

Desarrollo dirigido por hipótesis en productos de software sostenible: Un mapeo sistemático

Paola A. Noreña¹[0000-0002-3509-7354], Elizabeth Suescún¹[0000-0001-7872-7638], José Mejía¹

¹Escuela de Ciencias Aplicadas e Ingenierías – Universidad EAFIT
Medellín – Colombia

{panoreaac, esuescu1, jmejia05}@eafit.edu.co

Resumen. El desarrollo dirigido por hipótesis (HDD) es un enfoque de desarrollo de software que se basa en la hipótesis sobre el impacto que tiene un nuevo producto, funcionalidad o cambio en los usuarios. La sostenibilidad del software permite la gestión de recursos en el desarrollo minimizando el impacto ambiental, técnico, económico, social e individual. Por lo que HDD promueve que los productos de software sean sostenibles ya que, permite ciclos iterativos cortos, productos mínimos viables, optimización de recursos y retroalimentación de los usuarios. En este artículo se presenta un mapeo sistemático de la literatura, identificando herramientas y niveles de madurez necesarios para adoptar, ejecutar y medir los resultados de aplicar este enfoque en una empresa o proyecto de software. Con este mapeo, se pretende mostrar los beneficios de este enfoque en el desarrollo de software, proporcionando orientación sobre su implementación sostenible.

Palabras clave. Desarrollo de software, Desarrollo dirigido por hipótesis, sostenibilidad del software.

Hypothesis-driven development in Sustainable software products: A Systematic mapping

Abstract. Hypothesis-driven development (HDD) is a software development approach based on hypotheses about the impact of a new product, functionality, or change on users. Software sustainability enables resource management in development by minimizing environmental, technical, economic, social, and individual impacts. Therefore, HDD promotes software products being sustainable since it allows for short iterative cycles, minimum viable products, resource optimization, and user feedback. This article presents a systematic literature review, identifying necessary tools and maturity levels to adopt, execute, and measure the outcomes of applying this approach in a company or software project. Through this review, the aim is to demonstrate the benefits of this approach in software development, providing guidance on its sustainable implementation.

Keywords: Hypothesis-driven development, Software development, Software sustainability.

1 Introducción

El desarrollo dirigido por hipótesis o HDD por sus siglas *Hypothesis-driven development* en inglés, tiene su origen el “método científico”, el cual surge como una respuesta a la necesidad de evitar largos y costosos periodos de desarrollo y a los cuestionamientos: ¿esto si será útil?, ¿tendrá esto un beneficio? en formato de hipótesis (suposiciones críticas). Así pues, el HDD busca maximizar el valor entregado al centrarse en la validación continua de las hipótesis fundamentales que sustentan las decisiones de un producto de software [1]. Adoptar el desarrollo dirigido por hipótesis es pensar en el desarrollo de nuevas ideas, productos y servicios (incluso cambios organizacionales) como una serie de experimentos para determinar si se logrará el resultado esperado [2]. La aplicación del método científico (proceso sistemático para realizar experimentos y buscar la validación de descubrimientos científicos), permite realizar pruebas y validaciones rigurosas de la hipótesis, con el objetivo final de producir conocimiento que haya pasado las múltiples fases de validación con resultados reproducibles. De esta manera, los productos basados en hipótesis producen conocimientos firmemente fundamentados, pero dentro de un ámbito definido que limita la velocidad en el que se puede generar conocimiento [3]. Por lo que en el desarrollo de software, HDD permite diseñar, desarrollar y probar nuevos productos, conceptos o servicios, basándose en una hipótesis sobre el impacto que una nueva funcionalidad o cambio tendría sobre los usuarios finales y así obtener retroalimentación de forma rápida, de tal manera que se puede probar y desarrollar de acuerdo con la validación realizada, anticipándose a situaciones no deseadas en ámbitos productivos. HDD se enfoca también en ciclos iterativos de trabajo cortos, para prevenir gastos de tiempo y recursos en desarrollos extensos que pueden fallar con el lema “*fail fast*” (falla rápida) [1], [4].

La sostenibilidad del software es crucial para la creación de productos que incorporen procesos sostenibles a lo largo del ciclo de vida del software, o que respalden la sostenibilidad en diversas áreas, según lo indicado por [5]. La sostenibilidad del software es el conjunto de prácticas y soluciones de sostenibilidad que surgen desde el software, en el que una de sus dimensiones es el software sostenible o *green software*, el cual se puede interpretar de dos maneras [5], (1) procesos de desarrollo, despliegue y mantenimiento de productos de software sostenible, agnóstico al propósito, o (2) el propósito del software es apoyar los objetivos de sostenibilidad desde diferente área disciplinares. El software sostenible, también involucra aspectos técnicos de infraestructura tecnológica como la arquitectura de software y sus componentes [6]. Por lo tanto, el software sostenible debe ser eficiente energéticamente, minimizar el impacto ambiental de los procesos que apoya en reducción de datos y optimización del software, y generando un impacto positivo en la sostenibilidad humana y/o económica. Estos impactos pueden ocurrir directos (energía), indirectos (mitigados por el servicio) [7].

En este sentido, el desarrollo dirigido por hipótesis surge como una práctica para abordar la sostenibilidad del software, haciendo uso de productos ágiles como el producto mínimo viable o MVP (*minimum viable product* por sus siglas en inglés), sprints, historias de usuario, pruebas A/B (evaluación para comparar dos versiones del producto), herramientas de medición y técnicas estadísticas, los cuales permiten evaluar las funcionalidades, determinando su éxito o fracaso de manera objetiva. Este enfoque no solo contribuye a la eficiencia del desarrollo, sino que también favorece la alineación de los productos de software sostenible mediante los ciclos iterativos cortos, la optimización de recursos y la retroalimentación de usuarios [8].

De acuerdo con lo anterior, se procede a realizar un mapeo sistemático de la literatura con el objetivo de obtener una comprensión detallada sobre cómo el desarrollo dirigido por hipótesis ejerce un impacto en la creación de productos de software sostenible. El propósito de esta revisión fue identificar las prácticas más recurrentes relacionadas con el enfoque HDD, investigar las herramientas preponderantemente empleadas en este contexto, y finalmente, analizar el nivel de madurez necesario tanto a nivel de proyecto como de equipo de desarrollo

de software para instaurar un proceso de desarrollo más sostenible mediante la adopción exitosa del HDD. Por lo que los resultados permiten orientar en la adopción de este enfoque contribuyendo a la identificación de mejores prácticas y experiencias previas, un enfoque más efectivo y sostenible en el desarrollo de software, a la mejora continua y la toma de decisiones informadas en el ámbito de la ingeniería de software.

Finalmente, este artículo se estructura de la siguiente manera: En la Sección 2 se incluyen los trabajos previos, en la Sección 3 se expone la metodología, en la Sección 4 se incluyen los resultados y discusión. Finalmente se presentan las conclusiones y trabajos futuros en la Sección 5.

2 Trabajos previos

Varios autores han abordado el concepto de “experimentación continua” (EC), el cual está estrechamente vinculado con HDD. Dan McKinley introdujo este concepto en 2012 en una presentación sobre cómo Etsy –una empresa estadounidense de comercio electrónico, empleó la experimentación para evaluar el retorno de la inversión de funciones potenciales en el área de mercadeo [9]. Los estudios secundarios, aquellos que han usado y revisado trabajos con el concepto de EC, han sido objeto de estudio desde 2017 [9], [10], [11], [12], [13,] [14], [15], [16], [17]. Sin embargo, a pesar de que compartir una base conceptual similar, ninguno de estos estudios utiliza el término de HDD.

Entre los elementos comunes en los estudios de EC se encuentran la identificación y priorización de hipótesis, la experimentación, las pruebas A/B, la toma de decisiones y la implementación del MVP. Además, se destaca su relación con prácticas sostenibles como automatización de pruebas, integración continua, despliegue continuo y entrega continua [9], por lo que también induce a un interés frente a la utilización de este tipo de enfoques en software sostenible.

También se observa que desde la literatura gris y técnica reciente es más utilizado el término de HDD [2].

3 Metodología

Un mapeo sistémico de la literatura permite identificar, analizar e interpretar las evidencias disponibles relacionadas con una serie de preguntas de investigación sobre un tema en específico. Este mapeo se realizó siguiendo los lineamientos para estudios de mapeo sistemático en Ingeniería de Software [18], [19] para presentar los estudios relevantes en desarrollo dirigido por hipótesis en productos de software sostenible, en tres etapas: (i) creación de las preguntas de investigación; (ii) Definición de cadenas de búsqueda y (iii) esquemas de clasificación.

2.1 Preguntas de investigación

Para lograr el objetivo del mapeo sistémico, identificar las brechas y proponer nuevas oportunidades de investigación, se definen unas preguntas de investigación las cuales guiarán el estudio (véase Tabla 1). Con estas preguntas se trata de obtener una definición clara sobre HDD desde sus fundamentos más básicos, como lo son su origen y qué es, hasta como se aplica y qué herramientas se utilizan para adoptarlo. Esto con el fin de poder brindar recomendaciones al final de esta investigación, sobre posibles áreas adicionales de investigación con enfoques más profundos.

Tabla 1. Preguntas de investigación y motivación

Preguntas de investigación	Motivación
Q1. ¿Cuál es el origen de HDD y qué es?	Bases, fundamentos, de donde viene la necesidad de aplicar esta práctica.
Q2. ¿Qué frameworks (marco de trabajo) / metodología / estrategia / herramientas existen para llevar a cabo HDD?	Mejores prácticas, estrategias, herramientas y metodologías que se necesita para aplicar HDD.
Q3. ¿Qué se necesita como punto de inicio para comenzar a experimentar?	Criterios que se deben reunir para adoptar esta práctica.
Q4. ¿Qué nivel de madurez debe tener un proyecto o desarrollo de software para adoptar HDD?	Conocer en que etapa de un proyecto o desarrollo de software se puede aplicar.
Q5. ¿Qué prácticas o productos de trabajo de HDD permiten sostenibilidad en el software?	Conocer elementos de productos de software sostenible mediante HDD.

2.2 Estrategia de búsqueda

Para realizar la búsqueda de información relevante, se utilizaron motores de búsqueda académicos como Google Scholar, ACM Digital Library, IEEE Digital Library, Research Gate y Scopus, donde la utilización de conectores lógicos como, “AND” y “OR” se utilizó para construir cadenas de búsqueda con las palabras claves que se deseaban encontrar en los textos (véase Tabla 2), adicionalmente estas cadenas de búsqueda tuvieron que ser adaptadas para cada biblioteca digital con el fin de identificar de la mejor forma los trabajos disponibles.

Esta búsqueda se centró exclusivamente en obtener resultados en inglés, ya que se determinó que la información disponible en este idioma era más extensa y aportaba a la comprensión de la investigación abordada. Adicionalmente se agregó un parámetro de tiempo, el cuál limitó la búsqueda de información a un rango de 2018 a enero 2024. Se determinó este rango de años puesto que los resultados que se encontraban previos al 2018 eran escasos para el concepto “*hypothesis-driven development*”.

Tabla 2. Cadenas de búsqueda para cada base de datos

Base de datos	Cadena de búsqueda
Google Scholar	allintitle: Hypothesis-driven development "software" OR Hypothesis- driven OR Hypothesis driven" AND Sustainability OR Software sustainable OR Software sustainability
ACM Digital Library	(Hypothesis-driven development "software" OR Hypothesis-driven development OR Hypothesis- driven OR Hypothesis driven) AND (tools OR method OR methodology OR metrics OR strategy) AND (Sustainability OR Software sustainable OR Software sustainability)
IEEE Digital Library	("Hypothesis-driven development" "software" OR "Hypothesis-driven development" OR "Hypothesis- driven" OR "Hypothesis driven") AND ("tools" OR "method OR "methodology" OR "Metrics" OR "Strategy" "testing") AND ("Sustainability" OR "Software sustainable" OR "Software sustainability")
Research Gate	("Hypothesis-driven development" "software" OR "Hypothesis-driven development" OR "Hypothesis- driven" OR "Hypothesis driven") AND ("tools" OR "method OR "methodology" OR "Metrics" OR "Strategy" "testing") AND ("Sustainability" OR "Software sustainable" OR "Software sustainability")
Scopus	("Hypothesis-driven development" "software" OR "Hypothesis-driven development" OR "Hypothesis- driven" OR "Hypothesis driven") AND ("tools" OR "method OR "methodology" OR "Metrics" OR "Strategy" "testing") AND ("Sustainability" OR "Software sustainable" OR "Software sustainability")

Criterios de inclusión y exclusión. Posterior a recolectar la información en cada una de las bases de datos, se realizó una evaluación de los resultados teniendo en cuenta una serie de criterios de inclusión (véase Tabla 3) y exclusión (véase Tabla 4) como estrategia válida de búsqueda.

Tabla 3. Criterios de inclusión

Criterio	Descripción
IC1	Papers y capítulos de libros que sus títulos sean relacionados con HDD
IC2	Papers y capítulos de libros publicados entre 2018 – 2024
IC3	Papers y capítulos de libros disponibles en formato electrónico
IC4	Papers y capítulos de libros en inglés.

Tabla 4. Criterios de exclusión

Criterio	Descripción
EC1	Papers que su metodología o técnicas no sean aplicables a desarrollo de software
EC2	Papers que sean con fines educativos
EC3	Papers en forma de eventos, trabajos sin publicar
EC4	Textos en otros lenguajes diferentes a inglés fueron excluidos

Criterios de evaluación de calidad. Para medir la calidad de los estudios seleccionados se utilizó un cuestionario con un sistema de calificación con los posibles valores entre 0.00 (no cumplimiento) y 0.83 (máximo cumplimiento) para cada criterio, estos criterios son presentados en la Tabla 5. La sumatoria de puntos para cada estudio da como resultado valores entre 0.00 y 5.00, donde 0.00 es un estudio que no presenta relevancia pertinente en su contenido para el tema y 5.00 es un estudio que presenta mucha relevancia al cumplir con todos los criterios, a partir de un valor de 3.00 tiene relevancia. Dicha información se presenta en la Tabla 6, cuyo contenido presenta los resultados de la evaluación a partir de estos criterios y las preguntas de investigación de la Tabla 1.

Tabla 5. Criterios de evaluación de calidad

Criterio	Descripción
C1	¿El estudio tiene un enfoque primario en HDD?
C2	¿El estudio presenta técnicas para aplicar HDD?
C3	¿El estudio muestra áreas de aplicación de HDD?
C4	¿El estudio tiene como objetivo general HDD?
C5	¿El estudio expone pre-requisitos para la aplicación de HDD?
C6	¿El estudio se relaciona con HDD y aspectos de sostenibilidad?

2.3 Esquema de clasificación

Se llevó a cabo dos iteraciones. La primera iteración consistió en una revisión de la literatura gris. Por medio de esta revisión se identificaron palabras claves y términos asociados a HDD los cuales ayudaron a enfocar y refinar las cadenas de búsqueda finales. Posteriormente, en la

segunda iteración se analizaron y seleccionaron los estudios encontrados a través de las cadenas de búsqueda refinadas. En la Tabla 7 se muestra la cantidad de artículos analizados y seleccionados en las diferentes bases de datos.

Tabla 6. Estudios(E) y preguntas(Q) que responden, y criterios (C) que cumplen

Estudio	Q1	Q2	Q3	Q4	Q5	C1	C2	C3	C4	C5	C6	Puntaje
E1. [Schnappinger et al. 2018]	-	X	X	-	X	0.83	0.41	0.41	0.83	0	0.83	3.31
E2. [Reddy and Kothapalli 2019]	X	X	X	X	X	0.83	0.83	0.41	0.83	0	0.41	3.32
E3. [Sporse et al. 2021]	X	X	X	X	X	0.83	0.83	0.41	0.83	0.83	0.83	3.73
E4. [Melegati et al. 2020]	X	X	X	X	X	0.83	0.83	0.83	0.83	0.83	0.83	5.00
E5. [Alhammad and Moreno 2022]	-	X	X	-	-	0.41	0.83	0.83	0.41	0.41	0	3.33
E6. [Ferreira et al. 2021]	-	X	X	-	-	0.41	0.83	0.83	0.41	0.41	0	3.33
E7. [Khanna et al. 2018]	X	X	X	X	X	0.83	0.83	0.83	0.83	0.83	0.41	4.56
E8. [Kmieciak and Michna 2018]	-	X	X	-	X	0.41	0.83	0.83	0.41	0.41	0.41	3.33
E9. [Schnappinger et al. 2018]	X	X	X	X	X	0.83	0.83	0.83	0.41	0.83	0.83	4.56
E10. [Mollick 2020]	-	X	-	-	-	0.41	0.83	0.83	0.41	0.41	0	3.50
E11. [Silva et al. 2021]	-	X	X	X	X	0.41	0.83	0.83	0.41	0.83	0.83	4.14
E12. [Gottschalk et al. 2020]	-	X	X	X	X	0.83	0.83	0.83	0.83	0.83	0.41	4.56
E13. [Cowan 2023a]	-	X	X	X	-	0.83	0.83	0.83	0.83	0.83	0	4.15
E14. [Jololian and Bowman 2022]	X	X	X	-	X	0.83	0.83	0.83	0.83	0.83	0.41	4.56
E15. [Cowan 2023b]	-	X	X	X	-	0.83	0.83	0.83	0.83	0.83	0	4.15
E16. [Vargas et al. 2020]	-	X	X	-	X	0.83	0.83	0.41	0.41	0.41	0.41	3.33
E17. [Melegati et al. 2021]	X	X	X	X	X	0.83	0.83	0.83	0.83	0.83	0.41	4.56
E18. [Bowman 2022]	-	X	X	X	X	0.83	0.83	0.83	0.41	0.41	0.83	4.14
E19. [Klepper and Bruegge 2018]	X	X	X	-	X	0.83	0.83	0.83	0.83	0.83	0.83	5.00
E20. [Fleweling 2018]	-	X	X	-	X	0.41	0.83	0.83	0	0.83	0.41	3.31
E21. [Satyal 2019]	-	X	X	-	X	0.41	0.83	0.83	0	0.83	0.41	3.31
E22. [Peralta et al. 2020]	-	X	X	-	X	0.41	0.83	0.83	0	0.83	0.41	3.31

Tabla 7. Selección de estudios

Base de datos	Estudios Encontrados	Estudios Relevantes	Repetidos Relevantes	Selección final
Google Scholar	32	9	6	3
ACM Digital Library	61	5	0	5
IEEE Digital Library	102	10	0	10
ResearchGate	20	1	0	1
Scopus	31	3	0	3
Total	246	28	6	22

La información recolectada de los estudios obtenidos, se clasificaron de acuerdo con las preguntas realizadas al inicio de la investigación en la Tabla 1.

3 Resultados y Discusión(Q1) ¿Cuál es el origen de HDD y qué es?

3.1 (Q1) ¿Cuál es el origen de HDD y qué es?

El 36,6 % de estudios seleccionados responden a esta pregunta (véase la Tabla 6), indicando que HDD tiene su origen en áreas científicas, donde se utiliza software especializado para acompañar procesos de investigación, desarrollo y medición de resultados. En este sentido, desde estas mismas áreas se han fundamentado en el método científico, por lo que se buscó implementar la experimentación en el desarrollo de software, con el fin de validar hipótesis antes de comprometerse con la implementación de una funcionalidad o de todo el producto de software en el desarrollo.

Consecuentemente, el origen de HDD en el proceso de desarrollo de software se acuña a la metodología *Lean startup*, un enfoque científico propuesto por Eric Ries y refinado en colaboración con Thomas Eisenmann, para formular y evaluar un MVP con base en una hipótesis evaluada por los usuarios, cuya medición de resultados y aprendizajes permite la adaptación del producto (E19). Uno de los principios fundamentales de esta metodología, es minimizar el desperdicio en los recursos y en el proceso de desarrollo del producto de software, por lo que se busca maximizar el aprendizaje validado a través de la experimentación temprana e iterativa, minimizando el desperdicio de recursos [38] y costes de las nuevas funcionalidades del software (E7).

3.2 (Q2) ¿Qué frameworks/metodología/estrategia/herramientas existen para llevar a cabo HDD?

El 100 % de los estudios seleccionados responden a esta pregunta (véase la Tabla 6), ya que HDD tiene relación cercana con herramientas ágiles en su uso e implementación. Los principios más recurrentes se centran en el uso de MVP, desarrollos iterativos cortos, retroalimentación rápida y si se presentan fallas, qué se puedan solucionar a tiempo sin implicar costos altos en los cambios [3]. Por lo que HDD, se ve influenciado por metodologías como Lean UX (un enfoque de diseño en experiencia usuario), *Design Thinking* (un enfoque centrado en el usuario para idear soluciones innovadoras) y entrega continua (una práctica de DevOps para la automatización del desarrollo [39]), buscando que las nuevas versiones del producto de software sean más rápidas y sostenible mediante la minimización de desperdicios y un enfoque profundo hacia el usuario donde a través del aprendizaje y comprensión del problema se genera un mejor desarrollo y validación del producto.

Adicionalmente, se encuentran marcos de trabajo que presentan las entradas, el proceso y las salidas al adoptar HDD en desarrollo de software. Así pues, las entradas pueden ser problemas, preguntas, elementos previos de documentación, código fuente, configuraciones anteriores, costos, ambiente y elementos de calidad. En cuanto al proceso, los estudios coinciden en una ideación o formulación de la hipótesis después de formular el problema, selección de actividades para desarrollar un MVP y una serie de pruebas que permitan validar y retroalimentar el producto de software con acciones recomendadas para generar nuevas iteraciones con mejora y entregas continuas al producto. Por último, las salidas están directamente relacionadas con la retroalimentación del usuario, por lo que se encuentra el análisis de resultados y las acciones recomendadas para generar la iteración siguiente de acuerdo con los E9 y E18. Además, la literatura gris y técnica también apoya este proceso [2], [40]. Consecuentemente, en la Figura 1 se sintetizan las principales fases para desarrollar software utilizando HDD y se describen a continuación:

(1) Identificar el problema: mediante la interacción con los usuarios se busca investigar y comprender las necesidades y oportunidades que dan inicio al proceso mediante preguntas y

respuestas que ayuden a su comprensión. **(2) Idear la hipótesis:** el equipo formula una hipótesis mediante ideas que permitan solucionar los problemas identificados mediante preguntas que dirijan la hipótesis, se documentan las HU y además se diseñan los experimentos mediante la selección de funcionalidades, prototipos y pruebas que serán utilizadas en la validación. Usualmente se establece una hipótesis general con respecto al experimento principal (podría ser todo el producto de software o un módulo dentro del software) y luego se precisan las hipótesis específicas con base en los MVPs específicos (posibles funcionalidades con métricas para medir el resultado). **(3) Desarrollar un MVP:** se proponen, diseñan e implementan los MVP. **(4) Validar la hipótesis:** se prueba el MVP exponiendo al resultado a los usuarios o en un entorno de pruebas para obtener retroalimentación que ayude a dar respuesta a la hipótesis y se analizan los resultados para llegar a una conclusión sobre la efectividad del MVP y dar respuesta a la hipótesis.

En cuanto a los marcos de trabajo en lenguajes de programación, no hay uno específicamente asociado con HDD. En el ámbito de las herramientas de desarrollo de software, existe una amplia variedad de *frameworks* disponibles que se pueden utilizar sin estar limitados a un lenguaje de programación en particular. Por lo tanto, quienes opten por utilizar HDD tienen la libertad de seleccionar el *framework* que mejor se adapte a sus necesidades y preferencias.

Por otro lado, para la fase de validación hay un método de evaluación predominante en el uso de HDD: las pruebas A/B (E5, E8, E21), donde se pone a prueba la implementación del MVP para evaluar y comparar dos versiones diferentes del producto (A y B). El objetivo principal es determinar cuál de las dos versiones es más efectiva en términos de cumplir con un objetivo específico y de esta forma determinar la mejor opción para seguir desarrollando el producto en las siguientes iteraciones. La razón principal del uso de este tipo de pruebas es que permite hacer seguimiento de los resultados y aplicar el conocimiento o aprendizaje para mejorar el producto frente a la experiencia del usuario [41].

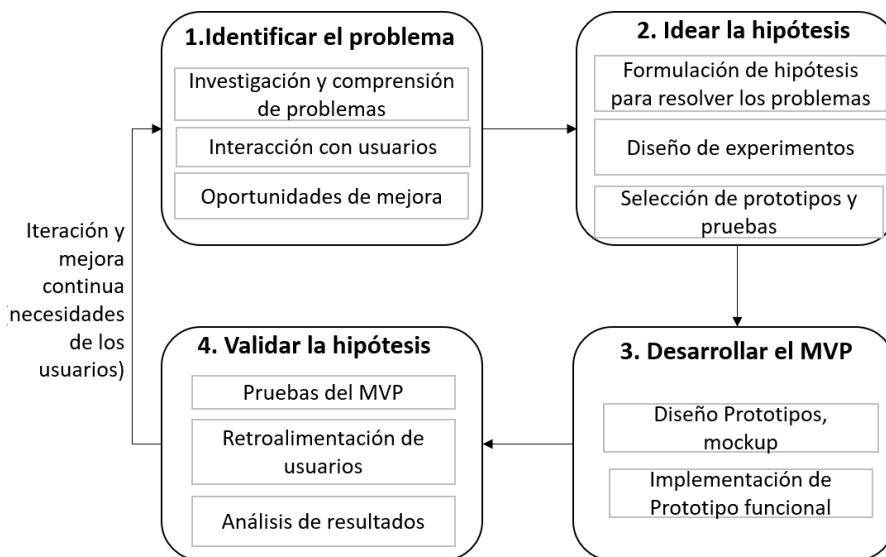


Figura 1. Fases de HDD. Los autores

3.3 (Q3) ¿Qué se necesita como punto de inicio para comenzar a experimentar?

El 95,4 % de los estudios coinciden en el punto de inicio para trabajar con HDD al presentar la “hipótesis” como el elemento diferenciador que se quiere probar (véase la Tabla 6), con el objetivo de tomar decisiones tempranas frente a lo que se debe realmente desarrollar, de este modo en ámbitos productivos se pueden anticipar a situaciones no deseadas (E8). Cuando se han analizado los problemas, se plantean las hipótesis que pueden solucionar el problema mediante un MVP y con base en los resultados, validar la mejor opción. La estructura para definir la hipótesis como HU es la siguiente: “*Si [hacemos algo por una determinada persona] entonces [responderán de alguna manera medible]* según [40]. Otra estructura posible es: *creemos que [esta capacidad] dará como resultado [esta salida] sabremos que hemos tenido éxito cuando [veamos una señal medible]*” según [2]. Ambas estructuras son semejantes en cuanto a la capacidad y la medida.

Para comprender mejor el punto de inicio en HDD, se presenta el siguiente ejemplo: se pretende mejorar una aplicación de redes sociales para noticias, en donde su tasa de retención de usuarios es muy baja. Siendo la pregunta inicial: ¿Cómo podemos mejorar la retención de usuarios en nuestra aplicación de redes sociales? y la hipótesis general: *Si personalizamos la alimentación de noticias de cada usuario en función de sus intereses y actividad previa, entonces la retención de usuarios aumentará, ya que recibirán contenido más relevante y atractivo.* El MVP general podría ser la *personalización de la alimentación de noticias* como un nuevo módulo de la aplicación. Además, se proponen unas posibles preguntas, hipótesis y experimentos para cada problema que se intenta solucionar en la hipótesis general (véase la Tabla 9).

Tabla 9. Ejemplo de preguntas, hipótesis y experimentos en un producto de software. Los autores

Pregunta	Hipótesis (HU)	Problema solucionado	Experimento (MVP)
¿Seleccionar categorías generales de interés durante la creación del perfil aumentará la participación y la interacción de los usuarios?	Si permitimos a los usuarios elegir categorías generales de interés, entonces la participación y la interacción de los usuarios aumentarán, ya que estarán más propensos a comprometerse con contenido que se alinee con sus preferencias.	Segmentación básica de intereses	Permitir a los usuarios seleccionar categorías generales de interés (por ejemplo, deportes, tecnología, entretenimiento) durante la creación de su perfil.
¿Registrar las interacciones de los usuarios, como clics, likes o compartidos, proporcionará datos útiles para personalizar la alimentación de noticias?	Si registramos las interacciones de los usuarios, entonces obtendremos datos valiosos para comprender sus preferencias y adaptar la alimentación de noticias de manera más efectiva.	Historial de actividad	Registrar las interacciones de los usuarios, como clics, likes o compartidos, en las noticias.
¿El uso de un algoritmo básico para personalizar la alimentación de noticias resultará en una experiencia de usuario más atractiva y relevante?	Si mostramos las categorías y noticias relevantes, entonces los usuarios experimentarán una mayor atracción y retención, ya que recibirán contenido más relevante y adaptado a sus intereses.	Alimentación de noticias personalizada	Utilizar la información de categorías de interés y actividades previas para mostrar noticias relevantes en una sección personalizada.

3.4 (Q4) ¿Qué nivel de madurez debe tener una empresa/proyecto para adoptar HDD?

El 50 % de los estudios seleccionados responden a esta pregunta (véase la Tabla 6). Una empresa o proyecto no requiere alcanzar un alto nivel de madurez o tener una larga trayectoria en el mercado con un producto. Por el contrario, metodologías como *Lean startup* están diseñadas con el objetivo de llevar a cabo HDD en *startups*, nuevos negocios que aún no tienen definido lo que quieren o no saben cuál es la mejor alternativa para su producto (E8 y E11). Incluso algunos estudios refieren la utilización de HDD/*Lean startup* en empresas grandes y de alto desempeño (aquellas que suelen tener una gestión eficiente y una cultura sólida con innovación y productividad), como una forma de innovar al interior de estas empresas, creando pequeños *startups* internos que ayuden a llevar a una empresa consolidada a un nuevo nivel y seguir innovando (E4 y E22).

3.5 (Q5) ¿Qué prácticas, características o productos de trabajo de HDD permiten sostenibilidad en el software?

El 77,2 % de estudios seleccionados responde a esta pregunta (véase la Tabla 6). En estos estudios se encuentran los siguientes productos de trabajo contribuyendo a la sostenibilidad del software: las HUs, los MVPs y sus validaciones (E7 y E8), los ciclos iterativos cortos “*fail fast*”, la retroalimentación de los usuarios (E4 y E17) la optimización de recursos desde la eficiencia, el reuso de experimentos para diferentes hipótesis (E12), y la evaluación de resultados (E14, E18, E19 y E21). En el E2, la medición del desempeño, la innovación [3], [28] y valor, las variables ambientales y la eficiencia en el desarrollo del producto de software son indispensables en la sostenibilidad del software.

Además, los autores afirman que la sostenibilidad para la arquitectura de software se fundamenta en el nivel de comprensión de la funcionalidad del sistema. Adicional a este aspecto, mencionan algunas características sostenibles respecto de la estructura de la arquitectura y el código: adaptabilidad (E20), flexibilidad, mantenibilidad, escalabilidad, extensibilidad, (E11), componentes reusables, consumo de energía y calidad del software para reducir riesgos sociales, ambientales, técnicos y económicos (E9, E18 y E19). Finalmente, una práctica sobresaliente es el desarrollo mediante HDD, ya que el enfoque permite el uso de las características mencionadas para que el ciclo de vida del software y el producto sean más sostenibles.

3.6 Discusión

El estudio sistemático provee luces del surgimiento de HDD, además es el primer estudio sistemático para este enfoque, aunque ya se encontraban estudios para técnicas de historias de usuario sin la inclusión de HDD. Basado en la estrategia de búsqueda y las preguntas de investigación se observan oportunidades de investigación en este enfoque para que sea profundizado frente a las etapas del ciclo de vida del software, las herramientas a utilizar y la sostenibilidad del software mediante HDD de acuerdo con los 22 estudios seleccionados entre los años 2018-2024 y las preguntas Q1 a Q5.

Si bien el agilismo puede tener alrededor de 24 años desde el Manifiesto Ágil, HDD es un enfoque relativamente nuevo, ya que sólo a partir del 2014 se encuentran textos referenciando el concepto de “desarrollo dirigido por hipótesis” en procesos de software, tratando de cerrar brechas en la flexibilidad de requisitos aún en el uso de marcos ágiles, cuando los requisitos son complejos con alta incertidumbre e incluso un desconocimiento de estos requisitos (Q1). “Como Lean y *Lean Startup* acelera el ritmo de innovación, apoyándose en iteraciones rápidas, en pequeñas cantidades y ciclos de tiempo cortos” [29] fundamentan las características de HDD, dando como resultado una metodología que permita las equivocaciones o fallas rápidas y económicas, eliminando el temor a la incertidumbre y facilitando la toma de decisiones, donde los resultados en cada iteración puedan proporcionar valor inmediato y sostenible (Q2). La

hipótesis resulta siendo el elemento diferenciador de HDD en los diferentes estudios, con el propósito de entender el problema y proponer un MVP que atienda a la solución y obteniendo validaciones tempranas que ayudan en esta toma de decisiones (Q3). Estas características de fallas rápidas, ciclos cortos, MVP, validaciones y retroalimentación temprana, así como la innovación y el valor en el producto a través de la optimización de recursos, eficiencia, reúso de experimentos, medición del desempeño, variables ambientales y características de calidad (adaptabilidad, flexibilidad, mantenibilidad, escalabilidad, extensibilidad, componentes reusables y consumo de energía), permite reducir riesgos sociales, ambientales, técnicos y económicos del software, promueven a HDD como una buena práctica que contribuye a la creación de productos de desarrollo de software más sostenibles según la gran mayoría de estudios (Q5).

Finalmente, la flexibilidad que ofrece HDD para el desarrollo de software y la innovación al integrarse con *Lean Startup*, puede aportar en términos de evolución del software para hacer los productos de software más escalables, resilientes y sostenibles. Por lo que, el gran potencial de HDD está en seguir utilizándolo en ambientes productivos para mejorar las aplicaciones en términos de escalabilidad, y fallas que puedan surgir independientemente del nivel de madurez de la empresa (Q4).

El mapeo sistemático de la literatura se centró en la recopilación de información sobre HDD. El estudio se planificó y realizó siguiendo el protocolo propuesto en [18] y [19]. En este contexto, y dada la naturaleza del estudio, es importante relacionar las amenazas a la validez y las limitaciones de este mapeo. (i) Las cadenas de búsqueda propuestas para el proceso pueden incluir términos de búsqueda inadecuados relacionados con el tema de investigación, (ii) sólo se aplicó la búsqueda automática, y (iii) Fueron considerados principalmente artículos en el rango de 2018 a enero 2024 y que incluían información sobre HDD y/o experimentación continua, esto porque era de interés su adopción como práctica sostenible de DevOps. Todo ello puede dar lugar a sesgos en la identificación de los estudios primarios [42]. Se identificó que el concepto de desarrollo basado en hipótesis tuvo alta adopción con DevOps. Esto no garantiza que se hayan seleccionado todos los estudios primarios relacionados ni que toda la información relacionada con la adopción de esta práctica en desarrollo de software. Es posible que se hayan descartado artículos que podrían contribuir a estos temas. Estas amenazas a la validez pueden afectar a la generalización de los resultados del mapeo. Sin embargo, para minimizar estas amenazas y evitar sesgos en la extracción de datos todo el proceso se llevó a cabo mediante comprobaciones cruzadas entre los autores.

Si bien el presente trabajo se enfoca en mostrar la importancia al utilizar un enfoque de desarrollo de software basado en hipótesis para que el proceso de software y los productos derivados de ello sean más sostenibles y se enunciaron principalmente las ventajas a lo largo del trabajo, también es importante enunciar las desventajas y limitaciones al abordar este enfoque. La mayoría de ellas descritas en [2] y [43], así: la incertidumbre, ya que las hipótesis son suposiciones no probadas que pueden llevar a decisiones erróneas si no se validan adecuadamente y en ocasiones esto puede emerger en el resultado final del proceso o del producto.

Por otro lado, el riesgo de fracaso cuando las hipótesis en las que se basa el desarrollo resultan ser incorrectas, el software puede no cumplir con los requisitos del cliente o presentar problemas significativos, lo que puede llevar al fracaso del proyecto. Adicional a eso, pueden existir casos donde se requiera mayor tiempo y recursos, si es comparado con un enfoque tradicional [45], en ese sentido, validar la hipótesis podría requerir más tiempo y recursos adicionales. Si las hipótesis cambian o se descartan, esto podría llevar a retrabajo, aumentando los costos y retrasando el cronograma del proyecto.

Otro tema por considerar es la limitación de datos, puede ser complejo obtener datos significativos y/o relevantes para validar las hipótesis, lo que puede limitar la eficacia de este enfoque. Otro aspecto es que algunas hipótesis pueden cambiar a medida que se desarrolla el software [47], lo que dificulta la toma de decisiones sólidas y la planificación a largo plazo. Finalmente, en el caso que las hipótesis iniciales resulten ser incorrectas, puede ser difícil adaptarse rápidamente y cambiar de dirección en el desarrollo del software, lo que limita la flexibilidad del equipo de desarrollo [48]. Estos aspectos son importantes considerarlos, gestionarlos ya que es importante estar preparado para abordar los desafíos asociados con un enfoque basado en hipótesis en el desarrollo de software y tomar medidas para mitigar los riesgos potenciales [44].

4 Conclusiones y Trabajo futuro

En este artículo se presentó un mapeo sistémico de la literatura, permitiendo una mayor comprensión sobre el estado actual del desarrollo dirigido por hipótesis, en cuanto a su fundamentación, uso y características para llevar a cabo productos de software más sostenibles en ambientes productivos. Así mismo el mapeo ayudó a identificar las preguntas y sus hipótesis como elementos de entradas principales al proceso de desarrollo con HDD.

HDD se beneficia de enfoques como Lean UX, *Design Thinking* y entrega continua de DevOps, con el objetivo de mejorar la velocidad y sostenibilidad de las nuevas versiones del software. Esto se logra minimizando desperdicios, adoptando un enfoque centrado en el usuario y utilizando el aprendizaje para desarrollar y validar el producto de manera más efectiva y generando valor en cualquier nivel de madurez de una empresa. Adicionalmente, este enfoque ayuda a minimizar el riesgo de desarrollar características que no son realmente valiosas para los usuarios o que no generan el impacto esperado. También fomenta la adaptabilidad y la mejora continua, ya que se basa en la retroalimentación constante del mercado y los usuarios permitiendo soluciones a problemas con MVPs esperados.

Ambientes productivos se pueden beneficiar de HDD en la creación productos de software sostenibles, al integrar HU con métricas y MVPs más rápidos y pertinentes, validaciones, ciclos iterativos cortos y retroalimentación temprana de usuarios. Además, HDD aporta a la sostenibilidad del software mediante la optimización de recursos, el reúso de experimentos con productos que contengan innovación y valor, medición del desempeño y de variables ambientales. Algunas de las características que pueden ser medidas en este tipo de productos son: funcionalidad, adaptabilidad, flexibilidad, mantenibilidad, escalabilidad, extensibilidad, componentes reusables y eficiencia energética en el producto desarrollado, el ciclo de vida del software y su infraestructura.

Como trabajos futuros, se considera importante realizar estudios sobre herramientas para realizar pruebas desde HDD y frente a la arquitectura con HDD. Adicionalmente se pueden encontrar otras prácticas, que puedan ser utilizadas junto con el desarrollo dirigido por hipótesis para lograr productos de software más sostenibles.

Agradecimientos. Este artículo surge como resultado del proyecto “Modelo de sostenibilidad humana en productos de desarrollo de software con infraestructura tecnológica limitada” en el programa de Mujeres en la ciencia 2023 patrocinado por Minciencias, Icetex y L’oreal-Unesco y la institución ejecutora, Universidad EAFIT.

Referencias

1. Klepper, S., Bruegge, B.: Impact of Hypothesis-Driven Development on Effectiveness, Quality, and Efficiency in Innovation Projects, Lecture Notes in Informatics (2018)

2. O'Reilly, B.: How to Implement Hypothesis-Driven Development, <https://www.thoughtworks.com/insights/articles/how-implement-hypothesis-driven-development>, (2014)
3. Bowman, A. D.: A Hybrid Data and Hypothesis-Driven Model for Software Development in Support of the Machine-Learning Paradigm, Doctoral dissertation, Alabama: The University of Alabama at Birmingham (2022)
4. Khanna, D., Nguyen-Duc, A., Wang, X.: From MVPs to pivots: a hypothesis-driven journey of two software startups, *Software Business: 9th International Conference, ICSOB 2018*, 172–186, Springer, Estonia (2018)
5. Calero, C., Moraga, M., Piattini, M.: Introduction to Software Sustainability. *Software Sustainability*. Springer, Switzerland (2021)
6. Restrepo, L.: A sustainable-development approach for self-adaptive cyber-physical systems life cycle: A systematic mapping study, *The Journal of Systems and Software*, **180**, 1–17 (2021)
7. Tardini, D.: Agilidad y desarrollo sostenible: Agilidad, sostenibilidad y tecnología en continua armonía. Publicado independientemente (2022)
8. Kmiecik, R., Michna, A.: Knowledge management orientation, innovativeness, and competitive intensity: evidence from Polish SMEs, *Knowledge Management Research and Practice*, **16**(4), 559–572 (2018)
9. Erthal, V. M., de Souza, B. P., dos Santos, P. S. M., Travassos, G. H.: Characterization of continuous experimentation in software engineering: Expressions, models, and strategies, *Science of Computer Programming*, **229**, 102961 (2023)
10. Giaimo, F., Andrade, H., Berger, C.: Continuous experimentation and the cyber-physical systems challenge: An overview of the literature and the industrial perspective, *Journal of Systems and Software*, **170**, 110781 (2020)
11. Fagerholm, F., Guinea, A. S., Mäenpää, H., Münch, J.: The RIGHT model for continuous experimentation. *Journal of Systems and Software*, **123**, 292–305 (2017)
12. Auer, F., Ros, R., Kaltenbrunner, L., Runeson, P., Felderer, M.: Controlled experimentation in continuous experimentation: Knowledge and challenges. *Information and Software Technology*, **134**, 106551 (2021)
13. Auer F., Felderer, M.: Current state of research on continuous experimentation: a systematic mapping study, 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 335–344, IEEE, Prague (2018)
14. Fabijan, A., Dmitriev, P., Olsson, H. H., Bosch, J.: The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale” 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), 770–780, IEEE (2017)
15. Fabijan, A., Dmitriev, P., Olsson, H. H., Bosch, J.: Online controlled experimentation at scale: an empirical survey on the current state of A/B testing, 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 68–72, IEEE, Prague (2018)
16. Ros, R., Runeson, P.: Continuous experimentation and a/b testing: a mapping study, IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE), 35–41, IEEE, New York (2018)
17. Mattos, D.I., Bosch, J., Olsson, H.H.: Challenges and strategies for undertaking continuous experimentation to embedded systems: industry and research perspectives, *International Conference on Agile Software Development*, 277–292, Portugal (2018)
18. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A.: *Experimentation in software engineering*, Springer Science and Business Media, New York (2012)
19. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update, *Information and software technology*, **64**, 1–18 (2015)
20. Schnappinger, M., Osman, M. H., Pretschner, A., Pizka, M., Fietzke, A.: Software quality assessment in practice: a hypothesis-driven framework, *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 1–6, Finland (2018)
21. Reddy, B. S., Kothapalli S. R.: Making the strategic goals as hypotheses in software startups: The role of hypothesis driven development and Agile methodologies, *International Journal of Management, IT and Engineering*, **9**(6), 328–337 (2019)
22. Sporse, T., Tkalic, A., Moe, N. B., Mikalsen, M.: Understanding barriers to internal startups in large organizations: evidence from a globally distributed company, IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE), 12–21, Madrid (2021)
23. Melegati, J., Edison, H., Wang, X.: XPro: a model to explain the limited adoption and implementation of experimentation in software startups, *IEEE Transactions on Software Engineering*, **48**(6), 1929–1946 (2020)

24. Alhammad, M. M. Moreno, A. M.: Integrating user experience into Agile: an experience report on lean UX and Scrum, Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training, 146–157, Pennsylvania (2022)
25. Ferreira, B. Kalinowski, M., Gomes, M. V., Marques, M. C., Lopes, H. Barbosa, S.: Investigating Problem Definition and End-User Involvement in Agile Projects that Use Lean Inceptions, XX Brazilian Symposium on Software Quality, 1–10, Brazil (2021)
26. Saltz, J. Sutherland, A.: SKI: A new agile framework that supports DevOps, continuous delivery, and lean hypothesis testing, 53rd Hawaii International Conference on System Sciences, 6617–6226, Hawaii (2020)
27. Mollick, E., *The Unicorn's Shadow: Combating the Dangerous Myths that Hold Back Startups, Founders, and Investors*, University of Pennsylvania Press, Pennsylvania (2020)
28. Silva, D. S. Ghezzi, A. Aguiar, R. B. D., Cortimiglia, M. N., ten Caten, C. S.: Lean startup for opportunity exploitation: adoption constraints and strategies in technology new ventures, *International Journal of Entrepreneurial Behavior and Research*, **27**(4), 944–969 (2021)
29. Gottschalk, S. Rittmeier, F., Engels, G.: Hypothesis-driven adaptation of business models based on product line engineering, IEEE 22nd Conference on Business Informatics, 134–143, Belgium (2020)
30. Cowan, A.: Continuous Design Using Hypothesis-Driven Development, Darden Case No. UVA-OM-1740, Available at SSRN: <https://ssrn.com/abstract=4419039> (2023)
31. Jololian, L., Bowman, A. D.: An Adaptation of the Spiral Model for the Integration of Hypothesis-and Data-Driven Workflows”, Available at SSRN: <https://ssrn.com/abstract=4210196> (2022)
32. Cowan, A.: Facilitating Digital Development with Agile User Stories, Darden Case No. UVA-OM-1739, Available at SSRN: <https://ssrn.com/abstract=4474666> (2023)
33. Vargas, B. P., Signoretti, I. Zorzetti, M. Marczak, S. Bastos R.: On the understanding of experimentation usage in light of lean startup in software development context”, 24th International Conference on Evaluation and Assessment in Software Engineering, 330–335, Norway (2020)
34. Melegati, J., Guerra, E., Wang, X.: Understanding hypotheses engineering in software startups through a gray literature review. *Information and Software Technology*, **133**, 106465 (2021)
35. Flewelling, P.: *The Agile Developer's Handbook: Get more value from your software development: get the best out of the Agile methodology*, Packt Publishing Ltd, Birmingham (2018)
36. Satyal, S.: *Business process improvement with performance-based sequential experiments* Doctoral dissertation, Sydney: UNSW Sydney (2019)
37. Peralta, C. B., Echeveste, M. E., Lermen, F. H., Marcon, A., Tortorella, G.: A framework proposition to identify customer value through lean practices, *Journal of Manufacturing Technology Management*, **31**(4), 725–747 (2020)
38. Eisenmann, T. R. Ries, E. Dillard, S.: Hypothesis-driven entrepreneurship: The lean startup. Harvard business school entrepreneurial management case, 812–095 (2012)
39. Martínez, A., Suescún, E. Noreña, P. A. González L., Pardo, C.: Implementación de prácticas DevOps en un Sistema de Mainframe Legado, *Investigación e Innovación en Ingenierías*, **10**(2), 1–18 (2022)
40. Cowan, A.: Hypothesis-Driven Development (Practitioner's Guide), https://www.alexandercowan.com/hypothesis-driven-development-practitioners-guide/#How_does_HDD_relate_to_agile_Design_Thinking_Lean_Startup_etc.
41. Oracle.: ¿Qué son las pruebas A/B? [https://www.oracle.com/co/cx/marketing/what-is-ab-testing/#:~:text=Las%20pruebas%20A%2FB%20\(tambi%C3%A9n,en%20funci%C3%B3n%20de%20m%C3%A9tricas%20clave](https://www.oracle.com/co/cx/marketing/what-is-ab-testing/#:~:text=Las%20pruebas%20A%2FB%20(tambi%C3%A9n,en%20funci%C3%B3n%20de%20m%C3%A9tricas%20clave).
42. Zhou, X., Jin, Y. Zhang, H., Li, S., Huang, X.: A map of threats to validity of systematic literature reviews in software engineering, *Asia-Pacific Software Engineering Conference*, 153–160, (2016)
43. Ekman, F. Johannesson, S. Peber, E. Sandberg, C.: Test-Driven Development: Drawbacks, Benefits, Industrial Usage and Complementary Methods.
44. Melegati, J., Guerra, E., Wang, X. Understanding Hypotheses Engineering in Software Startups through a Gray.
45. Goniwada, S. R., Goniwada, S. R.: Enterprise Cloud Native Software Engineering. *Cloud Native Architecture and Design: A Handbook for Modern Day Architecture and Design with Enterprise-Grade Examples*, 495–522 (2022)
46. Marlowe, T. J., Kirova, V., Chang, G., Hashmi, O., Masticola, S. P.: *Development and Evolution of Agile* (2020)
47. Arvedahl M. Åkersten, C.: *Exploring the Criticality and Impact of DevOps Practices* (2018)