

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE
HUAMANGA**

FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL

ESCUELA DE FORMACIÓN PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“Aplicación web para la gestión académica del Instituto de
Educación Superior Tecnológico Público Churcampa, Región
Huancavelica, 2016”**

Tesis presentado por : Bach. CONDE JAULES, Néstor
Para optar el título profesional de : Ingeniero de Sistemas
Tipo de investigación : Aplicada.
Área de investigación : Ingeniería de Software
Asesor : Ing. LAGOS BARZOLA, Manuel Avelino

Ayacucho-Perú

2017

DEDICATORIA

A Dios y a todas las personas que me apoyaron en el proceso de mi formación profesional y en especial a mis hermanos y a mi madre Juana Jaulis Huamán por su apoyo incondicional.

AGRADECIMIENTO

A la Universidad Nacional de San Cristóbal de Huamanga, mi alma mater en mi educación universitaria, a todos los docentes de la Universidad, en especial para mi Maestro y Asesor Ing. Manuel Avelino Lagos Barzola, quien supo guiarme hacia el buen camino en mi vida universitaria.

CONTENIDO

	Pág.	
DEDICATORIA	i	
AGRADECIMIENTO	ii	
CONTENIDO	iii	
RESUMEN	v	
INTRODUCCIÓN	vi	
CAPITULO I		
PLANTEAMIENTO DE LA INVESTIGACIÓN		
1.1	DIAGNOSTICO Y ENUNCIADO DEL PROBLEMA	1
1.2	PROBLEMA DE INVESTIGACIÓN	4
1.3	OBJETIVOS DE LA INVESTIGACIÓN	5
1.4	HIPOTESIS DE LA INVESTIGACIÓN	6
1.5	JUSTIFICACIÓN Y DELIMITACION DE LA INVESTIGACIÓN	6
1.5.1	IMPORTANCIA DEL TEMA	6
1.5.2	JUSTIFICACIÓN	7
1.5.3	DELIMITACIÓN	7
CAPITULO II		
REVISIÓN DE LA LITERATURA		
2.1	ANTECEDENTES DE LA INVESTIGACIÓN	8
2.2	MARCO TEORICO	8
2.2.1	AUTOMATIZACIÓN DE GESTIÓN ACADÉMICA	8
2.2.2	INGENIERIA WEB	11
2.2.3	METODOLOGÍA DE PROGRAMACIÓN EXTREMA	15
2.2.4	PROGRAMACIÓN ORIENTADA A OBJETOS	33
2.2.5	BASE DE DATOS RELACIONAL	34
2.2.6	ADMINISTRACIÓN DE BASE DE DATOS RELACIONAL	35
2.2.7	MODELO VISTA CONTROLADOR	36
2.2.8	TECNOLOGÍAS DE INTERNET	37
2.2.9	ARQUITECTURA CLIENTE SERVIDOR	39
CAPITULO III		
METODOLOGIA DE INVESTIGACIÓN		
3.1	TIPO DE INVESTIGACIÓN	41
3.2	DISEÑO DE LA INVESTIGACIÓN	41
3.3	POBLACIÓN Y MUESTRA	41
3.4	VARIABLES E INDICADORES	42
3.5	TÉCNICAS E INSTRUMENTOS	44
3.5.1	TECNICAS PARA RECOLECTAR INFORMACIÓN	44
3.5.2	INSTRUMENTO PARA RECOLECTAR INFORMACIÓN	44
3.5.3	HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN	44
3.5.4	TÉCNICA PARA APLICAR XP	47

CAPITULO IV		
RESULTADOS DE LA INVESTIGACIÓN		
4.1	RESULTADOS	51
4.1.1	ARTEFACTOS DEL SOFTWARE APLICANDO EL PROCESO XP	51
4.2	DISCUSIONES	96
CAPITULO V		
CONCLUSIONES Y RECOMENDACIONES		
5.1	CONCLUSIONES	97
5.2	RECOMENDACIONES	98
	BIBLIOGRAFÍA	99
	ANEXO A	103
	ANEXO B	104

RESUMEN

El Instituto de Educación Superior Tecnológico Público Churcampa, es una entidad, donde los procesos de gestión académica se realizan manualmente, en consecuencia, la información que fluye es desorganizada, escasa e inoportuna, no permitiendo tomar una adecuada decisión en un momento dado. Por lo cual es indispensable e importante una aplicación web que permita manejar la información de todos los procesos y actividades para la institución de manera eficiente.

El objetivo del presente trabajo investigación es desarrollar una aplicación web para la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, 2016. Mediante la metodología XP, utilizando lenguaje de programación orientada a objetos, base de datos relacionales y tecnologías de internet; con el propósito de automatizar el proceso de gestión académica, y la finalidad de optimizar los tiempos en actividades de procesos académicos.

La investigación es de tipo aplicada, de nivel descriptivo, y los métodos de investigación son el análisis, síntesis, y la metodología XP para el desarrollo de software.

Los beneficios esperados es contar con una aplicación web para la gestión académica de la institución y lograr la diferenciación a nivel de institutos superiores tecnológicos en la región Huancavelica y el país.

PALABRAS CLAVE

Gestión académica, aplicación web, población estudiantil, metodología XP, programación orientada a objetos y base de datos relacional.

INTRODUCCIÓN

La evolución de internet como red de comunicación global y el surgimiento y desarrollo de la web como servicio imprescindible para compartir información, creó un excelente espacio para la interacción del hombre con la información, a la vez sentó las bases para el desarrollo de una herramienta integradora de los servicios existentes en internet. Los sitios web, como una expresión de sistemas de información, deben estar formados por usuarios, mecanismos de entrada y salida de la información, base de datos y mecanismos para recuperar la información.

El tema de investigación a desarrollar fue elegido viendo la problemática del Instituto de no contar con sistemas de información que automaticen los procesos de gestión académica. Para lo cual se implementó una aplicación web con la metodología XP que estará orientada a mejorar la gestión académica y automatización de los procesos y actividades de ámbito académico.

El problema principal de la investigación es; ¿Cómo automatizar la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016?

Los objetivos específicos son; automatizar el registro de las evaluaciones de la gestión académica, automatizar la generación de indicadores académicos, automatizar la administración de semestres académicos de la gestión académica, automatizar la administración de las Unidades Didácticas como parte de la gestión académica, automatizar las matrículas como parte de la gestión académica y automatizar la administración de horarios de la gestión académica del Instituto.

CAPITULO I

PROBLEMA DE LA INVESTIGACIÓN

1.1. DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA

Actualmente las instituciones educativas de nivel superior, ya sea privada o público, se ven en la necesidad de utilizar los sistemas de información para realizar sus actividades académicas, porque las leyes, reglamentos y resoluciones actuales así lo consignan. (Ley N° 29394 Ley de Institutos y Escuelas de Educación Superior y su Reglamento DS. N° 004-2010-ED, RM. N° 023-2010-ED, Decreto Supremo N° 010-2015 que modifica la Ley 29394, Resolución Viceministerial N° 073-2015 y Ley N° 30512-2016). De la misma manera los institutos deben acreditar sus carreras profesionales, para lo cual uno de los estándares de calificación es la utilización de sistemas de información en la gestión académica. El presente año el instituto superior tecnológico Churcampa se encuentra en el proceso de acreditación con la carrera profesional de Enfermería Técnica y Computación e Informática.

La institución es, en la actualidad una de las principales casas superiores de estudios de la provincia Churcampa y que cuenta con tres carreras profesionales: Producción Agropecuaria, Enfermería Técnica y Computación e Informática; también cuenta con tres sedes y/o filiales en los distritos: Anco, con las carreras profesionales de Producción Agropecuaria y Enfermería Técnica; Paucarbamba con las carreras profesionales de Producción Agropecuaria y Computación e Informática y en el distrito de Locroja con la carrera profesional de Computación e Informática.

El personal docente y administrativo desarrollan sus actividades académicas y administrativas en forma manual siendo éstas: informes, generación de horarios académicos, matrícula, consolidado de notas e inscripción para examen de admisión.

Estas situaciones no permiten tomar decisiones oportunas dentro de la institución por falta de automatización de procesos; además interfiere la distancia geográfica de las sedes en la comunicación eficaz respecto al intercambio de documentación e información sobre la gestión académica, teniendo como consecuencia flujo de información tardía y poco confiable, propensa a cometer errores.

Los jefes de área académica, de las carreras profesionales, utilizan las computadoras

personales para hacer uso exclusivo de paquetes office que le permita realizar estadísticas, cuadros, evaluaciones, registro, matriculas etc., sobre la gestión académica y administrativa; paquetes que no ayudan a confiar en la correcta toma de decisiones, y a la vez desperdiciando hardware y tecnología.

Sin embargo, como resultado de las actividades asociadas a los servicios que se ofrecen a los alumnos, que realiza el personal administrativo, y en particular el personal del área de dirección académica, se generan problemas como gastos administrativos y de personal no planificados, sanciones administrativas impuestas por la Dirección Regional de Educación de Huancavelica y Ministerio de Educación, desprestigio del área ante los padres de familia al no contar con el registro actualizado de la asistencia de los alumnos a clase, entre otras. Estos problemas se acentúan cuando los procesos se desarrollan de forma manual o utilizando herramientas de uso genérico como una hoja de cálculo y un procesador de textos.

Además, la población estudiantil durante los 04 últimos años ha ido incrementando, quienes necesitan atención oportuna; que podemos demostrar en los siguientes cuadros:

Tabla 1.1 (Matriculados por Carrera semestre Impar, 2013)

Carrera	Producción Agropecuaria						Enfermería Técnica						Computación e Informática					
	I		III		V		I		III		V		I		III		V	
Sexo	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M
Cantidad	13	11	6	4	4	6	7	18	8	13	4	13	13	12	11	13	8	7
Total	24		10		10		25		21		27		25		24		15	
	44						73						64					
	181																	

Fuente: Elaboración propia (análisis documental).

Tabla 1.2 (Matriculados por Carrera semestre Par, 2013)

Carrera	Producción Agropecuaria						Enfermería Técnica						Computación e Informática					
	II		IV		VI		II		IV		VI		II		IV		VI	
Sexo	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M
Cantidad	13	10	6	3	4	5	7	18	8	13	4	13	13	12	10	13	8	7
Total	23		9		9		25		21		27		25		23		15	
	41						73						63					
	177																	

Fuente: Elaboración propia (análisis documental).

Tabla 1.3 (Matriculados por Carrera semestre Impar, 2014)

Carrera	Producción Agropecuaria						Enfermería Técnica						Computación e Informática					
	I		III		V		I		III		V		I		III		V	
Semestre	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M
Cantidad	13	9	8	8	14	2	9	24	6	26	6	14	13	19	12	12	4	8
Total	22		16		16		33		32		20		32		2		12	
	54						85						46					
	185																	

Fuente: Elaboración propia (análisis documental).

Tabla 1.4 (Matriculados por Carrera semestre Par, 2014)

Carrera	Producción Agropecuaria						Enfermería Técnica						Computación e Informática					
	II		IV		VI		II		IV		VI		II		IV		VI	
Semestre	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M
Cantidad	13	9	8	8	14	2	9	24	6	26	6	14	13	19	12	12	4	8
Total	22		16		16		33		32		20		32		2		12	
	54						85						46					
	185																	

Fuente: Elaboración propia (análisis documental).

Tabla 1.5 (Matriculados por Carrera semestre Impar, 2015)

Carrera	Producción Agropecuaria						Enfermería Técnica						Computación e Informática					
	I		III		V		I		III		V		I		III		V	
Semestre	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M
Cantidad	14	14	7	4	6	6	8	24	8	17	5	27	16	16	11	20	10	9
Total	28		11		12		32		25		32		32		31		19	
	51						89						82					
	222																	

Fuente: Elaboración propia (análisis documental).

Tabla 1.6 (Matriculados por Carrera semestre Par, 2015)

Carrera	Producción Agropecuaria						Enfermería Técnica						Computación e Informática					
	II		IV		VI		II		IV		VI		II		IV		VI	
Semestre	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M
Cantidad	15	13	7	4	6	6	9	21	8	18	4	27	15	14	11	18	10	9
Total	28		11		12		30		26		31		29		29		19	
	51						87						77					
	215																	

Fuente: Elaboración propia (análisis documental).

Tabla 1.7 (Matriculados por Carrera semestre Impar, 2016)

Carrera	Producción Agropecuaria						Enfermería Técnica						Computación e Informática					
Semestre	I		III		V		I		III		V		I		III		V	
Sexo	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M	V	M
Cantidad	20	14	13	10	07	05	06	30	09	19	07	15	15	19	17	12	09	17
Total	34		23		11		36		28		22		34		29		26	
	68						86						89					
	243																	

Fuente: Elaboración propia (análisis documental).

Por el aumento de población estudiantil de acuerdo a las tablas 1.6 y 1.7, los problemas tienen su origen en el uso de procedimientos manuales para realizar la gestión académica. Esto se puede traducir como la necesidad de implementar procedimientos eficientes que agilicen el proceso de atención a los alumnos e incremente la productividad de los trabajadores de la institución.

De las descripciones realizadas se puede concluir que el Instituto Superior, en estudio, necesita una estrategia que le permita agilizar y automatizar el proceso de gestión académica.

1.2. DEFINICIÓN DEL PROBLEMA

PROBLEMA PRINCIPAL

¿Cómo automatizar la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016?

PROBLEMAS SECUNDARIOS

- ¿Cómo realizar la automatización de registro evaluaciones como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016?
- ¿Cómo realizar la automatización de indicadores académicos como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016?
- ¿Cómo realizar la automatización de Semestres académicos como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público

- Churcampa, región Huancavelica, 2016?
- d. ¿Cómo realizar la automatización de Unidades Didácticas como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016?
 - e. ¿Cómo realizar la automatización de matrículas como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016?
 - f. ¿Cómo realizar la automatización de horarios como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016?

1.3. OBJETIVOS DE LA INVESTIGACIÓN

OBJETIVO GENERAL

Desarrollar una aplicación web para la gestión académica del Instituto de Educación Superior Tecnológico público Churcampa, 2016. Mediante la metodología XP, utilizando lenguaje de programación orientada a objetos, base de datos relacionales y tecnologías de internet; con el propósito de automatizar el proceso de gestión académica, y la finalidad de optimizar los tiempos en actividades de gestión académica.

OBJETIVOS ESPECÍFICOS

- a. Automatizar el registro de las evaluaciones de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016.
- b. Automatizar la generación de indicadores académicos de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016.
- c. Automatizar la administración de semestres académicos de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016.
- d. Automatizar la administración de las Unidades Didácticas como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016.

- e. Automatizar las matrículas como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016.
- f. Automatizar la administración de horarios de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa, región Huancavelica, 2016.

1.4. HIPÓTESIS DE LA INVESTIGACIÓN

Para Hernández et. al. (2010), no todas las investigaciones cuantitativas plantean hipótesis. La formulación de hipótesis depende de un factor esencial: el alcance inicial del estudio. Las investigaciones cuantitativas cuyo alcance de estudio es exploratorio, no se formula hipótesis; si es descriptivo, sólo se formulan hipótesis cuando se pronostica un hecho o dato; si es correlacional, formulan hipótesis correlacionales; y si es explicativo, formulan hipótesis causales. El presente trabajo tiene un nivel de investigación descriptivo y no pronostica ningún hecho o dato por lo que no se formula una hipótesis.

1.5. JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN

1.5.1. IMPORTANCIA DEL TEMA

Las tendencias actuales del mercado han llevado a los institutos de educación superior tecnológicos ya sean privados y públicos a incorporar tecnología y soluciones informáticas para lograr resultados a mediano y largo plazo, y que el servicio de esta solución concuerde con las exigencias de las estrategias de atención para poder alcanzar una verdadera estrategia competitiva y cumplir con las leyes y normas legales que emanan del gobierno peruano.

En general, las organizaciones actuales de cualquier sector ven necesario utilizar la tecnología web como herramienta de gestión para desarrollarse y garantizar la continuidad de su funcionamiento de manera eficiente en el mercado. En tal sentido todas las organizaciones actuales están utilizando las tecnologías de información para dar soluciones a sus actividades de manera óptima frente a sus usuarios y clientes.

La aplicación web de gestión de procesos académicos es de vital importancia para el IESTP Churcampa, porque en la actualidad sus procesos de gestión académica, se ejecutan manualmente y el tiempo de atención a la población estudiantil y comunidad en

general no es adecuado, en tal sentido es de necesidad primordial la aplicación web para el instituto; además, este proyecto provee a los docentes de una herramienta útil e importante para realizar las actividades académicas.

La importancia vital de la aplicación web en mención, es por las siguientes razones: reducción de errores provocados por los docentes a través del control de las entradas, velocidad en el procesamiento de datos, posibilidad de realizar tediosos análisis sobre los mismos, reducción de espacio físico destinado a su almacenamiento, agilidad al momento de buscar algún dato en particular, y otros tipos de ventajas que podrían lograrse en caso de enfocarse en el uso estratégico de los mismos.

1.5.2. JUSTIFICACION

El Instituto de Educación Superior Tecnológico Público Churcampa siendo el único instituto público en la provincia de Churcampa, observamos que año tras año se incrementa el ingreso de los postulantes a las diferentes carreras que ofrece; por lo cual se ve la necesidad de mejorar los servicios a los docentes, estudiantes y personal administrativo.

El presente proyecto de investigación es de gran ayuda para el instituto tecnológico con el fin lograr su ventaja competitiva; y también permite la ejecución de proceso de acreditación de las carreras profesionales de acuerdo a las normas legales que exigen en la actualidad la utilización e implementación de sistemas de información. Además, es de vital importancia para la población estudiantil la automatización de los procesos académicos en el instituto.

La aplicación web, apoya la gestión de los procesos académicos, brindando información operativa y táctica prioritaria, mejorando el servicio a los estudiantes, graduados, docentes y autoridades de la institución.

1.5.3. DELIMITACIÓN

DELIMITACIÓN ESPACIAL

La investigación se realizó en el Instituto de Educación Superior Tecnológico Público Churcampa.

DELIMITACIÓN TEMPORAL

La recolección de datos correspondió al año 2016.

CAPITULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Méndez (2012), en su tesis de pregrado titulado como “Sistema de gestión de la unidad educativa Manuel Guerrero Ecuador”, concluyó; el beneficio que proporciona un sistema web como el desarrollado, es sin duda un gran aporte para una institución educativa, en la cual se ha comprobado las innumerables ventajas de utilizar un sistema con estas características y condiciones que fortalecen el desarrollo de la institución, al tener toda su información correctamente almacenada en una base de datos, con acceso fácil y adecuado para el mantenimiento, consultas y para la toma de decisiones en ciertos casos.

Vinicio (2012), en su tesis de pregrado titulado como, “Desarrollo de Sistema de Gestión Académica para la Escuela Gonzalo Rubio Orbe de Otavalo”, concluyó; el sistema de información web de gestión académica de la escuela Gonzalo Rubio Orbe automatiza los procesos académicos y facilita el trabajo de los docentes y administrativos, mejorando los costos y el tiempo de ejecución de los procesos académicos.

Fernández (2009), en su tesis de pregrado titulado “Software para departamentos académicos de la UNSCH”; concluye; la aplicación para la gestión académica y administrativa de los departamentos académicos mejora los servicios a los docentes y personal administrativo, brindando información oportuna, rápida y precisa, porque las funcionalidades prioritarias de los procesos fueron implementadas.

2.2. MARCO TEÓRICO

2.2.1 AUTOMATIZACION DE GESTION ACADEMICA.

Para Fernández, (2014), la noción de la gestión se extiende hacia el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es también la dirección o administración de una compañía o de un negocio. En este caso, es un conjunto de actividades relacionadas al aspecto académico dentro de

una institución educativa donde se realizan proceso y se generan información, a través de un sistema de información que realiza automáticamente el proceso académico.

Según Delgado (2006), la calidad de gestión es un continuo, cuyos puntos representa combinaciones de funcionalidad, eficiencia y eficacia altamente correlacionadas y su grado máximo, la excelencia, supone un óptimo nivel de coherencia entre todos los componentes del sistema.

Para Fernández (2003), es un saber de síntesis capaz de ligar conocimiento y acción, ética y eficacia, política y administración de procesos que tienden al mejoramiento continuo de las prácticas educativas: a la exploración y explotación de todas las posibilidades; y a la innovación permanente como proceso sistemático.

Automatización

Según Smith (2011), es la sustitución de tareas tradicionalmente manuales por las mismas realizadas de manera automática por máquinas, robots o cualquier otro tipo de automatismo. La automatización tiene ventajas muy evidentes en los procesos industriales. Se mejora en costes, en servicio y en calidad. El trabajo es más rápido y no necesita de una cantidad determinada de operarios, que antes eran necesarios. Además, se producen menos problemas de calidad por realizarse el trabajo de una manera más uniforme debido a las especificaciones dadas al automatismo. Otras ventajas se obtienen de la automatización son el aumento de producción, menor gasto energético, mayor seguridad para los trabajadores.

A. Automatización de evaluaciones.

La evaluación es la acción de estimar, calcular o señalar el valor de algo. La evaluación es la determinación sistemática del mérito, el valor y el significado de algo o alguien en función de unos criterios respecto a un conjunto de normas.

El sistema de evaluación académica del alumno de los Institutos y Escuelas tiene las siguientes características: Integral, flexible, Permanente y pertinente. (Ley 29394 Art.19).

B. Automatización de indicadores académicos.

Los indicadores académicos permiten la consecución de las unidades de competencia. Cabe indicar que a cada unidad de competencia le corresponde un

conjunto de indicadores académico.

Para la identificación de los indicadores se debe considerar los siguientes criterios: Que corresponda a una unidad de competencia, que sean redactados como unidad de aprendizaje y que sean medibles que aseguren la correspondencia directa con el nivel educativo de la carrera profesional. (069-2015-MINEDU).

C. Automatización de semestres académicos.

Semestre es un concepto que tiene su origen etimológico en el vocablo latino “semestris”, que significa un determinado tiempo donde se ejecutan los procesos académicos dentro de una institución educativa (RALE)

El concepto hace referencia a un periodo de seis meses. Por ejemplo: “semestre académico de una institución educativa se refiere al tiempo que dura las clases”. (Ley 29394-2009-MINEDU)

D. Automatización de gestión de unidades didácticas.

Son unidades de programación de enseñanza con un tiempo determinado. La unidad didáctica es una propuesta de trabajo relativa a un proceso de enseñanza aprendizaje completo. Este modelo didáctico aparece muy ligado a las teorías constructivistas que en la actualidad se están implementando en los Institutos Tecnológicos. (069-2015-MINEDU).

E. Automatización de matrículas.

En las instituciones educativas, la matriculación, también llamada proceso de matrícula, suele consistir en la mayor parte de los casos en la cumplimentación de los formularios correspondientes y la aportación de la documentación adecuada. Las hojas de formulario se suelen encontrar en las secretarías de los centros de enseñanza. Allí se recogen y se entregan durante el periodo de matrícula. Este tiene lugar un tiempo antes del comienzo de las clases para que la administración del centro tenga tiempo de procesar los datos y organizar la información sobre los estudiantes.

Los requisitos para matricularse en las carreras ofrecidas por los Institutos y Escuelas son haber concluido la educación básica, aprobado el examen de admisión y cumplir con los requisitos establecidos por cada institución. (Ley 29394 Art.18).

F. Automatización de horarios.

Del latín horarius, el término horario hace referencia a aquello que tiene relación o que pertenece a las horas. Su uso más habitual está vinculado al periodo temporal durante el que se lleva a cabo una actividad. (RALE)

Se hace referencia al número de horas que debe comprender el desarrollo de la carrera profesional y que guarda relación con el número de créditos. (069-2015-MINEDU).

2.2.2 INGENIERIA WEB

Para Saldaña (2012), un sistema web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador web con cliente ligero. La facilidad para actualizar y mantener aplicaciones web sin destruir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Las aplicaciones como web mails, wikis, weblogs, tiendas en línea y comercio electrónico son ejemplos bien conocidos de las aplicaciones web.

Por otro lado, Pressman (2010), manifiesta que es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo.

Para Palomo y Montero (2005), la ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables, al desarrollo eficiente de aplicaciones de alta calidad en la web. Asimismo, hace referencia a las metodologías, técnicas y herramientas que se utilizan en el desarrollo de aplicaciones web complejas y de gran dimensión en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones.

A. Requisitos

Según Pressman (2010), los requisitos son el espectro amplio de tareas y técnicas que llevan a entender los requerimientos se denomina ingeniería de requerimientos. Desde la perspectiva del proceso del software, la ingeniería de requerimientos es una de las acciones importantes de la ingeniería de software que comienza durante la actividad de comunicación y continúa en la de modelado. Debe adaptarse a las necesidades del proceso, del proyecto, del producto y de las personas que

hacen el trabajo. La ingeniería de requerimientos tiende un puente para el diseño y la construcción. Pero, ¿dónde se origina el puente? Podría argumentarse que principia en los pies de los participantes en el proyecto (por ejemplo, gerentes, clientes y usuarios), donde se definen las necesidades del negocio, se describen los escenarios de uso, se delimitan las funciones y características y se identifican las restricciones del proyecto.

Los requisitos es una característica que debe tener el sistema o una restricción que debe satisfacer para que sea aceptado por el cliente. La ingeniería de requisitos pretende definir los requerimientos del sistema que se está construyendo. La ingeniería de requerimientos incluye dos actividades principales: la obtención de los requerimientos, que da como resultado una especificación del sistema que el cliente comprende, y el análisis, que da como resultado un modelo de análisis que los desarrolladores pueden interpretar sin ambigüedad. La obtención de requerimientos es la más retadora de los procesos, debido a que requiere la colaboración de varios grupos de participantes con diferentes niveles de conocimientos. Por un lado, el cliente y los usuarios son expertos en sus dominios y tienen una idea general de lo que debe hacer el sistema. Sin embargo, a menudo tienen muy poca experiencia en el desarrollo de software. Por otro lado, el desarrollador tiene experiencia en la construcción de sistemas, pero con frecuencia tienen muy poco conocimiento del ambiente diario del usuario.

Por otro lado, Rosenberg y Stephens (2007), manifiestan que definir lo que el sistema debe ser capaz de hacer. Dependiendo de la forma en que su proyecto está organizado, ya sea que usted participe en la creación de los requisitos funcionales o que los requisitos serán “dictados desde lo alto” de un cliente o un equipo de analista de negocio.

B. Análisis de Sistemas

Kendall y Kendall (2011), manifiestan, que análisis de sistemas es un conjunto de procedimientos o programas relacionados de manera que juntos forman una sola unidad. El proceso debe partir desde la información esencial hasta el detalle de la Implementación. La función del Análisis puede ser dar soporte a las actividades de un negocio, o desarrollar un producto que pueda venderse para generar beneficios.

Para Pressman (2010), lo primero que debemos hacer para construir un sistema de información es averiguar qué es exactamente lo que tiene que hacer el sistema. La etapa de análisis en el ciclo de vida del software corresponde al proceso mediante el cual se

intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer).

C. Diseño

Según Bruegge y Dutoit (2010), manifiestan que el diseño de sistemas es la transformación del modelo de análisis en un modelo de sistema. Durante el diseño del sistema los desarrolladores definen objetos de diseño del proyecto y descomponen el sistema en subsistemas más pequeños que puedan ser realizados por equipos individuales.

Por otro lado, Sommerville (2005), manifiesta que el diseño de sistemas se centra en proporcionar la funcionalidad del sistema a través de sus diferentes componentes:

- ✓ Dividir los requerimientos en grupos afines.
- ✓ Identificar los subsistemas para cumplir os requerimientos.
- ✓ Asignar los requerimientos a los subsistemas.
- ✓ Especificar la funcionalidad de los subsistemas.
- ✓ Definir las interfaces de subsistemas.

Todo ello implica el diseño de sistemas que es una parte muy importante de ciclo de vida de sistemas de información.

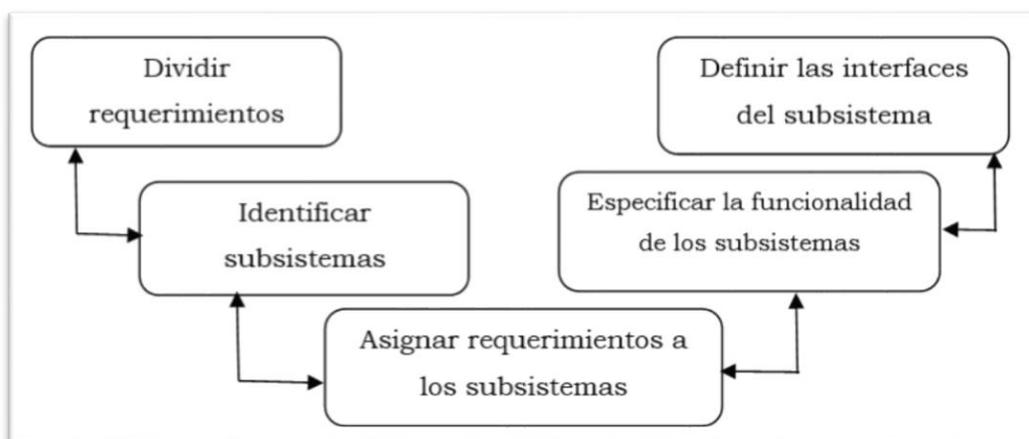


Grafico 2.1. diseño de sistema (Sommerville, 2005)

D. Implementación

Para Kendall y Kendall (2011), afirman que la fase de implementación del sistema, consiste en traducir la especificación de diseño, en una serie de instrucciones escritas en un lenguaje de programación con la finalidad de que se cumplan los requerimientos y que previamente se realice sus análisis correspondientes.

Por otro lado Sommerville (2005), los procesos de implementación del software están diseñados para producir software de forma rápida. Generalmente, son procesos iterativos en los que se entrelazan las especificaciones, el diseño, el desarrollo y pruebas. El software no se desarrolla y utiliza en su totalidad, sino es una serie de incrementos, donde en cada incremento se incluyen nuevas funcionalidades al sistema. A esto se denomina implementación de sistema con una herramienta ágil. Ejemplo, Programación Extrema (XP).

E. Pruebas

Según Bruegge y Dutoit (2010), manifiestan que las pruebas son el proceso de encontrar diferencias entre el comportamiento esperado, especificado de los modelos de sistema y el comportamiento observado del sistema. Las pruebas unitarias encuentran diferencias entre el modelo de diseño de objetos y sus componentes correspondientes del sistema.

Por otro lado, Pressman (2010), afirma que la prueba es un conjunto de actividades que pueden planearse por adelantado y realizarse de manera sistémica. Por esta razón, durante el proceso de software, debe definirse una plantilla para la prueba del software: un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de caso de prueba específicos. Una estrategia de prueba de software debe incluir pruebas de bajo nivel, que son necesarias para verificar que un pequeño segmento de código fuente se implementó correctamente, así como prueba de alto nivel, que validan las principales funciones del sistema a partir de los requerimientos del cliente.

El proceso de prueba de aplicaciones web comienza con pruebas que ejercitan la funcionalidad del contenido y la interfaz que son inmediatamente visibles para el usuario final. Conforme se avanza la prueba, se ejercitan aspectos de la arquitectura del diseño y de la navegación. Finalmente, la atención se centra en las pruebas que examinan las capacidades tecnológicas que no siempre son aparentes para los usuarios finales: los temas de infraestructura e instalación, implementación de las aplicaciones

web.

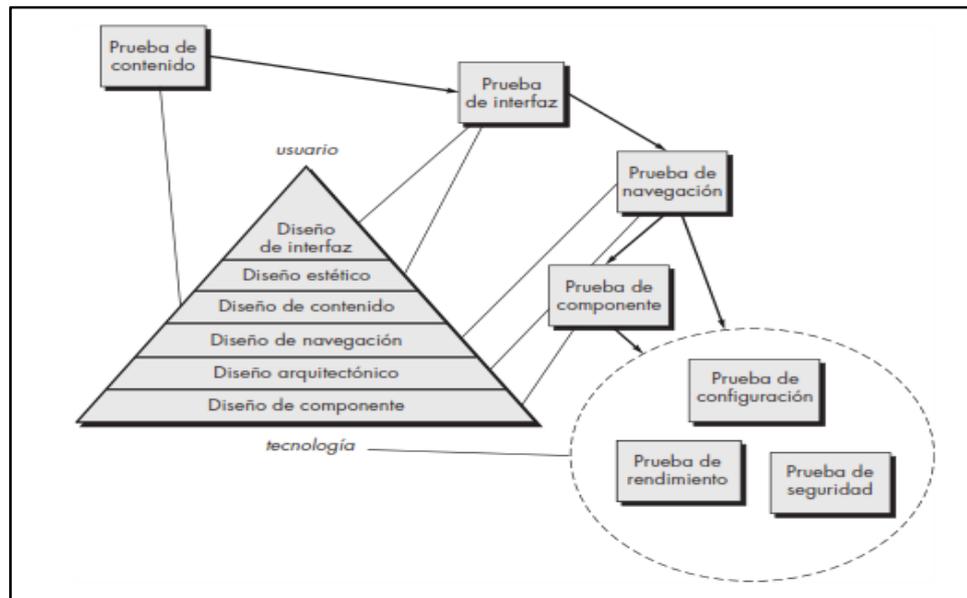


Grafico 2.2: Prueba se aplicaciones Web (Pressman, 2010)

2.2.3 METODOLOGIA DE PROGRAMACION EXTREMA.

Para Kent (2002), la programación extrema es una metodología ágil, eficiente y de bajo riesgo, flexible, predecible, científico, y una manera divertida de desarrollar software. La metodología XP es una disciplina de desarrollo de software.

Por otro lado, Beck (2000), la programación extrema es una metodología de desarrollo ligera (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas. Este modelo de programación se basa en una serie de metodologías de desarrollo de software en la que se da prioridad a los trabajos que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación.

A. VALORES EN XP

Según Beck (1999), la programación extrema se basa en cuatro valores, que deben estar presentes en el equipo de desarrollo para que el proyecto tenga éxito, siendo los siguientes:

COMUNICACIÓN

Beck, (1999), manifiesta que muchos de los problemas que existen en proyectos de software, se deben a problemas de comunicación entre las personas. La comunicación

permanente es fundamental en XP, dado que los artefactos son pocos, el diálogo frontal, cara a cara, entre desarrolladores, administrador y el cliente es el medio básico de comunicación. Una buena comunicación se debe mantener durante todo el proyecto.

Por otro lado, Ricardo (2016), menciona que en los métodos tradicionales de desarrollo de software, la comunicación de los requerimientos a los desarrolladores se realiza a través de la documentación, por ejemplo las Especificaciones de Diseño en el Rational Unified Process (RUP). XP rompe con este esquema, la comunicación se realiza por medio de transferencia de conocimientos en reuniones frecuentes cara a cara entre usuarios y desarrolladores, lo que da a ambos una visión compartida del sistema.

SIMPLICIDAD

Para Beck (1999), el proceso XP, es una metodología ágil, que apuesta por la sencillez, en su máxima expresión. Sencillez en diseño, en código, en los procesos, etc. La sencillez es fundamental para que todos entiendan el código y, se trata de mejorar mediante re-codificaciones continuas.

Por otro lado, Ricardo (2016), afirma que en XP se comienza desarrollando las soluciones más sencillas necesarias para solucionar los problemas (requerimientos) que se están viendo en ese momento, añadiendo funcionalidad extra más tarde, en la medida en que se obtiene más información de los requerimientos. La diferencia respecto a esquemas tradicionales es que se enfoca en las necesidades de hoy en lugar de las necesidades de mañana, la semana que viene o el mes que viene.

RETROALIMENTACIÓN

Según Porras (2010), la retroalimentación debe practicarse en forma permanente. El cliente debe brindar retroalimentación de las historias de usuario desarrolladas, a fin de considerar sus comentarios para la siguiente iteración, y para entender, cada vez más, sus necesidades. Los resultados de las pruebas unitarias, son también una retroalimentación permanente que tienen los desarrolladores acerca de la calidad de la aplicación.

Beck (1999), bajo este esquema, las fallas de sistema se pueden comunicar fácilmente, pues existen pruebas unitarias que demuestran que el sistema fallará si es puesto en producción. Asimismo, un cliente puede probar el sistema periódicamente, contrastando el funcionamiento con sus requerimientos funcionales o “Historias de usuario”.

CORAJE

Según Beck (1999), cuando se encuentran problemas serios en el diseño, o en cualquier fase del ciclo de XP, se debe tener el coraje suficiente para encontrar la solución, sin importar que tan difícil sea. Si es necesario cambiar completamente parte del código, hay que hacerlo, sin importar cuánto tiempo se ha invertido en desarrollar el código a cambiar.

Por otro lado, Ricardo (2016), la verdad en nuestros avances y estimados, no documentaremos excusas para el fracaso, pues planificamos para tener éxito. No tendremos miedo a nada pues sabemos que nadie trabaja solo. Nos adaptaremos a los cambios cuando sea que estos ocurran.

RESPETO

Según Kent (2002), el respeto se manifiesta de varias formas. Los miembros del equipo se respetan los unos a otros, porque los programadores no pueden realizar cambios que hacen que las pruebas existentes fallen o que demore el trabajo de sus compañeros. Los miembros respetan su trabajo porque siempre están luchando por la alta calidad en el producto y buscando el diseño óptimo o más eficiente para la solución a través de la refactorización del código. Los miembros del equipo respetan el trabajo del resto no haciendo menos a otros, una mejor autoestima en el equipo eleva su ritmo de producción.

B. ROLES DE LOS INTEGRANTES DE XP

ROL DEL CLIENTE

Según Jeffries, Anderson y Hendrickson (2001), el cliente elige lo que va a entregar, decide qué hacer primero y qué aplazar y, define las pruebas para demostrar que el sistema hace lo que necesita. El equipo será más eficaz si el cliente permanece en el lugar y esté presente con el equipo. El cliente, tiene la responsabilidad fundamental de elegir las historias de elementos más valiosos, de más alto valor comercial. Finalmente, especifica las pruebas que muestran si las historias se han desarrollado correctamente, las pruebas de aceptación, ya está construido por los programadores, por un testeador independiente, o por los clientes mismos.

Por otro lado, Kent (2002), afirma, escribe las historias de usuario y las pruebas funcionales para validar su implementación. Asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar el

mayor valor de negocio.

ROL DEL PROGRAMADOR

Para Jeffries et al. (2001), los programadores analizan, diseñan, prueban el programa, e integran el sistema. Los programadores estiman la dificultad de todas las historias y, realizan el seguimiento del ritmo al que pueden ofrecer las historias para el cliente.

Desarrolladores trabajan con el cliente para entender sus historias. De una historia, los desarrolladores decidan su aplicación. Los desarrolladores luego estiman la cantidad de trabajo que cada historia tendrá, en base a las decisiones de implementación y su experiencia en el proyecto hasta el momento. Estas estimaciones ayudan al cliente para programar la obra más valiosa para la próxima iteración, respondiendo a la pregunta de cuánto tiempo. (O'REILLY, s.f.).

ROL DEL ENCARGADO DE PRUEBAS (TESTER)

Para Kent, (2002), es quien ejecuta las pruebas y luego informa los resultados al equipo, además ayuda al cliente a escribir las pruebas funcionales. Es alguien que tiene que ejecutar todas las pruebas de forma regular (si no puede hacer funcionar su unidad y pruebas de funcionamiento en conjunto), resultados de prueba de difusión, y para asegurarse de qué herramientas de prueba funciona bien.

Por otro lado, Batalla (2006), afirma que el encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

ROL DEL ENCARGADO DE SEGUIMIENTO (TRACKER)

Kent (2002), es el que realiza el seguimiento del progreso de cada iteración y proporciona la realimentación al equipo de trabajo. Mantiene un registro de los resultados de las pruebas funcionales.

El tracker realiza el seguimiento de la programación. XP tracker de unos pocos indicadores. El más importante es la velocidad del equipo, que es la relación de momento ideal estimado para las tareas al tiempo real dedicado implementarlas. Otros datos importantes pueden incluir cualquier cambio en la velocidad, la cantidad de horas extras trabajadas, y la proporción de pasar las pruebas de pruebas fallidas. (O'REILLY, s.f.).

ROL DEL ENTRENADOR (COACH)

Para Kent (2002), es el responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente. El coach guía a su equipo a comprender XP y el software desarrollo. A veces se enseña directamente. A veces enrolla las mangas y se enseña haciendo. Se puede sugerir cambios en cómo se implementa una práctica, ofrecen ideas a resolver un problema técnico de espaldas, o servir de intermediario entre el equipo y la gestión de otros.

ROL DEL CONSULTOR

Según Kent (2002), es un miembro externo del equipo, quien posee conocimiento en algún tema necesario para el proyecto.

Por otro lado para, Batalla (2006), el consultor es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

ROL DEL ADMINISTRADOR

Para Jeffries et al. (2001), el administrador hace que el cliente y los desarrolladores estén juntos y los ayuda a participar en el funcionamiento correcto del equipo. Cuando se trata del proceso de planificación, diseño, pruebas, codificación, liberación, los administradores no realizan ninguna de estas cosas directamente. El administrador promoverá las cosas por hacer, coordinar las tareas, e informará los resultados. Como administrador, promoverá una sesión rápida, antes de la liberación de la planificación. Si hay conflictos en la programación, debe ponerse de acuerdo con los miembros del equipo y encontrar una fecha adecuada para la culminación de la historia. Si es necesario, fijar otra cita cuando existe conflicto. Es el dueño del equipo y sus problemas. Persona experta en tecnología y labores de gestión, su labor esencial es de coordinación. Es la imagen del equipo al exterior. Elige los miembros que conformaran el plantel, obtiene los recursos necesarios y maneja los problemas que se generan. Agenda reuniones (planes de iteración, agenda de compromisos, etc), verifica que se realicen de manera adecuada y registra lo referente a las mismas.

Según Batalla (2006), el administrador no le dice al grupo lo que tiene que hacer (el Cliente y el plan de iteración lo hacen), cuando hacerlo (la agenda de compromisos lo hace), ni verifica el avance de las tareas (Tracker).

C. CICLO DE VIDA DE UN PROYECTO XP

Para Kent y Fowler (2000), el ciclo de vida ideal de la programación extrema, consiste de seis fases, como presentamos a continuación.

FASE I: EXPLORACIÓN

Según Kent y Fowler (2000), en esta fase, los clientes plantean a grandes rasgos las historias de usuario, que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Por otro lado, Joskowicz, (2008), manifiesta; es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración.

FASE II: PLANIFICACIÓN DE LA ENTREGA

Kent y Fowler (2000), afirman que, en esta fase el cliente establece la prioridad de cada historia de usuario y, los programadores realizan una estimación del esfuerzo necesario para desarrollar cada una de las historias de usuario. Se toman acuerdos sobre el contenido de la primera iteración (entrega) y se determina un cronograma con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias de usuario, la fijan los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación, las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es

utilizada para establecer cuántas historias de usuario se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

Por otro lado, Joskowicz (2008), afirma sobre la planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas, o “Release Plan”, como se detallará en la sección “Reglas y Practicas”.

FASE III: ITERACIONES

Kent y Fowler (2000), manifiestan que, la fase incluye varias iteraciones sobre el sistema en construcción antes de ser entregado. El plan de entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura. Sin embargo, esto no siempre es posible ya que el cliente es quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Por otro lado, Joskowicz (2008), menciona, esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios.

FASE IV: PRODUCCION

Para Kent y Fowler (2000), la fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes que el sistema sea instalado en el entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas

características a la versión actual, debido a cambios durante esta fase. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (durante la fase de mantenimiento).

Según Joskowicz (2008), si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa. En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste (“fine tuning”).

FASE V: MANTENIMIENTO

Kent y Fowler (2000), manifiestan, mientras que la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Por otro lado afirman, Letelier y Penadés (2006), mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

FASE VI: MUERTE DEL PROYECTO

Según Kent y Fowler (2000), la muerte del proyecto es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Según Letelier y Penadés (2006), esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La

muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

D. REGLAS BÁSICAS DE XP

El proceso programación extrema, tiene un conjunto importante de reglas y prácticas, agrupadas según (Beck, 1999)

Para Jeffries et al. (2001), siendo; reglas y prácticas para la planificación, reglas y prácticas para el diseño, reglas y prácticas para el desarrollo, reglas y prácticas para las pruebas.

a) PLANIFICACIÓN

Según Jeffries et al. (2001), el proceso XP plantea la planificación mediante el diálogo continuo entre los integrantes del proyecto que son; cliente, programadores, coordinadores y administrador. El proyecto comienza recopilando “historias de usuarios”, que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “historias de usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de las historias presenta “riesgos” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba (“spikes”), para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los actores del proyecto (cliente, desarrolladores, administrador), a efectos de establecer un plan o cronograma de entregas (“Release Plan”) en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala algunas “historias de usuarios”.

Por otro lado, Joskowicz (2008), afirma, que la metodología XP plantea la planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando “Historias de usuarios”, las que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “historias de usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

HISTORIA DE USUARIOS

Según Jeffries et al. (2001), los clientes tienen derecho a obtener el máximo valor posible de cada momento de programación. Los programadores tienen derecho a saber lo que se necesita. Estos dos derechos se reúnen en la historia de usuario. Cada

historia del usuario es una breve descripción del comportamiento del sistema, desde el punto de vista del usuario del sistema. En XP, el sistema está totalmente especificado a través de historias. El análisis de las historias del usuario es el medio de comunicación entre el cliente y el programador.

De la misma manera, Joskowicz (2008), afirma, las “Historias de usuarios” (“User stories”) sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios.

PLAN DE ENTREGAS (“RELEASE PLAN”)

Según Jeffries et al. (2001), el cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega y, el orden de las mismas. Este cronograma será el resultado de un acuerdo entre todos los actores del proyecto (cliente, desarrolladores, administradores, etc.). En el proceso XP se denomina a esta reunión “Juego de planeamiento” (“Planninggame”), pero puede denominarse de una forma apropiada al tipo de empresa y cliente (Por ejemplo, Reunión de planeamiento, “Planningmeeting” o “Planningworkshop”). El cliente ordenará y agrupará según sus necesidades las historias de usuario.

Por otro lado, Joskowicz (2008), menciona, que el cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.). XP denomina a esta reunión “Juego de planeamiento” (“Planning game”), pero puede denominarse de la manera que sea más apropiada al tipo de empresa y cliente (por ejemplo, Reunión de planeamiento, “Planning meeting” o “Planning workshop”).

PLAN DE ITERACIONES (“ITERATION PLAN”)

Según Jeffries et al. (2001), las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración, cada historia de usuario se traduce en tareas específicas de programación. Por otro lado, Joskowicz (2008), menciona, para cada historia de usuario se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores.

REUNIONES DIARIAS DE SEGUIMIENTO (“STAND-UP MEETING”)

Para Jeffries et al. (2001), el objetivo de tener reuniones diarias es mantener la comunicación entre el equipo y, compartir problemas y soluciones. En la mayoría de estas reuniones, gran parte de los participantes solo escuchan, sin tener mucho que aportar. Para no quitar tiempo innecesario al equipo, se sugiere realizar estas reuniones en círculo y de pie.

Por otro lado, Joskowicz (2008), menciona, el objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar. Para no quitar tiempo innecesario del equipo, se sugiere realizar estas reuniones en círculo y de pie.

b) DISEÑO

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

SIMPLICIDAD

Según Jeffries et al. (2001), un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione, se sugiere que nunca debe adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

Por otro lado, Anaya (2007), menciona, que el sistema se diseña con la máxima simplicidad posible (YAGNY - "No vas a necesitarlo"), Se plasma el diseño en tarjetas CRC (Clase – Responsabilidad - Colaboración), no se implementan características que

no son necesarias, con esta técnica, las clases descubiertas durante el análisis pueden ser filtradas para determinar qué clases son realmente necesarias para el sistema.

SOLUCIONES “SPIKE”

Para Jeffries et al. (2001), al ocurrir problemas técnicos, o cuando es complejo estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados “spike”), para explorar diferentes soluciones. Estos programas solo sirven para probar o evaluar una solución y, son descartados luego de su evaluación.

Según Joskowicz (2008), cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados “spike”¹), para explorar diferentes soluciones. Estos programas son únicamente para probar o evaluar una solución, y suelen ser desechados luego de su evaluación.

RECODIFICACIÓN

Según Jeffries et al. (2001), la recodificación (“refactoring”), consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a efectos de hacerlo más simple, concreto y/o entendible. Muchas veces, al terminar de escribir un código de programa, pensamos que, si lo hacemos de nuevo, lo haríamos de forma diferente, más clara y eficientemente. Sin embargo, como “funciona”, rara vez es reescrito.

Por otro lado, Joskowicz (2008), menciona, que las metodologías de XP sugieren recodificar cada vez que sea necesario. Si bien, puede parecer una pérdida de tiempo innecesaria en el plazo inmediato, los resultados de ésta práctica tienen sus frutos en las siguientes iteraciones, cuando sea necesario ampliar o cambiar la funcionalidad. La filosofía que se persigue es, como ya se mencionó, tratar de mantener el código más simple posible que implemente la funcionalidad deseada.

DESARROLLO DEL CODIGO

DISPONIBILIDAD DEL CLIENTE

Para Jeffries et al. (2001), uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto, no solo como apoyo a los desarrolladores, sino formando parte del grupo, el cliente involucrado es fundamental para desarrollar un

proyecto con el proceso XP. Al inicio del proyecto, el cliente debe escribir las historias de usuarios. Las historias en este momento son cortas y de “alto nivel”, no tienen los detalles necesarios para realizar el desarrollo del código.

De acuerdo a Joskowicz (2008), si bien esto parece demandar del cliente recursos por un tiempo prolongado, debe tenerse en cuenta que en otras metodologías este tiempo es insumido por el cliente en realizar los documentos detallados de especificación. Adicionalmente, al estar el cliente en todo el proceso, puede prevenir a tiempo de situaciones no deseables, o de funcionamientos que no eran los que en realidad se deseaban. En otras metodologías, estas situaciones son detectadas en forma muy tardía del ciclo de desarrollo, y su corrección puede llegar a ser muy complicada.

PROGRAMACIÓN GUIADA POR LAS PRUEBAS (“TEST-DRIVEN PROGRAMMING”)

Para Jeffries et al. (2001), en las metodologías tradicionales, la fase de pruebas, incluyendo la definición de las pruebas, es realizada al final del proyecto, al final del desarrollo de cada módulo. El proceso XP propone un modelo inverso, lo primero que se escribe son los test que el sistema debe pasar, luego el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas.

De acuerdo a Joskowicz (2008), las pruebas a los que se refieren esta práctica, son las pruebas unitarias, realizados por los desarrolladores. La definición de estos test al comienzo, condiciona o dirige el desarrollo.

PROGRAMACIÓN EN PARES

Para Joskowicz (2008), XP propone codificar en pares de programadores, ambos trabajando juntos en una misma computadora, ésta práctica aparentemente duplica el tiempo asignado al proyecto y, por ende, los costos en recursos humanos, al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas.

Por otro lado, Letelier y Penadés (2006), manifiestan que toda la producción de código debe realizarse con trabajo en parejas de programadores, en un estudio realizado para identificar los costos y beneficios de la programación en parejas, las principales ventajas de introducir este estilo de programación son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y

el tamaño del código menor (continua discusión de ideas de los programadores), los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutan más su trabajo.

INTEGRACIÓN PERMANENTE

Según Joskowicz (2008), todos los desarrolladores necesitan trabajar siempre con la “última versión”, realizar cambios o mejoras sobre versiones antiguas originan graves problemas y, retrasan al proyecto, por eso XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores. Idealmente, todos los días deben existir nuevas versiones publicadas, para evitar errores, solo una pareja de desarrolladores puede integrar su código a la vez.

Por otro lado, Letelier y Penadés (2006), afirman, que cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido.

PROPIEDAD COLECTIVA DEL CÓDIGO

Para Joskowicz (2008), en un proyecto XP, todo el equipo puede contribuir con nuevas ideas para aplicar a cualquier parte del proyecto, cualquier pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o re codificar. En otras metodologías, este concepto parece extraño, muchas veces se asume que, si hay algo de propiedad colectiva, la responsabilidad también es colectiva y que “todos sean responsables”, muchas veces significa que “nadie es responsable”.

Por otro lado, Letelier y Penadés (2006), afirman, cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.

SEMANA DE 40 HORAS

Para Joskowicz (2008), lo importante no es si se trabajan, 35, 40 o 42 horas por semana, el concepto de esta práctica, es planificar el trabajo para mantener un ritmo constante y razonable, sin sobrecargar al equipo. Cuando un proyecto se retrasa, trabajar tiempo extra puede ser más perjudicial que beneficioso, el trabajo extra desmotiva inmediatamente al grupo e impacta en la calidad del producto.

Por otro lado, Letelier y Penadés (2006), afirman, que se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo. Los proyectos que requieren trabajo extra para intentar cumplir con los plazos suelen al final ser entregados con retraso. En lugar de esto se puede realizar el juego de la planificación para cambiar el ámbito del proyecto o la fecha de entrega.

c) PRUEBAS

PRUEBAS UNITARIAS

Para Joskowicz (2008), todos los módulos deben pasar las pruebas unitarias, antes de ser liberados o publicados, las pruebas unitarias deben ser definidas antes de realizar el código (“Test-driven programming”).

De acuerdo Letelier y Penadés (2006), las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

DETECCIÓN Y CORRECCIÓN DE ERRORES

Para Joskowicz (2008), cuando se encuentra un error (“bug”), éste debe ser corregido inmediatamente y, se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.

PRUEBAS DE ACEPTACIÓN

Joskowicz (2008), explica que las pruebas de aceptación son consideradas como “pruebas de caja negra” (“Black box system tests”). Los clientes son responsables de

verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información.

Por otro lado, Jeffries et al. (2001), explica, que con las pruebas de aceptación, antes que cometer el error de captura, más pronto podemos hacer que el programa funcione. El cliente es responsable de proporcionar esas pruebas de aceptación como parte de cada iteración. Si puede conseguir que los programadores hagan la mitad de cada iteración, el proyecto avanzara más rápido. Las pruebas de aceptación deben ser automatizados: insistir en ella como su derecho.

E. ARTEFACTOS XP HISTORIAS DE USUARIO

Para Anaya (2007), los historia de usuario representan una breve descripción del comportamiento del sistema, emplea terminología del cliente sin lenguaje técnico, se realiza una por cada característica principal del sistema, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación.

Customer Story and Task Card BIW Development / COLA

DATE: 3/19/98 TYPE OF ACTIVITY: NEW: FIX: ENHANCE: FUNC. TEST:

STORY NUMBER: ~~1275~~ / 1275 PRIORITY: USER: TECH:

PRIOR REFERENCE: _____ RISK: _____ TECH ESTIMATE: _____

TASK DESCRIPTION:
 SPLIT COLA: When the COLA rate chgs. in the middle of the BIW Pay Period we will want to pay the 1ST week of the pay period at the OLD COLA rate and the 2ND week of the Pay Period at the NEW COLA rate. Should occur "automatically based on system design.

NOTES:
 For the OT, we will run a m/ frame program that will pay or calc the COLA on the 2ND week of OT. The plant currently retransmits the hours data for the 2ND week exclusively so that we can calc COLA. This will come into the Model as a "2144" COLA

TASK TRACKING: Gross Pay Adjustment. Create RM Boundary and Place in DE Ent Excess COLA

Date	Status	To Do	Comments

Figura N° 2.3: Modelo propuesto para una historia de usuario (Beck, 2000).

TAREAS DE INGENIERÍA

Las task card se usan para describir las tareas que se realizan sobre el proyecto. Las tareas pueden ser: desarrollo, corrección, mejora, etc. Estas tareas tienen relación con una historia de usuario; se especifica la fecha de inicio y fin de la tarea, se nombra al programador responsable de cumplirla y describimos que se tratara de hacer en la tarea. (wikispace.com, 2016).

De acuerdo a Jeffries et al. (2001), las tareas de ingeniería son otro artefacto de la metodología XP, en este caso, se utilizó el modelo propuesto por Letelier y otros. Las tareas de ingeniería se establecieron, indicando por cada una de las iteraciones la forma en que se desarrollaría la aplicación.

PRUEBAS DE ACEPTACIÓN

Anaya (2006), explica la prueba de aceptación permite confirmar que la historia ha sido implementada correctamente. El uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de empezar a usarlo es muy difícil determinar si sus ventajas realmente justifican su uso. El mejor instrumento para esta determinación es la llamada 'prueba de aceptación'. En esta prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique.

Por otro lado, Kent (2002), afirma que, la prueba de aceptación es una prueba formal conducida para determinar si un sistema satisface los criterios de aceptación y permite al cliente determinar si acepta o no el sistema.

TARJETAS CRC

Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. En ellas se expresa el diseño del sistema, la sencillez de esta tarjeta hace del diseño una tarea fácil. (wikispace.com, 2016).

Según Jeffries et al. (2001), en esta etapa se busca describir las responsabilidades que tiene que cumplir cada clase y las colaboraciones entre ellas para poder cumplir con las historias de usuario. A medida que se van obteniendo las responsabilidades y las colaboraciones se anotan en la tarjeta CRC. Los campos a completar corresponden al nombre del proyecto en el cuál se está trabajando, la fecha en la cual se realiza la sesión de diseño y los datos respectivos a la tarjeta CRC.

PRUEBAS UNITARIAS

Según Jeffries et al. (2001), las pruebas unitarias y de aceptación como principales herramientas para dar soporte a los enfoques de pruebas. Ambas se aplican de forma implícita en el proceso de desarrollo, es decir, cada nueva parte del código se escribe y se integra con el sistema completo, por lo tanto, no es necesario escribir pruebas específicas de integración sino que éstas se encuentran incorporadas en el resto de pruebas que se han desarrollado. Por lo tanto, para poder dar soporte a estos enfoques de pruebas es necesario disponer de una mínima infraestructura que permita automatizar la ejecución del elevado número de pruebas que se diseñan y la toma de medidas que permitan conocer el nivel de calidad que está alcanzando el desarrollo. Esta infraestructura mínima recibe el nombre de entorno de integración continua. Estos entornos se caracterizan por la integración de múltiples herramientas que dan soporte al respaldo automatizado del código, su compilación, la ejecución de las pruebas, la toma de medidas de calidad, la generación de documentación y el despliegue de la aplicación.

PLAN DE ENTREGA

Según Jeffries et al. (2001), el plan de entregas es un documento que especifica exactamente que historias de usuario serán implementadas en cada entrega del sistema y sus prioridades, de modo que también permita conocer con exactitud qué historias de usuario serán implementadas en la próxima liberación. Debe ser negociado y elaborado en forma conjunta entre el cliente y el equipo desarrollador durante las reuniones de planificación de entregas, la idea es hacer entregas frecuentes para obtener una mayor retroalimentación. A continuación, se describen las principales actividades que se deben llevar a cabo, y los documentos que deben ser completados tanto por el cliente como por el equipo desarrollador como una pieza fundamental de la reunión de planificación de entregas. Una vez que la reunión de planificación ha concluido y han sido completados los documentos precedentes como actividades propias de la reunión, el documentador debe completar el plan que se presenta a continuación con la información correspondiente, éste debe ser debidamente aprobado tanto por el equipo desarrollador a cargo de su entrenador como por el cliente.

CÓDIGOS

Según Jeffries et al. (2001), el código es una propiedad colectiva compartida por todos los miembros del equipo de desarrollo. Según lo anterior, cualquier desarrollador puede cambiar cualquier línea de código, añadir funcionalidad, arreglar fallos o aplicar refactoring. Esta forma de trabajo debiera animar a todos los miembros del equipo de desarrollo a contribuir y aplicar refactoring con nuevas ideas en todos los apartados del sistema, con lo cual las personas dejarían de ser un cuello de botella en cuanto a la programación. El objetivo fundamental de esta forma de trabajo es reducir al mínimo las obstrucciones para distribuir e implementar rápidamente las tareas de programación.

2.2.4 PROGRAMACION ORIENTADA A OBJETOS.

De acuerdo a Llovet (2009), la POO se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo. No debemos confundir que esté basado en objetos con que sea orientado a objetos: para que sea orientado a objetos al margen que esté basado en objetos, necesita tener clases y relaciones de herencia entre ellas. El elemento fundamental de la POO es, como su nombre indica, el objeto. Podemos definir un objeto como un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

Según, Joyanes (2007), es una filosofía, un modelo de programación, con su teoría y su metodología, que conviene conocer y estudiar, antes de nada. Un lenguaje orientado a objetos es un lenguaje de programación que permite el diseño de aplicaciones orientadas a objetos. Dicho esto, lo normal es que toda persona que vaya a desarrollar aplicaciones orientadas a objetos aprenda primero la “filosofía” (o adquiera la forma de pensar) y después el lenguaje, porque “filosofía” sólo hay una y lenguajes muchos. En este documento veremos brevemente los conceptos básicos de la programación orientada a objetos desde un punto de vista global, sin particularizar para ningún lenguaje de programación específico.

De la misma manera, Izquierdo (2007), explica la programación orientada a objetos es otra forma de descomponer problemas. Este nuevo método de descomposición es la descomposición en objetos; vamos a fijarnos no en lo que hay que hacer en el problema, sino en cuál es el escenario real del mismo, y vamos a intentar simular ese escenario en nuestro programa. Los objetos en programación se usan para modelar objetos o

entidades del mundo real (el objeto hijo, madre, o farmacéutica, por ejemplo). Un objeto es, por tanto, la representación en un programa de un concepto, y contiene toda la información necesaria para abstraerlo: datos que describen sus atributos y operaciones que pueden realizarse sobre los mismos. La siguiente figura muestra una representación visual de un objeto. Una clase es una plantilla que define las variables y los métodos que son comunes para todos los objetos de un cierto tipo. El mecanismo de herencia permite definir nuevas clases partiendo de otras ya existentes. Las clases que derivan de otras heredan automáticamente todo su comportamiento, pero además pueden introducir características particulares propias que las diferencian.

2.2.5 BASE DE DATOS RELACIONAL.

De acuerdo a Heurtel (2009), una base de datos relacional presenta una organización de los datos basada en el modelo relacional desarrollado en 1970 por Edgar COOD. Es la estructura más extendida actualmente. Los datos se organizan en tablas enlazadas de manera lógica. Una tabla incluye columnas (o campos) que describen una fila (o registro). La relación entre las tablas se establece mediante una columna.

Por otro lado, Osorio (2008), explica es un paradigma que se ha adoptado en las tecnologías de la información, ninguno como el modelo relacional de las bases de datos se ha consolidado de una manera tan categórica y unánime, pudiéndose decir que la actual orientación a objetos debe su éxito a la consolidación de este modelo en la implementación de las bases de datos.

También manifiesta Rob (2004), es un depósito de datos en el que se mantiene su independencia. Sin embargo, el modelo de bases de datos relacional ofrece varias ventajas importantes como independencia estructural, simplicidad conceptual mejorada, diseño, ejecución, administración y uso más fácil de la base de datos.

2.2.6 ADMINISTRACION DE BASE DE DATOS RELACIONAL.

Para Sánchez (2007), un sistema gestor de bases de datos o SGBD (aunque se suele utilizar más a menudo las siglas DBMS procedentes del inglés, Data Base Management System) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. En estos Sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos,

garantizando además la seguridad de los mismos.

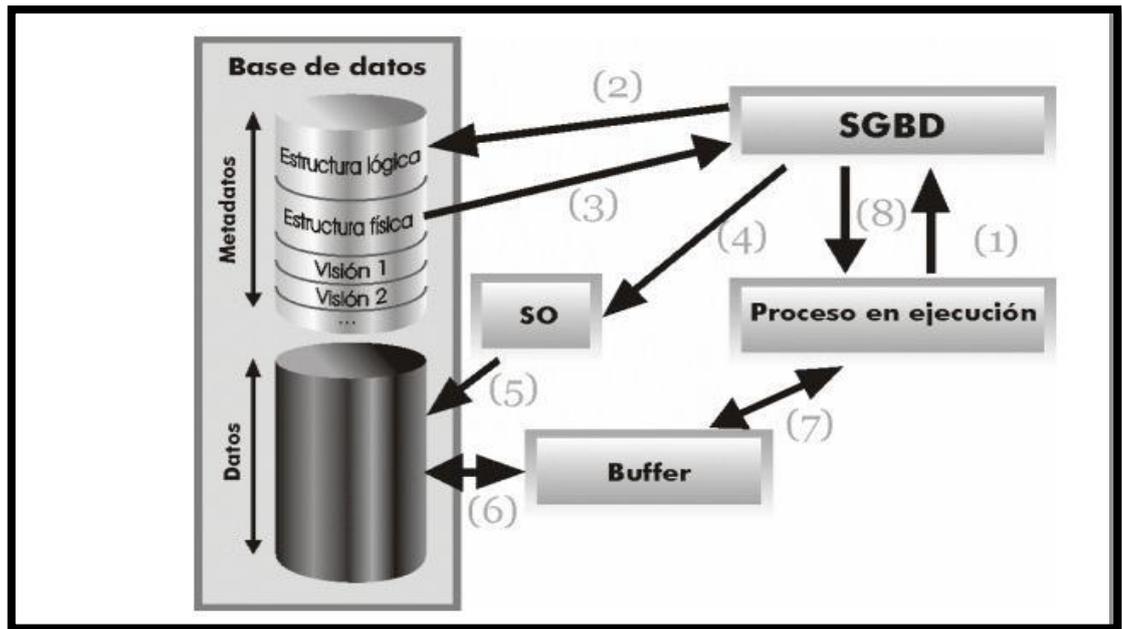


Figura N^o 2.4: Esquema del funcionamiento de un SGBD (Sánchez, 2007).

Para Mannino (2007), un sistema de administración de bases de datos es un grupo de programas que se usan como interfaz entre una base de datos y programas de aplicaciones, entre una base de datos y el usuario los administradores de base de datos se clasifican por el tipo de modelo de base de datos a los cuales da soporte con independencia del modelo que respalden los sistemas de administración de datos, algunas funciones comunes tales como proporcionar una vista de datos para el usuario, almacenar y recuperar físicamente los datos en una base de datos ,permite la modificación de la base de datos, manipular datos y elaborar informes.

Una base de datos relacional es básicamente un conjunto de tablas, similares a las tablas de una hoja de cálculo, formadas por filas (registros) y columnas (campos). Los registros representan cada uno de los objetos descritos en la tabla y los campos los atributos (variables de cualquier tipo) de los objetos. En el modelo relacional de base de datos, las tablas comparten algún campo entre ellas, estos campos compartidos van a servir para establecer relaciones entre las tablas. (www.um.es, s.f.).

2.2.7 MODELO VISTA CONTROLADOR

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC

propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.^{1 2} Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. (Wikipedia.org, 2016).

De acuerdo a Cabellé (2005), el modelo vista controlador es un patrón de diseño de aplicaciones que permite conseguir un alto grado de modularidad en las aplicaciones, en general y muy especialmente en la interfaz de usuario. Esencialmente, usando este patrón se puede realizar una clara separación de la aplicación en tres partes independientes que colaboran y se comunican entre sí para realizar la tarea.

MODELO

Para Cabellé (2005), es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tantas consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del "controlador.

VISTA

Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida. (Wikipedia.org, 2016).

Por otro lado, Cabellé (2005), la parte donde se interacciona con el usuario, se especifican cosas como posición de datos, y como se desplegarán.

CONTROLADOR

Según Cabellé (2005), pone orden entre los dos anteriores decide cuando se hace una llamada de datos, y cuando se despliega algo.

Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista'

asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo' (véase Middleware). (Wikipedia.org, 2016).

2.2.8 TECNOLOGIA DE INTERNET

De acuerdo a Silva y Mercerat (2002), el desarrollo de aplicaciones web involucra decisiones no triviales de diseño e implementación que inevitablemente influyen en todo el proceso de desarrollo, afectando la división de tareas. Los problemas involucrados, como el diseño del modelo del dominio y la construcción de la interfaz de usuario, tienen requerimientos disjuntos que deben ser tratados por separado. El alcance de la aplicación y el tipo de usuarios a los que estará dirigida son consideraciones tan importantes como las tecnologías elegidas para realizar la implementación. Así como las tecnologías pueden limitar la funcionalidad de la aplicación, decisiones de diseño equivocadas también pueden reducir su capacidad de extensión y reusabilidad. Es por ello que el uso de una metodología de diseño y de tecnologías que se adapten naturalmente a ésta, son de vital importancia para el desarrollo de aplicaciones complejas. Existen en la actualidad tecnologías ampliamente usadas para el desarrollo de aplicaciones web, pero muchas de ellas obligan al desarrollador a mezclar aspectos conceptuales y de presentación.

A. HTML (HyperTextMarkupLanguage)

Para Gauchat (2003), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares.

B. JSP(JAVA SERVER PAGES)

Para Holzner (2008), es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Esta tecnología es un desarrollo de la compañía Sun Microsystems. La especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1. Las JSP's permiten la utilización de código Java mediante scripts. Además,

es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de librerías de etiquetas (TagLibraries) externas e incluso personalizadas.

C. JAVASCRIPT

De acuerdo a Gauchat (2003), es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

D. AJAX

Para Gauchat (2003), es un enfoque de desarrollo basado en un conjunto de tecnologías ya existentes, agrupadas para presentar información e interactuar dinámicamente, de manera asíncrona, con un servidor Web.

E. CSS

Zevallos (2015), afirma, es un lenguaje de hojas de estilo desarrollado para controlar la presentación de documentos con lenguajes de marcado. Específicamente se utiliza para la presentación, estructuración y formato del contenido de una página web.

F. JAVA.

De acuerdo a Joyanes (2007), es un lenguaje de programación orientado a objetos que posee un sistema que interpreta y ejecuta los archivos de clase compilados en tiempo real, además posee un juego de herramientas de desarrollo y una interfaz de programación de aplicaciones (API). La utilidad del archivo java es el “jar” que genera los archivos java, estos son similares a los archivos zip en que comprimen los archivos a un tamaño más pequeño y proporciona una manera conveniente de distribuir clases de java compilados.

2.2.9 ARQUITECTURA CLIENTE SERVIDOR

Según Pressman (2010), la computación cliente servidor es un intento de equilibrar el proceso de una red hasta que se comparta la potencia de procesamiento entre computadoras que llevan a cabo servicios especializados tales como acceder a bases de datos (servidores), y aquellos que llevar a cabo tareas tales como la visualización IGU que es el más adecuado para el punto final dentro de la red. Por ejemplo, permite que las computadoras se ajusten a tareas especializadas tales como el procesamiento de bases de datos en donde se utilizan hardware y software de propósito especial para proporcionar un procesamiento rápido de base de datos comparado con el hardware que se encuentra en las mainframes que tienen que enfrentarse con una gran gama de aplicaciones.

A. CLIENTE

Según Pressman (2010), un servidor es una computadora que lleva a cabo un servicio que normalmente requiere mucha potencia de procesamiento.

Weitzenfeld (2005), explica, si se considera que el usuario, a través de una computadora local correspondiente al cliente, es el interesado en interactuar con los programas que existen en el Internet, el cliente tiene como función primordial facilitar la interacción del usuario. En otras palabras, el objetivo básico del cliente en la arquitectura cliente servidor es facilitar la presentación y control de la información administrada por la aplicación, algo similar al rol de las clases borde en relación con un actor de tipo usuario. Por tal motivo, la mayoría de las tecnologías que se procesan en el cliente están dirigidas a facilitar la visualización y control de la información, como en el caso de HTML, Flash, Javascript, VBScript, JScript.

B. SERVIDOR

Pressman (2010), explica, un cliente es una computadora que solicita los servicios que proporciona uno o más servidores y que también lleva a cabo algún tipo de procesamiento por sí mismo.

Para Sommerville (2005), es el responsable de prestar los servicios requeridos por los clientes. Es común que la mayor parte de una aplicación en una arquitectura cliente/servidor se encuentre del lado del servidor. Esto se hace por razones de costo,

eficiencia y facilidad para dar servicio a múltiples clientes de manera concurrente. En general en muchos de los lenguajes tradicionales de programación han sido utilizados para programar los servidores, pero existen ciertas tecnologías diseñadas específicamente para arquitecturas cliente servidor en Internet.

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1. TIPO DE INVESTIGACIÓN

Se desarrolló un producto software, “Una aplicación Web para apoyar la Gestión de Procesos Académicos del Instituto de Educación Superior Tecnológico Público Churcampa, 2016”, usando la metodología XP. Desde el punto de vista científico, describir es medir las características fundamentales del objeto en estudio y su descripción detallada de las partes, categorías o clases de dicho objeto (Hernández, 2010).

3.2. DISEÑO DE LA INVESTIGACIÓN

Según Hernández (2010), las investigaciones no experimentales son aquellas que se realizan sin manipular deliberadamente las variables, es decir, no se varía intencionalmente la variable independiente, simplemente lo que se hace es observar las funciones tal y como se dan en su contexto natural para después analizarlo. En la presente investigación, usamos la metodología XP, para construir el producto software, sin modificar las variables de la investigación.

Para Carrasco (2005), los diseños de investigación no experimentales; son aquellos cuyas variables independientes carecen de manipulación intencional, y no poseen grupo de control, ni mucho menos experimental. Analizan y estudian los hechos y fenómenos de la realidad después de su ocurrencia.

Por otro lado, Batista (2004), sobre los diseños de investigación transeccional o transversal manifiesta que se recolectan datos en un solo momento, en un tiempo único. Su propósito es describir variables y analizar su incidencia e interrelación en un momento dado.

3.3. POBLACIÓN Y MUESTRA

POBLACIÓN

Docentes del Instituto de Educación Superior Tecnológico Público Churcampa, Huancavelica.

Total docentes: 16.

MUESTRA

Se tomó un muestreo no probabilístico con juicio de experto y criterio de saturación, la muestra estuvo conformada por:

Docentes: 10.

3.4. VARIABLES E INDICADORES

DEFINICIÓN CONCEPTUAL DE LAS VARIABLES

VARIABLE DE ESTUDIO

X: AUTOMATIZACIÓN DE LA GESTIÓN ACADÉMICA. Realización de manera automatizada (con uso de tecnologías) el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es también la dirección o administración de una compañía o de un negocio. En este caso es un conjunto de actividades relacionadas al aspecto académico dentro de una institución educativa donde se realizan proceso y se generan un conjunto de información.

INDICADORES DE LA VARIABLE EN ESTUDIO

✓ **Automatización de registro de evaluaciones.**

La evaluación es la acción de estimar, calcular o señalar el valor de algo. La evaluación es la determinación sistemática del mérito, el valor y el significado de algo o alguien en función de unos criterios respecto a un conjunto de normas.

✓ **Automatización de indicadores académicos.**

Los indicadores académicos permiten la consecución de las unidades de competencia. Cabe indicar que a cada unidad de competencia le corresponde un conjunto de indicadores académicos.

✓ **Automatización de semestres académicos.**

Semestre es un concepto que tiene su origen etimológico en el vocablo latino semestris, que significa un tiempo de seis meses.

✓ **Automatización de unidades didácticas.**

Las unidades didácticas son unidades de programación de enseñanza con un tiempo determinado. La unidad didáctica es una propuesta de trabajo relativa a un proceso de enseñanza aprendizaje completo. Este modelo didáctico aparece muy ligado a las teorías constructivistas que en la actualidad se están implementando en los institutos tecnológicos.

✓ **Automatización de matrículas.**

En las instituciones educativas, la matriculación, también llamada proceso de matrícula, suele consistir en la mayor parte de los casos en la cumplimentación de los formularios correspondientes y la aportación de la documentación adecuada. Las hojas de formulario se suelen encontrar en las secretarías de los centros de enseñanza. Allí se recogen y se entregan durante el periodo de matrícula. Este tiene lugar un tiempo antes del comienzo de las clases para que la administración del centro tenga tiempo de procesar los datos y organizar la información sobre los estudiantes.

✓ **Automatización de horarios.**

Del latín horarius, el término horario hace referencia a aquello que tiene relación o que pertenece a las horas. Su uso más habitual está vinculado al periodo temporal durante el que se lleva a cabo una actividad académica.

DEFINICIÓN OPERACIONAL DE LAS VARIABLES

Variable	Indicadores
V.1. AUTOMATIZACION DE LA GESTION ACADEMICA	<ol style="list-style-type: none">1. Automatización de registro de evaluaciones2. Automatización de indicadores académicos3. Automatización de semestres académicos.4. Automatización de gestión de unidades didácticas5. Automatización de matrículas6. Automatización de horarios

3.5. TÉCNICAS E INSTRUMENTOS

3.5.1. TÉCNICA PARA RECOLECTAR INFORMACIÓN

Se empleó la técnica de encuesta a la muestra de docentes, en relación a las variables y dimensiones de la investigación.

3.5.2. INSTRUMENTO PARA RECOLECTAR INFORMACIÓN

Se levantó información usando el instrumento cuestionario para la encuesta a los docentes del instituto, que permite levantar información para dar respuesta a las preguntas, el logro de los objetivos, instrumento que se muestra en el anexo B.

3.5.3. HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN

Las herramientas tecnológicas que se utilizaron para implementar la aplicación web, fueron seleccionadas de acuerdo al marco teórico que se indica en el capítulo 2.2. En función al marco teórico seleccionamos las tecnologías según la tabla 3.1.

Tabla N° 3.1: Herramientas tecnológicas para el tratamiento de datos.

SOFTWARE	FABRICANTE	SERVICIO
WINDOWS 8.1	Producida por Microsoft Corporation	Es la versión más reciente de Microsoft Windows, línea de sistemas operativos producida por Microsoft Corporation.
POWERDESIGNER 16.1.0	Desarrollado por Sybase	Es una colaboración modelado empresarial herramienta producida por Sybase. PowerDesigner ejecuta bajo Microsoft Windows como un nativo de la aplicación, y se ejecuta en Eclipse a través de un plug-in. PowerDesigner es compatible con la arquitectura basada en modelos de diseño de software. PowerDesigner utiliza el .pdm formato de archivo.
NETBEANS IDE 8.1	Desarrollado por Sun Microsystems/Oracle	NetBeans es un entorno de desarrollo integrado libre, hecho principalmente

SOFTWARE	FABRICANTE	SERVICIO
	Corporation.	para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE2 es un producto libre y gratuito sin restricciones de uso.
JAVA	Desarrollado por James Gosling & Sun Microsystems (Oracle Corporation)	Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.
POSTGRESQL 9.6	Desarrollado por PostgreSQL Global Development Group	PostgreSQL es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD.
JAVASCRIPT 1.8.5	Diseñado por Netscape Communications Corp & Mozilla Foundation	Es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML.
HTML5	Desarrollado por World Wide Web Consortium & WHATWG	Es un conjunto de símbolos o palabras que definen varios componentes de un documento Web.
CSS3	Desarrollado por W3C	Es un lenguaje artificial usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

SOFTWARE	FABRICANTE	SERVICIO
SPRING MVC 4.1.1	Desarrollado por SpringSource. Licencia: Apache License 2.0	Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java. Basado en consulta URL de http, el DispatcherServlet llama al controlador correspondiente. Una vista es renderizada y enviada a la respuesta http.
BOOTSTRAP	Desarrollado por Twitter-año 2011	Bootstrap es una colección de varios elementos web personalizables y funciones completamente empaquetado en una sola herramienta. Cuando se diseña una web con Bootstrap, los desarrolladores pueden elegir qué elementos utilizar.
JQUERY 1.11	Creado por John Resig y desarrollado por el equipo de desarrollo http://jquery.com/	jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.
JSON	Desarrollado en Javascript bajo el estándar RFC 7159, ECMA-404	JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

SOFTWARE	FABRICANTE	SERVICIO
JQGRID	Desarrollado por .NET	jqGrid es un plugin parrilla por jQuery JavaScript Library. Se es libre, de código abierto y se distribuye bajo la licencia MIT.
iREPORT	Desarrollado por Jaspersoft Corporation,	iReport es el código abierto diseñador de informes libre para JasperReports y JasperReports Server, crear diseños muy sofisticados que contienen gráficos, imágenes, subinformes, tablas de contingencia y mucho más

Fuente: Elaboración Propia.

3.5.4. TÉCNICA PARA APLICAR XP

Se ha revisado las técnicas de XP y confrontado con el marco teórico indicado en capítulo 2.2, formulamos el proceso, que se considera las fases para desarrollar la aplicación web, como se muestra en las tablas 3.2 a 3.4.

Tabla N° 3.2: Fase de exploración.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Escribir historias de usuario	Historia de usuario	Describir brevemente la historia de usuario con la regla del negocio (lo que el sistema debe hacer) Dividir historias de usuario grandes	Cliente
Probar las tecnologías a utilizar	Arquitectura técnica inicial	Explorar posibilidades de uso de tecnologías. Probar el rendimiento de las tecnologías	Cliente Programador Entrenador
Estimar esfuerzo para historias de usuario	Plan de alto nivel.	Conocer previamente la historia de usuario	Hacer una implementación rápida de historia de usuario

Fuente: (Porras, 2010)

Tabla N° 3.3: Fase de planificación.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Rescribir las historias de usuario	Historia de usuario	Describir detalladamente la historia de usuario con la regla del negocio	Cliente
Formular el plan de versiones	Plan de versión (una iteración)	Introducir nuevos requisitos del software Definir prioridad para cada historia de usuario por necesidad del negocio	Cliente
		Utilizar técnicas de elaboración del plan de alto nivel Estimar y asignar esfuerzo (semana) para cada historia de usuario en función a tiempo para planear, diseñar, implementar y probar Estimar y asignar riesgo a cada historia de usuario en función a situación que afecta la estimación del esfuerzo Actualizar tarjeta de historia de usuario	programador

Fuente: (Porras, 2010)

Tabla N° 3.4: Fase de Iteración

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Definir la arquitectura técnica	Arquitectura técnica	Actualizar la arquitectura técnica inicial Usar características del negocio Utilizar arquitectura por capas Integrar frameworks	Cliente Programador Entrenador

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Escribir tareas de ingeniería	Tarea de ingeniería	Dividir cada historia de usuario en tareas, describir usando reglas del negocio cada tarea de ingeniería	Cliente Programador
Formular el plan de iteraciones	Plan de iteración	Estimar y asignar esfuerzo para desarrollar una tarea de ingeniería	Programador
		Asignar una tarea de ingeniería al programador	Entrenador Programador
		Utilizar el plan de versión Actualizar el plan con tareas de ingeniería de la siguiente iteración Actualizar el plan cuando fallo prueba de aceptación Actualizar el plan con tareas no concluidas Actualizar las tarjetas de tarea de ingeniería	Programador Entrenador Supervisar
Crear pruebas de aceptación	Caso de prueba de aceptación	Escribir pruebas de aceptación para cada historia de usuario por iteración	Cliente Encargado de pruebas
Implementar las interfaces	GUI	Diseñar con precisión la GUI relacionada a cada historia de usuario Generar código para la interfase usando una herramienta	Cliente Programador
Escribir tarjetas CRC para cada tarea de ingeniería	Tarjeta CRC	Diseñar para una tarea de ingeniería de forma simple Rediseñar por falla de prueba de aceptación una tarea Identificar responsabilidades Identificar colaboración Identificar atributos	Cliente Programador
Implementar la base datos física	Base de datos física	Escribir script usando tarjetas CRC Ejecutar script usando DBMS	Programador
Implementar código para clases entidad	Código fuente	Escribir código fuente o generar con una herramienta usando tarjetas CRC	Programador

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Crear pruebas unitarias para las clases control	Prueba unitaria	Escribir código fuente para una prueba unitaria, usando una herramienta	Programador
Implementar código fuente	Código fuente	Codificar una tarea de ingeniería Hacer refactoring Mover programadores	Programador Supervisor
Ejecutar pruebas unitarias	Reporte de prueba unitaria	Ejecutar el módulo de cada prueba unitaria Modificar código fuente si la prueba unitaria muestra resultado incorrecto	Programador
Realizar integración continua	Código fuente	Integrar las tareas para una historia de usuario Mantener sistema integrado todo el tiempo	Programador
Ejecutar pruebas de integración para una historia de usuario	Reporte pruebas de integración	Integrar continuamente al concluir las tareas de una historia de usuario Verificar que las pruebas de integración pasan al 100%	Programador
Ejecutar pruebas de aceptación	Reporte de pruebas de aceptación	Correr la última versión de una iteración Utilizar los casos de prueba de aceptación	Cliente Encargado de pruebas

Fuente: (Porras, 2010)

CAPITULO IV

ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

4.1. RESULTADOS

4.1.1. ARTEFACTOS DEL SOFTWARE APLICANDO EL PROCESO XP

A. FASE DE EXPLORACIÓN

Según el procedimiento desarrollado en las tablas 3.2 a 3.4 para el proceso XP, descrito en el capítulo 3, fase de exploración. Obtenemos las historias de usuario, arquitectura técnica inicial y el plan de alto nivel.

Tabla N° 4.1: Historias de usuario.

N°	HISTORIA DE USUARIO	DESCRIPCIÓN
01	Registrar usuario	El usuario (administrador) del instituto, podrá registrarse en el sistema.
02	Iniciar sesión	El usuario (administrador) del instituto, podrá iniciar sesión.
03	Cerrar sesión.	El usuario (administrador) del instituto, podrá cerrar sesión.
04	Actualizar perfil del usuario.	El usuario (administrador) del instituto, podrá actualizar su perfil.
05	Registrar Carrera Profesional	El usuario (administrador) del instituto, podrá registrar la carrera profesional.
06	Registrar Estudiante	El usuario (administrador) del instituto, podrá registrar estudiante.
07	Registrar docente.	El usuario (administrador) del Instituto, podrá registrar toda su información del Docente.
08	Registrar las unidades didácticas.	El usuario (Administrador) del instituto, podrá registrar las unidades didácticas.
09	Registrar semestre académico	El usuario (administrador) del instituto, podrá registrar el semestre académico.
10	Registrar el inicio del ciclo académico	El usuario (administrador) del instituto, podrá registrar el inicio del ciclo académico.

N°	HISTORIA DE USUARIO	DESCRIPCIÓN
11	Asignar unidad didáctica y docente por ciclo académico	El usuario (administrador) del instituto, podrá asignar unidad didáctica y docente por ciclo académico.
12	Matricular estudiante.	El usuario (administrador) del Instituto, podrá matricularse, registrando toda su información.
13	Registrar notas por unidad didáctica y por estudiante.	El usuario (administrador) del Instituto podrá registrar la calificación por unidad didáctica y estudiante.
14	Generar reporte de aprobados y desaprobados por semestre.	El usuario (administrador) del Instituto podrá generar el reporte de aprobados y desaprobados por semestre académico.
15	Generar reportes de notas por estudiante	El usuario (administrador) del instituto, podrá generar los reportes por estudiantes.
16	Generar ranking de alumnos por semestre académico	El usuario (administrador) del Instituto podrá generar el ranking de rendimiento de los alumnos por semestre académico
17	Generar los horarios semestrales	El usuario (administrador) del instituto, podrá registrar los horarios para cada carrera profesional.

Fuente: Elaboración Propia



Figura N° 4.1: Arquitectura técnica inicial

Tabla N° 4.2: Plan de alto nivel.

N°	HISTORIAS DE USUARIO	ESFUERZO (días)
1	Registrar usuario	2
2	Iniciar sesión	2
3	Cerrar sesión	2
4	Actualizar perfil del usuario.	2
5	Registrar carreras profesionales	3
6	Registrar estudiante	4
7	Registrar docente.	3
8	Registrar unidades didácticas	2
9	Registrar semestre académico	4

Nº	HISTORIAS DE USUARIO	ESFUERZO (días)
10	Registrar el inicio del ciclo académico.	4
11	Asignar unidad didáctica y docente por ciclo académico.	3
12	Matricular estudiante	5
13	Registrar notas por unidad didáctica y por estudiante	3
14	Generar reporte de aprobados y desaprobados por semestre.	2
15	Generar reportes de notas por estudiante	3
16	Generar ranking de alumnos por semestre académico	3
17	Generar los horarios semestrales	2

Fuente: Elaboración propia

B. FASE DE PLANIFICACIÓN

Las historias de usuario son un conjunto de fichas escritas que indican las funciones que debe realizar el sistema, constituyendo el mecanismo base de captura de requerimientos de la programación extrema.

Cada historia de usuario incluye una breve descripción, es importante procurar no incluir sintaxis técnica, de modo que se centren en las necesidades y no en la especificación del aspecto de las interfaces de usuario ni en la implementación, como base de datos o algoritmos específicos.

Típicamente las historias de usuario deben ser escritas en tarjetas, sin embargo, este documento provee una plantilla para que la utilice en la escritura de ellas.

Tabla N° 4.3: Registrar usuario.

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador del instituto.
Nombre historia: Registrar usuario	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Descripción: Como usuario tengo que registrarme en el sistema ingresando sus datos, con la finalidad de administrar la aplicación de acuerdo al rol que se le asigna.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.4: Iniciar Sesión.

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador del Instituto
Nombre historia: Iniciar sesión	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Descripción: Como usuario administrador, debo ingresar al sistema mediante el uso de cuenta de usuario y contraseña; con la finalidad de poner de un control adecuado de las operaciones que se realicen dentro del sistema.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.5: Cerrar sesión.

HISTORIA DE USUARIO	
Número: 3	Usuario: Administrador del Instituto
Nombre historia: Cerrar sesión.	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Descripción: Como administrador deseo realizar el proceso de salir del sistema, luego de haber interactuado con el sistema.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.6: Actualizar perfil del usuario.

HISTORIA DE USUARIO	
Número: 4	Usuario: Administrador del Instituto.
Nombre historia: Actualizar perfil del usuario.	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 2
<p>Descripción: Como administrador, puede aprobar o rechazar las solicitudes de creación de cuentas de usuario para acceder al sistema. A si mismo pueden dar de baja a los usuarios que incurran en faltas que deriven en la alteración de resultados del proceso académico. Esto con la finalidad de poder tener un control adecuado y que las opiniones sean las más fiable posibles.</p>	
Observaciones: Ninguno	

Fuente: Elaboración Propia

Tabla N° 4.7: Registrar Carrera Profesional.

HISTORIA DE USUARIO	
Número: 5	Usuario: Administrador del instituto.
Nombre historia: Registrar carreras profesionales.	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
<p>Descripción: Como administrador debo registrar los datos relacionados a las carreras profesionales con que cuenta el instituto, con la finalidad de utilizar la información, al momento de realizar la gestión académica,</p>	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.8: Registrar estudiante.

HISTORIA DE USUARIO	
Número: 6	Usuario: Administrador del Instituto.
Nombre historia: Registrar estudiante.	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Descripción: Como administrador deseo registrar a los estudiantes del instituto, con la finalidad de tener los datos de éstos registrados en el sistema para realizar la gestión académica.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.9: Registrar docente.

HISTORIA DE USUARIO	
Número: 7	Usuario: Administrador del Instituto.
Nombre historia: Registrar Docente.	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Descripción: Como administrador deseo registrar a los docentes y personal administrativo del instituto, con la finalidad de tener los datos de éstos registrados en el sistema para realizar la gestión académica.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.10: Registrar unidades didácticas.

HISTORIA DE USUARIO	
Número: 08	Usuario: Administrador del Instituto.
Nombre historia: Registrar Unidades Didácticas	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Descripción: Como administrar deseo registrar las unidades didácticas de las carreras profesionales de: producción agropecuaria, enfermería técnica y computación e informática, de acuerdo al itinerario formativo de cada carrera profesional.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.11: Registrar semestres académicos

HISTORIA DE USUARIO	
Número: 09	Usuario: Administrador del Instituto.
Nombre historia: Registrar Semestre Académico	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Descripción: Como administrador quiero registrar los semestres académicos de las carreras profesionales: producción agropecuaria, enfermería técnica y computación e informática, con la finalidad de matricular a los estudiantes.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.12: Registrar el inicio de semestre académico

HISTORIA DE USUARIO	
Número: 10	Usuario: Administrador del Instituto.
Nombre historia: Registrar el inicio de Semestre Académico	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Descripción: Como administrador deseo registrar la apertura del semestre académico, correspondiente a cada carrera profesional considerando las fechas que corresponde al cronograma de actividades académicas del semestre.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.13: Asignar unidad didáctica y docente por ciclo académico

HISTORIA DE USUARIO	
Número: 11	Usuario: Administrador del Instituto.
Nombre historia: Asignar unidad didáctica y docente por ciclo académico	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Descripción: Como administrador quiero asignar a cada unidad didáctica al personal docente por ciclo académico, de acuerdo a la distribución de la carga académica y la formación profesional de los docentes.	
Observaciones: Cada docente debe tener como máximo 22 horas pedagógicas semanales.	

Fuente: Elaboración Propia

Tabla N° 4.14: Registrar matrícula del estudiante.

HISTORIA DE USUARIO	
Número: 12	Usuario: Administrador del Instituto.
Nombre historia: Matricular estudiantes.	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 4	Iteración asignada: 2
Descripción: Como administrador debo realizar la matrícula registrando ingresando todos los datos, para considerar en la nómina oficial de matrícula, lo cual ser enviado a Dirección Regional de Educación de Huancavelica.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.15: Registrar notas por unidad didáctica y por estudiante

HISTORIA DE USUARIO	
Número: 13	Usuario: Administrador del Instituto.
Nombre historia: Registrar notas por unidad didáctica y por estudiante	
Prioridad en negocio: Alto	Riesgo en desarrollo: Medio
Puntos estimados: 4	Iteración asignada: 2
Descripción: Como administrador del instituto deseo registrar las notas por unidad didáctica y por estudiante, correspondientes a cada carrera profesional: producción agropecuaria, enfermería técnica y computación e informática, con la finalidad de generar el registro de evaluaciones.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.16: Generar reporte de aprobados y desaprobados por semestre.

HISTORIA DE USUARIO	
Número: 14	Usuario: Administrador del Instituto.
Nombre historia: Generar reporte de aprobados y desaprobados por semestre.	
Prioridad en negocio: Alto	Riesgo en desarrollo: Medio
Puntos estimados: 4	Iteración asignada: 2
Descripción: Como usuario administrador, quiero generar el reporte de aprobados y desaprobados de estudiantes por cada semestre académico de las carreras profesionales: producción agropecuaria, enfermería técnica y computación e informática, considerando las unidades didácticas matriculados.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.17: Generar reportes de notas por estudiante

HISTORIA DE USUARIO	
Número: 15	Usuario: Administrador del Instituto.
Nombre historia: Generar reportes de notas por estudiante	
Prioridad en negocio: Alto	Riesgo en desarrollo: Medio
Puntos estimados: 4	Iteración asignada: 2
Descripción: Como usuario administrador del Instituto, debo generar el reporte de notas por cada estudiante de cada carrera profesional y semestre correspondiente, que van ser de utilidad para generar boleta de notas de cada estudiante.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.18: Generar ranking de alumnos por semestre académico

HISTORIA DE USUARIO	
Número: 16	Usuario: Administrador del Instituto.
Nombre historia: Generar ranking de alumnos por semestre académico	
Prioridad en negocio: Alto	Riesgo en desarrollo: Medio
Puntos estimados: 4	Iteración asignada: 2
Descripción: Como usuario administrador del Instituto, quiero generar el ranking de alumnos por semestre académico y carrera profesional de acuerdo al rendimiento académico, para realizar reconocimiento mediante resolución directoral a los primeros y segundos puestos de cada carrera profesional y semestre académico.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.19: Generar Horarios.

HISTORIA DE USUARIO	
Número: 17	Usuario: Administrador del Instituto.
Nombre historia: Generar Horarios Semestrales	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Descripción: Como administrador del Instituto, debo de generar los horarios semestrales de acuerdo a las unidades didácticas de las carreras profesionales, asignación de unidades didácticas y distribución de carga académica, para la organización mental de los estudiantes.	
Observaciones: Ninguno	

Fuente: Elaboración propia

Tabla N° 4.20: Tabla de Versión.

N°	HISTORIA DE USUARIO	PRIORIDAD	RIESGO	ESFUERZO	ITERACIÓN
				(DÍAS)	
1	Registrar usuario	Medio	Medio	2	2
2	Iniciar sesión	Medio	Medio	2	2
3	Cerrar sesión.	Medio	Medio	2	2
4	Actualizar perfil del usuario.	Medio	Medio	2	2
5	Registrar carreras profesionales	Alto	Alto	3	2
6	Registrar estudiante.	Alto	Alto	4	2
7	Registrar docente.	Alto	Alto	3	2
8	Registrar unidades didácticas	Alto	Medio	2	2
9	Registrar semestre académico.	Alto	Medio	4	2
10	Registrar el inicio de ciclo académico.	Alto	Alto	4	2
11	Asignar unidad didáctica y docente por ciclo académico.	Medio	Alto	3	2
12	Matricular estudiante	Alto	Alto	5	2
13	Registrar notas por unidad didáctica y por estudiante	Alto	Alto	3	2
14	Generar reporte de aprobados y desaprobados por semestre.	medio	medio	2	2
15	Generar reportes de notas por estudiante	Alto	Alto	3	2
16	Generar ranking de alumnos por semestre académico	Alto	Alto	3	2
17	Generar los horarios semestrales	Alto	Alto	2	2

Fuente: Elaboración propia

C. FASE DE ITERACIÓN

La fase de iteración presentado en el título 2.2.3, capítulo II y, la técnica referenciada en la tabla 3.4, permite obtener los entregables; arquitectura técnica, tareas de ingeniería, plan de iteración, casos de prueba de aceptación, GUI, tarjetas CRC, base

de datos física, código fuente para clases entidad, pruebas unitarias, código fuente para tarea de ingeniería, reporte de pruebas unitarias, reporte de pruebas de integración y de aceptación.

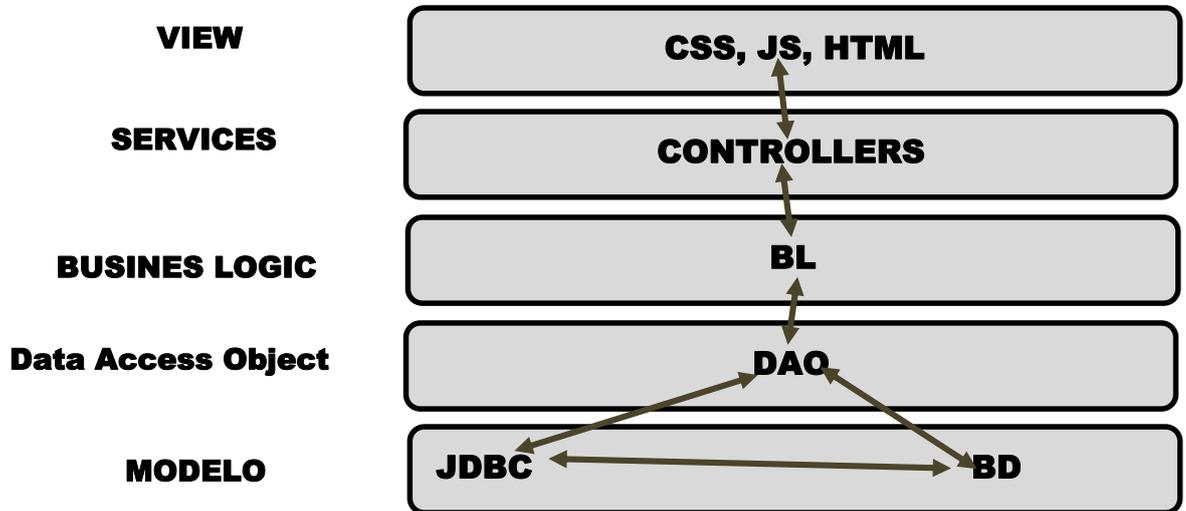


Figura N° 4.2: Arquitectura técnica final. Diagrama de componentes

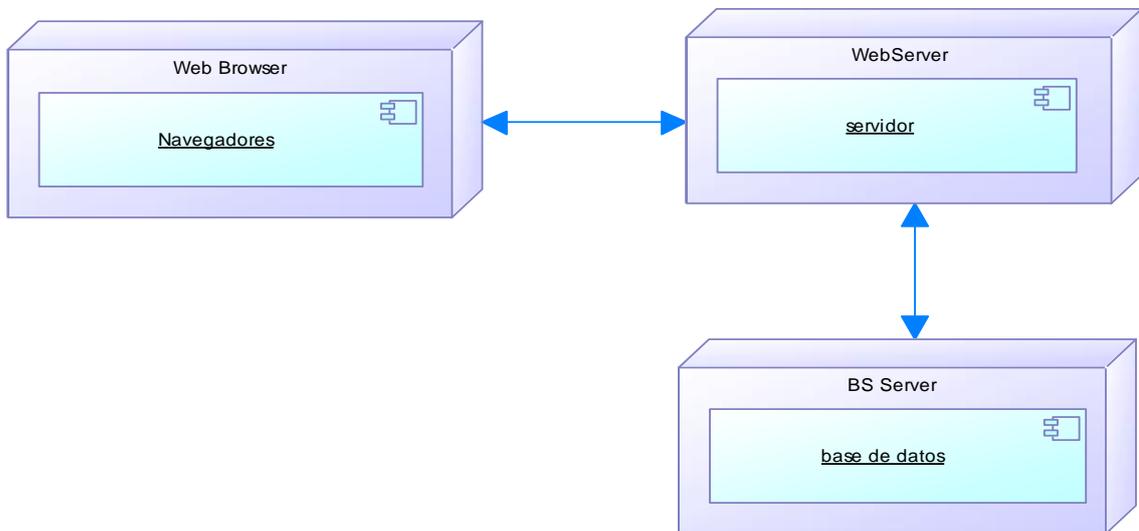


Figura N° 4.3: Arquitectura técnica final. Diagrama de despliegue.

Tabla N° 4.21: Registrar usuario.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Número historia de usuario: 1
Nombre tarea: Registrar usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 01/08/2016	Fecha fin: 03/08/2016
Programador responsable: Néstor Conde Jaules	
<p>Descripción: El usuario en la página “Menú Principal” selecciona el botón “Registrar”, el sistema le muestra el formulario de registro; el usuario ingresa nombres, apellidos, DNI, dirección, teléfono, usuario y password y luego hace clic en el botón “guardar” el sistema verifica la validación del campo y una vez validada guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.22: Iniciar sesión con usuario del sistema.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 2	Número historia de usuario: 2
Nombre tarea: Iniciar sesión.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 04/08/2016	Fecha fin: 06/08/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario en la página “Menú Principal” selecciona el botón “Iniciar sesión”, el sistema verifica la validación del campo y una vez validada muestra el formulario de “Registro del usuario”.</p>	

Fuente: Elaboración propia

Tabla N° 4.23: Cerrar sesión.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 3	Número historia de usuario: 3
Nombre tarea: Cerrar sesión.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 07/08/2016	Fecha fin: 09/08/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario en la página “Usuario” selecciona el botón “Cerrar sesión”, el sistema verifica la validación del campo y una vez validada muestra el formulario de “Iniciar sesión”.</p>	

Fuente: Elaboración Propia

Tabla N° 4.24: Actualizar perfil del usuario.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 4	Número historia de usuario: 4
Nombre tarea: Actualizar perfil del usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 10/08/2016	Fecha fin: 12/08/2016
Programador responsable: Néstor Conde Jaules	
<p>Descripción: El usuario en la página “Usuario” selecciona “Actualizar perfil usuario”, el sistema le muestra el formulario “perfil de usuario” luego ingresa los datos a actualizar y hace clic en el botón “guardar”, el sistema verifica la validación del campo ingresado y una vez validada el sistema guarda los datos ingresados en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.25: Registrar Carrera Profesional

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 5	Número historia de usuario: 5
Nombre tarea: Registrar carreras profesionales	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 13/08/2016	Fecha fin: 17/08/2016
Programador responsable: Néstor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Carrera" y luego selecciona el botón "nuevo" seguidamente ingresa código, nombre, Año de creación y descripción; una vez ingresado los datos hace clic en el botón "guardar" el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.26: Actualizar Carrera Profesional

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 6	Número historia de usuario: 5
Nombre tarea: Actualizar Carreras Profesionales	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 13/08/2016	Fecha fin: 16/08/2016
Programador responsable: Néstor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Carrera" y luego el icono "Actualizar " y seguidamente modifica: código, nombre, Año de creación y descripción; una vez ingresado los datos hace clic en el botón "guardar" el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración Propia

Tabla N° 4.27: Registrar estudiante.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 7	Número historia de usuario: 6
Nombre tarea: Registrar Estudiante	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 18/08/2016	Fecha fin: 22/08/2016
Programador responsable: Néstor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Estudiante" y luego selecciona el botón "nuevo" seguidamente ingresa códigos nombres, apellidos, DNI, número de celular, dirección, correo electrónico, carrera profesional y semestre; una vez ingresado los datos hace clic en el botón "guardar" el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.28: Actualizar estudiante.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 8	Número historia de usuario: 6
Nombre tarea: Actualizar estudiante.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 18/08/2016	Fecha fin: 22/08/2016
Programador responsable: Néstor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Estudiante" luego selecciona el icono "editar" del "menú" y luego corrige la información ingresada nombres, apellidos, DNI, número de celular, dirección, correo electrónico, carrera profesional, y semestre; una vez corregido hace clic en el botón "actualizar" en la página "Estudiante", el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.29: Eliminar estudiante.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 9	Número historia de usuario: 6
Nombre tarea: Eliminar estudiante.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 18/08/2016	Fecha fin: 22/08/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Estudiante" luego selecciona el icono "eliminar", el sistema muestra una ventana "desea eliminar el registro" luego el usuario hace clic en el botón "ok" para eliminar en caso contrario hace clic en "cancelar".</p>	

Fuente: Elaboración propia

Tabla N° 4.30: Registrar docente.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 10	Número historia de usuario: 7
Nombre tarea: Registrar docente	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 23/08/2016	Fecha fin: 27/08/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Docente" y luego selecciona el botón "nuevo" seguidamente ingresa grado, nombres, apellidos, DNI, número de celular, dirección y correo electrónico; una vez ingresado los datos hace clic en el botón "guardar" el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.31: Actualizar docente.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 11	Número historia de usuario: 7
Nombre tarea: Actualizar docente.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 23/08/2016	Fecha fin: 27/08/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Docente" luego selecciona el icono "editar" y luego corrige la información ingresada nombres, apellidos, número de celular, dirección, correo electrónico; una vez corregido hace clic en el botón "actualizar" en la página "Registro", el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración Propia

Tabla N° 4.32: Eliminar docente.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 12	Número historia de usuario: 7
Nombre tarea: Eliminar docente.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 23/08/2016	Fecha fin: 27/08/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Docente" luego selecciona en el "eliminar", el sistema muestra una página "desea eliminar el registro" luego el usuario hace clic en el botón "Eliminar" para eliminar en caso contrario hace clic en "cancelar".</p>	

Fuente: Elaboración propia

Tabla N° 4.33: Registrar Unidades didácticas.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 13	Número historia de usuario: 08
Nombre tarea: Registrar unidades didácticas.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 28/08/2016	Fecha fin: 30/08/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Docente" y luego selecciona el botón "nuevo" seguidamente y luego ingresa código, nombre, créditos y semestre y luego selecciona el botón "Guardar", el sistema verifica la validación de campos y lo guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.34: Actualizar unidades didácticas.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 14	Número historia de usuario: 08
Nombre tarea: Actualizar unidades didácticas.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 28/08/2016	Fecha fin: 30/08/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Unidades Didácticas" luego selecciona el icono "editar" y luego corrige la información ingresada de las unidades didácticas; una vez corregido hace clic en el botón "actualizar", el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración Propia

Tabla N° 4.35: Registrar semestre académico.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 15	Número historia de usuario: 09
Nombre tarea: Registrar semestre académico	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 03/09/2016	Fecha fin: 07/09/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Semestre" y luego selecciona el botón "Nuevo" seguidamente ingresa código, nombre del semestre, año y fecha; y hace clic en el botón "guardar" el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración Propia

Tabla N° 4.36: Registrar el inicio del ciclo académico.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 16	Número historia de usuario: 10
Nombre tarea: Registrar el inicio del ciclo académico	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 08/09/2016	Fecha fin: 11/09/2016
Programador responsable: Néstor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Apertura del ciclo académico" y luego selecciona el botón "nuevo" seguidamente ingresa denominación, fecha inicio, fecha final, fecha examen parcial, fecha examen final, fecha examen recuperación, fecha de entrega de actas y descripción; y hace clic en el botón "guardar" el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.37: Asignar unidad didáctica y docente por ciclo académico

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 17	Número historia de usuario: 11
Nombre tarea: Asignar unidad didáctica y docente por ciclo académico	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 12/09/2016	Fecha fin: 16/09/2016
Programador responsable: Néstor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Asignar docente curso" y luego selecciona el botón "Nuevo" seguidamente selecciona el docente, unidad didáctica y semestre académico; y hace clic en el botón "guardar" el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración Propia

Tabla N° 4.37: Matricular Estudiante

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 18	Número historia de usuario: 12
Nombre tarea: Matricular estudiante	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 17/09/2016	Fecha fin: 21/09/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario selecciona la opción "Matricula" y luego selecciona el botón "nuevo" seguidamente ingresa país, región, ciudad, código, nombres, apellidos, fecha de nacimiento, dirección, semestre, carrera profesional, correo y móvil; y hace clic en el botón "guardar" el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.38: Actualizar Matricula de Estudiante

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 19	Número historia de usuario: 12
Nombre tarea: Actualizar matrícula de estudiante.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 17/09/2016	Fecha fin: 21/09/2016
Programador responsable: Néstor Conde Jaules	
<p>Descripción: El usuario selecciona la opción “Matricula” luego selecciona el ícono “editar” y luego corrige la información ingresada de la matrícula del estudiante; una vez corregido hace clic en el botón “actualizar”, el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.39: Registrar calificación de los estudiantes.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 20	Número historia de usuario: 13
Nombre tarea: Registrar notas por unidad didáctica y por estudiante.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 22/09/2016	Fecha fin: 22/09/2016
Programador responsable: Nestor Conde Jaules	
<p>Descripción: El usuario selecciona la opción “Evaluaciones” luego selecciona el botón “Nuevo” y luego ingresa carrera profesional, docente, ciclo académico, unidad didáctica y calificación y hace clic en el botón “guardar” el sistema valida los datos ingresados y los guarda en la base de datos.</p>	

Fuente: Elaboración propia

Tabla N° 4.40: Generar reporte de aprobados y desaprobados.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 21	Número historia de usuario: 14
Nombre tarea: Generar reporte de aprobados y desaprobados por semestre.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 27/09/2016	Fecha fin: 30/09/2016
Programador responsable: Nestor Conde Jaules	
Descripción: El usuario selecciona la opción “generar reportes” , luego selecciona el botón “reporte” y luego ingresa semestre académico; y luego hace clic en el botón “visualizar” en la página “Reportes” , el sistema valida los datos ingresados y visualiza los datos.	

Fuente: Elaboración Propia

Tabla N° 4.41: Generar reporte de notas por estudiante.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 22	Número historia de usuario: 15
Nombre tarea: Generar Reporte de notas por estudiante	
Tipo de tarea : Desarrollo	Puntos estimados: 4
Fecha inicio: 01/10/2016	Fecha fin: 04/10/2016
Programador responsable: Néstor Conde Jaules	
Descripción: El usuario selecciona la opción “Indicadores” , luego selecciona el botón “reporte” e ingresa semestre académico, nombre del docente, unidad didáctica y luego hace clic opción “generar boleta de nota” y el sistema muestra visualiza boleta de nota del estudiante.	

Fuente: Elaboración Propia

Tabla N° 4.42: Generar ranking de alumnos por semestre académico.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 23	Número historia de usuario: 16
Nombre tarea: Generar ranking de alumnos por semestre académico.	
Tipo de tarea : Desarrollo	Puntos estimados: 4
Fecha inicio: 05/10/2016	Fecha fin: 10/10/2016
Programador responsable: Néstor Conde Jaules	
Descripción: El usuario selecciona la opción " Indicadores ", luego selecciona el botón " reporte " e ingresa carrera profesional, semestre académico y luego hace clic opción " generar ranking " y el sistema muestra visualiza el ranking de estudiantes de acuerdo al rendimiento académico.	

Fuente: Elaboración propia

Tabla N° 4.43: Generar horarios semestrales.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 24	Número historia de usuario: 17
Nombre tarea: Generar Horarios Semestrales	
Tipo de tarea : Desarrollo	Puntos estimados: 4
Fecha inicio: 11/10/2016	Fecha fin: 14/10/2016
Programador responsable: Néstor Conde Jaules	
Descripción: El usuario selecciona la opción " Horarios ", luego selección el botón " Nuevo " ingresa carrera profesional, semestre docente y nombre de la unidad didáctica y selecciona la opción " Generar ", el sistema muestra el formulario en la página " Horarios ", y luego el sistema genera el horario semestral.	

Fuente: Elaboración propia

Tabla N° 4.44: Plan de iteración

N° T.I.	TAREAS DE INGENIERIA	HISTORIAS DE USUARIO	N° H.U.
1	Registrar usuario	Registrar usuario	1
2	Iniciar sesión	Iniciar sesión	2
3	Cerrar sesión	Cerrar sesión	3
4	Actualizar perfil del usuario	Actualizar perfil del usuario	4
5	Registrar carreras profesionales	Registrar carreras profesionales	5
6	Actualizar Carreras Profesionales		
7	Registrar Estudiante	Registrar Estudiante	6
8	Actualizar Estudiante		
9	Eliminar estudiante		
10	Registrar Docente	Registrar Docente	7
11	Actualizar Docente		
12	Eliminar estudiante		
13	Registrar unidades didácticas.	Registrar unidades didácticas.	8
14	Actualizar unidades didácticas.		
15	Registrar semestre académico	Registrar semestre académico	9
16	Registrar el inicio del ciclo académico	Registrar el inicio del ciclo académico	10
17	Asignar unidad didáctica y docente por ciclo académico	Asignar unidad didáctica y docente por ciclo académico	11
18	Matricular estudiante	Matricular estudiante	12
19	Actualizar matricula de estudiante.		

N° T.I.	TAREAS DE INGENIERIA	HISTORIAS DE USUARIO	N° H.U.
20	Registrar notas por unidad didáctica y por estudiante.	Registrar notas por unidad didáctica y por estudiante.	13
21	Generar reporte de aprobados y desaprobados por semestre.	Generar reporte de aprobados y desaprobados por semestre.	14
22	Generar Reporte de notas por estudiante	Generar Reporte de notas por estudiante	15
23	Generar ranking de alumnos por semestre académico.	Generar ranking de alumnos por semestre académico.	16
24	Generar Horarios Semestrales	Generar Horarios Semestrales	17

Fuente: Elaboración propia

Tabla N° 4.45: Plan de iteración clasificados en fechas de desarrollo.

N° T.I.	N° H.U.	Fecha Inicio	Fecha Fin	Programador
1	1	01/08/2016	03/08/2016	Nestor Conde Jaules
2	2	04/08/2016	06/08/2016	Nestor Conde Jaules
3	3	07/08/2016	09/08/2016	Nestor Conde Jaules
4	4	10/08/2016	12/08/2016	Nestor Conde Jaules
5	5	13/08/2016	17/08/2016	Nestor Conde Jaules
6	5	13/08/2016	17/08/2016	Nestor Conde Jaules
7	6	18/08/2016	22/08/2016	Nestor Conde Jaules
8	6	18/08/2016	22/08/2016	Nestor Conde Jaules
9	6	18/08/2016	22/08/2016	Nestor Conde Jaules
10	7	23/08/2016	27/08/2016	Nestor Conde Jaules
11	7	23/08/2016	27/08/2016	Nestor Conde Jaules
12	7	23/08/2016	27/08/2016	Nestor Conde Jaules
13	8	28/08/2016	30/08/2016	Nestor Conde Jaules
14	8	28/08/2016	30/08/2016	Nestor Conde Jaules

Nº T.I.	Nº H.U.	Fecha Inicio	Fecha Fin	Programador
15	9	03/09/2016	07/09/2016	Nestor Conde Jaules
16	10	08/09/2016	11/09/2016	Nestor Conde Jaules
17	11	12/09/2016	16/09/2016	Nestor Conde Jaules
18	12	17/09/2016	21/09/2016	Nestor Conde Jaules
19	12	17/09/2016	21/09/2016	Nestor Conde Jaules
20	13	22/09/2016	26/09/2016	Nestor Conde Jaules
21	14	27/09/2016	30/09/2016	Nestor Conde Jaules
22	15	01/10/2016	04/10/2016	Nestor Conde Jaules
23	16	05/10/2016	10/10/2016	Nestor Conde Jaules
24	17	11/10/2016	14/10/2016	Nestor Conde Jaules

Fuente: Elaboración Propia

CASOS DE PRUEBA DE ACEPTACIÓN

Permite probar, que al poner una historia del usuario en ejecución se haya puesto correctamente. Una historia de usuario puede tener una o muchas pruebas de aceptación, lo que asegura a los trabajos su correcta funcionalidad. Una historia de usuario no se considera completa hasta que haya pasado sus pruebas de aceptación. Esto significa que las nuevas pruebas de aceptación se deben crear para cada iteración.

Tabla Nº 4.46: Casos de prueba de aceptación.

Nº H.U.	REQUISITOS	Nº C.P.	CASOS DE PRUEBA (C.P.)
1	El sistema debe ser capaz de registrar un usuario	1	El sistema debe verificar que los datos ingresados sean correctos y permitir crear la cuenta del usuario.
2	El sistema debe ser capaz de iniciar sesión	2	El sistema debe verificar que el login y la contraseña sean correctas y permitir el acceso al usuario.
	El sistema debe ser capaz de crear una nueva cuenta	3	El sistema debe verificar si el usuario entra por primera vez o ya inicio sesión, el usuario puede también ingresar con su cuenta del sistema.

Nº H.U.	REQUISITOS	Nº C.P.	CASOS DE PRUEBA (C.P.)
6	El sistema debe ser capaz matricular al estudiante	4	El sistema debe verificar que los datos ingresados sean correctos para guardar y mostrar.
	El sistema debe ser capaz de registrar estudiante.	5	El sistema debe verificar los datos ingresados y guardar.
4	El sistema debe ser capaz de registrar unidades didácticas.	6	El sistema debe verificar que los datos ingresados por el usuario estén correctos, en caso contrario debe mostrar los campos mal ingresados.
	El sistema debe ser capaz de registrar unidades didácticas.	7	El sistema una vez verificado debe ser capaz de guardar los datos ingresados.

Fuente: Elaboración propia

TARJETAS CLASE RESPONSABILIDAD Y COLABORACIÓN (CRC)

Para un diseño simple, se procede a definir tarjetas CRC, que permitan identificar las clases, sus responsabilidades y las colaboraciones que realiza para una historia de usuario y las tareas correspondientes, según se desarrolla en el capítulo II, sección 2.2.5.

Tabla Nº 4.47: CRC 001.

TARJETA CRC		
Número: 001	Escenario: Registrar usuario	
Nombre Clase: Usuario		
Responsabilidades	Colaboradores	Atributos
Registrar un usuario.	Usuario Docentes Estudiante	idusuario Nombre ApePaterno ApeMaterno

Fuente: Elaboración Propia

Tabla N° 4.48: CRC 002.

TARJETA CRC		
Número: 002	Escenario: Iniciar sesión	
Nombre Clase: Usuario		
Responsabilidades	Colaboradores	Atributos
Iniciar sesión con Facebook y con el sistema.	Usuario Docente. Estudiante.	Idusuario IdDocente IdEstudiante Nombre Apellidos

Fuente: Elaboración Propia

Tabla N° 4.49: CRC 003.

TARJETA CRC		
Número: 003	Escenario: Actualizar perfil del usuario	
Nombre Clase: Usuario		
Responsabilidades	Colaboradores	Atributos
Actualizar perfil del usuario.	Usuario Docente Estudiante	Idusuario IdDocente IdEstudiante Nombre Apellidos

Fuente: Elaboración propia

Tabla N° 4.50: CRC 004.

TARJETA CRC		
Número: 004	Escenario: Registrar Docente	
Nombre Clase: docente		
Responsabilidades	Colaboradores	Atributos
Registrar Docente	Usuario Docente	Idusuario Iddocente Nombre Apellidos Experiencia Conocimiento Competencia personal Formación académica

Fuente. Elaboración propia

Tabla N° 4.51: CRC 005.

TARJETA CRC		
Número: 005	Escenario: Registrar Estudiante	
Nombre Clase: Estudiante		
Responsabilidades	Colaboradores	Atributos
Registrar Estudiante	Usuario Estudiante	Idusuario IdoEstudiante Nombre Apellidos Dirección Semestre Correo electrónico

Fuente: Elaboración propia

Tabla N° 4.52: CRC 006.

TARJETA CRC		
Número: 006	Escenario: Buscar Estudiante	
Nombre Clase: Estudiante		
Responsabilidades	Colaboradores	Atributos
Buscar estudiante	Usuario Docente. Estudiante.	Idusuario Idestudiante Iddocente Nombre Dirección Correo electrónico

Fuente: Elaboración propia

Tabla N° 4.53: CRC 007.

TARJETA CRC		
Número: 007	Escenario: Buscar Docente	
Nombre Clase: Docente		
Responsabilidades	Colaboradores	Atributos
Buscar Docente	Usuario Docente Estudiante	Idusuario Iddocente Idestudiante Nombre Dirección Correo electrónico

Fuente: Elaboración Propia

Tabla N° 4.53: CRC 008.

TARJETA CRC		
Número: 008	Escenario: Registrar Calificaciones	
Nombre Clase: Usuario		
Responsabilidades	Colaboradores	Atributos
Registrar y actualizar la calificación de los estudiantes..	administrador Docente	Idusuario Iddocente Nombre Dirección Correo electrónico

Fuente: Elaboración propia

BASE DE DATOS FÍSICA

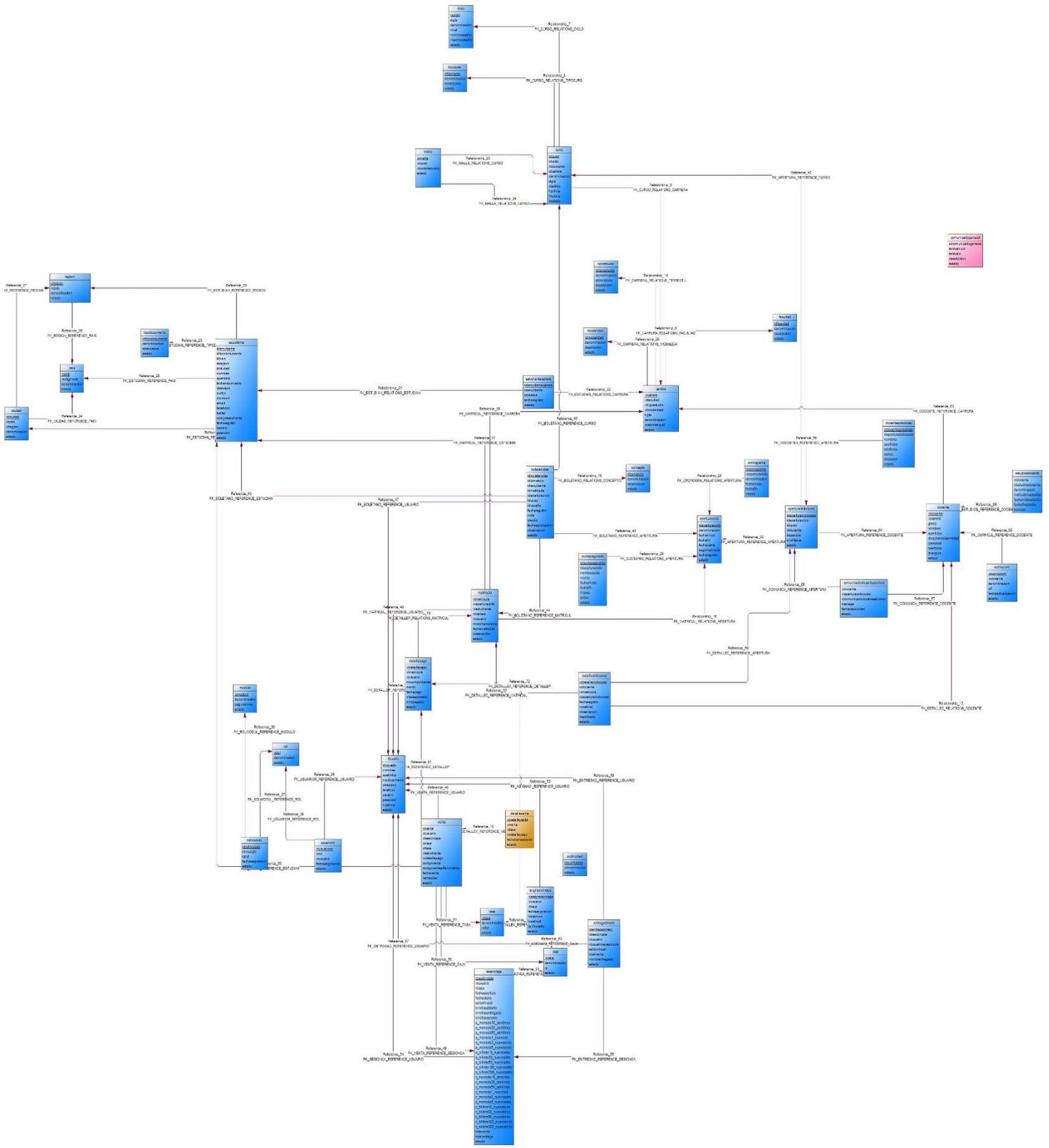


Figura N° 4.4: Base de Datos Física
Fuente: Elaboración propia

INTERFAZ DE USUARIO Y CODIFICACIÓN

Para el diseño de la interfaces gráficas de usuario de la aplicación web se consideró estándares de usabilidad, teniendo en cuenta que el objetivo de la tesis sí considera su evaluación; los factores para los estándares utilizados: botones, títulos de páginas, menús, fuentes, tamaño de fuente, colores; que permitan tener interfaces amigables y fáciles de usar.

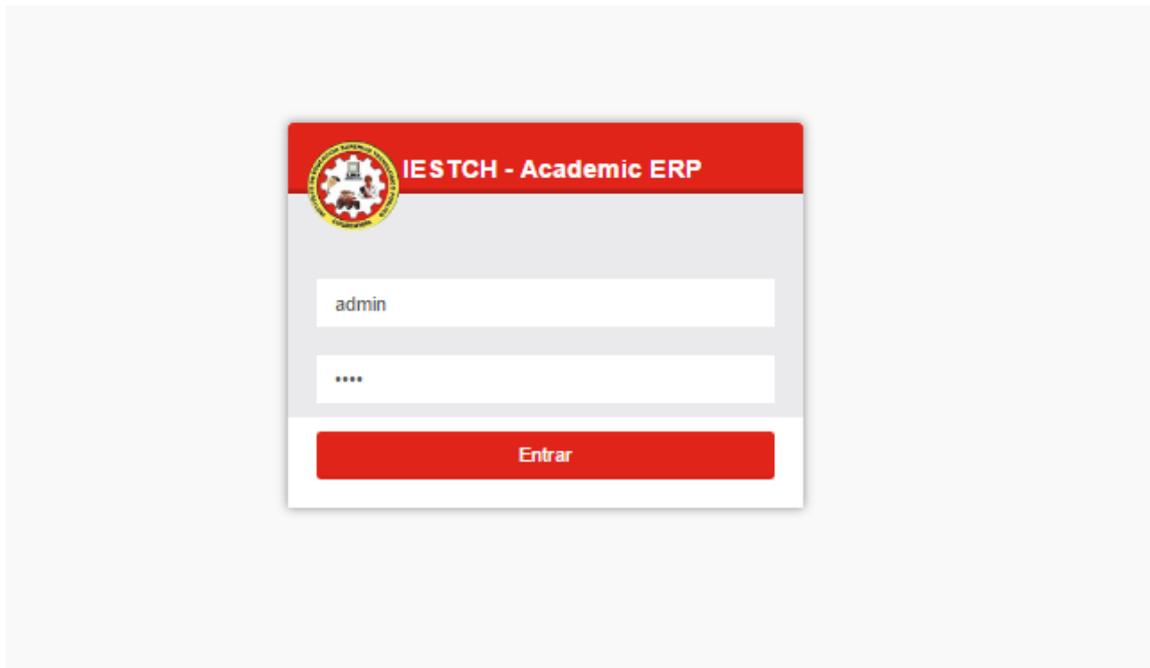


Figura N° 4.5: Página de Ingreso. Acceso a la aplicación web
Fuente: Elaboración propia

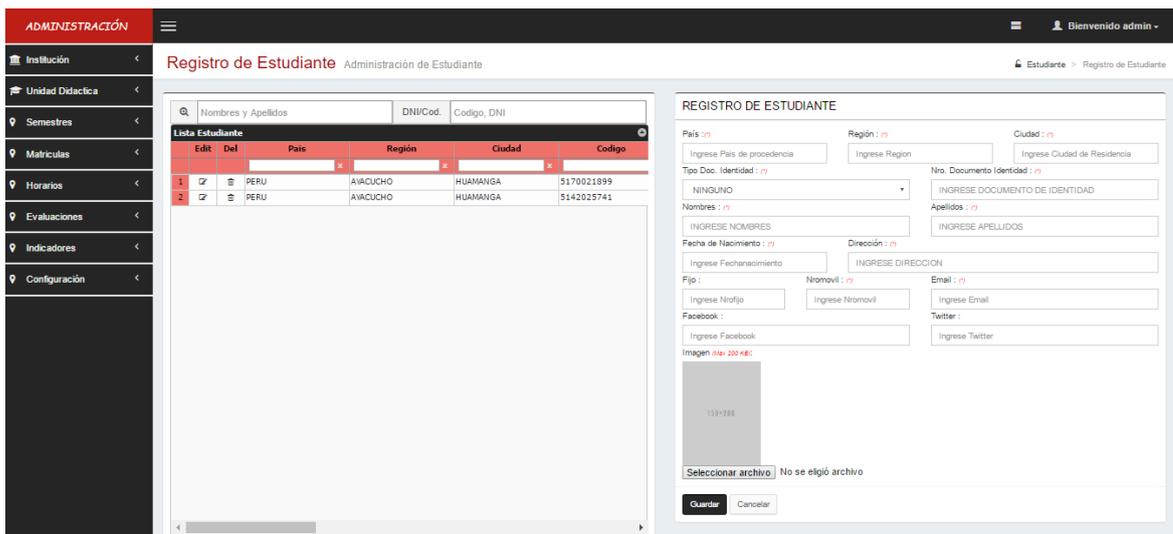


Figura N° 4.6: Página de Ingreso. Aplicación web para la Gestión Académica del IESTP
Churcampa
Fuente: Elaboración propia

Lista de usuarios

Lista Usuario						
	Editar	Eliminar	Nombres	Apellidos	DNI	Dirección
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	admin		admin	12345678
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DENIS JACK	OCHOA BERROCAL	43724871	SIN DIRECCION
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	GENARO	PALOMINO	28213485	LAS AMERICAS
4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DEBBIE	VILA MARTINEZ	42300841	URB. BANCO DI
5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NORMA	VILCATOMA JAYO	42960180	JRON MANCO C
6	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MARIA AURELIA	HUAMAN QUISPE	41983040	CARMEN ALTO
7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	KATIA	LOAYZA GUILLEN	41961983	Urb Mariscal C
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	HERLINDA	QUISPE ROCA	44695382	ASOC. JOSE MA
9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MARLENY	NARARRO GONZALEZ	42213810	AV EL EJERCITC
10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	VILMA	CARDENAS CHAMORRO	43975511	JOSE OLAYA 31
11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RHIVILINDA	HILARIO LUCANA	42839808	CIUDAD MAGIS
12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	YIERSNEIT	QUISPE GUTIERREZ	31183431	A
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ROSA LUZ	CCOPA ZAVALA	25848411	JR MICAELA BA

Mostrando 1 - 20 de 70

Registro de Usuario

Nombres :
Ingrese Nombres

Apellidos :
Ingrese Apellidos

DNI :
Ingrese Dni

Dirección :
Ingrese Dirección

Teléfono :
Ingrese Teléfono

Usuario :
Ingrese Usuario

Password :
Ingrese Password

Nuevo **Cancelar**

Figura N° 4.7: Interfaz. Registrar Usuarios del Sistema.
Fuente: Elaboración propia

ADMINISTRACIÓN

- Institución
- Carrera
- Unidad Didáctica
- Semestres
- Matriculas
- Horarios
- Evaluaciones
- Indicadores
- Configuración

Registro de Unidad Didáctica

Administración de Unidad Didáctica

LISTA DE UNIDAD DIDACTICA

Unidad Básica	Carrera Profesional	denominación	sigla	creditos
D08		PRODUCCION AGROPECUARIA LOGICA DE PROGRAMACION AC-241		2
AR03		ENFERMERIA TECNICA SALUD PUBLICA ENP		2
AR03		COMPUTACION E INFORMAS1ORGANIZACION Y ADMINISTR1OGBR-1		3
AR03		COMPUTACION E INFORMAS1MANTENIMIENTO DE LAS TECN1MIC-1		3
AR03		COMPUTACION E INFORMAS1MANTENIMIENTO DE EQUIP1N1EC-1		4
AR03		COMPUTACION E INFORMAS1SEGURIDAD INFORMATICA 1SE-1		2
AR03		COMPUTACION E INFORMAS1TECNICAS DE COMUNICACION1TC-1		2
AR03		COMPUTACION E INFORMAS1LOGICA Y FUNCIONES 1LF-1		2
AR03		COMPUTACION E INFORMAS1CULTURA FISICA Y DEPORTE1CFD-1		2
AR03		COMPUTACION E INFORMAS1INFORMATICA E INTERNET 1I-1		2
AR03		COMPUTACION E INFORMAS1REPARACION DE EQUIPOS DISCO-2		4
AR03		COMPUTACION E INFORMAS1INSTALACION Y CONFIGURACION1CAC-2		4
AR04		INFORMAS1MANTENIMIENTO DE EQUIPOS DE IMPRESION1MEI		3

Mostrando 1 - 20 de 29

REGISTRO DE UNIDAD DIDACTICA

Semestre : SEMESTRE 1

Tipo Unidad Básica : REGULARES

Carrera Profesional : ENFERMERIA TECNICA

Denominación :
Ingrese Denominación

Sigla :
Ingrese Sigla

Creditos :
Ingrese Creditos

Horas Teóricas :
Ingrese Horline

Horas Prácticas :
Ingrese Hitoria

Guardar **Cancelar**

Figura N° 4.8: Interfaz. Registrar Unidad didáctica.
Fuente: Elaboración propia

87

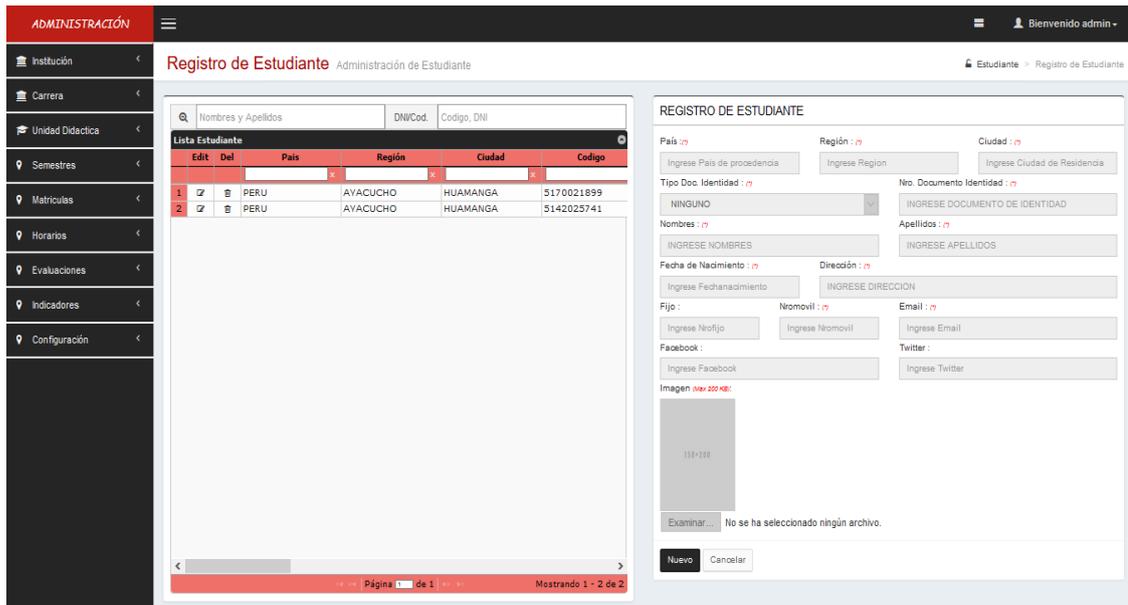


Figura N° 4.9: Interfaz. Registrar Estudiante.
Fuente: Elaboración propia

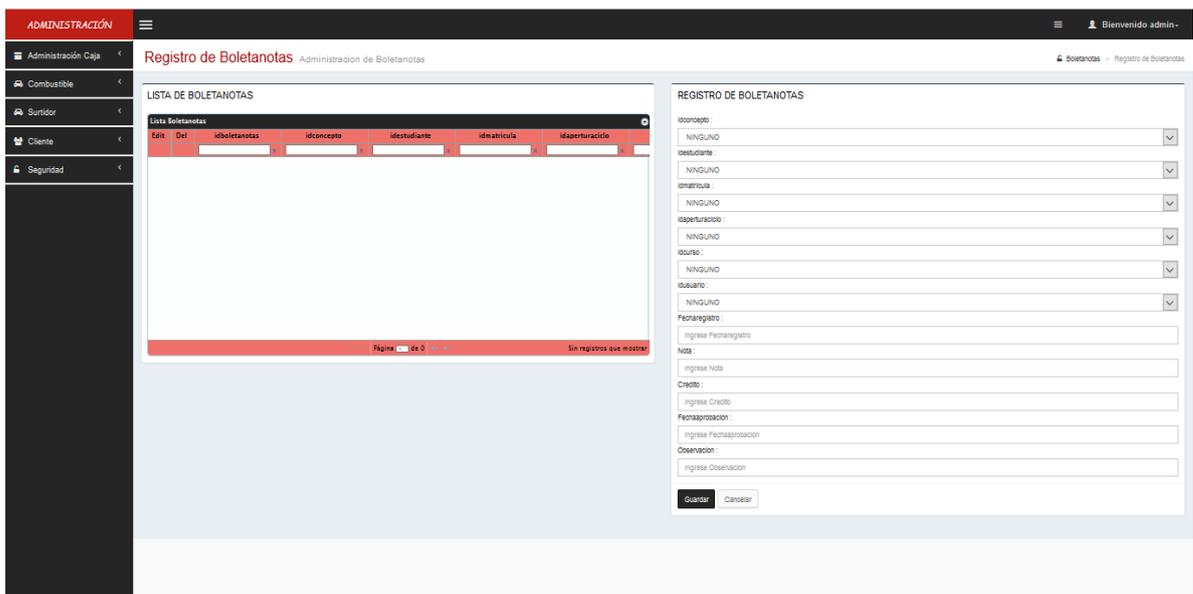


Figura N° 4.10: Interfaz. Registro Calificaciones de los Estudiantes.
Fuente: Elaboración propia



INSTITUTO DE EDUCACION SUPERIOR TECNOLOGICO DE CHURCAMP
 HORARIO DE CLASES DE LA CARRERA PROFESIONAL DE COMPUTACION E INFORMATICA
 SEMESTRE IMPAR 2017

Hora	LUNES			MARTES			MIÉRCOLES			JUEVES			VIERNES		
	II	IV	VI	II	IV	VI	II	IV	VI	II	IV	VI	II	IV	VI
8:00-8:30	ADMINISTRACION DE REDES (12)	TALLER DE PROGRAMACION CON CLASIFICACION DE DATOS (12)	PRODUCCION AUDIOVISUAL (14)	FUNDAMENTOS DE INSTALACION (12)	TALLER DE PROGRAMACION CON CLASIFICACION DE DATOS (12)	PRODUCCION AUDIOVISUAL (14)	INSTALACION DE EQUIPOS DE COMPUTO (12)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	CONEXION DE EQUIPOS DE COMPUTO (12)	INSTALACION Y CONFIGURACION DE REDES (14)	TALLER DE PROGRAMACION CON CLASIFICACION DE DATOS (12)	APLICACIONES MOVILES (13)	COMUNICACION Y REDES (14)	TALLER DE PROGRAMACION WEB (12)	TALLER DE PROGRAMACION WEB (12)
8:30-9:40	ADMINISTRACION DE REDES (12)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	ALGUNAS DE LAS TECNICAS DE PROGRAMACION (14)	OPINION (17)	INTRODUCCION DE DESARROLLO DE SOFTWARE (13)	TALLER DE PROGRAMACION WEB (12)	INSTALACION DE EQUIPOS DE COMPUTO (12)	ADMINISTRACION DE SERVIDORES (14)	CONEXION DE EQUIPOS DE COMPUTO (12)	QUINTAS SEMANAS DE REDES (14)	TALLER DE PROGRAMACION CON CLASIFICACION DE DATOS (12)	PRODUCCION AUDIOVISUAL (14)	PROYECTOS DE PROGRAMACION Y REDES (14)	TALLER DE PROGRAMACION WEB (12)	TALLER DE PROGRAMACION WEB (12)
9:40-10:30	ADMINISTRACION DE REDES (12)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	ALGUNAS DE LAS TECNICAS DE PROGRAMACION (14)	OPINION (17)	INTRODUCCION DE DESARROLLO DE SOFTWARE (13)	TALLER DE PROGRAMACION WEB (12)	INSTALACION DE EQUIPOS DE COMPUTO (12)	ADMINISTRACION DE SERVIDORES (14)	CONEXION DE EQUIPOS DE COMPUTO (12)	QUINTAS SEMANAS DE REDES (14)	TALLER DE PROGRAMACION CON CLASIFICACION DE DATOS (12)	PRODUCCION AUDIOVISUAL (14)	PROYECTOS DE PROGRAMACION Y REDES (14)	TALLER DE PROGRAMACION WEB (12)	TALLER DE PROGRAMACION WEB (12)
10:30-11:20	ADMINISTRACION DE REDES (12)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	ALGUNAS DE LAS TECNICAS DE PROGRAMACION (14)	OPINION (17)	INTRODUCCION DE DESARROLLO DE SOFTWARE (13)	TALLER DE PROGRAMACION WEB (12)	INSTALACION DE EQUIPOS DE COMPUTO (12)	ADMINISTRACION DE SERVIDORES (14)	CONEXION DE EQUIPOS DE COMPUTO (12)	QUINTAS SEMANAS DE REDES (14)	TALLER DE PROGRAMACION CON CLASIFICACION DE DATOS (12)	PRODUCCION AUDIOVISUAL (14)	PROYECTOS DE PROGRAMACION Y REDES (14)	TALLER DE PROGRAMACION WEB (12)	TALLER DE PROGRAMACION WEB (12)
11:20-11:40	R E C E S O														
11:40-12:30	INSTALACION Y CONFIGURACION DE REDES (14)	ADMINISTRACION DE SERVIDORES (14)	TALLER DE PROGRAMACION WEB (12)	INTERPRETACION Y PRODUCCION DE TEXTO (14)	QUINTAS SEMANAS DE REDES (14)	TALLER DE PROGRAMACION WEB (12)	ESTADISTICA GENERAL (13)	PROYECTOS DE PROGRAMACION DISTRIBUIDA (14)	REGULACION E INSPECCION LABORAL (14)	REPARACION DE EQUIPOS DE COMPUTO (12)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	PROYECTO SUPLENIMIENTAL (14)	INSTALACION Y CONFIGURACION DE REDES (14)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	INSTALACION Y CONFIGURACION DE REDES (14)
12:30-13:20	INSTALACION Y CONFIGURACION DE REDES (14)	ADMINISTRACION DE SERVIDORES (14)	TALLER DE PROGRAMACION WEB (12)	INTERPRETACION Y PRODUCCION DE TEXTO (14)	QUINTAS SEMANAS DE REDES (14)	TALLER DE PROGRAMACION WEB (12)	ESTADISTICA GENERAL (13)	PROYECTOS DE PROGRAMACION DISTRIBUIDA (14)	REGULACION E INSPECCION LABORAL (14)	REPARACION DE EQUIPOS DE COMPUTO (12)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	PROYECTO SUPLENIMIENTAL (14)	INSTALACION Y CONFIGURACION DE REDES (14)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	INSTALACION Y CONFIGURACION DE REDES (14)
13:20-14:10	INSTALACION Y CONFIGURACION DE REDES (14)	ADMINISTRACION DE SERVIDORES (14)	TALLER DE PROGRAMACION WEB (12)	INTERPRETACION Y PRODUCCION DE TEXTO (14)	QUINTAS SEMANAS DE REDES (14)	TALLER DE PROGRAMACION WEB (12)	ESTADISTICA GENERAL (13)	PROYECTOS DE PROGRAMACION DISTRIBUIDA (14)	REGULACION E INSPECCION LABORAL (14)	REPARACION DE EQUIPOS DE COMPUTO (12)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	PROYECTO SUPLENIMIENTAL (14)	INSTALACION Y CONFIGURACION DE REDES (14)	TALLER DE PROGRAMACION DISTRIBUIDA (14)	INSTALACION Y CONFIGURACION DE REDES (14)

LEYENDA	HORAS	
14	Ing. JORGE ALBERTO YACHI SOBERANES	24
11	Ing. ZOSIMO ANTONIO NAUPA ROMERO	19
13	Ing. JOSE CARLOS FLORES AYALA	19
12	Bach. NESTOR CONDE JAULES	24
03	Lic. NILVA PEÑA ALMIDON	02
10	Lic. EDITH POMA GARCIA	02
06	ING. VICTORIA VEGA	02
17	Doc. EDWARD MARQUEZ LUCAS	02

Figura N° 4.11: Interfaz. Generar Horarios Semestrales.
 Fuente: Elaboración propia

Figura N° 4.12: Interfaz. Generar indicadores académicos.
 Fuente: Elaboración propia

ADMINISTRACIÓN Bienvenido admin

Registro de Matricula Administración de Matricula

LISTA DE MATRICULA

Editar	Eliminar	Semestre	Estudiante	Carrera	N° Comprobante	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	SEMESTRE 2017 I	ERICK SIMON ESCALANTE O	ENFERMERIA TECNICA	09	11
<input checked="" type="checkbox"/>	<input type="checkbox"/>	SEMESTRE 2017 I	JOSE CARLOS	COMPUTACION E INFORMATICA	23	11
<input checked="" type="checkbox"/>	<input type="checkbox"/>	SEMESTRE 2017 I	JOSE CARLOS	COMPUTACION E INFORMATICA	23	11

Página 1 de 1 Mostrando 1 - 3 de 3

REGISTRO DE MATRICULA

Semestre: SEMESTRE 2017 I

Carrera: COMPUTACION E INFORMATICA

Estudiante: 42025741 HUAMANI QUISPE MARIA ISABEL

Nro Comprobante: 005

Fecha Matricula: 12/12/2016

Observacion: Ingrese Observacion

Cursos:

- SEMESTRE 1 - INTEGRACION DE LA
- SEMESTRE 1 - MANTENIMIENTO DE
- SEMESTRE 2 - REPARACION DE EQUI
- SEMESTRE 3 - MEDIO AMBIENTE Y D
- SEMESTRE 2 - OFIMATICA

Guardar Cancelar

Figura N° 4.13: Interfaz. Matricular Estudiante.
Fuente: Elaboración propia

ADMINISTRACIÓN Bienvenido admin

Registro de Unidad Didáctica Administración de Unidad Didáctica

LISTA DE UNIDAD DIDACTICA

Editar	Eliminar	Idcurso	Idciclo	Idtipocurso	Idcarrera
Sin registros que mostrar					

Página 1 de 0

REGISTRO DE UNIDAD DIDACTICA

Semestre:

Tipo Unidad Basica:

Carrera Profesional:

Denominación: LOGICA DE PROGRAMACION

Sigla: CI123

Creditos: 3

Horas Teoricas: 2

Horas Practicas: 6

Guardar Cancelar

Figura N° 4.14: Interfaz. Registro de Unidades Didácticas
Fuente: Elaboración propia

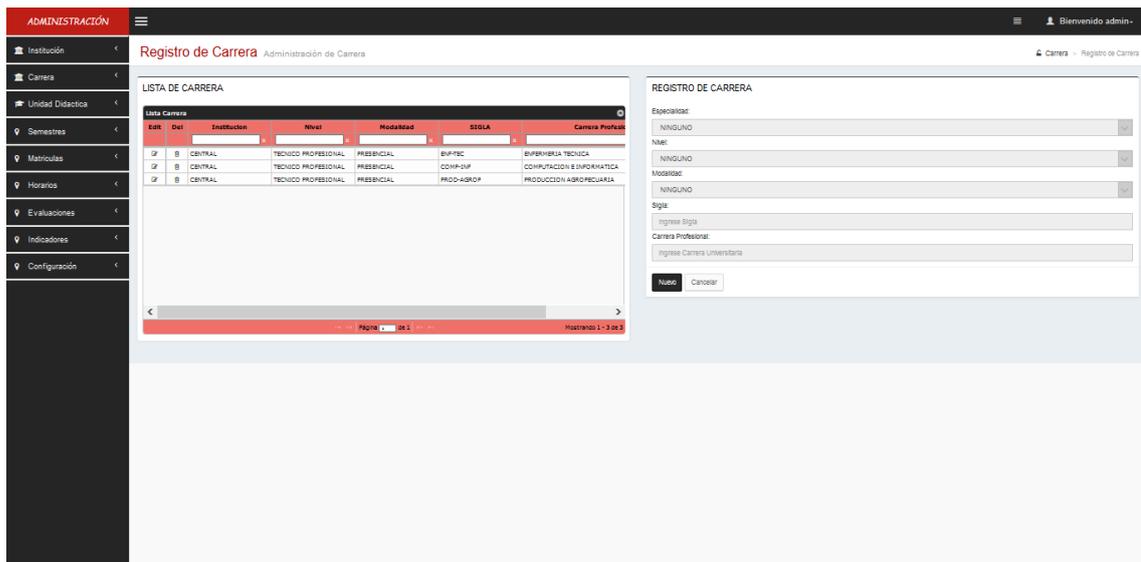


Figura N° 4.15: Interfaz. Registro de Carreras profesionales
Fuente: Elaboración propia

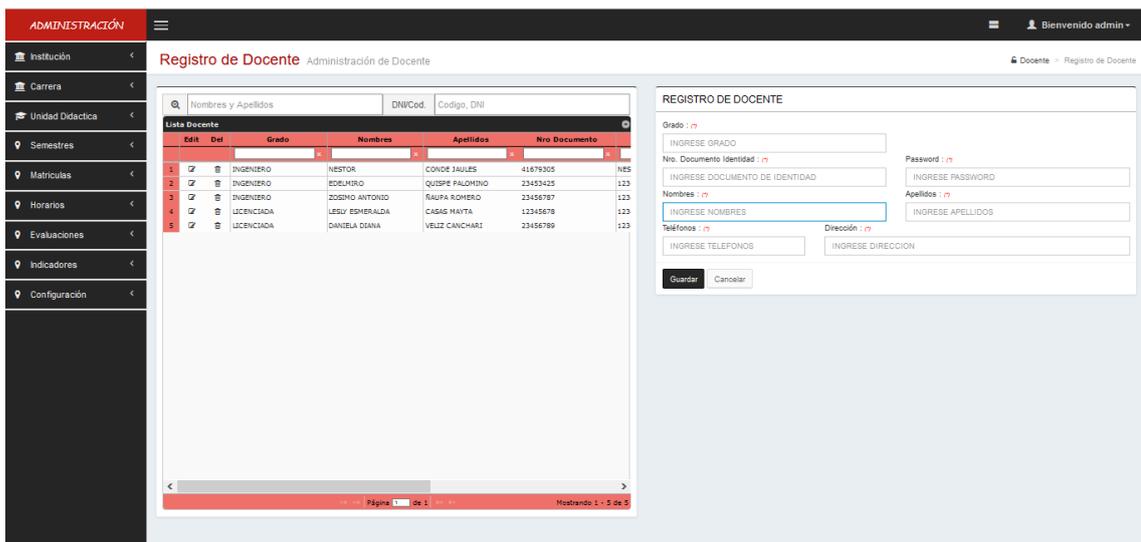


Figura N° 4.16: Interfaz. Registro de docentes
Fuente: Elaboración propia

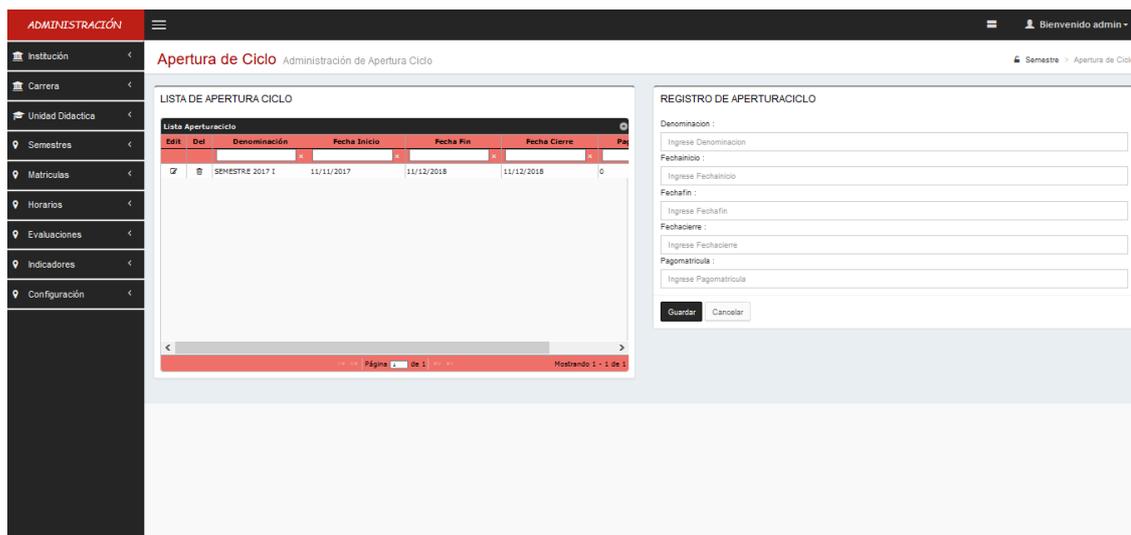


Figura N° 4.17: Interfaz. Apertura de semestre académico
Fuente: Elaboración propia

REPORTES DE PRUEBAS UNITARIAS

Tabla N° 4.54: Reporte de Pruebas de Integración

N° H.U.	N° T.I.	TAREAS DE INGENIERIA	RESULTADO
1	1	Registrar usuario	Satisfactorio
2	2	Iniciar sesión con usuario del sistema	Satisfactorio
3	3	Cerrar sesión	Satisfactorio
4	4	Actualizar perfil del usuario	Satisfactorio
5	5	Registrar carreras profesionales	Satisfactorio
	6	Actualizar carreras profesionales	Satisfactorio
6	7	Registrar estudiante	Satisfactorio
	8	Actualizar estudiante.	Satisfactorio
	9	Eliminar estudiante	Satisfactorio
7	10	Registrar docente	Satisfactorio
	11	Actualizar docente	Satisfactorio

N° H.U.	N° T.I.	TAREAS DE INGENIERIA	RESULTADO
	12	Eliminar docente	Satisfactorio
8	13	Registrar unidades didácticas	Satisfactorio
	14	Actualizar unidades didácticas	Satisfactorio
9	15	Registrar semestre académico.	Satisfactorio
10	16	Registrar inicio de semestre académico	Satisfactorio
11	17	Asignar unidad didáctica y docente por cada ciclo académico.	Satisfactorio
12	18	Matricular estudiante	Satisfactorio
	19	Actualizar matrícula de estudiante	Satisfactorio
13	20	Registrar notas por unidad didáctica y por estudiante.	Satisfactorio
14	21	Generar reporte de aprobados y desaprobados por semestre.	Satisfactorio
15	22	Generar Reporte de notas por estudiante	Satisfactorio
16	23	Generar ranking de alumnos por semestre académico.	Satisfactorio
17	24	Generar Horarios Semestrales	Satisfactorio

Fuente: Elaboración propia

REPORTES DE PRUEBA DE ACEPTACIÓN

Tabla N° 4.55: Reporte de Pruebas de Aceptación “Registrar usuario.

N° CASO PRUEBA	Número: 001
Propósito	El sistema debe verificar que los datos ingresados sean correctos y permitir crear la cuenta del usuario.
ACTIVIDAD	
Inicialización	Registrar usuario
Descripción de datos de entrada	El usuario ingresa su nombre. El usuario ingresa sus apellidos. El usuario ingresa contraseña
RESULTADOS	
Esperados	Registro de usuario con éxito.
Reales	Registra al usuario.

Fuente: Elaboración propia

Tabla N° 4.56: Reporte de Pruebas de Aceptación “Iniciar sesión”

N° CASO PRUEBA	Número: 002
Propósito	El sistema debe verificar que el login y la contraseña sean correctas y permitir el acceso al usuario
ACTIVIDAD	
Inicialización	Iniciar sesión
Descripción de datos de entrada	El usuario ingresa nombre de usuario o correo electrónico. El usuario ingresa la contraseña para acceder al sistema.
RESULTADOS	
Esperados	Usuario en sesión
Reales	Usuario inició sesión en el sistema

Fuente: Elaboración propia

Tabla N° 4.57: Reporte de Pruebas de Aceptación “Registrar docente”

N° CASO PRUEBA	Número: 004
Propósito	El sistema debe verificar que los datos ingresados sean correctos para guardar.
ACTIVIDAD	
Inicialización	Registrar docente
Descripción de datos de entrada	El usuario ingresa su información personal El usuario ingresa conocimiento El usuario ingresa experiencia laboral El usuario ingresa competencias personales El usuario ingresa formación académica
RESULTADOS	
Esperados	El registro de docente con éxito
Reales	El sistema muestra mensaje “Docente registrado”.

Fuente: Elaboración propia

Tabla N° 4.58: Reporte de Pruebas de Aceptación “Registrar estudiante”

N° CASO PRUEBA	Número: 005
Propósito	El sistema debe verificar los datos ingresados y después guardar.
ACTIVIDAD	
Inicialización	Registrar estudiante
Descripción de datos de entrada	El usuario ingresa su código El usuario ingresa nombres El usuario ingresa apellidos El usuario ingresa carrera profesional. El usuario ingresa semestre académico.
RESULTADOS	
Esperados	Se registró estudiante con éxito.
Reales	El sistema muestra el mensaje “Estudiante registrado”.

Fuente: Elaboración propia

Tabla N° 4.59: Reporte de Pruebas de Aceptación “Registrar unidades didácticas”

N° CASO PRUEBA	Número: 006
Propósito	El sistema debe verificar que los datos ingresados por el usuario estén correctos, en caso contrario debe mostrar los campos mal ingresados.
ACTIVIDAD	
Inicialización	Registrar unidades didácticas
Descripción de datos de entrada	El usuario selecciona registrar unidades didácticas y luego llena todos los campos del formulario de “registrar unidades didácticas”
RESULTADOS	
Esperados	Resultado del registro de unidades didácticas.
Reales	Resultado del registro de unidades didácticas y guardadas en la base de datos.

Fuente: Elaboración propia

4.2. DISCUSIONES

- a. Dentro de los resultados obtenidos, es importante resaltar que la aplicación web implementada, coincide en gran medida con otras aplicaciones desarrolladas anteriormente, tanto a nivel de funcionalidad y la finalidad que persiguen, con la diferencia que en otras aplicaciones, se utilizaron otras tecnologías de implementación y otros niveles de detalle, pero finalmente coinciden de manera general con el desarrollo de una aplicación web para automatizar la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa
- b. En los objetivos se hace énfasis la automatización de la gestión académica del instituto, y con los artefactos de software generados se logra implementar los objetivos planteados.
- c. Los resultados obtenidos en la presente investigación pueden ser aplicables a otros institutos públicos; ya que estas están sujetas a realizar procesos de automatización de gestión académica con la finalidad de acreditación y mejora continua de acuerdo las exigencias del ministerio de educación.
- d. Dentro de las limitaciones que existieron en el desarrollo de esta investigación, se puede citar a la falta de experiencia en procesos de automatización en el IESTPCH y la falta de apoyo de los especialistas a la hora de recabar información para la implementación de la aplicación.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- a. Se automatizó el registro de las evaluaciones de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa. El proceso de automatización puede verificarse en las historias de usuario que están plasmadas en la tabla: 4.15, tareas de ingeniería que se encuentran en la tabla: 4.39 e interfaz de usuario que se muestra en la figura N° 4.9.
- b. Se automatizó la generación de indicadores académicos de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa. El proceso de automatización puede verificarse en las historias de usuario que están plasmadas en las tablas: 4.17 y 4.18, tareas de ingeniería que se encuentran en las tablas: 4.41, 4.42 e interfaz de usuario que se muestra en la figura N° 4.11.
- c. Se automatizó la administración de semestres académicos de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa. El proceso de automatización puede verificarse en las historias de usuario que están plasmadas en las tablas: 4.11, 4.12 y 4.13, tareas de ingeniería que se encuentran en las tablas: 4.35, 4.36 y 4.37 e interfaz de usuario que se muestra en la figura N° 4.16.
- d. Se automatizó la administración de las unidades didácticas como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa. El proceso de automatización puede verificarse en las historias de usuario que están plasmadas en la tabla: 4.10, tareas de ingeniería que se encuentran en las tablas: 4.33, 4.34 e interfaz de usuario que se muestra en la figura N° 4.7.
- e. Se automatizó las matrículas como parte de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa. El proceso de automatización puede verificarse en las historias de usuario que están plasmadas en las tablas: 4.7, 4.8, 4.14, tareas de ingeniería que se

encuentran en las tablas: 4.25, 4.26, 4.27, 4.28, 4.37 y 4.38 e interfaz de usuario que se muestra en la figura N° 4.12.

- f. Se automatizó la administración de horarios de la gestión académica del Instituto de Educación Superior Tecnológico Público Churcampa. El proceso de automatización puede verificarse en las historias de usuario que están plasmadas en la tabla: 4.19, tareas de ingeniería que se encuentran en la tabla: 4.43 e interfaz que se muestra en la figura N° 4.10.
- g. Finalmente, podemos precisar que se desarrolló una aplicación web para la gestión académica del Instituto de Educación Superior Tecnológico público Churcampa.

5.2. RECOMENDACIONES

- a. Investigar e implementar aula virtual con la finalidad de mejorar la gestión académica en el Instituto de Educación Superior Tecnológico Público de Churcampa, y de esta manera mejorar la calidad educativa.
- b. Investigar e implementar una aplicación web para el proceso de admisión en el Instituto de Educación Superior Tecnológico Público de Churcampa.
- c. Investigar e implementar portal de transparencia de acuerdo a las normas de gobierno electrónico en el Instituto de Educación Superior Tecnológico Público de Churcampa.

BIBLIOGRAFÍA

1. Anaya, A. (2007). *Proposito de Programacion Extrema*. Colombia: Colombia S.A.
2. Batalla, P. (2006). *Extreme Programming*. Colombia: Pearson Educacion.
3. Batista, P. (2004). *Metodologia de Investigacion*. España: Mc Graw-Will.
4. Beck, F. (2000). *Una Explicacion de Programacion Extrema: Aceptar el Cambio*. España: Addison-Wesly Iberoamerica.
5. Beck, K. (1999). *Extreme Programming Explaing*. EE.UU: Addison Wesly Iberoamericana.
6. Bernal, C. (2010). *Metodologia de la Investigacion*. Colombia: Pearson Educacion.
7. Bruegge, B., & Dutoit, A. (2010). *Ingenieria de Software Orientada a Objetos*. New Jersy: Pretince Hall.
8. Cabellé, S. (2005). *Aplicaciones Dstribuidas en Java*. España: Delta.
9. Carrasco, S. (2005). *Metodologia de Investigacion Cientifica*. Lima: San Marcos.
10. Cuesta, P. (1999). *Desarrollo de Aplicaciones Distribuidas Basadas en Tecnologia Web*. España: Universidad de Vigo.
11. Delgado, L. (2006). *Mejoramiento de la Gestion Educativa a travez de Procesos de Calidad debe bajar los indices de desersion en el Colegio de María de Bogota*. Universidad de Granada: Granada.
12. Eckel, B. (2007). *Piensa en Java*. España: Pretince Hall.
13. Fernandez, E. (2003). *Evaluacion de la Gestion Institucional en Instituciones de Educacion Superior Privadas*. Mexico: Universidad Valle de Mexico.

14. Fernandez, E. (2009). *Software para los Departamentos Academicos de la UNSCH*. Ayacucho: UNSCH.
15. Fernandez, F. (2014). *Procesos Educativos Automatizados*. Mexico: Pretince Hall.
16. Ferrer, A., & Garcia, B. (2003). *Curso Completo de Html*. Mexico: Pretince Hall.
17. Fowler, M. (2000). *Planning Extreme Programing*. New Jersey: EE.UU.
18. Gauchat, J. (2003). *El gran Libro de HTML5, CSS y JavaScript*. Colombia: Marcombo.
19. Hernandez, R. (2010). *Metodologia de Investigacion*. Mexico: McGraw Hill.
20. Heurtel, P. (2009). *Php y MySQL Domine e Desarrollo de un sitio Web Dinamico e Interactivo*. Barcelona: ENI.
21. Holzner, S. (2008). *Manual Java Server Pages*. New Jersey: McGraw Hill.
22. Huaman, R. (2006). *Fidelizacion de Clientes Mediante la gestion de Conocimiento en una empresa Comercial*. Lima: UNI.
23. Izquierdo, L. (2007). *Introduccion a Programacion Orientada a Objetos*. Mexico: Pretince Hall.
24. Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme Programming*. USA: Addison-Wesley Pub Co.
25. Joskowicz, J. (2008). *Reglas y Practicas en Programacion Extrema*. España: Universidad de Vigo.
26. Joyanes, L. (2007). *algoritmos y Programacion Orientada a Objetos*. Mexico: McGraw Hill.
27. Kendall, K., & Kendall, J. (2011). *Analisis y Diseño de Sistemas*. EE.UU: McGraw Hill.

28. Kent, B. (2002). *Planning Extreme Programing*. Usa: Addison Wesley.
29. kent, B., & Fowler, M. (2000). *Planning Extreming Programing*. USA: Addison Wesley.
30. Letelier, P., & Penades , C. (2006). *Metodologias para el Desarrollo de Software: Extreme Programing (XP)*. Valencia: Universidad Politecnica de Valencia.
31. Llovet, A. (2009). *Introduccion a Programacion Orientad a Objetos en Java*. Valencia: Universidad Politecnica de Valencia.
32. Mannino, M. (2007). *Administracion de Base de Datos, Diseño y Desarrollo de Aplicaciones*. Mexico: McGraw Hill.
33. McConell, S. (2004). *Code Complete A Practical handbook of software construccion*. USA: EE.UU.
34. Mendez, F. (2012). *Sistema de Gestion Academica de la Unidad Educativa "Manuel Guerrero"*. Ecuador: Ecuador S.R.
35. O'Reilly, A. (2005). *Extreme Programing*. España: McGraw Hill.
36. Osorio, R. (2008). *Base de datos Relacional Teoria y Practica*. España: ITM.
37. Palomo, M., & Montero, S. (2005). *Ingenieria Web y Patrones de Diseño*. España: Pretince Hall.
38. Peru. (2009). Ley 29394. Lima: Peru.
39. Peru. (2015). Ley 069. Lima: Peru.
40. Porras, E. (2010). *Compracion de Procesos de Desarrollo de Software Usando Metodologia ICONIX y XP Caso: Comercializacion de Tara en la Region Ayacucho*. Lima: UNI.
41. Pressman, R. (2010). *Ingenieria de Software: Un enfoque Práctico*. España: McGraw Hill.

42. Ricardo, R. (17 de agosto de 2016). *Informatica*. Obtenido de <http://www.pmoinformatica.com/2016/11/los-5-valores-de-la-programacion.html>
43. Rob, C. (2004). *Sistema de Base de datos*. España: Nieto.
44. Rosenberg, D., & Stephens, M. (2007). *Use Case Driven Object Modeling*. España: McGraw Hill.
45. Saldaña, R. (2012). *Sistema Web*. España: Pearson Educacion.
46. Sanchez, J. (2007). *Sistema Gestores de Base Datos*. España: McGraw Hill.
47. Silva, D., & Mercerat, B. (2002). *Construyendo Aplicaciones Web con una Metodologia de Diseño Orientada a Objetos*. España: McGraw Hill.
48. Smith, C. (2011). *Control de Procesos Automaticos: Teoria y Practica*. Noriega: Noriega Editores.
49. Sommerville, I. (2005). *Ingenieria de Software*. Mexico: Pearson Educacion S.A.
50. UML. (setiembre de 23 de 2016). *UML*. Obtenido de <http://www.um.es/geograf/sigmur/sigpdf/temario-09.pdf>
51. Vinicio, M. (2012). *Desarrollo de Sistemas de Informacion de Gestion Academica "Gonzalo Rubio de Orbe"*. Otavalo: Otavalo.
52. Wikispace, O. (agosto de 17 de 2016). *Wikispace*. Obtenido de <http://programacion-extrema.wikispace.com/>
53. Zevallos, M. (2015). *Fundamentos de CSS3 y aplicaciones Practicas*. Mexico: Pearson Educacion.

ANEXO A
OPERACIONALIZACIÓN DE VARIABLE

NOMBRE DE LA VARIABLE	DIMENSIONES	INDICADORES	ITEMS O PREGUNTAS
GESTION ACADEMICA	MATRICULA	Cronograma	¿Cómo se determina el cronograma de matrícula?
		Medio de matricula	¿Cómo se realiza el proceso de la matrícula, explique el procedimiento?
		Datos de la matricula	¿Cuáles son los datos importantes que se considera en la matricula?
	EVALUACIONES	Notas	¿Usted como registra las notas?
		Aspectos de la evaluación	¿Cuál son los aspectos de la calificación de cada unidad didáctica?
		Cantidad de evaluaciones por curso	¿Cuántas evaluaciones realiza en cada unidad didáctica durante el semestre?
	INDICADORES ACADEMICOS	Boleta de notas	¿Qué datos se considera en la boleta de nota?
		Primero puestos	¿Cómo generan el orden de puestos los estudiantes?
		Méritos	¿Cuál es el procedimiento de generación de orden de méritos, detalle? ¿Para orden de méritos que datos se consideran?
	SEMESTRE ACADEMICO	Datos del semestre	¿Qué datos importantes se considera en un semestre académico?
		Unidades didácticas en el semestre	¿Cómo se asignan las unidades didácticas a cada semestre y carrera profesional?
		cronograma	¿Cuál es proceso de apertura de semestre académico, explique?
	UNIDADES DIDACTICAS	Datos de unidad didáctica	¿Qué datos importantes tiene la unidad didáctica?
		Itinerario formativo	¿Cómo está organizado el itinerario formativo?
		Contenido	¿Cómo están organizados las unidades didácticas?
	HORARIOS	Datos	¿Qué información se considera para generar horario de clases?
		Responsables	¿Quiénes son los responsables de elaborar el horario de clases?
		Horas teóricas - practicas	¿Cómo están organizado las horas de las unidades didácticas?



ANEXO B

CUESTIONARIO PARA LA ENCUESTA A ESTUDIANTES IESTP CHURCAMP

Formulario para la encuesta a los docentes del I.E.S.T.P- CHURCAMP

- 1 ¿Cómo se determina el cronograma de matrícula?
.....
.....
.....
- 2 ¿Cómo se realiza el proceso de la matrícula, explique el procedimiento?
.....
.....
.....
- 3 ¿Cuáles son los datos importantes que se considera en la matricula?
.....
.....
.....
- 4 ¿Cómo es el procedimiento de registro de notas, explique el procedimiento?
.....
.....
.....
- 5 ¿Cuál son los aspectos de la calificación de cada unidad didáctica?
 ✓ Competencias específicas ()
 ✓ Competencias de empleabilidad ()
 ✓ Situaciones reales en el centro de prácticas ()
 ✓ Otros () Especifique.....
- 6 ¿Cuántas evaluaciones realiza en cada unidad didáctica durante el semestre?
 ✓ 2 evaluaciones ()
 ✓ 3 evaluaciones ()
 ✓ 4 evaluaciones ()
 ✓ Más de 4 ()
- 7 ¿Qué datos se considera en la boleta de nota?
.....
.....
.....
- 8 ¿Cómo generan el orden de puestos los estudiantes?
.....
.....
.....
- 9 ¿Cuál es el procedimiento de generación de orden de méritos, detalle?
.....
.....
.....

.....
.....
10 ¿Para orden de méritos que datos se consideran?

.....
.....
.....

11 ¿Qué datos importantes se considera en un semestre académico?

.....
.....
.....

12 ¿Cómo se asignan las unidades didácticas a cada semestre y carrera profesional?

- ✓ Itinerario ()
- ✓ Distribución de carga académica ()
- ✓ De la carrera profesional ()
- ✓ Otro () Especifique.....

13 ¿Cuál es proceso de apertura de semestre académico, explique?

.....
.....
.....

14 ¿Qué datos importantes tiene la unidad didáctica?

.....
.....
.....

15 ¿Cómo está organizado el itinerario formativo?

.....
.....
.....

16 ¿Cómo están organizados las unidades didácticas?

.....
.....
.....

17 ¿Qué información se considera para generar horario de clases?

.....
.....
.....

18 ¿Quiénes son los responsables de elaborar el horario de clases?

- ✓ Docentes ()
- ✓ Personal administrativo ()
- ✓ Directivos ()

