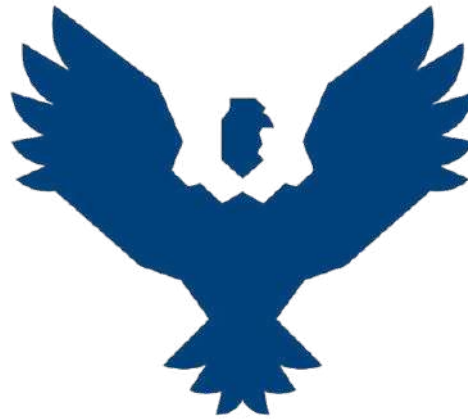




# UNIVERSIDAD ANDINA DEL CUSCO

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**TESIS**

---

**“DESARROLLO EN CLOUD COMPUTING DE UN SISTEMA DE  
INFORMACIÓN PARA EL PROCESO DE INSCRIPCIÓN AL EXAMEN DE  
ADMISIÓN DE UNA ENTIDAD EDUCATIVA SUPERIOR”**

---

Línea de investigación: Tecnologías de Información

PRESENTADO POR:

Br. Cano Alatrística Alberto

Br. Wiese Gutiérrez Esperanza

Para optar por el título Profesional de  
Ingenieros de Sistemas

ASESOR:

Ing. Ivan Molero Delgado

CUSCO – PERU

2020



### Metadatos

Datos del autor	
Nombres y apellidos	Alberto Cano Alatriza
Número de documento de identidad	46706708
Datos del autor	
Nombres y apellidos	Esperanza Wiese Gutiérrez
Número de documento de identidad	72803544
Datos del asesor	
Nombres y apellidos	Ing. Ivan Molero Delgado
Número de documento de identidad	25705893
Datos del jurado	
Presidente del jurado (jurado 1)	
Nombres y apellidos	Dr. Ing. Cristhian Eduardo Ganvini Valcarcel
Número de documento de identidad	23920560
Jurado 2	
Nombres y apellidos	Ing. Maria Isabel Acurio Gutierrez
Número de documento de identidad	29715419
Datos de la investigación	
Línea de investigación de la Escuela Profesional	Tecnologías de Información



# tesis 20200826

*by* Esperanza Wiese

---

**Submission date:** 26-Aug-2020 08:32AM (UTC-0500)

**Submission ID:** 1374344064

**File name:** T\_-\_01.docx (10.71M)

**Word count:** 23393

**Character count:** 122025



ORIGINALITY REPORT

19%

SIMILARITY INDEX

19%

INTERNET SOURCES

3%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1	<a href="http://docplayer.es">docplayer.es</a> Internet Source	3%
2	<a href="http://openaccess.uoc.edu">openaccess.uoc.edu</a> Internet Source	2%
3	<a href="http://www.hiberus.com">www.hiberus.com</a> Internet Source	1%
4	<a href="http://enmilocalfunciona.io">enmilocalfunciona.io</a> Internet Source	1%
5	<a href="http://cdn02.pucp.education">cdn02.pucp.education</a> Internet Source	1%
6	<a href="http://udesa.edu.ar">udesa.edu.ar</a> Internet Source	1%
7	<a href="http://issuu.com">issuu.com</a> Internet Source	1%
8	<a href="http://www.campusmvp.es">www.campusmvp.es</a> Internet Source	1%
9	<a href="http://videlcloud.wordpress.com">videlcloud.wordpress.com</a> Internet Source	1%

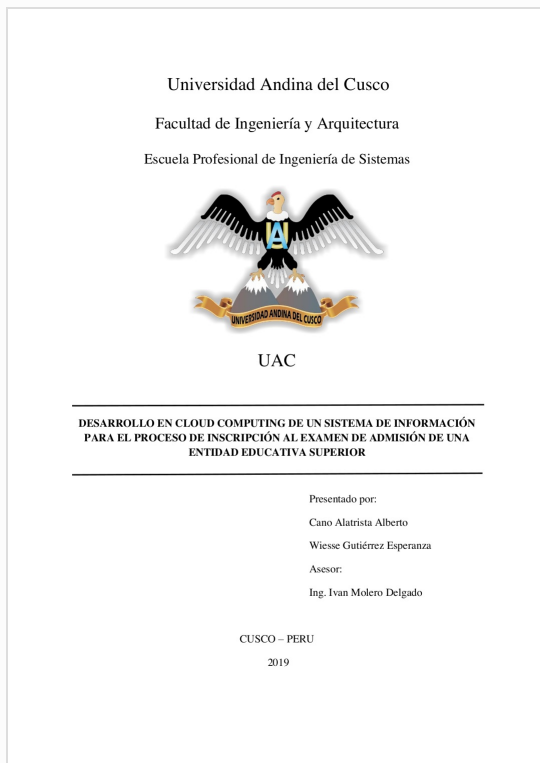


## Digital Receipt

This receipt acknowledges that **Turnitin** received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author: **Esperanza Wiese**  
Assignment title: **Tesis**  
Submission title: **tesis 20200826**  
File name: **T\_-\_01.docx**  
File size: **10.71M**  
Page count: **116**  
Word count: **23,393**  
Character count: **122,025**  
Submission date: **26-Aug-2020 08:32AM (UTC-0500)**  
Submission ID: **1374344064**





ÍNDICE	
Dedicatorias	1
Resumen	2
Abstrac	3
CAPÍTULO I	4
1. Problema de investigación	4
1.1. Ámbito de influencia	4
1.1.1. Ámbito de influencia teórica	4
1.1.2. Área de dominio	4
1.1.3. Línea de investigación	4
1.2. Planteamiento del problema	4
1.2.1. Descripción de la situación actual de lugar de intervención	4
1.2.2. Descripción del problema	5
1.2.3. Formulación del problema	6
1.2.4. Objetivos	6
1.2.5. Justificación	6
1.2.6. Alcance y limitaciones	6
CAPÍTULO II	8
2. Marco teórico	8
2.1. Antecedentes del desarrollo, implementación o transferencia tecnológica	8
2.1.1. Sistema de admisión de la PUCP	8
2.1.2. Evaluación de la reestructuración del proceso de admisión de estudiantes de la UNSAAC	10
2.1.3. Sistema de ingreso a las universidades, criterios para la admisión de alumnos. Acceso a la educación superior en universidades privadas argentinas	12



2.1.4.	El cambio en el sistema de admisión de la Universidad Nacional de Colombia: Un estudio de caso del Departamento de Sociología	13
2.2.	Base teórico – científicas	15
2.2.1.	Proceso de inscripción a un examen de admisión	15
2.2.2.	Condiciones básicas de calidad	15
2.2.3.	Indicador	16
2.2.4.	Componente	17
2.2.5.	Medio de verificación	17
2.2.6.	Open source	17
2.2.7.	MEAN Stack	18
2.2.8.	Cloud computing	21
2.2.9.	REST	27
2.2.10.	Heroku	29
2.2.11.	Mongoose	30
2.2.12.	Bcrypt	31
2.2.13.	Json web token	32
2.2.14.	JSPDF	34
2.2.15.	Usabilidad web	35
2.2.16.	Scrum	39
CAPÍTULO III		44
3.	Desarrollo, implementación o transferencia tecnológica	44
3.1.	Desarrollo	44
3.1.1.	Planteamiento de los sprints	44
3.1.2.	Producto backlog	44
3.1.3.	Diagrama de la Base de Datos	47
3.1.4.	Sprint 1	47
3.1.5.	Sprint 2	52



3.1.6. Sprint 3	59
3.1.7. Sprint 4	66
3.1.8. Sprint 5	72
3.2. Pruebas de usabilidad	76
3.3. Indicador 6 del componente 1.5	80
3.4. Costo de desarrollo de la aplicación	81
CAPÍTULO IV	83
4. Resultados	83
4.1. Comprobación de la usabilidad del sistema	83
4.1.1. Pruebas de usabilidad al sistema	83
4.2. Comprobación de la prospectiva	83
4.3. Cumplimiento de objetivos	84
4.4. Contribuciones (impacto)	92
Glosario	93
Conclusiones	94
Recomendaciones	95
Referencias	96
ANEXOS	98
ANEXO 1	98
ANEXO 2	104
ANEXO 3	113
ANEXO 4	119
ANEXO 5	125
ANEXO 6	127
ANEXO 7	129
ANEXO 8	130
ANEXO 9	145





## ÍNDICE DE TABLAS

TABLA 1 VENTAJAS Y DESVENTAJAS DEL PAAS .....	27
TABLA 2 PRINCIPIOS DE USABILIDAD .....	38
TABLA 3 HISTORIAS DE USUARIO .....	46
TABLA 4 HISTORIAS DE USUARIO .....	48
TABLA 5 HISTORIA DE USUARIOS .....	53
TABLA 6 HISTORIAS DE USUARIO .....	60
TABLA 7 HISTORIAS DE USUARIO .....	67
TABLA 8 HISTORIA DE USUARIO .....	73
TABLA 9 CUADRO DE USABILIDAD DE LA PREINSCRIPCIÓN.....	77
TABLA 10 CUADRO DE USABILIDAD DE LA PREINSCRIPCIÓN.....	77
TABLA 11 CUADRO DE USABILIDAD DE LA INSCRIPCIÓN .....	78
TABLA 12 CUADRO DE USABILIDAD DE LA INSCRIPCIÓN .....	79
TABLA 13 PROMEDIO DE LAS PRUEBAS DE USABILIDAD .....	79
TABLA 14 COSTOS DE DESARROLLO DE LA APLICACIÓN .....	82



## ÍNDICE DE FIGURAS

FIGURA 1 CONDICIONES BÁSICAS DE CALIDAD.....	16
FIGURA 2 CONDICIÓN I.....	16
FIGURA 3 CUADRO DE LA CONDICIÓN I.....	17
FIGURA 4 ESQUEMA DE FUNCIONAMIENTO DE MONGODB.....	19
FIGURA 5 SERVICIOS PRESTADOS POR LOS TIPOS DE NUBES.....	25
FIGURA 6 DEFINICIÓN DE UN ESQUEMA.....	30
FIGURA 7 DOCUMENTO JSON.....	30
FIGURA 8 DEFINICIÓN DEL ESQUEMA.....	31
FIGURA 9 CREACIÓN DEL MODELO.....	31
FIGURA 10 JWT.....	33
FIGURA 11 PROCESO DEL JWT.....	34
FIGURA 12 DIAGRAMA DE BASE DE DATOS.....	47
FIGURA 13 CHECK LIST.....	50
FIGURA 14 CHECK LIST.....	50
FIGURA 15 BURN DOWN CHARD.....	51
FIGURA 16 CAPTURA DE PANTALLA DEL MANTENIMIENTO DE USUARIOS DEL SISTEMA.....	52
FIGURA 17 CAPTURA DE PANTALLA DE LA CREACIÓN DE UN USUARIO .....	52
FIGURA 18 CHECK LIST.....	54
FIGURA 19 CHECK LIST.....	55
FIGURA 20 CHECK LIST.....	55
FIGURA 21 CHECK LIST.....	55
FIGURA 22 CHECK LIST.....	56
FIGURA 23 CHECK LIST.....	56
FIGURA 24 BURN DOWN CHART.....	57
FIGURA 25 MANTENIMIENTO DE PROCESOS.....	58
FIGURA 26 CREACIÓN DE UN PROCESO.....	58
FIGURA 27 MANTENIMIENTO DE LAS MODALIDADES.....	58
FIGURA 28 CREACIÓN DE UNA MODALIDAD.....	59
FIGURA 29 MANTENIMIENTO DE LAS CARRERAS.....	59



FIGURA 30 CREACIÓN DE UNA CARRERA .....	59
FIGURA 31 CHECK LIST.....	61
FIGURA 32 CHECK LIST.....	61
FIGURA 33 CHECK LIST.....	62
FIGURA 34 CHECK LIST.....	62
FIGURA 35 BURN DOWN CHARD.....	63
FIGURA 36 INTERFAZ DE INFORMACIÓN PARA EL POSTULANTE.....	64
FIGURA 37 FORMULARIO DE PREINSCRIPCIÓN.....	65
FIGURA 38 VALIDACIÓN DE LOS DATOS DE LA PREINSCRIPCIÓN .....	66
FIGURA 39 CHECK LIST.....	68
FIGURA 40 CHECK LIST.....	69
FIGURA 41 CHECK LIST.....	69
FIGURA 42 CHECK LIST.....	69
FIGURA 43 CHECK LIST.....	70
FIGURA 44 CHECK LIST.....	70
FIGURA 45 CHECK LIST.....	70
FIGURA 46 BURN DOWN CHARD.....	71
FIGURA 47 REGISTRO DE REQUISITOS .....	72
FIGURA 48 FORMATO DE INSCRIPCIÓN.....	72
FIGURA 49 CHECK LIST.....	74
FIGURA 50 CHECK LIST.....	74
FIGURA 51 BURN DOWN CHARD.....	75
FIGURA 52 REPORTE DE N° DE INSCRITOS .....	76
FIGURA 53 CLIENTE/SERVIDOR SIN ESTADO.....	86
FIGURA 54 PETICIONES HTTP.....	87
FIGURA 55 PETICIONES HTTP.....	88
FIGURA 56 DEFINICIÓN DEL PROTOCOLO.....	89
FIGURA 57 DOMINIO.....	89
FIGURA 58 RUTA.....	89
FIGURA 59 CONSULTA .....	90
FIGURA 60 FLUJO DEL DESARROLLO DEL SISTEMA.....	91



## Dedicatorias

*“A mis papas y mi hermana por siempre confiar en mí, y porque han estado presentes en este viaje conmigo, brindándome su apoyo incondicional en cada etapa.”*

Esperanza Wiese Gutiérrez.

*“A mi madre por ser el pilar de mi vida, a mi padre por el esfuerzo, cariño y comprensión y a mis hermanos por el apoyo que me brindaron en todo momento.”*

Alberto Cano Alatrasta.



## Resumen

En este trabajo de investigación se desarrolló un Sistema de Inscripción en Cloud Computing para el Registro de Postulantes a un Examen de Admisión de una entidad educativa superior, ya que en la actualidad existen entidad educativa superior que no cuenta con un Sistema de Información automatizado para el Proceso de Inscripción a los Exámenes de Admisión. Se pretende solucionar los siguientes problemas:

- No cuentan con un sistema web de inscripciones.
- La inscripción de los postulantes se realiza de manera manual.
- No se lleva un registro de los documentos entregados por el postulante.

En esta investigación se plantean los siguientes objetivos:

- a) Usar SCRUM como framework de gestión de proyecto.
- b) Evaluar la usabilidad del sistema de información desarrollado de acuerdo con los 10 principios de Nielsen.
- c) Generar información de apoyo para el cumplimiento de las Condiciones Básicas de Calidad de SUNEDU.

Para el desarrollo del Software se hizo uso del stack MEAN se decidió el uso de estas tecnologías porque todas tienen como base para su funcionamiento JavaScript; y para el despliegue en la nube Heroku, que es una empresa que se especializa en ofrecer servicios de plataforma administrada, por sus siglas en inglés PasS.



## Abstrac

In this research work, an Enrollment System Based on Cloud Computing was developed for the Registration of Applicants to a University Admission Exam, since currently there are universities that do not have an automated Information System for the Enrollment Process to the Admission Exams. It is intended to solve the following problems:

- They do not have a web registration system.
- The registration of applicants is done manually.
- There is no record of the documents delivered by the applicant.

In this research the following objectives are proposed:

- a) Use SCRUM as a project management methodology.
- b) Evaluate the usability of the information system developed in accordance with the 10 Nielsen principles.
- c) Generate information to support compliance with the Basic Quality Conditions of SUNEDU.

For the development of the Software, the MEAN stack was used, it was decided to use these technologies because they all have JavaScript as the basis for their operation; and for the deployment in the cloud Heroku, which is a company that specializes in offering managed platform services, for its acronym in English PasS.



## CAPÍTULO I

### 1. Problema de investigación

#### 1.1. Ámbito de influencia

##### 1.1.1. Ámbito de influencia teórica

El presente proyecto permite la automatización del Proceso de Inscripción a los Exámenes de Admisión de una entidad educativa superior, permitiendo acortar los tiempos de inscripción, la información recabada está sujeta a menos errores humanos, el tiempo de capacitación del personal es más corto. Por consiguiente, los beneficiados de la automatización del Proceso de Inscripción a los Exámenes de Admisión son los trabajadores de la Oficina de Admisión y los postulantes.

##### 1.1.2. Área de dominio

El área de dominio del proyecto de investigación “DESARROLLO EN CLOUD COMPUTING DE UN SISTEMA DE INFORMACIÓN PARA EL PROCESO DE INSCRIPCIÓN AL EXAMEN DE ADMISIÓN DE UNA ENTIDAD EDUCATIVA SUPERIOR” es el de Tecnologías de Información.

##### 1.1.3. Línea de investigación

La línea de investigación del proyecto de investigación “DESARROLLO EN CLOUD COMPUTING DE UN SISTEMA DE INFORMACIÓN PARA EL PROCESO DE INSCRIPCIÓN AL EXAMEN DE ADMISIÓN DE UNA ENTIDAD EDUCATIVA SUPERIOR” es el de Desarrollo de Software, para la automatización de procesos.

### 1.2. Planteamiento del problema

#### 1.2.1. Descripción de la situación actual de lugar de intervención

Algunas entidades educativas superiores, actualmente en funcionamiento, requieren de un sistema de Proceso de Inscripción a los Exámenes de Admisión razón por la cual se desarrolló este software de uso general. Esto en vista que sus procesos de registro



de inscripción de postulantes se vienen llevando de manera manual, lo cual no explota el uso de tecnologías actuales de manera adecuada.

### 1.2.2. Descripción del problema

En la actualidad existen entidades educativas superiores que no cuenta con un Sistema de Información automatizado para el Proceso de Inscripción a los Exámenes de Admisión, el Proceso de Inscripción se lleva acabo de manera manual siguiendo los siguientes pasos en la mayoría de los casos:

- a) Registro del número de la transacción del pago que realizo el postulante en el archivo de Excel
- b) Registro de los datos del postulante en el archivo de Excel
- c) Recepción de los requisitos presentados por el postulante
- d) Registro de las carreras elegidas por el postulante en el archivo de Excel
- e) Recepción de dos fotos tamaño carnet del postulante para la elaboración del carnet

Nosotros hemos realizado nuestras practicas pre-profesionales en la Universidad Andina del Cusco en el Área de admisión, lo que nos ha permitido imbuirnos del proceso de admisión en general, y hemos detectado que hay acciones que se realizan de manualmente y que podrían ser automatizadas, ya que no satisfacen los requerimientos de los usuarios.

1. Las universidades que estaban en proceso de acreditación no contaban con un sistema de inscripción automatizado.
2. No contaban con un registro automatizado de los requisitos entregados por el postulante.
3. La no entrega de una constancia que resuma los datos de inscripción del postulante.





### 1.2.3. Formulación del problema

¿Cómo se desarrolla un sistema de información en Cloud Computing para el Proceso de Inscripción a los Exámenes de Admisión de una entidad educativa superior?

### 1.2.4. Objetivos

#### 1.2.4.1. General

Desarrollar un Sistema de Información para el Proceso de Inscripción a los Exámenes de Admisión de una entidad educativa superior en Cloud Computing.

#### 1.2.4.2. Específicos

- a) Usar SCRUM como framework de gestión de proyecto para el desarrollo de un Sistema de Información para el Proceso de Inscripción a los Exámenes de Admisión de una entidad educativa superior.
- b) Evaluar la usabilidad del sistema de información desarrollado de acuerdo con los 10 principios de Nielsen.
- c) Generar información de apoyo al cumplimiento y sustentación del indicador 6 del componente 1.5 referido al Proceso de Admisión de las Condiciones Básicas de Calidad.
- d) Plantear un esquema de desarrollo del sistema.

### 1.2.5. Justificación

No todas las entidades educativas superiores cuentan con un Sistema de Información para el Proceso de Inscripción a los Exámenes de Admisión, por tanto, es difícil de gestionar y procesar la información, con el sistema que se propone se mejoraran los dos aspectos antes mencionados.

### 1.2.6. Alcance y limitaciones

#### 1.2.6.1. Alcance

El Sistema de Información para la Oficina de admisión solo contempla el Proceso de Inscripción al Examen de Admisión, el sistema cuenta con una Pre-Inscripción que realiza el



postulante, una Inscripción que lo realiza el personal de la Oficina de Admisión, el registro de la documentación entregada por el postulante para su Inscripción, reportes sobre el Proceso de Inscripción al Examen de Admisión.

#### **1.2.6.2. Limitaciones**

El Sistema de Información está desarrollado específicamente para la Oficina de Admisión y solo comprende el Proceso de Inscripción a los Exámenes de Admisión, por consiguiente, el sistema no gestionará la información del resto de procesos de un Examen de Admisión.



## CAPÍTULO II

### 2. Marco teórico

#### 2.1. Antecedentes del desarrollo, implementación o transferencia tecnológica

Es necesario precisar que el contenido en cada uno de los antecedentes es un resumen de las publicaciones mencionadas, para lo cual remito los links correspondientes en cada caso.

##### 2.1.1. Sistema de admisión de la PUCP

###### Resumen:

En junio del 2016 la oficina Central de Admisión e Informes de la Pontificia Universidad Católica del Perú realizó un estudio denominado Sistema de Admisión de la PUCP con la finalidad de mejorar su sistema de evaluación de admisión los objetivos planteados en este trabajo fueron:

- a) Contar con un perfil de ingreso
- b) Contar con un módulo e instrumentos de medición eficaces que permitan, además de ordenar al grupo para asignar vacantes, valorar la calidad académica de los admitidos y comparar a quienes postulan por distintos canales e ingresan al mismo ciclo o comparar entre los admitidos en diferentes años.
- c) Disponer de un sistema de retroinformación que proporcione información significativa a los postulantes admitidos y no admitidos, a las unidades académicas y a los colegios.
- d) Integrar a las unidades académicas que reciben alumnos en el primer ciclo del pregrado en el proceso de consulta y toma de decisiones sobre la admisión a la Universidad Católica.

Para poder lograr los objetivos plantearon una metodología de trabajo y proceso de consulta a expertos, esta metodología tuvo las siguientes etapas:

- Etapa 1: Revisión bibliográfica y asesoría de expertos (julio 2006 – marzo 2008).



- Etapa 2: Estudios de validez predictiva (julio de 2007 – abril de 2008).
- Etapa 3: Investigación sobre número de alternativas en preguntas objetivas (julio 2007).
- Etapa 4: Entrevistas grupales (marzo – abril de 2008).
- Etapa 5: Sistematización del perfil (mayo – julio de 2008).
- Etapa 6: Sistematización del sistema de admisión (mayo - julio de 2008).
- Etapa 7: Piloto de construcción y validación de preguntas para el banco. (julio – setiembre 2008).
- Etapa 8: Consulta a autoridades de las unidades académicas (agosto de 2008).
- Etapa 9: Consulta a la Dirección de Asuntos Académicos (agosto de 2008).
- Etapa 10: Estudios con postulantes (agosto –setiembre de 2008).
- Etapa 11: Elaboración de la propuesta para el Vicerrectorado Académico (setiembre de 2008).
- Etapa 12: Presentación del documento al Vicerrectorado Académico (octubre de 2008).
- Etapa 13: Validación de indicadores para calificar redacciones.

En este estudio se determinó en primer lugar el perfil del ingresante, con el fin de que el ingresante tenga un buen desempeño en su vida universitaria además de que las probabilidades de deserción del postulante sean bajas, en segundo lugar se terminó que aspectos del perfil del ingresante corresponden a la evaluación de admisión y que otras características que poseen los postulantes corresponden a servicio social de la universidad, en tercer lugar se investigó cual es el método de evolución más adecuado y los instrumentos de evaluación que se utilizarían. ( Rivera Scaramutti, Iza Rotta, Torreblanca Villavicencio, Tavera Peña, & Borda Pinedo, 2016).



### 2.1.2. Evaluación de la reestructuración del proceso de admisión de estudiantes de la UNSAAC

#### Resumen:

La Comisión de Admisión nombrada para el período 1982-1984, identificó el problema del uso continuo del “Banco de Preguntas”, preferentemente por los postulantes a la Carrera Profesional de Medicina Humana, lo conllevó, a que con el tiempo, estos postulantes, memorizaran las preguntas con la alternativa de solución correspondientes indicadas en el “Banco de Preguntas” (Al estudiante solo le bastaba escuchar o leer la parte inicial de la pregunta, e inmediatamente asociaban con la clave correspondiente, sin necesidad de desarrollar el tema)

Por estas razones, se autoriza el 10 de octubre del 1994, mediante Resolución VRAC N° 109-94 del Consejo Universitario, para que la Comisión Permanente de Admisión de Estudiantes a la UNSAAC, en coordinación con los Directivos del CPU, se deje de utilizar el “Banco de Preguntas”, así mismo proponga nuevas modalidades de examen de admisión.

La comisión planteó tres modalidades de ingreso, que detallamos para referencia:

PRIMERA PROPUESTA.- Establecimiento del CICLO PROPEDEUTICO, que consistía en que los postulantes cursarían dos semestres académicos no universitarios, durante el Primer Semestre desarrollarían los estudiantes 8 asignaturas básicas, y quienes aprobaran en forma invicta, estarían habilitados para continuar sus estudios en el Segundo Semestre, en el que también se consideraban 4 asignaturas generales y 4 orientadas a las Carreras Profesionales que desean postular, quienes aprobaran en calidad de invictos, tendrían la oportunidad de acceder a la Universidad.



Propuesta desechada, por resistencia a su implementación por los Decanos de las Carreras Profesionales de Medicina Humana e Ingeniería Civil, fundamentalmente.

SEGUNDA PROPUESTA. - Reimplantar con ciertas modificaciones los Estudios Generales, previo examen de Admisión basada en la evaluación de aptitudes matemáticas, aptitudes lingüísticas y cultura general. Propuesta no considerada por la Comisión Académica de la UNSAAC.

TERCERA PROPUESTA. - Agrupación de las Carreras profesionales de acuerdo a criterios de afinidad en el rendimiento académico durante el proceso de admisión en determinadas asignaturas. Propuesta aceptada por la Comisión Académica de la UNSAAC y aprobada por el Consejo Universitario, para que entre en vigencia desde el Primer Examen Ordinario del Año 1995.

La propuesta inicial consideraba los grupos de Carreras Profesionales denominadas: INGENIERÍAS, CIENCIAS BILÓGICAS, LETRAS, CIENCIAS DE SERVICIO SOCIAL Y CIENCIAS CONTABLES Y FINANCIERAS. Esta clasificación se consideró en base al record histórico de las asignaturas que mejor rendimiento hayan tenido en los anteriores procesos de admisión. Luego de una primera sustentación, ante la Comisión Académica de la UNSAAC, se replanteó la propuesta estableciendo códigos para las diferentes opciones propuestas:

- GRUPO "A": Arquitectura, Ing. Eléctrica, Ing. Geológica, Ing. Minas, Ing. Química, Ing. Civil, Ciencias Físicas, Químicas y Matemáticas e Ing. Agroindustrial (Sicuani).
- GRUPO "B": Agronomía, Biología, Enfermería, Medicina Humana, Zootecnia, Agronomía Tropical (Quillabamba) e Ing. Forestal (Puerto Maldonado).
- GRUPO "C": Ciencias Administrativas, Contabilidad, Economía y Turismo.



- GRUPO “D”: Antropología, Arqueología, Derecho y Ciencias Políticas, Historia, Ciencias de la Comunicación y Educación.

Las modalidades de ingreso de estudiantes a la UNSAAC, tenía tres variantes: El Examen Ordinario y el Examen de Dirimencia y vía CPU. (Wiesse Mujica , 1994)

### **2.1.3. Sistema de ingreso a las universidades, criterios para la admisión de alumnos. Acceso a la educación superior en universidades privadas argentinas**

#### **Resumen:**

En abril del 2012 se realizó una investigación denominada Sistema de Ingreso a las Universidades, Criterios para la Admisión de Alumnos. Acceso a la Educación Superior en Universidades Privadas Argentinas.

En la primera capítulo, se realizó un análisis de como la educación universitaria ha ido cambiando con el paso de los años y la influencia de cada gobierno en el modo de ingreso y enseñanza en las universidades.

En el segundo capítulo, se hace una comparación de las distintas modalidades de ingreso que existe en los países de: Inglaterra, España, Alemania, Francia y Estados Unidos.

En el tercer capítulo se realizó una descripción del sistema de admisión de las universidades privadas argentinas, para este estudio se tomó como referencia las universidades: Ciudad Autónoma de Buenos Aires y Gran Buenos Aires ya que son las universidades más representativas del sector privado. Para esto se analizó las condiciones o requisitos que deben cumplir y una clasificación de sistemas de admisión según los términos elaborados durante el 2004, son los siguientes:

- a) Admisión por competencia: dentro de la institución se establecen cupos y se otorgan las plazas a quienes, en orden de mérito del conjunto de postulantes, cumplen con los



requisitos establecidos (exámenes de selección y/u otros requisitos)

- b) Admisión por calificación: se fijan los requisitos mínimos considerados como los necesarios para que el postulante acceda a la condición de alumno de la institución, es decir, Se establecen requisitos de ingreso y quien los posee es admitido como alumno. En esta clasificación la universidad es quien establece el umbral que considera válido para certificar las aptitudes para iniciar la carrera de grado universitario.
- c) Admisión libre: no se establecen requisitos adicionales a la sola condición de haber finalizado el nivel medio. En algunos casos se pueden ofrecer cursos de nivelación de asistencia obligatoria con examen no eliminatorio o bien, sin examen.

En el cuarto capítulo, se mencionan algunas políticas educativas implementadas en Argentina, orientadas a que se disminuya la brecha entre la educación secundaria y universitaria y.

En el quinto capítulo, se estudia el caso de la Universidad Argentina de la Empresa, con el fin de saber si se acepta o rechaza la hipótesis de Macedo: “relación entre el rendimiento académico y el rendimiento en pruebas que se desarrollen en el momento de la admisión” (Macedo , 2012, pág. 12).

#### **2.1.4. El cambio en el sistema de admisión de la Universidad**

##### **Nacional de Colombia: Un estudio de caso del Departamento de Sociología**

###### **Resumen:**

En esta investigación realizada el 2014, por el Departamento de Sociología de la Universidad Nacional de Colombia, se analiza el cambio realizado en el sistema de admisión, en este nuevo sistema de admisión los aspirantes rinden un examen general y se les coloca en una escala del 1 al 7 según la nota que han logrado en el examen, los postulantes escogen las carreras según al orden de méritos en el que encuentren en la escala, la investigación tiene





como objetivo analizar los cambios que han comenzado a haber en el tipo de postulantes que reciben, los objetivos específicos de esta investigación son:

- a) Identificar los cambios que se han realizado en el nuevo sistema de admisión.
- b) Identificar la forma en como ha caminado el perfil de los postulantes con el nuevo sistema de admisión.
- c) Describir que tipo de expectativas tienen los postulantes de la carrera de sociología con el nuevo sistema de admisión, con respecto a las expectativas de los postulantes con el anterior sistema de admisión.

La investigación se realizó solo con alumnos de la carrera de sociología, estos alumnos estaban cursando el segundo y tercer semestre, se escogieron estos 2 grupos de estudios porque los alumnos de segundo semestre ingresaron con el nuevo sistema de admisión y los de tercer semestre con el antiguo sistema de admisión.

Las conclusiones a las que llegaron los investigadores son las siguientes:

- a) Disminuyo el nivel de conocimiento de los postulantes acerca de que trata cada carrera.
- b) Debido a lo anterior se encontró un alto nivel de descontento de los alumnos ya que la carrera no era los que esperaban los alumnos y eso genero un aumento en la tasa de deserción.
- c) También concluyeron que es necesario realizar un estudio más a fondo, con una muestra más grande y con acceso a la base de datos de la Universidad Nacional de Colombia.

El resumen de este trabajo de investigación fue elaborado en base a (Rubio Fonseca, Camacho Escamilla, Marín Ariza, & Gómez, 2014).



En los antecedentes que se ha mostrado se trata fundamentalmente la parte académica del sistema de admisión a las universidades y no del accionar de personal administrativo a través del trámite documentario, que creemos que es fundamental y muestra investigación va en ese sentido.

## **2.2. Base teórico – científicas**

### **2.2.1. Proceso de inscripción a un examen de admisión**

Este resumen se ha hecho en concordancia al artículo 98 de la Ley Universitaria N° 30220, que regula el proceso de admisión, del cual es parte el proceso de inscripción a un examen de admisión.

El proceso de inscripción a un examen de admisión, es el accionar de los administrativos a través de la tramitología, que es un paso inicial del proceso de admisión, es decir, el acto de inscribir a un postulante al examen de admisión con el propósito de tener el status de estudiante universitario, con la finalidad de adquirir una profesión; este acto administrativo está conformado por un conjunto de faces que el postulante tiene que completar, ya se les entrega documentos en forma física o virtual y brindando información en un plazo establecido, además de un pago por derechos que les permite rendir el Examen de Admisión y para someterse a un examen, que tiene diferentes modalidades en el sistema universitario, todo esto dentro de lo normado por la Ley Universitaria con las características propias de cada universidad. Resumen de los prospectos de las universidades Universidad San Antonio Abad del Cusco y Universidad Andina del Cusco (Admisión, Prospecto de Admisión de la UNSAAC, 2019) y (Admisión, Prospecto de Admisión de la UAC, 2019).

### **2.2.2. Condiciones básicas de calidad**

Según SUNEDU (SUNEDU, 2015), es un conjunto de estándares que establecen los requisitos mínimos para el proceso de evaluación de la calidad de las Universidades, respecto a la



prestación del servicio de educación superior y la autorización de funcionamiento.

La SUNEDU plantea los siguientes puntos como las condiciones básicas para un servicio de calidad:

CONDICIONES BÁSICAS DE CALIDAD

Condiciones	N° de Indicadores
Condición I. Existencia de objetivos académicos, grados y títulos a otorgar, y planes de estudios correspondientes.	8
Condición II. Oferta educativa a crearse compatible con los fines propuestos en los instrumentos de planeamiento.	7
Condición III. Infraestructura y equipamiento adecuado al cumplimiento de sus funciones (aulas, bibliotecas, laboratorios, entre otros).	15
Condición IV. Líneas de investigación a ser desarrolladas.	8
Condición V. Verificación de la disponibilidad de personal docente calificado con no menos de 25% de docentes a tiempo completo.	4
Condición VI. Verificación de los servicios educacionales complementarios básicos (servicio médico, social, psicopedagógico, deportivo, entre otros).	8
Condición VII. Existencia de mecanismos de mediación e inserción laboral (Bolsa de Trabajo u otros).	4
Condición VIII. CBC Complementaria: transparencia de universidades.	1
Total	55

Figura 1 condiciones básicas de calidad  
(SUNEDU, 2015)

### 2.2.3. Indicador

De acuerdo con el diccionario de Google (Google, 2020), un indicador es un dato o información que sirve para conocer o valorar las características y la intensidad de un hecho o para determinar su evolución futura.

El indicador en que se centra el presente trabajo es el siguiente:

**Condición I:**

**Existencia de objetivos académicos, grados y títulos a otorgar, y planes de estudios correspondientes**

La acción educativa requiere de objetivos claros por parte de los diversos actores de la comunidad universitaria. Estos deben formar parte de los planes de estudio. Asimismo, los programas académicos deben hacer explícito el procedimiento administrativo y las condiciones necesarias que todo estudiante debe cumplir desde su admisión hasta su graduación. Por lo tanto, la universidad debe prever que sus programas tengan objetivos, exista una jerarquía respecto a los objetivos institucionales y guarden coherencia entre ellos.

Figura 2 condición I  
(SUNEDU, 2015)



### 2.2.4. Componente

Un componente es aquello que forma parte de un todo, se trata de elementos que, a través de algún tipo de asociación o contigüidad, dan lugar a un conjunto uniforme. Entonces en el contexto del proceso de acreditación de las universidades, es un aspecto o elemento de proceso de evaluación que ayuda a determinar el nivel de calidad del servicio que brinda una universidad.

El componente a tratar en este trabajo es el referido al proceso de admisión.

### 2.2.5. Medio de verificación

Es el instrumento a través del cual se acredita el cumplimiento de los indicadores del objetivo de gestión

Condición I. Existencia de objetivos académicos, grados y títulos a otorgar, y planes de estudio correspondientes.		
Componente	Indicador	Medio de verificación
1.5 Procesos de admisión	5. Existencia de un documento normativo que regule los procesos de admisión.	Normatividad o Reglamento de Admisión aprobado por la autoridad competente de la universidad, que regule las modalidades de ingreso para todos los programas de estudios, en el que se indique su última fecha de actualización.
	6. La universidad cuenta con información sobre los procesos de admisión y los ingresantes según modalidades de ingreso por periodo académico.	Informe estadístico de admisión de los últimos 2 años, según corresponda (lo cual aplica para universidades existentes antes de la Ley N° 30220).

Figura 3 cuadro de la condición I

(SUNEDU, 2015)

### 2.2.6. Open source

Según (RedHat, 2019) “El software open source es un código diseñado de manera que sea accesible al público: todos pueden ver, modificar y distribuir el código de la forma que consideren conveniente.”

Los valores del open source:

- Revisión entre compañeros
- Transparencia
- Confiabilidad



- Flexibilidad
- Menor costo
- Sin dependencia de un solo proveedor
- Colaboración abierta

### 2.2.7. MEAN Stack

El MEAN STACK como indica (Sayago Heredia & Chango Sailema, 2018, pág. 86) “Comprende cuatro bloques principales: MongoDB como base de datos, Express como marco de trabajo del servidor web, Angular como marco de trabajo del cliente web y Node.js como plataforma del servidor”.

#### 2.2.7.1. MongoDB

MongoDB es una base de datos open-source desarrollada por 10gen en C++, fue lanzada en el año 2009 y pertenece a la categoría de base de datos documentales, esta base de datos surge como una nueva tendencia en el desarrollo de base datos y se refieren en general, a las bases de datos sin un esquema fijo. Además, tiene una seguridad de transacción a un nivel más bajo, pero es más rápido en el acceso a los datos y escala mejor que las bases de datos relacionales (Vele Zhingri, 2015, pág. 47).

MongoDB almacena los datos en formato JSON. En MongoDB los documentos JSON se agrupan en Colecciones (las colecciones son los equivalentes a las relaciones en un gestor de base de datos relacional), los documentos JSON de una colección no necesariamente tienen la misma estructura (Moreno Arboleda, Quintero Rendón, & Rueda Vásquez, 2016). En el siguiente diagrama se muestra la estructura de una base de datos de MongoDB.

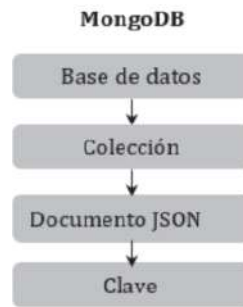


Figura 4 esquema de funcionamiento de MongoDB

(Moreno Arboleda, Quintero Rendón, & Rueda Vásquez, 2016)

### 2.2.7.2. Express

Express es un framework web maduro y flexible para construir aplicaciones web sobre el ecosistema Node.js. Por defecto, el framework Express utiliza el motor Pug para soportar plantillas. Express es un framework relativamente pequeño que se encuentra en la parte superior de la funcionalidad del servidor web de Node.js para simplificar sus APIs y añadir nuevas funciones útiles. Facilita la organización de la funcionalidad de su aplicación con middleware y enrutamiento; agrega utilidades útiles a los objetos HTTP de Node.js y el renderizado de vistas HTML dinámicas; define un estándar de extensibilidad fácilmente implementado. Express es fácil de configurar, implementar, controlar y proporcionar varios componentes clave para manejar las solicitudes web. Express ayuda en la creación de aplicaciones web y servidores HTTP simples ya que es un framework mínimo y flexible. En MEAN, Express funciona como un medio para transferir las solicitudes de un cliente a una base de datos y envía las respuestas de la base de datos al cliente (Sayago Heredia & Chango Sailema, 2018).

### 2.2.7.3. Angular

El siguiente concepto está basado en los artículos webs de (campusMVP, 2017) y (Cardenas, 2019); Angular usa la jerarquía de componentes, adoptando el cambio de enfoque de MVC (Modelo Vista Controlador) al uso de componentes



y directivas junto a las platillas, por tanto, se enfoca en el desarrollo de códigos más estructurados, modular y reutilizables. Angular ha sido desarrollado teniendo en cuenta la arquitectura para aplicaciones móviles, existen bibliotecas que ayudan a un desarrollo mucho más rápido orientado a aplicaciones móviles, además de renderizar el mismo código de manera distinta en navegadores web y para aplicaciones móviles.

“Angular ofrece la posibilidad de elegir entre varios lenguajes a la hora de programar. Para escribir código en Angular, puedes usar cualquiera de los lenguajes: ECMAScript 5, ES6, TypeScript o incluso Dart (de Google)” (campusMVP, 2017), siendo TypeScript el lenguaje por defecto utilizado para desarrollar en Angular.

Un componente en Angular es una porción de código que es posible reutilizar en otros proyectos de Angular sin apenas esfuerzo, lo que permite un desarrollo de aplicaciones mucho más ágiles. El diseño de Angular adopta el estándar de los componentes web. Se trata de un conjunto de APIs que te permiten crear nuevas etiquetas HTML personalizadas, reutilizables y auto-contenidas, que luego puedes utilizar en otras páginas y aplicaciones web. Estos componentes personalizados funcionarán en navegadores modernos y con cualquier biblioteca o framework de JavaScript que funcione con HTML (campusMVP, 2017).

#### **2.2.7.4. NodeJs**

Según (Sayago Heredia & Chango Sailema, 2018) Node.js es un framework de desarrollo, desarrollado originalmente en 2009 por Ryan Dahl, basado en el motor JavaScript V8 de Google. Node.js es una plataforma basada en el tiempo de ejecución JavaScript de Chrome para crear aplicaciones en red rápidas y escalables. Node.js utiliza un modelo



controlado por eventos y de no bloqueo de E/S que lo hace ligero y eficiente, perfecto para aplicaciones que requieren grandes cantidades de datos y que se ejecutan en dispositivos distribuidos. Node.js es un lenguaje de scripting del lado del servidor, que puede ser usado también en el lado del cliente e incluso puede ser un servidor web.

#### **2.2.8. Cloud computing**

El Instituto Nacional de Estándares y Tecnología y su laboratorio de tecnología de información, denominado NIST por el acrónimo en idioma inglés, han definido Cloud Computing de la siguiente manera: “Cloud Computing es un modelo para habilitar acceso conveniente por demanda a un conjunto compartido de recursos computacionales configurables, por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios, que pueden ser rápidamente provisionados y liberados con un esfuerzo mínimo de administración o de interacción con el proveedor de servicios. Este modelo de nube promueve la disponibilidad y está compuesto por cinco características esenciales, tres modelos de servicio y cuatro modelos de despliegue” (Mell & Timothy, 2011) traducción hecha por Guillermo Taylor.

Dentro de las características esenciales tenemos:

1. AUTOSERVICIO POR DEMANDA: Un consumidor puede provisionar de manera unilateral capacidades de cómputo, tales como tiempo de servidor y almacenamiento en red, en la medida en que las requiera sin necesidad de interacción humana por parte del proveedor del servicio.
2. ACCESO AMPLIO DESDE LA RED: Las capacidades están disponibles sobre la red y se acceden a través de mecanismos estándares que promueven el uso desde plataformas clientes heterogéneas, pesadas o livianas, como el PC, un teléfono móvil o un navegador Internet.





3. **CONJUNTO DE RECURSOS:** Los recursos computacionales del proveedor se habilitan para servir a múltiples consumidores mediante un modelo “multi-tenant”, con varios recursos tanto físicos como virtuales asignados y reasignados de acuerdo con los requerimientos de los consumidores. Existe un sentido de independencia de ubicación, en cuanto a que el consumidor no posee control o conocimiento sobre la ubicación exacta de los recursos que se le están proveyendo, aunque puede estar en capacidad de especificar ubicación a un nivel de abstracción alto; por ejemplo, país, estado o centro de datos. Algunos ejemplos incluyen almacenamiento, procesamiento, memoria, ancho de banda y máquinas virtuales.
4. **RÁPIDA ELASTICIDAD:** Las capacidades pueden ser rápida y elásticamente aprovisionadas, en algunos casos automáticamente, para escalar hacia fuera rápidamente y también rápidamente liberadas para escalar hacia dentro también de manera veloz. Para el consumidor, estas capacidades disponibles para aprovisionar a menudo aparecen como ilimitadas y pueden ser compradas en cualquier cantidad y en cualquier momento.
5. **SERVICIO MEDIDO:** Los sistemas en la nube controlan automáticamente y optimizan el uso de recursos mediante una capacidad de medición a algún nivel de abstracción adecuado al tipo de servicio; por ejemplo, almacenamiento, procesamiento, ancho de banda y cuentas de usuario activas. El uso de estos recursos puede ser monitoreado, controlado y reportado, proporcionando transparencia tanto para el proveedor como para el consumidor por el servicio utilizado.

Dentro de los tres modelos de servicio tenemos:

1. **SAAS – SOFTWARE AS A SERVICE:** Esta es una capacidad que se refiere a que el consumidor utiliza las aplicaciones del proveedor en una infraestructura de nube.



2. PAAS – PLATFORM AS A SERVICE: Esta capacidad le permite al consumidor desplegar en la infraestructura del proveedor aplicaciones creadas por el primero, incluso adquiridas, usando lenguajes de programación y herramientas del proveedor.
3. IAAS – INFRASTRUCTURE AS A SERVICE: Esta capacidad permite al consumidor aprovisionar recursos computacionales como almacenamiento, procesamiento, redes y otros elementos fundamentales, en donde el consumidor puede desplegar y correr software arbitrario, el cual puede incluir sistemas operacionales y aplicaciones.

Por último, los modelos de despliegue son:

1. NUBE PRIVADA: La infraestructura de esta nube es operada únicamente para una organización. Puede ser administrada por la organización o por un tercero y puede existir dentro de la misma, “on premises” o fuera de la misma, “off premises”.
2. NUBE COMUNITARIA: La infraestructura de esta nube es compartida por varias organizaciones y apoya las preocupaciones de una comunidad particular sobre un tema específico, por ejemplo, seguridad, investigación, políticas o cumplimientos. Puede ser administrada por la organización o por un tercero y puede existir dentro de la misma, “on premises” o fuera de la misma, “off premises”.
3. NUBE PÚBLICA: La infraestructura de esta nube está disponible para el público en general o para un gran grupo de industria y dicha infraestructura la provee una organización que vende servicios en la nube.
4. NUBE HÍBRIDA: Es la composición de dos o más nubes, por ejemplo, privada y pública, que permanecen como entidades únicas pero que coexisten por tener tecnología que permite compartir datos o aplicaciones entre las mismas. Piensen en un escenario en donde la aplicación se desarrolla y se prueba en una nube privada y luego se despliega a una nube pública.



### 2.2.8.1. PAAS

De acuerdo con (cloudflare, 2019) se puede acceder a “PaaS a través de cualquier conexión a Internet, lo que permite construir una aplicación completa en un navegador web. Debido a que el entorno de desarrollo no está alojado localmente, los desarrolladores pueden trabajar en la aplicación desde cualquier parte del mundo”. Esto permite que los equipos que se distribuyen en ubicaciones geográficas colaboren. También significa que los desarrolladores tienen menos control sobre el entorno de desarrollo, aunque esto conlleva mucho menos sobrecarga.

Los proveedores de PaaS generalmente incluyen:

- **HERRAMIENTAS DE DESARROLLO:** Los proveedores de PaaS ofrecen una variedad de herramientas que son necesarias para el desarrollo de software, incluido un editor de código fuente, un depurador, un compilador y otras herramientas esenciales. Estas herramientas se pueden ofrecer juntas como marco. Las herramientas específicas que se ofrecen dependerán del proveedor, pero las ofertas de PaaS deben incluir todo lo que un desarrollador necesita para construir su aplicación.
- **MIDDLEWARE:** Las plataformas que se ofrecen como servicio generalmente incluyen middleware, para que los desarrolladores no tengan que construirlo ellos mismos. Middleware es un software que se encuentra entre las aplicaciones orientadas al usuario y el sistema operativo de la máquina; Por ejemplo, el middleware es lo que permite que el software acceda a la entrada desde el teclado y el mouse. El middleware es necesario para ejecutar una aplicación, pero los usuarios finales no interactúan con él.



- **SISTEMAS OPERATIVOS:** Un proveedor de PaaS proporcionará y mantendrá el sistema operativo en el que los desarrolladores trabajan y la aplicación se ejecuta.
- **BASES DE DATOS:** Los proveedores de PaaS administran y mantienen bases de datos. Por lo general, también proporcionarán a los desarrolladores un sistema de gestión de bases de datos.
- **INFRAESTRUCTURA:** PaaS es la siguiente capa desde IaaS en el modelo de servicio de computación en la nube, y todo lo incluido en IaaS también se incluye en PaaS. Un proveedor de PaaS administra servidores, almacenamiento y centros de datos físicos, o los compra a un proveedor de IaaS.

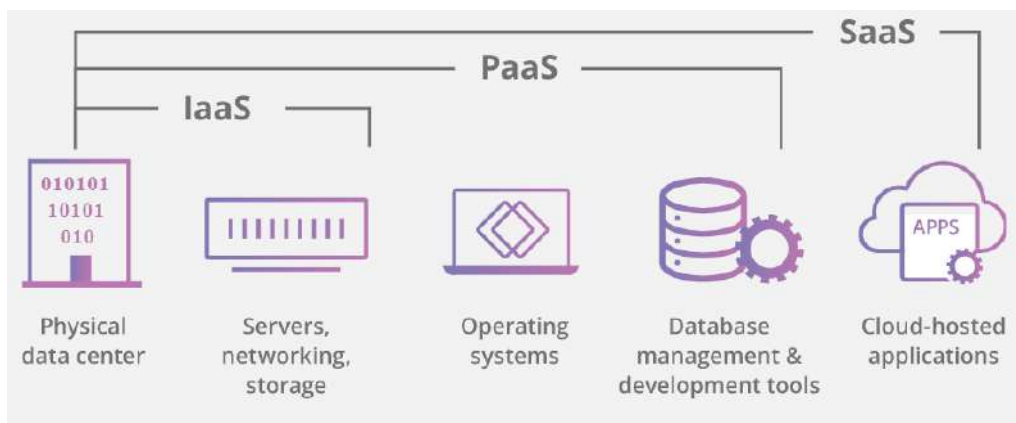


Figura 5 servicios prestados por los tipos de nubes

(cloudflare, 2019)

Ventajas y desventajas de usar PaaS:

Ventajas	Desventajas
<p><b>TIEMPO DE COMERCIALIZACIÓN MÁS RÁPIDO</b></p> <p>PaaS se usa para crear aplicaciones más rápidamente lo que no sería posible si los desarrolladores tuvieran que preocuparse por construir,</p>	<p><b>DEPENDENCIA DE UN PROVEEDOR</b></p> <p>Puede ser difícil cambiar de proveedor de PaaS, ya que la aplicación se crea utilizando las herramientas del proveedor y</p>



<p>configurar y aprovisionar sus propias plataformas e infraestructura de back-end. Con PaaS, todo lo que necesitan hacer es escribir el código y probar la aplicación, y el proveedor se encarga del resto.</p>	<p>específicamente para su plataforma. Cada proveedor puede tener diferentes requisitos de arquitectura. Es posible que diferentes proveedores no admitan los mismos idiomas, bibliotecas, API, arquitectura o sistema operativo que se utilizan para compilar y ejecutar la aplicación. Para cambiar de proveedor, es posible que los desarrolladores necesiten reconstruir o alterar en gran medida su aplicación.</p>
<p><b>UN ENTORNO DE PRINCIPIO A FIN</b> PaaS permite a los desarrolladores construir, probar, depurar, implementar, alojar y actualizar sus aplicaciones, todo en el mismo entorno. Esto permite a los desarrolladores asegurarse de que una aplicación web funcione correctamente antes de su lanzamiento, y simplifica el ciclo de vida de desarrollo de la aplicación.</p>	<p><b>DEPENDENCIA DEL PROVEEDOR</b> El esfuerzo y los recursos involucrados en el cambio de proveedores de PaaS pueden hacer que las empresas sean más dependientes de su proveedor actual. Un pequeño cambio en los procesos internos o la infraestructura del proveedor podría tener un gran impacto en el rendimiento de una aplicación diseñada para ejecutarse eficientemente en la configuración anterior. Además, si el proveedor cambia su modelo de precios, una aplicación puede volverse de repente más costosa de operar.</p>
<p><b>PRECIO</b> Los gastos generales se reducen porque los clientes de PaaS no necesitan administrar y aprovisionar máquinas virtuales. Además, algunos</p>	<p><b>DESAFÍOS DE SEGURIDAD Y CUMPLIMIENTO</b> En una arquitectura PaaS, el proveedor externo almacenará la mayoría o la totalidad de los datos de</p>



<p>proveedores tienen una estructura de precios de pago por uso, en la que el proveedor solo cobra por los recursos informáticos utilizados por la aplicación, generalmente haciendo ahorrar dinero a los clientes. Sin embargo, cada proveedor tiene una estructura de precios ligeramente diferente, y algunos proveedores de plataformas cobran una tarifa plana por mes.</p>	<p>una aplicación, junto con el alojamiento de su código. En algunos casos, el proveedor puede almacenar las bases de datos a través de un tercero adicional, un proveedor de IaaS. Aunque la mayoría de los proveedores de PaaS son grandes empresas con una fuerte seguridad, esto hace que sea difícil evaluar y probar completamente las medidas de seguridad que protegen la aplicación y sus datos. Además, para las empresas que tienen que cumplir con estrictas normas de seguridad de datos, verificar el cumplimiento de proveedores externos adicionales agregará más obstáculos para salir al mercado.</p>
<p><b>FACILIDAD DE LICENCIA</b> Los proveedores de PaaS manejan todas las licencias para sistemas operativos, herramientas de desarrollo y todo va incluido en su plataforma.</p>	

Tabla 1 ventajas y desventajas del PaaS

(cloudflare, 2019)

### 2.2.9. REST

Rest es un nuevo enfoque que fue desarrollado por Roy Fielding, cambiando por completo la ingeniería de software a partir del 2000. En el campo de las APIs, REST (Representational State Transfer- Transferencia de Estado Representacional) en la actualidad, es uno de los más usados en el desarrollo de servicios de aplicaciones (BBVAOPEN4U, 2016).



Según (BBVAOPEN4U, 2016) “REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON”.

BBVAOPEN4U indica que REST tiene las siguientes características:

1. PROTOCOLO CLIENTE/SERVIDOR SIN ESTADO: Cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni el cliente ni el servidor, necesiten recordar algún estado previo para satisfacer las peticiones.
2. OPERACIONES MÁS IMPORTANTES DE REST Y ESPECIFICACIONES HTTP: Las operaciones más importantes relacionadas con los datos son cuatro, POST (crear), GET (leer), PUT (editar), DELETE (eliminar).
3. LOS OBJETOS EN REST SIEMPRE SE MANIPULAN A PARTIR DE LA URI: Es la URI el identificador único de cada recurso del sistema REST, por tanto, la URI facilita el acceso a la información para poder realizar las peticiones.
4. INTERFAZ UNIFORME: La transferencia de datos de un sistema REST se realiza con acciones concretas (POST, GET, PUT, DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Así se tiene una interfaz uniforme que sistematiza el procesamiento de la información.

Las ventajas que ofrece REST según BBVAOPEN4U son:

1. SEPARACIÓN ENTRE EL CLIENTE Y EL SERVIDOR: REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos, lo que genera grandes ventajas al momento de desarrollar una aplicación; como por ejemplo aumenta la portabilidad de la interfaz a otro tipo de plataformas, amplía la escalabilidad de los proyectos y permite que los distintos componentes de desarrollo puedan evolucionar de forma independiente.



2. VISIBILIDAD, FIABILIDAD Y ESCALABILIDAD: la separación entre cliente y servidor permite migrar a otros servidores o realizar todo tipo de cambios en la base de datos, siempre y cuando los datos de cada una de las peticiones se envíen de la forma correcta. Esta separación facilita tener en servidores distintos el frontEnd y el backEnd y eso convierte a las aplicaciones en productos más flexibles a la hora de trabajar.
3. SIEMPRE ES INDEPENDIENTE DEL TIPO DE PLATAFORMAS O LENGUAJES: Se adapta al tipo de sintaxis o plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo; para esto lo indispensable es que las respuestas a las peticiones sean siempre en el lenguaje de intercambio de información usado, que suelen ser XML o JSON.

#### **2.2.10. Heroku**

Como plantea (Urrutia, 2018) Heroku es una empresa que se especializa en ofrecer servicios de plataforma administrada, por sus siglas en inglés PasS (Platform as a Service), en otras palabras ofrece servicios de servidores y redes administrados por Heroku en donde se pueden alojar aplicaciones de diferentes lenguajes de programación como Python, Java, PHP y más. Los servicios que Heroku ofrece están claramente orientados para desarrolladores, startups y empresas pequeñas. Heroku nació en el 2007 y fue adquirida en el 2010 por salesforce.com.

Urrutia también indica que Heroku utiliza contenedores Linux (Ubuntu) los cuales son llamados “dynos”, estos son utilizados para alojar las aplicaciones web, webservices o aplicaciones que se ejecutan del lado del servidor, así mismo cuenta con la posibilidad de instalar add-ons para agregar funcionalidades a dichos contenedores, por ejemplo, se pueden agregar servicios administrados de base de datos, almacenamiento en la nube o mensajería (MQ) por mencionar algunos.





### 2.2.11. Mongoose

Mongoose es una herramienta de modelado de objetos para MongoDB y Node.js. Lo que esto significa en términos prácticos es que puede definir su modelo de datos en un solo lugar, en su código. No tiene que crear un esquema en la base de datos, vincularlo a un ORM (mapeo objeto-relacional) o asignarlo a los objetos y clases de su proyecto. Simplemente puede definir su estructura de datos en JSON dentro de su proyecto (Holmes, 2013, pág. 12).

Una definición de esquema puede ser tan simple como el siguiente fragmento de código:

```
var userSchema = new mongoose.Schema({
  firstname: String,
  lastname: String,
  createdOn: Date
});
```

*Figura 6 definición de un esquema*

*(Holmes, 2013)*

Un documento en MongoDB creado a partir de este esquema sería como el siguiente fragmento de código:

```
{ "__v" : 0, "_id" : ObjectId("51412597e8e6d3e35c000001"),
  "createdOn" : ISODate("2013-03-14T01:19:19.866Z"),
  "firstname" : "Simon", " lastname " : "Holmes" }
```

*Figura 7 documento json*

*(Holmes, 2013)*

Según Simón Holmes Mongoose posee dos pilares, los que se desarrollan a continuación:

1. **ESQUEMAS DE MONGOOSE:** Este esquema describe fundamentalmente la construcción de datos de un documento. Este esquema define el nombre de cada elemento de datos y el tipo de datos, ya sea una cadena, número, fecha, booleano, etc.



```
var userSchema = new mongoose.Schema({
  name: String,
  email: String,
  createdAt: Date,
  verified: Boolean
});
```

*Figura 8 definición del esquema*

*(Holmes, 2013)*

En la mayoría de los escenarios, tendría un esquema para cada colección dentro de la base de datos. Los esquemas son un aspecto poderoso de Mongoose, que también se puede ampliar con funciones auxiliares y métodos adicionales.

2. **MODELO DE MONGOOSE:** Es una versión compilada del esquema. Una instancia del modelo que se asignará a un documento en la base de datos. En la imagen se crea una instancia del usuario basada en el esquema userSchema.

```
var User = mongoose.model('User', userSchema);
```

*Figura 9 creación del modelo*

*(Holmes, 2013)*

Es el modelo el que maneja: la lectura, creación, actualización y eliminación de documentos.

### **2.2.12. Bcrypt**

Como indica (Rohan, 2018) Bcrypt funciona haciendo uso de Hash que genera bytes aleatorios (el salt) y lo combina con la contraseña, antes del hash crea hashes únicos en la contraseña de cada usuario. Si dos usuarios tienen la misma contraseña, no tendrán el mismo hash de contraseña. Esto es para evitar ataques de la tabla del arco iris (Una tabla de arco iris es una tabla precalculada para revertir las funciones hash criptográficas, generalmente para descifrar hashes de contraseñas) que pueden revertir las contraseñas hash utilizando funciones de hashing comunes que no utilizan salt.



Utiliza algoritmos de hash que son funciones unidireccionales. Convierten cualquier cantidad de datos en una "huella digital" de longitud fija que no se puede revertir. También tienen la propiedad de que, si la entrada cambia incluso un poco, el hash resultante es completamente diferente. Esto es excelente para proteger las contraseñas, porque al almacenar las contraseñas de una manera que las proteja incluso si el archivo de contraseña en sí está comprometido, pero al mismo tiempo, debemos poder verificar que la contraseña de un usuario sea correcta.

El flujo de trabajo general para el registro y la autenticación de cuentas en un sistema de cuentas basado en hash es el siguiente:

1. El usuario crea una cuenta.
2. Su contraseña está cifrada y almacenada en la base de datos. En ningún momento se escribe la contraseña de texto sin formato (sin cifrar) en el disco duro.
3. Cuando el usuario intenta iniciar sesión, el hash de la contraseña que ingresó se compara con el hash de su contraseña real (recuperado de la base de datos).
4. Si los hashes coinciden, el usuario tiene acceso. De lo contrario, se le dice al usuario que ingresó credenciales de inicio de sesión no válidas.
5. Los pasos 3 y 4 se repiten cada vez que alguien intenta iniciar sesión en su cuenta.

### **2.2.13. Json web token**

Toda la explicación que se va a dar a continuación ha sido planteado por (Macias, 2018), señala que JWT es un estándar RFC 7519 para transmitir información con la identidad y claims de un usuario de forma segura entre un cliente/servidor. Dicha información puede ser verificada y confiable porque está firmada digitalmente.

En pocas palabras JWT es "simplemente" una cadena de texto que tiene 3 partes codificadas en Base64, separadas por un punto



(header.payload.firma) que generamos y entregamos a los clientes de nuestra API:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ
```

*Figura 10 JWT*

*(Macias, 2018)*

Es importante aclarar que la cadena/token esta codificado y lo crea la aplicación, esto permite de manera muy fácil inspeccionar su contenido.

- a) HEADER: **eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9** (indica el algoritmo y tipo de Token, en este caso: HS256 y JWT).
- b) PAYLOAD: **eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoiYWRtaW4iOnRydWV9** (datos de usuario/claims, fecha creación/caducidad Token y lo que necesite el API para validar la petición, se debe tener en cuenta que nosotros generamos el token y puede incluir todos los atributos que se desee).
- c) SIGNATURE: **TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ** (la firma, para verificar que el token es válido, aquí lo importante es el "secret" que es la o las palabras claves con la que firmamos).

El ciclo de vida de un token JWT se muestra en la siguiente ilustración:

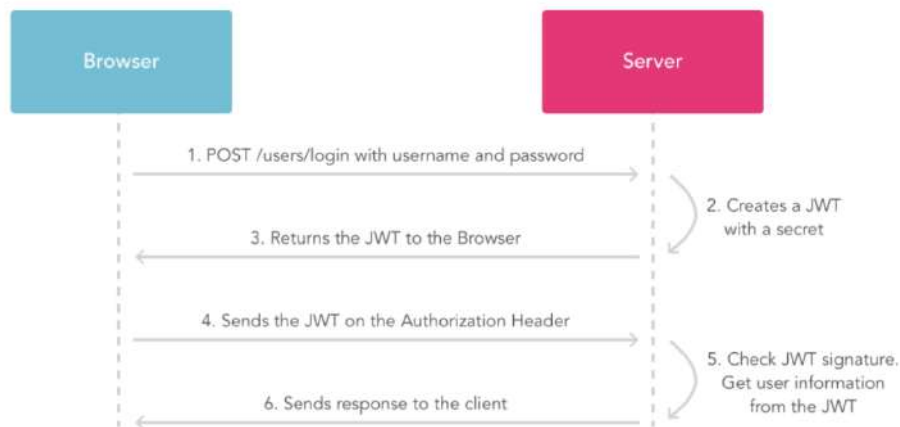


Figura 11 proceso del JWT

(Macias, 2018)

El proceso completo del JWT consta de los siguientes pasos:

1. El usuario de una aplicación web/móvil/desktop hace login con sus credenciales en el servidor donde esta publicada el API.
2. El usuario es validado en el servidor y se crea un nuevo Token JWT (usando nuestro "secret-key") para entregárselo al usuario.
3. El servidor retorna el JWT firmado que contiene los datos/claims referentes al usuario y caducidad del Token.
4. El cliente/browser almacena el JWT para su uso y lo envía en cada petición mediante "Authorization: Bearer".
5. El servidor verifica la firma del Token, su fecha de caducidad y comprueba si usuario tiene permisos al recurso leyendo los datos del payload.
6. El servidor responde al cliente, con la petición una vez ha confirmado el Token y los permisos del usuario son correctos.

#### 2.2.14. JSPDF

De acuerdo con (unipython, 2019) "jsPDF es una librería para Java Script que permite generar documentos PDF a partir de una plantilla HTML o directamente por programación", permitiendo generar un documento PDF de manera sencilla y adaptable.



Una ventaja muy importante que se obtiene al hacer uso de esta librería, es que no se usan recursos del lado del servidor, puesto que se ejecuta del lado del cliente.

### 2.2.15. Usabilidad web

Para entender bien que es la usabilidad web, daremos una definición de que es la usabilidad.

Usabilidad proviene de la palabra en ingles Usability que significa facilidad de uso, por lo tanto, usabilidad se refiere a la facilidad de interacción que tiene un usuario con una aplicación o producto. (Eugenia, 2018)

Componentes de la usabilidad:

1. Facilidad De Uso
2. Eficiencia
3. Memorabilidad
4. Errores
5. Satisfacción
6. Utilidad

Existen varios principios de usabilidad según diferentes expertos, en la tabla que mostramos a continuación se señala una lista de algunos principios según diferentes autores.

Autor	Principios de usabilidad
Jakob Nielsen	<ol style="list-style-type: none"><li>1. Visibilidad del estado del sistema</li><li>2. Encaje entre el sistema y el mundo real</li><li>3. Libertad y control por parte del usuario</li><li>4. Consistencia y estándares</li><li>5. Prevención de errores</li><li>6. Reconocimiento antes que recuerdo</li><li>7. Flexibilidad y eficiencia en el uso</li><li>8. Diseño estético y minimalista</li></ol>



	<ol style="list-style-type: none"><li>9. Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores</li><li>10. Ayuda y documentación</li></ol>
Larry Constantine	<ol style="list-style-type: none"><li>1. ESTRUCTURA: organiza con significado.</li><li>2. SIMPLICIDAD: has fáciles las tareas comunes.</li><li>3. VISIBILIDAD: muestra toda aquella información necesaria para una tarea.</li><li>4. RETROALIMENTACIÓN: mantén informados a los usuarios.</li><li>5. TOLERANCIA: permite cancelar, deshacer, volver.</li><li>6. REUTILIZACIÓN: reduce la necesidad de los usuarios de recordar.</li></ol>
Keith Instone	<ol style="list-style-type: none"><li>1. Diálogo simple y natural</li><li>2. Habla el lenguaje del usuario</li><li>3. Minimiza la carga de memoria del usuario</li><li>4. Consistencia</li><li>5. Retroalimentación</li><li>6. Salidas claramente marcadas</li><li>7. Atajos</li><li>8. Buenos mensajes de error</li><li>9. Preve errores</li><li>10. Ayuda y documentación</li></ol>
Bruce Tognazzini	<ol style="list-style-type: none"><li>1. Anticipación</li><li>2. Autonomía</li><li>3. Ceguera al color</li><li>4. Consistencia</li><li>5. Configuraciones por defecto</li><li>6. Eficiencia del usuario</li><li>7. Interfaces explorables</li><li>8. Ley de Fitts</li></ol>



	<ol style="list-style-type: none"><li>9. Objetos de interfaz humanos</li><li>10.Reducción de tiempos de latencia</li><li>11.Aprendizaje</li><li>12.Uso de metáforas</li><li>13.Protección del trabajo del usuario</li><li>14.Legibilidad</li><li>15.Seguir el estado</li><li>16.Navegación visible</li></ol>
Ben Schneiderman	<ol style="list-style-type: none"><li>1. Lucha por la consistencia</li><li>2. Crea atajos para los usuarios frecuentes</li><li>3. Ofrece feedback</li><li>4. Diseña el diálogo para mostrar trabajo pendiente</li><li>5. Ofrece una gestión sencilla de los errores</li><li>6. Permite una fácil recuperación de acciones</li><li>7. Soporta el control por el usuario</li><li>8. Reduce la carga de memoria reciente en el usuario</li></ol>
Deborah Mayhew	<ol style="list-style-type: none"><li>1. Compatibilidad de usuario</li><li>2. Compatibilidad de producto</li><li>3. Compatibilidad de tareas</li><li>4. Compatibilidad de procesos</li><li>5. Consistencia</li><li>6. Familiaridad</li><li>7. Simplicidad</li><li>8. Manipulación directa</li><li>9. Control</li><li>10.WYSIWYG (en español, "lo que ves es lo que obtienes")</li><li>11.Flexibilidad</li><li>12.Sensibilidad y feedback</li><li>13.Tecnología invisible</li><li>14.Robustez</li></ol>





	15. Protección 16. Facilidad de aprendizaje y facilidad de uso
--	-------------------------------------------------------------------

*Tabla 2 principios de usabilidad*

(Gimenez , 2017).

De los principios de usabilidad mostrados en la tabla anterior según los diferentes autores, el más usado es el de Jakob Nielsen que propone 10 Principios de Usabilidad, los cuales se desarrollan a continuación tomado en cuenta los conceptos desarrollados por (Gimenez , 2017).

1. VISIBILIDAD DEL ESTADO DEL SISTEMA: El sistema siempre debe informar al usuario de lo que está haciendo, es decir, proveer feedback en un tiempo razonable. Se revisan elementos cómo, dónde se encuentra el usuario en la arquitectura del sistema (títulos, migas de pan...); si es un listado de enlaces, cuáles han sido visitados, si hay un proceso en marcha que comunique su estado o cuántos pasos faltan para terminar el proceso.
2. ENCAJE ENTRE EL SISTEMA Y EL MUNDO REAL: El sistema y el usuario tienen que hablar en el mismo idioma, siguiendo las convenciones del entorno del usuario.
3. LIBERTAD Y CONTROL POR PARTE DEL USUARIO: Tener sensación de control y no tener miedo de efectuar acciones y probar cosas nuevas, es de vital importancia para que la persona interactúe y aprenda por sí misma.
4. CONSISTENCIA Y ESTÁNDARES: Es muy importante tener consistencia a lo largo de todo el sistema y no ir variando los elementos y su funcionamiento en cada pantalla.
5. PREVENCIÓN DE ERRORES: Lo ideal es que nunca haya errores, con instrucciones claras de que se debe hacer en cada pantalla, sistemas de ayuda. Si se pueden validar los errores antes de enviar la acción, como la validación en línea de un formulario.



6. RECONOCIMIENTO ANTES QUE RECUERDO: El usuario dispone de poca memoria a corto plazo, por lo que minimizar el uso de su memoria es importante, se debe colocar las opciones a la vista o con un fácil acceso. Es mucho más fácil que reconozca algo, a que lo recuerde.
7. FLEXIBILIDAD Y EFICIENCIA EN EL USO: El diseño debe servir tanto para usuarios inexpertos como expertos.
8. DISEÑO ESTÉTICO Y MINIMALISTA: No todas las acciones pueden estar a la vista, ya que cada unidad de información reduce la visibilidad de la información que de verdad importa. Manejar correctamente las jerarquías visuales y el espacio en blanco es importante.
9. AYUDA A LOS USUARIOS A RECONOCER, DIAGNOSTICAR Y RECUPERARSE DE LOS ERRORES: En el caso de que haya una situación de error, el sistema debe indicar al usuario que ha pasado y cómo resolverlo.
10. AYUDA Y DOCUMENTACIÓN: Aunque el sistema es suficientemente usable y no es necesario ninguna documentación, siempre habrá usuarios que puedan necesitarla. Por ello es importante verificar que el sistema ofrezca ayuda relevante al contexto del usuario.

#### **2.2.16. Scrum**

La siguiente información es un resumen de una guía elaborada por: Ken Schwaber y Jeff Sutherland. (Schwaber & Sutherland, 2017)

##### **2.2.16.1. Eventos de Scrum**

###### **2.2.16.1.1. Sprint planing**

Responde las siguientes preguntas:

- a) ¿Qué se puede hacer en el siguiente sprint?
  - Solo el equipo de desarrollo puede evaluar lo que se puede lograr en el próximo sprint



- El equipo de desarrollo crea un objetivo dentro del sprint a través de la implementación del product backlog
  - El product owner debe de conocer el objetivo plateado para el siguiente sprint
- b) ¿Cómo se hará el trabajo elegido?
- Al final de la planificación del sprint, el equipo debe poder explicar al product owner y scrum master, como pretenden trabajar, para poder completar el objetivo del sprint
  - Se planea suficiente trabajo, como para poder pronosticar que producto se entregara al final del sprint

#### **Sprint Goal:**

- a) Es el objetivo que se cumplirá a través de la implementación del product backlog
- b) Permite que el equipo trabaje en conjunto con coherencia y no con iniciativas separadas

#### **2.2.16.1.2. Daily Scrum**

- a) Tiene una duración de 15 minutos y se lleva a cabo todos los días del sprint
- b) Se planea el trabajo de las próximas 24 horas
- c) Se lleva a cabo todos los días a la misma hora y lugar para disminuir su complejidad
- d) Se usa para inspeccionar la tendencia de progreso, algunas preguntas que ayudan a evaluar el progreso son:
  - ¿Qué hiciste ayer, que ayudo al equipo de desarrollo a alcanzar el objetivo del sprint?
  - ¿Qué harás hoy, para ayudar a alcanzar el objetivo del sprint?
  - ¿Hay algún impedimento u obstáculo, que afecte al cumplimiento del objetivo del sprint?
- e) El Scrum Master se asegura de que se lleve a cabo la reunión, pero el equipo de desarrollo se hace cargo del desarrollo del daily Scrum



#### 2.2.16.1.3. Sprint review

- a) Para un sprint de 1 mes, tiene una duración aproximada de 4 horas
- b) Esta revisión incluye los siguientes elementos:
  - Los asistentes incluyen: al equipo de desarrollo, las principales partes interesadas invitadas por el Product Owner y el Scrum Master
  - El Product Owner explica que elementos del product backlog se han hecho y que no se han hecho
  - El equipo de desarrollo discute que fue lo que salió bien durante el sprint, con que problemas se encontraron y como se resolvieron esos problemas
  - El equipo de desarrollo demuestra el trabajo que ha hecho y responde las preguntas sobre el incremento
  - El Product Owner analiza la acumulación de los productos tal y como esta. Él o ella proyecta fechas de entrega y objetivos probables, según el cronograma hasta ese momento (si es necesario)
  - Todo el grupo colabora, en ¿qué hacer a continuación?, para que la revisión del sprint proporcione información valiosa, para la posterior planificación del siguiente sprint
  - Se revisa la línea del tiempo, el presupuesto y el mercado para las próximas versiones anticipadas de la funcionalidad o capacidad del producto

#### 2.2.16.1.4. Sprint retrospectiva

- a) Permite al equipo inspeccionarse a sí mismo, implementar mejoras durante el próximo sprint
- b) La retrospectiva del sprint ocurre después de la revisión del sprint y antes de la próxima planeación del sprint, tiene una duración aproximada de 3 horas
- c) Es propósito de este evento es:



- Inspeccionar como fue el último sprint con respecto a las personas, relaciones de equipo, el proceso y las herramientas
  - Identificar y ordenar los elementos principales que salieron bien y las mejoras potenciales
  - Crear un plan para implementar mejoras en la forma en que el equipo hace su trabajo
- d) La retrospectiva del sprint brinda una oportunidad formal para enfocarse en la inspección y adaptación

### **2.2.16.2. Scrum artifacts**

#### **2.2.16.2.1. Product backlog**

- a) Es una lista ordenada de todo lo que se necesita en el producto. Es la única fuente de requisitos para cualquier cambio que se realice en el producto
- b) El Product Backlog es dinámico, éste evoluciona a medida que evoluciona el producto y el entorno en que se usa.
- c) Tiene a menudo una definición Done (el producto está listo o hecho), que indica cuando se da por completado una parte del producto total
- d) El refinamiento del Product Backlog es el agregar detalles, estimaciones y orden al Product Backlog; el Product Owner puede actualizar el Product Backlog en cualquier momento aunque no es recomendable

#### **Monitoreo del progreso hacia los objetivos:**

- a) El Product Owner realiza el seguimiento del trabajo total
- b) Para evaluar el progreso se usan diversas herramientas como esquemas, flujos acumulativos, etc.

#### **2.2.16.2.2. Sprint backlog**

- a) Conjunto de elementos del Product Backlog seleccionados para el sprint, más un plan de entrega del producto y el objetivo del sprint.



- b) Es un pronóstico que hace el equipo de desarrollo para lograr un incremento en el producto
- c) A medida que el equipo de desarrollo avanza en el trabajo, se puede cambiar el Sprint Backlog, reestimando el trabajo restante

#### **2.2.16.2.3. Increment**

Es la suma de todos los elementos del Product Backlog completados durante el sprint y el valor del incremento de todos los sprints anteriores

#### **2.2.16.3. transferencia de los artefactos Scrum**

- a) El Scrum Master debe trabajar con todos los involucrados para cerciorarse de que todo el proceso sea transparente
- b) El Scrum Master puede detectar una transparencia incompleta, verificando si hay diferencias entre los resultados reales y los esperados

#### **2.2.16.3.1. Definición de lo listo (Done)**

- a) Cuando un elemento del Product Backlog se dé como listo, todos deben de entender que significa listo
- b) La definición de listo debe de entregar in incremento del producto que el Product Owner puede lanzar para ser usado



## CAPÍTULO III

### 3. Desarrollo, implementación o transferencia tecnológica

En el presente capítulo se muestra el proceso de desarrollo de Software que se realizó en cinco Sprints, las pruebas de usabilidad realizadas al sistema, el indicador 6 del componente 1.5 del Proceso de Licenciamiento de las Universidades y los costos de implementación y despliegue de la aplicación.

#### 3.1. Desarrollo

##### 3.1.1. Planteamiento de los sprints

Los lineamientos generales para el desarrollo de los sprints para el desarrollo de la aplicación son los siguientes:

- a) Roles:
  - a. Product Owner: Ingeniero Liu Phol Ramos Fernández
  - b. Scrum Master: Esperanza Wiese Gutiérrez
  - c. Development team members: Alberto Cano Alatriza, Esperanza Wiese Gutiérrez
- b) La duración de cada Sprint será de 4 semanas.

##### 3.1.2. Producto backlog

###### 3.1.2.1. Historia de usuarios

En el siguiente cuadro se muestra las historias de usuarios que hemos distribuido en tres Sprints, cada historia de usuario cuenta con sus respectivos pesos y prioridades, las prioridades se asignaron por Sprints.

N°	Historias de usuario	pesos	prioridad
1	Yo como jefe de la oficina de admisión quiero que el sistema me permita insertar, activar y desactivar procesos y modalidades, para poder gestionar los procesos de admisión	5	2
2	Yo como jefe de la oficina de admisión de admisión quiero que el sistema me permita insertar, activar y	8	2



	desactivar carreras y requisitos, para poder gestionar los procesos de admisión		
3	Yo como jefe de la oficina de admisión quiero que el sistema me permita gestionar el proceso, modalidad, carrera y requisitos de manera conjunta, para la apertura de un proceso de admisión	13	3
4	Yo como jefe de la oficina de admisión quiero poder crear usuarios en el sistema, para poder saber qué rol desempeña cada usuario y durante qué periodo de tiempo	21	1
5	Yo como postulante quiero poder realizar mi inscripción por internet, para entonces no tener que ir a hacer una larga cola para en la universidad para mi inscripción	8	1
6	Yo como operario quiero poder verificar que los datos del postulante en el sistema estén correctos, para entonces no tener problemas después al procesar la información	5	2
7	Yo como operario quiero poder registrar fácilmente que documentos está entregando el postulante, para entonces hacer más rápida la inscripción de los postulantes y para tener registrado que documentos entrego el postulante al momento de su inscripción	5	3
8	Yo como postulante quiero poder recibir una constancia de mi inscripción al Examen de Admisión, con la información de la inscripción, para entonces poder estar seguro de que realice mi inscripción de manera exitosa.	5	4
9	Yo como director de la oficina de admisión quiero que el postulante sea informado a través del sistema el plazo que tiene para poder regularizar sus documentos, para entonces asegurarme de que fueron notificados de la fecha límite de regularización de documentos.	3	1
10	Yo como director de la oficina de admisión quiero poder verificar cuantos postulantes hay en total en un	13	2





	determinado tiempo del proceso de admisión y cuantos postulantes hay por carreras, para entonces poder hacer un control constante de la cantidad de postulantes que se está recibiendo en general y por carreras.		
11	Yo como director de la oficina de admisión quiero poder saber cuántos postulantes entregaron completos sus documentos y cuantos no, para poder saber la cantidad de postulantes deben de regularizar sus documentos y la lista de nombres completos de los postulantes que entregaron completo sus documentos.	8	3
	total	94	

*Tabla 3 historias de usuario*

Elaboración propia

### 3.1.2.2. Definición de lo listo (Done)

Se dará como completada una historia de usuario cuando cumpla con los siguientes puntos:

- a) El producto cumpla con la funcionalidad definida en el product backlog
- b) El código se encuentre documentado (entiéndase por documentado cuando hay comentarios que ayudan a entender fácilmente el código)
- c) Se encuentre el producto desplegado en Heroku



### 3.1.3. Diagrama de la Base de Datos

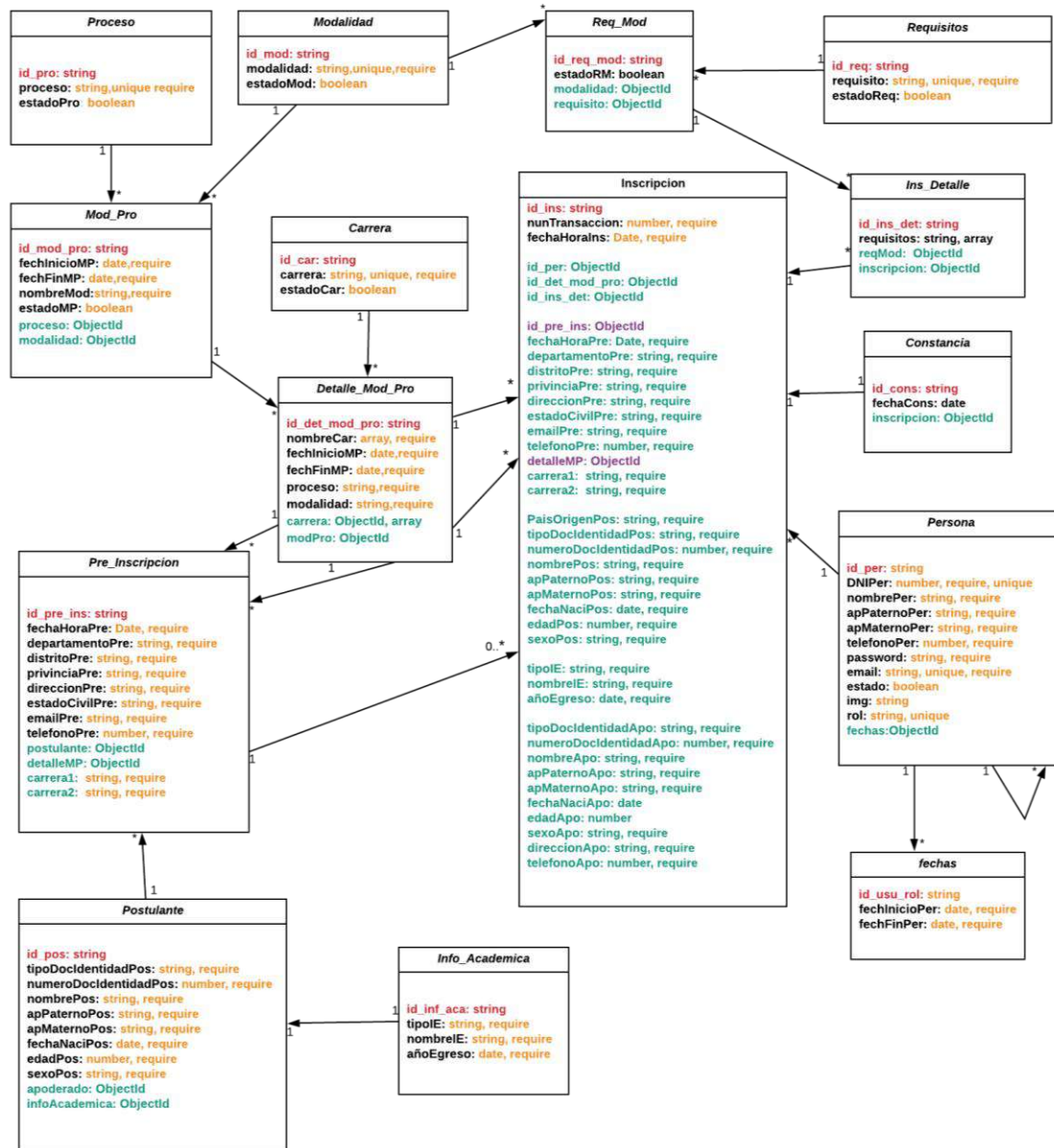


Figura 12 diagrama de base de datos

Elaboración propia

### 3.1.4. Sprint 1

#### 3.1.4.1. Planteamiento del Sprint

El Sprint dos tendrá una duración de cuatro semanas, para este se tiene previsto completar las historias de usuario N° 4, 3, 1. Se desarrollará la parte del mantenimiento del sistema (CRUAD de proceso, modalidad, carrera, requisitos, detalle y usuario) y un login para los usuarios del sistema y



el CRUAD de persona. Las tareas que se asignaron a cada miembro del equipo son las siguientes:

- a) Alberto Cano: FrontEnd del Sistema
- b) Esperanza Wiesse: BackEnd del Sistema

### 3.1.4.2. Sprint backlog

N°	Historias de usuario	pesos	prioridad
4	Yo como jefe de la oficina de admisión quiero poder crear usuarios en el sistema, para poder saber qué rol desempeña cada usuario y durante qué periodo de tiempo	21	1
3	Yo como jefe de la oficina de admisión quiero que el sistema me permita gestionar el proceso, modalidad, carrera y requisitos de manera conjunta, para la apertura de un proceso de admisión	13	3
1	Yo como jefe de la oficina de admisión quiero que el sistema me permita insertar, activar y desactivar procesos y modalidades, para poder gestionar los procesos de admisión	8	2
	total	42	

*Tabla 4 historias de usuario*

Elaboración propia

### 3.1.4.3. Daily meeting

En este primer sprint lo que se hizo en el frontEnd es hacer la estructura de archivos para que los componentes se consuman unos a otros, el sidebar, no page found, la estructura del componente breadCrums, creación de la estructura del logIn y su separación del resto de la estructura del proyecto, creación de la ruta principal y sus rutas hijas, en esta parte se tuvo dificultades al hacer la estructura de cómo se relaciona las rutas hijas, el CRUAD de persona. En el backEnd se estableció la conexión entre Node.js y mongo



y demás configuraciones iniciales, la creación del Schema de la tabla persona (con una función que evita que se envíe la contraseña al hacer una petición al servidor), elaboración de las peticiones CRUAD de persona, creación del Schema de la tabla fechas y su encadenamiento con la tabla personas para su creación, implementación de la búsqueda para realizar la búsqueda de una persona, la función de subir la imagen de usuario de persona, creación del login en el que se tuvo varios inconvenientes al hacer el manejo del token. En el anexo 1 se encuentra detallado el avance día a día del Daily Meeting.

Para la gestión de las historias de usuarios del presente sprint y de los siguientes se utilizó Trello, para esto el tablero se dividió en 4 partes, que son las siguientes:

- a) Definición
- b) En proceso de desarrollo
- c) Elaboración de la documentación, y
- d) La aceptación.

Luego para cada historia de usuario se elaboró una pequeña descripción y un checkList, que mostramos a continuación:

- a) Yo como jefe de la oficina de admisión quiero poder crear usuarios en el sistema, para poder saber qué rol desempeña cada usuario y durante qué periodo de tiempo.
  - checkList del FrontEnd



- hacer el login de persona
- hacer el logout de persona
- creación del servicio de persona
- mostrar la tabla de datos de persona
- creación de un persona
- actualización de una persona
- activación y des-activación de una persona
- actualizar la imagen de usuario de persona

*Figura 13 check list*

Elaboración propia

- **checkList del BackEnd**

- hacer el login de persona
- creación de la ruta para de persona
- creación del Schema de la tabla persona
- petición get (obtiene todos los datos de la tabla persona)
- petición post (creación de una persona)
- petición get (de un sola persona)
- petición put (actualización de una persona)
- petición put (activación y desactivación de una persona)
- actualizar la imagen de usuario de persona

*Figura 14 check list*

Elaboración propia

#### **3.1.4.4. Burn down chard**

En el siguiente grafico la línea negra representa el avance esperado, mientras que la línea roja el avance real del equipo; como se puede apreciar, en el grafico el primer sprint fue fallido, solo se completó la historia de usuario N° 4.

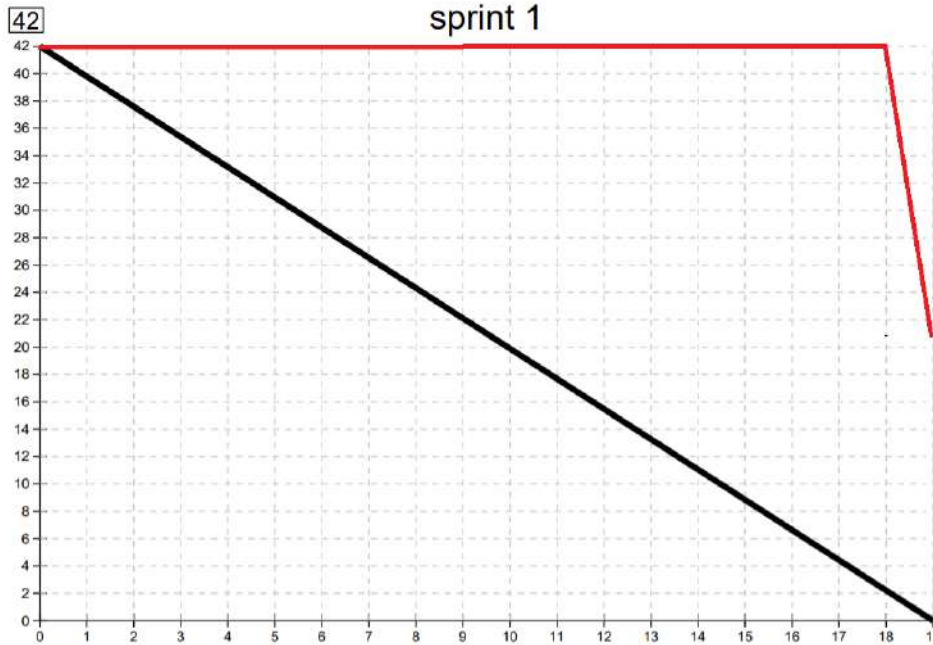


Figura 15 burn down chard

Elaboración propia

#### 3.1.4.5. Retrospectiva del sprint

- ¿Qué ha funcionado bien en este sprint?  
Hemos avanzado a buen ritmo en la programación
- ¿Cuáles con las cosas que se pueden mejorar para el siguiente sprint?  
Disminuir la carga de trabajo, debimos considerar el tiempo que llevaría armar la estructura de archivos del proyecto en el FrontEnd
- ¿Qué inconvenientes hemos tenido para poder progresar correctamente en este sprint?  
La preparación de la estructura del proyecto ha llevado más tiempo de lo planeado
- Recomendaciones para el siguiente sprint  
Tomar en cuenta el tiempo y la carga de trabajo para estimar mejor las metas

#### 3.1.4.6. Despliegue de la aplicación en Heroku

En las siguientes imágenes se puede apreciar las capturas de pantalla de la historia de usuario N°4 desplegada en Heroku.

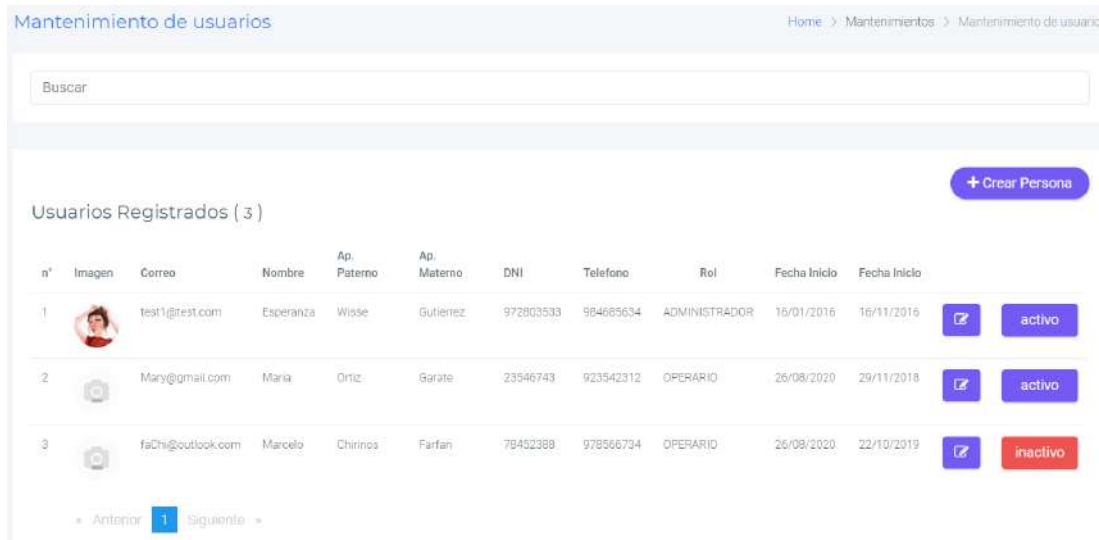


Figura 16 captura de pantalla del mantenimiento de usuarios del sistema

Elaboración propia

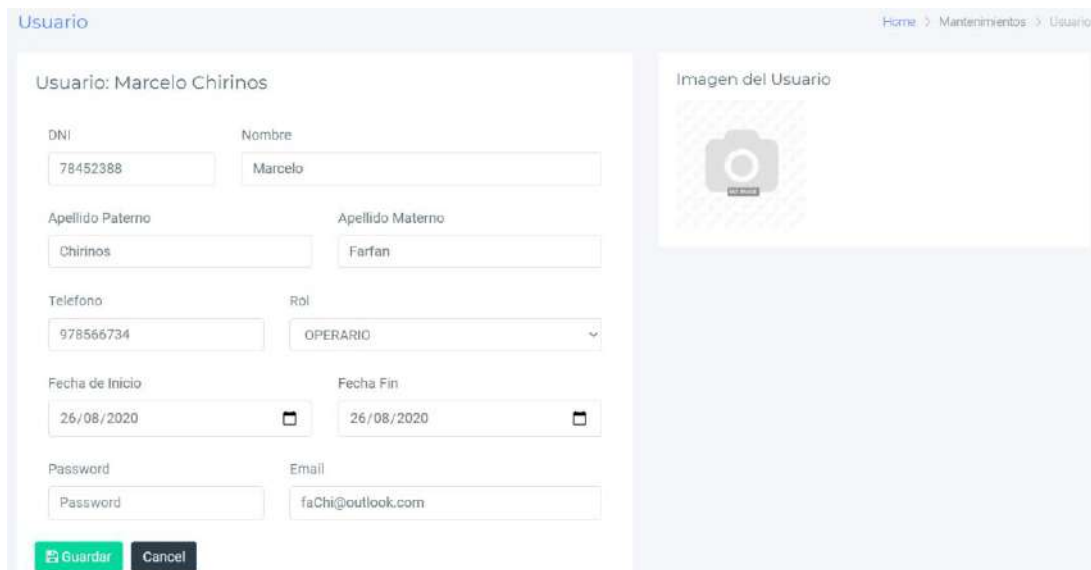


Figura 17 captura de pantalla de la creación de un usuario

Elaboración propia

### 3.1.5. Sprint 2

#### 3.1.5.1. Planteamiento del Sprint

El Sprint dos tendrá una duración de cuatro semanas, para este sprint pasan las historias de usuarios que no se completaron el en sprint 1 que son: N° 3 y 1, además se añade la historia de usuario N° 2. Se desarrollará el CREAD de proceso, modalidad, carrera, requisitos, modalidad-



proceso, detalleModPro. Las tareas que se asignaron a cada miembro del equipo son las siguientes:

- a) Alberto Cano: FrontEnd del Sistema
- b) Esperanza Wiesse: BackEnd del sistema

### 3.1.5.2. Sprint backlog

N°	Historias de usuario	pesos	prioridad
3	Yo como jefe de la oficina de admisión quiero que el sistema me permita gestionar el proceso, modalidad, carrera y requisitos de manera conjunta, para la apertura de un proceso de admisión	13	3
1	Yo como jefe de la oficina de admisión quiero que el sistema me permita insertar, activar y desactivar procesos y modalidades, para poder gestionar los procesos de admisión	5	2
2	Yo como jefe de la oficina de admisión quiero que el sistema me permita insertar, activar y desactivar carreras y requisitos, para poder gestionar los procesos de admisión	8	2
	total	26	

Tabla 5 historia de usuarios

Elaboración propia

### 3.1.5.3. Daily meeting

En este segundo sprint lo que se hizo en el FrontEnd es desarrollará el CREAD de proceso, modalidad, carrera, requisitos, modalidadProceso, detalleModPro, los problemas que se afrontaron fueron especialmente en el recurso modalidadProceso que el select no jalaba los datos de modalidad y en el recurso detalleModPro el multi select se demoró varios días al no poder visualizarse el select con los check box. En el backEnd se realizó también el CREAD de proceso, modalidad, carrera, requisitos,





modalidadProceso, detalleModPro, los problemas que se afrontaron fueron que no se podía extraer el nombre de las carreras en un arreglo aparte al hacer la búsqueda de carreras por los ids. En el anexo 2 se encuentra detallado el avance día a día del Daily Meeting.

Para la gestión de las historias de usuarios del presente sprint y de los siguientes se utilizó Trello, para esto el tablero se dividió en 4 partes, que son las siguientes:

- e) Definición
- f) En proceso de desarrollo
- g) Elaboración de la documentación, y
- h) La aceptación.

Los CheckLists de Trello para este sprint son los siguientes:

- a) Yo como jefe de la oficina de admisión quiero que el sistema me permita gestionar el proceso, modalidad, carrera y requisitos de manera conjunta, para la apertura de un proceso de admisión.

- checkList del FrontEnd

<input type="checkbox"/> creación del servicio de proceso	<input type="checkbox"/> creación del servicio de carrera
<input type="checkbox"/> mostrar la tabla de datos de proceso	<input type="checkbox"/> mostrar la tabla de datos de carrera
<input type="checkbox"/> creación de un proceso	<input type="checkbox"/> creación de una carrera
<input type="checkbox"/> actualización de un proceso	<input type="checkbox"/> actualización de una carrera
<input type="checkbox"/> creación del servicio de modalidad	<input type="checkbox"/> creación del servicio de requisito
<input type="checkbox"/> mostrar la tabla de datos de modalidad	<input type="checkbox"/> mostrar la tabla de datos de requisito
<input type="checkbox"/> creación de una modalidad	<input type="checkbox"/> creación de un requisito
<input type="checkbox"/> actualización de una modalidad	<input type="checkbox"/> actualización de un requisito

Figura 18 check list

Elaboración propia

- checkList del Backend



<input type="checkbox"/> creación de la ruta para proceso	<input type="checkbox"/> creación de la ruta para carrera
<input checked="" type="checkbox"/> petición get (obtiene todos los datos de la tabla procesos)	<input type="checkbox"/> petición get (obtiene todos los datos de la tabla carrera)
<input type="checkbox"/> petición post (creación de un proceso)	<input type="checkbox"/> petición post (creación de una carrera)
<input type="checkbox"/> petición get (de un solo proceso)	<input type="checkbox"/> petición get (de una sola carrera)
<input type="checkbox"/> petición put (actualización de un proceso)	<input type="checkbox"/> petición put (actualización de una carrera)
<input type="checkbox"/> creación de la ruta para modalidad	<input type="checkbox"/> creación de la ruta para requisito
<input type="checkbox"/> petición get (obtiene todos los datos de la tabla modalidad)	<input type="checkbox"/> petición get (obtiene todos los datos de la tabla requisito)
<input type="checkbox"/> petición post (creación de una modalidad)	<input type="checkbox"/> petición post (creación de un requisito)
<input type="checkbox"/> petición get (de una sola modalidad)	<input type="checkbox"/> petición get (de un solo requisito)
<input type="checkbox"/> petición put (actualización de una modalidad)	<input type="checkbox"/> petición put (actualización de un requisito)

Figura 19 check list

Elaboración propia

b) Yo como jefe de la oficina de admisión quiero que el sistema me permita insertar, activar y desactivar procesos y modalidades, para poder gestionar los procesos de admisión.

- checkList del FrontEnd

<input type="checkbox"/> activación y desactivación de un proceso
<input type="checkbox"/> activación y desactivación de una modalidad

Figura 20 check list

Elaboración propia

- checkList del BackEnd

<input type="checkbox"/> petición put (activación y desactivación de un proceso)
<input type="checkbox"/> petición put (activación y desactivación de una modalidad)

Figura 21 check list

Elaboración propia

c) Yo como jefe de la oficina de admisión de admisión quiero que el sistema me permita insertar, activar y desactivar carreras y requisitos, para poder gestionar los procesos de admisión.

- checkList del FrontEnd



- activación y desactivación de una carrera
- activación y desactivación de un requisito
- creación del servicio de modalidad-proceso
- mostrar la tabla de datos de modalidad-proceso
- creación de una modalidad-proceso
- actualización de una modalidad-proceso
- activación y desactivación de una modalidad-proceso
- creación del servicio de detalleModPro
- mostrar la tabla de datos de detalleModPro
- creación de un detalleModPro
- actualización de un detalleModPro

*Figura 22 check list*

Elaboración propia

- **checkList del BackEnd**

- petición put (activación y desactivación de una modalidad)
- petición put (activación y desactivación de un requisito)
- creación de la ruta para modalidad-proceso
- petición get (obtiene todos los datos de la tabla modalidad-proceso)
- petición post (creación de una modalidad-proceso)
- petición get (de un sola modalidad-proceso)
- petición put (actualización de una modalidad-proceso)
- petición put (activación y desactivación de una modalidad-proceso)
- creación de la ruta para detalleModPro
- petición get (obtiene todos los datos de la tabla detalleModPro)
- petición post (creación de un detalleModPro)
- petición get (de una solo detalleModPro)
- petición put (actualización de un detalleModPro)

*Figura 23 check list*

Elaboración propia

#### **3.1.5.4. Burn down chard**

En el siguiente grafico la línea negra representa el avance esperado, mientras que la línea roja el avance real del equipo; cómo se puede apreciar en el grafico el segundo sprint fue exitoso, se completó la historia de usuario N° 3 ,1 y 2.



Figura 24 burn down chart

Elaboración propia

### 3.1.5.5. Retrospectiva del sprint

- ¿Qué ha funcionado bien en este sprint?  
Hemos avanzado a buen ritmo en la programación
- ¿Cuáles con las cosas que se pueden mejorar para el siguiente sprint?  
Tener más conocimiento acerca de las tecnologías usadas
- ¿Qué inconvenientes hemos tenido para poder progresar correctamente en este sprint?  
Que no hemos podido solucionar relativamente rápido los problemas que hemos encontrado
- Recomendaciones para el siguiente sprint  
Verificar bien la documentación de las librerías usadas, buscar ejemplos o en foros nuestros errores y no dejar pasar el tiempo.

### 3.1.5.6. Despliegue de la aplicación den Heroku

En las siguientes imágenes se puede apreciar las capturas de pantalla de las historias de usuario 3,1 y 2 desplegadas en Heroku.



Mantenimiento de procesos Home > Mantenimientos > Mantenimiento de procesos

[+ Crear Proceso](#)

Procesos Registrados (7)

n°	Proceso		
1	2009-I		inactivo
2	2020-II		activo
3	2009-II		inactivo
4	2010-I		inactivo
5	2010-II		inactivo
6	2011-I		inactivo
7	2011-II		inactivo

Figura 25 mantenimiento de procesos

Elaboración propia

Proceso

Proceso: 2020-II

Nombre Proceso

[Guardar](#) [Cancel](#)

Figura 26 creación de un proceso

Elaboración propia

Mantenimiento de modalidades Home > Mantenimientos > Mantenimiento de modalidades

[+ Crear Modalidad](#)

Modalidades Registradas (2)

n°	Modalidad		
1	ordinario		activo
2	primera oportunidad		activo

< Anterior 1 Siguiente >

Figura 27 mantenimiento de las modalidades

Elaboración propia



Modalidad

Modalidad: ordinario

Nombre Modalidad

ordinario

Guardar Cancel

Figura 28 creación de una modalidad

Elaboración propia

Mantenimiento de carreras

Home > Mantenimientos > Mantenimiento de carreras

+ Crear Carrera

Carreras Registrados (3)

n°	Carrera		
1	derecho		activo
2	contabilidad		activo
3	administracion		activo

< Anterior 1 Siguiente >

Figura 29 mantenimiento de las carreras

Elaboración propia

Carrera

Carrera: administracion

Nombre Carrera

administracion

Guardar Cancelar

Figura 30 creación de una carrera

Elaboración propia

### 3.1.6. Sprint 3

#### 3.1.6.1. Planteamiento del Sprint

El Sprint tres tendrá una duración de cuatro semanas, para este sprint tiene las historias de usuarios N° 5, 6. Se



desarrollará la preinscripción y la inscripción del postulante. Las tareas que se asignaron a cada miembro del equipo son las siguientes:

- a) Alberto Cano: FrontEnd del Sistema
- b) Esperanza Wiese: BackEnd del sistema

### 3.1.6.2. Sprint backlog

N°	Historias de usuario	pesos	prioridad
5	Yo como postulante quiero poder realizar mi inscripción por internet, para entonces no tener que ir a hacer una larga cola para en la universidad para mi inscripción	8	1
6	Yo como operario quiero poder verificar que los datos del postulante en el sistema estén correctos, para entonces no tener problemas después al procesar la información	5	2
	total	13	

Tabla 6 historias de usuario

Elaboración propia

### 3.1.6.3. Daily meeting

En este tercer sprint lo que se hizo en el FrontEnd es la preinscripción del postulante, en el que antes de hacer la preinscripción, el postulante accede a una página donde se le indica todos los pasos del proceso de inscripción, dentro de la preinscripción se genera una tabla con la relacion de todos los postulantes, dicha tabla tiene una barra de búsqueda de postulantes por nombres y apellidos, luego se pasa a la validación de la preinscripción donde permite exportar a PDF el formulario, las dificultades que se enfrentaron fueron: el select no mostraba el nombre de la modalidad; luego con los dos select de las opciones de las carreras, que al tener doble ngFor se repetían los select tantas veces como número de carreras habían sido habilitadas en el sistema. En el backEnd se implementó la



búsqueda de las preinscripciones por nombre y apellido; también que la preinscripción almacene la fecha y hora actual de dicho acto y no el del servidor. En el anexo 3 se encuentra detallado el avance día a día del Daily Meeting.

Para la gestión de las historias de usuarios del presente sprint y de los siguientes se utilizó Trello, para esto el tablero se dividió en 4 partes, que son las siguientes:

- a) Definición
- b) En proceso de desarrollo
- c) Elaboración de la documentación, y
- d) La aceptación.

Los CheckLists de Trello para este sprint son los siguientes:

- a) Yo como postulante quiero poder realizar mi inscripción por internet, para entonces no tener que ir a hacer una larga cola para en la universidad para mi inscripción.

- checkList del FrontEnd

- creación del servicio de pre-inscripcion
- creación de la pre-inscripcion
- creación de una pagina de explicación del procedimiento de pre-inscripcion
- mostrar la tabla de datos de pre-inscripcion
- búsqueda de la pre-inscripcion

*Figura 31 check list*

Elaboración propia

- checkList del BackEnd

- creación de la ruta para de pre-inscripcion
- petición get (obtiene todos los datos de la tabla pre-inscripcion)
- petición post (creación de una pre-inscripcion)
- petición get (de un sola pre-inscripcion)
- petición put (actualización de una persona)
- búsqueda de la pre-inscripcion

*Figura 32 check list*





Elaboración propia

b) Yo como operario quiero poder verificar que los datos del postulante en el sistema estén correctos, para entonces no tener problemas después al procesar la información.

- checkList del FrontEnd

<input type="checkbox"/>	creación del servicio de inscripción
<input type="checkbox"/>	mostrar la tabla de datos de inscripción
<input type="checkbox"/>	creación de un inscripción
<input type="checkbox"/>	actualización de una inscripción
<input type="checkbox"/>	búsqueda de la inscripción por nombre o apellidos del postulante

*Figura 33 check list*

Elaboración propia

- checkList del BackEnd

<input type="checkbox"/>	creación de la ruta para de inscripción
<input type="checkbox"/>	petición get (obtiene todos los datos de la tabla inscripción)
<input type="checkbox"/>	petición post (validación de una pre-inscripción que genera la creación de la ruta para de inscripción)
<input type="checkbox"/>	petición get (de un sola inscripción)
<input type="checkbox"/>	búsqueda de la pre-inscripción

*Figura 34 check list*

Elaboración propia

#### 3.1.6.4. Burn down chard

En el siguiente grafico la línea negra representa el avance esperado, mientras que la línea roja el avance real del equipo; cómo se puede apreciar en el grafico el tercer sprint fue exitoso, se completó la historia de usuario N° 5 y 6.



Figura 35 burn down chard

Elaboración propia

### 3.1.6.5. Retrospectiva del sprint

a) ¿Qué ha funcionado bien en este sprint?

Que hemos podido cumplir con el plazo establecidos

b) ¿Cuáles con las cosas que se pueden mejorar para el siguiente sprint?

Debemos reaccionar de manera más acertada ante situaciones, como la que no avanzábamos durante varios días

c) ¿Qué inconvenientes hemos tenido para poder progresar correctamente en este sprint?

No conocemos a profundidad todas las directivas de angular y lo que se puede hacer con ellas

d) Recomendaciones para el siguiente sprint

Conocer más a profundidad angular

### 3.1.6.6. Despliegue de la aplicación den Heroku

En las siguientes imágenes se puede apreciar las capturas de pantalla de las historias de usuario 5 y 6 desplegadas en Heroku.



The screenshot shows a web interface for the admission process. At the top, there is a header with the university logo and name on the left, and 'Repositorio Digital' on the right. Below this is a light purple horizontal bar. The main content area is titled 'Admision 2019-II'. On the left side, there is a box with the text '¡Inscríbete ya!' and a red button labeled 'Pre-inscripción'. Below this, a list of dates and times is provided: 'INSCRIPCIONES: 3 de junio al 24 agosto', 'EXAMEN DE ADMISION: 25 de agosto - 09:00', 'COSTO DEL EXAMEN: S/ 200.00', 'RESULTADOS DE EXAMEN: 25 de agosto (14:30)', 'MATRÍCULAS: 25 al 31 de agosto', and 'REZAGADOS: 2 al 14 de septiembre (Con 10 % de incremento)'. On the right side, there is a section titled 'Admision 2019-II' with a sub-header 'A continuación te brindamos los pasos a seguir para el proceso de inscripción al examen de admisión:'. This section is divided into two parts: 'Pre-inscripción' and 'Inscripción'. 'Pre-inscripción' is noted as being done on the website and includes a 5-step list: 1. seleccionar la modalidad y carreras, 2. ingresar tus datos personales, 3. ingresar tu información académica, 4. hacer clic en enviar, and 5. se le mostrará los requisitos que debe de presentar según su modalidad. 'Inscripción' is noted as being done in the Admissions Office and includes a 5-step list: 1. entregar todos los requisitos correspondientes a la modalidad que seleccionó, 2. entregar el voucher de pago por derechos de Examen Admisión, 3. verificación de los datos de la Pre-inscripción, 4. verificación de los requisitos, and 5. entrega de la constancia de inscripción.

Figura 36 interfaz de información para el postulante

Elaboración propia



Pre-Inscripcion 2019-II

Todos los campos que tienen un asterisco (\*) son obligatorios

**1** Modalidad y Carreras

Proceso: 2020-II | Modalidad\*: Seleccione Modalidad

Opcion de carrera 1\*: Seleccione carrera | Opcion de carrera 2\*: Seleccione carrera

**2** Datos Personales:

Tipo de documento de identidad\*: Seleccione tipo doc. identidad | Numero de documento de identidad\*

Nombre\* | Apellido paterno\*

Apellido materno\* | Fecha de nacimiento\* (dd/mm/aaaa)

Edad\* | Sexo\*: Seleccione su sexo

Estado civil\*: Seleccione su estado civil | Email\*

Telefono\*

Datos del domicilio del postulante

Departamento\*: Seleccione un departamento | Distrito\*: Seleccione un distrito

Provincia\*: Seleccione un provincia | Direccion del domicilio\*

**3** Informacion Academica

Tipo de IE\* (Institucion educativa): Seleccione | Nombre de la IE\*

Mes y Año de egreso de la IE\* (--- de ---)

Figura 37 formulario de preinscripción

Elaboración propia



The screenshot shows a web form titled "Pre-inscripción" with the following sections and fields:

- Codigo Preinscripcion:** 5F3457e4255c911e46448d4
- Modalidad y Carreras:**
  - Proceso: 2020-II
  - Modalidad: primera oportunidad
  - Carrera 1: derecho
  - Carrera 2: contabilidad
- Pago por derechos de examen de admision:** n° de transaccion (empty)
- Datos de domicilio:**
  - Departamento: cusco
  - Provincia: espinar
  - Distrito: wanchac
  - Dirección: mallacaj gamarra
- Datos del postulante:**
  - Tipo de documento de identidad: DNI
  - n° de documento de identidad: 12345678
  - Nombre: Ruti
  - Apellido paterno: wiesse
  - Apellido materno: gutierrez
  - Fecha de nacimiento: 08/12/2002
  - Edad: 17
  - Sexo: FEMENINO
  - Estado civil: SOLTERO(A)
  - Email: esifto@gmail.com
  - Telefono: 912342145
- Informacion academica:**
  - Tipo de institucion educativa: PARTICULAR
  - Nombre de la institucion educativa: galleo
  - Mes y Año de egreso: agosto de 2020

Buttons: Guardar, Cancelar

Figura 38 validación de los datos de la preinscripción

Elaboración propia

### 3.1.7. Sprint 4

#### 3.1.7.1. Planteamiento del Sprint

El Sprint cuarto tendrá una duración de cuatro semanas, para este sprint tiene las historias de usuarios N° 7, 8, 9 y 10. Se desarrollará la preinscripción y la inscripción del postulante. Las tareas que se asignaron a cada miembro del equipo son las siguientes:

- a) Alberto Cano: FrontEnd del Sistema
- b) Esperanza Wiesse: BackEnd del sistema

#### 3.1.7.2. Sprint backlog

N°	Historias de usuario	pesos	prioridad
7	Yo como operario quiero poder registrar fácilmente que documentos está entregando el postulante, para entonces hacer más rápida la inscripción de los	5	3



	postulantes y para tener registrado que documentos entrego el postulante al momento de su inscripción		
8	Yo como postulante quiero poder recibir una constancia de mi inscripción al Examen de Admisión, con la información de la inscripción, para entonces poder estar seguro de que realice mi inscripción de manera exitosa.	5	4
9	Yo como director de la oficina de admisión quiero que el postulante sea informado a través del sistema, el plazo que tiene para poder regularizar sus documentos, para entonces asegurarme de que fueron notificados de la fecha límite de regularización de documentos.	3	1
10	Yo como director de la oficina de admisión quiero poder verificar cuantos postulantes hay en total en un determinado tiempo del proceso de admisión y cuantos postulantes hay por carreras, para entonces poder hacer un control constante de la cantidad de postulantes que se está recibiendo en general y por carreras.	13	2
	total	26	

Tabla 7 historias de usuario

Elaboración propia

### 3.1.7.3. Daily meeting

En este primer sprint lo que se hizo en el FrontEnd el CRUAD de reqMod, la registro de los requisitos que está entregando el postulante, la emisión de la constancia de inscripción con la notificación de la fecha límite para realizar las regularizaciones de los documentos, el reporte de cuantos postulantes ya inscritos y cuantos por carreras; uno de los problemas que se enfrento es, que no se podía pasar el id de la inscripción a la interfaz de entrega de registró de requisitos, el que detuvo el proceso de desarrollo por varios



días. En el backEnd se hizo la creación de los componen relacionados a la expedición de la constancia, las dificultada que se encuentra es al momento de hacer la creación de la constancia ya que primero se debe de recibir el id de la preinscripción para generar la constancia de la inscripción.

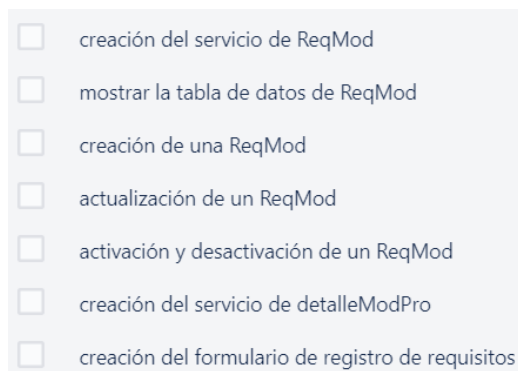
Para la gestión de las historias de usuarios del presente sprint y de los siguientes se utilizó Trello, para esto el tablero se dividió en 4 partes, que son las siguientes:

- a) Definición
- b) En proceso de desarrollo
- c) Elaboración de la documentación, y
- d) La aceptación.

Los CheckLists de Trello para este sprint son los siguientes:

- a) Yo como operario quiero poder registrar fácilmente que documentos está entregando el postulante, para entonces hacer más rápida la inscripción de los postulantes y para tener registrado que documentos entrego el postulante al momento de su inscripción.

- checkList del FrontEnd



*Figura 39 check list*

Elaboración propia

- checkList del BackEnd



- creación de la ruta para reqMod
- petición get (obtiene todos los datos de la tabla reqMod)
- petición post (creación de un reqMod)
- petición get (de un solo reqMod)
- petición put (actualización de un reqMod)
- petición put (activación y desactivación de un reqMod)
- petición get (obtiene todos los requisitos por postulante)
- petición post (creación de un registro de requisitos)

Figura 40 check list

Elaboración propia

b) Yo como postulante quiero poder recibir una constancia de mi inscripción al Examen de Admisión, con la información de la inscripción, para entonces poder estar seguro de que realice mi inscripción de manera exitosa.

- checkList del FrontEnd

- generación de la constancia de registro de requisitos y de inscripción

Figura 41 check list

Elaboración propia

- checkList del BackEnd

- petición get (obtiene todas las constancias de los postulante)
- petición post (creación de la constancia)

Figura 42 check list

Elaboración propia

c) Yo como director de la oficina de admisión quiero que el postulante sea informado a través del sistema el plazo que tiene para poder regularizar sus documentos, para entonces asegurarme de que fueron notificados de la fecha límite de regularización de documentos.

- checkList del FrontEnd





- agregara en la parte de bajo de la constancia el numero de días que tiene el postulante para regularizar sus papeles

Figura 43 check list

Elaboración propia

d) Yo como director de la oficina de admisión quiero poder verificar cuantos postulantes hay en total en un determinado tiempo del proceso de admisión y cuantos postulantes hay por carreras, para entonces poder hacer un control constante de la cantidad de postulantes que se está recibiendo en general y por carreras.

- checkList del FrontEnd

- creación del servicio de reespostes
- mostrar la tabla de datos de las cantidad de postulantes por carrera
- mostrar el total de inscritos

Figura 44 check list

Elaboración propia

- checkList del BackEnd

- petición get (obtiene todos los datos de la tabla inscripción contabilizando el numero de registros )
- petición get (de la cantidad de inscritos por carreras)

Figura 45 check list

Elaboración propia

#### 3.1.7.4. Burn down chard

En el siguiente grafico la línea negra representa el avance esperado, mientras que la línea roja el avance real del equipo; como se puede apreciar en el gráfico, el cuarto sprint fue exitoso, se completó la historia de usuario N° 7, 8, 9 y 10.

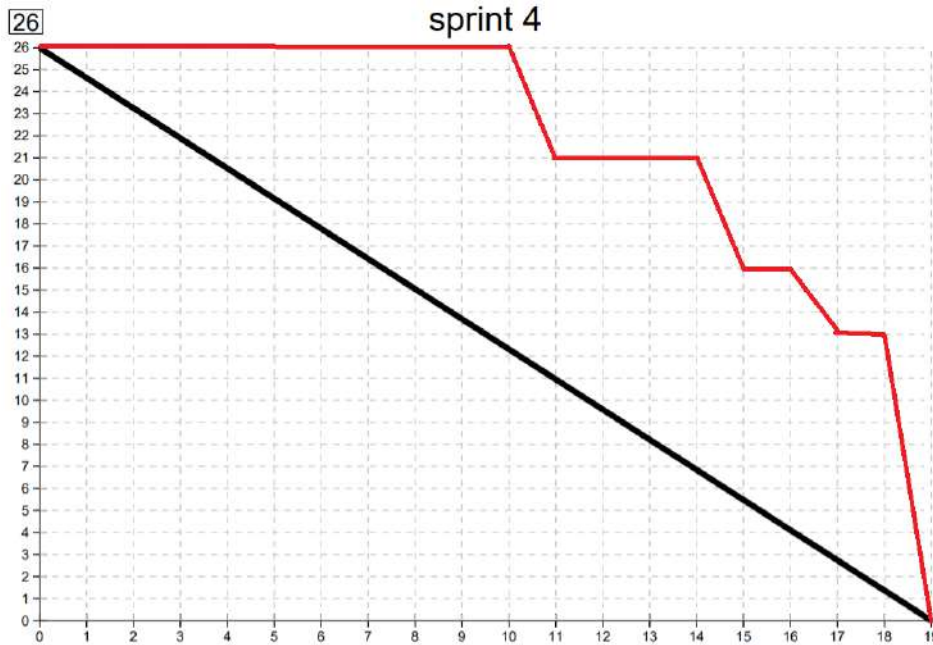


Figura 46 burn down chard

Elaboración propia

### 3.1.7.5. Retrospectiva del sprint

- ¿Qué ha funcionado bien en este sprint?  
Que ya tenemos implementadas las tablas a tiempo
- ¿Cuáles con las cosas que se pueden mejorar para el siguiente sprint?  
Avanzar más rápido con la programación de las historias de usuarios
- ¿Qué inconvenientes hemos tenido para poder progresar correctamente en este sprint?  
Que no tenemos algunos de los conocimientos necesarios para poder desarrollar el sistema
- Recomendaciones para el siguiente sprint: debemos medir mejor nuestro tiempo

### 3.1.7.6. Despliegue de la aplicación den Heroku

En las siguientes imágenes se puede apreciar las capturas de pantalla de las historias de usuario:7, 8, 9 y 10 desplegadas en Heroku



Detalle Inscripcion

Postulante: Rubi wiesse gutierrez

Codigo de Inscripcion:

5f345893255c9b11e4a4ad6e

Requisitos Entregados

DNI

certificado de estudios de 1° a 5° de se...

Figura 47 registro de requisitos

Elaboración propia

Datos de Inscripcion del Postulante

Codigo de Inscripcion: 5f345893255c9b11e4a4ad6e      Numero de Transaccion: 12344

Datos del Postulante

Nombre: Rubi	Ap. Paterno: wiesse
Ap. Materno: gutierrez	Edad: 17
Departamento: cusco	Provincia: escorse
Districto: wanchac	Direccion: maliscal gamama
Teléfono: 912242145	Estado Civil: SOLTERO(A)

Informacion Academica

Tipo I.E.: PARTICULAR	Nombre I.E.: galileo
-----------------------	----------------------

Datos de Postulacion

Proceso: 2019-I	Mediudad: Ordinario
1° Opcion de Carrera: derecho	2° Opcion de Carrera: contabilidad

Rubi wiesse gutierrez      Fecha: Wed Aug 26 2020 06:56:12 GMT-0500 (hora estandar de Peru)

Imprimir      1 hoja de papel

Destino: EPSON L210 Series

Paginas: Todo

Copias: 1

Color: Color

Más ajustes

Figura 48 formato de inscripción

Elaboración propia

### 3.1.8. Sprint 5

#### 3.1.8.1. Planteamiento del Sprint

El Sprint cinco tendrá una duración de 9 días, para esto se tiene previsto completar la historia de usuario N° 11. Se desarrollará la parte del reporte concerniente a la entrega de



requisitos de los postulantes. Las tareas que se asignaron a cada miembro del equipo son las siguientes:

- a) Alberto Cano: FrontEnd del Sistema
- b) Esperanza Wiesse: BackEnd del sistema

### 3.1.8.2. Sprint backlog

N°	Historias de usuario	pesos	prioridad
11	Yo como director de la oficina de admisión quiero poder saber cuántos postulantes entregaron completos sus documentos y cuantos no, para poder saber la cantidad de postulantes deben de regularizar sus documentos y la lista de nombres completos de los postulantes que entregaron completo sus documentos.	8	3
	total	8	

*Tabla 8 historia de usuario*

Elaboración propia

### 3.1.8.3. Daily meeting

En este quinto sprint lo que se hizo en el FrontEnd se realizó el reporte de los postulantes que entregaron completo sus documentos y cuantos son. En el backEnd se realizó las peticiones necesarias para poder dar el reporte, se tu una dificultad en la realización del filtro de que postulantes entregaron sus documentos completos. En el anexo 5 se encuentra detallado el avance día a día del Daily Meeting.

Para la gestión de las historias de usuarios del presente sprint y de los siguientes se utilizó Trello, para esto el tablero se dividió en 4 partes, que son las siguientes:

- a) Definición
- b) En proceso de desarrollo
- c) Elaboración de la documentación, y
- d) La aceptación.



Luego para cada historia de usuario se elaboró una pequeña descripción y un checkList, que mostramos a continuación:

- a) Yo como director de la oficina de admisión quiero poder saber cuántos postulantes entregaron completos sus documentos y cuantos no, para poder saber la cantidad de postulantes deben de regularizar sus documentos y la lista de nombres completos de los postulantes que entregaron completo sus documentos.

- checkList del FrontEnd

- creación del servicio de para el reporte de requisitos
- creación de la tabla para ver los nombres de los postulantes

*Figura 49 check list*

Elaboración propia

- checkList del BackEnd

- petición get (obtiene todos los postulante que entregaron requisitos completos)
- petición get (contabiliza el numero de postulantes que entregaron requisitos completos)

*Figura 50 check list*

Elaboración propia

#### 3.1.8.4. Burn down chard

En el siguiente grafico la línea negra representa el avance esperado, mientras que la línea roja el avance real del equipo; como se puede apreciar en el gráfico, el quinto sprint fue exitoso, se completó la historia de usuario N° 11.



Figura 51 burn down chard

Elaboración propia

### 3.1.8.5. Retrospectiva del sprint

- ¿Qué ha funcionado bien en este sprint?  
Hemos avanzado a buen ritmo en la programación
- ¿Cuáles con las cosas que se pueden mejorar para el siguiente sprint?  
Disminuir la carga de trabajo, debimos considerar el tiempo que llevaría
- ¿Qué inconvenientes hemos tenido para poder progresar correctamente en este sprint?  
No sabía exactamente cómo hacer la consulta a la base de datos
- Recomendaciones para el siguiente sprint  
Tomar en cuenta el tiempo y la carga de trabajo para estimar mejor las metas y los tiempos de desarrollo

### 3.1.8.6. Despliegue de la aplicación den Heroku

En las siguiente imagen se puede apreciar la captura de pantalla de la historia de usuario 11 desplegada en Heroku



Conteo de Inscritos

Home > Mantenimientos > Conteo de Inscritos

Total de Inscripciones Registradas (12)

Escuelas Profesionales	Total de Inscritos
Ciencias Contables y Finanzas	3
Economía y Negocios Internacionales	8
Ing. de Sistemas y Seguridad Informática	1
Turismo	1

Figura 52 reporte de n° de inscritos

Elaboración propia

### 3.2. Pruebas de usabilidad

Las pruebas de usabilidad se realizaron tomando como base los principios de usabilidad de Jakob Nielsen, ya que ésta es la más usada.

Las pruebas de usabilidad del sistema se hicieron con 2 grupos de usuarios, estos 2 grupos son:

- a) LOS USUARIOS DE PREINSCRIPCIÓN: en el que se tomaron a 2 que fueron voluntarias.
- b) LOS USUARIOS DE INSCRIPCIÓN: en el que se tomaron 2 personas de la Oficina de Admisión.

Las pruebas se realizaron por separado a cada persona, a la persona seleccionada se le asignó una tarea que debió completar haciendo usos del sistema desarrollado, el equipo de desarrollo del sistema observo a la persona que realiza la prueba sin tener algún tipo de intervención.

La evaluación de la usabilidad se realizó después de la observación de la interacción del usuario con el sistema, para indicar el grado de usabilidad se tomó la escala del 1 al 5, siendo 5 la valoración más óptima y 1 el menos óptimo; esta valoración del grado de usabilidad se realizó para cada uno de los principios de Nielsen.

LOS USUARIOS DE PREINSCRIPCIÓN:

- Persona N° 1: Postulante



N°	Principio	Escala de usabilidad
1	Visibilidad del estado del sistema	5
2	Encaje entre el sistema y el mundo real	5
3	Libertad y control por parte del usuario	4
4	Consistencia y estándares	5
5	Prevención de errores	3
6	Reconocimiento antes que recuerdo	4
7	Flexibilidad y eficiencia en el uso	5
8	Diseño estético y minimalista	4
9	Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	3
10	Ayuda y documentación	3

*Tabla 9 cuadro de usabilidad de la preinscripción*

Elaboración propia

- Persona N° 2: Postulante

N°	Principio	Escala de usabilidad
1	Visibilidad del estado del sistema	4
2	Encaje entre el sistema y el mundo real	5
3	Libertad y control por parte del usuario	5
4	Consistencia y estándares	5
5	Prevención de errores	4
6	Reconocimiento antes que recuerdo	4
7	Flexibilidad y eficiencia en el uso	5
8	Diseño estético y minimalista	4
9	Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	3
10	Ayuda y documentación	3

*Tabla 10 cuadro de usabilidad de la preinscripción*

Elaboración propia





LOS USUARIOS DE INSCRIPCIÓN:

- Persona N° 3: Personal de la Oficina de Admisión

N°	Principio	Escala de usabilidad
1	Visibilidad del estado del sistema	4
2	Encaje entre el sistema y el mundo real	5
3	Libertad y control por parte del usuario	5
4	Consistencia y estándares	5
5	Prevención de errores	4
6	Reconocimiento antes que recuerdo	4
7	Flexibilidad y eficiencia en el uso	5
8	Diseño estético y minimalista	5
9	Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	5
10	Ayuda y documentación	2

*Tabla 11 cuadro de usabilidad de la inscripción*

Elaboración propia

- Persona N° 4: jefe de la Oficina de Informática (Inscripción y mantenimiento del sistema)

N°	Principio	Escala de usabilidad
1	Visibilidad del estado del sistema	5
2	Encaje entre el sistema y el mundo real	5
3	Libertad y control por parte del usuario	4
4	Consistencia y estándares	5
5	Prevención de errores	4
6	Reconocimiento antes que recuerdo	5
7	Flexibilidad y eficiencia en el uso	4
8	Diseño estético y minimalista	5
9	Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	5



10	Ayuda y documentación	2
----	-----------------------	---

*Tabla 12 cuadro de usabilidad de la inscripción*

Elaboración propia

El siguiente cuadro contiene los promedios de las 4 pruebas realizadas:

N°	Principio	Escala de usabilidad
1	Visibilidad del estado del sistema	4.5
2	Encaje entre el sistema y el mundo real	5
3	Libertad y control por parte del usuario	4.5
4	Consistencia y estándares	4.7
5	Prevención de errores	3.7
6	Reconocimiento antes que recuerdo	4.25
7	Flexibilidad y eficiencia en el uso	4.7
8	Diseño estético y minimalista	4.5
9	Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	4
10	Ayuda y documentación	2.5

*Tabla 13 promedio de las pruebas de usabilidad*

Elaboración propia

Como se puede observar, tanto en las tablas correspondientes a la preinscripción, como de inscripción, los resultados de usabilidad son positivos. En los principios del 1 al 9 se ha obtenido resultados bastante satisfactorios, en el caso del principio N° 10 se ha obtenido un puntaje relativamente bajo, porque en la preinscripción al estar bien definido los pasos a seguir no ha sido necesario realizar algún tipo de documentación extra para que el postulante pueda consultar, al momento de realizar la preinscripción, en el caso de la inscripción y la parte de mantenimiento del sistema, aunque es sistema no cuenta con un apartado de documentación, se elaboró un manual de funcionamiento del sistema que se encuentra en el anexo 8 en cumplimiento de los estándares de algunas entidades educativas superiores.



### 3.3. Indicador 6 del componente 1.5

El sistema desarrollado se aboca a la tarea específica de inscribir a los postulantes en un proceso de admisión, para el cumplimiento de esta tarea se realiza una serie de pasos que se detallan a continuación:

Del lado del jefe de la Oficina de Admisión:

- a) Debe crear el Proceso de Admisión.
- b) En el apartado “Modalidad Proceso”, se debe de asignar uno por uno las Modalidades al Proceso de Admisión.
- c) En “Detalle Modalidad Proceso” debe de seleccionar todas las Carreras que corresponden a cada “Proceso Modalidad”
- d) En “Requisito Modalidad” se debe de asignar los Requisitos de acuerdo con cada modalidad de ingreso.
- e) En el apartado de repostes, se puede ver el número de inscritos por carrera y el total de inscritos al examen de admisión de la institución educativa superior, en el apartado de “Inscripción” existe un botón con el cual se puede descargar un documento de Excel que contiene toda la información de inscripción al examen de admisión de los postulantes.

Una vez terminado un Proceso de Admisión se debe desactivar cada registro para dar paso a un nuevo Proceso de Admisión.

Del lado del postulante:

- a) Al ingresar a la página del Examen de Admisión, primero visualizara en el lado derecho de la página una descripción paso a paso del Proceso de Inscripción al Examen de Admisión, en el lado izquierdo podrá ver información como las fechas, pagos, y demás información referida al Examen de Admisión y un botón que le redireccionara a la preinscripción.
- b) En la preinscripción deberá de llenar sus datos como: las carreras y la modalidad por la que postula, sus datos personales y su información académica.

Del lado del operador:



- a) Selecciona la preinscripción del postulante al que tiene que atender.
- b) Verifica que los datos del postulante estén correctos.
- c) Ingresa en el sistema los requisitos que entrega el postulante.
- d) Genera e imprime la constancia correspondiente.

Como se puede ver, en base a la descripción brindada acerca del sistema, el sistema apoya al cumplimiento del Indicador 6 al brindar y guardar la información sobre el Proceso de Inscripción a un Examen de Admisión.

### 3.4. Costo de desarrollo de la aplicación

Dentro del cálculo de los costos de desarrollo de la aplicación, se debe tener en cuenta las siguientes consideraciones:

Primero: las tecnologías usadas para el desarrollo del software son open source.

Segundo: el despliegue de la aplicación en la nube haciendo uso de Heroku se está tomando el costo mínimo de pago que ofrece dicha plataforma.

Tercero: para estimar el precio del desarrollo del software, se está teniendo en cuenta, que el proyecto se desarrolló en 5 meses y se trabajó 8 horas al día (solo se trabajó los días hábiles).

<b>Costos de Desarrollo de la Aplicación</b>				
<b>Rubro</b>	<b>N° de Meses</b>	<b>Cantidad</b>	<b>Costo Unitario</b>	<b>Costo Total</b>
<b>Miembros del Equipo</b>				
Scrum members*	5	2	S/3 000.00	S/30 000.00
<b>Hardware</b>				
Depreciación o uso de las laptops para el desarrollo del sistema**	5	2	S/68.00	S/136.00
<b>Software</b>				
Visual Studio Code	5	2	S/0.00	S/0.00
Sweetalert 2	5	2	S/0.00	S/0.00
ngx-pagination	5	2	S/0.00	S/0.00
make PDF	5	2	S/0.00	S/0.00



node js	5	2	S/0.00	S/0.00
mongo DB	5	2	S/0.00	S/0.00
Angular	5	2	S/0.00	S/0.00
Express	5	2	S/0.00	S/0.00
Heroku	5	1	\$25.00	\$125.00
<b>servicios</b>				
Internet**	5	30 megas	S/65.00	S/325.00
			<b>Total</b>	<b>S/30 879.75</b>

Tabla 14 costos de desarrollo de la aplicación

Elaboración propia

\* No existe un estándar de costos de diseño y elaboración de software, tanto en la empresa privada como pública y en la SUNAT, para tomar como referencia el estándar de costos en México, que es de 17 000 pesos mexicanos, equivalente a 3 076.77 soles peruanos según (talentcom, 2019-2020).

\*\* Los precios considerados están referidos al periodo en el cual se hizo el uso.

El cambio de los costos de dólares a soles que se realizó fue tomando en cuenta un cambio de ese entonces que fue de 3.35.



## CAPÍTULO IV

### 4. Resultados

#### 4.1. Comprobación de la usabilidad del sistema

##### 4.1.1. Pruebas de usabilidad al sistema

Las pruebas de usabilidad se realizaron tomando como base los principios de usabilidad de Jakob Nielsen, ya que esta es la más usada.

Las pruebas de usabilidad del sistema se hicieron con 2 grupos de usuarios, estos 2 grupos son:

- a) LOS USUARIOS DE PREINSCRIPCIÓN: en el que se tomaron a 2 estudiantes de 5° de secundaria y 1 persona que culminó recién sus estudios secundarios.
- b) LOS USUARIOS DE INSCRIPCIÓN: en el que se tomaron 2 personas de la oficina de admisión en este caso de la Universidad Peruana Austral del Cusco, que en ese entonces estaba en pleno funcionamiento; 1 una persona ajena a la oficina de admisión (la oficina de admisión contrata por ciertos periodos personal temporal)

Las pruebas se realizaron por separado a cada persona, a la persona seleccionada se le asignó una tarea que debió completar haciendo usos del sistema desarrollado, el equipo de desarrollo del sistema observó a las personas que realizaron la prueba sin tener algún tipo de intervención, en las que al final de la evaluación se obtuvieron resultados satisfactorios.

#### 4.2. Comprobación de la prospectiva

El presente trabajo de investigación generará los siguientes beneficios:

- a) El Sistema de Información desarrollado no requiere para su uso de personal altamente capacitado, para la Oficina de Admisión, para la ejecución de funcionalidades planteadas dentro del sistema.



- b) Los datos que contendrá el sistema no estarán expuestos a pérdida o a ser afectados por problemas técnicos dentro de las entidades educativas superiores, porque los datos serán almacenados en la nube.
- c) El Sistema de Información para los Procesos de Inscripción a los Exámenes de Admisión de las entidades educativas superiores facilita la integración de datos y generación de reportes, sin necesidad del uso adicional de herramientas externas, para generar reportes.
- d) La obtención de reportes relacionados al Proceso de Inscripción a los Exámenes de Admisión, sirven como herramienta de gestión para la Oficina de Admisión.

#### **4.3. Cumplimiento de objetivos**

Para el cumplimiento de los objetivos se contó con el apoyo de la Universidad Austral del Cusco para la realización de las pruebas y la elaboración de los requerimientos; los objetivos planteados se muestran a continuación, con la explicación de cómo es que se cumplieron estos objetivos.

- a) El objetivo de desarrollar un Sistema de Información para el Proceso de Inscripción a los Exámenes de Admisión de una entidad educativa superior en Cloud Computing.

Como se puede evidenciar, en el epígrafe 3.1 y el anexo N° 6 el sistema se termina de desarrollar y es desplegado en el PaaS Heroku, actualmente se encuentra bajo la disposición y control de la Oficina de Admisión de la Universidad Peruana Austral del Cusco. Sin que este hecho impida el uso en otras instituciones educativas superiores.

Al usar la modalidad de PaaS, el sistema de información desarrollado es desplegado en la nube por un tercero que también brinda la infraestructura de TI, en este caso Heroku es el que se encarga del despliegue del sistema en la nube. Por tanto, el



desarrollo de la aplicación que ésta contiene es responsabilidad del equipo de desarrollo. El manejo de la data sobre la veracidad de La información que es ingresada al sistema es responsabilidad de la entidad educativa superior.

La API REST que se desarrolló cumple con las características indicadas en estándar:

1. PROTOCOLO CLIENTE/SERVIDOR SIN ESTADO: las peticiones HTTP del sistema desarrollado contiene toda la información que el cliente y el servidor necesitan, como se puede apreciar en la imagen.





```
137 app.put('/:id', mdAutenticacion.verificaToken, (req, res) => {
138   var id = req.params.id;
139   var body = req.body;
140
141   //busacando en la base de datos por el id
142   DetallerModPro.findById(id, (err, detalleMP) => {
143
144     if (err) {
145       return res.status(500).json({
146         ok: false,
147         mensaje: 'error al buscar detalle modalidad proceso',
148         errors: err
149       });
150     }
151
152     if (!detalleMP) {
153       return res.status(400).json({
154         ok: false,
155         mensaje: 'el detalle modalidad proceso con el id' + id + 'no existe',
156         errors: { message: 'no existe un detalle modalidad proceso con ese id' }
157       });
158     }
159
160     ModPro.findById(body.modPro, (err, modProNuevo) =>{
161       var modalidadNueva = modProNuevo.modalidad;
162       var procesoNueva = modProNuevo.proceso;
163
164       detalleMP.modalidad = modalidadNueva;
165       detalleMP.proceso = procesoNueva;
166       detalleMP.carreras = body.carreras;
167       detalleMP.modPro = body.modPro;
168       detalleMP.nombreCar = body.nombreCar;
169
170
171       detalleMP.save((err, detalleMPGuardado) => {
172
173         if (err) {
174           return res.status(400).json({
175             ok: false,
176             mensaje: 'error al actualizar detalle modalidad proceso',
177             errors: err
178           });
179         }
180
181         res.status(200).json({
182           ok: true,
183           detalleMP: detalleMPGuardado
184         });
185
186       });
187
188     });
189
190   });
191 });
192 });
193
```

Figura 53 cliente/servidor sin estado

Elaboración propia

Como ejemplo se toma la petición put de “Detalle Modalidad Proceso”, en la que se evidencia que esta petición funciona de manera independiente ya que no requiere más información que



las enviadas por el usuario final, también se puede ver que los estados son asignados dentro de la función (no requiere verificar estados previos o de otros componentes para funcionar).

2. OPERACIONES MÁS IMPORTANTES DE REST Y ESPECIFICACIONES HTTP: Como se aprecia en las imágenes (figura 54, figura 55) se hace uso de las peticiones HTTP para cada una de las operaciones que realiza el sistema.



Figura 54 peticiones HTTP

Elaboración propia



```
//=====
//          OBTENER TODOS LAS MODALIDADES PROCESOS
//=====
app.get('/', (req, res, next) => {
|

//=====
//          OBTENER TODOS LAS MODALIDADES PROCESOS ACTIVOS
//=====
app.get('/activas', (req, res, next) => {
|

//=====
//          OBTENER UN DETALLE MODALIDAD PROCESO
//=====

app.get('/:id', (req, res) => {
|   var id = req.params.id;
|

//=====
//          ACTUALIZAR DETALLE MODALIDAD PROCESO
//=====

app.put('/:id', mdAutenticacion.verificaToken, (req, res) => {
|   . . .
|

//=====
//          CREAR UN NUEVO DETALLE MODALIDAD PROCESO
//=====

app.post('/', mdAutenticacion.verificaToken, async(req, res) => {
|
```

Figura 55 peticiones HTTP

Elaboración propia

### 3. LOS OBJETOS EN REST SIEMPRE SE MANIPULAN A PARTIR DE LA URI:

- Esquema: indica el protocolo a usar se puede ver en la figura 56.
- Identificador del dominio: que es proporcionado por Heroku, figura 57.
- Ruta: que se puede ver en la figura 58.
- Consulta: que se parecía en la figura 59.



```
directConnect: true,  
baseUrl: 'https://',  
framework: 'jasmine',
```

Figura 56 definición del protocolo

Elaboración propia

```
.herokuapp.com/
```

Figura 57 dominio

Elaboración Heroku

```
//rutas  
app.use('/persona', personaRoutes);  
app.use('/login', login);  
app.use('/proceso', procesoRoutes);  
app.use('/modalidad', modalidadRoutes);  
app.use('/carrera', carreraRoutes);  
app.use('/requisito', requisitoRoutes);  
  
app.use('/modPro', modProRoutes);  
app.use('/detalleMP', detalleMPRoutes);  
app.use('/reqMod', reqModRoutes);  
  
app.use('/preInscripcion', preInscripcionRoutes);  
app.use('/inscripcion', inscripcionesRoutes);  
app.use('/detalleInscrip/:idP', detalleInscripRoutes);  
app.use('/constancia/:idI', constanciaRoutes);  
  
app.use('/resportes', repostesRoutes);  
  
app.use('/busqueda', busquedaRoutes); //busqueda  
app.use('/upload', uploadRoutes); //subir imagen  
app.use('/img', imagenesRoutes); //mostrar imagen
```

Figura 58 ruta

Elaboración propia



Figura 59 consulta

Elaboración propia

4. INTERFAZ UNIFORME: como se evidencia en epígrafe 3.1 y el Anexo 8 las interfaces del sistema tienen relación con cada solicitud HTTP que el usuario realizara.

b) El objetivo de implementar un Sistema de Información adecuado a los requerimientos de las entidades educativas superiores usando Scrum.

En el epígrafe 3.1 de este trabajo de investigación se encuentra detallado todo el desarrollo del Sistema de Información para el Proceso de Inscripción a los Exámenes de Admisión de las entidades educativas superiores haciendo uso de Scrum. Siendo 4 sprints exitosos y 1 fallido como se puede evidenciar en la constancia emitida por la Oficina que facilitó el desarrollo del sistema, como se observa en el anexo N° 6.

c) El objetivo de evaluar la usabilidad del Sistema de Información desarrollado de acuerdo con los 10 principios de Nielsen.

Como se evidencia en el epígrafe 3.2 las pruebas de usabilidad se realizaron con usuarios voluntarios, en donde los cuadros de evolución de la usabilidad del Sistema de Información indican que los resultados son favorables, así mismo esto es ratificado por el certificado de operatividad del sistema que se encuentra en el anexo N° 7, que fue expedido por la Oficina que facilitó el desarrollo del sistema, donde indica que se ha cumplido con la funcionalidad



esperada y que no presenta inconvenientes operacionales o técnicos.

- d) El objetivo planteado de generar información de apoyo al cumplimiento y sustentación del indicador 6 del componente 1.5 referido al Proceso de Admisión de las Condiciones Básicas de Calidad.

Como se puede evidenciar en el Anexo 9 el sistema desarrollado apoya al cumplimiento del indicador 6 al brindar y guardar la información sobre el Proceso de Inscripción a un Examen de Admisión y permitir que los datos de los postulantes sean descargador en un archivo Excel, para que posteriormente el jefe de la Oficina de Admisión pueda realizar un informe estadístico como lo indica la Ley Universitaria.

- e) El objetivo de plantear un esquema del desarrollo del sistema es el siguiente:

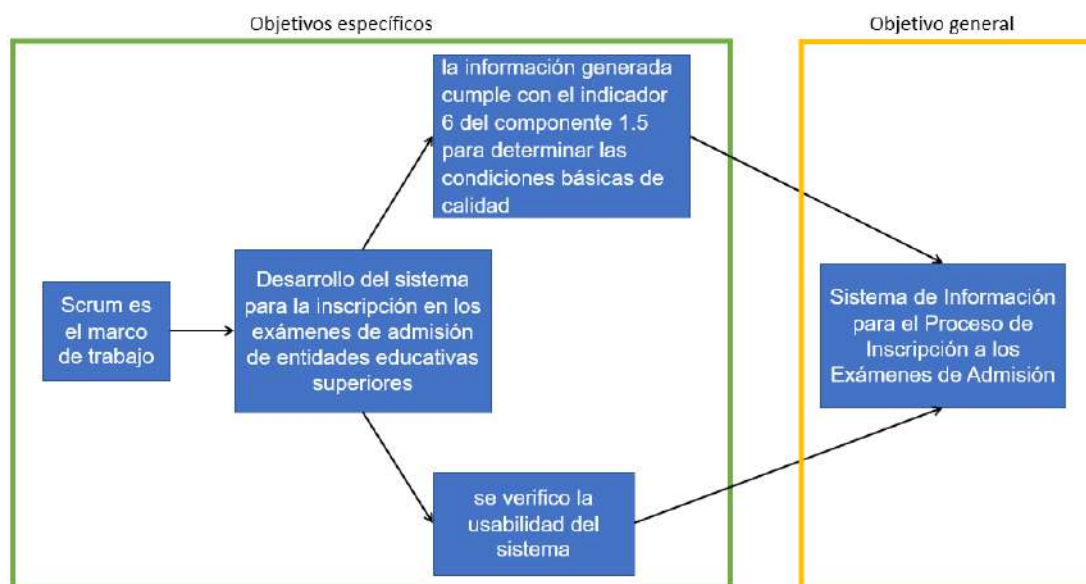


Figura 60 flujo del desarrollo del sistema

Elaboración propia



#### 4.4. Contribuciones (impacto)

Las contribuciones del presente trabajo de investigación son las siguientes:

- a) El jefe de la Oficina de Admisión tiene al alcance de manera más rápida y estandarizada un conjunto de reportes al día, como se puede apreciar en el epígrafe 3.1.2 historia de usuario N° 10 e historia de usuario N° 11.
- b) Mejorará la imagen institucional de las entidades educativas superiores, al contar con una preinscripción, y dar al postulante un comprobante de su inscripción.



## Glosario

1. **API:** Interfaz de programación de aplicaciones (Application Programming Interface).
2. **CRUAD:** Crear, leer, actualizar, activar y desactivar.
3. **HTTP:** El protocolo de transferencia de hipertexto (Hypertext Transfer Protocol).
4. **IaaS:** Infrestructure as a Service.
5. **JSON:** JavaScript Object Notation.
6. **JWK:** Json web token.
7. **MEAN:** MongoDB, Express, Angular, Node.js.
8. **NIST:** El Instituto Nacional de Estándares y Tecnología (National Institute of Standards and Technology).
9. **PaaS:** plataforma como servicio.
10. **REST:** Transferencia de estado representacional (Representational State Transfer).
11. **SaaS:** Software as a Service.
12. **URI:** identificador de recursos uniforme (uniform resource identifier).
13. **WYSIWYG:** lo que ves es lo que obtienes (What You See Is What You Get).





## Conclusiones

- 1) Scrum como framework de gestión de proyectos es eficaz, para desarrollar un sistema de información para el proceso de inscripción a los exámenes de admisión de una entidad educativa superior.
- 2) Se ha probado la usabilidad del sistema de información desarrollado de acuerdo con los 10 principios de Nielsen.
- 3) La información generada por el sistema desarrollado, es completa y cumple con los requisitos del indicador 6 del componente 1.5, para determinar las condiciones básicas de calidad en una institución educativa superior.
- 4) El flujo de la secuencia ordenada de los objetivos específicos nos conduce al desarrollo del sistema aplicado al proceso de inscripción a los exámenes de admisión de una entidad educativa superior.



### Recomendaciones

Al haber concluido el desarrollo del presente trabajo de investigación, proponemos las siguientes recomendaciones:

1. Se recomienda utilizar scrum porque permite recibir constantemente retroalimentación por parte de los usuarios, además facilita la construcción de un sistema que responda a las necesidades del usuario y que es muy amigable.
2. Es preferible utilizar Trello, un tablero virtual que facilita el seguimiento de la lista de tareas o actividades, facilitando la coordinación del trabajo en equipo al asignar una tarea cada tarea a un miembro o miembros del equipo, además indicando el plazo estimado de desarrollo.
3. Si se tiene en cuenta que existe una comunidad bastante numerosa de desarrolladores que usan Angular, es más accesible encontrar información que la comunidad ha generado.
4. Se recomienda el Stack MEAN que es un conjunto de tecnologías basadas en Javascript, que permite utilizar una sintaxis uniforme para desarrollar cualquier tipo de sistema.
5. Se recomienda el uso de Heroku para desplegar una aplicación en la nube, porque es un gestor económico, fiable y óptimo, con un plan gratuito que te permite iniciar el despliegue de tu aplicación a mayor escala.



## Referencias

- Admisión, C. d. (2019). *Prospecto de Admisión de la UAC*. Cusco: UAC.
- Admisión, C. d. (2019). *Prospecto de Admisión de la UNSAAC*. Cusco: UNSAAC.
- BBVAOPEN4U. (23 de marzo de 2016). *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos*. Obtenido de <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- campusMVP. (10 de mayo de 2017). *Las 10 principales diferencias entre AngularJS y Angular*. Obtenido de <https://www.campusmvp.es/recursos/post/las-10-principales-diferencias-entre-angularjs-y-angular.aspx>
- Cardenas, A. (24 de febrero de 2019). *¿AngularJS o Angular?* Obtenido de <https://medium.com/ngesyfirebase/angularjs-o-angular-6e27eef1ed96>
- cloudflare. (02 de diciembre de 2019). *What Is Platform-as-a-Service (PaaS)?* Obtenido de <https://www.cloudflare.com/learning/serverless/glossary/platform-as-a-service-paas/>
- Eugenia. (15 de enero de 2018). *UXTips*. Obtenido de <https://www.youtube.com/watch?v=Hp3s9QTVuUE>
- Gimenez , M. (23 de noviembre de 2017). *Análisis Heurístico para UX: evalúa la usabilidad de tu web*. Obtenido de <https://www.hiberus.com/crecemos-contigo/analisis-heuristico-para-ux-evalua-la-usabilidad-de-tu-web/>
- Google. (09 de febreo de 2020). *indicador*. Obtenido de <https://www.google.com/search?sxsrf=ACYBGNSioRN3CUb4AYd7v6Y4HeQduKkYksW:1581499685153&q=Diccionario&stick=H4sIAAAAAAAAAAONQesSoyS3w8sc9YSmFSWtOXmOU4uLzL0jNc8IMLsnMz0ssqrTiUGJLLdY1MbTkWcTKDRRPBotn5gMANTZuyTwAAAA&zx=1581499748851>
- Holmes, S. (2013). *Mongoose for Application*. Packt Publishing.
- Macedo , A. P. (2012). *Resumen de Tesis de Maestría en Educación con Orientación en Gestión Educativa*. Buenos Aires: Universidad de San Andres .
- Macias, S. (21 de marzo de 2018). *Construyendo una Web API REST segura con JSON Web Token en .NET (Parte I)*. Obtenido de <https://enmilocalfunciona.io/construyendo-una-web-api-rest-segura-con-json-web-token-en-net-parte-i/>
- Mell, P., & Timothy, G. (septiembre de 2011). *The NIST Definition of Cloud*.
- Moreno Arboleda, F. J., Quintero Rendón, J. E., & Rueda Vásquez, R. (15 de marzo de 2016). *UNA COMPARACIÓN DE RENDIMIENTO ENTRE ORACLE Y MONGODB*. Bogotá, Colombia: Universidad Militar Nueva Granada, Ciencia e Ingeniería Neogranadina.
- RedHat. (2019). *Red Hat*. Obtenido de <https://www.redhat.com/es/topics/open-source/what-is-open-source>
- Redondo, F. (18 de enero de 2017). *Paradigma*. Obtenido de <https://www.paradigmadigital.com/dev/postman-gestiona-construye-tus-apis-rapidamente/>



- Rivera Scaramutti, F., Iza Rotta, M., Torreblanca Villavicencio, A., Tavera Peña, I., & Borda Pinedo, V. (2016). Sistema de Admisión a la PUCP. Lima , Perú.
- Rohan, P. (11 de agosto de 2018). *How bcryptjs works*. Obtenido de <https://medium.com/javascript-in-plain-english/how-bcryptjs-works-90ef4cb85bf4>
- Rubio Fonseca, B. A., Camacho Escamilla, D. A., Marín Ariza, J. A., & Gómez, V. M. (2014). *El cambio en el sistema de admisión de la Universidad Nacional de Colombia: un estudio de caso del departamento de sociología*. Bogota .
- Sayago Heredia, J., & Chango Sailema, G. (27 de diciembre de 2018). Análisis comparativo para aplicaciones web basados en servicios REST: stack MEAN y stack Java EE. Knowledge E.
- Schwaber, K., & Sutherland, J. (noviembre de 2017). *scrum.org*. Obtenido de <https://www.scrumguides.org/scrum-guide.html>
- SUNEDU. (noviembre de 2015). modelo de licenciamiento y su implementacion en el sistema universitario peruano. Lima, Perú.
- talentcom. (2019-2020). *talentcom*. Obtenido de <https://mx.talent.com/salary?job=de+desarrollador+de+software>
- unipython. (8 de octubre de 2019). *CREAR UN PDF EN JAVA SCRIPT (JSPDF)*. Obtenido de <https://unipython.com/crear-un-pdf-en-java-script-jspdf/#:~:targetText=jsPDF%20es%20una%20librer%C3%ADa%20para,de%20manera%20sencilla%20y%20adaptable>.
- Universidad Peruana Austral del Cusco*. (04 de 02 de 2019). Obtenido de <http://uaustral.edu.pe/organigrama-de-la-universidad-peruana-austral-del-cusco/>
- uniwebsidad. (2 de noviembre de 2018). Obtenido de uniwebsidad: <https://librosweb.es/libro/css/capitulo-1.html>
- Urrutia, V. (22 de diciembre de 2018). *QUE ES HEROKU, PARA QUÉ SIRVE, VENTAJAS Y DESVENTAJAS*. Obtenido de <https://videlcloud.wordpress.com/2018/12/22/que-es-heroku-para-que-sirve-ventajas-y-desventajas/>
- Vele Zhingri, C. A. (2015). analisis de rendimiento de la base de datos relacional: MySQL y na base de datos no relacional: MongoDB. Cuenca, Ecuador: universidad del azuay, escuela de ingenieria de sistemas y telematica.
- VILLA, L. (06 de octubre de 2003). *alzado.org*. Obtenido de [https://www.alzado.org/articulo.php?id\\_art=221](https://www.alzado.org/articulo.php?id_art=221)
- Wiesse Mujica , M. (1994). Resolucion VRAC N° 109-94. Cusco, Perú.



ANEXOS

ANEXO 1

Fecha	Integrante del equipo	¿Qué hiciste ayer?	¿Qué tienes planeado hacer hoy?	¿Qué obstáculos encontraste en el camino?
03/12/18	Alberto C.	-----	Generación de los componentes básicos del proyecto	-----
	Esperanza W.	-----	Establecer la conexión entre node y mongo	-----
04/12/18	Alberto C.	La estructura de archivos del sistema, establecer una jerarquía de componentes	creación del componente breadCrumbs y el HTML de breadCrumbs	
	Esperanza W.	La conexión entre mongo y node	Las configuraciones iniciales de express y la creación del schema persona	
05/12/18	Alberto C.	creación del componente breadCrumbs y el HTML de breadCrumbs	creación del componente breadCrumbs y el HTML de header	
	Esperanza W.	Ceración del schema de persona	Buscar cómo se hace para no manda la contraseña del usuario cuando se haga una petición get o put	Como establecer los valores pre determinados para el campo rol
06/12/18	Alberto C.	creación del componente breadCrumbs y el HTML de header	Creación del HTML del componente no page fund y ceración del componente sidebar con la estructura HTML de sidebar	
	Esperanza W.	Buscar como no mandar la contraseña del usuario al hacer peticiones al servidor	Creación de las peticiones get, post, put de persona	
07/12/18	Alberto C.	Creación del HTML del	Creación del del componente login y	



		componente no page fund y ceración del componente sidebar con la estructura HTML de sidebar	la separación de este del resto del template	
	Esperanza W.	Creación de las peticiones get, post, put de persona y el link reference de consumo de las peticiones	Volver a hacer el método de no envió de la contraseña del usuario, he implementar la encriptación de la contraseña en la petición post	El método que implemente para que no mande la contraseña no funcionaba y tuve que investigar y volver ha hacerlo
10/12/18	Alberto C.	Creación del del componente login	Ver cómo implementar el routerOutlet para la separación del login del resto del template	Problemas con el routerOutlet para la generación de las rutas hijas
	Esperanza W.	Solucionar el problema del envió de la contraseña en las peticiones get y put	Implementación del código para la encriptación de la contraseña de persona	
11/12/18	Alberto C.	Funcionamiento del routerOutlet para separar el login	Creación y estructuración del módulo pages que contendrá todos los componentes y el modulo share para la administración del sidebar, breadcrumbs, header, no pagesfound	
	Esperanza W.	Encriptación de la contraseña del persona	Encadenamiento de las promesas para la creación de fechas y persona	
12/12/18	Alberto C.	creación y estructuración de pages y share	Ceración de las rutas hijas con forchild	
	Esperanza W.	Encadenamiento de las promesas para la creación de persona y fechas	Buscar la forma de que primero se cree fechas y su id pase a persona y se	Problemas al implementar el encadenamiento, no funcionan las promesas, el id



			almacene junto con el resto de datos	de la tabla fechas no se llega a guardar
13/12/18	Alberto C.	Ceración de las rutas hijas con forchild	Implementación del manejo del sidebar desde su servicio y la configuración del breadcrums	
	Esperanza W.	Anidado de las promesas para guardar persona y fechas	Creación de la petición get de todas la personas	
14/12/18	Alberto C.	Implementación del manejo del sidebar desde su servicio y la configuración del breadcrums	Creación e elaboración del modelo y servicio de persona, implementación de las funciones de para el consumo de la petición get de todas las personas y mostrar a las personas registradas en una tabla	
	Esperanza W.	Creación de la petición get de todas la personas	Petición put para desactivar y activar a una persona Petición put para actualizar los datos de una persona y de sus correspondientes fechas	Hacer populate de fechas para que me muestre las fechas ingresadas por persona Hacer la paginación del get del total de personas registradas
17/12/18	Alberto C.	Creacion del modelo de persona y la visualización de las personas en una tabla	Los datos de la tabla persona se muestren paginados	Problemas para la paginación de la tabla persona
	Esperanza W.	Peticiones put para la actualización de datos de persona y para desactivarla y activarla	Eliminación de la paginación de la petición get de persona  Petición get de una persona especifica	



			Generación de la petición post para el login de la persona	
18/12/18	Alberto C.	Hacer la paginación de los datos de la tabla persona sin que vengan paginados desde el backend	Implementación del login de persona  Elaboración del guard para la protección las rutas del sistema	
	Esperanza W.	Petición get de una persona específica y el logging de persona	Middleware para la verificacopn del token  Agregar a las peticiones port y put el middleware de verificación del token	Tuve que averiguar cómo se hace la verificación del token
19/12/18	Alberto C.	Login de persona y el guard de protecciones rutas	Logout de la persona  En el servicio de persona elaborara la función de creación de persona y hacer el HTML para la creación de la persona	Problemas para inicializar el valor del token No reconoce la función esta- logueado
	Esperanza W.	Middleware para la verificacopn del token  Agregar a las peticiones port y put el middleware de verificación del token	Generación de la funcionalidad de búsqueda de una persona por nombres y apellidos	
20/12/18	Alberto C.	Logout y creación de una persona	Actualización de los datos de una persona usando el mismo HTML que se elaboró para la creación de una persona	Al cerrar la cesión no se eliminaba los valores del locastorage
	Esperanza W.	Búsqueda de una persona	Creación de una petición put que permita la	Como hacer que la búsqueda no distinga entre





			actualización de la imagen del usuario	minúsculas y mayúsculas
21/12/18	Alberto C.	Actualización de los datos del usuario	Creación de la función activar y desactivar en el servicio y la implementación en el componente persona	En el servicio hacer una función que permita identificar si la persona la tiene id o no para según eso cree o actualice los datos de la persona Mostar las fechas en formato yyy-mm-dd y que permita desplegar el time picker
	Esperanza W.	Verificación de extensiones las imágenes	Cambiar el nombre de la imagen para poder ser subida	
26/12/18	Alberto C.	Activar y desactivar a una persona	Implementación de la funcionalidad de búsqueda de una persona	
	Esperanza W.			Enferma
27/12/18	Alberto C.	Implementación de la búsqueda de una persona	Creación del pipe para subir una imagen de usuario de la persona	
	Esperanza W.			Enferma
28/12/18	Alberto C.	Creación del pipe para subir una imagen de usuario de la persona	En la tabla muestre la imagen por defecto en caso la persona no haya actualizado la imagen	
	Esperanza W.	Parte de la funcionalidad para la actualización de la imagen de usuario de persona	Cambio de nombre de las imágenes para ser guardada	La verificación de extensiones de imágenes validadas
2/01/19	Alberto C.	Que el modal se oculte al	Corregir el funcionamiento del	El modal para la actualización de



		presionar el botón cancelar	modal para que se oculte cuando se da a cancelar o guardar	la imagen no se está ocultando al darle cancelar o guardar
	Esperanza W.	Cambio de nombre de las imágenes para ser guardada	Optimización del middleware de verificación del token	
03/01/19	Alberto C.	Corregir el funcionamiento del modal para que este oculto	notificación al localStorage del cambio de imagen	
	Esperanza W.	-----	-----	-----

En esta tabla se puede apreciar el avance del desarrollo del sistema día a día, que se realizó durante el primer sprint 1, concluyéndose solo la historia de usuario N° 4.



ANEXO 2

Fecha	Integrante del equipo	¿Qué hiciste ayer?	¿Qué tienes planeado hacer hoy?	¿Qué obstáculos encontraste en el camino?
10/01/19	Alberto C.	-----	Creación del componente y servicio de proceso  elaboración de la tabla proceso, creación y actualización de un proceso	-----
	Esperanza W.	-----	Creación del schema de proceso y la ruta de proceso Creación de la petición post, get de todos los procesos, get de un solo proceso, put para la actualización	-----
11/01/19	Alberto C.	Creación, actualización de un proceso y la tabla para mostrar los procesos	Creación del componente y servicio de modalidad  elaboración de la tabla modalidades, creación y actualización de una modalidad	
	Esperanza W.	Peticiones get (tanto para todos los procesos, como de un proceso), post, put para la actualización de un proceso	Creación del schema de modalidad y la ruta de modalidad Creación de la petición post, get de todas las modalidades, get de una sola modalidad, put para la actualización	
14/01/19	Alberto C.	Creación, actualización de una modalidad y la tabla para mostrar las modalidades	Creación del componente y servicio de carrera  elaboración de la tabla carreras, creación y actualización de una carrera	



	Esperanza W.	Peticiones get (tanto para todas las modalidades, como de una modalidad), post, put para la actualización de una modalidad	Creación del schema de carrera y la ruta de carrera Creación de la petición post, get de todas las carreras, get de una sola carrera, put para la actualización	
15/01/19	Alberto C.	Creación, actualización de una carrera y la tabla para mostrar las carreras	Creación del componente y servicio de requisito  elaboración de la tabla requisito, creación y actualización de un requisito	
	Esperanza W.	Peticiones get (tanto para todas las carreras, como de una carrera), post, put para la actualización de una carrera	Creación del schema de requisito y la ruta de requisito Creación de la petición post, get de todos los requisitos, get de un solo requisito, put para la actualización	
16/01/19	Alberto C.	Creación, actualización de un requisito y la tabla para mostrar los requisitos	Elaboración de la funcionalidad activar y desactivar proceso, modalidad, carrera, requisito	
	Esperanza W.	Peticiones get (tanto para todos los requisitos, como de un requisito), post, put para la actualización de un requisito	Elaboración de la petición put para activar y desactivar proceso, modalidad, carrera, requisito	
17/01/19	Alberto C.	Activar y desactivar proceso, modalidad, carrera, requisito	Creación del componente y servicio de modalidad-proceso  elaboración de la tabla modalidad-proceso, creación y actualización de una modalidad-proceso	



			activación y desactivación de una modalidad-proceso	
	Esperanza W.	Petición put para activar y desactivar proceso, modalidad, carrera, requisito	Creación del schema de modalidad-proceso y la ruta de modalidad-proceso Creación de la petición post, get de todas las modalidades-procesos, get de una sola modalidad-proceso, put para la actualización activación y desactivación de una modalidad-proceso	
18/01/19	Alberto C.	Creación del componente y servicio de modalidad-proceso  elaboración de la tabla modalidad-proceso  activación y desactivación de una modalidad-proceso	arreglar el problema del select  creación y actualización de una modalidad-proceso haciendo uso del servicio que lista las modalidades que están activas y los procesos activos	El select no jala los datos de proceso para poder almacenarlos.  Backend no hizo una promesa que devuelva solo las modalidades que están activas y otra que devuelva los procesos activos
	Esperanza W.	Creación del schema de modalidad-proceso y la ruta de modalidad-proceso Creación de la petición post (pero sin el almacenaje del nombre de la modalidad), get de todas las	Solucionar el problema de que luego de hacer la búsqueda de la modalidad pueda retornar el nombre de la modalidad para que se almacene en la tabla modalidad-proceso	No pude hacer que en la petición post, se busque y almacene el nombre de la modalidad



		modalidades- procesos, get de una sola modalidad- proceso, put para la actualización activación y desactivación de una		
24/01/19	Alberto C.	creación y actualización de una modalidad- proceso haciendo uso del servicio que lista las modalidades que están activas y los procesos activos	Creación del componente y servicio de detalleModPro  elaboración de la tabla detalleModPro, creación y actualización de detalleModPro	
	Esperanza W.	Solucionar el problema de que luego de hacer la búsqueda de la modalidad pueda retornar el nombre de la modalidad para que se almacene en la tabla modalidad- proceso	Creación del schema de detalleModPro y la ruta de detalleModPro Creación de la petición post, get de todas los detalleModPros, get de un detalleModPro, put para la actualización	
25/01/19	Alberto C.	Creación del componente y servicio de detalleModPro  elaboración de la tabla detalleModPro (sin la visualización de los nombres de las carreras)	Solucionar el problema de la visualización de multi select que sea con check box	El multi select no funciona bien no envía todo el arreglo de datos  El multi select no muestra los check box para poder seleccionar varias carreras



		creación y actualización de detalleModPro		
	Esperanza W.	Creación del schema de detalleModPro y la ruta de detalleModPro Creación de la petición get de todas los detalleModPros, get de un detalleModPro	Voy a intentar con el operador map para poder almacenar los nombres de las carreras en otro arreglo	Al mandar los datos a la petición post, al realizar la búsqueda de los ids del arreglo de carreras no puedo hacer que extraiga los nombres de las carreras y los almacene en otro arreglo que contenga solo los nombres
28/01/19	Alberto C.	Intentar solucionar el problema de la visualización de multi select que sea con check box	Hacer el multi select con librería Multiselect Dropdown	El multi select no muestra los check box para poder seleccionar varias carreras
	Esperanza W.	Intentar almacenar los nombres de las carreras en un array	Hacer que los nombres de las carreras se almacenen en un solo array	He logrado extraer los nombres de las carreras pero las almacena en una array distinto cada una, debe ser todo en uno solo
29/01/19	Alberto C.	Intentar hacer el multi select con librería Multiselect Dropdown	Hacer una función que consuma el servicio para poder listar los nombres de las carreras	el multi select muestra los check box en las opciones pero no estoy pudiendo mandar el servicio para que me muestre en las opciones las carreras
	Esperanza W.	Tratar de hacer que los nombres de las carreras	Intentar con otro tipo de estructuras for que me permitan almacenar todo en un solo arreglo	Al sacar los valores del buque que realiza las



		se almacenen en un solo array		búsquedas por ids, se pierden los valores y aparecen como undefine
30/01/19	Alberto C.	Tratar de hacer una función que consuma el servicio para poder listar los nombres de las carreras	Hacer una función que consuma el servicio para poder listar los nombres de las carreras	la función de Multiselect Dropdown no recibe un array y backEnd envía un array con los nombres de las carreras
	Esperanza W.			
31/01/19	Alberto C.	Intentar hacer una función que consuma el servicio para poder listar los nombres de las carreras	Intentare hacer el multi select con la librería chosen	No puede hacer una función que consuma el servicio de detalleModPro
	Esperanza W.	Probar distintas estructuras for para poder almacenar los nombres de las carreras	Intentar extraer los valores fuera del bucle	Sigo con el mismo problema de que se pierden los valores y aparecen como undefine al sacarlos del bucle
1/02/19	Alberto C.	Utilizar la librería chosen para hacer el multi select	Buscar cómo hacer funcionar la librería chosen en angular 7	No funciona el multi select, aparece como un cuadro grande que contiene todas las opciones
	Esperanza W.	Intentar extraer los valores fuera del bucle	Tarar de almacenar los datos sin salir del bucle	Sigo con el mismo problema de que se pierden los valores y aparecen como undefine al sacarlos del bucle





4/02/19	Alberto C.	Buscar cómo hacer funcionar la librería chosen en angular 7	Buscar y probar otras librerías de npm para hacer el multi select	No he encontrado una solución para el no funcionamiento de la librería chosen
	Esperanza W.	Tarar de almacenar los datos sin salir del bucle	Buscar como encadenar una promesa en al que solo se busquen u se almacene en una contante los nombres	No funciono el tratar e almacenar los datos dentro del bucle porque almacena el resto de datos por cada nombre de carrera que ha encontrado
5/02/19	Alberto C.	Buscar y probar otras librerías de npm para hacer el multi select	Usar angular material para hacer el multi select	No las librerías de npm que he consultado funcionan de manera similar a la de Multiselect Dropdown
	Esperanza W.	Buscar como encadenar una promesa en al que solo se busquen u se almacene en una contante los nombres	Se tomó la decisión de que frontend se encargue de hacer la búsqueda de los nombres de en base a los ids que obtiene y que los concatene todos dentro de un arreglo par que en backEnd haga el procesamiento de ese arreglo	Los datos al pasarlos de la promesa a la petición post se pierden
6/02/19	Alberto C.	Usar la librería angular material para tratar de implementar el multi select	Hacer la búsqueda de los nombres de las carreras y andarlos con sus ids a backEnd  Seguir buscando una solución para el problema de angular material	me parece un error que dice que el elemento mat-select No está definido en el schema de angular material notifications
	Esperanza W.	Seguí intentado hacer el proceso en backEnd y al	Hacer el filtrado de los detallesModPro que es estén relacionados con	No encontré la solución en



		final se decidio que frontend sea el que haga la búsqueda	modalidades-procesos que estén activos	backEnd a ese problema
7/02/19	Alberto C.	Hacer la búsqueda de los nombres de las carreras ´para el backEnd  Buscar la solución al problema de angular material	Buscar en más foros información para poder solucionar el problema	Al importar los componentes de la Liberia en el pages.module y en el app,module tego errores por la redundancia pero si quito la importación regreso al problema anterior
	Esperanza W.	Intentar hacer el filtrado de la detalleModPro por el estado de modalidad-proceso	Buscar alguna propiedad en mongoose que me permita hacer el filtrado	No se puede filtrar los datos, puesto que el estdo esta en otra tabla que es pilada con populate
8/02/19	Alberto C.	Creación de ngModule en el pages. Module para exportar los componentes de angular material	Hacer el listado de en la tabla de detalleModPros de los nombres de las carreras	
	Esperanza W.	Hacer el mach con populate para poder filtrar los datos	Filtrar los resultados obtenidos con JavaScript para eliminar lis valores null	Con populate ago el filtrado de los detalleModPros por el estado de modalidad-proceso pero de los que su estado es false me devuelve null
11/02/19	Alberto C.	Hacer el listado de en la tabla de detalleModPros de los nombres de las carreras	Hice un doble ngFor para que me pueda listar los nombres de las carreras dentro de una fila	Me muestra en la tabla los nombres de las carreras todo junto en una sola línea



	Esperanza W.	Tratar de usar JavaScript para poder eliminar los valores null	Usar Lodash para filtrar los resultados obtenidos	
--	--------------	----------------------------------------------------------------	---------------------------------------------------	--

En esta tabla se puede apreciar el avance del desarrollo del sistema día a día, que se realizó durante el primer sprint 2, concluyéndose solo la historia de usuario N° 3, 1, 2.



ANEXO 3

Fecha	Integrante del equipo	¿Qué hiciste ayer?	¿Qué tienes planeado hacer hoy?	¿Qué obstáculos encontraste en el camino?
19/02/19	Alberto C.	-----	Creación de los componentes y servicio de pre-inscripción  elaboración el formulario de pre-inscripción	-----
	Esperanza W.	-----	Creación del schema de proceso y la ruta de pre-inscripción Creación de la petición post, get de todos los pre-inscripciones, get de un solo pre-inscripción, put para la actualización	-----
20/02/19	Alberto C.	Elaboración de una página previa donde se le indica al postulante los pasos a seguir Y el formulario de pre-inscripción	Almacenamiento de los datos de la pre-inscripción	
	Esperanza W.	Creación del schema de proceso y la ruta de pre-inscripción	Terminar la petición post de la pre-inscripción	Hacer que la pre-inscripción almacene la fecha y la hora en la que se realizó y no la del servidor
21/02/19	Alberto C.	Almacenamiento de los datos de la pre-inscripción	Revisar las directiva ngFor para ver porque no jala nos nombres de las modalidades activas	Para listar las modalidades en un select me parece object ID
	Esperanza W.	Terminar la petición post de la pre-inscripción	Hacer la petición get de todos los pre-inscripciones, get de	



			un solo pre-inscripción	
22/02/19	Alberto C.	Arreglar la directiva ngFor para que me liste las modalidades y nombre de las carreras	En el componente de listado de preinscripciones hacer la tabla que muestre todas las pre-inscripciones	
	Esperanza W.	Hacer la petición get de todos los pre-inscripciones, get de un solo pre-inscripción	Implementación de la búsqueda de la pre-inscripción por nombre o apellidos del postulante	
25/02/19	Alberto C.	Tabla para mostrar las pre-inscripciones	Implementación de la búsqueda de las pre-inscripciones	
	Esperanza W.	La búsqueda de la una pre-inscripción	Petición pot para la validación de los datos del postulante que ingreso en la pre-inscripción y se haga la inscripción	
26/02/19	Alberto C.	Implementación de la búsqueda de las pre-inscripciones	Creación del formulario de validación de los datos del postulante	
	Esperanza W.	Petición post que crea la inscripción del postulante	Resolver el problema de la búsqueda de la pre-inscripción	Me al hacer la búsqueda de la pre- inscripción para almacenar los datos me parece que la pre-inscripción no se ha encontrado
27/02/19	Alberto C.	Implementación de la búsqueda de las pre-inscripciones	Solucionar el problema de la generación de 2 select por cada opción de carrera del postulante	Problemas al mostrar el select de las opciones de carreras, al poner el ngfor doble para que me liste los nombres de las carreras activas por la modalidad elegida se publica cada select (aparecen 4 en total)



	Esperanza W.	Resolver el problema de la búsqueda de la pre-inscripción	Resolver el problema de la búsqueda de la pre-inscripción buscando en foros la una situación similar	El problema de la búsqueda de la pre-inscripción persiste
28/02/19	Alberto C.	El problema de la duplicidad el select continua	Buscar como hacer un ngFor doble en un mismo select	Solucionar el problema de la generación de 2 select por cada opción de carrera del postulante
	Esperanza W.	Resolver el problema de la búsqueda de la pre-inscripción	Resolver el problema de la búsqueda de la pre-inscripción	El problema de la búsqueda de la pre-inscripción persiste
1/03/19	Alberto C.	El problema de la duplicidad el select continua	Buscar como usar el pipe de angular pala eliminar la duplicidad de select	Solucionar el problema de la generación de 2 select por cada opción de carrera del postulante
	Esperanza W.	Resolver el problema de la búsqueda de la pre-inscripción	Ayudar a resolver el problema de la duplicidad el select continua	
4/03/19	Alberto C.	Buscar como usar el pipe de angular pala eliminar la duplicidad de select	Hacer el llamado a los datos de las tablas relacionadas a la pre-inscripción	
	Esperanza W.	Buscar cómo usar el pipe de angular pala eliminar la duplicidad de select	resolver el problema de la duplicidad del select con el pipe de angular	
5/03/19	Alberto C.	Hacer el llamado a los datos de las tablas relacionadas a la pre-inscripción	Buscar el problema de que HTML no reconoce los rutas puestas en el ngModel	la consola del HTML muestra el error de no reconocer los campos de la tabla información académica en la directiva ngModel



	Esperanza W.	resolver el problema de la duplicidad del select con el pipe de angular	Buscar cómo hacer que reconozca los ids del html el pipe elaborado	Al marcar con una etiqueta id los select para evitar la publicidad la función elaborada con el pipe no reconoce los ids
6/03/19	Alberto C.	Buscar el problema de que HTML no reconoce los rutas puestas en el ngModel	Buscar el problema de que HTML no reconoce los rutas puestas en el ngModel	la consola del HTML muestra el error de no reconocer los campos de la tabla información académica en la directiva ngModel
	Esperanza W.	Buscar cómo hacer que reconozca los ids del html el pipe elaborado	Buscar como eliminar con el pipe la duplicidad de estructuras HTML	No elimina la duplicidad de el select por que no es un registro sino una estructura HTML
7/03/19	Alberto C.	Buscar el problema de que HTML no reconoce los rutas puestas en el ngModel	Buscar el problema de que HTML no reconoce los rutas puestas en el ngModel	la consola del HTML muestra el error de no reconocer los campos de la tabla información académica en la directiva ngModel
	Esperanza W.	Buscar como eliminar con el pipe la duplicidad de estructuras HTML	Buscar como encapsular el primer ngFor en otro tipo de estructura HTML	Solucionar el problema de la generación de 2 select por cada opción de carrera del postulante
8/03/19	Alberto C.	Buscar el problema de que HTML no reconoce los rutas puestas en el ngModel	Buscar el problema de que HTML no reconoce los rutas puestas en el ngModel	Buscar el problema de que HTML no reconoce los rutas puestas en el ngModel



	Esperanza W.	Buscar como encapsular el primer ngFor en otro tipo de estructura HTML	Buscar cómo poner en el select un ngFor y otro en las opción del select	Solucionar el problema de la generación de 2 select por cada opción de carrera del postulante
11/03/19	Alberto C.	Buscar el problema de que HTML no reconoce las rutas puestas en el ngModel	Generación del formato de inscripción en base al formulario de validación	
	Esperanza W.	Buscar cómo poner en el select un ngFor y otro en las opción del select	Buscar cómo poner en el select un ngFor y otro en las opción del select	Solucionar el problema de la generación de 2 select por cada opción de carrera del postulante
12/03/19	Alberto C.	Generar un pdf del formulario de validación	Buscar cómo hacer que capture los valores que están en los inputs	Jspdf no imprime los valores que se encuentran dentro de los inputs
	Esperanza W.	Buscar cómo poner en el select un ngFor y otro en las opción del select	Burcar como hacer el anidado de ngFor	Solucionar el problema de la generación de 2 select por cada opción de carrera del postulante
13/03/19	Alberto C.	Agregar la librería canvas para la captura de los datos	Que el pdf incluya los valores contenidos en los inputs	Jspdf no imprime los valores que se encuentran dentro de los inputs
	Esperanza W.	Burcar como hacer el anidado de ngFor	Buscar como concatenar 2 ngFor	Solucionar el problema de la generación de 2 select por cada opción de carrera del postulante
14/03/19	Alberto C.	Solucionar el problema de los inputs al generar el pdf	Buscar como excluir algunas estructuras html de la generación del pdf	Canvas me captura toda la interfaz del sistema





				incluyendo el menú de navegación
	Esperanza W.	Buscar como concatenar 2 ngFor	Que al concatenar los ngFor muestre solo las carreras correspondientes a la modalidad elegida	Solucionar el problema de la generación de 2 select por cada opción de carrera del postulante
15/03/19	Alberto C.	Buscar como excluir algunas estructuras html de la generación del pdf	Buscar como excluir algunas estructuras html de la generación del pdf	Canvas me captura toda la interfaz del sistema incluyendo el menú de navegación
	Esperanza W.	concatenar los ngFor muestre solo las carreras correspondientes a la modalidad elegida	Que al concatenar los ngFor muestre solo las carreras correspondientes a la modalidad elegida	Al concatenar muestra un todas las carreras, y no solo las que pertenecen a la modalidad seleccionada
18/03/19	Alberto C.	Buscar como excluir algunas estructuras html de la generación del pdf	Buscar como excluir algunas estructuras html de la generación del pdf	Canvas me captura toda la interfaz del sistema incluyendo el menú de navegación
	Esperanza W.	-----	-----	-----

En esta tabla se puede apreciar el avance del desarrollo del sistema día a día, que se realizó durante el primer sprint 3, concluyéndose solo la historia de usuario N° 5, 6.



ANEXO 4

Fecha	Integrante del equipo	¿Qué hiciste ayer?	¿Qué tienes planeado hacer hoy?	¿Qué obstáculos encontraste en el camino?
25/03/19	Alberto C.	-----	Creación del componente y servicio de reqMod	-----
	Esperanza W.	-----	Creación del schema de reqMod y la ruta de reqMod Creación de la petición post	-----
26/03/19	Alberto C.	Creación del componente y servicio de reqMod	Creación de la tabla reqMod	
	Esperanza W.	Creación del schema de reqMod y la ruta de reqMod Creación de la petición post	Creación de las peticiones get tanto para lista todos como para hacer la búsqueda con el id de uno solo reqMod	
27/03/19	Alberto C.	Creación de la tabla reqMod	Implementación de la interfaz de creación y actualización de reqMod	
	Esperanza W.	Creación de las peticiones get tanto para lista todos como para hacer la búsqueda con el id de uno solo reqMod	Ceración de la petición put de actualizar un reqMod	
28/03/19	Alberto C.	Implementación de la interfaz de creación y actualización de reqMod	Implementación de la activación y desactivación de reqMod	
	Esperanza W.	Ceración de la petición put de	La petición put pata activar y desactivar ReqMod	



		actualizar un reqMod		
29/03/19	Alberto C.	Implementación de la activación y desactivación de reqMod	Creación del componente y servicio de constancia	
	Esperanza W.	La petición put para activar y desactivar REqMod	Creación del schema de constancia y la ruta de constancia Creación de la petición post	
01/04/19	Alberto C.	Creación del componente y servicio de constancia	Ceración de la interfaz de registro de requisitos del postulante	
	Esperanza W.	Creación del schema de constancia y la ruta de constancia Creación de la petición post	Creación de una petición get de constancia por el id	
02/04/19	Alberto C.	Ceración de la interfaz de registro de requisitos del postulante	Hacer que el router navigate pase como parámetro el id de la inscripción del postúlate	
	Esperanza W.	Creación de una petición get de constancia por el id	Hacer la búsqueda por el id de la inscripción en la constancia	Cómo hacer para que pensando el id de la inscripción me busque la constancia generada
03/04/19	Alberto C.	Hacer que el router navigate pase como parámetro el id de la inscripción del postúlate	Buscar una solución al problema de pasar el id de la inscripción	Al pasar como parámetro el id de la inscripción en el router navigate aparece como indefinido
	Esperanza W.	Hacer la búsqueda por el id de la	Buscar una solución al problema de pasar el id de la inscripción	Cómo hacer para que pensando el id de la inscripción me



		inscripción en la constancia		busque la constancia generada
<b>04/04/19</b>	Alberto C.	Buscar una solución al problema de pasar el id de la inscripción	Buscar una solución al problema de pasar el id de la inscripción	Al pasar como parámetro el id de la inscripción en el router navigate aparece como indefinido
	Esperanza W.	Hacer la búsqueda por el id de la inscripción en la constancia	Buscar una solución al problema de pasar el id de la inscripción	Cómo hacer para que pensando el id de la inscripción me busque la constancia generada
<b>05/04/19</b>	Alberto C.	Buscar una solución al problema de pasar el id de la inscripción	Intentar pasar el id de la inscripción como id parámetro aparte	Al pasar como parámetro el id de la inscripción en el router navigate junto con la ruta de redireccionamiento me envía a una pagina que no existe
	Esperanza W.	problema de pasar el id de la inscripción	Ayudar con la generación de los datos de para la elaboración de la constancia en frontend	
<b>08/04/19</b>	Alberto C.	Intentar pasar el id de la inscripción como id parámetro aparte	Solucionar el problema del submit del id de la inscripción	Al obtener el id en el formulario de constancia de entrega de requisitos y querer mandar el id de la inscripción el ngModel no reconoce el valor del id
	Esperanza W.	Ayudar con la generación de los datos de para la elaboración de la constancia en frontend	Buscar como funciona la directiva ngNavigate	



09/04/19	Alberto C.	Solucionar el problema del submit del id de la inscripción	Solucionar el problema del submit del id de la inscripción	No se está guardando el id de inscripción
	Esperanza W.	Buscar como funciona la directiva ngNavigate	Al pasar como otro parámetro independiente el id no se llega a capturar el valor del id	No se está guardando el id de inscripción
10/04/19	Alberto C.	Solucionar el problema del submit del id de la inscripción	Pasar junto con la ruta de re-direccionamiento el id de inscripción	No se está guardando el id de inscripción
	Esperanza W.	Solucionar el problema del submit del id de la inscripción	Pasar junto con la ruta de re-direccionamiento el id de inscripción	No se está guardando el id de inscripción
11/04/19	Alberto C.	Solucionar el problema del submit del id de la inscripción	hacer que una vez creado la constancia ya no busque por el id de la url sino por la generada en el backEnd que es el de constancia	Busca una constancia con el id de la inscripción
	Esperanza W.	Solucionar el problema del submit del id de la inscripción	hacer que una vez creado la constancia ya no busque por el id de la url sino por la generada en el backEnd que es el de constancia	Busca una constancia con el id de la inscripción
12/04/19	Alberto C.	Solucionar el problema del submit del id de la inscripción	hacer que una vez creado la constancia ya no busque por el id de la url sino por la generada en el backEnd que es el de constancia	No se está guardando el id de inscripción
	Esperanza W.	Solucionar el problema del submit del id de la inscripción	hacer que una vez creado la constancia ya no busque por el id de la url sino por la generada en el backEnd que es el de constancia	No se está guardando el id de inscripción
16/04/19	Alberto C.	Solucionar el problema del submit del id de la inscripción	hacer que una vez creado la constancia ya no busque por el id de la url sino por la generada en el	No se está guardando el id de inscripción



			backEnd que es el de constancia	
	Esperanza W.	Solucionar el problema del submit del id de la inscripción	hacer que una vez creado la constancia ya no busque por el id de la url sino por la generada en el backEnd que es el de constancia	No se está guardando el id de inscripción
17/04/19	Alberto C.	Solucionar el problema del submit del id de la inscripción	Dividir en dos ya no hacer junto el crear y el actualiza para que no exista conflicto	No se está guardando el id de inscripción
	Esperanza W.	Solucionar el problema del submit del id de la inscripción	Creación de la ruta de reposte de conteo Creación de la petición get que hace el conteo	No se está guardando el id de inscripción
22/04/19	Alberto C.	Dividir en dos ya no hacer junto el crear y el actualiza para que no exista conflicto	Creación de una función separad para la actualización de la entrega de requisitos	
	Esperanza W.		Intentare con lodash para hacer el filtro de carreras y luego el conteo de inscritos	Función conteo para que haga el conteo por careras no funciona
23/04/19	Alberto C.	Creación de una función separad para la actualización de la entrega de requisitos	Hacer que se genere un pdf para la impresión de la constancia donde se incluya el plazo de regularización de requisitos	
	Esperanza W.	Intentare con lodash para hacer el filtro de carreras y luego el conteo de inscritos	Hacer una función con find anidado para búsqueda y conteo	No devuelve el nuero de inscritos sino un true o false
24/04/19	Alberto C.	Hacer que se genere un pdf para la impresión de la constancia	Creación del componente y servicio de reportes Elaboración de la tabla para mostrar los conteos por carrera y general	



	Esperanza W.	-----	-----	-----
--	-----------------	-------	-------	-------

En esta tabla se puede apreciar el avance del desarrollo del sistema día a día, que se realizó durante el primer sprint 4, concluyéndose solo la historia de usuario N° 7, 8, 9, 10.



ANEXO 5

Fecha	Integrante del equipo	¿Qué hiciste ayer?	¿Qué tienes planeado hacer hoy?	¿Qué obstáculos encontraste en el camino?
02/05/19	Alberto C.		Creación del componente y servicio de reporte de número y nombres de postulantes que entregaron sus requisitos completos	
	Esperanza W.		Creación del schema de proceso y la ruta de de reporte de número y nombres de postulantes que entregaron sus requisitos completos	
03/05/19	Alberto C.			Enfermo
	Esperanza W.	Creación del schema de proceso y la ruta de de reporte de número y nombres de postulantes que entregaron sus requisitos completos	Creación de la petición get del total de postulantes que entregaron completos sus requisitos	
06/05/19	Alberto C.			Enfermo
	Esperanza W.	Creación de la petición get del total de postulantes que entregaron completos sus requisitos	Hacer una función que evalué requisito por requisito si el postulante lo entrego	No puedo hacer el conteo por todos los requisitos completos
07/05/19	Alberto C.			Enfermo
	Esperanza W.	Hacer una función que evalué requisito por requisito si el postulante lo entrego	Unificar los resultados del conteo de requisito por requisito del postulante	No sé cómo unificar los resultados de cada una de las búsquedas por requisitos
08/05/19	Alberto C.			Enfermo





	Esperanza W.	Unificar los resultados del conteo de requisito por requisito del postulante	Buscar la forma de hacer una consulta que me permita evaluar todos los requisitos con una sola función	No me sale el resultado de que postulantes entregaron sus documentos completos
09/05/19	Alberto C.			Enfermo
	Esperanza W.	Buscar la forma de hacer una consulta que me permita evaluar todos los requisitos con una sola función	Buscar la forma de hacer una consulta que me permita evaluar todos los requisitos con una sola función	No me sale el resultado de que postulantes entregaron sus documentos completos
10/05/19	Alberto C.			Enfermo
	Esperanza W.	Buscar la forma de hacer una consulta que me permita evaluar todos los requisitos con una sola función	Hacer funciones anudadas que permitan evaluar si tiene los requisitos completos	No he podido evaluar todos los requisitos con una sola función
13/05/19	Alberto C.		elaboración de la tabla proceso, y el servicio que recibirá los datos	
	Esperanza W.	Hacer funciones anudadas que permitan evaluar si tiene los requisitos completos con Lodash	hacer el listado de los postulantes que tiene completos sus requisitos	No puedo recopilar todos los resultados en una sola respuesta
14/05/19	Alberto C.	elaboración de la tabla proceso, y el servicio que recibirá los datos	Verificar que los datos se vean de la manera esperada al consumir el servicio del backend	
	Esperanza W.	Tratar de hacer el listado de los postulantes que entregaron todos los requisitos	Almacenar todos los resultados en un arreglo usar un bucle para que se guarden todos en un campo del schema	Como la búsqueda es por cada postulante por id no estoy pidiendo mandar todo

En esta tabla se puede apreciar el avance del desarrollo del sistema día a día, que se realizó durante el primer sprint 5, concluyéndose solo la historia de usuario N° 11.



ANEXO 6



**AÑO DE LA LUCHA CONTRA LA CORRUPCIÓN E IMPUNIDAD  
CUSCO PATRIMONIO HISTÓRICO Y CULTURAL DE LA HUMANIDAD**

EL JEFE DE LA UNIDAD DE INFORMÁTICA DE LA UNIVERSIDAD PERUANA AUSTRAL DEL CUSCO, QUE SUSCRIBE:

**HACE CONSTAR:**

Que, la Bachiller Esperanza Wiese Gutiérrez que desempeña el Rol de Scrum Master con fecha 06 de diciembre del 2018, con expediente N° 1556, ha presentado una solicitud pidiendo autorización para puesta en prueba del Proyecto de Investigación denominado: "INFLUENCIA DE LA IMPLEMENTACIÓN DE UN SISTEMA BASADO EN CLOUD COMPUTING EN EL PROCESO DE INSCRIPCIÓN DE LA UNIVERSIDAD PERUANA AUSTRAL DEL CUSCO", adjuntando el Plan de tesis correspondiente, la que ha sido derivada a esta Unidad. El proceso de desarrollo y verificación del proyecto se desarrolló en los términos siguientes:

**Sprint 1:** Presentada el 07 de enero del 2019, la que ha resultado fallida por no haber sido desarrollada plenamente el sistema previsto, consistente en:

- Historia de Usuario N° 4, referido a la creación de usuarios en el sistema.
- Historia de Usuario N° 3, referido a la gestión de las tablas proceso, modalidad, carrera, requisitos.
- Historia de Usuario N° 1, referido a la activación y desactivación de los registros de tablas proceso, modalidad, carrera, requisitos.

De las cuales la Historia de Usuario N° 4 es entregada satisfactoriamente, mientras que las Historias de Usuarios N° 3 y 1 han sido fallidas.

Por lo tanto, la Historia de Usuario N° 4, cumple la definición de listo, de acuerdo a los siguientes parámetros:

- a) El producto cumpla con la funcionalidad definida en el product backlog
- b) El código se encuentre documentado (entiéndase por documentado cuando hay comentarios que ayudan a entender fácilmente el código)
- c) Se encuentre el producto desplegado en Heroku.

Dejando establecido que siguiente Sprint constará de: Historia de Usuario N° 3, Historia de Usuario N° 1.

**Sprint 2:** Del 13 de febrero del 2019, consistente en:

- Historia de Usuario N° 3, referido a la gestión de las tablas proceso, modalidad, carrera, requisitos.
- Historia de Usuario N° 1, referido a la activación y desactivación de los registros de tablas proceso, modalidad, carrera, requisitos.
- Historia de Usuario N° 2, referido las interfaces de modalidad - proceso, detalle - modalidad - proceso.

Las que tienen condición de satisfactorias, por lo que cumplen la definición de listo, de acuerdo a los siguientes parámetros:

- a) El producto cumpla con la funcionalidad definida en el product backlog
- b) El código se encuentre documentado (entiéndase por documentado cuando hay comentarios que ayudan a entender fácilmente el código).
- c) Se encuentre el producto desplegado en Heroku.

**Sprint 3:** Del 20 de marzo del 2019, consistente en:

- Historia de Usuario N° 5, referido a la pre inscripción que realiza el postulante por internet.
- Historia de Usuario N° 6, referido a la verificación de la pre inscripción.

Las que tienen condición de satisfactorias, por lo que cumplen la definición de listo, de acuerdo a los siguientes parámetros:

- a) El producto cumpla con la funcionalidad definida en el product backlog

Calle Sacsayhuamán L-13  
(084) 241189  
informes@uaustral.edu.pe



**AÑO DE LA LUCHA CONTRA LA CORRUPCIÓN E IMPUNIDAD  
CUSCO PATRIMONIO HISTÓRICO Y CULTURAL DE LA HUMANIDAD**

- b) El código se encuentre documentado (entiéndase por documentado cuando hay comentarios que ayudan a entender fácilmente el código).
- c) Se encuentre el producto desplegado en Heroku.

**Sprint 4:** Del 26 de abril del 2019, consistente en:

- Historia de Usuario N° 7, referido a la verificación de requisitos entregados por el postulante.
- Historia de Usuario N° 8, referido a la emisión de constancia de inscripción al examen de admisión.
- Historia de Usuario N° 9, referido a la notificación que se le hace al postulante sobre los plazos de regularización de requisitos faltantes, que va inserto en la constancia de inscripción al examen de admisión.
- Historia de Usuario N° 10, referido a la emisión de la estadística sobre los resultados del proceso de inscripción al examen de admisión a la fecha del reporte: en forma total y por escuelas profesionales.

Las que tienen condición de satisfactorias, por lo que cumplen la definición de listo, de acuerdo a los siguientes parámetros:

- a) El producto cumpla con la funcionalidad definida en el product backlog
- b) El código se encuentre documentado (entiéndase por documentado cuando hay comentarios que ayudan a entender fácilmente el código).
- c) Se encuentre el producto desplegado en Heroku.

**Sprint 5:** Del 17 de mayo del 2019, consistente en:

- Historia de Usuario N° 11, referido a la emisión de la estadística sobre los resultados del proceso de inscripción al examen de admisión a la fecha del reporte, indicando el número de postulantes que presentaron documentación completa en forma nominativa: del total y por escuelas profesionales.

La que tiene condición de satisfactorias, por lo que cumplen la definición de listo, de acuerdo a los siguientes parámetros:

- a) El producto cumpla con la funcionalidad definida en el product backlog
- b) El código se encuentre documentado (entiéndase por documentado cuando hay comentarios que ayudan a entender fácilmente el código)
- c) Se encuentre el producto desplegado en Heroku.

La que se expide para los fines consiguientes.

Cusco, 27 de mayo del 2019.



Calle Sacsayhuamán L-13  
(084) 241189  
informes@uaustral.edu.pe



ANEXO 7



**AÑO DE LA LUCHA CONTRA LA CORRUPCIÓN E IMPUNIDAD  
CUSCO PATRIMONIO HISTÓRICO Y CULTURAL DE LA HUMANIDAD**

EL JEFE DE LA UNIDAD DE INFORMÁTICA DE LA UNIVERSIDAD PERUANA AUSTRAL DEL CUSCO, QUE SUSCRIBE:

Por medio de la presente, el que suscribe pone de manifiesto que el Sistema de Inscripción al Examen de Admisión, desarrollado en el marco del trabajo de investigación **"INFLUENCIA DE LA IMPLEMENTACIÓN DE UN SISTEMA BASADO EN CLOUD COMPUTING EN EL PROCESO DE INSCRIPCIÓN DE LA UNIVERSIDAD PERUANA AUSTRAL DEL CUSCO"**, ha cumplido con la funcionabilidad esperada y no presento inconvenientes operacionales o técnicos, por lo que es un aporte importante para la Oficina de Admisión.

El siguiente documento se expide a solicitud de los interesados, para los fines consiguientes.

Cusco, 4 de septiembre del 2019.



Calle Sacsayhuamán L-13  
(084) 241189  
informes@uaustral.edu.pe



ANEXO 8



MANUAL DEL SISTEMA DE  
INFORMACIÓN PARA EL  
PROCESO DE INSCRIPCIÓN AL  
EXAMEN DE ADMISIÓN DE LA  
UNIVERSIDAD AUSTRAL DEL  
CUSCO

2019

Esperanza W.G. / Alberto C.A.  
[Dirección de correo electrónico]



## PRE INSCRIPCION DEL POSTULANTE

1. Para que el postulante realice la pre-inscripción primero se le presenta una página en donde se le indica los pasos a seguir para su postulación

Admisión 2019-II

A continuación te brindamos los pasos a seguir para el proceso de inscripción al examen de admisión:

**Pre-Inscripción**  
se realiza en la página web

1. selecciona la modalidad y carreras
2. ingresa tus datos personales
3. ingresa tu información académica
4. haga click en enviar
5. se le mostrará los requisitos que debe de presentar según su modalidad

**Inscripción**  
se realiza en la oficina de Admisión

1. entregue todos los requisitos correspondientes a la modalidad que seleccionó
2. entregue el voucher de pago por derechos de Examen Admisión
3. verificación de los datos de la Pre\_Inscripción
4. verificación de los requisitos
5. entrega de la constancia de inscripción

2. Luego debe proceder a llenar cada uno de los campos de los 3 pasos que se le indican, debe dar clic en enviar y ya habrá realizado su preinscripción

Pre-Inscripción 2019-II

todos los campos que tienen un asterisco ( \* ) son obligatorios

**1** Modalidad y Carreras

Proceso: 2019-II

Modalidad \*: Seleccione Modalidad

Opción de carrera 1 \*: Seleccione carrera

Opción de carrera 2 \*: Seleccione carrera

**2** Datos Personales

Tipo de documento de identidad \*: Seleccione tipo doc. identidad

Numero de documento de identidad \*

Nombre \*

Apellido paterno \*



Fecha del domicilio del postulante

Departamento \*  
Seleccione un departamento

Distrito \*  
Seleccione un distrito

Provincia \*  
Seleccione un provincia

Dirección del domicilio \*

**3** Información Académica

Tipo de IE \* (Institución Educativa)  
Seleccione

Nombre de la IE \*

Mes y Año de egreso de la IE \*

— de —

Enviar Cancelar

## MANTENIMIENTO DEL SISTEMA E INSCRIPCIÓN DEL POSTULANTE

### 1. Primero el usuario debe de loguearse



test1@test.com

Contraseña

Recordarme  ¿Olvide mi contraseña?

**INICIAR SESION**

pre-inscripcion



a) Si es administrador del sistema en el menú tendrá habilitado la parte de mantenimiento



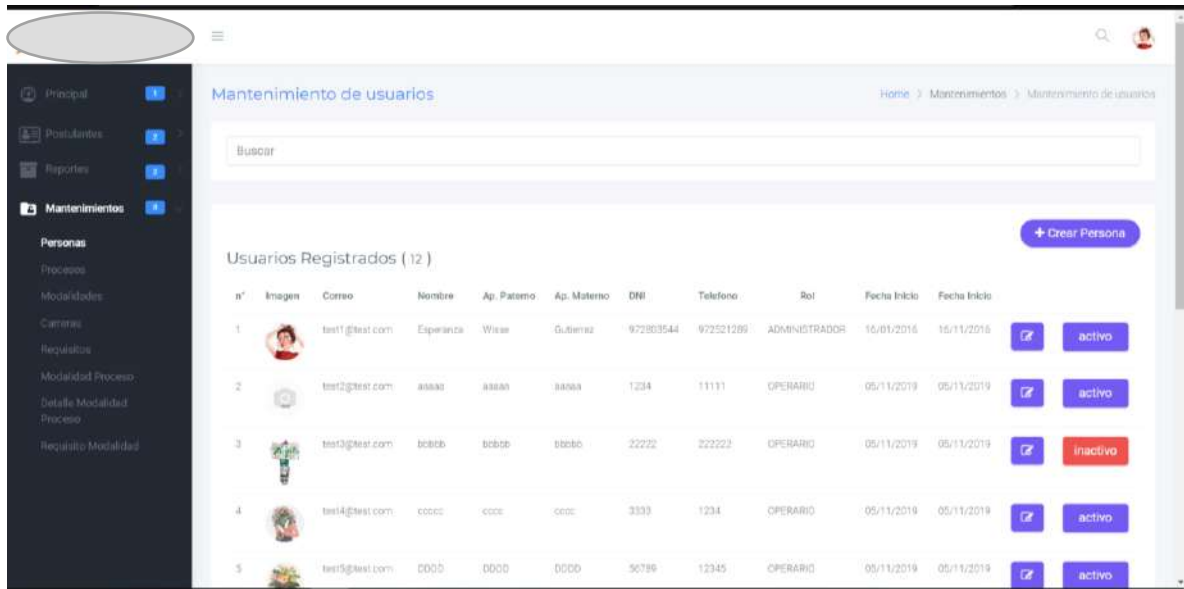
b) En caso de que sea un operario no tendrá acceso al mantenimiento del sistema



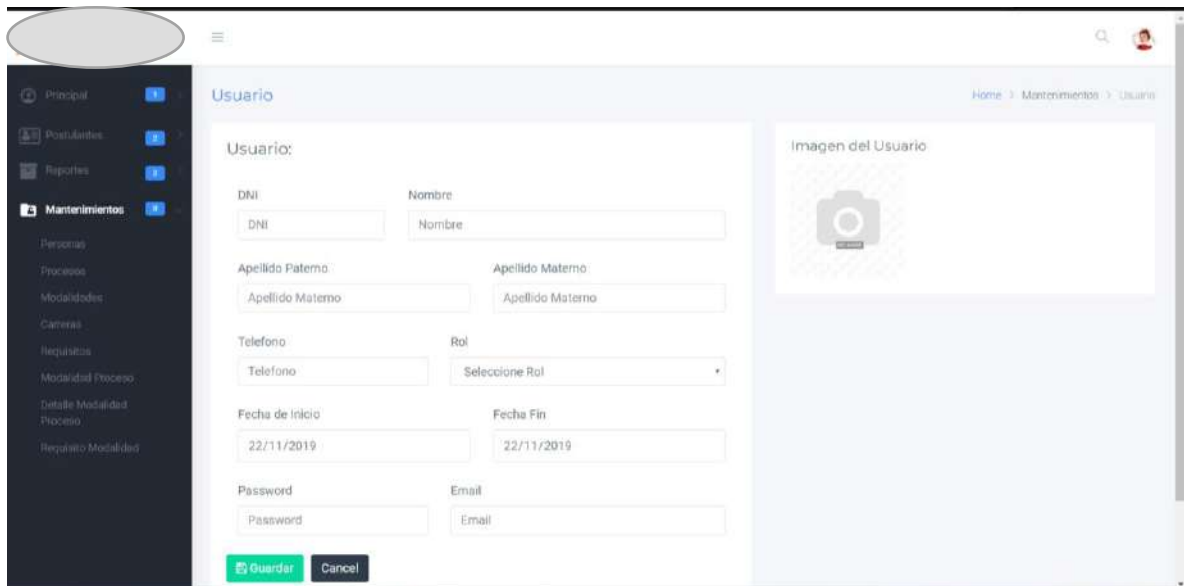




2. El administrador del sistema puede visualizar y hacer búsqueda de los usuarios por nombre o apellidos

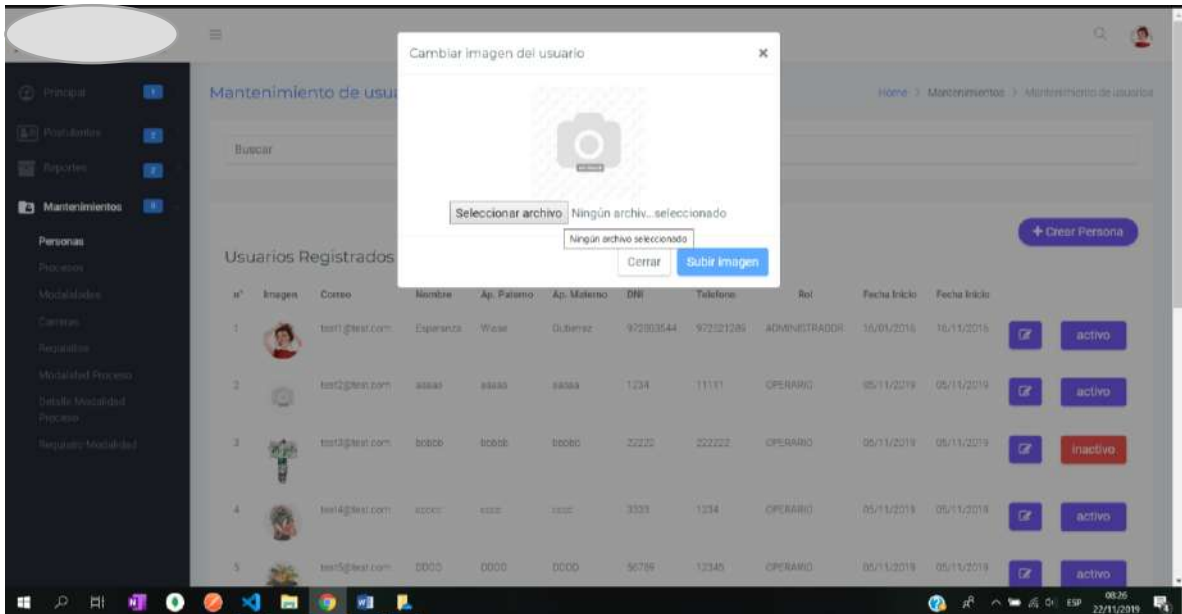


3. Si desea agregar a un usuario debe de hacer clic en crear nueva persona y llenar los datos que le pide el sistema

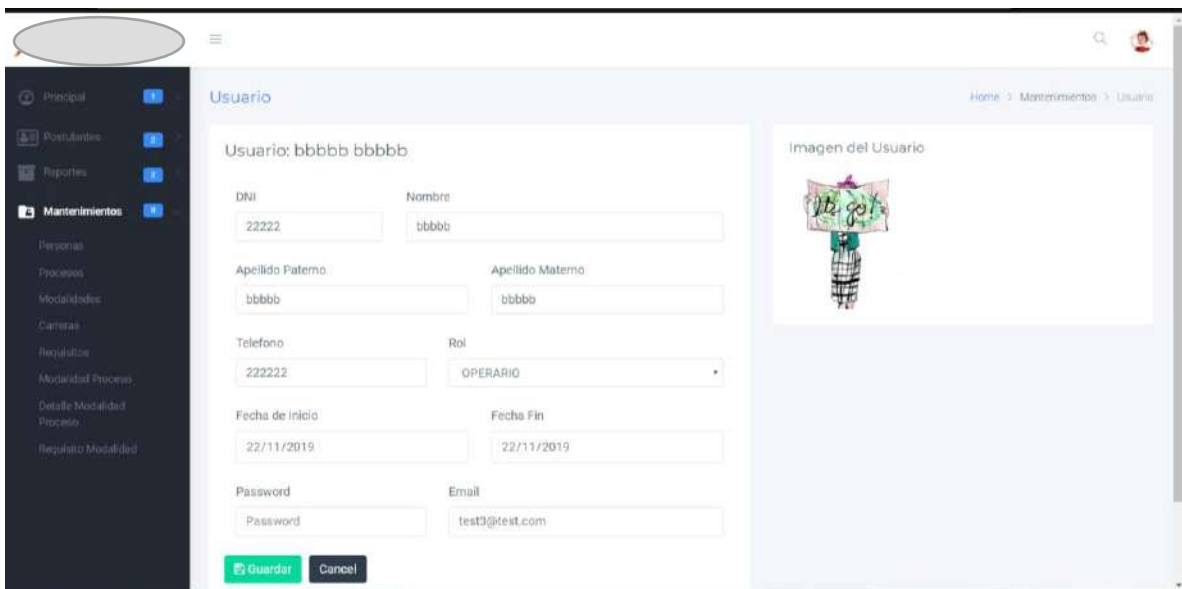




4. Para hacer el cambio de la imagen de uno de los usuarios solo debe dar clic en cima de la imagen u seleccionar una nueva



5. También el administrador podrá cambiar cualquiera de los datos el usuario en caso lo requiera





6. Para el caso de la creación de un proceso, modalidad, requisito o carrera puede activarlos y desactivarlos y crear y editar los datos

c) proceso

Mantenimiento de procesos

Procesos Registrados ( 5 )

n°	Proceso		
1	2018-I	<input checked="" type="checkbox"/>	Inactivo
2	2018-II	<input checked="" type="checkbox"/>	Inactivo
3	2018-III	<input checked="" type="checkbox"/>	Inactivo
4	2019-I	<input checked="" type="checkbox"/>	activo
5	2019-II	<input checked="" type="checkbox"/>	activo

< Anterior 1 Siguiente >

Proceso

Proceso:

Nombre Proceso

Nombre del Proceso



Proceso

Proceso: 2018-I

Nombre Proceso:

2018-I

Guardar Cancelar

d) modalidad

Mantenimiento de modalidades

Modalidades Registradas (6)

+ Crear Modalidad

n°	Modalidad		
1	ordinario	<input checked="" type="checkbox"/>	activo
2	primera oportunidad	<input checked="" type="checkbox"/>	activo
3	atletas destacados	<input checked="" type="checkbox"/>	inactivo
4	personas con discapacidad	<input checked="" type="checkbox"/>	inactivo
5	primeros y segundos puestos de educación secundaria	<input checked="" type="checkbox"/>	inactivo
6	graduados	<input checked="" type="checkbox"/>	activo

< Anterior 1 Siguiente >



e) carrera

Mantenimiento de carreras

Home > Mantenimientos > Mantenimiento de carreras

+ Crear Carrera

Carreras Registrados ( 4 )

n°	Carrera		
1	ciencias contables y finanzas		activo
2	economía y negocios internacionales		activo
3	ing. de sistemas y seguridad informática		activo
4	turismo		activo

< Anterior 1 Siguiente >

f) requisitos

Mantenimiento de requisitos

Home > Mantenimientos > Mantenimiento de requisitos

+ Crear Requisito

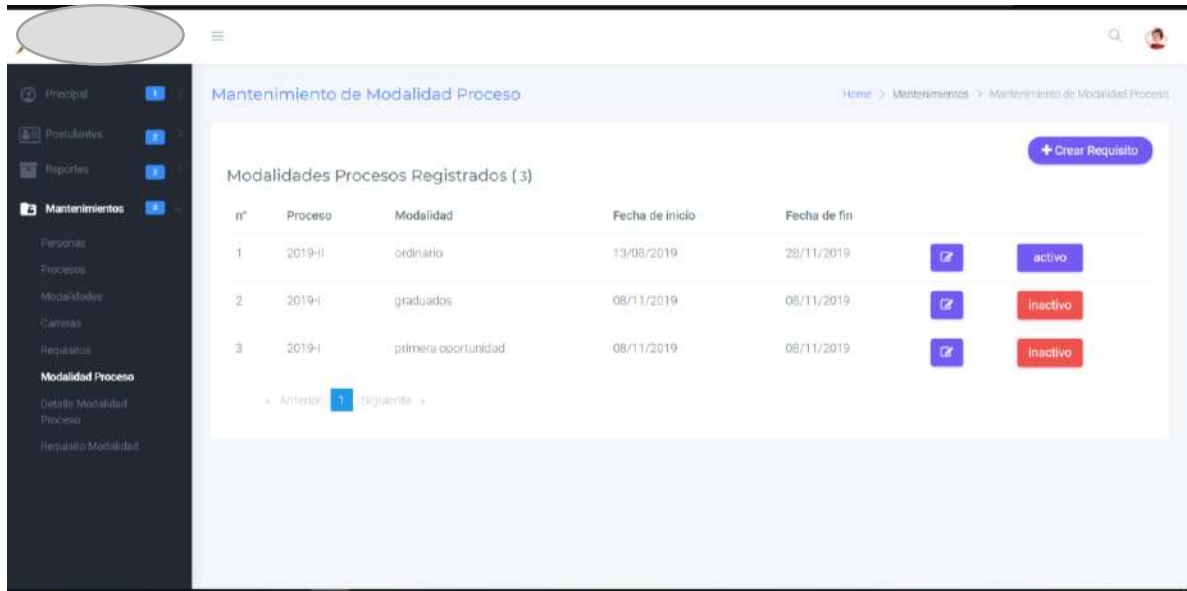
Requisitos Registrados ( 6 )

n°	Requisito		
1	DNI		activo
2	certificado de estudios		activo
3	partida de nacimiento		activo
4	declaración jurada		activo
5	constancia de ser atleta destacado		inactivo
6	constancia de ocupar el 1° o 2° puesto en el colegio		inactivo

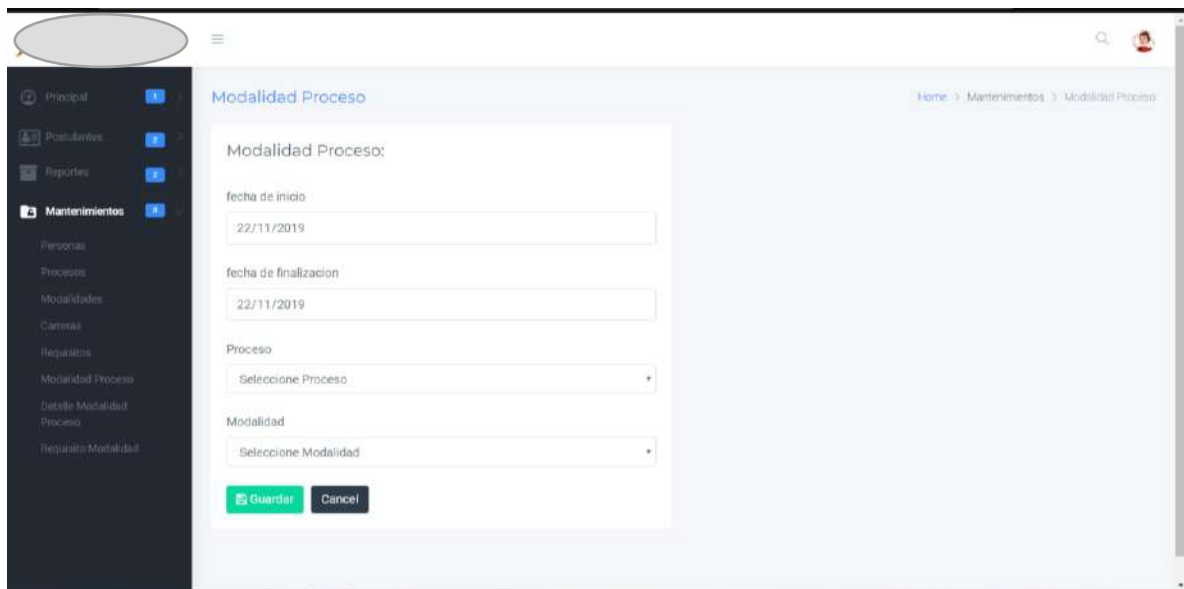
< Anterior 1 Siguiente >



7. para la apertura de un proceso de admisión el administrador del sistema tendrá:



a) primero que crear una modalidad proceso e indicar el intervalo de fechas que el proceso estará activo





b) en detalle modalidad proceso asignar las carreras a la modalidad-  
proceso

Mantenimiento de Detalle Modalidad Proceso

Home > Mantenimientos > Mantenimiento de Detalle Modalidad Proceso

+ Crear

Detalle Modalidades Procesos Registrados ( 1 )

n°	Proceso	Modalidad	Carreras	Fecha de inicio	Fecha de fin	Estado	
1	2019-8	ordinario	<ul style="list-style-type: none"><li>ciencias contables y finanzas</li><li>economia y negocios internacionales</li><li>ing. de sistemas y seguridad informatica</li><li>turismo</li></ul>	13/08/2019	28/11/2019	true	

< Anterior 1 Siguiente >

Detalle Modalidad Proceso

Home > Mantenimientos > Detalle Modalidad Proceso

Detalle Modalidad Proceso:

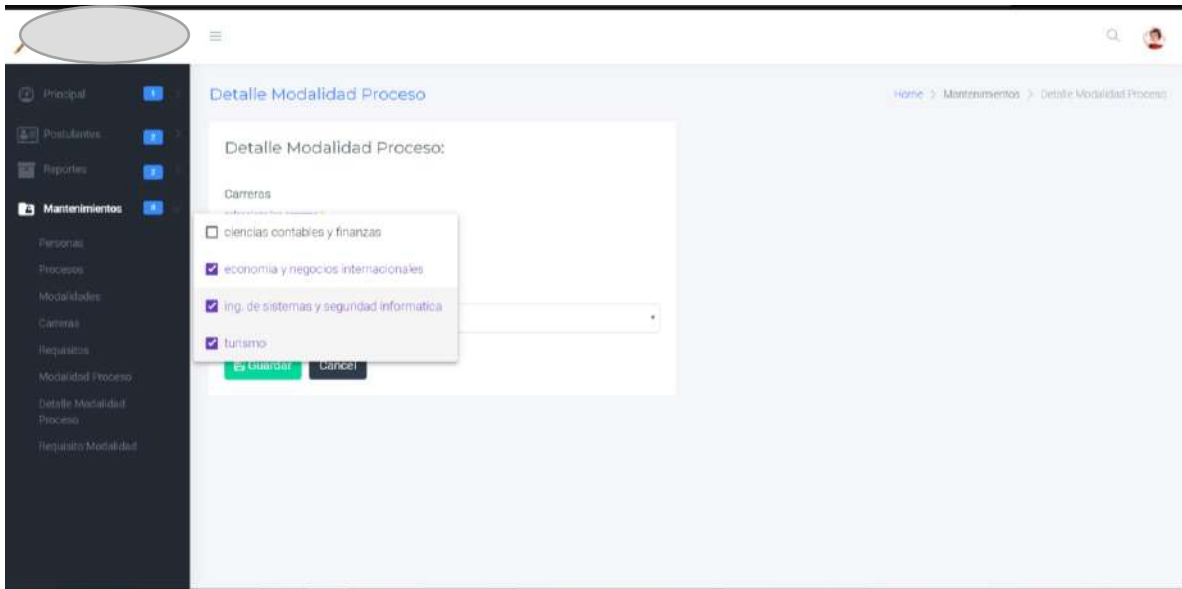
Carreras

seleccione las carrer...

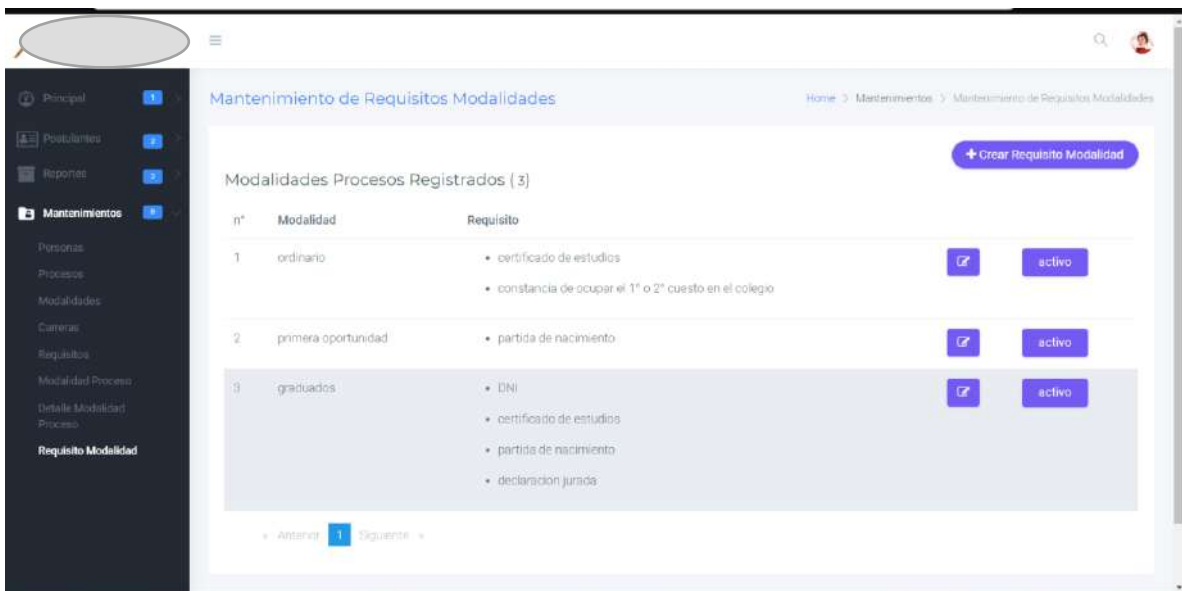
Proceso y Modalidad

Seleccione Proceso y Modalidad

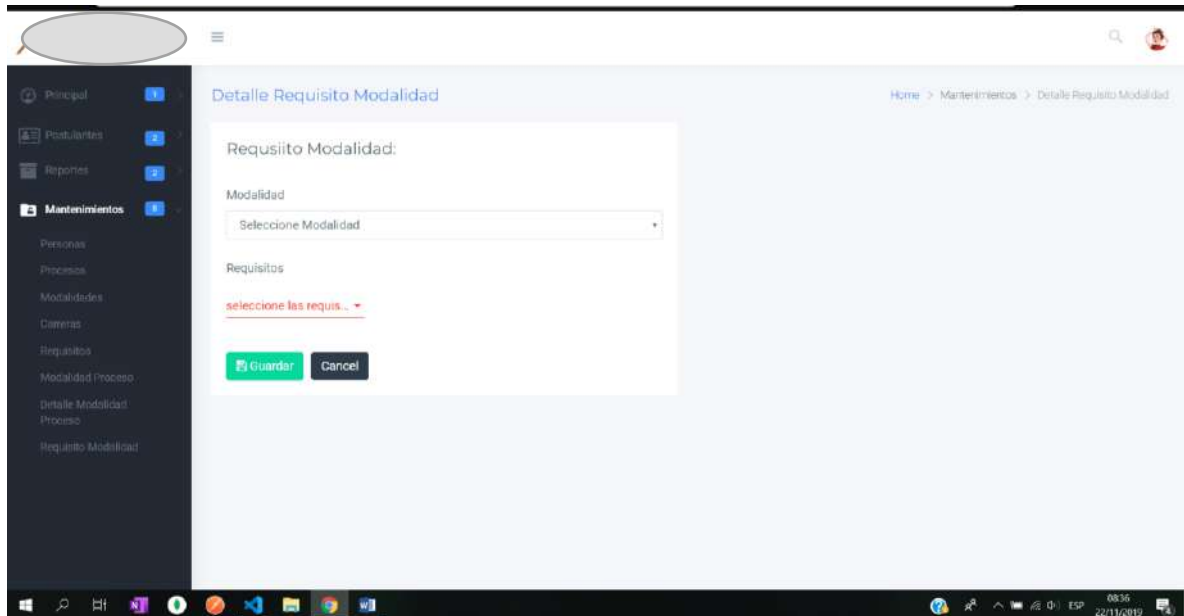
Guardar Cancel



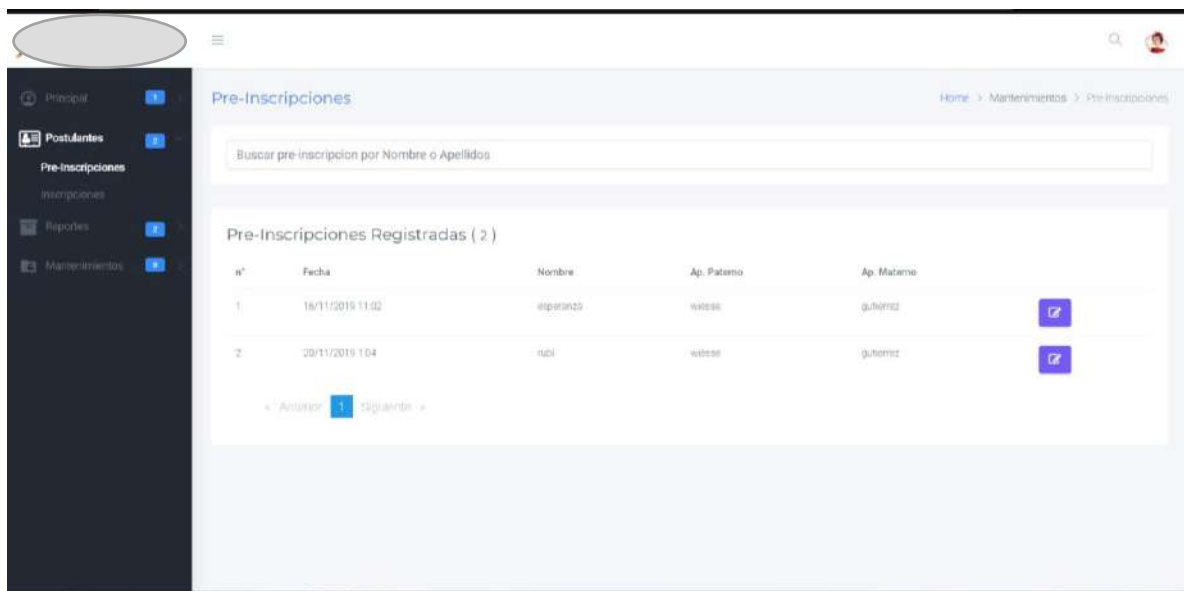
8. para la asignación de los requisitos a la modalidad debe de indicar primero la modalidad y luego asignar los requisitos que corresponden a la modalidad







9. para la inscripción de un postulante:
  - a) seleccionar la preinscripción y dar clic en editar





b) verificar que los datos de la preinscripción sean los correctos y llenar el campo n° de transacción

Pre-Inscripción

Código Preinscripción:  
5d94f713d765b4418a10e3

Modalidad y Carreras

Proceso: 2019-II Carrera 1: Ing. de sistemas y seguridad informatic

Modalidad: [ ] Carrera 1: turismo

Pago por derechos de examen de admisión  
n° de transacción: [ ]

Datos de domicilio

Departamento: cusco Distrito: cusco

Provincia: cusco Dirección: 23456

Datos del postulante

Tipo de documento de identidad: DNI n° de documento de identidad: 72803533

Nombre: rubi Apellido paterno: wiesse

Apellido materno: gutierrez Fecha de nacimiento: 08/12/2002

Información académica

Tipo de institución educativa: PARTICULAR Nombre de la institución educativa: galileo

c) debe de imprimir el formulario de inscripción y dar clic en guardar

Modalidad: [ ] Carrera 1: turismo

Datos de domicilio

Departamento: cusco Distrito: cusco

Provincia: cusco Dirección: 23456

Datos del postulante

Tipo de documento de identidad: DNI n° de documento de identidad: 72803533

Nombre: rubi Apellido paterno: wiesse

Apellido materno: gutierrez Fecha de nacimiento: 08/12/2002

Información académica

Tipo de institución educativa: PARTICULAR Nombre de la institución educativa: galileo

Mes y Año de egreso: Noviembre de 2019

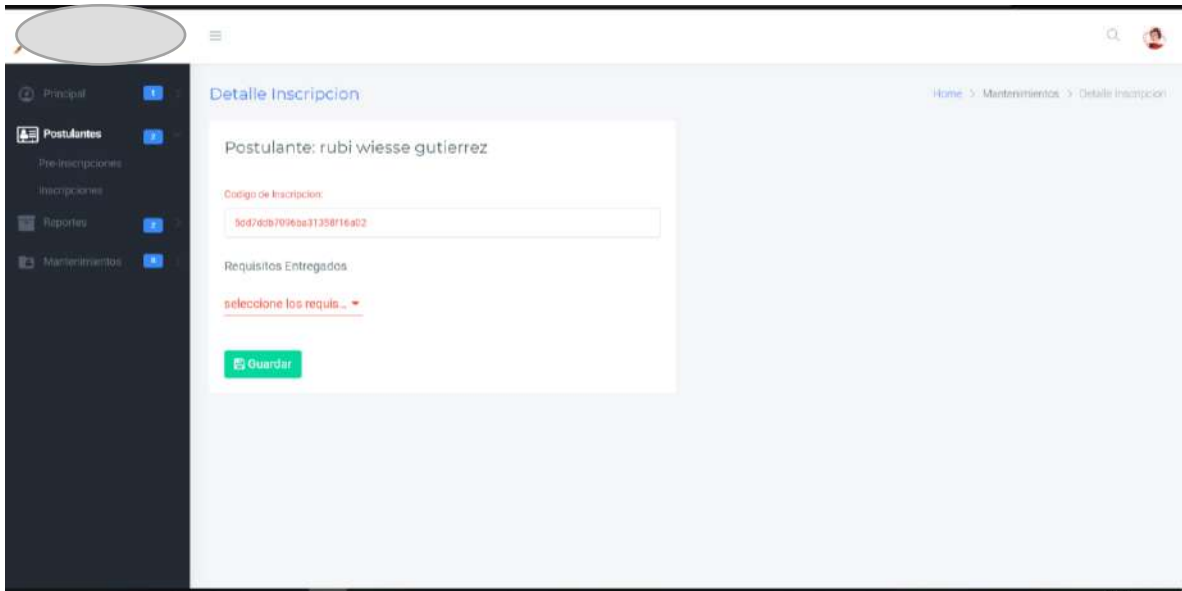
Estado civil: SOLTERO(A) Email: 1234@qwe.com

Telefono: 123456789

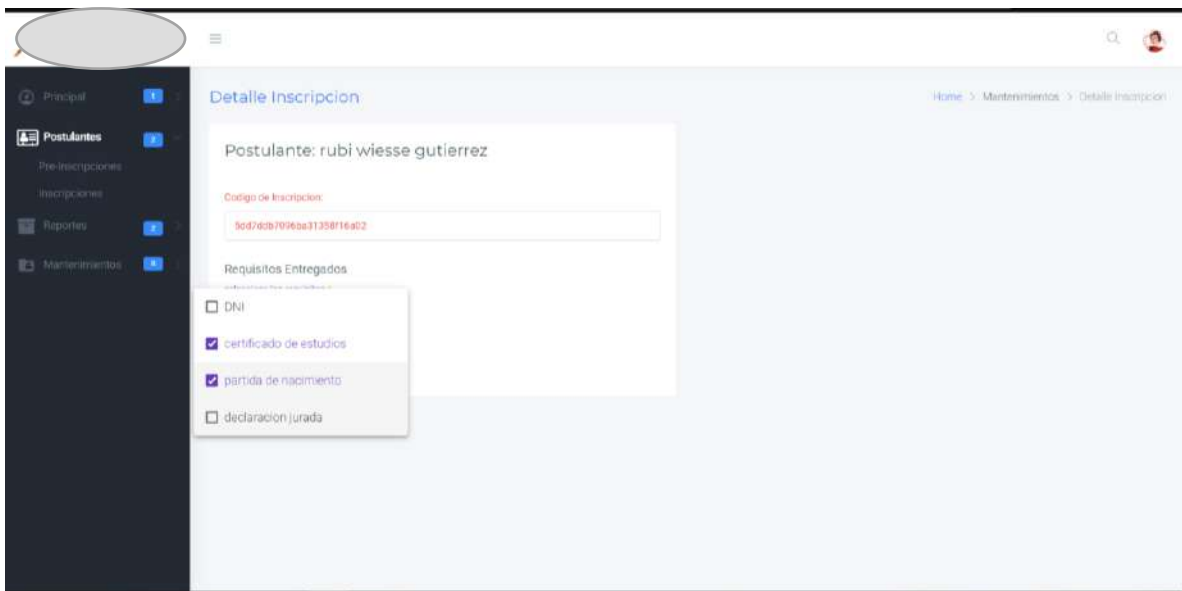
Imprimir Guardar Cancelar



d) luego debe de registrar los requisitos que es postulante esta presentando en momento de la inscripción



e) dar clic en guardar



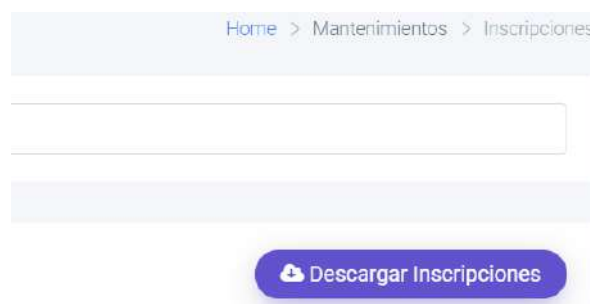


## ANEXO 9

En la imagen se puede ver el Excel que genera el sistema con toda la información de los postulantes por cada proceso de admisión. Los campos que contienen el Excel son:

- Teléfono del postulante
- Carreras que eligió el postulante
- Numero de documento de identidad
- Nombres del postulante
- Apellido paterno del postulante
- Apellido materno del postulante
- Fecha de nacimiento del postulante
- Edad del postulante
- Sexo del postulante
- Tipo de I.E. en que estudio el postulante
- Nombre de la I.E
- Año de egreso del postulante de la I.E
- Código de preinscripción
- Modalidad de ingreso
- Nombre del proceso de admisión
- Requisitos entregados

Para descargar el Excel del proceso de admisión, se hace desde la pantalla de inscripciones. Como se aprecia en la siguiente imagen.



Después de obtendrá un Excel con todos los campos ya mencionados, en que se encontrará la información de todos los postulantes.



A	B	C	D	E	F	G	H	I
tipoDocIdent	_id	nunTransacci	departament	distritoPre	provinciaPre	direccionPre	estadoCivilPr	emailPre
DNI	5f980d68c2ff	7865	Cusco	Cusco	Cusco	los incas f-3	SOLTERO(A)	mati@gm:
DNI	5f980fc8b13f	2354	Cusco	CACHIMAYO	Anta	canto c-6	CASADO(A)	maricita@
DNI	5f984aad7d7	7865	Cusco	Lamay	Calca	girasoles H-6	SOLTERO(A)	rapo@gm: