



Revista UIS Ingenierías

ISSN: 1657-4583

ISSN: 2145-8456

revistaingenierias@uis.edu.co

Universidad Industrial de Santander

Colombia

Báez-Pérez, Carmen Inés; Clunie-Beaufond, Clifton Eduardo  
**El modelo tecnológico para la implementación de un proceso de  
educación ubicua en un ambiente de computación en la nube móvil**

Revista UIS Ingenierías, vol. 19, núm. 4, 2020, Octubre-, pp. 77-88

Universidad Industrial de Santander

Bucaramanga, Colombia

DOI: <https://doi.org/10.18273/revuin.v19n4-2020007>

Disponible en: <https://www.redalyc.org/articulo.oa?id=553768213008>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica Redalyc

Red de Revistas Científicas de América Latina y el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso  
abierto

# El modelo tecnológico para la implementación de un proceso de educación ubicua en un ambiente de computación en la nube móvil

## Technological model for the implementation of a ubiquitous learning process on mobile cloud computing

Carmen Inés Báez-Pérez <sup>1</sup>, Clifton Eduardo Clunie-Beaufond <sup>2</sup>

<sup>1</sup> Giprocas, Programa de Ingeniería de Sistemas, Universidad de Boyacá, Colombia. Orcid: 0000-0001-5084-1631.  
Correo electrónico: cibaez@uniboyaca.edu.co

<sup>2</sup> Ingeniería de Software, Facultad de Ingeniería de Sistemas Computacionales, Universidad Tecnológica de Panamá, Panamá. Orcid: 0000-0002-6642-0900. Correo electrónico: clifton.clunie@utp.ac.pa

Recibido: 22 enero, 2020. Aceptado: 4 mayo, 2020. Versión final: 23 julio, 2020.


### Resumen

Este artículo presenta una propuesta de un modelo tecnológico que buscar servir de punto de partida para implementaciones de soluciones educativas, que apoyen procesos de formación por medio de la educación ubicua; este modelo se implementó sobre una plataforma tecnológica basada en Computación en la Nube Móvil (Mobile Cloud Computing - MCC), que se soporta sobre tecnologías en la nube computacional (Cloud Computing). Para validar la aplicación del modelo, se implementó un caso de estudio, el cual se basa en el tema de Resolución de Problemas Matemáticos, tema que se desarrolla en asignaturas de primeros semestres a nivel universitario. La implementación se realizó mediante el desarrollo de una app (aplicación para dispositivos móviles) utilizando la plataforma Microsoft Azure, como plataforma de Computación de Nube Móvil y las herramientas de desarrollo C# y ASP.NET. El manejador de base de datos usado es MySQL.

**Palabras clave:** educación ubicua; computación en la nube; computación móvil en la nube; aprendizaje basado en problemas; resolución de problemas matemáticos.

### Abstract

This paper presents a proposal for a technological model that seeks to serve as a starting point for implementations of educational solutions that support training processes through ubiquitous education; This model was implemented on a technological platform based on Mobile Cloud Computing (MCC), which is supported on cloud computing technologies (Cloud Computing). To validate the application of the model, a case study was implemented, which is based on the topic of Mathematical Problems Resolution, an issue that is developed in subjects of first semesters at the university level. The implementation was carried out by developing an app (application for mobile devices) using the Microsoft Azure platform, as a Mobile Cloud Computing platform and the C # and ASP: NET development tools. The database manager used is MySQL.

ISSN impreso: 1657 - 4583. ISSN en línea: 2145 – 8456, CC BY-ND 4.0 

Como citar: C. Báez, C. Clunie, “El modelo tecnológico para la implementación de un proceso de educación ubicua en un ambiente de computación en la nube móvil,” *Rev. UIS Ing.*, vol. 19, no. 4, pp. 77-88, 2020, doi: <https://doi.org/10.18273/revuin.v19n4-2020007>

**Keywords:** ubiquitous learning; cloud computing; mobile cloud computing; problem-based learning; solving math problems.

## 1. Introducción

La computación en la nube ha tenido un gran auge en los últimos años, debido a las facilidades que tiene [1], así mismo, tiene la capacidad de ofrecer recursos de procesamiento y almacenamiento de acuerdo con la demanda de cada usuario, los cuales pueden ser accedidos por medio de internet, sin que los usuarios necesiten conocer detalles técnicos acerca de la plataforma computacional [2], con lo cual se tiene la ventaja de no requerir personal experto para su administración. Otra de las ventajas de este tipo de computación, es pago por demanda [3], es decir, a medida que se requieren los recursos se paga por ellos y en caso de disminuir su demanda, también va disminuyendo su costo.

La incorporación de dispositivos móviles en las actividades cotidianas también ha logrado una importante consolidación, por los servicios que presta y las facilidades que ofrece para disminuir la complejidad de algunas tareas. Estos aspectos, la computación en la nube y los dispositivos móviles, sumados a la disponibilidad del uso de internet, han hecho que cada día se haga más necesario el uso de herramientas de software, particularmente, de aplicaciones móviles que apoyen el desarrollo de las tareas diarias. De acuerdo con el [4], el número total de suscripciones móviles alrededor del mundo para el periodo julio- septiembre de 2019 fue de 8 mil millones, lo cual demuestra la penetración de estas tecnologías en la vida cotidiana.

Por las razones mencionadas, aparece otro concepto que agrupa lo anterior y es la Computación en la Nube Móvil, la cual busca aprovechar las características de la Computación en la Nube y suplir o disminuir las desventajas que tienen los dispositivos móviles como son: capacidad de almacenamiento, capacidad de procesamiento, entre otras. Así mismo, el desarrollo de apps (aplicaciones para dispositivos móviles) en áreas como: salud, juegos, aprendizaje, entre otros, basados en la computación en la nube móvil están en aumento cada día [5].

Es por ello que, la dinámica en el uso de este tipo de dispositivos y tecnologías ha permitido la incorporación de las mismas de una manera casi obligatoria en diferentes aspectos de la vida diaria, tal es el caso de las actividades educativas, particularmente, dicha disponibilidad ha hecho que se creen nuevos escenarios donde el docente tenga la capacidad para incorporar estos recursos dentro del proceso de formación. Una forma de

implementación de aplicaciones móviles en los procesos de aprendizaje se ha identificado como educación ubicua (ubiquitous learning), la cual está teniendo un gran desarrollo y ha logrado su incorporación en diferentes escenarios de formación, permitiendo la incorporación de nuevas características y funcionalidad que permiten enriquecer los recursos digitales que se incorporan en el proceso de formación.

La educación ubicua está definida como aprender en cualquier momento y en cualquier lugar, basado en el contexto y mediado por el uso de Tecnologías de la Información y las Comunicaciones [6], [7].

A nivel mundial, ya se tienen experiencias sobre desarrollo de educación ubicua, como los que se describen en:

Computación en la nube brinda características de seguridad, escalabilidad, confiabilidad y acceso global, lo cual hoy en día son grandes ventajas, dado que permite la implementación de proyectos a gran escala, sin necesidad de tener una infraestructura propietaria y personal experto para su administración. Por otra parte, la posibilidad que ofrece de disponer de recursos computacionales como procesamiento, almacenamiento, herramientas de desarrollo, según la demanda o necesidad del proyecto hacen que esta sea una tecnología apropiada para la implementación de diferentes tipos de proyectos, entre ellos, proyectos de tipo académico, dado que se pueden masificar a gran escala.

El modelo que se planteó en la investigación se ha desarrollado desde el punto de vista teórico, donde se define la estructura general de la arquitectura del mismo, con características generales y desde el punto de vista de una implementación de un caso de estudio, donde de manera concreta, se realiza la implementación para un caso específico. En el caso particular de la implementación del caso de estudio, se tomó como referente un tema académico que se desarrolla en los primeros semestres de programas de pregrado de algunas universidades, tal es el caso de la Resolución de Problemas, refiriéndose específicamente a Problemas Matemáticos.

## 2. El modelo

El modelo teórico que se plantea presenta una propuesta donde se analizan diferentes aspectos tecnológicos que se requieren para la implementación de una App Educativa, que integre diferentes aspectos tanto pedagógicos como

tecnológicos, partiendo de la definición identificada por [7] sobre los elementos que componen el aprendizaje ubicuo, entre los cuales se destacan: nivel de formación, modelos de aprendizaje, áreas de conocimiento, tecnologías computacionales y características.

A continuación, se presenta la propuesta del middleware que soportará la implementación de la app educativa.

## 2.1. Perspectiva del modelo desde el punto de vista del middleware

El middleware que se propone, se estructura en tres capas, 1) capa de usuario final, esta es la capa superior, en la cual se encuentran los estudiantes y el docente; los estudiantes descargan la app en su dispositivo móvil (celular o tableta) y se conectan a la misma mediante una conexión con acceso a internet; 2) la capa intermedia, o Capa Cloud, es la capa que permite hacer el enlace entre el usuario y la base de datos, a esta capa se accede mediante una API, que es la que consume el servicio y genera la conexión con la base de datos; 3) la capa de recursos, en esta capa se encuentran todos los recursos necesarios para el funcionamiento de la app, se encuentran: la API, la conexión a la red, los recursos de procesamiento, almacenamiento y las herramientas de desarrollo, que permiten la implementación de la App. Lo anterior, se puede observar en la Fig. 1.

Como se presenta en la figura 1, el middleware del modelo propuesto, se compone de tres capas principales, las cuales se describen a continuación.

La primera es la capa de recursos, la cual proporciona todos los recursos y servicios computacionales necesarios (API (Application Program Interface, servicios de redes de comunicaciones, recursos de procesamiento, base de datos y herramientas de desarrollo de software) para la implementación de la App Educativa. En este caso se obvia el Sistema Operativo, dado que es un recurso transparente para la implementación del proyecto y dependerá del proveedor de MCC que se seleccione.

La segunda capa hace referencia a la nube, aquí se cuenta con un proveedor de computación en la nube que ofrezca servicio para desarrollo e implementación de aplicaciones móviles.

La tercera capa es la del usuario final, aquí el usuario por medio de un dispositivo móvil (celular o tableta) vía inalámbrica podrá interactuar con la app educativa, se plantea que sea por medio inalámbrico y a través de celular o tableta, dado que un principio de la educación ubicua, es que se pueda acceder en cualquier momento y en cualquier lugar.

## 2.2. Caso de Estudio

Partiendo del modelo tecnológico planteado, se realizó la implementación de una app educativa que permita el aprendizaje de la Resolución de Problemas Matemáticos, de acuerdo con la documentación presentada en el libro Lógica una Síntesis Didáctica [8], se define que se realizará una implementación mediante un modelo de aprendizaje basado en problemas, con características tipo juego, con el fin de que la app presente cierta

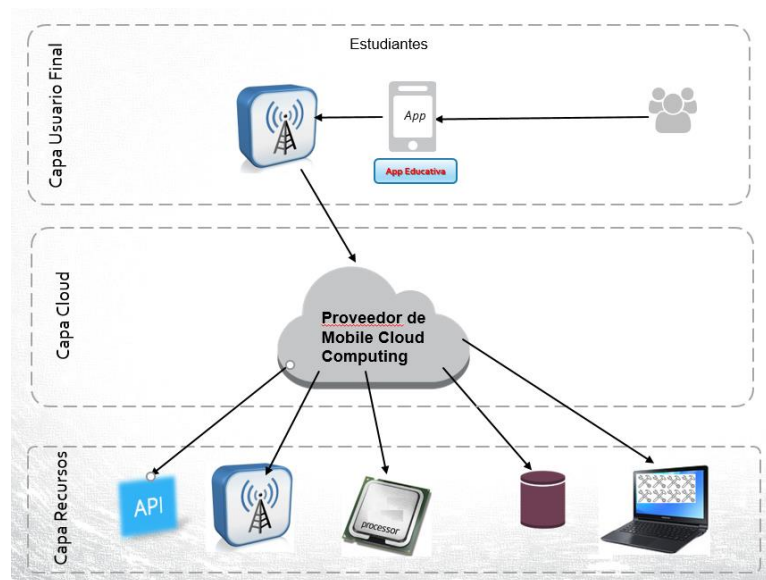


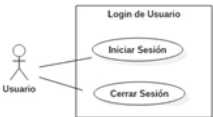
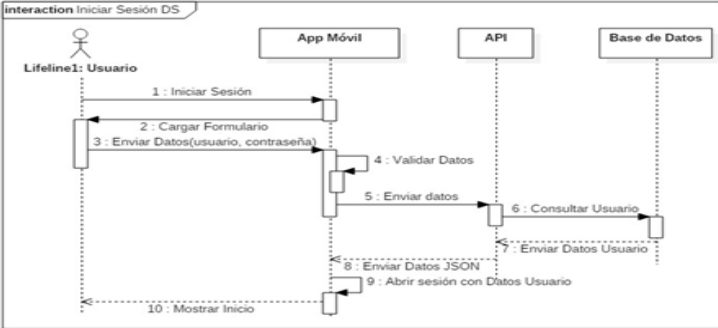
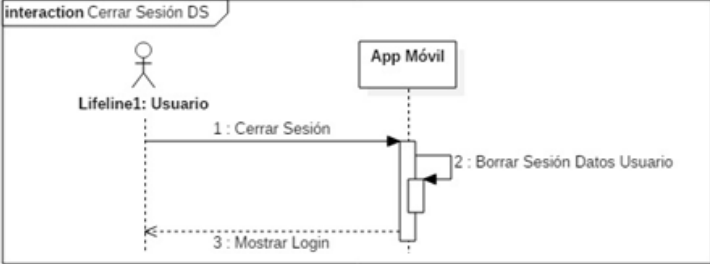
Figura 1. Middleware de modelo propuesto. Fuente: elaboración propia.

interactividad y permita que el usuario a medida que avanza en el estudio y refuerzo del tema pueda acceder a un incentivo como es una bonificación en monedas y obtener insignias, que le indiquen su progreso.

### 2.2.1. Requisitos Funcionales

Dada la descripción anterior, se identifican los requisitos funcionales, con su respectiva descripción.

Tabla 1. Requerimientos Funcionales – Inicio de sesión

Proyecto	Solución de problemas
Código	01-RF
Descripción	Para acceder a la aplicación el usuario debe iniciar sesión y debe tener la opción de cerrar dicha sesión. Para ello se debe pedir al usuario el correo electrónico o nickname (nombre de usuario), y la contraseña (está definida como el código del estudiante).
Entradas	Iniciar Sesión: nickname o correo electrónico, y contraseña
Procesos	<p>Inicio de Sesión:</p> <ol style="list-style-type: none"> <li>1. Verificar que el usuario ingrese nickname o correo electrónico, y contraseña</li> <li>2. Enviar los datos a través del API</li> <li>3. Consultar los datos a la base de datos</li> <li>4. Enviar respuesta del servidor.</li> </ol> <p>Cerrar Sesión:</p> <ol style="list-style-type: none"> <li>1. Presionar en el menú de aplicación la opción de cerrar sesión</li> <li>2. Eliminar datos de sesión</li> </ol> <p>Mostrar interfaz de inicio de sesión</p>
Salidas	<p>Inicio de Sesión:</p> <ol style="list-style-type: none"> <li>1. Si el usuario existe: mostrar interfaz de inicio.</li> <li>2. Si el usuario no existe: mostrar mensaje de validación.</li> </ol> <p>Cerrar Sesión:</p> <p>Mostrar interfaz de inicio de sesión.</p>
Caso(s) de uso	
Diagrama(s) de secuencia(s)	 

Fuente: elaboración propia.

Tabla 2. Requerimientos Funcionales – Inicio de la aplicación



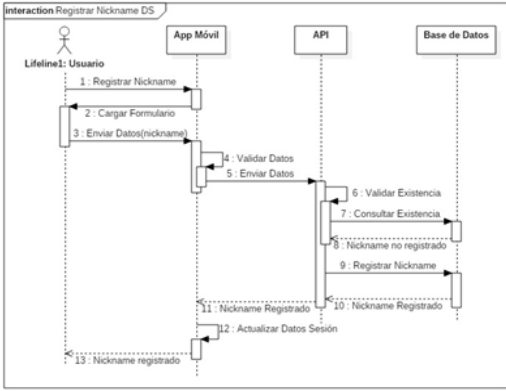
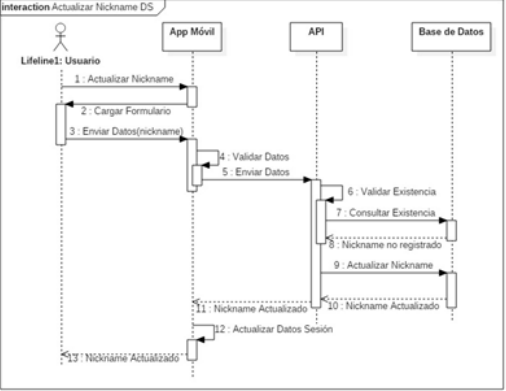
Proyecto	Solución de problemas
Código	02-RF
Descripción	En el inicio de la aplicación se debe mostrar un menú donde el usuario puede navegar a: 1. Instrucciones 2. Configuración 3. Explicación 4. Jugar
Entradas	
Procesos	
Salidas	Interfaz de inicio de la aplicación
Caso(s) de uso	

Tabla 3. Requerimientos Funcionales – Instrucciones


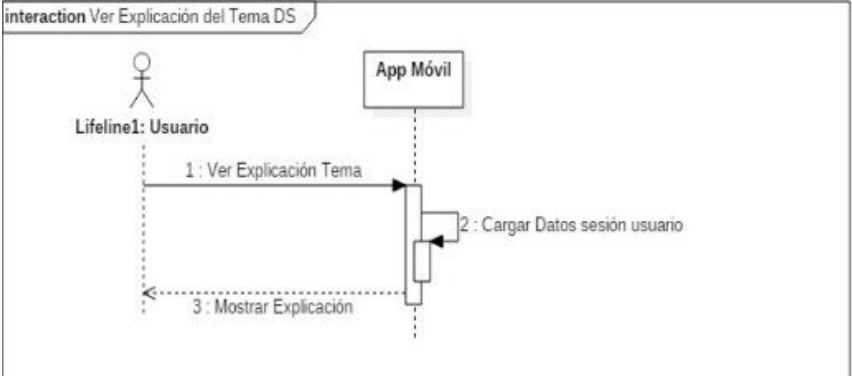
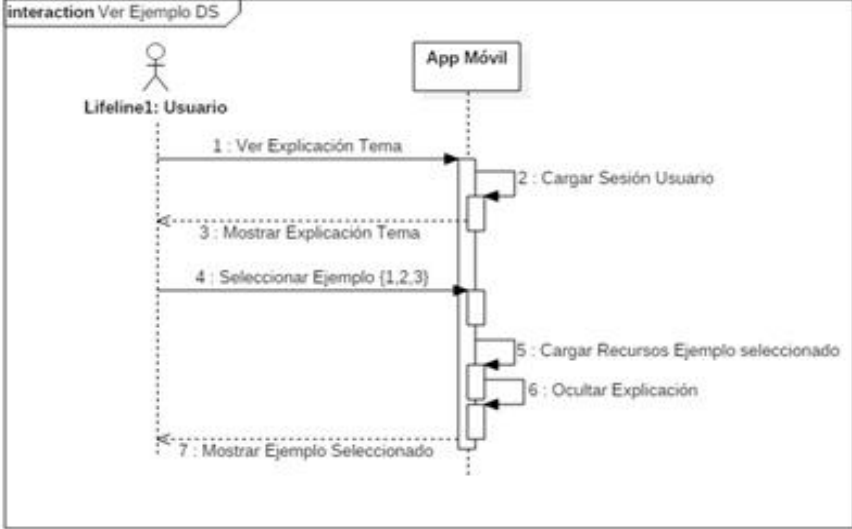
Proyecto	Solución de problemas
Código	03-RF
Descripción	Se debe mostrar una interfaz donde se explique el funcionamiento de la aplicación
Entradas	
Procesos	
Salidas	Interfaz de inicio de la aplicación

Tabla 4. Requerimientos funcionales – Configuración

Proyecto	Solución de problemas
Código	04-RF
Descripción	Se debe mostrar una interfaz donde el usuario pueda definir su nickname. Si el usuario ingresa por primera vez, el sistema debe abrir dicha interfaz para ingresar el nickname. También se debe dejar la posibilidad de modificación.
Entradas	Nickname
Procesos	1. Ingresar el nickname 2. Enviar el dato al API 3. Validar si el nickname existe 4. Registrar o modificar el nickname 5. Enviar respuesta del servidor Actualizar datos de sesión de usuario
Salidas	1. Si el nickname fue registrado o actualizado, se actualiza los datos del usuario en la sesión de la aplicación Si el nickname no pudo ser registrado o actualizado, mostrar un mensaje de alerta
Caso(s) de uso	
Diagrama(s) de secuencia(s)	<div> <p>Interaction Registrar Nickname DS</p>  </div> <div> <p>Interaction Actualizar Nickname DS</p>  </div>

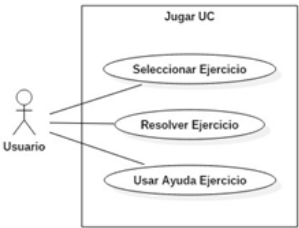
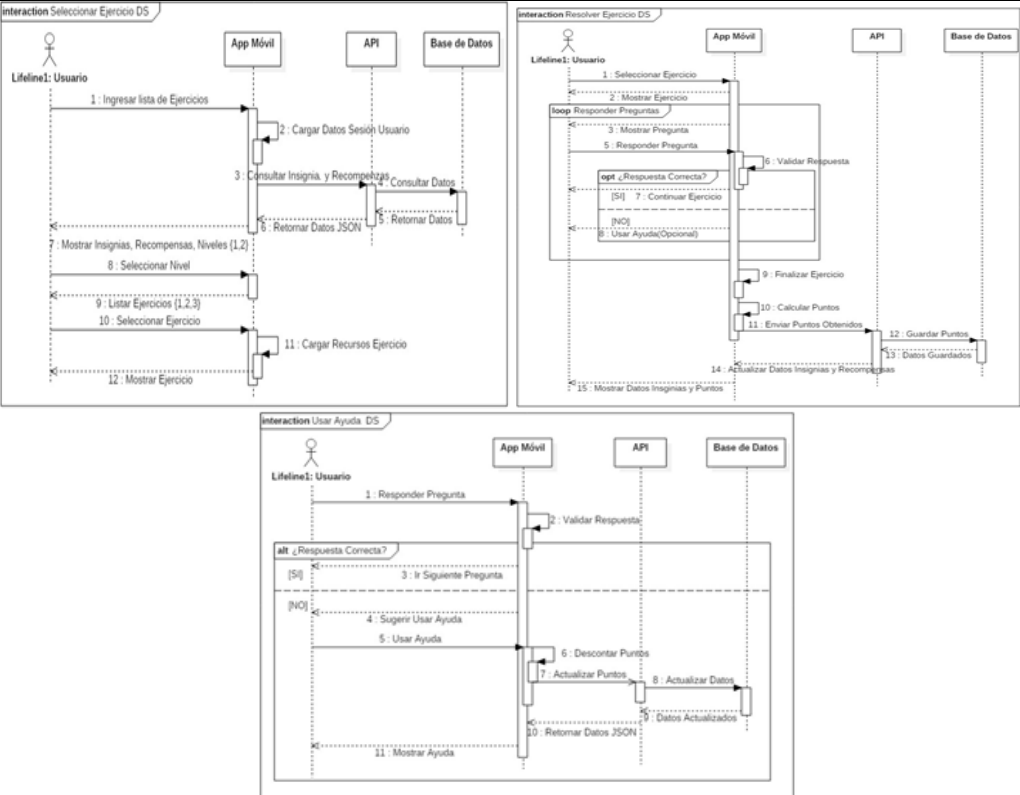
Fuente: elaboración propia.

Tabla 5. Requerimientos Funcionales – Explicación del tema

Proyecto	Solución de problemas
Código	05-RF
Descripción	Se debe mostrar una interfaz donde el usuario pueda revisar la explicación de los conceptos del tema de la aplicación y ver los ejemplos que contienen dicha explicación.
Entradas	
Procesos	
Salidas	1. Interfaz para ver la explicación del tema Interfaz para ver los ejemplos
Caso(s) de uso	
Diagrama(s) de secuencia(s)	<p><b>interaction Ver Explicación del Tema DS</b></p>  <pre> sequenceDiagram     participant U as Lifeline1: Usuario     participant A as App Móvil     U-&gt;&gt;A: 1: Ver Explicación Tema     activate A     A-&gt;&gt;A: 2: Cargar Datos sesión usuario     A--&gt;U: 3: Mostrar Explicación     deactivate A   </pre> <p><b>interaction Ver Ejemplo DS</b></p>  <pre> sequenceDiagram     participant U as Lifeline1: Usuario     participant A as App Móvil     U-&gt;&gt;A: 1: Ver Explicación Tema     activate A     A-&gt;&gt;A: 2: Cargar Sesión Usuario     A--&gt;U: 3: Mostrar Explicación Tema     U-&gt;&gt;A: 4: Seleccionar Ejemplo {1,2,3}     activate A     A-&gt;&gt;A: 5: Cargar Recursos Ejemplo seleccionado     A-&gt;&gt;A: 6: Ocultar Explicación     A--&gt;U: 7: Mostrar Ejemplo Seleccionado     deactivate A   </pre>

Fuente: elaboración propia.

Tabla 6. Requerimientos Funcionales – Jugar

Proyecto	Solución de problemas
Código	06-RF
Descripción	<p>El usuario debe tener una interfaz donde pueda seleccionar los ejercicios configurados en la aplicación y que se pueda visualizar los resultados obtenidos al resolver los ejercicios.</p> <p>Cada ejercicio contará con una serie de ayudas que al ser usadas descuentan puntos acumulados del usuario, dichas ayudas son opcionales para el uso del usuario.</p> <p>Al finalizar cada ejercicio, el usuario ganará una insignia y una serie de puntos que serán registrados en la base de datos.</p>
Entradas	Seleccionar un ejercicio, datos del usuario
Procesos	<p>Resolver Ejercicio:</p> <ol style="list-style-type: none"> <li>1. Seleccionar un ejercicio</li> <li>2. Responder las preguntas para avanzar</li> <li>3. Si el usuario se equivoca, podrá usar una ayuda</li> <li>4. Al usar la ayuda se descontarán puntos del usuario</li> <li>5. Hasta tanto, el usuario no escoja la respuesta correcta, no podrá avanzar a la siguiente pantalla.</li> <li>6. Al finalizar el ejercicio el usuario obtendrá una insignia y los puntos ganados</li> <li>7. Se registra el progreso en la base de datos</li> </ol> <p>Se regresa a la interfaz del listado de ejercicios</p>
Salidas	<ol style="list-style-type: none"> <li>1. Interfaz para ver el listado de ejercicios</li> </ol> <p>Registro de los puntos y las insignias del usuario</p>
Caso(s) de uso	
Diagrama(s) de secuencia(s)	

Fuente: elaboración propia.



### 2.2.2. Requisitos Funcionales

Una vez identificados los requisitos funcionales, se definen los requisitos no funcionales, los cuales se detallan a continuación.

Tabla 7. Requerimientos no funcionales – Jugar

Nombre	Código	Descripción
Tipo de aplicación y dispositivos donde se instala la aplicación	01-RNF	La aplicación será móvil para solo dispositivos Android.
Framework de desarrollo de la aplicación	02-RNF	La aplicación se desarrolla en ReactNative.
Motor de base de datos	03-RNF	El motor de base de datos que se usará es mysql
Infraestructura física	04-RNF	La infraestructura será cloud, usando los servicios de Azure
Arquitectura de la aplicación	05-RNF	La arquitectura será del tipo cliente-servidor y se estructura en dos partes backend y frontend. Para el backend se desarrollará un API basado en MVC (Modelo-Vista-Controlador) con código c# y framework API Web de Visual Studio. Para el frontend se desarrollará las interfaces con ReactNative. La base de datos y el API estarán configurados en el servidor cloud de azure.
Conexión a internet	06-RNF	Se requiere conexión a internet para poder consumir el API.

Fuente: elaboración propia.

### 2.3. Arquitectura de la aplicación

La arquitectura para implementar la aplicación se definió como Cliente/Servidor (C/S), donde el Cliente maneja la parte de la interfaz gráfica y la parte del servidor, donde se tienen todos los recursos tecnológicos y el software que permiten el funcionamiento de la app.

La parte del Cliente (FRONT-END), esta implementada dentro de la app, es decir, toda la explicación del tema va dentro de la interfaz gráfica (tablas, formularios, imágenes, textos), con esto se tiene la ventaja que, una vez descargada la app, el estudiante puede revisar el tema sin tener que estar conectado todo el tiempo a la red de datos. Solo se requiere la conexión a una red de datos

cuando va a realizar el ingreso a la app por primera vez, dado que se valida el usuario y la clave. Cuando el estudiante se vuelva a conectar a la app, si los datos quedan guardados en el dispositivo móvil, puede conectarse sin tener conexión a internet.

La otra parte es el servidor (BACK END), en este caso, en la parte del servidor se tiene el proveedor de los servicios de nube, quien es el que tiene configurado el manejador de base de datos y la lógica del negocio (en la API (por medio de servicios web)), la cual permite la conexión a la base de datos y la manipulación (operaciones de consulta, actualización, registro y eliminación) de la información almacenada.

La Figura 2, presenta la arquitectura implementada en el desarrollo de la app educativa.

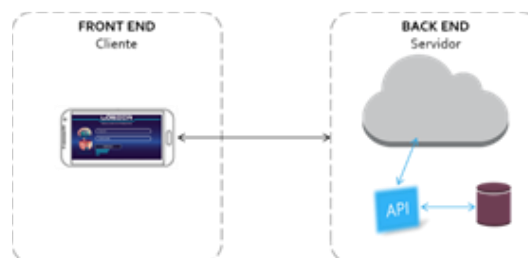


Figura 2. Arquitectura de la aplicación. Fuente: elaboración propia.

#### 2.3.1. Herramientas de desarrollo utilizadas en la implementación de la app

La implementación del Back End se realizó utilizando como proveedor del servidor de nube a la herramienta Microsoft Azure, la cual permite crear una cuenta gratuita e implementar el desarrollo con un crédito inicial, lo que ofrece unos recursos básicos, en caso de requerir mayores recursos, podrá pagar para tenerlos disponibles. Dado que se trabajó con Microsoft Azure, se utilizó para la implementación de la API, el lenguaje de programación C# y el framework ASP.NET Web API 2, el cual permite estructurar el API al modelo MVC y configurar los controladores y rutas definidos para el consumo del API y su posterior implementación, se desarrolló con la API Web de los proyectos creados con Visual Studio .NET. Para la implementación de la API es necesario configurar el host del servidor en donde se va a publicar el proyecto, e ingresar el usuario y la contraseña de Azure.

Como gestor de la base de datos se utilizó MySQL, el cual se habilitó en el Servidor Azure, siendo esta otra

ventaja de Azure, dado que como MySQL es software libre, no debe pagarse por su configuración, pero si por el almacenamiento de los datos y mediante Mysql Workbench se realizó la administración a la base de datos.

En la implementación del Cliente, se utilizó el framework ReactNative, el cual ofrece la posibilidad de desarrollo de aplicaciones móviles multiplataforma, aunque para este caso, la app solo fue desarrollada para sistemas con Android. Se diseñaron las interfaces gráficas y controles como cajas de texto, botones de chequeo, entre otros. Una ventaja que presenta ReactNative es que utiliza XMLHttpRequest (el cual es un estándar del W3C, lo cual garantiza la interoperabilidad entre sistemas) para realizar las peticiones http y https a servidores web y permite también hacer transferencia de datos. También, se usó la estructura JSON para el intercambio de los datos, el cual es un formato más sencillo que XML y que permite una mejor lectura en el desarrollo de este tipo de proyectos.

Para la realización de las pruebas se utilizó una librería conocida como Expo, la cual es un servicio gratuito de ReactNative, y permite realizar las pruebas sin necesidad de instalar la app en el dispositivo.

La Figura 3. Presenta un esquema general de las herramientas utilizadas en la implementación del proyecto.

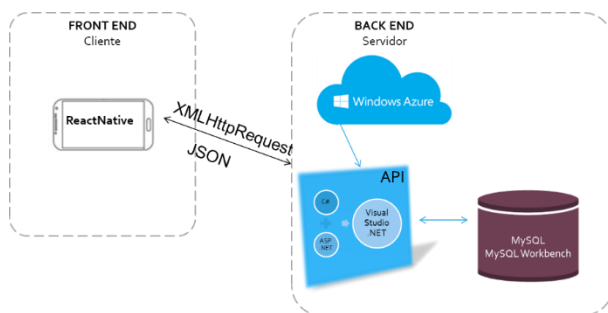


Figura 3. Herramientas de implementación. Fuente: elaboración propia.

### 2.3.2. Modelo Entidad-Relación

El modelo entidad relación, figura 4, permite llevar un control acerca de qué ejercicios ha desarrollado el estudiante, con el fin de conocer el avance. Al final, se puede obtener un reporte para conocer qué estudiantes ingresaron a la app y cuántas insignias obtuvieron, así como si utilizaron monedas para consultar las ayudas que explican las respuestas.

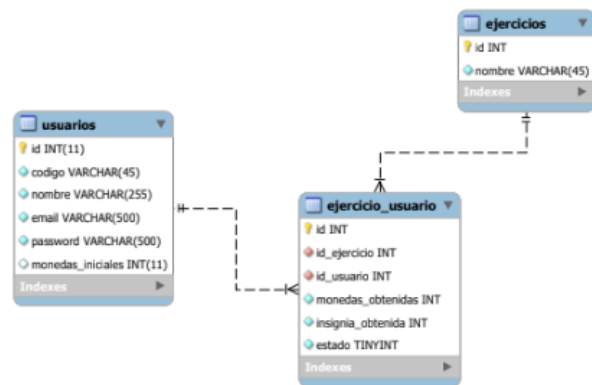


Figura 4. Modelo entidad-relación. Fuente: elaboración propia.

### 2.3.3. Diagrama de flujo de interfaces de usuario – User Flow

El diagrama de flujo de las interfaces de usuario, figura 5, describe el flujo de navegación del usuario en cada interfaz gráfica de la aplicación. En algunas de las interfaces se cuenta con opciones de Menú, que ayudan a una mejor navegación por la aplicación.

## 3. Resultados y discusión

EL modelo teórico planteado, presenta un panorama general que permite entender la estructura general desde el punto de vista de las Tecnologías de Información y Comunicaciones, que se puede emplear para la implementación de un proyecto de este tipo. Esta estructura conocida como middleware se ha diseñado en tres capas, de manera que se garantice la independencia entre las mismas, a la vez permite una mejor integración del proyecto y a futuro su mantenimiento sea más fácil, dado que, si se requiere ajustar o ampliar algo en alguna de las capas, el proceso es totalmente transparente para las otras.

Una arquitectura por capas como la plateada, ofrece versatilidad para la adaptación de nuevas funcionalidades, así como, el poder implementar API's garantiza independencia entre las capas de la arquitectura y el usuario final, esto permite implementar mejoras sin que la funcionalidad se afecte.

La arquitectura propuesta, ofrece la posibilidad de seleccionar la tecnología de información que mejor se adapte a los equipos de desarrollo que se establezcan de tal forma que este no sea un obstáculo para la implementación de una solución de este tipo.

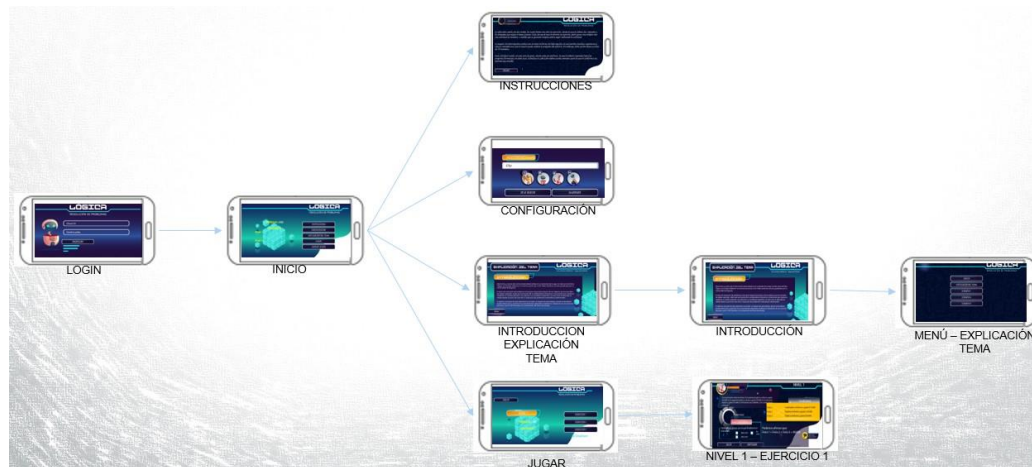


Figura 5. User flow. Fuente: elaboración propia.

El análisis de requerimientos de la solución que se proponga, podrá tomar como base los modelos de diseño aquí planteados y adecuarlos al caso específico que se quiera implementar, de modo que sirvan de guía para la implementación de la solución.

La arquitectura planteada permite que se puedan plantear diferentes alternativas de visualización de la solución propuesta, como puede ser desarrollar una app o desarrollar un sitio web en el cual se puedan implementar más funcionalidades.

El caso de estudio implementado, se desarrolló de acuerdo con el middleware definido, por lo cual también se identificaron las tres capas planteadas, como se presenta en la figura 6.

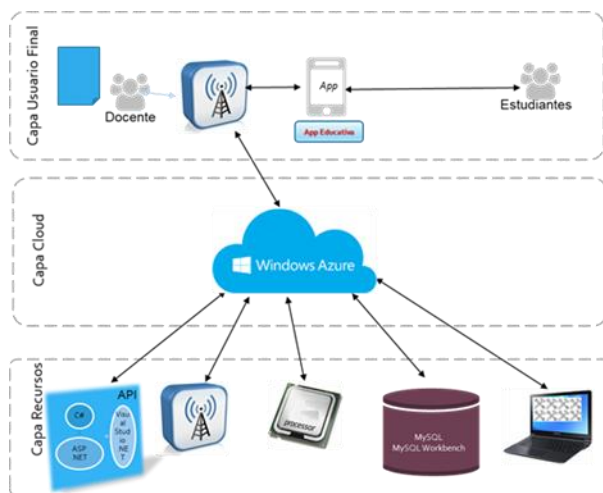


Figura 6. Middleware Caso de Estudio. Fuente: elaboración propia.

Así mismo, al desarrollar el proyecto por medio de una API, se logra la independencia entre capas, lo cual representa una ventaja significativa para el mantenimiento de la misma.

La app, fue desplegada en la Play Store, servicio ofrecido por Google, y se realizó el proceso que ellos tienen establecido para garantizar que no sea una aplicación con software malicioso.

De acuerdo con la literatura que existe sobre el tema y teniendo en cuenta algunos estudios que se han implementado se puede establecer que:

De acuerdo con [9], una de las características de la educación ubicua es poder ofrecer a los estudiantes que aprovechen diferentes espacios de tiempo para adelantar sus procesos de formación, lo cual se logra con la implementación de la app propuesta. Es importante resaltar que en una siguiente versión se podrían adicionar opciones que permitan la interacción de los estudiantes entre ellos con el fin de compartir conocimientos que adquieran.

El estudio realizado en [10] presentan una implementación de un caso de estudio de educación ubicua para el área de educación física, donde definen una serie de actividades que se apoyan en el uso de recursos como google drive, en este caso, el uso de herramientas ya desarrolladas permite incorporar algunas características mediante las cuales se puede realizar seguimiento dentro del proceso de aprendizaje, pero no permiten una adaptación a características especiales que se quieran incorporar, mientras el modelo planteado, presenta una personalización del proceso, donde se plasma la experiencia y conocimiento del docente,

haciendo una experiencia más personalizada para los estudiantes.

De acuerdo con [11] las implementaciones de educación ubicua ofrecen la posibilidad de incorporar “lectura, audio, imágenes, videos o actividades interactivas” (pág. 11), lo cual permite desarrollar contenidos con mayores recursos que brinden nuevas experiencias en el proceso de aprendizaje; para el caso de la app implementada no se utilizaron recursos como videos, lo cual se podría incorporar en una segunda versión, con el fin de ir complementando la app ya implementada.

Otra característica que debe tenerse en cuenta es que la aplicación que se desarrolle se adapte a diferentes plataformas y a diferentes tipos y tamaños de pantallas, como lo presentado en [12], en el caso de la app desarrollada tiene como limitante que se desarrolló para celulares con sistema operativo android, por lo cual para una siguiente versión se deberá implementar para dispositivos con otros sistemas operativos u otras plataformas, con el fin de que tenga una mayor versatilidad para su uso.

En [13] presentan un análisis de la implementación de herramientas que pueden propiciar la incorporación de las mismas en ambientes de educación ubicua, destacando el uso de Moodle y Black board como unas de las analizadas, estas herramientas facilitan la incorporación de ambientes ubicuos, pero tienen limitantes, dado que son para uso genérico, es decir, tienen configuradas opciones que pueden ser adaptadas según las necesidades, pero con la limitante de poder incorporar recursos personalizados donde la experiencia del docente sea incorporada de una manera adecuada. La app implementa personaliza el proceso para el tema específico logrando incorporar de manera detallada la experiencia del docente en el tema específico, pero cualquier cambio o mejora, deberá ser diseñado y programado por un experto en desarrollo de software.

Es importante resaltar que, la app fue implementada sólo para ser desplegada en dispositivos con sistema operativo Android, por lo cual un próximo paso será realizar los ajustes para poderla desplegar en otros dispositivos con diferentes sistemas operativos, tales como: IOS, Windows, entre otros.

#### 4. Conclusiones

Teniendo en cuenta los resultados obtenidos, se logran concluir los siguientes aspectos: 1) El modelo tecnológico teórico puede servir de base para la implementación de proyectos futuros; 2) el diseño del caso de estudio, permite materializar el modelo teórico y

brinda herramientas que sirven de base en la implementación de casos de estudio con condiciones similares; 3) el desarrollo de este tipo de herramientas permiten la incorporación de nuevos elementos al desarrollo de recursos educativos, brindando nuevas posibilidades de interacción para los estudiantes y de seguimiento a los docentes; 4) una tecnología como Mobile Cloud Computing permite crear proyectos que puedan ser extrapolados a una mayor población, con una posibilidad de crecimiento relativamente económica y sin requerir de expertos en la gestión de la infraestructura que soporta el mismo; 5) el poder desarrollar procesos más personalizados hace que se potencien las opciones de mejorar los recursos de aprendizaje, logrando un proceso más personalizado; 6) la incorporación de nuevas tecnologías de información para procesos de formación ofrece la posibilidad de incorporar de manera más personalizada los recursos desarrollados, logrando así, una mejor experiencia para quien está recibiendo la formación, 7) la implementación de la app, permitió que los estudiantes pudieran contar con una herramienta tecnológica diferente a lo que normalmente usan en el aula de clase, lo cual generó gran expectativa por parte de ellos y les permitió reforzar los conocimientos adquiridos sobre el tema y 8) para la implementación de una solución de este tipo se debe contar con la experiencia del docente que conoce y orienta el tema específico y un desarrollador que entienda lo que el docente quiere transmitir y lo plasme en la solución tecnológica propuesta.

#### Referencias

- [1] D. Puthal, B.P.S. Sahoo, S. Mishra, S. Swain, “Cloud Computing Features, Issues, and Challenges: A Big Picture”, en *2015 International Conference on Computational Intelligence and Networks*, Bhubaneswar, 2015, pp. 116-123, doi: 10.1109/CINE.2015.31
- [2] P. Li, J. Li, Z. Huang *et al.*, “Privacy-preserving outsourced classification in cloud computing”, *Cluster Computing*, vol. 21, no. 1, pp. 277-286, 2018, doi: 10.1007/s10586-017-0849-9
- [3] J. Del Vecchio, F. Paternina, C. Henríquez, “La computación en la nube: un modelo para el desarrollo de las empresas”, *Prospect*, Vol 13, no. 2, pp. 81-87, 2015, doi: 10.15665/rp.v13i2.490
- [4] Ericsson, “Ericsson Mobility Report June 2019”, Ericsson Corp., 2019 [En línea]. Disponible en: <https://www.ericsson.com/en/press->

releases/2019/6/ericsson-mobility-report-5g-uptake-even-faster-than-expected

[5] D. Debashis, *Mobile Cloud Computing: Architectures, Algorithms and Applications*. New York, Estados Unidos: CRC Press Taylor & Francis Group, 2016.

[6] G. M. Ramírez, C. A. Collazos, F. Moreira, H. Fardoun, “Relación entre el U-Learning, aprendizaje conectivo y el estándar xAPI: Revisión Sistemática”, *Campus Virtuales*, vol 7, no. 1, pp. 51-62, 2018.

[7] C. I. Báez, C. Clunie, “Una mirada a la Educación Ubicua”, *Revista Iberoamericana de Educación a Distancia*, vol 22, no. 1, pp. 325-344, 2019, doi: 10.5944/ried.22.1.22422

[8] F. Gutiérrez, *Lógica una Síntesis Didáctica*. Tunja-Colombia: Ediciones Universida de Boyacá, 2015.

[9] S. E. Hidalgo Pérez, M. S. Orozco Aguirre, M. T. Daza Ramírez, “Trabajando con Aprendizaje Ubicuo en los alumnos que cursan la materia de Tecnologías de la Información”, *Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, vol 6, no. 11, pp. 605-621, 2015

[10] V. GallegoLema, J. A. MuñozCristóbal, H. F. Arribas Cubero, B. RubiaAvi, “Aprendizaje ubicuo: un proceso formativo en educación física en el medio natural”, *Revista Latinoamericana de Tecnología Educativa*, vol. 15, no. 1, pp 59-73, 2016, doi: 10.17398/1695288X.15.1.59

[11] E. Vázquez-Cano, J. M. Fernández Batanero, E. López Meneses, “Movilidad y ubicuidad en la Educación Superior”, en *Innovación e investigación sobre el aprendizaje ubicuo y móvil en la Educación Superior*, Barcelona, Ediciones Octaedro 2019, pp. 7-22.

[12] G. A. Moreno-López, E. J. Ramírez-Monsalve, y J. A. Jiménez-Builes, “Ubiquitous learning model based on platforms of multi-screen TV (uLMTV)”, *Revista DYNA*, vol. 84, pp. 160-169, 2017, doi: 10.15446/dyna.v84n203.64160

[13] S. A. Aljawarneh, “Reviewing and exploring innovative ubiquitous learning tools in higher education”, *Journal Comput Higher Education*, vol. 32, pp. 57-73, 2020, doi: 10.1007/s12528-019-09207-0