

Capítulo 4

Arquitectura del sistema ubicuo: escenario M-Learning

Este capítulo describe la implementación del prototipo de una aplicación móvil multiplataforma para la Institución Universitaria Antonio José Camacho que permita apoyar los procesos de formación ubicua por medio de cursos cortos afines a los diferentes programas ofrecidos por la universidad. Se aprovechan los avances tecnológicos en el desarrollo de aplicaciones y de los Learning Management Systems (LMS) para lograr realizar una integración que brinde a los usuarios una plataforma interactiva, de fácil acceso y que integre los componentes didácticos del M-Learning. De igual forma, se especifica el desarrollo tecnológico en Android con las integraciones asociadas al LMS para desplegar a estudiantes y docentes una solución que dé apoyo a esta modalidad de formación a distancia.

La humanidad se encuentra actualmente en un punto de transformación tecnológica sin precedentes y, en efecto, el desarrollo de las TIC ha facilitado la generación de conocimiento, convirtiéndose en un punto clave en los procesos formativos. En afinidad, Carneiro, Toscano y Díaz (2009) refieren que las TICS en el campo educativo han abierto grandes posibilidades para el mejoramiento de los procesos de enseñanza y de aprendizaje.

Uno de los aspectos que más ha contribuido en la evolución del uso e impactos de las TIC en la educación tiene que ver con el despliegue de escenarios de formación ubicua, que les permite a los educandos acceder a recursos de formación a distancia en cualquier instante, lugar y dispositivo de comunicación electrónico con conexión a Internet. Uno de los recursos tecnológicos que más ha contribuido a la masificación de este tipo de alternativa de formación es el de los servicios móviles educativos o M-Learning.

Este capítulo describe la implementación del prototipo de una aplicación móvil multiplataforma para la Institución Universitaria Antonio José Camacho que permite apoyar los procesos de formación ubicua por medio de cursos cortos afines a los diferentes programas ofrecidos por la universidad. Se aprovechan los avances tecnológicos en el desarrollo de aplicaciones y de los Learning Management Systems (LMS) para lograr realizar una integración que brinde a los usuarios una plataforma interactiva, de fácil acceso y que integre los componentes didácticos del M-Learning.

Se definió una arquitectura orientada a servicios que cumpla con los requerimientos y necesidades del proyecto, adicionalmente será utilizada la metodología ICONIX en el desarrollo del proyecto que tiene 4 fases las cuales permiten tener un levantamiento de información preciso y una implementación acoplada a las interacciones definidas para el proyecto.

Con la implementación de este desarrollo tecnológico se busca dejar la base documental necesaria para brindarle a los estudiantes y profesores que necesiten realizar proyectos similares y brindarle a la UNIAJC un prototipo que amplíe la educación virtual brindada actualmente.

4.1 Metodología utilizada en la implementación de la solución móvil multiplataforma

Teniendo en cuenta la necesidad de contar con una metodología que permita documentar de forma estructurada los procesos de software, pero que a la vez se enfoque en el desarrollo

ágil, se tomó como referencia ICONIX, que consiste en una metodología híbrida (tradicional-ligera) que se halla entre las metodologías RUP (Rational Unified Process) y XP (eXtreme Programming), que integra un conjunto de métodos orientados a objetos con el fin de tener un control estricto durante el ciclo de vida de un proyecto de software. Este cuenta con una serie actividades claramente definidas y muestra los pasos a ser implementados.

Iconix cuenta con 3 características principales:

- Iterativo o Incremental
- Trazabilidad
- Dinámica del UML

Para el desarrollo de este proyecto se definieron 7 iteraciones, en donde se aplicaron los diagramas de uso para cada iteración, el modelo de dominio general de todo el proyecto, se realizaron los prototipos de interfaz para cada iteración y, por último, la implementación en cada una de las iteraciones, como se muestra en la descripción de la iteración 6.

4.1.1 Iteración 6: Foros

La implementación de la iteración consiste en crear una funcionalidad que permita consultar los foros relacionados a los cursos del estudiante y realizar publicaciones en cada uno. Las iteraciones están compuestas por las fases de análisis de requerimientos, análisis y diseño preliminar, implementación y pruebas.

Fase: Análisis de requerimientos

Los requerimientos asociados a la iteración son:

- El sistema debe permitir la comunicación asincrónica por medio de foros, con el fin de que estudiantes y docentes de prueba tengan un canal de comunicación.
- El sistema permitirá la funcionalidad de mensajería instantánea (chat) entre los estudiantes y el docente de prueba, cuando se realice una clase en vivo.

Las figuras 8, 9 y 10 muestran las propuestas de Mockups definidos en la fase de requerimientos:



Figura 8. Prototipo – Listado de foros.
 Fuente: los autores.

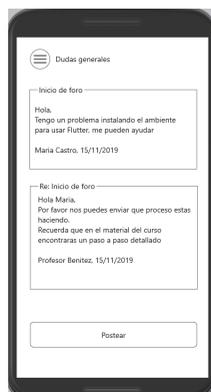


Figura 9. Prototipo – Información de foros.
 Fuente: los autores.



Figura 10. Realizar Post Prototipo – Listado de foros.
 Fuente: los autores.

Fase: Análisis y diseño preliminar

Esta fase describe los casos de uso extendidos relacionados con el listado de foros y la participación que tendrán los participantes en el curso. En las tablas 6 y 7 se muestra la descripción de estas acciones por medio de las plantillas

Concepto	Descriptivo
ID	IT6_CU1
Iteración	6
Nombre	Listar foros
Propósito	Permita a los usuarios visualizar la lista de foros en los que puede participar
Descripción	Se requiere un módulo del sistema que permita a los estudiantes ver el listado de foros del curso seleccionado
Actores	Estudiante
Precondiciones	Haber seleccionado un curso
Postcondiciones de éxito	Visualizar lista de foros creados por el docente
Postcondiciones de fracaso	Mostrar mensaje de que no se han creado foros hasta la fecha

Tabla 6. Caso de uso extendido – Listar foros
 Fuente: los autores.

Concepto	Descriptivo
ID	IT6_CU2
Iteración	6
Nombre	Participar en foro
Propósito	Permita a los usuarios hacer participaciones en los foros
Descripción	Se requiere un módulo del sistema que permita a los estudiantes participar y responder en los foros creados por el docente.
Actores	Estudiante
Precondiciones	Haber seleccionado un curso y que el docente haya creado foros
Postcondiciones de éxito	Visualizar la participación realizada en el foro
Postcondiciones de fracaso	Mostrar mensaje de que no se ha podido hacer la participación en el foro

Tabla 7. Caso de uso extendido – Participar en foro
 Fuente: los autores.

En la Figura 11 se encuentra el diagrama de robustez donde se muestra el proceso para la visualización de los foros y la publicación de posts.

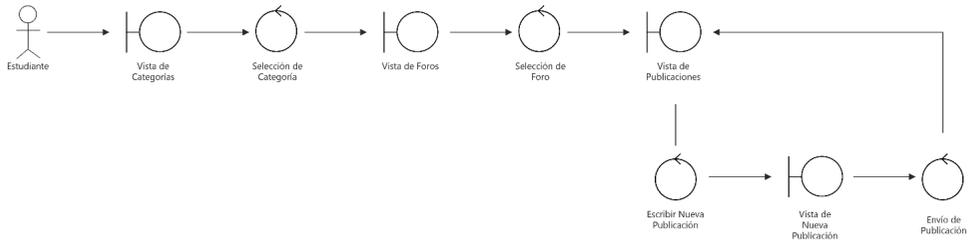


Figura 11. Diagrama de robustez – Foros.
Fuente: los autores.

Fase: Diseño

Se definen pruebas de carga en la funcionalidad de la iteración para determinar cuál es el tiempo de respuesta del servidor al realizar varias peticiones. Se parametriza lo siguiente:

- Concurrencia de 60 usuarios realizando la petición.
- Tiempo de respuesta menor a 1500 milisegundos.

Fase: Implementación

En lo referente al consumo de servicios, se implementó el patrón de arquitectura MVVM (Model-View-ViewModel), el cual es el más usado para el desarrollo de aplicaciones móviles, y el que Google adoptó como arquitectura oficial para aplicaciones en Android. Su aplicación en esta iteración fue la siguiente:

Model: en el modelo se crea una clase PostApi, que es la que se conecta al servicio de autenticación de Chamilo, el cual, para hacer la petición, recibe el título y contenido del post para hacer un nuevo post en el foro, y también contiene la función de obtener los posts del foro, como se muestra en la Figura 12.

```

Map<String, String> data = {
    "op": "ObtainPosts",
    "C": locator<LocalStorageService>().course,
    "T": thread,
    "wsToken": locator<LocalStorageService>().token,
};

try {
    final response = await http.post(endpoint,
        body: data,
        headers: {"Content-type": "application/x-www-form-urlencoded"});
    final resBody = response.body;

    if (response.statusCode == 200) {
        if (json.decode(resBody)["success"] == 0) {
            return Future.error("No hay datos");
        }
        if (json.decode(resBody)["success"] == 2) {
            closeSession();
            return Future.error("Credenciales inválidas");
        }
        final dataJson = json.decode(resBody)["posts"];
        dataJson.forEach((item) => posts.add(Post.fromJson(item)));
    }
}

Map<String, String> data = {
    "op": "WritePost",
    "C": locator<LocalStorageService>().course,
    "T": threadId,
    "F": forumId,
    "TIT": post.title,
    "CON": post.text,
    "wsToken": locator<LocalStorageService>().token,
};

try {
    final response = await http.post(endpoint,
        body: data,
        headers: {"Content-type": "application/x-www-form-urlencoded"});
    final resBody = response.body;

    if (response.statusCode == 200) {
        if (json.decode(resBody)["success"] == 0) {
            return Future.error("Error al insertar");
        }
        if (json.decode(resBody)["success"] == 2) {
            closeSession();
            return Future.error("Credenciales inválidas");
        }
    }
    success = true;
}
    
```

Figura 12. . Model – Foros.
Fuente: los autores.

View: es donde se visualizan todos posts de un foro en específico y también se tiene la opción de agregar un nuevo post, para el cual le solicita al ViewModel la creación de un nuevo post, como se muestra en la Figura 13.

```

Widget _postItem(Post post) {
  return Container(
    margin: EdgeInsets.only(bottom: 5),
    child: Card(
      elevation: 4.0,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.all(Radius.circular(10)),
      ), // RoundedRectangleBorder
      child: Padding(
        padding: EdgeInsets.all(10.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.max,
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Text(post.title, style: textTitlePlace),
            Html(data: post.text),
            Text(post.user),
            Text(post.date),
          ], // <Widget>[]
        ), // Column
      ), // Padding
    ), // Card
  ); // Container
}

```

```

void _onClickCreatePost() async {
  if (postProvider.isBusy()) return;

  if (_formKey.currentState.validate()) {
    final post = Post(
      title: _titleController.text,
      text: _contentController.text,
    );

    if (await postProvider.uploadPost(post, threadId, forumId)) {
      locator<NavigationService>()
        ..navigatorKey.currentState.pop()
        ..navigateTo(Routes.posts, pop: true, arguments: data);
    }
  }
}

```

Figura 13. View – Foros.
 Fuente: los autores.

ViewModel: es el que comunica a la vista con el modelo, esta recibe la acción de la vista, que en este caso es “obtainPost” o “uploadPost” y, en el momento que el ViewModel obtenga una respuesta del Model, le notificará a la vista del resultado, como se muestra en la Figura 14.

```

Future<List<Post>> obtainPosts(String thread) async {
  try {
    return await _postRepository.obtainPosts(thread);
  } catch (error) {
    return Future.error(error.toString());
  }
}

Future<bool> uploadPost(Post post, threadId, forumId) async {
  setBusy();
  bool uploadedPost;
  try {
    uploadedPost = await _postRepository.uploadPost(post, threadId, forumId);
  } catch (error) {
    this.error = error.toString();
    uploadedPost = false;
  }
  setIdle();
  return uploadedPost;
}

```

Figura 14. ViewModel – Foros.
 Fuente: los autores.

Se realiza la prueba de cargue para determinar el tiempo de respuesta de la solicitud como se muestra en la Figura 15.

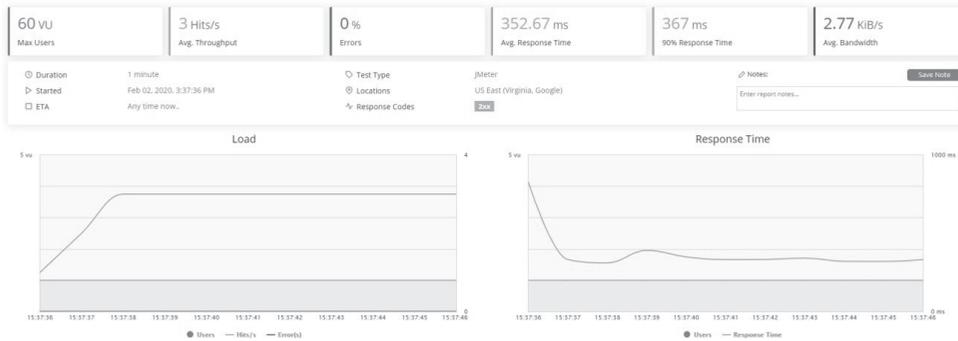


Figura 15. Prueba de carga –Foros
Fuente: los autores.

Se identifica que al realizar la solicitud con una concurrencia de 60 usuarios no se presentan errores y el tiempo promedio es de 352.67 milisegundos, adicionalmente se puede apreciar lo siguiente:

- El 90% de las solicitudes tienen un tiempo de respuesta de 367 milisegundos
- La banda ancha promedio de la prueba fue 2.77 kilobytes por segundo
- Se soporta un promedio de 3 solicitudes por segundo

4.1.2 Diagrama de secuencia de Foros

Al tener acceso a la información de los cursos desde la aplicación móvil, el estudiante tiene acceso a los foros abiertos en cada curso, una vez en ellos puede ver los mensajes de los participantes o realizar un post. En la Figura 16 se especifica el proceso que debe seguir el estudiante para acceder a la opción.

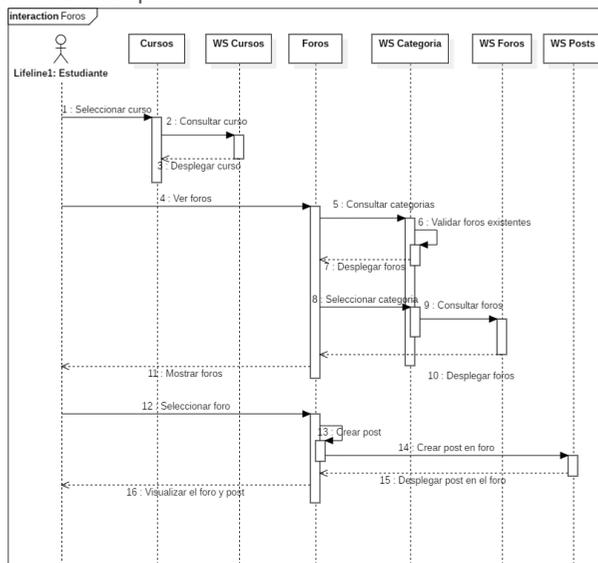


Figura 16. Diagrama de secuencia –Foros.
Fuente: los autores.

4.2 Componentes pedagógicos requeridos en la implementación de la solución M-Learning

Fueron identificados los siguientes componentes didácticos relacionados a la modalidad M-Learning que se adaptan al modelo pedagógico de la Institución Universitaria Antonio José Camacho, como se muestra en la Tabla 8:

Aspecto pedagógico	M-Learning
Medios Informativos	<ul style="list-style-type: none"> • Anuncios • Notificaciones push
Pautas de Organización	<ul style="list-style-type: none"> • Contenido del Curso
Materiales de Apoyo	<ul style="list-style-type: none"> • Documentos de apoyo • Material multimedia
Aprendizaje Colaborativo	<ul style="list-style-type: none"> • Foros • Chat • Clase en Vivo
Sistema de Evaluación	<ul style="list-style-type: none"> • Participación en colaboración conjunta

Tabla 8. Componentes didácticos de M-Learning
 Fuente: Vasco & Steven (2017)

Para la administración de los componentes didácticos fue utilizado Chamilo, un LMS (Learning Management System), en este se define el contenido educativo y la administración de usuarios de la aplicación. Adicionalmente fueron desarrollados servicios web para realizar la comunicación con la aplicación móvil que facilitan el paso de información y la ejecución de actividades.

4.3 Arquitectura de la solución tecnológica

Se utiliza el modelo 4+1 para representar la arquitectura del proyecto desde varios puntos usando vistas concurrentes. Las vistas representan lo siguiente:

- Vista lógica, tiene la información sobre varias partes del sistema y es enfocada al desarrollador. Se usan los diagramas de clases, objetos y secuencias.
- Vista de proceso, se describen los procesos concurrentes del sistema, engloba el performance y disponibilidad. Son usados los diagramas de actividades.
- Vista de desarrollo, se enfoca en representar los módulos y subsistemas con los diagramas de paquetes y componentes.
- Vista física, se describe el despliegue físico del sistema. Se usan los diagramas de despliegue.
- Vista de casos de uso o escenarios, representa la funcionalidad del sistema desde una perspectiva externa y es usada como guía para las demás vistas. Son usados los diagramas de casos de uso.

4.4 Vista lógica

Se representa de forma gráfica el funcionamiento del proyecto en cada uno de sus procesos utilizando los diagramas de dominio y secuencia. En la Figura 17 se muestran los dominios que modelan la información a manejar y sus relaciones entre ellos.

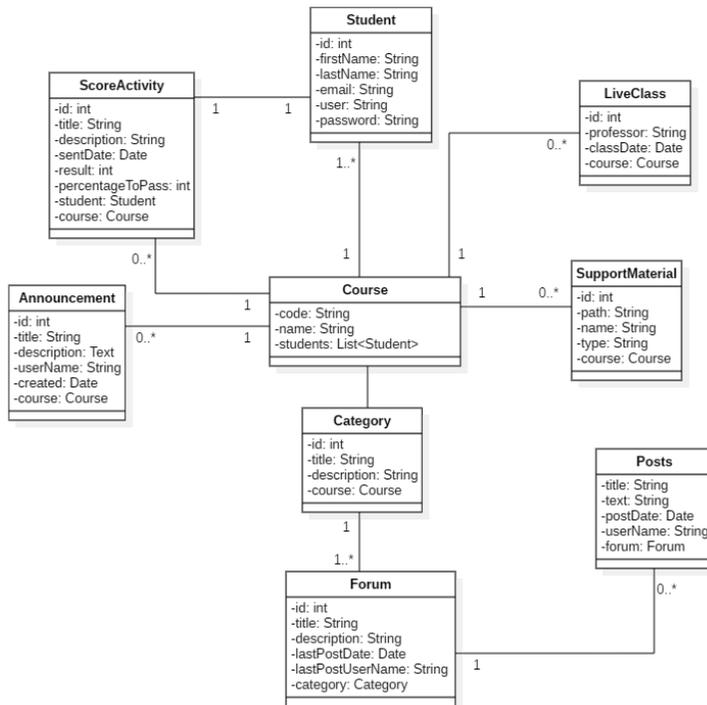


Figura 17. Diagrama de dominio.
 Fuente: los autores.

Se definen los modelos de:

- **Student:** modela los estudiantes dentro de la plataforma LMS, que será asociado a un curso mediante el cual accederá a toda la información que se asocie al curso.
- **Course:** modela los cursos que se creen dentro del LMS, en el cual se asociarán los elementos que a continuación se van a nombrar.
- **Announcement:** son todos los anuncios que los docentes del curso.
- **Category:** contiene la información, relaciona con los foros que el docente crea para la comunicación asincrónica con los estudiantes, para el manejo de esta se tiene también Forum y Posts que complementan el modelado de Category.
- **SupportMaterial:** modela todo el material de apoyo que el docente suba al LMS.
 LiveClass: contiene la información referente a las clases en vivo que el docente imparta para un grupo.
- **ScoreActivity:** contiene la información de las calificaciones que obtienen los estudiantes sobre las actividades que suben a la plataforma.

4.5 Vista de desarrollo

El modelo utilizado en el proyecto debe tener una estructura, se definen las dependencias de cada paquete por medio del diagrama de la Figura 18.

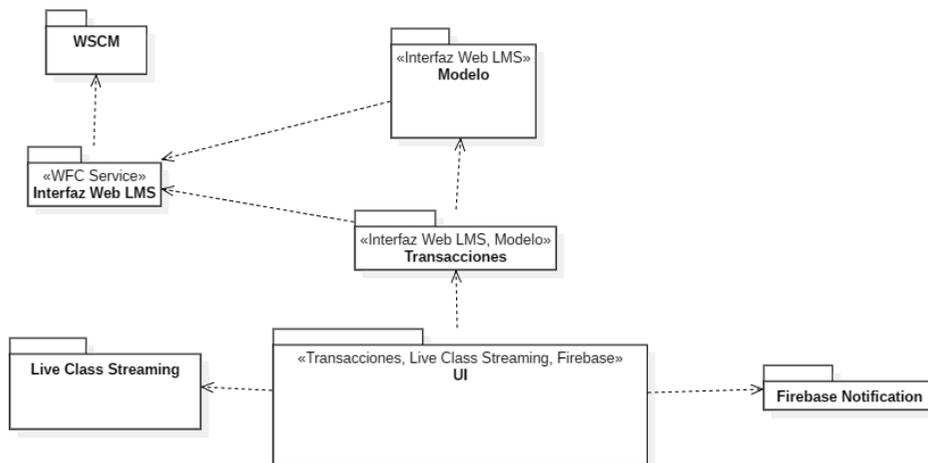


Figura 18. Diagrama de paquetes.
 Fuente: los autores.

El diagrama de paquetes lo conforman los siguientes componentes de software

- **WSCM:** este paquete es donde se encuentra todo el núcleo del LMS, en el cual se hace la gestión de todos los cursos.
- **Interfaz Web LMS:** este paquete es el que le permite a la aplicación móvil acceder a toda la información mediante peticiones HTTP por medio de una interfaz web.
- **Modelo:** es la capa donde encontramos el modelado de las entidades de la aplicación, la cual permite compartir información entre las aplicaciones.
- **Transacciones:** este paquete tiene todas las transacciones con la lógica del negocio entre las interacciones de la aplicación móvil y los servicios web.
- **UI:** este paquete contiene las interfaces móviles con las que interactúan los estudiantes.
- **Firebase Notification:** este paquete permite recibir las notificaciones PUSH en la aplicación móvil.
- **Live Class Streaming:** este paquete contiene los servicios para las transmisiones en vivo.

4.6 Vista de proceso

Se debe definir el funcionamiento y flujo del sistema a través del tiempo, para esto se utiliza un diagrama de actividades para representar la vista de proceso. En la Figura 19 se representa el flujo del estudiante desde su ingreso a la App hasta la interacción con cada uno de los componentes.

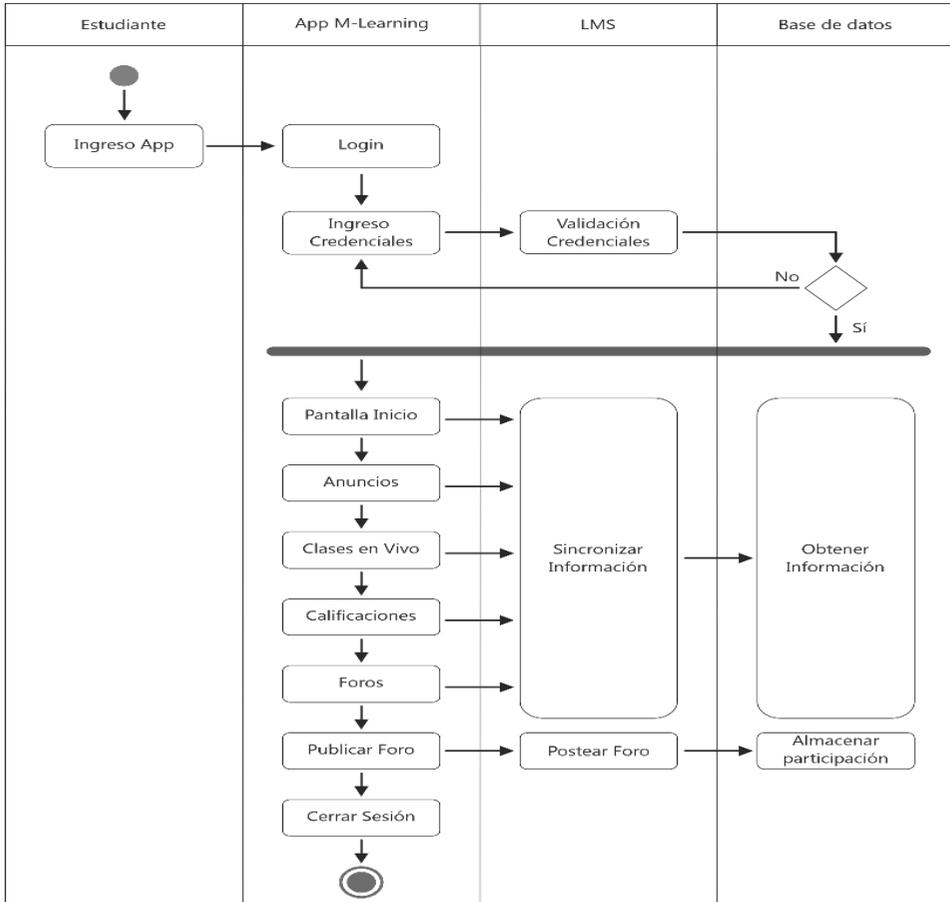


Figura 19. Diagrama de actividades.
 Fuente: los autores.

4.7 Vista física

Debido a que se trabaja con diferentes componentes del sistema, mediante el diagrama de despliegue que se ve en la Figura 20 se representan cada uno de ellos y la manera entre cómo se comunican.

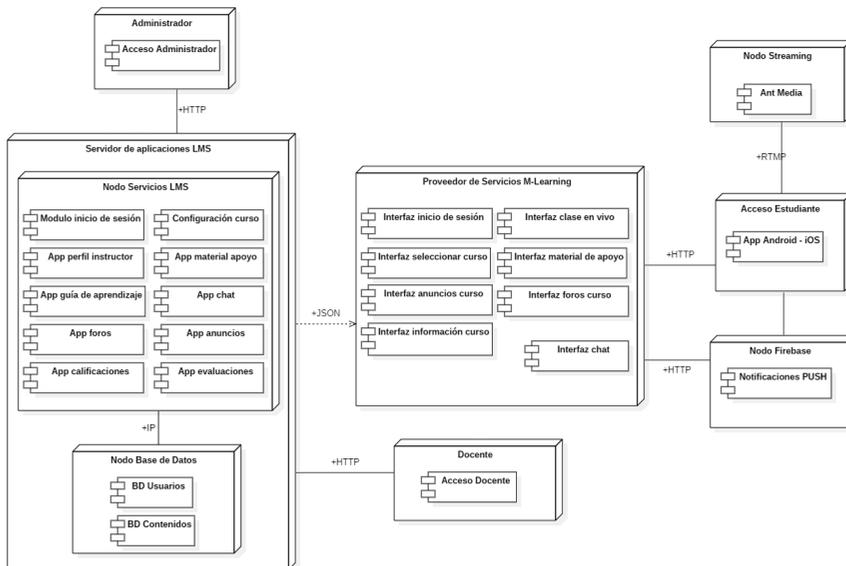


Figura 20. Diagrama de despliegue.
 Fuente: los autores.

La vista física del sistema la conforman los siguientes nodos:

- **Nodo gestión (Administrador y docente):** a través de una interfaz web, tanto el administrador de los contenidos del LMS como los docentes tendrán acceso de todo el nodo de Servicios LMS, en donde podrán crear cursos y asignar docentes (Administrador) y gestionar lo referente a la información y contenido del curso (Docente).
- **Nodo servicios LMS:** en este nodo es donde se despliegan los servicios del LMS, los cuales serán accedidos por los administradores del LMS y docentes; este componente es con el cual el proveedor de Servicios M-Learning se conectará y obtendrá toda la información referente al curso que el estudiante esté visualizando.
- **Nodo Base de datos:** este nodo almacena toda la información del servidor LMS.
- **Proveedor de Servicios M-Learning:** este nodo contiene todos los servicios que la aplicación móvil de los estudiantes obtiene desde la información que reposa en el LMS, este nodo es el intermediario entre la aplicación móvil y el nodo de Servicios LMS.
- **Nodo Streaming:** este nodo contiene el servidor Ant Media que permite realizar el Streaming de video y ser consumido desde la aplicación móvil
- **Nodo Acceso Estudiante:** aquí es donde los estudiantes accederán a la información del curso, el cual es por medio de la aplicación móvil. Cada estudiante debe contar con credenciales de ingreso. Para poder visualizar información e interactuar, el estudiante debe estar matriculado en un curso del LMS.
- **Nodo Firebase:** con este nodo se trabajan las notificaciones PUSH cada que haya mensajes en el chat de la sesión en vivo.

4.8 Escenarios

En esta vista se especifican por medio de los casos de uso el despliegue funcional del comportamiento que tendrán los estudiantes y el docente con la solución M-Learning propuesta. Se definen 2 roles para el funcionamiento del sistema:

- **Docente:** se encarga de la administración de los cursos que tenga asignados, esto lo realiza directamente desde el LMS y adicionalmente en la aplicación realiza la creación de las sesiones en vivo para interactuar con los estudiantes.
- **Estudiante:** tiene los permisos para acceder a la aplicación y a todos los módulos necesarios para ver y obtener la información de los cursos, acceder a sesiones en vivo, acceso a foros y chats para la comunicación entre los participantes del curso. Al interactuar con la mayoría de la aplicación es considerado el rol principal.

Teniendo en cuenta lo anterior, en las figuras 21 y 22 se definen los diagramas de caso de uso para cada rol:

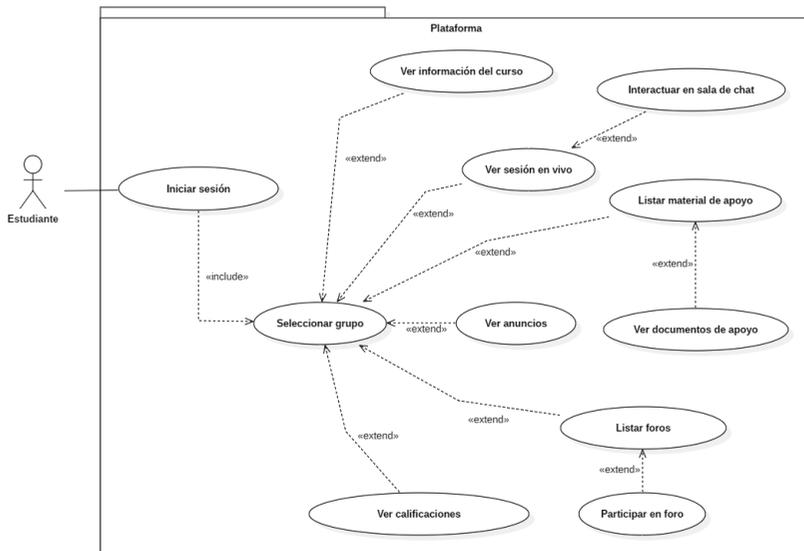


Figura 21. Diagrama de secuencia – Diagrama de casos de uso estudiante.
Fuente: los autores.

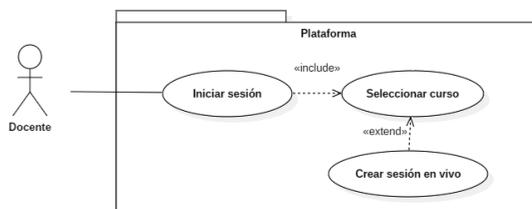


Figura 22. Diagrama de secuencia – Diagrama de casos de uso docente.
Fuente: los autores.

4.9 Implementación de la solución M-Learning

La aplicación móvil fue desarrollada con el framework Flutter utilizando la arquitectura para la vista MVVM (Model – View – ViewModel) y el uso del patrón Repository. Con esta arquitectura se logran consumir los servicios brindados por el LMS para ser usados en las respectivas funcionalidades. Los módulos definidos para la aplicación fueron los siguientes:

Inicio de sesión: es la pantalla principal y por medio de esta se realiza el ingreso al sistema. Las credenciales ingresadas son brindadas por el LMS, en la Figura 23 se puede apreciar la vista final:



Figura 23. Inicio de Sesión solución M-Learning.
Fuente: los autores.

Información de cursos: cada curso tiene una información general y su propio menú, en la Figura 24 se pueden apreciar ambas vistas.

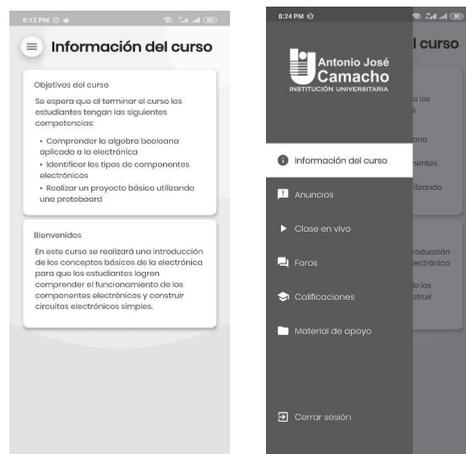


Figura 24. Información del curso.
Fuente: los autores.

Anuncios del curso: los anuncios son información publicada por el docente para el curso, al ingresar a la opción se muestra la vista de la Figura 25.

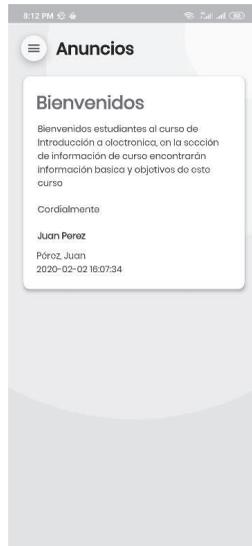


Figura 25. Información del curso.
 Fuente: los autores.

Foros: esta opción permite al estudiante visualizar las publicaciones o realizar posts en temas relacionados al curso para así lograr tener una comunicación asincrónica con los demás participantes, en la Figura 26 se puede visualizar el módulo de foros desde que se registran los comentarios y se visualizan las respuestas desde el POST, como se muestra en la Figura 26.

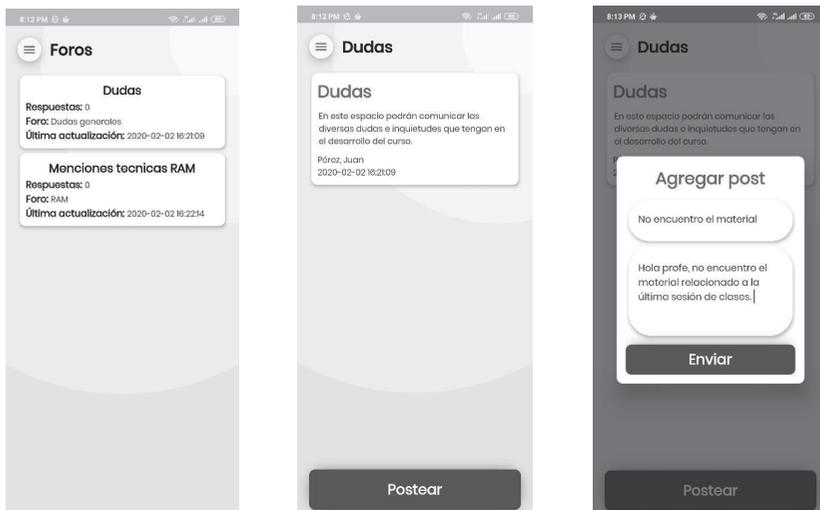


Figura 26. Listado de foros.
 Fuente: los autores.

Clases en vivo y chat con notificaciones push: para este módulo se tuvo que desplegar un servidor de transmisión de servicios multimedia llamado Ant Media que permite transmitir la sesión en vivo creada desde el software libre Open Broadcaster Studio (OBS Studio) a la aplicación móvil. El contenido multimedia que se envía a la aplicación es de protocolo RTMP (Real-Time Messaging Protocol), por lo cual se utiliza la librería GPlayer que permite reproducir este tipo de contenido. Por último, es usada una base de datos en vivo de Firebase para lograr notificar al estudiante sobre los nuevos mensajes del chat. En la Figura 27 se puede apreciar vista del módulo.

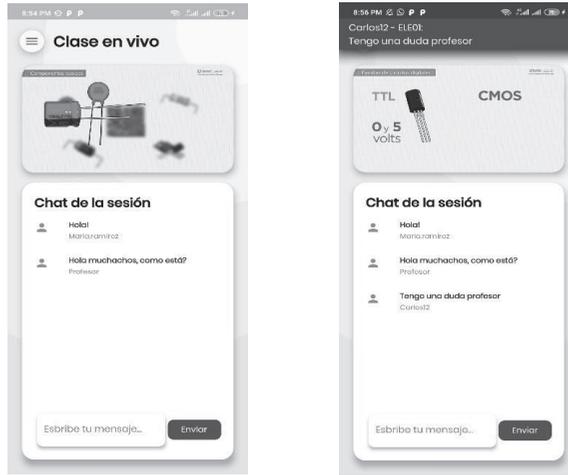


Figura 27. Clase en vivo y chat.
 Fuente: los autores.

Material de apoyo: por medio de este módulo el estudiante puede visualizar los materiales de apoyo del curso, los cuales se encuentran alojados en el LMS. En la Figura 28 se muestra la visualización de los videos y PDF.

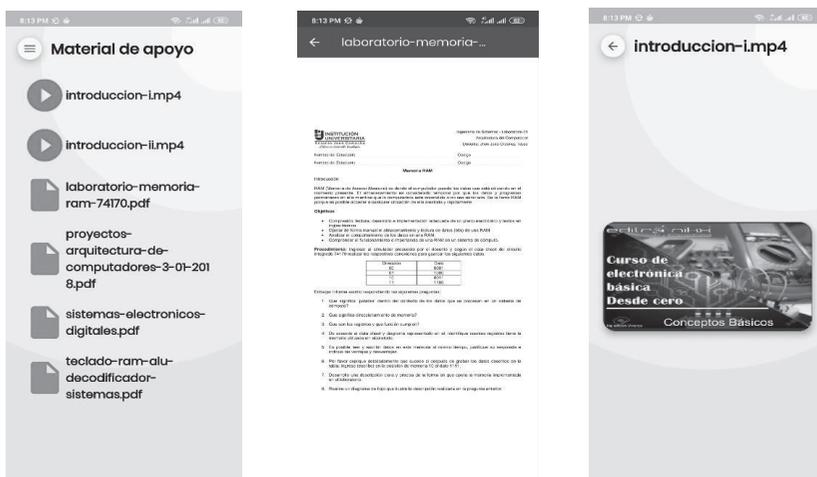


Figura 28. Listado de los materiales de apoyo.
 Fuente: los autores.

Algunos de los aspectos a resaltar en este capítulo en relación con el aporte del proyecto de investigación son:

1. Al haber realizado la caracterización de los componentes didácticos y los estándares de diseño de las soluciones M-Learning se lograron identificar los principales componentes y funcionalidades que permitieron cumplir con las necesidades de una aplicación móvil para los cursos complementarios de la UNIAJC, de esta forma se plantea una referencia para los próximos proyectos vinculados a la integración de la tecnología con la educación ubicua.
2. El uso del patrón de arquitectura para la vista MVVM (Model – View – ViewModel) y el uso del patrón Repository (para el Model), hacen que la lógica del proyecto no esté acoplada a un servicio web en específico, debido a que, para el presente proyecto, aunque se estén consumiendo los servicios que provee el LMS Chamilo, si en algún futuro se cambia de servidor, la aplicación está totalmente desacoplada a ello y, por lo tanto, no sufriría cambios en su funcionamiento.
3. El prototipo de plataforma interactiva actualmente apoya los procesos de formación complementaria de la UNIAJC a través de una aplicación móvil, permitiendo a los estudiantes recibir formación a través de sus dispositivos móviles con cursos cortos afines a sus programas de formación para fortalecer sus conocimientos. Todo esto se logró a partir de la integración una plataforma LMS para la gestión del contenido relacionado con la formación y el consumo de los servicios del LMS por medio de la aplicación móvil.
4. Este desarrollo tecnológico promueve facilitar la administración y divulgación de información entre estudiantes, docentes y compañeros de la institución universitaria, sin necesidad de requerir archivos, libros u otros elementos físicos; desde esta perspectiva, se contribuye al cuidado del medio ambiente en tanto se evita el uso de papel. Por otro lado, esta herramienta hace posible el concepto de aula virtual, el cual se constituye en un espacio interactivo en donde se comparte información entre la comunidad educativa, se disponen bibliotecas virtuales y se dan sesiones formativas en vivo, pero sin recurrir al desplazamiento físico.