



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Computación

**SMARTLEARN: Aplicación móvil para la organización y
exploración eficiente de repositorios de aprendizaje**

Tesis para obtener el grado de
Licenciado en Ingeniería en Ciencias de la Computación

Presenta

DAVID CONSTANTINO GUTIÉRREZ GUTIÉRREZ

Directora de tesis

Dra. María Teresa Torrijos Muñoz

Asesora de tesis

M.I. María del Consuelo Molina García

22 de noviembre de 2024

DEDICATORIA

Esta tesis está dedicada a las pocas personas que representan un enorme lugar en mi corazón, aquellas cuya esperanza prevaleció en todo momento y siempre creyeron en mí. Primeramente, a mi Mamá, que con su amor y cariño siempre supo que lograría o lograré cada una de mis metas, a la Maestra Consuelo Molina, quien fue la primera persona de la facultad en ayudarme, cuando se me presentó el problema con mis materias, y ella prontamente me proporcionó una solución, a la Doctora Tere Torrijos quien me brindo todo su apoyo para poder seguir adelante con mis estudios,

gracias a toda su experiencia y conocimiento es que es posible esta tesis, les agradezco enormemente a cada una de estas personas que, aunque esté escrito aquí, las llevare en el corazón siempre, ya que fueron quienes gracias a todas y cada una de sus acciones, me dieron un enorme motivo para poder concluir hoy mis estudios y que hoy en día el sueño de dejar una marca en mi historia se hiciera realidad.

Muchas gracias a todas.

Índice

INTRODUCCIÓN	5
Capítulo 1	6
Alcance del proyecto	6
1.1 Antecedentes	6
1.2 Planteamiento del problema	7
1.3 Objetivo General	9
1.4 Metodología	10
Capítulo 2	12
Modelado de procesos	12
2.1 Consultar Repositorio	12
2.2 Agregar Repositorio	13
2.2 Modificar Repositorio	14
2.2 Revisar Repositorio	14
Capítulo 3	15
Levantamiento de requerimientos	15
3.1 Requisitos Funcionales	15
3.1.1 Registro de usuario	15
3.1.2 Inicio de sesión	15
3.1.3 Elegir opción	16
3.1.4 Agregar repositorios	16
3.1.5 Consultar repositorio	16
3.1.6 Modificar repositorio	17
3.1.7 Validar repositorio	18
3.2 Requisitos no Funcionales	18
3.2.1 Usabilidad	18
3.2.2 Rendimiento	19
3.2.3 Seguridad	19
3.2.4 Disponibilidad	19

3.2.5 Compatibilidad	19
Capítulo 4	21
Análisis y Diseño	21
4.1 Casos de uso	21
4.2 Diagrama de secuencia para el Usuario	31
Capítulo 5	33
5.1 Framework	33
5.2 Codificación	34
Conclusiones	46
Bibliografía	47

INTRODUCCIÓN

La revolución tecnológica está cambiando nuestras vidas, actualmente podemos participar en conversaciones prácticamente desde cualquier lugar con la infraestructura tecnológica que lo hace posible. En el ámbito educativo, los estudiantes están dejando a un lado los medios como libretas y lápiz, y están sustituyéndolos por herramientas como computadoras portátiles y teléfonos móviles; esta situación motiva el aprovechamiento de estos recursos electrónicos y de la tecnología que los soporta para diseñar nuevas estrategias, en búsqueda de mejorar el proceso de aprendizaje.

En este sentido, el aprendizaje electrónico móvil o m-learning (*mobile learning* en inglés), es una estrategia educativa que soporta la utilización de tecnología móvil, a fin de facilitar el aprendizaje en cualquier momento y lugar, apegándose al paradigma de aprendizaje “anytime and anywhere”.

Los estudiantes de la carrera de Ingeniería en Ciencias de la Computación, que se imparte en Ciudad Universitaria de la BUAP, cuentan con dispositivos móviles personales y también cuentan con acceso a internet inalámbrico y cableado, como un servicio administrado por la Dirección General de Cómputo y Tecnologías de la Información y Comunicaciones (DCyTIC). Lo anterior, aunado a que hoy en día un gran porcentaje de ellos usa sus dispositivos para actividades académicas LAVÍN ZATARAIN, ZALDÍVAR COLADO, RODELO MORENO, & ZALDÍVAR MARTÍNEZ, (2019), genera las condiciones necesarias y abre la oportunidad para desarrollar el prototipo de una aplicación móvil que funcione en cualquier smartphone y que integre, en el mismo, repositorios y metadatos de información útil para complementar el aprendizaje en el aula de las materias que cursan los estudiantes de Ingeniería en Ciencias de la Computación.

Capítulo 1

Alcance del proyecto

1.1. Antecedes

En la investigación realizada por VACA ORELLANA, NARANJO TORO, BESANTES ANDRADE, & BENÍTEZ BASTIDAS, (2016), pág. 11, se enmarcan los elementos para el diseño de recursos de aprendizaje para móviles considerando en la capa exterior los objetivos, contenidos, actividades de aprendizaje, elementos de conceptualización y evaluación; y en la capa interior, la accesibilidad, portabilidad, interoperabilidad, durabilidad y reusabilidad del objeto de aprendizaje. Asimismo, resaltan la importancia de la intención del aprendizaje, el uso de estándares y concluyen que no hay diferencia significativa en el aprovechamiento de los estudiantes que consultaron objetos de aprendizaje en otros medios y objetos de aprendizaje en dispositivos móviles.

RAMÍREZ ARELLANO, RODRÍGUEZ MANCERA, & ACOSTA GONZAGA, (2014), pág. 2, realizaron un guía útil para poder construir un objeto de aprendizaje para los dispositivos móviles, el cual tiene una intención de aprendizaje y que además cumple con las siguientes características: Reusable, Interoperable, Modular, Accesible, Portable. En su guía se menciona la integración de sistemas que utilizaron en AOM para que este cumpliera con ser de aprendizaje, un sistema que utilizaron fue un Sistema de Administración del Aprendizaje (LMS por sus siglas en Inglés) el cual mantiene comunicación con AOM para compartir información acerca de la interacción del estudiante con la misma. Esta comunicación se lleva a cabo mediante la API ECMAScript de la IEEE.

CALDERÓN LOEZA & SÁNCHEZ ESCOBEDO (2021), pág. 3, analizaron el impacto del uso de dispositivos móviles en el aprendizaje. Realizaron una comparación del impacto en el aprendizaje con dispositivos móviles en adolescentes de entre 10 a 21 años, para dar a conocer si el impacto es positivo, neutro o negativo. En su investigación descubrieron que, de acuerdo con diversos estudios realizados por otros investigadores dio como resultado que el 70% presentan un impacto positivo en el aprendizaje; 20% un impacto neutro; y menos del 10% un impacto negativo, concluyendo así un gran cambio tras la pandemia COVID-19 en el mundo.

CRUZ BARRAGÁN & BARRAGÁN LÓPEZ (2014), pág. 1, realizaron una investigación con el objetivo de describir la experiencia del aprendizaje móvil (m-learning) que es una forma de enseñanza a través de los dispositivos móviles. Dicha experiencia relata que fue obtenida durante un curso en enfermería, donde los alumnos pudieron interactuar con Apps educativas y gratuitas, para enfermería, en sus dispositivos móviles. En esta investigación relatan como esta experiencia fue positiva ya que se pudo observar cómo los estudiantes tuvieron apertura a la interactividad y participación en la actividad, principalmente en la investigación para la construcción de su aprendizaje, ya que descubrieron que pueden usar sus dispositivos móviles como un apoyo para su formación académica a través de las Apps educativas.

1.2. Planteamiento del problema

El programa educativo de Ingeniería en Ciencias de la Computación (ICC) de la BUAP se imparte en modalidad escolarizada, está programado para que el estudiante concluya en 10 semestres y en cada semestre se cursan entre 3 y 5 asignaturas disciplinares.

En el programa de cada asignatura se establece el número de horas que se deben dedicar para abordar la teoría y para realizar las prácticas bajo la conducción del docente, particularmente para las asignaturas disciplinares se establece que, para propiciar el aprendizaje en el estudiando y lograr el objetivo propuesto, se requiere de 3 horas de teoría y 2 horas de práctica a la semana, más el trabajo independiente del estudiante en función de sus conocimientos, aptitudes y habilidades.

El rendimiento académico es considerado una medida de la capacidad de respuesta del individuo, que expresa en forma de estimación lo que una persona ha aprendido como resultado de un proceso de instrucción o formación VERGARA MESA, (2011).

De acuerdo con GAETA GONZÁLEZ & CAVAZOS ARROYO, (2016) y con MARCÉN & MARTÍNEZ CARABALLO, (2012), los estudiantes con promedio más alto dedican más tiempo al estudio, adicional al que se invierte en el aula, además hacen uso de estrategias para autorregular este tiempo.

En este sentido, el interés auténtico que tiene los estudiantes de la carrera de ICC de la BUAP en su formación académica, muchas veces, los lleva a invertir más tiempo para lograr el desarrollo exitoso de sus ejercicios, tareas o proyectos y como consecuencia obtener mejores promedios. Sin embargo, frecuentemente se genera frustración y un doble esfuerzo, cuando el tiempo que invierten en encontrar los recursos confiables que les ayuden al desarrollo de su actividad es mayor o igual al tiempo real que invierten en el desarrollo propiamente de la actividad.

Por otra parte, considerando que el 78 % de la población mundial de diez años en adelante tiene un teléfono celular (Organización de las Naciones Unidas, 2023), que en México, el porcentaje de las personas usuarias de teléfono celular equivale a 79.2% (Instituto Nacional de Estadística y Geografía, 2023) que todos los estudiantes activos de ICC cuentan con al menos un dispositivo móvil y que, además cuentan con acceso a internet inalámbrico y cableado, como un servicio administrado por la Dirección General de Cómputo y Tecnologías de la Información y Comunicaciones (DCyTIC) de la BUAP; y considerando también que, hoy en día, un gran porcentaje de estudiantes usa sus dispositivos móviles para actividades académicas (Lavín, et. al, 2019), se generan las condiciones necesarias y se abre la oportunidad para desarrollar soluciones específicas, en el marco de los programas de asignatura, que contribuyan a optimizar el tiempo y disminuir la frustración y el esfuerzo en la búsqueda de recursos de aprendizaje que ayuden a los estudiantes para que puedan realizar sus ejercicios, tareas y proyectos de forma efectiva, sin distraerse de su objetivo haciendo uso de su dispositivo móvil.

1.3. Objetivo General

En el escenario donde los estudiantes de ICC cuentan con un dispositivo móvil y también cuentan con el servicio de Internet proporcionado por la institución educativa, se busca desarrollar un prototipo de aplicación móvil que integre, en el mismo, repositorios y metadatos de información útil para complementar el aprendizaje en el aula de las asignaturas de su programa educativo.

Se pretende que la aplicación funcione como herramienta de autoapoyo y que optimice el tiempo que invierten en el desarrollo de una actividad, proyecto o tarea a realizar, de forma

individual o en equipo. Se espera que el uso de esta aplicación ayude al estudiante en su trayectoria académica y contribuya a la mejora del aprendizaje proporcionando la información precisa y detallada, de estudiante a estudiante (E2E).

1.4. Metodología

El uso de una metodología formal para el desarrollo de una aplicación móvil es fundamental para organizar las tareas, administrar eficientemente la complejidad y controlar el desarrollo para obtener al final, el producto requerido.

Para el desarrollo de la aplicación se hará uso de Scrum, esto con base al estudio realizado por MOLINA RIOS, HONORES TAPIA, PEDREIRA SOUTO, & PARDO LEÓN, (2021), en el que se precisa que, en el desarrollo de aplicaciones móviles, no es posible hacer uso de cualquier metodología ágil debido a la parte estructural que generalmente se encuentra enfocada al desarrollo de software o sistemas web y concluye que la metodología Scrum se considera con mayor frecuencia para el desarrollo de aplicaciones móviles.

Scrum es una metodología formal de tipo ágil, permite adaptar la aplicación a las necesidades que van surgiendo lo que facilita la construcción, y es una metodología incremental que considera ciclos cortos y rápidos lo que permite un buen control al ir incorporando poco a poco funcionalidad al producto final.

Scrum se centra en crear productos adecuados por lo que su nivel de flexibilidad es mayor con respecto a otras metodologías más rígidas, la propiedad y transparencia se traslada a todos los involucrados en el proyecto, la resolución de problemas es eficiente por la autonomía de cada

integrante del equipo. Por otra parte, no menos importante, se encuentra el nivel de madurez para plantear la solución al problema planteado.

La presente investigación requiere de la implementación de las áreas que conforman una aplicación móvil (framework, arquitectura, lógica de programación, diseño de interfaz) para desarrollo de una Aplicación Móvil que se nombra ESUN como acrónimo de *Estudiante Universitario*.

Hay que aprovechar todas las técnicas, herramientas y tecnologías que existen para poder cubrir a la enorme cantidad de estudiantes en cuanto a asesoramiento o consulta sobre alguna problemática que afecta a su proceso en el desarrollo educativo, las dudas surgen a diversas horas del día más cuando se realizan trabajos como: tareas, proyectos, investigación. Para poder solventar esta situación y minimizar este problema, se recurre a lo que se encuentra a nuestro alcance con la tecnología hoy presente y que podemos llevar a todos lados, formando ya parte de nuestro día a día casi las 24 horas, con lo anterior se refiere a los teléfonos celulares, que hoy en día la gran mayoría de la población tiene la posibilidad de adquirir al menos un dispositivo.

De lo anterior, se pretende llevar a cabo el desarrollo de una aplicación móvil la cual permitirá poder almacenar repositorios de datos e información útil para su posterior consulta, en cualquier tiempo y en cualquier lugar. La razón del por qué se pensó en una aplicación móvil y no en otro tipo de aplicación como web, es porque es más sencillo portar un smartphone que un dispositivo portátil como lo es una laptop, o una tableta. Hoy en día, si bien la consulta de información puede ser a través de blogs web, páginas web, artículos de científicos, foros, etc., esa misma información muy probablemente no cumpla con los puntos específicos del tema que se está estudiando para

propósitos específicos o bien puede que la información sea errónea, y esto provoca pérdida de tiempo valiosa que bien pudiera ser sustituida por información que puede ser consultada, testeada, avalada por autores creadores específicos del problema, volviéndola una información veraz.

Capítulo 2

Modelado de procesos

Para el modelado de procesos se hizo uso de la nomenclatura estándar para modelado de procesos BPMN y como herramienta el software Bonita Soft

2.1 Proceso: Consultar Repositorio

El proceso para poder hacer una consulta de un repositorio es similar en ambos casos, ya sea para un usuario, o para un docente, es necesario autenticarse y si las credenciales de autenticación son correctas entonces se despliega la GUI de inicio a partir de ahí se muestran tanto las materias como los docentes asociados a cada repositorio desplegado en un listado, el cual puede ser seleccionado siguiendo la trayectoria tal y como se muestra en la figura 1 usando BPMN.

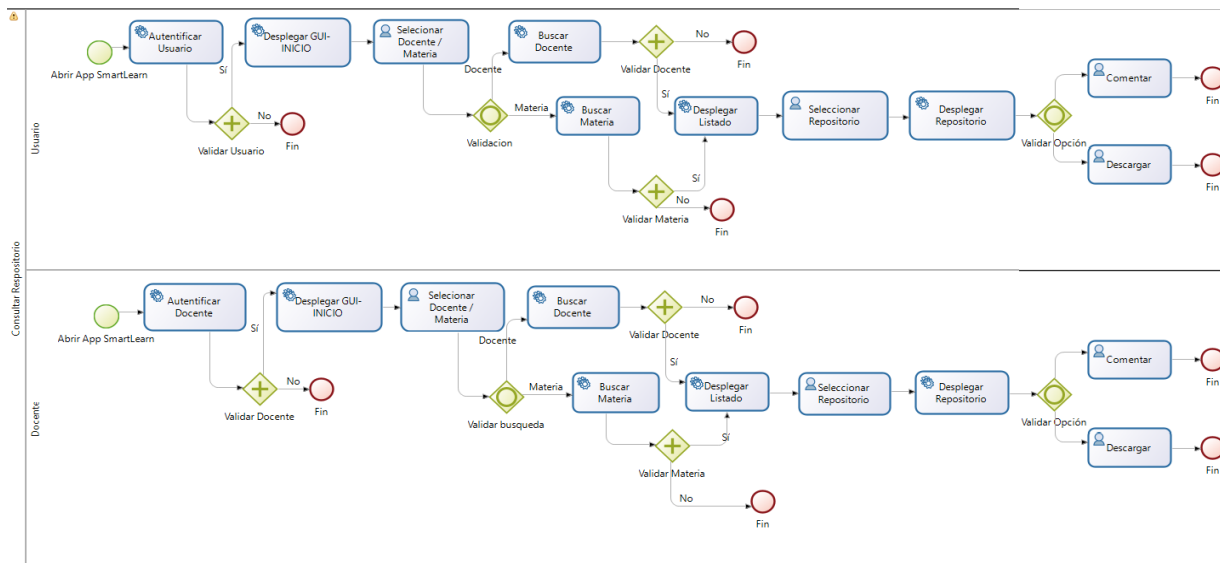


Figura 1. Proceso consultar un repositorio

2.2 Proceso: Agregar Repositorio

Para poder Agregar un repositorio similar al proceso anterior, es necesario que ya sea el usuario o docente sea autentiquen, una vez validadas las credenciales se desplegará la GUI el cual les muestra un menú del cual se selecciona la opción de Agregar Repositorio, al terminar la edición se procede a guardar y se publica siguiendo la trayectoria como se muestra en la figura 2. Usando BPMN.

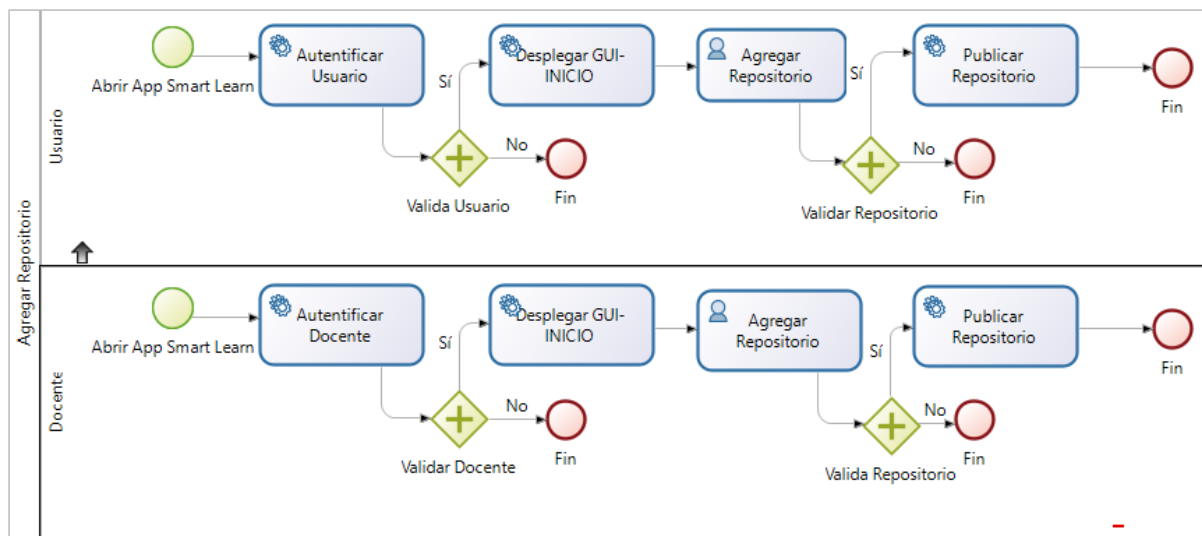


Figura 2. Proceso agregar un repositorio

2.3 Proceso: Modificar Repositorio

Para poder modificar un repositorio es necesario que el usuario o docente se autentiquen, una vez validadas sus credenciales el sistema GUI se despliega mostrando el menú de opciones, vamos a búsqueda del repositorio que se desea modificar, y posteriormente se editan los campos que se requieren cambiar, terminando la actualización de los datos al guardar, automáticamente se publican los resultados siguiendo la trayectoria de la figura 3, usando BPMN.

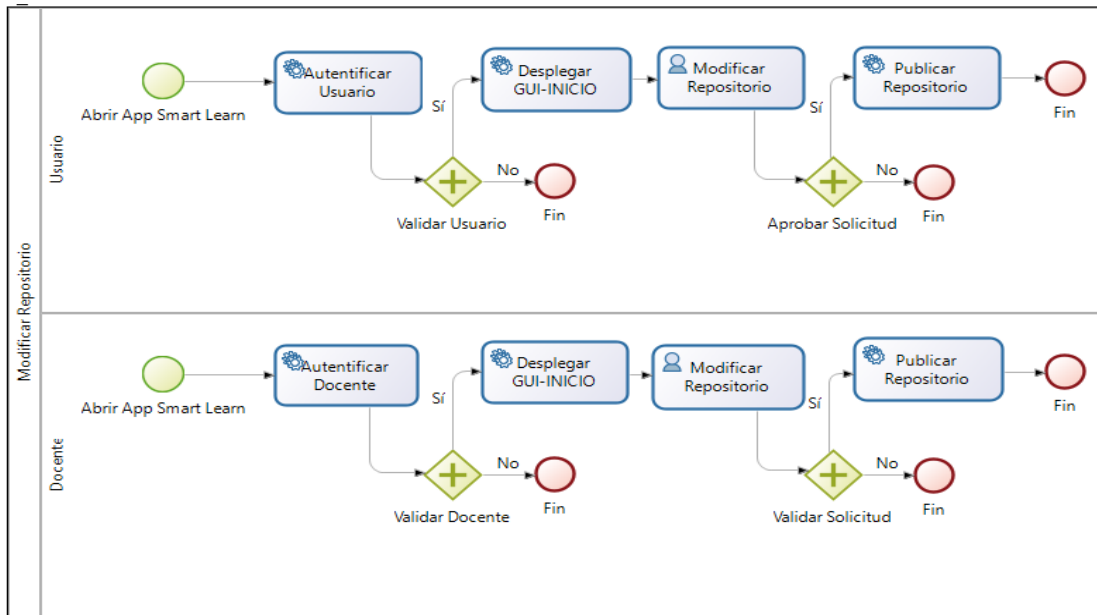


Figura 3. Proceso modificar un repositorio

2.4 Proceso: Revisar Repositorio

El proceso para la validación de un repositorio no está habilitada para usuarios, sino para docentes los cuales deben autenticarse, una vez que se validen las credenciales se despliega el GUI del sistema mostrando los repositorios que pueden ser revisados y actualizados, una vez terminada su edición y guardando los cambios se publica automáticamente, estos pasos siguen la trayectoria de la figura 4, usando BPMN.

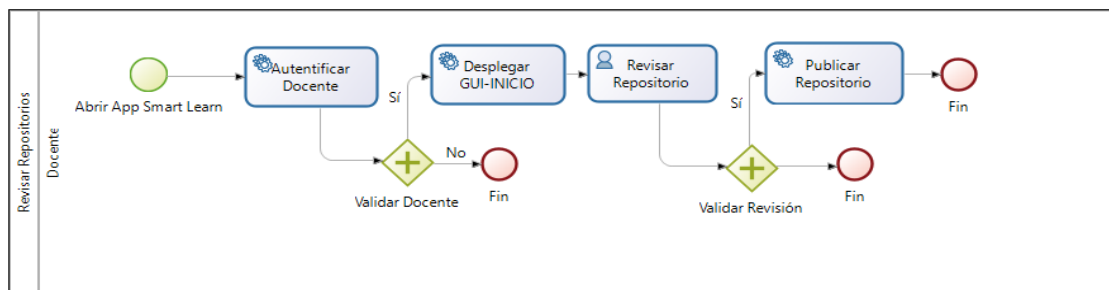


Figura 4. El diagrama muestra el proceso que lleva revisar un repositorio

Dentro de todo este proyecto de tesis se han seleccionado tecnologías muy específicas por practicidad, facilidad de entendimiento de lenguaje reservado, así como también la facilidad de su interconectividad a nivel de producción.

Capítulo 3

Levantamiento de requerimientos

Se llevó a cabo conforme al estándar IEEE 830, el cual proporciona directrices para la especificación de requisitos de software, así mismo, este documento contiene información detallada sobre aspectos como los requisitos funcionales y requisitos no funcionales que a continuación se mencionan.

3.1. Requisitos Funcionales

3.1.1. Registro de usuario

- El sistema debe permitir a usuarios registrarse proporcionando un nombre de usuario, una dirección de correo electrónico y una contraseña.
- Para usuarios Docentes es importante llenar un campo más que es la cédula profesional.
- Después del registro, el usuario debe recibir un correo electrónico de confirmación.
- El sistema debe validar que la dirección de correo electrónico, el nombre y la cédula (en el caso Docente) proporcionada sea única en la base de datos.

3.1.2. Inicio de sesión:

- Los usuarios registrados deben poder iniciar sesión utilizando el correo electrónico y contraseña.

- El sistema debe autenticar las credenciales del usuario y permitir el acceso a la aplicación si son correctas.

3.1.3. Elegir opción

- Una vez que el usuario haya validado sus credenciales, el sistema debe mostrarle al usuario diferentes opciones dependiendo del tipo de usuario que ha iniciado sesión puede ser usuario estudiante o usuario docente.
- El usuario estudiante será capaz de ver 3 opciones: agregar repositorios, consultar repositorios, modificar repositorios.
- El usuario docente será capaz de ver 4 opciones: agregar repositorios, consultar repositorios, modificar repositorios y validar repositorios.

3.1.4. Agregar Repositorios

- Los usuarios deben poder crear nuevos repositorios especificando un título de la materia, un nombre del docente, descripción, sección, observaciones o anotaciones, agregar archivos, también debe cumplir ciertas restricciones, tales como: la sección de la materia que no puede ser repetido en caso de existir un repositorio con el mismo nombre, deben ser omitidos los caracteres especiales y palabras que sean ofensivas.
- Al terminar de agregar el repositorio y guardarlo, el sistema se encarga de subir los archivos a la base de datos, para posteriormente mostrarlos en las secciones de consultar y modificar repositorios.

3.1.5. Consultar Repositorio

- El sistema debe mostrar al usuario una lista de todos los repositorios agregados hasta la fecha actual de consulta, el usuario podrá elegir entre toda esta lista el que sea de su interés.

- El sistema también muestra al usuario un apartado de búsqueda, donde el usuario puede filtrar una búsqueda por docente o por materia. Aun que un docente puede dar varias materias, también las materias pueden ser impartida por más de un docente por ello es, que podemos encontrar más de un repositorio con el mismo nombre el filtro ayuda a reducir el tiempo de búsqueda ya que también se muestra la sección y fecha en la que fue impartido para encontrar más rápido la información (en caso de que el repositorio exista).
- El sistema mostrará al usuario toda la información contenida del repositorio una vez que el usuario haya elegido de la lista o del filtro de búsqueda una sola materia o docente, así como todas las anotaciones que haya indicado el usuario que haya subido el repositorio.
- El sistema también es capaz de brindarle al usuario la capacidad de poder descargar todo el paquete de archivos desde la base de datos.

3.1.6. Modificar Repositorio

- Si el usuario decide entrar al apartado de modificar repositorio, el sistema es capaz de la misma manera que en la sección de consulta de repositorio, mostrarle al usuario un listado de todos los docente y materias que han sido agregados hasta la fecha actual de consulta.
- El sistema es capaz también de mostrarle al usuario un repositorio específico usando el apartado de búsqueda, siempre y cuando el repositorio filtrado se encuentre en la base de datos. Una vez que el usuario ha encontrado el repositorio que requiere, al seleccionarlo pasara a una pantalla donde estarán todos los datos del repositorio y en esa misma sección encontrará un botón que indica modificar, al seleccionarlo

todos los datos pasaran a una pantalla más donde se podrá modificar todo lo que ya se encuentre en la base de datos actualmente.

- Una vez que el usuario haya modificado o actualizado los datos que requirió modificar, puede dar en el botón guardar, y el sistema debe mostrarle al usuario que los datos han sido guardados con éxito y que debe revisar en su correo electrónico, ya que ahí se le enviará el estatus de su modificación al repositorio.

3.1.7. Validar Repositorio

- Esta sección es únicamente visible para el usuario docente, un repositorio necesitará la validación de un docente siempre y cuando se cumpla lo siguiente: el repositorio agregado no ha sido por él, o el repositorio ha sido modificado.
- Si el repositorio no fue agregado por el usuario docente, y éste mismo ya se encuentra registrado en el sistema, le llegará una notificación al iniciar sesión, esta misma notificación podrá llevarlo a donde se encuentra lo que requiere validar el sistema, si todo es correcto y el usuario docente valida el repositorio, el sistema enviará un mensaje por correo al usuario que agregó el repositorio indicándole que el repositorio ya se encuentra publicado.
- Si el repositorio es modificado el sistema mostrará al usuario docente una notificación donde espera su aprobación o validación, si el usuario docente aprueba dicho repositorio sucederá el mismo proceso anterior.
- Si por el contrario el repositorio ya sea agregado o modificado, el usuario docente lo invalida, entonces el sistema enviará un email, indicándole al usuario que el repositorio no cumple con las especificaciones para ser publicado.

3.2. Requisitos no Funcionales

3.2.1. Usabilidad

- La interfaz de usuario debe ser intuitiva y fácil de usar, con un diseño limpio y claro.
- El tiempo de carga de la aplicación no debe exceder los 3 segundos en cualquiera de sus vistas.

3.2.2. Rendimiento

- La aplicación debe ser capaz de manejar al mismo tiempo al menos a 50 usuarios activos sin experimentar degradación del rendimiento.
- El sistema debe ser escalable para admitir un aumento del 20% en al menos 6 meses a partir de su lanzamiento.

3.2.3. Seguridad

- La información de inicio de sesión y los datos de las tareas debe estar protegidos mediante el cifrado FDE (Full Disk Encryption o cifrado de disco completo).
- El sistema debe implementar medidas de autenticación y autorización adecuadas para garantizar que solo los usuarios autorizados puedan acceder a ciertas funciones o datos.

3.2.4. Disponibilidad

- La aplicación debe tener una disponibilidad del 99.9% durante las horas de trabajo normales.
- Se deben realizar copias de seguridad diarias de la base de datos, y estas copias de seguridad deben mantenerse durante al menos 30 días.

3.2.5. Compatibilidad

- La aplicación es compatible a partir de la versión kit kat 4.4, no es compatible con IOS.
- El espacio de almacenamiento para la aplicación es de un mínimo de 1.5 GB para su correcto funcionamiento.

- La aplicación debe proporcionar una experiencia de usuario adecuada en pantallas de diferentes tamaños.

Capítulo 4

Análisis y Diseño

4.1. Casos de uso

La fig. 5 nos muestra los casos de uso de ambos usuarios, y como comparten las funcionalidades del sistema, tanto en gestión de usuarios, como en gestión de repositorios.

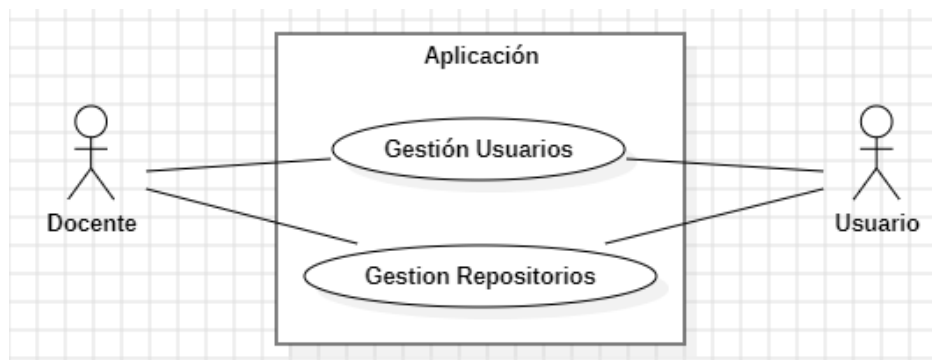


Figura 5.

CU1. Gestión de Usuarios

Para que los usuarios puedan hacer uso del sistema es necesario cumplir con un conjunto de funcionalidades y procesos implementados en la aplicación para poder administrar la autenticación, autorización y el perfil de los usuarios que interactúan con la misma.

La fig. 6 nos muestra algunos aspectos clave de la gestión de usuarios en el sistema.

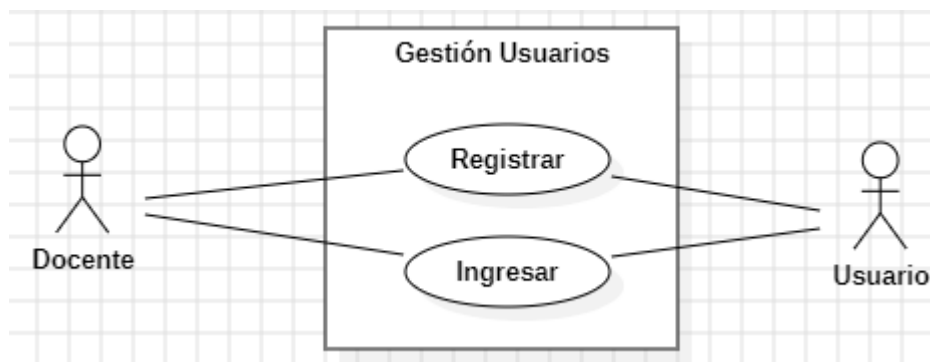


Figura 6.

1.1 Registrar

Este apartado es únicamente para los usuarios nuevos ya sean usuarios docentes o usuarios estudiantes, requieren proporcionar datos a la aplicación como son: nombre, correo electrónico, una contraseña, adicional a esto y únicamente para usuarios docentes, es necesario llenar el apartado de cédula profesional, esto con el fin de poder asociar las demás actividades que puede realizar un usuario dentro de la aplicación.

El nombre es para ser mostrado en la sección de comentarios, este campo es requerido y no debe mostrarse nombres ofensivos, el correo electrónico es para que una vez registrado el usuario pueda ser autenticado mediante el mismo, ya que se le hará llegar un email, pidiendo que acceda a un enlace para corroborar que se trata de él, y una contraseña para asegurar el resguardo de su cuenta y el uso de esta.

1.2 Prerrequisitos

1.2.1 Tener un dispositivo móvil con almacenamiento de 1.5 GB para poder instalar la aplicación.

1.2.2 El dispositivo móvil debe contar con un mínimo de memoria RAM de 2GB.

1.2.3 La versión requerida en el dispositivo debe ser Android 10 o superior.

1.2.4 El dispositivo debe contar con conexión a internet estable.

1.3 Flujo Básico 1 (FB1)

La fig.7 muestra la secuencia que lleva el registrar un nuevo usuario, mostrado en la pantalla de inicio de sesión.

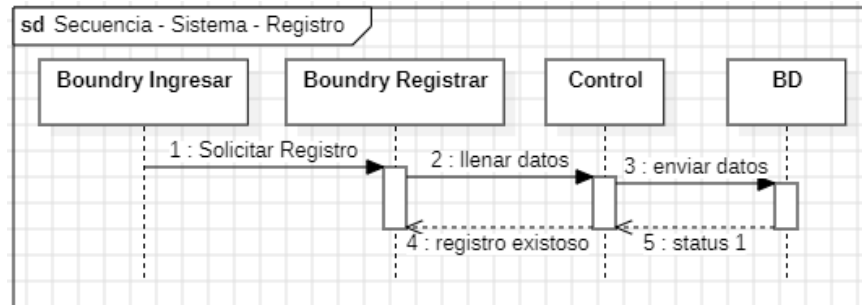


Figura 7.

1.3.1 Solicitar Registro.

1.3.2 Llenar formulario de registro.

1.3.3 Si el email NO está duplicado en el sistema continuo en **FB1-1.3.5**.

1.3.4 Si el email está duplicado entonces **1.5 FE1**.

1.3.5 Registro exitoso.

1.3.6 Iniciar sesión o salir.

1.4 Flujo por Excepción 1 (FE1)

En caso de existir ya en la base de datos un perfil con un email ya registrado y el usuario nuevo quiere registrar el mismo correo, la fig. 8 nos muestra la secuencia de dichas acciones.

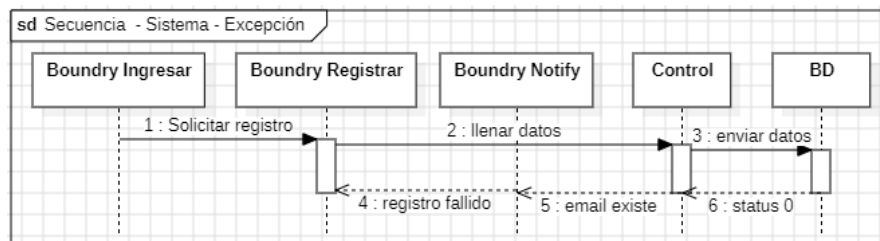


Figura 8.

- 1.4.1 Solicitar registro.
- 1.4.2 Llenar formulario de registro.
- 1.4.3 Enviar datos.
- 1.4.4 Notificación: el email ya se encuentra registrado en el sistema, proporcione otro email.
- 1.4.5 Regresar **FB1-1.3.1** o salir.

1.5 Flujo Básico 2 (FB2)

La fig. 9 únicamente la secuencia que se realiza a elegir la opción de ingresar al sistema.

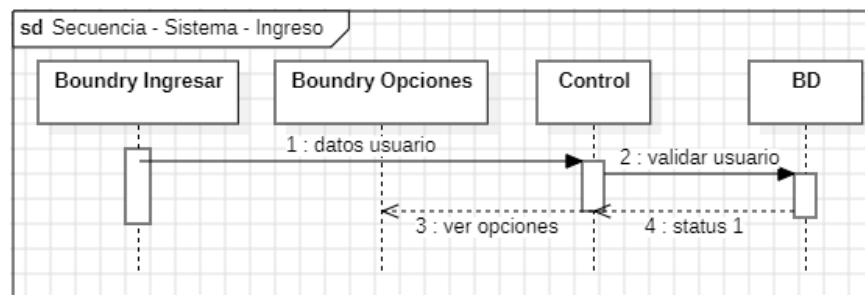


Figura 9.

- 1.5.1 Iniciar sesión con sus datos (email y contraseña).
- 1.5.2 Enviar datos.
- 1.5.3 Si los datos son correctos, continuo **FB2-1.5.5**.
- 1.5.4 Si los datos NO son válidos entonces **FE2**.
- 1.5.5 Sistema muestra boundry de opciones al usuario entre ellas, la opción de salir del sistema.

1.6 Flujo por Excepción 2 (FE2)

En caso de que los datos para el inicio de sesión no correspondan a datos que ya estén registrados en el sistema, ocurrirá la secuencia que se muestra en la fig. 10.

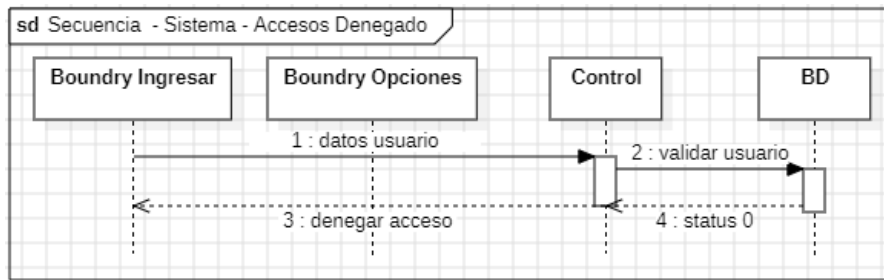


Figura 10.

- 1.6.1 Iniciar sesión con sus datos (email y contraseña).
- 1.6.2 Enviar datos.
- 1.6.3 Acceso Denegado, el usuario o la contraseña no son válidos.
- 1.6.4 Regresar a **FB2-1.5.1** o salir.

CU2. Gestión Repositorios

Los repositorios pueden ser manipulados por los usuarios con diferentes acciones que brinda el sistema, a continuación, la fig. 11 nos muestra el caso de uso que tienen las opciones que nos muestra el sistema.

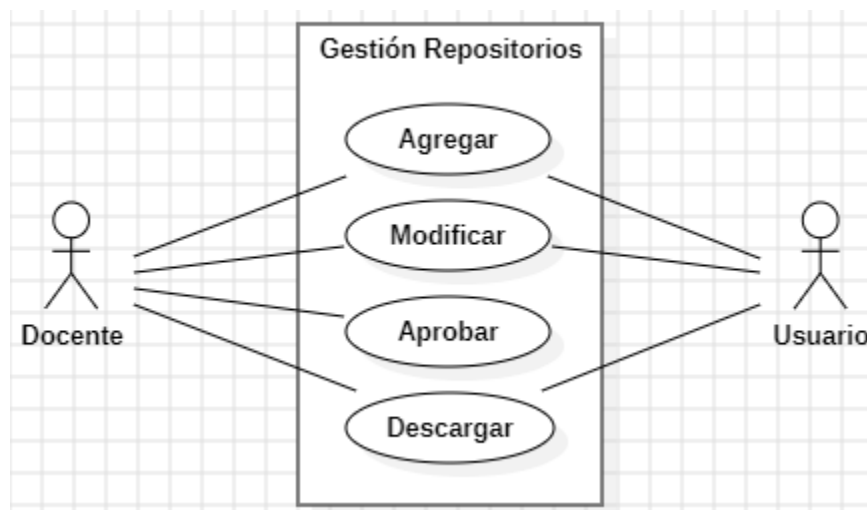


Figura 11.

2.1 Agregar

Esta acción está dentro de las opciones mostradas por el sistema a los dos tipos de usuario (Usuario Docente / Usuario Estudiante), lo que permite esta opción es poder agregar un nuevo repositorio en el sistema, para poder agregar un nuevo repositorio, es necesario haberse autenticado ya que de no ser así el sistema no mostrará las opciones, una vez que el sistema haya permitido el acceso al usuario y el usuario haya elegido esta opción, el sistema le mostrará un formulario que el usuario deberá llenar, los campos que se requieren como obligatorios se mostrarán con un indicador (*) el cual si se deja vacío, el sistema de forma visual resaltará en color rojo indicando que el campo es obligatorio, al haber terminado el formulario habrá una sección más abajo donde se podrá subir el contenido necesario al repositorio, estos pueden ser archivos que sean únicamente visibles, como office, PDF, imágenes, etc., así como también, los programas hechos por los usuarios, con la condición de que no contengan contraseñas, no sean archivos maliciosos, y estén con alguna extensión de archivo de compresión.

2.2 Prerrequisitos

2.3 Flujo Básico 1 (FB1)

Consultar repositorio: una vez que el usuario haya accedido a la aplicación con los datos previamente solicitados, puede elegir entre diferentes opciones, una de ellas es consultar un repositorio, para ello es necesario acceder a este apartado el cual pasará a una vista siguiente, en donde bien se puede filtrar una búsqueda por docente o por materia, si la opción es docente, el usuario debe proporcionar el nombre del docente, si este existe arrojará un resultado en donde el docente podrá tener varios repositorios, esto debido a que un docente puede enseñar diversas materias, lo último que bastará por hacer es seleccionar la materia que el usuario requiera y a continuación se mostrará todos los datos del repositorio el cual podrá descargar si así lo requiere,

también podrá ver un apartado de comentarios, donde verá los comentarios de los demás usuarios, en este mismo apartado el usuario podrá dejar también un comentario si así lo desea, ya que no es obligatorio, por último al final de esta pantalla se muestran dos opciones, solicitar la modificación del repositorio o bien regresar a la pantalla anterior.

3 Modificar Repositorio: una vez que el usuario inició sesión en la aplicación, puede buscar un repositorio, al elegir el deseado y abrirlo, se mostrará todo el contenido del repositorio, así como la sección de comentarios (en caso de que los haya), dentro de esta misma vista existe un apartado que dice modificar repositorio, esto es con el único fin de actualizarlo, completarlo, o bien enviar posibles modificaciones que mejoren el mismo, al hacer click en el mismo, pasaremos a una vista en donde se encuentra un formulario, como si fuera a agregarse por primera vez, al terminar la edición y llenar los correspondientes cambios el usuario podrá enviar los cambios haciendo click en el botón guardar, esta última acción mostrará una notificación en pantalla indicando que se revisa la modificación y una vez terminada se enviará el estatus al correo asociado.

4 Comentar Repositorio: Para poder realizar esta acción es requerido acceder con los datos de inicio de sesión que pide la aplicación, una vez dentro es necesario dirigirse a cualquier repositorio (o el que uno desee revisar) una vez dentro puede observar que ahí se encuentra todo el repositorio, y en la parte de abajo se encuentra una sección de comentarios, en esa misma puede dejar su comentario y al finalizar guardarlo, en cuestión de segundos o minutos dependiendo de la transferencia de datos hacia internet del dispositivo, se verá reflejado el comentario.

Casos de Uso para el Usuario:

El siguiente caso de uso fig.12 muestra las actividades que puede realizar el Docente con la aplicación móvil:

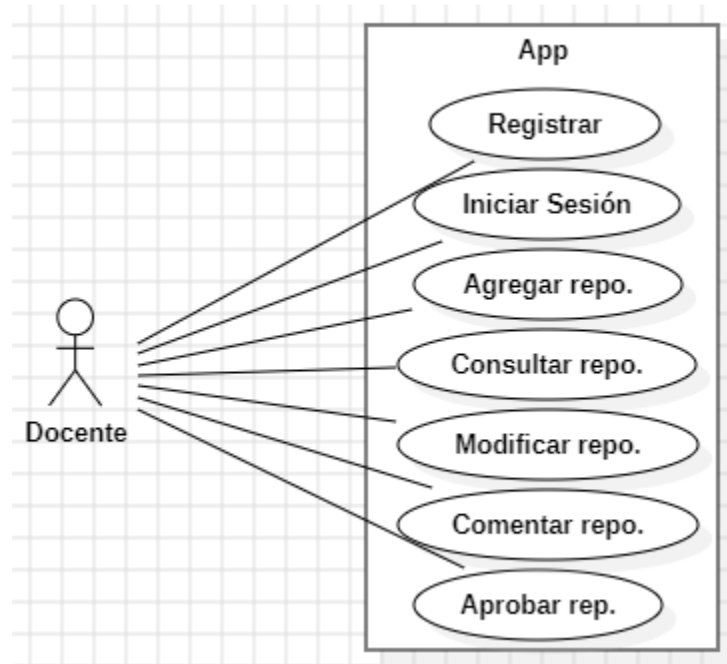


Figura 12.

1. **Registrar:** este apartado es únicamente para los Docentes nuevos, requieren proporcionar datos a la aplicación como son: nombre, correo electrónico, y una contraseña, adicional a esto es necesario que para validar que es un docente, es necesario llenar el campo de cédula profesional, esto con el fin de poder asociar las demás actividades que puede realizar un docente dentro de la aplicación bajo el perfil de profesional de la educación.

El nombre es para ser mostrado en la sección de comentarios, este campo es requerido y no debe mostrarse nombres ofensivos, el correo electrónico es para que una vez registrado el usuario pueda ser autenticado mediante el mismo, ya que se le hará llegar un email, pidiendo que acceda a un link para corroborar que se trata de él, una contraseña para asegurar el resguardo de su

cuenta y el uso de la misma y por último el número de cédula para comprobar que se trata de un profesional de la educación y así garantizar que el repositorio sirva de apoyo a los estudiantes de la carrera ICC (el apartado de la cédula está en propuesta para futuro desarrollo).

- 2. Iniciar sesión:** esta actividad es únicamente para Docentes ya con registro previo, ya que de otro modo será imposible poder acceder a las funciones de la aplicación móvil, el Docente únicamente debe iniciar sesión llenando el formulario con su nombre o su correo, y la contraseña proporcionada en el registro.
- 3. Agregar repositorio:** una vez que el Docente haya accedido a la aplicación con los datos previamente solicitados, puede elegir entre diferentes opciones, una de ellas es agregar un repositorio, para ello es necesario acceder a este apartado el cual pasará a una vista siguiente, donde se le pedirá que llene un formulario, este formulario tendrá varios apartados comenzando por añadir el nombre de la materia, seguido del nombre del docente, fecha en que se está subiendo el nuevo repositorio, y archivos, ya sean en formato de paquetería de Office, o bien PDF, archivos de programas siempre y cuando estén comprimidos con algún empaquetador con extensión .ZIP o .RAR, llenar el apartado de las tecnologías usadas, referencias web o bibliográficas (opcional) y finalmente agregando algún comentario (opcional), en esta misma pantalla al finalizar el repositorio puede ser guardado, o bien cancelar y regresar al apartado anterior, si la opción es guardar, entonces el Docente podrá ver una ventana emergente indicando que su repositorio fue guardado con éxito y que su repositorio fue publicado.
- 4. Consultar repositorio:** una vez que el Docente haya accedido a la aplicación con los datos previamente solicitados, puede elegir entre diferentes opciones, una de ellas es consultar un repositorio, para ello es necesario acceder a este apartado el cual pasará a una vista siguiente, en

donde bien se puede filtrar una búsqueda por docente o por materia, si la opción es docente, el Docente debe proporcionar el nombre del docente, si este existe arrojará un resultado en donde el docente podrá tener varios repositorios, esto debido a que un docente puede enseñar diversas materias, lo último que bastará por hacer es seleccionar la materia que el Docente requiera y a continuación se mostrará todos los datos del repositorio el cual podrá descargar si así lo requiere, también podrá ver un apartado de comentarios, donde verá los comentarios de los demás usuarios, en este mismo apartado el Docente podrá dejar también un comentario si así lo desea, ya que no es obligatorio, por último al final de esta pantalla se muestran dos opciones, solicitar la modificación del repositorio o bien regresar a la pantalla anterior.

5. **Modificar Repositorio:** una vez que el Docente inició sesión en la aplicación, puede buscar un repositorio, al elegir el deseado y abrirlo, se mostrara todo el contenido del repositorio, así como la sección de comentarios (en caso de que los haya), dentro de esta misma vista existe un apartado que dice modificar repositorio, esto es con el único fin de actualizarlo, completarlo, o bien enviar posibles modificaciones que mejoren el mismo, al hacer click en el mismo, pasaremos a una vista en donde se encuentra un llenado, como si fuera a agregarse por primera vez, al terminar la edición y llenar los correspondientes cambios el Docente podrá enviar los cambios haciendo click en el botón guardar, esta última acción mostrará una notificación en pantalla indicando que se revisa la modificación y una vez terminada se enviará el estatus al correo asociado.
6. **Comentar Repositorio:** Para poder realizar esta acción es requerido acceder con los datos de inicio de sesión que pide la aplicación, una vez dentro es necesario dirigirse a cualquier repositorio (o el que uno desee revisar) una vez dentro puede observar que ahí se encuentra todo el repositorio, y en la parte de abajo se encuentra una sección de comentarios, en esa misma puede dejar su comentario y al finalizar guardarlo, en cuestión de segundo o minutos

dependiendo de la transferencia de datos del dispositivo hacia internet, se verá reflejado el comentario.

- 7. Aprobar Repositorio:** Para poder realizar esta acción el Docente deberá haber iniciado sesión con sus credenciales dentro de la aplicación, tendrá a diferencia de un usuario normal, un apartado de notificaciones, donde se mostraran los repositorios que estén a su nombre y que pueda revisar y aprobar, una vez aprobados, estos se verán reflejados en la aplicación junto a todos los demás repositorios, pero si por el contrario el repositorio no es aprobado, el mismo será borrado y al mismo tiempo se notificará a quien haya subido el repositorio.

4.2. Diagrama de Secuencia para el Usuario

El siguiente diagrama fig.13 muestra como seria el proceso o secuencia de acciones que podría realizar el usuario.

1. Diagrama de Secuencia Registrar – Usuario

La secuencia de Registro sigue los siguientes pasos:

- 1- El usuario necesita abrir la aplicación, cuando esto ocurre la aplicación le muestra en la pantalla la sección donde debe registrar sus datos
- 2- El usuario llena el formulario de registro correctamente y al terminar presiona el botón guardar.
- 3- La aplicación se encarga de enviar los datos obtenidos en el registro a una base de datos.
- 4- La base de datos regresa un mensaje a la aplicación, indicando que los datos fueron almacenados correctamente.
- 5- En el apartado de inicio de sesión aparece un mensaje de registro exitoso.
- 6- La aplicación le indica al usuario que revise el correo para corroborar datos.

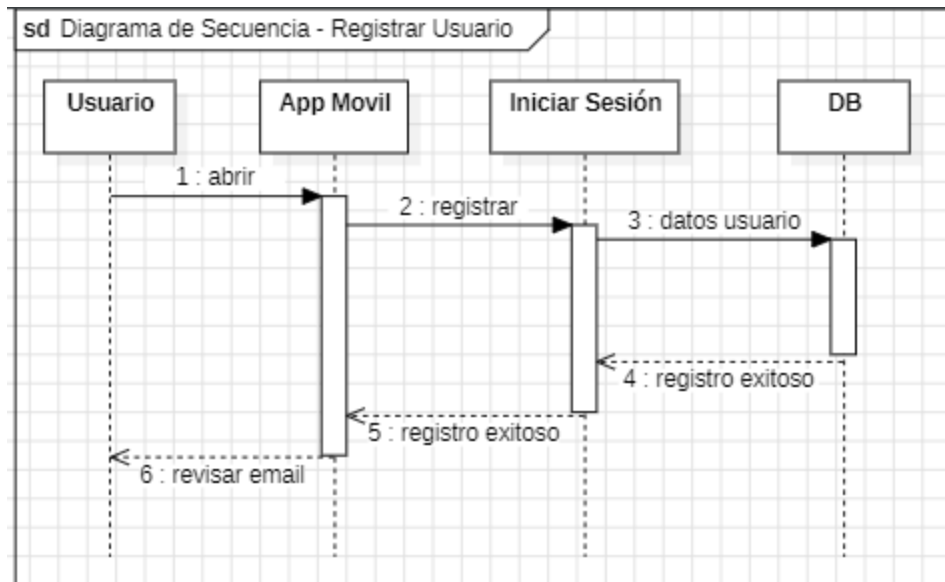


Figura 13.

Capítulo 5

Construcción y pruebas

5.1. Framework

Para la construcción de la aplicación móvil se utilizó de la tecnología Flutter que es un framework de Google y que tiene como lenguaje de uso base a Dart, este lenguaje está basado en C++. El tipo de arquitectura que se utilizó para poder implementar bien las funcionalidades de la aplicación fue la arquitectura limpia.

Adaptar esta arquitectura permitió poder separar los módulos para traer la información del backend de una manera más detallada, de tal manera que, si la API Rest consumida desde la aplicación llega a cambiar, o bien se necesita agregar o adquirir data de otra API no es necesario reestructurar o modificar partes importantes de la aplicación y así simplemente puede modificarse o bien agregarse otro archivo que contenga las características de la API.

El uso de la arquitectura limpia en el proyecto fue de mucha importancia ya que, al usar paquetes para la comunicación de estados entre pantallas si se usaba otro tipo de arquitectura era muy probable que el código aparte de que si ocurría un cambio en la API y tener que modificar gran parte del código para su mantenimiento, sería también difícil poder hacer legible el mismo código, la ventaja de haber usado arquitectura limpia es que el código en cada archivo es más legible, entendible y que todo el código en un solo archivo siempre guarda relación y coherencia con lo que se está haciendo.

5.2. Codificación

A continuación, se agregan capturas de código con la implementación de arquitectura limpia.

Los datos que requiere la API para la creación, modificación, y lectura se encuentran en un solo archivo que fue llamado Repositorio, estos datos gracias a la arquitectura limpia son los que, declarados en este archivo, van a ser manejados a lo largo de toda mi aplicación cuando se genere un nuevo repositorio, se edite o sea leído desde el backend:

A screenshot of a code editor with a dark background and light-colored text. The code is written in Dart and defines a class named 'Repositorio'. The class has several attributes: 'id' (String), 'title' (String), 'docente' (String), 'materia' (String), 'seccion' (int), 'anotacion' (String?), 'comentario' (String?), 'tt' (String), 'archivoComprimido' (List<String>), and 'user' (User?). The constructor 'Repositorio()' is annotated with 'required' for all attributes. The code is numbered from 1 to 26.

```
1 class Repositorio {
2   String id;
3   String title;
4   String docente;
5   String materia;
6   int seccion;
7   String? anotacion;
8   String? comentario;
9   String tt;
10  List<String> archivoComprimido;
11  User? user;
12
13  Repositorio({
14    required this.id,
15    required this.title,
16    required this.docente,
17    required this.materia,
18    required this.seccion,
19    required this.anotacion,
20    required this.comentario,
21    required this.tt,
22    required this.archivoComprimido,
23    this.user,
24  });
25
26 }
```

Fig. 14 Repositorio.dart

Así como repositorio también para almacenar datos del usuario como el inicio de sesión o el registro, fue necesario agregar un archivo que pudiera manejar todos los datos solicitados por mi API y gracias a la arquitectura limpia se detalla de una manera muy similar a la de Repositorio.dart:

```
1 class User {
2
3   final String id;
4   final String email;
5   final String fullName;
6   final List<String> roles;
7   final String? cedula;
8   final String token;
9
10  const User({
11    required this.id,
12    required this.email,
13    required this.fullName,
14    required this.roles,
15    this.cedula = '',
16    required this.token,
17  });
18
19  bool get isAdmin {
20    return roles.contains('admin');
21  }
22
23
24 }
```

Figura 15. User.dart

Para poder comunicar cada módulo dentro de la aplicación, fue necesario hacer uso la tecnología de Riverpod que se caracteriza por su manejo de estados reactivos dentro de aplicaciones móviles, su adaptación es sumamente sencilla al igual que su implementación una vez definidas sus características.

Esos archivos se componen de tres etapas, la primera es el estado:



```
1 class RepositoriosState {
2
3     final bool isLastPage;
4     final int limiteEntidades;
5     final int pagina;
6     final bool isLoading;
7     final List<Repositorio> repositorios;
8
9     RepositoriosState({
10         this.isLastPage = false,
11         this.limiteEntidades = 10,
12         this.pagina = 0,
13         this.isLoading = false,
14         this.repositorios = const[],
15     });
16
17     RepositoriosState copyWith({
18         bool? isLastPage,
19         int? limiteEntidades,
20         int? pagina,
21         bool? isLoading,
22         List<Repositorio>? repositorios,
23     }) => RepositoriosState(
24         isLastPage: isLastPage ?? this.isLastPage,
25         limiteEntidades: limiteEntidades ?? this.limiteEntidades,
26         pagina: pagina ?? this.pagina,
27         isLoading: isLoading ?? this.isLoading,
28         repositorios: repositorios ?? this.repositorios,
29     );
30
31 }
```

Figura 16. Repositorio State

La segunda etapa es la de notificación, que es quien comunicará los estados y propagará la información por toda la aplicación.

```
1 class RepositoriosNotifier extends StateNotifier<RepositoriosState> {
2
3   final RepositoriosRepository repositoriosRepository;
4
5   RepositoriosNotifier({
6     required this.repositoriosRepository
7   }): super( RepositoriosState() ){
8     loadNextPage();
9   }
10
11   Future<bool> createOrUpdateRepositorio( Map<String,dynamic> repositorioLike ) async {
12
13     try {
14       final repositorio = await repositoriosRepository.createUpdateRepositorio(repositorioLike);
15       final isRepositorioInList = state.repositorios.any((elemento) => elemento.id == repositorio.id);
16       if(!isRepositorioInList){
17         state = state.copyWith(
18           repositorios: [...state.repositorios, repositorio],
19         );
20         return true;
21       }
22
23       state = state.copyWith(
24         repositorios: state.repositorios.map(
25           (elemento) => (elemento.id == repositorio.id) ? repositorio : elemento,
26         ).toList()
27       );
28       return true;
29     } catch (e) {
30       return false;
31     }
32   }
33
34   Future loadNextPage() async {
35
36     if( state.isLoading || state.isLastPage ) return;
37
38     state = state.copyWith( isLoading: true );
39
40     final repositorios = await repositoriosRepository
41       .getRepositoriosByPage(limiteEntidades: state.limiteEntidades, pagina: state.pagina);
42
43     if( repositorios.isEmpty ){
44       state = state.copyWith(
45         isLoading: false,
46         isLastPage: true,
47       );
48       return;
49     }
50
51     state = state.copyWith(
52       isLastPage: false,
53       isLoading: false,
54       pagina: state.pagina + 10,
55       repositorios: [...state.repositorios, ...repositorios]
56     );
57   }
58 }
59 }
```

Figura 17. Repositorio Notifier

Por último, necesitamos el State Notifier Provider, que es lo que será llamado en cada una de las pantallas donde se requiera la gestión de un estado.

```
1 final repositoriosProvider = StateNotifierProvider<RepositoriosNotifiaer, RepositoriosState>((ref) {
2
3   final repositoriosRepository = ref.watch(repositoriosRepositoryProvider);
4
5   return RepositoriosNotifiaer(repositoriosRepository: repositoriosRepository);
6 });
```

Figura 18. Repositorio State Notifier Provider

Como ya lo había citado antes la aplicación móvil consume una API Rest que es lo que fue diseñado en el backend y la forma en cómo se comunica con la base de datos, esta misma fue desarrollada con el framework Nest que está basado en Node.js.

Para este proyecto es tan importante el uso de arquitectura limpia que también fue adoptado por la parte del backend, ya que, al tratarse de un proyecto con cierto grado de robustez era imperativo implementar una forma de organizar y estructurar de manera adecuada, eficaz, y sin tantos cambios el proyecto.

Dentro de la creación del backend se tomaron en cuenta dos cosas importantes, la facilidad del consumo de la API por parte de quien lo requiriera y también la seguridad, sin estos dos principales es muy probable que el proyecto no tuviera la facilidad de comunicarse con las aplicaciones web, móvil o escritorio, ya que, al no tener una clara información brindada por parte del backend hacia el frontend era muy posible que la aplicación que consumiera esta API se enfrentara a diversos problemas que el backend debía resolver.

Los problemas más usuales que se suelen contener dentro del backend, son el tipo de respuesta al consumir la API, si es para subida de archivos que estatus son los que maneja, si es para consulta de datos que tipos de respuesta son los que puede brindar después de realizar las operaciones de éxito o fracaso, la edición de los mismos archivos o datos.

Dentro de la creación de la API como recién se cita también fue muy importante proveer de seguridad para impedir la filtración de datos o la modificación de esta sin consentimiento, esta misma seguridad fue implementada con JWT (Json Web Token) en donde se generan tokens dependiendo del rango del usuario de la aplicación, en particular son 4 usuarios, que van de menor a mayor rango como se citan a continuación:

Not user: es el rango más bajo dentro de la clasificación de seguridad y la única función posible que puede realizar, es consultar todos los datos de la base de datos.

user: es el rango siguiente, el cual tiene permitido, consultar datos mediante la API, puede crear un repositorio nuevo, y por último que, aunque está restringido, puede enviar una sugerencia de edición a un repositorio. Cuando se refiere a sugerencia de edición, quiere decir que el rango user puede editar las características del repositorio, sin embargo, estos al guardarse son generan cambios, por eso es la parte restrictiva, sin embargo, los cambios realizados se almacenan en un end point que eventualmente se envían a revisión para saber si su estatus es aprobado o no.

superuser: rango superior dentro del backend con permisos de creación de repositorios, modificación de repositorios, aprobación de estatus para su publicación, y consulta de repositorios.

admin: el rango más alto dentro de los 4 mencionados, este rango permite, crear, editar, consultar, aprobar, inhabilitar y eliminar, tanto repositorios, como usuarios de la aplicación.

El backend también por la parte de seguridad fue diseñado de tal manera que requiere a quien consume su end point obedecer las directrices de tipos, ya que de no ser así se manejan estatus de servicio desde el estatus 1 hasta el 500 dependiendo de la consulta hacia la API.

En la fig.6 se muestra como es definida la entidad para el backend.

```
1  @Entity('user')
2  export class User {
3
4      @PrimaryGeneratedColumn('uuid')
5      id: string;
6
7      @Column('text',{
8          unique: true
9      })
10     email: string;
11
12     @Column('text', {
13         select: false,
14     })
15     password: string;
16
17
18     @Column('text')
19     fullName: string;
20
21
22     @Column('bool', {
23         default: true
24     })
25     isActive: boolean;
26
27     @Column('text',{
28         array: true,
29         default: ['user']
30     })
31     roles: string[];
32
33     @Column('text',{
34         nullable: true
35     })
36     cedula?: string
37
38     @OneToMany(
39         () => Repositorio,
40         (repositorio) => repositorio.user,
41     )
42     repositorio: Repositorio;
43
44     @BeforeInsert()
45     checkFieldBeforeInsert(){
46         this.email = this.email.toLowerCase().trim();
47     }
48
49     @BeforeUpdate()
50     checkFieldBeforeUpdate(){
51         this.checkFieldBeforeInsert();
52     }
53 }
```

Figura 19. Entidad Usuario

Creación del Dto para el usuario.

```
1  export class CreateUserDto {
2
3      @ApiModelProperty({
4          description: 'email usuario',
5          nullable: false,
6      })
7      @IsString()
8      @IsEmail()
9      email: string;
10
11     @ApiModelProperty({
12         description: 'password user',
13         nullable: false,
14         minLength: 6,
15         maxLength: 50
16     })
17     @IsString()
18     @MinLength(6)
19     @MaxLength(50)
20     @Matches(
21         /(?:(?=.*\d)(?=.*\W+))(?![\.\n])(?=.*[A-Z])(?=.*[a-z]).*$/, {
22             message: 'The password must have a Uppercase, lowercase letter and a number'
23         })
24     password: string;
25
26     @ApiModelProperty({
27         description: 'fullName user',
28         nullable: false,
29         minLength: 3
30     })
31     @IsString()
32     @MinLength(3)
33     fullName: string;
34
35     @ApiModelProperty({
36         description: 'cedula user',
37         nullable: true,
38         default: null
39     })
40     @IsString()
41     @IsOptional()
42     cedula?: string;
43 }
```

Figura 20. Create User Dto

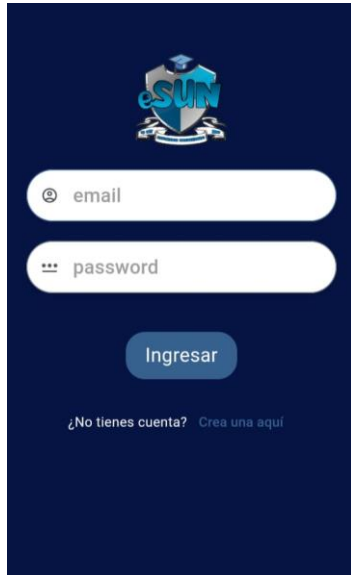
Y por último un poco de cómo se implementó el código para el CRUD.

```
1 import { Controller, Get, Post, Body, Patch, Param, Delete, ParseUUIDPipe, Query } from '@nestjs/common';
2 import { ApiResponse, ApiTags } from '@nestjs/swagger';
3
4 import { RepositorioService } from '../repositorio.service';
5 import { CreateRepositorioDto } from '../dto/create-repositorio.dto';
6 import { UpdateRepositorioDto } from '../dto/update-repositorio.dto';
7 import { PaginacionDto } from '../common/dtos/paginacion.dto';
8
9 import { Auth, GetUser } from '../auth/decorators';
10 import { User } from '../auth/entities/user.entity';
11 import { ValidRoles } from '../auth/interfaces';
12 import { Repositorio } from '../entities';
13
14 @ApiTags('Repositorio')
15 @Controller('repositorio')
16 export class RepositorioController {
17   constructor(private readonly repositorioService: RepositorioService) {}
18
19   @Post()
20   @Auth(ValidRoles.superUser, ValidRoles.admin, ValidRoles.user)
21   @ApiResponse({ status: 201, description: 'El producto a sido creado', type: Repositorio })
22   @ApiResponse({ status: 400, description: 'Solicitud erronea' })
23   @ApiResponse({ status: 403, description: 'Renovar Tokken' })
24   create(
25     @Body() createRepositorioDto: CreateRepositorioDto,
26     @GetUser() user: User
27   ) {
28     return this.repositorioService.create(createRepositorioDto, user);
29   }
30
31   @Get()
32   findAll(@Query() paginacionDto: PaginacionDto) {
33     // console.log(paginacionDto);
34     return this.repositorioService.findAll(paginacionDto);
35   }
36
37   @Get('/:id')
38   findOne(@Param('id') id: string) {
39     return this.repositorioService.findOnePlain(id);
40   }
41
42   @Patch('/:id')
43   @Auth(ValidRoles.admin, ValidRoles.superUser)
44   @ApiResponse({ status: 200, description: 'Repositorio actualizado', type: Repositorio })
45   @ApiResponse({ status: 400, description: 'Solicitud erronea' })
46   @ApiResponse({ status: 401, description: 'No autorizado' })
47   update(
48     @Param('id', ParseUUIDPipe) id: string,
49     @Body() updateRepositorioDto: UpdateRepositorioDto,
50     @GetUser() user: User
51   ) {
52     return this.repositorioService.update(id, updateRepositorioDto, user);
53   }
54
55   @Delete('/:id')
56   @Auth(ValidRoles.admin)
57   @ApiResponse({ status: 201, description: 'Repositorio a sido eliminado', type: Repositorio })
58   @ApiResponse({ status: 400, description: 'Solicitud erronea' })
59   @ApiResponse({ status: 401, description: 'No autorizado' })
60   remove(@Param('id', ParseUUIDPipe) id: string) {
61     return this.repositorioService.remove(id);
62   }
63 }
```

Figura 21. CRUD del repositorio

A continuación, se muestran capturas de diferentes pantallas de la aplicación móvil en funcionamiento.

Pantalla Login o Inicio de sesión



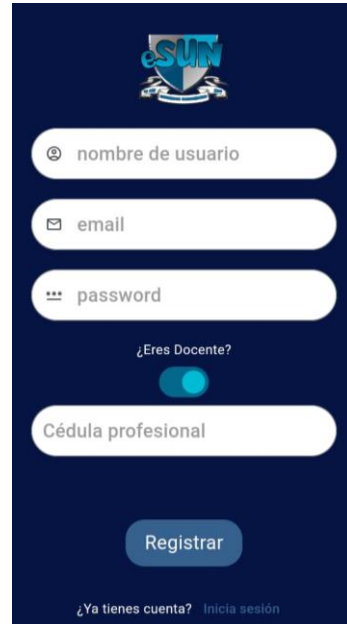
email

password

Ingresar

¿No tienes cuenta? Crea una aquí

Pantalla Registro con cedula



nombre de usuario

email

password

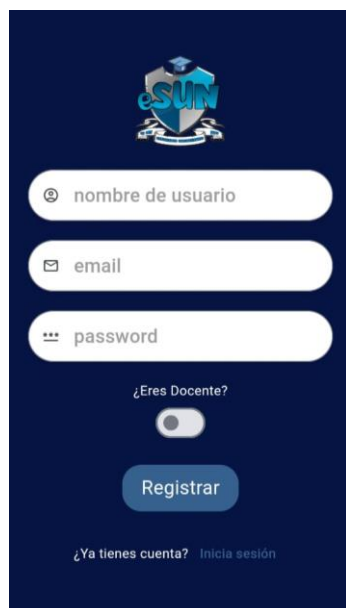
¿Eres Docente?

Cédula profesional

Registrar

¿Ya tienes cuenta? Inicia sesión

Pantalla Registro sin cedula



nombre de usuario

email

password

¿Eres Docente?

Registrar

¿Ya tienes cuenta? Inicia sesión

Pantalla validación de credenciales



davcogut@gmail.com

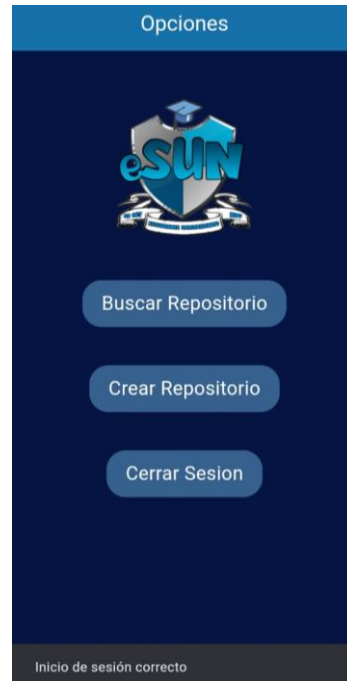
.....

¿No tienes cuenta? Crea una aquí

Pantallas credenciales incorrectas



Pantalla opciones



Pantalla validación formulario vacío



Pantalla Buscar Repositorio



Pantalla Ver / Editar Repositorio



Pantalla Crear Repositorio



Conclusiones

Hoy en día las aplicaciones móviles impactan de manera importante dentro de nuestro día a día haciéndonos las tareas un poco más fáciles, en particular esta herramienta logra su propósito al ser de utilidad hacia mis pares dentro de la carrera de Ingeniería en Ciencias de la Computación, ya que, al tener la información al alcance de un clic, el tiempo de consulta de información para ayudar en la tarea de un estudiante se ve enormemente beneficiada.

Bibliografía

- Calderón Loeza, G. Y., & Sánchez Escobedo, P. (8 de Enero de 2021). *Impacto del uso de dispositivos móviles en el aprendizaje de estudiantes adolescentes*. Obtenido de revistas.ujat.mx: <https://revistas.ujat.mx/index.php/emerging/article/view/4040>
- Calderón Loeza, G. Y., & Sánchez Escobedo, P. (7 de Enero de 2021). *Impacto del uso de dispositivos móviles en el aprendizaje de estudiantes adolescentes*. Obtenido de <https://revistas.ujat.mx/>: <https://revistas.ujat.mx/index.php/emerging/article/view/4040>
- CALDERÓN LOEZA, G. Y., & SÁNCHEZ ESCOBEDO, P. (7 de 1 de 2021). *Impacto del uso de dispositivos móviles en el aprendizaje de estudiantes adolescentes*. Obtenido de <https://revistaemerging.ujat.mx/>: <https://revistaemerging.ujat.mx/index.php/emerging/article/view/4040/3274>
- Cruz Barragán, A., & Barragán López, A. D. (1 de diciembre de 2014). *Aplicaciones Móviles para el proceso de Enseñanza-Aprendizaje en Enfermería*. Obtenido de revista.unsis.edu.mx: <https://revista.unsis.edu.mx/index.php/saludyadmon/article/view/81>
- CRUZ BARRAGÁN, A., & BARRAGÁN LÓPEZ, A. D. (1 de 12 de 2014). *Aplicaciones Móviles para el Proceso de Enseñanza-Aprendizaje en Enfermería*. Obtenido de <https://revista.unsis.edu.mx/>: https://www.unsis.edu.mx/revista/doc/vol1num3/A4_Aplic_Mov.pdf
- GAETA GONZÁLEZ, M., & CAVAZOS ARROYO, J. (12 de 2016). *Relación entre tiempo de estudio, autorregulación del aprendizaje y desempeño académico en estudiantes universitarios*. Obtenido de <https://www.scielo.org.mx/>: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-53082016000200142&lng=es&tlng=es
- Instituto Nacional de Estadística y Geografía. (19 de 6 de 2023). *ENCUESTA NACIONAL SOBRE DISPONIBILIDAD Y USO DE TECNOLOGÍAS DE LA INFORMACION EN LOS HOGARES (ENDUTIH) 2022*. Obtenido de <https://www.inegi.org.mx/>: https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2023/ENDUTIH/ENDUTIH_22.pdf
- Instituto Nacional de Estadística y Geografía. (19 de 6 de 2023). *La ENDUTIH registró 93.8 millones de personas usuarias de teléfono celular. La cifra equivale a 79.2 % de la población de estudio*. Obtenido de <https://www.inegi.org.mx/>: https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2023/ENDUTIH/ENDUTIH_22.pdf
- LAVÍN ZATARAIN, S., ZALDÍVAR COLADO, A., RODELO MORENO, J. A., & ZALDÍVAR MARTÍNEZ, J. J. (5 de 10 de 2019). *Utilización del smartphone por estudiantes de nivel superior*. Obtenido de <https://riti.es/>: <https://riti.es/index.php/riti/article/view/97/118>
- MARCÉN, M., & MARTÍNEZ CARABALLO, N. (3 de 2012). *Gestión eficiente del tiempo de los universitarios: evidencias para estudiantes de primer curso de la Universidad de Zaragoza*. Obtenido de <http://www.scielo.org.co/>: http://www.scielo.org.co/scielo.php?pid=S0121-50512012000100010&script=sci_arttext
- MOLINA RIOS, J. R., HONORES TAPIA, J. A., PEDREIRA SOUTO, N., & PARDO LEÓN, H. (14 de 6 de 2021). *Estado del arte: metodologías de desarrollo de aplicaciones móviles*. Obtenido de <https://ruc.udc.es/dspace/>:

- [https://ruc.udc.es/dspace/bitstream/handle/2183/28449/Pedreira-Souto_Nieves_2021_Metodolog%
c3%adas_aplicaciones_m%c3%b3viles.pdf?sequence=3&isAllowed=y](https://ruc.udc.es/dspace/bitstream/handle/2183/28449/Pedreira-Souto_Nieves_2021_Metodolog%c3%adas_aplicaciones_m%c3%b3viles.pdf?sequence=3&isAllowed=y)
- Organización de las Naciones Unidas. (26 de 12 de 2023). *Más del 75% de la población mundial tiene un teléfono celular y más del 65% usa el internet*. Obtenido de <https://news.un.org/es/>: <https://news.un.org/es/story/2023/12/1526712>
- Organización de las Naciones Unidas. (26 de 12 de 2023). *Más del 75% de la población mundial tiene un teléfono celular y más del 65% usa el internet*. Obtenido de <https://news.un.org/es/>: <https://news.un.org/es/story/2023/12/1526712>
- Organization de las Naciones Unidas para la Educación, la Ciencia y la Cultura. (2013). *Directrices para las políticas de aprendizaje móvil*. Obtenido de Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO): <https://unesdoc.unesco.org/ark:/48223/pf0000219662.locale=es>
- RAMÍREZ ARELLANO, A., RODRÍGUEZ MANCERA, J. A., & ACOSTA GONZAGA, E. (10 de 11 de 2014). *Guías útiles para la construcción de Objetos de*. Obtenido de <https://www.researchgate.net/>: https://www.researchgate.net/profile/Elizabeth-Gonzaga/publication/264871508_Guias_utiles_para_la_construccion_de_Objetos_de_Aprendizaje_para_dispositivos_Moviles_AOM/links/5460f4170cf295b561638237/Guias-utiles-para-la-construccion-de-Objetos-de-Aprendiza
- Ramírez Arellano, A., Rodríguez Mancera, J. A., & Acosta Gonzaga, E. (10 de Noviembre de 2014). *Guías útiles para la construcción de Objetos de Aprendizaje para dispositivos Móviles (AOM)*. Obtenido de [researchgate.net](https://www.researchgate.net/): https://www.researchgate.net/publication/264871508_Guias_utiles_para_la_construccion_de_Objetos_de_Aprendizaje_para_dispositivos_Moviles_AOM
- VACA ORELLANA, C. F., NARANJO TORO, M. E., BESANTES ANDRADE, A., & BENÍTEZ BASTIDAS, N. (30 de 6 de 2016). *Objeto de Aprendizaje Móvil en el Aula, para estudiantes de la Carrera de Nutrición y Salud Comunitaria, Universidad Técnica Del Norte*. Obtenido de <http://201.159.222.149/index.php>: <http://201.159.222.149/index.php/lauinvestiga/article/view/229>
- VERGARA MESA, M. I. (2011). *FUNCIONES EJECUTIVAS Y DESEMPEÑO ACADEMICO EN ESTUDIANTES DE PRIMER AÑO DE PSICOLOGÍA DE LA CORPORACION UNIVERSITARIA MINUTO DE DIOS, EN BELLO ANTIOQUIA*. Obtenido de <https://bibliotecadigital.usb.edu.co/home>: <https://bibliotecadigital.usb.edu.co/server/api/core/bitstreams/827ccc64-6951-4a43-b498-eb78e9744a59/content>