

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO
UQUEST FCPN: PLATAFORMA VIRTUAL DE OPINIÓN
ESTUDIANTIL
CASO: FACULTAD DE CIENCIAS PURAS Y NATURALES

Proyecto de Grado para obtener el Título de Licenciatura en Informática

Mención Ingeniería de Sistemas Informáticos

POSTULANTE: RONALD ALCIDES GUARACHI ENRIQUEZ

TUTOR METODOLÓGICO: M.Sc. ROSA FLORES MORALES

ASESOR: M.Sc. MOISES MARTIN SILVA CHOQUE

LA PAZ – BOLIVIA

2021

Dedicatoria

Este proyecto de grado va dedicado a las personas que estén leyendo, que por algún motivo quien sabe porque, este proyecto llego a tus manos, estas invirtiendo tu tiempo, tus horas de trabajo, por el cual me sentiría feliz que llegaras a leer el gran paso que yo di al finalizar este proyecto que planteé para ser tanto como mejor persona y un gran profesional.

Además de dedicarte unas palabras: “Tropiézate, cáete miles de veces porque de cada tropiezo aprendes, equivócate programando, no te sientas mal si lo haces, solucióvalo, no descanses hasta solucionarlo, eso te hará más fuerte mentalmente porque no hay nada más lindo que llenarte de conocimiento de algo que te gusta hacer”

Agradecimientos

A mis padres que me enseñaron los valores que necesito para seguir creciendo me alimentaron y me dieron un hogar para que cumpla esta gran meta, por su gran paciencia que aguantaron miles de errores que tuve, por la comprensión que me tuvieron, gracias por la sabiduría que me inculcaron, por esas cosas tan simples en la vida que no sería nada sin ellas.

A mi primo Mauricio Enriquez que a pesar de la distancia logro enseñarme y asesorarme con la programación llevada a cabo, sin el este proyecto se hubiera demorado más tiempo de lo previsto, además de enseñarme un mundo de posibilidades y a no conformarme con nada, porque siempre existen soluciones.

A mi asesor: Lic. Moises Silva y a mi tutora: M.Sc. Rosa Flores. No hay palabras que describan cuan agradecido estoy con ustedes por su tiempo y dedicación para que este proyecto siga adelante. Gracias por su paciencia conmigo, por sus recomendaciones, observaciones, ideas y sugerencias, gracias por haberme guiado en esta última parte de esta vida universitaria.

Quiero agradecer también a la empresa CommIT SRL que además de darme un trabajo con un gran líder con buenos ideales también llevo a ayudarme a aprender herramientas que nunca había visto en mi vida, ya que sin ese conocimiento no entendería varios factores necesarios para que un proyecto pueda ser escalable.

A mi tía Jhenny y mi primo Luis que siempre apoyaron a mi y a mi familia en momentos difíciles, mis especiales gratificaciones porque la familia es lo ultimo que se pierde en la vida.

Por ultimo y no menos importante a mi enamorada Corali que no dejo de apoyarme en los momentos más difíciles en esta etapa universitaria siempre dando ese granito de arena para que esto sea posible, dándome consejos y alternativas a problemas que surgían y quitándome esa nube de la cabeza.

RESUMEN

El presente proyecto de grado se ha implementado una Plataforma Virtual de Opinión Estudiantil enfocada para la Facultad de Ciencias Puras y Naturales, este proyecto surge por la escasa y poca información existente respecto a la opinión de los estudiantes en las distintas carreras de la Facultad.

Este proyecto se centró en el desarrollo de un sistema para la realización de encuestas para posteriormente compartir a los estudiantes, docentes o administrativos que deseen participar en encuestas elaboradas por un usuario en particular y responderlas individualmente. En su desarrollo se aplicó la metodología de desarrollo ágil SCRUM y para el modelado y diseño de esquemas la metodología UWE (*UML-based web engineering*), así mismo se utilizaron diversas herramientas de diseño como mapeo de historias de usuario, casos de uso, y diagramas de secuencia; para la implementación del desarrollo del sistema se utilizó el framework Angular basado en JavaScript y TypeScript que facilitó el desarrollo y posibilitó la eficacia del proyecto, a su vez se hizo uso del framework Spring Boot basado en JAVA para la administración de los datos que proveerá el servidor.

Una vez finalizado el desarrollo del sistema, se realizaron las pruebas respectivas por los usuarios dentro la Facultad de Ciencias Puras y Naturales, posteriormente fue sometido a valoración según la metodología de calidad GQM para los atributos expuestos en los estándares ISO/IEC 9126 y la ISO/IEC 25000, concluyendo así que el sistema cumple con criterios de calidad.

Finalmente, se ha logrado el desarrollo de la plataforma virtual para la creación y realización de encuestas que facilitará conocer la opinión de docentes, estudiantes y/o administrativos dentro la Facultad.

Palabras clave: encuesta online, spring boot, scrum, plataforma web, generador de encuestas.

SUMMARY

This graduate project has implemented a Virtual Student Opinion Platform focused on the Pure and Natural Science School. This project arises from the limited information that exists regarding the student's opinion enrolled on the different School Careers.

This project is focused on the development of a system to conduct surveys and later give access to students, teachers and administrative staff who wish to participate on survey. These surveys can be made by any user and be answered individually. The development of this project used the agile development methodology SCRUM and to model and design of diagrams we used the UWE methodology (UML-based web engineering). In addition, we used various design tools such as mapping user history, use cases, and sequence diagrams. For system development we used the Angular framework based on JavaScript and TypeScript, which made the development easier and made the project more efficient. Additionally, we used the Spring Boot framework based on JAVA for the administration of the data that the server will provide.

Once the development of the system was completed, the users in the Pure and Natural Science School carried the respective tests. Subsequently, the system was subjected to an evaluation according to the GQM quality methodology for the attributes set in the standards ISO / IEC 9126 and ISO / IEC 25000. It concluded that the system meets the quality criteria.

Finally, the development of the virtual platform to create and conduct surveys has been achieved. It will help to know the opinion of teachers, students and / or administrators within the School.

Keywords: online survey, spring boot, scrum, web platform, survey generator.

ÍNDICE

Dedicatoria.....	2
Agradecimientos	3
CAPÍTULO I	10
MARCO INTRODUCTORIO	10
1.1. INTRODUCCIÓN	11
1.2. ANTECEDENTES.....	11
1.3. DESCRIPCIÓN DEL PROBLEMA	12
1.4. OBJETIVO PRINCIPAL.....	13
1.5. OBJETIVOS SECUNDARIOS	13
1.6. JUSTIFICACIÓN	14
1.7. LÍMITES Y/O ALCANCES	14
1.8. DISEÑO METODOLÓGICO.....	15
1.9. HERRAMIENTAS	19
CAPÍTULO II.....	21
MARCO TEÓRICO.....	21
2.1. DESARROLLO MULTIPLATAFORMA.....	22
2.2. HERRAMIENTA DE DESARROLLO MULTIPLATAFORMA	22
2.3. METODOLOGIAS DE DESARROLLO	23
2.4. SCRUM.....	25
2.5. UWE (<i>UML-Based Web Engineering</i>).....	26
2.5. SPRING MODELO VISTA CONTROLADOR (MVC).....	27
2.6. FRAMEWORK SPRING	28
2.7. SPRING BOOT	29
2.8. INYECCIÓN DE DEPENDENCIAS	29
2.9. HIBERNATE.....	30
2.10. ANGULAR.....	30
2.11. ANGULAR MATERIAL	31
2.12. JSON WEB TOKEN (JWT)	31
2.13. BCRYPT	32
2.14. POSTMAN	33
2.15. HERRAMIENTAS DE DESARROLLO.....	33
2.16. ATRIBUTOS DE CALIDAD.....	34
CAPÍTULO III.....	39
MARCO APLICATIVO.....	39

3.1.	RESUMEN	40
3.2.	OBTENCIÓN DE REQUERIMIENTOS	40
3.2.	PRODUCT-BACKLOG	42
3.3.	ROLES SCRUM.....	44
3.4.	PLANIFICACIÓN DE SPRINTS.....	44
3.5.	DESARROLLO DE SPRINTS	45
3.5.1.	PRIMER SPRINT: MÓDULO DE DISEÑO E IMPLEMENTACIÓN DE BASE DE DATOS	45
3.5.2.	SEGUNDO SPRINT: MÓDULO DE REGISTRO Y LOGIN DE USUARIO	47
3.5.3.	TERCER SPRINT: MÓDULO DE ADMINISTRACIÓN DE CUENTAS DE USUARIO	50
3.5.4.	CUARTO SPRINT: MÓDULO DE ADMINISTRACIÓN DE CARRERAS E INSTITUTOS	56
3.5.5.	QUINTO SPRINT: MÓDULO DE ADMINISTRACIÓN Y CREACIÓN DE ENCUESTAS	62
3.5.6.	SEXTO SPRINT: MÓDULO DE RESPUESTA DE ENCUESTAS	69
3.6.	DESARROLLO DE INTERFAZ.....	75
3.7.	RESPONSABILIDAD DE AUTOR.....	88
CAPÍTULO IV.....		89
RESULTADOS.....		89
4.1.	PRUEBAS UNITARIAS Y DE INTEGRACIÓN.....	90
4.2.	PRUEBAS DE ACEPTACIÓN Y FUNCIONALIDAD	91
4.3.	PRUEBAS DE RENDIMIENTO.....	93
4.4.	ENCUESTA A USUARIOS	95
4.5.	EVALUACIÓN DE CALIDAD	96
CAPÍTULO V.....		101
CONCLUSIONES Y RECOMENDACIONES.....		101
5.1.	CONCLUSIONES	102
5.2.	RECOMENDACIONES	103
ANEXOS		107
Base de datos en MySQL (desarrollo).....		107
MANUAL DE USUARIO.....		108
DESCRIPCIÓN DE VISTAS		109
Ventana de registro		109
Ventana de “login”.....		111
Ventana de Encuestas públicas		111

Ventana de Encuestas propias del Usuario	115
Ventana de perfil.....	117



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON EstrictAMENTE ACADÉMICOS

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVA EN INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

CAPÍTULO I

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

La Universidad Mayor de San Andrés (UMSA) durante varios años obtuvo información de los estudiantes mediante encuestas; afortunadamente la tecnología ha avanzado aceleradamente ofreciendo una variedad de ventajas, entre ellos la facilidad de acceso al “Internet” y que hoy en día los estudiantes lo usan para sus actividades académicas, es por eso que tanto como estudiantes docentes y/o administrativos en la UMSA consideran elementos de gran importancia, entre ellos está la aceptación y consideración de la opinión de los estudiantes y la comunidad para la toma de decisiones dentro de los ámbitos facultativos.

La necesidad de progresar tecnológicamente se hizo muy importante hoy en día, es por eso que también obtener información de los estudiantes y de la comunidad también lo es, el proyecto actual se enfoca en realizar consultas vía WEB para toda la UMSA, para facilitar a los estudiantes información pública de la opinión social o académica, ya sea de sus propias carreras como por ejemplo la obtención de información en tiempo real en las temporadas electorales.

El actual proyecto presenta una plataforma virtual que fue desarrollada bajo las tecnologías y requerimientos por la Unidad de Tecnologías de la Información y Comunicación (ITIC) brindando a los estudiantes, docentes un área para la elaboración de encuestas dentro la Facultad de Ciencias Puras y Naturales, esto con el fin de obtener información para la mayor comprensión para la toma de decisiones académicas.

1.2. ANTECEDENTES

1.2.1. ANTECEDENTES INTERNACIONALES

Se encontraron plataformas similares a la propuesta externas a Bolivia, una de ellas es “Online Encuesta” lanzada el 2007 en Zurich, Switzerland por la empresa “Enuvo”, en esta plataforma solo se puede participar con una cuenta, ingresar mediante Facebook o un *login* simple, aun así, la ventaja de esta plataforma es que se puede generar una encuesta y la plataforma se

encarga de distribuir la encuesta a todo el mundo con diferentes costos dependiendo de las necesidades de la encuesta.

Otra plataforma y la más popular conocida hoy en día como “Google Forms” que en sus inicios comenzó como “Google Sheets” en el 2008 desarrollada por Google y está orientada en la creación de formularios, tiene una interfaz amigable y fácil de entender para el usuario y de fácil acceso además de ser gratuita llega a ser una potencial competencia para el sistema actual.

1.2.2. ANTECEDENTES NACIONALES

En la actualidad, se encontró una plataforma similar en Bolivia conocida como “U-Report Bolivia” lanzada por UNICEF (2019) disponible para todo público. El objetivo de esta plataforma es obtener la opinión de la población adolescente y joven para trazar una nueva perspectiva de participación social, muchas encuestas planteadas en esta plataforma han sido de gran ayuda al tratar temas como ser: “¿Conoces casos de adolescentes embarazadas?”, además de preguntas o encuestas que la plataforma realiza, también dispone de un apartado para obtener información importante de Bolivia, como un periódico *online* histórico. Los datos que proporcionan son sistematizados y compartidos con autoridades nacionales, municipales y comunitarias y, así, contribuyen a mejorar los programas y acciones de organizaciones dedicadas a jóvenes y adolescentes y a llamar la atención sobre temas urgentes e importantes.

1.3. DESCRIPCIÓN DEL PROBLEMA

La Universidad Mayor de San Andrés, hoy en día no cuenta con una plataforma enfocada a los estudiantes y docentes para conocer la opinión respecto a las diversas decisiones que se toman en la parte académica; en pocas ocasiones estas han sido consultadas al estamento estudiantil y/o docentes, y cuando se llevó a cabo se realizó de manera poco eficiente y con demoras en cuanto a la obtención de los resultados.

La escasa información que se obtiene de la comunidad estudiantil impide la toma de decisiones para las diversas actividades académicas y eso converge en que la universidad tome decisiones que no beneficien directamente a los estudiantes.

Los estudiantes no tienen la posibilidad de obtener información de muchas encuestas realizadas a los estudiantes que podrían ser beneficiosas para ciertas investigaciones que se podrían llevar a cabo en muchas de las carreras.

La Facultad no posee una fuente centralizada de información para poder acceder a los datos que se suscitaron en previas encuestas que llegan a ser relevantes tanto para los estudiantes, a nivel facultad, como a los docentes o administrativos en las distintas áreas, desconocer la opinión de la comunidad universitaria impide planificar y tomar decisiones con mayor certeza, esto puede terminar en una disconformidad docente-estudiantil.

1.4. OBJETIVO PRINCIPAL

Desarrollar una plataforma de monitoreo de opinión dentro de la población universitaria para facilitar tanto a los estudiantes como docentes la libre y privada opinión de las actividades universitarias en las distintas carreras e institutos dentro los ambientes de la Facultad de Ciencias Puras y Naturales en la UMSA.

1.5. OBJETIVOS SECUNDARIOS

- Seleccionar el software y hardware con el cual se trabajará utilizando tecnologías similares que se utilizan actualmente en los ambientes del ITIC de la Facultad.
- Realizar y obtener los requerimientos del sistema con el ITIC de la Facultad
- Diseñar y desarrollar un entorno responsivo y fácil de entender para el usuario.
- Implementar la plataforma en los servidores del ITIC de la Facultad de Ciencias Puras y Naturales
- Realizar pruebas y analizar resultados con la metodología de calidad QGM.

1.6. JUSTIFICACIÓN

La opinión de los estudiantes es importante para la UMSA debido a que son conscientes de las desigualdades, disparidades, tensiones y conflictos actuales, que son motivo de inquietud para ellos y exigen que sean resueltos, es por eso que su aporte es importante para que la universidad pueda progresar de la mejor manera posible, UQuest será una herramienta vital para obtener información.

- Justificación social: En la facultad no existe un sistema de información dedicada para obtener opiniones de los estudiantes, por lo que el presente proyecto está enfocado a facilitar dicha fuente.
- Justificación tecnológica: El uso de un nuevo sistema orientado específicamente a los estudiantes de las carreras de la facultad es un progreso tecnológico y además para facilitar el acceso a la opinión que nos aportan.
- Justificación económica: El funcionamiento de la plataforma estará en funcionamiento dentro los servidores de la facultad, el costo del sistema es variable pero óptimo.

1.7. LÍMITES Y/O ALCANCES

Para cumplir el objetivo principal de este proyecto de grado, la plataforma cumple con las siguientes funcionalidades:

- ✓ Publicar un tema en específico que posea preguntas varias relevantes para el ámbito académico
- ✓ Desplegar información de manera anónima en valor porcentual de la cantidad de personas que votaron hasta el momento
- ✓ Desplegar información en valor porcentual independientemente para cada pregunta para los resultados
- ✓ Mostrar el número total de encuestados hasta la fecha
- ✓ De no tener preguntas de selección múltiple, mostrar las palabras clave usadas con mayor frecuencia
- ✓ Indicar la fecha límite de la encuesta

- ✓ Mostrar resultados previos de anteriores consultas realizadas
- ✓ En un apartado indicar la cantidad de encuestados en las distintas fechas en los diferentes meses

1.8. DISEÑO METODOLÓGICO

1.8.1. METODOLOGÍA DE INVESTIGACIÓN

La metodología de investigación que se utilizó es la investigación aplicada, para poder direccionar el trabajo actual, las fases a seguir en esta metodología de investigación son planeación, ejecución y comunicación (figura 1).



Figura 1 – Fases de la investigación. (Ministerio de Educación, 2017)

En estas fases se establecen tareas que aseguran el cumplimiento de sus objetivos. El detalle de las mismas se observa en la tabla nro. 1.

Tabla 1 - Fases de investigación aplicada

Planeación	Elección del tema
	Preguntas a responder
	Antecedentes
	Proponer metodología
	Determinar acciones a llevar a cabo
Ejecución	Solución en seguimiento a la metodología
	Recolección de datos e información
	Descripción de pasos realizados
Comunicación	Analizar información
	Publicación de los resultados en algún formato

1.8.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE

La metodología de desarrollo que se utilizó fue Scrum debido a que está basada en la división del trabajo en iteraciones, es decir, fases con objetivos y tareas específicas. Esto hace que necesariamente aporte beneficios en aspectos como los siguientes:

- Gestión de las expectativas del usuario. Los usuarios pueden participar en cada una de las etapas del proceso y proponer soluciones. De hecho, el proceso en su conjunto está pensado para un tipo de evaluación conjunta.
- Resultados anticipados. Cada etapa del proceso arroja una serie de resultados. No es necesario, por tanto, que el cliente espere hasta el final para ver el resultado.
- Flexibilidad y adaptación a los contextos. Se adapta a cualquier contexto, área o sector de la gestión. Es decir, no es una técnica exclusiva de ninguna disciplina.
- Gestión sistemática de riesgos. Del mismo modo, los problemas que aparecen durante los procesos de gestión que pueden afectar a un proyecto son gestionados en el mismo momento de su aparición. Esto es posible debido a que la intervención de los equipos de trabajo puede ser inmediata (Español, 2019).

Scrum es reconocida por ser la metodología ágil más prestigiosa internacionalmente en el sector empresarial y ha tenido una gran aceptación desde su creación, en el año 1992, cuando el teórico norteamericano Jeff Sutherland sentó las bases para su posterior desarrollo. (Español, 2019)

En pocas palabras y apoyándose en la figura nro. 2 las fases operativas de esta metodología son las siguientes:

- Planificación de Sprint. - es un periodo definido de tiempo en el que se hace todo el trabajo.
- StandUp Meeting. - Un tiempo dedicado a saber que se realizó y que se planifica por cada día.
- Incremento del Producto. – el aporte que se resta al producto backlog por cada fase operativa o sprint
- Sprint Review. – Es la fase donde se verifica que fue realizado en cada sprint basándonos en cada parte que se planifico en el product-backlog¹
- Retrospectiva de Sprint. – Esta ocurre después de un Sprint Review y en esta se visualiza que es lo que estuvo bien en el sprint, en que se puede mejorar, y cuál es el compromiso del para mejorar en el siguiente sprint
- Actualización del product-backlog. – En esta fase se actualiza el product-backlog según los avances realizados por cada sprint, agregando y modificando su contenido.

Una vez que se realizan los pasos anteriormente mencionados se realiza otra iteración hasta que el producto que finalizado hasta su lanzamiento.

¹ Un listado ordenado y priorizado de los requisitos necesarios para la implementación de un proyecto.

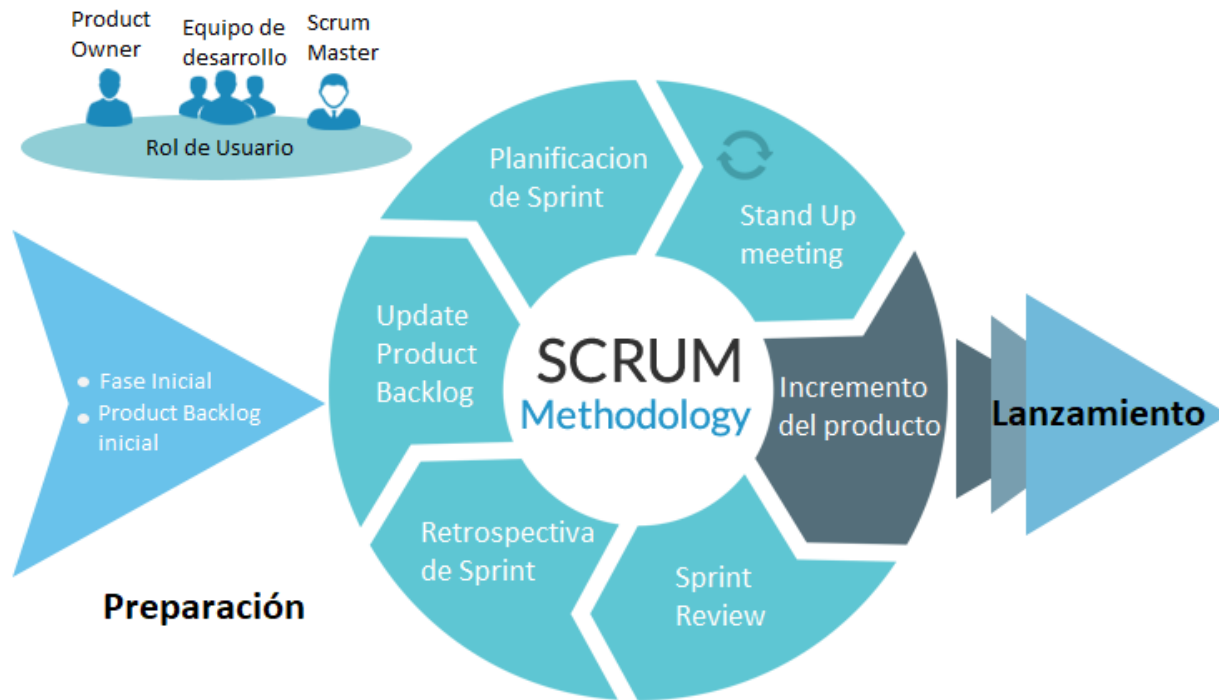


Figura 2 - Fases de la metodología Scrum

1.8.3. TÉCNICAS DE INVESTIGACIÓN

En primer lugar, se realizó un análisis de la tecnología existente en los ámbitos académicos para tener conocimiento del área de enfoque, realizar un estudio de caso identificando el problema principal y por qué necesitamos de esta herramienta o plataforma.

Por otra parte, también necesitamos recaudar información de la Universidad, para esto la mejor herramienta es la de la observación, debido a que no se tiene documentos previos de plataformas similares.

Las encuestas son una herramienta popular utilizada para obtener información de los estudiantes, que servirán para obtener una opinión directa respecto a la necesidad de una herramienta para poder expresar su opinión libremente. Para la aplicación de dicha encuesta necesitamos realizar un estudio previo de la cantidad de estudiantes y realizar una muestra (muestreo estratificado).

1.8.4. INSTRUMENTOS Y ESCALAS DE MEDIDA

Para la evaluación y calidad de software uso del estándar ISO/IEC 9126 y la ISO/IEC 25000, al mismo tiempo se empleó la metodología de evaluación QGM (Objetivo, Pregunta y Métrica según sus siglas en inglés), este fue probado por usuarios de la Facultad de Ciencias Puras y Naturales respondiendo encuestas haciendo uso de la misma plataforma.

La usabilidad es la experiencia que tiene el usuario en la plataforma y que es analizada bajo su comportamiento en éste. En el caso de “UQUEST” se midió la usabilidad una prueba piloto donde se observaron criterios como el nivel de aprendizaje, satisfacción, etc.

En situaciones donde exista una gran demanda del uso del sistema como por ejemplo temporadas de elecciones, una gran cantidad de usuarios desearán ingresar al sistema para poder dejar su opinión y también ver la opinión general de los demás usuarios, el sistema debe estar preparado para recibir toda la información que el usuario envíe sin que se perciba pérdida o algún fallo, para medir la saturación de la plataforma y servidor mediante una conexión simultánea a la plataforma y obtener un límite de usuarios conectados óptimo, este aspecto fue medido gracias a la herramienta de JMeter donde se realizó pruebas de estrés con múltiples peticiones al servidor.

Otro factor importante es la fiabilidad del almacenamiento de la información, es decir, que no existan ataques en el sistema que puedan irrumpir la información y volverla no fiable para el público, y si es que existieran, el sistema debe tener tolerancia a fallos. Se realizó una prueba de seguridad realizando pruebas unitarias en los distintos endpoints² accesibles a la base de datos.

1.9. HERRAMIENTAS

1.9.1. SOFTWARE

Para futuros desarrollos y actualizaciones del proyecto, se utilizó las siguientes tecnologías/herramientas para el desarrollo de “UQUEST”

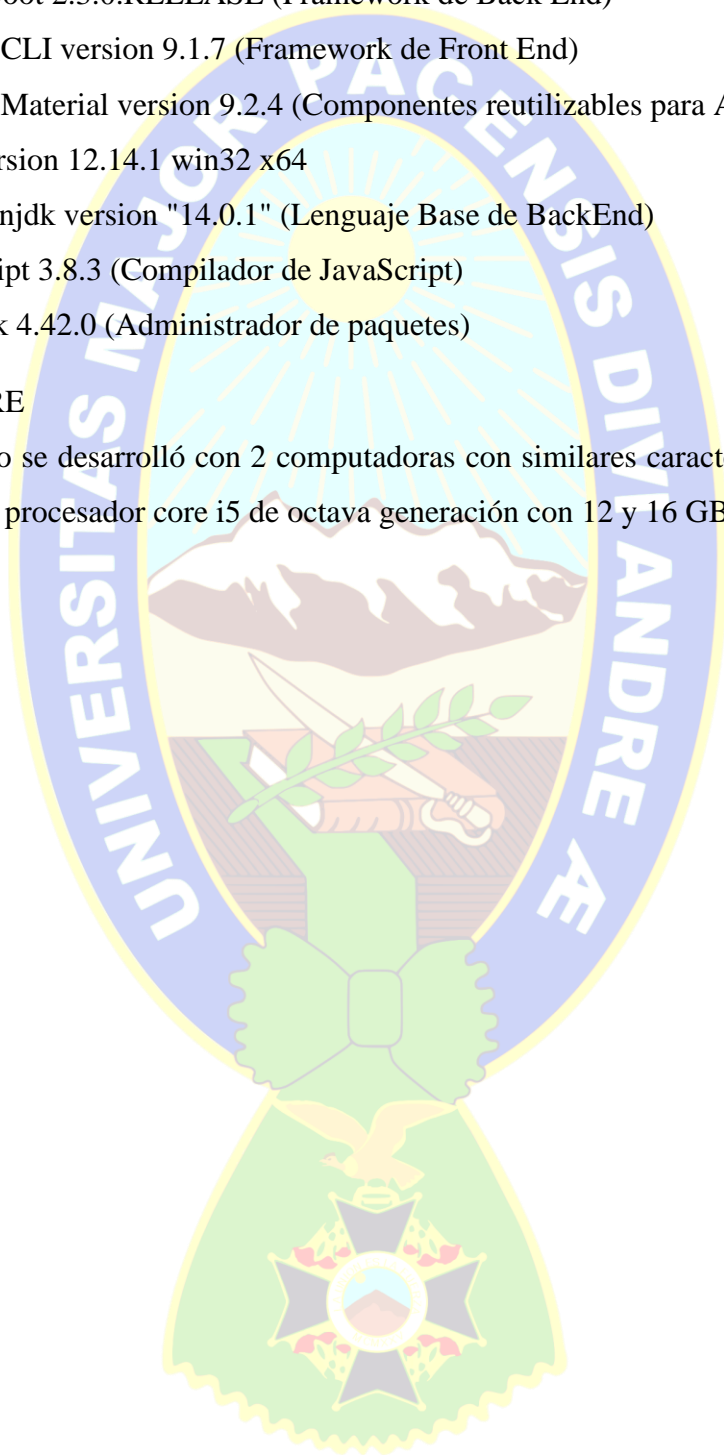
- Microsoft Windows 10 Pro, 64bits Edition (Sistema operativo del equipo de desarrollo)
- IntelliJ IDEA 2020.2.1 Ultimate Edition (Entorno de desarrollo IDE)
- Git versión 2.25.0.windows.1 (Administrador de código)
- GitHub (Plataforma de almacenamiento de código en la nube)

² Característica usada en el backend para definir una URL utilizada para ingresar u obtener información de la base de datos.

- MySQL versión 8.0.19 (Base de Datos)
- MySQL Workbench 8.0.19 CE (64 bits) Community (Administrador de Base de Datos)
- Spring Boot 2.3.0.RELEASE (Framework de Back End)
- Angular CLI version 9.1.7 (Framework de Front End)
- Angular Material version 9.2.4 (Componentes reutilizables para Angular)
- Node version 12.14.1 win32 x64
- Java openjdk version "14.0.1" (Lenguaje Base de BackEnd)
- TypeScript 3.8.3 (Compilador de JavaScript)
- Webpack 4.42.0 (Administrador de paquetes)

1.9.2. HARDWARE

El presente Proyecto se desarrolló con 2 computadoras con similares características; laptop y pc de escritorio con un procesador core i5 de octava generación con 12 y 16 GB de memoria RAM.





CAPÍTULO II

MARCO TEÓRICO

2.1. DESARROLLO MULTIPLATAFORMA

Uno de los mayores retos al que se enfrenta un desarrollador de software a la hora de diseñar su aplicación es la de decidir sobre qué sistema o sistemas operativos podrá funcionar. Existen varias desventajas cuando se desea realizar desarrollo multiplataforma por parte del programador, algunos de los dilemas pueden ser los siguientes:

- Aprender las tecnologías específicas de cada sistema operativo como, por ejemplo:
 - Lenguaje de programación
 - Entorno de desarrollo
 - APIs de cada sistema operativo móvil en específico.
- Tener dominio de las tecnologías anteriormente mencionadas se vuelve más complicado.

El desarrollo multiplataforma, a diferencia del desarrollo nativo, se centra en el reuso de código. La construcción de aplicaciones web móviles constituye un ejemplo que representa este enfoque. Sin embargo, las limitaciones derivadas de su ejecución dentro de un navegador, ha motivado a los ingenieros de software a dirigir su atención hacia otro tipo de aplicaciones multiplataforma con el que se obtienen resultados más cercanos a las soluciones nativas (Delía, 2017).

2.2. HERRAMIENTA DE DESARROLLO MULTIPLATAFORMA

Una herramienta de desarrollo multiplataforma permite a un desarrollador de software escribir aplicaciones para distintos sistemas operativos, generalmente móviles, utilizando una base de código fuente escrita en un solo lenguaje de programación, y trabajando sobre solo un entorno de desarrollo.

Estas herramientas hacen que el desarrollador de software pueda apuntar a más ámbitos de mercado, con la menor inversión de tiempo y dinero, enfocándose simplemente en solo un ámbito de desarrollo.

En resumen, una herramienta de desarrollo multiplataforma es “traducir” el código de un desarrollador de software en un lenguaje de programación, al lenguaje de programación destino que el sistema operativo requiere.

2.3. METODOLOGIAS DE DESARROLLO

Desarrollar un software es una tarea compleja, y si no se utiliza por parte del equipo de trabajo una metodología para su construcción desde la fase inicial, aumenta la probabilidad de obtener un resultado que no satisfaga las expectativas de los clientes y usuarios finales (Boaventura, Peña, Verdecia, & Fustiel, 2016).

La selección de la metodología adecuada para desarrollar determinado proyecto es clave para el éxito del mismo, ya que se llevarían a cabo actividades requeridas de acuerdo las características específicas del proyecto y su entorno.

En la actualidad existen dos grupos de metodologías para el desarrollo de software, las metodologías tradicionales y las metodologías ágiles, las metodologías tradicionales proporcionan importancia al seguimiento y la planificación predictiva, que establecen un rigor en el proceso de desarrollo de software, por otro lado las metodologías ágiles están basadas en entregas frecuentes de versiones de software funcionales, con mayor relevancia a la planificación adaptativa, colaboración con el cliente y respuestas ante los cambios en el desarrollo de software (Boaventura, Peña, Verdecia, & Fustiel, 2016).

2.8.1. METODOLOGIAS TRADICIONALES

Las metodologías tradicionales en el desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de licitación de requerimientos, previo a etapas de análisis y diseño. Con esto se aseguran resultados con alta calidad fijados a fechas acordadas.

En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado el desarrollo del software (Khurana & Sohal, 2011).

2.8.2. METODOLOGIAS ÁGILES

Las metodologías ágiles son alterables con el transcurso del proyecto, flexibles y pueden ser modificadas según la comodidad del equipo.

Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características. Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, de entre dos y seis semanas (Navarro Cadavid, 2013), la comunicación con el cliente es constante al punto de requerir un representante del mismo durante el desarrollo. Los proyectos son altamente colaborativos y se adaptan mejor a los cambios (Ghosh, 2018).

2.8.3. COMPARACION DE METODOLOGIAS

Resulta interesante comparar ambas metodologías de tal forma que se pueda llegar a una decisión al momento de decidir qué metodología se debe usar, y para esto se tiene la tabla nro. 2

Tabla 2. Comparación de metodología ágil y tradicional

Metodologías Tradicionales	Metodologías Agiles
Resistencia a los cambios	Preparados para cambios
Impuestas por el equipo	Impuestas externamente
Arquitectura esencial, expresada mediante modelos	Menos énfasis en la arquitectura del software
Mas roles	Pocos roles
Mas artefactos	Pocos artefactos
Grupos grandes y distribuidos	Grupos pequeños, en el mismo sitio
Proceso controlado, con muchas normas y políticas	Proceso menos controlado, con pocos principios
Proceso rígido	Proceso flexible con adaptación
El cliente interactúa con el equipo de desarrollo	El cliente es parte del equipo de desarrollo
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código

Fuente: (Kuz, Falco, & Giandini, 2018)

2.4. SCRUM

Una de las metodologías más conocidas y más usadas para la gestión de proyectos es la metodología Scrum, debido a que los proyectos se los planifica en bloques o “Sprints”.

La metodología Scrum se centra en ajustar sus resultados y responder a las exigencias reales y exactas del cliente. De ahí, que se vaya revisando cada entregable, ya que los requerimientos van variando a corto plazo. El tiempo mínimo para un Sprint es de una semana y el máximo es de cuatro semanas.

Scrum es un proceso, marco de trabajo o framework, usado en equipos que trabajan en proyectos complejos; una metodología de trabajo ágil que tiene como finalidad la entrega de valor en períodos cortos de tiempo, basada tres pilares: la transparencia, inspección y adaptación (Araque, 2017).

Es de suma importancia respetar y hacer cumplir los roles en cada proceso del proyecto para hacer una adecuada aplicación de la metodología SCRUM. Es por eso que en este artículo explico los roles de cada uno de los integrantes del equipo y cuáles son las responsabilidades de cada uno.

- ❖ Propietario del Producto (Product Owner): Representa los intereses de la comunidad de interesados. Es la Voz del Cliente, dado a que asegura que las necesidades explícitas e implícitas del cliente se conviertan en Historias de Usuario en la Lista Priorizada de Pendientes del Producto y posteriormente sean utilizadas para crear los entregables del Proyecto. Debe comprender y apoyar los intereses de todas las partes interesadas y también entender las necesidades y el funcionamiento del Equipo Scrum.
- ❖ Scrum Master: Su principal responsabilidad es la garantizar que todos los miembros del Equipo Scrum sigan correctamente los procesos Scrum, incluyendo el Propietario del Producto. Debe asegurar que la gestión de proyectos avance sin problemas, y que los miembros del Equipo Scrum cuenten con las herramientas necesarias para realizar el trabajo. Debe ser un Facilitador y Líder Servicial.
- ❖ Equipo Scrum (Scrum Team): Es un grupo o equipo de personas que son responsables de entender los requerimientos del negocio especificados por el Propietario del Producto, estimar Historias de Usuario y crear los Sprints del Proyecto. Deben ser Auto-organizados,

Interfuncionales, utilizar comunicación frontal y realizarán entregas iterativas del producto.

El Equipo Scrum es el núcleo de cualquier Proyecto Scrum y seleccionar a los miembros apropiados es importante para una entrega exitosa de los proyectos. Esto no puede hacerse de forma aislada, sino que el Propietario del Producto y el Scrum Master deben participar de manera conjunta en la selección del Equipo Scrum. (Bahit, 2011).

2.5. UWE (*UML-Based Web Engineering*)

El enfoque UWE conocido por sus siglas en inglés, “UML-based web engineering” proporciona una notación específica de dominio, un proceso de desarrollo basado en modelos y soporte de herramientas para la ingeniería de aplicaciones web. La característica de UWE es el hecho de ser un enfoque basado en estándares que no se limita al uso de la "lengua franca" UML, sino que también utiliza XMI como formato de intercambio de modelos, MOF para meta-modelado, los principios basados en modelos del enfoque MDA, el lenguaje de transformación de modelos QVT y XML. (Maximilians, 2016)

Existen dos clases principales de tipos de diagramas: diagramas estructurales y diagramas de comportamiento. Estos últimos incluyen varios que representan diferentes aspectos de las interacciones. Estos diagramas pueden ser categorizados jerárquicamente como se muestra en el siguiente diagrama de clases:

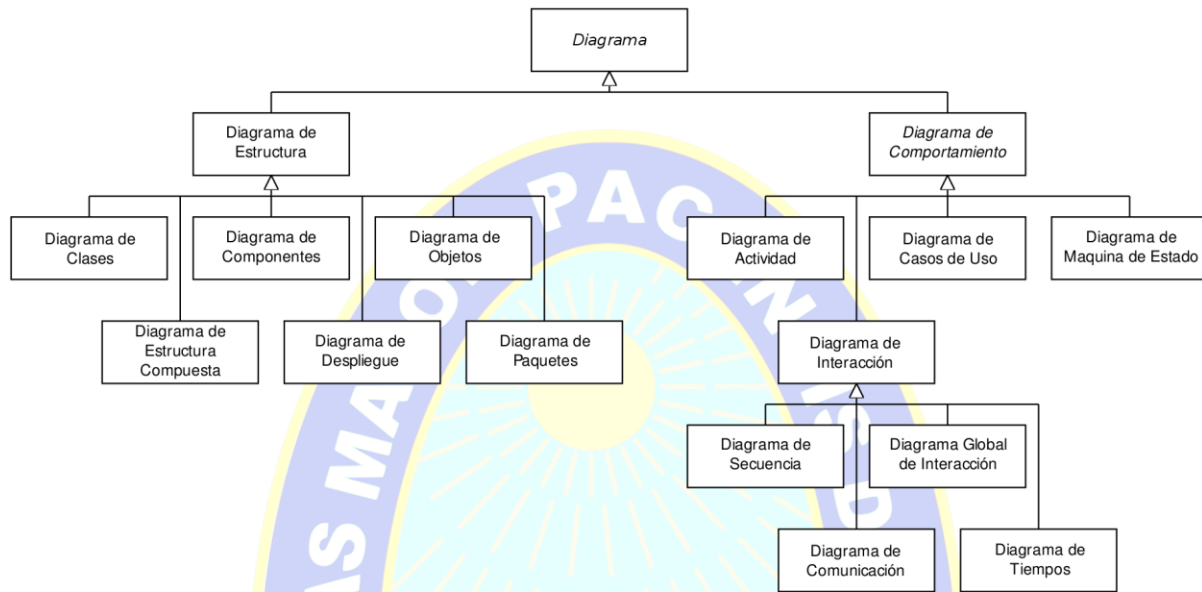


Figura 3 - Jerarquía UML

En el trabajo actual se usaron diagramas dentro de la jerarquía UML que se consideraron relevantes los siguientes diagramas:

- Diagrama de Clases
- Diagrama de Secuencia
- Diagrama de Casos de Uso
- Diagrama de Iteración o de Secuencia

2.5. SPRING MODELO VISTA CONTROLADOR (MVC)

La arquitectura MVC es uno de los principales componentes y tecnológicas, como su propio nombre nos indica implementa una arquitectura MVC (Modelo – Vista – Controlador). Soporta varias tecnologías para la generación de las vistas, entre ellas JSP, Thymeleaf, FreeMarker, Velocity, Tiles, iText y POI (Java API para archivos Microsoft Office), entre ellas, las que se usará en el actual proyecto será JSP y Thymeleaf.

El controlador se encarga básicamente de controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista. El modelo se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, etc. todo eso va aquí, en el modelo. La vista es la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. ni el modelo ni el controlador se preocupan

de cómo se verán los datos, esa responsabilidad es únicamente de la vista. (Hernandez, 2015), el MVC nos facilita mucho las cosas en entornos de trabajo donde cada elemento tiene que estar relacionado con otros elementos, y para su fácil manipulación (véase Figura 4.) y de ubicación de elementos precisos haremos uso de MVC

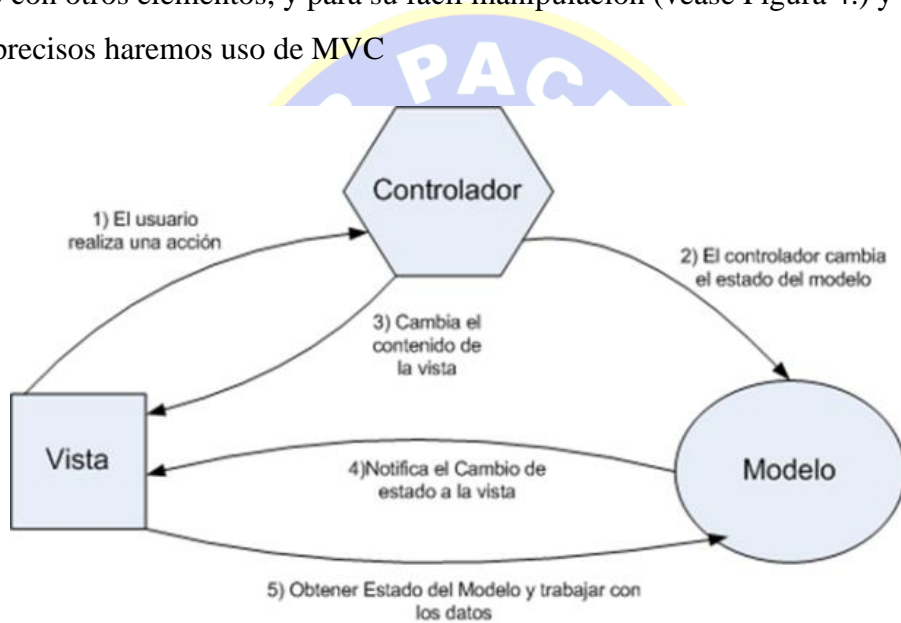


Figura 4 - Patron MVC (Model View Controller)

2.6. FRAMEWORK SPRING

Un framework (marco de trabajo) es un conjunto de convenciones estándares paradigmas, buenas prácticas, y funcionalidades costosas ya desarrolladas, ahorrándonos trabajo, facilitando el código, seguir los estándares. Si una persona quiere continuar el proyecto que está bajo el mismo Framework será mucho más fácil habituarse debido a funcionalidades ya conocidas, de esta forma mejorar la productividad del software en desarrollo evitando tareas repetitivas.

Spring es un framework en JAVA y de software libre el cual su principal característica es brindar un “Factory” de objetos basados en inyección de dependencia, es un robusto Framework para el desarrollo de Aplicaciones Empresariales en el lenguaje JAVA, para el actual proyecto se usará Spring Framework 5 junto con Spring Boot.

Spring posee modularidad de componentes a través de inyección de dependencia promoviendo la composición y modularidad entre las partes que componen una aplicación, las clases POJO ³ mantienen su código limpio, simple y modular, bajo acoplamiento y alta cohesión.

Las aplicaciones con Spring son simples y requieren mucho menos código (Java y XML), las clases POJO pueden ser testeadas unitariamente sin estar atadas al framework.

2.7. SPRING BOOT

Spring Boot es una infraestructura ligera que elimina mayor parte del trabajo de configurar aplicaciones basadas en Spring, es decir que usa una configuración mínima para poder crear aplicaciones que simplemente puedes ejecutar (Perry, 2017).

La ventaja de Spring Boot, a diferencia de Spring MVC es más fácil, evita configuraciones, mucha parte del código viene preconfigurado, además de traer manejo de dependencias, y un servidor embebido.

En resumen, Spring Boot facilita la programación minimizando configuraciones tanto del servidor, dependencias, despliegue, enfocándonos más en nuestra aplicación y la lógica de negocios, manteniendo la arquitectura básica de Modelo – Vista – Controlador.

2.8. INYECCIÓN DE DEPENDENCIAS

El objetivo es lograr un bajo acoplamiento entre los objetos de nuestra aplicación. Con este patrón de diseño, los objetos no crean o buscan sus dependencias (objetos con los cuales colabora) sino que éstas son dadas al objeto. El contenedor (la entidad que coordina cada objeto en el sistema) es el encargado de realizar este trabajo al momento de instanciar el objeto. Se invierte la responsabilidad en cuanto a la manera en que un objeto obtiene la referencia a otro objeto.

De esta manera, los objetos conocen sus dependencias por su interfaz. Así la dependencia puede ser intercambiada por distintas implementaciones a través del contenedor. En resumen, programaremos orientado a interfaces e inyectaremos las implementaciones a través del contenedor.

³ “Plain Old Java Object” en español hace referencia a clases simples en java y que terminan siendo independientes y trabajan solos ante cualquier framework

Inyectar es justamente suministrar a un objeto una referencia de otros que necesite según la relación, El mecanismo de inyección (injection) permite a un componente A obtener de un contexto una referencia de una instancia de un componente B Haciendo que el contenedor de aplicaciones “inyecte” el componente B en una variable del componente A, en tiempo de ejecución. (Guzmán, 2018)

2.9. HIBERNATE

Hibernate es una herramienta de Mapeo objeto-relacional (ORM⁴) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.

En otras palabras, Hibernate es un Framework que agiliza la relación entre la aplicación y la base de datos. Para poder aprender a utilizarlo es necesario contar con los conocimientos básicos de base de datos y SQL, así como manejar el lenguaje Java. (Guzmán, 2018).

2.10. ANGULAR

Angular es un framework open-source desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application).

Angular separa completamente el Frontend y el BackEnd en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones. (Devs, 2019).

Este tipo de páginas SPA suelen ser un poco lentas en la primera carga, debido a que carga toda la página, subpáginas y/o submenús que puede llegar a tener, pero una vez que carga se obtiene mayor fluidez en la página debido a que se trata de realizar la menor carga de vistas continuas.

⁴ “ORM” Es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador.

2.11. ANGULAR MATERIAL

Angular Material es una librería de estilos (como Bootstrap) basada en la guía de diseño de Material Design, realizado por el equipo de Angular para integrarse perfectamente con Angular.

una guía de estilo creada por Google para Android y para sus aplicaciones. En los últimos tiempos se ha puesto muy de moda el estilo material design, y han salido un montón de librerías y frameworks que lo implementan, entre ellas Angular Material.

En el actual proyecto se hizo uso de esta librería para facilitar el desarrollo de componentes en cuanto a accesibilidad, fiabilidad y estética, para que de esta forma resulte más sencillo y atractivo para los usuarios de UQUEST

2.12. JSON WEB TOKEN (JWT)

JSON Web Token es un estándar abierto (RFC 7519) que define una forma compacta e independiente de transmitir la información de forma segura entre partes como un objeto JSON.

Esta información se puede verificar y confiar porque está firmada digitalmente. Los JWT se pueden firmar utilizando una clave secreta (un algoritmo HMAC) o un par de claves públicas y privadas mediante RSA o ECDSA. (JWT, n.d.)

Estos tokens se dividen en tres partes los cuales se los menciona en inglés y son los siguientes:

- Header (Cabecera)
- Payload (Carga Útil)
- Signature (Firma)

Por parte de la cabecera podemos obtener el siguiente tipo de contenido:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

El cual denota como “alg” al tipo de algoritmo como HMAC SHA256 o RSA, y por otra parte “typ” señala el tipo de del token, en este caso JWT.

El payload contiene las notificaciones o reclamos conocido en inglés como “claims” contiene instrucciones o información de las entidades generalmente usada para almacenar al usuario y otros datos adicionales, existen tres tipos de claims: registered claims, public claims y private claims, en la siguiente figura podemos ver un pequeño ejemplo de payload:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

Por último, se tiene la firma (Signature) el cual para crear la parte de firma debe tomar el encabezado codificado, el payload, una clave secreta, el algoritmo especificado en el encabezado y firmarlo.

```
HMACSHA256 (
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

Y como resultado se obtiene el JWT final con la información anteriormente detallada como ejemplo:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gR91IiwiaXNjb2NpYWwiOiOnRydWV9.4pcPyMD09o1PSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

2.13. BCrypt

Bcrypt es una función de hash de contraseña basado en la criptografía Blowfish (codificador de bloques simétricos) para desarrollar un nuevo tipo de algoritmo llamado “eksblowfish” el cual se puede editar la cantidad de rondas a encriptar una contraseña el cual fue utilizada para el actual sistema, este tipo de cifrado es lento pero seguro y garantiza seguridad con las contraseñas utilizadas usando la siguiente estructura:


```
$2a$10$N9qo8uLOiCKgX2ZMRZoMyeIjZAgcf17p92ldGxad68LJZdL17lhwy
```

Alg Cost Salt Hash

Donde:

2.14. POSTMAN

Postman es una herramienta que se utiliza, sobre todo, para el testing de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de sistemas.

Gracias a esta herramienta, además de testear, consumir y depurar API REST, podremos monitorizarlas, escribir pruebas automatizadas para ellas, documentarlas, *mockearlas*⁵, simularlas, etc. (López, 2019)

2.15. HERRAMIENTAS DE DESARROLLO

- a. jQuery es una biblioteca de JavaScript que simplifica la forma de desarrollar aplicaciones web. Las aplicaciones que utilizan jQuery suelen necesitar menos tiempo y menos código que las aplicaciones hechas con JS puro además permite manipular elementos del DOM (textos, imágenes, enlaces, etc.), cambiar el diseño CSS o realizar peticiones Ajax utilizando instrucciones simples, a través de un código muy conciso y sencillo.
- b. AJAX significa JavaScript asíncrono y XML (*Asynchronous JavaScript and XML*). Es un conjunto de técnicas de desarrollo web que permiten que las aplicaciones web funcionen de forma asíncrona, procesando cualquier solicitud al servidor en segundo plano.
- c. IntelliJ es un IDE (Entorno de desarrollo integrado) para el desarrollo de programas informáticos. Esta tiene dos versiones, la versión “*Community Edition*” que sería la versión gratuita, la que se estaría usando en el actual proyecto que es la “*Ultimate Edition*” el cual es la versión de pago. La razón por la cual se escogió este IDE entre muchos se debe a que tiene un ambiente muy amigable para el desarrollo en el lenguaje JAVA.

⁵ Mock.- un mock te indica algo como «espero que llames a este método y que si lo haces con un específico argumento de entrada, obtendrás una respuesta concreta».

2.16. ATRIBUTOS DE CALIDAD

El presente proyecto debe ser un producto de calidad, para ello se optó por cumplir con los estándares catalogados en la ISO/IEC 9126 y la ISO/IEC 25000. SQuaRE (Software Product Quality Requirements and Evaluation) establece varios factores a tomar en cuenta para la evaluación de productos de software con el objetivo de guiar el desarrollo y evaluación de requisitos de calidad; a continuación, se presenta la figura nro. 5 extraída de la página oficial (ISO25000, 2020) el cual presenta características con mayor uso.

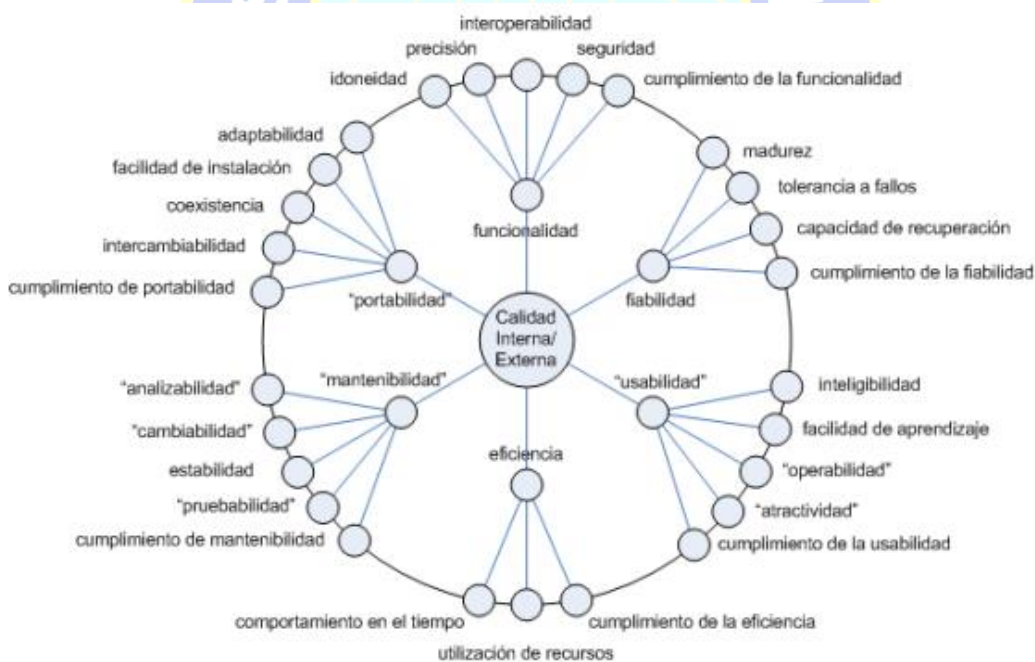
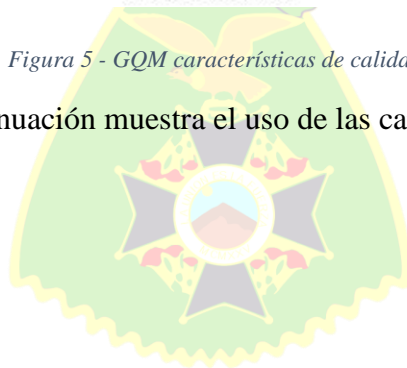


Figura 5 - GQM características de calidad

La siguiente tabla nro. 3 a continuación muestra el uso de las características de calidad con más detalle:



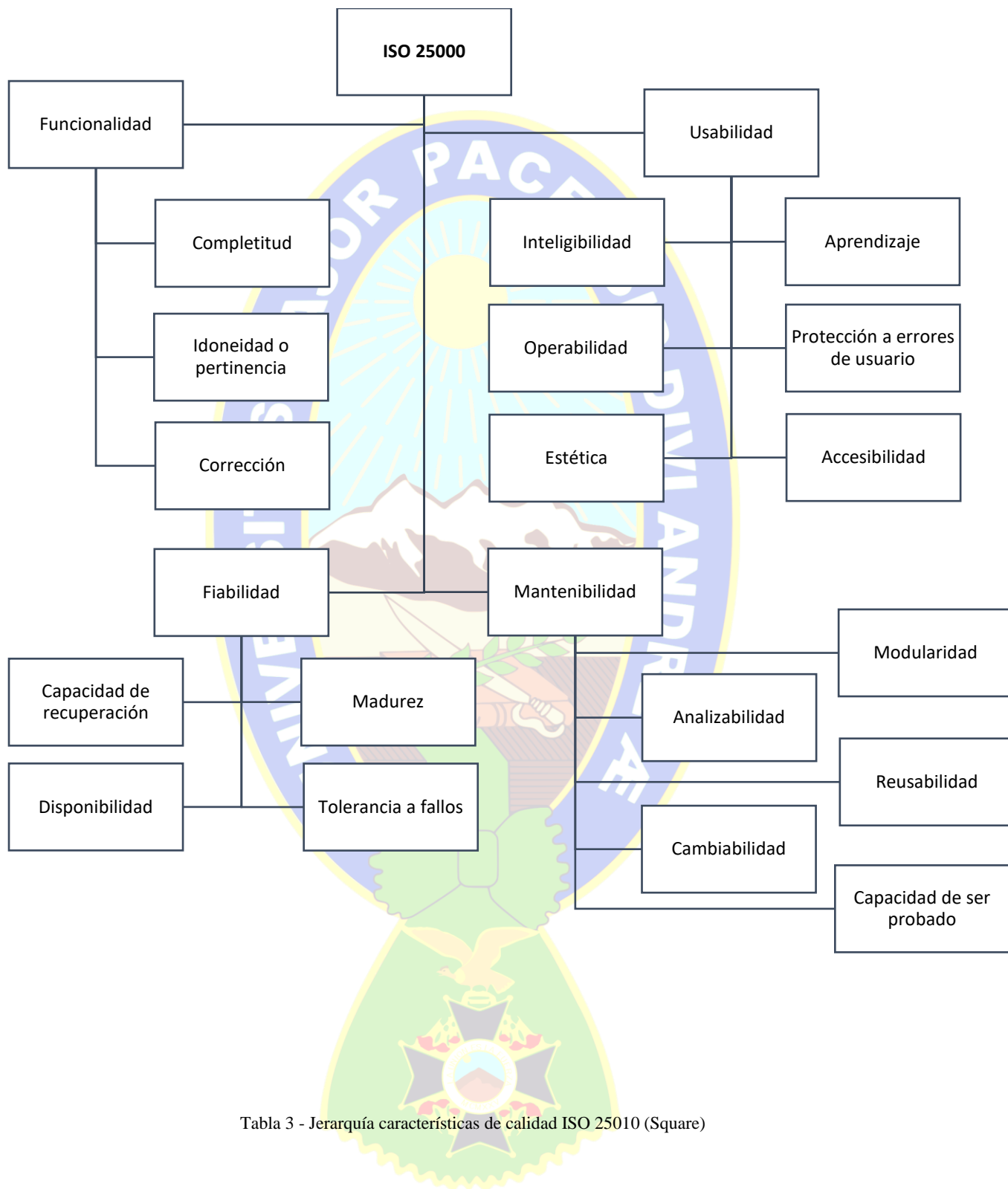


Tabla 3 - Jerarquía características de calidad ISO 25010 (Square)

El estándar ISO/IEC 2500 (ISO25000, 2020) detalla estas características y a continuación se menciona y explica elementos que son pertinentes para este tipo de sistema web:

- Funcionalidad: la capacidad que tiene un producto software para otorgar las funciones que satisfacen los requerimientos planteados.

Compleitud funcional	Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
Corrección funcional	Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
Pertinencia funcional	Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

- Usabilidad: la capacidad que tiene un producto software para ser entendido, aprendido, operable, atractivo para el usuario y conforme a estándares/guías, cuando es utilizado bajo unas condiciones específicas.

Atractividad	capacidad de la interfaz de usuario de agrandar y satisfacer la interacción con el usuario
Facilidad de aprendizaje	capacidad del producto que permite al usuario aprender su aplicación
Operabilidad	capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
Protección contra errores de usuario	capacidad del sistema para proteger a los usuarios de hacer errores.
Inteligibilidad	capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.

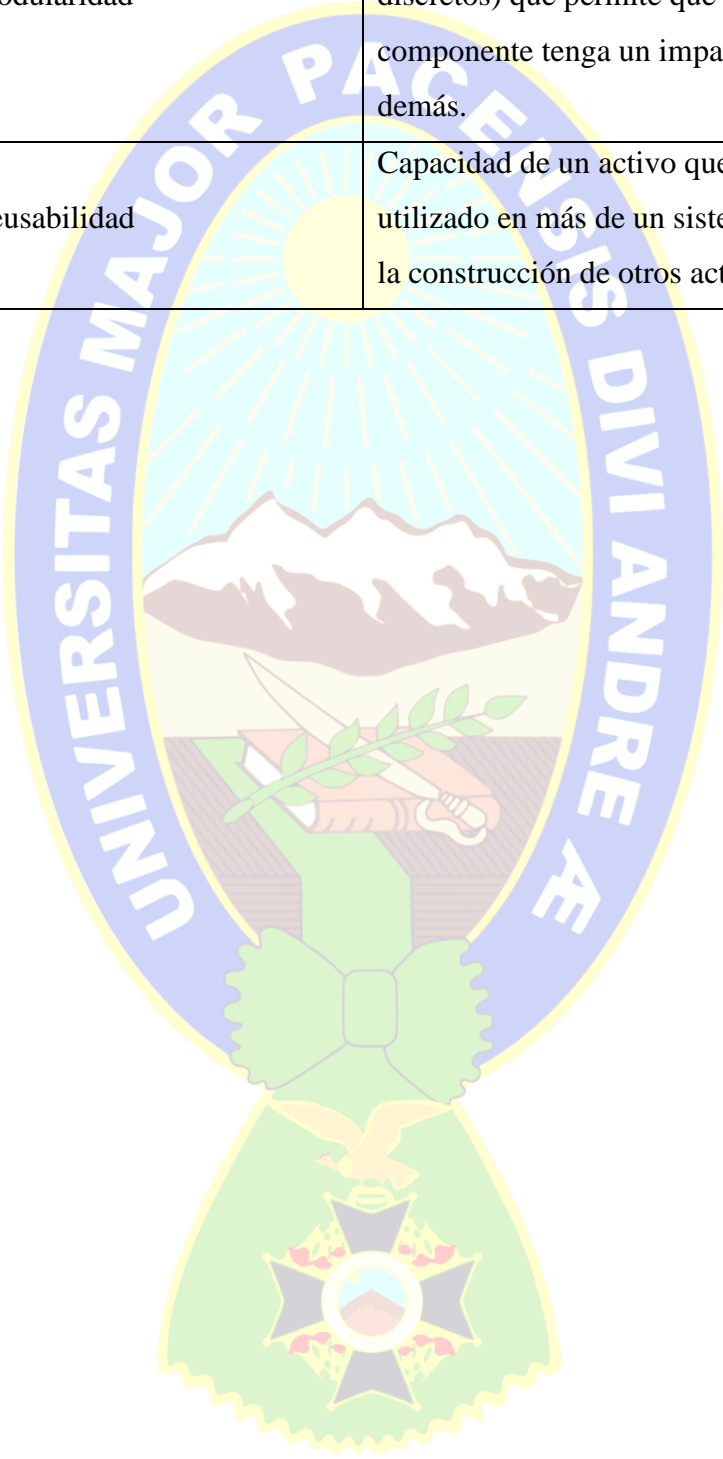
- Eficiencia: en otras palabras, la eficiencia en la ejecución, almacenamiento, y el uso de los recursos del sistema

Comportamiento en el tiempo	los tiempos de respuesta y procesamiento y las tasas de throughput de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (benchmark) establecido.
Utilización de recursos	las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
Capacidad o cumplimiento	grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

- Mantenibilidad: es el esfuerzo requerido para localizar y arreglar un fallo en el programa dentro de su entorno operativo

Analizabilidad	facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
Cambiabilidad	Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
Pruebabilidad	Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

<p>Modularidad</p>	<p>Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.</p>
<p>Reusabilidad</p>	<p>Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.</p>



CAPÍTULO III



MARCO APLICATIVO

3.1. RESUMEN

Este capítulo se enfoca más en el desarrollo del sistema utilizando las metodologías anteriormente mencionadas para su elaboración se tomó en cuenta un análisis de requerimientos obtenido en el TIC Facultativo. Para el desarrollo del sistema se dividieron en Sprints que constan de módulos en cada uno; módulo de diseño e implementación de base de datos, módulo de administración de cuentas de usuario, módulo de registro y *login* de usuario, módulo administración de carreras e institutos, módulo administración y creación de encuestas y módulo de respuesta de encuestas, cada uno especificara los diagramas necesarios para su elaboración y desarrollo, y posteriormente la planeación y desarrollo de ventanas por parte del usuario para el uso del sistema.

3.2. OBTENCIÓN DE REQUERIMIENTOS

La obtención de requerimientos se realizó junto con el encargado del TIC Facultativo (Tecnologías de la Información y comunicación) y también las prioridades de cada uno de estos detallando las necesidades que pueden tener los estudiantes en las distintas carreras para el posterior desarrollo del sistema, obteniendo los resultados que se pueden apreciar en la Tabla 4.

Tabla 4 Tabla de requerimientos

ID	Requerimientos	Prioridad	Descripción
R1	Autenticación	Media	El sistema debe tener incluido un sistema de autenticación para poder identificar entre un administrador y un usuario utilizando una autenticación con Spring Security o Google, tomar en cuenta que no es obligatorio.
R2	Administración de Encuestas	Alta	El usuario tiene la opción de realizar encuestas nuevas una vez ingresado en el sistema. Por parte del administrador debe tener las mismas opciones

			del usuario, pero para todos los usuarios <i>logueados</i> y también encuestas creadas por el mismo administrador.
R3	Administración de usuarios	Media	Es necesario que el administrador pueda contar con este módulo, tanto como para poder asignar un rol y además adicionar eliminar y editar usuarios.
R4	Desplegar resultados de encuestas	Alta	En este módulo se muestran los resultados actuales o finales de una encuesta, desplegando información estadística de las encuestas, dependiendo el tipo de pregunta.
R5	Clasificación los usuarios	Baja	Los usuarios pueden ser de dos tipos, administrador, que puede editar, eliminar a otros usuarios, y el usuario, que es el que ingresara al sistema creándose una cuenta por sí mismo
R6	Categorización de encuestas	Media	Las encuestas deben pertenecer a un área para poder obtener información que sea relevante para los usuarios, y tener estadísticas concisas.
R7	Encuestas abiertas y cerradas	Alta	Los usuarios pueden responder encuestas de manera abierta, para eso la plataforma tendrá a

			disposición publica dicha encuesta donde usuarios que no necesiten ingresar al sistema podrán responder, además se tendrán encuestas de forma cerrada donde solo usuarios con cuenta podrán responder.
R8	Múltiples tipos de preguntas - respuestas	Media	Cada pregunta deberá tener un tipo de respuesta, estos estarán enlazados porque las respuestas tienen el mismo tipo de pregunta al ser generada, es decir una categorización de preguntas y/o respuestas

3.2. PRODUCT-BACKLOG

A partir de la obtención de requerimientos en la Tabla 3. se realizó el análisis correspondiente junto con el ITIC de la Facultad para la obtención del Product Backlog que se puede apreciar en la Tabla nro. 5.

Tabla 5 Product-Backlog

ID	Nombre	Prioridad	Módulo
R1	Diseño de tablas en la base de datos	Alta	Módulo de diseño e implementación de base de datos
R2	Creación de controladores básicos para el uso DAO	Alta	Módulo de diseño e implementación de base de datos
R3	Creación de los servicios	Alta	Módulo de diseño e implementación de base de datos

R4	Enlazar Servicios con CRUD	Media	Módulo de diseño e implementación de base de datos
R5	Diseño de <i>login</i> y <i>logout</i>	Media	Módulo de registro y <i>login</i> de usuario
R6	Creación de controlador de user para uso de CRUD	Alta	Módulo de registro y <i>login</i> de usuario
R7	Autenticación	Alta	Módulo de registro y <i>login</i> de usuario
R8	Asignación de roles	Media	Módulo de administración de cuentas de usuario
R9	Validar roles	Media	Módulo de administración de cuentas de usuario
R10	Creación de controlador de carreras e institutos para uso CRUD	Media	Módulo de administración de carreras e institutos
R11	Asignación de carreras e institutos a usuarios	Media	Módulo de administración de carreras e institutos
R12	Diseño en la creación de encuestas	Alta	Módulo de administración y creación de encuestas
R13	Creación de controlador encuesta para uso de CRUD	Alta	Módulo de administración y creación de encuestas
R14	Creación de encuestas	Alta	Módulo de administración y creación de encuestas

R15	Desplegar encuestas realizadas por usuario	Alta	Módulo de despliegue de encuestas
R16	Filtro de encuestas abiertas y cerradas para los usuarios	Alta	Módulo de despliegue de encuestas
R17	Información estadística en tiempo real de las encuestas	Alta	Módulo de despliegue de encuestas

3.3. ROLES SCRUM

Para el actual proyecto se tomaron en cuenta los siguientes roles para el desarrollo según la metodología scrum:

- Product Owner: Lic. Jesús Reynaldo Perez Benavides
- Scrum Master: M.Sc. Rosa Flores Morales
- Equipo de desarrollo: Ronald Alcides Guarachi Enriquez

3.4. PLANIFICACIÓN DE SPRINTS

Para la planificación de los Sprints se tendrá en cuenta los módulos incluidos en el Product-Backlog.

- Primer Sprint: Módulo de registro y *login* de usuario
- Segundo Sprint: Módulo de administración de cuentas de usuario
- Tercer Sprint: Módulo de administración de carreras e institutos
- Cuarto Sprint: Módulo de administración y creación de encuestas
- Quinto Sprint: Módulo para responder encuestas

3.5. DESARROLLO DE SPRINTS

En esta sección se describirán los Sprints a realizar con rangos de fechas y tiempos a concluir junto con sus diagramas de clase y casos de uso respectivos para cada Sprint.

3.5.1. PRIMER SPRINT: MÓDULO DE DISEÑO E IMPLEMENTACIÓN DE BASE DE DATOS

En este Sprint se realizará la parte del, diseño, estructuración, e implementación de la base de datos, el cual la Tabla nro. 6 se representan las tareas junto a la duración respectiva y el estado del mismo.

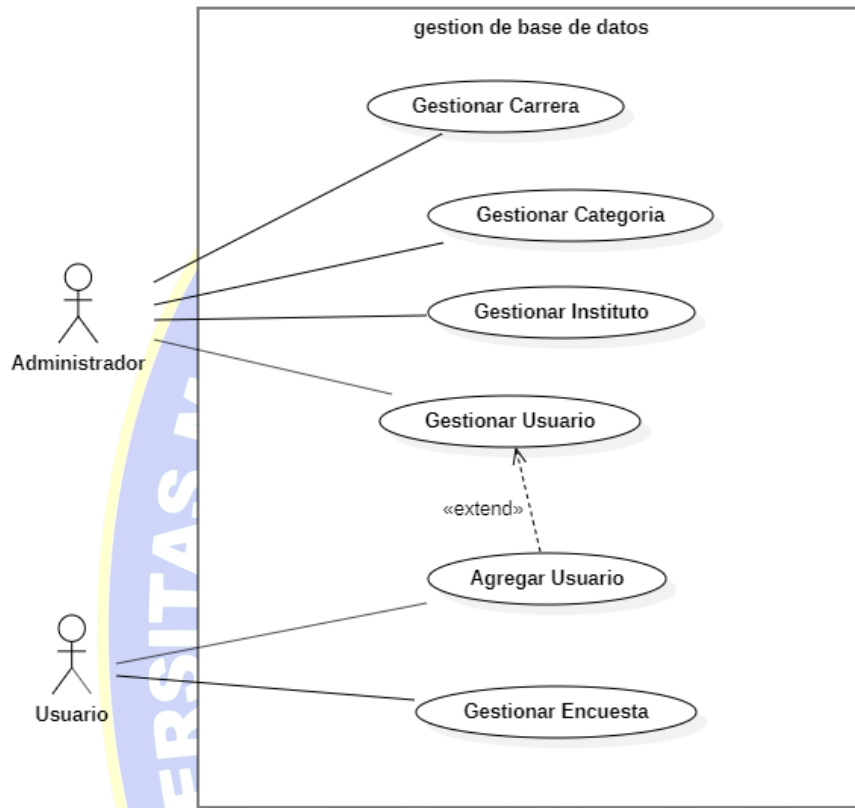
Tabla 6 Planificación Sprint N° 1

Sprint N°1. Módulo de diseño e implementación de base de datos		Inicio	Fin	Duración
		25/11/2019	09/12/2019	14 días
Nro.	Tarea	Duración Tarea		Estado
1	Diseño de tablas en la base de datos	96 horas		Completado
2	Creación de controladores	48 horas		Completado
3	DAO de las tablas	24 horas		Completado
4	Creación de los servicios	48 horas		Completado
5	Enlazar Servicios con CRUD	36 horas		Completado
6	Integrar Servicios dentro los controladores	48 horas		Completado
7	Test de servicios independientes	24 horas		Completado

3.5.1.1. DIAGRAMA DE CASOS DE USO.

La siguiente figura representa la manera de uso de la base de datos, indicando que solo el administrador tendrá acceso directo a cada una de las opciones, como se puede mostrar en la Figura nro. 6.

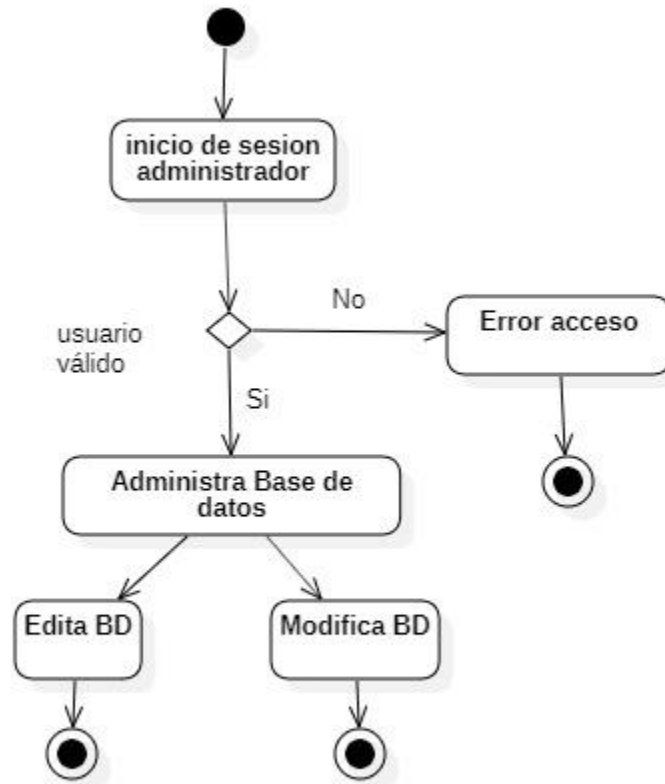
Figura 6 - Gestión base de datos, caso de uso



3.5.1.2. DIAGRAMA DE ESTADOS.

Se realizó el diagrama de estados para la validación de un administrador para poder ingresar al sistema y poder realizar modificaciones, adiciones y demás estados en la base de datos sin el uso directo de una consola MySQL, el cual se visualiza en la Figura nro. 7.

Figura 7 - Gestión base de datos, diagrama de estados



3.5.2. SEGUNDO SPRINT: MÓDULO DE REGISTRO Y LOGIN DE USUARIO

Este sprint es exclusivo para el apartado de registro e ingreso de un usuario en el sistema, para eso tenemos la Tabla 7 detallando las actividades a realizar.

Tabla 7 - Planificación Sprint nro. 2

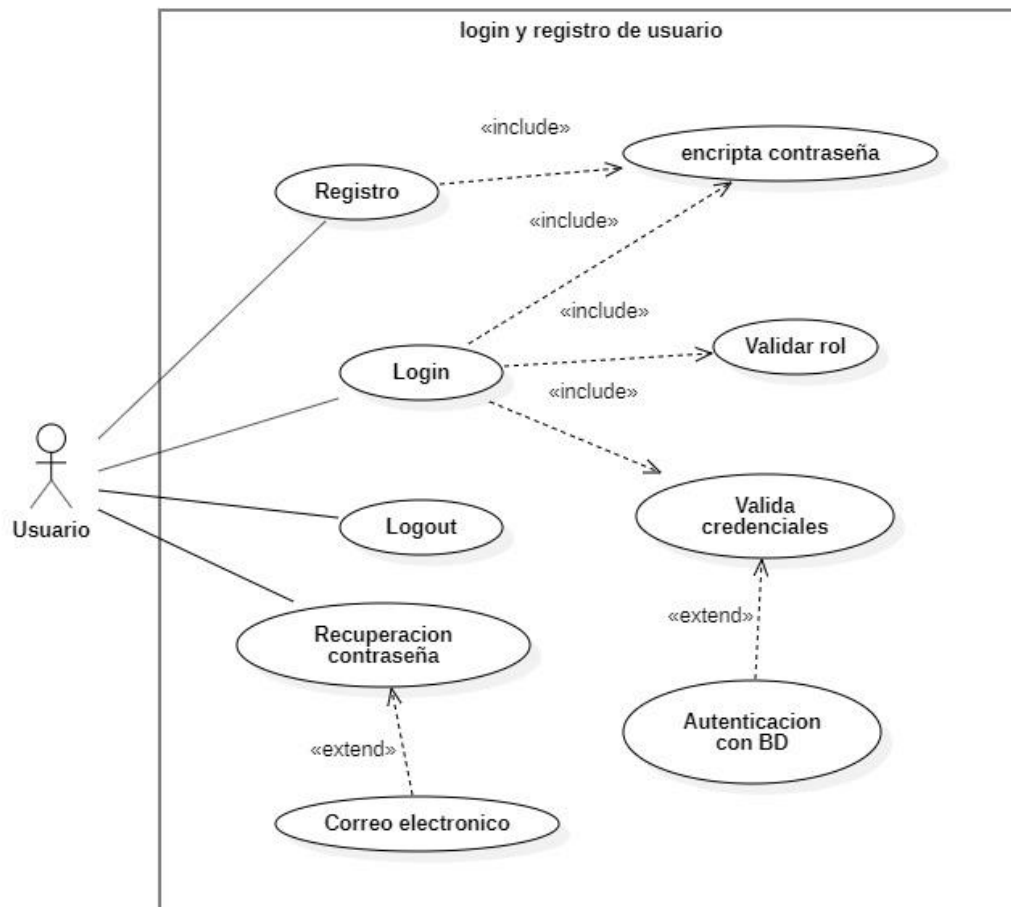
Sprint N°1. Módulo de registro y login de usuario		Inicio	Fin	Duración
		09/12/2019	23/12/2019	14 días
Nro.	Tarea	Duración Tarea		Estado
1	Interfaz de registro de usuario	48 horas		Completado
2	Interfaz y lógica de inicio de sesión	48 horas		Completado
3	Interfaz y lógica de cerrar sesión	48 horas		Completado
4	Implementación y desarrollo de recuperación de contraseña	62 horas		Completado

5	Seguridad implementada con Spring Security	48 horas	Completado
6	Spring Security + JWT (Json Web Token)	48 horas	Completado
7	Test de servicios usando Postman	24 horas	Completado

3.5.2.1. DIAGRAMA DE CASOS DE USO

En este caso, para un usuario no administrador se tendrá también la misma interfaz de *logueo*, pero solo será accesible con los roles de “user”, para mayor entendimiento se puede apreciar en la Figura nro. 10 a continuación.

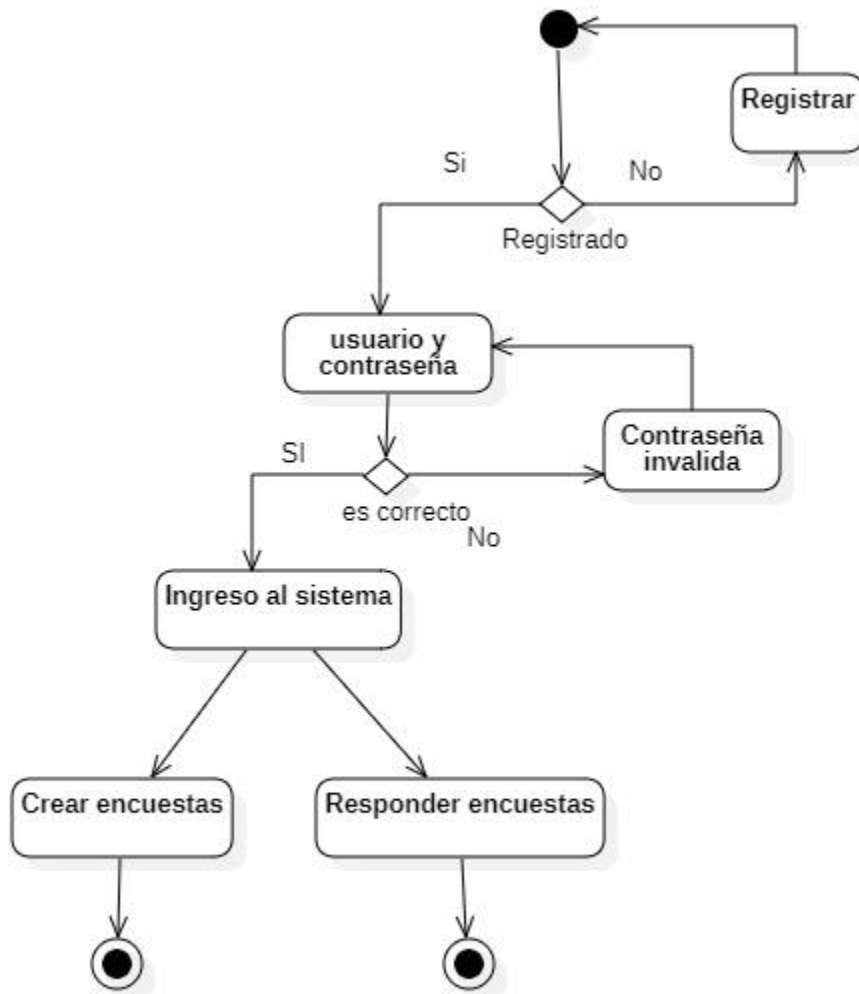
Figura 8 - Registro y Login, casos de uso



3.5.2.2. DIAGRAMA DE ESTADOS

Se realizó el diagrama de estados para los requisitos: interfaz y lógica de registro de usuarios, interfaz y lógica de iniciar sesión, interfaz y lógica cerrar sesión, implementación y desarrollo de recuperación de contraseña, lo cual se visualiza en la Figura nro. 11.

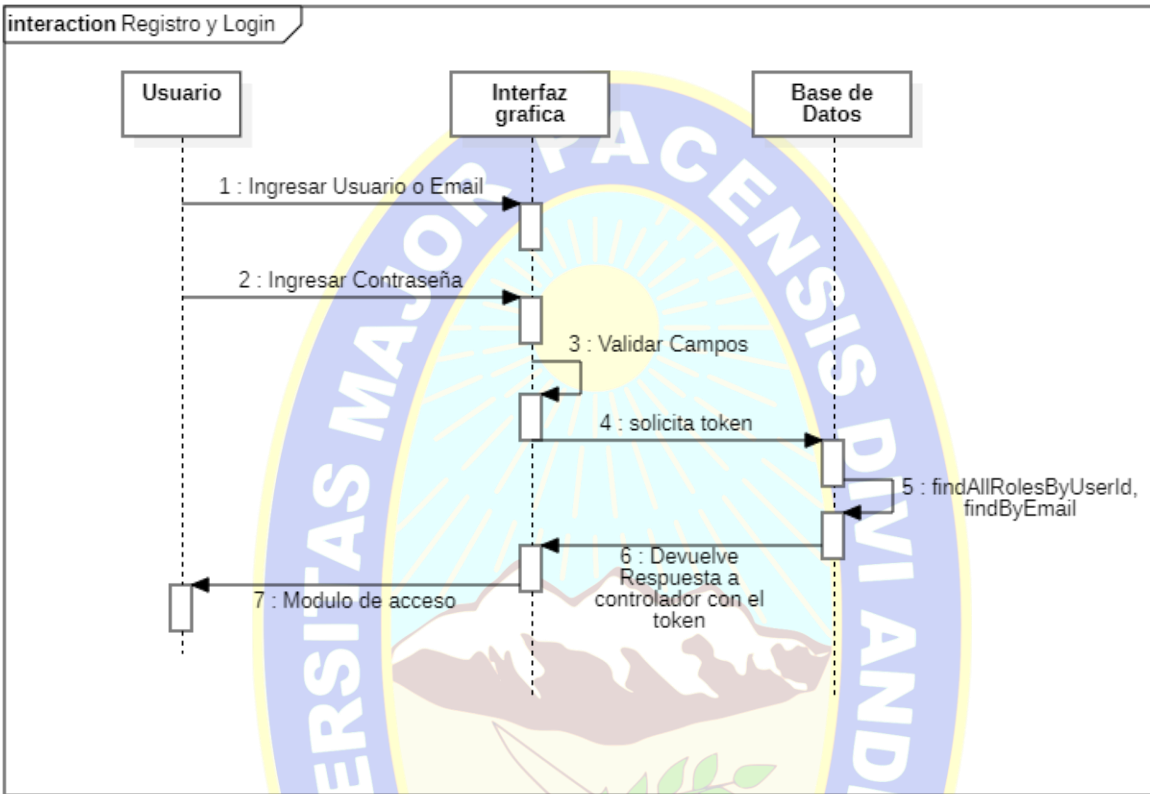
Figura 9 - Diagrama de estados, registro y login de usuario



3.5.2.3. DIAGRAMA DE SECUENCIA

Se diseñó el diagrama de secuencia para los requisitos de login de usuario, registro de usuarios para los mismos con asignación de roles automáticos, como se visualiza en la Figura nro. 12.

Figura 10 - Diagrama de secuencia, registro y login



Como se observa en la Figura nro. 12 la validación de los datos dentro el servidor se lo realiza dentro la entidad de Usuarios que se muestra en la Figura nro. 16 más adelante.

3.5.3. TERCER SPRINT: MÓDULO DE ADMINISTRACIÓN DE CUENTAS DE USUARIO

El módulo de administración de cuentas de usuario tiene como objetivo todas altas bajas y edición de roles para los usuarios que ingresen al sistema, en la Tabla nro. 8

Tabla 8-Planificación de administración de cuentas de usuario

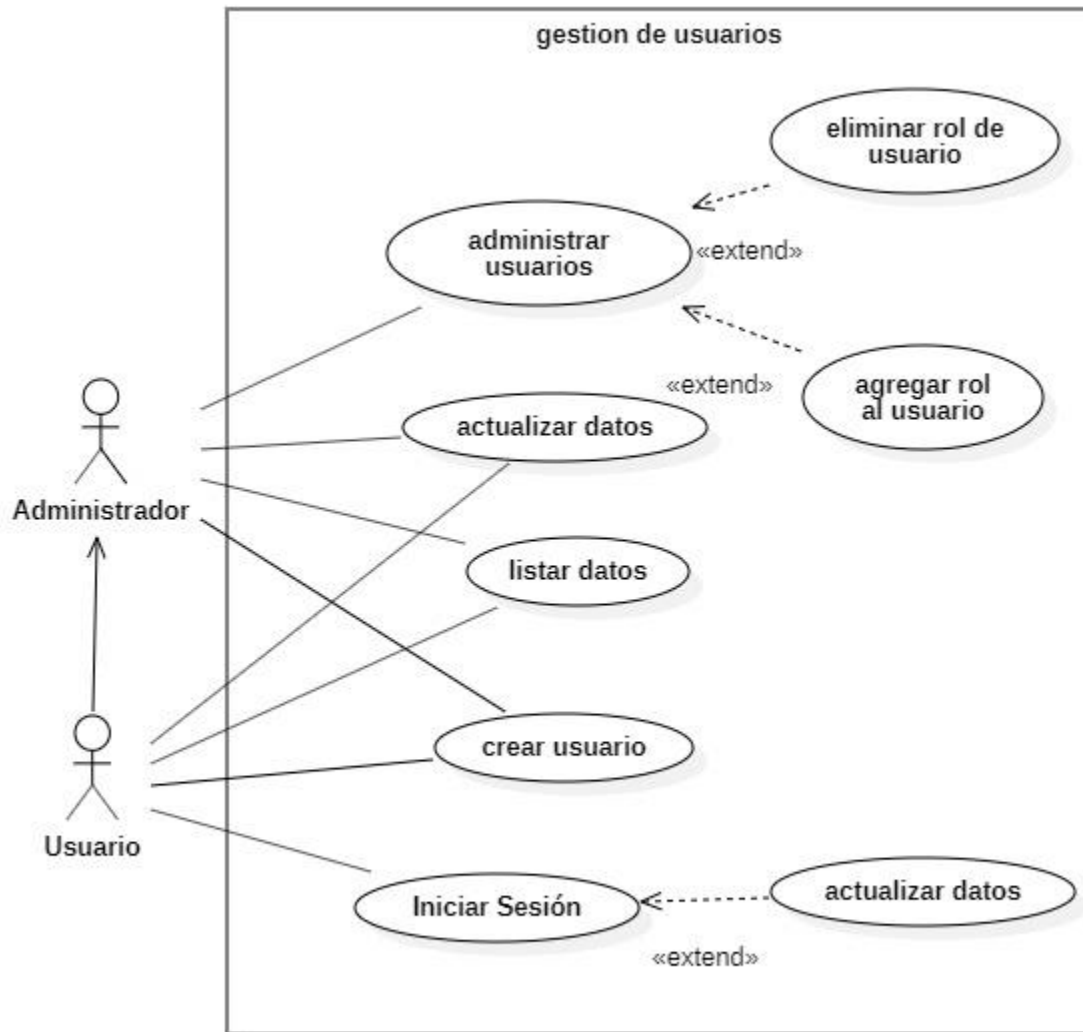
Sprint N°1. Módulo de administración de cuentas de usuario		Inicio	Fin	Duración
		01/01/2020	01/02/2020	30 días
Nro.	Tarea	Duración Tarea		Estado

1	Interfaz de administración para información de usuario	96 horas	Completado
2	Api para obtención de roles	96 horas	Completado
3	Autenticación de api con JWT para obtención de roles	120 horas	Completado
4	Validar registro de usuario lado servidor	168 horas	Completado
5	CRUD usuarios	120 horas	Completado
6	Control de api para alteraciones de token JWT	96 horas	Completado
7	Prueba de servicio de administración de usuarios	24 horas	Completado

3.5.3.1. DIAGRAMA DE CASOS DE USO

El usuario-cliente podrá generar una cuenta propia, además de poder editar información, ver la información propia que se ingresó al crear la cuenta del usuario, el administrador también tendrá la opción de editar a gusto y además eliminar usuarios el cual el usuario con un rol normal no podrá realizar, para mayor comprensión se tienen los casos de uso en la Figura nro. 13.

Figura 11 - Casos de uso, gestión de usuarios



3.5.3.2. DIAGRAMA DE ESTADOS

Se realizó el diagrama de estados para los requisitos: validar registro por lado del cliente, altas y bajas por parte del administrador de usuarios, lo cual se visualiza en la Figura nro. 14

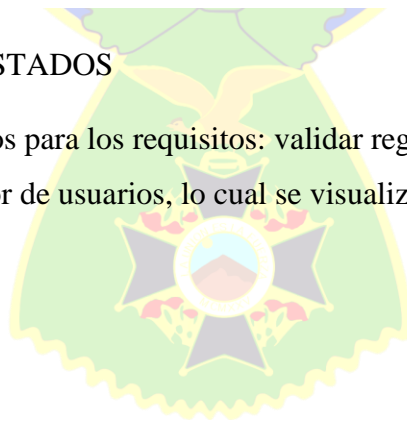
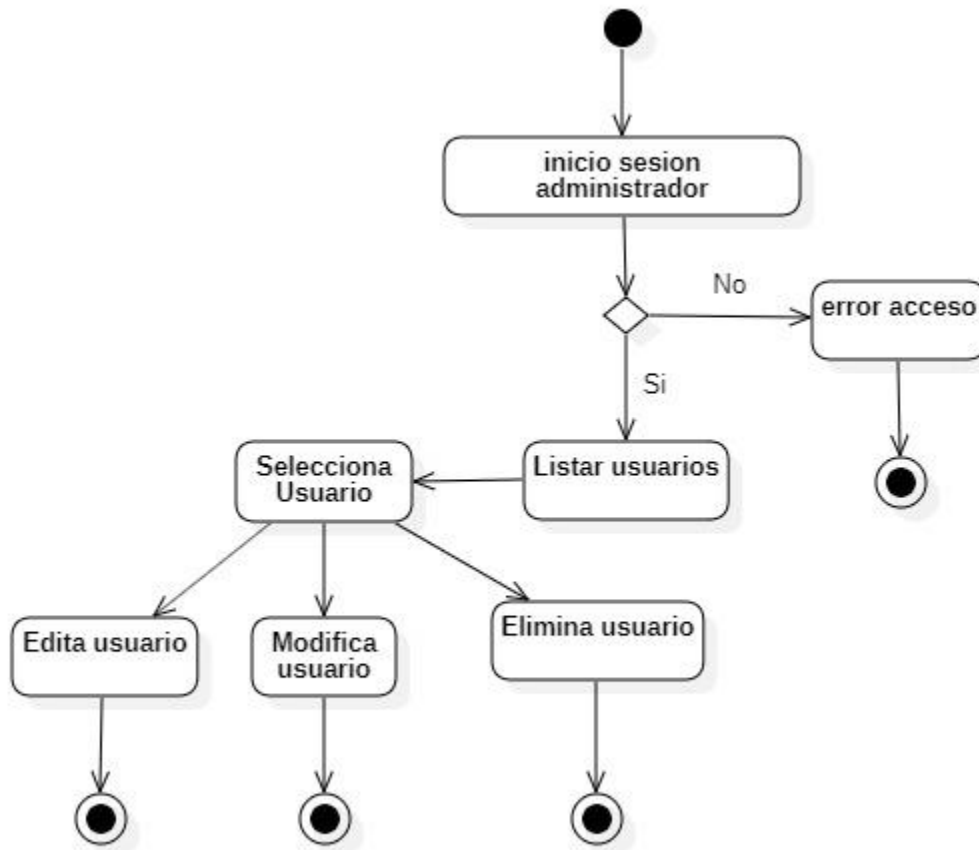


Figura 12 - Diagrama de estados, administración de usuarios



3.5.3.3. DIAGRAMA DE SECUENCIA

Se diseñó el diagrama de secuencia para la administración de usuarios para el lado del administrador, como se visualiza en la Figura nro. 15.

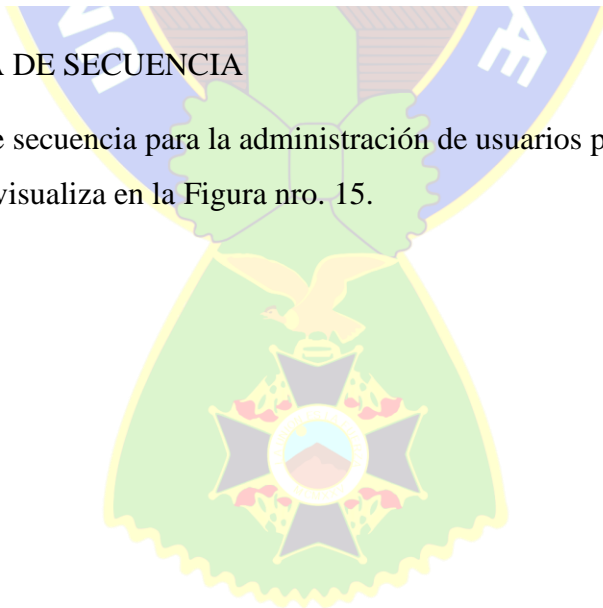
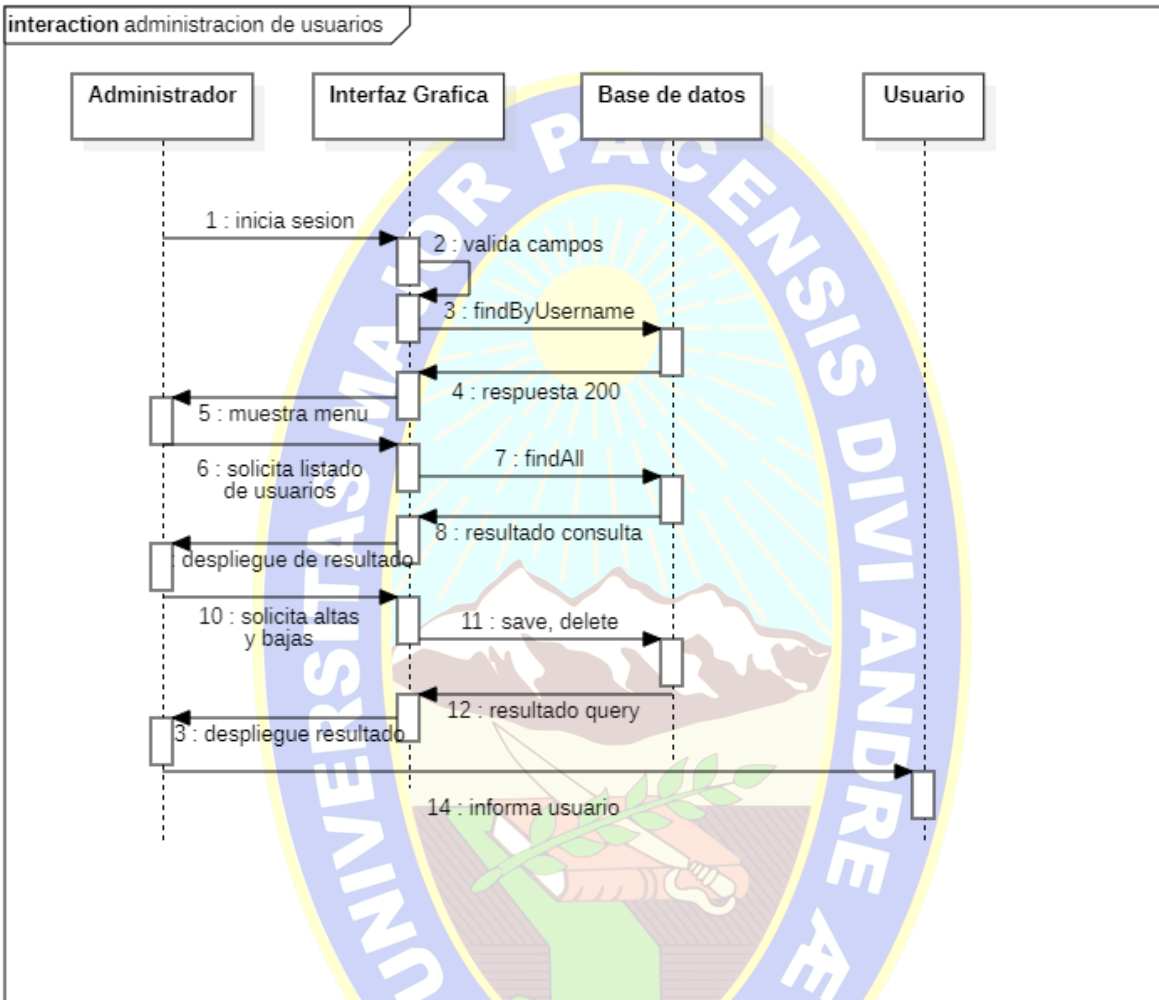


Figura 13 - Diagrama de secuencia, registro y login



3.5.3.4. DIAGRAMA DE CLASES

Se realizo el diseño del diagrama de clases para el módulo de “administración de cuentas de usuario” el cual se muestra como en la Figura nro. 16 existen diferentes métodos para la entidad “usuario” el cual se explica a continuación:

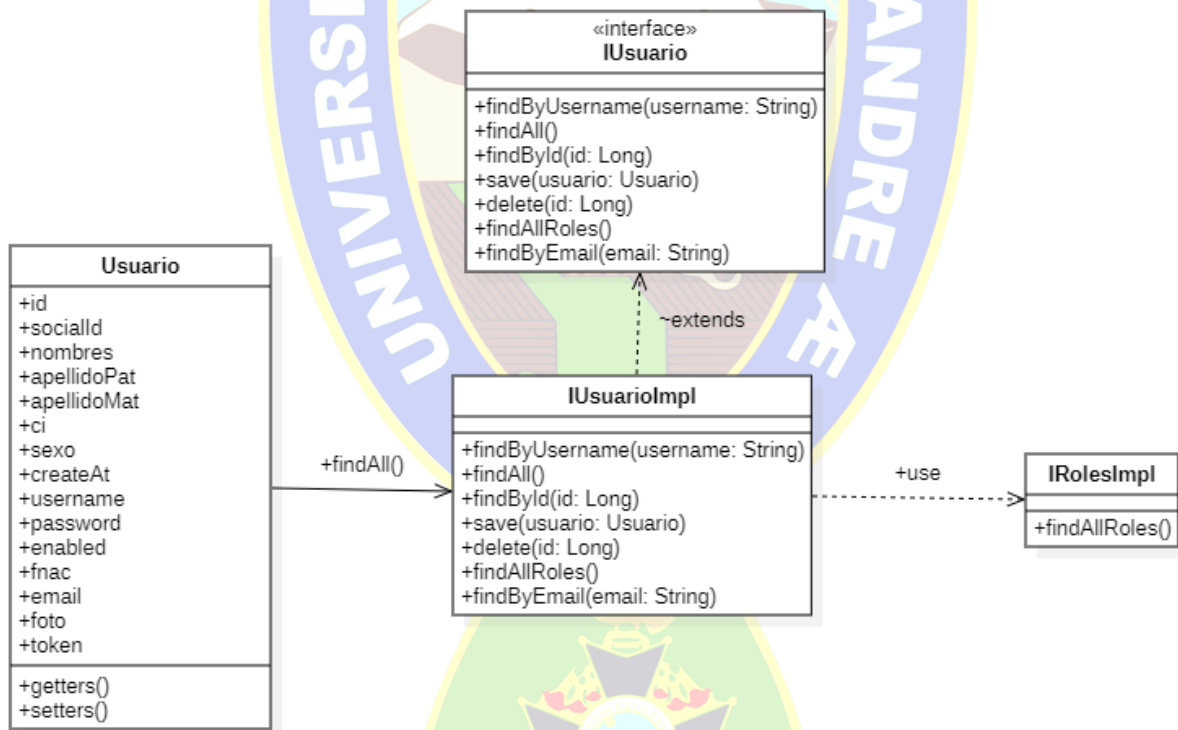
- findByUsername(username: String). - retorna un usuario buscado por el identificador de username
- findAll(). - retorna una lista de todos los usuarios
- findById(id: Long) .- retorna un usuario filtrado por id

- save(usuario: Usuario) .- guarda un usuario en la base de datos retornando estado 201 (CREATED) cuando se realiza la acción satisfactoriamente
- delete(id: Long) .- elimina un usuario con el parámetro recibido id retornando estado 200 (OK) cuando se elimina correctamente
- findAllRoles() .- retorna todos los roles existentes por usuario
- findByEmail() .- retorna un usuario filtrándolo por email (único)

Se tiene también una entidad simple con los siguientes métodos:

- listarRoles() .- retorna todos los roles existentes
- save(rol: Rol) .- guarda un usuario en la base de datos retornando estado 201 (CREATED) cuando se realiza la acción satisfactoriamente

Figura 14 - Diagrama de clases, administración de cuentas de usuario

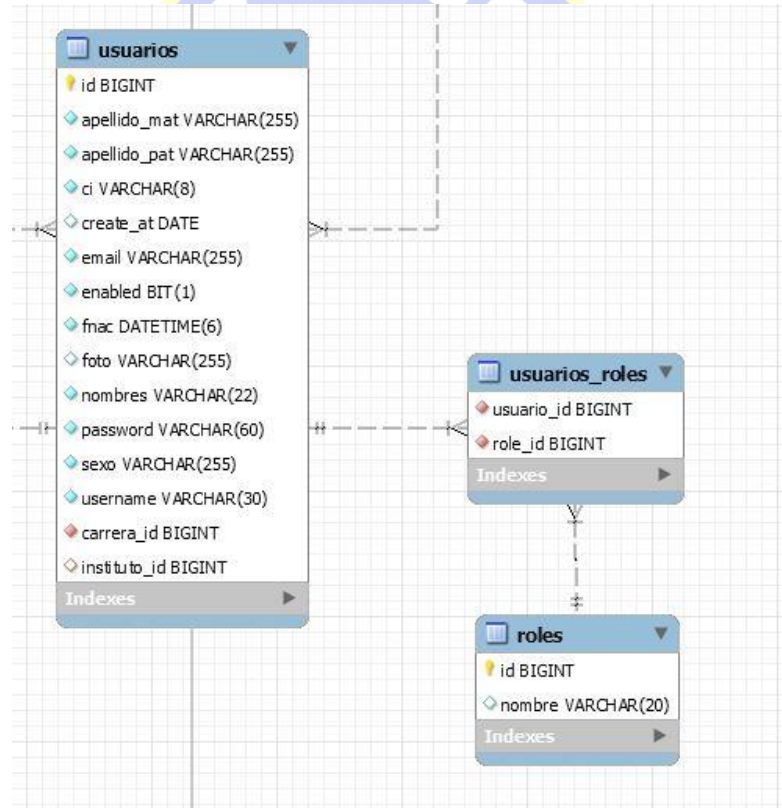


3.5.3.5. DIAGRAMA RELACIONAL

El siguiente diagrama muestra la tabla de usuarios relacionada con roles con una relación de N:M, un usuario puede tener varios roles (como ser ADMIN y USER) como también un rol puede ser usado en varios usuarios como se aprecia en la Figura nro. 17.

Los atributos del usuario son los más amplios de toda la base de datos y más completa, considerando que se tomara en cuenta para las estadísticas de la plataforma si esta fuera requerida.

Figura 15 - Diagrama relacional, módulo de administración de cuentas de usuario

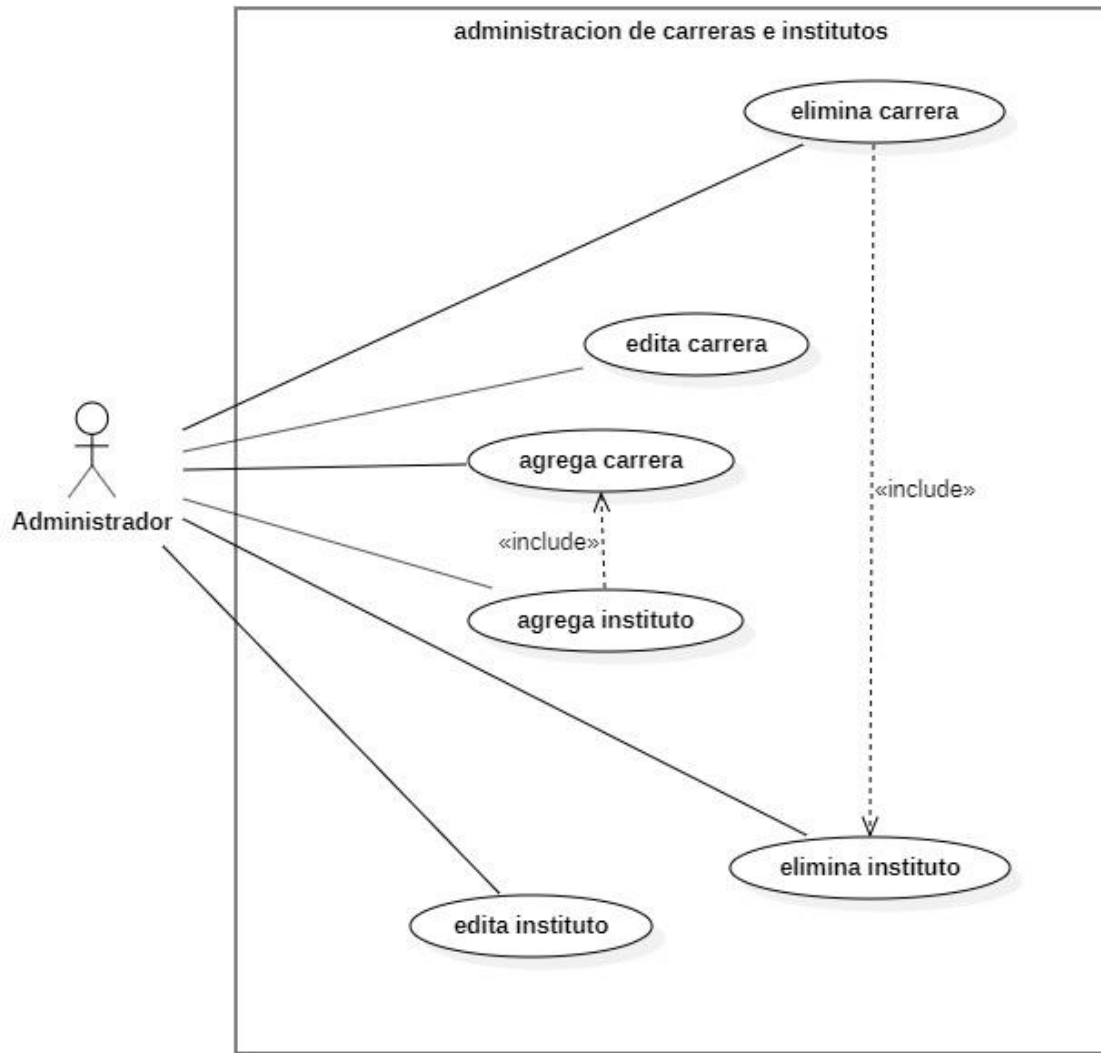


3.5.4. CUARTO SPRINT: MÓDULO DE ADMINISTRACIÓN DE CARRERAS E INSTITUTOS

3.5.4.1. DIAGRAMA DE CASOS DE USO

Un usuario sin permisos de administrador no podrá ser capaz de administrar ninguna de las partes que pasan a segundo plano, en este caso serían las carreras e institutos que existen en nuestra base de datos y que un usuario debe ingresar al registrarse y no hacerse anónimo. Por otra parte, el administrador tiene acceso a toda la parte de administración de las carreras e institutos como se observa en la Figura nro. 18 a continuación.

Figura 16 - Diagrama de casos de uso, administración de carreras e institutos



3.5.4.2. DIAGRAMA DE ESTADOS

Se realizó el diagrama de estados para los requisitos: eliminación de carrera, eliminación de institutos lo cual muestra la lógica que ocurre por debajo cuando se elimina una carrera junto con todos sus institutos como se muestra en la Figura nro. 19 a continuación.

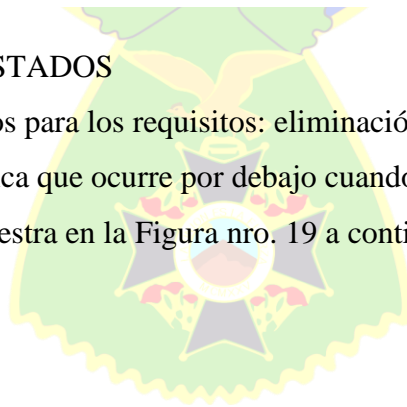
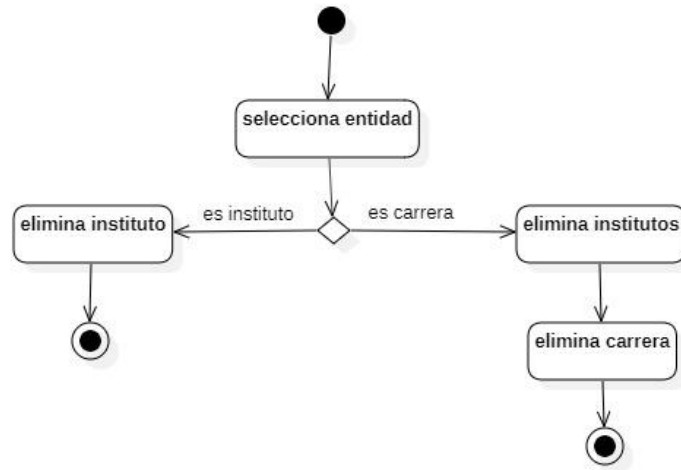


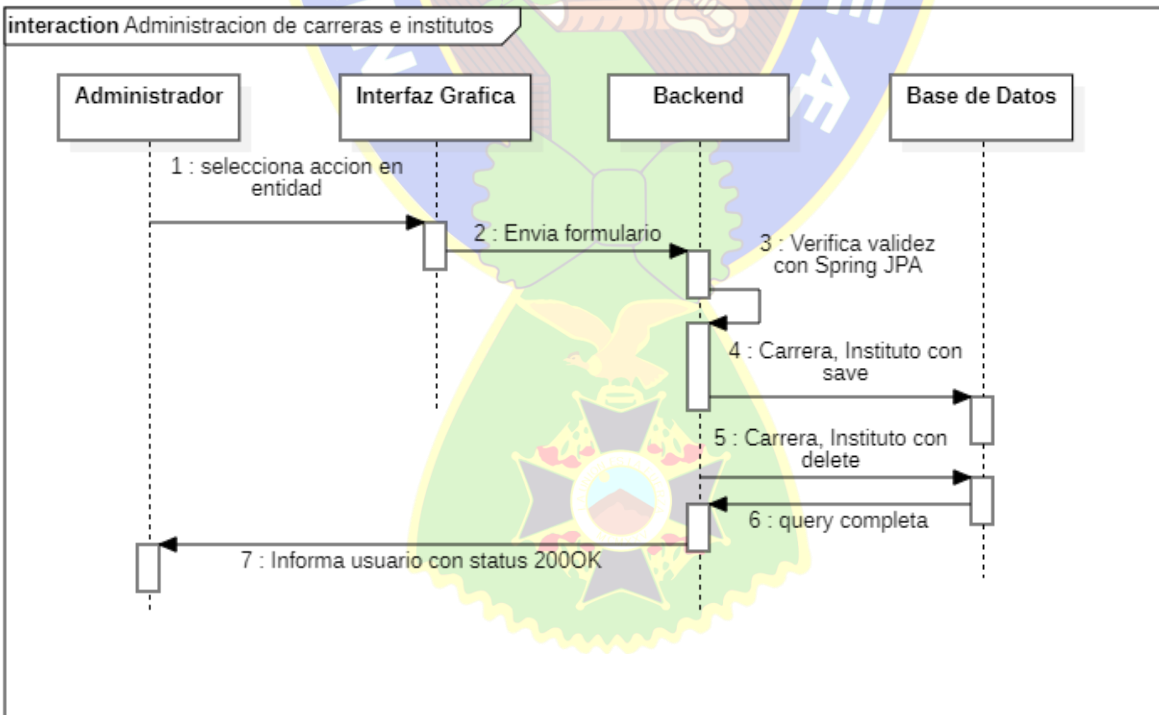
Figura 17 - Diagrama de estados, eliminación de instituto



3.5.4.3. DIAGRAMA DE SECUENCIA

Se diseñó el diagrama de secuencia para la administración de carreras e institutos, haciendo la validez con la herramienta de Spring por parte del back-end como se visualiza en la Figura nro. 20 a continuación.

Figura 18 - Diagrama de secuencia, administración de carreras e institutos



3.5.4.4. DIAGRAMA DE CLASES

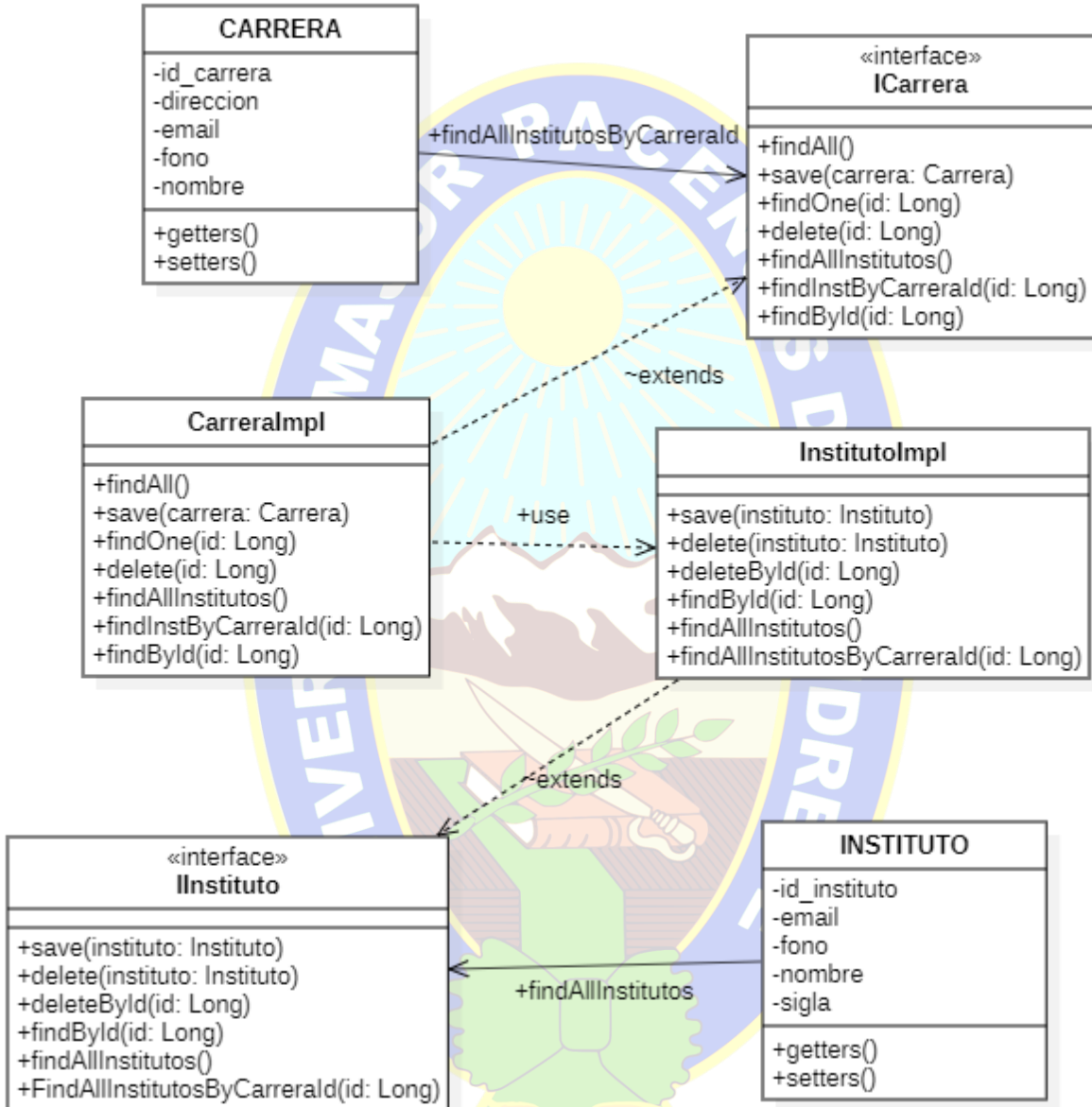
Se realizó el diseño del diagrama de clases para el módulo de “administración de carreras e institutos” el cual se muestra como en la Figura nro. 21 existen diferentes métodos para la entidad “carrera” el cual se explica a continuación:

- findAllCarreras() .- retorna un listado de todas las carreras disponibles
- save(carrera: Carrera) .- guarda una carrera en la base de datos retornando estado 201 (CREATED) cuando se realiza la acción satisfactoriamente
- delete(id: Long) .- elimina una carrera con el parámetro recibido id retornando estado 200 (OK) cuando se elimina correctamente

Para la entidad de “instituto” se tienen métodos similares a carrera

- findAllInstutos() .- retorna un listado de todos los institutos disponibles
- save(instituto: Instituto) .- guarda un instituto en la base de datos retornando estado 201 (CREATED) cuando se realiza la acción satisfactoriamente
- delete(id: Long) .- elimina un instituto con el parámetro recibido id retornando estado 200 (OK) cuando se elimina correctamente

Figura 19 - Diagrama de clases, administración de carreras e institutos

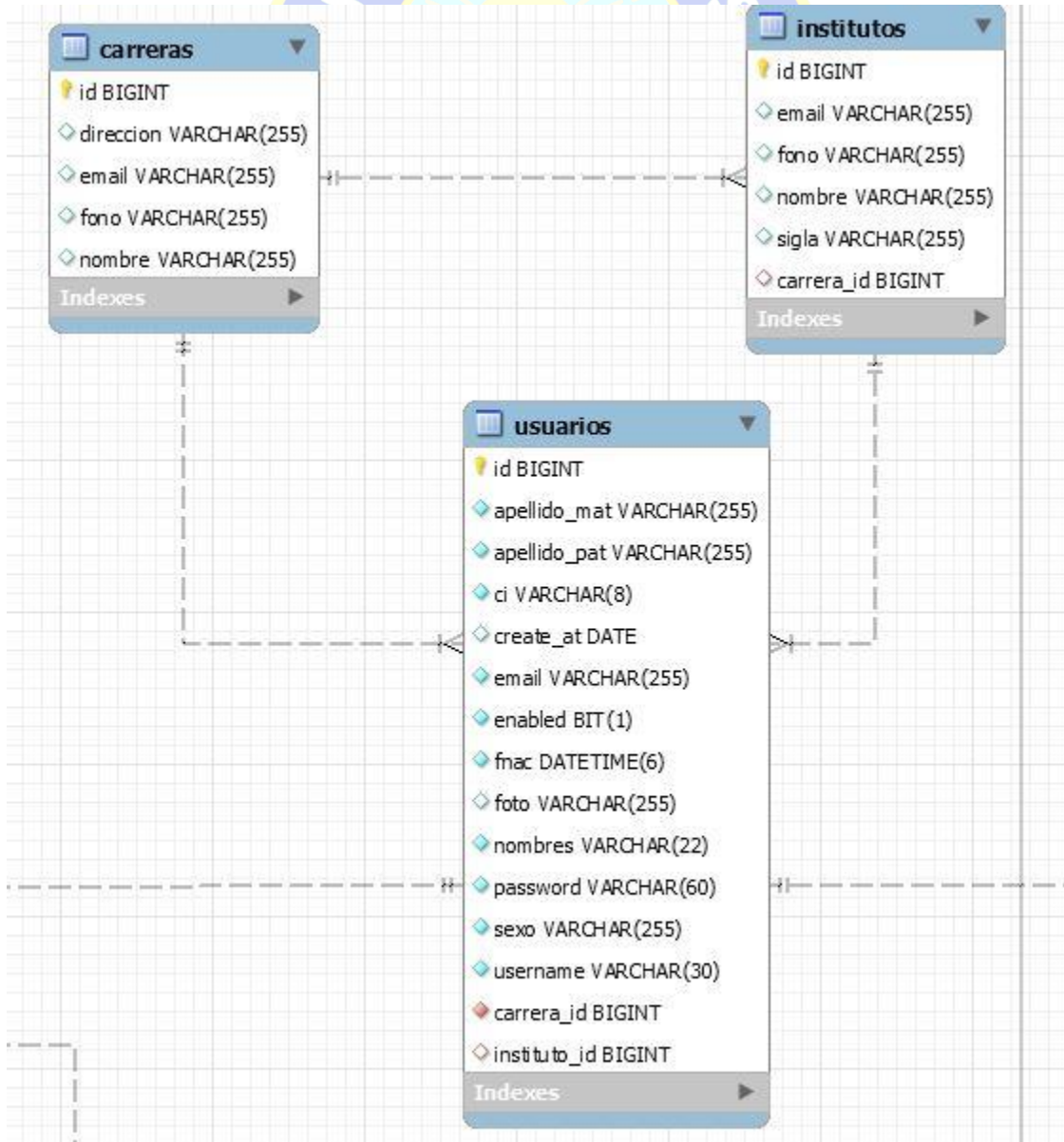


3.5.4.5. DIAGRAMA RELACIONAL

Una limitación del sistema es la parte de las carreras e institutos que dependerán de la administración del encargado administrador para poder agregar información requerida, por defecto estas vendrán con las carreras actuales en el año 2020.

Suponiendo algún cambio necesario a futuro se proporcionó una interfaz donde estos podrán ser editables para el administrador, agregando ya sea nuevos institutos y/o carreras. Para este caso un usuario solo podrá pertenecer a una carrera y en caso que fuera necesario podrá registrarse con un instituto, apoyándose en la Figura nro. 22 se puede verificar que un usuario pertenecerá a una carrera y también puede pertenecer a un instituto (no obligatoriamente).

Figura 20 – Diagrama relacional, administración de carreras e institutos

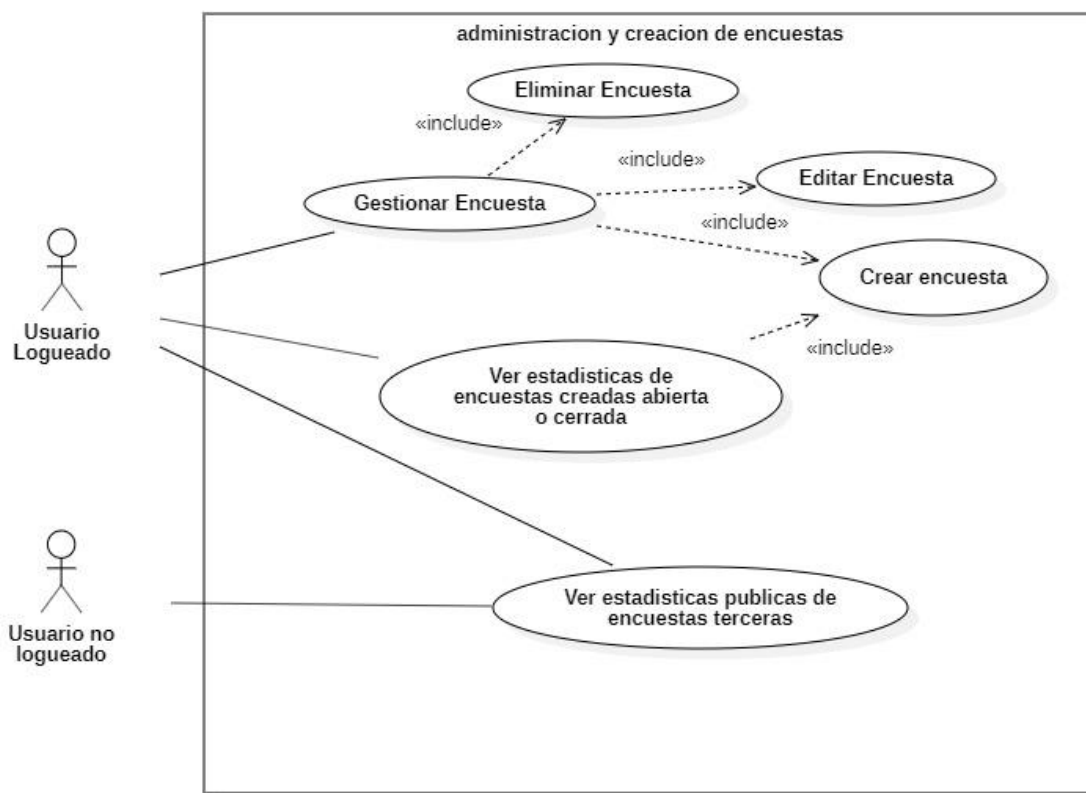


3.5.5. QUINTO SPRINT: MÓDULO DE ADMINISTRACIÓN Y CREACIÓN DE ENCUESTAS

3.5.5.1. DIAGRAMA DE CASOS DE USO

Un usuario podrá administrar de manera directa encuestas creadas por su misma persona, tanto como crearla y editarla antes de la fecha límite de publicación de la encuesta, por otra parte un usuario que no se encuentre *logueado* en el sistema solo podrá ver estadísticas públicas que se encuentren en la plataforma y no crearlas. En la Figura nro. 23 a continuación se puede observar con mayor facilidad y de manera reducida las limitaciones que posee un usuario *logueado* y no *logueado* para la creación o la visibilidad de estas encuestas.

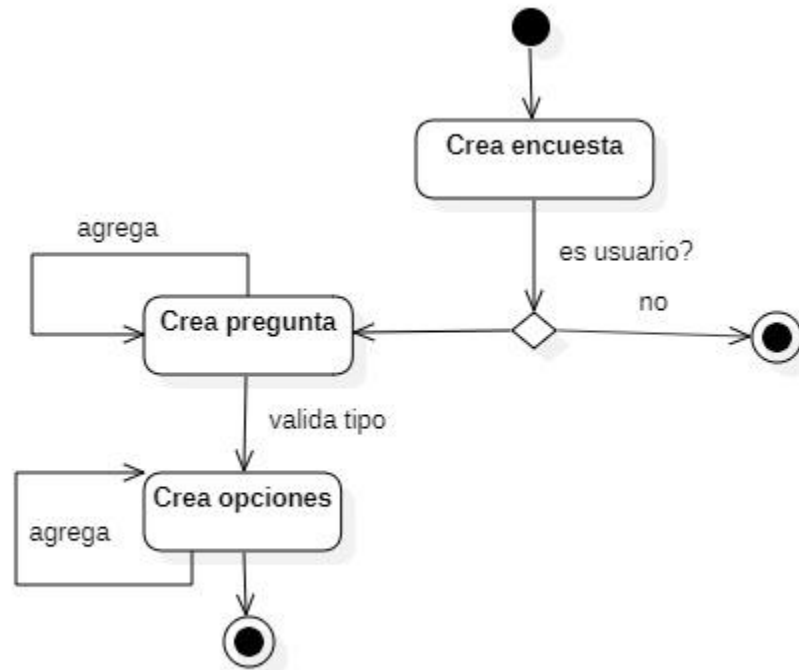
Figura 21 - Diagrama de casos de uso, administración y creación de encuestas



3.5.5.2. DIAGRAMA DE ESTADOS

Un usuario al tener la capacidad de generar una encuesta también tiene la posibilidad de crear varias preguntas por encuesta y a su vez crear opciones por cada pregunta como se muestra en la Figura nro. 24 a continuación:

Figura 22 - Diagrama de estados, administración y creación de encuestas



3.5.5.3. DIAGRAMA DE SECUENCIA

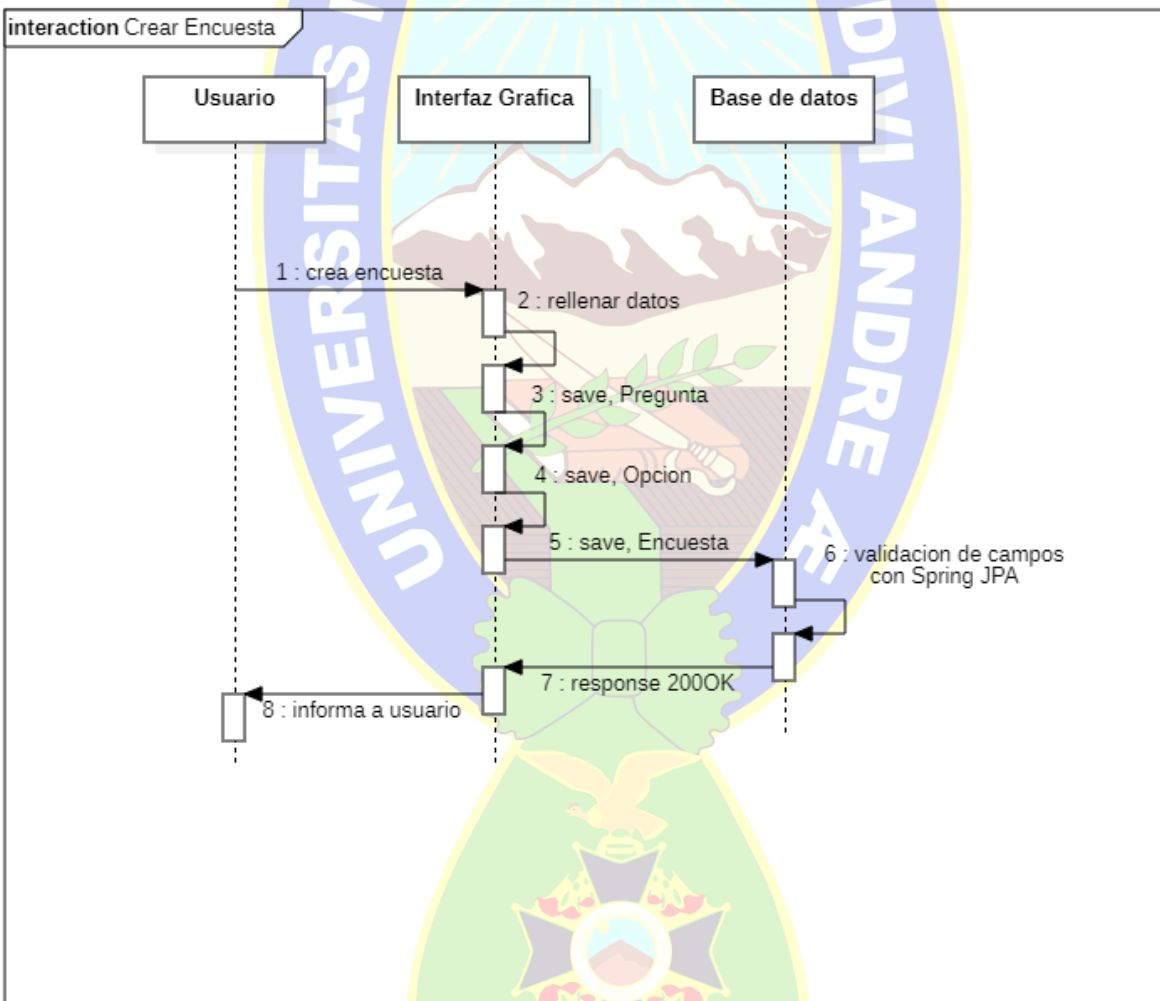
Una vez concluida la encuesta a enviar, la validación de los campos se los realiza por parte del Front End, para poder solventar algún tipo de valor no coincidente al que debería estar y que se termine enviando al Backend sin una validación previa por parte del Frontend responderá con un error del tipo HTTP o más conocido como “Códigos de estado de respuesta HTTP” como por ejemplo el código 401 *Unauthorized* que señala un api no permitido por la petición enviada.

Según Nasos Michas (Software Engineer y Scrum Master), es mucho mejor y prudente realizar la validación por parte del Backend, pero también es cierto que las validaciones por parte del Frontend terminan siendo útiles para el servidor en cuanto a las peticiones que terminan enviándose cuando existen múltiples usuarios intentando crear una encuesta o también

responderla, es por eso que la validación que se realiza en el FrontEnd está también pensada para no saturar los servidores del ITIC que es donde se hará *deploy*⁶ del sistema, debido a la famosa frase “nunca confíes en el usuario” porque se sabe cuándo se puede enviar información maliciosa al servidor (Michas, 2020).

Los pasos que seguirá el sistema tanto el Frontend como el Backend para la petición de creación de encuesta se detallan en la Figura nro. 25 a continuación.

Figura 23 - Diagrama de secuencia, administración y creación de encuestas.



⁶ Deploy: este término es muy conocido entre los desarrolladores web en resumen su función es: implantar, colocar en posición, habilitar para uso o, simplemente, publicar que en lo general suele ser el código desarrollado.

3.5.5.4. DIAGRAMA DE CLASES

Se realizó el diseño del diagrama de clases para el módulo de “administración y creación de encuestas” el cual se muestra como en la Figura nro. 26 existen diferentes métodos para la entidad “encuesta” el cual se explica a continuación:

- `findAllEncuestasByUsuarioId()` .- retorna todas las encuestas por usuario
- `findAllEncuestasByUsername(user: String)` .- retorna todas las encuestas con el parámetro `username` recibido
- `findAllPublic()` .- retorna todas las encuestas de tipo “publica”
- `save(encuesta: Encuesta)` .- guarda una encuesta en la base de datos retornando estado 201 (CREATED) cuando se realiza la acción satisfactoriamente
- `delete(id: Long)` .- elimina una encuesta con el parámetro recibido `id` retornando estado 200 (OK) cuando se elimina correctamente
- `findAll()` .- retorna todas las encuestas en un listado de encuestas
- `setPreguntas(preguntas: List<Pregunta>)` .- guarda las preguntas de la encuesta

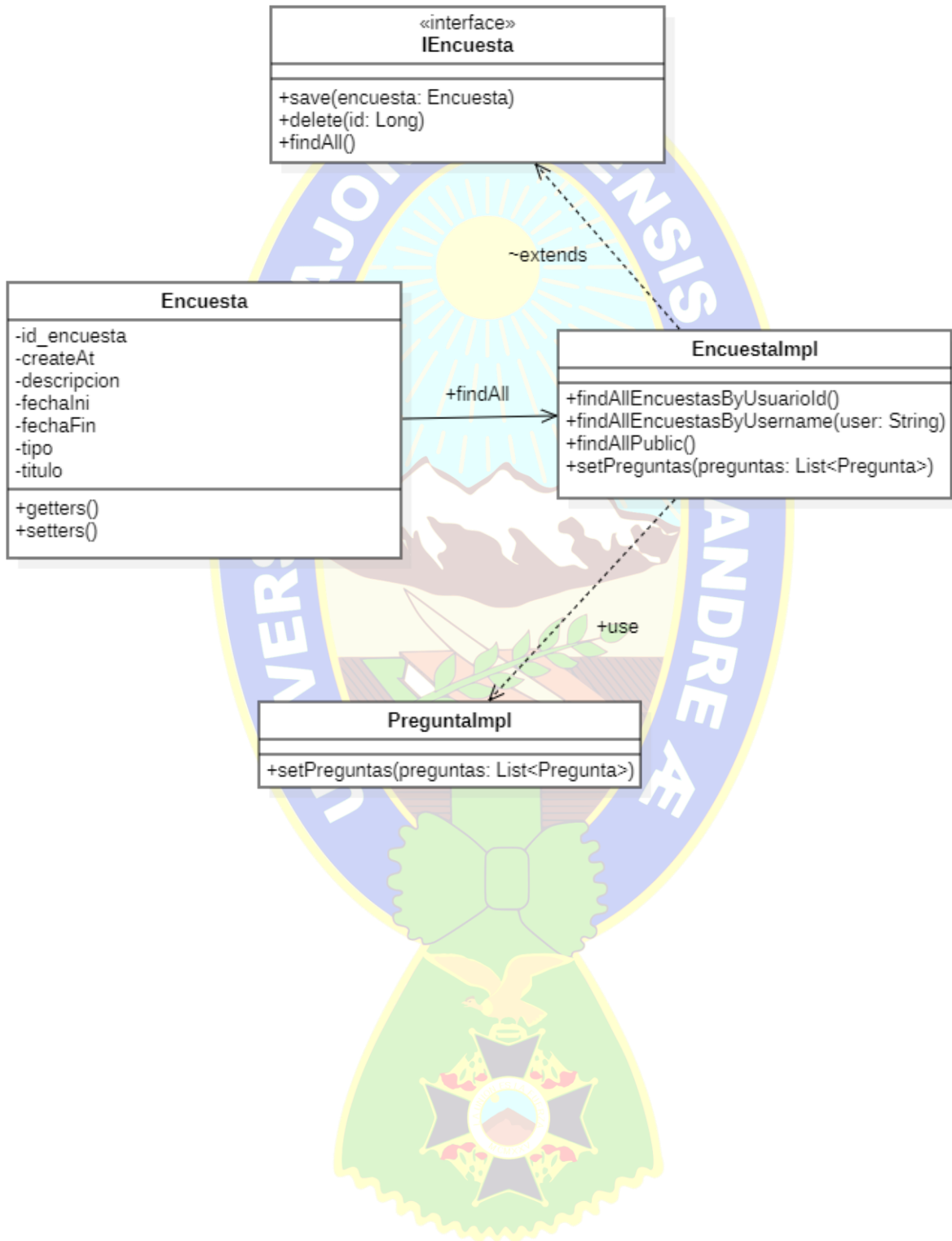
Para la entidad pregunta se tiene los siguientes atributos:

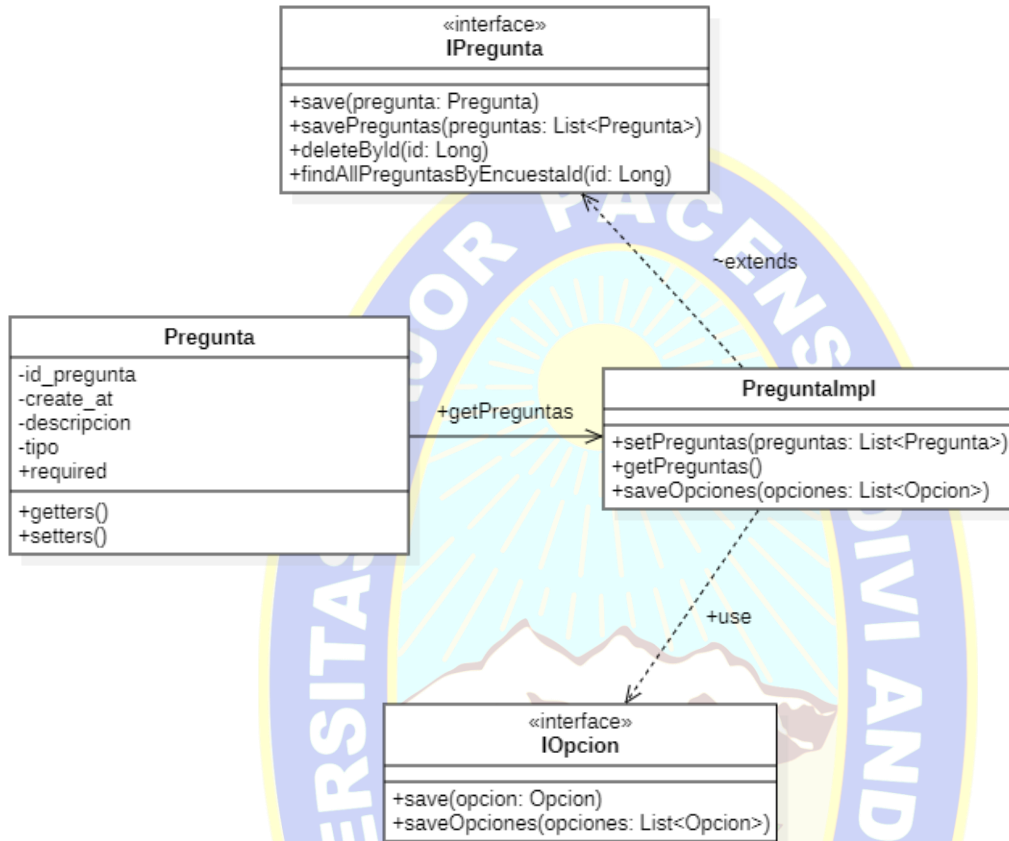
- `save(pregunta: Pregunta)` .- guarda una pregunta en la base de datos retornando estado 201 (CREATED) cuando se realiza la acción satisfactoriamente
- `delete(id: Long)` .- elimina una pregunta con el parámetro recibido `id` retornando estado 200 (OK) cuando se elimina correctamente
- `getPreguntas()` .- retorna una lista de todas las preguntas existentes
- `setOpciones(opciones: List<Opcion>)` .- guarda una lista de opciones dentro de la entidad pregunta

Para la entidad opción se tiene los siguientes atributos:

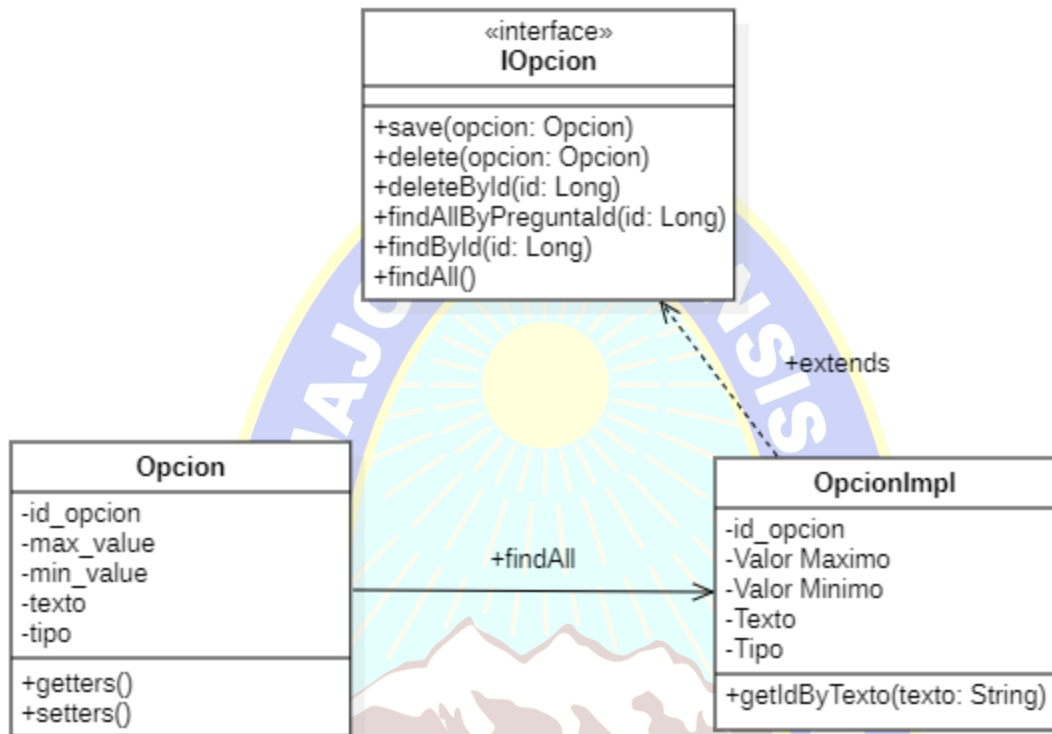
- `getIdByText(texto: String)` .- retorna el primer `id` que se obtiene por texto (útil para casos donde se requiera agregar más cantidad de respuestas con el mismo texto)
- `save(option: Option)` .- guarda una opción en la base de datos retornando estado 201 (CREATED) cuando se realiza la acción satisfactoriamente
- `delete(id: Long)` .- elimina una opción con el parámetro recibido `id` retornando estado 200 (OK) cuando se elimina correctamente

Figura 24 - Diagrama de clases, administración y creación de encuestas





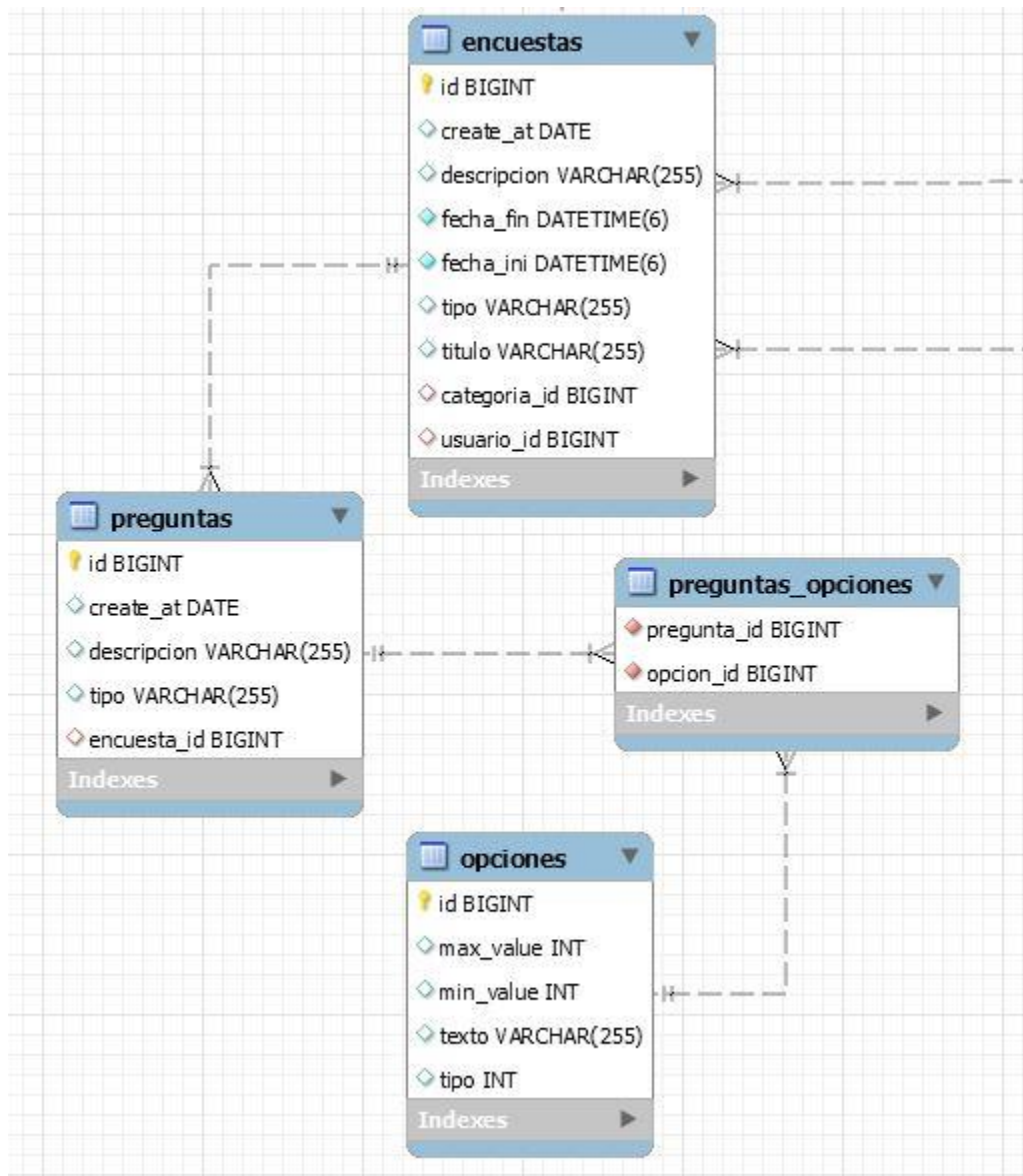
Por último, se tiene dentro la relación el diagrama de clases a continuación:



3.5.5.5. DIAGRAMA RELACIONAL

Para la estructuración de cada encuesta esta deberá contener una lista de preguntas única por cada encuesta y a su vez cada pregunta deberá tener varias opciones, véase Figura nro. 27.

Figura 25 - Diagrama relacional, administración y creación de encuestas



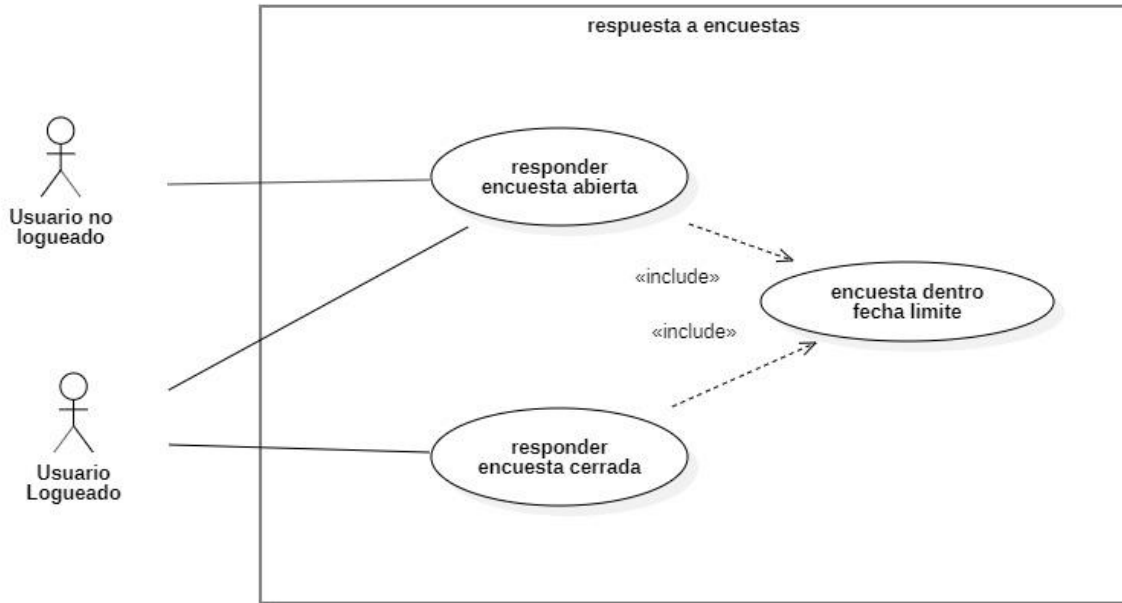
3.5.6. SEXTO SPRINT: MÓDULO DE RESPUESTA DE ENCUESTAS

3.5.6.1. DIAGRAMA DE CASOS DE USO

Todos los usuarios tienen la capacidad de responder preguntas, pero se restringe por el tipo de encuesta a la que se quiere responder; si la encuesta es de tipo “abierta” cualquier usuario puede responderla, ya sea un usuario con una cuenta existente y que se encuentre *logueado* en el sistema o también sin la necesidad de estar *logueado*, a diferencia de una encuesta que sea de tipo “cerrada” esta no podrá ser respondida o entregada a usuarios que no posean una cuenta, el

sistema los redireccionara a la pantalla de *login*, esto se debe a motivos de calidad de uso, para facilitar el entendimiento se tiene la Figura nro. 28 a continuación.

Figura 26 - Diagrama de casos de uso, respuesta a encuestas

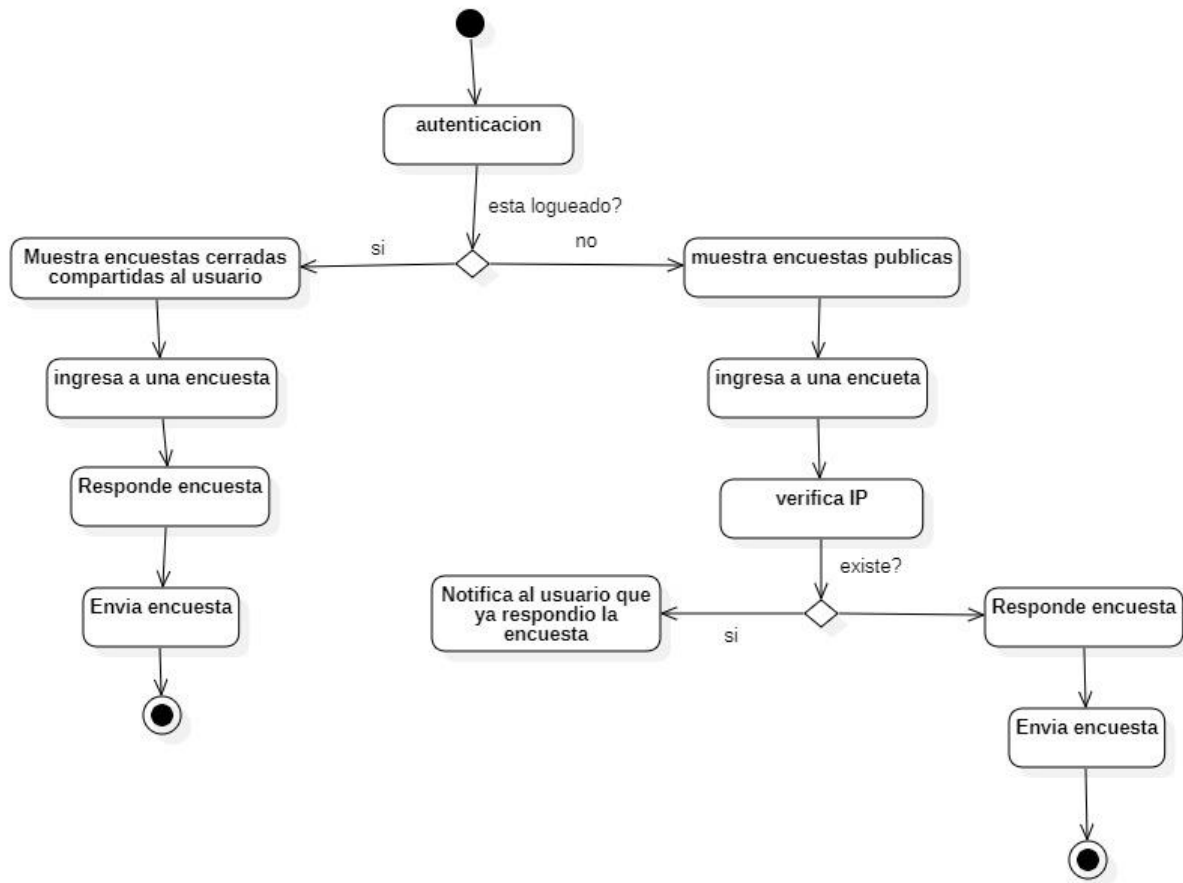


3.5.6.2. DIAGRAMA DE ESTADOS

Cualquier usuario sin tener una cuenta creada podrá responder encuestas que se encuentren en modalidad “abierto” y limitado con el IP del ordenador para que no pueda responder varias veces una encuesta, para el caso de que una encuesta se encuentre en la modalidad “cerrada” este solo podrá ser accesible mediante enlace y no necesariamente ingresado en el sistema previamente.

Existe una diferencia cuando un usuario ingreso previamente al sistema a uno que no lo realizo, para las respuestas de encuestas como se puede ver en la Figura nro. 29.

Figura 27 - Diagrama de estados, módulo de respuesta a encuestas



3.5.6.3. DIAGRAMA DE SECUENCIA

La forma lineal de crear una encuesta por parte de un usuario que se encuentra *logueado* y otro que no lo está se detalla a continuación en la Figura nro. 30.

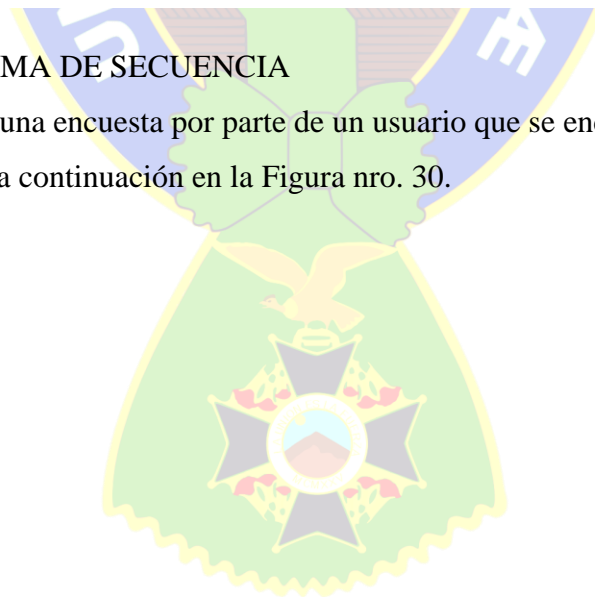
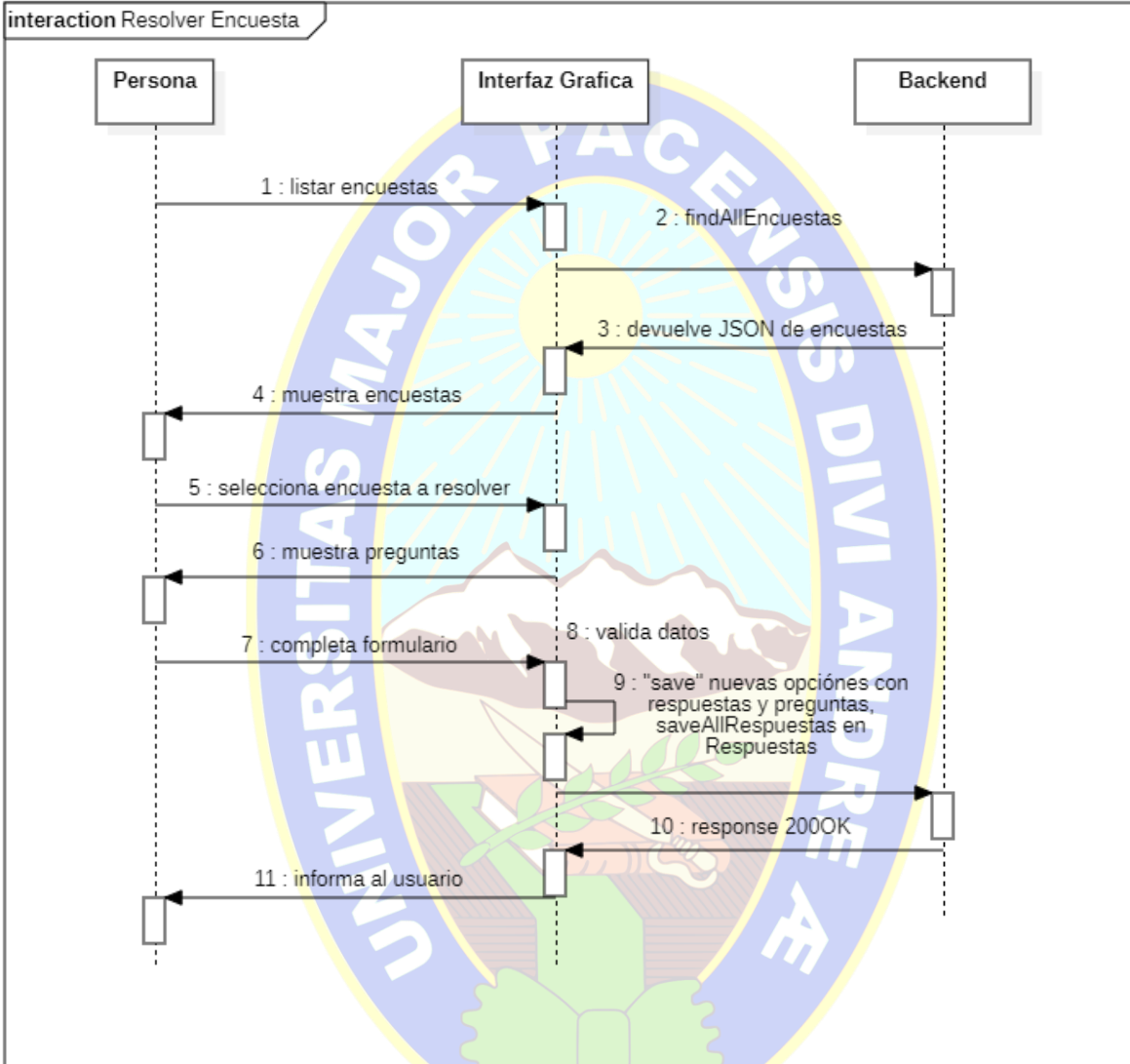


Figura 28 - Diagrama de secuencia, módulo de respuesta de encuestas



3.5.6.4. DIAGRAMA DE CLASES

Se hizo la planeación cautelosa de la clase de respuesta (véase Figura nro. 31), a causa de ser la más importante del proyecto donde se guardará información valiosa y llegaría a ser el núcleo del proyecto actual, para poder llegar a una optimización se optó por los siguientes atributos a continuación:

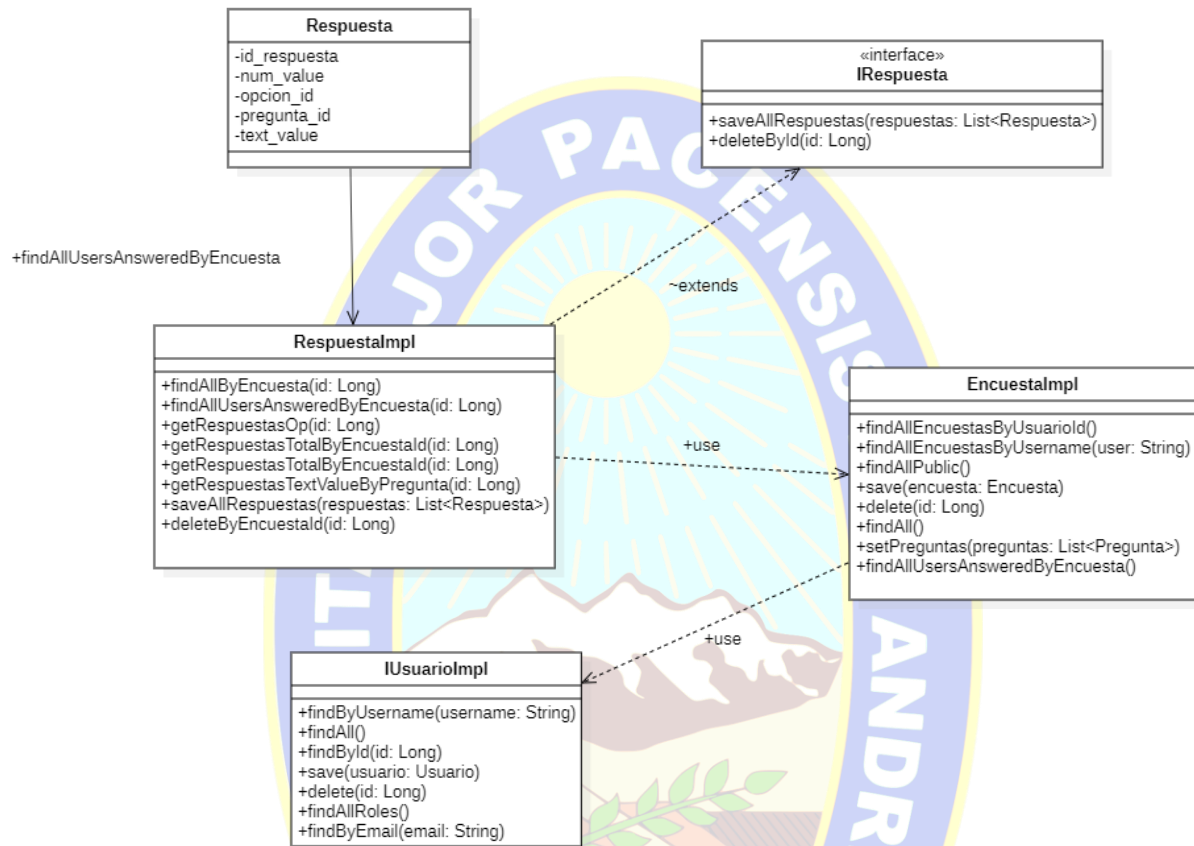
- id_respuesta: identificador de la clase
- num_value: en el caso de que el usuario responda a una pregunta que sea del tipo “escala lineal” este valor se almacenaría en esta casilla

- opción_id: si la pregunta fuera de tipo “opción múltiple” o de tipo “casillas de verificación” el valor que el usuario responde por cada pregunta se almacenara en este atributo.
- pregunta_id: lleva un identificador a la pregunta que pertenece.
- text_value: si la pregunta fuera de tipo “respuesta simple” o de tipo “párrafo”, la respuesta textual del usuario se almacenaría en este atributo

Los métodos siguientes estarían enlazados al DAO por parte del Backend

- findAllByEncuesta(id: Long) .- retorna una lista de todas las respuestas enviadas por el usuario filtrados por el parámetro recibido “id”
- findAllUsersAnsweredByEncuesta(id: Long) .- retorna el total de usuarios que participaron en la encuesta con el parámetro recibido considerado como identificador “id”
- getRespuestaOp(id: Long) .- retorna una lista de todas las respuestas de tipo “opción multiple” o casillas de verificacion.
- getRespuestasTotalByEncuestaId(id: Long) .- retorna el total de respuestas realizadas por cada encuesta.
- getRespuestasTextValueByPregunta(id: Long) .- retorna una lista de todas las respuestas de tipo “respuesta simple” o “párrafo”.

Figura 29 - Diagrama de clases, respuesta de encuestas



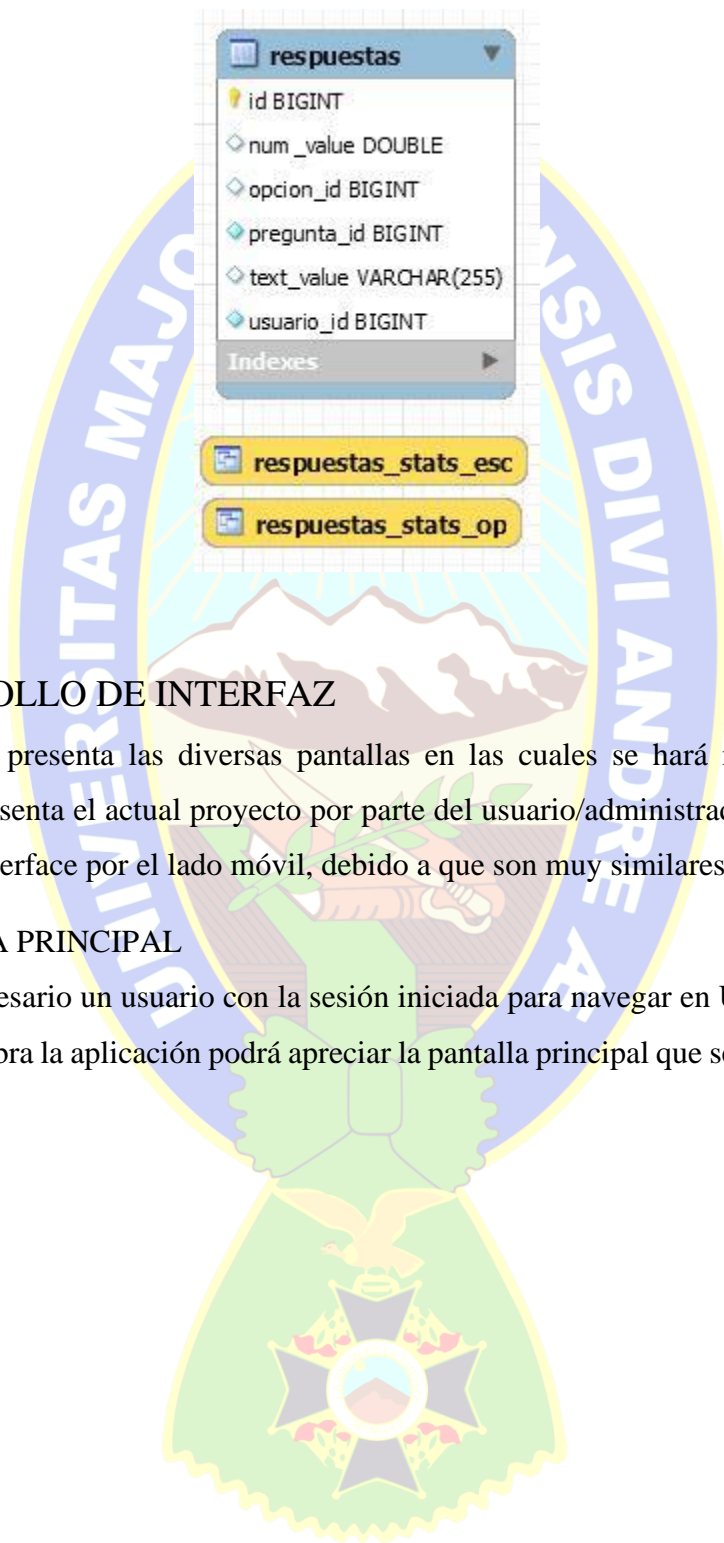
3.5.6.5. DIAGRAMA RELACIONAL

En el módulo de respuesta de encuestas se tomó muy en cuenta las relaciones que esta entidad podría tener, y se optó por la más óptima que es estar relacionado con el usuario teniendo así en cada respuesta el id del usuario

Se implemento también 2 vistas integradas en la base de datos para facilitar la obtención de información (véase Figura nro. 32):

- respuestas_stats_op. - cuenta la cantidad de respuestas por cada opción seleccionada por cada pregunta en los casos de “opción múltiple”, “casillas de verificación” y “párrafo”
- respuestas_stats_esc. - cuenta la cantidad de respuestas para el caso de “escala lineal”

Figura 30 - Diagrama relacional, respuestas



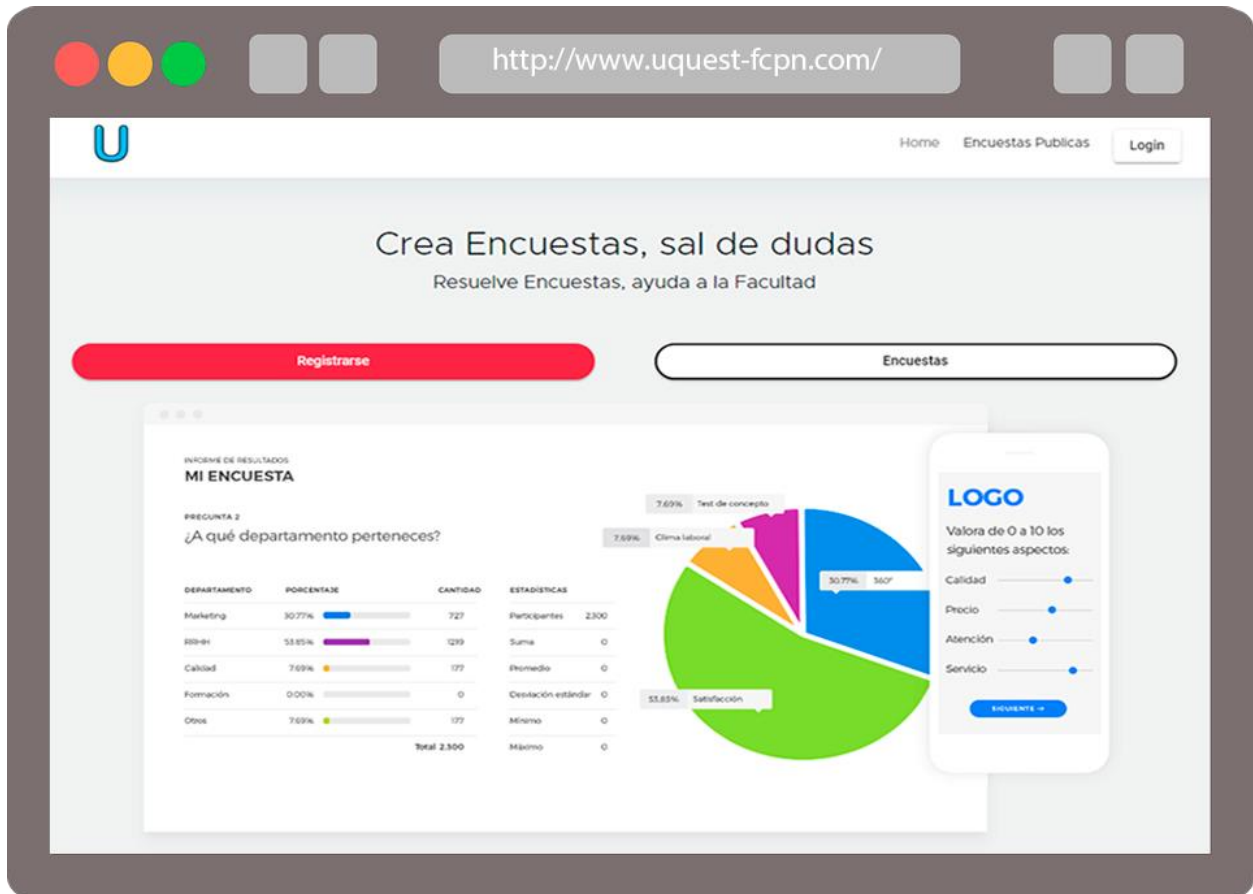
3.6. DESARROLLO DE INTERFAZ

En esta sección se presenta las diversas pantallas en las cuales se hará mención de las más importantes que presenta el actual proyecto por parte del usuario/administrador en un navegador, no se presentó la interface por el lado móvil, debido a que son muy similares en ambos casos.

3.6.1. PANTALLA PRINCIPAL

Dado que no es necesario un usuario con la sesión iniciada para navegar en UQUEST, la primera vez que el usuario abra la aplicación podrá apreciar la pantalla principal que se aprecia en la Figura nro. 33.

Figura 31 - UQUEST, Pantalla Principal



3.6.2. PÁGINA DE REGISTRO DE CUENTA

La ventana de registro de cuenta es un poco amplia con varios campos a llenar, pero necesaria para obtener resultados posteriores más precisos si es que se llegara a desarrollar a más profundidad, el ejemplo de los campos y la demostración de la pantalla se puede apreciar en la Figura nro. 34.

Figura 32 - UQUEST, registro usuario

UQUEST user registration page. The browser address bar shows `http://www.uquest-fcpn.com/`. The page has a header with a 'U' logo, 'Home', 'Encuestas Publicas', and 'Login' links. The main content is a registration form titled 'Registro' with fields for 'Nombres *', 'Apellido paterno *', 'Apellido materno *', 'CI *', 'Sexo *', 'Fecha de nacimiento *', 'Carrera *' (set to 'Informatica'), 'Instituto', 'Email *', and 'Username *' (set to 'fema'). A red error message 'Usuario ya existente' is shown below the username field. A 'Password *' field is also present with a red error message 'This field is required'. A green 'Finalizar' button is at the bottom.

3.6.3. PÁGINA DE INGRESO AL SISTEMA

El usuario una vez registrado previamente para ingresar al sistema este deberá ingresar a la página de “login” es decir ingresar al sistema mediante la siguiente sencilla interfaz de la Figura nro. 35:

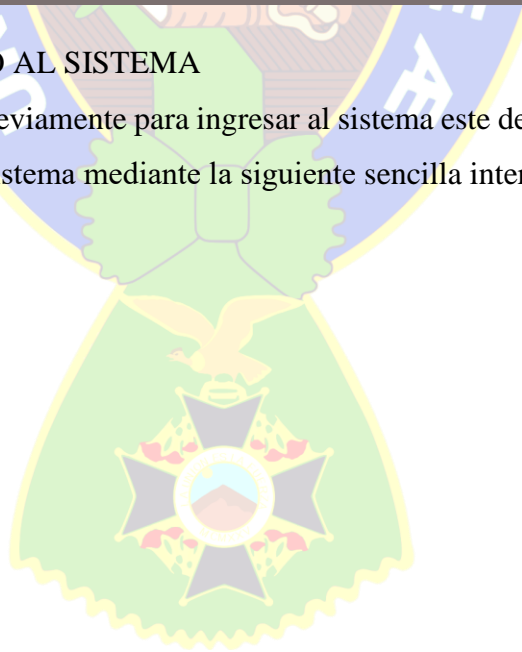
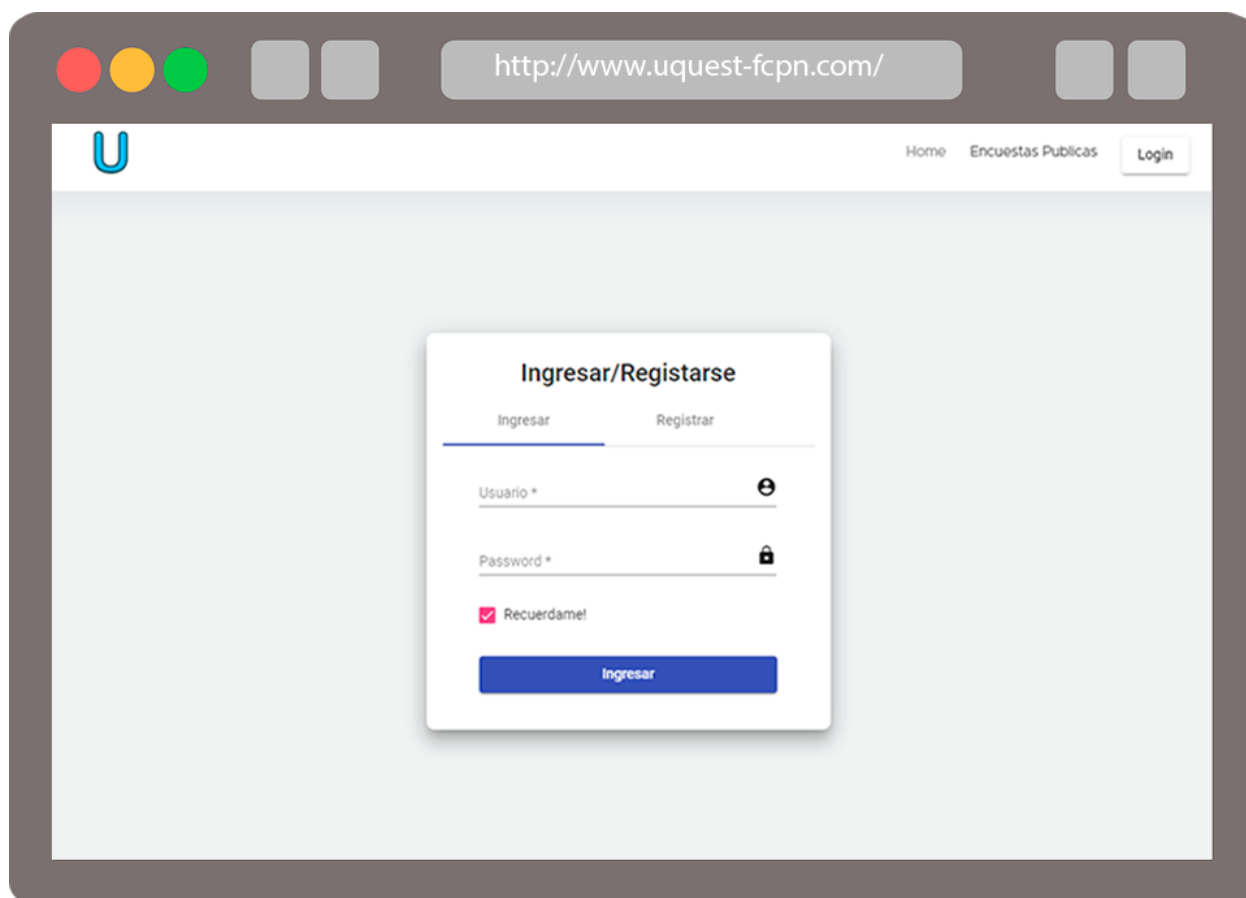


Figura 33 - UQUEST, ingreso al sistema



3.6.4. PERFIL DEL USUARIO

El usuario podrá ver los detalles de su perfil y si lo deseara también podrá editarlos con una opción en la misma ventana, las características del usuario podrá verlos a continuación en la figura nro. 36.

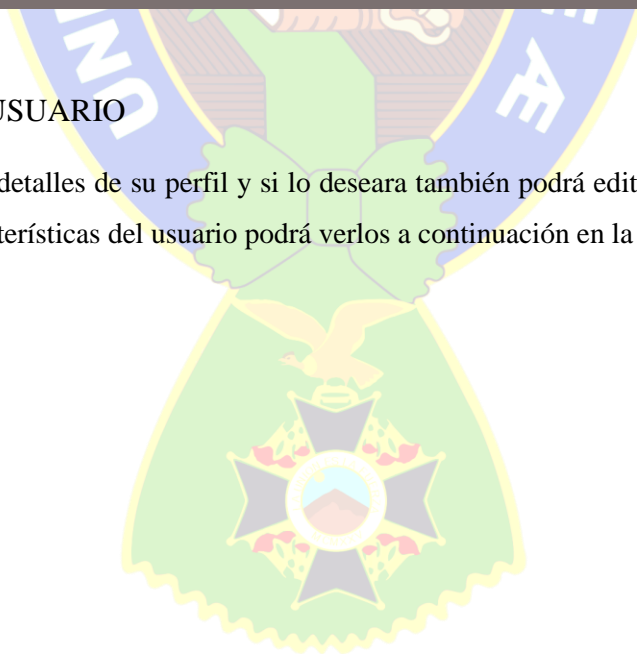
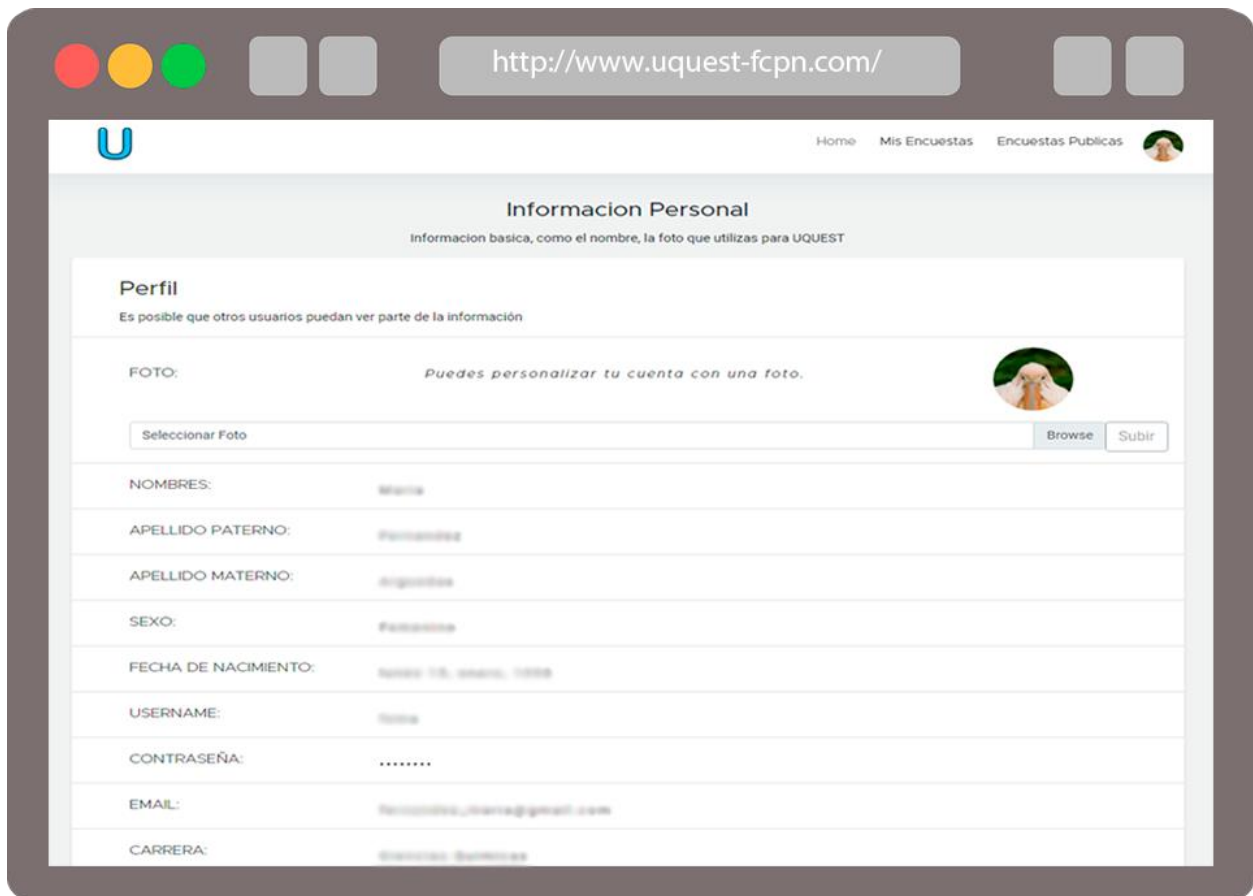


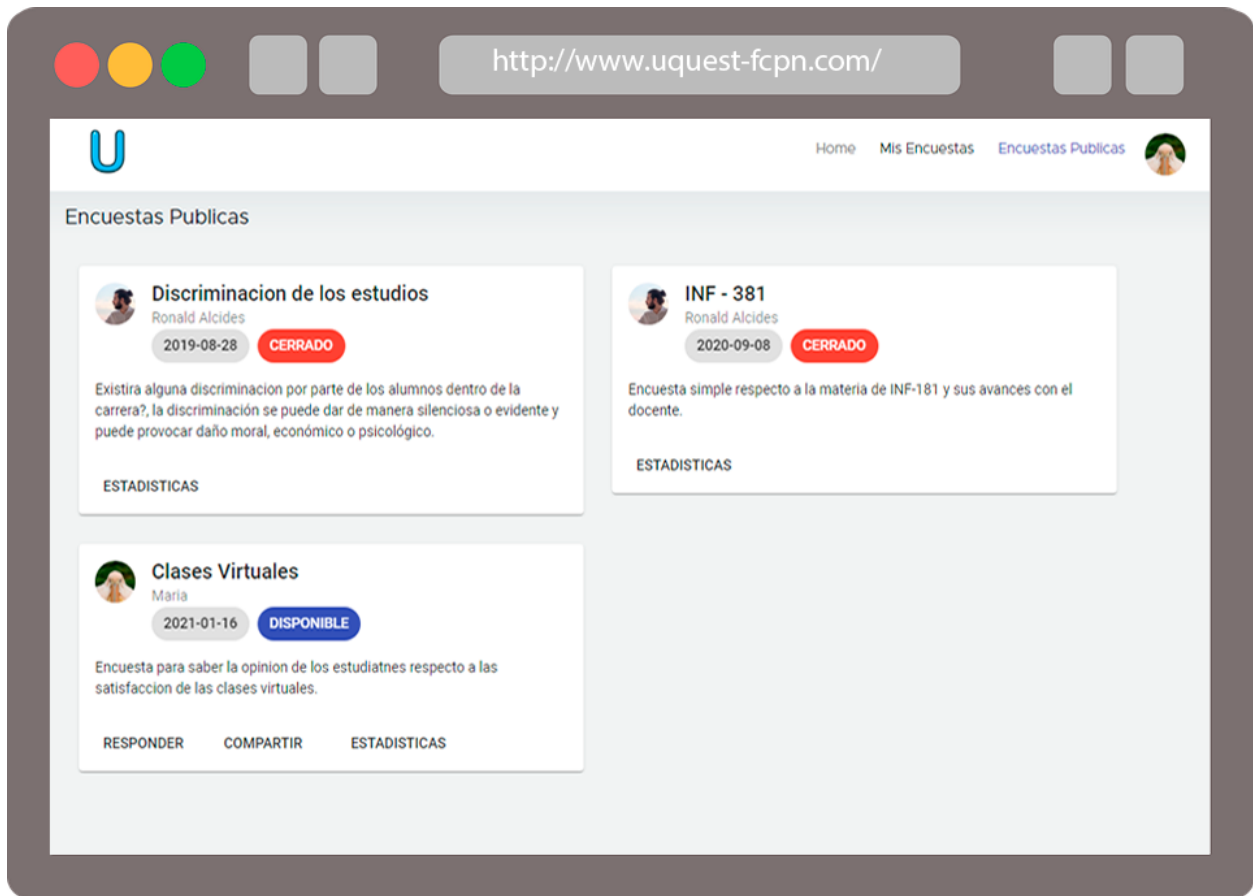
Figura 34 - UQUEST, perfil usuario



3.6.5. LISTADO DE ENCUESTAS PÚBLICAS

En este apartado el usuario, ya sea que haya ingresado al sistema o no, como lo indica el diagrama de casos de uso en la Figura nro. 37, para mayor comprensión se tiene la siguiente imagen (Figura nro. 36) que contiene casos de ejemplo con encuestas ficticias elaboradas por usuarios de prueba.

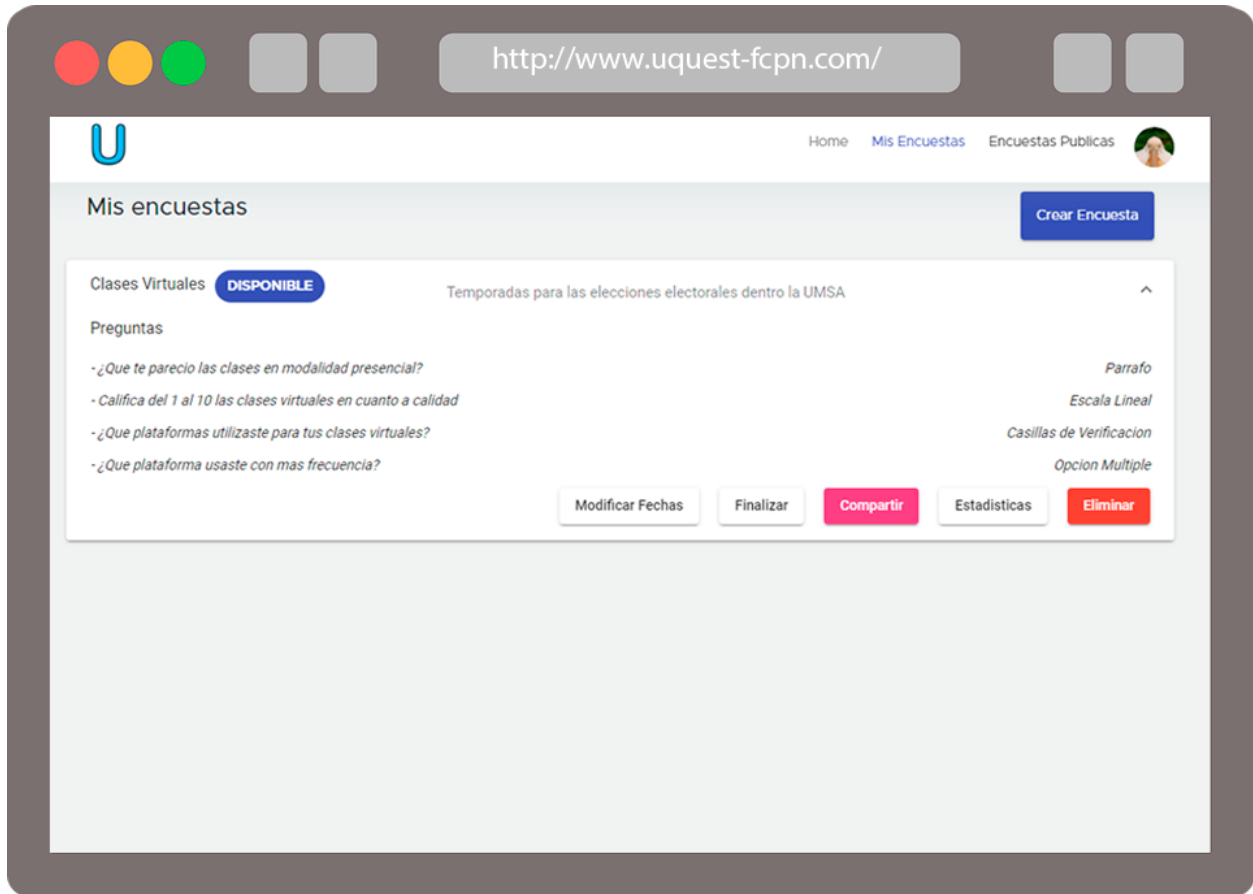
Figura 35 - UQUEST, encuestas públicas



3.6.6. LISTADO DE ENCUESTAS CREADAS

Si el usuario se encuentra previamente *logueado* en el sistema este tendrá disponibles múltiples nuevas acciones, como ser la de mostrar encuestas que creo previamente y visualizar resultados, como también modificar fechas de la encuesta, terminar la encuesta para no recibir más respuestas, acceder a las estadísticas, eliminar la encuesta, compartir la encuesta (para el caso de una encuesta que sea de tipo cerrada, será la única forma de acceder mediante un link proporcionado en la página), tomar en cuenta que el usuario debe haber ingresado al sistema previamente para poder responder encuestas cerradas, para una mayor apreciación se tiene la Figura nro. 38 a continuación, con datos ficticios.

Figura 36 - UQUEST, encuestas del usuario



3.6.7. CREACION DE LA ENCUESTA

Un usuario *logueado* tiene la opción de crear una encuesta propia como el desee con limitaciones al crear preguntas por tipo que son las siguientes detallados en la Figura nro. 39:

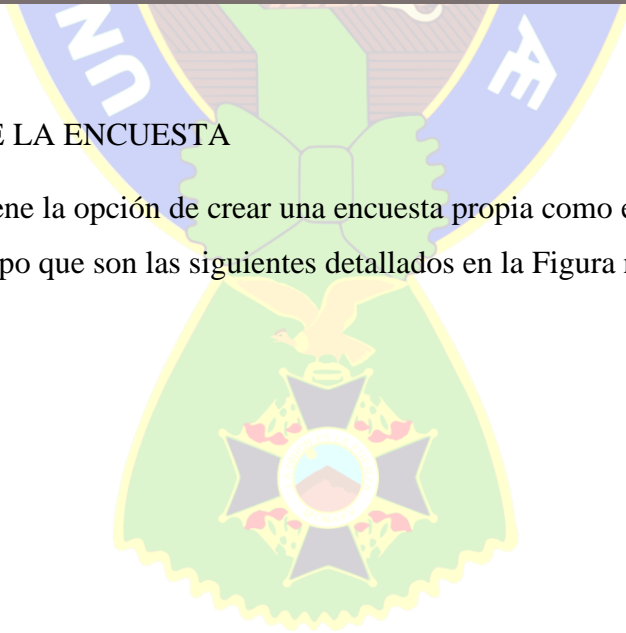
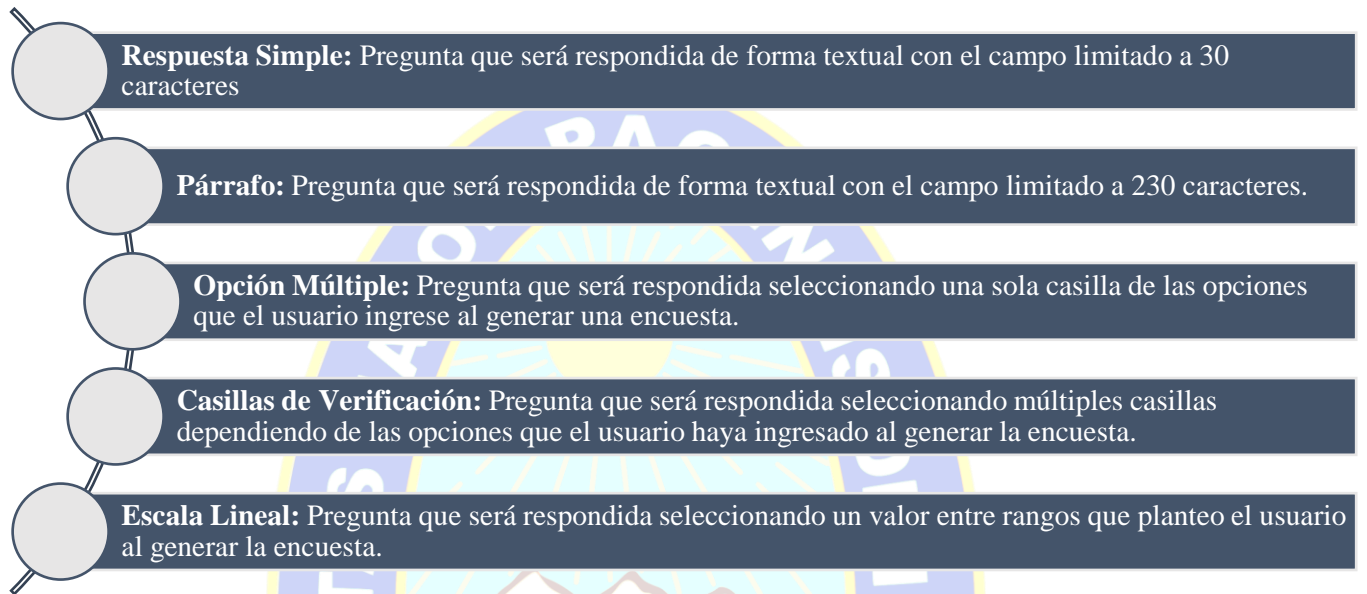


Figura 37 - Tipos de Preguntas



Se tiene la Figura nro. 40 con la pregunta de tipo opción múltiple seleccionada como sencillo ejemplo.

Figura 38 - UQUEST, generar encuesta

The screenshot shows the UQUEST web application interface for creating a survey. The browser address bar displays <http://www.uquest-fcpn.com/>. The page features a navigation menu with 'Home', 'Mis Encuestas', and 'Encuestas Publicas', along with a user profile icon. The main form includes a 'Nombre de la encuesta:' field, a 'Descripcion:' field, and dropdown menus for 'Tipo' and 'Categoria'. It also has date pickers for 'Fecha Inicio de Encuesta' and 'Fecha Fin de Encuesta'. A note states: '* Tipo cerrado solo se accedera mediante un link o codigo QR *'. The 'Pregunta 1:' section has an 'Obligatorio' toggle and a text input for the question. Below it, there is a radio button for 'Opcion 1' and buttons for 'Agregar Opcion +' and 'Quitar Opcion -'. A dropdown menu on the right lists question types: 'Respuesta Simple', 'Parrrafo', 'Opcion Multiple' (highlighted), 'Casillas de Verificacion', and 'Escala Lineal'. At the bottom, there are buttons for 'Agregar Pregunta +', 'Quitar Pregunta -', 'Limpiar', and 'Guardar'.

3.6.8. RESOLVER LA ENCUESTA

Para resolver la encuesta se debe ingresar al link proporcionado por el usuario que genero la encuesta, si la encuesta es de tipo abierta el usuario podrá responder la encuesta sin necesidad de ingresar al sistema, por el otro lado si la encuesta es de tipo cerrada, requiere que el usuario tenga una cuenta para poder responderla, los campos a responder de la encuesta dependen de los tipos de preguntas anteriormente ilustrados en la figura nro. 39, se realizó una encuesta de prueba titulada “Clases Virtuales” con las preguntas ficticias para la presentación de la interfaz mostrada a continuación en la Figura nro. 41:

Figura 39 - UQUEST, responder encuesta

U Home Mis Encuestas Encuestas Publicas

Clases Virtuales

¿Que te parecio las clases en modalidad presencial?

Ingrese su respuesta
Muy malas a comparación de las clases virtuales, porque resultan mas cómodas

Maximo 230 caracteres 76/230

Califica del 1 al 10 las clases virtuales en cuanto a calidad

1 — 10

¿Que plataformas utilizaste para tus clases virtuales?

Zoom
 Microsoft Teams
 Jitsi
 Google Meet
 Otros

¿Que plataforma usaste con mas frecuencia?

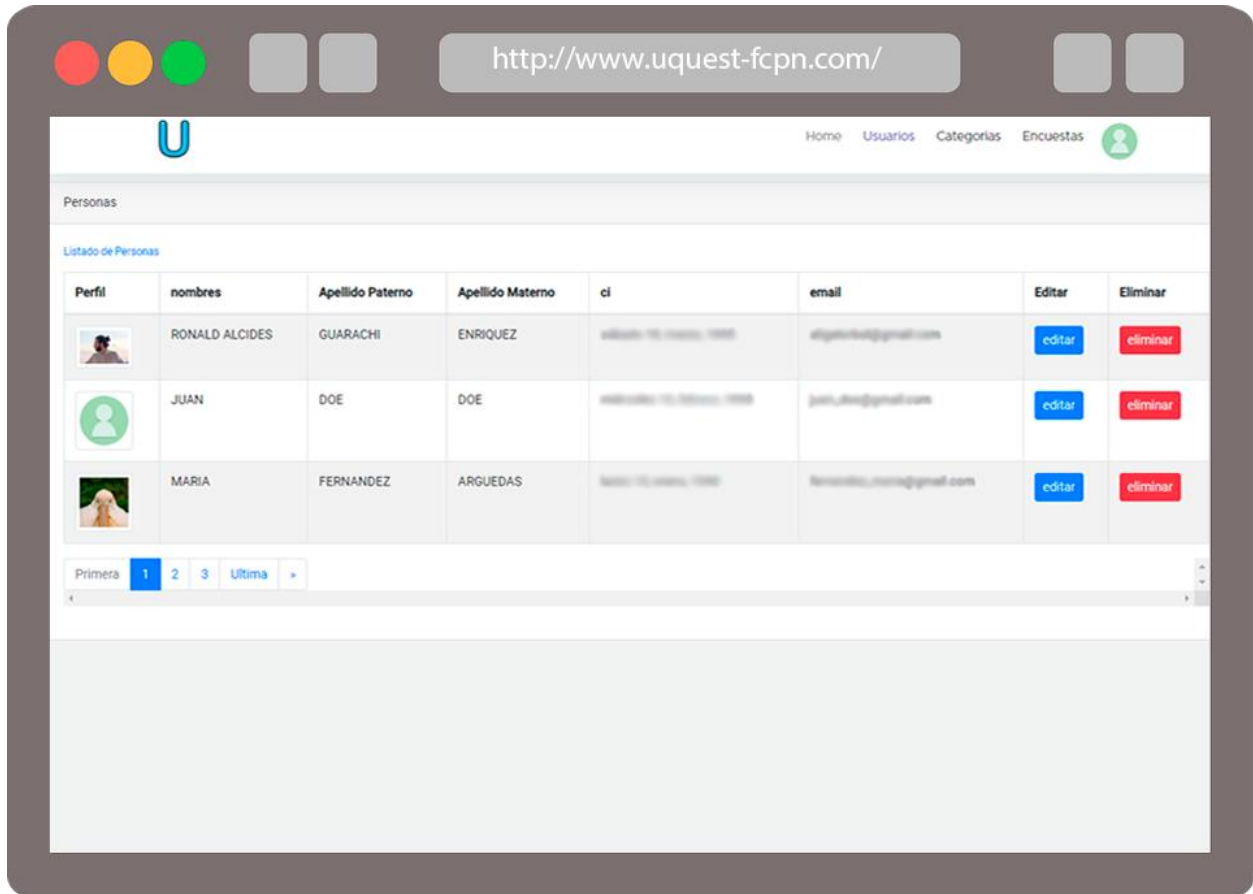
Zoom Microsoft Teams Jitsi Google Meet Otros

Enviar

3.6.9. ADMINISTRACION DE LA BASE DE DATOS

El administrador tendrá una interfaz como ser el de la administración de usuarios, donde podrá eliminar usuarios como también administrar su información si llegara a suceder algún acontecimiento no previsto, la interfaz del administrador es la que se muestra en la Figura nro. 42 a continuación:

Figura 40 - UQUEST, usuarios (administrador)



También tendrá la opción de agregar y quitar categorías de las encuestas, es decir un CRUD de categorías mostradas en la Figura nro. 43.

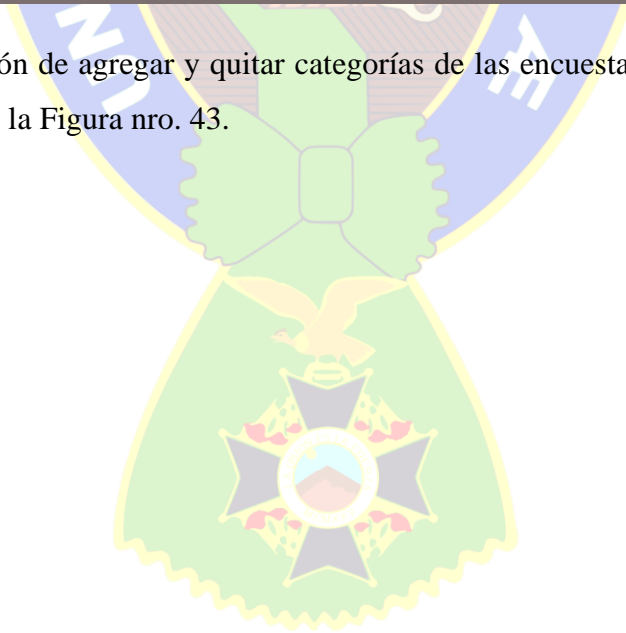
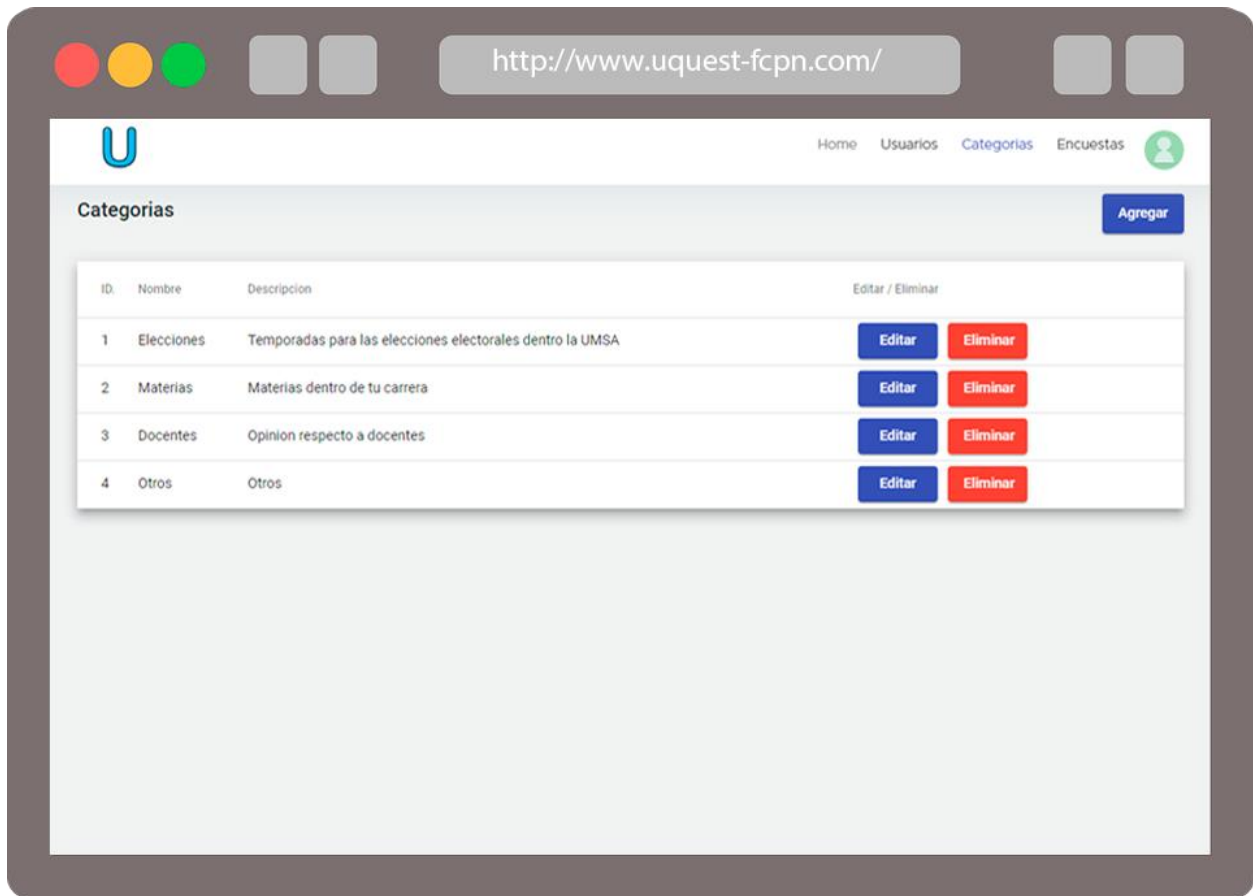


Figura 41 - UQUEST, CRUD categorias (interfaz)



También como la administración de las encuestas generadas, pero en esta solo se podrá eliminar las existentes, debido al conflicto que puede generar al mostrar las estadísticas con información ya existente de preguntas con distinto tipo, la interface por parte del administrador en esta parte es la que se muestra en la Figura nro. 44 a continuación.

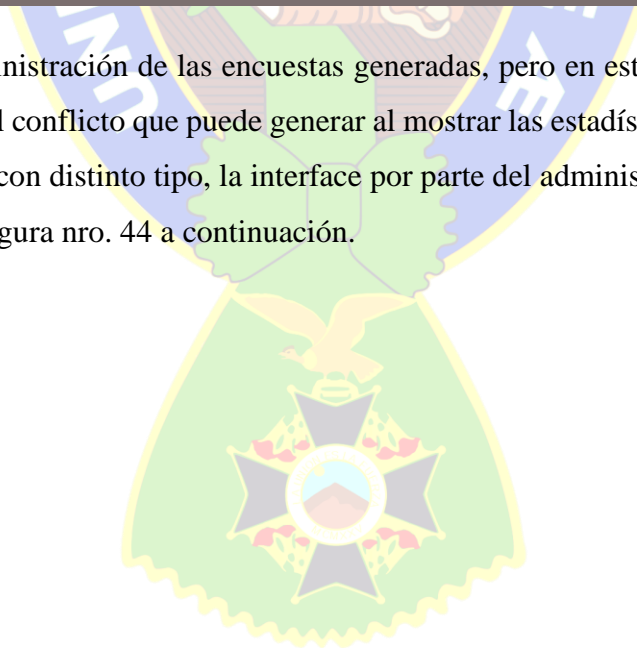
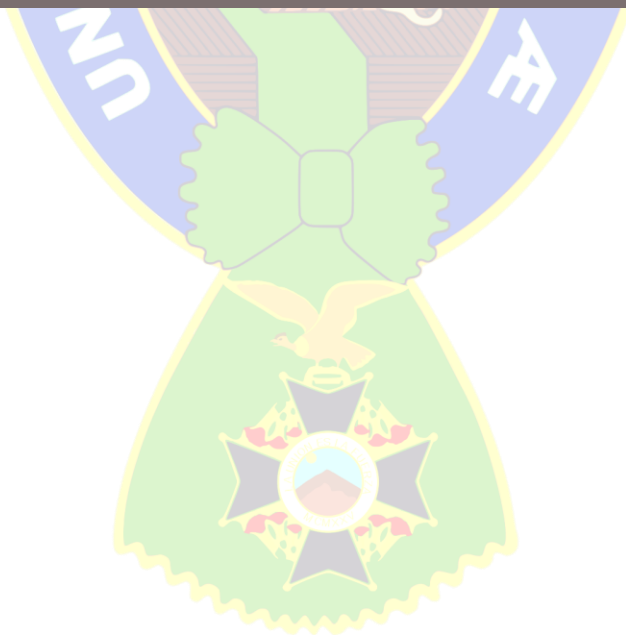
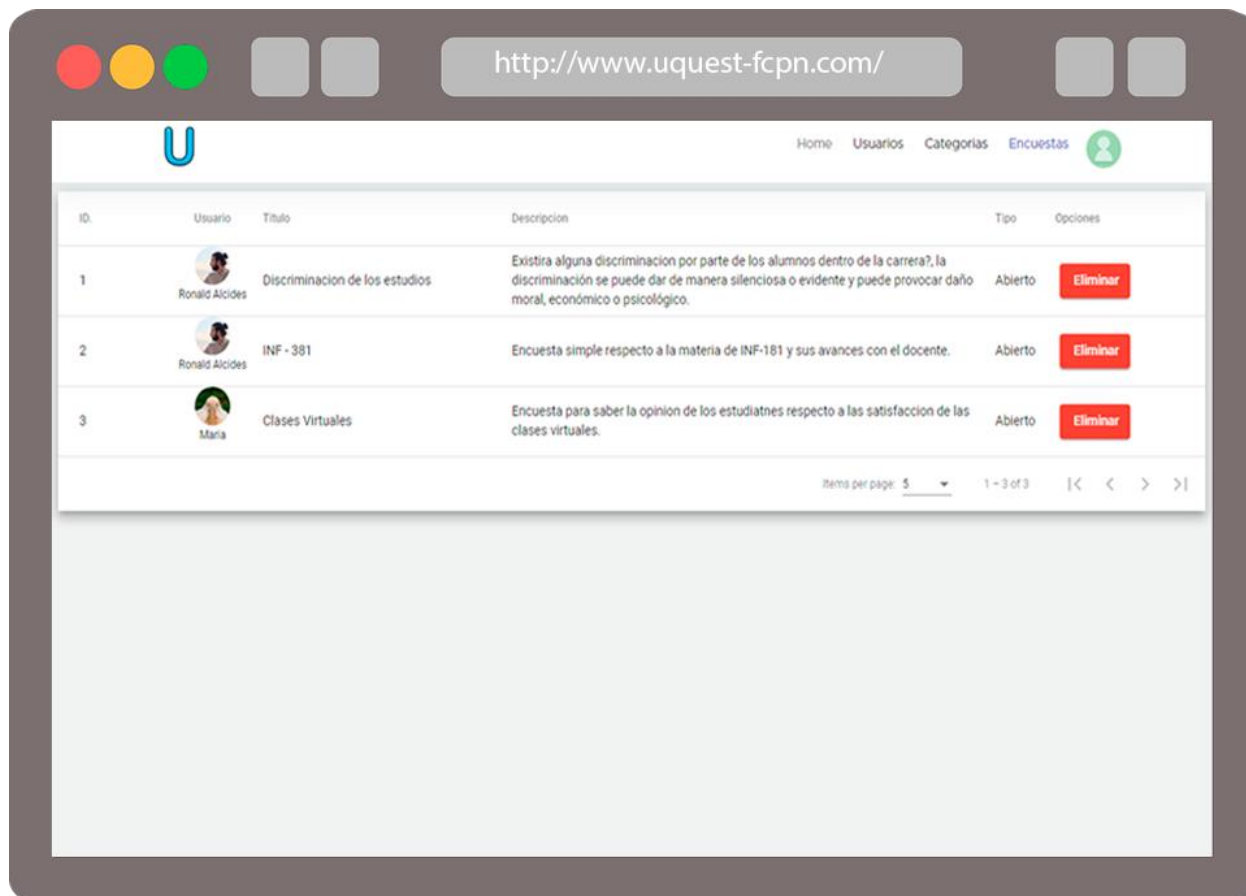


Figura 42 - UQUEST, encuestas (administrador)



3.7. RESPONSABILIDAD DE AUTOR

El presente documento (Proyecto de Grado) es un trabajo escrito en su integridad por Ronald Alcides Guarachi Enriquez (El autor), excluyendo las partes que son extraídas de libros y artículos hallados en el internet, que están propiamente referenciados en la bibliografía.

El código fuente de la aplicación desarrollada en el presente proyecto de grado está alojado como proyecto OpenSource en GitHub bajo licencia MIT, si se desea descargar el código fuente bajo previa autorización enviar un correo a aligatorbol@gmail.com , para acceder al siguiente repositorio: <https://github.com/skynoorz/uquest-angular>. Debajo, los detalles de la licencia MIT.

MIT License

Copyright (c) 2021 Ronald Alcides Enriquez Guarachi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



CAPÍTULO IV

RESULTADOS

Para este capítulo de resultados, se presenta las pruebas y análisis posterior al desarrollo e implementación de la plataforma cumpliendo con los estándares ISO/IEC 9126 y 25000, debido a que un sistema con las cualidades que presenta el actual proyecto necesita aprobar una serie de pruebas e inspecciones para asegurar que se cumple con los requisitos asignados.

4.1. PRUEBAS UNITARIAS Y DE INTEGRACIÓN

Dentro de las actividades que realiza el sistema se corroboró el apropiado funcionamiento de cada módulo haciendo pruebas unitarias para cada Endpoint, es decir cada API REST que posea el sistema. La interacción entre el Frontend y el Backend es vital para un funcionamiento óptimo, realizando peticiones HTTP como ser get, post, put, delete se pudo evidenciar el funcionamiento CRUD⁷ respectivo.

Para pruebas Unitarias del servidor se hizo uso de Postman que permite hacer solicitudes HTTP sin necesidad de tener un Frontend configurado lo que nos permitió corroborar un buen funcionamiento de los endpoints en el Backend, en la Figura nro. 45 puede apreciarse una demostración del uso de postman interactuando con el sistema.

⁷ Crear, Leer, Actualizar y Borrar (del original del inglés: Create, Read, Update and Delete)

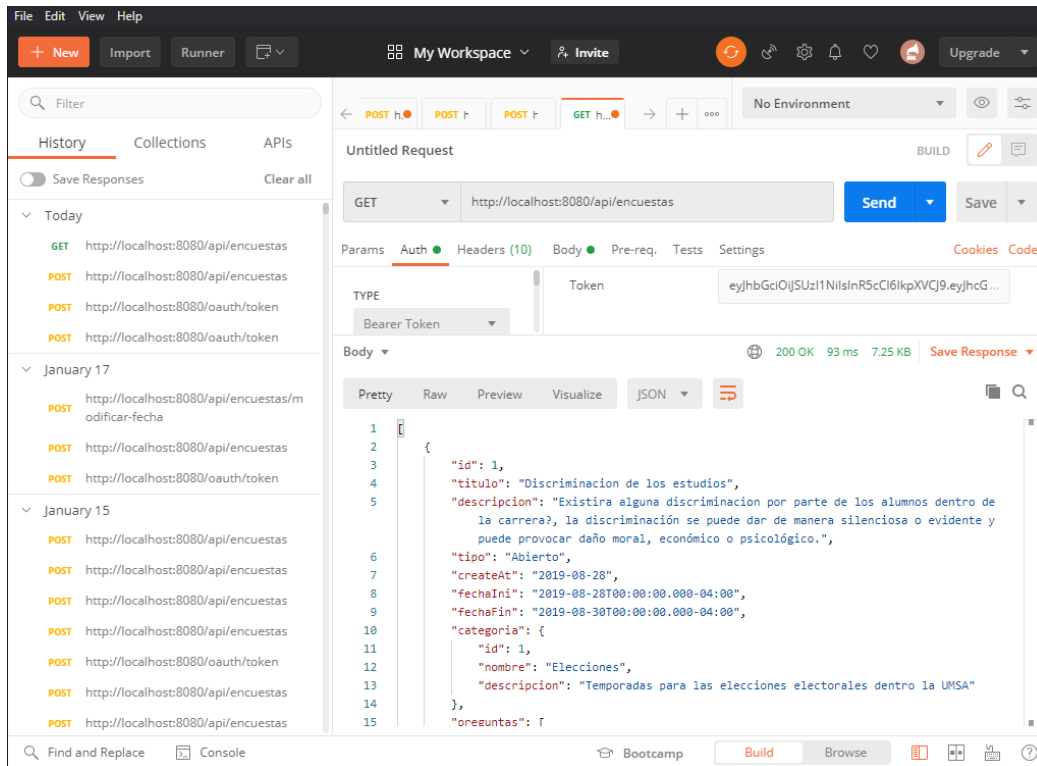


Figura 43 - demostración postman

4.2. PRUEBAS DE ACEPTACIÓN Y FUNCIONALIDAD

Se necesita aprobación de los usuarios para las diversas actividades que se presentan en el product-backlog, para eso se usaron casos de prueba en usuarios que demuestren la aceptación de cada uno.

En el primer caso se tiene la actividad de responder una encuesta, el cual detalla las evaluaciones que se necesitan para concluir el caso como aceptado detallado en la Tabla nro. 9 a continuación:

Código de prueba: 1 **Actividad del cliente:** responder una encuesta
Tipo de prueba: Caso de prueba **Resolución:** aceptado

Descripción de la prueba: Prueba del usuario al responder encuestas generadas por otros usuarios
Evaluación: - Recibir enlace de la encuesta - Llenado de encuesta - Envío de encuesta
Errores detectados: - La dirección URL al recibir un enlace de una encuesta se encontraba errónea direccionándose a otra ubicación.

Corrección ante los casos de prueba:

- Se realizó la corrección de las direcciones URL a la ubicación correcta.

Tabla 9 - Caso de prueba, responder una encuesta

En la Tabla nro. 10 a continuación se muestra los casos de prueba para el caso de los usuarios al consultar resultados de encuestas anteriormente realizadas por otros usuarios.

Código de prueba: 2 **Actividad del cliente:** resultados de

Tipo de prueba: Caso de prueba **encuestas terceras**

Resolución: aceptado

Descripción de la prueba:

Prueba del usuario al visualizar resultados de encuestas generadas por terceros usuarios

Evaluación:

- Visualización listado de encuestas publicas
- Visualización de estadísticas por encuesta
- Generar archivo pdf con datos estadísticos

Errores detectados:

- Las fechas mostradas al generar un archivo pdf no poseían un formato diferente a la zona horaria local mostrando así un día menos a la esperada.

Corrección ante los casos de prueba:

- Se realizó la corrección de fechas al generar un archivo pdf.

Tabla 10 - Caso de prueba, resultados de encuestas terceras

En la Tabla nro. 11 se presenta el caso de prueba para los usuarios que deseen registrarse e ingresar al sistema.

Código de prueba: 3 **Actividad del cliente:** registro y log in

Tipo de prueba: Caso de prueba **Resolución:** aceptado

Descripción de la prueba:

Prueba de registro del usuario y posterior ingreso al sistema

Evaluación:

- Registro del usuario
- Ingreso al sistema mediante usuario y contraseña

Errores detectados:

- Error con la validación de correos ya registrados en el sistema.
- Al realizar el registro de un usuario se omitía el llenado de captcha considerándose como no obligatorio.

Corrección ante los casos de prueba:

- Se corrigió la validación con los correos al realizar un registro de un usuario.
- Se corrigió el uso de captcha incluyéndolo en el registro como obligatorio.

Tabla 11 - Caso de uso, registro y log in

En la Tabla nro. 12 se presenta el caso de uso de un usuario al crear su propia encuesta.

Código de prueba: 4 **Actividad del cliente:** creación de una

Tipo de prueba: Caso de prueba **encuesta**

Resolución: aceptado

Descripción de la prueba: Prueba de generación y creación de una encuesta
Evaluación: - Generación de una encuesta - Validación de campos - Finalización de la encuesta y posterior creación.
Errores detectados: - ninguno
Corrección ante los casos de prueba: - ninguno

Tabla 12 - Caso de uso, creación de una encuesta

Posterior a la creación de una encuesta propia se tiene el caso de uso presentada en la Tabla nro. 13 al revisar estadísticas de la encuesta generada.

Código de prueba: 5 **Actividad del cliente:** estadísticas de encuestas propias del usuario
Tipo de prueba: Caso de prueba **Resolución:** aceptado

Descripción de la prueba: Prueba de visualización de encuestas propias
Evaluación: Visualización de las estadísticas de la encuesta generada por el usuario Exportación por archivo pdf.
Errores detectados: - la visualización de las estadísticas de una encuesta no eran responsivas
Corrección ante los casos de prueba: -xxx

Tabla 13 - Caso de uso, estadísticas de encuestas propias del usuario

4.3. PRUEBAS DE RENDIMIENTO

Para realizar las pruebas de rendimiento se utilizó la popular herramienta JMeter para ejecutar pruebas de stress y cargar muchos de los endpoints que brinda el sistema, esto es muy importante debido a que para responder encuestas se asume que múltiples usuarios estarán conectados al servidor y estos deberán enviar y recibir respuestas en tipo JSON simultáneamente, cuando el usuario responde encuestas es cuando se recibe una gran carga de respuesta, debido a que recibe toda la encuesta completa, el usuario quien lo creo, la carrera, el instituto y otros datos secundarios para poder validar la encuesta para posterior envío, analizando los datos y considerando que la api a la que se hizo la prueba tiene la mayor carga de API REST con un tamaño de 7440 bytes en todos los casos.

Se establecieron a 30 usuarios simultáneos que hicieran la petición a la obtención de toda la información de las encuestas que pudieran conectarse al servidor, obteniendo así un resultado óptimo con una media de 292ms por cada petición, siendo 427ms con mayor tiempo de retorno y 251ms el menor, todos ellos con un estado de respuesta 200OK. En la Figura nro. 46 se puede apreciar con mayor detalle las conexiones simultaneas.

Figura 44 - JMeter, conexiones de usuarios múltiples

The screenshot displays the JMeter interface. On the left, a tree view shows 30 'Petición HTTP' items, each with a green checkmark icon. The right pane shows the 'Resultado del Muestreador' for a selected item. The data is as follows:

Nombre del hilo	Grupo de usuarios simultaneos 1-30
Comienzo de muestra	2021-01-24 16:14:17 BOT
Tiempo de carga	308
Connect Time	129
Latencia	307
Tamaño en bytes	7440
Sent bytes	140
Headers size in bytes	441
Body size in bytes	6999
Conteo de muestra	1
Conteo de error	0
Código de respuesta	200

The response headers section shows:

```

HTTP/1.1 200
Server: Cowboy
Connection: keep-alive
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Type: application/json
Transfer-Encoding: chunked
Date: Sun, 24 Jan 2021 20:14:19 GMT
Via: 1.1 vegur
  
```

The body section shows:

```

Campo adicional
Type Result: HTTPSampleResult
ContentType: application/json
DataEncoding:
En bruto: Parseado
  
```

De las pruebas realizadas con esta herramienta se puede concluir que el sistema trabaja en sus mejores condiciones cuando se accederán a él entre 1 a 30 usuarios simultaneas, en las mejores condiciones, es decir que el servidor tenga a disposición el procesamiento necesario para este.

4.4. ENCUESTA A USUARIOS

Para las encuestas a los usuarios se utilizó la plataforma actual aprovechando que el sistema cumple las características y así medir la satisfacción de los usuarios respecto a la misma. Para poder realizar esta encuesta se utilizó la escala de Likert para ver los niveles de conformidad que tiene el usuario con la plataforma presentadas con anterioridad el cual en la plataforma actual se utilizó un tipo de pregunta de escala lineal donde 1 el usuario se encuentra “En desacuerdo” y en 5 cuando el usuario se encuentra “Muy de acuerdo”, a su vez se utilizó estos valores para el cálculo porcentual.

Los resultados que las encuestas exponen presentan que, según la opinión de un usuario estudiante, la plataforma presenta un nivel considerable de inteligibilidad de un porcentaje de 83.3% brindando facilidad a los usuarios el aprendizaje del uso del sistema y el trabajo con este como se muestra en la tabla nro. 14. El diseño, la generación de las encuestas, la presentación de resultados fue significativo para la consideración de los estudiantes.

	Muy de acuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo	Algo en desacuerdo	En desacuerdo
1. El uso de la aplicación me resultó fácil.	26	20	2	0	0
2. Visualizar las encuestas me parece fácil y de rápido acceso.	20	10	17	1	0
3. Responder encuestas es intuitivo	17	24	6	0	1
4. Fue fácil para mí el aprender a usar el sistema	17	18	13	0	0
5. La interfaz del sistema es atractiva	23	20	5	0	0
6. Los colores utilizados en el sistema son adecuados	18	14	9	5	2
TOTAL	121	106	52	6	3

Tabla 14 - Encuesta a usuarios estudiantes

En la Tabla nro. 14 se muestran las preguntas realizadas a los usuarios que generaron su propia encuesta y preguntas referentes a esta sección.

	Muy de acuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo	Algo en desacuerdo	En desacuerdo
1. La generación de las encuestas me parece óptimo	21	24	3	0	0
2. Encontré los tipos de preguntas que deseaba generar	15	13	19	0	1
3. La información requerida es justa	8	24	15	1	0
4. Generar encuestas me parece intuitivo	21	18	8	1	0
5. La interfaz del sistema es atractiva	14	27	6	1	0
6. Recomendaría el sistema a otras personas	27	13	6	2	0
TOTAL	106	119	57	5	1

Tabla 15 - Encuestas a usuarios de creación de encuestas

Como conclusión a opinión de los usuarios se puede observar que una gran parte de la población está de acuerdo a muchas de las características que se presenta en UQUEST, además como sistema para poder evaluar la usabilidad, dando así un valor de 82.5% en promedio.

4.5. EVALUACIÓN DE CALIDAD

Para esta sección se tomó en cuenta la metodología de calidad GQM mencionada con anterioridad, tomando en cuenta los puntos de la ISO/IEC 25000 se realizarán las preguntas de evaluación según las categorías denotadas en él y asignando la métrica correspondiente. Complementando la evaluación realizada a los usuarios mediante encuestas se tiene la evaluación de calidad según GQM denotada en la tabla nro. 16 junto a sus factores de usabilidad representados por sus iniciales:

- I – Inteligibilidad
- A – Aprendizaje
- O – Operabilidad
- P – Protección a errores de usuario
- E - Estética

ID	PREGUNTAS DE USABILIDAD	RESP.	VALOR	FACTOR
----	-------------------------	-------	-------	--------

U1	¿El sistema posee errores con la interface?	negativo	1	E
U2	¿El sistema presenta textos difíciles de comprender?	negativo	1	I y E
U3	¿El sistema presenta fallos ortográficos?	negativo	1	
U4	¿El sistema da a conocer las acciones que realiza con mensajes?	positivo	1	O
U5	¿El sistema da a conocer en qué dirección se encuentra el usuario?	positivo	1	
U6	¿El sistema otorga facilidad de acceso a las secciones?	positivo	1	
U7	Si el sistema presenta algún error ¿se da a conocer al usuario lo sucedido?	positivo	1	P
U8	¿Existe validación correcta de datos al llenar encuestas?	positivo	1	O y P
U9	¿Existe validación correcta de datos cuando se registra un usuario?	positivo	1	
U10	¿El sistema posee iconos para el acceso a diferentes funcionalidades?	positivo	1	O
U11	¿El sistema posee una sección de ayuda o manual para el usuario?	negativo	0	A, I y O
U12	¿El sistema posee una sección de preguntas frecuentes?	negativo	0	
U13	¿El sistema tiene apoyo con la formulación de encuestas, como ser mensajes flotantes o ventanas emergentes?	positivo	1	A
U14	¿La interface del sistema es comprensible al usuario?	positivo	1	I

Tabla 16 – QGM, preguntas usabilidad

Finalizada la evaluación de usabilidad se llegó a un valor de usabilidad de 85% aprobando así con los criterios de calidad que se plantearon concluyendo con una buena capacidad de uso por parte del usuario con el sistema.

Las preguntas que se utilizó para la evaluación de calidad en cuanto a la funcionalidad del sistema se las puede apreciar en la Tabla nro. 17 con las siguientes iniciales a tomar en cuenta:

C – Completitud Funcional
CO – Corrección Funcional
P – Pertinencia Funcional

ID	PREGUNTAS DE FUNCIONALIDAD	RESP.	VALOR	FACTOR
----	----------------------------	-------	-------	--------

F1	¿El sistema cumple con los requerimientos que tiene el usuario?	positivo	1	C
F2	¿El sistema genera encuestas de manera correcta a la interfaz otorgada?	positivo	1	CO
F3	¿El sistema muestra los datos correctos en la sección de estadísticas, de acuerdo a las respuestas de usuarios?	positivo	1	
F4	¿El sistema puede diferenciar entre administrador y un usuario normal?	positivo	1	P
F5	¿El sistema deja libre el acceso a direcciones que solo el administrador puede acceder?	negativo	1	

Tabla 17 - GQM, preguntas funcionalidad

Para la evaluación de funcionalidad se obtuvo un valor de 100% según las características necesarias para esta sección concluyendo que las funciones otorgadas en el sistema satisfacen los requerimientos planteados.

La eficiencia del sistema se puede apreciar en la Tabla nro. 17 y se logró medir bajo los atributos de la metodología con las iniciales denotadas a continuación:

T – Comportamiento en el Tiempo

R – Utilización de Recursos

C – Capacidad

ID	PREGUNTAS DE EFICIENCIA	RESP.	VALOR	FACTOR
E1	¿La respuesta de las peticiones con la página es menor a 1000 milisegundos?	positivo	1	T
E2	¿Se suscitaron errores en pruebas de stress en baja cantidad, o no se dieron?	negativo	1	R
E3	¿El software que utilizó para utilizar el sistema cumple con los requisitos para que el sistema funcione?	positivo	1	
E4	¿El sistema es compatible con el servidor objetivo?	positivo	1	
E5	¿El sistema tiene un límite de 1Mb para cargar archivos en el navegador?	positivo	1	C

E6	¿El servidor cumple con el espacio necesario para instalar la plataforma incluyendo dependencias?	positivo	1	
-----------	---	----------	---	--

Tabla 18 - GQM, preguntas eficiencia

Bajo los criterios observados en la Tabla 18 obteniendo un valor de 100% se puede concluir que la plataforma cumple con funcionalidades que tanto al software utilizado JMeter como el ambiente donde se instaló cumple con las características necesarias para ser un sistema eficiente.

En la tabla nro. 19 se presenta las preguntas elaboradas para medir la mantenibilidad del sistema junto a las iniciales de cada atributo en específico:

- A – Analizabilidad
- C – Cambiabilidad
- P – Pruebabilidad
- M – Modularidad
- R – Reusabilidad

ID	PREGUNTAS DE MANTENIBILIDAD	RESP.	VALOR	FACTOR
M1	¿El sistema está elaborado bajo la arquitectura MVC, para mejorar la organización?	positivo	1	A
M2	¿El sistema posee servicios para mejorar la modularización?	positivo	1	M
M3	¿El sistema es apto a los cambios futuros?	positivo	1	C
M4	Si el sistema presentara fallos ¿Se podría sustentar el fallo?	positivo	1	
M5	¿Existen variables inteligibles y/o entendibles en el sistema?	positivo	1	A
M6	¿Se puede implementar pruebas unitarias futuras?	positivo	1	P
M7	¿Existe una cantidad razonable de comentarios?	negativo	0	A
M8	¿Los endpoints de los servicios son sencillos?	positivo	1	
M9	¿Los endpoints son comprensibles a simple vista?	positivo	1	
M10	¿Los endpoints pueden ser reutilizados por otras plataformas?	positivo	1	R

Tabla 19 - GQM, preguntas mantenibilidad

Finalizando se tiene la evaluación de la mantenibilidad que según la tabla nro. 19 con un valor de 90% del sistema apreciando que el código es limpio y con buenas prácticas estructurando la

información, realizando varias pruebas para evitar fallos, modularizar código, para que así en un futuro el sistema pueda ser compatible a cambios y dar paso a una mejora continua.



CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES



5.1. CONCLUSIONES

Luego de tener claros los objetivos planteados gracias a la problemática del proyecto, además de diseñar, desarrollar y probar la plataforma web UQUEST, aplicando todas las metodologías de análisis y diseño de software se logró cumplir con el objetivo general planteado en un inicio, y el desarrollo del sistema fue exitoso.

Tomando en cuenta los objetivos previamente planteados se llegó a las siguientes conclusiones:

- Se cumplió con el objetivo principal planteado debido a que la plataforma virtual desarrollada recibirá opiniones dentro de la población universitaria para posteriormente compartir la información generada al público si fuera necesario y el usuario así lo quisiera.
- Se obtuvo los requerimientos necesarios para el desarrollo del software lo que permitió concluir el sistema en los ambientes del ITIC en la facultad. La plataforma se desarrolló bajo la tecnología de Spring solicitada, de tal manera que el código quede comprensible a futuras actualizaciones y sea de fácil entendimiento.
- Se estableció una comunicación cliente y servidor con los lenguajes de HTML5, CSS y JavaScript utilizado en el *framework* Angular por parte del cliente y se utilizó PostgreSQL para la base de datos, mejorando así la compatibilidad con los servidores del ITIC FCPN.
- Se implementó una interfaz responsiva mediante la herramienta de Angular Material y Bootstrap para mejorar la interactividad del usuario con la plataforma y así tener un entorno amigable y elegante con el usuario.
- Se realizó el despliegue de la plataforma en los servidores del ITIC de la Facultad para que el costo del sistema sea mínimo.
- Se realizaron las pruebas correspondientes gracias a la metodología SQuaRE para que el sistema cumpla con los atributos de calidad planteados. Gracias a las pruebas realizadas se obtuvo valores de XX en cuanto a funcionalidad, de XX en eficiencia, de XX en usabilidad y de XX en mantenibilidad, cabe mencionar que para que el sistema sea calificado como “seguro” se utilizó el estándar JWT para así mantener la confidencialidad del sistema con el servidor.

De esta manera se logró alcanzar de manera óptima el objetivo general junto a los objetivos específicos, planteado para facilitar el acceso y conocimiento de las diversas opiniones que surgen

en la facultad generando así un foco de opiniones, para así mejorar tanto a nivel carrera como facultad las diversas decisiones en las actividades que se llegara a cabo gracias a esta plataforma.

5.2. RECOMENDACIONES

A culminación del presente proyecto se efectúan las siguientes recomendaciones:

- Se recomienda para migrar el sistema a otro servidor este cumpla con las características mencionadas para el desarrollo, puede que en futuras versiones tanto de Spring Boot como Angular estas queden obsoletas o eliminen funciones que terminan en *deprecated*⁸, si este fuera el caso requeriría una actualización de código para mantener la mantenibilidad del sistema
- El sistema puede ser actualizado para poder recibir más tipos de preguntas, esto puede ser importante en un futuro, para cuando el usuario requiera de nuevas funcionalidades en el sistema.
- Hacer uso de las matrículas de los estudiantes para poder ingresar al sistema y así poder diferenciar si es un estudiante u otro tipo de usuario.
- Si el servidor donde se almacena la información fallara y se perdiera la información el cual almacena los datos de los usuarios como de las encuestas, es recomendable tener una base de datos almacenando así la información que podría llegar a ser crítica para algunos usuarios.
- Agregar un “*dashboard*”⁹ por la parte del administrador que facilite la visión de las encuestas/usuarios, además de recursos utilizados en el sistema y dar a conocer si existe algún componente en mal funcionamiento por parte del servidor.

⁸ significa que sólo está ahí para que las aplicaciones antiguas funcionen

⁹ Una herramienta para obtener información de los datos y centralizarlos para saber que está pasando con la información.

BIBLIOGRAFÍA

- Araque, M. (2017, Febrero 08). *we are marketing*. Retrieved from wam, agencia de marketing digital: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>
- Bahit, E. (2011, Septiembre 14). *Metodologías Ágiles para el Desarrollo de Software*. Retrieved from desarrolloweb: <https://desarrolloweb.com/articulos/roles-scrum.html>
- Bardilla, N. I. (2013, Marzo 12). *actividadreconocimiento*. Retrieved from blog: <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>
- Bernal, J. (2015, Abril 7). *prezi*. Retrieved from <https://prezi.com/qfgafhb2yi35/fiabilidad-iso-9126-para-la-calidad-del-software/>
- Boaventura, J., Peña, E., Verdecia, P., & Fustiel, Y. (2016). Elección entre una metodología ágil y tradicional basado en técnicas de soft computing. *Revista Cubana de Ciencias Informaticas*, 145-158.
- Camara Malaga. (2019, 11 28). Retrieved from Master Malaga Web: <https://www.master-malaga.com/empresas/criterios-nivel-calidad-producto/>
- Delía, L. N. (2017, Mayo 24). *Sedici*. Retrieved from Sedici: Repositorio Institucional de la UNLP: <http://sedici.unlp.edu.ar/handle/10915/60497>
- Devs, Q. (2019, Septiembre 16). *QualityDevs*. Retrieved from Quality Devs: <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>
- Españon, A. P. (2019, 12 03). *WeareDrew*. Retrieved from WeareDrew Website: <https://blog.wearedrew.co/ventajas-y-desventajas-de-la-metodologia-scrum>
- Espinosa, H. (2014, Noviembre 5). *prezi*. Retrieved from https://prezi.com/surwzan_rji_/mantenibilidad-iso-9126/
- Ghosh, S. (2018, Febrero 21). *Systemic comparison of the application*. Retrieved from Project Management Center for Excellence: <https://pm.umd.edu/files/public>
- Guzmán, A. J. (2018, Agosto 21). Retrieved from <https://www.udemy.com/course/spring-framework-5/>
- Hernandez, U. (2015, Febrero 22). Retrieved from <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- ISO25000. (2020). *ISO 25000*. Retrieved from ISO 25000: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- JWT. (n.d.). *jwt.io*. Retrieved from jwt.io: <https://jwt.io/introduction/>
- Khurana, H., & Sohal, J. (2011). The necessitate of contemporary. *International Journal*, 1031-1039.
- Kuz, A., Falco, M., & Giandini, R. S. (2018). Comprendiendo la Aplicacion de Scrum. *Articulos Originales*.

- López, A. (2019, Junio 03). *OpenWebinars*. Retrieved from Desarrollo:
<https://openwebinars.net/blog/que-es-postman/>
- Lovelle, J. C. (n.d.). Retrieved from
<http://di002.edv.uniovi.es/~cueva/asignaturas/masters/2005/MetricasUsabilidad.pdf>
- Maximilians, L. (2016). *Reserch Unit of Programming and Software Engineering*. Múnich: Universität München.
- Michas, N. (2020, Junio 26). *medium*. Retrieved from BetterProgramming:
<https://betterprogramming.pub/back-end-data-validations-73ea9004c6d7>
- Ministerio de Educación. (2017, Septiembre 22). YouTube. Peru. Retrieved from Youtube:
https://www.youtube.com/watch?v=2_87761_yJg
- Navarro Cadavid, F. M. (2013). Revision de metdologias ágiles para el desarrollo de software. *PROSPECTIVA*, 30-39.
- Olsina, L., & Rossi, G. (2013). Measuring Web application quality with WebQEM. *Reserch Gate*, 1-11.
- Opini3n. (2019, Junio 08). *Unicef lanza en Bolivia la plataforma digital U-Report*.
- Perry, J. S. (2017, Julio 11). *IBM Developer*. Retrieved from IBM Developer:
<https://developer.ibm.com/es/languages/java/tutorials/j-spring-boot-basics-perry/>
- Yepes, J. F. (2013, Noviembre 10). *Prezi*. Retrieved from <https://prezi.com/xpg0h7trtnjg/norma-iso-9126-eficiencia/>

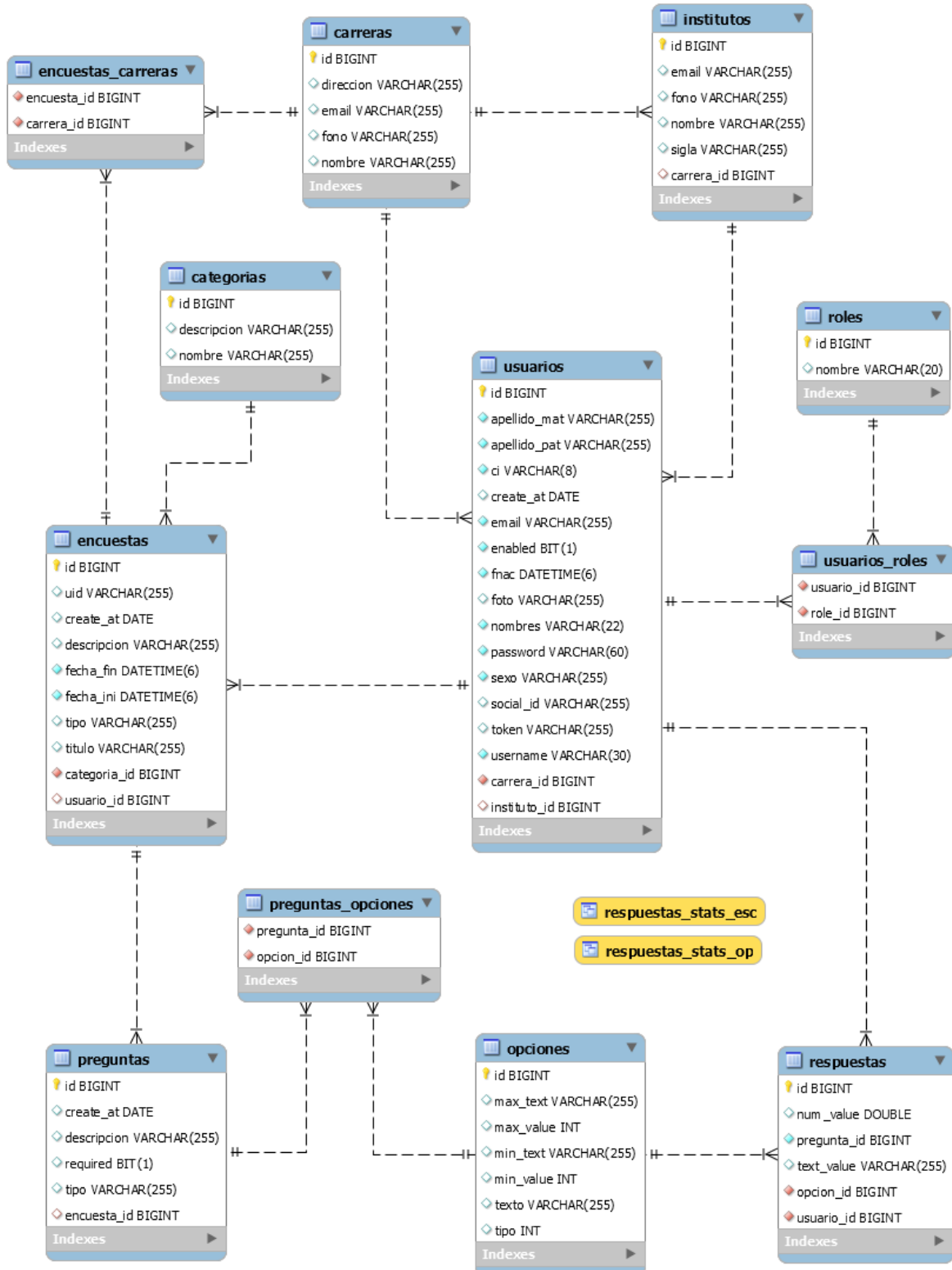
TABLA DE ILUSTRACIONES

Figura 1 – Fases de la investigaci3n. (Ministerio de Educaci3n, 2017).....	15
Figura 2 - Fases de la metodología Scrum.....	18
Figura 3 - Jerarquía UML.....	27
Figura 4 - Patron MVC (Model View Controller).....	28
Figura 5 - GQM características de calidad.....	34
Figura 6 - Gesti3n base de datos, caso de uso.....	46
Figura 7 - Gesti3n base de datos, diagrama de estados.....	47
Figura 10 - Registro y Login, casos de uso.....	48
Figura 11 - Diagrama de estados, registro y login de usuario.....	49
Figura 12 - Diagrama de secuencia, registro y login.....	50
Figura 13 - Casos de uso, gesti3n de usuarios.....	52
Figura 14 - Diagrama de estados, administraci3n de usuarios.....	53
Figura 15 - Diagrama de secuencia, registro y login.....	54
Figura 16 - Diagrama de clases, administraci3n de cuentas de usuario.....	55
Figura 17 - Diagrama relacional, m3dulo de administraci3n de cuentas de usuario.....	56
Figura 18 - Diagrama de casos de uso, administraci3n de carreras e institutos.....	57
Figura 19 - Diagrama de estados, eliminaci3n de instituto.....	58
Figura 20 - Diagrama de secuencia, administraci3n de carreras e institutos.....	58
Figura 21 - Diagrama de clases, administraci3n de carreras e institutos.....	60

Figura 22 – Diagrama relacional, administración de carreras e institutos	61
Figura 23 - Diagrama de casos de uso, administración y creación de encuestas	62
Figura 24 - Diagrama de estados, administración y creación de encuestas	63
Figura 25 - Diagrama de secuencia, administración y creación de encuestas.....	64
Figura 26 - Diagrama de clases, administración y creación de encuestas.....	66
Figura 27 - Diagrama relacional, administración y creación de encuestas	69
Figura 28 - Diagrama de casos de uso, respuesta a encuestas.....	70
Figura 29 - Diagrama de estados, módulo de respuesta a encuestas.....	71
Figura 30 - Diagrama de secuencia, módulo de respuesta de encuestas	72
Figura 31 - Diagrama de clases, respuesta de encuestas	74
Figura 32 - Diagrama relacional, respuestas	75
Figura 33 - UQUEST, Pantalla Principal.....	76
Figura 34 - UQUEST, registro usuario	77
Figura 35 - UQUEST, ingreso al sistema	78
Figura 36 - UQUEST, perfil usuario.....	79
Figura 37 - UQUEST, encuestas públicas	80
Figura 38 - UQUEST, encuestas del usuario	81
Figura 39 - Tipos de Preguntas	82
Figura 40 - UQUEST, generar encuesta	83
Figura 41 - UQUEST, responder encuesta.....	84
Figura 42 - UQUEST, usuarios (administrador).....	85
Figura 43 - UQUEST, CRUD categorías (interfaz).....	86
Figura 44 - UQUEST, encuestas (administrador).....	87
Figura 45 - demostración postman.....	91
Figura 46 - JMeter, conexiones de usuarios múltiples.....	94
Tabla 1 - Fases de investigación aplicada.....	16
Tabla 2. Comparación de metodología ágil y tradicional	24
Tabla 3 - Jerarquía características de calidad ISO 25010 (Square).....	35
Tabla 4 Tabla de requerimientos.....	40
Tabla 5 Product-Backlog	42
Tabla 6 Planificación Sprint N° 1	45
Tabla 7 - Planificación Sprint nro. 2.....	47
Tabla 8-Planificación de administración de cuentas de usuario	50
Tabla 9 - Caso de prueba, responder una encuesta	92
Tabla 10 - Caso de prueba, resultados de encuestas terceras.....	92
Tabla 11 - Caso de uso, registro y log in	92
Tabla 12 - Caso de uso, creación de una encuesta.....	93
Tabla 13 - Caso de uso, estadísticas de encuestas propias del usuario	93
Tabla 14 - Encuesta a usuarios estudiantes.....	95
Tabla 15 - Encuestas a usuarios de creación de encuestas.....	96
Tabla 16 – QGM, preguntas usabilidad	97
Tabla 17 - GQM, preguntas funcionalidad	98
Tabla 18 - GQM, preguntas eficiencia.....	99
Tabla 19 - GQM, preguntas mantenibilidad	99

ANEXOS

ANEXO A - Base de datos en MySQL (desarrollo)

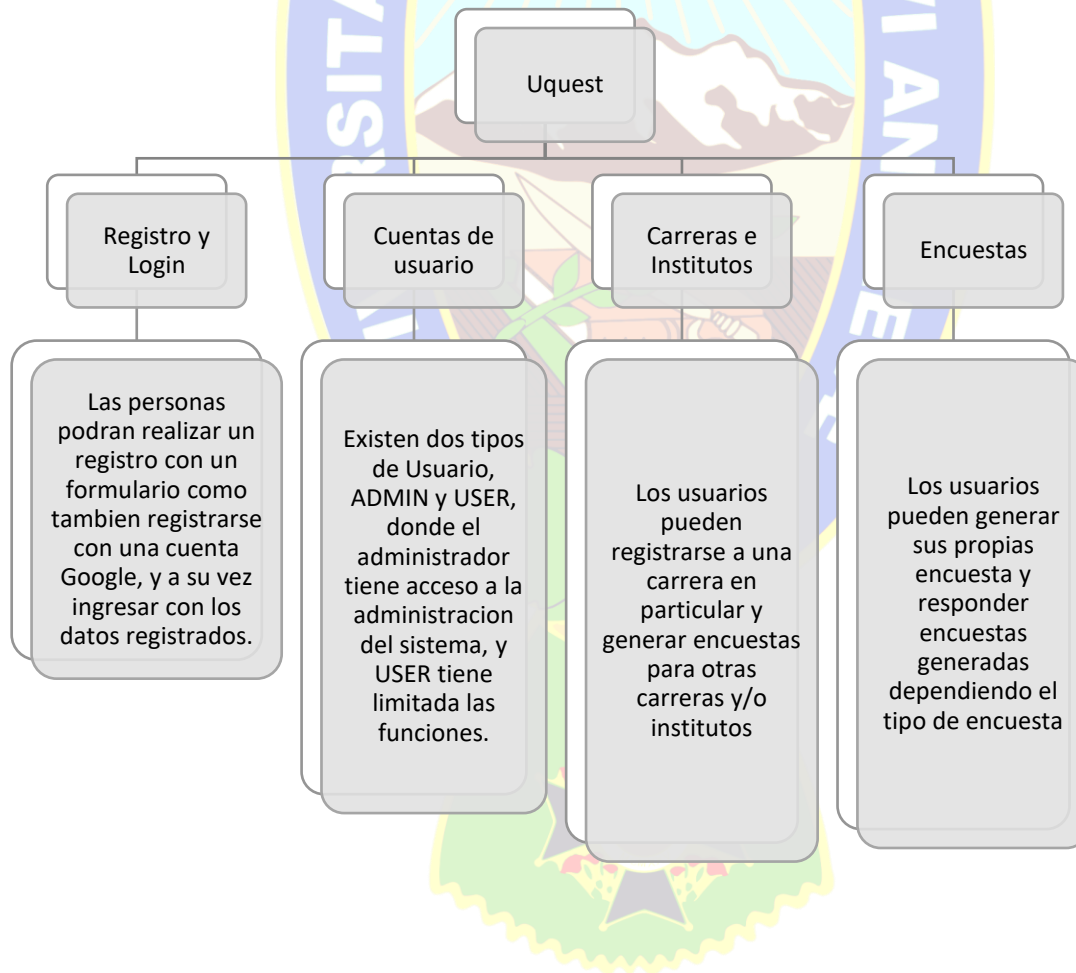


ANEXO B - MANUAL DE USUARIO

MAPA DEL SISTEMA

Modelo Lógico

En este apartado se hará una descripción del sistema mediante diagramas en formato libre con un enfoque top-down. Es decir, se comenzará describiendo el sistema en su entorno, se continuará con una descomposición lógica del sistema por módulos, y a continuación se describirá cada módulo.



Navegación

En este punto se describirá la navegación a través de un grafo de ventanas. En este diagrama se representarán las ventanas del sistema y mediante flechas las navegaciones entre las mismas. Se ha de representar los caminos más significativos.

DESCRIPCIÓN DE VISTAS

En esta sección se describirá la interfaz gráfica con las principales características de la aplicación. Se expondrán las pantallas anteriormente expuestas, así como las dependientes. Explicando las distintas pantallas de la aplicación siguiendo los caminos lógicos que el usuario realizaría.

Para cada pantalla, se explican los posibles errores que pueden tener.

Ventana de registro

Cabe mencionar que la sección mas morosa del sistema es la del registro debido a la cantidad de datos solicitados pero necesarios como se observa en la ilustración nro. 1. A continuación:

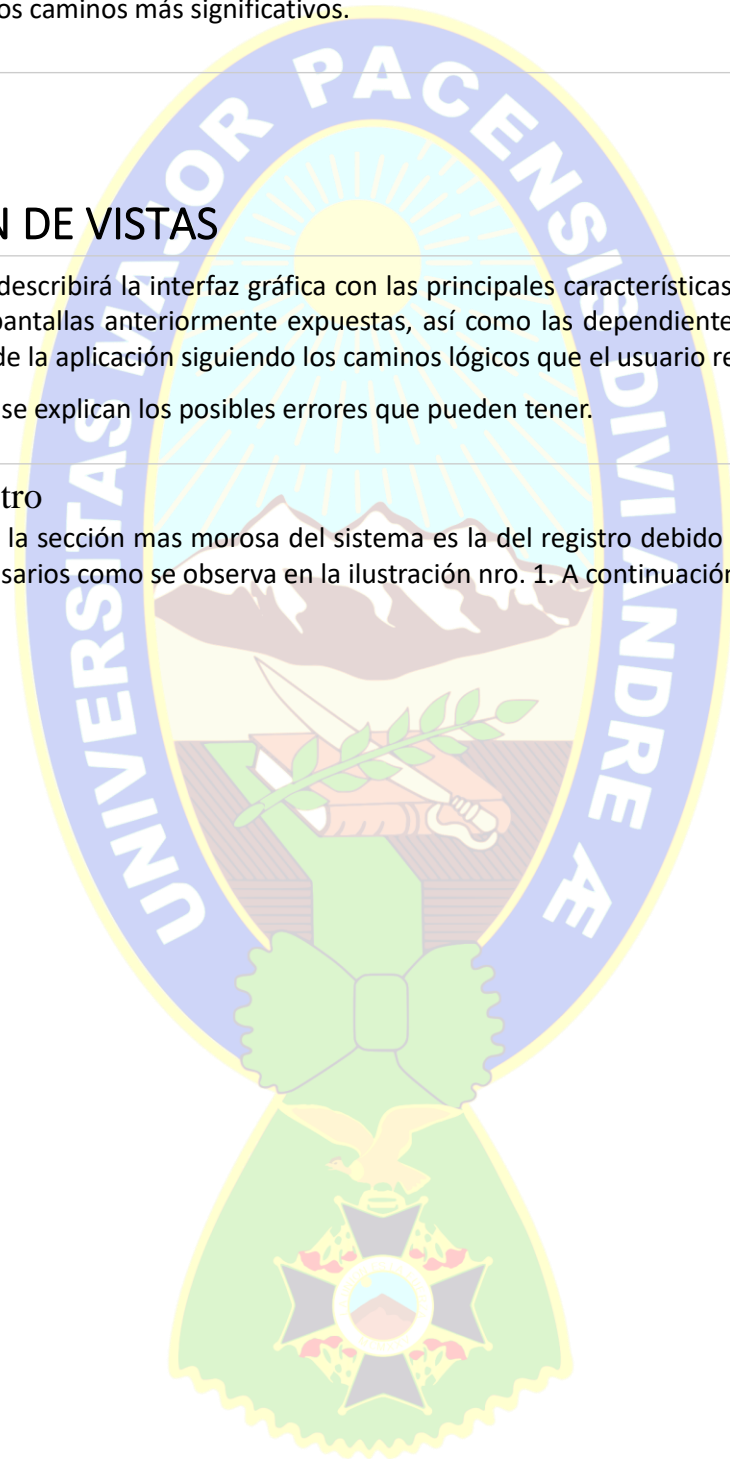


Ilustración 1 - Registro usuario

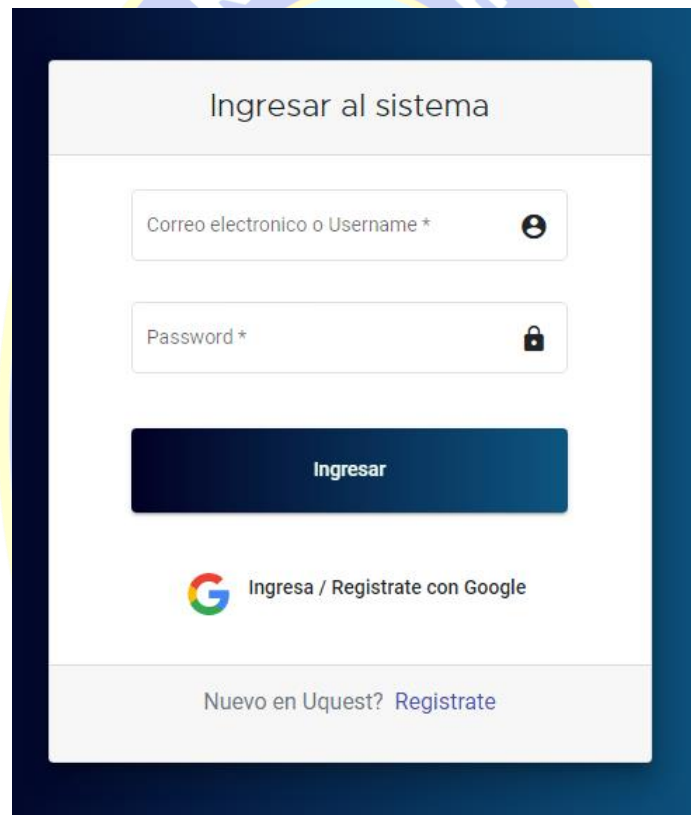
Se pueden apreciar diversos campos obligatorios:

- Nombres: nombres del usuario (no acepta caracteres especiales)
- Apellido Paterno: apellido paterno del usuario (no acepta caracteres especiales)
- Apellido Materno: apellido materno del usuario (no acepta caracteres especiales)
- CI: Cedula de identidad o carnet del usuario (no acepta letras)
- Sexo: sexo del usuario, limitado a la selección de “Femenino” y “Masculino”
- Fecha de Nacimiento: la fecha de nacimiento del usuario (selección y validado con dd/mm/yyyy)
- Carrera: la carrera donde pertenece el usuario de la facultad como ser “Informática”, “Física”, “Química”, “Matemática” y “Estadística” actuales hasta la fecha del segundo semestre del 2021
- Instituto: campo no obligatorio donde se seleccionará si el usuario pertenece a un instituto en específico dentro de la carrera perteneciente.
- Email: correo electrónico del usuario (valor unico)
- Usuario: nombre de usuario para poder ingresar al sistema

- Contraseña: contraseña del usuario donde se requiere al menos un número.
- Captcha: campo de validación y evitar la generación de usuarios “bots”

Ventana de “login”

Para poder generar una encuesta propia para el usuario y llegar a compartirla al público, este tiene la obligación de estar registrado en el sistema, para ello el usuario tiene la opción de ingresar al sistema mediante la ventana de la ilustración nro. 2.



The image shows a login window with a white background and a dark blue border. At the top, the title "Ingresar al sistema" is centered. Below the title are two input fields: "Correo electronico o Username *" with an eye icon and "Password *" with a lock icon. A dark blue button labeled "Ingresar" is positioned below the fields. Underneath the button is a Google logo followed by the text "Ingresa / Regístrate con Google". At the bottom, there is a link that says "Nuevo en Uquest? Regístrate".

Ilustración 2 – Login

Como se puede apreciar en la ilustración nro. 1. El usuario además de ingresar con una cuenta propia este también tendrá la opción de ingresar con su cuenta de Google y además de registrarse con los datos existentes.

Ventana de Encuestas públicas

Dentro de la ventana accesible al botón de la barra de navegación, “Encuestas públicas” se podrá observar la ilustración nro. 2

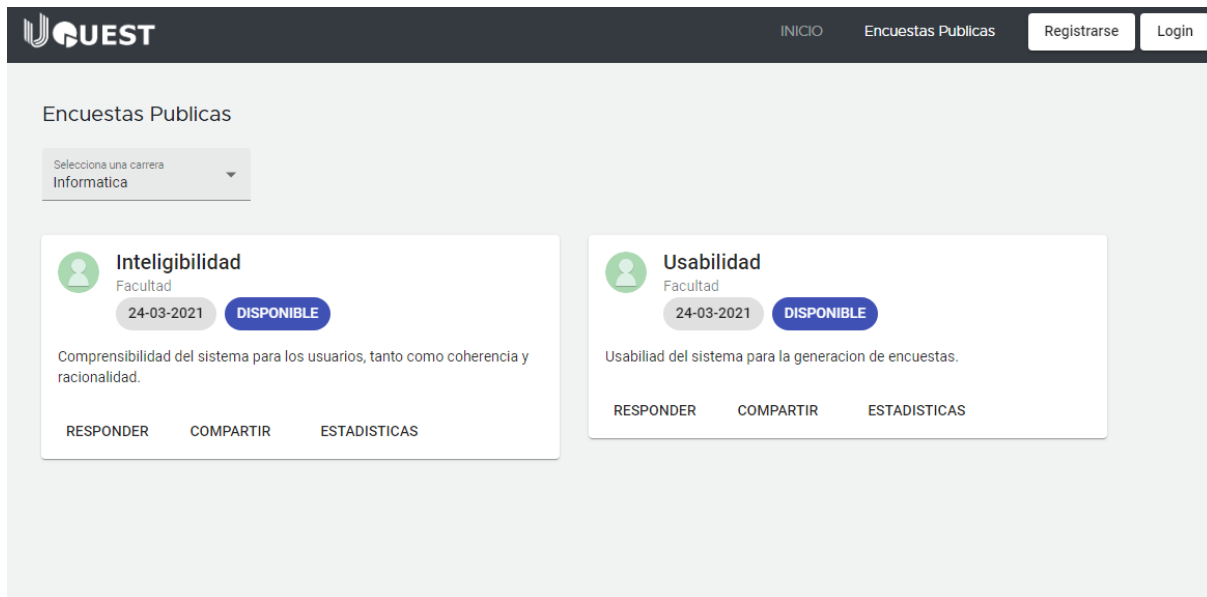


Ilustración 3 - Encuestas públicas

Se puede observar en la ilustración nro. 3 donde se puede acceder a las 3 opciones cuando la encuesta se encuentra disponible entre los rangos programados según el que creo la encuesta

Responder una encuesta pública

Una vez dentro la ventana de encuestas publicas se pueden responder las encuestas publicas generadas en el sistema, para esto con seleccionar "responder" independientemente para cada encuesta este generara la ventana de ilustración nro. 4.

The screenshot shows the QUEST survey interface. At the top, there is a dark header with the QUEST logo on the left and navigation links for 'INICIO', 'Mis Encuestas', and 'Encuestas Publicas' on the right, along with a user profile icon. The main content area is titled 'Encuesta ejemplo' and contains three question blocks. The first block, 'Pregunta 1 Ejemplo *', features a text input field with the placeholder 'Ingrese su respuesta *', a character count '0/30', and a note 'Maximo 30 caracteres'. The second block, 'Pregunta 2 Ejemplo', contains two radio button options: 'Opcion nro. 1 ejemplo' and 'Opcion nro. 2 ejemplo'. The third block, 'Pregutna 3 Ejemplo *', includes a slider control ranging from 'En desacuerdo' to 'De acuerdo'. An 'Enviar' button is located at the bottom right of the form.

Ilustración 4 - Responder encuesta ejemplo.

Los campos a llenar son dependientes a los campos generados por el administrador de la encuesta para ello el usuario puede decidir si una respuesta es obligatoria o no marcados con un asterisco * color rojo para denotarlo.

Compartir Encuesta

El acceso a las encuestas puede realizarse mediante un enlace URL que un usuario puede compartir a otro para llenar las encuestas, existen 2 tipos de encuestas las “Abiertas” y las “Cerradas” las abiertas pueden ser llenadas por cualquier persona que ingrese al sistema entrando en la pestaña de Encuestas abiertas, y las encuestas cerradas pueden accederse solamente mediante un enlace URL y con el requerimiento de que el usuario tenga un previo ingreso al sistema con su cuenta.

Para poder compartir la encuesta a otro usuario el sistema generará el enlace URL o código QR por el usuario haciendo click en el botón “Compartir” mostrando una ventana flotante y habilitando la opción de copiar si lo deseara, como se puede ver en la ilustración nro. 5.



Ilustración 5 - Compartir Encuesta

Resultados de la Encuesta

En esta pantalla el usuario podrá ver los resultados de la encuesta hasta la fecha o antes de la finalización de aceptación de respuestas

Además, el usuario tendrá las opciones de exportar los resultados tanto independientemente como generalmente. Para exportar todos los resultados se presentan los botones de “**exportar pdf**” para exportar en formato pdf y “**exportar csv**” para exportar en formato csv, como también individualmente cada imagen por separado.

A continuación, una pequeña vista de la ventana de resultados mostrada en la ilustración nro. 6.

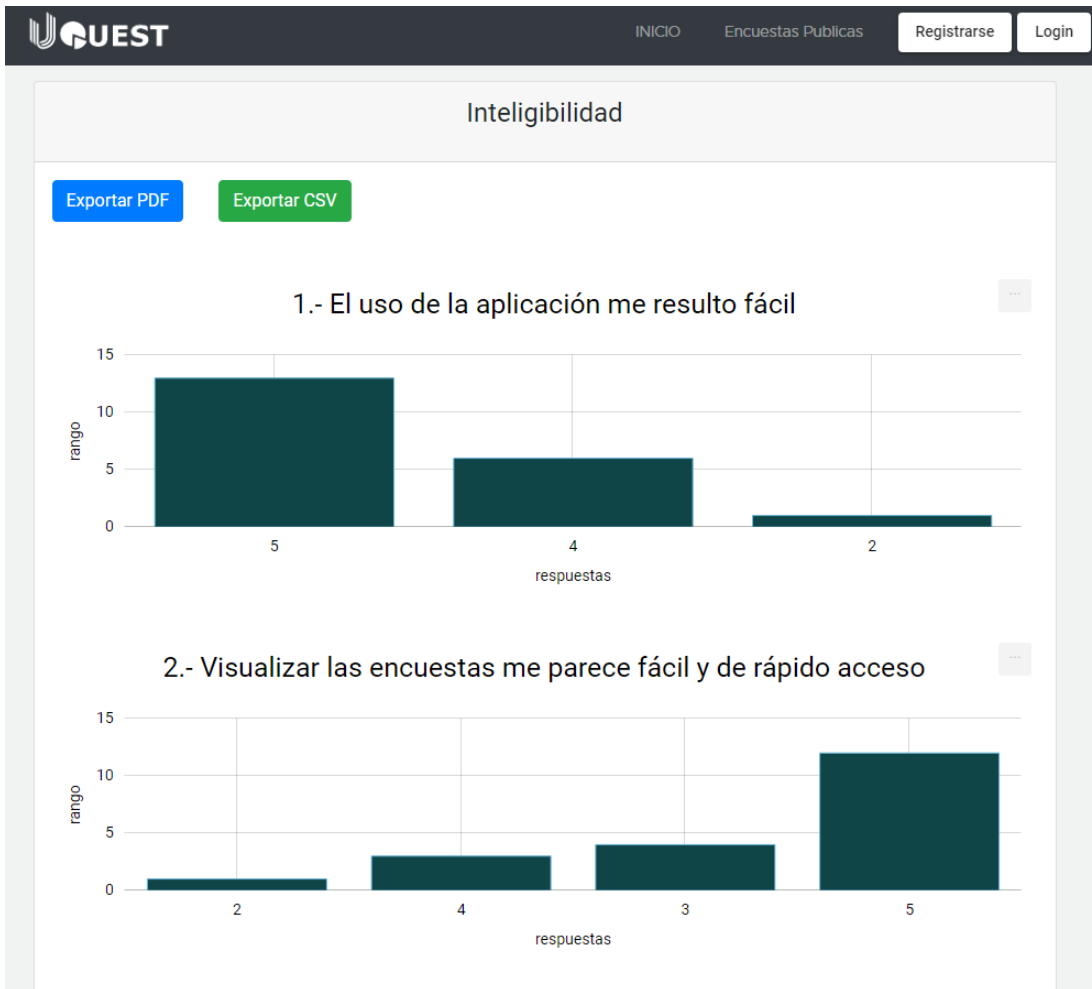


Ilustración 6 - Resultados de encuesta

La presentación de resultados puede variar según el tipo de pregunta, ya sea tipo pie o de escala.

Ventana de Encuestas propias del Usuario

Dentro de la ventana accesible al botón de la barra de navegación, “Mis encuestas” se podrá observar la ilustración nro. X

- Copiar Encuesta
- Modificar Fechas
- Finalizar
- Compartir
- Estadísticas
- Eliminar

Crear Encuesta

Cuando se accede al apartado de crear encuesta al seleccionar el botón “crear encuesta” la ilustración nro.

7 muestra un ejemplo de llenado de encuesta con los distintos campos a llenar como un usuario lo requiera.

Ilustración 7 - Crear encuesta

- Tipo: el tipo de la encuesta puede ser “Abierto” donde la encuesta estará disponible para todo el publico que acceda a la página, en segundo se tendrá el tipo “Cerrado” el cual para responder a la pregunta solo se podrá acceder mediante un enlace y además el usuario deberá estar registrado en el sistema para poder responder la encuesta.
- Categoría: la categoría de la encuesta sirve para poder identificar el objetivo de la encuesta e identificar un público en específico.
- Carreras Objetivo: las carreras objetivo son las carreras destino de la encuesta el cual se seleccionará en el apartado de encuestas publicas como ser accediendo a un link donde el usuario registrado pertenezca a dicha carrera.
- Fecha Inicio: la fecha de inicio de la encuesta
- Fecha Final: la fecha final de la encuesta, una vez que la fecha final sea menor a la actual la encuesta pasará a tener el estado de “Cerrado” donde no se podrá responder mas preguntas dentro la encuesta y solo se podrá ver resultados.

Tipos de Preguntas

- Respuesta Simple: Una respuesta simple donde la respuesta es literal y limitada a 30 caracteres.
- Párrafo: un párrafo es cuando la respuesta es más extensa que una respuesta simple y se limita a 230 caracteres.

- Opción Múltiple: la opción múltiple posee una selección de opciones donde un usuario al responder la pregunta de este tipo solo podrá seleccionar una opción de preferencia.
- Casillas de verificación: las casillas de verificación son similares a las de opción múltiple, que se diferencian en la cantidad de opciones a seleccionar, en este caso no tiene límite de selección y puede seleccionar varios a la vez.
- Escala Lineal: la escala lineal se selecciona un valor dentro el rango proporcionado por el creador de la encuesta, como por ejemplo seleccionar un valor entre 0 y 10 (límite de selección actual)

Ventana de perfil

En esta ventana el usuario puede acceder a su perfil que mostrará toda la información, para acceder a este basta con seleccionar el icono de avatar en la parte superior derecha de la barra de navegación donde se podrá observar como la ilustración nro.

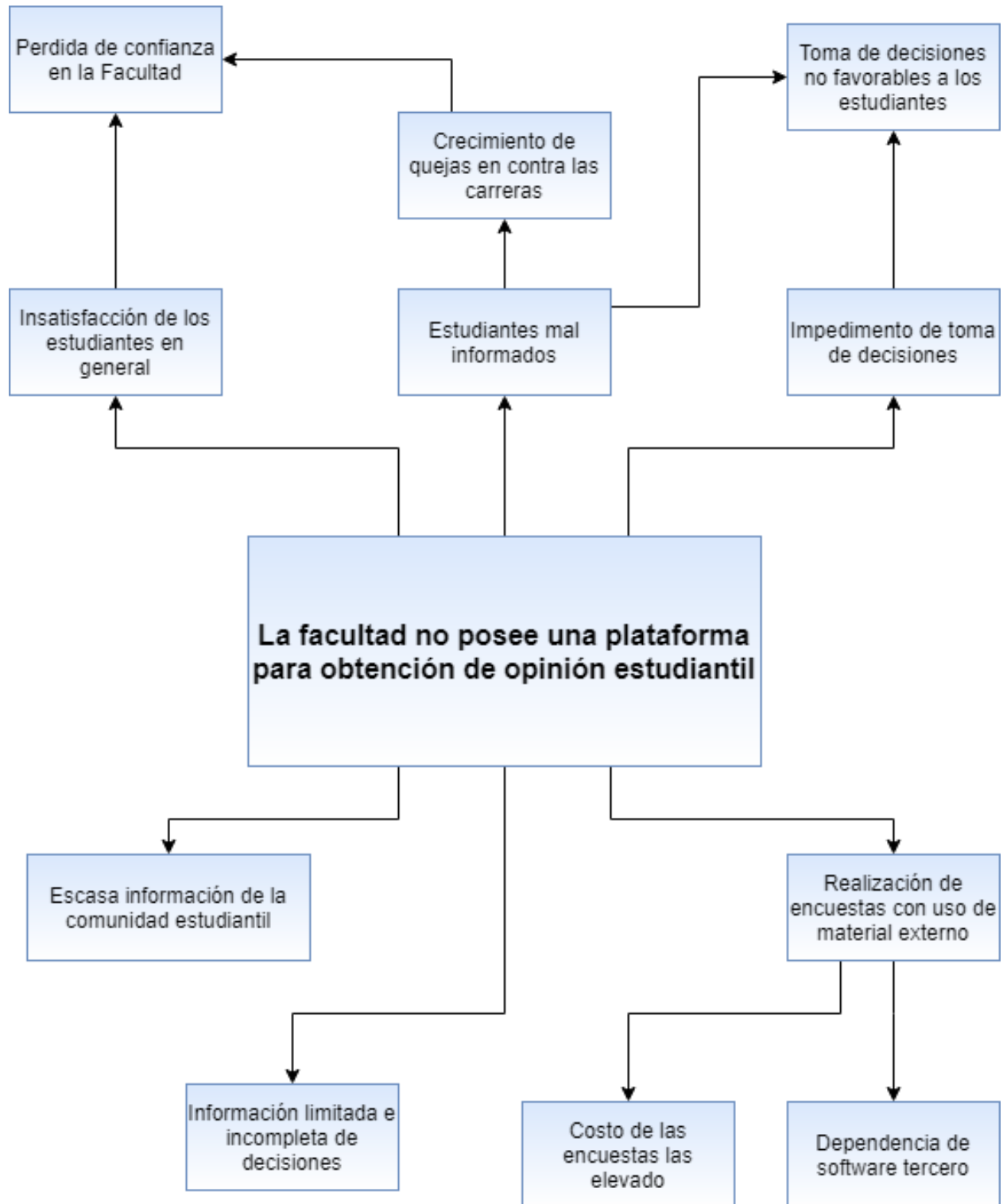
The screenshot shows the 'Información Personal' page on the UQUEST platform. At the top, there is a navigation bar with the UQUEST logo and links for 'INICIO', 'Mis Encuestas', and 'Encuestas Publicas'. Below the navigation bar, the page title is 'Información Personal' with a subtitle: 'Información básica, como el nombre, la foto que utilizas para UQUEST'. The main content area is titled 'Perfil' and includes a note: 'Es posible que otros usuarios puedan ver parte de la información'. The profile details are as follows:

FOTO:	<i>Puedes personalizar tu cuenta con una foto.</i>	
<input type="text" value="Seleccionar Foto"/> <input type="button" value="Browse"/> <input type="button" value="Subir"/>		
NOMBRES:	[Redacted]	
APELLIDO PATERNO:	[Redacted]	
APELLIDO MATERNO:	[Redacted]	
SEXO:	Masculino	
FECHA DE NACIMIENTO:	[Redacted]	
USERNAME:	[Redacted]	
CONTRASEÑA:	
EMAIL:	[Redacted]	

Ilustración 8 - Perfil usuario

El usuario podrá acceder a sus datos personales y además de tener la opción de subir una imagen con un límite de carga actual de 2MB (personalizable), además de poder editar el perfil accediendo al botón inferior derecho “editar” el cual puede cambiar los datos personales, como también la contraseña.

ANEXO C – ARBOL DE PROBLEMAS



ANEXO D – ARBOL DE OBJETIVOS

