



Facultad de Ciencias de la Administración

**Carrera de Ingeniería en Sistemas y
Telemática**

**UNA REVISIÓN SISTEMÁTICA DE LA
LITERATURA SOBRE LA INGENIERÍA
DIRIGIDA POR MODELOS ENFOCADA EN LA
INTERFAZ DE USUARIO DE APLICACIONES
DE SOFTWARE**

**Trabajo de titulación previo a la obtención del
grado en Ingeniero en Sistemas y Telemática**

Autor:

Steven Bladimir Suquisupa Roldán

Director:

Ing. Lenin Erazo Garzón MSc.

Cuenca – Ecuador

2023

DEDICATORIA

Este trabajo está dedicado a toda mi familia,
cuyo éxito ha sido una importante fuente de
inspiración para mis logros.

AGRADECIMIENTO

Agradezco el gran apoyo del ingeniero Lenin Erazo Garzón como director de este trabajo; a Dios por hacerme una mejor persona y a mis padres por su apoyo incondicional.

Índice de Contenidos

Dedicatoria.....	i
Agradecimiento.....	ii
Índice de contenidos.....	iii
Índice de tablas.....	iv
Índice de figuras.....	v
Resumen.....	vi
Abstract.....	vi
1. Introducción.....	1
2. Trabajos relacionados	1
3. Materiales y métodos.....	2
3.1 Planeación de la revisión.....	2
3.2 Ejecución de la revisión.....	4
4. Resultados y discusión.....	5
5. Conclusión y trabajo futuro.....	18
6. Referencias.....	19
7. Anexos.....	23

Índice de Tablas

Tabla 1. Cadena de búsqueda automática.....	2
Tabla 2. Lista de control de calidad.....	3
Tabla 3. Formulario de extracción de datos.....	3
Tabla 4. Frecuencia del tipo de paradigma	6
Tabla 5. Frecuencia del tipo de plataforma	7
Tabla 6. Frecuencia del tipo de solución desarrollada.....	8
Tabla 7. Frecuencia del lenguaje de modelado y serialización utilizado.....	10
Tabla 8. Frecuencia de la herramienta de desarrollo utilizada para construir el DSL..	11
Tabla 9. Frecuencia de la herramienta de desarrollo utilizada para apoyar las transformaciones	11
Tabla 10. Frecuencia del lenguaje de programación en el que se generan los artefactos de la interfaz de usuario	12
Tabla 11. Frecuencia del sistema operativo en el que se ejecutan los artefactos generados de la interfaz de usuario	13
Tabla 12. Frecuencia del tipo de validación.....	14
Tabla 13. Frecuencia del tipo de estudio.....	16
Tabla 14. Frecuencia del alcance del enfoque.....	17

Índice de Figuras

Figura 1. Ejecución de la revisión sistemática	4
Figura 2. Distribución de los estudios primarios por año.....	5
Figura 3. Evaluación de la calidad de los estudios primarios.....	5
Figura 4. Distribución porcentual del tipo de paradigma.....	6
Figura 5. Distribución porcentual del tipo de plataforma.....	7
Figura 6. Distribución porcentual del tipo de solución desarrollada	9
Figura 7. Paradigmas de MDE por tipo de plataforma y solución.....	9
Figura 8. Distribución porcentual del lenguaje de modelado y serialización utilizado	10
Figura 9. Distribución porcentual de la herramienta de desarrollo utilizada para construir el DSL	11
Figura 10. Distribución porcentual de la herramienta de desarrollo utilizada para apoyar las transformaciones	12
Figura 11. Distribución porcentual del lenguaje de programación en el que se generan los artefactos de la interfaz de usuario	13
Figura 12. Distribución porcentual del sistema operativo en el que se ejecutan los artefactos generados de la interfaz de usuario	14
Figura 13. Distribución porcentual del tipo de validación	15
Figura 14. Tipo de validación realizada por el tipo de solución MDE propuesto	16
Figura 15. Distribución porcentual del tipo de estudio.....	17
Figura 16. Distribución porcentual del alcance del enfoque.....	18

Resumen:

El presente trabajo de titulación tiene como propósito elaborar una revisión sistemática de la literatura para comprender el estado del arte de la ingeniería de interfaces de usuario basada en modelos, siguiendo las directrices de Kitchenham. El estudio pretende responder a las preguntas de investigación: ¿Cómo se utiliza MDE para construir y mantener las interfaces de usuario de software? y ¿Cómo se aborda la investigación en los estudios relacionados con el uso de MDE para construir y mantener las interfaces de usuario de software? Se recopilaron 1708 estudios primarios y luego de aplicar criterios de inclusión y exclusión; se seleccionaron 51 artículos sobre propuestas relevantes de MDE centradas en interfaces de usuario de software. Se aplicaron métodos cuantitativos y cualitativos basados en criterios de extracción para responder a las preguntas de investigación, se determinaron las ventajas y desventajas de las propuestas existentes e identificaron las brechas y oportunidades de investigación.

Palabras clave: Desarrollo Dirigido por Modelos (MDD), Ingeniería Dirigida por Modelos (MDE), interfaz de usuario (UI), models@run.time, revisión sistemática.

Abstract:

The present degree work aimed to elaborate a systematic literature review to understand the state of the art of model-based user interface engineering, following Kitchenham's guidelines. The study aimed to answer the research questions: How is MDE used to build and maintain software user interfaces? and How is research addressed in studies related to the use of MDE to build and maintain software user interfaces? We collected 1708 primary studies and, after applying inclusion and exclusion criteria, 51 articles on relevant approaches to DEM focused on software user interfaces were selected. Quantitative and qualitative methods based on extraction criteria were applied to answer the research questions, determine the advantages and disadvantages of existing approaches, and identify research gaps and opportunities.

Keywords: Model-Driven Development (MDD), Model-Driven Engineering (MDE), models@run.time, systematic review, user interface (UI)



Este certificado se encuentra en el repositorio digital de la Universidad del Azuay, para verificar su autenticidad escanee el código QR

Este certificado consta de: 1 página

1. Introducción

La creciente complejidad de los sistemas modernos resulta un reto medular para el desarrollo de software (Zeferino & Vilain, 2014; France & Rumpe, 2007). En particular, estos sistemas operan en escenarios altamente distribuidos, en los que predomina la heterogeneidad de los dispositivos, plataformas informáticas y perfiles de usuario; por ejemplo, el tamaño y la resolución de la pantalla, la capacidad de almacenamiento, el ancho de banda, los sistemas operativos, los conocimientos, la experiencia y las capacidades de los usuarios, entre otros (Mitrović et al., 2017; Bendaly et al., 2019). Además, los sistemas modernos se caracterizan por su ubicuidad, pues buscan la adaptación permanente de los requerimientos del usuario en función de su movilidad y contexto (Bendaly et al., 2019; López-Jaquero et al., 2008).

La interfaz de usuario (UI) representa la frontera entre el hombre y la máquina; por lo tanto, garantiza que la experiencia del usuario sea satisfactoria (Zeferino & Vilain, 2014). Sin embargo, la UI es uno de los aspectos más afectados por la creciente complejidad de los sistemas. Como consecuencia, sus actividades de desarrollo y mantenimiento requieren más esfuerzo, tiempo y recursos (Khan et al., 2022; Zhang et al., 2020). Además, cerca de la mitad del presupuesto se invierte en tareas de implementación de la UI (Frey et al., 2014). Por lo tanto, resulta esencial disponer de métodos y herramientas de Ingeniería de Software que faciliten y agilicen las tareas de implementación y mantenimiento de los requisitos de la UI (Zeferino & Vilain, 2014).

La Ingeniería Dirigida por Modelos (MDE) surge como un enfoque adecuado para abstraer la complejidad de los sistemas y mejorar su evolución (Khan et al., 2022). MDE propone el uso de modelos de dominio como artefactos de primera clase para el desarrollo y operación de los sistemas (Erazo-Garzón et al., 2022). Estos modelos describen el sistema desde diferentes puntos de vista o perspectivas (por ejemplo, arquitectura, comportamiento, interfaz de usuario), considerando diferentes niveles de abstracción (Molina et al., 2012). En particular, los modelos pueden utilizarse para capturar las características comunes de la UI y, mediante transformaciones, producir múltiples implementaciones ajustadas a la heterogeneidad de las plataformas tecnológicas, el contexto o las necesidades de los usuarios (Botterweck, 2006). Además, estos modelos se utilizan para razonar sobre el estado y comportamiento del sistema en tiempo de ejecución con el fin de ajustar su UI (Criado et al., 2010). Por lo tanto, MDE proporciona importantes beneficios para aumentar la productividad de los desarrolladores y reducir el mantenimiento de la interfaz de usuario del sistema.

En la literatura, varias revisiones sistemáticas (Gottardi & Vaccare, 2018; Bencomo et al., 2019; Tufail et al., 2019; Bucchiarone et al., 2020; Shamsujjoha et al., 2021; Ordóñez et al., 2022) abordan, en general, el estado del conocimiento de MDE o estudian otras perspectivas del sistema diferentes a la UI. Por lo tanto, no existe una revisión sistemática actualizada orientada a la ingeniería de la UI dirigida por modelos que evidencien su estado particular para comprender aspectos, tales como: tipos de plataformas y soluciones a los que se orientan las propuestas existentes, los modelos, lenguajes y herramientas que utilizan, los métodos de evaluación empírica que aplican, entre otros.

En consecuencia, este trabajo de titulación tiene como objetivo elaborar una revisión sistemática de la literatura para conocer el estado del arte sobre el uso de MDE para apoyar el desarrollo de la UI de las aplicaciones de software, para lo cual se han seguido las directrices de Kitchenham & Charters (2007). Este estudio pretende responder a las siguientes preguntas de investigación: i) ¿Cómo se utiliza MDE para construir y mantener las interfaces de usuario de software? y ii) ¿Cómo se aborda la investigación en los estudios relacionados sobre el uso de MDE para construir y mantener las interfaces de usuario de software? En primer lugar, se recuperaron automáticamente 1708 estudios primarios de múltiples fuentes de investigación. A continuación, aplicando criterios de inclusión y exclusión, se seleccionaron 51 artículos sobre propuestas de MDE centradas en la UI de software. Además, se realizó una evaluación de la calidad de los estudios seleccionados para clasificarlos según su relevancia científica. Por último, se aplicaron métodos cuantitativos y cualitativos basados en criterios de extracción para responder a las preguntas de investigación propuestas.

La estructura de este trabajo es la siguiente: la sección 2 presenta una revisión de los trabajos relacionados; la sección 3 describe la metodología y el protocolo utilizados en la revisión sistemática; la sección 4 incluye una discusión de los resultados de la revisión sistemática, destacando las fortalezas y débiles de los estudios existentes, así como las brechas, retos y oportunidades de investigación en esta área. Por último, la sección 5 presenta las conclusiones y las líneas de trabajo futuro.

2. Trabajos relacionados

Esta sección presenta una descripción de las principales revisiones y mapeos sistemáticos existentes en la literatura que tiene como objetivo mostrar el estado tecnológico de MDE.

Gottardi & Vaccare (2018) presentan un mapeo sistemático para identificar los dominios tecnológicos y de aplicación en los que MDE tiene éxito, así como los retos de la aplicación de esta metodología a los procesos de desarrollo de propósito general. Asimismo, Bucchiarone et al. (2020) presentan un debate sobre los retos

y las oportunidades de investigación para la comunidad científica de MDE. En cuanto a la aplicación de los modelos en tiempo de ejecución, Bencomo et al. (2019) analizan el estado del arte de esta línea de investigación. Para ello, utilizan una taxonomía para comparar los principales enfoques y los resultados de investigación en esta área durante la última década.

Otro trabajo más específico, Shamsujjoha et al. (2021) investigan qué técnicas y metodologías de Desarrollo Dirigido por Modelos (MDD) se han utilizado para apoyar el desarrollo de aplicaciones móviles y cómo se han empleado estas técnicas, identificando los beneficios, limitaciones, brechas y oportunidades de investigación futuras. Así también, en su revisión sistemática, Tufail et al. (2019) exhiben un análisis de las herramientas de MDD propuestas para aplicaciones móviles. Por último, Ordóñez et al. (2022) ejecutan una revisión sistemática de MDD y la accesibilidad del software, incluyendo la revisión de las normas de accesibilidad y una evaluación cualitativa de las propuestas existentes.

En conclusión, se puede determinar que las revisiones y mapeos sistemáticos existentes sobre MDE no profundizan su estudio desde el punto de vista de las interfaces de usuario. Por lo tanto, este artículo pretende ayudar a los investigadores a suplir esta limitación de información e identificar brechas y oportunidades de investigación en este campo para fomentar futuros proyectos.

3. Materiales y métodos

Esta revisión sistemática se basa en los procedimientos y directrices propuestos por Kitchenham & Charters (2007) para garantizar un proceso fiable, replicable y auditable. Esta metodología consta de tres etapas: i) planificación de la revisión, para identificar los objetivos y validar el protocolo de revisión; ii) realización de la revisión, para llevar a cabo el protocolo previamente definido; y iii) difusión de la revisión, para elaborar un informe con los resultados de la revisión.

3.1 Planeación de la revisión

Preguntas de investigación. El objetivo general de esta revisión sistemática es comprender el estado del conocimiento tecnológico sobre la ingeniería de interfaces de usuario basada en modelos. Las preguntas de investigación propuestas son:

RQ1. ¿Cómo se utiliza MDE para construir y mantener las interfaces de usuario de software?

RQ2. ¿Cómo se aborda la investigación en los estudios relacionados con el uso de MDE para construir y mantener las interfaces de usuario de software?

Fuentes de datos y estrategia de búsqueda. Las fuentes de información seleccionadas para la búsqueda automática de estudios primarios fueron los indexadores Scopus y Google Scholar. Para la búsqueda manual se seleccionaron las conferencias, revistas y libros más significativos sobre MDE orientados a la UI de software. Además, se aplicó la técnica de bola de nieve para encontrar otros artículos relevantes. La Tabla 1 presenta la cadena de búsqueda utilizada, que se aplicó a los metadatos del artículo: título, resumen y palabras clave.

Tabla 1

Cadena de búsqueda automática

Concepto	Conector	Sinónimos y acrónimos
user interface	AND	UI, GUI
model-driven engineering	OR	MDE, based-model
model-driven development	OR	MDD
model-driven architecture	OR	MDA
domain-specific language	OR	DSL, DSML
metamodel	OR	meta-model
model at runtime	OR	runtime model, models@runtime, models@run.time

Cadena de búsqueda:

(user interface OR ui OR gui) AND (model-driven OR based-model OR mde OR mdd OR mda OR domain-specific language OR dsl OR dsml OR metamodel OR meta-model OR model at runtime OR runtime model OR models@runtime OR models@run.time)

Selección de estudios primarios. Los estudios primarios obtenidos a partir de la búsqueda automática y manual fueron evaluados y seleccionados por cuatro investigadores basándose en el título, el resumen y las

palabras clave. Las discrepancias en la selección de los estudios se resolvieron por consenso, tras revisar el documento completo.

Se incluyeron los estudios primarios que cumplieran cualquiera de los siguientes criterios de inclusión:

- IC1. Estudios primarios que aborden paradigmas, enfoques, métodos o técnicas de MDE para diseñar y construir interfaces de usuario de software.
- IC2. Estudios primarios que proponen herramientas o aplicaciones basadas en MDE para diseñar y construir interfaces de usuario de software.

Se excluyeron los estudios primarios que cumplieran al menos uno de los siguientes criterios de exclusión:

- EC1. Editoriales, prólogos, opiniones, entrevistas, noticias o carteles.
- EC2. Estudios duplicados en diferentes fuentes.
- EC3. Artículos cortos de menos de cuatro páginas.
- EC4. Artículos escritos en un idioma distinto del inglés.
- EC5. Literatura gris, tesis y documentos de introducción.

Evaluación de la calidad de los estudios primarios. Se estableció una lista de comprobación de tres preguntas para evaluar la calidad de los estudios seleccionados (ver Tabla 2). Cada pregunta se evaluó utilizando una escala de 0 a 1. Las puntuaciones obtenidas en las tres preguntas se sumaron para determinar la puntuación total de cada artículo. La evaluación de la calidad se utilizó únicamente para clasificar los estudios según su relevancia científica, y para tener una síntesis adecuada en la presentación de los resultados.

Tabla 2

Lista de control de calidad

No.	Pregunta	Respuesta y puntuación
QAQ1	¿Cuántas citas tiene el estudio primario?	Más de 10 citas, muy relevante (1) Entre 1 y 10 citas, relevante (0,5) No se cita, irrelevante (0)
QAQ2	¿Se ha publicado el estudio primario en revistas o conferencias relevantes?	Muy relevante (1) Relevante (0,5) Irrelevante (0)
QAQ3	¿El estudio presenta una evaluación empírica de la solución propuesta?	Sí (1) No (0)

Estrategia de extracción de datos. Se utilizó un formulario de extracción de datos para facilitar y estandarizar la recolección y sistematización de la información de los artículos, así como para responder a las preguntas de investigación inicialmente propuestas (ver Tabla 3).

Tabla 3

Formulario de extracción de datos

RQ1. ¿Cómo se utiliza MDE para construir y mantener las interfaces de usuario de software?		
EC1.	Tipo de paradigma.	[Desarrollo dirigido por modelos, models@run.time]
EC2.	Tipo de plataforma.	[Aplicación web, Aplicación móvil, Aplicación de escritorio, No específica]
EC3.	Tipo de solución desarrollada.	[Metodología, Framework, Arquitectura, Middleware, Metamodelo, DSL, Herramientas de transformación]
EC4.	Lenguaje de modelado y serialización utilizado.	[MOF, ECORE-XMI, UML, IFML, OCL, XML, KERMETA, GOPRR, OWL, JSON]
EC5.	Herramienta de desarrollo utilizada para construir el DSL.	[Sirius, Xtext, EFM, MetaEdit+, Papyrus]
EC6.	Herramienta de desarrollo utilizada para apoyar las transformaciones.	[ATL, Acceleo, M2T, Xtend, Xpand, MofScript]
EC7.	Lenguaje de programación en el que se generan los artefactos de la interfaz de usuario.	[Java, JavaScript, Python, C++, C#, Objective-C, HTML, XAML, ASP.NET].

EC8.	Sistema operativo en el que se ejecutan los artefactos generados de la interfaz de usuario.	[Windows, Android, iOS, Windows Phone, BlackBerry OS]
RQ2. ¿Cómo se aborda la investigación en los estudios relacionados con el uso de MDE para construir y mantener las interfaces de usuario de software?		
EC9.	Tipo de validación.	[Prueba de concepto, Encuesta, Estudio de caso, Experimento, Ninguno]
EC10.	Tipo de estudio.	[Nuevo, Extensión]
EC11.	Alcance del enfoque.	[Industria, Academia]

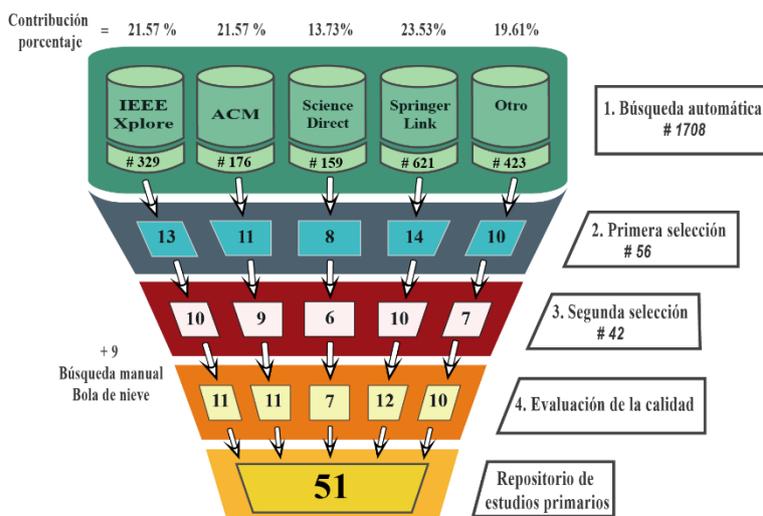
Métodos de análisis y síntesis. Se utilizaron dos métodos para el análisis y síntesis de la información recogida: i) cuantitativo, basado en la construcción de diferentes tipos de gráficos para representar la frecuencia de las respuestas para cada criterio de extracción o combinación de criterios de extracción; y ii) cualitativo, descripción de las propuestas más relevantes, identificando sus puntos fuertes, débiles y las oportunidades de investigación.

3.2 Ejecución de la revisión

En esta subsección se describe el proceso de recogida, selección y evaluación de la calidad de los estudios primarios, utilizando los criterios de inclusión, exclusión y calidad definidos en el protocolo de revisión. Aunque la búsqueda automática se realizó mediante indexadores, la Figura 1 muestra los resultados obtenidos en cada actividad de este proceso, clasificados según las principales bibliotecas digitales informáticas. A continuación, se describe brevemente cada actividad del proceso:

1. *Búsqueda automática.* La cadena de búsqueda se adaptó y ejecutó en cada indexador, recogiendo 1708 estudios primarios para su posterior análisis.
2. *Primera selección.* Los títulos, resúmenes y palabras clave de los estudios recuperados se evaluaron de acuerdo con los criterios de inclusión y exclusión para determinar su pertinencia con respecto al tema de investigación. Como resultado, se seleccionaron 56 estudios.
3. *Segunda selección.* En esta actividad, las discrepancias y dudas sobre la selección de algunos artículos se resolvieron por consenso entre todos los investigadores participantes una vez revisado el artículo completo. Como resultado, el repositorio se redujo a 42 estudios. A su vez, se realizó una búsqueda manual de artículos; y se aplicó la técnica de bola de nieve a los artículos seleccionados, incluyendo 9 estudios adicionales, teniendo un total de 51 estudios definitivos.
4. *Evaluación de la calidad.* Como actividad final, se clasificaron los estudios primarios seleccionados, según su nivel de rigor científico y relevancia.

Figura 1
Ejecución de la revisión sistemática



4. Resultados y discusión

Como resultado de la ejecución de la revisión sistemática, se han seleccionado 51 estudios primarios sobre propuestas de MDE orientadas a la UI, los cuales se encuentran enlistados y codificados en el Anexo 2. A su vez, una síntesis de los resultados obtenidos en este trabajo ha sido publicada en Erazo-Garzón et al. (2022). La Figura 2 presenta la distribución de los estudios por años. Se puede observar que a partir del 2005 comienzan a publicarse artículos sobre MDE aplicados a la UI, siendo los años de mayor publicación 2010, 2013 y 2015 con 9, 7 y 6 artículos, respectivamente. Sin embargo, también se observan varios años con un bajo número de publicaciones, lo que lleva a pensar que es una línea de investigación que no tiene un interés permanente por parte de los investigadores, sobre todo a partir del 2015 se muestra una disminución considerable de publicaciones. Según el tipo de publicación, 32 estudios han sido publicados en conferencias, seguido de 17 estudios en revistas; y, finalmente, 2 estudios son capítulos de libros.

En cuanto a la evaluación de la calidad de los estudios (ver Figura 3), la pregunta QAQ1 ha obtenido resultados muy satisfactorios, con 27 estudios puntuados como muy relevantes (más de 10 citas), 22 estudios como relevantes (entre 1 y 10 citas), y apenas 2 estudios como irrelevantes (no citados). Del mismo modo, la pregunta QAQ2 presenta una valoración favorable, con 25 y 13 estudios publicados en revistas o congresos muy relevantes y relevantes, respectivamente. Por último, la pregunta QAQ3 muestra un número significativo de estudios (15) que no han sido objeto de una evaluación empírica. Además, de los 36 estudios que incluyen una evaluación de su solución propuesta, varios estudios (13) solo aplican una prueba de concepto. En consecuencia, la falta de evaluación o el tipo de evaluación utilizado es la principal amenaza de la mayoría de las soluciones propuestas en cuanto a su validez, fiabilidad y generalización.

Figura 2

Distribución de los estudios primarios por año

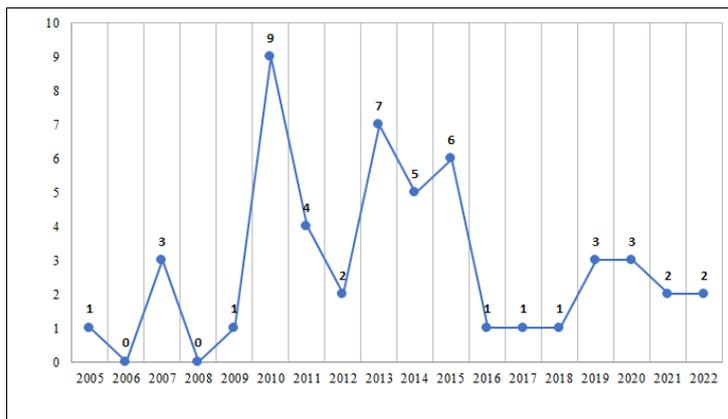
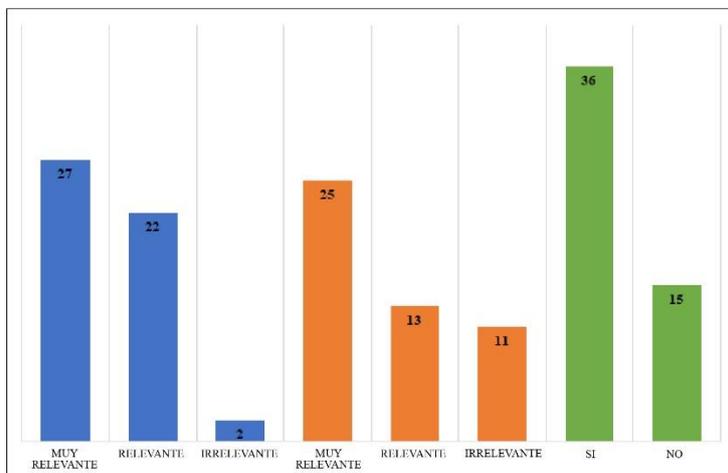


Figura 3

Evaluación de la calidad de los estudios primarios



RQ1. ¿Cómo se utiliza MDE para diseñar y construir las interfaces de usuario de software?

EC1: Tipo de paradigma.

Este criterio de extracción pretende determinar en qué medida los investigadores utilizan los paradigmas Desarrollo dirigido por modelos (MDD) y *models@run.time* para apoyar el desarrollo y la operación de la interfaz de usuario del software, respectivamente. En este sentido, de los 51 estudios, 36 (70,59%) aplican los modelos solo en tiempo de diseño para aumentar la abstracción y automatización del proceso de desarrollo de artefactos de interfaz de usuario, mientras que solo 15 (29,41%) cubren el uso de modelos en tiempo de ejecución. Por lo tanto, una importante oportunidad de investigación se relaciona con ampliar el uso de los modelos en tiempo de ejecución para mantener una relación causal bidireccional entre el modelo y el sistema en operación. De este modo, se podría disponer de soluciones más robustas y autónomas que analicen los modelos en tiempo de ejecución para apoyar la adaptación de la UI, según el comportamiento del entorno operativo del sistema (ver Tabla y Figura 4).

Tabla 4

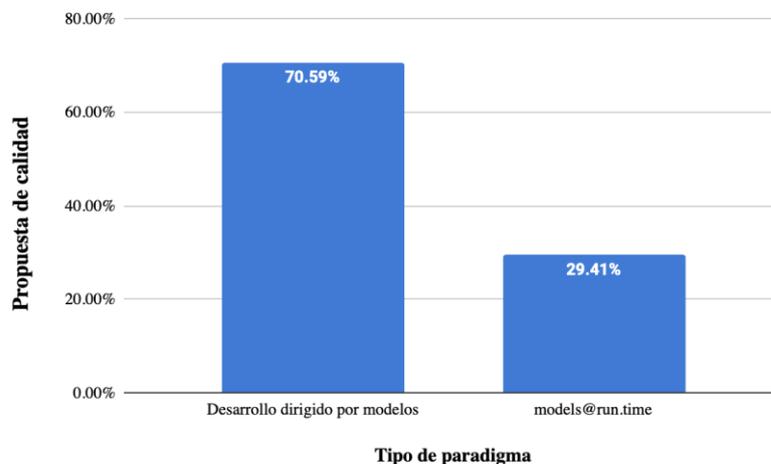
Frecuencia del tipo de paradigma

Tipo de Paradigma	Estudios Relevantes	Frecuencia
Desarrollo dirigido por modelos	EP2, EP3, EP4, EP5, EP6, EP7, EP8, EP10, EP11, EP12, EP13, EP14, EP15, EP16, EP17, EP18, EP19, EP20, EP23, EP24, EP26, EP28, EP29, EP30, EP31, EP32, EP33, EP34, EP35, EP37, EP38, EP39, EP40, EP41, EP42, EP48	36
<i>models@run.time</i>	EP1, EP9, EP21, EP22, EP25, EP27, EP36, EP43, EP44, EP45, EP46, EP47, EP49, EP50, EP51	15

Entre los estudios más relevantes sobre el desarrollo dirigido por modelos se encuentra el de Bouchelligua et al. (2010), quienes proponen un enfoque basado en modelos para la generación de una UI adaptativa mediante transformaciones parametrizadas desde una interfaz abstracta a una interfaz concreta. Estos parámetros de transformación se determinan en función del contexto de uso de la aplicación. Por su parte, Bendaly et al. (2019) presentan un enfoque genérico que toma como entradas un modelo de contexto de accesibilidad de un usuario determinado y el modelo de interfaz de usuario para proporcionar como salida un modelo de interfaz de usuario adaptado. Con relación a *models@run.time*, Criado et al. (2014) diseñan una propuesta para auto adaptar la estructura de la interfaz de usuario basada en componentes en tiempo de ejecución mediante la transformación dinámica del modelo, ya que las reglas que describen su comportamiento no están preestablecidas, sino más bien son seleccionadas de un repositorio según el contexto de operación.

Figura 4

Distribución porcentual del tipo de paradigma



EC2: Tipo de plataforma

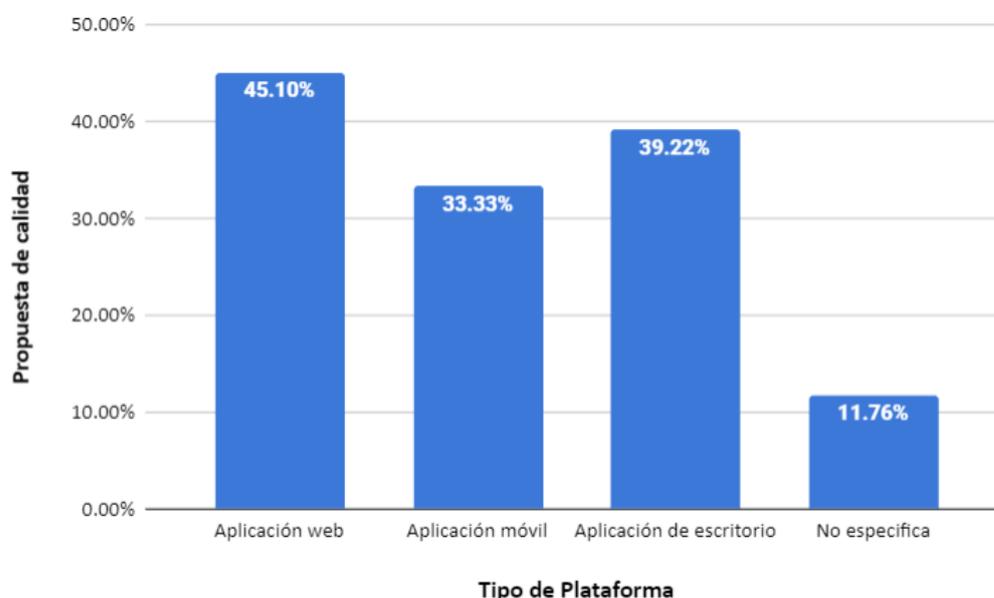
Este criterio pretende conocer el tipo de plataformas a las que se orientan las propuestas de ingeniería de interfaz de usuario basada en modelos. La Tabla y Figura 5 muestra una contribución uniforme de las propuestas para las diferentes plataformas; es decir, el 45,10% de los estudios se orientan a la web, seguido del 39,22% a aplicaciones de escritorio, y el 33,33% a aplicaciones móviles. Entre los estudios más importantes, Sottet et al. (2015) proponen un enfoque de gestión de la variabilidad integrado a un proceso de prototipado rápido de la interfaz de usuario web, lo cual implica la combinación de MDE y líneas de productos de software; mientras que Diep et al. (2013) construyen un entorno de desarrollo integrado basado en modelos de interfaz de usuario independientes de la plataforma para crear interfaces de usuario para aplicaciones móviles en tres plataformas diferentes: Android, iOS y Windows Phone.

Tabla 5
Frecuencia del tipo de plataforma

Tipo de Plataforma	Estudios Relevantes	Frecuencia
Aplicación Web	EP1, EP5, EP7, EP8, EP9, EP11, EP13, EP15, EP16, EP18, EP20, EP21, EP23, EP25, EP27, EP28, EP32, EP36, EP39, EP40, EP43, EP46, EP49	23
Aplicación móvil	EP1, EP2, EP3, EP8, EP10, EP11, EP13, EP19, EP23, EP24, EP32, EP34, EP36, EP42, EP45, EP48, EP50	17
Aplicación de escritorio	EP4, EP6, EP8, EP11, EP12, EP22, EP29, EP31, EP32, EP35, EP36, EP37, EP38, EP41, EP42, EP44, EP45, EP46, EP50, EP51	20
No específica	EP14, EP17, EP26, EP30, EP33, EP47	6

Un aspecto destacable es la existencia de algunas propuestas multiplataforma, lo que demuestra la preocupación de los investigadores por crear soluciones que abstraigan la heterogeneidad de las plataformas y permitan, a partir de un mismo modelo de alto nivel, generar artefactos de UI para varios tipos de plataformas. Un ejemplo claro es MANTRA (Botterweck, 2006), un enfoque basado en modelos abstractos de UI para generar código de implementación para diferentes plataformas de UI (web, móvil, escritorio). Además, en los últimos años han aparecido notables propuestas sobre MDE para apoyar la construcción de cuadros de mando e infografías, especialmente en dominios como IoT y smart city (De Morais et al., 2018; Rojas et al., 2020; De Sanctis et al., 2022).

Figura 5
Distribución porcentual del tipo de plataforma



EC3: Tipo de solución desarrollada. Los metamodelos representan el tipo de solución que se incluye principalmente en los estudios (66,67%). Sin embargo, un número menor de estudios (45,10%) da un paso más adelante para proponer una herramienta DSL. Un aspecto destacable es que el 78,26% de los DSL propuestos tienen una notación gráfica de alto nivel. A su vez, también hay un número importante de propuestas de herramientas de transformación (66,67%) para alcanzar la automatización de los modelos traduciéndolos a código. En concreto, el 54,90% de las propuestas incluyen transformaciones de modelo a modelo, mientras que el 35,29% transformaciones de modelo a texto; existen estudios que combinan ambos tipos de transformaciones (ver Tabla y Figura 6).

En cuanto a las soluciones metodológicas, la revisión encontró un grupo importante de estudios (54,90%) que describen en detalle el proceso propuesto para diseñar modelos de dominio independientes de la plataforma y transformarlos en artefactos de interfaz de usuario concretos. Igualmente, existen varios estudios (39,22%) alineados con la Arquitectura Dirigida por Modelos (MDA) propuesta por el Object Management Group (OMG), extendiendo y adaptando esta arquitectura al dominio de la UI. A su vez, en menor medida, se han propuesto marcos de trabajo (27,45%) basados en modelos como solución integral para automatizar la construcción y evolución de UIs. Por último, existen escasas soluciones de tipo middleware (3,92%) centradas principalmente en el uso de modelos en tiempo de ejecución para adaptar la UI de acuerdo al entorno operativo (ver Tabla y Figura 6).

En resumen, es necesario aprovechar los metamodelos existentes para la construcción de DSL con una notación (sintaxis concreta) que proporcione una interfaz de modelado fácil de entender y utilizar para los desarrolladores y expertos del dominio. Asimismo, en consonancia con la recomendación del criterio anterior, es necesario profundizar en la investigación de middlewares basados en modelos de tiempo de ejecución.

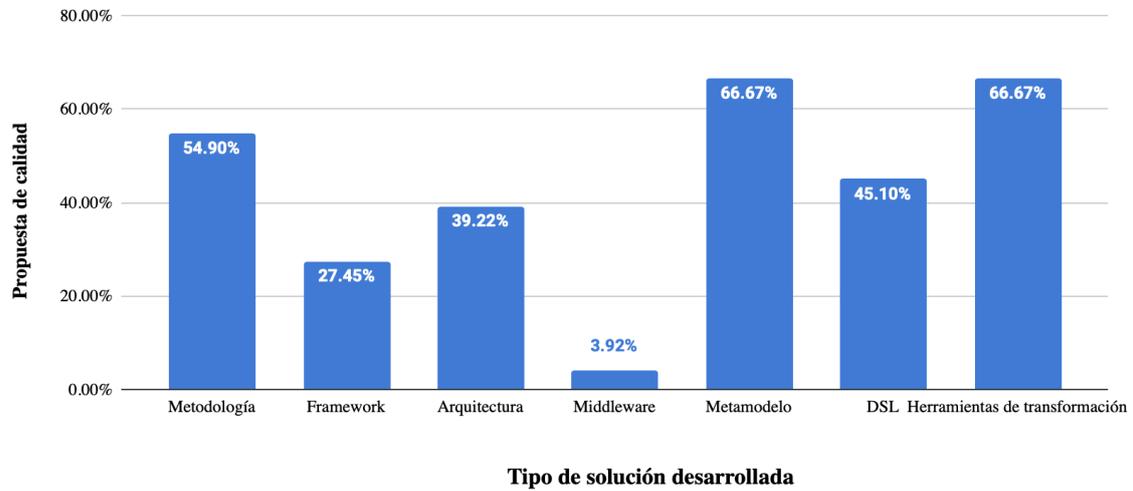
Tabla 6
Frecuencia del tipo de solución desarrollada

Tipo de Solución	Estudios Relevantes	Frecuencia
Metodología	EP1, EP6, EP7, EP8, EP14, EP15, EP17, EP19, EP20, EP23, EP24, EP25, EP27, EP28, EP29, EP30, EP32, EP35, EP37, EP38, EP39, EP40, EP41, EP42, EP45, EP46, EP47, EP49	28
Framework	EP1, EP2, EP3, EP5, EP9, EP14, EP19, EP20, EP21, EP22, EP31, EP39, EP43, EP44	14
Arquitectura	EP1, EP2, EP5, EP10, EP13, EP14, EP15, EP17, EP19, EP22, EP24, EP28, EP29, EP32, EP34, EP38, EP41, EP49, EP50, EP51	20
Middleware	EP25, EP44	2
Metamodelo	EP1, EP2, EP3, EP4, EP5, EP6, EP8, EP10, EP11, EP12, EP13, EP15, EP16, EP17, EP19, EP20, EP23, EP24, EP25, EP26, EP27, EP29, EP32, EP33, EP36, EP38, EP40, EP42, EP46, EP47, EP48, EP49, EP50, EP51	34
DSL	EP1, EP2, EP3, EP5, EP6, EP10, EP11, EP13, EP18, EP19, EP22, EP23, EP24, EP28, EP29, EP32, EP35, EP36, EP37, EP41, EP48, EP50, EP51	23
Herramientas de transformación	EP1, EP2, EP3, EP4, EP5, EP6, EP8, EP10, EP12, EP13, EP16, EP18, EP19, EP20, EP23, EP24, EP25, EP27, EP28, EP29, EP32, EP33, EP34, EP35, EP36, EP38, EP40, EP41, EP42, EP45, EP46, EP47, EP49, EP50	34

Un estudio relevante es el propuesto por Molina et al. (2012), quienes presentan una solución MDD completa para construir interfaces gráficas de usuario a partir de modelos declarativos. La solución comprende un método llamado CIAT-GUI y un conjunto de artefactos y herramientas que soportan este método, como metamodelos, DSL gráficos y generadores automáticos de la interfaz de usuario concreta. El estudio también presenta la arquitectura de la solución y su relación con los niveles de MDA. Por su parte, Zeferino y Vilain (2014) proponen modelar la interacción usuario-sistema a un alto nivel de abstracción y generación automática de código utilizando la transformación automática de modelos y el diagrama de interacción del usuario (UID). Por otro lado, Khan et al. (2022) proporcionan una descripción general del lenguaje de modelado unificado para interfaces de usuario móviles (UMMUI), que utiliza la notación estándar del lenguaje de modelado unificado (UML) para representar los requisitos de la interfaz

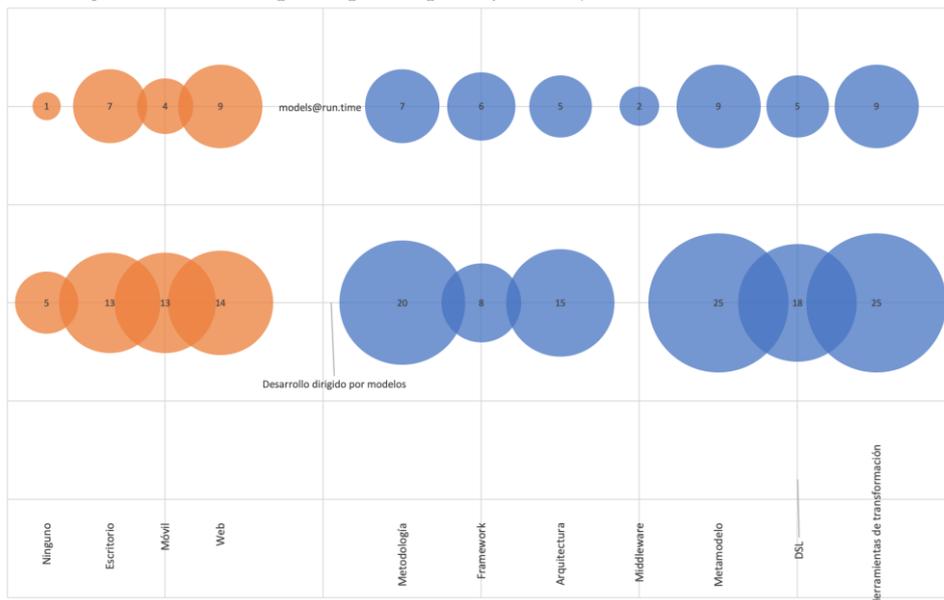
de usuario móvil en un alto nivel de abstracción y desarrollando un motor de generación de código abierto para transformar automáticamente los modelos.

Figura 6
Distribución porcentual del tipo de solución desarrollada



La Figura 7 presenta un gráfico de burbuja con la distribución de las propuestas de MDE según una combinación de tres criterios de extracción: i) tipo de paradigma (EC1), ii) tipo de plataforma (EC2) y iii) tipo de solución desarrollada (EC3). En este gráfico se puede observar una equilibrada propuesta de soluciones de Desarrollo dirigido por modelos para todos los tipos de plataforma (web, escritorio y móvil); sin embargo, las soluciones basadas en el paradigma *models@run.time* se han orientado fundamentalmente a la web y escritorio, existiendo pocas propuestas de este tipo para móvil. Con respecto a tipos de soluciones basadas en el Desarrollo dirigido por modelos, en orden de frecuencia, predominan metamodelos, herramientas de transformación, métodos, DSLs y arquitecturas; mientras que, para soluciones basadas en *models@run.time* su distribución es similar al Desarrollo dirigido por modelos.

Figura 7
Paradigmas de MDE por tipo de plataforma y solución



EC4: Lenguaje de modelado y serialización utilizado. Entre los principales lenguajes de modelado y serialización utilizados por las propuestas de MDE para la UI, ordenados según su nivel de recurrencia, se encuentran: XML con 19 estudios (37.25%); ECORE – XMI con 13 estudios (25.49%), UML con 12

estudios (23.52%); MOF con 10 estudios (19.61%); a su vez, OCL, OWL y KERMETA con 2 estudios (3.92%) cada uno; y, finalmente JSON con 1 estudio (1.96%) (ver Tabla 7 y Figura 8).

Tabla 7

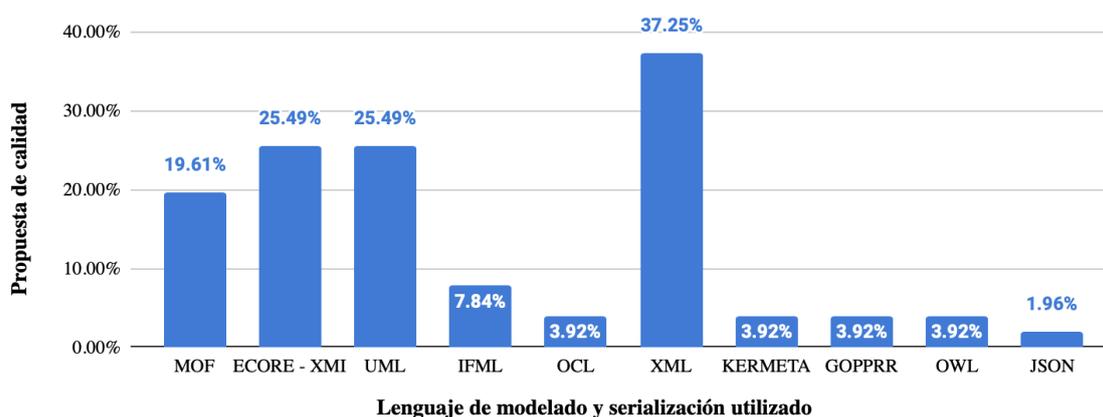
Frecuencia del lenguaje de modelado y serialización utilizado

Lenguaje	Estudios Relevantes	Frecuencia
MOF	EP5, EP6, EP12, EP13, EP19, EP24, EP29, EP33, EP40, EP43	10
ECORE -XMI	EP3, EP4, EP5, EP6, EP8, EP12, EP13, EP16, EP18, EP23, EP32, EP48, EP51	13
UML	EP2, EP4, EP10, EP12, EP13, EP19, EP23, EP24, EP29, EP32, EP35, EP41	12
IFML	EP7, EP11, EP15, EP39	4
OCL	EP27, EP49	2
XML	EP3, EP4, EP5, EP6, EP12, EP19, EP20, EP24, EP32, EP33, EP34, EP36, EP37, EP40, EP41, EP42, EP43, EP44, EP45	19
KERMETA	EP20, EP38	2
GOPRR	EP41, EP48	2
OWL	EP17, EP43	2
JSON	EP6	1

Entre los trabajos más relevante basados en UML está el de Molina et al. (2012), quienes mencionan que el desarrollo de sistemas interactivos es un proceso complejo, por lo que, proponen un modelo para simplificar y automatizar el proceso de desarrollo de software; y, a partir de este modelo alcanzar la portabilidad, compatibilidad e independencia de la plataforma tecnológica. Otro trabajo importante, es el de Alí et al. (2021), proponen un nuevo marco basado en modelos, con el fin de diseñar rápidamente interfaces de usuario adaptadas a diferentes tamaños de pantalla. El marco incluye una capa de modelo/metamodelo M2 ECORE que permite modelar escenarios de interfaz de usuario y generar código ejecutable específico en la plataforma Java.

Figura 8

Distribución porcentual del lenguaje de modelado y serialización utilizado

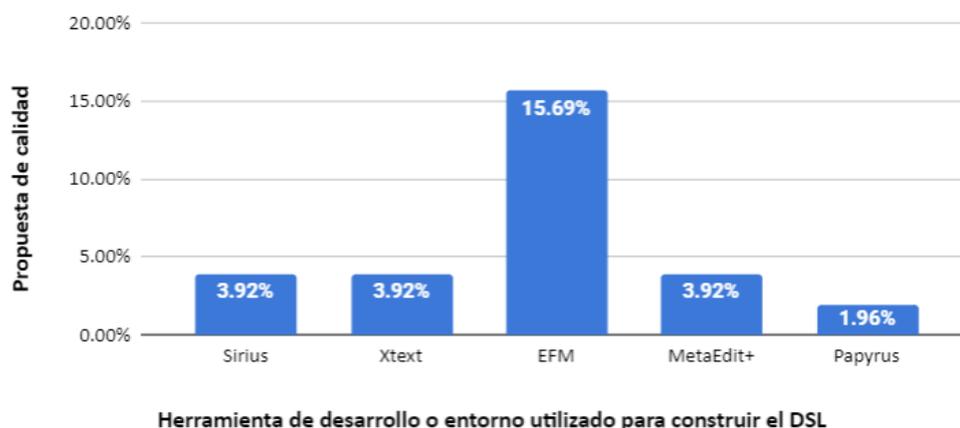


EC5: Herramienta de desarrollo utilizada para construir el DSL. En la Tabla 8 y Figura 9 se presenta la distribución de las herramientas utilizada por los investigadores para construir DSLs. En este sentido, EMF (Eclipse Model Framework) es la herramienta más empleada, con 8 estudios relevantes que corresponden al 15.69%. A su vez, muy por debajo se encuentran las herramientas Sirius, MetaEdit+ y Xtext con 2 estudios (3.92%) cada uno; finalmente Papyrus con 1 estudio (1.96%).

Tabla 8*Frecuencia de la herramienta de desarrollo utilizada para construir el DSL*

Herramienta de desarrollo	Estudios Relevantes	Frecuencia
Sirius	EP3, EP5	2
Xtext	EP10, EP13	2
EFM	EP5, EP6, EP13, EP24, EP32, EP49, EP51	8
MetaEdit+	EP41, EP48	2
Papyrus	EP2	1

Uno de los estudios más relevantes sobre herramientas de desarrollo para la construcción de DSL es el propuesto por Rojas et al. (2020), quienes presentan Cities-Board un marco para el desarrollo automatizado de tableros de ciudades inteligentes basado en técnicas de modelado. Cities-Board, a través de Sirius, propone la creación de un lenguaje gráfico de dominio específico (DSL) como un proyecto de software de Eclipse, mientras que el uso de Eclipse Modeling Framework (EMF) permite la creación de modelos de tablero conceptualmente más cercanos a las ciudades. En otro estudio, Lachgar y Abdali (2015) sugieren construir un DSL basado en la propia solución Xtext para preservar el código optimizado, importante para el desarrollo de software, pues asegura modularidad, legibilidad y mantenibilidad. Xtext proporciona varias funciones para mantener las aplicaciones DSL limpias y modulares, gracias a su descomposición en diferentes aspectos.

Figura 9*Distribución porcentual de la herramienta de desarrollo utilizada para construir el DSL*

EC6: Herramienta de desarrollo utilizada para apoyar las transformaciones. De acuerdo con los resultados obtenidos en la revisión, 9 de las propuestas estudiadas, es decir, el 17.65 % de los estudios construyen las transformaciones con ATL, debido a que su implementación es bastante robusta; además de ser un lenguaje híbrido que permite construcciones tanto declarativas como imperativas. En menor proporción, se utiliza ACCELEO en 4 estudios (7.84%), seguido de M2T en 3 estudios (5.88%), Xtend en 2 estudios (3.92%); y, Xpand y MofScript en 1 estudio (1.96%) cada uno (ver Tabla 9 y Figura 10).

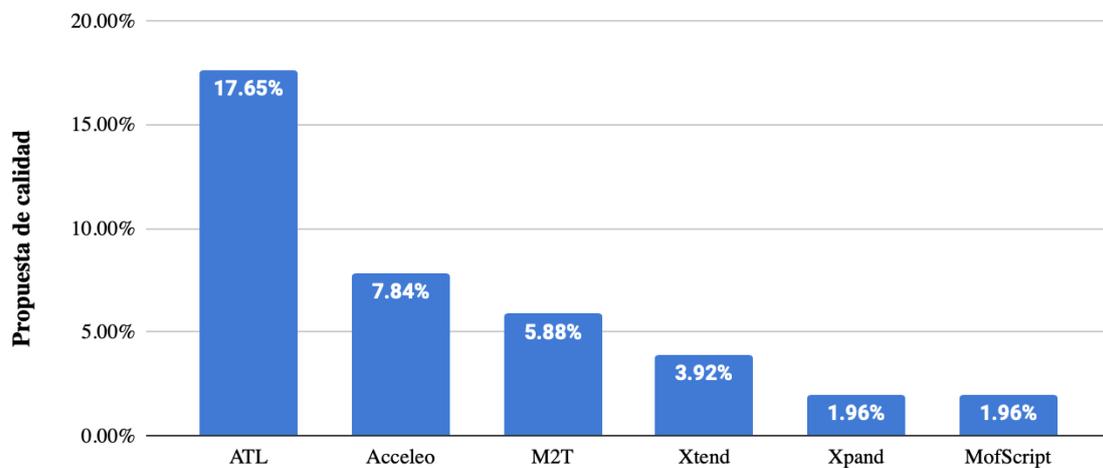
Tabla 9*Frecuencia de la herramienta de desarrollo utilizada para apoyar las transformaciones*

Herramienta de desarrollo	Estudios Relevantes	Frecuencia
ATL	EP4, EP5, EP16, EP23, EP24, EP32, EP33, EP47, EP49	9
Acceleo	EP3, EP5, EP12, EP16	4
M2T	EP1, EP18, EP19	3
Xtend	EP10, EP13	2
Xpand	EP23	1
MofScript	EP24	1

Sabraoui et al. (2010), proponen un enfoque basado en MDA para generar GUI para aplicaciones móviles. Este estudio utiliza el lenguaje ATL para la transformación de modelo a modelo, debido a su facilidad de uso, cercanía a los estándares y al completo conjunto de herramientas de desarrollo que permite ATL, al estar integrado a Eclipse. Para automatizar el proceso de transformación de modelo a código, es necesario definir generadores encargados de interpretar la semántica del modelo de dominio a la semántica del lenguaje de programación correspondiente. En esta propuesta se utiliza la generación de código basada en el lenguaje Xpand para generar automáticamente código GUI para aplicaciones móviles en plataformas Android y BlackBerry. A su vez, Ian y Favre (2005) proponen un enfoque basado en modelos para satisfacer las necesidades de personalización de las interfaces de usuario visuales, que se realiza mediante el uso de EMF para definir un metamodelo en Java y luego mediante transformaciones ATL convertir estos modelos de Java en modelos de vista. Finalmente, el modelo de vista se representa utilizando un conjunto de herramientas de visualización interactiva.

Figura 10

Distribución porcentual de la herramienta de desarrollo utilizada para apoyar las transformaciones



Herramienta de desarrollo utilizada para apoyar las transformaciones.

EC7: Lenguaje de programación en el que se generan los artefactos de la interfaz de usuario. Los resultados de este criterio demuestran que los enfoques estudiados generan el código de los artefactos de la interfaz de usuario de destino en una variedad de lenguajes; sin embargo, mayoritariamente se usa Java con 22 estudios (43.14%), seguido de HTML con 12 estudios (23.53%) y JavaScript con 8 estudios (15.69%). Mientras que muy por debajo, se encuentran Python, C++, C# y XAML con 2 estudios (3.92%), respectivamente. Finalmente, Objective-C y ASP.NET con un estudio (1.96%) cada uno (ver Tabla 10 y Figura 11).

Tabla 10

Frecuencia del lenguaje de programación en el que se generan los artefactos de la interfaz de usuario

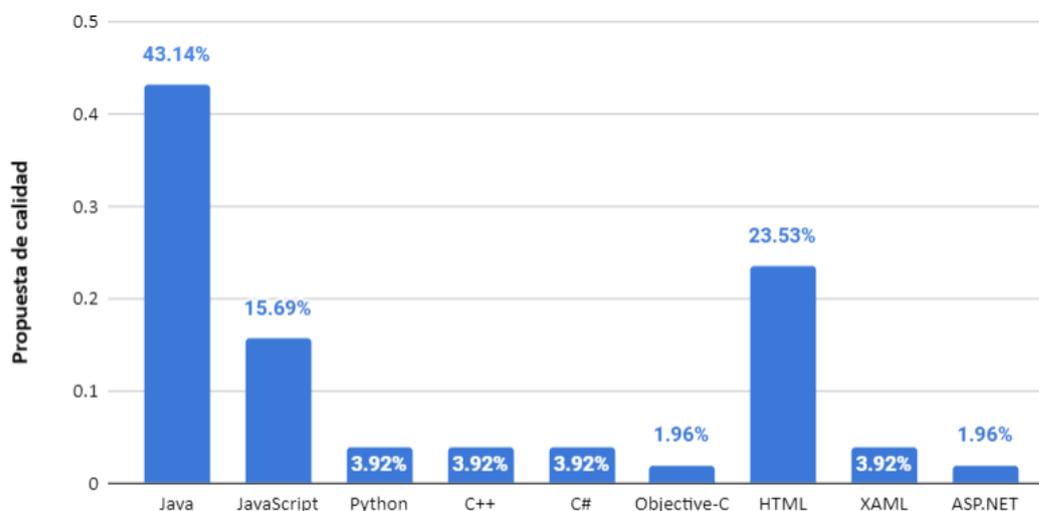
Lenguaje de programación	Estudios Relevantes	Frecuencia
Java	EP1, EP2, EP3, EP8, EP10, EP12, EP13, EP16, EP21, EP23, EP24, EP25, EP28, EP29, EP32, EP33, EP41, EP44, EP45, EP46, EP49, EP51	22
JavaScript	EP1, EP2, EP5, EP6, EP9, EP11, EP34, EP36	8
Python	EP6, EP22	2
C++	EP37, EP44	2
C#	EP 32, EP13	2

Objective-C	EP13	1
HTML	EP1, EP5, EP6, EP8, EP9, EP13, EP28, EP32, EP36, EP39, EP40, EP43	12
XAML	EP13, EP29	2
ASP.NET	EP5	1

Entre los estudios más relevantes sobre este criterio se tiene la propuesta de Mikado et al. (2010), quienes aplican un método para definir, evaluar y visualizar automáticamente los KPI de ciudades inteligentes. En particular, este enfoque proporciona varios artefactos para el modelado de ciudades inteligentes y KPIs, así como un motor de evaluación para el cálculo automático de KPIs y su visualización. En este caso, la generación de la vista gráfica se implementa en tres plantillas que generan código en HTML y Javascript para mostrar gráficos con los resultados de los KPIs. Por su parte, desarrollar aplicaciones para plataformas móviles resulta una tarea compleja debido a la diversidad de dispositivos móviles y sistemas operativos. La propuesta de Sabraoui et al. (2013) presenta un enfoque basado en un modelo MDA para generar una GUI de aplicación móvil utilizando plantillas Xpand para transformar el modelo (XMI/XML) a artefactos en Java.

Figura 11

Distribución porcentual del lenguaje de programación en el que se generan los artefactos de la interfaz de usuario



Lenguaje de programación en el que se generan los artefactos de la interfaz de usuario

EC8: Sistema operativo en el que se ejecutan los artefactos generados de la interfaz de usuario.

Asimismo, este criterio demuestra la preocupación de los investigadores por crear soluciones para múltiples plataformas. En este sentido, ordenados según su nivel de recurrencia, la revisión evidencia estudios que generan artefactos de interfaz de usuario para ejecutarse en diferentes sistemas operativos: 19 estudios dirigidos a Windows (37.25%), 13 estudios a Android (25.49%) y 7 estudios a iOS (13.73%). Por último, un pequeño grupo de estudios generan soluciones para Windows Phone (5.88%) y BlackBerry OS (1.96%) (ver Tabla 11 y Figura 12).

Tabla 11

Frecuencia del sistema operativo en el que se ejecutan los artefactos generados de la interfaz de usuario

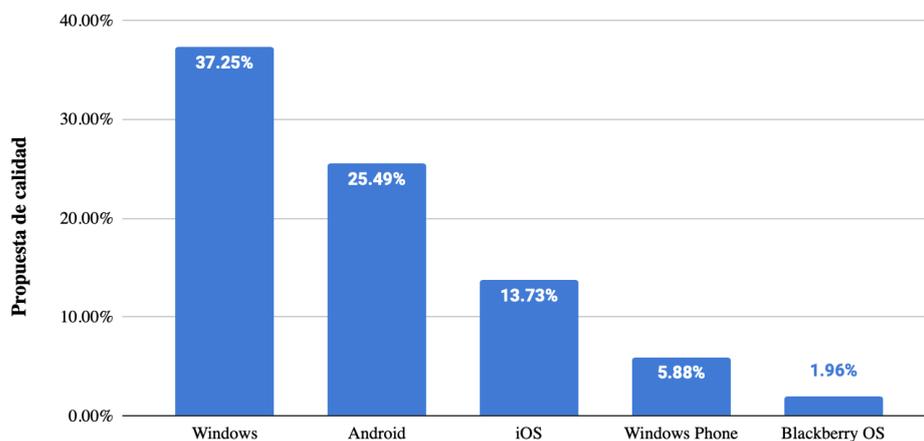
Sistema operativo	Estudios Relevantes	Frecuencia
Windows	EP4, EP8, EP11, EP12, EP22, EP29, EP31, EP32, EP35, EP36, EP37, EP38, EP41, EP42, EP44, EP45, EP46, EP50, EP51	19

Android	EP1, EP2, EP3, EP8, EP10, EP11, EP13, EP19, EP23, EP32, EP34, EP42, EP50	13
iOs	EP2, EP11, EP13, EP19, EP23, EP34, EP36	7
Windows Phone	EP13, EP23, EP34	3
BlackBerry OS	EP23	1

Uno de los artículos más importantes relacionados con este criterio es el de Brambilla et al. (2017), donde se propone un enfoque basado en modelos para diseñar la interfaz de usuario de sistemas IoT, mediante la definición de componentes y patrones de diseño. Este enfoque está orientado a sistemas multiplataforma de automatización del hogar, específicamente puede generar interfaces de usuario para iOS, Android y Windows. Por otro lado, Boterweck (2007) presenta un enfoque general para adaptar las interfaces gráficas de usuario ejecutadas en plataformas de escritorio, móvil o web, al contexto de accesibilidad del usuario. En lugar de crear procesos de adaptación separados para cada plataforma, se desarrolla un procedimiento de adaptación único basado en modelos que genera artefactos de UI concretos para las diferentes plataformas.

Figura 12

Distribución porcentual del sistema operativo en el que se ejecutan los artefactos generados de la interfaz de usuario



Sistema operativo en el que se ejecutan los artefactos generados de la interfaz de usuario

RQ2. ¿Cómo se aborda la investigación en los estudios relacionados con el uso de MDE para diseñar y construir las interfaces de usuario de software?

EC9: Tipo de validación. Un aspecto esencial de la revisión se relaciona con comprender el tipo de validación utilizado por los estudios seleccionados. Los resultados muestran que los trabajos que realizaron un estudio de caso representan el 31,37%, mientras que la prueba de concepto y el experimento: 25,49% y 15,69%, respectivamente. Finalmente, la encuesta fue utilizada en único estudio (1,96%). Por otro lado, los trabajos que no han utilizado ninguna evaluación representan el 29,41% (ver Tabla 12 y Figura 13). En resumen, los estudios que no incluyen una evaluación empírica y los que se limitan a utilizar una prueba de concepto representan más de la mitad de los estudios (54,90%), lo que constituye la principal limitación de esta área de investigación. Por ello, los investigadores deberían utilizar métodos de evaluación más rigurosos para determinar la validez de los resultados.

Tabla 12

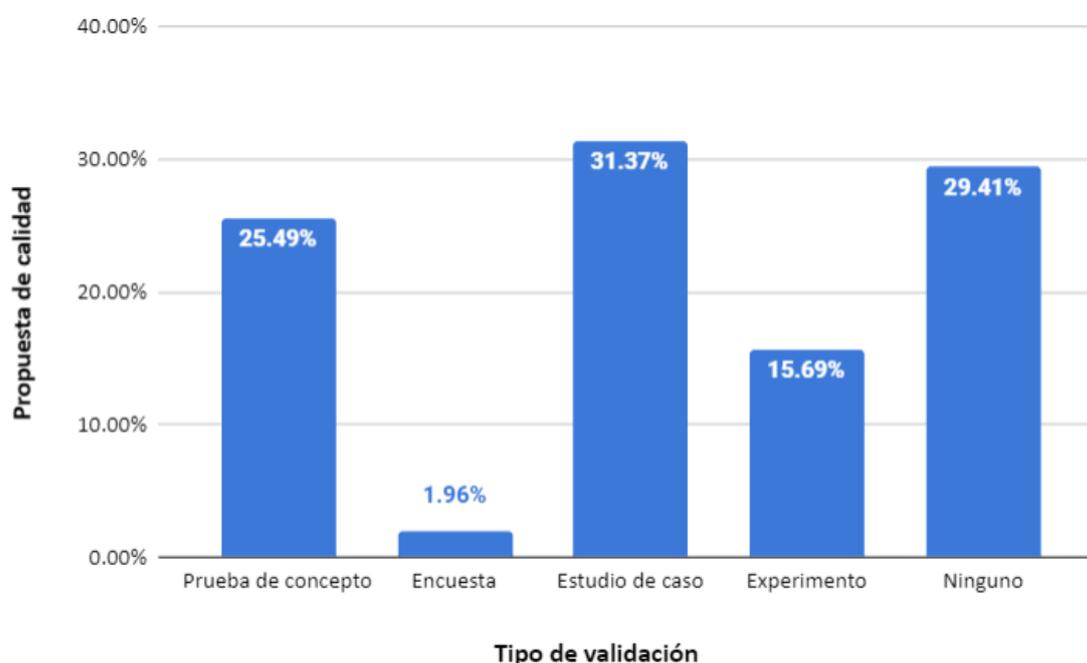
Frecuencia del tipo de validación

Tipo de validación	Estudios Relevantes	Frecuencia
Prueba de concepto	EP3, EP10, EP12, EP13, EP19, EP32, EP33, EP37, EP39, EP40, EP45, EP46, EP47	13

Encuesta	EP1	1
Estudio de caso	EP1, EP2, EP8, EP11, EP16, EP20, EP22, EP23, EP24, EP26, EP27, EP31, EP36, EP38, EP48, EP50	16
Experimento	EP1, EP5, EP8, EP21, EP28, EP34, EP41, EP49	8
Ninguno	EP4, EP6, EP7, EP9, EP14, EP15, EP17, EP25, EP29, EP30, EP35, EP42, EP43, EP44, EP51	15

Entre los trabajos de investigación a destacar en este apartado, en primer lugar, De Sanctis et al. (2022) utilizan una combinación de tipos de evaluación, consistente en un estudio de caso, una familia de experimentos y una encuesta, con el fin de probar la escalabilidad, usabilidad y comprensibilidad de un marco de modelado de evaluación de los KPI de las ciudades inteligentes. En otro estudio, Brambilla et al. (2017) implementan tres estudios de casos industriales reales para demostrar la viabilidad de su enfoque basado en modelos para el diseño del front-end (interfaz de usuario) de los sistemas de IoT. Por último, el estudio de caso de Molina et al. (2012) evaluó la percepción de los ingenieros de software de una empresa farmacéutica sobre la utilidad del método y las herramientas de diseño de interfaces de usuario basadas en modelos propuestas.

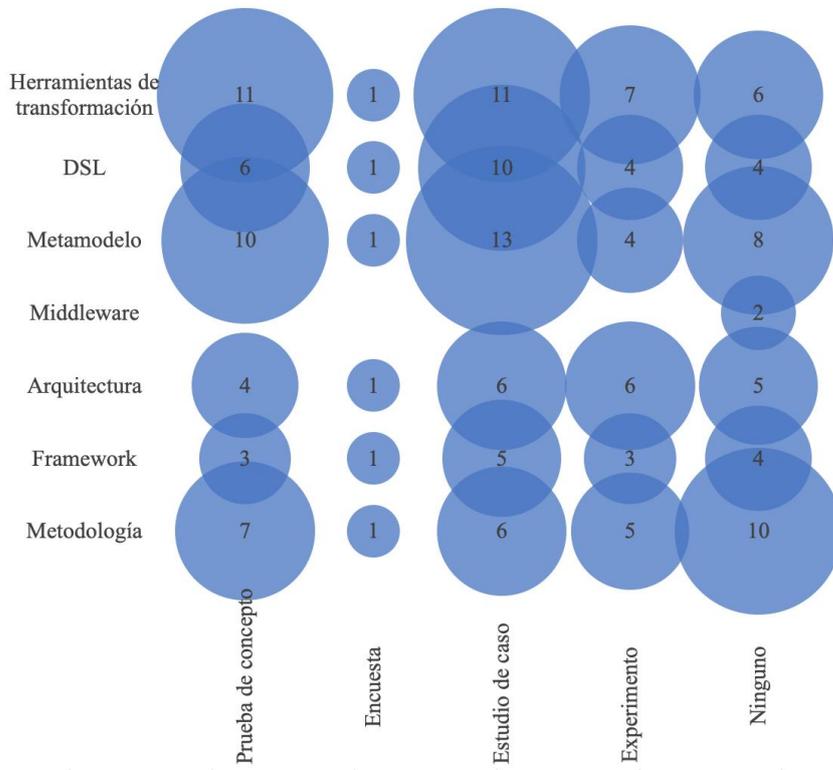
Figura 13
Distribución porcentual del tipo de validación



La Figura 14 muestra la distribución de las propuestas de MDE mediante una combinación de dos criterios de extracción: i) tipo de solución desarrollada y ii) tipo de verificación. En la gráfica se puede observar que las validaciones mediante estudios de caso, pruebas de concepto y experimentos se han orientado en su mayoría a metamodelos, transformaciones y DSLs; y, en menor proporción a metodologías, arquitecturas y frameworks.

Figura 14

Tipo de validación realizada según el tipo de solución MDE propuesto



EC10: Tipo de estudio. En términos generales se comprobó que el 72,55% son estudios nuevos que no presentan una fase previa, mientras que las propuestas restantes (27,45%) corresponden a trabajos que son una extensión de investigaciones anteriores. Estos resultados evidencian la necesidad de aprovechar las contribuciones existentes en la literatura para alcanzar mejores resultados (ver Tabla 13 y Figura 15).

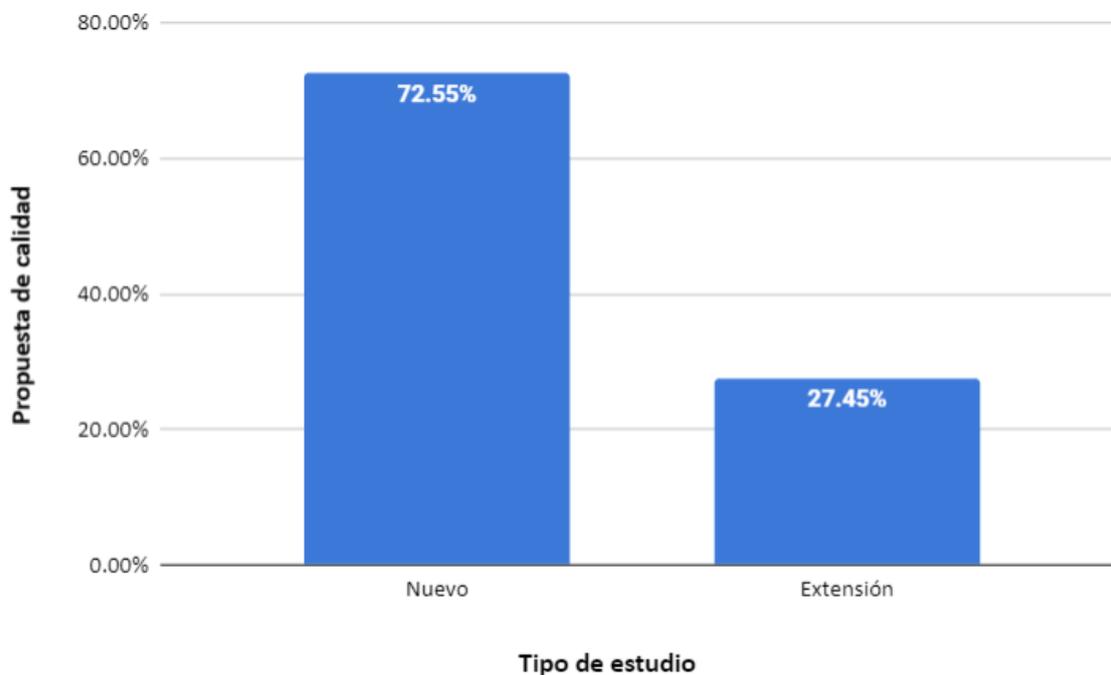
Tabla 13

Frecuencia del tipo de estudio

Tipo de estudio	Estudios Relevantes	Frecuencia
Nuevo	EP2, EP3, EP6, EP8, EP10, EP11, EP12, EP13, EP14, EP15, EP16, EP17, EP19, EP20, EP21, EP22, EP24, EP25, EP26, EP27, EP28, EP29, EP33, EP34, EP36, EP37, EP38, EP40, EP41, EP42, EP44, EP45, EP46, EP47, EP48, EP49, EP50, EP51	38
Extensión	EP1, EP4, EP5, EP7, EP9, EP18, EP23, EP30, EP31, EP32, EP33, EP35, EP39, EP43	14

Figura 15

Distribución porcentual del tipo de estudio



EC11: Alcance del enfoque. De las 51 propuestas de MDE, 39 estudios muestran que el 76,47% representan los trabajos de investigación desarrollados únicamente en un entorno académico, mientras que solo el 23,53% se llevó a cabo con la participación de la industria. En este sentido, resulta recomendable la incorporación de la industria, como un actor fundamental en proyectos futuros, a fin de validar las propuestas en entorno reales (ver Tabla 14 y Figura 16).

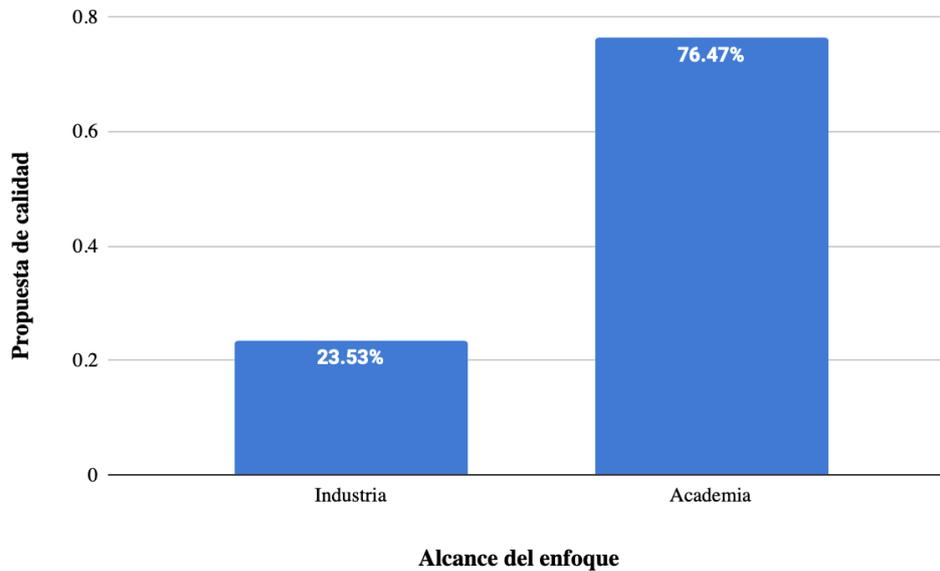
Tabla 14

Frecuencia del alcance del enfoque

Alcance del enfoque	Estudios Relevantes	Frecuencia
Industria	EP9, EP11, EP12, EP18, EP20, EP23, EP29, EP30, EP33, EP36, EP45, EP46	12
Academia	EP1, EP2, EP3, EP4, EP5, EP6, EP7, EP8, EP10, EP13, EP14, EP15, EP16, EP17, EP19, EP21, EP22, EP24, EP25, EP26, EP27, EP28, EP31, EP32, EP34, EP35, EP37, EP38, EP39, EP40, EP41, EP42, EP43, EP44, EP47, EP48, EP49, EP50, EP51	39

Figura 16

Distribución porcentual del alcance del enfoque



5. Conclusión y trabajo futuro

La interfaz de usuario (front-end) es uno de los componentes fundamentales del software para garantizar una experiencia de usuario satisfactoria. Sin embargo, debido a la complejidad e incertidumbre intrínsecas de los sistemas modernos y a las exigencias de las múltiples partes interesadas, se requieren métodos emergentes, como MDE, para mejorar la calidad del desarrollo y el funcionamiento de las interfaces de usuario del software. Este trabajo de titulación cubre la necesidad de una revisión sistemática de la literatura actualizada sobre la Ingeniería Dirigida por Modelos aplicada a las interfaces de usuario que ofrezca una visión general del estado actual y una orientación para investigaciones futuras. A continuación, se detallan las principales conclusiones y resultados de las preguntas de investigación:

- RQ1. Entre los puntos fuertes de este campo de investigación está la existencia de un importante grupo de propuestas de MDE multiplataforma que combinan metodologías, arquitecturas, metamodelos, DSLs y motores de transformación. Asimismo, en los últimos años han aparecido estudios centrados en la construcción de cuadros de mando. Sin embargo, los estudios se han concentrado fundamentalmente en el tiempo de diseño, por lo que un reto importante de la investigación es profundizar en el uso de modelos en tiempo de ejecución para promover la adaptación autónoma de la UI. Una forma de hacer operativa esta necesidad sería proponer un middleware basado en modelos en tiempo de ejecución. Asimismo, es necesario aprovechar los diversos metamodelos propuestos para complementarlos con una herramienta visual DSL de alto nivel.
- RQ2. La principal preocupación es el gran número de estudios que no han sido evaluados empíricamente o han aplicado una prueba de concepto. De ahí que se deban promover nuevos estudios que contemplen tipos de evaluación más rigurosos para validar las propuestas existentes, buscando una mayor participación de la industria.

En futuros trabajos, se propone construir un middleware basado en modelos de tiempo de ejecución para soportar la adaptación de UIs en múltiples plataformas, considerando las fortalezas de los estudios analizados.

6. Referencias

- Ahmed, S. & Ashraf, G. (2007). Model-Based User Interface Engineering with Design Patterns. *Journal of Systems and Software*, 80(8), 1408-1422.
- Akiki, P. A., Bandara, A. K. & Yu, Y. (2013). Cedar Studio: An IDE Supporting Adaptive Model-Driven User Interfaces for Enterprise Applications. In: *5th Symp. on Engineering Interactive Computing Systems*, pp. 139-144.
- Ali, A., Rashid, M., Azam, F., Rasheed, Y. & Anwar, M. W. (2021). A Model-Driven Framework for Android Supporting Cross-Platform GUI Development. In: *4th National Computing Colleges Conference*, pp. 1-6.
- Aquino, N., Vanderdonck, J. & Pastor, O. (2010). Transformation Templates: Adding Flexibility to Model-Driven Engineering of User Interfaces. In: *Symp. on Applied Computing*, pp. 1195-1202.
- Ben, L. & Mahfoudhi, A. (2013). Usability Driven Model Transformation. In: *6th Inter. Conf. on Human System Interactions*, pp. 110-116.
- Bencomo, N., Götz, S. & Song, H. (2019). Models@run.time: A Guided Tour of the State of the Art and Research Challenges. *Softw. Syst. Model*, 18(5), 3049-3082.
- Bendaly, Y., Zouhaier, L. & Ben, L. (2019). Model Driven Approach for Adapting User Interfaces to the Context of Accessibility: Case of Visually Impaired Users. *Journal on Multimodal User Interfaces*, 13(4), 293-320.
- Bezerra, J. D. H. & De Souza, C. T. (2019). A Model-Based Approach to Generate Reactive and Customizable User Interfaces for the Web of Things. In: *25th Brazilian Symp. on Multimedia and the Web*, pp. 57-60.
- Botterweck, G. (2006). A Model-Driven Approach to the Engineering of Multiple User Interfaces. In: *Int. Conf. on Model Driven Engineering Languages and Systems*, pp. 106-115.
- Bouchelligua, W., Mahfoudhi, A., Benammar, L., Rebai, S. & Abed, M. (2010). An MDE Approach for User Interface Adaptation to the Context of Use. In: *Inter. Conf. on Human-Centred Software Engineering*, pp. 62-78.
- Brambilla, M., Umuhzoza, E. & Acerbis, R. (2017). Model-Driven Development of User Interfaces for IoT Systems via Domain-Specific Components and Patterns. *Journal of Internet Services and Applications*, 8(1), 1-21.
- Bucchiarone, A., Cabot, J., Paige, R. F. & Pierantonio, A. (2020). Grand Challenges in Model-Driven Engineering: An Analysis of the State of the Research. *Softw. Syst. Model*, 19(1), 5-13.
- Coutaz, L., Balme, L., Alvaro, X., Calvary, G., Demeure, A. & Sottet, J. S. (2007). An MDE-SOA Approach to Support Plastic User Interfaces in Ambient Spaces. In: *Inter. Conf. on Universal Access in Human-Computer Interaction*, pp. 63-72.
- Criado, J., Iribarne, L., Padilla, N., Troya, J. & Vallecillo, A. (2011). Adapting Component-Based User Interfaces at Runtime Using Observers. In: *16th Jornadas de Ingeniería del Software y Bases de Datos*, pp. 707-712.
- Criado, J., Iribarne, L., Padilla, N., Troya, J. & Vallecillo, A. (2012). An MDE Approach for Runtime Monitoring and Adapting Component-Based Systems: Application to WIMP User Interface Architectures. In: *38th Euromicro Conf. on Software Engineering and Advanced Applications*, pp. 150-157.
- Criado, J., Rodríguez-Gracia, D., Iribarne, L. & Padilla, N. (2014). Toward the adaptation of component-based architectures by model transformation: behind smart user interfaces. *Software: Practice and Experience*, 45(12), 1677-1718.
- Criado, J., Vicente-Chicote, C., Iribarne, L. & Padilla, N. (2010). A Model-Driven Approach to Graphical User Interface Runtime Adaptation. In: *Int. Conf. on Model Driven Engineering Languages and Systems*, pp. 49-59.
- Da Costa, S. L., Graciano, V. V. & De Oliveira, J. L. (2014). A User Interface Stereotype to Build Web Portals. In: *9th Latin American Web Congress*, pp. 10-18.
- Da Cruz, A. M. R. & Faria, J. P. (2010). A Metamodel-Based Approach for Automatic User Interface Generation. In: *Int. Conf. on Model Driven Engineering Languages and Systems*, pp. 256-270.

- De Morais, C. M., Kelner, J., Sadok, D. & Lynn, T. (2018). SiMoNa: A proof-of-concept domain specific modeling language for IoT infographics. In: *Symp. on Visual Languages and Human-Centric Computing*, pp. 199-203.
- De Sanctis, M., Iovino, L., Rossi, M. T. & Wimmer, M. (2022). MIKADO: A Smart City KPIs Assessment Modeling Framework. *Softw. Syst. Model*, 21(1), 281-309.
- Diep, C. K., Tran, Q. N. & Tran, M. T. (2013). Online Model-driven IDE to Design GUIs for Cross-Platform Mobile Applications. In: *4th Symp. on Inf. and Commun. Technol.*, pp. 294-300.
- Erazo-Garzon, L., Erraez, J., Cedillo, P. & Illescas-Peña L. (2019). Quality Assessment Approaches for Ambient Assisted Living Systems: A systematic Review. In: *Int. Conf. on Applied Technologies*, 1193, pp. 421-439.
- Erazo-Garzón, L., Priscilla, C., Gustavo, R. & Moyano, J. (2022). A Domain-Specific Language for Modeling IoT System Architectures That Support Monitoring. *IEEE Access*, 10, 61639-61665.
- France, R. & Rumpe, B. (2007). Model-Driven Development of Complex Software: A Research Roadmap. In: *Future of Software Engineering*, pp. 37-54.
- Frey, A. G., Calvary, G. & Dupuy-Chessa, S. (2010). Xplain: An Editor for Building Self-Explanatory User Interfaces by Model-Driven Engineering. In: *2nd Symp. on Engineering Interactive Computing Systems*, pp. 41-46.
- Frey, A. G., Céret, E., Dupuy-Chessa, S. & Calvary, G. (2011). QUIM111ERA: A Quality Metamodel to Improve Design Rationale. In: *3rd Symp. on Engineering Interactive Computing Systems*, pp. 265-270.
- Frey, A. G., Sottet, J. S. & Vagner, A. (2014). Towards a Multi-Stakeholder User Interface Engineering Approach with Adaptive Modelling Environments. In: *Symp. on Engineering Interactive Computing Systems*, pp. 33-38.
- García, A., Calvary, G., Dupuy-Chessa, S. & Mandran, N. (2013). Model-Based Self-explanatory UIs for Free, but Are They Valuable? In: *IFIP Conf. on Human-Computer Interaction*, pp. 144-161.
- Gibbs, I., Dascalu, S. & Harris, F. C. (2015). A Separation-Based UI Architecture with a DSL for Role Specialization. *Journal of Systems and Software*, 101, 69-85.
- Gottardi, T. & Vaccare, R. T. (2018). Understanding the Successes and Challenges of Model-Driven Software Engineering-a Comprehensive Systematic Mapping. In: *XLIV Latin American Computer Conference (CLEI)*, pp. 129-13.
- Ian, R. & Favre, J. M. (2005). *Visualization in the Context of Model Driven Engineering*. MDDAUI@MoDELS.
- Iribarne, L., Padilla, N., Criado, J. & Vicente-Chicote, C. (2010). An Interaction Meta-Model for Cooperative Component-Based User Interfaces. In: *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, pp. 259-268.
- Kai, R. & Liu, X. (2020). IFML-Based Web Application Modeling. *Procedia Computer Science*, 166, 129-133.
- Kapłański, P., Seganti, A., Cieslinski, K., Chrabrowa, A. & Ługowska, I. (2017). Automated Reasoning Based User Interface. *Expert Systems with Applications*, 71, 125-137.
- Khaddam, I., Mezhoudi, N. & Vanderdonckt, J. (2015). Adapt-First: A MDE Transformation Approach for Supporting User Interface Adaptation. In: *2nd World Symposium on Web Applications and Networking*, pp. 1-9.
- Khan, M., Azam, F., Rashid, M., Samea, M., Anwar, M. W., Muzaffar, A. W. & Butt, W. H. (2022). A Retargetable Model-Driven Framework for the Development of Mobile User Interfaces. *Journal of Circuits, Systems and Computers*, 31(01), 2250018.
- Kitchenham, B. & Charters, S. (2007). *Guidelines for Performing Systematic Literature Reviews in Software Engineering Version 2.3*. University of Durham.
- Lachgar, M. & Abdali, A. (2014). Generating Android Graphical User Interfaces Using an MDA Approach. In: *3rd Inter. Colloquium in Information Science and Technology*, pp. 80-85.
- Lehmann, G., Blumendorf, M. & Albayrak, S. (2010). Development of Context-Adaptive Applications on the Basis of Runtime User Interface Models. In: *2nd Symp. on Engineering Interactive Computing Systems*, pp. 309-314.

- López-Jaquero, V., Vanderdonckt, J., Montero, F. & González, P. (2008). Towards an Extended Model of User Interface Adaptation: The Isatine Framework. In: *Int Conf. on Engineering for Human-Computer Interaction*, pp. 374-392.
- Miñón, R., Moreno, L., Martínez, P. & Abascal, J. (2014). An Approach to the Integration of Accessibility Requirements into a User Interface Development Method. *Science of Computer Programming*, 86, 58-73.
- Mitrović, N., Bobed, C. & Mena, E. (2017). Dynamic User Interface Architecture for Mobile Applications Based on Mobile Agents. In: *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, pp. 282-292.
- Molina, A. I., Giraldo, W. J. Gallardo J. & Redondo, M. A. (2012). CIAT-GUI: A MDE-Compliant Environment for Developing Graphical User Interfaces of Information Systems. *Adv. Eng. Softw.*, 52, 10-29.
- Ordóñez, K., Hilera, J. & Cueva, S. (2022). Model-Driven Development of Accessible Software: A Systematic Literature Review. *Universal Access in the Inf. Society*, 21, 295-324.
- Paterno, F., Santoro, C. & Spano, L. D. (2009). MARIA: A Universal, Declarative, Multiple Abstraction-Level Language for Service-Oriented Applications in Ubiquitous Environments. *ACM Transactions on Computer-Human Interaction*, 16(4), 1-30.
- Raneburger, D. (2010). Interactive Model Driven Graphical User Interface Generation. In: *2nd Symp. on Engineering Interactive Computing Systems*, pp. 321-324.
- Raneburger, D., Popp, R., Kavaldjian, S., Kaindl, H. & Falb, J. (2011). Optimized GUI Generation for Small Screens. In: H. Hussmann, G. Meixner & Zuehlke (eds.), *Model-Driven Development of Advanced User Interfaces*, pp. 107-122. Springer Link.
- Rojas, E., Bastidas, V. & Cabrera, C. (2020). Cities-Board: A Framework to Automate the Development of Smart Cities Dashboards. *Internet of Things Journal*, 7(10), 10128-10136.
- Roscher, D., Lehmann, G., Schwartz, V., Blumendorf, M. & Albayrak, S. (2011). Dynamic Distribution and Layouting of Model-Based User Interfaces. In: *Model-Driven Development of Advanced User Interfaces*, pp. 171-197.
- Roubi, S., Erramdani, M. & Mbarki, S. (2015). Generating Graphical User Interfaces Based on Model Driven Engineering. *Int. Rev. on Comput. and Softw.*, 10(5), 520-528.
- Sabraoui, A., Abouzahra, A., Afdel, K. & Machkour, M. (2019). MDD Approach for Mobile Applications Based on DSL. In: *Inter. Conf. of Computer Science and Renewable Energies*, pp. 1-6.
- Sabraoui, A., El Koutbi, M. & Khriiss, I. (2013). A MDA-Based Model-Driven Approach to Generate GUI for Mobile Applications. *Int. Rev. Comput. Softw.*, 8(3), 844-852.
- Schuler, A. & Franz, B. (2013). Rule-Based Generation of Mobile User Interfaces. In: *10th Inter. Conf. on Information Technology: New Generations*, pp. 267-272.
- Shamsujjoha, M., Grundy, J., Li, L., Khalajzadeh, H. & Lu, Q. (2021). Developing Mobile Applications Via Model Driven Development: A Systematic Literature Review. *Information and Software Technology*, 140, 106693.
- Shirogane, J., Mukaeda, H., Iwata, H. & Fukazawa, Y. (2010). GUI Prototype Generation Based on an MDA Process. In: *Inter. Conf. Information Systems*, pp. 478-482.
- Sottet, J. S., Vagner, A. & Frey, A. G. (2015). Model Transformation Configuration and Variability Management for User Interface Design. In: *Inter. Conf. on Model-Driven Engineering and Software Development*, pp. 390-404.
- Erazo, L., Suquisupa, S., Bermeo, A. & Cedillo, P. (2022). *Model-Driven Engineering Applied to User Interfaces. A Systematic Literature Review* [Manuscrito enviado para publicación]. LIDI, Universidad del Azuay.
- Thomas, M., Mihaela, I., Andrianjaka, R. M., Germain, D. W. & Sorin, I. (2021). Metamodel Based Approach to Generate User Interface Mockup from UML Class Diagram. *Procedia Computer Science*, 184, 779-784.
- Tufail, H., Azam, F., Anwar, M. W. & Qasim, I. (2019). Model-Driven Development of Mobile Applications: A Systematic Literature Review. In: *9th Annual Information Technology, Electronics and Mobile Communication Conference*, pp. 1165-1171.

- Vázquez-Ingelmo, A., García-Peñalvo, F. J., Therón, R. & Conde, M. Á. (2020). Representing Data Visualization Goals and Tasks Through Meta-Modeling to Tailor Information Dashboards. *Applied. Sciences*, 10(7), 2306.
- Zeferino, N. V. & Vilain, P. (2014). A model-driven approach for generating interfaces from user interaction diagrams. In: *16th Int. Conf. on Information Integration and Web-based Applications & Services*, pp. 474-478.
- Zhang, L., Qu, Q. X., Chao, W. Y. & Duffy, V. G. (2020). Investigating the Combination of Adaptive UIs and Adaptable UIs for Improving Usability and User Performance of Complex UIs. *Int. Journal of Human-Computer Interaction*, 36(1), 82-94.
- Zouhaier, L., Bendaly, Y. & Ben Ayed, L. J. (2015). A Model Driven Approach for Improving the Generation of Accessible User Interfaces. In: *10th Inter. Joint Conf. on Software Technologies*, pp. 1-6. Science and Technology Publications, Lda.
- Zouhaier, L., Hlaoui, Y. B. & Ben, L. J. (2014). Generating Accessible Multimodal User Interfaces Using MDA-Based Adaptation Approach. In: *38th Annual Computer Software and Applications Conf.*, pp. 535-540.

7. Anexos

1. Listado de estudios primarios relevantes obtenidos de la revisión

#	Título del estudio	Año de publicación	Autor	Referencia	Tipo de publicación	Biblioteca digital
EP1	MIKADO: a smart city KPIs assessment modeling framework	2022	De Sanctis, M., Iovino, L., Rossi, M. T., Wimmer, M.	51	Revista	Otros
EP2	A Retargetable Model-Driven Framework for the Development of Mobile User Interfaces	2022	Khan, M., Azam, F., Rashid, M., Samea, F., Anwar, M. W., Muzaffar, A. W., Butt, W. H.	6	Revista	Springer Link
EP3	A Model-Driven Framework for Android Supporting Cross-Platform GUI Development	2021	Ali, A., Rashid, M., Azam, F., Rasheed, Y., Anwar, M. W.	21	Conferencia	IEEE Explore
EP4	Metamodel based approach to generate user interface mockup from UML class diagram	2021	Thomas, M., Mihaela, I., Andrianjaka, R. M., Germain, D. W., Sorin, I.	22	Revista	IEEE Explore
EP5	Cities-Board: A Framework to Automate the Development of Smart Cities Dashboards	2020	Rojas, E., Bastidas, V., Cabrera, C.	23	Revista	IEEE Explore
EP6	Representing data visualization goals and tasks through meta-modeling to tailor information dashboards	2020	Vázquez-Ingelmo, A., García-Peñalvo, F. J., Therón, R., Conde, M. A.	24	Revista	Otros
EP7	IFML-based web application modeling	2020	Cao, R. K., Liu, X.	25	Revista	Springer Link
EP8	Model driven approach for adapting user interfaces to the context of accessibility: case of visually impaired users	2019	Bendaly Hlaoui, Y., Zouhaier, L., Ben Ayed, L.	4	Revista	Otros
EP9	A model-based approach to generate reactive and customizable user interfaces for the web of things	2019	Bezerra, J. D. H., De Souza, C. T.	52	Conferencia	IEEE Explore
EP10	MDD Approach for Mobile Applications Based on DSL	2019	Sabraoui, A., Abouzahra, A., Afdel, K., Machkour, M.	26	Conferencia	Springer Link
EP11	Model-driven development of user interfaces for IoT systems via domain-specific components and patterns	2017	Brambilla, M., Umuhoza, E., Acerbis, R.	27	Revista	Springer

EP12	Generating graphical user interfaces based on model driven engineering	2015	Roubi, S., Erramdani, M., Mbarki, S.	28	Revista	ResearchGate
EP13	Generating Android graphical user interfaces using an MDA approach	2015	Lachgar, M., Abdali, A.	29	Conferencia	IEEE explore
EP14	A model driven approach for improving the generation of accessible user interfaces	2015	Zouhaier, L., Hlaoui, Y. B., Ben Ayed, L. J.	30	Conferencia	IEEE explore
EP15	A user interface stereotype to build Web portals	2014	Costa, S. L. D., Neto, V. V. G., Oliveira, J. L. D.	31	Conferencia	IEEE explore
EP16	A model-driven approach for generating interfaces from user interaction diagrams	2014	Zeferino, N. V., Vilain, P.	1	Conferencia	ACM Digital Library
EP17	Generating accessible multimodal user interfaces using MDA-based adaptation approach	2014	Zouhaier, L., Hlaoui, Y. B., Ayed, L. J. B.	32	Conferencia	IEEE explore
EP18	Towards a multi-stakeholder user interface engineering approach with adaptive modelling environments	2014	Frey, A. G., Sottet, J. S., Vagner, A.	8	Conferencia	Association for Computing Machinery
EP19	Rule-based generation of mobile user interfaces	2013	Schuler, A., Franz, B.	33	Conferencia	IEEE explore
EP20	Usability driven model transformation	2013	Ben Ammar, L., Mahfoudhi, A.	34	Conferencia	IEEE explore
EP21	Model-based self-explanatory UIs for free, but are they valuable? (pag. 171)	2013	García Frey, A., Calvary, G., Dupuy-Chessa, S., Mandran, N.	53	Conferencia	Springer
EP22	Cedar studio: An IDE supporting adaptive model-driven user interfaces for enterprise applications	2013	Akiki, P.A., Bandara, A. K., Yu, Y.	54	Conferencia	ACM Digital Library
EP23	A MDA-based model-driven approach to generate GUI for mobile applications (pag.168)	2013	Sabraoui, A., El Koutbi, M., Khriiss, I.	35	Revista	ResearchGate
EP24	CIAT-GUI: A MDE-compliant environment for developing Graphical User Interfaces of information systems	2012	Molina, A. I., Giraldo, W. J., Gallardo, J., Redondo, M. A., Ortega, M., García, G.	10	Revista	ScienceDirect
EP25	Adapting component-based user interfaces at runtime using observers	2011	Criado, J., Iribarne, L., Padilla, N., Troya, J., Vallecillo, A.	55	Conferencia	HAL

EP26	QUIMERA: A quality metamodel to improve design rationale	2011	Frey, A. G., Céret, E., Dupuy-Chessa, S., Calvary, G.	36	Conferencia	ACM Digital Library
EP27	An interaction meta-model for cooperative component-based user interfaces (pag.286)	2010	Iribarne, L., Padilla, N., Criado, J., Vicente-Chicote, C.	56	Conferencia	Springer
EP28	Gui prototype generation based on an MDA process	2010	Shirogane, J., Mukaeda, H., Iwata, H., Fukazawa, Y.	37	Conferencia	Waseda
EP29	A metamodel-based approach for automatic user interface generation	2010	Da Cruz, A. M. R., Faria, J. P.	38	Conferencia	Springer
EP30	Interactive model driven graphical user interface generation	2010	Raneburger, D.	39	Conferencia	ACM Digital Library
EP31	Model-based user interface engineering with design patterns	2007	Kaindl, H.	40	Revista	ScienceDirect
EP32	A model-driven approach to the engineering of multiple user interfaces	2007	Botterweck, G.	11	Conferencia	Springer
EP33	Visualization in the context of model driven engineering	2005	Ian Bull, R., Favre, J. M.	41	Conferencia	ceur-ws.org
EP34	Online Model-driven IDE to Design GUIs For Cross-platform Mobile Applications	2013	Diep, C. K., Tran, Q. N., Tran, M. T.	42	Conferencia	ACM Digital Library
EP35	Optimized GUI Generation for Small Screens	2011	Raneburger, D., Popp, R., Kavaldijian, S., Kaindl, H., Falb, J.	43	Capítulo de libro	Springer
EP36	MARIA: A Universal, Declarative, Multiple Abstraction-Level Language for Service-Oriented Applications in Ubiquitous Environments	2009	Paterno, F., Santoro, C., Spano, L. D.	57	Revista	ACM Digital Library
EP37	Xplain: an Editor for building Self-Explanatory User Interfaces by Model-Driven Engineering	2010	Frey, A. G., Calvary, G., Dupuy-Chessa, S.	44	Conferencia	ACM Digital Library
EP38	An MDE Approach for User Interface Adaptation to the Context of Use	2010	Bouchelligua, W.	45	Conferencia	Springer

EP39	Model Transformation Configuration and Variability Management for User Interface Design	2015	Sottet, J. S., Vagner, A., Frey, A. G.	46	Conferencia	Springer
EP40	An approach to the integration of accessibility requirements into a user interface development method	2013	Miñón, R., Moreno, L., Martínez, P., Abascal, J.	47	Revista	ScienceDirect
EP41	A separation-based UI architecture with a DSL for role specialization	2015	Gibbs, I., Dascalu, S., Harris, F. C.	48	Revista	ScienceDirect
EP42	Transformation Templates: Adding Flexibility to Model-Driven Engineering of User Interfaces	2010	Aquino, N., Vanderdonckt, J., Pastor, O.	49	Conferencia	ACM Digital Library
EP43	Automated reasoning based user interface	2017	Kapłański, P., Seganti, A., Cieslinski, K., Chrabrowa, A., Ługowska, I.	58	Revista	ScienceDirect
EP44	An MDE-SOA Approach to Support Plastic User Interfaces in Ambient Spaces	2007	Coutaz, L., Balme, L., Alvaro, X., Calvary, G., Demeure, A., Sottet, J. S.	59	Conferencia	Springer
EP45	Adapt-First: a MDE transformation approach for supporting User Interface Adaptation	2015	Khaddam, I., Mezhoudi, N., Vanderdonckt, J.	60	Conferencia	IEEE explore
EP46	An MDE approach for Runtime Monitoring and Adapting Component-based Systems: Application to WIMP User Interface Architectures	2012	Criado, J., Iribarne, L., Padilla, N., Troya, J., Vallecillo, A.	61	Conferencia	IEEE explore
EP47	A model driven approach graphical user interface runtime adaptation	2010	Criado, J., Vicente-Chicote, C., Iribarne, L., Padilla, N.	12	Conferencia	repo.sitorio.upct.es
EP48	SiMoNa: A Proof-of-concept Domain-Specific Modeling Language for IoT Infographics	2018	De Morais, C. M., Kelner, J., Sadok, D., Lynn, T.	50	Conferencia	Hawaii International Conference on System Sciences
EP49	Toward the adaptation of component-based architectures by model transformation: behind smart user interfaces	2014	Criado, J., Rodríguez-Gracia, D., Iribarne, L., Padilla, N.	62	Revista	Wol
EP50	Dynamic Distribution and Layouting of Model-Based User Interfaces in Smart Environments	2011	Roscher, D., Lehmann, G., Schwartze, V., Blumendorf, M., Albayrak, S.	63	Revista	Springer
EP51	Development of Context-Adaptive Applications on the Basis of Runtime User Interface Models	2010	Lehmann, G., Blumendorf, M., Albayrak, S.	64	Conferencia	ACM Digital Library

2. Resumen de resultados de los criterios de extracción

<i>Criterio de extracción</i>		<i>Posibles respuestas</i>	<i># Propuestas de MDE</i>	<i>% Porcentaje</i>
Tipo de publicación:		Revista	17	33.33%
		Conferencia	32	62.75%
		Capítulo libro	2	3.92%
		Otro	0	0.00%
Tipo de búsqueda:		Búsqueda automática	1708	99.47%
		Búsqueda manual	4	0.23%
		Estrategia de bola de nieve	5	0.59%
Biblioteca digital:		IEEE Xplore Digital Library	329	19.26%
		ACM Digital Library	176	10.30%
		Springer Link	621	36.36%
		Science Direct	159	9.31%
		Otro	423	24.77%
RQ1. ¿Cómo se utiliza MDE para construir y mantener las interfaces de usuario de software?				
EC1.	Tipo de paradigma	Desarrollo dirigido por modelos	36	70.59%
		Modelos en tiempo de ejecución	15	29.41%
EC2.	Tipo de plataforma	Aplicación Web	23	45.10%
		Aplicación móvil	17	33.33%
		Aplicación de escritorio	20	39.22%
		No especifica	6	11.76%
EC3.	Tipo de solución Desarrollada	Metodología	28	54.90%
		Framework	14	27.45%
		Arquitectura	20	39.22%
		Middleware	2	3.92%
		Metamodelo	34	66.67%
		DSL	23	45.10%
		Transformaciones	34	66.67%

EC4.	Lenguaje de modelado y serialización utilizado	MOF	10	19.61%
		ECORE	13	25.49%
		UML	13	25.49%
		IFML	4	7.84%
		OCL	2	3.92%
		XML	19	37.25%
		KERMETA	2	3.92%
		GOPRR	2	3.92%
		OWL	2	3.92%
		JSON	1	1.96%
EC5.	Herramientas de desarrollo utilizadas para construir el DSL	Sirius	2	3.92%
		EFM	8	15.69%
		MetaEdit+	2	3.92%
		Xtext	2	3.92%
		Papyrus	1	1.96%
EC6.	Herramientas de desarrollo utilizadas para apoyar las transformaciones.	ATL	9	17.65%
		ACCELEO	4	7.84%
		XTEND	2	3.92%
		XPAND	1	1.96%
		M2T	3	5.88%
		MofScript	1	1.96%
EC7.	Lenguaje de programación en el que se generan los artefactos de la interfaz de usuario.	Java	22	43.14%
		JavaScript	8	15.69%
		Python	2	3.92%
		C++	2	3.92%
		C#	2	3.92%
		ASP.NET	1	1.96%
		Objective-C	1	1.96%
		Html	12	23.53%
		XAML	2	3.92%
EC8.	Sistema operativo en el que se ejecutan los	Windows	19	37.25%
		Android	13	25.49%
		iOS	7	13.73%

	artefactos generados de la interfaz de usuario.	Windows Phone	3	5.88%
		BlackBerry OS	1	1.96%
RQ2. ¿Cómo se aborda la investigación en los estudios relacionados con el uso de MDE para construir y mantener las interfaces de usuario de software?				
EC9.	Tipo de validación	Prueba de concepto	13	25.49%
		Encuesta	1	1.96%
		Caso de estudio	16	31.37%
		Experimento	8	15.69%
		Ninguno	15	29.41%
EC10.	Tipo de estudio	Nuevo	37	72.55%
		Extensión	14	27.45%
EC11.	Alcance del enfoque	Industria	39	76.47%
		Academia	12	23.53%