



“Sistema tutor afectivo para apoyar el proceso de enseñanza aprendizaje de estructuras de datos”

**INSTITUTO TECNOLÓGICO SUPERIOR DE  
TEZIUTLÁN**

**Tesis**



**ALUMNO:**

Ricardo Alberto Serrano Herrera

**Número de Control:**

19TE0012P

**Maestría en**

Sistemas Computacionales

**ASESORA:**

Cristina Joaquín Salas

Teziutlán, Puebla; mayo 2021

“La Juventud de hoy, Tecnología del Mañana”



## **DEDICATORIA**

Dedico este trabajo principalmente a mi novia Cristy, por su amor, cariño, confianza y creer en mí, gracias a su apoyo he alcanzado mis metas, me ha ayudado a enfrentar mis miedos y a crecer como persona y profesionalista. Por estar a mi lado en cada decisión que he tomado, y a pesar de todo siempre está para escucharme y darme consejos cuando los necesito. Este es uno de los muchos proyectos que nos esperan juntos.

A mi madre y mis hermanos que a pesar de nuestras diferencias siempre estaremos unidos, apoyándonos, escuchándonos y queriéndonos como una familia.

Este no solo es mi logro, también es el suyo.

## **AGRADECIMIENTOS**

A mi madre por inspirarme a siempre superarme y no rendirme, por confiar en mí y nunca dudar de mis habilidades, por todas sus enseñanzas, cariño y comprensión.

A mis hermanos por motivarme y siempre estar unidos a pesar de la distancia, en particular a mi hermano Alejandro por su tiempo y confianza para lograr este proyecto y a mi hermano Roberto por estar siempre apoyándome en cada paso de mi vida.

A mi novia Cristy porque sin ella no lo hubiera logrado, por su tiempo, enseñanzas, palabras de apoyo, y saber que siempre puedo confiar en ella y aunque días pensaba que no lo lograría ella siempre me apoyaba a seguir adelante.

A mis maestros por cada enseñanza que me han transmitido desde la licenciatura hasta este momento.

Al Tec de Teziutlán, por darme la oportunidad de realizar mis estudios.

## RESUMEN

La presente investigación presenta el desarrollo de un sistema tutor afectivo, para apoyar el proceso de enseñanza aprendizaje de estructuras de datos. Se define una aplicación móvil como la parte del sistema enfocada a la interacción con el alumno y docente. El desarrollo de estos se llevó a cabo utilizando el framework de Ionic, los servicios de firebase y el api de Google Vision AI. Para la implementación se definió un grupo de estudio de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico Superior de Teziutlán. Como primer paso el grupo realizo un pretest sobre las estructuras de control para identificar las áreas de oportunidad de los estudiantes, posteriormente se aplica la herramienta propuesta. Finalmente, se aplicó un postest en para evaluar la utilidad de la herramienta en la conceptualización de la información en el proceso de enseñanza-aprendizaje.

DEDICATORIA.....	2
AGRADECIMIENTOS .....	3
RESUMEN .....	4
ÍNDICE DE ILUSTRACIONES .....	8
ÍNDICE DE TABLAS .....	9
ÍNDICE DE DIAGRAMAS.....	10
ÍNDICE DE GRÁFICAS.....	11
CAPITULO I. INTRODUCCIÓN.....	12
1.1. PLANTEAMIENTO DEL PROBLEMA.....	13
1.2 OBJETIVO GENERAL .....	14
1.2.1 OBJETIVOS ESPECÍFICOS .....	14
1.3 JUSTIFICACIÓN .....	15
1.4 PREGUNTAS DE INVESTIGACIÓN.....	16
1.5 HIPÓTESIS Y VARIABLES.....	17
1.5.1 Hipótesis de trabajo.....	17
1.5.2 Hipótesis nula .....	17
1.5.3 Variables.....	17
1.6 PROPUESTA TECNOLÓGICA.....	18
1.7 ESTUDIO DE FACTIBILIDAD .....	19
1.7.1 Económica.....	19
1.7.2 Técnica.....	19
1.7.3 Humana .....	19
1.8 ANTECEDENTES .....	20
CAPITULO II. MARCO TEORICO .....	24
2.1 Emociones en el aprendizaje.....	24
2.2 Computación afectiva .....	25
2.3 Reconocimiento facial.....	25
2.4 Sistema tutor inteligente y Afectivo.....	26
2.5 Gamificación .....	27

2.6 Herramientas de diseño .....	27
2.6.1 Balsamiq.....	27
2.6.2 StarUML .....	28
2.7 Herramientas de desarrollo.....	28
2.7.1 Visual Studio Code .....	28
2.7.2 GIT.....	29
2.7.3 Gitlab .....	30
2.7.4 Android Studio.....	30
2.8 Metodología de desarrollo de software Kuali-Beh.....	31
2.9 Tecnologías .....	31
2.9.1 NodeJs .....	31
2.9.2 Angular .....	31
2.9.3 Ionic Framework .....	32
2.9.4 Capacitor .....	33
2.9.5 Rextester .....	33
2.9.6 Google Cloud.....	33
2.9.7 Vision AI .....	34
2.9.8 Firebase.....	34
2.9.9 Firebase Cloud Firestore .....	35
2.9.10 Firebase Authentication .....	35
2.9.11 Firebase Cloud Functions.....	35
2.9.12 Firebase Cloud Storage.....	35
CAPITULO III. METODOLOGIA DE LA INVESTIGACIÓN.....	36
3.1 DISEÑO METODOLÓGICO.....	36
3.1.1 Tipo de investigación .....	36
3.1.2 Enfoque de la investigación.....	37
3.1.3 Alcance y tipo de estudio.....	37
3.1.4 Instrumentos de recolección de datos .....	37
3.2 DISEÑO DE LA INVESTIGACIÓN .....	37
3.3 MUESTRA.....	38
3.4 INSTRUMENTOS DE MEDICIÓN.....	38

3.4.1 Operacionalización de las variables .....	38
3.4.2 Instrumentos de medición.....	39
CAPITULO IV. METODOLOGIA DE DESARROLLO DE SOFTWARE .....	42
4.1 Metodología de desarrollo.....	42
4.2 Especificación de requerimientos.....	43
4.2.1 Diagramas de caso de uso .....	43
4.2.2 Descripción de casos de uso .....	48
4.3 Diseño conceptual.....	56
4.4 Diseño navegacional.....	57
4.5 Diseño de Interfaces .....	57
4.5 Pruebas e implementación.....	62
4.5.1 Pruebas .....	62
4.5.2 Implementación .....	66
CAPITULO V. RESULTADOS Y DISCUSIÓN.....	71
5.1 Resultados Pretest y Postest.....	71
5.2 Análisis de expresiones faciales .....	78
CONCLUSIONES.....	81
REFERENCIAS.....	82

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 Inicio se sesión.....	58
Ilustración 2 Vista de actividades (Alumno).....	58
Ilustración 3 Vista de perfil (Alumno).....	58
Ilustración 4 Editar perfil (Alumno) .....	58
Ilustración 5 Vista de ejercicios de programación (Alumno).....	59
Ilustración 6 Vista de resolver ejercicios (Alumno) .....	59
Ilustración 7 Vista de resolver ejercicios mostrando una Alerta (Alumno).....	59
Ilustración 8 Vista principal (Docente) .....	59
Ilustración 9 Vista de Administrar alumnos (Docente).....	60
Ilustración 10 Vista de consultar resultados (Docente)vez .....	60
Ilustración 11 Vista de consultar datos individuales (Docente).....	60
Ilustración 12 Vista de actividades (Docente).....	60
Ilustración 13 Vista principal (Docente) .....	61
Ilustración 14 Vista de descarga de metadatos (docente).....	61
Ilustración 15 Página de logueo y principal .....	66
Ilustración 16 Página principal y configuración.....	67
Ilustración 17 Pantalla de Pretest.....	68
Ilustración 18 Pregunta Pretest .....	68
Ilustración 19 Pantalla de actividades Algoritmos .....	69
Ilustración 20 Editor de código en actividad de algoritmos.....	69
Ilustración 21 Pantalla de retroalimentación.....	70

## ÍNDICE DE TABLAS

Tabla 1 Operacionalización de las variables de investigación.....	38
Tabla 2 Cronograma de etapas de desarrollo .....	43
Tabla 3 Descripción de caso de uso autenticación de usuario.....	48
Tabla 4 Descripción de caso de uso realización pretest .....	49
Tabla 5 Descripción de caso de uso realización posttest.....	50
Tabla 6 Descripción de caso de uso análisis de expresiones faciales .....	51
Tabla 7 Descripción de caso de uso realizar actividad.....	52
Tabla 8 Descripción de caso de uso consultar ayuda escrita.....	53
Tabla 9 Descripción de caso de uso consultar ayuda audiovisual .....	54
Tabla 10 Descripción de caso de uso activar o desactivar actividad.....	55
Tabla 11 Prueba unitaria Inicio de sesión.....	62
Tabla 12 Prueba unitaria Pretest.....	63
Tabla 13 Prueba unitaria posttest.....	64
Tabla 14 Prueba unitaria ejercicio .....	65
Tabla 15 Base de datos de emociones .....	78

## ÍNDICE DE DIAGRAMAS

Diagrama 1 Etapas de desarrollo en el marco de trabajo Quali-Beh .....	42
Diagrama 2 Diagrama de caso de uso Actores .....	44
Diagrama 3 Diagrama general de caso de uso .....	45
Diagrama 4 Diagrama de caso de uso Usuario .....	45
Diagrama 5 Diagrama de caso de uso Alumno .....	46
Diagrama 6 Diagrama de caso de uso Alumno experimental.....	46
Diagrama 7 Diagrama de caso de uso Docente .....	47
Diagrama 8 Diagrama de caso de uso Investigador.....	47
Diagrama 9 Diagrama de dominio .....	56
Diagrama 10 Diagrama navegacional .....	57

## ÍNDICE DE GRÁFICAS

Gráfica 1 Respuestas pregunta 1 .....	71
Gráfica 2 Respuestas pregunta 2 .....	72
Gráfica 3 Respuestas pregunta 3 .....	73
Gráfica 4 Respuestas pregunta 4 .....	73
Gráfica 5 Respuestas pregunta 5 .....	74
Gráfica 6 Respuestas pregunta 6 .....	75
Gráfica 7 Respuestas pregunta 7 .....	75
Gráfica 8 Respuestas pregunta 8 .....	76
Gráfica 9 Respuestas pregunta 9 .....	77
Gráfica 10 Respuestas pregunta 10 .....	77
Gráfica 11 Clasificación en 7 grupos .....	79
Gráfica 12 Grupos principales de estudiantes con emociones similares .....	80
Gráfica 13 Componentes principales .....	80

# CAPITULO I. INTRODUCCIÓN

Los avances tecnológicos y el proceso de digitalización en la industria hacen notable cada vez más la demanda de trabajadores altamente calificados para los sectores del software, tecnologías de comunicación y la información (Afi, 2017). De acuerdo con el Informe sobre el desarrollo mundial 2019: La naturaleza cambiante del trabajo, la evolución rápida del mercado laboral exige realizar una inversión en la salud y educación de las personas. El reto actual de las universidades es preparar a una mayor cantidad de jóvenes a la velocidad que las TI se mueven (Stromquist, 2019). En el caso de las ingenierías la programación es parte fundamental de la formación de un estudiante, ya que esta les permite enfrentar actividades de solución de problemas empleando dicha competencia. Aprender a programar consiste en plasmar la solución a un problema mediante un lenguaje de programación, cada programador lo realiza de manera distinta de acuerdo con su razonamiento y habilidades para crear una solución, y es en este último aspecto donde se encuentra la mayoría de las dificultades al momento de codificar y que resulta en un motivo de reprobación y deserción en estudiantes que cursan una ingeniería, sobre todo en el área de sistemas computacionales (Fuentes-Rosado & Moo-Medina, 2017)

En México la educación superior atendió en el ciclo escolar 2016-2017 a 3 millones 762 mil 679 estudiantes, con una tasa de deserción escolar del 6.8% (Dirección General de Planeación & Educativa, 2017). Por otra parte, de manera particular en el Tecnológico Nacional de México el 80% de su oferta académica corresponde a ingeniería, con una tasa de eficiencia terminal del 58%, lo cual hace evidente la necesidad de generar estrategias que permitan mejorar dicho porcentaje. El presente trabajo tiene la finalidad de mostrar el desarrollo e implementación de un sistema tutor afectivo que permita a los estudiantes de Ingeniería en Sistemas Computacionales, mejorar su desempeño académico para intentar así disminuir los índices de reprobación y deserción registrados en el Instituto Tecnológico Superior de Teziutlán.

## 1.1. PLANTEAMIENTO DEL PROBLEMA

En el caso de las ingenierías la programación es parte fundamental de la formación de un estudiante, ya que esta les permite enfrentar actividades de solución de problemas empleando dicha competencia, por lo cual es importante que los estudiantes aprendan programación básica en sus primeros años de preparación universitaria. Aprender a programar consiste en plasmar la solución a un problema mediante un lenguaje de programación, cada programador lo realiza de manera distinta de acuerdo con su razonamiento y habilidades para crear una solución, y es en este último aspecto en donde se encuentra la mayoría de las dificultades al momento de codificar y que resulta en un motivo de reprobación y deserción en estudiantes que cursan una ingeniería, sobre todo en el área de sistemas computacionales. En específico, las estructuras de datos al ser representaciones algorítmicas dinámicas suelen ser difíciles de entender para los estudiantes. Estas estructuras hacen uso de matemáticas abstractas que, si son basadas únicamente con fundamento teórico el estudiante no es capaz de comprenderlo. La principal característica de estos conceptos es para la manipulación de datos mediante la búsqueda, ordenación, almacenamiento y control. Su uso es indispensable en el desarrollo de software al tener que manipular datos continuamente y su poca o mala implementación producen aplicaciones de bajo rendimiento y poco optimizadas provocando una mala experiencia al usuario.

Aunado a lo anterior, además de los factores cognitivos como el razonamiento y la memorización, se encuentran los factores emocionales que juegan un papel importante en el aprendizaje, ya que estados como la frustración y el aburrimiento en el estudiante lo conducen hacia el abandono de la actividad, llevándolo a la reprobación y posteriormente a la deserción de la carrera.

## **1.2 OBJETIVO GENERAL**

Desarrollar e implementar un sistema tutor afectivo para la enseñanza de estructuras de datos usando técnicas de gamificación y el marco de trabajo KualiBeh.

### **1.2.1 OBJETIVOS ESPECÍFICOS**

- Analizar los requerimientos para el desarrollo de la aplicación.
- Generar los ejercicios que resolverán los estudiantes.
- Desarrollar los módulos requeridos de la aplicación
- Realizar pruebas de funcionamiento.
- Implementar el software con un grupo de control en el Instituto Tecnológico Superior de Teziutlán.

## 1.3 JUSTIFICACIÓN

La carrera de Ingeniería en Sistemas Computacionales ofertada por el Instituto Tecnológico Superior de Teziutlán, contiene dentro de su retícula el curso de Estructura de datos, asignatura del área de programación que aborda el contenido temático necesario, que sirve de base para futuras asignaturas de la misma área, ya que permite a los estudiantes manejar grandes cantidades de datos de manera eficiente en aplicaciones como grandes bases de datos y servicios de indización de Internet o para el diseño de algoritmos eficientes en el desarrollo de software. Sin embargo, ya que una estructura de datos es una forma particular de organizar datos en una computadora, muchas de las ocasiones resulta incompresible para el alumno el manejo de diversos tipos de estructuras de dato, lo cual genera estrés y ansiedad en los estudiantes representado un factor significativo que puede llevar a la desmotivación del estudiante por acreditar la asignatura o más allá, por continuar con la carrera, haciendo evidente la importancia de emplear herramientas que permitan mejorar el ambiente de aprendizaje de los estudiantes, prestando atención además de los factores cognitivos a los emocionales, que juegan un papel sumamente importante al momento de aprender a programar, pues si la motivación se pierde, los alumnos que cursan dicha asignatura pueden desistir en el intento de aprender y llevarlos a tomar decisiones que los encaminan a la reprobación y posterior deserción de la carrera.

Por lo anterior, es necesario generar herramientas tecnológicas de apoyo al proceso de enseñanza aprendizaje que permitan trabajar de manera particular con el estudiante, con un enfoque en el manejo de las emociones y de los estados afectivos presentados durante su proceso de aprendizaje, ya que cada individuo posee habilidades y características únicas que definen su capacidad de adaptación a un entorno, en este sentido al aprendizaje de estructuras de datos.

## 1.4 PREGUNTAS DE INVESTIGACIÓN

De lo anterior surgen las siguientes preguntas de investigación que guiarán este trabajo de tesis:

- ¿Es posible mejorar el desempeño escolar de un estudiante de estructuras de datos si se atiende de manera particular su estado emocional?
- ¿El uso de una herramienta tecnológica como apoyo al proceso de enseñanza aprendizaje puede ser un factor motivante para un estudiante de estructuras de datos?
- ¿Identificar las emociones de un estudiante de programación mientras resuelve un problema puede ayudar al docente a generar estrategias particulares para apoyarlo?

## **1.5 HIPÓTESIS Y VARIABLES**

### **1.5.1 Hipótesis de trabajo**

Un sistema de tutor afectivo puede mejorar el desempeño escolar de un estudiante de estructura de datos.

### **1.5.2 Hipótesis nula**

Un sistema tutor afectivo no es un factor relevante para mejorar en el desempeño escolar de un estudiante de estructura de datos.

### **1.5.3 Variables**

Las variables de estudio son:

- Emociones
- Calificaciones

## 1.6 PROPUESTA TECNOLÓGICA

El presente trabajo busca generar una herramienta tecnológica de apoyo en el proceso de enseñanza aprendizaje, para el docente que imparte la materia de Estructura de Datos de la carrera de Ingeniería en Sistemas Computacionales dentro del Instituto Tecnológico Superior de Teziutlán, con el fin de mejorar el desempeño escolar de los estudiantes. Esto, debido a la complejidad de las estructuras manejadas que resultan difícil de conceptualizar por los estudiantes, ocasionando problemas en la asimilación del conocimiento, y por ende afectando su rendimiento académico.

La herramienta propuesta es un Sistema Tutor Afectivo que contendrá actividades de aprendizaje sobre estructuras de datos, dichas actividades se presentaran en forma de juego y por medio de recompensas para motivar al estudiante, al mismo tiempo se hará un reconocimiento en tiempo real de las emociones presentadas por el estudiante para identificar los estados afectivos en los que se encuentra mientras resuelve las actividades, con el objetivo de generar la retroalimentación oportuna o la guía necesaria para mantener por mayor tiempo al estudiante en un estado de flujo y concentración, buscando así logre apropiarse del conocimiento y por ende mejorar su desempeño académico.

Las herramientas con las que se desarrollara el sistema son el framework de Ionic, los servicios de firebase y el api de Google Vision AI, esto debido a su utilidad gratuita para todos los elementos de desarrollo y ejecución que requiere el sistema.

## **1.7 ESTUDIO DE FACTIBILIDAD**

La factibilidad del proyecto se puede describir desde tres perspectivas que se definen de la siguiente manera.

### **1.7.1 Económica**

El diseño del sistema tutor afectivo es factible económicamente ya que las herramientas utilizadas para el desarrollo de este son totalmente gratuitas, por lo que no se requiere el pago de ningún tipo de licenciamiento. Por otra parte, el instituto donde se desarrolla el proyecto cuenta con el equipo suficiente para poder realizar adecuadamente el trabajo y su posterior uso por los docentes. Así mismo, para la implementación de la propuesta al tratarse de una aplicación móvil cada estudiante podrá utilizar el tutor por medio de su equipo personal.

### **1.7.2 Técnica**

En cuanto a la factibilidad técnica, se cuenta con el conocimiento y habilidades necesarias de las diferentes herramientas tecnológicas para el desarrollo del proyecto por lo que no existe inconveniente alguno en este sentido.

### **1.7.3 Humana**

El instituto en donde se implementará el sistema cuenta con el personal necesario y capacitado en el uso de herramientas tecnológicas para su uso y mantenimiento si así fuera necesario.

## 1.8 ANTECEDENTES

La sociedad actual se encuentra cada día más digitalizada, por lo que, saber programar es una competencia básica que debe poseer cualquier profesionista, ya que ésta le permite pensar computacionalmente, así como, ser capaz de leer y escribir a través de diversos lenguajes de programación (M. R. González, 2014). En todas las ingenierías, aprender a programar puede resultar complicado, debido a que la gran mayoría de los estudiantes recibe este tipo de formación hasta su preparación universitaria. La programación, puede ser en los primeros intentos un reto de aprendizaje, ya que requiere de un razonamiento que permita resolver problemas computacionalmente de ahí, que en muchas ocasiones aprender a programar puede generar frustración, ansiedad e incluso aburrimiento. Sin embargo, en los mejores casos los retos de programación pueden ser un motivante para seguir aprendiendo.

Lo anterior resalta la importancia de las emociones dentro del proceso de enseñanza aprendizaje, ya que están juegan un factor importante que definen gran parte del desempeño académico del estudiante. Así, las emociones centradas en el aprendizaje se producen cuando los estudiantes al realizar diferentes actividades manifiestan una serie de estados afectivos tales como la frustración o el aburrimiento, considerados como negativos donde generalmente el estudiante termina por abandonar la actividad.

Por otra parte, se presentan estados afectivos positivos como la concentración y relajación, que colocan al estudiante en un estado de flujo permitiéndole comprender adecuadamente la actividad que está realizando y llevándolo así hacia la apropiación del conocimiento (Daniel, 2015).

El uso de herramientas computacionales para apoyar el proceso de enseñanza es muy común. Una de estas herramientas son los Sistemas Tutores Inteligentes (ITS por sus siglas en inglés), los cuales utilizan técnicas de inteligencia artificial para dotar a la computadora la capacidad de involucrarse en el proceso de enseñanza del estudiante al proveer asistencia a este de manera en que lo haría un instructor

humano (Huapaya, 2009). Su aplicabilidad en la educación radica, en que son un tipo de ambiente interactivo que está diseñado para propiciar el aprendizaje individual, distinguiéndose de otras herramientas por su capacidad para modelar el estado cognitivo del usuario, permitiendo guiar al estudiante y proporcionándole la retroalimentación necesaria en cada una de las etapas de su proceso de aprendizaje (Durango Hernández & Pascuas Rengifo, 2016).

Evidentemente, existe un debate entre si un STI puede ser igual de funcional que un tutor humano, si bien este ofrece una mayor flexibilidad y sobre todo sutileza durante el proceso de enseñanza aprendizaje, el proceso que ambos siguen es muy similar, por lo que puede servir claramente como una herramienta de apoyo al proceso de aprendizaje de bastante utilidad (Roschelle, 2009).

Los STI que prestan atención a las emociones presentadas por el estudiante durante el proceso de aprendizaje, se conocen como Sistemas Tutores Afectivos (STA), los cuales se enfocan en identificar los estados afectivos presentados por el estudiante mientras resuelve una actividad, si bien estos pueden resultar complejos de atender ya que pueden tener diversos orígenes, el objetivo principal del STA es evaluar las emociones presentadas y definir a partir de estas acciones que permitan guiar al estudiante y retroalimentarlo de la mejor manera para llevarlo al logro de la actividad y promover así en el la apropiación del conocimiento (Petrovica, AnohinaNaumeca, & Ekenel, 2016). Sin embargo, el uso de sistemas tutores afectivos no es tan común, debido al tipo de aspectos como la expresividad presentada por el tutor, o la capacidad de reconocimiento de las emociones, así como la facilidad de uso, por lo que resulta importante identificar claramente las ventajas de estos al poder atender las emociones presentadas por el estudiante y así poder implementarlos adecuadamente en las instituciones educativas (Mao & Li, 2010).

Específicamente en al área de programación existen algunos trabajos relacionados con la implementación de sistemas tutores inteligentes para la enseñanza de esta.

Uno de estos es STIPB, el cual es un sistema tutor inteligente para programación básica en C, el cual mostro resultados favorables al generar un ambiente flexible e

interactivo, proporcionando al estudiante un abanico de posibilidades para la navegación en la realización de las actividades presentadas (Guilcatoma Panchi & Quimbita Cadena, 2006).

Master-Pro, es otro sistema tutor inteligente desarrollado para gestionar el conocimiento referido a temas de programación, el cual realiza análisis en relación a las inteligencias múltiples con base en conceptos, proporcionándole al docente la información necesaria para que este pueda generar estrategias adecuadas al tipo de inteligencia que muestran sus estudiantes. Además, se enmarca nuevamente la capacidad de este tipo de sistemas para la personalización de actividades dependiendo de las necesidades propias de cada estudiante (Teresa MachinTorres, 2017).

Otro tutor inteligente es el denominado JITS, el cual está enfocado en la enseñanza de programación en un entorno de JAVA, caracterizando al sistema por su capacidad de detectar, predecir y corregir errores lexicográficos en las actividades de codificación de los estudiantes obteniendo resultados favorables en el desempeño de los estudiantes con la implementación de este como estrategia de enseñanza aprendizaje (Sykes, 2010).

De manera particular, para el caso de sistemas tutores afectivos, se han realizado algunas investigaciones sobre la implementación de estos, tales como el sistema denominado Java Sensei cuyo objetivo es enseñar a codificar en JAVA de manera individualizada, este sistema enfoco el análisis de sentimiento específicamente en la retroalimentación para identificar como podría llegar a cambiar el estado de ánimo del estudiante según la retroalimentación proporcionada y a partir de ello generar la información pertinente tanto para análisis del docente como para una mejor interacción con el estudiante (Barron-Estrada, Zatarain-Cabada, Oramas-Bustillos, & Gonzalez-Hernandez, 2017). Finalmente, otra herramienta que denota las ventajas de los sistemas tutores afectivos es EasyLogic, el cual proporciona un ambiente interactivo para el aprendizaje de lógica algorítmica y programación, mediante técnicas de gamificación, que resalta la importancia de identificar sobre todo los estados afectivos negativos para lograr encaminar al estudiante

oportunamente hacia un estado positivo que le permita seguir motivado y continuar con la actividad (Zatarain Cabada, 2018).

Lo anteriormente descrito, resalta la importancia de generar nuevas herramientas que puedan ser útiles para el acompañamiento del estudiante durante su proceso de aprendizaje y más aun para el apoyo emocional y de motivación que se requiere cuando se presentan situaciones complejas al abordar un tema, en este caso del área de programación.

## **CAPITULO II. MARCO TEORICO**

A continuación, se explican los fundamentos teóricos que sustenta la presente investigación.

### **2.1 Emociones en el aprendizaje**

El procesamiento de las emociones se realiza principalmente por el sistema límbico, conformado entre otras estructuras por la amígdala y el hipocampo, los cuales son a su vez, las principales regiones cerebrales dedicadas al aprendizaje (“¿Qué importancia tienen las emociones en el aprendizaje? - NeuroClass,” n.d.).

La afectación que tiene un individuo al relacionarse con diferentes estímulos, estados o situaciones del entorno se conoce como inteligencia emocional, al trasladarse al ámbito educativo, las emociones, los estados de ánimo y la motivación influyen en cómo se perciben las situaciones lo cual a su vez interviene en el proceso de enseñanza aprendizaje (GARCIA, HURTADO, QUINTERO, RIVERA, & UREÑA, 2018).

Dentro del proceso de enseñanza aprendizaje, las emociones surgen espontáneamente desde la alegría por acreditar un examen hasta la frustración por no comprender un tema o la tristeza por no acreditar una evaluación. Dichos estados afectivos positivos o negativos pueden desencadenar diferentes repercusiones en el aprendizaje del alumno. Así, cuando un estudiante está realizando alguna actividad, pasa por algunos estados emocionales como confusión, frustración, aburrimiento y compromiso. Cuando está atendiendo temas difíciles puede darse que estos estados afectivos lleven al estudiante al abandono de la actividad (D’Mello & Graesser, 2011).

Los estudios realizados en el área de la neurociencia en relación a la importancia de las emociones en el aprendizaje dan paso al debate de si los docentes deben ser encargados del desarrollo emocional de sus estudiantes, debido a que las emociones se presentan en las mismas zonas cerebrales que los procesos cognitivos (Elizondo Moreno, Rodríguez Rodríguez, & Rodríguez Rodríguez, 2018).

## **2.2 Computación afectiva**

La computación afectiva se define como una disciplina dentro del campo de la inteligencia artificial enfocada en el desarrollo de métodos de computación centrados en reconocer y generar emociones humanas (“Computación afectiva - InnoWiki,” n.d.). También, puede considerarse como el estudio y desarrollo de sistemas y dispositivos que pueden reconocer, interpretar, procesar y simular las emociones humanas (Navarro, 2018).

Las emociones juegan un papel importante tanto en la cognición como en la percepción humana, por lo que, la computación afectiva provee a las computadoras la capacidad de tomar mejores decisiones a través del reconocimiento de estas (Lisetti, 1998).

Existen diferentes medios para reconocer las emociones tales como el reconocimiento facial, señales del habla, pulsaciones, presión sanguínea, entre otras, siendo la combinación de estas la que ofrece mejores resultados (Gina, Rocio, Ángel, José, & Luis, 2018).

## **2.3 Reconocimiento facial**

Se puede definir al reconocimiento facial como la capacidad de todos los individuos de reconocer formas básicas de expresión en el rostro humano (Iglesias-Hoyos, del Castillo Arreola, & Muñoz-Delgado, 2016). El rostro es uno de los canales más ricos de expresión, el cual comunica emociones y señales sociales (Gina et al., 2018).

Existen siete emociones universales que todos los seres humanos sienten y que trascienden las diferencias lingüísticas, regionales, culturales y étnicas, estas son: enojo, desprecio, disgusto, alegría, miedo, tristeza, sorpresa (“Universal Emotions | What are Emotions? | Paul Ekman Group,” n.d.).

El Sistema de Codificación de Acción Facial (FACS por sus siglas en inglés) creado por Ekman y Friesen en 1978, es la taxonomía más objetiva y usada para para las

investigaciones que buscan desarrollar tecnología que detecten las emociones en tiempo real para diferentes objetivos (Gina et al., 2018).

## **2.4 Sistema tutor inteligente y Afectivo**

Los sistemas tutores inteligentes, se definen como sistemas computacionales diseñados para impartir instrucción y apoyar al estudiante durante su proceso de enseñanza aprendizaje, de manera muy similar a como lo haría un tutor humano (Durango Hernández & Pascuas Rengifo, 2016).

Un ITS se compone de cuatro elementos clave: el modelo de alumno, modelo de dominio, modelo de tutor y módulo de interfaz o comunicación para la interacción con el usuario (C. González, Mora, & Toledo, 2014).

El principal objetivo de los sistemas tutores inteligentes es comunicar el conocimiento efectivamente, y no necesariamente comunicarse con el de manera en cómo lo haría un tutor humano. Es decir, busca lograr que el estudiante asimile en conocimiento de manera progresiva y limitada a lo que está preparado a entender (Romdhani, 2016).

Un sistema tutor afectivo (ATS por sus siglas en inglés) es una variación de un ITS, que incluye la integración del reconocimiento de emociones para actuar en relación a estas dentro del proceso de enseñanza aprendizaje de un estudiante (Ben Ammar, Neji, Alimi, & Gouardères, 2010). Así, los ATS poseen la habilidad para reconocer el estado afectivo de los estudiantes e interactuar con el mediante ejercicios que estimulen su estado emotivo (Zatarain-Cabada, Barrón-Estrada, & GarcíaLizárraga, 2016).

Los ATS han demostrado en diversas investigaciones la importancia de las emociones, para mejorar el desempeño académico de los estudiantes, siendo imprescindible la capacidad de empatía con el estudiante (Ponce, Pintado, & Biset, 2019).

## 2.5 Gamificación

La gamificación consiste en el uso de mecánicas y estrategias de juego en contextos diferentes a este. Es actualmente utilizada en el aspecto educativo, donde se busca alentar al estudiante en el cambio del comportamiento y mejorar la solución al problema presentado (C. González et al., 2014).

Existen muchos beneficios que se asocian al juego, entre ellos aspectos de motivación, razonamiento lógico, interactividad y experiencia de desarrollo que pueden fomentar el aprendizaje (Zatarain Cabada, 2018).

## 2.6 Herramientas de diseño

A continuación, se describen las herramientas de diseño utilizadas para el desarrollo del sistema tutor afectivo.

### 2.6.1 Balsamiq

Balsamiq es un software para la creación de Wireframes. (*Balsamiq*, 2021). Para entender cuál es el funcionamiento de Balsamiq, primero es necesario comprender que es un Wireframe, un Wireframe es un boceto con ideas de un producto o el esqueleto de una interfaz de una aplicación o sitio web. Los Wireframes comúnmente se realizan papel, pero tienen desventajas con respecto a utilizar una aplicación como Balsamiq, como la interactividad portabilidad y simplicidad. (Tutorialspoint, 2014)

Entre las ventajas que podemos destacar de Balsamiq son:

- Fácil uso: Puedes crear wireframes en cuanto abres la aplicación.
- Herramienta simple: Balsamiq es fácil de entender y de transferir tus ideas de tu cabeza a la computadora fácil y de manera limpia.
- Portabilidad: Los archivos de Balsamiq son fácil de compartir entre tus colaboradores o clientes. Con Balsamiq tu puedes diseñar y compartir entre

múltiples sistemas operativos y enviar a través de correo, imprimir u compartir de manera online.

- Asequibilidad: El costo de Balsamiq es menor con respecto a otras herramientas de diseño como Photoshop o Visio, además de contar con licencias gratuitas para estudiantes, docentes o investigadores. (Faranello, 2012)

## **2.6.2 StarUML**

StarUML es una herramienta para crear diagramas de clases UML y generar automáticamente código. También puede aplicar ingeniería inversa al código fuente para producir diagramas UML correspondientes. (Wong, 2007).

Esta herramienta entra en la categoría de herramientas CASE (Computer Aided Software Engineering) o Ingeniería de Software Asistida por Computadora las cuales son herramientas que nos permiten visualizar, especificar, construir y documentar los artefactos de un sistema de software. (Booch et al., 1996).

En particular StarUML es la herramienta más conocida dentro de la categoría de herramientas CASE, es la que cuenta con la mayor cantidad de descargas con respecto a sus competidores, cuenta con soporte a UML 2.0 y la mayoría de sus modelos, además de ser open-source. Es posible generar la estructura de las clases a través de los diagramas diseñados y posteriormente generar código auxiliar para agregar la funcionalidad del código especificando lo que debe hacer cada método.

## **2.7 Herramientas de desarrollo**

En este apartado se definen las herramientas de desarrollo y codificación utilizadas para el sistema tutor afectivo.

### **2.7.1 Visual Studio Code**

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft y disponible bajo licencia Open-Source. Está disponible para prácticamente cualquier

sistema operativo (Windows, macOS y Linux) y cuenta con soporte incorporado para Javascript, TypeScript y Nodejs aunque instalando plugins es posible incluir compatibilidad con otros lenguajes de programación tales como C++, c#, Java, PHP, entre otros, además de runtimes como .NET y Unity.

Entre las características que podemos destacar de VS Code son:

- Colaborar y codificar de forma remota mediante extensiones como LiveShare.
- Resaltado de palabras clave en su código además de incluir a través de extensiones de IntelliSense y PeekDefinition que permiten identificar errores sintácticos y semánticos desde el desarrollo.
- Brinda soluciones rápidas para errores comunes semánticos y sintácticos, además de la integración de un depurador que puedes configurar para diferentes lenguajes de programación.
- Personalización del entorno utilizando temas, fuentes, iconos, accesos directos, entre otras opciones.
- Terminal y gestor de versiones git integrados en su interfaz, lo que permite concentrar el desarrollo en una sola aplicación.

Aunque VS Code no es considerado un Entorno de Desarrollo Integrado (IDE), cuenta con las herramientas suficientes para desarrollar aplicaciones robustas y colaborativas. (*Visual Studio Code*, 2021).

## **2.7.2 GIT**

Git es un sistema de control de versiones gratuito y de código abierto para el manejo de proyectos de software.

Git es una herramienta de fácil uso, no consume muchos recursos y es el control de versiones más popular en el mercado por una gran diferencia.

Esta herramienta implementa un modelo de ramificación lo que permite tener múltiples versiones locales que trabajan de forma independiente, pero de ser necesario, es posible realizar una fusión entre ellas. Esta función permite mantener cambios sin comprometer el funcionamiento en un entorno de producción, además, si desea realizar

pruebas puede realizarlo en una rama diferente y, decidir si desechar o realizar una fusión. (*Git-Scm.Com*, 2021).

### **2.7.3 Gitlab**

Gitlab es una aplicación web que permite gestionar repositorios Git en la nube. A diferencia de sus competidores, Gitlab es opensource lo que permite montar tu propia plataforma en tus servidores, ofrece alojamiento a repositorios ilimitados y sin límite de miembros colaboradores. Además de servir como alojamiento de repositorios cuenta con herramientas DevOps para la automatización del despliegue de aplicaciones, Wikis, y seguimiento de errores. (*Gitlab.Com*, 2021).

De acuerdo con su sitio web, las herramientas de DevOps son las más utilizadas en el mercado.

### **2.7.4 Android Studio**

Es una herramienta para la creación de aplicaciones enfocadas a dispositivos Android, cuenta con un editor de diseño visual lo que le permite ir generando vistas utilizando una interfaz gráfica, analizador de APK que le permiten examinar los archivos de su aplicación, emulador integrado para simular dispositivos físicos con diferentes configuraciones y plataforma, editor de código inteligente que incluye colores distintivos para palabras clave, sugerencias de código, refactorización entre otras herramientas, Sistema de construcción flexible que le permite personalizar a través de Gradle diferentes versiones de compilación, perfiladores de tiempo real para examinar el uso de CPU, memoria ram, actividad de red y muchas otras herramientas.

Esta herramienta es desarrollada y mantenida por Google y constantemente recibe actualizaciones de seguridad e incorporación de herramientas. Además de una integración fácil con otros servicios de Google como Firebase o Google Cloud. (*Developer.Android.Com*, 2021).

## **2.8 Metodología de desarrollo de software Kuali-Beh**

De las palabras náhuatl Kuali (bueno) y Beh (Camino), es una propuesta de marco de trabajo mexicana para la expresión, comparación y mejora de las formas de trabajo de Ingeniería de Software, forma parte del estándar ESENCE y describe los conceptos comunes y las relaciones presentes en los proyectos de software.

KUALI-BEH consta de dos componentes principales, vista estática, donde, a través de conceptos comunes entre cualquier proyecto de software se definen elementos básicos para construir prácticas que definan la forma de trabajo de un equipo; y la vista operacional que abarca la ejecución de los métodos definidos mediante los elementos de la vista estática.(Arroyo-lópez et al., 2015)

## **2.9 Tecnologías**

En el siguiente apartado se describen las tecnologías utilizadas para el desarrollo del sistema tutor afectivo.

### **2.9.1 NodeJs**

Es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome orientado a eventos asíncronos, diseñado principalmente para crear aplicaciones escalables. A diferencia de tecnologías similares maneja un sistema de eventos diferente, más parecido al de Event Machine de Ruby o de Twisted de Python. NodeJs es una tecnología cada vez más utilizada para el desarrollo de aplicaciones web al compartir lenguaje tanto en front-end como en back-end reduciendo el costo de desarrollo, y reutilizando partes del código y por su baja latencia en operaciones de multiproceso y streaming se ha vuelto muy adecuado para estos proyectos. (Nodejs.Org, 2021).

### **2.9.2 Angular**

Es una plataforma que permite desarrollar aplicaciones web utilizando HTML, CSS y JavaScript para que el cliente asuma la mayor parte de la lógica y descargue al servidor con la finalidad de que las aplicaciones ejecutadas a través de internet sean más

rápidas. Es mantenido por Google y es uno de los Frameworks SPA (Simple Pages Application) más populares. (Boada Oriols & Gómez Guitiérrez, 2018)

Al tener un mejor rendimiento con respecto a aplicaciones tradicionales permite realizar aplicaciones móviles híbridas con un buen uso de recursos computacionales.

Su principal premisa es el uso de componentes que permiten encapsular la funcionalidad en pequeños módulos independientes que en conjunto forman una aplicación.

Aunque sus nombres son muy parecidos no se debe confundir con AngularJS, que son completamente diferentes. AngularJs fue la primera propuesta de Google de desarrollar una plataforma para el desarrollo de aplicaciones SPA, y aunque se pensaba seguir el proyecto, los cambios realizados dentro del core fueron tan diferentes que se decidió por definirlo como uno nuevo.

Angular usa el patrón MVC y es compatible con el lenguaje TypeScript además de que implementa herramientas para testing, Lazy loading, y ejecución a través del servidor.

### **2.9.3 Ionic Framework**

Ionic Framework es un marco para el desarrollo de aplicaciones móviles con HTML, CSS y Javascript, las aplicaciones en ionic se ejecutan como aplicaciones nativas y su principal ventaja es que tienen una “apariencia” nativa por el uso de componentes visuales parecidos a los de un desarrollo nativo.

Ionic inicialmente estaba basados en AngularJs, pero en sus últimas versiones es compatible con Angular, React, Vue, entre otros, cuenta con una colección de componentes visuales que permiten construir aplicaciones móviles rápidamente.

Las aplicaciones desarrolladas en ionic son construidas sobre Apache Cordova o Capacitor que permiten tener acceso a funciones nativas del celular como lo son la cámara, micrófono, ubicación, entre otros.

Al ser aplicaciones híbridas que se ejecutan a través de un webview, permiten ser multiplataforma permitiendo exportar aplicaciones para iOS, Android, Web, Escritorio

(implementando electrón) o como una WebApp progresiva.(SUPARATCHADECH, 2017)

#### **2.9.4 Capacitor**

Capacitor es un runtime nativo multiplataforma que facilita la creación de aplicaciones híbridas para iOS, Android y la Web. Proporciona un enfoque moderno de contenedor nativo para la construcción de aplicaciones nativas.

Capacitor proporciona un conjunto de API's que permiten la creación de aplicaciones basado en el "one code base for multiple platforms" que permiten utilizar los elementos nativos del celular sin preocuparse por la compatibilidad con las distintas plataformas. Además, permite agregar funcionalidad utilizando Swift o Java creando complementos que se integran fácilmente.

Capacitor es considerado el sucesor no oficial de Apache Cordova y de Adobe PhoneGap debido a que cuenta con un mayor performance y compatible con muchos de los complementos de Cordova.(*Capacitorjs.Com*, 2021).

#### **2.9.5 Rextester**

Es un compilador online compatible con múltiples lenguajes de programación. Implementa una API para procesar las solicitudes y retorna el resultado de la ejecución. La ventaja de un compilador online es que no requiere de ninguna instalación para funcionar, lo que permite realizar pruebas, ejemplos o ejecución de secciones de código de manera rápida.

#### **2.9.6 Google Cloud**

Google Cloud es un conjunto de servicios modulares basados en la nube que proporcionan componentes básicos para el desarrollo de aplicaciones, desde sitios web estáticos hasta aplicaciones de varias capas y niveles. (Krishnan & Gonzalez, 2015).

Google Cloud cuenta con servicios para la creación de aplicaciones web, bases de datos relacionales y no relacionales, manejo de contenedores a través de Docker,

herramientas para el Big Data y Business Analytics, servidores virtuales, entre algunas otras más. (Google Cloud Platform, 2015).

Google Cloud consiste en un conjunto de recursos físicos, como computadoras y unidades de disco duro, y recursos virtuales, como máquinas virtuales (VM), que se encuentran en los centros de datos de Google en todo el mundo. Cada centro de datos está ubicado en una región, específicamente en Asia, Australia, Europa, América del Norte y América del Sur. Cada región es una colección de zonas aisladas entre sí dentro de cada región. (Cloud.Google.Com, 2021b).

### **2.9.7 Vision AI**

Es un servicio de Google Cloud que permite extraer información de imágenes en la nube a través de reconocimiento de patrones lo que permite detectar emociones, interpretar texto, localizar objetos y muchas cosas más. (Cloud.Google.Com, 2021a)

La API de Vision ofrece modelos de aprendizaje automático predeterminados y muy potentes a través de las APIs REST y RPC.

Entre las principales ventajas se destaca:

- Detecta objetos automáticamente.
- Obtiene información valiosa en el perímetro.
- Comprende texto mediante un OCR para detectar texto en más de 50 idiomas
- Detecta contenido explícito para su filtrado.

### **2.9.8 Firebase**

Firebase es una plataforma que contiene herramientas para programar aplicaciones móviles sin preocuparse de implementar un backend. Ya que sus herramientas proveen servicios de base de datos, almacenamiento, ejecución de código, autenticación y muchas otras herramientas sin necesidad de configurar un servidor. (Wieruch, 2019).

Google adquirió Firebase en el año 2014 y actualmente es un producto integrado con Google Cloud Platform.

### **2.9.9 Firebase Cloud Firestore**

Es una base de datos no relacional flexible y escalable enfocada para dispositivos móviles y web que se integra con los servicios de Firebase y Google Cloud. Está basado en documentos y colecciones. Cuenta con un sistema de sincronización para múltiples dispositivos y escuchas en tiempo real, soporte para datos sin conexión que se sincronizan una vez restablecida la conexión. (Moroney, 2017)

### **2.9.10 Firebase Authentication**

Es un servicio de Firebase que permite administrar la autenticación de usuarios mediante correo electrónico y contraseña, número de teléfono o servicios externos como Google, Facebook, Twitter. Al ser un servicio integrado con la plataforma Google Cloud, es posible controlar el acceso a los distintos recursos que ofrece desde esta API o incluirlo en servicios externos a través de su SDK. (Moroney, 2017)

### **2.9.11 Firebase Cloud Functions**

Este servicio provee de ejecución de código por parte del servidor sin la necesidad de configurar uno. Corre en un entorno de NodeJs utilizando v8. Sus versiones actuales son compatibles con las versiones 12 y 14. (Moroney, 2017)

### **2.9.12 Firebase Cloud Storage**

Firebase Cloud Storage es un servicio más de Firebase que permite la integración con el resto de los servicios de manera fácil y dinámica lo que lo vuelve diferente a Google Cloud Storage. Permite almacenar fotos, videos o archivos en la nube y mediante en conjunto con Firebase Authentication permite generar reglas de acceso a los archivos almacenados. (Moroney, 2017).

## **CAPITULO III. METODOLOGIA DE LA INVESTIGACIÓN**

La metodología establecida para el desarrollo de la presente investigación consta de diferentes etapas y características que se explican a continuación.

### **3.1 DISEÑO METODOLÓGICO**

La primera etapa de la investigación se enfoca en la revisión de la literatura, para establecer las motivaciones y oportunidades dentro del campo de la computación afectiva. Comprender como las emociones pueden ser influidas por medios tecnológicos y afectar positivamente en este caso el proceso de aprendizaje de un estudiante de programación.

La segunda etapa consiste en el desarrollo del sistema tutor afectivo, realizando el análisis de las actividades de aprendizaje de estructuras de datos, propuestas en el plan de estudios vigente del Tecnológico Nacional de México. Posteriormente, el diseño y codificación de estas, la incorporación de la capacidad de análisis facial para la detección de las emociones y las pruebas de funcionamiento.

Finalmente, la tercera etapa consiste en la implementación del sistema tutor afectivo en un grupo experimental para poder generar los datos necesarios y realizar la comparación con el grupo de control, generar un análisis de los resultados y hacer la presentación de estos.

#### **3.1.1 Tipo de investigación**

Para la presente investigación se llevará a cabo una investigación de tipo cuasi experimental ya que se contará con un grupo de estudio predefinido por la asignación correspondiente en la materia.

### **3.1.2 Enfoque de la investigación**

La investigación se llevará a cabo bajo un enfoque cuantitativo que permita establecer objetivamente la relación existente entre las emociones presentadas por los alumnos y las calificaciones obtenidas.

### **3.1.3 Alcance y tipo de estudio**

Se realizará un estudio correlacional ya que busca encontrar la relación entre las emociones y las calificaciones obtenidas por los estudiantes, para poder definir estrategias para la mejora de su desempeño escolar.

### **3.1.4 Instrumentos de recolección de datos**

Para la obtención de datos se hará uso de un cuestionario (pretest y postest), así como de la aplicación móvil para el registro de las emociones en tiempo real.

## **3.2 DISEÑO DE LA INVESTIGACIÓN**

La presente investigación se trata de una investigación de tipo cuasi experimental ya que el grupo de control y experimental son definidos previamente por cuestiones de planificación escolar por lo que se tiene poco control de los individuos que serán objeto de estudio. La investigación sigue también un enfoque de tipo transversal, ya que se busca encontrar la correlación existente entre los estados afectivos presentes en el aprendizaje y el desempeño escolar del estudiante reflejado en sus calificaciones.

La investigación se diseñó de dicha manera ya que la implementación se llevará a cabo con estudiantes de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico Superior de Teziutlán, quienes llevan en su plan curricular la materia de Estructuras de Datos. Debido a cuestiones de planeación por parte de

las instancias administrativas, la implementación de la propuesta se hará únicamente durante dos semanas, tratando de definir en la medida de lo posible los mejores hallazgos respecto a la utilización del sistema tutor afectivo.

### 3.3 MUESTRA

La implementación de la propuesta se llevará a cabo con una muestra no probabilista de los estudiantes que cursan la carrera de Ingeniería en Sistemas Computacionales en el Instituto Tecnológico Superior de Teziutlán, debido a que se llevará a cabo en el grupo A del cuarto semestre de la carrera que cursan la materia de Estructura de Datos, el cual cuenta con 20 estudiantes en edad promedio de 19 años. Designado de esta manera por cuestiones de planeación escolar.

### 3.4 INSTRUMENTOS DE MEDICIÓN

A continuación, se describen los instrumentos de medición utilizados para registrar los datos de los alumnos que interactúen con la aplicación, así como la definición de las variables de estudio.

#### 3.4.1 Operacionalización de las variables

La Tabla 1 muestra la descripción de las variables en virtud de su significado para la investigación definiendo el propósito de estas. Para esta investigación únicamente se consideran las emociones presentadas y la calificación obtenida.

*Tabla 1 Operacionalización de las variables de investigación*

<b>Variable</b>	<b>Definición</b>
<b>Emoción</b>	Emoción reconocida para la definición del estado afectivo que permitirá la interacción con el usuario.

<b>Calificación</b>	Escala de desempeño escolar obtenida por el estudiante en una actividad.
---------------------	--

### 3.4.2 Instrumentos de medición

El instrumento para utilizar en la medición del desempeño escolar será un cuestionario que se utilizará como pretest y postest. A continuación, se presenta el cuestionario a aplicar.

1. Es un modelo donde los datos están organizados, manejados y almacenados de manera eficiente para su acceso y modificación de los datos.
  - a) Estructura de datos
  - b) Algoritmo
  - c) Árbol
  - d) Clases Abstractas
2. Son ejemplos de estructuras de datos (Selección múltiple)
  - a) Pila
  - b) String
  - c) Grafo
  - d) Lista enlazada
  - e) Condicionales
  - f) Clases
3. ¿Cuál de los siguientes algoritmos es el óptimo para realizar la suma de los primeros 15 números?
  - a) `int suma = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15;`
  - b) `int suma = 0;`  
`for (int i = 1; i <= 15; i++) { suma += i; }`
  - c) `int suma = 15 * (15 + 1) / 2; // Formula Gaussiana`

4. ¿Qué elemento NO es un elemento clave para tomar en cuenta para escoger el mejor algoritmo?
  - a) Grado de orden
  - b) Cantidad de datos
  - c) Tipo de datos
  - d) Tamaño de los datos
  - e) Número de líneas
5. Es un método que se llama a sí mismo
  - a) Método heredado
  - b) Método sobrecargado
  - c) Método recursivo
  - d) Método con valor de retorno
6. ¿Qué es LIFO?
  - a) Ultimo en entrar, primero en salir
  - b) Ultimo en entrar, ultimo en salir
7. ¿En qué casos es útil ocupar un método de ordenamiento secuencial?
  - a) Cuando son muchos datos
  - b) Cuando los datos están desordenados
  - c) Cuando los datos se encuentran en tablas hash
8. Son ejemplos de estructuras NO lineales (múltiple)
  - a) Lista doblemente enlazada
  - b) Árbol
  - c) Pila
  - d) Grafo
  - e) Cola circular
  - f) Arreglo
  - g) Diccionario de datos
9. ¿Cuál de los siguientes métodos tiene una menor complejidad?
  - a) Ordenamiento de burbuja
  - b) Ordenamiento de inserción
  - c) Ordenamiento por casilleros

d) Ordenamiento Radix

e) Quicksort

10. Son tipos de corrimiento que se le pueden aplicar a un árbol

a) Amplitud

b) Profundidad

c) Simple

d) Secuencial

e) Lineal

Las preguntas diseñadas tienen la finalidad de identificar la conceptualización que poseen los alumnos sobre estructuras de datos antes y posterior al uso del sistema tutor afectivo.

# CAPITULO IV. METODOLOGIA DE DESARROLLO DE SOFTWARE

El presente capítulo explica el proceso de desarrollo de software llevado a cabo explicado en sus diferentes etapas para la implementación del software de investigación.

## 4.1 Metodología de desarrollo

La metodología de desarrollo empleada para el desarrollo del sistema tutor afectivo fue el marco de trabajo Kuali-Beh, el cual fue diseñado para la expresión, comparación y mejora de las formas de trabajo de grupos pequeños de desarrollo de software. El Diagrama 1 muestra las prácticas propuestas para el desarrollo del sistema tutor afectivo.

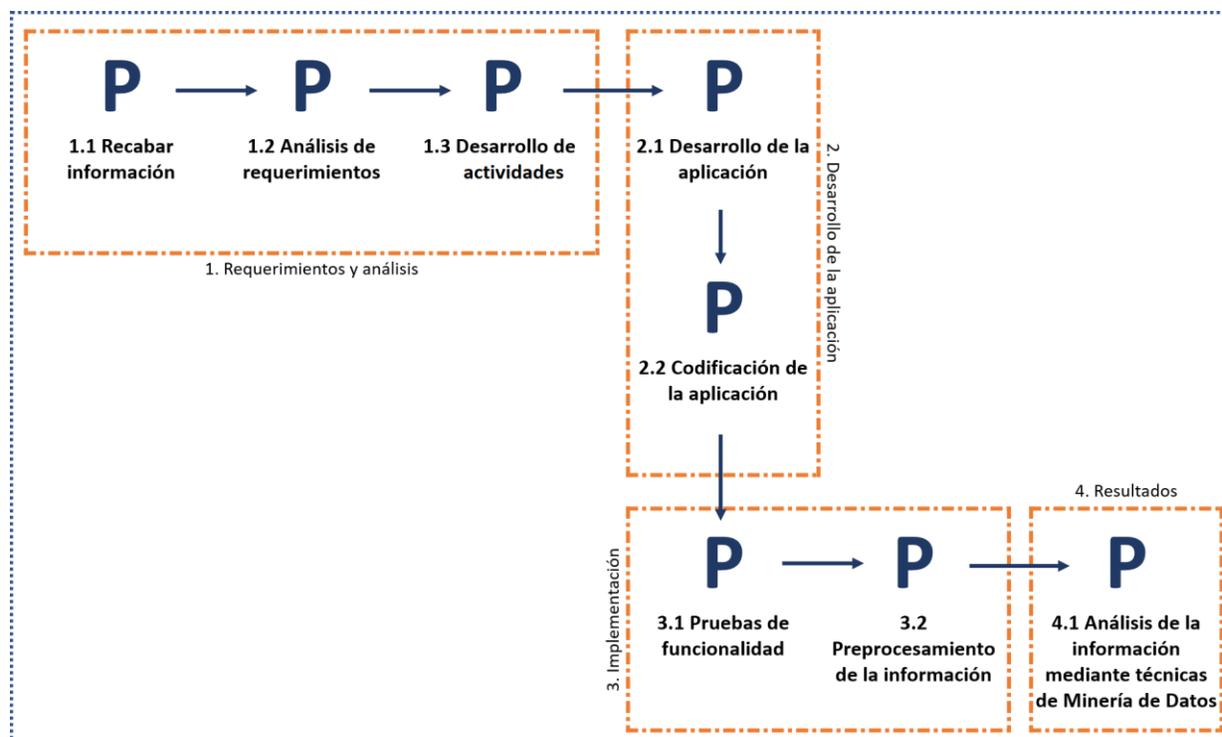


Diagrama 1 Etapas de desarrollo en el marco de trabajo Kuali-Beh

Por otra parte, el cronograma de desarrollo establecido se puede observar en la Tabla 2, el cual muestra el número de semanas (S) empleadas para cada una de las etapas de desarrollo. El desarrollo del sistema tutor afectivo se llevó a cabo en un periodo de 6 meses en los que se consiguió cubrir todas las etapas de desarrollo.

Tabla 2 Cronograma de etapas de desarrollo

Actividad	Mes 1				Mes 2				Mes 3				Mes 4				Mes 5				Mes 6			
	S1	S2	S3	S4																				
Análisis de requerimientos	■	■	■	■																				
Diseño de casos de uso y secuencial para cada módulo					■	■	■	■																
Diseño de interfaces									■	■	■	■												
Codificación de interfaces y módulos													■	■	■	■	■	■	■	■				
Pruebas y correcciones																					■	■	■	■
Implementación																								■

## 4.2 Especificación de requerimientos

A continuación, se describen los requerimientos del sistema mediante diagramas de casos de uso y su descripción.

### 4.2.1 Diagramas de caso de uso

En el sistema existen 5 actores diferentes que interactúan con él, de los cuales Docente, Investigador y Alumno extienden del actor Usuario y Alumno experimental extiende de Alumno como se muestra en el Diagrama 2.

El Diagrama 3 muestra de manera global cuales casos de uso realiza cada actor además de las relaciones de inclusión y extensión.

El actor usuario solo cuenta con el caso de uso CU001 Iniciar Sesión y los actores Docente, Investigador, Alumno extienden la funcionalidad como se puede observar en el Diagrama 4.

El actor Alumno tiene las funciones de realizar pretest y el post-test que servirán únicamente como grupo de control sin que puedan realizar las actividades dentro del sistema, el usuario Alumno experimental también realiza las evaluaciones y este si es capaz de realizar las actividades donde podrá consultar ayuda a través de videos o mediante una explicación escrita. Cada actividad que realicen ambos actores descritos anteriormente, a través de la cámara serán analizadas sus emociones, esto se puede notar en el Diagrama 5 y el Diagrama 6.

En el Diagrama 7 se aprecian las acciones que realiza el docente en el sistema las cuales incluyen la administración de alumnos, asignación de actividades y visualización de los resultados.

El sistema al ser ocupado principalmente para la obtención de resultados de una investigación relacionada con el proceso de enseñanza-aprendizaje de las estructuras de datos, se incluye en el sistema al actor Investigador, el cual tendrá acceso a consultar los resultados y el análisis de emociones de los alumnos cuando hicieron uso de la aplicación como se puede visualizar en el Diagrama 8.

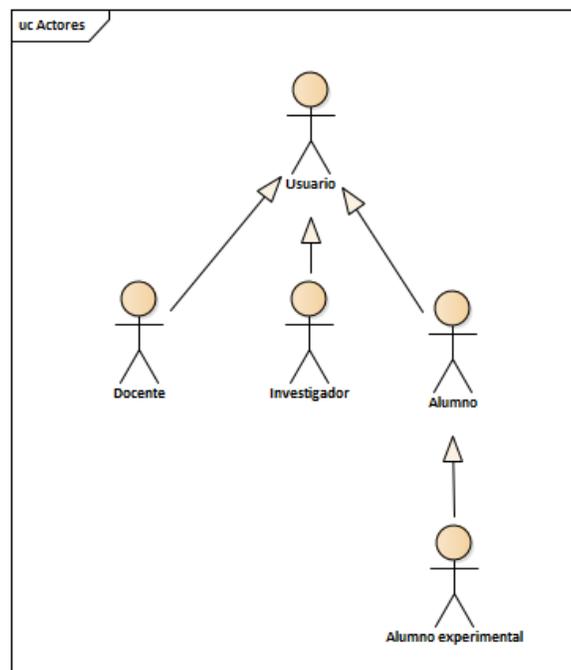


Diagrama 2 Diagrama de caso de uso Actores

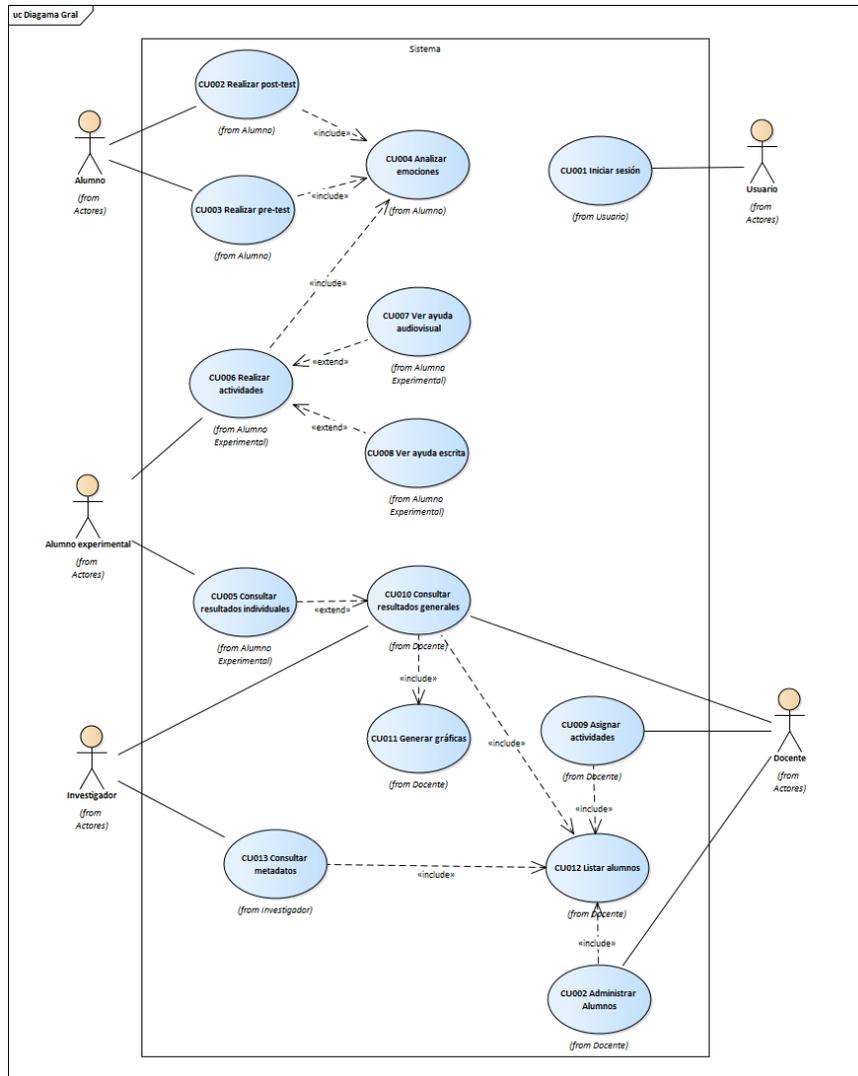


Diagrama 3 Diagrama general de caso de uso

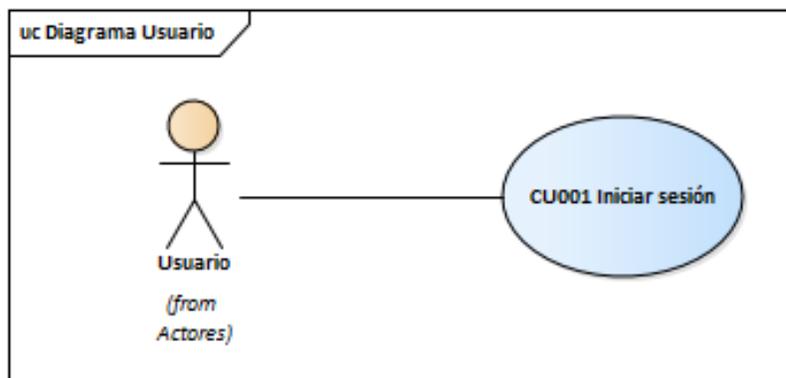


Diagrama 4 Diagrama de caso de uso Usuario

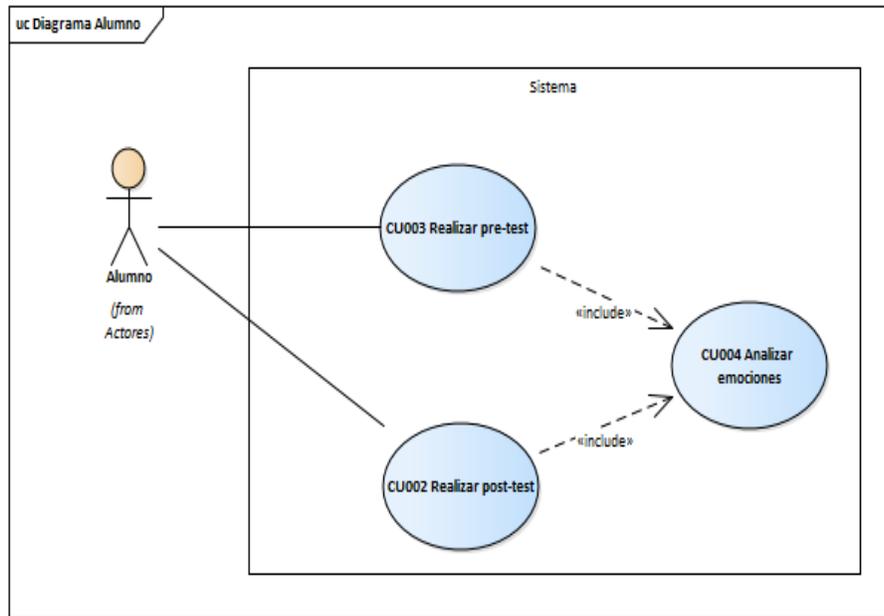


Diagrama 5 Diagrama de caso de uso Alumno

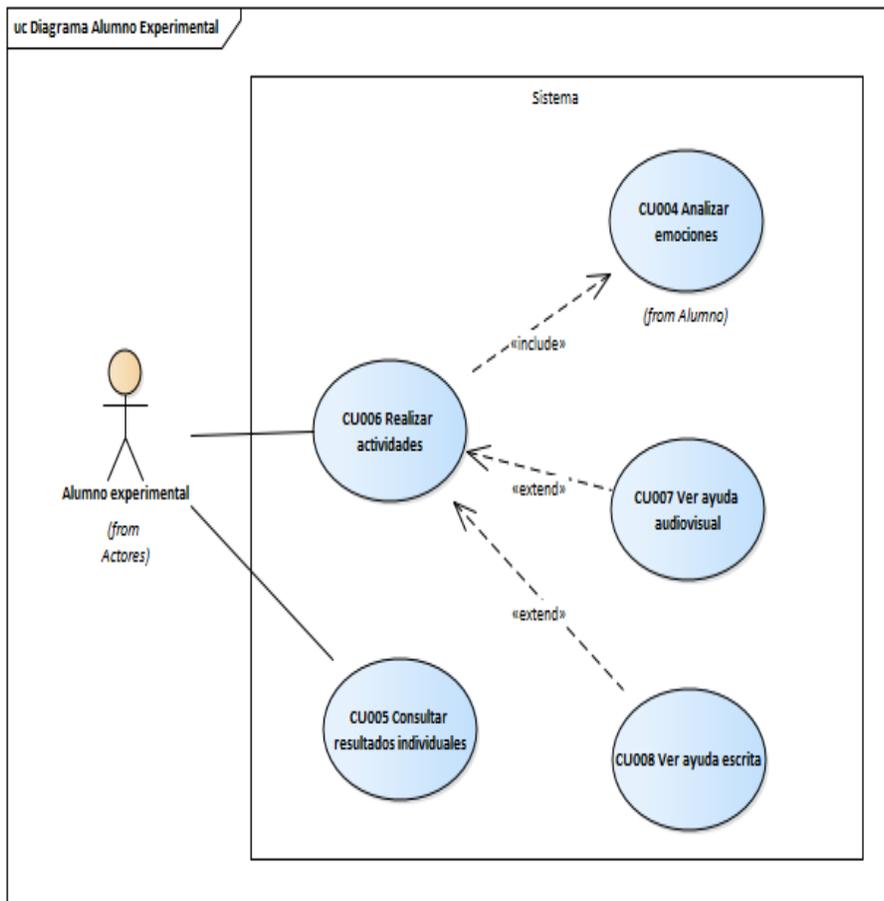


Diagrama 6 Diagrama de caso de uso Alumno experimental

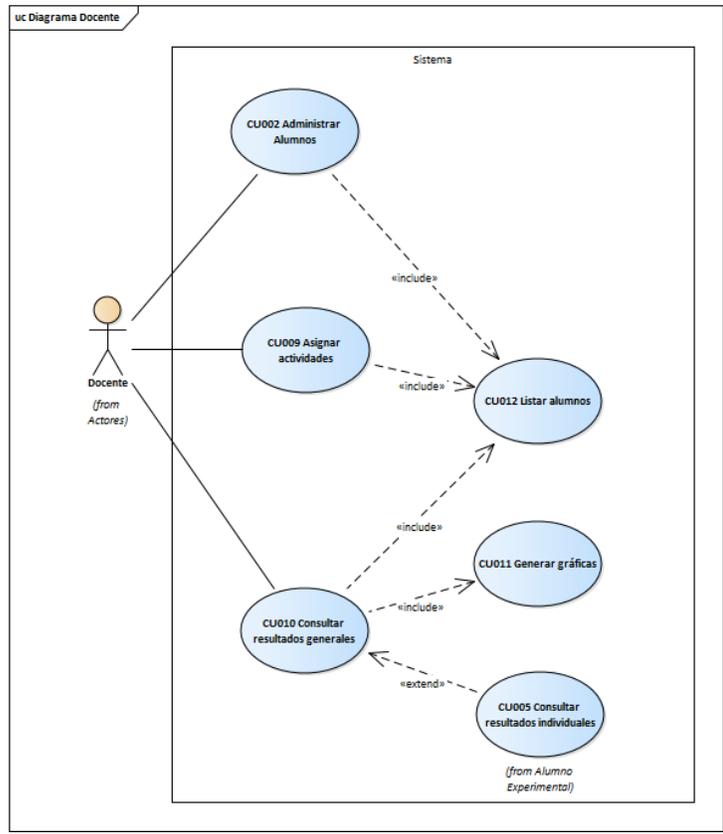


Diagrama 7 Diagrama de caso de uso Docente

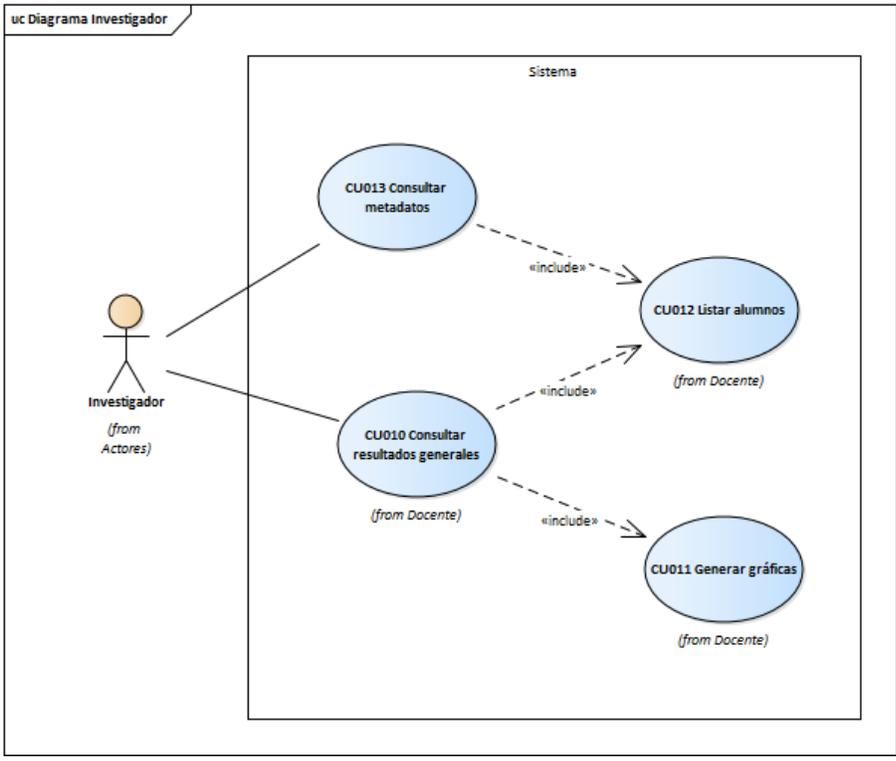


Diagrama 8 Diagrama de caso de uso Investigador

## 4.2.2 Descripción de casos de uso

A continuación, se muestran las descripciones de caso de uso, que se encuentran en los diagramas anteriores.

Tabla 3 Descripción de caso de uso autenticación de usuario

<b>Nombre:</b>	<b>Iniciar Sesión</b>	<b>Folio:</b>	<b>CU001</b>
<b>Actores:</b>	Usuario		
<b>Descripción:</b>	El Usuario se autentica para validar su identidad mediante correo y contraseña		
<b>Precondición</b>	<i>No aplica</i>		
<b>Poscondición</b>	El usuario es autenticado		
<b>Flujo Principal:</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>	
<b>1.1</b>	Ingresa a la aplicación	Carga la ventana de Login.	
<b>1.2</b>	Ingresa su correo	Se valida que el correo tenga un formato valido, en caso de que no ir al flujo alternativo CU001.a	
<b>1.3</b>	Ingresa su contraseña	Se valida que la contraseña tenga un formato valido, en caso de que no ir al flujo alternativo CU001.b	
<b>1.4</b>	Da un toque al botón de iniciar Sesión	Envía una petición al Back para corroborar que el usuario y contraseña son válidos, si todo es correcto ir a ventana principal, en caso de que no ir al flujo alternativo CU001.c, si es error del sistema ir al flujo alternativo CU001.d	
<b>1.5</b>	Fin del caso de uso		
<b>Flujo Alternativo CU001.a</b>			
<b>1.a.1</b>		Muestra en la parte inferior del input correo el mensaje "El correo no tiene el formato correcto".	

1.a.2	Corrige el correo con un formato valido.	Regresar al paso 1.2
<b>Flujo Alternativo CU001.b</b>		
1.b.1		Muestra en la parte inferior del input correo el mensaje “El correo no tiene el formato correcto”.
	Corrige el correo con un formato valido.	Regresar al paso 1.3
<b>Flujo Alternativo CU001.c</b>		
		Muestra un toast indicando que le usuario y contraseña son incorrectos.
	Vuelve a ingresar sus datos.	Regresa al paso 1.2
<b>Flujo Alternativo CU001.d</b>		
		Muestra un alert indicando el siguiente error “Error de conexión, inténtelo más tarde”.
	Fin del caso de uso	

Tabla 4 Descripción de caso de uso realización pretest

<b>Nombre:</b>	<b>Realizar Pretest</b>	<b>Folio:</b>	<b>CU002</b>
<b>Actores:</b>	Alumno		
<b>Descripción:</b>	El Alumno realiza el pretest para poder comenzar a realizar las actividades		
<b>Precondición</b>	El Alumno debe estar autenticado		
<b>Poscondición</b>	Los resultados del pretest del alumno son almacenados en el servidor		
<b>Flujo Principal:</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>	
2.1	Selección al icono de pretest de la ventana principal	Carga la ventana “Examen”	

2.2	Lee las instrucciones y da en el botón continuar	Carga las preguntas
2.3	Contesta la pregunta seleccionando las opciones que se le muestran y da en el botón siguiente.	Guarda la respuesta de la pregunta en el servidor, si existe otra pregunta repetir el punto 1.3 si no es así mostrar el mensaje de "Tus datos han sido guardados con éxito".
2.4	Da un toque en el botón de continuar	El sistema redirige a la página principal.
2.5	Fin del caso de uso	
<b>Flujo Alternativo CU003.a</b>		
<i>Sucede en cualquier momento</i>		
2.a.1		Muestra un alert indicando el siguiente error "Error de conexión, inténtelo más tarde". Y redirige a la ventana principal.
2.a.2	Fin del caso de uso	

Tabla 5 Descripción de caso de uso realización postest

<b>Nombre:</b>	Realizar Postest	Folio:	CU003
<b>Actores:</b>	Alumno		
<b>Descripción:</b>	El Alumno realiza el Postest para terminar el uso de la aplicación		
<b>Precondición</b>	El Alumno debe estar autenticado		
<b>Poscondición</b>	Los resultados del postest del alumno son almacenados en el servidor		
<b>Flujo Principal:</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>	
3.1	Selección al icono de Postest que se encuentra al final de la lista de actividades de la ventana principal	Carga la ventana "Examen"	

3.2	Lee las instrucciones y da en el botón continuar	Carga las preguntas
3.3	Contesta la pregunta seleccionando las opciones que se le muestran y da en el botón siguiente.	Guarda la respuesta de la pregunta en el servidor, si existe otra pregunta repetir el punto 1.3 si no es así mostrar el mensaje de “Tus datos han sido guardados con éxito”.
3.4	Da un toque en el botón de continuar	El sistema redirige a la página principal.
3.5	Fin del caso de uso	
<b>Flujo Alternativo CU002.a</b>		
<i>Sucede en cualquier momento</i>		
3.a.1		Muestra un alert indicando el siguiente error “Error de conexión, inténtelo más tarde”. Y redirige a la ventana principal.
3.a.2	Fin del caso de uso	

Tabla 6 Descripción de caso de uso análisis de expresiones faciales

<b>Nombre:</b>	<b>Analizar emociones</b>	<b>Folio:</b>	<b>CU004</b>
<b>Actores:</b>	Alumno		
<b>Descripción:</b>	Se analiza el rostro del alumno para identificar sus emociones al realizar las actividades		
<b>Precondición</b>	El Alumno debe estar autenticado		
<b>Poscondición</b>	Las emociones obtenidas son convertidas en objeto JSON y almacenadas en base de datos.		
<b>Flujo Principal:</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>	
4.1	Inicia un caso de uso predecesor (Consultar el diagrama de casos de uso)	El sistema activa la cámara y comienza a sacar fotografías del Alumno casa segundo, estas fotografías son comprimidas y mandadas al back para su procesamiento, posteriormente es almacenado el objeto JSON de las	

		emociones. Si no se tienen los permisos para activar la cámara ir al flujo alternativo CU004.a
<b>4.2</b>	El alumno finaliza la actividad	El sistema deja de tomar fotografías.
<b>4.3</b>	Fin del caso de uso	
<b>Flujo Alternativo CU004.a</b>		
<b>4.a.1</b>		Muestra un alert indicando que se requiere el permiso de la cámara para poder continuar.
<b>4.a.2</b>	El Alumno selecciona la opción de Activar, si no ir a Flujo Alternativo CU004.b	Continúa en el paso 3.1
<b>Flujo Alternativo CU004.b</b>		
<b>4.b.1</b>		El sistema muestra un alert con un botón de aceptar indicando que se requieren los permisos para realizar la actividad
<b>4.b.2</b>	El usuario toca el botón de aceptar	Regresa a la página principal.
<b>4.b.4</b>	Fin del caso de uso	

Tabla 7 Descripción de caso de uso realizar actividad

<b>Nombre:</b>	<b>Realizar actividades</b>	<b>Folio:</b>	<b>CU006</b>
<b>Actores:</b>	Alumno experimental		
<b>Descripción:</b>	Realiza una actividad		
<b>Precondición</b>	El Alumno experimental debe estar autenticado		
<b>Poscondición</b>	Los resultados obtenidos son almacenados en base de datos		
<b>Flujo Principal:</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>	

6.1	Inicia una actividad dando un toque a cualquier icono de la página Principal	El sistema muestra la página actividades donde se carga el contenido de la actividad.
6.2	Lee el ejercicio y da en continuar	Si todavía hay información repetir el paso 6.2
6.3		Si la información se ha acabado mostrar el ejercicio a resolver.
6.4	Analiza la actividad para resolverla, si requiere ayuda da un toque en Ayuda Escrita (Ir al caso de uso CU007 Ver ayuda escrita) o en Mostrar Ayuda Audiovisual (Ir al caso de uso CU008 Mostrar ayuda Audiovisual)	
6.5	Da un toque en finalizar	El sistema almacena su respuesta y valida que sea correcta, posteriormente manda un mensaje con sus resultados
6.6	Fin del caso de uso	

Tabla 8 Descripción de caso de uso consultar ayuda escrita

<b>Nombre:</b>	<b>Ver ayuda escrita</b>	<b>Folio:</b>	<b>CU007</b>
<b>Actores:</b>	Alumno experimental		
<b>Descripción:</b>	Consulta la ayuda para realizar la actividad		
<b>Precondición</b>	El Alumno experimental debe estar autenticado		
<b>Poscondición</b>	No aplica		
<b>Flujo Principal:</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>	
7.1	Da un toque en Ver ayuda escrita	El sistema abre un modal con la explicación resumida del ejercicio y el estatus de acción realizada cambia a "consultando ayuda escrita"	

7.2	Lee la ayuda y da en continuar	Si existe mas contenido se repite el paso 7.2
7.3	Da un toque en cerrar	El modal se cierra.
7.4	Fin del caso de uso	
<b>Flujo Alternativo CU007.a</b>		
<b>Sucede a cualquier momento</b>		
7.a.1	Da u toque en cerrar sin haber terminado la ayuda	El sistema muestra un alert preguntando si desea cerrar o continuar si le da en continuar regresar al punto en 7.2 en caso contrario cerrar el modal
7.a.2	Fin del caso de uso	

Tabla 9 Descripción de caso de uso consultar ayuda audiovisual

<b>Nombre:</b>	<b>Ver ayuda audiovisual</b>	<b>Folio:</b>	<b>CU008</b>
<b>Actores:</b>	Alumno experimental		
<b>Descripción:</b>	Consulta la ayuda para realizar la actividad		
<b>Precondición</b>	El Alumno experimental debe estar autenticado		
<b>Poscondición</b>	No aplica		
<b>Flujo Principal:</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>	
8.1	Da un toque en Ver ayuda audiovisual	El sistema abre un modal con un video explicativo y el estatus de acción realizada cambia a “consultando ayuda audiovisual”	
8.2	Visualiza y escucha la ayuda	El video se reproduce hasta terminar	
8.3	Da un toque en cerrar	El modal se cierra.	
8.4	Fin del caso de uso		
<b>Flujo Alternativo CU008.a</b>			
<b>Sucede a cualquier momento</b>			

<b>8.a.1</b>	Da u toque en cerrar sin haber terminado la ayuda	El sistema muestra un alert preguntando si desea cerrar o continuar si le da en continuar regresar al punto en 7.2 en caso contrario cerrar el modal
<b>8.a.2</b>	Fin del caso de uso	

Tabla 10 Descripción de caso de uso activar o desactivar actividad

<b>Nombre:</b>	<b>Asignar actividades</b>	<b>Folio:</b>	<b>CU009</b>
<b>Actores:</b>	Docente		
<b>Descripción:</b>	Activa o desactiva las actividades a los estudiantes		
<b>Precondición</b>	El docente tiene que estar autenticado		
<b>Poscondición</b>	No aplica		
<b>Flujo Principal:</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>	
<b>9.1</b>	Da un toque en Administrar actividades que se encuentra en el menu principal del docente	Muestra el menú de actividades	
<b>9.2</b>	Visualiza las actividades y selecciona cuales va a activar mediante un toggle que se encuentra a un costado de cada actividad	Si el toggle se alguna actividad cambia se guarda en base de datos el estatus (activo o inactivo) de cada actividad.	
<b>9.3</b>	Da un toque en regresar	La ventana se cierra y regresa al menú principal.	
<b>9.4</b>	Fin del caso de uso		

## 4.3 Diseño conceptual

A continuación, se muestra el Diagrama 9, que muestra el dominio del sistema.

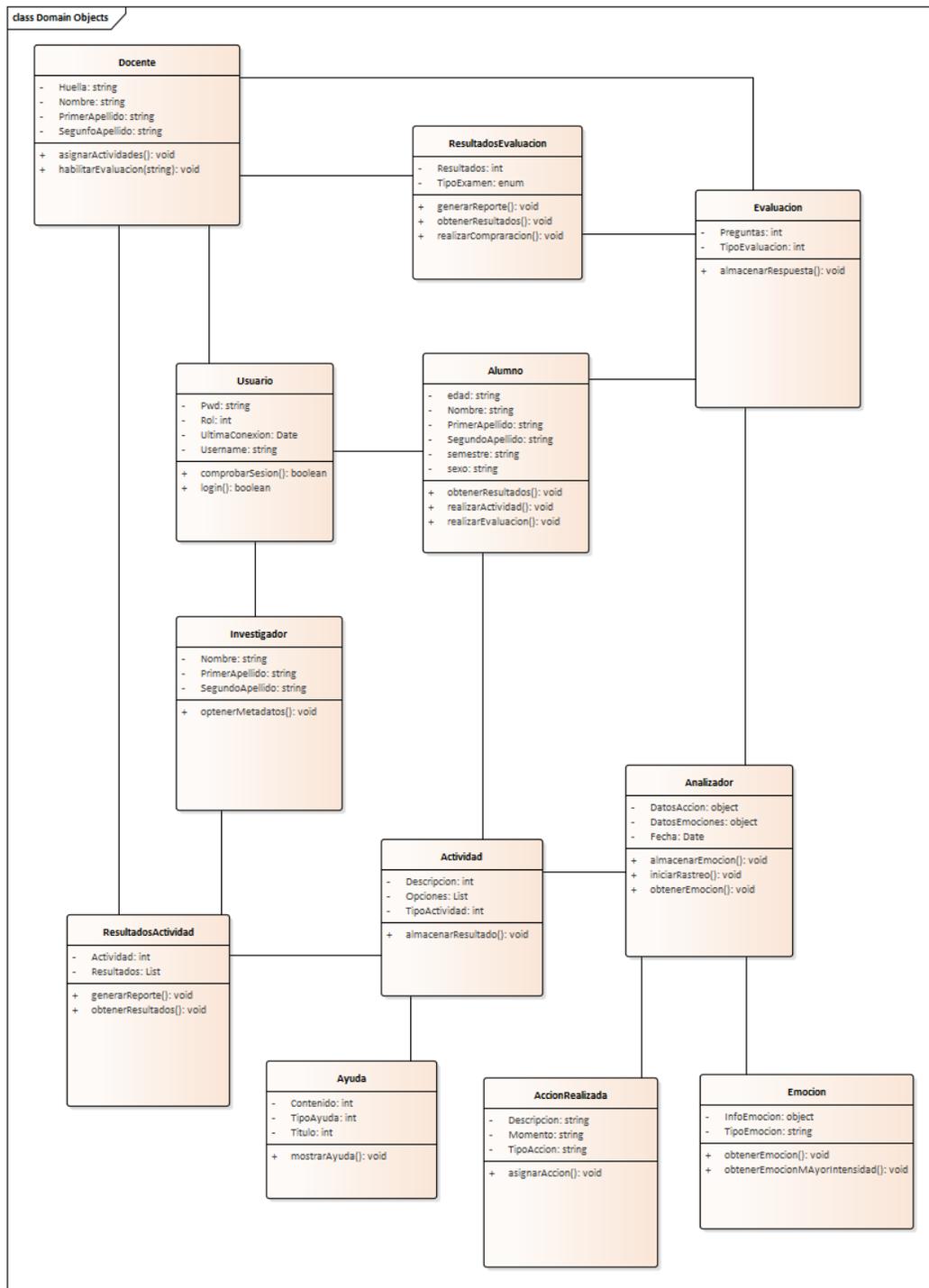


Diagrama 9 Diagrama de dominio

## 4.4 Diseño navegacional

A continuación, se muestra el Diagrama 10 que muestra la estructura navegacional del sistema.

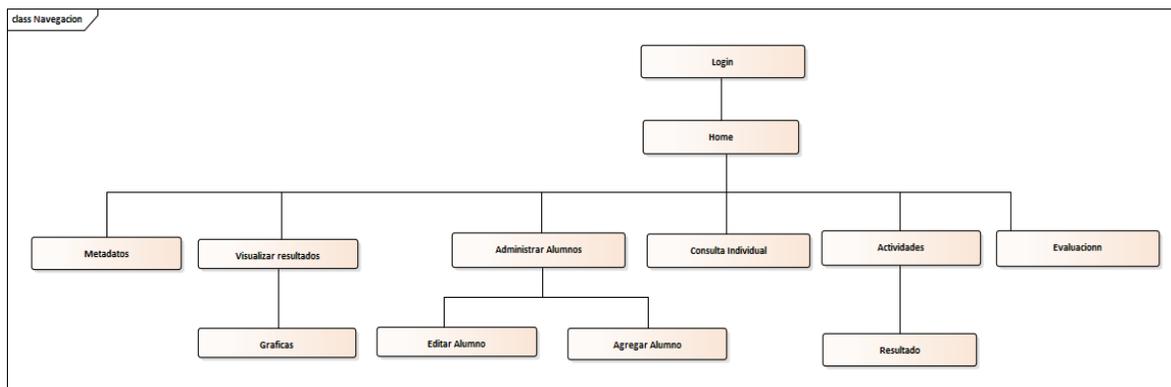


Diagrama 10 Diagrama navegacional

## 4.5 Diseño de Interfaces

El diseño de interfaces consiste en realizar un diseño previo de lo que será la programación de las interfaces o vistas visuales del sistema mejor conocido como Mockups. Se realizaron 12 diseños, que se presentan en las siguientes ilustraciones.

En la Ilustración 1, Ilustración 2, Ilustración 3 e Ilustración 4 se observan los diseños correspondientes al inicio de sesión, pantalla principal y perfil del alumno. Por otra parte, en la Ilustración 5, Ilustración 6, Ilustración 7 e Ilustración 8 se puede observar las secciones correspondientes a la administración de las actividades por parte del docente, donde puede observar los resultados obtenidos por los estudiantes en cada una de las actividades. Finalmente, en la Ilustración 9, Ilustración 10, Ilustración 11 e Ilustración 12 estas vistas permitirán al docente obtener datos estadísticos respecto a la actividad del estudiante dentro de la aplicación, es decir lo referente a las emociones presentadas, así como a los resultados obtenidos por el estudiante en cada una de las actividades propuestas en la aplicación.

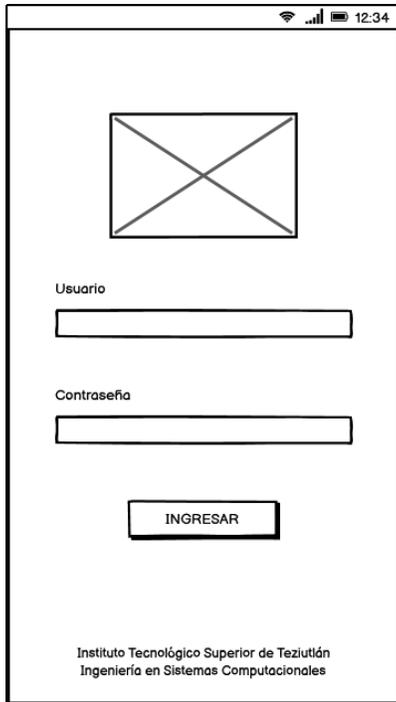


Ilustración 1 Inicio se sesión

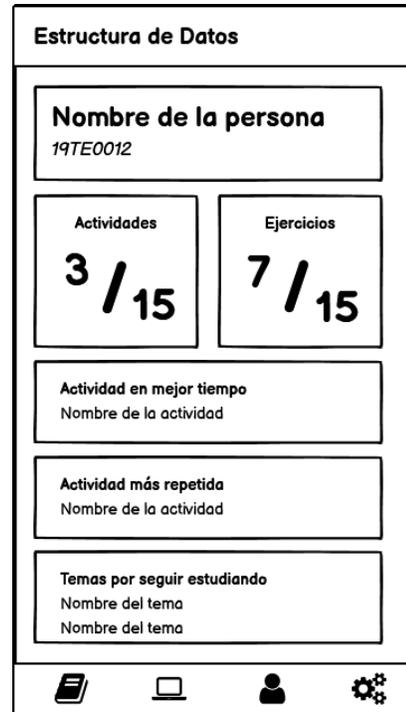


Ilustración 3 Vista de perfil (Alumno)

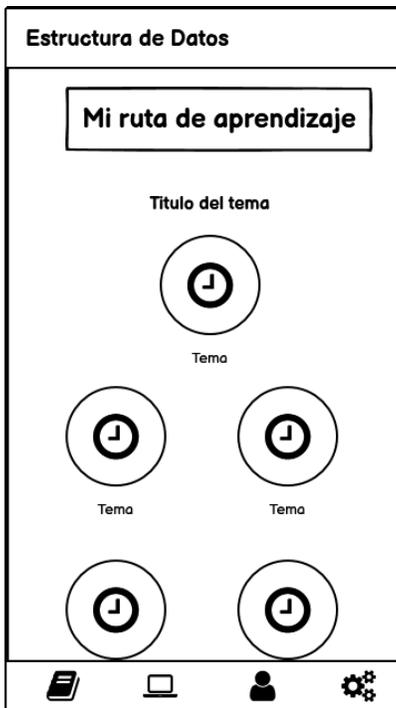


Ilustración 2 Vista de actividades (Alumno)

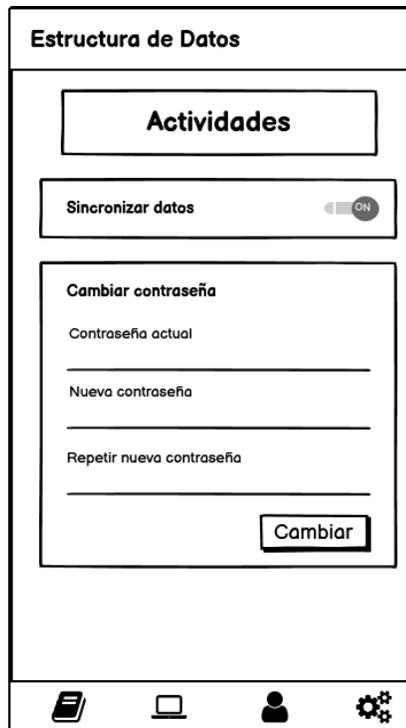


Ilustración 4 Editar perfil (Alumno)

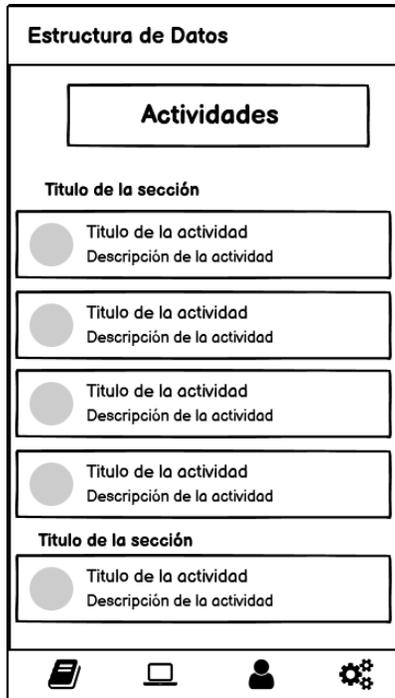


Ilustración 5 Vista de ejercicios de programación (Alumno)

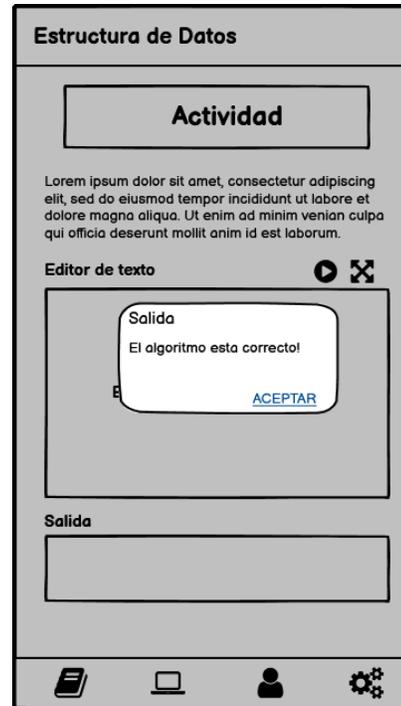


Ilustración 7 Vista de resolver ejercicios mostrando una Alerta (Alumno)

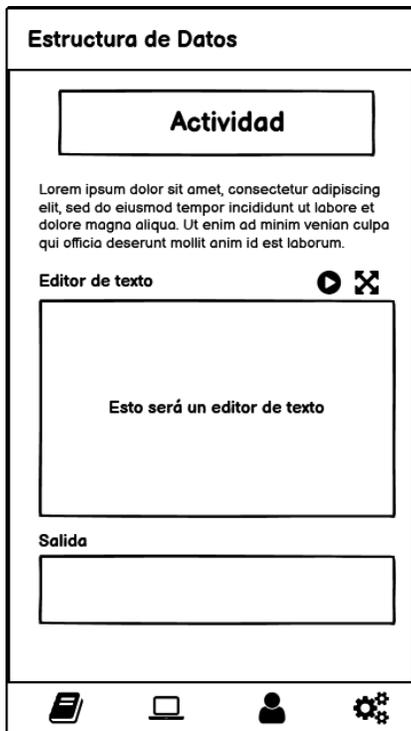


Ilustración 6 Vista de resolver ejercicios (Alumno)

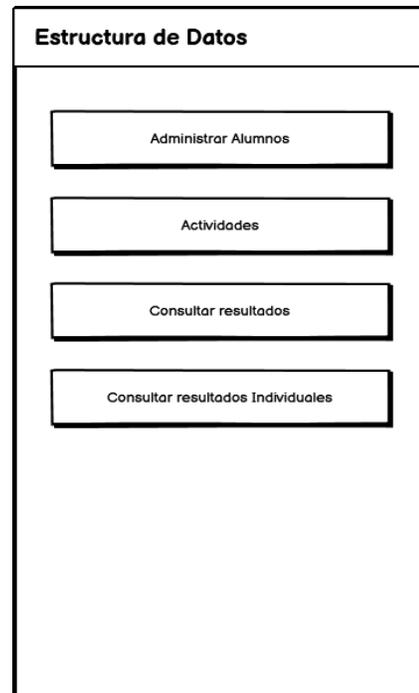


Ilustración 8 Vista principal (Docente)

**Estructura de Datos**

**Estudiantes**

Nombre

---

Numero de control

---

Correo

---

**AGREGAR**

Nombre^	Activo	Eliminar
Giacomo Guillizzoni Founder & CEO	<input checked="" type="checkbox"/>	D
Marco Botton Tuttofare	<input checked="" type="checkbox"/>	D
Mariah Maclachlan Better Half	<input checked="" type="checkbox"/>	D
Valerie Liberty Head Chef	<input checked="" type="checkbox"/>	D

Ilustración 9 Vista de Administrar alumnos (Docente)

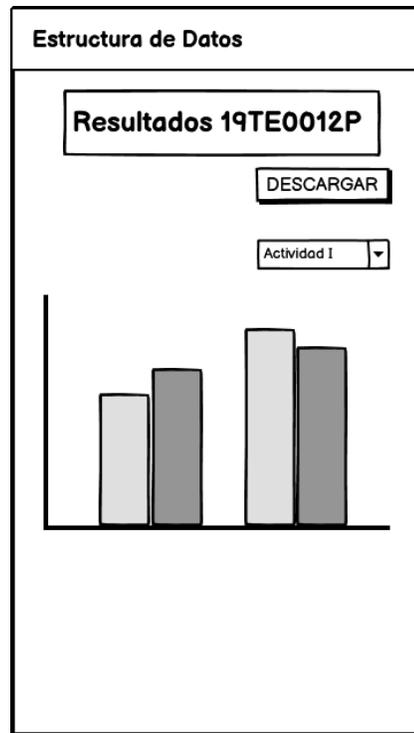


Ilustración 11 Vista de consultar datos individuales (Docente)

**Estructura de Datos**

**Resultados**

**DESCARGAR**

Nombre^	Activo	Resultados
Giacomo Guillizzoni Founder & CEO	<input checked="" type="checkbox"/>	D
Marco Botton Tuttofare	<input checked="" type="checkbox"/>	D
Mariah Maclachlan Better Half	<input checked="" type="checkbox"/>	D
Valerie Liberty Head Chef	<input checked="" type="checkbox"/>	D

Ilustración 10 Vista de consultar resultados (Docente)vez

**Estructura de Datos**

**Actividades**

Tema ▾  Todos

● Título de la actividad  ON

Descripción de la actividad

● Título de la actividad  ON

Descripción de la actividad

● Título de la actividad  ON

Descripción de la actividad

● Título de la actividad  ON

Descripción de la actividad

● Título de la actividad  ON

Descripción de la actividad

Ilustración 12 Vista de actividades (Docente)

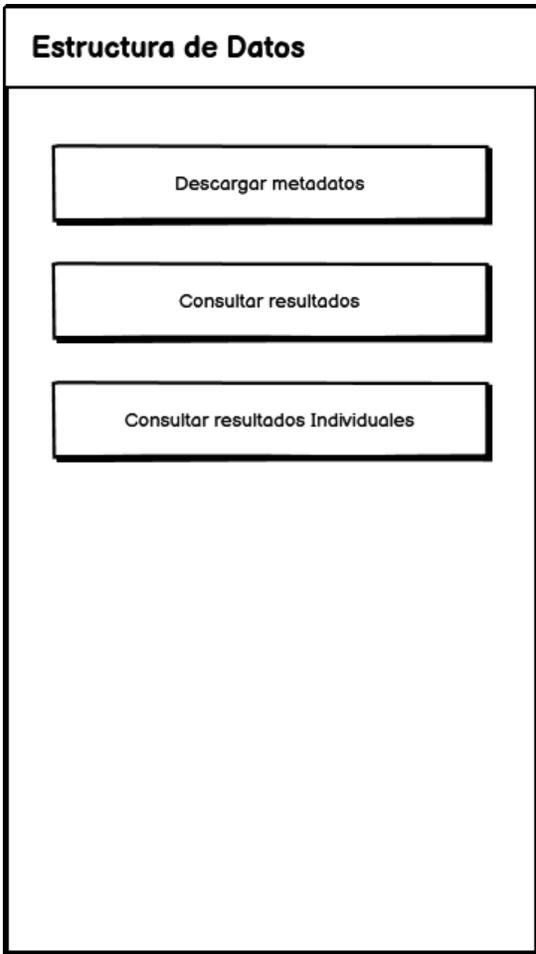


Ilustración 13 Vista principal (Docente)

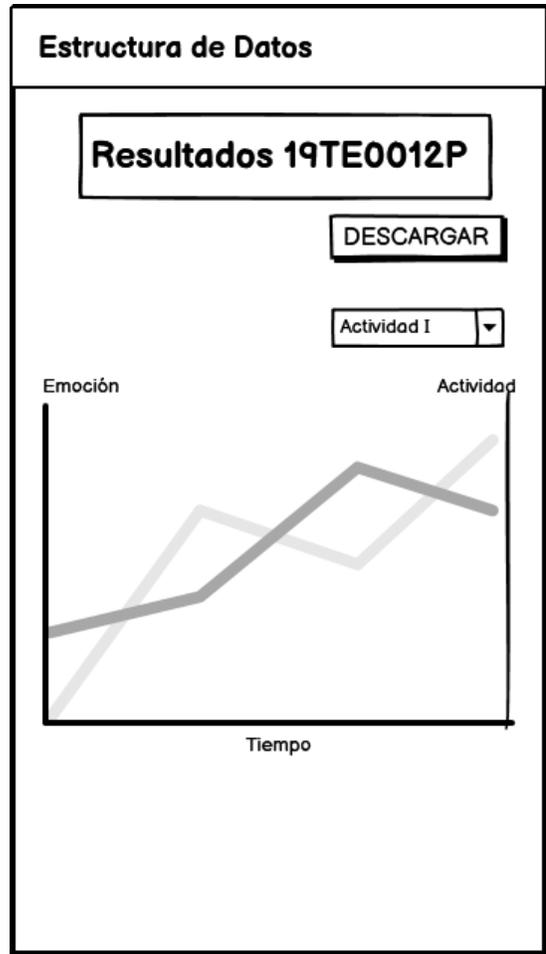


Ilustración 14 Vista de descarga de metadatos (docente)

## 4.5 Pruebas e implementación

Una vez realizada la codificación de las interfaces considerando los mockups, se obtuvieron las pantallas finales presentadas en la aplicación móvil y se realizaron las pruebas correspondientes de funcionalidad.

### 4.5.1 Pruebas

Se realizan pruebas unitarias sobre los módulos principales como el inicio de sesión (Tabla 11), el almacenamiento y funcionamiento del pretest y postest (Tabla 12 y Tabla 13), además del funcionamiento de los algoritmos de la aplicación (Tabla 14) utilizando los siguientes dispositivos:

- Dispositivo físico: Xiaomi Mi 9 SE, procesador Octa-core 2.3Ghz 6Gb de RAM, S.O. Android 10
- Dispositivo físico: Xiaomi MI 8 lite, Procesador Snapdragon 660 a 1.8 GHz, 4 de RAM, S.O. Android 8.1 Oreo con MIUI 10.
- Dispositivo virtual: Pixel 3A, Procesador 4 núcleos, 2Gb de RAM, S.O. Android 11.

Las pruebas realizadas concluyeron que el sistema está funcionando correctamente.

Tabla 11 Prueba unitaria Inicio de sesión

#### **Código PA001**

<i>Caso de prueba</i>	Inicio de sesión
<i>Descripción</i>	Validar que el inicio de sesión valide correctamente el usuario y contraseña, en caso de no ser validos o existir un problema de conexión, mostrar una alerta con el mensaje del error. Los tres posibles casos son: <ul style="list-style-type: none"><li>• Usuario y contraseña correcta</li><li>• Usuario y/o contraseña incorrecta</li></ul>

	<ul style="list-style-type: none"> <li>• Problemas de conexión</li> </ul>
<i>Fecha</i>	21 de marzo de 2021
<i>Acciones de entrada</i>	Ingresar usuario y contraseña y presionar enter o clic en el botón INICIAR SESIÓN.
<i>Resultado esperado</i>	Si el inicio de sesión es exitoso mostrar la ventana principal. En caso contrario mostrar una alerta con el mensaje de error.
<b>Información para el seguimiento</b>	
<i>Resultado obtenido</i>	Muestra mensajes de error correctamente y navegación correcta.
<i>Estado</i>	Exitoso
<i>Ultima fecha de estado</i>	28 de marzo de 2021
<i>Observaciones</i>	N/A

Tabla 12 Prueba unitaria Pretest

**Código PA002**

<i>Cado de prueba</i>	Pretest
<i>Descripción</i>	Validar que el pretest se muestre de manera correcta con las preguntas en orden y almacene las respuestas de manera correcta
<i>Fecha</i>	21 de marzo de 2021
<i>Acciones de entrada</i>	Presionar el botón de pretest y contestar las preguntas (algunas correctas y otras incorrectas) y finalizar el cuestionario

<i>Resultado esperado</i>	En base de datos se almacenan todas las respuestas con los valores seleccionados.
<b>Información para el seguimiento</b>	
<i>Resultado obtenido</i>	Las respuestas se almacenan correctamente.
<i>Estado</i>	Exitoso
<i>Ultima fecha de estado</i>	28 de marzo de 2021
<i>Observaciones</i>	N/A

Tabla 13 Prueba unitaria posttest

**Código PA003**

<i>Cado de prueba</i>	Postest
<i>Descripción</i>	Validar que el postest se muestre de manera correcta con las preguntas en orden y almacene las respuestas de manera correcta
<i>Fecha</i>	21 de marzo de 2021
<i>Acciones de entrada</i>	Presionar el botón de postest y contestar las preguntas (algunas correctas y otras incorrectas) y finalizar el cuestionario
<i>Resultado esperado</i>	En base de datos se almacenan todas las respuestas con los valores seleccionados.
<b>Información para el seguimiento</b>	
<i>Resultado obtenido</i>	Las respuestas se almacenan correctamente.
<i>Estado</i>	Exitoso

<i>Ultima fecha de estado</i>	28 de marzo de 2021
<i>Observaciones</i>	N/A

Tabla 14 Prueba unitaria ejercicio

**Código PA004**

<i>Cado de prueba</i>	Ejercicios de algoritmos validos
<i>Descripción</i>	Validar que las respuestas de los algoritmos enviados al servidor sean evaluadas correctamente
<i>Fecha</i>	21 de marzo de 2021
<i>Acciones de entrada</i>	Seleccionar un ejercicio de ordenamiento y escribir el código de la solución, presionar el botón de “probar” y validar que el resultado sea el correcto
<i>Resultado esperado</i>	El servidor captura la respuesta y almacena en el servidor el tiempo de ejecución y si el resultado es correcto.

**Información para el seguimiento**

<i>Resultado obtenido</i>	El tiempo de ejecución y el resultado de correcto o incorrecto se almacena en base de datos
<i>Estado</i>	Exitoso
<i>Ultima fecha de estado</i>	28 de marzo de 2021
<i>Observaciones</i>	N/A

## 4.5.2 Implementación

Para la implementación se presentó a usuario una aplicación móvil con la funcionalidad explicada en apartados anteriores. A continuación, se describen las pantallas implementadas.

La aplicación móvil funciona como sistema tutor afectivo presentado una interfaz interactiva y atractiva para el usuario.

La Ilustración 15 muestra las pantallas referentes al inicio de sesión para un usuario y la página principal donde se muestran las actividades que el estudiante puede realizar para practicar estructuras de datos.

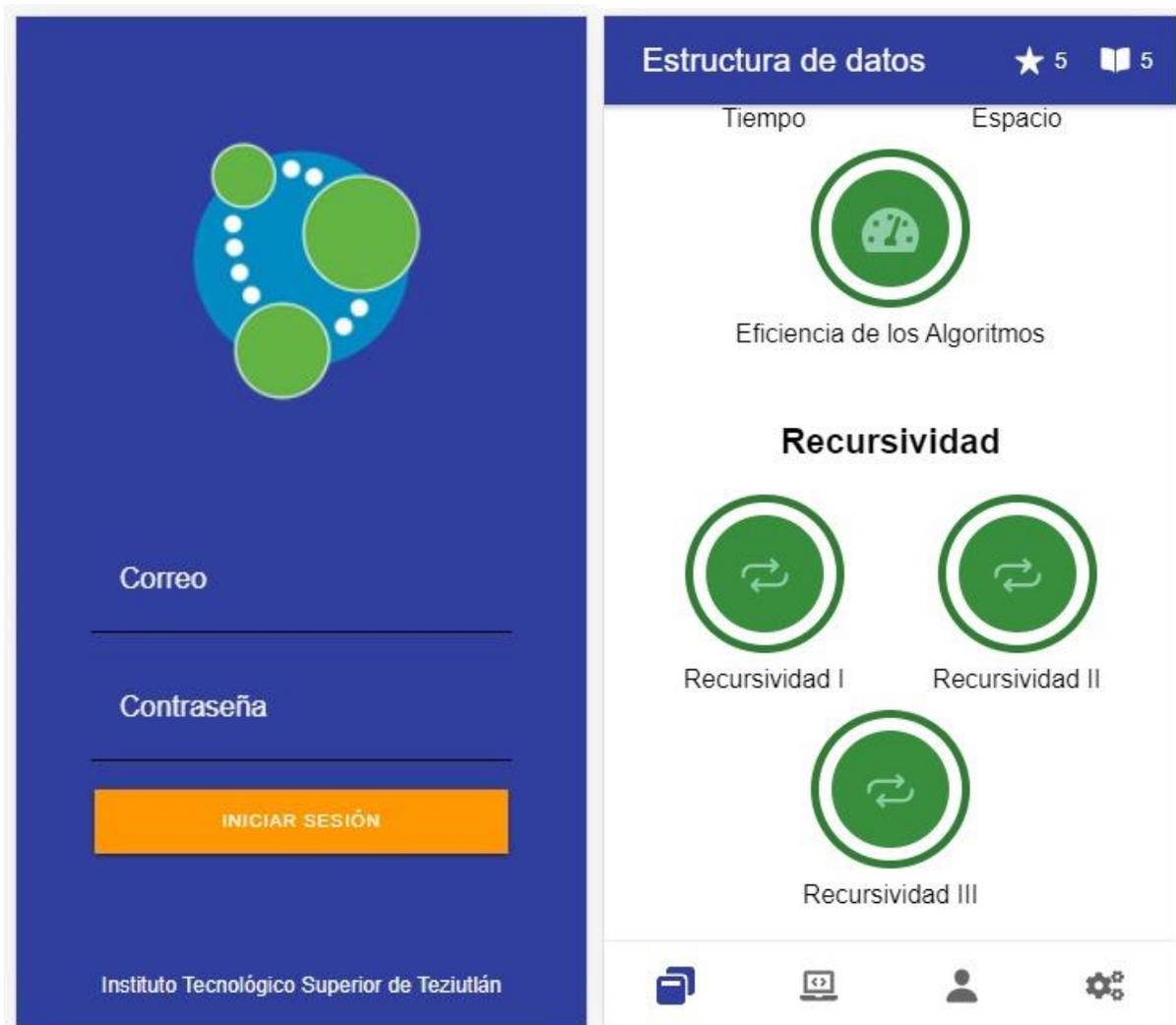


Ilustración 15 Página de logueo y principal

Las pantallas que se muestra en la Ilustración 16, corresponden al inicio de las actividades que puede realizar el estudiante, como primera actividad se muestra el pretest que sirve como diagnostico para conocer el nivel del conocimiento que posee el estudiante y cuáles son las áreas de estructura de datos que debe reforzar. Además, se muestra el apartado de configuración en donde el estudiante puede configurar su perfil y contraseña.

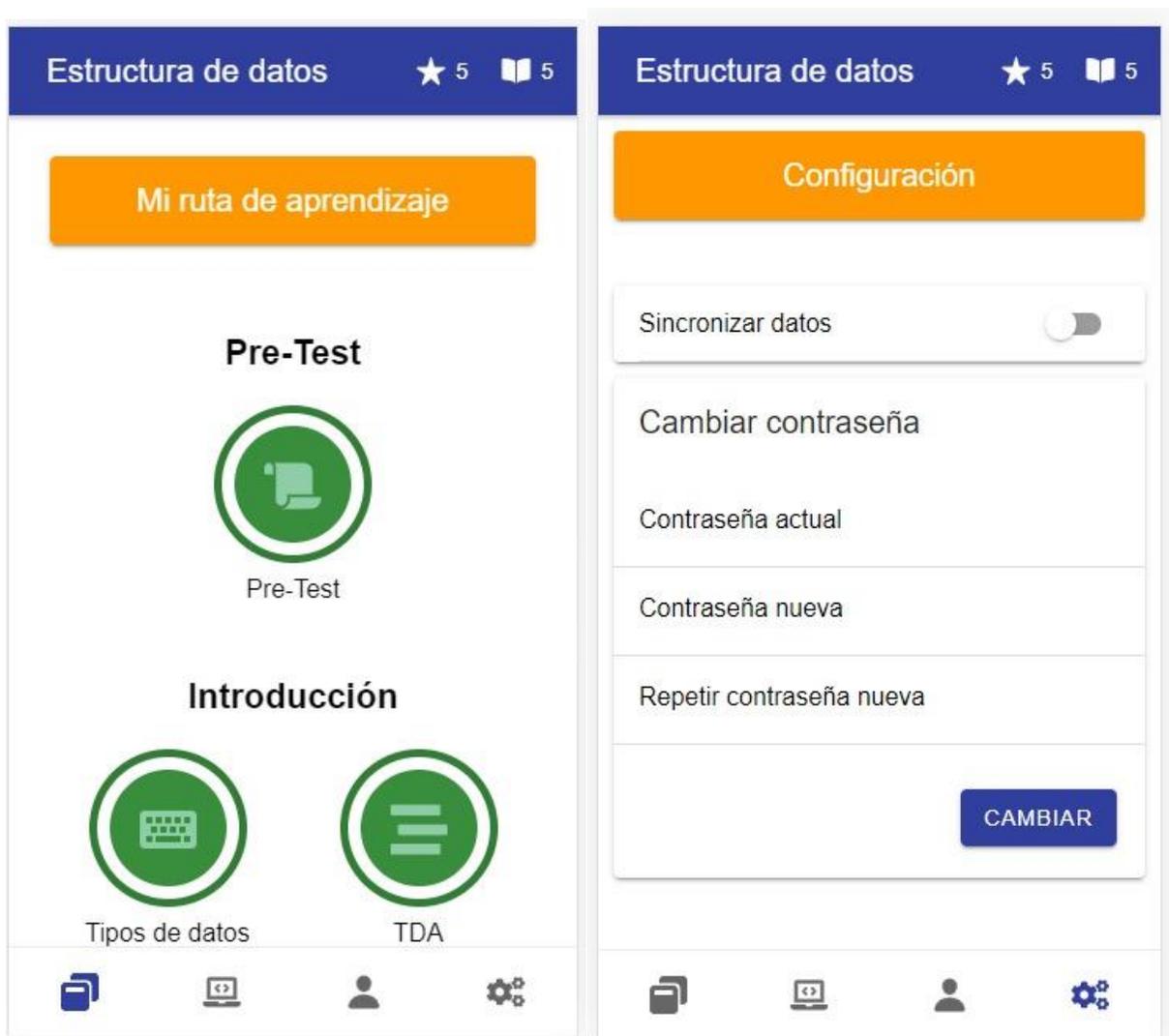


Ilustración 16 Página principal y configuración

En la Ilustración 17 y la Ilustración 18 se muestran las pantallas correspondientes al pretest donde se presentan las instrucciones generales al estudiante, para que pueda responder tranquilamente sin presión alguna y contestar solo aquellas preguntas de las que tenga conocimiento previo. Se muestra también un ejemplo del tipo de preguntas que aparece en la aplicación, estas son de tipo selección múltiple, con una o varias respuestas.

The image shows two side-by-side screenshots of a 'Pre Test' interface. Both have a blue header with the text 'Pre Test'.

The left screenshot contains the following text: 'Contesta las preguntas que se resenan a continuación si no sabes la respuesta de alguna pregunta puedes dejarla en blanco, no te preocupes solo es un examen diagnostico para saber cual es tu conocimiento previo sobre las estructuras de datos'. Below this text is a blue button labeled 'SIGUIENTE'.

The right screenshot contains the text 'Son ejemplos de estructuras de datos' followed by a list of items with checkboxes: 'Pila' (unchecked), 'String' (checked), 'Grafo' (unchecked), 'Lista enlazada' (checked), 'Condicionales' (unchecked), and 'Clases' (unchecked). At the bottom of this list is a blue button labeled 'SIGUIENTE'.

Ilustración 17 Pantalla de Pretest

The image shows a single screenshot of a 'Pre Test' interface with a blue header labeled 'Pre Test'. The main text reads: 'Es un modelo donde los datos estan organizados, manejados y almacenados de manera eficiente para su acceso y modificación de los datos'. Below this text is a list of four options, each with a radio button: 'Estructura de datos', 'Algoritmo', 'Arbol', and 'Clases Abstractas'. At the bottom of the list is a blue button labeled 'SIGUIENTE'.

Ilustración 18 Pregunta Pretest

En cuanto a las actividades diseñadas para el aprendizaje de estructura de datos algunos ejercicios más destacados corresponden al diseño algorítmico, donde se presenta un tipo editor de código que les muestra a los estudiantes el código resultante del algoritmo que ellos hayan diseñado. En la Ilustración 19 y la Ilustración 20 se observa las pantallas que corresponden a las actividades de algoritmos.

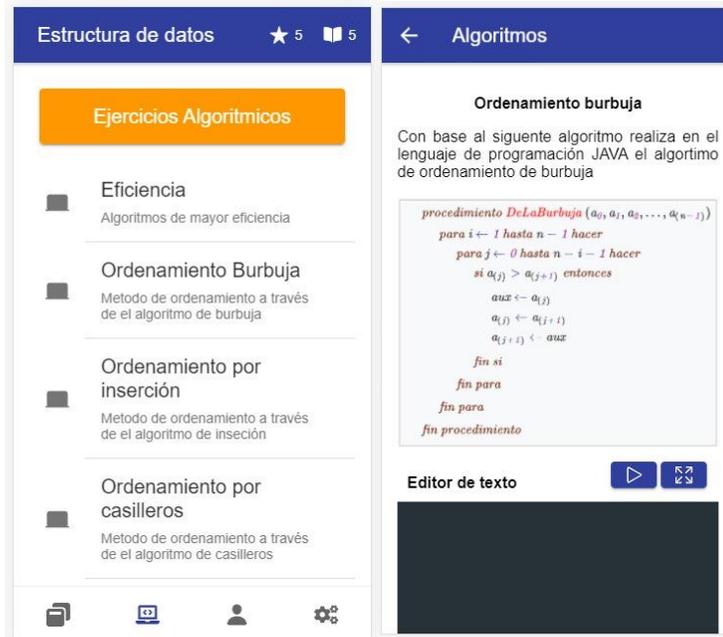


Ilustración 19 Pantalla de actividades Algoritmos



Ilustración 20 Editor de código en actividad de algoritmos

Finalmente, una pantalla importante para el estudiante es la que se muestra en la Ilustración 21, ya que le permite visualizar cuales son las actividades que ha realizado y cuales han sido sus resultados. Además, proporciona una recomendación de los temas que es necesario seguir repasando para que este pueda reforzar adecuadamente su aprendizaje.



Ilustración 21 Pantalla de retroalimentación

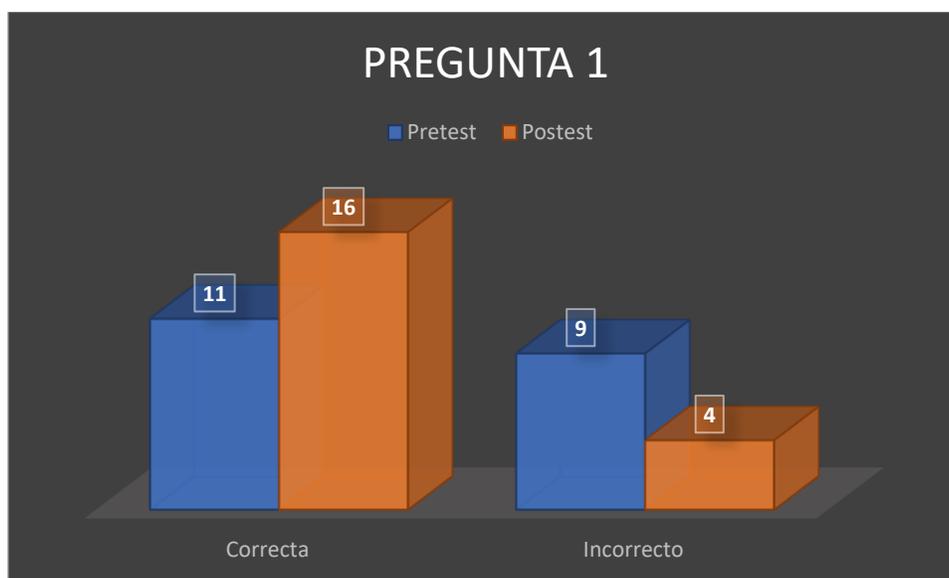
## CAPITULO V. RESULTADOS Y DISCUSIÓN

En este apartado se describen los resultados obtenidos en las diferentes etapas de la investigación, en el grupo de investigación que consta de 20 estudiantes de cuarto semestre de la carrera de Ingeniería en Sistemas Computacionales.

### 5.1 Resultados Pretest y Postest

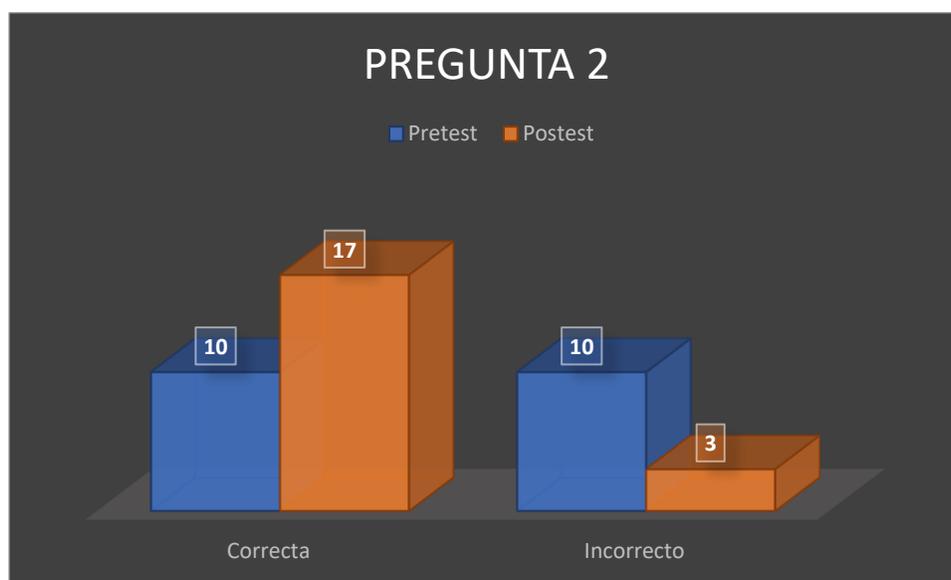
Los resultados en cuanto a cuestionario aplicado demuestran lo importante de utilizar herramientas de apoyo en las sesiones de clase que permitan reforzar el aprendizaje de los estudiantes.

En referencia a la pregunta 1: *Es un modelo donde los datos están organizados, manejados y almacenados de manera eficiente para su acceso y modificación de los datos*, como se puede observar en la Gráfica 1 en el pretest 9 estudiantes respondieron de manera incorrecta, aunque después de aplicar el sistema tutor afectivo y responder el postest, solo 4 respondieron de manera incorrecta, esto demuestra que estos poseen deficiencias en cuanto al concepto de estructura de datos, probablemente por confusión en las palabras técnicas utilizadas.



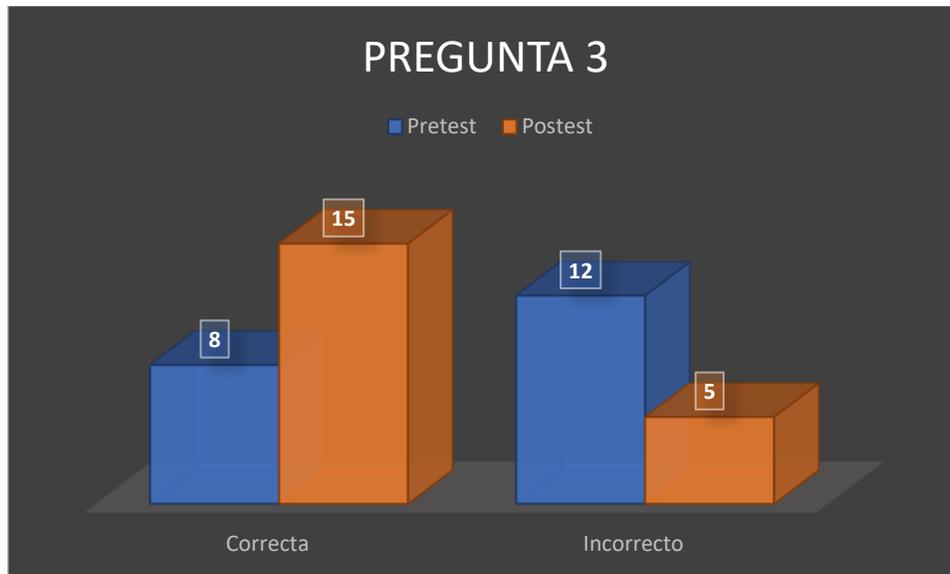
Gráfica 1 Respuestas pregunta 1

Con respecto a la pregunta 2: *Son ejemplos de estructuras de datos*, los alumnos mostraron falta de conocimiento sobre dichos tipos, lo cual resalta la importancia de conocer adecuadamente el concepto de estas ya que se utilizan constantemente en codificación. Como se puede observar en la Gráfica 2, en el pretest la mitad del grupo contestó de manera incorrecta, posterior al uso del sistema tutor afectivo se tuvo una mejora considerable y solo 3 estudiantes respondieron de manera equivocada. Sin embargo, aún existen alumnos en los que es necesario seguir reforzando dichos conceptos.



Gráfica 2 Respuestas pregunta 2

En relación con la pregunta 3: *¿Cuál de los siguientes algoritmos es el óptimo para realizar la suma de los primeros 15 números?* Los resultados obtenidos que se pueden observar en la Gráfica 3, denotan que los estudiantes tienen bastantes deficiencias en cuestión de análisis del código y sobre como es que las estructuras de datos pueden resultar eficientes el proceso computacional, por tal motivo en el pretest más de la mitad de los estudiantes, es decir, 12 de ellos contestó de manera incorrecta, lo cual mejoro posteriormente a la aplicación del sistema tutor afectivo, sin embargo aun en el postest una cantidad considerable, 5 estudiantes respondió de manera incorrecta.



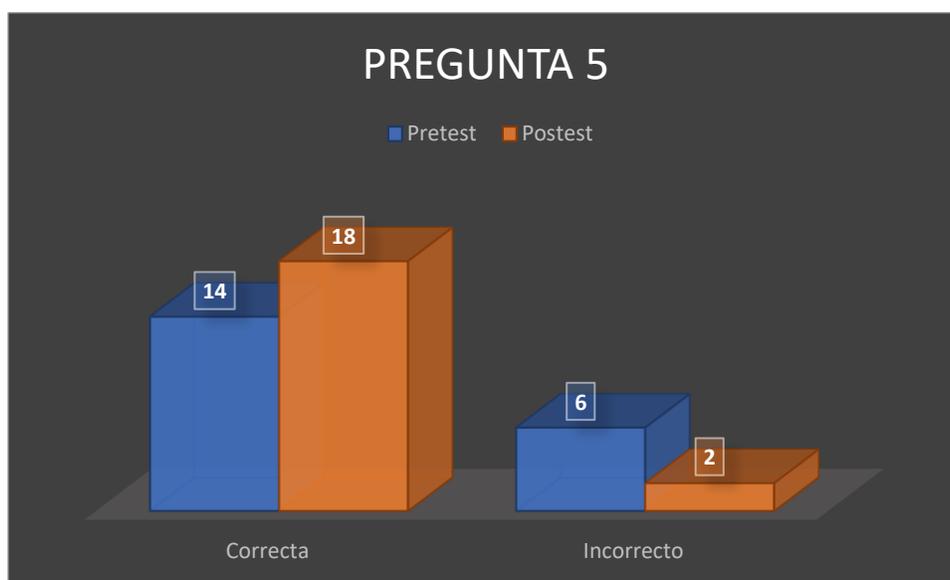
*Gráfica 3 Respuestas pregunta 3*

En la pregunta 4: ¿Qué elemento NO es un elemento clave para tomar en cuenta para escoger el mejor algoritmo? Este tipo de pregunta requiere nuevamente del análisis de los estudiantes con respecto a los conceptos que se manejan en estructuras de datos. La Gráfica 4, muestra que en el pretest poco más de la tercera parte del grupo pudo responder correctamente, es decir, solo 7 alumnos. Después de la aplicación del sistema tutor afectivo, los resultados mejoraron, y solo 4 respondieron incorrectamente.



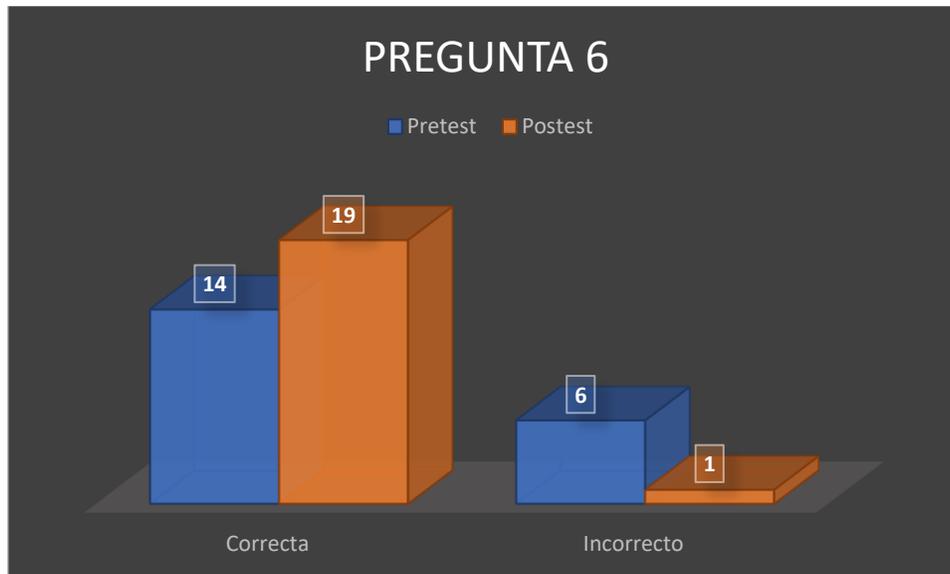
*Gráfica 4 Respuestas pregunta 4*

Para la pregunta 5: Es un método que se llama a sí mismo, la mayor parte de los estudiantes respondió correctamente y solo una tercera parte contestó de manera incorrecta. Como se puede observar en la Gráfica 5 después de aplicar el sistema tutor afectivo, en el postest solo dos estudiantes respondieron de manera incorrecta, lo cual es satisfactorio ya que es una cantidad muy pequeña de estudiantes a los que se les puede proporcionar más actividades de reforzamiento, ya que el concepto de recursividad es sumamente importante al momento de codificar ya que facilita muchas tareas y sobre todo optimiza varios procesos.



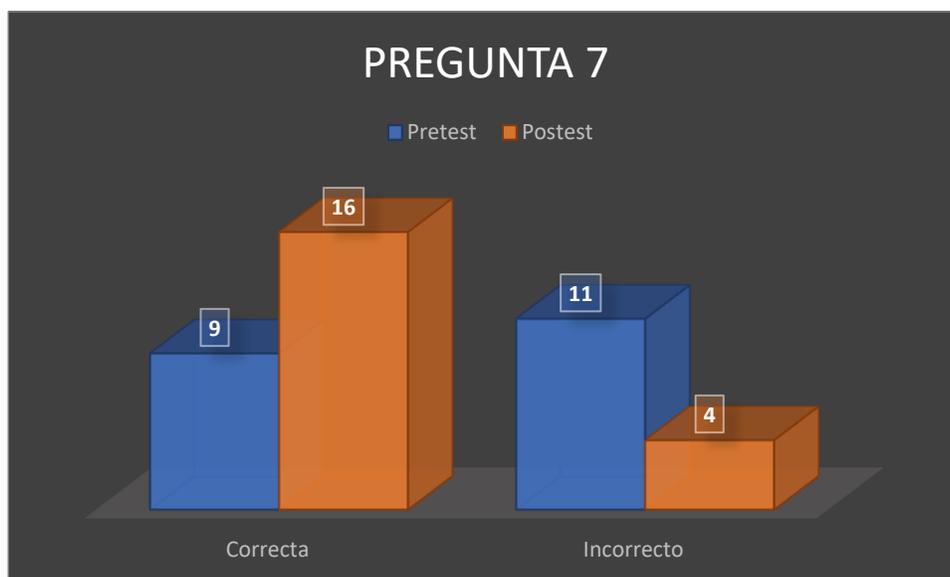
Gráfica 5 Respuestas pregunta 5

En referencia a la pregunta 6: ¿Qué es LIFO? Esta pregunta corresponde más al conocimiento de las siglas que al concepto, por lo que era una pregunta sencilla. En la Gráfica 6, se puede observar que la mayor parte de los estudiantes respondió adecuadamente y que posteriormente al uso del sistema tutor afectivo al responder el postest, casi todos respondieron de manera correcta, solo un estudiante cometió un error nuevamente al responder. Lo que permite entender que el sistema tutor facilitó la comprensión de las siglas para casi todos los estudiantes.



*Gráfica 6 Respuestas pregunta 6*

En el caso de la pregunta 7: ¿En qué casos es útil ocupar un método de ordenamiento secuencial? Nuevamente requiere del análisis del estudiante para reconocer en qué momento es recomendable aplicar sus conocimientos de estructuras de datos. Poco mas de la mitad del grupo respondió de manera incorrecta. Después del uso del tutor el resultado en el postest fue mejor, pero aún existen 4 estudiantes que deben reforzar dicha área, como se observa en la Gráfica 7.



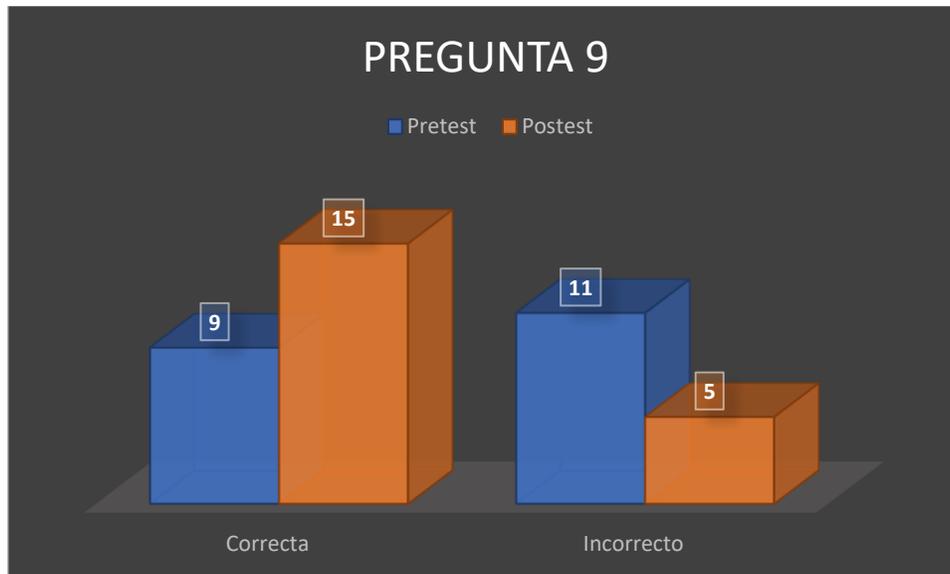
*Gráfica 7 Respuestas pregunta 7*

La pregunta 8: Son ejemplos de estructuras NO lineales (múltiple), corresponde a la identificación de los tipos de estructuras múltiples. La Gráfica 8, muestra que en el pretest casi la mitad de los estudiantes no pudieron identificar correctamente las estructuras múltiples, posterior a la aplicación del sistema tutor afectivo solo cuatro estudiantes respondieron erróneamente lo cual resalta hasta el momento lo importante que es que los estudiantes puedan comprender los conceptos teóricos.



Gráfica 8 Respuestas pregunta 8

En cuanto a la pregunta 9: ¿Cuál de los siguientes métodos tiene una menor complejidad? Nuevamente se trata de una pregunta de análisis donde los estudiantes deben conocer los tipos de estructuras que se presentan y a partir de eso definir cual de ellas corresponde a la respuesta correcta. Como se puede observar en la Gráfica 9 en el pretest poco menos de la mitad del grupo respondió de manera errónea al no identificar claramente el método con menor complejidad, pero posteriormente al uso del sistema tutor afectivo una mejoría, aunque aun existe una cantidad considerable de alumnos que no comprendieron adecuadamente como identificar que método representa menor complejidad.



*Gráfica 9 Respuestas pregunta 9*

Finalmente, en la pregunta 10: Son tipos de corrimiento que se le pueden aplicar a un árbol, esta pregunta nuevamente se trata de conceptos teóricos en cuanto a los tipos de corrimiento. En este caso como se puede observar en la Gráfica 10, en el pretest una tercera parte del grupo respondió de manera incorrecta pero posteriormente al uso del sistema tutor afectivo en el postest solo dos estudiantes respondieron de manera incorrecta notándose que la mayoría de los estudiantes pudo identificarlos los tipos.



*Gráfica 10 Respuestas pregunta 10*

## 5.2 Análisis de expresiones faciales

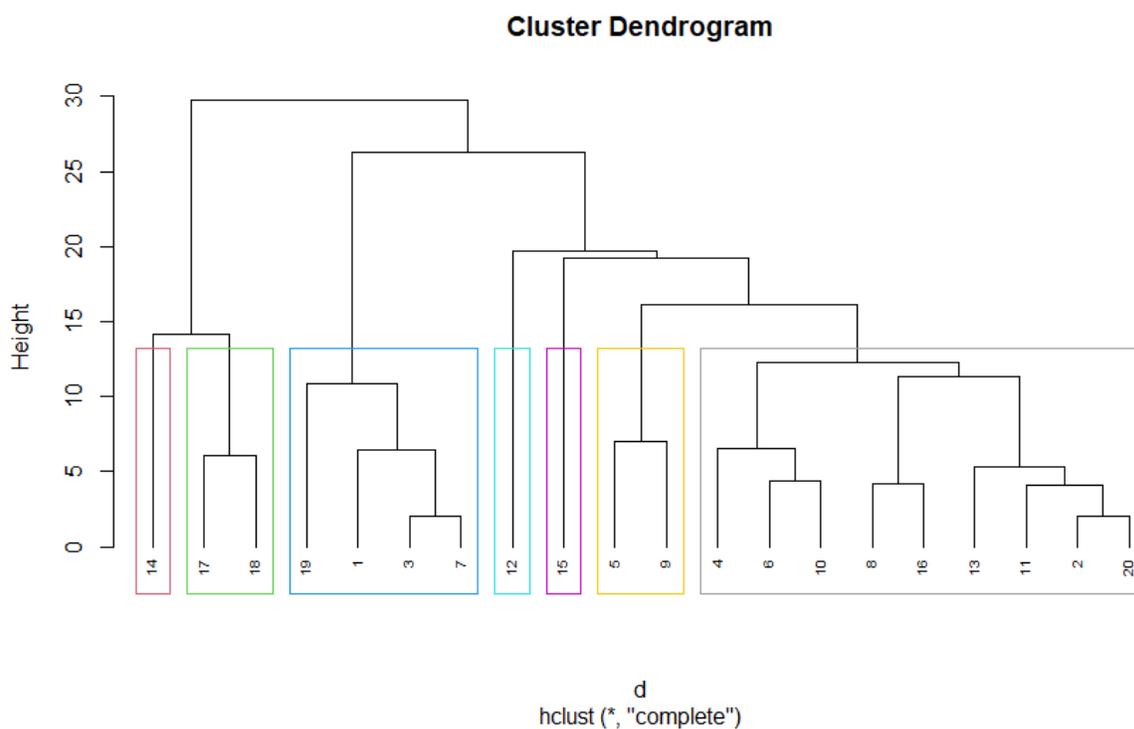
En relación con el análisis facial que lleva a cabo el sistema tutor afectivo debido a que se trata de una aplicación móvil este se realiza mediante la cámara frontal del dispositivo móvil, después de implementar el tutor y realizar el preprocesamiento de los datos se obtuvo una base de datos (Tabla 15) con el registro de las emociones que presentaron los estudiantes mientras estuvieron interactuando con el tutor.

Tabla 15 Base de datos de emociones

Alumno	Disgusto	Alegría	Desprecio	Tristeza	Miedo	Sorpresa	Enojo
1	0	0	0	0	0	18	0
2	9	0	0	0	0	0	0
3	1	0	5	0	0	14	0
4	6	0	4	2	0	8	0
5	0	0	0	0	0	0	7
6	2	3	5	0	0	6	3
7	2	0	4	0	0	15	1
8	12	0	3	0	0	7	4
9	0	0	0	0	0	0	0
10	2	0	5	0	0	7	0
11	11	0	0	0	0	3	0
12	9	0	16	0	0	1	0
13	6	1	1	0	0	4	0
14	0	23	0	0	1	0	0
15	18	0	0	0	4	0	5
16	12	0	0	0	0	10	4
17	0	14	0	0	0	0	3
18	1	9	1	0	0	1	0
19	0	10	1	0	0	15	0
20	9	0	2	0	0	0	0

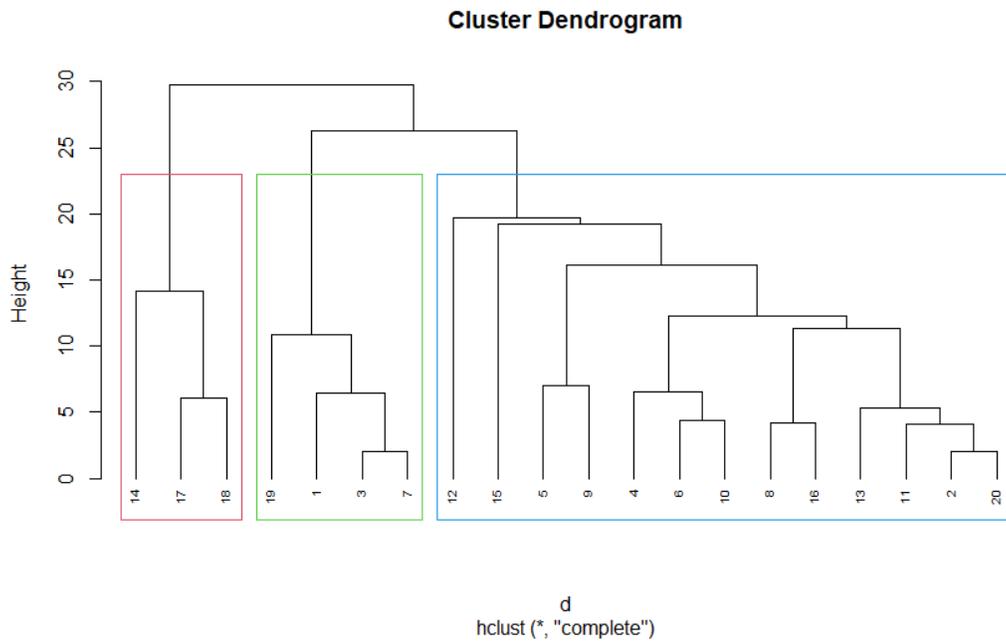
Después de realizar un agrupamiento de por clústeres utilizando el método de k-means se obtuvieron siete grupos para lograr identificar a las 7 emociones. Como se puede observar en la Gráfica 11, los resultados se encuentran variados ya que las emociones presentadas por los estudiantes en los diferentes momentos de interacción permite clasificarlos en grupos pequeños con similitudes en las emociones presentadas, como se puede notar el grupo siete enmarcado en color gris posee la mayor cantidad de alumnos lo que sin duda denota un conjunto de emociones que tuvo mayor presencia

que otras. Así mismo se observan, a tres estudiantes etiquetados como 12, 14 y 15 que presentaron una única emoción diferente cada uno de ellos, durante toda la interacción que tuvieron con el tutor, por lo que también son un caso de estudio que puede ser analizado en investigaciones posteriores. Si bien el agrupamiento presentado permite realizar algunos análisis es importante establecer un análisis más a detalle, con un agrupamiento menor que permita visualizar otras características de la población de estudio.



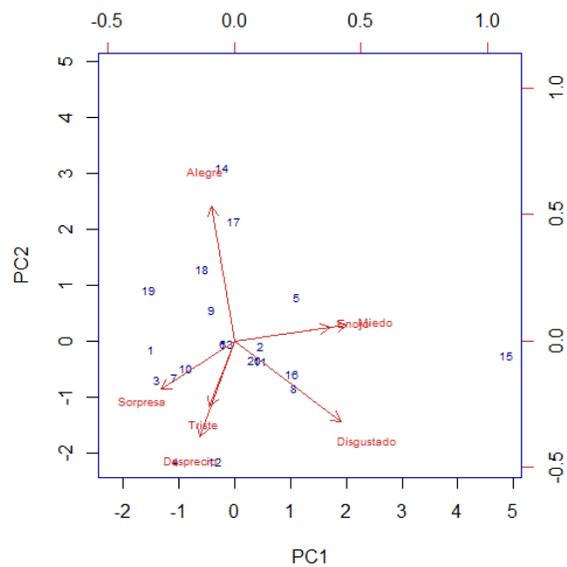
*Gráfica 11 Clasificación en 7 grupos*

Al realizar un nuevo agrupamiento con respecto a la similitud de las emociones presentadas, se obtienen tres grupos principales, como se muestra en la Gráfica 12, donde se puede observar que los alumnos pertenecientes al grupo de estudio presentaron de manera general tres emociones principalmente y que caracterizan a la población de estudio.



*Gráfica 12 Grupos principales de estudiantes con emociones similares*

Considerando lo anterior, se realizó un análisis de componentes principales para definir las dos componentes que caracterizan a la población observando en la Gráfica 13 el primer componente se refiere a la emoción de disgusto mientras que el segundo componente a la sorpresa, las cuales son las dos emociones que predominan en la población.



*Gráfica 13 Componentes principales*

## CONCLUSIONES

En referencia a la implementación de la aplicación móvil en modalidad de sistema tutor afectivo, se obtuvo que gracias a las actividades presentadas los estudiantes pudieron reforzar sus conocimientos tanto teóricos como prácticos al poseer diferentes actividades que les permitieron llevar un proceso de aprendizaje más dinámico y sobre todo con relación a las necesidades que cada uno presentaba, debido que los estudiantes podían acceder a las actividades según su gusto o desempeño en el pretest realizado al ingresar a la aplicación.

La aplicación resulto sumamente atractiva para los estudiantes ya que su interfaz permitió un adecuado uso y sobre todo facilidad al momento de acceder a cualquier sección de esta.

Con referencia a las emociones que pudo analizar el tutor, fueron la sorpresa y disgusto las emociones con mayor presencia, y estas caracterizaron a la población en general. La presencia del disgusto se debió principalmente a los errores al realizar las actividades, por lo que puede deducirse que los estudiantes presentaron varias dificultades para comprender adecuadamente los conceptos de estructura de datos.

Por otra parte, la sorpresa también como emoción predominante se refiere a las actividades en donde los estudiantes realizaban practicas de codificación en el editor presentado en el tutor, al observar como sus códigos no son tan eficientes como ellos creían.

De manera general podemos concluir que el manejo de las emociones en proceso aprendizaje es de suma importancia y que la implementación de una aplicación móvil en modalidad de tutor afectivo es una herramienta bastante eficaz para apoyar el proceso de aprendizaje de estudiantes que cursan ingeniería, pero sobre todo para aquellos que estudian estructuras de datos en el área de programación computacional.

## REFERENCIAS

1. ¿Qué importancia tienen las emociones en el aprendizaje? - NeuroClass. (n.d.). Retrieved December 3, 2020, from <https://neuro-class.com/queimportancia-tienen-las-emociones-en-el-aprendizaje/>
2. Afi. (2017). El trabajo del futuro. 17. Retrieved from [www.afi.es](http://www.afi.es)
3. Barron-Estrada, M. L., Zatarain-Cabada, R., Oramas-Bustillos, R., & Gonzalez-Hernandez, F. (2017). Sentiment Analysis in an Affective Intelligent Tutoring System. Proceedings - IEEE 17th International Conference on Advanced Learning Technologies, ICAALT 2017, 394–397. <https://doi.org/10.1109/ICALT.2017.137>
4. Ben Ammar, M., Neji, M., Alimi, A. M., & Gouardères, G. (2010). The Affective Tutoring System. Expert Systems with Applications, 37(4), 3013–3023. <https://doi.org/10.1016/j.eswa.2009.09.031>
5. Computación afectiva - InnoWiki. (n.d.). Retrieved December 2, 2020, from [http://185.5.126.23/innowiki/index.php/Computación\\_afectiva](http://185.5.126.23/innowiki/index.php/Computación_afectiva)
6. D'Mello, S., & Graesser, A. (2011). The half-life of cognitive-affective states during complex learning. Cognition and Emotion, 25(7), 1299–1308. <https://doi.org/10.1080/02699931.2011.613668>
7. Daniel, L. H. (2015). Reconocimiento de emociones centradas en el aprendizaje por medio de expresiones faciales.
8. Dirección General de Planeación, P. y E., & Educativa. (2017). Principales cifras del Sistema Educativo Nacional 2016-2017. Retrieved from [http://www.planeacion.sep.gob.mx/Doc/estadistica\\_e\\_indicadores/principales\\_cifras/principales\\_cifras\\_2016\\_2017\\_bolsillo.pdf](http://www.planeacion.sep.gob.mx/Doc/estadistica_e_indicadores/principales_cifras/principales_cifras_2016_2017_bolsillo.pdf)
9. Durango Hernández, J. A., & Pascuas Rengifo, Y. S. (2016). Los Sistemas Tutores Inteligentes y su aplicabilidad en la educación. Horizontes Pedagógicos, 17(2), 104–116. Retrieved from <https://revistas.iberamericana.edu.co/index.php/rhpedagogicos/article/view/826/772>

10. Elizondo Moreno, A., Rodríguez Rodríguez, J. V., & Rodríguez Rodríguez, I. (2018). La importancia de la emoción en el aprendizaje: Propuestas para mejorar la motivación de los estudiantes. *Cuaderno de Pedagogía Universitaria*, 15(29), 3–11. <https://doi.org/10.29197/cpu.v15i29.296>
11. Fuentes-Rosado, J. I., & Moo-Medina, M. (2017). Dificultades de aprender a programar. *Revista Educación En Ingeniería*, 12(24), 76. <https://doi.org/10.26507/rei.v12n24.728>
12. GARCIA, M., HURTADO, P. A., QUINTERO, D. M., RIVERA, D. A., & UREÑA, Y. C. (2018). La gestión de las emociones, una necesidad en el contexto educativo y en la formación profesional. *Revista ESPACIOS*, 39(49).
13. Gina, M., Rocio, B. M., Ángel, C., José, S., & Luis, C. (2018). LA COMPUTACIÓN AFECTIVA: EMOCIONES, TECNOLOGÍAS Y SU RELACIÓN CON LA EDUCACIÓN VIRTUAL. (1), 95–104.
14. González, C., Mora, A., & Toledo, P. (2014). Gamification in intelligent tutoring systems. *ACM International Conference Proceeding Series*, 221–225. <https://doi.org/10.1145/2669711.2669903>
15. González, M. R. (2014). Aprender a programar “apps” como enriquecimiento curricular en alumnado de alta capacidad. *Bordon*, 66(4), 135–155. <https://doi.org/10.13042/Bordon.2014.66401>
16. Guilcatoma Panchi, B. E., & Quimbita Cadena, N. A. (2006). Sistema tutor inteligente para programación básica “C” bajo el modelo de objetos de aprendizaje “STIPB”. Retrieved from <https://repositorio.espe.edu.ec/handle/21000/3426>
17. Huapaya, C. R. (2009). *Sistemas Tutoriales Inteligentes Un análisis crítico*. 74.
18. Iglesias-Hoyos, S., del Castillo Arreola, A., & Muñoz-Delgado, J. I. (2016). Reconocimiento facial de expresión emocional: diferencias por licenciaturas. *Acta de Investigación Psicológica*, 6(3), 2494–2499. <https://doi.org/10.1016/j.aiprr.2016.07.001>

19. Lisetti, C. L. (1998). Affective computing. *Pattern Analysis and Applications*, 1(1), 71–73. <https://doi.org/10.1007/bf01238028>
20. Mao, X., & Li, Z. (2010). Agent based affective tutoring systems: A pilot study. *Computers and Education*, 55(1), 202–208. <https://doi.org/10.1016/j.compedu.2010.01.005>
21. Navarro, E. (2018). Socio-Cognitive and Affective Computing. *Socio-Cognitive and Affective Computing*. <https://doi.org/10.3390/books978-303897-199-3>
22. Petrovica, S., Anohina-Naumeca, A., & Ekenel, H. K. (2016). Emotion Recognition in Affective Tutoring Systems: Collection of Ground-truth Data. *Procedia Computer Science*, 104(December 2016), 437–444. <https://doi.org/10.1016/j.procs.2017.01.157>
23. Ponce, J. P., Pintado, P., & Biset, J. (2019). ANÁLISIS DE IMPLEMENTACIONES DE SISTEMAS TUTORES INTELIGENTES Y AFECTIVOS. REVISIÓN SISTEMÁTICA. 7(2), 218–234.
24. Romdhani, A. (2016). PROTOTIPO DE TUTOR INTELIGENTE PARA EL APRENDIZAJE DE LA PROGRAMACIÓN DE COMPUTADORES. X, 1–21.
25. Roschelle, J. (2009). Effective Tutoring Techniques: A Comparison of Human Tutors and Intelligent Tutoring Systems. (March 2015), 37–41. <https://doi.org/10.1207/s15327809jls0203>
26. Stromquist, N. P. (2019). World Development Report 2019: The changing nature of work. In *International Review of Education* (Vol. 65). <https://doi.org/10.1007/s11159-019-09762-9>
27. Sykes, E. (2010). Design, development and evaluation of the Java Intelligent Tutoring System. *Technology, Instruction, Cognition & Learning*, 8(1), 25–65.
28. Teresa Machin-Torres, I. (2017). Sistema tutor para la aplicación de la programación mediante análisis de inteligencias. *Revista Científica*, 2(29), 219–229. <https://doi.org/10.14483/udistrital.jour.RC.2016.29.a9>

29. Universal Emotions | What are Emotions? | Paul Ekman Group. (n.d.). Retrieved December 3, 2020, from <https://www.paulekman.com/universalemotions/>
30. Zatarain-Cabada, R., Barrón-Estrada, M. L., & García-Lizárraga, J. (2016). Sistema tutor afectivo para el aprendizaje de las matemáticas usando técnicas de gamificación. *Research in Computing Science*, 111(1), 83–96. <https://doi.org/10.13053/rcs-111-1-7>
31. Zatarain Cabada, R. (2018). Reconocimiento afectivo y gamificación aplicados al aprendizaje de Lógica algorítmica y programación. *Revista Electrónica de Investigación Educativa*, 20(3), 115. <https://doi.org/10.24320/redie.2018.20.3.1636>
32. Arroyo-lópez, E., Ríos-silva, T., Morales-trujillo, M., Rico-martínez, A., & Oktaba, H. (2015). Expresando Nuestra Manera de Trabajo con KUALI-BEH : Lecciones Aprendidas por Tic-Tac-S Expresando Nuestra Manera de Trabajo con KUALI-BEH : Lecciones Aprendidas por Tic-Tac-S. December 2018.
33. Balsamiq. (2021). <https://balsamiq.com/wireframes/>
34. Boada Oriols, M., & Gómez Guitiérrez, J. A. (2018). El gran libro de Angular.
35. Booch, G., Complejidad, C., El, C., Objetos, M. D. E., Objetos, C. C. Y., & Clasificació, N. (1996). *Análisis y Diseño Orientado a Objetos*.
36. capacitorjs.com. (2021). <https://capacitorjs.com/docs>
37. cloud.google.com. (2021a). <https://cloud.google.com/vision/>
38. Cloud.Google.Com. (2021b). <https://cloud.google.com/anthos/docs/concepts/overview>
39. developer.android.com. (2021). <https://developer.android.com/studio>
40. Faranello, S. (2012). *Balsamiq Wireframes Quickstart Guide*. <https://books.google.com/books?id=gk6b4gGQjjsC&pgis=1>
41. Git-Scm.Com. (2021). <https://git-scm.com/>
42. gitlab.com. (2021). <https://about.gitlab.com/>
43. Google Cloud Platform. (2015). Storing and Analyzing Your Data in Google's Cloud. 1–14. [https://cloud.google.com/files/articles/google-cloud\\_technical-](https://cloud.google.com/files/articles/google-cloud_technical-)

article\_overview-of-storage-options.pdf

44. Krishnan, S. P. T., & Gonzalez, J. L. U. (2015). Building Your Next Big Thing with Google Cloud Platform. Building Your Next Big Thing with Google Cloud Platform, August. <https://doi.org/10.1007/978-1-4842-1004-8>
45. Moroney, L. (2017). The Definitive Guide to Firebase. In The Definitive Guide to Firebase. <https://doi.org/10.1007/978-1-4842-2943-9>
46. nodejs.org. (2021). <https://nodejs.org/es/about/>
47. SUPARATCHADECH, P. (2017). Ionic Framework คืออะไร. <http://ionicframework.com#>
48. Tutorialspoint. (2014). Balsamiq Mockups. Time, 17–26.
49. Visual Studio Code. (2021). <https://code.visualstudio.com/docs>
50. Wieruch, R. (2019). The Road to React with Firebase. <http://leanpub.com/the-road-to-react-with-firebase>
51. Wong, S. (2007). StarUML Tutorial. 1–18.