****

**DESARROLLO DE APP EDUCATIVA 4.0**

**MEMORIA DE ESTADÍA PROFESIONAL**

R E P O R T E T É C N I C O

**PARA OBTENER EL TÍTULO DE**

**TÉCNICO SUPERIOR UNIVERSITARIO EN TECNOLOGÍAS DE LA INFORMACIÓN ÁREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA**

*P R E S E N T A*

**JOEL GONZÁLEZ CRUZ**

ASESORA DE LA ORGANIZACIÓN: DRA. MORAMAY RAMÍREZ HERNÁNDEZ   
ASESORA ACADÉMICA: MTRA. EN C. LIZETH AGUILAR CARRILLO  
 ORGANIZACIÓN: “UNIVERSIDAD TECNOLÓGICA DE TECÁMAC”

GENERACIÓN: ENERO 2023 – DICIEMBRE 2024

CUATRIMESTRE DE TÉRMINO: SEPTIEMBRE - DICIEMBRE 2024

CARTA DE AUTORIZACION

CARTA DE CESIÓN

AGRADECIMIENTO Y DEDICATORIAS

Quiero agradecer a la Universidad Tecnológica de Tecámac por brindarme la oportunidad de aplicar de forma profesional los conocimientos y herramientas brindados por la misma, así mismo también quiero agradecer a la Dra. Moramay Ramírez Hernández por brindarme un espacio y proyecto en mi estadía profesional así como actuar como guía en mi formación académica y formal, también quiero agradecer a mi familia por otorgarme la ayuda económica y recursos necesarios para mi formación académica, en especial a Ana María Cruz García y Simón González Najar.

# ÍNDICE

[RESUMEN 1](#_Toc183510488)

[ABSTRACT 2](#_Toc183510489)

[INTRODUCCIÓN 3](#_Toc183510490)

[OBJETIVOS 5](#_Toc183510491)

[MARCO TEÓRICO 10](#_Toc183510494)

[METODOLOGÍA 15](#_Toc183510495)

[CAPÍTULO 1. ANÁLISIS 17](#_Toc183510496)

[1.1 Necesidades del cliente 17](#_Toc183510497)

[1.2 Identificación y definición de roles de usuario 17](#_Toc183510498)

[1.3 Diagramas generales 18](#_Toc183510499)

[1.4 Especificación de casos de uso 22](#_Toc183510500)

[1.5 Diagramas de secuencia 27](#_Toc183510501)

[CAPÍTULO 2. INICIALIZACIÓN 31](#_Toc183510502)

[2.1 Diagrama relacional de base de datos 31](#_Toc183510503)

[2.2 Diccionario de datos 31](#_Toc183510504)

[2.3 Diseño de interfaces 36](#_Toc183510505)

[CAPÍTULO 3. PRODUCCIÓN 46](#_Toc183510506)

[CONCLUSIONES 60](#_Toc183510507)

[LISTADO DE SIGLAS O ACRÓNIMOS 62](#_Toc183510508)

[GLOSARIO 64](#_Toc183510509)

[ANEXOS 68](#_Toc183510510)

[REFERENCIAS 88](#_Toc183510511)

# RESUMEN

El presente documento tiene como finalidad mostrar el desarrollo del prototipo de aplicación educativa “Desarrollo App Educativa 4.0”, este documento desglosa sus partes tales como: objetivos, alcances, herramientas empleadas para su desarrollo, metodología, especificación de requerimientos, diseño, desarrollo e implementación del proyecto.

El prototipo “Desarrollo App Educativa 4.0” está destinado a ayudar a los estudiantes de la Universidad Tecnológica de Tecámac a reforzar sus habilidades cognitivas a través de la asignación de temarios, los cuales tendrán un examen diagnóstico y contenidos variados los cuales a su vez poseen material de apoyo y un ejercicio práctico que ayude a comprender el contenido de cada uno de los temas asignados.

El contenido de la App es gestionado por uno o varios usuarios administradores, los cuales podrán editar el temario, los exámenes diagnósticos, actividades, contenidos, material de apoyo y otra información relacionada al contenido didáctico dentro de la App.

Los estudiantes podrán registrarse y acceder a la App para realizar los temarios registrados, al completar algún examen diagnóstico se les asignará contenido de acuerdo a su resultado obtenido y la configuración del examen. Las preguntas de examen y de actividades se generan al momento de que un estudiante los realice, haciendo que las preguntas aparezcan en diferente orden y algunas preguntas sean reemplazadas por otras preguntas.

El propósito del prototipo “Desarrollo App Educativa 4.0” es disminuir el rezago escolar y deficiencias de conocimiento de alumnos de nuevo ingreso en la división de TIC, otorgando material de apoyo que facilite la comprensión de temas de soporte al plan de estudios de la Universidad Tecnológica de Tecámac.

# ABSTRACT

The porpuse of this document is to show the development of the prototype “Desarrollo App Educativa 4.0”, this document explains all of its parts, such as: the objectives, scope, tools used for the development, methodology, specification of requirements, design, development of the project and the implementation.

The prototype “Desarrollo App Educativa 4.0” is intended to help the Technologycal University of Tecámac’s Students to improve their cognitive skills through assigment of topics, these will have a test and a variety of contents which in turn contains supporting material and a excercise that it helps to understand the content of each topic.

The App’s content is manage by one or more administrator users, they can edit the topics, tests, activities, contents, material support, and other information related to the educational content in the App. The students can sign up and log in to the App to do the registred topics, upon completing a diagnostic test, content will be assigned based on the results obtained and the test configuration. The test’s questions and activity’s questions are generated at the time that the student does, the questions will be shuffled and some of them by other questions.

The porpuse of the prototype “Desarrollo App Educativa 4.0” is reduce school backlog and Knowledge deficiencies in the new Students of the Technologycal University of Tecámac in the TIC’s division, giving supporting material that facilitate understanding the topic from the studying plan of the Technologycal University of Tecámac.

# INTRODUCCIÓN

La Universidad Tecnológica de Tecámac es un organismo público descentralizado del Gobierno del Estado de México, institución educativa creada en el año de 1996, con el objetivo de formar técnicos superiores y profesionales universitarios, aptos para el desarrollo de conocimientos aplicados a la solución creativa de los problemas del sector productivo y a los requerimientos del crecimiento económico y social que requiere los Estados Unidos Mexicanos.

En esta Institución se imparten 11 programas educativos de Técnico Superior Universitario, 9 de nivel Licenciatura y una Maestría; se atiende cerca de 6,000 estudiantes, de los cuales, el 65% cursa estudios de Técnico Superior Universitario y el 35% restante estudios de nivel Licenciatura, en programas educativos reconocidos por su buena calidad. En sus diferentes aulas, talleres y laboratorios se realizan actividades científicas y se extiende el conocimiento y los servicios hacia la sociedad.

Es práctica constante de esta universidad actualizar sus planes y programas de estudio de acuerdo con las necesidades del sector productivo; renovar su equipamiento y mejorar sus instalaciones. Sus profesores se capacitan y actualizan constantemente para ofrecer a los estudiantes mejores experiencias educativas y conocimientos que contribuyan a su formación integral.

El Cuerpo Académico de Servicios Tecnológicos detectó que algunos estudiantes de nuevo ingreso inscritos en las carreras de la División de Tecnologías de la Información y Comunicación (*DTIC*), presentan dificultades en algunas materias técnicas, lo que reduce su desempeño y entorpece las habilidades de los estudiantes en entornos profesionales, por lo que en el presente documento se presenta el prototipo denominado “Desarrollo App Educativa 4.0”, el cual tiene como propósito, actuar como un apoyo didáctico para dichos alumnos en temas básicos como: pensamiento abstracto, lógica, computo, etc. Que facilite la compresión de temas de las asignaturas de los planes de estudio de la *DTIC*.

El presente documento se divide en tres capítulos, los cuales abarcaran el desarrollo del proyecto “Desarrollo App Educativa 4.0”.

El capítulo uno está dirigido a la planeación general del proyecto, esto incluye la recopilación de requerimientos, tales como las necesidades del cliente, especificación de roles de usuario, diagramas generales y de secuencia. Los cuales están dirigidos a definir la estructura general del proyecto “Desarrollo App Educativa 4.0” para poder abordarlo correctamente.

El capítulo dos abarcará el diseño de la aplicación, el cual incluye el diseño de la base de datos que se empleará en el proyecto “Desarrollo App Educativa 4.0”, con un diagrama relacional y un diccionario de datos. Además también contendrá el diseño de interfaces o vistas de la aplicación.

Finalmente en el capítulo tres se centra en la implementación o codificación de la aplicación, en ella se detallará la estura y comportamiento general de algunas partes del prototipo, con la finalidad de explicar la funcionalidad de dichas partes mencionadas durante este capítulo.

# OBJETIVOS

## Objetivo General

Desarrollar una App interactiva en la cual, personal administrativo pueda gestionar temas, actividades y exámenes diagnósticos que permitan reforzar las habilidades cognitivas de los estudiantes de DTIC de la Universidad Tecnológica de Tecámac a fin de disminuir el rezago escolar y las deficiencias de conocimiento en dichos estudiantes.

## Objetivos específicos

Desarrollar un sistema de registro y autenticación para acceder a exámenes diagnósticos y actividades relacionadas con los temas disponibles, garantizando la seguridad y privacidad de la información.

Desarrollar un módulo administrativo que permita a los administradores gestionar información sobre asignaturas, así como el contenido de los exámenes diagnósticos y actividades, asegurando la actualización constante del material de apoyo.

Optimizar la experiencia de aprendizaje de los estudiantes a través de la integración de recursos interactivos y exámenes diagnósticos que faciliten la identificación de deficiencias de conocimiento.

|  |
| --- |
| **UNIVERSIDAD TECNOLÓGICA DE TECÁMAC** |
| ***DIVISIÓN TIC*** |
| ***PROGRAMA DE ESTADÍAS PROFESIONALES*** |

|  |  |  |
| --- | --- | --- |
| ***PROGRAMA DE TRABAJO*** |  | |
|  | | *FECHA: 02/09/2024* |

*DATOS DE LA ESTUDIANTE*

|  |  |
| --- | --- |
| NOMBRE: | Joel González Cruz |
| DIVISIÓN: | Tecnologías de la Información y Comunicación |
| CARRERA: | Técnico Superior Universitario en Tecnologías de la Información Área Desarrollo de Software Multiplataforma |
| MATRÍCULA: | 2523260021 |
| GENERACIÓN: | enero 2023 – diciembre 2024 |

*ASESORA ACADÉMICA*

|  |  |
| --- | --- |
| NOMBRE: | Lizeth Aguilar Carrillo |
| CARGO: | Profesora de Tiempo Completo Asociado “C” |

*DATOS DE LA ORGANIZACIÓN*

|  |  |
| --- | --- |
| NOMBRE DE LA ORGANIZACIÓN: | Universidad Tecnológica de Tecámac |
| DEPARTAMENTO: | Tecnologías de la Información y Comunicación |
| ÁREA: | Desarrollo de Software Multiplataforma |
| DIRECCIÓN: | Carretera Federal México – Pachuca km 37.5, 55749 Estado de México |
| TELÉFONO: | 55 6499 7632 |
| E-MAIL: | [ditc@uttecamac.edu.mx](mailto:ditc@uttecamac.edu.mx) |

*ASESOR DE LA ORGANIZACIÓN*

|  |  |
| --- | --- |
| NOMBRE: | Moramay Ramírez Hernández |
| CARGO: | Profesora De Tiempo Completo Asociado “C” |

*PERÍODO*

|  |  |
| --- | --- |
| DURACIÓN: | 15 semanas |
| FECHA DE INICIO: | 02 de septiembre de 2024 |
| FECHA DETERMINACIÓN: | 06 de diciembre de 2024 |
| HORARIO: | 09:00 a.m. – 16:00 p.m. |

*PROYECTO*

|  |  |
| --- | --- |
| NOMBRE: | Desarrollo App Educativa 4.0 |
| DESCRIPCIÓN: | Aplicación móvil que permita reforzar las habilidades cognitivas de los estudiantes |
| OBJETIVO GENERAL: | Desarrollar una App interactiva en la cual, personal administrativo pueda gestionar temas, actividades y exámenes diagnósticos que permitan reforzar las habilidades cognitivas de los estudiantes de DTIC de la Universidad Tecnológica de Tecámac a fin de disminuir el rezago escolar y las deficiencias de conocimiento en dichos estudiantes. |
| OBJETIVOS ESPECÍFICOS: | Desarrollar un sistema de registro y autenticación para acceder a exámenes diagnósticos y actividades relacionadas con los temas disponibles, garantizando la seguridad y privacidad de la información.  Desarrollar un módulo administrativo que permita a los administradores gestionar información sobre asignaturas, así como el contenido de los exámenes diagnósticos y actividades, asegurando la actualización constante del material de apoyo.  Optimizar la experiencia de aprendizaje de los estudiantes a través de la integración de recursos interactivos y exámenes diagnósticos que faciliten la identificación de deficiencias de conocimiento. |
| ALCANCE(S): | El prototipo “Desarrollo App Educativa 4.0” abarcara las 3 primeras fases de la metodología Mobile-D (Exploración, Inicialización y Producción). Los estudiantes podrán registrarse a la aplicación, iniciar sesión, realizar exámenes diagnósticos y actividades de los temas que estén disponibles en la aplicación. Por otro lado, los administradores podrán iniciar sesión, agregar y editar información de los temas, el contenido de exámenes diagnósticos y de las actividades que se vayan a estar disponibles dentro de la aplicación. |
| META(S): | Disminuir el rezago escolar y deficiencias de conocimiento de alumnos de nuevo ingreso en la división de TIC, otorgando material de apoyo que facilite la comprensión de temas de soporte al plan de estudios de la Universidad Tecnológica de Tecámac en un lapso de 15 semanas. |
| RECURSOS: | Hardware: Equipo de cómputo con 4 núcleos, 8GB de memoria RAM, 50GB de almacenamiento, puertos y memorias USB de 16GB y con conexión a internet.  Software: Android Studio, Java y SQLite. |

*PLAN DE TRABAJO*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ACTIVIDAD** | | **DESCRIPCIÓN** | **SEMANA** | | **FECHAS** | |
| **INICIO** | **TÉRMINO** | **INICIO** | **TÉRMINO** |
| 1 |  | Exploración | 1 | 5 | 02/09/24 | 01/10/24 |
|  | 1.1 | Identificar las necesidades del cliente | 1 | 2 | 02/09/24 | 09/09/24 |
|  | 1.2 | Definir roles de usuario | 2 | 2 | 09/09/24 | 12/09/24 |
|  | 1.3 | Diseñar diagramas generales | 2 | 3 | 12/09/24 | 17/09/24 |
|  | 1.4 | Especificar casos de uso | 3 | 4 | 17/09/24 | 24/09/24 |
|  | 1.5 | Diseñar diagramas de secuencia | 4 | 5 | 24/09/24 | 01/10/24 |
| 2 |  | Inicialización | 5 | 8 | 01/10/24 | 23/10/24 |
|  | 2.1 | Diseñar diagrama de base de datos (relacional) | 5 | 6 | 01/10/24 | 08/10/24 |
|  | 2.2 | Definir diccionario de datos | 6 | 6 | 08/10/24 | 11/10/24 |
|  | 2.3 | Diseñar interfaces | 6 | 8 | 11/10/24 | 23/10/24 |
| 3 |  | Producción | 8 | 14 | 23/11/24 | 04/12/24 |

CRONOGRAMA DE ACTIVIDADES (PROGRAMA)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | ACTIVIDADES | CONTROL | septiembre | | | | | octubre | | | | noviembre | | | | diciembre | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | Exploración | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Inicialización | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Producción | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*FIRMAS*

|  |  |
| --- | --- |
| Dra. Moramay Ramírez Hernández  *ASESORA INSTITUCIONAL* | |
| Joel González Cruz  *ESTUDIANTE* | Mtra en C. Lizeth Aguilar Carrillo  *ASESORA ACADÉMICA* |

# MARCO TEÓRICO

**Android Studio**

Android Studio es el entorno de desarrollo integrado (*IDE*) oficial que se usa en el desarrollo de *Apps* para Android. Basado en el potente editor de código y las herramientas para desarrolladores de *IntelliJ IDEA*, Android Studio ofrece funciones como: sistema de compilación flexible basado en *Gradle*, emuladores virtuales de dispositivos móviles, integración con *Git* y *GitHub*, variedad de marcos de trabajo y herramientas de prueba, entre otras funciones.

Entre las funcionalidades más destacadas de Android Studio se encuentran: Sistema de compilación flexible, emuladores virtuales de dispositivos móviles, variedad de marcos de trabajo, herramientas de *testing*, diseño de interfaces atractiva, refactorización y análisis de código, además de contar soporte para varios idiomas.

En el contexto del proyecto “Desarrollo App Educativa 4.0”, se utilizará Android Studio como la plataforma principal para desarrollar, compilar y depurar la aplicación. Esto permitirá aprovechar al máximo las características y herramientas que ofrece el *IDE*, asegurando un desarrollo ágil y eficiente, y facilitando la implementación de funcionalidades educativas innovadoras. Gracias a su robustez y versatilidad, Android Studio se convierte en la elección ideal para llevar a cabo este proyecto educativo, garantizando una experiencia de usuario óptima y un producto final de alta calidad.

**Java**

Es un lenguaje de programación orientado a objetos y una plataforma de *software* ampliamente utilizado que se ejecuta en miles de millones de dispositivos, que incluyen computadoras portátiles, dispositivos móviles, consolas de videojuegos, dispositivos médicos y muchos otros. Una de las principales ventajas de desarrollar *software* con Java es su portabilidad. Una vez que haya escrito el código para un programa Java en una computadora portátil, es muy fácil mover el código a un dispositivo móvil. El objetivo principal es "escribir una vez, ejecutar en cualquier lugar".

Algunas de las características principales de Java son: Paradigma orientado a objetos, seguridad de datos, gran cantidad de bibliotecas y *API,* procesamiento multi-hilo y una comunidad activa.

Se empleará este lenguaje de programación para escribir el código fuente de la aplicación. Su naturaleza orientada a objetos y su portabilidad permitirá construir una aplicación robusta y adaptable que funcione en una variedad de dispositivos Android.

**XML**

Es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos. Permite definir y almacenar datos de forma compartible. También admite el intercambio de información entre sistemas de computación, como sitios *web*, bases de datos y aplicaciones de terceros. Las reglas predefinidas facilitan la transmisión de datos como archivos *XML* a través de cualquier red, ya que el destinatario puede usar esas reglas para leer los datos de forma precisa y eficiente.

Algunas características de *XML* son: Estructura jerárquica, legibilidad, compatibilidad con diferentes sistemas, validación, flexibilidad y extensibilidad.

Se empleará este lenguaje de marcado para definir la estructura de las vistas o pantallas de la aplicación. Al utilizar *XML* para diseñar estas interfaces, se podrá separar claramente la lógica de la aplicación de su presentación, lo que mejora la mantenibilidad y la escalabilidad del código.

**SQLite**

Es un motor de bases de datos relacional compatible con *ACID*, contenida en una pequeña biblioteca escrita en C. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos es guardado como un solo fichero estándar en la máquina host o principal.

Las características más destacables de SQLite son: Integracion sencilla, rendimiento optimizado, ligereza y portabilidad y soporte para consultas *SQL*.

Se utilizará como motor de base de datos para almacenar, consultar, actualizar y, en caso de ser necesario, eliminar la información necesaria para la ejecución de la aplicación “Desarrollo App Educativa 4.0”

**Git**

Es un sistema de control de versiones distribuido en donde cada desarrollador, programador o colaborador tiene una copia integral del mismo. A diferencia de los sistemas de control de versiones centralizados, los *DVCS* no necesitan una conexión constante al repositorio central. Este sistema de control de versiones distribuido permite participar en un flujo colaborativo, manipular el historial de cambios, además de la creación y manipulación de ramas o trabajos temporales.

Las características de Git son: Distribución, historial de cambios, colaboración fluida y ramas o trabajos temporales.

Se hará uso de Git para llevar un **control consistente de los cambios** realizados en el código. Esta herramienta será fundamental por varias razones: mantenimiento de la calidad del código, seguridad y documentación.

**Blender**

Es un programa informático *open-source* multiplataforma que se ha consolidado como una de las herramientas más potentes y versátiles para el modelado, la animación y la creación de gráficos tridimensionales. Desde su inicio, Blender ha evolucionado para incluir una amplia gama de funciones que abarcan no solo la creación de modelos 3D, sino también iluminación, renderizado, animación, edición de vídeo, escultura y pintura digital. Su naturaleza de código abierto permite a una comunidad activa de desarrolladores y artistas contribuir a su mejora continua, lo que resulta en un software en constante evolución y altamente adaptable.

Las características clave de Blender son: Modelado 3D avanzado, animación, iluminación y renderizado, edición de video, pintura, escultura, texturización, amplia cantidad de recursos y una comunidad activa.

Se usará este programa para diseñar modelos tridimensionales que sean necesarios como recursos ilustrativos de apoyo en algunas partes de la aplicación “Desarrollo App Educativa 4.0”

**StarUML**

Es una herramienta de modelado de *software* de código abierto, basado en *UML*. Esta herramienta es ampliamente reconocida por su capacidad para soportar todos los tipos de diagramas *UML*, lo que la convierte en una opción ideal para la planificación y diseño de sistemas de software. Además de su funcionalidad de modelado, StarUML permite generar código y documentación directamente a partir de los modelos, lo que optimiza el flujo de trabajo y mejora la calidad del producto final.

Alguna de las características de StarUML son: Documentación automática, extensibilidad, colaboración en equipo e interfaz intuitiva.

Se utilizará este programa, para modelar diagramas de caso de uso, de secuencia, de paquetes, entre otros, para sustentar el desarrollo de la aplicación “Desarrollo App Educativa 4.0”.

**Canva**

Es una *web* de diseño gráfico y composición de imágenes para la comunicación fundada en 2012, y que ofrece herramientas online para crear tus propios diseños, tanto si son para ocio como si son profesionales. Su método es el de ofrecer un servicio *freemium*, que puedes utilizar de forma gratuita, pero con la alternativa de pagar para obtener opciones avanzadas. Sirve tanto para diseñadores aficionados como para los más experimentados, incluyendo su propio banco de imágenes y una serie de herramientas variadas.

Las características de Canva son: interfaz intuitiva, amplia biblioteca de plantillas, recursos gráficos, herramientas de edición, colaboración en tiempo real y exportación sencilla.

Se utilizará esta herramienta para diseñar algunos recursos gráficos que sean necesarios emplear en la aplicación.

# METODOLOGÍA

Para el desarrollo de este proyecto se eligió la metodología Mobile-D debido a ser una metodología ágil, ideal para equipos pequeños y enfoca en el desarrollo de aplicaciones móviles.

El objetivo de esta metodología es conseguir ciclos de desarrollo muy rápidos. Tiene distintas fases: exploración, inicialización, fase de producto, fase de estabilización y la fase de pruebas. Cada una tiene un día de planificación y otro de entrega.

Fase exploración: se centra la atención en la planificación y en los conceptos básicos del proyecto. Aquí es donde se define el alcance del proyecto y su establecimiento con las funcionalidades donde se quiere llegar.

Fase de iniciación: configurar el proyecto identificando y preparando todos los recursos necesarios. En esta fase se acuerda un día a la planificación y el resto al trabajo y publicación.

Fase de producto: se repiten iterativamente las subfases. Se usa el desarrollo dirigido por pruebas (*TDD*), antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento. En esta fase se puede decir que se lleva a acabo toda la implementación.

Fase de estabilización: se realizan las acciones de integración para enganchar los posibles módulos separados en una única aplicación.

Fase de pruebas: una vez parado totalmente el desarrollo se pasa una fase de testeo hasta llegar a una versión estable según lo establecido en las primeras fases por el cliente. Si es necesario se reparan los errores, pero no se desarrolla nada nuevo.

En el presente documento únicamente se abarcará hasta la tercera fase “Producción” debido a que en un lapso de quince semanas o cuatro meses, no es posible abarcar correctamente todas las fases de la metodología *Mobile-D*, además de que se prefirió disponer de un ligero margen en los tiempos de cambio de fase para asegurar la correcta integración de cada una de las mismas.

# CAPÍTULO 1. ANÁLISIS

El primer capítulo contiene las necesidades del cliente, el análisis del problema, la forma de abordarlo, así como definición de roles de usuario, casos de uso, especificaciones del proyecto.

## Necesidades del cliente

El Cuerpo Académico de Servicios Tecnológicos ha identificado de manera recurrente deficiencias en las materias técnicas de algunos estudiantes que ingresan a la División de Tecnologías de la Información y Comunicación (*DTIC*), especialmente en el primer cuatrimestre. Esta situación ha dado lugar a una alta tasa de deserción escolar y una elevada cantidad de asignaturas no acreditadas. Ante este panorama, se propuso el desarrollo de un prototipo de aplicación educativa que permita a los alumnos reforzar sus habilidades cognitivas y mejorar su desempeño académico, contribuyendo a reducir las tasas de reprobación y deserción.

## Identificación y definición de roles de usuario

Dentro del prototipo “Desarrollo App Educativa 4.0” se han definido dos roles principales de usuario: Estudiante y Administrador, cada uno con funcionalidades específicas orientadas a optimizar el uso de la aplicación y cumplir con los objetivos educativos planteados.

**Rol de Estudiante**

Este rol está diseñado para cualquier alumno inscrito en los programas educativos de la División de Tecnologías de la Información y Comunicación (DTIC) de la Universidad Tecnológica de Tecámac. Los estudiantes tendrán acceso a diversas funcionalidades y recursos educativos que incluyen:

* **Exámenes diagnósticos**: Evaluaciones iniciales para identificar sus conocimientos previos y áreas de mejora.
* **Actividades interactivas**: Material diseñado para fortalecer habilidades específicas en las asignaturas de los programas.
* **Progreso personalizado**: Visualización de avances y resultados en actividades, promoviendo la autogestión del aprendizaje.

Este rol busca ser una herramienta complementaria para reforzar el aprendizaje, disminuir el rezago académico y facilitar la transición de los estudiantes al entorno académico universitario.

Rol de Administrador

Este rol está enfocado en el personal encargado de la gestión y administración de la aplicación, garantizando su funcionamiento efectivo y alineado con las necesidades de los estudiantes y docentes. Las principales funciones del administrador incluyen:

* **Gestión de contenidos**: Creación, modificación y actualización de exámenes, actividades y recursos educativos dentro de la plataforma.
* **Supervisión del progreso estudiantil**: Monitoreo de los resultados y avances de los estudiantes para identificar áreas que requieran ajustes o mejoras.
* **Ajustes y mantenimiento**: Implementación de configuraciones y resolución de problemas técnicos para asegurar una experiencia fluida y eficiente en el uso de la *App*.

Este rol es clave para garantizar que la aplicación se mantenga actualizada y cumpla con los objetivos pedagógicos establecidos.

La definición clara de estos roles permite establecer un entorno colaborativo, funcional y adaptado a las necesidades de los usuarios, fomentando un aprendizaje efectivo y una administración eficiente.

## 1.3 Diagramas generales

La figura 1.1 representa el diagrama de paquetes general de la aplicación “Desarrollo App Educativa 4.0”. Este diagrama proporciona una visión general de la estructura de la aplicación, mostrando la organización de los diferentes módulos y componentes que interactúan entre sí para ofrecer una experiencia de aprendizaje completa.



**Figura 1.1** Diagrama de paquetes general

La figura 1.2 muestra la especificación del paquete “Inicio”, cuyo objetivo es permitir a los usuarios acceder a la aplicación y gestionar la información asociada a su cuenta. A través de este módulo, los usuarios pueden modificar sus datos personales, actualizar credenciales y gestionar preferencias, asegurando un acceso personalizado y eficiente a la plataforma.



**Figura 1.2** Diagrama de paquete Inicio

La figura 1.3 muestra la especificación del paquete “Temas”, cuyo objetivo es proporcionar un sistema *CRUD* (Crear, Leer, Actualizar, Eliminar) para los temas de aprendizaje dentro de la aplicación. Este paquete permite a los administradores gestionar de manera eficiente los contenidos educativos, creando nuevos temas, consultando los existentes, actualizando información relevante y eliminando aquellos que ya no son necesarios. De esta manera, se asegura que los estudiantes siempre tengan acceso a contenido actualizado y pertinente, facilitando el proceso de aprendizaje dentro de la plataforma.



**Figura 1.3** Diagrama de paquete Temas

La figura 1.4 muestra la especificación del paquete “Test”, cuyo objetivo es proporcionar un sistema *CRUD* (Crear, Leer, Actualizar, Eliminar) para la gestión de exámenes diagnósticos dentro de la aplicación. Este paquete permite a los administradores crear nuevos exámenes, consultar los existentes, actualizar su contenido según sea necesario y eliminar aquellos exámenes que ya no sean relevantes. Además, facilita la implementación y personalización de pruebas diagnósticas para evaluar el conocimiento y las habilidades de los estudiantes al inicio de su formación, contribuyendo así a una mejor identificación de áreas de oportunidad para su aprendizaje. Gracias a esta funcionalidad, se asegura una gestión eficiente y flexible de los exámenes diagnósticos, lo que optimiza la experiencia educativa en la plataforma.

El paquete también incluye opciones de configuración que permiten personalizar cada examen según los requisitos del curso o materia. Esto incluye la capacidad de establecer la cantidad de preguntas, resultados de aplicación y otros parámetros relevantes. Esta flexibilidad asegura que los exámenes diagnósticos se adapten a las necesidades específicas de los estudiantes y de los programas educativos.

Asimismo, el paquete "Test" facilita el seguimiento del desempeño de los estudiantes a lo largo del tiempo. Al permitir la creación de exámenes periódicos o de evaluación continua, los administradores pueden monitorear el progreso de los estudiantes, identificar áreas de mejora y ajustar los planes de estudio o contenidos en función de los resultados obtenidos. Esto contribuye a ofrecer una experiencia educativa más personalizada y efectiva.



**Figura 1.4** Diagrama de paquete Test

La figura 1.5 muestra la especificación del paquete “Actividades”, cuyo propósito es ofrecer un sistema *CRUD* (Crear, Leer, Actualizar, Eliminar) para la gestión de actividades asociadas a cada tema de aprendizaje dentro de la aplicación. Este paquete permite a los administradores crear nuevas actividades, asignarlas a los temas correspondientes, y establecer parámetros como la dificultad de cada una. Además, ofrece la capacidad de actualizar las actividades para adaptarlas a nuevas necesidades o cambios en el contenido educativo, así como eliminar aquellas que ya no sean relevantes o necesarias.

A través de esta funcionalidad, los estudiantes pueden acceder a las actividades de manera organizada, realizando tareas específicas relacionadas con los temas que están estudiando. Esto fomenta un aprendizaje activo y refuerza los conocimientos adquiridos. Por otro lado, el sistema permite a los administradores revisar el progreso de los estudiantes en cada actividad, lo que facilita el seguimiento y la evaluación del desempeño de los mismos. Esta estructura flexible y eficiente asegura que el contenido educativo sea continuamente actualizado y accesible para todos los usuarios de la plataforma.



**Figura 1.5** Diagrama de paquete Actividades

## 1.4 Especificación de casos de uso

El prototipo “Desarrollo App Educativa 4.0” cuenta con 21 casos de uso, de los cuales solo se mostrarán 5 de los más importantes. El resto de especificaciones de casos de uso se encuentra en la sección de documentos anexos

La figura 1.6 muestra la especificación del caso de uso “Registrar cuenta”. El estudiante debe de ingresar a la aplicación y en una vista colocar sus datos personales como matricula, nombre, apellido paterno materno, y establecer una contraseña de acceso.

|  |  |  |
| --- | --- | --- |
| CU-0001 | Registrar cuenta | |
| Versión | 1.1 (04/10/2024) | |
| Prioridad | Alta | |
| Autor/es | Joel González Cruz | |
| Actor/es | Estudiante | |
| Dependencias | Ninguna | |
| Descripción | El sistema deberá registrar en la base de datos la información ingresada por el estudiante para su posterior acceso | |
| Precondición | El estudiante deberá ingresar a la aplicación | |
| Flujo normal | Paso | Acción |
| 1 | El usuario hará click en el botón “Registrarse” |
| 2 | El sistema mostrará un formulario de registro con matrícula, nombre completo, contraseña y otros datos personales |
| 3 | El usuario ingresará el número de su matrícula |
| 4 | El usuario completará los campos del formulario con su información personal. |
| 5 | El usuario hará click en el botón “Registrarme” |
| 6 | El sistema validará la informacion ingresada |
| 7 | El sistema guardará la informacion ingresada en la base de datos |
| 8 | El sistema notificará al usuario del registro exitoso |
| Flujo alternativo | Paso | Acción |
| 6.1 | En caso de que algún campo tenga información incorrecta o incompatible, el sistema mostrará un mensaje notificando el error |
| 6.2 | En caso de que los datos ingresados ya existan en la base de datos, el sistema mostrará un mensaje notificando que el registro ya es existente |
| Postcondición | La base de datos almacenará los datos del estudiante | |

**Figura 1.6** Especificación del caso de uso “Registrar cuenta”

La figura 1.7 muestra la especificación del caso de uso “Iniciar sesión”. Cuando los administradores o estudiantes abran la *App* deben de colocar su matrícula y contraseña para que puedan accesar y realizar sus actividades.

|  |  |  |
| --- | --- | --- |
| CU-0002 | Iniciar sesión | |
| Versión | 1.1 (04/10/2024) | |
| Prioridad | Alta | |
| Autor/es | Joel González Cruz | |
| Actor/es | Estudiante, Administrador | |
| Dependencias | Ninguna | |
| Descripción | El sistema consultará en la base de datos la información asociada al usuario a partir de su matrícula y su contraseña | |
| Precondición | El usuario debe haber hecho un registro al sistema previamente | |
| Flujo normal | Paso | Acción |
| 1 | El sistema muestra un formulario de inicio de sesión |
| 2 | El usuario ingresará el número de su matricula |
| 3 | El usuario ingresará su contraseña |
| 4 | El usuario hará click en el botón “Ingresar” |
| 5 | El sistema valida la informacion ingresada |
| Flujo alternativo | Paso | Acción |
| 5.1 | En caso de que algún campo tenga informacion incorrecta o incompatible, el sistema mostrará un mensaje notificando el error |
| 5.2 | En caso de que los datos no existan en la base de datos o alguno de ellos sea incorrecto, el sistema mostrará un mensaje de error diciendo “Datos incorrectos, inténtalo de nuevo” |
| 5.3 | En caso de que los datos sean correctos y pertenecientes a un Estudiante, el sistema otorgará acceso al usuario en la vista de estudiantes |
| 5.4 | En caso de que los datos sean correctos y pertenecientes a un Administrador, el sistema otorgará acceso al usuario en la vista de administradores |
| Postcondición | El sistema otorgará acceso al usuario, pudiendo diferenciar si es estudiante o administrador y mandarlos a una vista correspondiente a su información | |

**Figura 1.7** Especificación del caso de uso “Iniciar sesión”

La figura 1.8 muestra la especificación del caso de uso “Consultar tema”. Tanto los estudiantes como administradores pueden consultar la información de los temas de acuerdo a las necesidades de la actividad que estén realizando en ese momento.

|  |  |  |
| --- | --- | --- |
| CU-0005 | Consultar tema | |
| Versión | 1.0 (30/09/2024) | |
| Prioridad | Alta | |
| Autor/es | Joel González Cruz | |
| Actor/es | Estudiante, Administrador | |
| Dependencias | Ninguna | |
| Descripción | El sistema consultará informacion sobre los temas almacenados en la base de datos de acuerdo a las peticiones del usuario | |
| Precondición | El usuario deberá de haber iniciado sesión | |
| Flujo normal | Paso | Acción |
| 1 | El sistema consultará información básica (como el nombre y resumen) sobre los temas registrados en la base de datos |
| 2 | El sistema mostrará la informacion obtenida de la consulta en pantalla |
| 3 | El usuario hará click en cualquier tema en pantalla |
| 4 | El sistema consultará toda la informacion del tema seleccionado por el usuario |
| 5 | El sistema muestra el examen diagnóstico y actividades asociados al tema consultado |
| Flujo alternativo | Paso | Acción |
| 4.1 | En caso de que haya ocurrido algún problema con la consulta, el sistema notificará al usuario del problema ocurrido |
| Postcondición | El sistema devuelve y/o muestra la información asociada a dicho tema de la base de datos | |

**Figura 1.8** Especificación del caso de uso “Consultar tema”

La figura 1.9 muestra la especificación del caso de uso “Consultar test”. Los estudiantes pueden consultar informacion básica del examen diagnóstico, mientras que los administradores pueden acceder a la informacion completa de cada examen diagnóstico.

|  |  |  |
| --- | --- | --- |
| CU-0009 | Consultar test | |
| Versión | 1.0 (30/09/2024) | |
| Prioridad | Alta | |
| Autor/es | Joel González Cruz | |
| Actor/es | Estudiante, Administrador | |
| Dependencias | Ninguna | |
| Descripción | El sistema consultará la informacion del test a las peticiones del usuario | |
| Precondición | El usuario deberá de seleccionar un tema previamente | |
| Flujo normal | Paso | Acción |
| 1 | El usuario hará click en la sección “Examen diagnóstico” |
| 2 | El sistema consultará el examen diagnóstico asociado al tema actual |
| 3 | El sistema obtiene la información del examen diagnóstico consultado |
| 4 | El sistema muestra la informacion obtenida en la pantalla |
| Flujo alternativo | Paso | Acción |
| 2.1 | En caso de que no exista algún examen diagnostico asociado al tema actual, el sistema notificará al usuario del problema ocurrido |
| Postcondición | El sistema devuelve y/o muestra la información asociada a dicho examen diagnóstico de la base de datos | |

**Figura 1.9** Especificación del caso de uso “Consultar test”

La figura 1.10 muestra la especificación del caso de uso “Realizar test”. La *App* extrae las preguntas del examen, las desordena y las aplica al estudiante, este debe de responder el examen en su totalidad para obtener su resultado.

|  |  |  |
| --- | --- | --- |
| CU-0016 | Realizar test | |
| Versión | 1.0 (30/09/2024) | |
| Prioridad | Alta | |
| Autor/es | Joel González Cruz | |
| Actor/es | Estudiante | |
| Dependencias | Ninguna | |
| Descripción | El sistema extraerá la informacion del examen diagnóstico y las preguntas asociadas a él en la base de datos, posteriormente generará un examen para el estudiante el cual deberá ser respondido, al final el sistema arrojará un resultado de acuerdo a las respuestas ingresadas por el estudiante | |
| Precondición | El usuario deberá dar click en el botón “Realizar examen diagnostico” | |
| Flujo normal | Paso | Acción |
| 1 | El sistema consultará todas las preguntas asociadas al examen diagnostico actual |
| 2 | El sistema desordenará las preguntas obtenidas en la consulta y seleccionará las primeras “n” cantidad de preguntas |
| 3 | El sistema mostrará la pregunta en curso con su sentencia, material de apoyo en caso de que lo tenga y sus posibles respuestas |
| 4 | El usuario seleccionará la respuesta que considere más adecuada |
| 5 | El usuario dará click en el botón “Siguiente” |
| 6 | Mientras existan preguntas posteriores, repetir los pasos 3 al 5 |
| 7 | El usuario hará click en el botón “Terminar” |
| 8 | El sistema calcula el nivel de conocimiento obtenido de acuerdo al modelo de asignación diagnostica (consulte los documentos anexos) |
| 9 | El sistema devuelve y muestra el resultado del examen diagnóstico |
| Flujo alternativo | Paso | Acción |
| 1.1 | En caso de que no existan preguntas asociadas al examen diagnóstico actual, el sistema notificará al usuario del error |
| 2.1 | En caso de que la “n” cantidad de preguntas sea menor al total de preguntas disponibles asociadas al examen diagnostico actual, el sistema notificará al usuario del error |
| 5.1 | En caso de que el usuario no haya seleccionado alguna respuesta, el sistema notificará al usuario de que en caso de continuar, la pregunta actual no sumara algún puntaje, el mensaje podrá aceptarse (Omitir pregunta) o rechazarse (Responder pregunta) |
| 5.2 | En caso de que el usuario omita la pregunta, el sistema asignará un puntaje de 0 a la pregunta actual y continuará el flujo en el paso 6 |
| 5.3 | En caso de que el usuario prefiera responder la pregunta, el sistema continuará con flujo normal |
| Postcondición | El sistema almacenara en la base de datos el resultado del nivel de conocimiento del alumno y otorgará acceso a las actividades del tema actual | |

**Figura 1.10** Especificación del caso de uso “Realizar test”

## 1.5 Diagramas de secuencia

Para representar gráficamente las especificaciones de caso de uso del proyecto “Desarrollo App Educativa 4.0” se emplearán diagramas de secuencia, de igual manera solo se mostraran 5 de los más importantes.

La figura 1.11 muestra el diagrama de secuencia del caso de uso “Registrar cuenta”. El usuario ingresa sus datos personales además de su matrícula y contraseña, si la matrícula ya existe en la base de datos, muestra un mensaje de usuario existente, en caso contrario registra la nueva información.



**Figura 1.11** Diagrama de secuencia “Registrar cuenta”

La figura 1.12 muestra el diagrama de secuencia del caso de uso “Iniciar sesión”. El usuario ingresa su matrícula y contraseña, si los datos son correctos, se verifica si el usuario es administrador o estudiante para otorgar o denegar acceso a vistas de administrador. Posterior a ello se le da acceso a la *App*.



**Figura 1.12** Diagrama de secuencia “Iniciar sesión”

La figura 1.13 muestra el diagrama de secuencia del caso de uso “Consultar tema”. El usuario hace click en algún tema y este consulta los exámenes y actividades asignadas a él.



**Figura 1.13** Diagrama de secuencia “Consultar tema”

La figura 1.14 muestra el diagrama de secuencia del caso de uso “Consultar test”. El usuario hace click en el examen diagnóstico del tema previamente seleccionado. Si existe el examen registrado al tema previo, lo muestra, en caso contrario no.



**Figura 1.14** Diagrama de secuencia “Consultar test”

La figura 1.15 muestra el diagrama de secuencia del caso de uso “Realizar test”. El usuario hace click en el botón “Realizar examen”. El sistema extrae las preguntas asociadas al examen seleccionado, desordena las preguntas y selecciona el número de preguntas configurado en el examen. Por cada pregunta muestra el enunciado y las respuestas asignadas a dicha pregunta. Al terminar las preguntas se calcula el resultado y se le muestra al usuario.



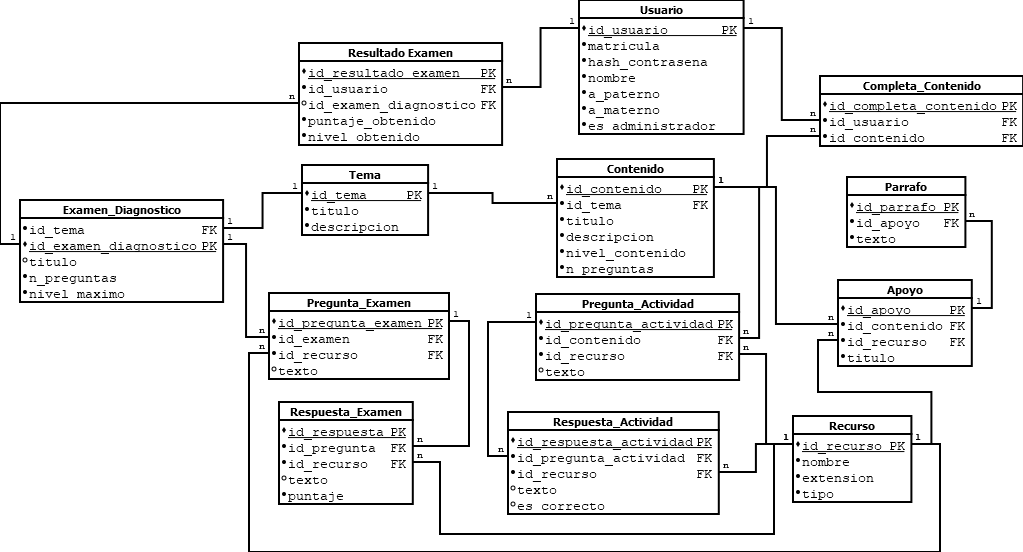
**Figura 1.15** Diagrama de secuencia “Realizar test”

# CAPÍTULO 2. INICIALIZACIÓN

El segundo capítulo contiene el diseño de la base de datos, incluyendo su diagrama relacional y diccionario de datos. Además de contener el diseño de algunas vistas de la aplicación.

## 2.1 Diagrama relacional de base de datos

La figura 2.1 muestra el diagrama relacional de la base de datos de “Desarrollo App Educativa 4.0”



**Figura 2.1** Diagrama relacional de la base de datos de “Desarrollo de App Educativa 4.0”

## 2.2 Diccionario de datos

El diccionario de datos es la especificación de cada uno de los campos de todas las tablas de la base de datos empleada en “Desarrollo App Educativa 4.0”

La figura 2.2 muestra el diccionario de datos de la tabla “Usuario”. Esta tabla almacena informacion básica como: matricula, nombre, apellido paterno, materno y el hash de la contraseña de usuario.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB USUARIO** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_usuario | serial | N/A | PK | Llave primaria de registro de usuario |
| matricula | varchar | 10 | N/A | Número de matrícula del usuario |
| hash\_contrasena | varchar | 32 | N/A | Hash decimal de la contraseña del usuario |
| nombre | varchar | 50 | N/A | Nombre del usuario |
| a\_paterno | varchar | 50 | N/A | Apellido paterno del usuario |
| a\_meterno | varchar | 50 | N/A | Apellido materno del usuario |
| es\_administrador | boolean | N/A | N/A | Valor lógico sobre usuario administrador |

**Figura 2.2** Diccionario de datos de la tabla “Usuario”

La figura 2.3 muestra el diccionario de datos de la tabla “Resultado\_Examen”. Esta tabla almacena el id de usuario que haya realizado al examen así como el id del examen, el puntaje obtenido por el usuario y el nivel obtenido de acuerdo a la configuración establecida por el examen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB RESULTADO EXAMEN** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_resultado\_examen | serial | N/A | PK | Llave primaria de registro del resultado obtenido por el usuario en un examen diagnostico |
| id\_usuario | int | N/A | FK | Llave foránea obtenida de la tabla “Usuario” |
| id\_examen\_diagnostico | int | N/A | FK | Llave foránea obtenida de la tabla “Examen\_Diagnostico” |
| puntaje\_obtenido | int | N/A | N/A | Puntuación obtenida por el estudiante al finalizar el examen |
| nivel\_obtenido | int | N/A | N/A | Nivel de conocimiento del usuario obtenido al final del examen |

**Figura 2.3** Diccionario de datos de la tabla “Resultado\_Examen”

La figura 2.4 muestra el diccionario de datos de la tabla “Examen\_Diagnostico”. Esta tabla almacena la llave primaria del tema, titulo de examen y la configuración del mismo como es el número de preguntas aplicadas y el nivel de conocimiento máximo que se puede obtener por el estudiante.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB EXAMEN DIAGNOSTICO** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_examen\_diagnostico | serial | N/A | PK | Llave primaria de registro de examen diagnóstico |
| id\_tema | int | N/A | FK | Llave foránea obtenida de la tabla “Tema” |
| titulo | varchar | 100 | N/A | Título del examen |
| n\_preguntas | int | N/A | N/A | Número de preguntas aplicables durante el examen |
| nivel\_maximo | int | N/A | N/A | Nivel máximo de conocimiento obtenible por el usuario |

**Figura 2.4** Diccionario de datos de la tabla “Examen\_Diagnostico”

La figura 2.5 muestra el diccionario de datos de la tabla “Tema”. Esta tabla almacena información básica del tema como puede ser su título y una breve descripción que ayuda a comprender de que trata o que contenidos puede tener el tema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB TEMA** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_tema | serial | N/A | PK | Llave primaria de registro de tema |
| titulo | varchar | 100 | N/A | Título del tema |
| descripcion | varchar | 250 | N/A | Descripción del tema |

**Figura 2.5** Diccionario de datos de la tabla “Tema”

La figura 2.6 muestra el diccionario de datos de la tabla “Pregunta\_Examen”. Esta tabla guarda el texto de la pregunta perteneciente a un examen diagnóstico, este puede poseer la llave primaria de un recurso grafico si es que lo requiere.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB PREGUNTA EXAMEN** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_pregunta\_examen | serial | N/A | PK | Llave primaria de registro de pregunta de examen diagnostico |
| id\_examen | int | N/A | FK | Llave foránea obtenida de la tabla “Examen Diagnostico” |
| id\_recurso | int | N/A | FK | Llave foránea obtenida de la tabla “Recurso” |
| texto | varchar | 250 | N/A | Enunciado de la pregunta |

**Figura 2.6** Diccionario de datos de la tabla “Pregunta\_Examen”

La figura 2.7 muestra el diccionario de datos de la tabla “Respuesta\_Examen”. Esta tabla guarda la informacion de una posible respuesta asociada a una pregunta asociada a un examen diagnóstico, en caso de necesitar algún recurso gráfico se puede guardar la llave primaria de algún recurso.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB RESPUESTA EXAMEN** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_respuesta\_examen | serial | N/A | PK | Llave primaria de registro de respuesta |
| id\_pregunta\_examen | int | N/A | FK | Llave foránea obtenida de la tabla “Pregunta\_Examen” |
| id\_recurso | int | N/A | FK | Llave foránea obtenida de la tabla “Recurso” |
| texto | varchar | 50 | N/A | Enunciado de la respuesta |
| puntaje | int | N/A | N/A | Puntaje de la respuesta |

**Figura 2.7** Diccionario de datos de la tabla “Respuesta”

La figura 2.8 muestra el diccionario de datos de la tabla “Completa\_Contenido”. Esta tabla guarda el registro de los contenidos completados por cada uno de los estudiantes, almacena la llave primaria del usuario y la llave primaria del contenido para indicar que dicho contenido está completo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB COMPLETA CONTENIDO** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_completa\_contenido | serial | N/A | PK | Llave primaria de registro de contenido completado por el usuario |
| id\_usuario | int | N/A | FK | Llave foránea obtenida de la tabla “Usuario” |
| id\_contenido | int | N/A | FK | Llave foránea obtenida de la tabla “Contenido” |

**Figura 2.8** Diccionario de datos de la tabla “Completa\_Contenido”

La figura 2.9 muestra el diccionario de datos de la tabla “Contenido”. Esta tabla guarda información básica como título y descripción, además de una pequeña configuración que incluye nivel de contenido y número de preguntas aplicables como actividad.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB CONTENIDO** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_respuesta | serial | N/A | PK | Llave primaria de registro de contenido |
| id\_tema | int | N/A | FK | Llave foránea obtenida de la tabla “Tema” |
| titulo | varchar | 50 | N/A | Título del contenido |
| descripción | varchar | 100 | N/A | Descripción del contenido |
| nivel\_contenido | int | N/A | N/A | Nivel del contenido |
| n\_preguntas | int | N/A | N/A | Número de preguntas aplicables durante la actividad |

**Figura 2.9** Diccionario de datos de la tabla “Contenido”

La figura 2.10 muestra el diccionario de datos de la tabla “Pregunta\_Actividad”. Esta tabla guarda el texto de la pregunta de la actividad y en caso de necesitar algún recurso gráfico puede almacenar la llave primaria del recurso necesario.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB PREGUNTA ACTIVIDAD** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_pregunta\_actividad | serial | N/A | PK | Llave primaria de registro de pregunta de actividad |
| id\_contenido | int | N/A | FK | Llave foránea obtenida de la tabla “Contenido” |
| id\_recurso | int | N/A | FK | Llave foránea obtenida de la tabla “Recurso” |
| texto | varchar | 250 | N/A | Enunciado de la pregunta |

**Figura 2.10** Diccionario de datos de la tabla “Pregunta\_Actividad”

La figura 2.11 muestra el diccionario de datos de la tabla “Respuesta\_Actividad”. Esta tabla guarda una posible respuesta a una pregunta de actividad, almacena su texto, si la respuesta es correcta y la llave primaria de algún recurso gráfico en caso de necesitarlo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB RESPUESTA ACTIVIDAD** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_respuesta\_actividad | serial | N/A | PK | Llave primaria de registro de respuesta de actividad |
| id\_pregunta\_actividad | int | N/A | FK | Llave foránea obtenida de la tabla “Pregunta\_Actividad” |
| id\_recurso | int | N/A | FK | Llave foránea obtenida de la tabla “Recurso” |
| texto | varchar | 50 | N/A | Enunciado de la respuesta |
| es\_correcto | boolean | N/A | N/A | Valor lógico de si la respuesta es correcta o no |

**Figura 2.11** Diccionario de datos de la tabla “Respuesta\_Actividad”

La figura 2.12 muestra el diccionario de datos de la tabla “Recurso”. Esta tabla almacena información básica y de referencia para los recursos gráficos que se necesiten emplear. Tal como un nombre, extensión de archivo y tipo de archivo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB RECURSO** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_recurso | serial | N/A | PK | Llave primaria de registro de recurso |
| nombre | varchar | 32 | N/A | Nombre del recurso |
| extensión | varchar | 5 | N/A | Terminación o extensión del recurso |
| tipo | varchar | 24 | N/A | Tipo de recurso |

**Figura 2.12** Diccionario de datos de la tabla “Recurso”

La figura 2.13 muestra el diccionario de datos de la tabla “Apoyo”. Esta tabla guarda la información del apoyo didáctico perteneciente a algún contenido, almacena un título y la llave primaria de algún recurso gráfico en caso de necesitarla.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB APOYO** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_apoyo | Serial | N/A | PK | Llave primaria de registro de apoyo |
| id\_contenido | Int | N/A | FK | Llave foránea obtenida de la tabla “Contenido” |
| id\_recurso | Int | N/A | FK | Llave foránea obtenida de la tabla “Recurso” |
| titulo | varchar | 100 | N/A | Título o encabezado del apoyo |

**Figura 2.13** Diccionario de datos de la tabla “Apoyo”

La figura 2.14 muestra el diccionario de datos de la tabla “Parrafo”. Esta tabla almacena un párrafo de texto perteneciente a algún apoyo didáctico a fin de que los estudiantes comprendan de una manera más adecuada el material de apoyo que se les asigna.

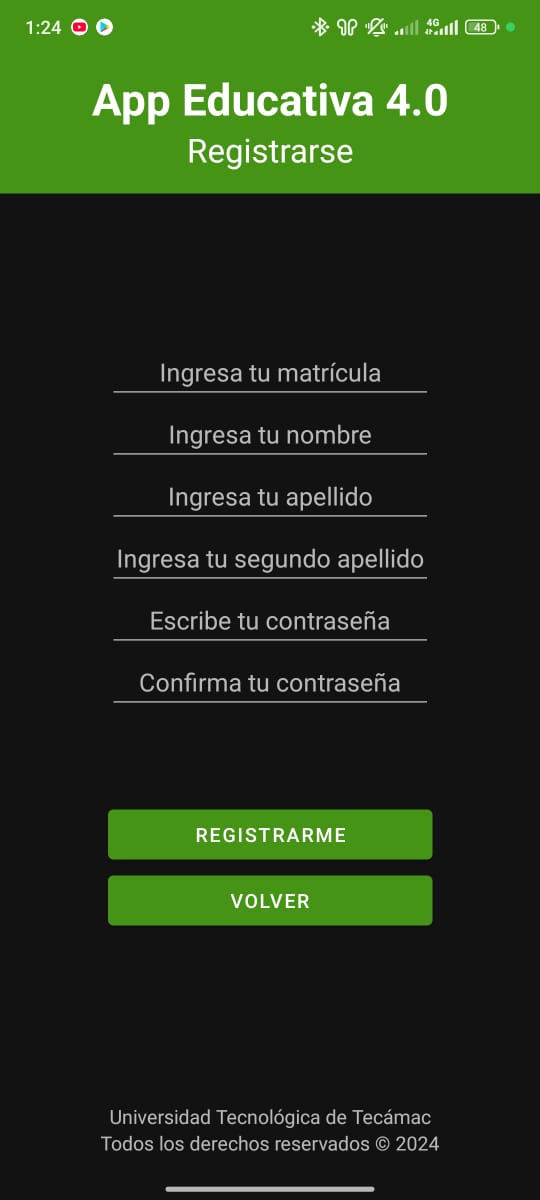
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TB PARRAFO** | | | | |
| **Campo** | **Tipo** | **Extensión** | **Restricción** | **Observaciones** |
| id\_parrafo | serial | N/A | PK | Llave primaria de registro de párrafo |
| id\_apoyo | int | N/A | FK | Llave foránea obtenida de la tabla “Apoyo” |
| texto | varchar | 250 | N/A | Texto del parrafo |

**Figura 2.14** Diccionario de la tabla “Parrafo”

## 2.3 Diseño de interfaces

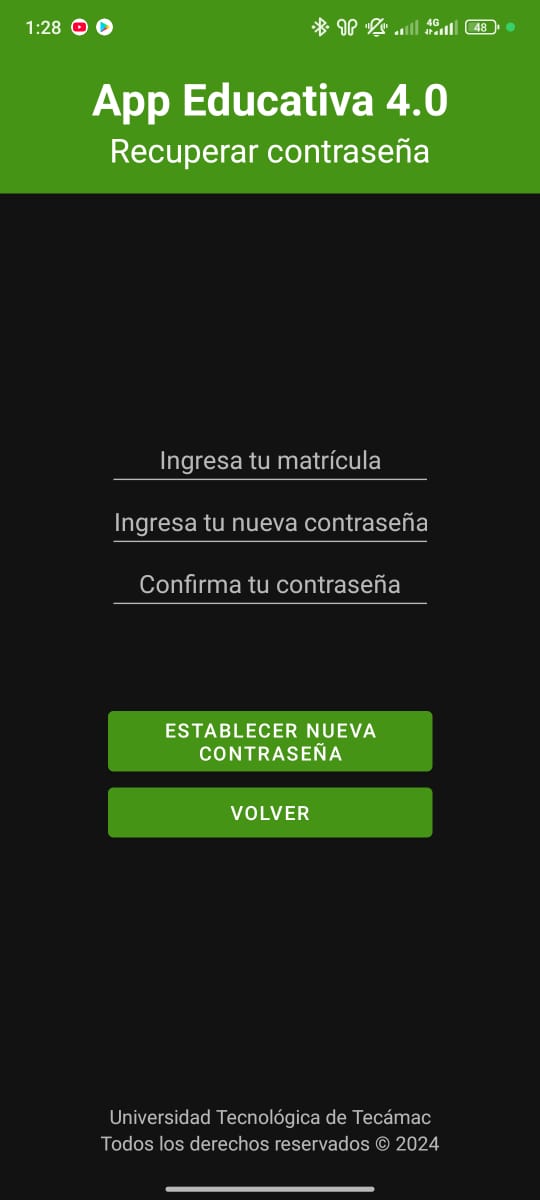
Se tomó la decisión de pasar directamente al diseño y no a un maquetado, ya que previo a esto se contaban con bocetos de la aplicación, los cuales eran suficientes para comenzar con el diseño de interfaces. Únicamente se mostrarán las pantallas más importantes de la *App* “Desarrollo App Educativa 4.0”.

La figura 2.15 muestra la pantalla de registro. En esta pantalla los estudiantes pueden registrarse a la App, ingresando su matrícula, nombre y estableciendo una contraseña de acceso.



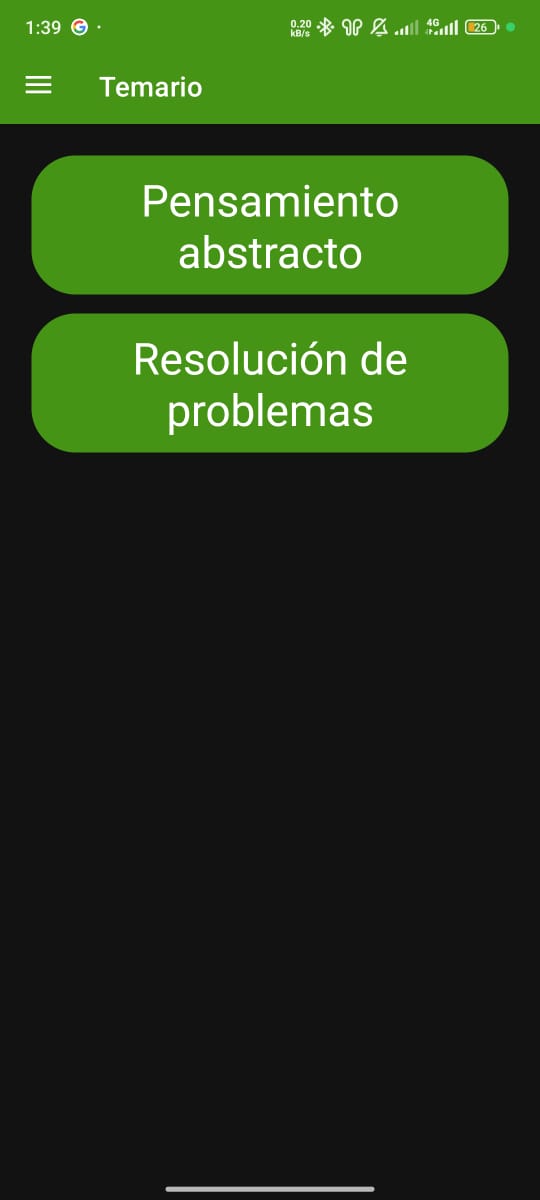
**Figura 2.15** Pantalla de registro

La figura 2.16 muestra la pantalla de recuperación de contraseña. Los usuarios registrados en la *App* ingresan su matrícula y establecen una nueva contraseña.



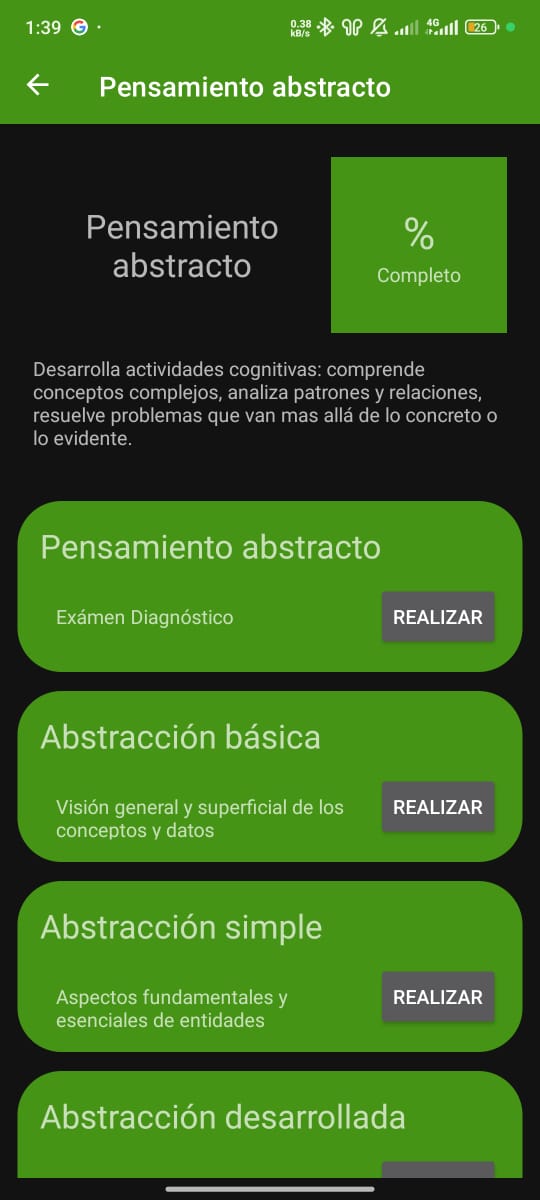
**Figura 2.16** Pantalla de recuperación de contraseña

La figura 2.17 muestra la pantalla de la sección “Temario”. Una vez el usuario haya accedido con su matrícula y contraseña, se le mandará a esta vista en donde podrá consultar los temas disponibles en la base de datos. Al hacer click, el usuario accede a los contenidos del tema seleccionado.



**Figura 2.17** Pantalla de sección “Temario”

La figura 2.18 muestra la pantalla de tema. Esta pantalla muestra el título, porcentaje de actividades completadas, y contenido relacionado al tema seleccionado previamente por el usuario en la vista “Temario”.



**Figura 2.18** Pantalla de tema

La figura 2.19 muestra la pantalla de cuestionario. Esta vista se muestra cuando el usuario entra a realizar algún examen diagnóstico o actividad. Se muestra el título del elemento al que pertenece, el número de pregunta actual y total de preguntas, así como la pregunta y las posibles respuestas.



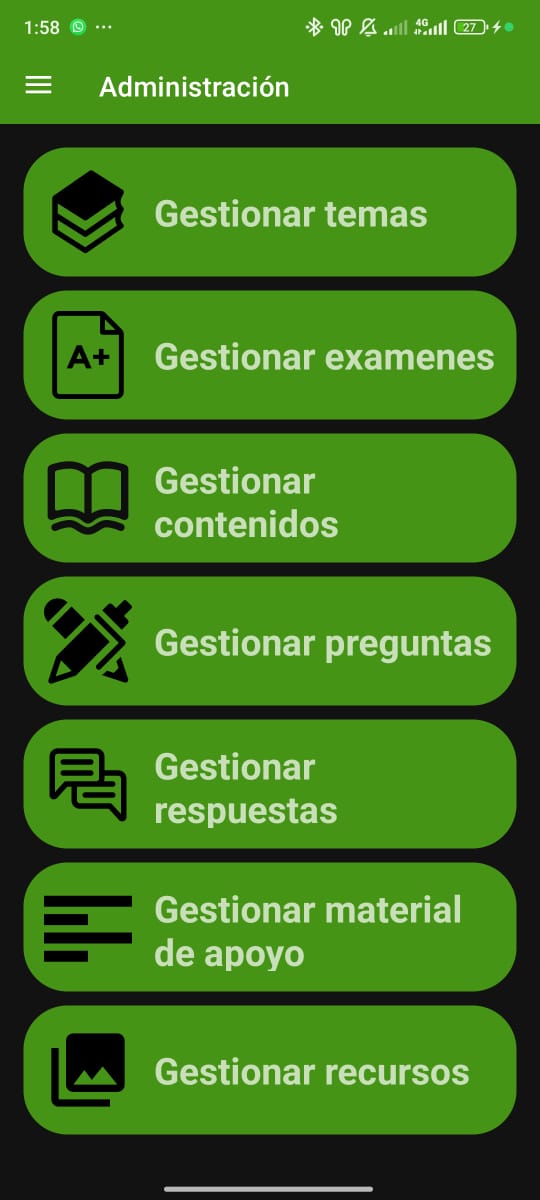
**Figura 2.19** Pantalla de cuestionario

La figura 2.20 muestra la pantalla de la sección “Perfil”. En el menú de navegación el usuario puede dar click en la sección “Perfil”, en esta vista se pueden editar algunos datos personales y establecer una nueva contraseña si es necesario.



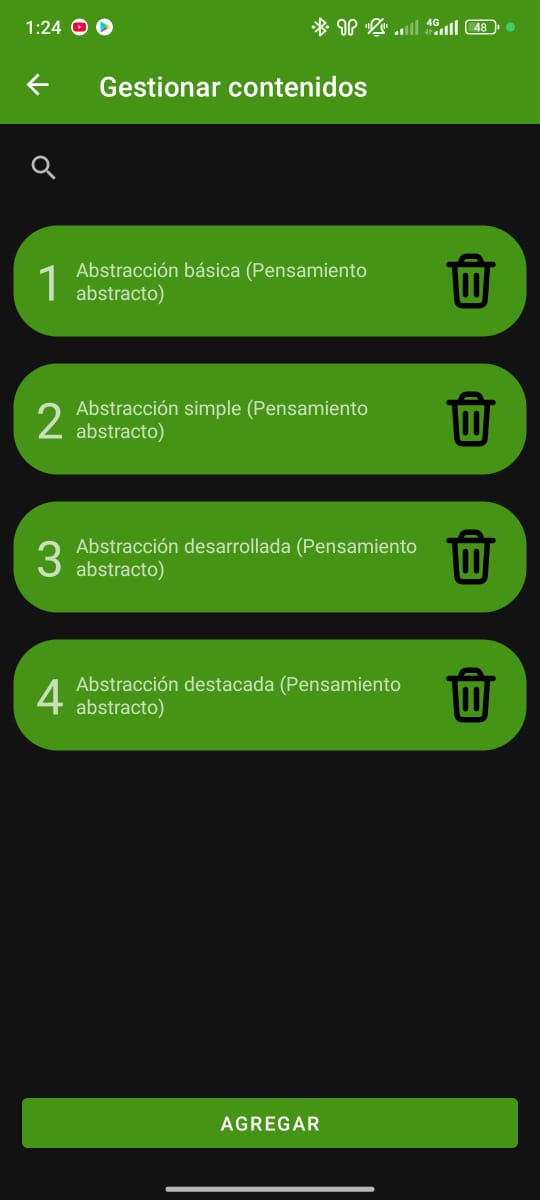
**Figura 2.20** Pantalla de sección “Perfil”

La figura 2.21 muestra la pantalla de la sección “Administración”. Esta sección solo está disponible para usuarios administradores. En ella, los administradores pueden consultar, agregar, actualizar y eliminar información como temas, exámenes, preguntas, recursos, etc.



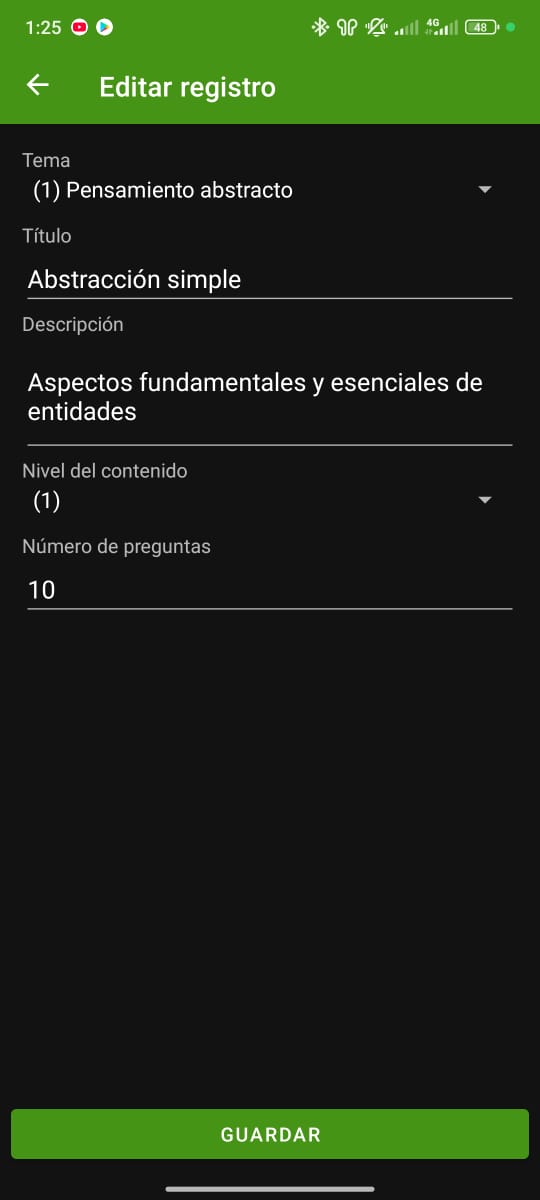
**Figura 2.21** Pantalla de la sección “Administración”

La figura 2.22 muestra la pantalla de gestión de la sección “Administración”. Esta vista extrae los registros asociados al botón que se presionó anteriormente. Cada uno de los elementos posee su identificador, un texto representativo y un botón para eliminarlo.



**Figura 2.22** Pantalla de gestión de administración

La figura 2.23 muestra la pantalla de formulario de registro. El formulario se genera de acuerdo a la información requerida, además de que este mismo se usa para modificar la información de un registro existente.



**Figura 2.23** Pantalla de formulario

# CAPÍTULO 3. PRODUCCIÓN

El tercer capítulo presenta la especificación de interfaces o vistas del prototipo “Desarrollo App Educativa 4.0”. Cada una de las vistas detalla sus elementos y comportamiento general.

La figura 3.1 Muestra la vista de inicio de sesión. Los usuarios previamente registrados ingresan su matrícula y contraseña para acceder a la *App,* también pueden registrarse en caso de no tener una cuenta o establecer una nueva contraseña en caso de que hayan olvidado o extraviado su contraseña. La vista se encuentra en “res.layout.activity\_welcome.xml” y su controlador se encuentra en “app.welcome\_views.Welcome.java”.



**Figura 3.1** Pantalla de inicio sesión

La figura 3.2 muestra una descripción de elementos de la vista activity\_welcome.xml. Estos elementos incluyen dos *EditTexts* para ingresar la matrícula y contraseña del usuario y cuatro *Buttons* para acceder, navegar a otra vista o salir de la *App*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Tipo de elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | input\_matricula | EditText | Entrada de la matricula del usuario | String | matricula |
| 2 | input\_password | EditText | Entrada de la contraseña del usuario | String | hash\_contrasena |
| 3 | btn\_access | Button | Botón para ingresar a la *App* | N/A | N/A |
| 4 | btn\_signup | Button | Botón para registrarse | N/A | N/A |
| 5 | btn\_forgot | Button | Botón para restablecer contraseña | N/A | N/A |
| 6 | btn\_exit | Button | Botón para cerrar la *App* | N/A | N/A |

**Figura 3.2** Tabla de elementos de la vista de inicio de sesión

La figura 3.3 muestra la vista de registro. Los usuarios no registrados ingresan su matrícula, ingresan su nombre, apellido paterno y materno, establecen una contraseña mayor a 8 caracteres con al menos un número, una letra mayúscula y una letra minúscula. La vista se encuentra en “res.layout.activity\_sing\_up.xml” y su controlador se encuentra en “app.welcome\_views.SignUp.java”.



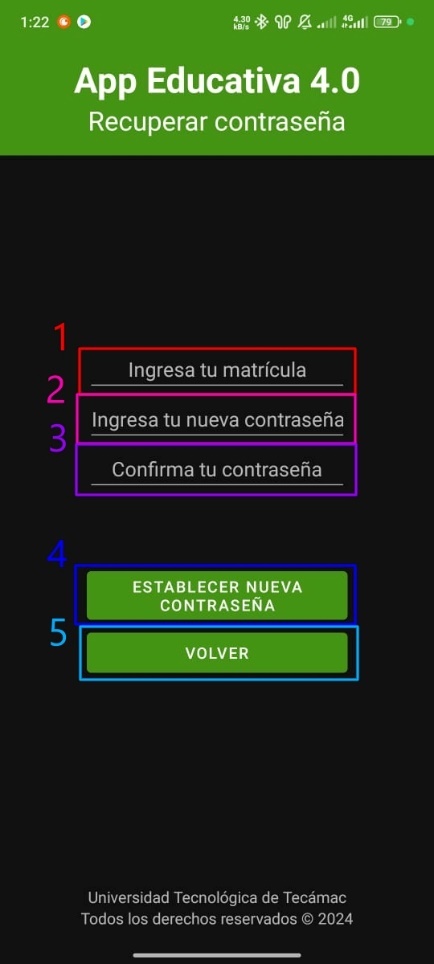
**Figura 3.3** Pantalla de registro

La figura 3.4 muestra una descripción de elementos de la vista activity\_sing\_up.xml. Estos elementos incluyen seis *EditTexts* para ingresar los datos personales del estudiante y dos *Buttons* para registrar al estudiante o volver a la vista anterior.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Tipo de elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | input\_matricula | EditText | Entrada de la matricula del usuario | String | matricula |
| 2 | input\_name | EditText | Entrada del nombre de usuario | String | nombre |
| 3 | input\_surname | EditText | Entrada del apellido paterno del usuario | String | a\_paterno |
| 4 | input\_surname2 | EditText | Entrada del apellido materno del usuario | String | a\_materno |
| 5 | input\_password | EditText | Entrada de la contraseña del usuario | String | hash\_contrasena |
| 6 | input\_confirm\_password | EditText | Entrada de confirmación de la contraseña del usuario | String | hash\_contrasena |
| 7 | btn\_sign\_up | Button | Botón para registrar la información | N/A | N/A |
| 8 | btn\_to\_sign\_in | Button | Botón para volver a la vista de inicio de sesión | N/A | N/A |

**Figura 3.4** Tabla de elementos de la vista de registro

La figura 3.5 muestra la vista de reestablecer contraseña. Los usuarios registrados ingresan su matricula y establecen una nueva contraseña la cual debe tener los mismos criterios para establecer una contraseña: mayor a 8 caracteres con al menos un número, una letra mayúscula y una letra minúscula. Tienen que volver a escribir la misma contraseña que definieron nuevamente para confirmar que la contraseña es idéntica en ambas entradas. La vista se encuentra en “res.layout.activity\_forgot\_password.xml” y su controlador se encuentra en “app.welcome\_views.ForgotPassword.java”.



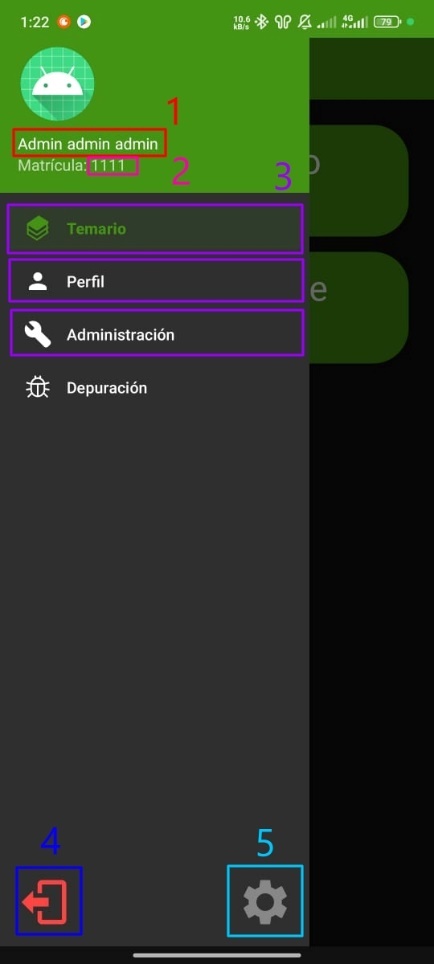
**Figura 3.5** Pantalla de recuperación

La figura 3.6 muestra una descripción de los elementos de la vista activity\_forgot\_password.xml. Estos elementos incluyen tres *EditTexts* para que el usuario coloque su matrícula, establezca y confirme una nueva contraseña y 2 *Buttons* para reestablecer la contraseña o volver a la vista anterior.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Tipo de elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | matricula | EditText | Entrada de la matricula del usuario | String | N/A |
| 2 | input\_new\_password | EditText | Entrada de la nueva contraseña del usuario | String | hash\_contrasena |
| 3 | input\_password\_confirm | EditText | Entrada de confirmación de la nueva contraseña del usuario | String | hash\_contrasena |
| 4 | btn\_set\_new\_password | Button | Botón para reestablecer la contraseña | N/A | N/A |
| 5 | btn\_to\_sign\_in | Button | Botón para volver a la vista de inicio de sesión | N/A | N/A |

**Figura 3.6** Tabla de elementos de vista de recuperación

La figura 3.7 muestra el menú despegable, este cuenta con una imagen predeterminada, el nombre completo del usuario activo y su matrícula posee elementos para navegar a las diferentes secciones de la aplicación que en el caso de los estudiantes se muestran menos, también contiene iconos presionables para cerrar la sesión actual y acceder a las configuraciones. La vista se encuentra estructurada en diversas vistas y controladores, los cuales son activity\_main\_drawer.xml, activity\_main.xml, app\_bar\_main.xml, content\_main.xml, nav\_header\_main.xml y MainActivity.java



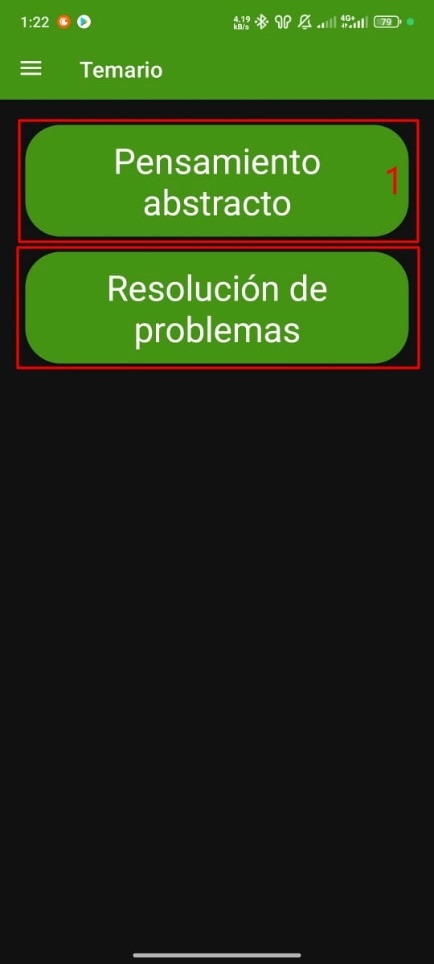
**Figura 3.7** Menú desplegable

La figura 3.8 muestra una descripción de los elementos de que componen el menú desplegable. Estos incluyen un conjunto de *TextViews* para mostrar el nombre completo del usuario actual otro *TextView* para mostrar la matrícula del usuario, también un conjunto de *Items* para navegar a las diferentes secciones de la *App* y dos *ImageButtons* para cerrar la sesión actual y acceder a las configuraciones respectivamente.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Tipo de elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | user\_name, user\_surname, user\_surname2 | TextView | Nombre completo del usuario activo | N/A | N/A |
| 2 | user\_matricula | TextView | Matrícula del usuario activo | N/A | N/A |
| 3 | nav\_themes,  nav\_provile, nav\_admin | Item | Elemento navegable | N/A | N/A |
| 4 | btn\_logout | ImageButton | Imagen presionable para cerrar la sesión | N/A | N/A |
| 5 | btn\_settings | ImageButton | Imagen presionable para acceder a las configuraciones | N/A | N/A |

**Figura 3.8** Tabla de elementos del menú desplegable

Figura 3.9 muestra la vista principal de la sección “Temario” los estudiantes que hayan accedido a la *App* visualizan el título de todos los temas disponibles, al presionarlos estos navegan hasta el contenido de dichos temas. La vista se encuentra en “res.layout.fragment\_themes.xml” y su controlador se encuentra en “ui.themes.ThemesFragment.java”.



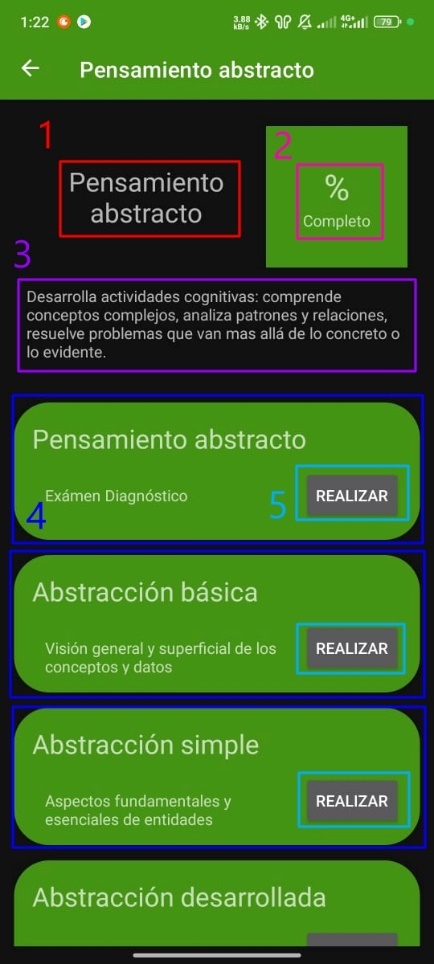
**Figura 3.9** Pantalla de sección “Temario”

La figura 3.10 muestra una descripción de los elementos de la vista fragment\_themes.xml. Estos elementos son *TextView* donde se genera uno distinto por cada tema disponible dentro de la *App*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Tipo de elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | theme | TextView | Texto presionable del título del tema extraído de la base de datos | N/A | N/A |

**Figura 3.10** Tabla de elementos de la vista de la sección “Temario”

La figura 3.11 muestra la vista de tema. La carga de esta vista depende del título enviado a través de un *bundle* desde la vista anterior “Temario”. En esta vista se muestra el título del tema, porcentaje de contenidos completados, descripción de tema, un examen diagnóstico y contenidos de tema. En caso de que el estudiante no haya realizado el examen diagnóstico únicamente se le muestra el examen realice el examen diagnóstico, si es que existe algún registro del examen diagnostico completado, se le asigna el contenido correspondiente a los resultados obtenidos del examen. La vista se encuentra en “res.layout.fragment\_theme.xml” y su controlador se encuentra en “ui.themes.theme.ThemeFragment.java”.



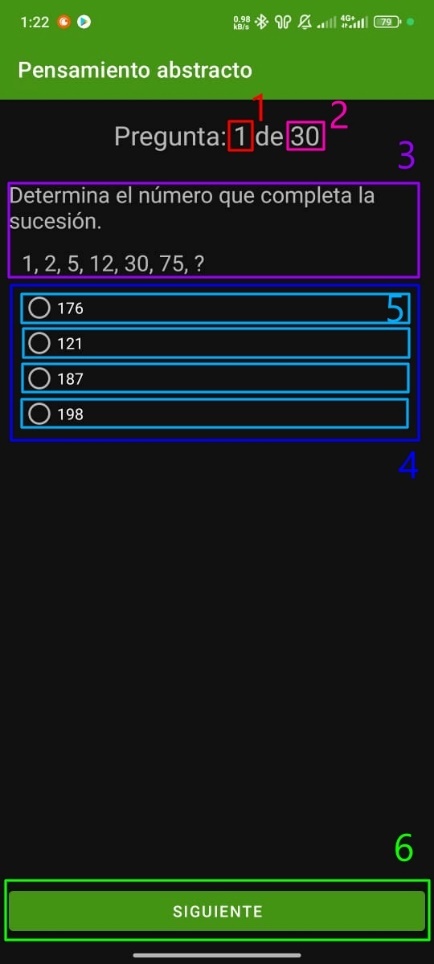
**Figura 3.11** Pantalla de tema

La figura 3.12 muestra una descripción de los elementos de la vista fragment\_theme.xml. Estos elementos son tres *TextViews* que muestran título, porcentaje de contenido completo y descripción del tema respectivamente, el examen diagnóstico y los contenidos asociados al tema generan un *LinearLayout* con un título, descripción y un *Button* para realizar la actividad correspondiente al elemento.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Tipo de elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | theme\_name | TextView | Texto del título del tema actual | N/A | N/A |
| 2 | theme\_per\_cent | TextView | Texto del porcentaje de contenidos completados | N/A | N/A |
| 3 | theme\_description | TextView | Texto de la descripción del tema | N/A | N/A |
| 4 | layout | LinearLayouy | Recuadro de examen diagnostico o contenido según corresponda | N/A | N/A |
| 5 | button | Button | Botón para realizar cuestionario | N/A | N/A |

**Figura 3.12** Tabla de elementos de la vista de tema

La figura 3.13 muestra la vista de cuestionario. Para el caso de un examen diagnóstico o alguna actividad de un contenido, se extrae todas las preguntas asociadas a dicho cuestionario y se desordenan, se selecciona solo las primeras “n” cantidad de preguntas impuestas por la configuración del cuestionario, por último se le aplica al estudiante. La vista muestra el número de pregunta actual, el total de preguntas, la pregunta y sus posibles respuestas, en caso de que estas requieran alguna imagen también se muestra. Al terminar el cuestionario se realiza el cálculo de resultado correspondiente y se devuelve al estudiante a la vista anterior. La vista se encuentra en “res.layout.fragment\_excercise.xml” y su controlador se encuentra en “ui.themes.excercise.ExcerciseFragment.java”.



**Figura 3.13** Pantalla de cuestionario

La figura 3.14 muestra una descripción de los elementos de la vista fragment\_excercise.xml. Los elementos son dos *TextViews* para indicar el número de pregunta actual y el número de preguntas totales respectivamente, además de un conjunto de *TextViews* que componen la estructura de la pregunta, así como un *RadioGroup* que contiene dos o más *RadioButtons* los cuales son las posibles respuestas a la pregunta actual, por ultimo contiene un *Button* para pasar a la siguiente pregunta.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Tipo de elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | i\_question | TextView | Texto de número de pregunta actual | N/A | N/A |
| 2 | max\_question | TextView | Texto de número total de preguntas aplicadas | N/A | N/A |
| 3 | tvs | TexView | Texto del enunciado de la pregunta | N/A | N/A |
| 4 | rg | RadioGroup | Grupo de respuestas posibles | N/A | N/A |
| 5 | rb | RadioButton | Posible respuesta de pregunta actual | N/A | N/A |
| 6 | btn\_next | Button | Botón para responder la siguiente pregunta | N/A | puntaje\_obtenido, nivel\_obtenido |

**Figura 3.14** Tabla de elementos de la vista de cuestionario

La figura 3.15 muestra la vista de la sección “Perfil”. Los usuarios pueden visualizar su matrícula actualizar su nombre y contraseña en caso de que sea necesario. La vista se encuentra en “res.layout.fragment\_profile.xml” y su controlador se encuentra en “ui.profile.ProfileFragment.java”.



**Figura 3.15** Pantalla de la sección “Perfil”

La figura 3.16 muestra una descripción de los elementos de la vista fragment\_profile.xml. Estos elementos incluyen un *TextView* que muestra el número de matrícula del usuario actual, cinco *EditTexts* para actualizar la información de usuario y dos *Buttons* para descartar los cambios o actualizar la informacion respectivamente.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. Elemento | Elemento | Tipo de elemento | Interacciones | Tipo de dato de entrada | Campo que afecta |
| 1 | user\_matricula | TextView | Texto de la matrícula del usuario | N/A | N/A |
| 2 | user\_name | EditText | Entrada del nombre de usuario | String | nombre |
| 3 | user\_surname | EditText | Entrada del apellido paterno del usuario | String | a\_paterno |
| 4 | user\_surname2 | EditText | Entrada del apellido materno del usuario | String | a\_materno |
| 5 | new\_password | EditText | Entrada de la nueva contraseña de usuario | String | hash\_contrasena |
| 6 | confirm\_password | EditText | Entrada de confirmación de la nueva contraseña de usuario | String | hash\_contrasena |
| 7 | btn\_clear | Button | Botón para descartar los cambios | N/A | N/A |
| 8 | btn\_save | Button | Botón para actualizar la informacion de usuario | N/A | N/A |

**Figura 3.16** Tabla de elementos de la vista de la sección “Perfil”

# CONCLUSIONES

El prototipo “Desarrollo App Educativa 4.0” se desarrolló hasta la tercera fase de la metodología *Mobile-D*: “Producción”, se cumplió con el objetivo general y los dos primeros objetivos específicos, ya que con el estado actual del proyecto no se puede asegurar que la experiencia de aprendizaje sea la más óptima, ya que no se pudo desarrollar una fase de pruebas ni contar con una retroalimentación de los usuarios finales, sea estudiantes, docentes o personal administrativo.

El prototipo “Desarrollo App Educativa 4.0” dispone de todos sus elementos planteados en este documento, aunque no todos son completamente fiables, estos pueden presentar comportamientos anormales o extraños en caso de que la información introducida no esté completa, de que no esté correctamente relacionada, en caso de rotar el dispositivo, hacer gestos en la pantalla, o cualquier otra excepción no esperada. Adicionalmente la *App* cuenta con información precargada como registros de usuarios, temas, exámenes, etc. a modo de demostración.

Algunos de los puntos que no se lograron cubrir fueron:

* **Implementación de material de apoyo**: Aunque la base de datos está completa dentro de la *App*, no fue posible incluir el *CRUD* de la tabla “Apoyo” y “Parrafo”, por lo que la utilización de los registros, así como vistas de apoyo pertenecientes a dichas tablas es inexistente en el estado actual del prototipo.
* **Temas precargados incompletos**: Dentro de la información precargada incluye 2 temas: “Pensamiento abstracto” y “Resolución de problemas”, este último no tiene ningún tipo de información asociada, mientras que el otro dispone de un examen diagnóstico y un contenido declarado. El resto de elementos asociados están incompletos.
* **Porcentaje de actividades completadas**: El prototipo del proyecto “Desarrollo App Educativa 4.0” no dispone de algún algoritmo que calcule el porcentaje de actividades completadas por el estudiante en un tema concreto.

Alguna de las implementaciones que se pueden hacer a futuro es reestructurar el comportamiento de la aplicación ya que los controladores, modelos y clases auxiliares no siempre tienen sus métodos correctamente distribuidos entre ellos, provocando que estas puedan ser confusas de comprender y mantener, así como refactorizar el código existente.

Se pueden expandir las tablas de la base de datos empleada para almacenar más informacion relevante como fechas, tiempos de aplicación durante un examen diagnóstico, etc. Así mismo se pueden mejorar las vistas de administración ya que estas pueden resultar incómodas de usar en caso de que se te tenga que manipular mucha información.

Por último se puede definir un algoritmo que identifique las deficiencias de un estudiante en un tema o contenido específico, ya que el prototipo “Desarrollo App Educativa 4.0” solo dispone de un algoritmo de asignación diagnóstica y no es capaz de ajustar el contenido de actividades o material de apoyo en caso de que el estudiante falle la actividad con frecuencia o presente dificultades en la misma.

El desarrollo del prototipo “Desarrollo App Educativa 4.0” fue una labor desafiante y satisfactoria, ya que se emplearon técnicas nuevas para el equipo de desarrollo como: archivos de navegación, uso de imágenes vectoriales, menús, *bundles* y adaptaciones que se tuvieron que hacer para lograr los requisitos de las especificaciones planteadas por el proyecto.

# LISTADO DE SIGLAS O ACRÓNIMOS

ACID: Atomicity, Consistency, Isolation, Durability

ACID: Atomicidad, Consistencia, Integridad, Durabilidad

API: Application programming interface

API: Interfaz de programación de aplicación

CRUD: Create, Read, Update, Delete

CRUD: Crear, Leer, Actualizar, Eliminar

CU: Caso de uso

DTIC: División de Tecnologías de la Informacion y Comunicación

DVCS: Distributed Version Control System

DVCS: Sistema de Control de Versiones Distribuido

FHD: Full HD (High Definition)

FHD: Alta definición completa

IDE: Integrated Development Environment

IDE: Interfaz de Desarrollo Integrado

RAM: Random Access Memory

RAM: Memoria de Acceso Aleatorio

SQL: Structured Query Language

SQL: Lenguaje de consulta estructurada

TDD: Test Driven Development

TDD: Desarrollo Guiado por Pruebas

TIC: Tecnologías de Información y Comunicación

UML: Unified Modeling Language

UML: Lenguaje de Modelado Unificado

USB: Universal Serial Bus

USB: Bus Universal en Serie

XML: Extensible Markup Language

XML: Lenguaje de Marcado Extensible

# GLOSARIO

Actor: Entidad externa ajena al sistema informático que guarda una relación con este y demanda funcionalidad del mismo, casi siempre se refiere al usuario que usa el sistema.

API: Es un conjunto de reglas o protocolos que permite a las aplicaciones informáticas comunicarse entre sí para intercambiar datos, características y funcionalidades.

Biblioteca: Conjunto de funciones y rutinas predefinidas que facilitan el desarrollo de software al proporcionar una serie de herramientas listas para usar.

Bundle: Colección de archivos o recursos combinados en un solo paquete.

Caso de uso: Secuencia de transacciones desarrolladas por el sistema informático en respuesta a un evento que inicia un actor sobre el propio sistema.

Compilación: Proceso de convertir un programa informático a lenguaje máquina para que este pueda ser ejecutado por una computadora.

CRUD: Es un concepto en base de datos que permite la estandarización de operaciones, lo que permite gestionar información en aplicaciones.

Evento: Acción detectada por un programa informático.

Freemium: Es una estrategia de venta centrada en ofrecer al consumidor una versión gratuita y básica un producto o servicio, pero que dispone de una versión de pago en la que permite el acceso a características o funcionalidades adicionales.

Gestor de base de datos: Software que crea, gestiona y administra la información contenida en una base de datos.

Gráfico: Cualquier imagen generada por computadora.

Hardware: Partes físicas, tangibles, de un sistema informático, sus componentes eléctricos, electrónicos y electromecánicos.

Hilo: Es una unidad de ejecución más pequeña dentro de un proceso informático.

Host: Se refiere al anfitrión conectado a una red que proveen y utilizan servicios de ella, estos son computadoras, servidores, o dispositivos móviles. También hace referencia al mismo dispositivo en caso de que este no esté conectado a una red.

IDE: Tipo de aplicación informática que proporciona servicios integrales para facilitar al desarrollador o programador el desarrollo de software.

Lenguaje de marcado: Forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

Lenguaje de programación: Lenguaje formal o artificial, que proporciona un desarrollador o programador, el cual declara una serie de instrucciones ordenadas para que un sistema informático pueda realizarlas.

Llave foránea: campo o conjunto de campos en una tabla que hacen referencia a la llave principal de otra tabla en una base de datos

Llave primaria: columna especial o conjunto de columnas en una tabla de base de datos que identifica de manera única cada registro en la tabla

Offline: Se traduce como “Fuera de línea”, se refiera a todo aquello que no esté relacionado con internet.

Online: Se traduce como “En línea”, se utiliza para nombrar a algo que está conectado o a alguien que está haciendo uso de una red.

Open-Source: es un modelo de desarrollo de software basado en la colaboración abierta. Se enfoca en los beneficios prácticos (acceso al código fuente) y en cuestiones éticas o de libertad que tanto se destacan en el software libre.

Paradigma: Conjunto de principios y directrices que define un enfoque particular para diseñar, estructurar y escribir código.

Parametrizable: Consiste en una creación de un elemento informático flexible, que pueda modificarse sin necesidad de reescribirse o redefinirse nuevamente.

RAM: Sistema de almacenamiento temporal de datos y programas que el equipo de cómputo este usando en el momento.

Refactorización: Proceso de mantenimiento del código que no arregla errores ni añade funcionalidad. El objetivo, por el contrario, es mejorar la facilidad de comprensión del código o cambiar su estructura y diseño y eliminar código muerto o poco útil.

Renderizado: Consiste en crear una imagen tridimensional a partir de una escena virtual, generada por un software de modelado que incluye elementos como texturas, luces, sombras y cámaras.

Sistema: Conjunto de elementos físicos y lógicos (software) necesarios para captar informacion, almacenarla, y procesarla.

Sistema de control de versiones: Software que ayuda a hacer un seguimiento de los cambios realizados en el código a lo largo del tiempo.

Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en un equipo de cómputo.

Testing: Es una prueba o conjunto de ellas que consiste en analizar si un software o programa informático funciona correctamente.

TIC: Conjunto de recursos, herramientas, equipos de cómputo, programas informáticos, aplicaciones, redes y medios que permiten la compilación, procesamiento, almacenamiento, transmisión de informacion como: datos, texto, imágenes, voz, video, entre muchos otros.

UML: Representación gráfica de una o varias piezas de software, que facilita la comprensión de su estructura y flujo de instrucciones.

Vista: representación visual de la información que el usuario puede ver e interactuar. Muchas veces se comprende como “lo que está en pantalla”.

Web: Red informática.

# ANEXOS

Documento anexo 1, Especificaciones de caso de uso

En el capítulo 1. Análisis, en el punto 1.4 Especificación de casos de uso, se muestran unos pocos casos de uso, a continuación se muestran los 16 casos de uso restantes donde 2 son de alta prioridad, 5 de media prioridad, y 9 de baja prioridad.

La figura 4.1 muestra la especificación del caso de uso “Actualizar información”

|  |  |  |
| --- | --- | --- |
| CU-0003 | Actualizar información | |
| Versión | 1.0 (03/10/2024) | |
| Prioridad | Baja | |
| Autor/es | Joel González Cruz | |
| Actor/es | Estudiante, Administrador | |
| Dependencias | Ninguna | |
| Descripción | El usuario podrá actualizar su información de usuario, permitiendo que el sistema guarde la información nueva | |
| Precondición | El usuario deberá de haber iniciado sesión | |
| Flujo normal | Paso | Acción |
| 1 | El usuario hará click en la sección “Actualizar información” |
| 2 | El sistema cargará la vista de “Actualizar información” |
| 3 | El sistema consultará la información en la base de datos del usuario actual y la mostrará al usuario en un formulario |
| 4 | El usuario modifica la información que desee actualizar |
| 5 | El usuario dará click en el botón “Actualizar” |
| 6 | El sistema verifica la información ingresada por el usuario |
| 7 | El sistema edita la información del usuario en la base de datos |
| 8 | El sistema notifica al usuario sobre la actualización exitosa |
| Flujo alternativo | Paso | Acción |
| 4.1 | En caso de que el usuario haga click en el botón “Cancelar”, el sistema limpiará todos los campos y recolocará la informacion original |
| 6.1 | En caso de que algún campo tenga informacion incorrecta o incompatible, el sistema mostrará un mensaje notificando el error |
| 6.2 | En caso de que algún campo obligatorio este vacío, el sistema notificará al usuario sobre los campos obligatorios son requeridos |
| 7.1 | En caso de que exista algún problema al actualizar la información en la base de datos, el sistema notificará al usuario sobre el problema |
| Postcondición | El sistema guarda la nueva información del usuario actual en la base de datos | |

**Figura 4.1** Especificación del caso de uso “Actualizar información”

La figura 4.2 muestra la especificación del caso de uso “Crear tema”

|  |  |  |
| --- | --- | --- |
| CU-0004 | Crear tema | |
| Versión | 1.0 (02/10/2024) | |
| Prioridad | Media | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El usuario creará temas los cuales podrán ser editados posteriormente y asociar actividades a los mismos | |
| Precondición | El usuario deberá haber iniciado sesión como “Administrador” | |
| Flujo normal | Paso | Acción |
| 1 | El usuario dará click en el botón “Crear tema” |
| 2 | El sistema mostrará en pantalla un formulario para crear el tema, el formulario requerirá información básica del tema como un nombre único |
| 3 | El usuario llenará los campos del formulario con la información requerida |
| 4 | El usuario dará click en el botón “Guardar” |
| 5 | El sistema validará la información ingresada en el formulario |
| 6 | El sistema guardará la información ingresada del formulario en la base de datos |
| 7 | El sistema notificará al usuario sobre el registro exitoso del nuevo tema |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que el usuario de click en el botón “Atrás”, el sistema saldrá de la vista actual y volverá a la vista anterior |
| 3.2 | En caso de que el usuario de click en el botón “Limpiar”, el sistema limpiará los campos del formulario |
| 5.1 | En caso de que los datos ingresados sean incorrectos o incompatibles, el sistema notificará al usuario del problema |
| 5.2 | En caso de que el nombre del tema ya exista en la base de datos, el sistema no guardará la informacion y notificará al usuario que el nombre del tema debe ser diferente al nombre ya existente |
| Postcondición | El sistema registrará y guardará el nuevo tema en la base de datos | |

**Figura 4.2** Especificación del caso de uso “Crear tema”

La figura 4.3 muestra la especificación del caso de uso “Actualizar tema”

|  |  |  |
| --- | --- | --- |
| CU-0006 | Actualizar tema | |
| Versión | 1.0 (03/10/2024) | |
| Prioridad | Baja | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El usuario podrá actualizar la informacion de los temas, permitiendo que el sistema guarde la información nueva | |
| Precondición | El usuario deberá de haber iniciado sesión y seleccionado un tema | |
| Flujo normal | Paso | Acción |
| 1 | El usuario hará click en la sección “Actualizar tema” |
| 2 | El sistema cargará la vista de “Actualizar tema” |
| 3 | El sistema consultará la información en la base de datos del tema actual y la mostrará al usuario en un formulario |
| 4 | El usuario modifica la información que desee actualizar |
| 5 | El usuario dará click en el botón “Actualizar” |
| 6 | El sistema verifica la información ingresada por el usuario |
| 7 | El sistema edita la información del tema en la base de datos |
| 8 | El sistema notifica al usuario sobre la actualización exitosa |
| Flujo alternativo | Paso | Acción |
| 4.1 | En caso de que el usuario haga click en el botón “Cancelar”, el sistema limpiará todos los campos y recolocará la informacion original |
| 6.1 | En caso de que algún campo tenga informacion incorrecta o incompatible, el sistema mostrará un mensaje notificando el error |
| 6.2 | En caso de que algún campo obligatorio este vacío, el sistema notificará al usuario sobre los campos obligatorios son requeridos |
| 7.1 | En caso de que exista algún problema al actualizar la información en la base de datos, el sistema notificará al usuario sobre el problema |
| Postcondición | El sistema guarda la nueva información del tema actual en la base de datos | |

**Figura 4.3** Especificación del caso de uso “Actualizar tema”

La figura 4.4 muestra la especificación del caso de uso “Eliminar tema”

|  |  |  |
| --- | --- | --- |
| CU-0007 | Eliminar tema | |
| Versión | 1.0 (03/10/2024) | |
| Prioridad | Baja | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El sistema elimina el registro existente del tema seleccionado por el usuario | |
| Precondición | El usuario deberá de haber iniciado sesión y seleccionado un tema | |
| Flujo normal | Paso | Acción |
| 1 | El usuario dará click en el botón “Eliminar tema” |
| 2 | El sistema mostrará un mensaje de confirmación al usuario |
| 3 | El usuario confirmará el mensaje de eliminación |
| 4 | El sistema elimina el registro del tema en la base de datos |
| 5 | El sistema muestra un mensaje de eliminación exitosa |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que el usuario cancele el mensaje, el sistema cancelará la operación también |
| 4.1 | En caso de que exista algún problema con la base de datos, el sistema notificará al usuario del problema |
| Postcondición | El sistema elimina el registro del tema seleccionado de la base de datos | |

**Figura 4.4** Especificación del caso de uso “Eliminar tema”

La figura 4.5 muestra la especificación del caso de uso “Crear test”

|  |  |  |
| --- | --- | --- |
| CU-0008 | Crear test | |
| Versión | 1.0 (02/10/2024) | |
| Prioridad | Media | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El usuario creará exámenes diagnósticos los cuales podrán ser editados y asociarse a algún tema posteriormente. | |
| Precondición | El usuario deberá haber iniciado sesión como “Administrador” | |
| Flujo normal | Paso | Acción |
| 1 | El usuario dará click en el botón “Crear examen” |
| 2 | El sistema mostrará en pantalla un formulario para crear el examen diagnóstico, el formulario requerirá información básica del examen como un número de preguntas y los niveles de conocimiento que pueda devolver |
| 3 | El usuario llenará los campos del formulario con la información requerida |
| 4 | El usuario dará click en el botón “Guardar” |
| 5 | El sistema validará la información ingresada en el formulario |
| 6 | El sistema guardará la información ingresada del formulario en la base de datos |
| 7 | El sistema notificará al usuario sobre el registro exitoso del nuevo examen diagnóstico |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que el usuario de click en el botón “Atrás”, el sistema saldrá de la vista actual y volverá a la vista anterior |
| 3.2 | En caso de que el usuario de click en el botón “Limpiar”, el sistema limpiará los campos del formulario |
| 5.1 | En caso de que los datos ingresados sean incorrectos o incompatibles, el sistema notificará al usuario del problema |
| Postcondición | El sistema registrará y guardará el nuevo examen diagnóstico en la base de datos | |

**Figura 4.5** Especificación del caso de uso “Crear test”

La figura 4.6 muestra la especificación del caso de uso “Actualizar test”

|  |  |  |
| --- | --- | --- |
| CU-0010 | Actualizar test | |
| Versión | 1.0 (03/10/2024) | |
| Prioridad | Baja | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El usuario podrá actualizar la informacion de los exámenes diagnósticos, permitiendo que el sistema guarde la información nueva | |
| Precondición | El usuario deberá de haber seleccionado el examen diagnóstico de algún tema | |
| Flujo normal | Paso | Acción |
| 1 | El usuario hará click en la sección “Actualizar examen” |
| 2 | El sistema cargará la vista de “Actualizar examen” |
| 3 | El sistema consultará la información en la base de datos del tema actual y la mostrará al usuario en un formulario |
| 4 | El usuario modifica la información que desee actualizar |
| 5 | El usuario dará click en el botón “Actualizar” |
| 6 | El sistema verifica la información ingresada por el usuario |
| 7 | El sistema edita la información del examen diagnóstico en la base de datos |
| 8 | El sistema notifica al usuario sobre la actualización exitosa |
| Flujo alternativo | Paso | Acción |
| 4.1 | En caso de que el usuario haga click en el botón “Cancelar”, el sistema limpiará todos los campos y recolocará la informacion original |
| 6.1 | En caso de que algún campo tenga informacion incorrecta o incompatible, el sistema mostrará un mensaje notificando el error |
| 6.2 | En caso de que algún campo obligatorio este vacío, el sistema notificará al usuario sobre los campos obligatorios son requeridos |
| 7.1 | En caso de que exista algún problema al actualizar la información en la base de datos, el sistema notificará al usuario sobre el problema |
| Postcondición | El sistema guarda la nueva información del examen diagnóstico actual en la base de datos | |

**Figura 4.6** Especificación del caso de uso “Actualizar test”

La figura 4.7 muestra la especificación del caso de uso “Eliminar test”

|  |  |  |
| --- | --- | --- |
| CU-0011 | Eliminar test | |
| Versión | 1.0 (03/10/2024) | |
| Prioridad | Baja | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El sistema elimina el registro existente del examen diagnostico seleccionado por el usuario | |
| Precondición | El usuario deberá de haber seleccionado el examen diagnóstico de algún tema | |
| Flujo normal | Paso | Acción |
| 1 | El usuario dará click en el botón “Eliminar examen” |
| 2 | El sistema mostrará un mensaje de confirmación al usuario |
| 3 | El usuario confirmará el mensaje de eliminación |
| 4 | El sistema elimina el registro del tema en la base de datos |
| 5 | El sistema muestra un mensaje de eliminación exitosa |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que el usuario cancele el mensaje, el sistema cancelará la operación también |
| 4.1 | En caso de que exista algún problema con la base de datos, el sistema notificará al usuario del problema |
| Postcondición | El sistema elimina el registro del examen diagnóstico seleccionado de la base de datos | |

**Figura 4.7** Especificación del caso de uso “Eliminar test”

La figura 4.8 muestra la especificación del caso de uso “Crear pregunta”

|  |  |  |
| --- | --- | --- |
| CU-0012 | Crear pregunta | |
| Versión | 1.0 (02/10/2024) | |
| Prioridad | Media | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El usuario creará preguntas los cuales podrán ser editados y asociarse a alguna actividad o examen posteriormente. | |
| Precondición | El usuario deberá haber iniciado sesión como “Administrador” | |
| Flujo normal | Paso | Acción |
| 1 | El usuario dará click en el botón “Crear pregunta” |
| 2 | El sistema mostrará en pantalla un formulario para crear la pregunta, el formulario requerirá información básica de la pregunta como sentencia de la misma |
| 3 | El usuario llenará los campos del formulario con la información requerida |
| 4 | El usuario registrará la información de cada una de las posibles respuestas necesarias |
| 5 | El usuario dará click en el botón “Guardar” |
| 6 | El sistema validará la información ingresada en el formulario |
| 7 | El sistema guardará la información ingresada del formulario en la base de datos |
| 8 | El sistema notificará al usuario sobre el registro exitoso de la nueva pregunta |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que el usuario de click en el botón “Atrás”, el sistema saldrá de la vista actual y volverá a la vista anterior |
| 3.2 | En caso de que el usuario de click en el botón “Limpiar”, el sistema limpiará los campos del formulario |
| 6.1 | En caso de que los datos ingresados sean incorrectos o incompatibles, el sistema notificará al usuario del problema |
| 6.2 | En caso de que el número de respuestas sea menor a 2, el sistema notificará al usuario que las preguntas deben ser 2 o más |
| 6.3 | En caso de que el puntaje máximo de las respuestas sea menor a 1, el sistema notificará al usuario de que al menos una respuesta debe de tener un puntaje de 1 o superior |
| 6.4 | En caso de que el puntaje mínimo de las preguntas sea igual al puntaje máximo o menor a 0, el sistema notificara el usuario de que el puntaje mínimo de las respuestas debe ser 0 o superior y menor al puntaje máximo |
| Postcondición | El sistema registrará y guardará la nueva pregunta en la base de datos | |

**Figura 4.8** Especificación del caso de uso “Crear pregunta”

La figura 4.9 muestra la especificación del caso de uso “Consultar pregunta”

|  |  |  |
| --- | --- | --- |
| CU-0013 | Consultar pregunta | |
| Versión | 1.0 (02/10/2024) | |
| Prioridad | Media | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El sistema consultará la información del de acuerdo a las peticiones del usuario | |
| Precondición | El usuario deberá haber iniciado sesión como “Administrador” | |
| Flujo normal | Paso | Acción |
| 1 | El usuario da click en la sección “Buscar” |
| 2 | El sistema mostrará una vista con 2 secciones, una donde se mostrarán los exámenes diagnósticos y otra donde se mostrarán las preguntas asociadas al examen seleccionado |
| 3 | El sistema consultará el nombre y tema asociado de todos los exámenes diagnósticos registrados en la base de datos y los colocará en pantalla |
| 4 | El usuario seleccionará el examen diagnóstico que desee |
| 5 | El sistema consultará todas las preguntas del examen diagnostico seleccionado |
| 6 | El sistema obtiene las preguntas del examen diagnostico consultado |
| 7 | El sistema muestra la informacion obtenida en la pantalla |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que exista algún problema durante la consulta, el sistema notificará al usuario del problema |
| 5.1 | En caso de que exista algún problema durante la consulta, el sistema notificará al usuario del problema |
| Postcondición | El sistema muestra todas las preguntas asociadas al examen diagnostico seleccionado para su posterior edición en caso de ser necesaria | |

**Figura 4.9** Especificación del caso de uso “Consultar pregunta”

La figura 4.10 muestra la especificación del caso de uso “Actualizar pregunta”

|  |  |  |
| --- | --- | --- |
| CU-0014 | Actualizar pregunta | |
| Versión | 1.0 (03/10/2024) | |
| Prioridad | Baja | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El usuario podrá actualizar la informacion de las preguntas de los exámenes diagnósticos, permitiendo que el sistema guarde la información nueva | |
| Precondición | El usuario deberá de seleccionado haber alguna pregunta de un examen diagnóstico | |
| Flujo normal | Paso | Acción |
| 1 | El usuario hará click en la sección “Actualizar pregunta” |
| 2 | El sistema cargará la vista de “Actualizar pregunta” |
| 3 | El sistema consultará la información en la base de datos de la pregunta actual y la mostrará al usuario en un formulario |
| 4 | El usuario modifica la información que desee actualizar |
| 5 | El usuario dará click en el botón “Actualizar” |
| 6 | El sistema verifica la información ingresada por el usuario |
| 7 | El sistema edita la información de la pregunta en la base de datos |
| 8 | El sistema notifica al usuario sobre la actualización exitosa |
| Flujo alternativo | Paso | Acción |
| 4.1 | En caso de que el usuario haga click en el botón “Cancelar”, el sistema limpiará todos los campos y recolocará la informacion original |
| 6.1 | En caso de que algún campo tenga informacion incorrecta o incompatible, el sistema mostrará un mensaje notificando el error |
| 6.2 | En caso de que algún campo obligatorio este vacío, el sistema notificará al usuario sobre los campos obligatorios son requeridos |
| 7.1 | En caso de que exista algún problema al actualizar la información en la base de datos, el sistema notificará al usuario sobre el problema |
| Postcondición | El sistema guarda la nueva información de la pregunta actual en la base de datos | |

**Figura 4.10** Especificación del caso de uso “Actualizar pregunta”

La figura 4.11 muestra la especificación del caso de uso “Eliminar pregunta”

|  |  |  |
| --- | --- | --- |
| CU-0015 | Eliminar pregunta | |
| Versión | 1.0 (03/10/2024) | |
| Prioridad | Baja | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El sistema elimina el registro existente de la pregunta del examen diagnostico seleccionada por el usuario | |
| Precondición | El usuario deberá de seleccionado haber alguna pregunta de un examen diagnóstico | |
| Flujo normal | Paso | Acción |
| 1 | El usuario dará click en el botón “Eliminar pregunta” |
| 2 | El sistema mostrará un mensaje de confirmación al usuario |
| 3 | El usuario confirmará el mensaje de eliminación |
| 4 | El sistema elimina el registro del tema en la base de datos |
| 5 | El sistema muestra un mensaje de eliminación exitosa |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que el usuario cancele el mensaje, el sistema cancelará la operación también |
| 4.1 | En caso de que exista algún problema con la base de datos, el sistema notificará al usuario del problema |
| Postcondición | El sistema elimina el registro de la pregunta seleccionada de la base de datos | |

**Figura 4.11** Especificación del caso de uso “Eliminar pregunta”

La figura 4.12 muestra la especificación del caso de uso “Crear actividad”

|  |  |  |
| --- | --- | --- |
| CU-0017 | Crear actividad | |
| Versión | 1.0 (02/10/2024) | |
| Prioridad | Media | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El usuario creará actividades los cuales podrán ser editados y asociarse a algún tema posteriormente. | |
| Precondición | El usuario deberá haber iniciado sesión como “Administrador” | |
| Flujo normal | Paso | Acción |
| 1 | El usuario hará click en el botón “Crear actividad” |
| 2 | El sistema mostrará en pantalla un formulario para crear la actividad, el formulario requerirá información de la actividad como un título |
| 3 | El usuario llenará los campos del formulario con la información requerida |
| 4 | El usuario hará click en el botón “Guardar” |
| 5 | El sistema validará la información ingresada en el formulario |
| 6 | El sistema guardará la información ingresada del formulario en la base de datos |
| 7 | El sistema notificará al usuario sobre el registro exitoso de la nueva actividad |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que el usuario haga click en el botón “Atrás”, el sistema saldrá de la vista actual y volverá a la vista anterior |
| 3.2 | En caso de que el usuario haga click en el botón “Limpiar”, el sistema limpiará los campos del formulario |
| 5.1 | En caso de que los datos ingresados sean incorrectos o incompatibles, el sistema notificará al usuario del problema |
| Postcondición | El sistema registrará y guardará la actividad en la base de datos | |

**Figura 4.12** Especificación del caso de uso “Crear actividad”

La figura 4.13 muestra la especificación del caso de uso “Consultar actividad”

|  |  |  |
| --- | --- | --- |
| CU-0018 | Consultar actividad | |
| Versión | 1.0 (30/09/2024) | |
| Prioridad | Alta | |
| Autor/es | Joel González Cruz | |
| Actor/es | Estudiante, Administrador | |
| Dependencias | Ninguna | |
| Descripción | El sistema consultará la informacion de la actividad de acuerdo a las peticiones del usuario | |
| Precondición | El usuario deberá de seleccionar un tema e ir a la sección de contenidos | |
| Flujo normal | Paso | Acción |
| 1 | El sistema consultará informacion breve sobre las actividades asociadas al tema actual, y los mostrará en pantalla |
| 2 | El usuario dará click en la actividad que desee |
| 3 | El sistema consultará toda la información de la actividad seleccionada por el usuario |
| 4 | El sistema mostrará en pantalla los resultados obtenidos por la consulta |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que la actividad carezca de informacion, el sistema notificará al usuario del error e interrumpirá el flujo |
| Postcondición | El sistema muestra la informacion asociada a la actividad que para que el usuario pueda visualizarla | |

**Figura 4.13** Especificación del caso de uso “Consultar actividad”

La figura 4.14 muestra la especificación del caso de uso “”

|  |  |  |
| --- | --- | --- |
| CU-0019 | Actualizar actividad | |
| Versión | 1.0 (03/10/2024) | |
| Prioridad | Baja | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El usuario podrá actualizar la informacion de las actividades de los temas, permitiendo que el sistema guarde la información nueva | |
| Precondición | El usuario deberá de seleccionado haber alguna actividad de un tema | |
| Flujo normal | Paso | Acción |
| 1 | El usuario hará click en la sección “Actualizar actividad” |
| 2 | El sistema cargará la vista de “Actualizar actividad” |
| 3 | El sistema consultará la información en la base de datos de la actividad actual y la mostrará al usuario en un formulario |
| 4 | El usuario modifica la información que desee actualizar |
| 5 | El usuario dará click en el botón “Actualizar” |
| 6 | El sistema verifica la información ingresada por el usuario |
| 7 | El sistema edita la información de la actividad en la base de datos |
| 8 | El sistema notifica al usuario sobre la actualización exitosa |
| Flujo alternativo | Paso | Acción |
| 4.1 | En caso de que el usuario haga click en el botón “Cancelar”, el sistema limpiará todos los campos y recolocará la informacion original |
| 6.1 | En caso de que algún campo tenga informacion incorrecta o incompatible, el sistema mostrará un mensaje notificando el error |
| 6.2 | En caso de que algún campo obligatorio este vacío, el sistema notificará al usuario sobre los campos obligatorios son requeridos |
| 7.1 | En caso de que exista algún problema al actualizar la información en la base de datos, el sistema notificará al usuario sobre el problema |
| Postcondición | El sistema guarda la nueva información de la actividad actual en la base de datos | |

**Figura 4.14** Especificación del caso de uso “Actualizar actividad”

La figura 4.15 muestra la especificación del caso de uso “Eliminar actividad”

|  |  |  |
| --- | --- | --- |
| CU-0020 | Eliminar actividad | |
| Versión | 1.0 (03/10/2024) | |
| Prioridad | Baja | |
| Autor/es | Joel González Cruz | |
| Actor/es | Administrador | |
| Dependencias | Ninguna | |
| Descripción | El sistema elimina el registro existente de la pregunta del examen diagnostico seleccionada por el usuario | |
| Precondición | El usuario deberá de seleccionado haber alguna actividad de un tema | |
| Flujo normal | Paso | Acción |
| 1 | El usuario dará click en el botón “Eliminar actividad” |
| 2 | El sistema mostrará un mensaje de confirmación al usuario |
| 3 | El usuario confirmará el mensaje de eliminación |
| 4 | El sistema elimina el registro del tema en la base de datos |
| 5 | El sistema muestra un mensaje de eliminación exitosa |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que el usuario cancele el mensaje, el sistema cancelará la operación también |
| 4.1 | En caso de que exista algún problema con la base de datos, el sistema notificará al usuario del problema |
| Postcondición | El sistema elimina el registro de la actividad seleccionada de la base de datos | |

**Figura 4.15** Especificación del caso de uso “Eliminar actividad”

La figura 4.16 muestra la especificación del caso de uso “Realizar actividad”

|  |  |  |
| --- | --- | --- |
| CU-0021 | Realizar actividad | |
| Versión | 1.0 (30/09/2024) | |
| Prioridad | Alta | |
| Autor/es | Joel González Cruz | |
| Actor/es | Estudiante | |
| Dependencias | Ninguna | |
| Descripción | El sistema mostrará una serie de preguntas al estudiante para que este las responda, al final mostrará un resultado de actividad | |
| Precondición | El estudiante deberá de haber completado el examen diagnóstico asociado al tema actual e ir a la sección de “Contenidos” | |
| Flujo normal | Paso | Acción |
| 1 | El usuario dará click en el botón “Realizar actividad” |
| 2 | El sistema hará una consulta sobre el nivel de conocimiento del alumno y las actividades asociadas a dicho nivel y del tema actual |
| 3 | El sistema conservará únicamente las actividades que el alumno no haya completado o respondido incorrectamente |
| 4 | El sistema desordenará las preguntas restantes y conservará solo la primera “n” cantidad de actividades |
| 5 | El sistema mostrará la pregunta en curso con su respectiva sentencia, material de apoyo en caso de que lo tenga, y sus posibles respuestas |
| 6 | El usuario seleccionará la respuesta que considere más adecuada |
| 7 | El usuario dará click en el botón “Siguiente” |
| 8 | Mientras existan preguntas posteriores, repetir los pasos 5 al 7 |
| 9 | El usuario dará click en el botón “Terminar” |
| 10 | El sistema mostrará los resultados obtenidos por el usuario |
| Flujo alternativo | Paso | Acción |
| 3.1 | En caso de que el usuario tenga el nivel de conocimiento máximo y todas las actividades completadas, el sistema seleccionará todas las actividades disponibles para el tema actual |
| 4.1 | En caso de que “n” cantidad de preguntas sea menor al total de preguntas, el sistema asignará el valor de la cantidad de preguntas a “n” |
| 7.1 | En caso de que el usuario no haya seleccionado alguna respuesta, el sistema notificará al usuario de que en caso de continuar, la pregunta actual no sumara algún puntaje, el mensaje podrá aceptarse(Saltar pregunta) o rechazarse(Responder pregunta) |
| 7.2 | En caso de que el usuario salte la pregunta, el sistema asignará un puntaje de 0 a la pregunta actual y continuará el flujo en el paso 8 |
| 7.3 | En caso de que el usuario prefiera responder la pregunta, el sistema continuará con flujo normal |
| Postcondición | El sistema muestra los resultados de la actividad en pantalla al usuario | |

**Figura 4.16** Especificación del caso de uso “Realizar actividad”

Documento anexo 2, Modelo matemático de asignación diagnóstica

El prototipo de la aplicación nativa “Desarrollo App Educativa 4.0” tiene la capacidad de incluir módulos de aprendizaje, los cuales se pueden entender como asignaturas destinadas a adquirir y reforzar conocimiento previo a áreas de informática como desarrollo, infraestructura de redes, programación, etc. Para así actuar como una introducción previa, facilitando la comprensión de temas básicos como: abstracción, lógica, cómputo, entre otros.

Estos módulos de aprendizaje cuentan con un examen diagnóstico y su respectiva sección de contenidos. El examen diagnostico posee una cantidad determinada de preguntas las cuales contendrán 4 posibles respuestas, cada respuesta asigna de 0 hasta 3 puntos de acuerdo si la respuesta no es acertada y si la respuesta es muy acertada respectivamente. A continuación, se muestra un ejemplo de pregunta de examen diagnóstico (para este ejemplo las respuestas están ordenadas de correcta a incorrecta).

7. De acuerdo a la sucesión (a, b, e, c, i, ?)   
¿Cuál es la letra que completa correctamente la serie?

A) d *(3 puntos)*

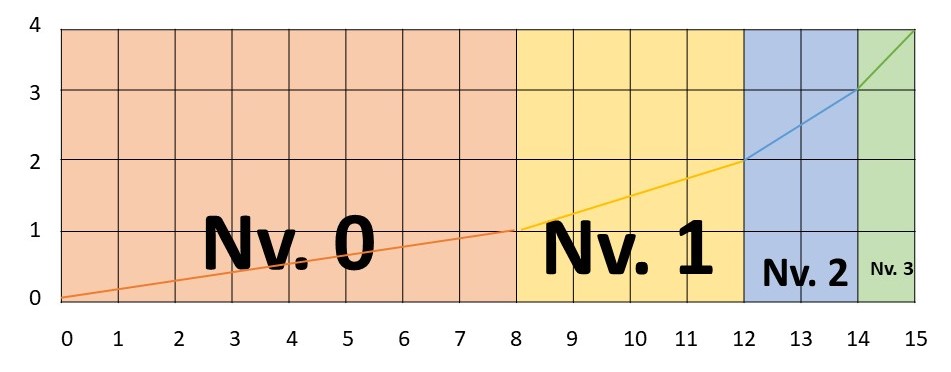
B) o *(2 puntos)*

C) z *(1 punto)*

D) ñ *(0 puntos)*

**Figura 5.1** Ejemplo de pregunta diagnóstica con respuestas ordenadas

Al finalizar el examen diagnóstico, se calcula una serie de proporciones de acuerdo a la cantidad de niveles de conocimiento requeridos por el examen diagnóstico. Estas proporciones facilitan la asignación de niveles de conocimiento. A continuación, se muestra un ejemplo de este comportamiento gráfico tomando 4 niveles como ejemplo.



**Figura 5.2** Ejemplo gráfico de modelo de asignación diagnóstica

Como primer paso, se calcula el multiplicador de rango por cada uno de los niveles que se hayan planteado previamente (la cantidad de rangos es igual a la cantidad de niveles planteados disminuido en 1)

Rx =

Donde:

* n = Cantidad de niveles
* x = Número de rango ()
* Rx = Multiplicador de rango “x”

A continuación, se muestra un ejemplo con: n = 4

R0

R1

R2

Como segundo paso se necesita calcular los segmentos de puntaje para la asignación de niveles. Estos segmentos se calculan como el producto de la puntuación máxima posible del examen con los multiplicadores de rango.

Sx = Rx(3(q))

Donde:

* x = Número de segmento ()
* Sx = Segmento “x”
* Rx = Multiplicador de rango “x”
* q = Cantidad de preguntas aplicadas en el examen diagnóstico

A continuación, se da un ejemplo de cada uno de los segmentos a calcular usando como multiplicador de rango el ejemplo anterior y una cantidad de preguntas: q = 50

NOTA: en caso de que los resultados sean números racionales, estos se redondearán al entero más cercano.

S0

S1

S2

Una vez calculados los segmentos de puntaje se debe comparar la puntuación obtenida por el estudiante con el primer segmento, si este es mayor se le asignará el nivel n -1, en caso contrario si la puntuación es menor o igual segmento anterior y mayor al siguiente segmento, se le asignará el nivel n -2. Repetir este proceso hasta comprobar hasta llegar al último caso que donde se interpretará que la puntuación es menor al último segmento. En caso cualquier afirmación sea correcta, se obtiene el resultado y se detiene el proceso.

(S0 < Po)Nv. (n-1)

(S1 < Po ≤ S0)Nv. (n-2)

(S2 < Po ≤ S1)Nv. (n-3)

**. . .**

(S(n-2) < Po ≤ S(n-3))Nv. 1

(Po ≤ S(n-2))Nv. 0

Donde:

* n = Cantidad de niveles
* Po = Puntaje obtenido por el estudiante

Por último, se muestra un ejemplo del proceso de comparación utilizando los ejemplos anteriores con una puntuación de obtenida de: Po = 119

(140 < 119)Nv. 3

(120 < 119 ≤ 140)Nv. 2

(80 < 119 ≤ 120)Nv. 1

(119 ≤ 80)Nv. 0

Como se pude observar se realizaron 4 afirmaciones en donde la 3ra sentencia es correcta, haciendo que en el caso de que cálculo se procese con 4 niveles, 150 puntos de puntuación máxima y una puntuación obtenida de 119, dé como resultado una respuesta de 1.

Documento anexo 3, Examen diagnóstico de pensamiento abstracto

[Documento completo]

# REFERENCIAS

Alonso de Castro, M. G. (2014). Educational projects based on mobile learning. Revista Teoría de la Educación: Educación y Cultura en la Sociedad de la Información. 15(1), pp. 10-19.

Amavizca, S. y Alvarez-Flores, E. P. (2022). Comprensión lectora en universitarios: comparativo por áreas de conocimiento. Revista Electrónica de Investigación Educativa, 24, e20, 1-13. <https://doi.org/10.24320/redie.2022.24.e20.3986>

Arellano Pimentel, J. J., Solar González, R. y Armería Zavala, L. (2024). Estrategias y recursos didácticos utilizados para aprender programación estructurada. Una revisión sistemática. IE Revista de Investigación Educativa de la REDIECH, 15, e1872. <https://doi.org/10.33010/ie_rie_rediech.v15i0.1872>

Awouda, A., Traini, E., Asranov, M. (2024). Bloom’s IoT Taxonomy towards an effective Industry 4.0 education: Case study on Open-source IoT laboratory. Educ Inf Technol. <https://doi.org/10.1007/s10639-024-12468-7>

Aznar Díaz, I., Cáceres Reche, M. P., y Romero Rodríguez, J. M. (2018). Indicadores de calidad para evaluar buenas prácticas docentes de «mobile learning» en Educación Superior. Education in the Knowledge Society, 19(3), pp. 53-68. <https://doi.org/10.14201/eks20181935368>

Blasco, F. (2021). El cultivo de la inteligencia a través del lenguaje matemático | Cultivating intelligence through mathematical language. Revista Española de Pedagogía, 79(278), pp. 59-75. doi: <https://doi.org/10.22550/REP79-1-2021-07>

Cáceres-Serrano, P. A., & Conejeros Solar, M. L. (2011). Efecto de un modelo de metodología centrada en el aprendizaje sobre el pensamiento crítico, el pensamiento creativo y la capacidad de resolución de problemas en estudiantes con talento académico. Revista Española de Pedagogía, 69(248).

Cantú-Cervantes, D., Amaya-Amaya, A., & Baca-Pumarejo, J. R. (2019). Modelo para el reforzamiento del aprendizaje con dispositivos móviles. CienciaUAT, 13(2), pp. 56-70. <https://doi.org/10.29059/cienciauat.v13i2.1161>

Casany Guerrero, M. J. y Barceló García, M. (2013). Como definir proyectos de m-learning más sostenibles. Revista Teoría de la Educación: Educación y Cultura en la Sociedad de la Información. 14(2), pp. 271-291.

Castro-Benavides, L. M., Tamayo-Arias, J. A., y Burgos, D. (2022). Escenarios de la docencia frente a la transformación digital de las Instituciones de Educación Superior. Education in the Knowledge Society, 23(2022), pp. 1-17. <https://doi.org/10.14201/eks.27569>

De los Santos Lorenzo, M. (2021). Evaluación de competencias informacionales en estudiantes universitarios de la República Dominicana. Education in the Knowledge Society, 22(2021), pp. 1-13. <https://doi.org/10.14201/eks.23650>

Escalante Ferrer, A. E., Coronado Fernández, S. E. y Moctezuma Ramírez, E. E. (2023). La dimensión metacognitiva de la competencia aprender a aprender en titulaciones españolas. Sinéctica, Revista Electrónica de Educación, (60), e1457. <https://doi.org/10.31391/S2007-7033(2023)0060-004>

Escudero, C. (2009). Una mirada alternativa acerca del residuo cognitivo cuando se introducen nuevas tecnologías. El caso de la resolución de problemas en ciencias. Revista Electrónica Teoría de la Educación: Educación y Cultura en la Sociedad de la Información, 10(1), pp. 272-292.

Fernandes Gomes, N., & Hernández Serrano, M. J. (2014). Tecnologias e modelos de aprendizagem emergentes no ensino superior. Propostas e aplicações de inovações. Teoría de la Educación. Educación y Cultura en la Sociedad de la Información, 15(4), pp. 134-159.

Ferrer-Rojas, A. (2017). VirPLC: una metodología para el desarrollo de capacidades, habilidades y autoestima mediante la estimulación de la lógica con una herramienta sencilla, funcional y dinámica. Education in the Knowledge Society, 18(2), pp. 59-69. <https://doi.org/10.14201/eks20171825969>

Florez-Florez, R. (1982). Educación del pensamiento crítico. Revista Española de Pedagogía, 40(158).

Forero-Corba, W., & Negre Bennasar, F. (2024). Techniques and applications of Machine Learning and Artificial Intelligence in education: a systematic review. [Técnicas y aplicaciones del Machine Learning e Inteligencia Artificial en educación: una revisión sistemática]. RIED-Revista Iberoamericana de Educación a Distancia, 27(1), pp. 209-253. <https://doi.org/10.5944/ried.27.1.37491>

Fornons, V., y Palau, R. (2021). Flipped Classroom en la enseñanza de las Matemáticas: una revisión sistemática. Education in the Knowledge Society, 22(2021), pp. 1-20. <https://doi.org/10.14201/eks.24409>

Fuentes, J. L., Albertos, J. E., y Torrano, F. (2019). Hacia el Mobile-Learning en la escuela: análisis de factores críticos en el uso de las tablets en centros educativos españoles. Education in the Knowledge Society, 20(2019), pp. 1-17. <https://doi.org/10.14201/eks2019_20_a3>

García-Peñalvo, F. J. (2024). Inteligencia artificial generativa y educación: Un análisis desde múltiples perspectivas. Education in the Knowledge Society, 25(2024), pp. 1-10. <https://doi.org/10.14201/eks.29407>

García-Santiago, L., Díaz-Millón, M. (2024). Pedagogical and communicative resilience before industry 4.0 in higher education in translation and interpreting in the twenty-first century. Educ Inf Technol. <https://doi.org/10.1007/s10639-024-12751-7>

Gilbert Delgado, R. P., Naranjo Vaca, G. E., & Gorina Sánchez, A. (2023). Comprensión textual en la resolución de problemas matemáticos. Acta Universitaria 33, e3809. doi: <http://doi.org/10.15174.au.2023.3809>

González-Martínez, J., Estebanell Minguell, M., y Peracaula Bosch, M. (2018). ¿Robots o programación? El concepto de Pensamiento Computacional y los futuros maestros. Education in the Knowledge Society, 19(2), pp. 29-45. <https://doi.org/10.14201/eks20181922945>

González-Pérez, L. I., Ramírez-Montoya, M. S., y García-Peñalvo, F. J. (2022). Habilitadores tecnológicos 4.0 para impulsar la educación abierta: aportaciones para las recomendaciones de la UNESCO. RIED-Revista Iberoamericana de Educación a Distancia, 25(2), pp. 23-48. <https://doi.org/10.5944/ried.25.2.33088>

Latorre-Cosculluela, C., Vázquez-Toledo, S., Rodríguez-Martínez, A. y Liesa-Orús, M. (2020). Design Thinking: creatividad y pensamiento crítico en la universidad. Revista Electrónica de Investigación Educativa, 22, e28, 1-13. <https://doi.org/10.24320/redie.2020.22.e28.2917>

Llanos-Ruiz, D. Ausín-Villaverde, V., y Abella García, V. (2023). Percepción de alumnos y familias sobre la robótica educativa en la educación no formal. Education in the Knowledge Society, 24(2023), pp. 1-12. <https://doi.org/10.14201/eks.31351>

Llorens-Largo, F., García-Peñalvo, F. J., Molero Prieto, X., y Vendrell Vidal, E. (2017). La enseñanza de la informática, la programación y el pensamiento computacional en los estudios preuniversitarios. Education in the Knowledge Society,18(2), pp. 7-17. <https://doi.org/10.14201/eks2017182717>

Montes-León, H., Hijón-Neira, R., Pérez-Marín, D., y Montes-León S. R. (2020). Mejora del Pensamiento Computacional en Estudiantes de Secundaria con Tareas Unplugged. Education in the Knowledge Society, 21(2020), pp. 1-12. <https://doi.org/10.14201/eks.23002>

Navas López, E. A. (2024). Relaciones entre la matemática, el pensamiento algorítmico y el pensamiento computacional. IE Revista de Investigación Educativa de la REDIECH, 15, e1929. <https://doi.org/10.33010/ie_rie_rediech.v15i0.1929>

Patiño, A., Ramírez-Montoya, M.S. & Buenestado-Fernández, M. (2023). Active learning and education 4.0 for complex thinking training: analysis of two case studies in open education. Smart Learn. Environ. 10, 8(2023). <https://doi.org/10.1186/s40561-023-00229-x>

Quitério-Figueiredo, J. A. (2017). Cómo mejorar el pensamiento computacional: un estudio de caso. Education in the Knowledge Society, 18(4), pp. 35-51. <https://doi.org/10.14201/eks20171843551>

Ramírez-Montoya, M. S., McGreal, R., y Obiageli Agbu, J.-F. (2022). Horizontes digitales complejos en el futuro de la educación 4.0: luces desde las recomendaciones de UNESCO. RIED-Revista Iberoamericana de Educación a Distancia, 25(2), pp. 09-21. <https://doi.org/10.5944/ried.25.2.33843>

Reynders, G., Lantz, J., Ruder, S.M. (2020). Rubrics to assess critical thinking and information processing in undergraduate STEM courses. IJ STEM Ed 7, 9. <https://doi.org/10.1186/s40594-020-00208-5>

Roll, M., Ifenthaler, D. (2021). Learning Factories 4.0 in technical vocational schools: can they foster competence development?. Empirical Res Voc Ed Train 13, 20(2021). <https://doi.org/10.1186/s40461-021-00124-0>

Sánchez Prieto, J. C., Olmos Migueláñez, S. y García-Peñalvo, F. J. (2014). Understanding mobile learning: devices, pedagogical implications and research lines. Revista Teoría de la Educación: Educación y Cultura en la Sociedad de la Información. 15(1), pp. 20-42.

Segredo, E., Miranda, G., y León, C. (2017). Hacia la educación del futuro: El pensamiento computacional como mecanismo de aprendizaje generativo. Education in the Knowledge Society, 18(2), pp. 33-58. <https://doi.org/10.14201/eks20171823358>

Sein-Echaluce, M.L., Fidalgo-Blanco, Á., Balbín, A.M. (2024). Flipped Learning 4.0. An extended flipped classroom model with Education 4.0 and organisational learning processes. Univ Access Inf Soc 23, pp. 1001–1013. <https://doi.org/10.1007/s10209-022-00945-0>

Suárez Gómez, R:, Crescenzi Lanna, L. y Grané i Oro, M. (2013). Análisis del entorno colaborativo creado para una experiencia de mobile learning. Revista Teoría de la Educación: Educación y Cultura en la Sociedad de la Información. 14(1), pp. 101-121.

Vera, J., Villalba-Condori, K., y Cuba-Sayco, S. C. (2018). Modelo de sistema de recomendación basado en el contexto a partir del análisis de código estático para el desarrollo del Pensamiento Computacional: Caso de Programación Web. Education in the Knowledge Society, 19(2), pp. 103-126. <https://doi.org/10.14201/eks2018192103126>

Wei, H., Bos, R. & Drijvers, P. (2024). Developing Functional Thinking: from Concrete to Abstract Through an Embodied Design. *Digit Exp Math Educ*. <https://doi.org/10.1007/s40751-024-00142-z>

Zapata-Roz, M. (2019). Pensamiento computacional desenchufado. Education in the Knowledge Society, 20(2019), pp. 1-29. <https://doi.org/10.14201/eks2019_20_a18>