

TRANSFERENCIA DE DISEÑO DE INTERFACES DE USUARIO A CÓDIGO: REVISIÓN SISTEMÁTICA

Jonathan Stalin Delgado Guerrero¹, Leandro Antonelli², Diego Firmenich³

¹Facultad de Informática, UNLP, calle 50 esq 120, La Plata, Bs As, Argentina; SENESCYT, Ecuador

²Lifia, Facultad de Informática, UNLP, calle 50 esq 120, La Plata, Bs As, Argentina

²CAETI - Facultad de Tecnología Informática - Universidad Abierta Interamericana

³Departamento de Informática, Facultad de Ingeniería, UNPSJB, Chubut, Argentina.

¹e-mail: jonathan.delgado@info.unlp.edu.ar, ²e-mail: lanto@lifia.info.unlp.edu.ar,

³e-mail: dafirmenich@ing.unp.edu.ar

RESUMEN

La transferencia de diseño de interfaces de usuario (UI) a código es esencial en el desarrollo web actual. Existen diversos métodos, técnicas y herramientas disponibles para facilitar este proceso, como herramientas de prototipado, frameworks de diseño responsivo y generadores de código. Este estudio realiza una revisión sistemática de la literatura para identificar y analizar los métodos utilizados en la transferencia de diseño de interfaces de usuario web a código. Este aspecto es relevante para los desarrolladores al ser un potencial insumo que se puede usar desde el comienzo del desarrollo con principal impacto en el ahorro de tiempo. Se busca responder a la pregunta clave: ¿Qué métodos, técnicas y herramientas de diseño de interfaces de usuario web permiten la transferencia al código? Se identifican dos categorías principales: a partir de mockups, utilizando herramientas como Figma o Sketch, y a partir de la detección de elementos con algoritmos y redes neuronales. Se discuten los resultados y posibles beneficios de este proceso para futuras investigaciones.

PALABRAS CLAVES: diseño de interfaz de usuario, transferencia de diseño, diseño a código, métodos transferencia de diseño a código.

TRANSFER OF USER INTERFACES DESIGN TO CODE: A SYSTEMATIC REVIEW

ABSTRACT

Transferring user interface (UI) design to code is essential in today's web development. There are various methods, techniques and tools available to facilitate this process, such as prototyping tools, responsive design frameworks and code generators. This study conducts a systematic literature review to identify and analyze the methods used in transferring web user interface design to code. This aspect is relevant for developers as it is a potential input that can be used from the beginning of development with the main impact on time savings. It seeks to answer the key question: What web user interface design methods, techniques and tools allow the transfer to code? Two main categories are identified: from mockups, using tools such as Figma or Sketch, and from the detection of elements with algorithms and neural networks. The results and possible benefits of this process for future research are discussed.

KEY WORDS: user interface design, design transfer, design to code, design to code transfer methods.

1. INTRODUCCIÓN

El desarrollo de software web está estrechamente relacionado con el diseño de interfaces de usuario, permitiendo representar objetos del mundo real en computadoras conectadas en red [1]. Nuevas tecnologías como Ajax, JavaScript y sus frameworks, junto con la evolución de módulos de CSS, han potenciado las mejoras visuales en el desarrollo web, aumentando su popularidad [2] [3] [4]. La constante evolución de los estándares web ha generado enfoques que se centran en el diseño de la interfaz de usuario, brindando oportunidades para investigar y mejorar la usabilidad y experiencia del usuario. Esta

evolución ha transformado el desarrollo de software web y la forma en que interactuamos con la tecnología [5] [6]. Una herramienta usada para representar un diseño de interfaz de usuario lo constituyen los prototipos. Estas son herramientas empleadas para materializar y evaluar diseños de interfaces de usuario (UI) [7].

Algunos tipos comunes de prototipos de UI corresponden al nivel de detalle que ofrecen. Así tenemos prototipos de baja y alta fidelidad. Los primeros son representaciones de UI simples y abstractas, por lo general realizadas a mano o con la ayuda de herramientas de dibujo relativamente simple creadas desde cero. Su propósito principal es capturar la esencia conceptual del modelo y recopilar la retroalimentación temprana sobre su facilidad de uso [8].

Los prototipos de alta fidelidad son similares a la interfaz final, con detalle visual y funcionalidad interactiva. Se usan para validar el diseño y obtener retroalimentación precisa del usuario. Las herramientas permiten exportar a HTML y CSS [9]. Esta función permite tener un recurso inicial que permita transferir el diseño prototipado hacia código que podría ser usado en una etapa temprana de desarrollo de software. Tener un prototipo de alta fidelidad al inicio del desarrollo de software ofrece beneficios al usuario final [10] [11].

Además, este enfoque facilita la detección y corrección de errores o problemas de usabilidad antes de la fase de codificación, lo que ahorra tiempo y recursos a largo plazo. La participación activa de los usuarios en la evaluación del prototipo también contribuye a refinar los requisitos del software, incorporando escenarios y necesidades que podrían no haberse detectado en la etapa inicial de recopilación de información.

Existen trabajos basados en aumentación web como Wamri [12] y Crowdmock [13], donde los usuarios finales definen sus requerimientos utilizando técnicas de aumentación. Estos artefactos son utilizados en el proceso de desarrollo de aplicaciones web tradicionales y en el proceso de desarrollo de artefactos de aumentación.

En este estudio se realiza una revisión sistemática de la literatura con el objetivo de identificar y analizar los métodos, técnicas y herramientas actuales empleados en el diseño de interfaces de usuario (UI) web y su posterior transferencia a código. La investigación se centra en responder a la pregunta clave: ¿Qué métodos, técnicas y herramientas actuales de diseño de UI web, permiten una transferencia del diseño al código?

A continuación, se expone la metodología empleada, detallando el proceso de búsqueda, los criterios de inclusión y exclusión utilizados y la selección de los artículos relevantes para el análisis. Posteriormente, se presenta una descripción detallada de los artículos seleccionados, sintetizando los principales hallazgos relacionados con los métodos, técnicas y herramientas actuales para la transferencia de diseño de UI web a código.

Finalmente, se presentan las conclusiones del estudio, resumiendo los principales aportes y hallazgos relacionados con las metodologías, técnicas y herramientas actuales para la transferencia de diseño de UI web a código. Se discuten las implicaciones prácticas de la investigación, además de proponer trabajos futuros.

2. METODOLOGÍA

Para la realización de la revisión sistemática de la literatura, seguimos la metodología propuesta por Kitchenham [14]. Iniciamos definiendo las preguntas de investigación para preparar un protocolo de revisión que se usó como guía para la realización de la revisión y la recopilación de los datos obtenidos de los artículos. Los artículos seleccionados se leyeron completamente.

Finalmente, se sintetizaron los resultados para la presentación del informe. A continuación, se describen las preguntas que orientaron la investigación, la estrategia de búsqueda para la recuperación de artículos y los criterios de selección.



del 25 de noviembre al 29 de noviembre de 2024

21 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA

PALACIO DE CONVENCIONES DE LA HABANA

60
ANIVERSARIO
cujae

Preguntas de investigación

Se estableció como objetivo principal del estudio investigar e identificar los principales métodos, técnicas y herramientas actuales de diseño de interfaces de usuarios. Con base en el objetivo, se plantean las siguientes preguntas de investigación:

RQ1: ¿Qué métodos existen para la transferencia de diseños de interfaces de usuario en el desarrollo web?

RQ2: ¿Cuáles son las técnicas y herramientas utilizadas para la adaptación de diseños de interfaces de usuario del lado del cliente en el desarrollo web?

RQ3: ¿Qué métodos existen para incorporar patrones de interacción en la recolección de requisitos para el desarrollo de software web?

RQ4: ¿Qué herramientas y tecnologías existen para transferir el diseño y comportamiento de un prototipo de interfaz de usuario web a código HTML/CSS/JS?

Estrategia de búsqueda

Se realizaron búsquedas automáticas utilizando las herramientas de búsqueda avanzadas dispuestas sobre las siguientes bases de datos y bibliotecas digitales: Scopus, IEEE Xplore Digital Library, ScienceDirect, Web of Science (WoS), ACM Digital Library. También se incorporaron artículos realizando una búsqueda manual con las referencias de los artículos recuperados en la búsqueda automática que tenían coincidencia con el título y el resumen.

Se identificaron términos clave, detallados en la Tabla 1, para establecer cadenas de búsqueda en Scopus y ACM Digital Library. Se incluyeron palabras alternativas para ampliar la búsqueda. Se realizaron dos pruebas piloto para validar los resultados.

Tabla 1: Términos de búsqueda según preguntas de investigación.

ID	Términos	Términos alternativos	Usado en
T1	User interface	a) Graphical user interface b) Visual component c) Frontend d) GUI e) UI	RQ1, RQ2, RQ4
T2	Reusable component	a) Modular component b) Resusable design c) Atomic design d) Reusable elements	RQ4
T3	Design transfer	a) Design to code b) Code generation c) Code automation	RQ1
T4	Web development	a) Web based b) Front end development c) Web design d) Web creation e) Web programming	RQ1, RQ2, RQ3, RQ4
T5	Interaction pattern	a) User interaction pattern b) Design pattern interaction c) Interaction guideline d) Human-computer interaction pattern e) HCI	RQ3
T6	Requirement gathering	a) Elicitation b) Requirement análisis c) Requirement scoping d) Requirement definition e) Requirement capture	RQ3
T7	Prototype behavior	a) High Fidelity prototype b) Mockup	RQ4
T8	Client-side adaptation	a) Client-side personalization b) Cliente-side optimization c) Client-side rendering d) Client-side processing e) Web augmentation f) Web augmented	RQ2

Criterios de selección

Como criterios de inclusión, se seleccionan artículos que respondan a uno o varias de las preguntas de investigación, de texto completo, que hayan tenido revisión de pares (journals, revistas, conferencias, libros y capítulos de libros) y que hayan sido escritos en idioma inglés. No se aplican filtros por fecha de publicación del artículo para el relevamiento. Además, se escogieron artículos relacionados con el dominio de ciencias de la computación e ingeniería de software con enfoque en el desarrollo de software para ambiente web.

En cuanto a los criterios de exclusión quedaron determinados por aquellos artículos escritos en idioma distinto al inglés, no arbitrados, o que no correspondan a la temática de estudio y no respondan a ninguna pregunta de investigación.

3. RESULTADOS Y DISCUSIÓN

Este apartado describe las características clave de los artículos analizados y expone los hallazgos que responden a las preguntas de investigación planteadas.

Descripción de artículos seleccionados

En la Fig. 1 se muestra el diagrama PRISMA con los resultados de la estrategia de búsqueda, que recuperó un total de 2347 documentos. Entre ellos, 691 fueron indexados en IEEE, 423 en WoS, 279 en ScienceDirect, 802 en ACM, 136 en SpringerLink y 16 en Scopus. Se identificaron 398 documentos duplicados mediante la comparación de títulos, autores y DOIs. Tras revisar los criterios de inclusión, se descartaron 1830 trabajos.

Se añadieron dos trabajos de forma manual, que se obtuvieron al realizar la revisión de las referencias de los 120 artículos potenciales a través de la herramienta en línea www.connectedpapers.com. La inclusión de [15] se debe a que permite evidenciar la necesidad de cerrar la brecha entre diseñadores y desarrolladores a través de herramientas de prototipado de alta fidelidad. En [16] se sustentan más herramientas de aprendizaje automático que permiten pasar de prototipos manuales a los de alta fidelidad.

Finalmente, se obtuvieron 56 estudios primarios a partir de los cuales se realiza el relevamiento de la investigación.

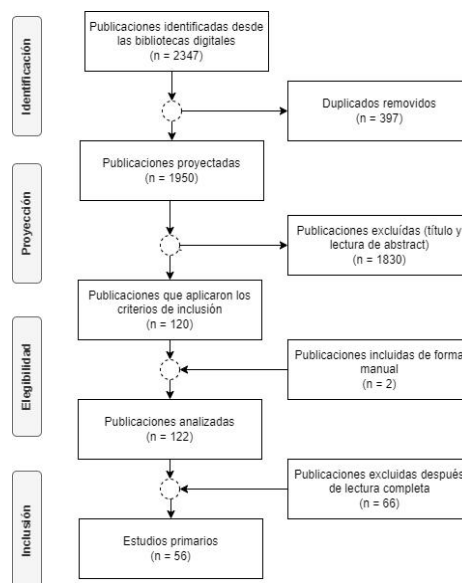


Figura 1: Diagrama PRISMA de la estrategia de búsqueda y criterios de selección.

En la Fig. 2 se establece la demografía de los artículos seleccionados. En esta se evidencia trabajos desde el 2007 hasta el presente año. También se observa que en los últimos cinco años se han venido realizando trabajos mayormente expuestos en Conferencias y journals.

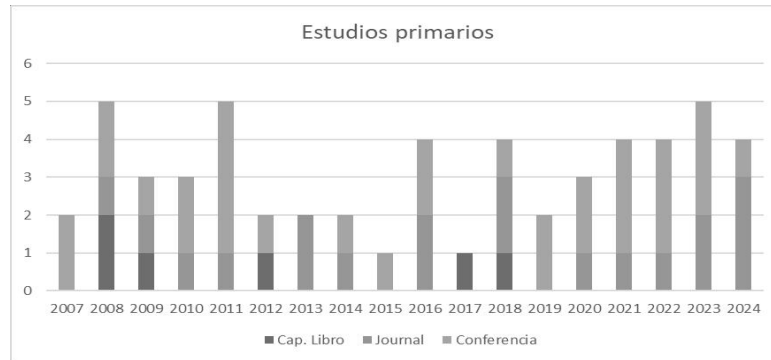


Figura 2: Demografía de los 56 estudios primarios seleccionados en el presente trabajo.

Validez de los estudios seleccionados

Con base en la recomendación de Kitchenham se incluye una fase de validación para asegurar la calidad de los estudios primarios seleccionados. Se plantearon tres criterios: 1. Definición o aplicación de métodos relacionados a la transformación de diseño de interfaz de usuario a código, 2. Definición de la motivación del trabajo de investigación, 3. Exposición clara de resultados y trabajos futuros. Se utilizó la escala planteada por Kitchenham [14], esta va de 0 a 3 (0=No; 1=Algo; 2=Casi; 3=Completa), obteniendo un puntaje promedio por cada artículo solicitado. Se evidencia que no todos los artículos seleccionados como primarios cumplen con todos los criterios indicados, sin embargo, al menos 87.5% de los estudios seleccionados cumplen con un puntaje promedio mayor de 2.5 que es un valor indicativo para estudios con bastante buena calidad.

Transferencia de diseño de interfaces de usuarios web

La transferencia de diseño de UI hacia código se le denomina a la conversión de una representación visual prototipada hacia código, principalmente HTML y CSS. En la transferencia se procura que el código generado sea el fiel reflejo del diseño del interfaz plasmado en la herramienta de prototipado. Su principal función es la de proveer un insumo inicial que pueda ser usado en las etapas de desarrollo de software. Este recurso al poseer la representación visual definida en etapas previas de levantamiento de requerimientos [17]. La calidad de la transferencia del diseño al código tendrá relación a la fidelidad del prototipo y mientras mayor fidelidad se cuente, más cerca de parecerse al producto final se encuentra, según el proceso de diseño de doble diamante. Existen herramientas especializadas que permiten crear prototipos de alta fidelidad como Adobe XD, InVision, Sketch, o Figma [11].

De acuerdo a la revisión de literatura se pueden determinar los siguientes beneficios de la transferencia de diseño:

Reducción de errores de diseño: al tener código HTML y CSS bien estructurado y que respete el diseño de la interfaz de usuario, puede ayudar a reducir los errores relacionados con la mala aplicación de los estilos. Una correcta aplicación de un diseño especificado en el prototipo debe garantizarse en la etapa de desarrollo [17].

Comunicación entre diseñadores y desarrolladores: la colaboración constituye un factor fundamental al permitir establecer la relación entre la visión creativa de los diseñadores con la especificación funcional de los desarrolladores [7] [11] [15].

Mejora la experiencia de usuario: contar con un diseño funcional validado desde etapas tempranas, puede brindar mayor tiempo para realizar iteraciones que promuevan la introducción de mejoras relacionadas en la experiencia del usuario [18].

Incentiva la participación del usuario final: el tener un prototipo de alta fidelidad y de forma funcional en etapas tempranas puede involucrar la participación del usuario en el proceso de mejoras de la interfaz [10] [11] [19].

RQ1: Métodos de transferencia de diseño de interfaz de usuario web a código

En el [Anexo 1](#) se detallan los métodos encontrados que contemplan la transferencia del diseño de interfaces de usuario hacia código. Se permitió determinar que gran parte de las investigaciones (n = 31; 55%) se centran en la aplicación de técnicas como el aprendizaje automático y profundo provistas desde el área de la inteligencia artificial. A partir de la categorización propuesta por [4], se establecen dos categorías para la clasificación de los trabajos estudiados: 1. A partir de mockups creados desde herramientas de diseño, como Figma, Sketch, etc.; 2. A partir de detección de elementos (en papel o captura de imágenes).

Con la revisión de los artículos se puede determinar la presencia de varios métodos donde resaltan aquellos basados en la aplicación de herramientas y técnicas de inteligencia artificial. Estos se concentran en la detección de elementos de interfaz de usuario (diseñados en papel o en imágenes) a través de algoritmos de detección y redes neuronales. Esto posee una limitante ya que la detección correcta está ligada a la calidad del insumo y a la precisión de los algoritmos de detección [1] [15].

Apoyados en la inteligencia artificial existen métodos enfocados en la generación automática de código para elementos específicos de la interfaz (íconos, etiquetas, botones, entre otros [61] [62] [52]) que también pueden lograr a realizar una composición para hacer algo más complejo (inserción de varios de estos componentes a la vez) pero carecen de la posibilidad de reusar código, ya que la generación se hace acotada al contenedor de elementos.

En [65] se discute la necesidad de tener un enfoque orientado a la transformación del diseño de interfaces de usuario hacia código que pueda ser usado directamente en el desarrollo de las aplicaciones. Se propone un método denominado GUIRA que se basa en diseños creados en herramientas como Figma, Sketch, AdobeXD, que se utilizan como insumos que atraviesan por una etapa de Agregación, donde se seleccionan los elementos de la interfaz a pasar para finalmente entregarlos a otra etapa denominada Generador, que es donde finalmente se genera el código con posibilidad de dirigirlo hacia frameworks específicos de JavaScript y TypeScript. Las ventajas incluyen una mayor eficiencia en el tiempo de desarrollo y la capacidad de mantener una alta calidad de código comparable al enfoque tradicional. Una desventaja mencionada es que, aunque el enfoque tradicional puede ofrecer una calidad de código ligeramente superior, la diferencia no es estadísticamente significativa.

RQ2: Técnicas y herramientas para la adaptación de diseños de interfaces de usuario del lado del cliente

La adaptación de diseño de interfaces de usuario se define como el mecanismo para modificar y ajustar un diseño de interfaces existente para que funcione en diferentes contextos o dispositivos. Con base en la revisión de literatura se pudo identificar dos técnicas principales para realizar la adaptación del diseño de interfaces de usuario. Estas son: Diseño responsivo (n = 4; 7%) y Aumentación web (n = 8; 14%).

Diseño responsivo: se denomina a la adaptación de páginas HTML utilizando reglas, llamadas media queries, de hojas de estilos (CSS) con base en las dimensiones de la pantalla, adapta su contenido pudiendo dar una variante muy diferente al diseño original. Esta técnica involucra la necesidad de tener un diseño original (generalmente en dimensiones de pantallas de computadoras de escritorio) para poder extenderse. Estas se caracterizan por seguir el enfoque WCAG 2.2 y WAI-ARIA [71] [72].

Aumentación web: se conoce como la extensión de un sitio web existente en uno nuevo sin necesidad de modificarlo. Superpone contenido, diseño o navegación en las páginas web existentes para personalizar la experiencia del usuario [48]. Por ser una técnica del lado del cliente se apoya en tecnologías como JavaScript y extensiones del navegador. Existen trabajos que han extendido esta técnica con la incorporación de métodos y metodologías que la involucran desde etapas tempranas del desarrollo como herramientas para definir y evolucionar requisitos de aumento web y enfoques dirigidos por modelos que manejan el soporte tanto del lado del cliente como del servidor [13] [54].

RQ3: Métodos para incorporación de patrones de interacción en la recolección de requisitos

Se encontraron nueve artículos ($n = 9$; 16%) que hacen mención a la pregunta de investigación. Se determina que existen herramientas como WebSpec que es un lenguaje visual específico de dominio para especificar requisitos de navegación, interacción y UI en aplicaciones web. Facilita la validación automática de requisitos, la simulación de interfaces de usuario y la gestión de cambios [33].

La Arquitectura Dirigida por Modelos (MDA) permite la separación de preocupaciones y el uso de estándares como UML [25]. Existen herramientas como UWE que incorpora factores humanos en el diseño de aplicaciones de software para mejorar la experiencia del usuario. También se proponen el "Modelo de Prototipo Intensivo" para identificar y analizar estos factores [19].

La generación automática de interfaces de usuario (UI) es un área de investigación activa que busca cumplir con los estándares de HCI y asegurar la consistencia de los artefactos a lo largo del SDLC. Quid es una herramienta de modelado web que utiliza un lenguaje específico de dominio (DSL) y un editor WYSIWYG para prototipar componentes web. También se propone el método TRAIN para desarrollar interfaces de usuario de aplicaciones de RIAs de manera estandarizada [31].

RQ4: Herramientas para transferir diseño y comportamiento de un prototipo de interfaz de usuario

Con base en la revisión de los trabajos seleccionados, no se evidencian trabajos que involucren la incorporación de patrones de interacción de usuario (comportamiento) como parte de la transformación de diseño a código. Sin embargo, existen trabajos que incorporan patrones de interacción en la etapa del modelado de la aplicación utilizando UML [25] [37] [47]. En estos trabajos, se menciona la importancia de incorporar no solo la abstracción de detalles visuales y técnicos, sino también de los eventos de usuario provocando que se obtenga un entregable más completo y funcional desde etapas tempranas.

4. CONCLUSIONES Y TRABAJO FUTURO

La presente revisión sistemática ha permitido determinar que existen diversidad de métodos y herramientas para facilitar la transferencia de diseño, desde herramientas de prototipado tradicionales hasta soluciones basadas en inteligencia artificial y aprendizaje automático. Esto permite determinar dos principales frentes para la transferencia de diseño: la basada en detección de elementos de interfaz de usuario y aquella basada en mockups transferidos a código. Si bien existen trabajos enfocados en la detección de elementos, estos dependen de una correcta definición del insumo utilizado, así como del algoritmo de detección aplicado.

Se necesitan más métodos y herramientas que puedan automatizar tareas complejas en la transferencia de diseño basados en mockups, reduciendo el tiempo y esfuerzo manual requeridos. La transferencia de diseño desde mockups brinda potenciales beneficios como la reducción de errores al contar con una representación visual detallada del diseño final, se minimiza el riesgo de errores en la implementación, optimizando la calidad del código y reduciendo los tiempos de corrección. En la revisión también se aclara la importancia de establecer mejoras en el proceso de comunicación entre los diseñadores y los desarrolladores.

La comunicación efectiva entre ambas partes es esencial para asegurar la traducción precisa de la visión creativa del diseño en código funcional, evitando malentendidos y discrepancias. La transferencia de

diseño también potencia la mejora de la experiencia del usuario al permitir validar y refinar la interfaz de usuario en etapas tempranas del desarrollo, garantizando una experiencia de usuario fluida, intuitiva y satisfactoria.

Finalmente, con base en la revisión de artículos relacionados a la pregunta de investigación RQ4 se logra identificar la necesidad de incorporar patrones de interacción para que la transferencia de diseño no solo deba limitarse a la conversión de elementos visuales, sino que incluya la integración de patrones de interacción que definan el comportamiento de la interfaz de usuario. Esto permite crear interfaces más usables y que respondan a las expectativas del usuario.

REFERENCIAS

1. N. A. Shamat, S. Sulaiman, y J. S. Sinpang, «A systematic literature review on user interface design for web applications», *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, n.º 3-4, pp. 57-61, 2017.
2. H. Cao y Y. Gao, «Research and implementation of using AJAX technology to improve the user experience», en *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, IEEE, 2011, pp. 2539-2542.
3. Z. Lin, J. Wu, Q. Zhang, y H. Zhou, «Research on web applications using ajax new technologies», en *2008 International Conference on MultiMedia and Information Technology*, IEEE, 2008, pp. 139-142.
4. T. Kaluarachchi y M. Wickramasinghe, «A systematic literature review on automatic website generation», *Journal of Computer Languages*, vol. 75, p. 101202, jun. 2023, doi: 10.1016/j.col.2023.101202.
5. C. Crumlish y E. Malone, *Designing social interfaces: Principles, patterns, and practices for improving the user experience*. O'Reilly Media, Inc., 2009.
6. B. Scott y T. Neil, *Designing web interfaces: Principles and patterns for rich interactions*. O'Reilly Media, Inc., 2009.
7. E. Salinas, R. Cueva, y F. Paz, «A systematic review of user-centered design techniques», en *Design, User Experience, and Usability. Interaction Design: 9th International Conference, DUXU 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings, Part I 22*, Springer, 2020, pp. 253-267.
8. B. Suranto, «Software prototypes: Enhancing the quality of requirements engineering process», en *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*, IEEE, 2015, pp. 148-153.
9. F. Staiano, *Designing and Prototyping Interfaces with Figma: Learn essential UX/UI design principles by creating interactive prototypes for mobile, tablet, and desktop*. Packt Publishing Ltd, 2022.
10. I. Rodríguez-Calero, S. R. Daly, G. Burleson, y K. H. Sienko, «Prototyping strategies to engage stakeholders during early stages of design: a study across three design domains», *Journal of Mechanical Design*, vol. 145, n.º 4, p. 041413, 2023.
11. K. K. Feng, T. W. Li, y A. X. Zhang, «Understanding collaborative practices and tools of professional UX practitioners in software organizations», en *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1-20.
12. L. G. Marticorena, L. A. Morales, L. Antonelli, G. Rossi, y D. Firmenich, «Development iterations based on web augmentation and context tasks», *Multimedia Tools and Applications*, vol. 82, n.º 8, pp. 11793-11817, 2023.
13. D. Firmenich, S. Firmenich, J. Rivero, L. Antonelli, y G. Rossi, «CrowdMock: an approach for defining and evolving web augmentation requirements», *REQUIREMENTS ENGINEERING*, vol. 23, n.º 1, pp. 33-61, mar. 2018, doi: 10.1007/s00766-016-0257-3.
14. B. Kitchenham, «Procedures for performing systematic reviews», *Keele, UK, Keele University*, vol. 33, n.º 2004, pp. 1-26, 2004.
15. M. Bexiga, S. Garbatov, y J. C. Seco, «Closing the gap between designers and developers in a low code ecosystem», en *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, 2020, pp. 1-10.



del 25 de noviembre al 29 de noviembre de 2024

21 CONVENCION CIENTIFICA
DE INGENIERIA Y ARQUITECTURA
PALACIO DE CONVENCIONES DE LA HABANA

60
ANIVERSARIO
cujae

16. D. de Souza Baulé, C. G. von Wangenheim, A. von Wangenheim, y J. C. Hauck, «Recent Progress in Automated Code Generation from GUI Images Using Machine Learning Techniques.», *J. Univers. Comput. Sci.*, vol. 26, n.º 9, pp. 1095-1127, 2020.
17. M. Bochicchio y R. Paiano, «Prototyping web applications», en *Proceedings of the 2000 ACM symposium on Applied computing-Volume 2*, 2000, pp. 978-983.
18. E. Jongmans, F. Jeannot, L. Liang, y M. Dampérat, «Impact of website visual design on user experience and website evaluation: the sequential mediating roles of usability and pleasure», *Journal of Marketing Management*, vol. 38, n.º 17-18, pp. 2078-2113, 2022.
19. S. Abduljalil y D. -K. Kang, «Analysis of human factors in software application design for effective user experience», en *13th International Conference on Advanced Communication Technology (ICACT2011)*, feb. 2011, pp. 1446-1451.
20. X. Lu, J. Wan, y J. Hou, «A Model Based Heuristic Design of Web User Interface», en *2007 IEEE International Conference on Information Reuse and Integration*, ago. 2007, pp. 6-11. doi: 10.1109/IRI.2007.4296589.
21. M. Anastassova, C. Mégard, y J.-M. Burkhardt, «Prototype evaluation and user-needs analysis in the early design of emerging technologies», en *Human-Computer Interaction. Interaction Design and Usability: 12th International Conference, HCI International 2007, Beijing, China, July 22-27, 2007, Proceedings, Part I 12*, Springer, 2007, pp. 383-392.
22. A. G. Cuesta, J. C. Granja, y R. V. O'Connor, «A Model Driven Architecture Approach to Web Development», en *Software and Data Technologies*, vol. 22, J. Filipe, B. Shishkov, M. Helfert, y L. A. Maciaszek, Eds., en *Communications in Computer and Information Science*, vol. 22. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 101-113. doi: 10.1007/978-3-540-88655-6_8.
23. P. Dolog, «Designing Adaptive Web Applications», en *SOFSEM 2008: Theory and Practice of Computer Science*, vol. 4910, V. Geffert, J. Karhumäki, A. Bertoni, B. Preneel, P. Návrat, y M. Bieliková, Eds., en *Lecture Notes in Computer Science*, vol. 4910. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 23-33. doi: 10.1007/978-3-540-77566-9_3.
24. S. Lukosch y M. Bourimi, «TOWARDS AN ENHANCED ADAPTABILITY AND USABILITY OF WEB-BASED COLLABORATIVE SYSTEMS», *INTERNATIONAL JOURNAL OF COOPERATIVE INFORMATION SYSTEMS*, vol. 17, n.º 4, pp. 467-494, dic. 2008, doi: 10.1142/S0218843008001944.
25. R. A. C. de Souza y R. S. M. de Barros, «A Model-Driven Method for the Development of Web Applications User Interaction Layer», en *2008 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering*, jun. 2008, pp. 91-98. doi: 10.1109/TASE.2008.11.
26. T. Memmel, H. Ziegler, R. Oed, y H. Reiterer, «User interface specification in complex web-based information spaces», en *Proceedings of the Third IASTED International Conference on Human Computer Interaction*, en HCI '08. USA: ACTA Press, feb. 2008, pp. 180-185.
27. G.-J. Houben, N. Koch, y A. Vallecillo, «Model-Driven Web Engineering (MDWE 2008)», en *Models in Software Engineering*, vol. 5421, M. R. V. Chaudron, Ed., en *Lecture Notes in Computer Science*, vol. 5421. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 151-152. doi: 10.1007/978-3-642-01648-6_16.
28. J. Ginzburg, D. Distante, G. Rossi, y M. Urbieta, «OBLIVIOUS INTEGRATION OF VOLATILE FUNCTIONALITY IN WEB APPLICATION INTERFACES», *JOURNAL OF WEB ENGINEERING*, vol. 8, n.º 1, pp. 25-47, mar. 2009.
29. M. Linaje, J. C. Preciado, R. Morales-Chaparro, R. Rodríguez-Echeverría, y F. Sánchez-Figueroa, «Automatic generation of rias using rux-tool and webratio», en *Web Engineering: 9th International Conference, ICWE 2009 San Sebastián, Spain, June 24-26, 2009 Proceedings 9*, Springer, 2009, pp. 501-504.
30. P. Fraternali, S. Comai, A. Bozzon, y G. Carughi, «Engineering Rich Internet Applications with a Model-Driven Approach», *ACM TRANSACTIONS ON THE WEB*, vol. 4, n.º 2, abr. 2010, doi: 10.1145/1734200.1734204.
31. F. J. Martínez-Ruiz, «The triad-based design of rich user interfaces for internet applications», en *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, en EICS '10. New York, NY, USA: Association for Computing Machinery, jun. 2010, pp. 345-348. doi: 10.1145/1822018.1822077.
32. J. M. Rivero, G. Rossi, J. Grigera, J. Burella, E. R. Luna, y S. Gordillo, «From mockups to user interface models: an extensible model driven approach», en *Current Trends in Web Engineering:*



del 25 de noviembre al 29 de noviembre de 2024

21 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA
PALACIO DE CONVENCIONES DE LA HABANA

60
ANIVERSARIO
cujae

- 10th International Conference on Web Engineering ICWE 2010 Workshops, Vienna, Austria, July 2010, Revised Selected Papers 10*, Springer, 2010, pp. 13-24.
33. E. Robles Luna, G. Rossi, y I. Garrigós, «WebSpec: a visual language for specifying interaction and navigation requirements in web applications», *Requirements Eng*, vol. 16, n.º 4, pp. 297-321, nov. 2011, doi: 10.1007/s00766-011-0124-1.
 34. J. d. Monte-Mor, E. O. Ferreira, H. F. Campos, A. M. da Cunha, y L. A. V. Dias, «Applying MDA Approach to Create Graphical User Interfaces», en *2011 Eighth International Conference on Information Technology: New Generations*, abr. 2011, pp. 766-771. doi: 10.1109/ITNG.2011.206.
 35. F. Kruger, C. Winkelholz, y C. M. Schlick, «System for a model based analysis of user interaction patterns within web-applications», en *2011 IEEE International Conference on Systems, Man, and Cybernetics*, oct. 2011, pp. 1274-1279. doi: 10.1109/ICSMC.2011.6083835.
 36. S. Firmenich, G. Rossi, S. Gordillo, y M. Winckler, «A flexible architecture for client-side adaptation», en *International Conference on Web Engineering*, Springer, 2011, pp. 327-331.
 37. N. Koch y S. Kozuruba, «Requirements Models as First Class Entities in Model-Driven Web Engineering», en *Current Trends in Web Engineering*, vol. 7703, M. Grossniklaus y M. Wimmer, Eds., en *Lecture Notes in Computer Science*, vol. 7703. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 158-169. doi: 10.1007/978-3-642-35623-0_16.
 38. O. Díaz, J. De Sosa, C. Arellano, y S. Trujillo, «Web-Based tool integration: a web augmentation approach», en *International Conference on Web Engineering*, Springer, 2012, pp. 431-434.
 39. L. Rivero y T. Conte, «Using an empirical study to evaluate the feasibility of a new usability inspection technique for paper based prototypes of web applications», *J Softw Eng Res Dev*, vol. 1, n.º 1, p. 2, dic. 2013, doi: 10.1186/2195-1721-1-2.
 40. L. Colombo-Mendoza, G. Alor-Hernández, A. Rodríguez-González, y R. Colomo-Palacios, «ALEXANDRIA: A VISUAL TOOL FOR GENERATING MULTI-DEVICE RICH INTERNET APPLICATIONS», *JOURNAL OF WEB ENGINEERING*, vol. 12, n.º 3-4, pp. 317-359, jul. 2013.
 41. J. M. Rivero, J. Grigera, G. Rossi, E. Robles Luna, F. Montero, y M. Gaedke, «Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering», *Information and Software Technology*, vol. 56, n.º 6, pp. 670-687, jun. 2014, doi: 10.1016/j.infsof.2014.01.011.
 42. D. Firmenich, S. Firmenich, J. M. Rivero, y L. Antonelli, «A platform for web augmentation requirements specification», en *Web Engineering: 14th International Conference, ICWE 2014, Toulouse, France, July 1-4, 2014. Proceedings 14*, Springer, 2014, pp. 1-20.
 43. M. Brambilla, «Modeling and execution of software user interfaces», en *2015 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, feb. 2015, p. IS-9.
 44. J. Hermida, et al., «XANUI: A TEXTUAL PLATFORM-INDEPENDENT MODEL FOR RICH USER INTERFACES», *JOURNAL OF WEB ENGINEERING*, vol. 15, n.º 1-2, pp. 45-83, mar. 2016.
 45. F. P. Basso, R. M. Pillat, T. C. Oliveira, F. Roos-Frantz, y R. Z. Frantz, «Automated design of multi-layered web information systems», *Journal of Systems and Software*, vol. 117, pp. 612-637, jul. 2016, doi: 10.1016/j.jss.2016.04.060.
 46. S. Roubi, M. Erramdani, y S. Mbarki, «Modeling and generating graphical user interface for MVC Rich Internet Application using a model driven approach», en *2016 International Conference on Information Technology for Organizations Development (IT4OD)*, abr. 2016, pp. 1-6. doi: 10.1109/IT4OD.2016.7479249.
 47. S. Roubi, et al. «Extending graphical part of the Interaction Flow Modeling Language to Generate Rich Internet Graphical User Interfaces», en *2016 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, feb. 2016, pp. 161-167.
 48. I. Aldalur, et al., «Web Augmentation as a Promising Technology for End User Development», en *New Perspectives in End-User Development*, F. Paternò y V. Wulf, Eds., Cham: Springer International Publishing, 2017, pp. 433-459. doi: 10.1007/978-3-319-60291-2_17.
 49. J. M. Rivero et al., «Improving Legacy Applications with Client-Side Augmentations», en *Web Engineering*, vol. 10845, T. Mikkonen, R. Klamma, y J. Hernández, Eds., en *Lecture Notes in Computer Science*, vol. 10845. , Cham: Springer International Publishing, 2018, pp. 162-176. doi: 10.1007/978-3-319-91662-0_12.
 50. C. D'Souza, V. Deufemia, A. Ginige, y G. Polese, «Enabling the generation of web applications from mockups», *Software: Practice and Experience*, vol. 48, n.º 4, pp. 945-973, 2018.



del 25 de noviembre al 29 de noviembre de 2024

21 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA
PALACIO DE CONVENCIONES DE LA HABANA

60
ANIVERSARIO
cujae

51. A. Tarasiev, M. Filippova, K. Aksyonov, y O. Aksyonova, «Developing Prototype of CASE-Tool to Create Automation Systems Based on Web Applications Using Code Generation», en *2018 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, nov. 2018, pp. 1-4. doi: 10.1109/Dynamics.2018.8601443.
52. B. Aşıroğlu *et al.*, «Automatic HTML Code Generation from Mock-Up Images Using Machine Learning Techniques», en *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, abr. 2019, pp. 1-4. doi: 10.1109/EBBT.2019.8741736.
53. P. J. Molina, «Quid: prototyping web components on the web», en *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, en EICS '19. New York, NY, USA: Association for Computing Machinery, jun. 2019, pp. 1-5. doi: 10.1145/3319499.3330294.
54. M. Urbietta, S. Firmenich, G. Bosetti, P. Maglione, G. Rossi, y M. A. Olivero, «MDWA: a model-driven Web augmentation approach—coping with client- and server-side support», *Softw Syst Model*, vol. 19, n.º 6, pp. 1541-1566, nov. 2020, doi: 10.1007/s10270-020-00779-5.
55. M. Xie, S. Feng, Z. Xing, J. Chen, y C. Chen, «UIED: a hybrid tool for GUI element detection», en *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, en ESEC/FSE 2020. New York, NY, USA: Association for Computing Machinery, nov. 2020, pp. 1655-1659. doi: 10.1145/3368089.3417940.
56. Y. Xu, L. Bo, X. Sun, B. Li, J. Jiang, y W. Zhou, «image2emmet: Automatic code generation from web user interface image», *JOURNAL OF SOFTWARE-EVOLUTION AND PROCESS*, vol. 33, n.º 8, ago. 2021, doi: 10.1002/smr.2369.
57. J. Pacheco, S. Garbatov, y M. Goulão, «Improving Collaboration Efficiency Between UX/UI Designers and Developers in a Low-Code Platform», en *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, oct. 2021, pp. 138-147. doi: 10.1109/MODELS-C53483.2021.00025.
58. C. Machado y J. C. Campos, «Towards the integration of user interface prototyping and model-based development», en *2021 International Conference on Graphics and Interaction (ICGI)*, nov. 2021, pp. 1-8. doi: 10.1109/ICGI54032.2021.9655284.
59. S. Reyas *et al.*, «An Investigation into UI generation compliant with HCI standards ensuring artifact consistency across SDLC», en *2021 21st International Conference on Advances in ICT for Emerging Regions (ICTer)*, dic. 2021, pp. 93-98. doi: 10.1109/ICTer53630.2021.9774787.
60. D. Firmenich, S. Firmenich, G. Rossi, M. Wimmer, I. Garrigós, y C. Gonzalez-Mora, «Engineering Web Augmentation software: A development method for enabling end-user maintenance», *INFORMATION AND SOFTWARE TECHNOLOGY*, vol. 141, ene. 2022, doi: 10.1016/j.infsof.2021.106735.
61. B. Aşıroğlu *et al.*, «A Deep Learning Based Object Detection System for User Interface Code Generation», en *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, jun. 2022, pp. 1-5. doi: 10.1109/HORA55278.2022.9799941.
62. D. Yashaswini, Sneha, y N. Kumar, «HTML Code Generation from Website Images and Sketches using Deep Learning-Based Encoder-Decoder Model», en *2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, oct. 2022, pp. 133-138. doi: 10.1109/ICCCMLA56841.2022.9989298.
63. Y. Wang, L. Luo, y H. Liu, «Bridging the Semantic Gap Between Customer Needs and Design Specifications Using User-Generated Content», *IEEE Transactions on Engineering Management*, vol. 69, n.º 4, pp. 1622-1634, ago. 2022, doi: 10.1109/TEM.2020.3021698.
64. M. Arefin, K. M. Hossen, y R. Khan, «pic2code: Generating HTML Code from Handwritten Picture», en *2023 26th International Conference on Computer and Information Technology (ICCIT)*, dic. 2023, pp. 1-6. doi: 10.1109/ICCIT60459.2023.10441098.
65. G. E. Raptis y C. Katsini, «Transforming Ideas into Code: A Novel Approach to Design-to-Development Workflow», en *2023 International Conference on Computer and Applications (ICCA)*, nov. 2023, pp. 1-6. doi: 10.1109/ICCA59364.2023.10401412.
66. A. S. Chapai y E. J. Rapos, «SkeMo: Sketch Modeling for Real-Time Model Component Generation», en *2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, oct. 2023, pp. 173-183. doi: 10.1109/MODELS58315.2023.00038.



del 25 de noviembre al 29 de noviembre de 2024

21 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA

PALACIO DE CONVENCIONES DE LA HABANA

60
ANIVERSARIO
cujae

67. T. Kaluarachchi y M. Wickramasinghe, «WebDraw: A machine learning-driven tool for automatic website prototyping», *SCIENCE OF COMPUTER PROGRAMMING*, vol. 233, mar. 2024, doi: 10.1016/j.scico.2023.103056.
68. I. Aldalur, «Web Augmentation: A systematic mapping study», *Science of Computer Programming*, vol. 232, p. 103045, ene. 2024, doi: 10.1016/j.scico.2023.103045.
69. M. Samir, A. Elsayed, y M. I. Marie, «A Model for Automatic Code Generation from High Fidelity Graphical User Interface Mockups using Deep Learning Techniques», *Intl. J. Adv. Comput. Sci. Appl.*, vol. 15, n.º 3, pp. 677-691, 2024, doi: 10.14569/IJACSA.2024.0150369.
70. S. R, S. P. K. R, V. R. S, V. W, y L. Srinivasan, «STC (Sketch To Code)-An Enhanced Html & CSS Autocode Generator from Handwritten Text and Image Using Deep Learning», en *2024 2nd International Conference on Networking and Communications (ICNWC)*, abr. 2024, pp. 1-6. doi: 10.1109/ICNWC60771.2024.10537336.
71. G. Xu, S. Wang, Y. Hu, y X. Shen, «A Multilayer Component Pane 3D Layout Frame Design for Responsive Websites», en *2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, oct. 2023, pp. 474-478. doi: 10.1109/ISMAR-Adjunct60411.2023.00103.
72. S. Wongsalam y T. Senivongse, «Visual Design and Code Generation of User Interface Based on Responsive Web Design Approach», en *Proceedings of the 2019 3rd International Conference on Software and e-Business*, en ICSEB '19. New York, NY, USA: Association for Computing Machinery, ene. 2020, pp. 51-59. doi: 10.1145/3374549.3374560.

Sobre los autores

Jonathan Stalin Delgado Guerrero, estudiante del Doctorado de Ciencias Informáticas de la Facultad de Informática de la Universidad Nacional de La Plata, Argentina. Es también becario por la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación de la República del Ecuador.

Dr. Leandro Antonelli, con grado de Doctor en Ciencias Informáticas por la Universidad Nacional de La Plata. Profesor en materias de grado y postgrado en la Universidad Nacional de La Plata. También es docente de postgrado en la Universidad Abierta Interamericana, como así también profesor visitante de otras Universidades del país (Universidad Nacional de la Patagonia Austral, Universidad Nacional de Río Negro, Universidad Nacional de San Juan y Universidad Nacional de Cuyo).

El Dr. Firmenich Diego, obtuvo su doctorado en Ciencias Informáticas en la Facultad de Informática de la Universidad Nacional de La Plata, Argentina. Es Profesor Investigador de dedicación exclusiva, en el Departamento de Informática de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia, Argentina.