso sistemático para identificar temas y patrones desde una fuente de datos cualitativos extraídos de una búsqueda. En el caso de nuestra investigación de bots aplicados al desarrollo de software se realizará una codificación inductiva en la cual se extraen códigos resultantes de una obtención de datos para identificar líneas de investigación de los temas y patrones producidos durante el análisis. Para realizar este análisis cualitativo mediante una codificación inductiva, se redactará un informe del proceso de elaboración del MSL desde la recolección de datos por medio de una búsqueda automática y extracción de datos hasta el análisis ejecutado durante la síntesis de datos. Debido a la síntesis temática seleccionada para el tipo de síntesis, surgirán como resultados códigos de los cuales se obtendrán temas y patrones a plasmar en un modelo de orden superior.

Entonces durante la elaboración del informe es clave mostrar claramente una buena trazabilidad para brindarle al lector un claro vínculo desde los datos crudos de estudios primarios hasta la codificación inductiva para poder responder las preguntas de investigación planteadas al inicio de este trabajo de investigación.

Se debe lograr llegar a la audiencia objetiva, las cuales son principalmente investigadores e interesados en bots dentro de la ingeniería de software, publicando en conferencia: International Conference in Software Engineering Research and Innovation (CONISOFT) con la ayuda y guía de nuestro director.

También demostrar una repetibilidad asegurándonos de que la metodología se definió y detallo claramente a diferencia de otros estudios. Siguiendo la metodología de Kitchenham et. al. (2015) se realizará un reporte de estudios con la siguiente estructura:

Punto del informe	Descripción
Abstracto	Ayudará a la selección, proporcionando al lector (o motor de búsqueda) suficiente información para sugerir relevancia.
Introducción	Establecerá un contexto y deja en claro por qué el estudio es una contribución útil a el campo de investigación y por qué una revisión sistemática es apropiada.
Trabajo Previo	Ampliará la descripción proporcionada en la introducción y proporcionará información sobre estudios anteriores y sus contribuciones.
Método	Incluirá los elementos centrales del protocolo de investigación justificando la elección del tipo de revisión y el plan.
Conducción	Resaltara cualquier divergencia con el plan, así como proporcionara información sobre acuerdos entre el equipo sobre cuestiones como la inclusión y la exclusión
Resultados	Aquí se describirán los resultados de la búsqueda y la inclusión / exclusión, así como de la extracción de datos.
Análisis	Redactara los resultados del proceso de síntesis
Discusión	Interpretará y considerará todos los resultados del análisis.
Conclusión	Finalmente abordara qué tan bien se han respondido las preguntas de investigación y cuáles son las respuestas que se obtuvieron.

Con este informe a realizar se busca:

- Responder las preguntas de investigación de una manera clara y completa
- Informar correctamente la metodología de la investigación
- Brindar una trazabilidad completa desde las preguntas hasta los datos obtenidos

Finalmente, después de realizar el mapeo sistemático de literatura se deben resumir en un apartado las conclusiones y hallazgos derivados de los datos resultantes del MSL. De los datos resultantes se espera principalmente un Mapa Temático resultante de la síntesis temática de Cruzes & Dybå (2011) a realizar. Junto con los resultados se deben plasmar las conclusiones resultantes que incluyen una categorización de bots conforme a tres diferentes medidas. Estas son: actividad en la cual se usa un bot, tipo de bot y nivel de inteligencia del bot, y los factores humanos, incluyendo beneficios reportados, que conducen el uso de bots.

## 8 Contenido del trabajo de investigación

En los siguientes puntos se mencionan los apartados contemplados para esta monografía.

- 1 Introducción
  - 1.1 Antecedentes
  - 1.2 Definición del problema
  - 1.3 Motivo
    - 1.3.1 Objetivo General
    - 1.3.2 Objetivos Específicos
  - 1.4 Justificación
  - 1.6 Delimitación del proyecto
    - 1.6.1 Alcance
    - 1.6.2 Limitaciones
- 2 Marco Teórico
  - 2.1 Ingeniería de software
    - 2.1.1 Desarrollo de software
    - 2.1.2 Ciclo de vida del desarrollo de software
  - 2.2 Herramientas de desarrollo de software
  - 2.3 Técnicas de Inteligencia Artificial aplicadas al desarrollo de bots
    - 2.3.1 Procesamiento de lenguaje natural
  - 2.4 Bots
    - 2.4.1 Devbots
      - 2.4.1.1 Chat bots
      - 2.4.1.2 Review bots
      - 2.4.1.3 Repair bots
      - 2.4.1.4 Analysis bots
    - 2.4.2 Nivel de inteligencia de bots

- 3 Conducción del método de investigación
  - 3.1 Preguntas de investigación
  - 3.2 Procedimiento de búsqueda
    - 3.2.1 Conceptos de búsqueda
    - 3.2.2 Términos de búsqueda
    - 3.2.3 Cadenas de búsqueda
    - 3.2.4 Selección de fuentes
  - 3.3 Selección de estudios primarios
    - 3.3.1 Criterios de selección
    - 3.3.2 Procedimiento de selección
    - 3.3.3 Estudios primarios seleccionados
  - 3.4 Evaluación de calidad de estudios primarios
  - 3.5 Extracción de los datos
  - 3.6 Síntesis Temática
  - 3.7 Resultados de síntesis temática
- 4 Capitulo por tema de síntesis temática
- 5 Discusión
- 6 Validez del estudio y limitantes
  - 7.1 Confiabilidad de la búsqueda de literatura
  - 7.2 Confiabilidad de la selección de estudios
  - 7.3 Confiabilidad de la extracción de datos y clasificación
  - 7.4 Confiabilidad de la síntesis de datos
- 7 Conclusiones y trabajo futuro
- 8 Referencias
- 9 Apendices

## 9 Cronograma

En seguida se muestra el cronograma de actividades a realizar para la investigación planteada. Estas actividades incluyen la elaboración de 3 artefactos principales. Estos son: Protocolo de la investigación, Revisión Sistemática de la literatura y Monografía.

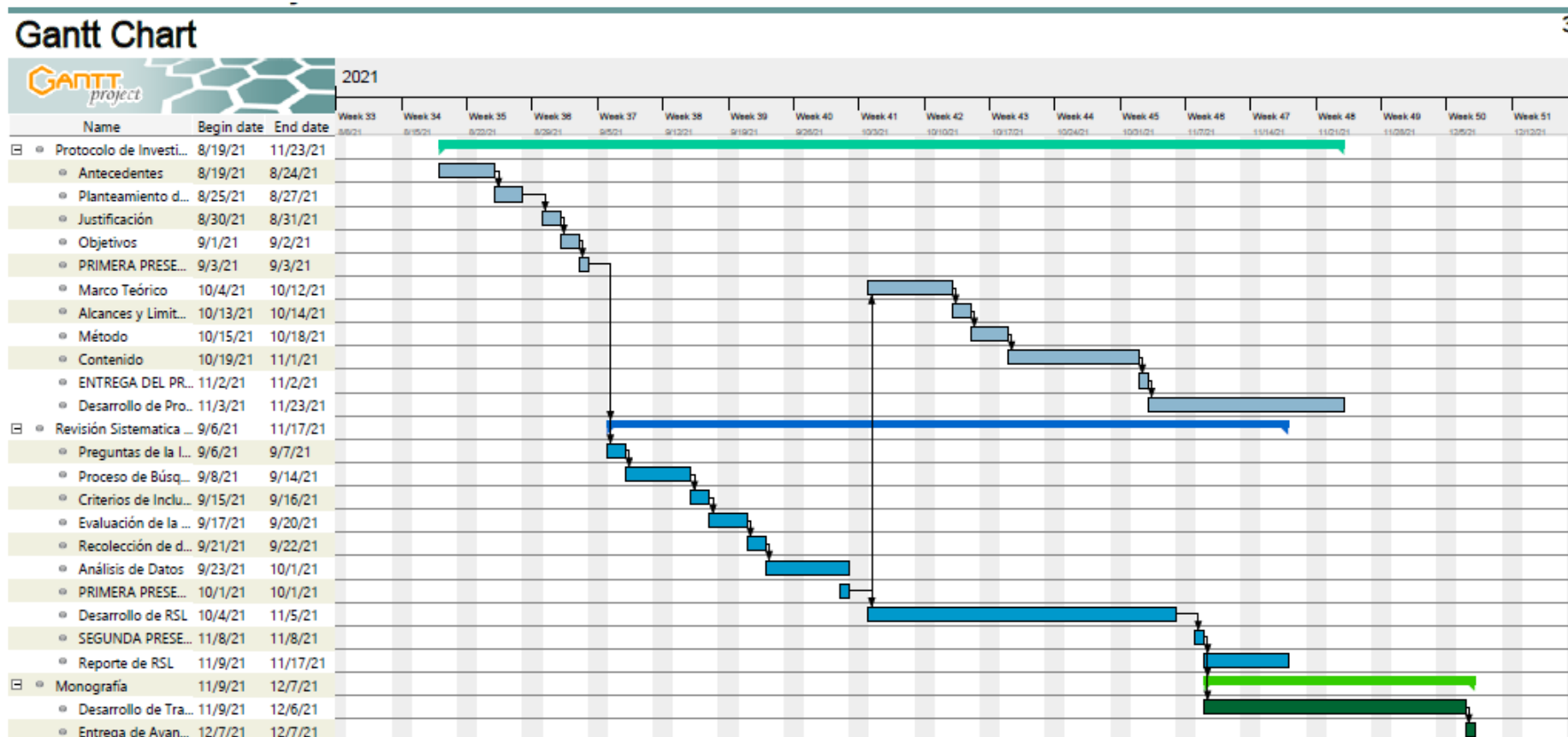


Ilustración 2 Cronograma del proyecto de investigación

## 10 Bibliografía

1. Bhaveet, N., & Tracy, H. (2015). How Software Developers Mitigate their Errors. IEEE Transactions on Software Engineering.
2. C. Lebeuf, A. Z. (2019). Defining and Classifying Software Bots:A Faceted Taxonomy. 1st International Workshop on Bots in Software Engineering (BotSE).
3. Carlene R., L. (2018). A Taxonomy of Software Bots:Towards a Deeper Understanding of Software Bot Characteristics. Victoria.
4. I. Qasse, S. M. (2021). iContractBot: A Chatbot for Smart Contracts' Specification and Code Generation. IEEE/ACM Third International Workshop on Bots in Software Engineering (BotSE), (págs. 35 - 38). doi:10.1109/BotSE52550.2021.00015.
5. Linda Erlenhov, F. G. (2019). Current and Future Bots in Software Development. 2019. Montreal: University of Gothenburg.
6. Mancl, S. F. (2008). No Silver Bullet: Software Engineering Reloaded. IEEE Software, 25, págs. 91-94. doi:10.1109/MS.2008.14
7. Margaret-Anne, S., & Zagalsky, A. (2016). Disrupting developer productivity one bot at a time. In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering.
8. McCarthy, J. (Noviembre de 2007). What is Artificial Intelligence?
9. Monperrus, M. (2019). Explainable Software Bot Contributions: Case Study of Automated Bug Fixes. IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE), (págs. 12-15). doi:10.1109/BotSE.2019.00010
10. Parnin, C. B. (2021). Nudging Students Toward Better Software Engineering Behaviors. IEEE/ACM Third International Workshop on Bots in Software Engineering (BotSE), (págs. 11-15). doi:10.1109/BotSE52550.2021.00010
11. Russell, S., & Norvig, P. (2010). Artificial Intelligence A Modern Approach (Third Edition). New Jersey: Pearson Education, Inc.



12. V. Alizadeh, M. A. (2019). RefBot: Intelligent Software Refactoring Bot. 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), (págs. 823-834). doi:10.1109/ASE.2019.00081.
13. V. S. Sharma, R. M. (2019). A Smart Advisor for Software Delivery - A Bot Framework for Awareness, Alerts and Advice. IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE), (págs. 22-23). doi:10.1109/BotSE.2019.00014.
14. V. Subramanian, N. R. (2019). TutorBot: Contextual Learning Guide for Software Engineers. IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE), (págs. 16-17). doi:10.1109/BotSE.2019.00011
15. Cruzes, D., & Dybå, T. (2011). Recommended Steps for Thematic Synthesis in Software Engineering. International Symposium on Empirical Software Engineering and Measurement, 275–284. <https://doi.org/10.1109/ESEM.2011.36>
16. Wohlin, C. (2014). Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. <https://doi.org/10.1145/2601248.2601268>
17. Zhang, H., Babar, M. A., & Tell, P. (2011). Identifying relevant studies in software engineering. Information and Software Technology, 53(6), 625–637. <https://doi.org/https://doi.org/10.1016/j.infsof.2010.12.010>
18. Briand, L. (2012). Embracing the engineering side of software engineering. IEEE Software, vol. 29, no. 4, pp. 96-96, July-Aug. 2012, doi: 10.1109/MS.2012.86.
19. M. Harman, "The role of Artificial Intelligence in Software Engineering," 2012 First
20. International Workshop on Realizing AI Synergies in Software Engineering (RAISE), 2012, pp. 1-6, doi: 10.1109/RAISE.2012.6227961.
21. Zemčík T. (2019). A brief history of chatbots. DEStech Transactions on Computer Science and Engineering,

22. Riddle, W. E., & Fairley, R. E. (2012). Software development tools. Springer Science & Business Media.
23. Krishnamoorthy, C. S., & Rajeev, S. (2018). Artificial Intelligence and Expert Systems for Artificial Intelligence Engineers. CRC press.
24. Robbins, J. E. (1999). Cognitive support features for software development tools. University of California, Irvine.
25. Prasad, C., & Schulte, W. (2013). Taking control of your engineering tools. Computer, 46(11), 63-66.
26. Rajlich, V. T., & Bennett, K. H. (2000). A staged model for the software life cycle. Computer, 33(7), 66-71.
27. Leach, R. J. (2018). Introduction to software engineering. CRC Press.