

## **Prototipo de aplicación móvil con Ionic & .Net core para Escolme**

### **Adrián Gallego Ospina**

Administración de Sistemas Informáticos, Institución Universitaria Escolme, Medellín, aagallegoo@escolme.edu.co

### **Daniel Londoño Sánchez**

Administración de Sistemas Informáticos, Institución Universitaria Escolme, Medellín, dflondonos@escolme.edu.co

### **Juan Camilo Betancur**

Administración de Sistemas Informáticos, Institución Universitaria Escolme, Medellín, cbetancurm@escolme.edu.co

### **Christian Hernán Obando Ibarra**

Docente, Institución Universitaria Escolme, Medellín, Colombia, chobandoi@escolme.edu.co

**Recibido:** 06/06/2020 - **Aceptado:** 24/06/2020 - **Publicado:** 31/07/2020

## **RESUMEN**

La Institución Universitaria Escolme de carácter privado ofrece servicios educativos, en esta se llevan a cabo múltiples eventos para la comunidad universitaria por tanto necesita mejorar la gestión de este proceso que actualmente se realiza de manera manual, sistematizando de manera óptima y a bajo costo. Este trabajo tiene por objeto crear un prototipo de aplicación móvil con Ionic & Web Api .Net Core, implementar la arquitectura base propuesta en este prototipo en un ambiente desarrollo, verificando este como una posible solución de software para la gestión de eventos de la institución con alta disponibilidad, escalabilidad y usabilidad antes de presentar una propuesta a la institución. Ionic es un framework que permite crear aplicaciones móviles para los sistemas operativos Android, IOS con un solo código fuente escrito en JavaScript bajo el patrón de arquitectura MVC (cliente), .Net Core es un framework que con su Web Api permite exponer los métodos que serán consumidos por la aplicación móvil, además de facilitar la escalabilidad de la aplicación debido a su capacidad real de funcionar en múltiples plataformas (Linux o macOS) con su código escrito en C# bajo el patrón de arquitectura n capas (servidor), ambos marcos de trabajo son de código abierto, por último para la gestión de los bases de datos se propone utilizar SQL Server Express Edition bajo licencia MIT.

**Palabras clave:** código abierto; escalabilidad; licenciamiento; sistema operativo; software.

## ABSTRACT

The private Escolme University Institution offers educational services, in which multiple events will be held for the university community, therefore it needs to improve the management of this process, which is currently carried out manually, systematizing optimally and at low cost. The purpose of this work is to create a prototype of a mobile application with Ionic and Web Api .Net Core, implement the base architecture proposed in this prototype in a development environment, verify this as a possible software solution for the management of events of the institution with high availability, scalability and usability before presenting a proposal to the institution. Ionic is a framework that allows creating mobile applications for Android operating systems, IOS with a single source code written in JavaScript under the MVC (client) architecture pattern, .Net Core is a framework that with its Web Api allows exposing the methods that they will be consumed by the mobile application, in addition to facilitating the scalability of the application due to its real ability to operate on multiple platforms (Linux or macOS) with its code written in C # under the layered architecture (server) pattern, both frameworks work are open source, finally for database management it is proposed to use SQL Server Express Edition under MIT license.

**Keywords:** open source; scalability; licensing; operating system; software.

## 1. INTRODUCCIÓN

Nuevos tiempos, nuevas tecnologías, nuevos retos, son las características del actual mundo digital, donde los programas informáticos están presentes en todos los aspectos de la vida humana. Esto representa un desafío para las organizaciones sin importar el tamaño o el fin para el que han sido creadas, pues la gestión de procesos a través de programas informáticos se ha convertido en una obligación (Tavarez, 2014).

Con la aparición, el uso masivo y el crecimiento de los dispositivos móviles, especialmente de los Smartphone, las aplicaciones móviles han constituido un ecosistema propio y un potente motor de innovación. La consolidación de las aplicaciones móviles como interfaz dominante en la gestión de contenido permitiendo llegar a mayor número de personas y a su vez ofrecer una mejor experiencia de usuario comparado con las aplicaciones web tradicionales, sin embargo, la creación de una aplicación móvil puede representar costos y tiempos de desarrollo altos (Aguado, Martínez & Cañete-Sanz, 2015; Gómez-Ramírez Valencia-Arias & Duque, 2019).

Este trabajo se realiza con el fin de lograr identificar las herramientas de software que permitan realizar un prototipo y sugerir la futura implementación de una aplicación móvil para la gestión de eventos de la Institución Universitaria Escolme, considerando que en los últimos años son más frecuentes los eventos dentro del centro educativo y que en la actualidad la institución no cuenta con un sistema informático que facilite dicho proceso, es por esto que se propone una aplicación móvil basado en herramientas de código abierto que permitan implementar una solución a bajo costo y mejorar el proceso existente.

Entre las posibles soluciones de código abierto disponible para la creación de una aplicación móvil los que sobresalen son: Xamarin, React JS, Phonegap, Ionic. entre otros.

## 2. MARCO TEÓRICO Y/O ANTECEDENTES

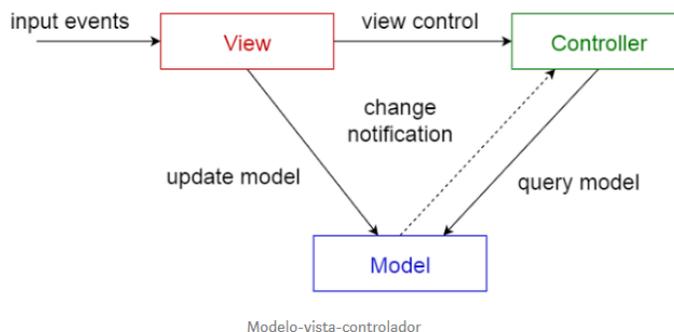
### 2.1. Patrón de diseño de software

Tener buenas prácticas al programar es importante ya que permite llevar el código de una manera formal y una estructura de un proyecto entendible, por eso existen diferentes patrones de diseño.

### 2.2. Patrón MVC

El patrón MVC está estructurado mediante tres componentes: modelo, vista y controlador, los tres están separados en diferentes objetos los cuales no se pueden combinar dentro de una misma clase, estos son:

- a) Modelo: contiene la funcionalidad y los datos básicos.
- b) Vista: muestra la información al usuario y se puede definir más de una vista.
- c) Controlador: maneja la entrada del usuario (Huaman, 2018).



**Ilustración 1. Patrón Vista controlador. Fuente: (Huaman, 2018).**

### 2.3. Patrón N Capas

Es una arquitectura de tipo cliente servidor, esta arquitectura se enfoca en la distribución de roles y responsabilidades de forma jerárquica garantizando una forma muy efectiva y de separación de responsabilidades. Además, permite distribuir el trabajo en una aplicación por niveles, de este modo cada grupo de trabajo es totalmente abstraído.

- Capa de presentación: es la que ve el usuario está presente el sistema al usuario y le comunica información y captura información.
- La capa de negocio: es donde están los programas que se ejecutan y se reciben las peticiones del usuario.
- La capa de datos: esta se encarga de acceder a los datos, formada por uno o más gestores de base de datos, estos realizan todo el almacenamiento de datos y reciben

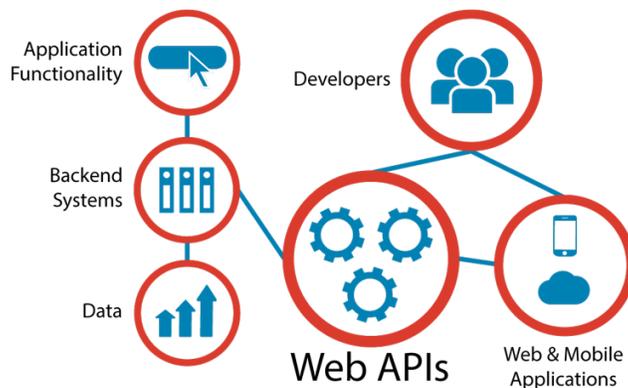
solicitudes de almacenamiento o recuperación de información desde la capa de negocio (Acosta, Álvarez & Gordillo, 2006).



**Ilustración 2. N Capas. Fuente: (Acosta et al., 2006).**

#### 2.4. Web API

Las API en la web se exponen a través de servicios web, que permiten que las aplicaciones clientes obtenga y realicen operaciones con los datos que el servicio expone, existen destinitos servicios web que se caracterizan principalmente por la forma que realizan la comunicación con el cliente y el formato en el que intercambian información. En la actualidad una opción para exponer APIs en la web es mediante servicios REST ASP.NET WEB API, que es un framework que forma parte de ASP.NET MVC y que permite construir APIs WEB habilitadas para RESET (Web Master, 2018).



**Ilustración 3. Web APIs. Fuente: (Web Master, 2018).**

#### 2.5. Frameworks

Cuando se habla de framework se indica de un referente para la programación, es un esquema de trabajo en el cual se establecen esquemas y estructuras las cuales son aprovechadas al momento de programar. Por medio de un framework se puede realizar

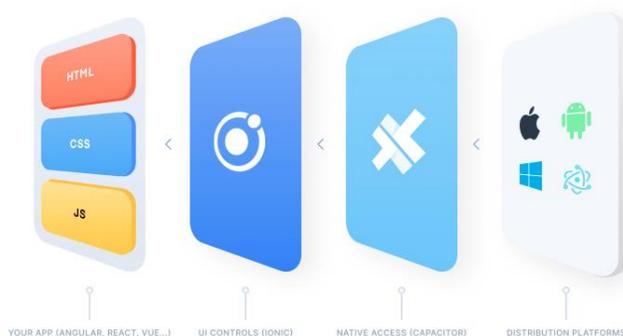
tareas más comunes de una forma muy sencilla sin tener que programar estas desde cero, lo cual podría resultar costoso tanto en tiempo como desarrollo (NeoAttack, s.f.).

En los últimos años se han desarrollado una cantidad de frameworks por diferentes comunidades para facilitar diferentes tipos de tareas tanto en la parte del frontend como en la parte del backend. Entre tantos framework se puede encontrar los más reconocidos como son: .Net Framework, Meteor, Ionicframework, Angular, Bootstrap, entre otros.

## 2.6. Ionic frameworks

Ionic framework permite programar aplicaciones móviles para los sistemas operativos Android, Ios y Web con un solo código fuente, se centra en la parte del Front-End de la aplicación creando aplicaciones ricas en contenido y amigables al usuario final.

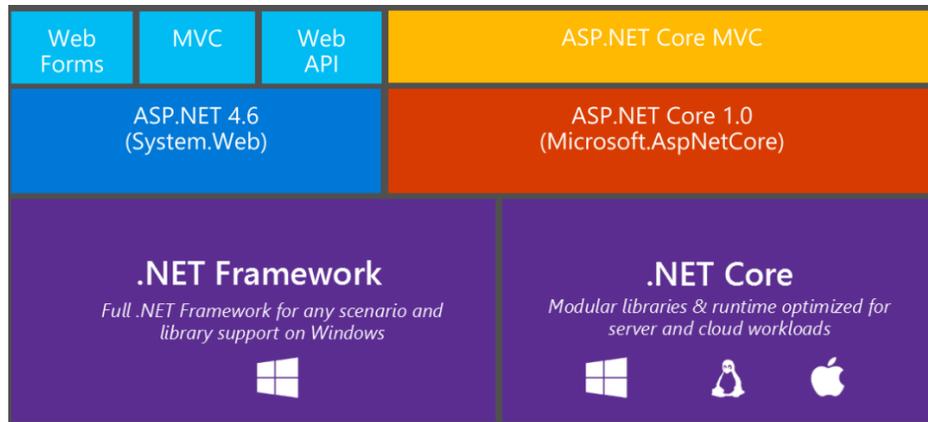
Ionic está desarrollado en su base con angular y cordova como transpilador de código, el cual permite que una aplicación creada con HTML, CSS y JavaScript pueda ser interpretada por un dispositivo móvil, vio la luz en 2013 con la única intención de que desarrolladores pudieran crear aplicaciones móviles híbridas (se refiere a aplicaciones con HTML, CSS y JavaScript) con la particularidad y beneficios de los dos framework sobre los que fue construida. Una de las principales ventajas de trabajar con Ionic es que aprovecha todos los plugins (Hardware, software, imágenes, texto, códigos QR, etc) del marco de desarrollo móvil Cordova (Quality devs, 2019).



**Ilustración 4. Ionic. Fuente:** (Quality devs, 2019).

## 2.7. Net Core

ASP.NET Core es un nuevo framework de código abierto y multiplataforma para la creación de aplicaciones modernas conectadas a Internet, como aplicaciones web y APIs Web. Se diseñó para proporcionar un framework de desarrollo optimizado para las aplicaciones que se implementan tanto en la nube como en servidores dedicados en las instalaciones del cliente. Se pueden desarrollar y ejecutar aplicaciones ASP.NET Core en Windows, Mac y Linux (Recio, 2017).



**Ilustración 5. ASP.NET Core. Fuente: (Recio, 2017).**

## 2.8. IDE (Integrated Development Environment)

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk u Objective-C. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. Este es el caso de Eclipse, que mediante plugins se le puede añadir soporte de lenguajes adicionales (Software de programación, s.f.).

Los componentes que debe tener los IDEs son:

- Un editor de texto.
- Un compilador.
- Un intérprete.
- Herramientas de automatización.
- Un depurador.
- Posibilidad de ofrecer un sistema de control de versiones.
- Factibilidad para ayudar en la construcción de interfaces gráficas de usuarios.

## 2.9. Visual Studio Community

Un completo IDE extensible y gratuito para crear aplicaciones modernas para Windows, Android e iOS, además de aplicaciones web y servicios en la nube (Obadel, 2015).

## 2.10. Bases de datos

En cualquier tipo de actividad, se genera una cantidad indeterminada de información, la cual, según su importancia, es necesario almacenarla para su uso posterior. El lugar donde se almacena esta información se conoce con el nombre de “Base de Datos”, este lugar puede ser tangible (físico) o intangible (lógico) (Monsalve & Narváez, 2014).

## 2.11. SQL Server Express

SQL Server 2019 Express es una edición gratuita de SQL Server ideal para el desarrollo y la producción de aplicaciones de escritorio, aplicaciones web y pequeñas aplicaciones de servidor, es un gestor de base de datos el que permite la administración de esta de una

forma más amigable y sencilla permitiendo la escalabilidad en la evolución del proyecto (Microsoft, s.f.).

### **3. METODOLOGÍA O DESCRIPCIÓN DEL PROCESO**

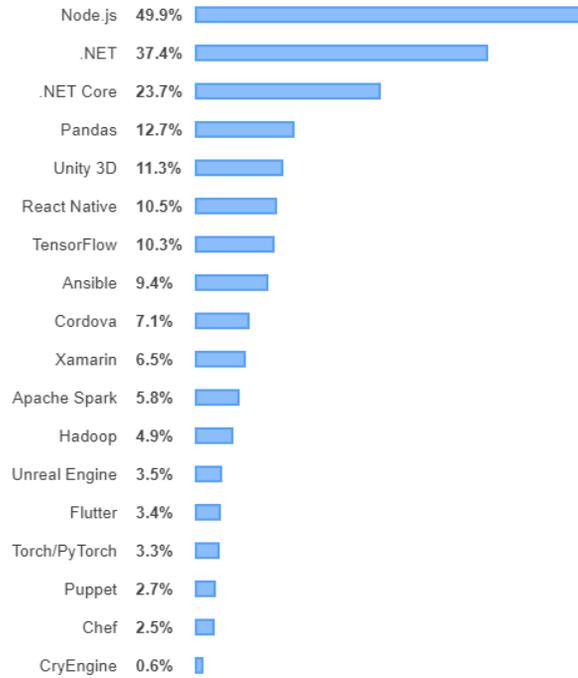
La elaboración de este artículo se realizó en cuatro fases, las cuales se describen a continuación:

#### **Fase 1. Exploratoria**

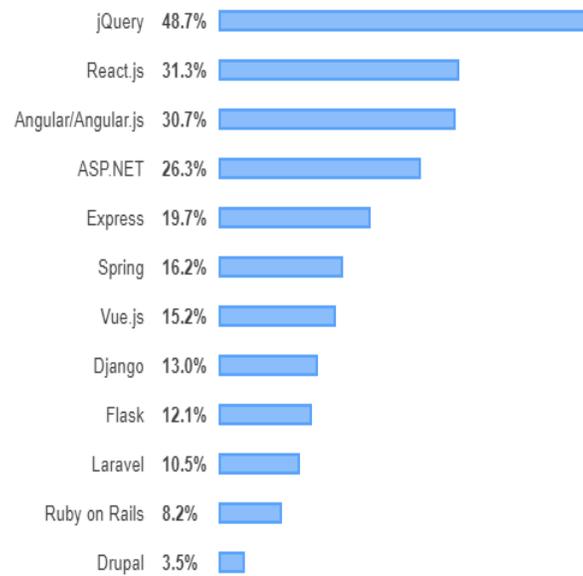
En esta fase se consultaron múltiples artículos de sitios web para programadores como son: stackoverflow y github, y fuentes como google académico. En esta se logró justificar los elementos para la elaboración de un prototipo para el desarrollo de un software a la medida y a su vez determinar qué lenguajes de programación, frameworks, patrones de arquitectura de software y gestor de base de datos deberían ser considerados para realizar el prototipo y la futura implementación de este.

Para obtener el resultado deseado en esta fase fue necesario identificar las herramientas open source más utilizadas por los desarrolladores de software, a la hora de realizar la implementación de una aplicación móvil híbrida de idénticas características a la del prototipo que se presenta en este artículo.

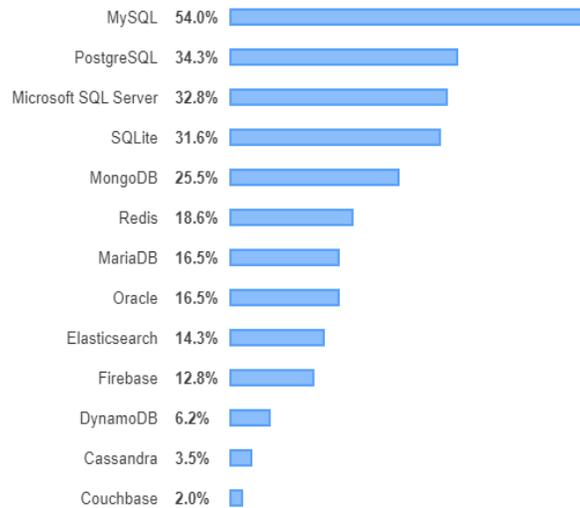
Los ítems de mayor relevancia en esta lista frameworks backend son Node.js, .Net y .Net Core (ilustración 6). Mientras que en los frameworks Webs del lado del cliente la lista la encabezan jquery y angular (ilustración 7), en cuanto se refiere a la gestión de base de datos MySQL, PostgreSQL y Microsoft SQL Server son las soluciones más ranqueadas (ilustración 8), por último, las plataformas que sobresalen para la instalación de los proyectos de software son Linux, Windows, Android (ilustración 9) (Stack overflow, 2019).



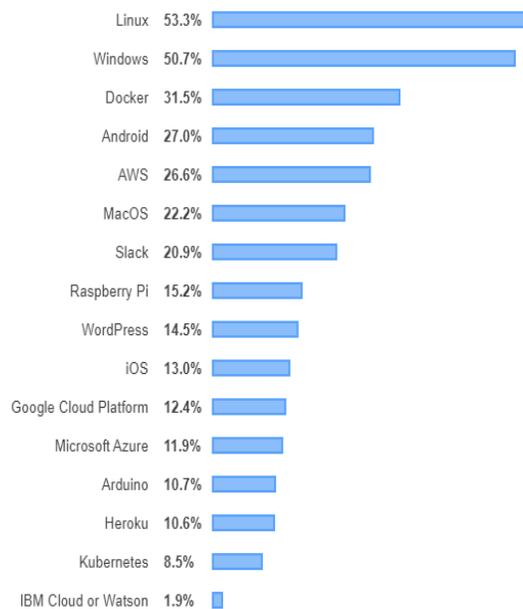
**Ilustración 6. Frameworks backend Fuente: (Stack overflow, 2019).**



**Ilustración 7. Frameworks Webs. Fuente: (Stack overflow, 2019).**



**Ilustración 8. Base de datos. Fuente: (Stack overflow, 2019).**



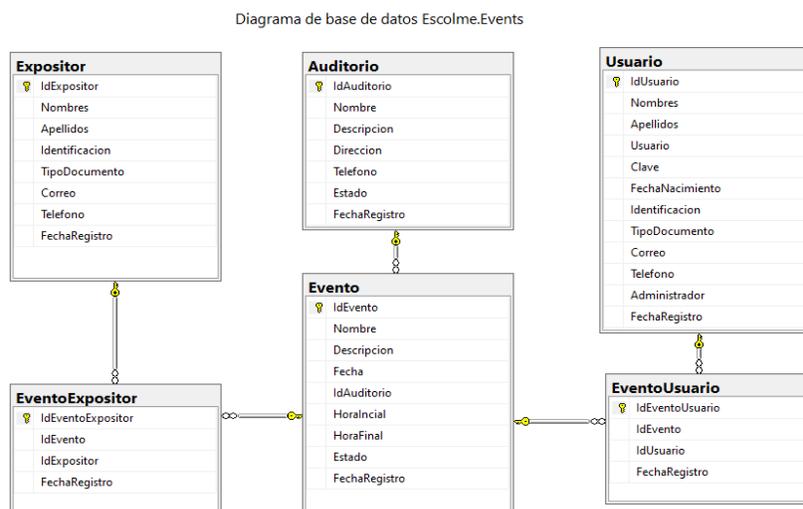
**Ilustración 9. Plataformas. Fuente: (Stack overflow, 2019).**

Basado en el resultado de la exploración de acuerdo a los datos evidenciados en las anteriores ilustraciones se decidió utilizar el STACK de .Net y el SDK De Ionic para realizar el prototipo de la aplicación móvil y se sugiere las mismas soluciones de software para realizar una futura implementación.

## **Fase 2. Creación y diseño del Modelo de base de datos**

En esta fase se analizaron los requisitos básicos para la elaboración del proyecto, identificando los modelos bases para el almacenamiento de la información deseada, a partir

de este análisis se generó el modelo de la base de datos propuesta para almacenar dicha información. Se presenta el modelo entidad relación a continuación.

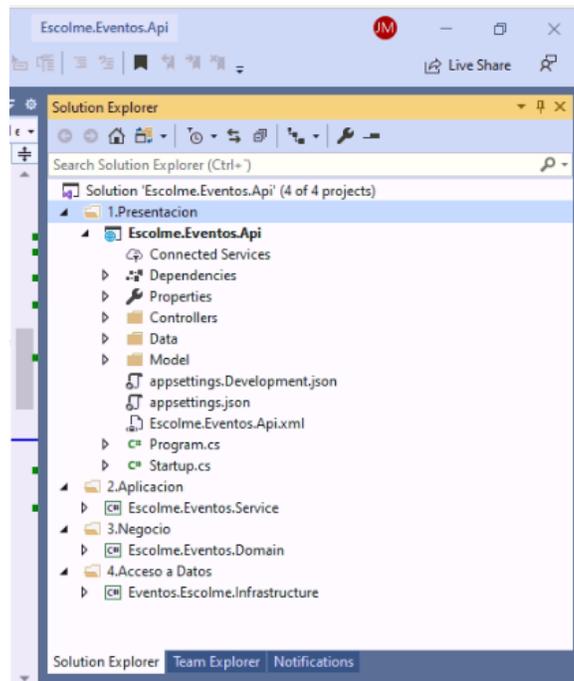


**Ilustración 10. Diagrama de base de datos Escolme. Fuente: elaboración propia en el gestor de bases de datos SQL Server Express 2019.**

### Fase 3. Creación del prototipo Backend de la aplicación

Cuando se habla del backend de la aplicación, hace referencia al proceso encargado de administrar toda la información y la lógica de negocio de esta. Por medio de controladores se expone los métodos y se interactúa con el cliente (App móvil) permitiendo la transferencia de información de extremo a extremo.

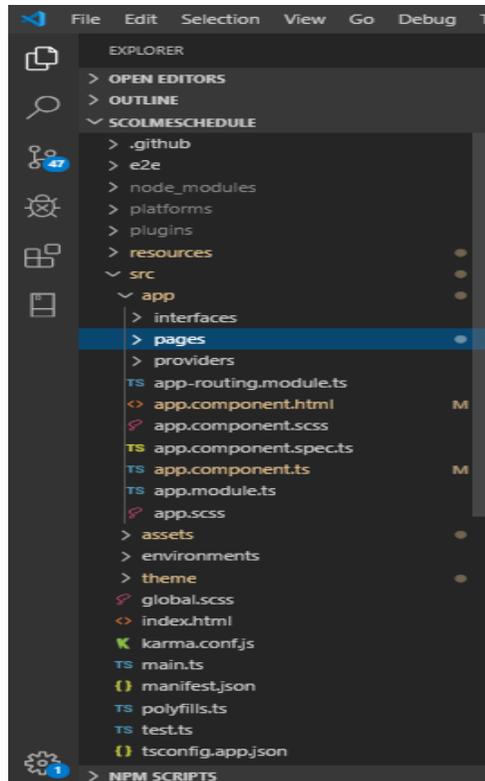
Para este prototipo, se elaboró una web Api construida en .net Core Framework bajo la arquitectura N-Capas implementando el modelo MDD (Modelo dirigido por dominio), el cual da la estructura para distribuir las capas de la manera adecuada, para que el código sea limpio y tenga las mejores prácticas de desarrollo para su escalabilidad.



**Ilustración 11. Código. Fuente: elaboración propia en Visual Studio Community Edition 2019.**

#### **Fase 4. Creación del prototipo Frontend de la aplicación**

Cuando se habla del Frontend de la aplicación, esto hace referencia a la aplicación móvil como tal, esta se encarga de presentar toda la información al usuario final mostrando de una manera amigable los eventos y la información deseada. Para este prototipo se elaboró un proyecto Ionic, el cual crea la estructura principal para el desarrollo de la parte visual del prototipo. La aplicación se comunica por medio de proveedores con el Api, el cual expone los métodos para obtener la información que se desea visualizar.



**Ilustración 12. Código. Fuente: elaboración propia en Visual Studio Code.**

#### **4. ANÁLISIS DE RESULTADOS O HALLAZGOS**

Como parte final del prototipo realizado, se presenta la captura de los resultados obtenidos de la implementación del proyecto.

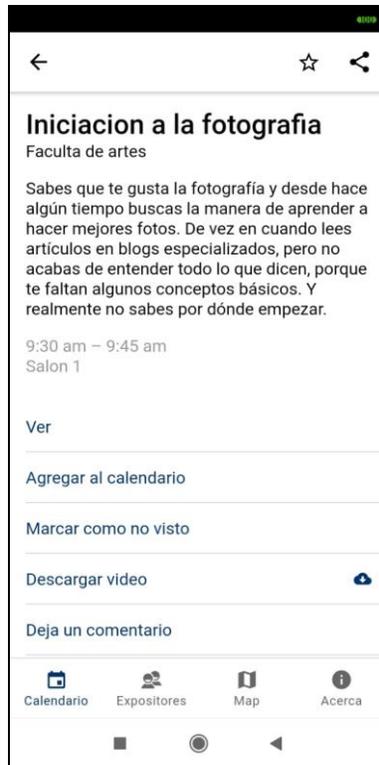
Con la arquitectura presentada en los puntos anteriores de la metodología, se deja un inicio para un desarrollo de la aplicación con las mejores prácticas investigadas e implementadas por la comunidad y el mundo real.



**Ilustración 13. Bienvenida. Fuente: elaboración propia.**



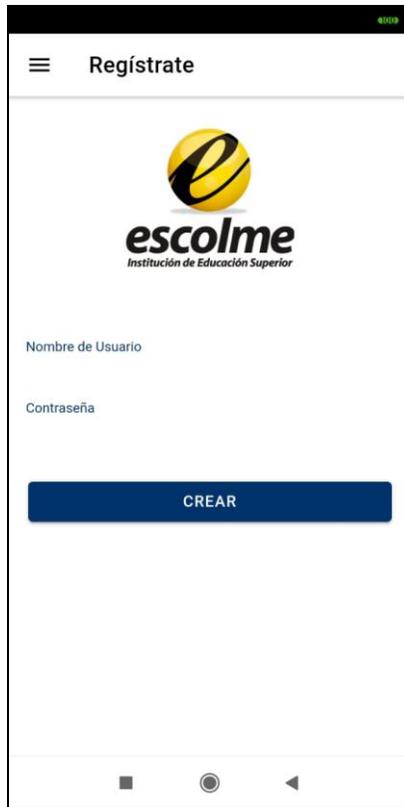
**Ilustración 14. Sesiones. Fuente: elaboración propia.**



**Ilustración 15. Iniciación a la fotografía. Fuente: elaboración propia.**



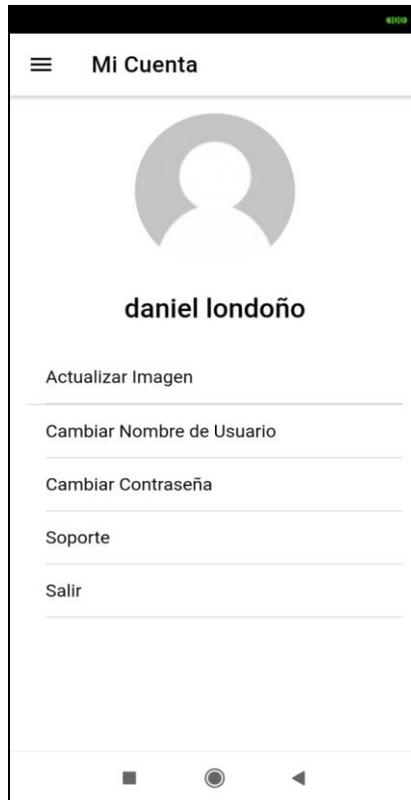
**Ilustración 16. Ingresar. Fuente: elaboración propia.**



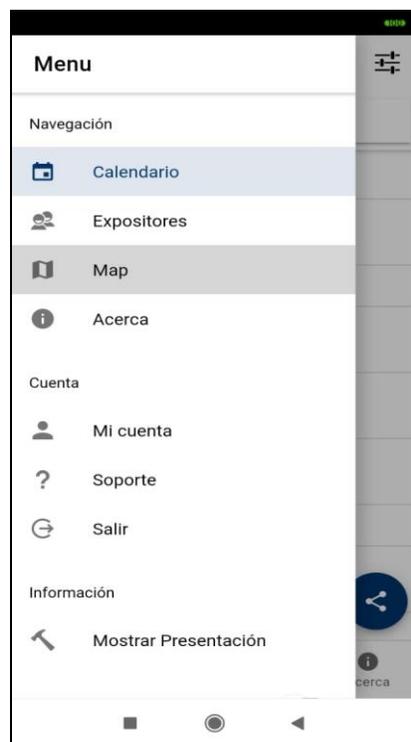
**Ilustración 17. Registrarse. Fuente: elaboración propia.**



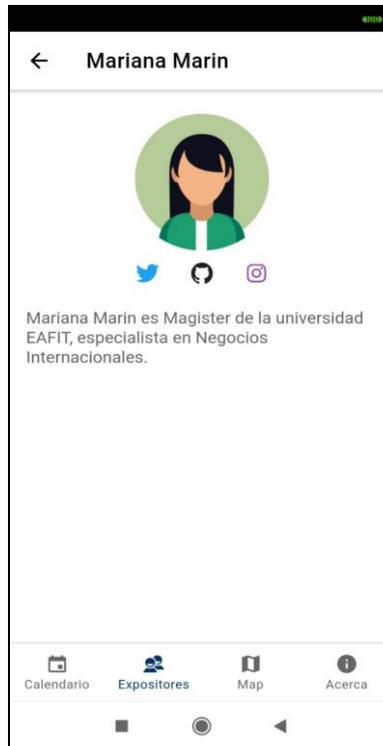
**Ilustración 18. Soporte. Fuente: elaboración propia.**



**Ilustración 19. Mi cuenta. Fuente: elaboración propia.**



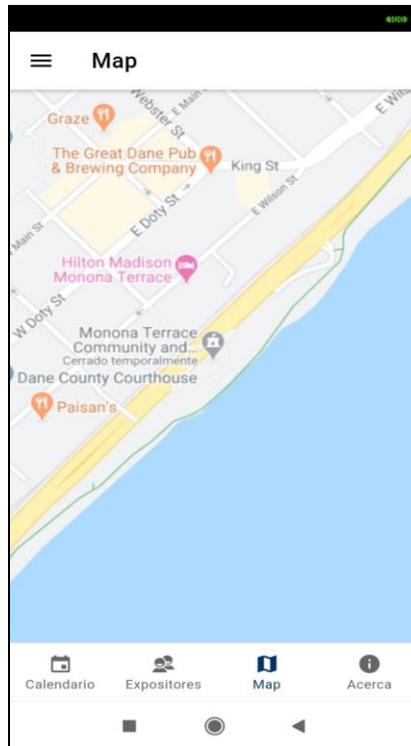
**Ilustración 20. Menú. Fuente: elaboración propia.**



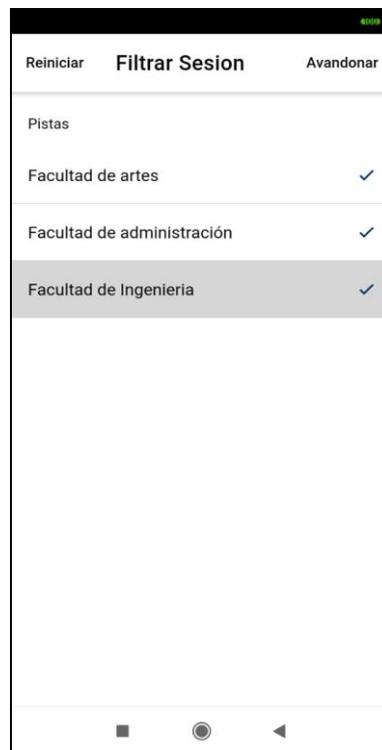
**Ilustración 21. Expositores. Fuente: elaboración propia.**



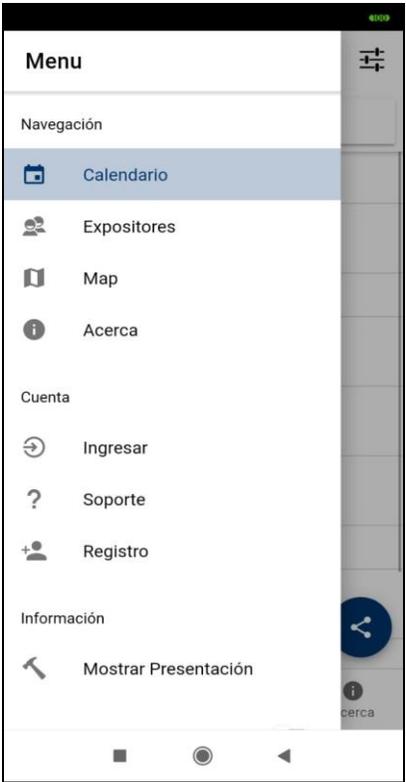
**Ilustración 22. Acerca. Fuente: elaboración propia.**



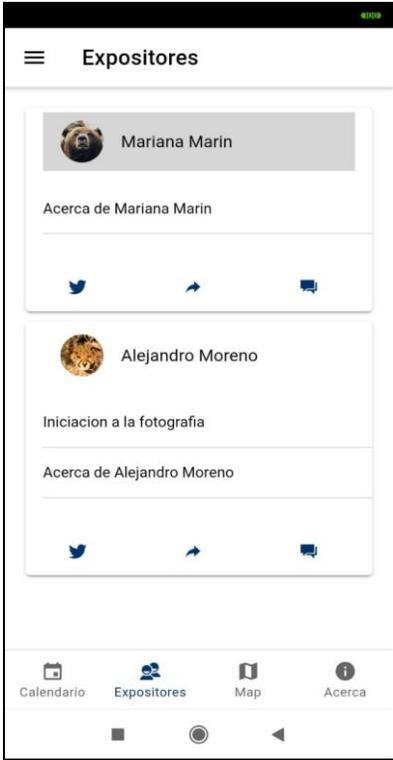
**Ilustración 23. Mapa. Fuente: elaboración propia.**



**Ilustración 24. Filtrar sesión. Fuente: elaboración propia.**



**Ilustración 25. Menú. Fuente: elaboración propia.**



**Ilustración 26. Expositores. Fuente: elaboración propia.**

## 5. CONCLUSIONES

Las mejores herramientas por el momento en el mercado para construir un aplicativo móvil son .Net Core y Ionic framework, ya que facilitan el desarrollo y la implementación, garantizando calidad, seguridad y confianza en el funcionamiento y la estabilidad del código.

Se implementa este prototipo para proponer a Escolme como puede mejorar la gestión de sus eventos, ya que a medida que va evolucionando la tecnología y las telecomunicaciones, se debe adaptar a los cambios que estas ingieren, presentando herramientas y soluciones que permitan facilitar la administración de la información de una forma sencilla para con esta realizar análisis de los datos obtenidos y tomar las mejores decisiones en presentación de eventos para el campus educativo.

Se recomienda que este proyecto pueda ser tomado como base referente para terminar su implementación por otro equipo de proyecto de grado. Además, que la Institución Universitaria Escolme, se apoye fuertemente en la tecnología para mejorar los procesos, incluida la gestión de eventos internos que de alguna manera ayuda a cuidar el medio ambiente, al evitar el registro de datos en papel y el desplazamiento de las personas a las sedes, estas herramientas son una óptima solución para obtener tener información más ágil y rápida en tiempo real.

## 6. REFERENCIAS

Acosta, E., Álvarez, J., & Gordillo, A. (2006). Arquitecturas en n-Capas: Un Sistema Adaptivo. *Polibits*, (34), 34-37. Recuperado de <https://www.redalyc.org/pdf/4026/402640447007.pdf>

Aguado, J. M., Martínez, I. J., & Cañete-Sanz, L. (2015). Tendencias evolutivas del contenido digital en aplicaciones móviles. *El profesional de la información*, 24(6), 787-795. <http://dx.doi.org/10.3145/epi.2015.nov.10>

Gómez-Ramírez, I., Valencia-Arias, A., & Duque, L. (2019). Approach to M-learning Acceptance Among University Students. *The International Review of Research in Open and Distributed Learning*, 20(3).

Huaman, W. (2018). Los 10 patrones comunes de arquitectura de software. Recuperado de <https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b>

Microsoft. (s.f). Prueba SQL Server on-premises o en el cloud. Recuperado de <https://www.microsoft.com/es-es/sql-server/sql-server-downloads>

Monsalve, C. & Narváez, J. (2014). *Desarrollo de una solución de inteligencia de negocios para apoyar la toma de decisiones en la dependencia de recursos educativos de la universidad del Magdalena*. (Trabajo de grado, Universidad del Magdalena). Recuperado de <http://repositorio.unimagdalena.edu.co/jspui/bitstream/123456789/3519/1/IS-00119.pdf>

NeoAttack. (s.f.). Concepto de Framework. Recuperado de <https://neoattack.com/neowiki/framework/>

Obadel. (2015). Comunidad de Visual Studio. Recuperado de <https://obadelblog.wordpress.com/2015/12/01/comunidad-de-visual-studio-un-completo-ide-extensible-y-gratuito-para-crear-aplicaciones-modernas-para-windows-android-e-ios-ademas-de-aplicaciones-web-y-servicios-en-la-nube/>

Quality devs. (2019). Qué es Ionic y por qué te interesa conocerlo si eres desarrollador web. Recueprado de <https://www.qualitydevs.com/2019/05/31/que-es-ionic-desarrollador-web/>

Recio, S. (2017). Lo que necesita saber sobre el núcleo ASP.net. Recuperado de <https://codeburst.io/what-you-need-to-know-about-asp-net-core-30fec1d33d78>

Software de programación. (s.f.). Entorno de desarrollo integrado. Recuperado de <https://sites.google.com/site/softwaredeprogramacion2/entorno-de-desarrollo-integrado>

Stack overflow. (2019). Developer Survey Results 2019. Recuperado de <https://insights.stackoverflow.com/survey/2019#technology>

Tavarez, J. (2014). El software y su importancia. Recuperado de <https://eldia.com.do/el-software-y-su-importancia/>

Web Master. (2018). What is Web API? Recuperado de <http://www.dselva.co.in/blog/what-is-web-api/>