



Plataforma No-Code para la Recolección de datos en Campo por Diferentes Roles

Johnny Andrés Chinchajoa Taimal

Anteproyecto presentada(o) como requisito parcial para optar al título de:
Magister en Ingeniería de Software

Director(a):

Título (Ph.D., MSc) y nombre del director(a)

Pontificia Universidad Javeriana Cali

Facultad de Ingeniería

Departamento de Electrónica y Ciencias de la Computación

Cali, Colombia

25 de mayo de 2022

Índice

1. Definición del problema	3
1.1. Planteamiento del problema	3
1.2. Formulación del problema	4
2. Objetivos del proyecto	6
2.1. Objetivo General	6
2.2. Objetivos específicos	6
3. Alcance	7
4. Justificación del trabajo de grado	8
5. Marco teórico de referencia y antecedentes	9
5.1. Bases Teóricas	9
5.2. Estado del Arte	11

Índice de figuras

Índice de tablas

1. Definición del problema

1.1. Planteamiento del problema

De acuerdo a Galindo (2019) la transformación digital ha ~~echo~~ que las empresas sin importar su tamaño o mercado adapten sus modelos de negocio para seguir siendo competitivas. Por ejemplo la recolección de datos ha empezado a migrar a formularios digitales.

En la recolección de datos Cisneros-Caicedo et al. (2022) ~~destaca~~ la encuesta como la técnica más usada para obtener información cualitativa o cuantitativa, que además se puede realizar en entornos virtuales online y offline sobre cuestionarios estructurados en diferentes pasos que en algunos casos son diligenciado por distintos roles.

En busca de la transformación digital diferentes compañías están generando sus propias soluciones tecnológicas, conformando equipos de desarrollo de software que las construyen y mantienen. Sin embargo, de acuerdo a Bock and Frank (2021) la falta de desarrolladores de software se ha convertido en un obstáculo para que muchas empresas enfrenten con éxito la transformación digital, principalmente porque las soluciones son muy costosas.

Por esta razón surge como alternativa el software como servicio (SaaS) ~~que~~ según la definición de Mohammed and Zeebaree (2021), hacen que las empresas solo se preocupen de cómo utilizar las herramientas, olvidándose del hardware, sistemas operativos, aplicaciones, etc. Algunos ejemplos de éxito de estas soluciones son la suite de Google, que con el servicio de Google Forms se están usando para realizar recolección de información.

A la lista de opciones para construir herramientas digitales Bock and Frank (2021) suman las aplicaciones low-code y no-code como una nueva clase de entornos de desarrollo de software que ofrecen a los usuarios el empoderamiento para construir sus propias soluciones. Sin embargo ellos mencionan que por la heterogeneidad de las aplicaciones low-code y no-code, aún existen empresas que no han podido empezar con la transformación digital, pues las herramientas existentes pueden ser demasiado

complejas y costosas para cumplir las necesidades específicas de cada negocio.

~~Por ejemplo~~ en el análisis de Hervás (2020) sobre Power Apps, una plataforma low-code de Microsoft, se destaca la facilidad y ahorro de tiempo a la hora de persistir datos, sin embargo la plataforma esta limitada a bases de datos compatibles con Power Apps y requiere conocimiento técnico para modelar una estructura de datos si se requiere asignar partes de un formulario a diferentes roles. Esta responsabilidad podría causar que las aplicaciones tengan modelos de datos rígidos, que impiden agregar o eliminar campos de una entidad haciendo compleja la evolución del modelo para adaptarse a las características cambiantes del negocio.

~~De acuerdo al análisis de Moskal (2021)~~ sigue siendo un reto para las aplicaciones low-code y no-code adjuntar eventos dinámicos a un formulario, pues sigue sin una respuesta aceptada en más del 70 por ciento de respuestas de Stack Overflow. Por esta razón modelos de datos dinámicos generan costos adicionales en el procesos de construcción, modificación y mantenimiento de los formularios.

De esta manera, la complejidad y altos costos que siguen teniendo las aplicaciones low-code y no-code, que son la solución digital factible a la hora de crear formularios, hace que diferentes negocios con esta necesidad no se hayan migrado a formulario digitales. Soluciones como Appian, Google App Maker, Microsoft Powerapps y Salesforce Lightning, que son algunas de las plataformas low-code populares de acuerdo a Moskal (2021), añaden una complejidad adicional en el momento de segmentar la recolección de información en diferentes pasos, asignarlas a diferentes roles y funcionen offline, que es importante en el momento de recolección de información en campo.

1.2. Formulación del problema

- ¿Un modelo de datos no relacional permitirá agregar, actualizar o eliminar propiedades de un formulario sin generar costos adicionales y sin afectar los datos ya recolectados?
- ¿Una arquitectura de software para recolectar información en modo offline y con un modelo de datos no relacional flexible permitirá ~~construir~~ soluciones asequibles para la creación de formularios?

- ¿Una plataforma no-code para elaborar formularios y asignarlos a diferentes roles para recolección de información en campo, permitirá a las empresas empezar a migrar la recolección de información a formularios digitales?

Creo que el contexto del problema no está presentando de forma adecuada los términos que usas en tus preguntas

2. Objetivos del proyecto

Todos esos términos que estas usando aquí deberían aparecer en el planteamiento del problema. El tema de la flexibilidad, el tema offline, etc.
En cuanto a que sea un modelo de datos no relacional eso es de la solución no del problema

2.1. Objetivo General

Diseñar una solución digital a partir de un modelo de datos no relacional flexible, que permita la recolección de datos incluso offline, con la opción de segmentar formularios para la participación de múltiples actores en el levantamiento de la información.

Tu quieres lograr un modelo de datos flexible, en este punto realmente no sabes cuál es, el objetivo es definir un modelo de datos flexible, no que sea no relacional per se

2.2. Objetivos específicos

- Definir un modelo de datos no relacional que permita agregar, actualizar o eliminar propiedades.
- Añadir al modelo de datos la capacidad de segmentar los formularios a múltiples actores y diferentes roles de usuario.
- Diseñar una arquitectura de software que implemente el modelo de datos flexible y tenga en cuenta la recolección de información en campo (online/offline).
- Construir una aplicación no-code para elaborar formularios, a partir de la arquitectura de software y el modelo de datos, para la recolección de información en campo por diferentes roles.

Estos objetivos están planteados desde el punto de vista de la solución y no desde el problema.
Es como si ya existiera la solución y estás describiendo los objetivos en términos de las características de tu solución

Esta muy detallado, parecen requerimientos. El alcance no es lo que la aplicación podrá hacer sino hasta donde vas a llegar con la investigación

3. Alcance

- El modelo de datos debe soportar como mínimo 10 tipos de entradas para formularios digitales.
- La arquitectura software diseñada deberá cumplir como mínimo los siguientes atributos de calidad: seguridad, mantenibilidad, Escalabilidad y desempeño.
- La arquitectura diseñada tendrá la capacidad de desplegarse en al menos una nube.
- La aplicación web no-code permitirá a los usuarios finales construir formularios con la funcionalidad seleccionar, arrastrar y soltar, con la opción de agruparlos en secciones.
- La aplicación web no-code permitirá crear roles y asignar secciones del formulario a los roles.
- La aplicación web no-code permitirá disponer el formulario a través de una URL y mostrara la sección del formulario que corresponda a cada rol.
- La aplicación web no-cocce permitirá descargar los datos recolectados.

Ideal que tuvieras más cifras y datos aquí

4. Justificación del trabajo de grado

Cuando las empresas buscan migrar sus formularios a formatos digitales se encuentran con variedad de herramientas, muchas de ellas **no-code y low-code**, fáciles de usar. La solución de este trabajo tendrá en **cuenta este aspecto** pero además priorizará la modificación de las propiedades de los formularios, la opción de ejecutar partes de los formularios por diferentes roles y la recolección de información en campo. Estas son características que difícilmente se encuentran juntas en las herramientas actuales. De esta manera, muchas organizaciones podrán construir formularios, actualizarlos en el tiempo, con la posibilidad de segmentarlos por roles para recolectar información en modo offline.

Esta solución al tener un modelo de datos no relacional permitirá que cuando el negocio lo pida, sea posible agregar mas propiedades o eliminar propiedades permitiendo que ~~la data~~ recopilada se adapte al modelo. **Es muy común que para recolectar información en un formulario existan componentes gráficos asociados a cada tipo de dato por ejemplo: Text Area, InputText, Date, Number, SelectBox, CheckBox, UploadImage etc. En las aplicaciones tradicionales la relación entre datos se modela de tal manera que la información no gaste mucho espacio, para eso se procede a realizar actividades de normalización, sin embargo al tener un modelo de datos no relacional como se plantea en este trabajo la normalización ya no sería necesaria pues ya se estaría disminuyendo la redundancia de datos.**

Por otro lado al no tener un modelo de datos rígido facilita hacer búsquedas sobre la información. Hoy en día es posible almacenar archivos JSON con múltiples propiedades y hacer búsquedas sobre su contenido de manera rápida sin necesidad de tener los datos estructurados.

Estás teniendo en cuenta la complejidad de las búsquedas?
Cómo estás midiendo la "rapidez" o cuáles son los criterios que te permiten decidir que una búsqueda es rápida o no

5. Marco teórico de referencia y antecedentes

5.1. Bases Teóricas

- Transformación digital: ~~Según Martinelli et al. (2021)~~ este es proceso que están realizando las organizaciones como estrategia competitiva y modelo de negocio en el que se adoptan herramientas y servicios digitales. Así pues buscando apoyar este proceso de transformación el Gobierno (2022) colombiano a definido unos niveles de madurez de acuerdo a la implementación de: Herramientas intercomunicadas y analítica de datos (4), software especializado o aplicaciones web (3), herramientas ofimáticas (2) y la digitalización de la documentación (1). ~~Ahora bien uno~~ de los muchos procesos que se están migrando dentro de los niveles 2 y 3 es la recolección de datos.
- Recolección de datos: La recolección de datos es el proceso que se realiza para adquirir información a través de formularios impresos, formularios digitales, entrevistas telefónicas o personales. Este es un proceso que muchas organizaciones realizaban a menudo con cuestionarios físicos, los cuales estaban expuestos a manipulaciones o errores humanos difíciles de controlar. Con la digitalización surge la oportunidad de realizar esta tarea sobre formularios digitales.
- Formulario digital: es el equivalente de un documento papel que se utiliza para recopilar datos. Estos reproducen absolutamente todos los elementos de un formulario en papel, como por ejemplo redactar texto, dibujar un boceto, rellenar un checkbox o añadir una firma y además permiten recolectar datos de una manera rápida, ordenada y validando los datos ingresados. Esta es la razón por la que las organizaciones han empezado a buscar alternativas para la construcción de estos formularios, y para ello han empezado a usar alternativas como el software como servicio.
- Software as a Service (SaaS): ~~De acuerdo con Bello et al. (2021)~~ estas son soluciones en la nube que proveen de software con un modelo de pago por su uso. El proveedor se encarga de administrar el hardware y software y el usuario solo se

debe conectar a través de internet a la aplicación. De esta manera las empresas solo deben preocuparse de como estructurar y construir los formularios y para ello surgen las herramientas no-code y low-code.

- Plataformas No-Code y Low-code: ~~En Hylton et al. (2021)~~ se considera a una plataforma No-code a una aplicación web o móvil que puede usarse para crear soluciones digitales sin escribir una sola línea de código. Por otro lado las herramientas low-code ~~de acuerdo a Bock and Frank (2021)~~ son plataformas que permiten el desarrollo, la implementación, la ejecución y la administración de aplicaciones mediante abstracciones de programación declarativas de alto nivel, como lenguajes de programación basados en metadatos y basados en modelos. Según Moskal (2021) los beneficios de usar plataformas de código bajo también incluyen flexibilidad y agilidad, tiempo de desarrollo que permite una respuesta rápida a las demandas del mercado, corrección de errores reducida, menor esfuerzo de implementación y mantenimiento más fácil. De esta manera esto les permite a las empresas construir los formularios de forma sencilla, por ejemplo seleccionar y arrastrar los campos que requiere para la recolección de información y en algunos casos, si la solución es muy compleja, diseñar el modelo de datos.
- Modelo de datos: Son representaciones visuales de los elementos de datos de una empresa y de las conexiones entre ellos. Los más usados son los modelos de datos conceptuales también conocidos como modelos de dominio, modelos de datos lógicos, modelos de datos físicos. A partir de los cuales se generan las estructuras de datos.
- Estructura de datos: son ~~aquellas~~ que les permiten a los desarrolladores, organizar la información de manera eficiente, y diseñar la solución correcta para un determinado problema, estos pueden ser árboles, grafos, tablas, etc. Una estructura de datos permite trabajar en alto nivel de abstracción almacenando información para luego acceder a ella, modificarla y manipularla. Entre las alternativas para implementar las estructuras de datos, muy usadas hoy en día,

están las bases de datos NoSql.

- NoSql: ~~De acuerdo a Davoudian et al. (2018)~~ son opciones de almacenamiento de información que adoptan modelos de datos flexibles, estos pueden tener o no esquemas de datos, permitiendo que el almacenamiento escale sin afectar la disponibilidad y latencia de una aplicación. Esta opción de almacenamiento es una buena opción si se necesita modificabilidad dentro de una arquitectura software.
- Arquitectura software: representa la decisión de diseño más temprana. Es determinada básicamente por criterios de calidad como la modificabilidad, mantenibilidad, seguridad y el rendimiento.

5.2. Estado del Arte

Plataformas No-code y Low-code

- Microsoft Power Apps Microsoft (2022): Este es un servicio soportado por Microsoft para crear y lanzar aplicaciones web a partir de plantillas, con la técnica de arrastrar y soltar. sin embargo el low-code tiene una importancia secundaria, pues al aumentar la complejidad de la solución se requiere de conocimientos técnicos para modelar las estructuras de datos. Esta solución entonces empieza a ser muy compleja cuando el usuario requiere construir una formulario para la recolección de información.
- Mendix disponible en (Mendix, 2022): Es una plataforma que permite a desarrolladores de varios niveles técnicos crear aplicaciones de forma rápida, abstractando y automatizando el proceso de desarrollo. Mendix se ofrece principalmente como soluciones para el desarrollo rápido de aplicaciones, bajo el modelo de plataforma como servicio (PaaS), lo que disminuye la dificultad de los usuarios al llevar sus soluciones a producción. Sin embargo el desarrollo de una solución combina componentes para diseño de la interfaz, el modelado de datos, la gestión de persistencia de datos, el modelado de procesos o la lógica

de aplicación y el soporte de la solución, todo lo anterior aumenta la dificultad en el momento de querer asociar partes de un formulario a diferentes roles.

- Appian disponible en (Appian, 2022): Es una plataforma Low-Code para diseñar y automatizar procesos a nivel empresarial. Con ella es posible crear aplicaciones y flujos de trabajo rápidamente. Sin embargo al tener un historial como herramienta para la gestión de procesos empresariales (BPM), requiere que los usuarios construyan modelos conceptuales de flujo de trabajo, definan la interfaz de usuario, la estructura de datos. Appian permite las conexiones a sistemas externos y la gestión de roles de usuario, que es un buen acercamiento para la segmentación de formularios por roles. Finalmente para el paso a producción requieren instalar la solución en un servidor web que también puede presentarse como un obstáculo.
- Wavemaker disponible en (WaveMaker, 2022): Es una plataforma de código bajo altamente escalable, fácil de adoptar e integrar para potenciar plataformas empresariales y desarrollar aplicaciones. permite a los usuarios crear, dentro de los límites de una arquitectura de aplicación genérica, aplicaciones de complejidad baja o moderada. Pero en cualquier caso se requiere una comprensión básica de desarrollo de software, pues si es necesario implementar requisitos avanzados, generalmente se requiere estar familiarizado con tecnologías como Java, JavaScript, SQL, HTML, CSS, SOAP y una variedad de API comunes. Finalmente esta plataforma no permite definir roles.
- Pega disponible en (Pega, 2022): Es una plataforma No-code con la cual se realiza la configuración y actualización de aplicaciones sin codificación ni secuencias de comandos. Esta plataforma también permite la definición y cambios de procesos, así como el diseño y la modificación de la interfaz de usuario, que es una buena aproximación para la modificación de los formularios después de haber sido usados y sin afectar la información que ya ha sido recolectada. Sin embargo al ser una solución para la automatización de procesos robóticos, así como para la gestión de procesos, la complejidad para construir una solución

es grande independientemente de la complejidad del problema. Finalmente se requiere del levantamiento de servidor web para llevar la solución a producción.

- Quickbase disponible en (Quickbase, 2022): es una plataforma flexible No-code que permite resolver problemas mediante la creación sencilla de soluciones personalizables. Quickbase Permite a los usuarios configurar entornos de aplicaciones simples y administrar los datos, sin necesidad de comprender los detalles de las tecnologías de base de datos. brindan soporte a distintos dispositivos como computadoras personales, tabletas y teléfonos inteligentes que podría ayudar en la recolección de información en campo. Para su paso a producción se requiere de la configuración de un ambiente productivo.
- Bonita disponible en (Bonitasoft, 2022): Esta plataforma utiliza el estándar para el modelado de procesos de negocio (BPM). Ofrece herramientas de programación visual para no programadores y entornos de desarrollo para que los programadores puedan extender la plataforma y personalizar las aplicaciones a múltiples niveles. Sin embargo Bonita se caracteriza por ofrecer muchas opciones de implementación que a su vez la vuelve más compleja de implementar. Por otro lado queda en duda la recolección de información en campo pues esto no esta soportado en sus características.

Pulir el marco teórico, parece un glosario, por las viñetas y porque hace falta un poco de profundidad en las definiciones.

En el esatdo del arte es neceasrio definir los criterios que permitirán comparar las soluciones que ya existen y presentar una clasificación de las soluciones que ya existen de acuerdo a esos criterios para hacer visible que la solución a la que se está apuntando puede cubrir las deficiencias de las demás soluciones

Referencias

- Appian (2022). Low-code platform for workflow, automation process mining. <https://appian.com/>.
- Bello, S. A., Oyedele, L. O., Akinade, O. O., Bilal, M., Delgado, J. M. D., Akanbi, L. A., Ajayi, A. O., and Owolabi, H. A. (2021). Cloud computing in construction industry: Use cases, benefits and challenges. *Automation in Construction*, 122:103441.
- Bock, A. C. and Frank, U. (2021). Low-code platform. *Business and Information Systems Engineering*, 63:733–740.
- Bonitasoft (2022). Plataforma open-source de automatización de procesos de negocio. <https://es.bonitasoft.com/>.
- Cisneros-Caicedo, A. I. J., III, J. J. U.-C., Guevara-García, A. I. F., and Garcés-Bravo, J. I. E. (2022). Técnicas e instrumentos para la recolección de datos que apoyan a la investigación científica en tiempo de pandemia. *Dominio de las Ciencias*, 8:1165–1185.
- Davoudian, A., Chen, L., and Liu, M. (2018). A survey on nosql stores. *ACM Computing Surveys (CSUR)*, 51.
- Galindo, E. Z. (2019). Modelos de madurez digital en pymes-caso de estudio de una pyme de telecomunicaciones de colombia.
- Gobierno (2022). Transformación digital. <https://www.centrosdetransformaciondigital.gov.co/695/w3-channel.html>.
- Hervás, D. R. (2020). Tecnologías low-code y no-code: Un caso práctico para estudiar su potencial y limitaciones.
- Hylton, D., Sung, S., and Xie, C. (2021). Adopting no-code methods to visualize computational thinking.

- Martinelli, E. M., Farioli, M. C., and Tunisini, A. (2021). New companies' dna: the heritage of the past industrial revolutions in digital transformation. *Journal of Management and Governance*, 25:1079–1106.
- Mendix (2022). Welcome to mendix docs. <https://docs.mendix.com/>.
- Microsoft (2022). ¿qué es power apps? - power apps. <https://docs.microsoft.com/es-es/power-apps/powerapps-overview>.
- Mohammed, C. M. and Zeebaree, S. R. (2021). Sufficient Comparison Among Cloud Computing Services: IaaS, PaaS, and SaaS: A Review. *International Journal of Science and Business*, 5(2):17–30.
- Moskal, M. (2021). No-code application development on the example of logotec app studio platform. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, 11:54–57.
- Pega (2022). Software para arrasar con la complejidad del negocio. <https://www.pega.com/es>.
- Quickbase (2022). The 1 no-code platform for managing unique projects. <https://www.quickbase.com/>.
- WaveMaker (2022). Enterprise-grade low-code solution for the java world. <https://www.wavemaker.com/>.