

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA DE CONTROL Y SEGUIMIENTO DE PROYECTOS
EDUCATIVOS”

CASO: GOBIERNO AUTÓNOMO MUNICIPAL DE PUERTO PÉREZ

Para optar al título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

POSTULANTE	:	UNIV. ANCHELHO SANGALLI PACO
TUTOR METODOLÓGICO	:	ING. ENRIQUE FLORES BALTAZAR
TUTOR ESPECIALISTA	:	M.SC. LIC. JUAN JESÚS ROCHA VERA
TUTOR REVISOR	:	ING. RUBÉN DAVID QUISPE AMARU

EL ALTO – BOLIVIA

2020

DEDICATORIA

Dedico el presente trabajo a las personas más importantes de mi vida que sin ellas este trabajo nunca se hubiera concluido.

A mi madre Eufracia Paco Condori quien me dio la vida y me inculco los valores del bien, por su infinito amor, el sacrificio y su ejemplo de fortaleza para luchar en la vida sin importar las circunstancias, a mi padre Gabino Sangalli Medrano quien con su fortaleza y consejos fortaleció mi camino para afrontar los retos.

A mis hermanos y hermanas quienes me enseñaron el valor de sonreír en los momentos malos y darme el apoyo constante durante todo este tiempo.

AGRADECIMIENTOS

Un trabajo de Proyecto de Grado, constituye un gran esfuerzo personal para alcanzar con éxito el objetivo propuesto, dicho trabajo no sería posible realizar sin el apoyo, ayuda, orientación, consejos, sugerencias, conocimientos y experiencias de otras personas.

Un agradecimiento muy especial a mi tutor metodológico, el Ing. Enrique Flores Baltazar, por haberme brindado su apoyo y conocimiento, su experiencia y consejos sin los cuales no hubiera sido posible la culminación del trabajo.

De la misma manera un agradecimiento muy sincero a mi tutor especialista M.Sc. Lic. Juan Jesús Rocha Vera, por guiarme en las diferentes etapas de mi proyecto y por su asesoramiento. Un agradecimiento muy sincero a mi tutor revisor Ing. Rubén David Quispe Amaru por su colaboración y consejos, por sus observaciones y seguimiento constante, pero sobre todo por su paciencia y tiempo necesario para poder concluir este trabajo.

Al Gobierno Autónomo Municipal de Puerto Pérez, Quienes me abrieron las puertas brindándome la confianza la elaboración del proyecto, el apoyo constante que me brindaron y la disponibilidad de tiempo durante la elaboración del proyecto.

Agradezco a mi familia, pero sobre todo a mi madre Eufracia Paco Condori, padre Gabino Sangalli Medrano y hermanos por toda la colaboración en todas las etapas de mi vida.

A mis amigos y amigas, pero sobre todo a quienes me brindaron su amor y cariño, por no dejarme caer, por aceptarme como soy y así permitirme ser parte de sus vidas, por ayudarme a ser más fuerte, por enseñarme a ser mejor cada día, por darme la bendición de tenerlos en mi vida.

A todos muchas gracias, desde el fondo de mi corazón les deseo todo lo mejor, nunca olvidare, y que Dios los bendiga y los cuide siempre.

RESUMEN

Tener la información actualizada y en el momento oportuno es una necesidad primordial en todas las instituciones municipales para que puedan cumplir con sus objetivos, dicha información puede ser manipulada desde cualquier dispositivo y cualquier explorador que tenga una conexión a internet.

El municipio de Puerto Pérez, referente y viendo la falencia de contar con un sistema que automatice todo el seguimiento de los proyectos educativos, hace que el sistema desarrollado sea de gran ayuda a la población, así tener la información al alcance de todos los colegios para ver el estado de cada uno de sus proyectos, donde también se hará el ahorro de tiempo.

En el desarrollo del sistema web se hace con la metodología ágil UWE, UWE detalla el proceso de las aplicaciones, cuenta con cinco modelos que son el análisis de requerimientos, modelo de contenidos, modelo navegacional, modelo de presentación y modelo de procesos.

Para la calidad del sistema web se utiliza la norma ISO 9126 y para la seguridad del sistema se realiza por niveles, tanto del lado del cliente como del lado del servidor.

Palabras clave: Metodología ágil UWE, ISO 9126, Sistema Web, COCOMO II

ABSTRACT

Having the information updated and in a timely manner is a primary need in all municipal institutions so that they can meet their objectives, such information can be manipulated from any device and any browser that has an internet connection.

The municipality of Puerto Pérez, a benchmark and seeing the failure of having a system that automates all the monitoring of educational projects, makes the developed system a great help to the population, thus having the information available to all schools to See the status of each of your projects, where you will also save time.

In the development of the web system it is done with the agile UWE methodology, UWE details the application process, it has five models that are the requirements analysis, content model, navigational model, presentation model and process model.

For the quality of the web system the ISO 9126 standard is used and for the security of the system it is carried out by levels, both on the client side and on the server side.

Keywords: UWE Agile Methodology, ISO 9126, Web System, COCOMO II

ÍNDICE

1.MARCO PRELIMINAR.....	2
1.1. INTRODUCCIÓN.....	2
1.2. ANTECEDENTES.....	3
1.2.1. ANTECEDENTES INSTITUCIONALES.....	3
1.3. ANTECEDENTES ACADÉMICOS.....	4
1.4. PLANTEAMIENTO DEL PROBLEMA.....	5
1.5. PROBLEMA GENERAL.....	6
1.5.1. PROBLEMAS ESPECÍFICOS.....	6
1.6. OBJETIVOS.....	7
1.6.1. OBJETIVO GENERAL.....	7
1.6.2. OBJETIVO ESPECÍFICOS.....	7
1.7. TÉCNICA.....	7
1.8. ECONÓMICA.....	8
1.9. SOCIAL.....	8
1.10. METODOLOGÍA UWE.....	9
1.10.1. FASES DE LA METODOLOGÍA UWE.....	9
1.11. MÉTRICAS DE CALIDAD DE SOFTWARE.....	10
1.12. INGENIERÍA DE COSTOS.....	11
1.13. HERRAMIENTAS.....	11

1.13.1. YII	12
1.13.2.PHP	12
1.13.3. MYSQL	13
1.14. LÍMITES Y ALCANCES	14
1.14.1. LÍMITES	14
1.14.2. ALCANCES	14
1.14.3. APORTES	15
2.MARCO TEÓRICO	17
2.1. INTRODUCCIÓN.....	17
2.2. CONCEPTOS BÁSICOS.....	17
2.3. COMPONENTES O ELEMENTOS DE UN SISTEMA INFORMÁTICO....	17
2.3.1. ACTIVIDADES QUE REALIZA UN SISTEMA INFORMÁTICO	18
2.3.2. SISTEMA.....	20
2.3.3. INFORMACIÓN.....	21
2.3.4. CONTROL	22
2.3.5. SISTEMA DE CONTROL.....	22
2.3.6. SEGUIMIENTO.....	22
2.3.7. PROYECTO.....	23
2.3.8. REQUERIMIENTO	25

2.3.9. EJECUCIÓN	26
2.4. INGENIERÍA DE SOFTWARE	26
2.5. METODOLOGÍA UWE	27
2.5.1. MODELO DE CASOS DE USO	28
2.5.2. MODELO CONCEPTUAL	29
2.5.3. MODELO DE NAVEGACIÓN	30
2.5.4. MODELO DE ESTRUCTURA DE NAVEGACIÓN	31
2.2.5. MODELO DE PRESENTACIÓN.....	32
2.5.6. FASES DE LA UWE	33
2.6. LENGUAJE UNIFICADO DE MODELADO UML	36
2.7. ARQUITECTURA CLIENTE/SERVIDOR	42
2.8. MÉTRICAS DE CALIDAD	44
2.8.1. NORMA ISO/IEC 9126	44
2.8.2. FUNCIONALIDAD:.....	45
2.8.3. FIABILIDAD:	45
2.8.4. USABILIDAD:	45
2.8.5. EFICIENCIA:.....	46
2.8.6. MANTENIBILIDAD:	46
2.8.7. PORTABILIDAD:	46

2.9. COCOMO II.....	49
2.10. HERRAMIENTAS.....	57
2.10.1. FRAMEWORK YII	57
2.10.2.PHP.....	61
2.10.3. MYSQL.....	63
2.10.4. XAMPP	67
2.10.5. MAGICDRAW	70
3.MARCO APLICATIVO	73
3.1. INTRODUCCIÓN.....	73
3.2. PROCESO DE DESARROLLO DE LA METODOLOGÍA UWE.....	74
3.2.1. ANÁLISIS DE REQUERIMIENTOS	74
3.2.2. MODELO DE CONTENIDOS	79
3.2.3. MODELO NAVEGACIONAL.....	80
3.2.4. MODELO DE PRESENTACIÓN.....	81
3.2.5. MODELO DE PROCESO.....	83
3.3. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.....	85
3.4. MODELO VISTA CONTROLADOR MVC.....	85
3.5. MODELO CONCEPTUAL	87
3.6. IMPLEMENTACIÓN DEL SISTEMA	88

3.7. APLICACIÓN DE MÉTRICAS DE CALIDAD.....	96
3.7.1. FUNCIONALIDAD.....	96
3.7.2. CONFIABILIDAD.....	101
3.7.3. USABILIDAD.....	102
3.7.4. MANTENIBILIDAD.....	103
3.7.5. PORTABILIDAD.....	104
3.7.6. EFICIENCIA.....	105
3.8. COSTOS.....	106
3.8.1. COCOMO II.....	106
3.8.2. COSTO DE ELABORACIÓN DE PROYECTO.....	109
3.8.3. COSTO TOTAL DEL SISTEMA.....	110
4. PRUEBAS Y RESULTADOS.....	112
4.1. PRUEBAS.....	112
4.2. RESULTADOS.....	113
5. CONCLUSIONES Y RECOMENDACIONES.....	115
5.1. CONCLUSIONES.....	115
5.2. RECOMENDACIONES.....	115
MANUAL DE USUARIO.....	128
MANUAL DE INSTALACIÓN.....	139

ÍNDICE DE IMÁGENES

<i>Ilustración 1 Esquema ISO/IEC 9126</i>	10
<i>Ilustración 2 Modelo Cocomo</i>	11
<i>Ilustración 3 Framework VII</i>	12
<i>Ilustración 4 Logo PHP</i>	12
<i>Ilustración 5 Base de Datos Mysql</i>	13
<i>Ilustración 6 Componentes de un Sistema</i>	18
<i>Ilustración 7 Monitoreo o Seguimiento de un Programa</i>	23
<i>Ilustración 8 Clasificación de los sistemas de información</i>	24
<i>Ilustración 9 Ciclo de un Proyecto</i>	25
<i>Ilustración 10 Metodología UWE</i>	27
<i>Ilustración 11 Diagrama de Casos de Uso</i>	29
<i>Ilustración 12 Diagrama de Contenidos</i>	29
<i>Ilustración 13 Diagrama de Navegación UWE</i>	30
<i>Ilustración 14 Estereotipos e Iconos del Modelo de Navegación UWE</i>	30
<i>Ilustración 15 Modelo de Estructura de Navegación de UWE</i>	31
<i>Ilustración 16 Modelo de Presentación de UWE</i>	32
<i>Ilustración 17 Estereotipos y sus Iconos para el Modelo de Presentación</i>	33
<i>Ilustración 18 Análisis y Especificación de requisitos</i>	34
<i>Ilustración 19 Diseño de Sistema</i>	34
<i>Ilustración 20 Códigos de Programación</i>	35
<i>Ilustración 21 Prueba del Sistema</i>	35
<i>Ilustración 22 Instalación de Sistemas</i>	36

<i>Ilustración 23 Mantenimiento de Sistemas</i>	36
<i>Ilustración 24 Diagrama de Casos de Uso</i>	38
<i>Ilustración 25 Diagrama de Secuencia</i>	38
<i>Ilustración 26 Modelo de Diagrama de Colaboración</i>	39
<i>Ilustración 27 Diagrama de Clases</i>	39
<i>Ilustración 28 Elementos de un Diagrama de Estado</i>	40
<i>Ilustración 29 Diagrama de Actividad</i>	40
<i>Ilustración 30 Diagrama de Componentes</i>	41
<i>Ilustración 31 Diagrama de Despliegue</i>	42
<i>Ilustración 32 Esquema de Modelo Cliente – Servidor</i>	42
<i>Ilustración 33 Arquitectura Cliente/Servidor</i>	43
<i>Ilustración 34 Atributos de Calidad</i>	44
<i>Ilustración 35 Factores Cocomo</i>	50
<i>Ilustración 36 Distribución del Mercado de Software Actual y Futuro</i>	52
<i>Ilustración 37 Concepto Operacional Cocomo</i>	56
<i>Ilustración 38 Yii Framework</i>	57
<i>Ilustración 39 Flujo de Tareas Yii</i>	59
<i>Ilustración 40 Estructura PHP</i>	61
<i>Ilustración 41 Gestor de Base de Datos Mysql</i>	63
<i>Ilustración 42 Funcionamiento de Mysql Servidor – Cliente</i>	66
<i>Ilustración 43 Servidor Local Xampp</i>	68
<i>Ilustración 44 MagicDraw</i>	70
<i>Ilustración 45 Esquema del Sistema</i>	73

<i>Ilustración 46 Caso de Uso: Administrar Usuario</i>	<i>75</i>
<i>Ilustración 47 Caso de Uso: Solicitud de Proyectos Educativos</i>	<i>75</i>
<i>Ilustración 48 Caso de uso: Control de proyectos Educativos</i>	<i>76</i>
<i>Ilustración 49 Caso de uso: Seguimiento de proyectos educativos</i>	<i>76</i>
<i>Ilustración 50 Modelo Conceptual.....</i>	<i>80</i>
<i>Ilustración 51 Modelo Navegacional.....</i>	<i>81</i>
<i>Ilustración 52 Nombre de estereotipos y sus iconos del Modelo de Presentación</i>	<i>82</i>
<i>Ilustración 53 Modelo de Presentación</i>	<i>82</i>
<i>Ilustración 54 Modelo De Proceso Registro de Proyecto.....</i>	<i>83</i>
<i>Ilustración 55 Modelo de Procesos Seguimiento de Proyecto y Cambio</i>	<i>84</i>
<i>Ilustración 56 Estereotipos y sus Iconos Modelo de Procesos</i>	<i>84</i>
<i>Ilustración 57 Modelo Vista Controlador</i>	<i>86</i>
<i>Ilustración 58 Modelo Conceptual de la Base de Datos</i>	<i>88</i>
<i>Ilustración 59 Menú de Inicio</i>	<i>89</i>
<i>Ilustración 60 Código Fuente - Menú Inicio.....</i>	<i>89</i>
<i>Ilustración 61 Menú - Login - Modulo Login de Usuario.....</i>	<i>90</i>
<i>Ilustración 62 Código Fuente - Login.....</i>	<i>90</i>
<i>Ilustración 63 Menú - Unidad Educativa.....</i>	<i>91</i>
<i>Ilustración 64 Código- Unidad Educativa</i>	<i>91</i>
<i>Ilustración 65 Menú Dirección de Educación - Sub Menú - Proyecto Listado de Proyectos.....</i>	<i>92</i>
<i>Ilustración 66 Código - Listado de Proyecto</i>	<i>92</i>
<i>Ilustración 67 Sub Menú - Carpeta.....</i>	<i>93</i>

<i>Ilustración 68 Código - Sub Menú Carpeta</i>	<i>93</i>
<i>Ilustración 69 Menú Administración - Reportes</i>	<i>94</i>
<i>Ilustración 70 Código - Administración.....</i>	<i>94</i>
<i>Ilustración 71 Administración - Usuarios.....</i>	<i>95</i>
<i>Ilustración 72 Administración - Usuarios.....</i>	<i>95</i>

ÍNDICE DE TABLAS

<i>Tabla 1 Tabla de Modelo Iso 9126</i>	47
<i>Tabla 2 Esquema Cocomo</i>	51
<i>Tabla 3 Modelo Composición de Aplicación</i>	54
<i>Tabla 4 Requerimientos no Funcionales</i>	55
<i>Tabla 5 Estimación de Esfuerzo</i>	57
<i>Tabla 6 Factor Escala</i>	57
<i>Tabla 7 Caso de Uso - Administrador del Sistema</i>	77
<i>Tabla 8 Caso de Uso - Solicitud de Proyecto</i>	77
<i>Tabla 9 Control de Proyectos</i>	78
<i>Tabla 10 Caso de uso - Seguimiento de proyectos</i>	79
<i>Tabla 11 Lista de entradas del usuario del sistema</i>	96
<i>Tabla 12 Lista de salidas de usuario del sistema</i>	97
<i>Tabla 13 Lista de peticiones del usuario al sistema</i>	97
<i>Tabla 14 Lista de archivos lógicos del sistema</i>	98
<i>Tabla 15 Factores de ponderación</i>	98
<i>Tabla 16 Factor de ajuste de complejidad</i>	99
<i>Tabla 17 Preguntas para obtener el grado de usabilidad</i>	102
<i>Tabla 18 Evaluación de mantenibilidad</i>	103
<i>Tabla 19 Resultado General Norma ISO 9126</i>	105
<i>Tabla 20 Constantes a,b,c,d COCOMO II</i>	107
<i>Tabla 21 Conversión de PF a KLDC</i>	107

<i>Tabla 22 Costos de recursos empleados para la elaboración del sistema.</i>	<i>109</i>
<i>Tabla 23 Costo total del sistema</i>	<i>110</i>



CAPÍTULO

I



MARCO

PRELIMINAR

1.MARCO PRELIMINAR

1.1. INTRODUCCIÓN

Hoy en día la mayoría de las instituciones ya sean privadas o públicas, cuentan con sistemas informáticos, puesto que conocen las diversas ventajas y necesidades que pueden satisfacer dichos sistemas. En concreto, un sistema informático nos permite almacenar, procesar y/o acceder a la información cualquiera que sea este, de una manera más rápida, oportuna y actualizada

El sistema de control y seguimiento de proyectos tiene una gran importancia, ya que el personal administrativo podrá acceder a dicha información de cada uno de sus proyectos el cual el principal objetivo es hacer conocer a toda la población la ubicación exacta de ello, así teniendo una información al alcance de cualquier persona, En tal sentido, el Gobierno Autónomo Municipal de Puerto Pérez tienen la necesidad de contar con un sistema informático que facilite y mejore el control y seguimiento de los proyectos educativos de distritales inscritas en el POA. Dentro de los proyectos distritales o menores quedan contemplados las obras, compras, dotaciones, equipamientos y servicios. La cual este desarrollo se realizará con la metodología UWE “UML-BASED WEB ENGINEERING”.

Las herramientas que se utilizan para el desarrollo del sistema de información de seguimiento y control de proyectos, para lo cual se utilizara: Base de datos mysql, lenguaje de desarrollo php, framework Yii, servidor web Apache.

1.2. ANTECEDENTES

1.2.1. ANTECEDENTES INSTITUCIONALES

El Gobierno Autónomo Municipal de Puerto Pérez es la cuarta sección municipal de la provincia Los Andes del departamento de La Paz. Limita al norte con el lago Titicaca hasta la frontera con la República del Perú, al oeste, sur y este con Pucarani y al noroeste con Batallas.

Se accede al Municipio a través de la carretera La Paz - Batallas - Puerto Pérez, ruta que es transitable durante todo el año. Presenta un relieve de planicie ligeramente ondulada a montaña, con una temperatura promedio de 13°C, con precipitaciones marcadas de diciembre a abril. Debido a la presencia del lago Titicaca, esta región presenta una flora y fauna típicas de la cuenca del lago. La población es de origen aymara y está organizada en sindicatos agrarios, centrales y subcentrales.

En su calidad de Máxima Autoridad Ejecutiva (MAE) dentro el Gobierno Municipal de Puerto Pérez tiene la responsabilidad de plantear políticas, planes y programas de carácter estratégico que tengan impacto sustancial en la población del Municipio de Puerto Pérez a través de una gestión municipal orientada a resultados a partir del análisis de la realidad y escenarios aplicando las normas como la Ley N° 1178 de Administración y Control Gubernamentales.

También se hizo la verificación que no se desarrollaron sistemas ante el crecimiento de las tecnologías para dar un alcance máximo a diferentes comunidades que acopla el municipio, ya que el área de educación no cuenta con ningún sistema que realice el control de proyectos, solo hacen el control de cada proyecto de acuerdo a la poa en hojas Excel, del cual hacen la ejecución presupuestaria.

Existen diferentes antecedentes de trabajo de acuerdo al tipo de proyecto que se va realizando.

1.3. ANTECEDENTES ACADÉMICOS

- **Antecedentes Internacionales**

“SISTEMA INFORMÁTICO PARA CONTROL Y SEGUIMIENTO DE PROYECTOS DE LA DIRECCIÓN DE ORDENAMIENTO FORESTAL, CUENCAS Y RIEGO”.

El cual se encarga de realizar el seguimiento de proyectos que viene realizando la Dirección General de Ordenamiento Forestal, Cuencas y Riego (DGFC), Ministerio de Agricultura y Ganadería; este proyecto se encarga de mejorar los procesos de toma de decisiones y actividades gerenciales. Se podrá potenciar la retroalimentación de la información en las actividades relacionadas con el control y seguimiento de proyectos estableciendo un canal oficial de comunicación, permitiendo de esta manera que la información este centralizada y actualizada. (TOBAR, 2013).

- **Antecedentes Nacionales**

SISTEMA WEB DE SEGUIMIENTO Y MONITOREO DE PROYECTOS ORIENTADOS A RESULTADOS CASO: COLEGIO DE ADMINISTRADORES DE EMPRESAS DE LA PAZ “CADELP”

De acuerdo a la revisión de la biblioteca de la UMSA se tiene un sistema similar al cual está desarrollado que consiste en dar seguimiento y monitoreo de los proyectos que se realiza en beneficio de los socios evitando la extensión de tiempo y gasto innecesario para obtener dicha información. (Umsa, 2015).

SISTEMA WEB PARA EL CONTROL Y ADMINISTRACIÓN DE RECURSOS HUMANOS CASO: EMPRESA DE LIMPIEZA INDUSTRIAL TOTES LTDA

El presente proyecto en la Empresa de Limpieza Industrial “TOTE’s LTDA” específicamente en la unidad de Recursos Humanos, que realiza tareas de reclutamiento de personal. El Sistema Web para el Control y Administración de Recursos Humanos fue implementado de acuerdo a las necesidades y requerimientos de la institución, dando como resultados: reportes del estado de la documentación de cada funcionario, una mejor

administración y control de las faltas y permisos del personal, informar sobre contratos en donde existen puestos vacantes, generación de planillas y almacenamiento de la documentación de cada funcionario y contrato en su formato digital para que esta pueda ser consultada de forma inmediata. (CASILLO, 2014).

1.4. PLANTEAMIENTO DEL PROBLEMA

A medida que pasa el tiempo, las instituciones necesitan ser más capaces de ejecutar procesos más exigentes, lo que ha desatado un sin fin de proyectos de acuerdo a las necesidades de cada una de ellas saltando a solucionar problemas en diferentes campos, como el académico, empresarial y hasta personal.

Actualmente el Gobierno Autónomo Municipal de Puerto Pérez, lleva acabo varios proyectos a nivel municipal, cantonal, comunidad; quienes los beneficiarios con los habitantes del municipio. En su gran mayoría el personal dentro de la alcaldía, incluso el alcalde necesita saber la ubicación exacta y el estado de cada uno de los proyectos, el cual surge una necesidad de tener información veraz sobre cada uno de los proyectos que se vayan ejecutando según la unidad educativa que pertenezca, así el concejo educativo o cualquier persona interesada a su unidad educativa pueda tener el estado de su proyecto y el avance que tiene para su ejecución.

Para la adjudicación de los diferentes proyectos distritales es importante la ejecución del POA (dentro del cual se encuentran los proyectos menores de nuestro interés) y ello contempla dos aspectos muy importantes la ejecución financiera y física de los mismos. Estos dos aspectos son algunos de los parámetros para medir la eficiencia y eficacia de la institución. Entonces se hace imprescindible el control y seguimiento de los proyectos.

En la actualidad cada una de las unidades; proyectos, licitación, supervisión y fiscalización, por las que atraviesan cada uno de los proyectos, realizan de manera independiente el seguimiento y control de los proyectos en hojas de cálculo de Excel. Esta

información, es posteriormente centralizada por un operador para la emisión de informes, reportes estadísticos y consiguientemente la toma de decisiones. Por esta situación es que la información no se la obtiene de manera inmediata, fiable y oportuna. Por lo tanto, no se tiene mucha información precisa y fiable con respecto a los diferentes proyectos que se están ejecutando en las diferentes unidades educativas del municipio.

Actualmente no se cuenta con un sistema semiautomatizado para el control y seguimiento de proyectos menores. Tanto el control como el seguimiento se realizan de manera manual registrando como archivos y carpetas.

1.5. PROBLEMA GENERAL

La información relacionada a los proyectos se encuentra en oficinas de Gobierno Autónomo Municipal de Puerto Pérez, los cuales, para ser consultados, son morosas para las partes interesadas, les es difícil saber el estado de cada uno de sus proyectos, información que la institución demora en dar a las unidades educativas, lo que genera que los usuarios internos y externos no estén satisfechos en sus requerimientos de manera oportuna.

1.5.1. PROBLEMAS ESPECÍFICOS

El inadecuado control de los proyectos educativos, frecuentemente no es el más apropiado originando el desconocimiento de su ubicación y el estado del mismo.

- ✓ No se tiene un acceso inmediato a la información referente a los proyectos educativos.
- ✓ La centralización manual de la información genera errores en los datos, además que existe una pérdida de tiempo.
- ✓ Falta de control de los recursos humanos que invierten en la ejecución de los proyectos, haciendo conocer la eficiencia de los mismos.

- ✓ Demora en la emisión de reportes actualizados: por estado, ejecución física y financiera, por tal situación la toma de decisiones es tardía.
- ✓ Dificultad de obtener información acerca de los proyectos educativos.

¿Cómo podrá optimizar el control y seguimiento a los proyectos educativos dentro del Municipio de Puerto Pérez?

1.6. OBJETIVOS

1.6.1. OBJETIVO GENERAL

El objetivo general del proyecto es:

Desarrollar un sistema de control y seguimiento de proyectos educativos para el gobierno autónomo municipal de Puerto Pérez, el cual contribuirá en el acceso a la información de avances y ejecución financiera de proyectos.

1.6.2. OBJETIVO ESPECÍFICOS

- ✓ Implementar una base de datos para el registro de proyectos educativos.
- ✓ Desarrollar módulos para el control y seguimiento de los proyectos educativos.
- ✓ Desarrollar módulos para poder administrar permisos de accesos al sistema para los diferentes usuarios.
- ✓ Capacitar al personal para el buen funcionamiento del Sistema
- ✓ Realizar el diseño para crear un entorno amable con el usuario y satisfacer las necesidades de la institución.

1.7. TÉCNICA

El sistema informático de control y seguimiento de proyectos, beneficia a la institución con el manejo adecuado y responsable de la información, para tal hecho establece nuevas políticas de orden y de manejo de proyectos. La forma en la que dará a conocer informes previos en base a reportes dará significancia en el orden y en el tiempo de ejecución.

De acuerdo a las técnicas de entrevista, donde se hace una conversación dirigida al personal administrativo de la alcaldía en una serie de preguntas tipo entrevistas para poder desarrollar el sistema siempre usando las herramientas y métodos para el desarrollo, por el cual también se optó por la metodología UWE, el cual adecua con todos los procesos de verificación para la entrega de un software.

1.8. ECONÓMICA

Debido a que se trabajará desde el gobierno se recomendó migrar con software libre, esto implica que el código fuente estará abierto a cualquier desarrollador involucrado con la institución, por lo tanto, esto implica independencia total en cuestión de licencia y de desarrollo. Además, que el sistema permitirá un control riguroso sobre el personal administrativo con el objetivo de no realizar gastos en nuevos viajes de verificación por la falsedad de información.

Por otra parte, el uso de esta herramienta y el aprovechamiento de las tecnologías que se usan, mostrando que el Gobierno Autónomo Municipal de Puerto Pérez este a la vanguardia de la tecnología.

1.9. SOCIAL

La necesidad de un control social sobre los proyectos que ejecuta el Gobierno Autónomo Municipal de Puerto Pérez es evidente, debido al alto compromiso que se tienen con las juntas escolares del municipio, en gran medida por las malas experiencias del pasado, es en este sentido que el Software a desarrollar beneficiara tanto a la institución como también a la población en general.

Con la implantación del sistema los usuarios finales; funcionarios de la Alcaldía, responsables de los proyectos, realizarán de manera más fácil, adecuada, ordenada y responsable el control y seguimiento de los diferentes proyectos correspondientes de cada comunidad, contribuyendo así a una administración eficiente y eficaz.

Asimismo, la población en general del municipio tendrá a su disposición información actualizada de los proyectos que se están ejecutando permitiendo así la transparencia administrativa, pues ellos también serán partícipes de dar seguimiento y control a los proyectos.

1.10. METODOLOGÍA UWE

(UML-Based Web Engineering) es una herramienta para modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos).

UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito, (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza el soporte que proporciona la definición de restricciones sobre los modelos, Corporación Universitaria Centro Superior (2017).

1.10.1. FASES DE LA METODOLOGÍA UWE

Captura, análisis y especificación de requisitos: Durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Diseño del sistema: Se basa en la especificación de requisitos producido por el análisis de los requerimientos, el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

Codificación del software: Se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

Pruebas: Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

La Instalación o Fase de Implementación: Proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino.

El Mantenimiento: Es el proceso de control, mejora y optimización del software ya desarrollado e instalado

1.11. MÉTRICAS DE CALIDAD DE SOFTWARE

NORMA DE EVALUACIÓN ISO/IEC 9126 CALIDAD DEL SOFTWARE, Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado “Information technology-Software product evaluation- Quality characteristics and guidelines for their use”; o también conocido como ISO 9126 (o ISO/IEC 9126). Este estándar describe 6 características generales: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad.

Ilustración 1 Esquema ISO/IEC 9126

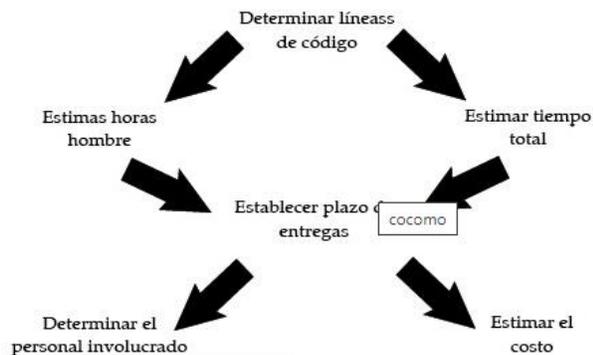


Fuente: (Propia, 2020)

1.12. INGENIERÍA DE COSTOS

Es un modelo con el cual calcularemos cuánto es el coste general del proyecto, para lo cual se empleará el Modelo Constructivo de Costos (o COCOMO, por su acrónimo del inglés (constructive Cost Model) es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Ilustración 2 Modelo Cocomo



Fuente: (Propia, 2020)

1.13. HERRAMIENTAS

Estas herramientas tienen como finalidad, el disminuir el estrés y los tiempos de cada fase, para además mejorar los resultados obtenidos y dar mejores propuestas al cliente. Además de que claro, el nivel de satisfacción con el uso de herramientas es mucho mayor que sin ellas.

Por eso a continuación, vamos a ver las mejores herramientas para el desarrollo de software en cada una de las etapas o fases que conlleva el proceso.

1.13.1. YII

Ilustración 3 Framework YII



Fuente: (Propia, 2020)

Yii es un framework genérico de programación web, lo que significa que se puede utilizar para desarrollar todo tipo de aplicaciones web en PHP. Debido a su arquitectura basada en componentes y a su sofisticada compatibilidad de caché, es especialmente apropiado para el desarrollo de aplicaciones de gran envergadura, como portales, foros, sistemas de gestión de contenidos (CMS), proyectos de comercio electrónico, servicios web compatibles con la arquitectura REST y muchas más.

1.13.2.PHP

Ilustración 4 Logo PHP



Fuente: (Propia, 2020)

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

1.13.3. MYSQL

Ilustración 5 Base de Datos Mysql



Fuente: (Propia, 2020)

Primero lo primero, tienes que saber Como pronunciarlo: MY-ES-KYU-EL' [maɪ,ɛs,kju:'ɛl]. Una compañía sueca llamada MySQL AB originalmente desarrolló MySQL en 1994. La compañía de tecnología de los Estados Unidos Sun Microsystems luego tomó el control por completo cuando compró MySQL AB en el 2008. El gigante de la tecnología estadounidense Oracle adquirió Sun Microsystems en el 2010, y MySQL ha sido propiedad de Oracle desde entonces.

1.14. LÍMITES Y ALCANCES

1.14.1. LÍMITES

El Sistema de Control y Seguimiento de Proyectos Educativos a realizar el monitoreo de los proyectos dentro del Gobierno Autónomo Municipal De Puerto Pérez:

- La aplicación no estará disponible fuera de la institución.
- No se puede realizar la sincronización en línea ya que la institución no cuenta con dominio actualizado.
- Se limitará básicamente al desarrollo del objetivo principal y los objetivos específicos:

1.14.2. ALCANCES

En cuanto al alcance del proyecto, será el diseño y la implementación de un sistema de control y seguimiento físico y financiero de ejecución de proyectos, brindando información actualizada de cada proyecto y la capacitación del personal del municipio en el manejo del sistema automatizado.

Geográficamente se pretende llegar a nivel municipal, ya que los proyectos que se encuentran en ejecución se encuentran en toda la jurisdicción municipal.

- Módulo de Autenticación (Login).
- Módulo de Registro de Proyectos
- Módulo de Seguimiento de Proyectos
- Contemplará la documentación como ser un manual de usuario para el uso de la plataforma.
- Contemplará la documentación como ser un manual de instalación del sistema.

1.14.3. APORTES

Los aportes del sistema son:

El principal aporte a la institución es el software del sistema de información, que apoyara a la alcaldía municipal, automatizando los procesos, manejando la información de manera oportuna y segura, esto con el fin de mejorar el tiempo de respuesta y servicio a la población en general.

El sistema de información será una herramienta útil ya que para su desarrollo se emplea métodos que permitirán una mayor capacidad a los cambios, facilitando la operación y mantenimiento del software.

Se entregará la documentación del proyecto con el manual de usuario a la unidad interesada para las actualizaciones o desarrollo de nuevos proyectos.



CAPÍTULO

II



MARCO

TEÓRICO

2.MARCO TEÓRICO

2.1. INTRODUCCIÓN

El marco teórico constituye el sustento que fundamenta el problema de investigación y que se elabora a partir de conceptos y teorías ya existentes en fuentes diversas. Es decir, dando una pequeña descripción de los conceptos fundamentales de las herramientas y técnicas que contribuyen en el desarrollo e implementación del software.

A continuación, se hace referencia de los términos de proyecto y seguimiento. “El Proyecto no es más que un modelo del emprendimiento a ser realizado, con las previsiones de recursos, de tiempo de ejecución y de los resultados esperados” (IBARROLA, 1972)

2.2. CONCEPTOS BÁSICOS

DEFINICIONES

“Un sistema de información automatizado o basado en computadoras, es la integración de hardware, software, personas, procedimientos y datos. Todos estos elementos se conjugan, trabajando juntos, para proporcionar información básica para la conducción de la empresa. Esta información hace posible que las empresas lleven a cabo sus tareas con mayor calidad y facilidad”. (RENA, 2008).

“Un sistema de información es un conjunto de elementos que interactúan entre si con el fin de apoyar las actividades de una empresa o negocio. Los elementos que interactúan entre si son: personas, hardware, software y datos”. (PERALTA, 2008).

2.3. COMPONENTES O ELEMENTOS DE UN SISTEMA INFORMÁTICO

Los elementos de un sistema de información son:

Personas: Son los que utilizan el SI. Se pueden dividir en dos grandes grupos: los usuarios finales y los especialistas o profesionales. Los usuarios finales son aquellos que operan o interaccionan directamente con el sistema e incluso, quienes reciben reportes o

información generada por el sistema. Entre los profesionales se encuentran: los analistas de sistemas, programadores, administradores del sistema y los capacitadores.

Hardware: Consiste en los equipos, dispositivos y medios necesarios que constituyen la plataforma física mediante la cual, el sistema de información puede funcionar. Se incluyen aquí, por supuesto, los que permiten las comunicaciones y los enlaces de red. Estos recursos son, por ejemplo, computadoras, monitores, impresoras, disquetes o componentes de almacenamiento de información externos, disco óptico, papel de impresión, cableado de red, y otros.

Software o programas: Son el componente lógico, es decir, los programas, las rutinas e instrucciones que conforman el sistema de información.

Datos: Unidades de información que son almacenadas y generadas en el transcurrir de la labor de la empresa. Los datos son almacenados en las denominadas bases de datos o bases de conocimientos (Rena, 2008).

Ilustración 6 Componentes de un Sistema



Fuente: (Rena, 2008)

2.3.1. ACTIVIDADES QUE REALIZA UN SISTEMA INFORMÁTICO

Un Si realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

Entrada de Información: Es el proceso mediante el cual el SI toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomando de otros sistemas o módulos. Este último se denomina interfaces automáticas.

Procesamiento de Información: Es la capacidad del SI para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Estas características de los sistemas permiten la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

Almacenamiento de Información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominados archivos. La unidad típica de almacenamiento es los discos magnéticos o discos duros, los discos duros, los discos flexibles y los discos compactos (CD-ROM).

Salida de Información: La salida es la capacidad de un SI para mostrar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Importante aclarar que la salida de un SI puede constituir la entrada de otro SI o modulo. En este caso también existe una interfaz automática de salida. Por ejemplo, el sistema de control de clientes tiene una interfaz automática de salida con el sistema de contabilidad, ya que genera las pólizas contables de los movimientos procesales de los clientes. (IZAMORAR, 2015).

2.3.2. SISTEMA

Dato: Los datos son, así, la información (valores o referentes) que recibe el computador a través de distintos medios, y que es manipulada mediante el procesamiento de los algoritmos de programación. Su contenido puede ser prácticamente cualquiera: estadísticas, números, descriptores, que por separado no tienen relevancia para los usuarios del sistema, pero que en conjunto pueden ser interpretados para obtener una información completa y específica.

Tipos de Datos: En la informática, cuando hablamos de tipos de datos (o simplemente “tipo”) nos referimos a un atributo que se indica al computador respecto a la naturaleza de los datos que se dispone a procesar. Esto incluye delimitar o restringir los datos, definir los valores que pueden tomar, qué operaciones se puede realizar con ellos, etc.

Algunos tipos de datos son:

Caracteres: Dígitos individuales que se pueden representar mediante datos numéricos (0-9), letras (a-z) u otros símbolos.

Caracteres Unicode: Unicode es un estándar de codificación que permite representar más eficazmente los datos, permitiendo así hasta 65535 caracteres diferentes.

Numéricos: Pueden ser números reales o enteros, dependiendo de lo necesario.

Booleanos: Representan valores lógicos (verdadero o falso).

Ejemplo de Datos:

Algunos ejemplos de datos informáticos pueden ser los que componen la jerarquía de los datos informáticos, en una pirámide ascendente de lo más básico a lo más complejo:

Bits. Cada entrada del lenguaje de código binario, es decir, 1 o 0.

Caracteres. Números, letras o caracteres especiales, formados cada uno a partir de una combinación de bits. Por ejemplo: el número decimal 99 corresponde a 1100011 en binario.

Campos. Conjunto ordenado de caracteres, por ejemplo, una palabra, como el nombre y/o el apellido del usuario que llena un formulario en línea.

Registros. Conjuntos de campos ordenados, como los necesarios para iniciar sesión en nuestro correo electrónico.

Archivos. Conjuntos ordenados de registros, como las cookies que las páginas web guardan en nuestro sistema y contienen la información de las sesiones que hemos iniciado.

2.3.3. INFORMACIÓN

Para la informática, comprendida justamente como la ciencia del manejo de la información, este concepto se comprende como un conocimiento explícito, acumulado por seres vivos o sistemas expertos al interactuar con su entorno. Tiene el valor de poder almacenarse, organizarse y recuperarse.

Tipos de información

La información puede clasificarse de maneras muy distintas, conforme a numerosos criterios. Uno de los más comunes tiene que ver con la relación establecida entre los emisores de la información y sus eventuales o posibles receptores, de la siguiente manera:

Información confidencial o clasificada: Aquella a la que sólo puede acceder un pequeño conjunto de personas, dada la naturaleza secreta, peligrosa, delicada o privada de los datos contenidos en ella.

Información pública: Aquella que, por el contrario, permite el acceso general de cualquiera a su contenido, sin requerir permisos especiales y sin tener ningún grado de privacidad.

Información personal: Aquella que le pertenece a cada persona, es decir, que emana de un individuo concreto, el cual puede decidir con quién compartirla o a quién ofrecérsela.

Información externa: Aquella que emana de un organismo, institución o empresa, y cuyos destinatarios son instancias o personas externas a la misma.

Información interna: Aquella, por el contrario, que emana de un organismo, institución o empresa, con el fin de ser consumida de manera interna, sin salir al exterior de la organización.

2.3.4. CONTROL

Controla diariamente que todas las actividades de los sistemas de información sean realizadas cumpliendo los procedimientos, estándares y normas fijados por la dirección de informática, así como los requerimientos legales.

2.3.5. SISTEMA DE CONTROL

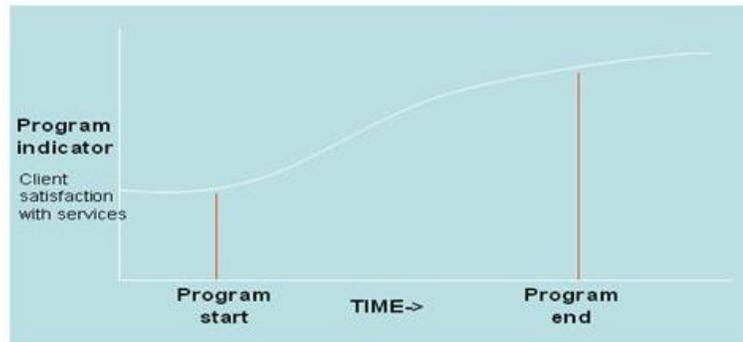
Un sistema de control es un conjunto de dispositivos encargados de administrar, ordenar, dirigir o regular el comportamiento de otro sistema, con el fin de reducir las probabilidades de fallo y obtener los resultados deseados.

Existen dos clases comunes de sistemas de control, sistemas de lazo abierto y sistemas de lazo cerrado. En los sistemas de control de lazo abierto la salida no interviene en la acción de control; mientras que en los de lazo cerrado si se va a requerir conocer la salida para ejercer el control del sistema. Un sistema de lazo cerrado es llamado también sistema de control con realimentación.

2.3.6. SEGUIMIENTO

Es un proceso de implantación del seguimiento tanto del cumplimiento del plan, como sus posibles cambios desde que se realizó la planificación. Se establece mecanismos de control de presupuestos, recursos y fechas fijadas para ver si el plan se ajusta a lo inicialmente programado. Es el proceso sistemático de recolectar, analizar y utilizar información para hacer seguimiento al progreso de un programa.

Ilustración 7 Monitoreo o Seguimiento de un Programa



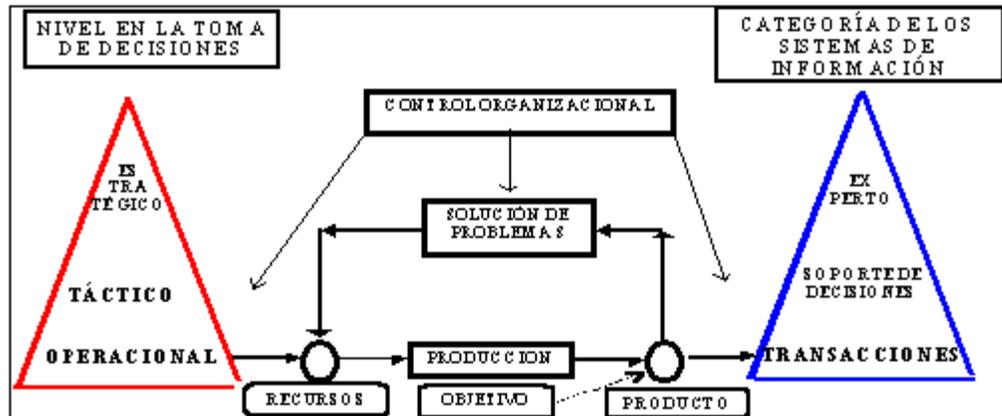
Fuente: (Piaget Arana)

2.3.7. PROYECTO

De la definición de proyectos, vista en el punto anterior, podemos aplicarla a los proyectos informáticos; y decir que: un proyecto informático es un sistema de cursos de acción simultáneos y/o secuenciales que incluye personas, equipamientos de hardware, software y comunicaciones, enfocados en obtener uno o más resultados deseables sobre un sistema de información.

El inicio de un proyecto informático generalmente está dado en la solicitud de requerimientos de los usuarios, y siendo que los diferentes sistemas de Información abordan los diferentes tipos de problemas organizacionales; podemos clasificar a los Sistemas de Información según sean las aplicaciones que necesite cada usuario en: Sistemas de Transacciones, Sistemas de Soporte para la toma de decisiones, y Sistemas Expertos.

Ilustración 8 Clasificación de los sistemas de información



Fuente: (<http://www.cyta.com.ar/>, s.f.)

Etapas de un Proyecto:

Idea: Consiste en establecer la necesidad u oportunidad a partir de la cual es posible iniciar el diseño del proyecto. La idea de proyecto puede iniciarse debido a alguna de las siguientes razones:

Porque existen necesidades insatisfechas actuales o se prevé que existirán en el futuro si no se toma medidas al respecto.

Porque existen potencialidades o recursos aprovechados que pueden optimizarse y mejorar las condiciones actuales.

Porque es necesario complementar o reforzar otras actividades o proyectos que se producen en el mismo lugar y con los mismos involucrados.

Diseño: Etapa de un proyecto en la que se valoran las opciones, tácticas y estrategias a seguir, teniendo como indicador principal el objetivo a lograr. En esta etapa se produce la aprobación del proyecto, que se suele hacer tras la revisión del perfil de proyecto y/o de los estudios de prefactibilidad, o incluso de factibilidad. Una vez dada la aprobación, se realiza la planificación operativa, un proceso relevante que consiste en prever los diferentes recursos y los plazos de tiempo necesarios para alcanzar los fines del proyecto, y, asimismo en establecer la asignación o requerimiento de personas.

Ejecución: Consiste en poner en práctica la planificación llevada a cabo previamente.

Evaluación: Etapa final de un proyecto en la que este es revisado, y se llevan a cabo las valoraciones pertinentes sobre lo planeado y lo ejecutado, así como sus resultados, en consideración al logro de los objetivos planteados

Ilustración 9 Ciclo de un Proyecto



Fuente: (Propia, 2020)

2.3.8. REQUERIMIENTO

Un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

Tipos de Requerimientos

Los requerimientos de software pueden dividirse en 2 categorías: requerimientos funcionales y requerimientos no funcionales.

Los requerimientos funcionales: son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Los requerimientos no funcionales: tienen que ver con características que de una u otra forma puedan limitar el sistema, como, por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad,

2.3.9. EJECUCIÓN

Ejecución se denomina todo proceso en el cual una computadora lee y realiza las operaciones establecidas por un programa o software. En otras palabras, la ejecución es el proceso mediante el cual una computadora lleva a cabo las instrucciones de un programa informático donde se pone en marcha el sistema desarrollado.

2.4. INGENIERÍA DE SOFTWARE

La ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web.

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la Web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que las personas tiendan a realizar todas sus actividades por esta vía. Esta área investiga las tecnologías Web Semántica y las estructuras de datos Linked data desde el prisma de los sistemas abiertos (open source y open data). Estas tecnologías posibilitan estructuras de datos enriquecidas que se utilizan para:

Facilitar la búsqueda de contenidos y servicios.

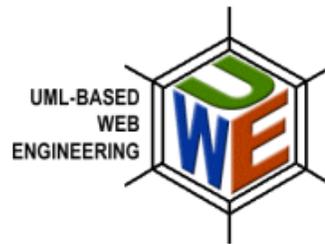
- Crear agentes inteligentes para la automatización de procesos.
- Extraer, procesar y almacenar conocimiento (Knowledge management).
- Integrar sistemas heterogéneos.
- Componer sistemas complejos.

- Filtrar información
- Posibilitar el procesamiento semántico por parte de las máquinas
- Estandarizar y flexibilizar vocabularios

2.5. METODOLOGÍA UWE

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Ilustración 10 Metodología UWE



Fuente: (München, 2020)

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

ACTIVIDADES DE MODELADO DE UWE

El proceso de UWE está dividido en cuatro actividades, que en el siguiente capítulo será profundizado con claridad, estos son los siguientes:

- Análisis de Requerimientos
- Modelo Conceptual
- Modelo de Navegación
- Modelo de Presentación

2.5.1. MODELO DE CASOS DE USO

Para describir los requerimientos funcionales de una aplicación se puede usar un modelo de caso de uso. Este describe un trozo de comportamiento de la aplicación sin revelar su estructura interna.

Un caso de uso es una técnica de modelado usada para describir lo que debería hacer un sistema nuevo o lo que hace un sistema que ya existe.

Los componentes primarios de un modelo de casos de uso (case-use model) son los casos de uso (use cases), los actores y el sistema modelado.

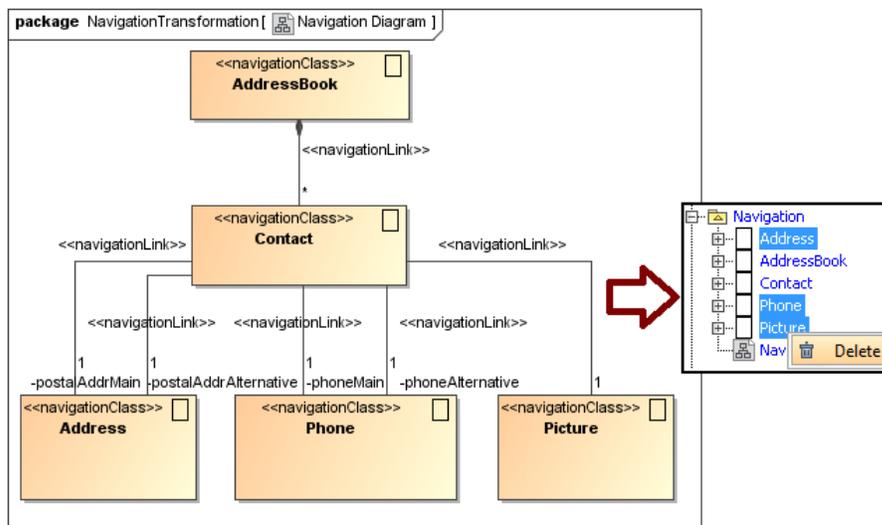
Los casos de uso son descripciones funcionales del sistema; describen cómo los actores pueden usar un sistema.

2.5.3. MODELO DE NAVEGACIÓN

En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos (nodes) y enlaces (links).

Se obtienen el modelo de espacio de navegación y modelo de estructura de navegación, que muestra cómo navegar a través del espacio de navegación. Se obtienen diagramas de clases que representan estos modelos.

Ilustración 13 Diagrama de Navegación UWE



Fuente: (Koch, 2003)

El modelo de navegación se lo realiza con los siguientes estereotipos:

Ilustración 14 Estereotipos e Iconos del Modelo de Navegación UWE

nombres de estereotipos y sus iconos

- | | |
|---|--|
|  clase de navegación |  menú |
|  índice |  pregunta |
|  visita guiada |  clase de proceso |
|  nodo externo | |

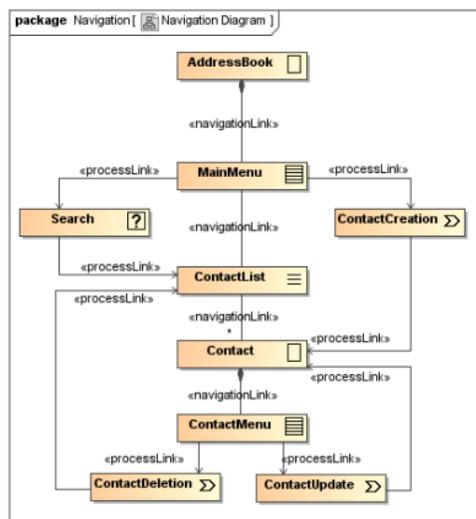
Fuente: (Koch, 2003)

2.5.4. MODELO DE ESTRUCTURA DE NAVEGACIÓN

Este refinamiento consiste en mejorar el modelo de espacio de navegación añadiendo estructuras de acceso como: índices, guías de ruta, consultas y menús.

Las clases estereotipadas para estas estructuras son: <<index>> (especifica acceso directo a todas las instancias del destino proporcionando una lista de todos los elementos desde donde se puede seleccionar para la navegación en la aplicación web), <<guided tour>> (proporciona acceso secuencial a instancias del nodo destino), <<query>> (representa la posibilidad de buscar por instancias en el nodo destino), <<menu>> (son utilizados para estructurar la salida de enlaces desde un nodo).

Ilustración 15 Modelo de Estructura de Navegación de UWE



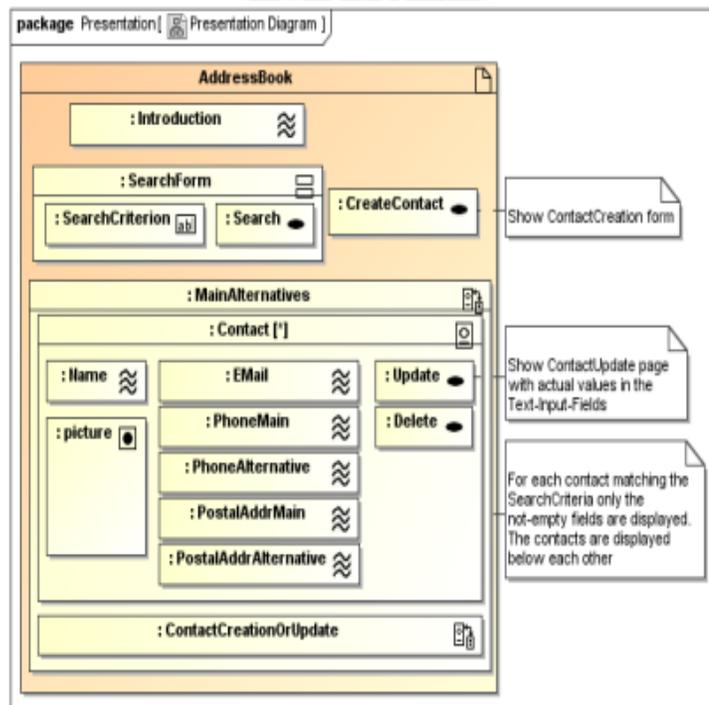
Fuente: (Koch, 2003)

2.2.5. MODELO DE PRESENTACIÓN

Basándose en el modelo anterior y en información adicional recopilada durante el análisis de requerimientos se elabora el modelo de presentación, cuyo propósito es el diseño abstracto de interfaces de usuario.

Se obtienen el modelo de espacio de navegación y modelo de estructura de navegación, que muestra cómo navegar a través del espacio de navegación. Se obtienen diagramas de clases que representan estos modelos.

Ilustración 16 Modelo de Presentación de UWE



Fuente: (Koch, 2003)

El modelo de presentación se los realiza con los siguientes estereotipos:

Ilustración 17 Estereotipos y sus Iconos para el Modelo de Presentación

nombres de estereotipos y sus iconos

 grupo de presentación	 página de presentación
 texto	 entrada de texto
 anda	 fileUpload
 botón	 imagen
 formulario	 componente de cliente
 alternativas de presentación	 selección

Fuente: (Koch, 2003)

2.5.6. FASES DE LA UWE

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrandose además su atención en aplicaciones personalizadas o adaptativas.

Las fases o etapas a utilizar son:

- 1) Captura, análisis y especificación de requisitos:** En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web. Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

Ilustración 18 Análisis y Especificación de requisitos



Fuente: (Google, s.f.)

- 2) Diseño del sistema:** Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

Ilustración 19 Diseño de Sistema



Fuente: (Google, s.f.)

- 3) Codificación del software:** Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

Ilustración 20 Códigos de Programación



Fuente: (Google, s.f.)

4) Pruebas: Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

Ilustración 21 Prueba del Sistema



Fuente: (Google, s.f.)

5) La Instalación o Fase de Implementación: es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final. Esto incluye la implementación de la arquitectura, de la estructura del hiperspacio, del modelo de usuario, de la interfaz de usuario,

de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

Ilustración 22 Instalación de Sistemas



Fuente: (Google, s.f.)

6) El Mantenimiento: es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

Ilustración 23 Mantenimiento de Sistemas



Fuente: (Google, s.f.)

2.6. LENGUAJE UNIFICADO DE MODELADO UML

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Sea

convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grade Booch, James Rumbaugh e Ivar Jacobson. Estos autores fueron contratados por la Empresa Racional Software. Para crear una notación unificada en la que se basa la construcción de sus herramientas CASE. [Bouch, 1999].

Sin embargo, no dice que modelos crear ni cuando se debería de crear, esta es la tarea del proceso de desarrollo de software.

UML es un lenguaje para documentar; cubre toda la documentación de todas las decisiones de análisis, diseño e implementación que debe realizarse al desarrollar un sistema, proporciona un lenguaje para modelar las actividades de planificación de proyectos.

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas. [LAR, 2000].

Diagrama de Casos de Uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores de sistemas, esta es una herramienta muy valiosa, puesto que es una técnica para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. Es decir, ayuda a obtener los requerimientos desde el punto de vista de los usuarios.

Ilustración 24 Diagrama de Casos de Uso

Actor: es que un Actor es un rol que un usuario juega con respecto al sistema.



Casos de Uso: Es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.



Asociación: elemento que conecta el actor con el caso de uso.



Extensión, un caso de uso extiende al original dado que agrega otros pasos a la secuencia del caso de uso existente



Inclusión, Si los pasos en una situación dentro de un caso de uso son los mismos que los de otro, entonces podemos adicionar otro caso de uso que incluya los pasos similares.



Fuente: (Google, s.f.)

Diagrama de Secuencia

Describe el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos.

Ilustración 25 Diagrama de Secuencia

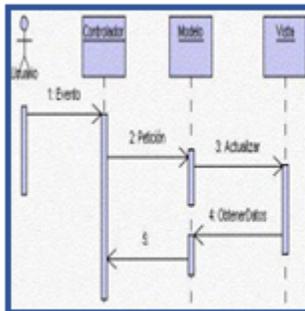


Diagrama de Secuencia: Puede verse con facilidad cómo se distribuyen las tareas entre los componentes. Pueden identificarse los modelos de interacción que dificultan la actualización de software.

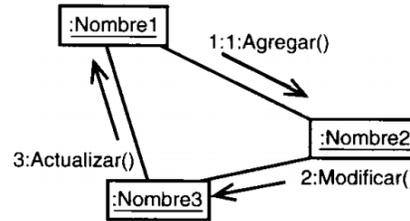
Fuente: (Google, s.f.)

Diagrama de Colaboración

Muestra la interacción entre varios objetos y los enlaces que existen entre ellos. Representa las interacciones entre objetos organizados alrededor de los objetos y sus

vinculaciones. A diferencia de diagrama de secuencia, un diagrama de colaboraciones muestra las relaciones entre los objetos, no la secuencia en el tiempo en que se producen los mensajes.

Ilustración 26 Modelo de Diagrama de Colaboración

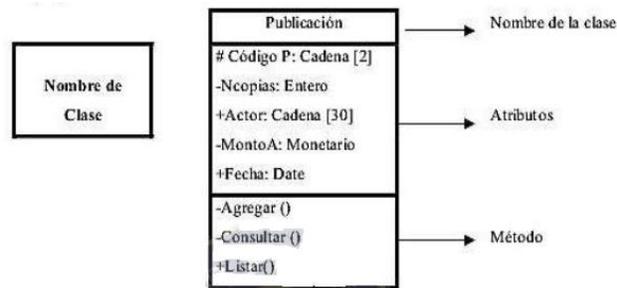


Fuente: (Google, s.f.)

Diagrama de Clases

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones y las relaciones entre estos; los diagramas de clases muestran el diseño del sistema desde un punto de vista estático.

Ilustración 27 Diagrama de Clases



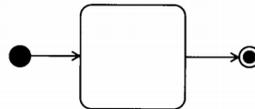
Fuente: (Google, s.f.)

Diagrama de Estados

Representan la secuencia de estados por los que un objeto o una interacción entre objetos pasa durante su tiempo de vida en respuesta a estímulos (eventos) recibidos. Representa lo que en conjunto se puede denominar, una máquina de estados.

Cuando un objeto o una interacción pasa de un estado a otro por la ocurrencia de un evento se dice que ha sufrido una transición; existen varios tipos de transición entre objetos; simples (normales y reflexivas) y complejas.

Ilustración 28 Elementos de un Diagrama de Estado



Fuente: (Google, s.f.)

Diagramas de Actividad

El diagrama de actividad del UML, es muy parecido a los viejos diagramas de flujo. Le muestra los pasos (conocidos como actividades) así como de decisiones y bifurcaciones. Es útil para mostrar lo que ocurre en un proceso de negocios y operación.

Ilustración 29 Diagrama de Actividad



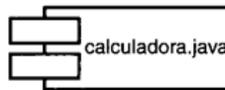
Fuente: (Google, s.f.)

Diagrama de Componentes

Muestra un conjunto de componentes y sus relaciones. Un componente de software es una parte física de un sistema, y se encuentra en la computadora, no en la mente del analista. Puede tomarse como un componente: una tabla, archivo de datos, ejecutables, documentos, Existen tres tipos de componentes:

Componentes de distribución, conforman el fundamento de los sistemas ejecutables (p.e. DLL, ejecutables, controles ActiveX). Componentes para trabajar en el producto, a partir de los cuales se han creado los componentes de distribución (archivos de bases de datos y de códigos).

Ilustración 30 Diagrama de Componentes



Fuente: (Google, s.f.)

Diagrama de plataformas o despliegue

Ilustra la forma en que luce un sistema físicamente cuando sea conjugado. Muestra la configuración de los componentes de hardware los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. A cada actividad se presenta por un rectángulo angosto con las esquinas redondeadas. Una flecha representa la transición de una a otra actividad. Cuenta

con un punto inicial y final. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componente dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes.

Ilustración 31 Diagrama de Despliegue



Fuente: (Google, s.f.)

2.7. ARQUITECTURA CLIENTE/SERVIDOR

Un modelo Cliente Servidor, es un Sistema distribuido entre múltiples procesadores donde hay clientes que solicitan servicios y servidores que los proporcionan. En el modelo cliente-servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.

Ilustración 32 Esquema de Modelo Cliente – Servidor

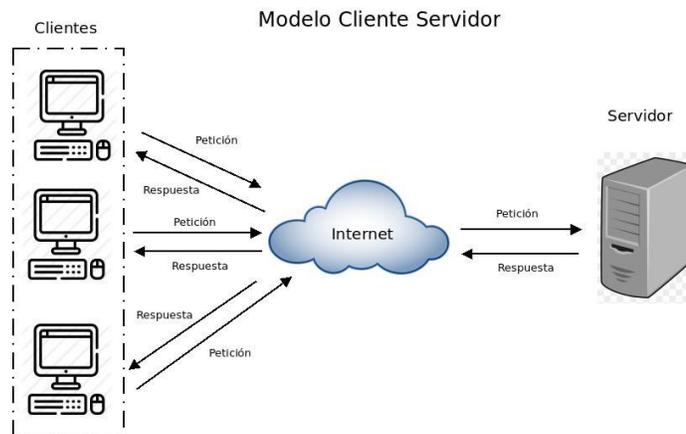


Figura 31: (wikipedia, s.f.)

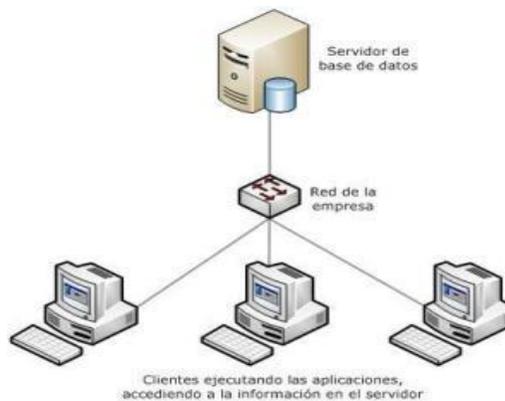
Los servidores hardware tienen fundamentalmente dos funciones, bien “servidores de aplicaciones”, que alojan distintos tipos de programas que pueden llamarse desde y

ejecutarse en los terminales, bien “servidores de bases de datos”, que alojan archivos con datos que pueden ser consultados y/o editados y modificados en las máquinas terminales o clientes; así también pueden ser servidores de ambos tipos simultáneamente.

Si se trata de una red de área local, la interconexión entre el o los servidores y los clientes es directa, mediante un sistema de cable o red inalámbrica; si es una red corporativa distribuida o a través de Internet, la interconexión es indirecta, y la alternativa más común es mediante un modem y vía telefónica.

Una aplicación cliente/servidor típico es un servidor de base de datos al que varios usuarios realizan consultas simultáneamente. El proceso cliente realiza una consulta, el proceso servidor le envía las tablas resultantes de la consulta y el proceso cliente las interpreta y muestra el resultado en pantalla. Los sistemas distribuidos pueden consistir en diversos servidores que alojen datos, de forma que el cliente no tiene por qué conocer exactamente donde se encuentran, simplemente hace una petición de servicio, y es el sistema servidor encargado de localizarlos y proporcionar el resultado de la consulta al usuario que hizo la petición.

Ilustración 33 Arquitectura Cliente/Servidor



Fuente: (wikipedia, s.f.)

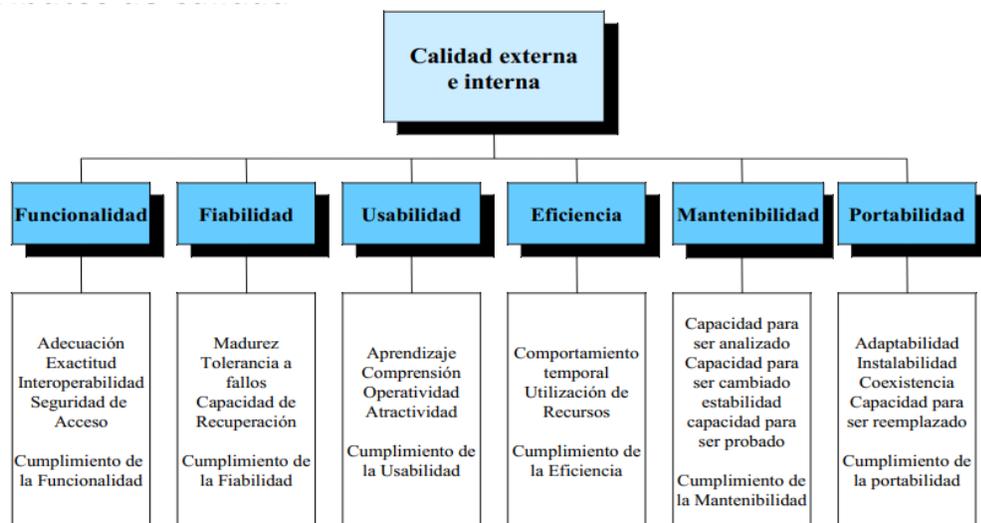
2.8. MÉTRICAS DE CALIDAD

2.8.1. NORMA ISO/IEC 9126

La ISO 9126 es un estándar internacional para evaluar la calidad del software en base a un conjunto de características y sub - características de la calidad. Cada sub - característica consta de un conjunto de atributos que son medidos por una serie de métricas.

Estas métricas miden artefactos obtenidos en etapas tardías del desarrollo de software, aumentando el costo de detección y corrección de errores. Por esta razón, en la literatura ha surgido un mayor interés por la definición de métricas que pretenden evaluar una o varias de las características de calidad definidas en el estándar ISO 9126, en etapas tempranas del desarrollo de software. El modelo de calidad establecido en la primera parte del estándar, ISO 9126, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

Ilustración 34 Atributos de Calidad



Fuente: (Cuomo Castares)

2.8.2. FUNCIONALIDAD: conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen lo indicado o implica necesidades.

Idoneidad: Se enfoca a evaluar si el SW cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.

Exactitud: Permite evaluar si el SW presenta resultados o efectos acordes a las necesidades para las cuales fue creado.

Interoperabilidad: Permite evaluar la habilidad del SW de interactuar con otros sistemas previamente especificados.

Seguridad: Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o promediado, a los programas y datos.

Conformidad: Evalúa si el SW se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.

2.8.3. FIABILIDAD: Conjunto de atributos relacionados con la capacidad de mantener un nivel de presentación bajo condiciones establecidas durante un periodo de tiempo establecido.

Madurez: Permite medir la frecuencia de falla por errores en el SW.

Recuperación: Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesario para lograrlo.

Tolerancia de fallos: Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del SW o de cometer infracciones de su interfaz específica.

2.8.4. USABILIDAD: Conjunto de atributos relacionados con el esfuerzo necesitado para el uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

Comprensión: Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del SW.

Facilidad de Aprender: Establece atributos del SW relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.

Operatividad: Agrupa los conceptos que evalúan la operación y el control del sistema.

2.8.5. EFICIENCIA: Conjunto de atributos relacionados con la relación entre el nivel de desempeño del SW y la cantidad de recursos necesitados bajo condiciones establecidas.

Comportamiento en el tiempo: Atributos del SW relativos a los tiempos de respuesta y de procedimiento de los datos.

Comportamiento de recursos: Atributos de SW relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

2.8.6. MANTENIBILIDAD: Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema SW.

Estabilidad: Capacidad del SW de tener un desempeño normal a pesar de hacerse modificaciones.

Facilidad de análisis: Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.

Facilidad de cambios: Capacidad del que tiene el SW para que la modificación pueda ser válida.

Facilidad de pruebas: Capacidad del que tiene el SW para que la modificación pueda ser válida.

2.8.7. PORTABILIDAD: Conjunto de atributos relacionados con la capacidad de un sistema SW para ser transferido desde una plataforma a otra.

Adaptabilidad: Evalúa la oportunidad para adaptar el SW a diferentes ambientes sin necesidad de aplicarle modificaciones.

Facilidad de instalación: Es el esfuerzo necesario para instalar el SW en un ambiente determinado.

Cumplimiento: Permite evaluar si el SW de adhiere a estándares o convenciones relativas a portabilidad.

Capacidad de reemplazo: Se refiere a la oportunidad y el esfuerzo usando en sustituir el SW por otro producto con funciones similares.

Tabla 1 Tabla de Modelo Iso 9126

Nombre de la métrica: Precisión esperada	Nombre de la métrica: Exactitud de cálculo
Propósito de la métrica: ¿Cuán frecuente no son aceptables las diferencias entre los resultados reales y los resultados razonablemente esperados?	Propósito de la métrica: ¿Cuán frecuente los usuarios finales encuentran resultados inexactos?
Método de aplicación: Hacer una prueba de casos de entrada versus salida y comparar la salida con los resultados razonablemente esperados.	Método de aplicación: Registrar el número de cálculos inexactos basado en especificaciones.
Medición, fórmula y cálculo de elementos de datos: $X = A / T$ A = Número de casos encontrados por los usuarios con una diferencia respecto a los resultados razonablemente esperados más allá de lo aceptable	Medición, fórmula y cálculo de elementos de datos: $X = A / T$ A= Número de cálculos inexactos encontrados por los usuarios T = Tiempo de operación

T = Tiempo de operación

Interpretación del valor medido: $0 < = X$, lo más cercano a 0,0 es lo mejor	Interpretación del valor medido: $0 < = X$, lo más cercano a 0,0 es lo mejor
Tipo de escala de métrica: Ratio	Tipo de escala de métrica: Ratio
Tipo de medida: X = Cantidad / Tiempo A = Cantidad T = Tiempo	Tipo de medida: X = Cantidad / Tiempo A = Cantidad T = Tiempo
Entrada para la medición: Especificación de requerimientos / Manual de operación del usuario / Opinión de usuarios / Reporte de pruebas	Entrada para la medición: Especificación de requerimientos / Reporte de pruebas
Referencia PCVS ISO/IEC 12207: 6.5 Validación 6.3 Aseguramiento de calidad	Referencia PCVS ISO/IEC 12207: 6.5 Validación 6.3 Aseguramiento de calidad
Audiencia objetivo: Desarrollador y usuario	Audiencia objetivo: Desarrollador y usuario
Nota: Los resultados razonablemente esperados podrían ser descritos en una especificación de requerimiento,	Nota

en un manual de usuario o en las expectativas de los usuarios.

Fuente: (Propia, 2020)

2.9. COCOMO II

Breve historia

El modelo COCOMO ha evolucionado debido a los constantes avances en el mercado de desarrollo de software. En el año 1981 Barry Boehm se publicó el modelo COCOMO, acorde a las prácticas de desarrollo de software de aquel momento [Boehm 1981]. Durante la década de los 80, el Modelo se continuó perfeccionando y consolidando, siendo el modelo de estimación de costos más ampliamente utilizado en el mundo.

En el año 1983 se introduce el lenguaje de programación Ada (American National Standard Institute) para reducir los costos de desarrollo de grandes sistemas. Algunos aspectos de Ada provocaron un gran impacto en los costos de desarrollo y mantenimiento, así Barry Boehm y Walker Royce definieron un modelo revisado, llamado Ada COCOMO [Boehm 1989]. En los 90, las técnicas de desarrollo de software cambiaron dramáticamente, surgieron la necesidad de reusar software existente, la construcción de sistemas usando librerías, etc. Estos cambios comenzaron a generar problemas en la aplicación del modelo COCOMO. La solución fue reinventar el modelo. Después de algunos años y de un esfuerzo combinado de USC-CSE (University of Southern California- Center For Software Engineering), IRUS at UC Irvine y organizaciones privadas, aparece COCOMO II. Las incorporaciones a este modelo lo reforzaron e hicieron apto para ser aplicado en proyectos vinculados a tecnologías como orientación a objetos, desarrollo incremental, composición de aplicación, y reingeniería. COCOMO II consta de tres modelos, cada uno de los cuales ofrece una precisión acorde a cada etapa de desarrollo del proyecto. Enunciados en orden creciente de fidelidad son, modelo de Composición de Aplicación, Diseño Temprano y Post Arquitectura.

El USC- CSE implementó los dos últimos modelos en una herramienta de software. Esta herramienta le permite al planificador hacer rápidamente una exploración de las posibilidades de un proyecto, analizando qué efectos provoca el ajuste de requerimientos, recursos y staff sobre la estimación de costos y tiempos. Para evitar confusión el modelo COCOMO original fue modificado con el nombre COCOMO' 81. Así todas las referencias de COCOMO encontradas en la literatura antes de 1995 se refieren a lo que ahora llamamos COCOMO'81. La mayoría de las referencias publicadas a partir de 1995 se refieren a COCOMO II.

Modelo COCOMO II, modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (Modelo Constructivo de Coste). El modelo COCOMO original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria.

Ilustración 35 Factores Cocomo



Fuente: (wikipedia, s.f.)

Los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.

Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.

Implementar una herramienta de software que soportara el modelo.

Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

COCOMO II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura.

Éstos surgen en respuesta a la diversidad del mercado actual y futuro de desarrollo de software.

Esta diversidad podría representarse con el siguiente esquema:

Tabla 2 Esquema Cocomo

Aplicaciones Desarrolladas por Usuarios		
Finales		
Generador	Aplicaciones	Sistemas
de	con	Integrados
aplicaciones	Componentes	
Infraestructura		

Fuente: (Propia, 2020)

Ilustración 36 Distribución del Mercado de Software Actual y Futuro

	<p>Aplicaciones desarrolladas por Usuarios Finales:</p> <ul style="list-style-type: none"> •En este sector se encuentran las aplicaciones de procesamiento de información generadas directamente por usuarios finales, mediante la utilización de generadores de aplicaciones tales como planillas de cálculo, sistemas de consultas, etc. Estas aplicaciones surgen debido al uso masivo de estas herramientas, conjuntamente con la presión actual para obtener soluciones rápidas y flexibles.
	<p>Generadores de Aplicaciones:</p> <ul style="list-style-type: none"> •En este sector operan firmas como Lotus, Microsoft, Novell, Borland con el objetivo de crear módulos pre-empaquetados que serán usados por usuarios finales y programadores.
	<p>Aplicaciones con Componentes:</p> <ul style="list-style-type: none"> •Sector en el que se encuentran aquellas aplicaciones que son específicas para ser resueltas por soluciones pre-empaquetadas, pero son lo suficientemente simples para ser construidas a partir de componentes interoperables. •Componentes típicas son constructores de interfaces gráficas, administradores de bases de datos, buscadores inteligentes de datos, componentes de dominio-específico (medicina, finanzas, procesos industriales, etc.). Estas aplicaciones son generadas por un equipo reducido de personas, en pocas semanas o meses.
	<p>Sistemas Integrados: Sistemas de gran escala, con un alto grado de integración entre sus componentes, sin antecedentes en el mercado que se puedan tomar como base. Porciones de estos sistemas pueden ser desarrolladas a través de la composición de aplicaciones. Entre las empresas que desarrollan software representativo de este sector, se encuentran grandes firmas que desarrollan software de telecomunicaciones, sistemas de información corporativos, sistemas de control de fabricación, etc.</p>
	<p>Infraestructura: Área que comprende el desarrollo de sistemas operativos, protocolos de redes, sistemas administradores de bases de datos, etc. Incrementalmente este sector direccionará sus soluciones, hacia problemas genéricos de procesamiento distribuido y procesamiento de transacciones, a soluciones middleware. Firmas representativas son Microsoft, Oracle, SyBase, Novell y NeXT.</p>

Fuente: (Propia, 2020)

Modelos de COCOMO II

Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información. Estos tres modelos son:

Modelo I: Nivel inicial de prototipado - Modelo Composición de Aplicación

El esfuerzo necesario para concretar un proyecto de desarrollo de software, cualquiera sea el modelo empleado, se expresa en meses/persona (PM) y representa los meses de trabajo de una persona fulltime, requeridos para desarrollar el proyecto.

Estimación del Esfuerzo

Estimaciones realizadas con puntos de objeto y una fórmula simple para el cálculo del esfuerzo

Soporta proyectos con prototipado y proyectos que hacen uso intensivo de la reutilización.

Basado en estimaciones estándar de la productividad del desarrollador en puntos objeto/mes.

Tiene en cuenta el uso de herramientas CASE.

La fórmula es:

$$PM = (NOP * (1 - \%reuse/100)) / PROD$$

NOP (Nuevos Puntos Objeto): Tamaño del nuevo software a desarrollar expresado en Puntos Objeto y se calcula de la siguiente manera:

%reuso: Porcentaje de reusó que se espera lograr en el proyecto

PROD: Es la productividad promedio determinada a partir del análisis de datos de proyectos en [Banker 1994], mostrada en.

Tabla 3 Modelo Composición de Aplicación

Experiencia y Capacidad de los Desarrolladores	Muy Bajo	Bajo	Normal	Alto	Muy Alto
Madurez y Capacidad del ICASE	Muy Bajo	Bajo	Normal	Alto	Muy Alto
PROD	4	7	13	25	50

Fuente: (Barry Boehm, 1995)

Modelo II: Modelo para Diseño Temprano (EDM)

Este modelo se usa en las etapas tempranas de un proyecto de software, cuando se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o detalles específicos del proceso a utilizar. Este modelo podría emplearse tanto en productos desarrollados en sectores de Generadores de Aplicación, Sistemas Integrados o Infraestructura.

El modelo de Diseño Temprano ajusta el esfuerzo nominal usando siete factores de costo. La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM \text{ nominal} = PM \text{ nominal}$$

$$PM \text{ nominal} = A * (KSLOC)^B$$

$$B = 1.01 + 0.01 * \sum_{j=1}^5 W_j$$

Donde

PM Estimado: es el esfuerzo nominal ajustado por 7 factores, que reflejan otros aspectos propios del proyecto que afectan al esfuerzo necesario para la ejecución del mismo.

KSLOC: es el tamaño del software a desarrollar expresado en miles de líneas de código fuente.

A es una constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño, ($A=2.94$).

B es el factor exponencial de escala, toma en cuenta las características relacionadas con las economías y deseconomías de escala producidas cuando un proyecto de software incrementa su tamaño.

EMi corresponde a los factores de costo que tienen un efecto multiplicativo sobre el esfuerzo, llamados Multiplicadores de Esfuerzo (Effort Multipliers). Cada factor se puede clasificar en seis niveles diferentes que expresan el impacto del multiplicador sobre el esfuerzo de desarrollo. Esta escala varía desde un nivel Extra Bajo hasta un nivel Extra Alto. Cada nivel tiene un peso Asociado. El peso promedio o nominal es 1.0. Si el factor provoca un efecto nocivo en el esfuerzo de un proyecto, el valor del multiplicador correspondiente será mayor que 1.0, caso contrario el multiplicador será inferior a 1.0.

Los multiplicadores reflejan la capacidad de los desarrolladores, requerimientos no funcionales, la familiaridad con la plataforma de desarrollo, etc.

Tabla 4 Requerimientos no Funcionales

	Abrev.	Descripción	Ponderación
1	RCPX	Fiabilidad de producto y complejidad	0 a 1
2	RUSE	Reutilización requerida	0 a 1
3	PDIF	Dificultad de la plataforma	0 a 1
4	PREX	Experiencia del personal	0 a 1
5	PERS	Capacidad del personal	0 a 1
6	SCED	Agenda requerida	0 a 1
7	FCIL	Facilidad de soporte de grupo	0 a 1

Fuente: Propia

Modelo III: Nivel Post-Arquitectura.

Es el modelo de estimación más detallado y se aplica cuando la arquitectura del proyecto está completamente definida. Este modelo se aplica durante el desarrollo y mantenimiento de productos de software incluidos en las áreas de Sistemas Integrados, Infraestructura y

Generadores de Aplicaciones.

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente:

Ilustración 37 Concepto Operacional Cocomo



Fuente: (UTEA, 2011)

Tabla 5 Estimación de Esfuerzo

Personas Mes Nominales	
$PM = A \text{ Tamaño}^B * EM_j$	(A = 2.94)

Fuente: (Propia, 2020)

$B < 1$. Los esfuerzos de desarrollo mejoran cuando escalan. Si se dobla el tamaño, el esfuerzo es menor del doble.

$B = 1$. Los proyectos están balanceados. Los aumentos son proporcionales.

$B > 1$. Los esfuerzos de desarrollo empeoran cuando escalan. Si se dobla el tamaño, el esfuerzo es menor del doble.

Tabla 6 Factor Escala

FACTOR ESCALA

$$B = 0,91 + 0,01 \times \sum W_i$$

2.10. HERRAMIENTAS

2.10.1. FRAMEWORK YII

Framework Yii de PHP, este es un Framework de alto rendimiento escrito en PHP y diseñado para crear robustas aplicaciones web.

Ilustración 38 Yii Framework



Fuente: (Yii, s.f.)

En mis propias palabras yo lo definiría como un conjunto de herramientas que nos ayudará a crear una aplicación de una forma más arquitectónica, es decir, sabemos por definición que en el mundo de la programación un Framework por sí mismo son un conjunto de componentes que nos ayudan a nosotros como desarrolladores a crear una aplicación en un lenguaje de programación en particular, bueno pues Yii nos permite hacerlo de una manera más sencilla y profesional utilizando estándares de desarrollo; como patrones de diseño, POO, y todas esas cosas.

Si estás familiarizado con otros framework, puedes apreciar cómo se compara Yii con ellos:

Como la mayoría de los framework de PHP, Yii implementa el patrón de diseño MVC (Modelo-Vista-Controlador) y promueve la organización de código basada en este patrón.

La filosofía de Yii consiste en escribir el código de manera simple y elegante, sin sobrediseñar nunca por el mero hecho de seguir un patrón de diseño determinado.

Yii es un framework completo que provee muchas características probadas y listas para usar, como los constructores de consultas y la clase ActiveRecord para las bases de datos relacionales y NoSQL, la compatibilidad con la arquitectura REST para desarrollar API, la compatibilidad de caché en varios niveles y muchas más.

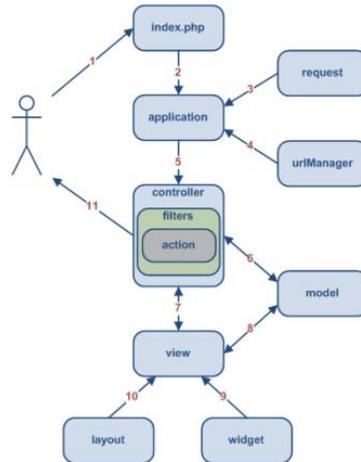
Yii es extremadamente extensible. Puedes personalizar o reemplazar prácticamente cualquier pieza de código de base, como se puede también aprovechar su sólida arquitectura de extensiones para utilizar o desarrollar extensiones distribuibles.

El alto rendimiento es siempre la meta principal de Yii.

Yii no es un proyecto de una sola persona, detrás de Yii hay un sólido equipo de desarrollo, así como una gran comunidad en la que numerosos profesionales contribuyen constantemente a su desarrollo. El equipo de desarrollo de Yii se mantiene atento a las últimas tendencias de desarrollo web, así como a las mejores prácticas y características de otros frameworks y proyectos. Las buenas prácticas y características más relevantes de otros proyectos se incorporan regularmente a la base del framework y se exponen a través de interfaces simples y elegantes.

CARACTERÍSTICAS DE YII

Ilustración 39 Flujo de Tareas Yii



Fuente: (Yii, s.f.)

Usa el patrón de Arquitectura de Software MVC (Modelo Vista Controlador) que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Provee un Nivel de Abstracción hacia la Base de Datos (DAO, Data Access Object) y se apoya en el patrón de arquitectura de datos Active Record.

Incluye soporte para autenticación de usuarios basado en el modelo RBAC role-based access control o control de acceso basado en roles.

Herramientas para automatización y validación de Formularios

Integración con jQuery.

Avanzado control de Themes

Web services, herramientas para simplificar la tarea de generar y consumir servicios Web en diferentes formatos.

Internacionalización y Localización.

Sistema de manejo de cache.

Manejo de Errores.

Manejo de Seguridad (Sql injection,XSS, CSRF, cookie tampering, etc.).

Generador de código automático, muy útil a la hora de implementar formularios o bien operaciones CRUD. Además de que el código generado cumple XHTML standards.

Amigable y extensible, por lo cual se pueden usar sin problemas librerías de terceros como Pear, Zend, etc.

El manejo de errores y logging. Los errores son manejados y personalizados, y los logs de mensajes pueden ser categorizados, filtrados y movidos a diferentes destinos.

Las medidas de seguridad incluyen la prevención cross-site scripting (XSS), prevención cross-site request forgery (CSRF), prevención de la manipulación de cookies, etc.

Herramientas para pruebas unitarias y funcionales basados en PHPUnit y Selenium.

Generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD, etc.

Generación de código por componentes de Yii y la herramienta por línea de comandos cumple con los estándares de XHTML.

Cuidadosamente diseñado para trabajar bien con código de terceros. Por ejemplo, es posible usar el código de PHP o Zend Framework en una aplicación Yii. (Yii Framework, 2020).

Versión actual de Yii

A la fecha de estar escribiendo esta introducción, la versión más reciente de Yii es la 2.0.6 y no es compatible hacia atrás con la versión 1. Si deseas puedes visitar el sitio web oficial de Yii Framework para verificar cual es la versión más actualizada.

Pero en el caso de nuestro software del Seguimiento de proyectos escolares utilizaremos la versión 1.1. para su desarrollo e implementación.

2.10.2.PHP

Para explicar qué es PHP primero debemos mostrar qué significa sus siglas: Hypertext Pre-Processor. Ahora ya podemos decir que tras este nombre se esconde un lenguaje de programación de código abierto muy utilizado en el desarrollo web y que puede ser incrustado en HTML.

Los orígenes de este lenguaje de programación se remontan al año 1994 cuando Rasmus Lerdorf decidió escribir un pequeño CGI en Perl que le ayuda a contabilizar las visitas que llegaban a su curriculum vitae.

Su crecimiento desde entonces se ha debido fundamentalmente a los miembros que forman parte de la comunidad PHP, que han ido desarrollando numerosas versiones de este lenguaje de programación.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

Ilustración 40 Estructura PHP



Fuente: (Google, s.f.)

Funcionamiento

Enviamos una petición al servidor, ejemplo `www.sistemasUpea /blog/index.php`

El servidor recibe la petición y busca la página a entregar.

Si la página contiene la extensión «.php», el intérprete de PHP la procesa.

El servidor ejecuta el código PHP de la página y prepara el resultado final, el HTML.

Se envía la página HTML al cliente final.

Ventajas

Lenguaje multiplataforma.

Fácil de aprender.

Orientado para desarrollar aplicaciones web donde la información esté en una base de datos.

Buena integración con la mayoría de conectores a base de datos. MySQL, PostgreSQL, Oracle, etc.

Lenguaje modular.

Mucha documentación debido a su gran popularidad y una gran comunidad. (Web Oficial php.net).

Programación orientada a objetos.

Lenguaje de código libre y gratuito.

Biblioteca muy amplia de funciones nativas.

Múltiples Frameworks PHP que permiten que tu código sea más ordenado, estructurado, manejable y permiten trabajar utilizando patrones de diseño como Modelo-Vista-Controlador (MVC).

Desventajas

Se necesita instalar un servidor web.

Se realiza todo el trabajo en la parte del servidor, por esto, si se tienen muchas peticiones, el rendimiento de nuestra aplicación podría verse afectado sensiblemente.

Al mezclar HTML + PHP, dificulta la legibilidad de nuestro código.

Seguridad. Como es un lenguaje de código abierto, todas las personas pueden ver el código fuente, y si hay errores, la gente puede utilizar estas debilidades de codificación.

Es un lenguaje de programación NO tipado.

Difícil de mantener.

2.10.3. MYSQL

Primero lo primero, tienes que saber cómo pronunciarlo: MY-ES-KYU-EL' [maɪ,ɛs,kju:'ɛl]. Una compañía sueca llamada MySQL AB originalmente desarrolló MySQL en 1994. La compañía de tecnología de los Estados Unidos Sun Microsystems luego tomó el control por completo cuando compró MySQL AB en el 2008. El gigante de la tecnología estadounidense Oracle adquirió Sun Microsystems en el 2010, y MySQL ha sido propiedad de Oracle desde entonces.

En cuanto a la definición general, MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional. Ahora, echemos un vistazo más de cerca a cada término:

Ilustración 41 Gestor de Base de Datos Mysql



Fuente: (Mysql © 2020, s.f.)

Base de datos

Una base de datos es simplemente una colección de datos estructurados. Imagina que te tomas una selfie: presionas un botón y capturas una imagen de ti mismo. Tu foto es información y la galería de tu teléfono es la base de datos. Una base de datos es un lugar en el que los datos son almacenados y organizados. La palabra «relacional» significa que los datos almacenados en el conjunto de datos son organizados en forma de tablas. Cada tabla se relaciona de alguna manera. Si el software no es compatible con el modelo de datos relacionales, simplemente se llama DBMS.

Código abierto

Código abierto significa que eres libre de usarlo y modificarlo. Cualquiera puede instalar el software. También puedes aprender y personalizar el código fuente para que se adapte mejor a tus necesidades. Sin embargo, la GPL (licencia pública de GNU) determina lo que puedes hacer según las condiciones. La versión con licencia comercial está disponible si necesitas una propiedad más flexible y un soporte avanzado.

Modelo cliente-servidor

Las computadoras que tienen instalado y ejecutan el software RDBMS se llaman clientes. Siempre que necesitan acceder a los datos, se conectan al servidor RDBMS. Esa es la parte «cliente-servidor».

MySQL es una de las muchas opciones de software RDBMS. Suele pensarse que RDBMS y MySQL son lo mismo debido a la popularidad de MySQL. Para nombrar algunas aplicaciones web grandes como Facebook, Twitter, YouTube, Google y Yahoo!, todas usan MySQL para el almacenamiento de datos. Aunque inicialmente se creó para un uso limitado, ahora es compatible con muchas plataformas de computación importantes como Linux, macOS, Microsoft Windows y Ubuntu.

MySQL y SQL no son lo mismo. Ten en cuenta que MySQL es una de las marcas más populares de software RDBMS, que implementa un modelo cliente-servidor. Entonces, ¿cómo se comunican el cliente y el servidor en un entorno RDBMS? Utilizan un lenguaje

específico del dominio: lenguaje de consulta estructurado (SQL, Structured Query Language). Cuando veas otros nombres que contienen SQL, como PostgreSQL y el servidor Microsoft SQL, es muy probable que sean marcas que también utilizan la sintaxis SQL. El software RDBMS a menudo se escribe en otros lenguajes de programación, pero siempre usa SQL como lenguaje principal para interactuar con la base de datos. MySQL como tal está escrito en C y C ++. Pasa como con los países sudamericanos, todos son geográficamente diferentes y tienen historias diferentes, pero todos hablan principalmente español.

El científico informático Ted Codd desarrolló SQL a principios de la década de 1970 con un modelo relacional basado en IBM. Se volvió más común en 1974 y reemplazó rápidamente a lenguajes similares, por entonces obsoletos, ISAM y VISAM.

Dejando de lado la historia, SQL le dice al servidor qué hacer con los datos. Es similar a tu contraseña o código de WordPress. Lo ingresas en el sistema para obtener acceso al área del panel de control. En este caso, las declaraciones de SQL pueden indicarle al servidor que realice ciertas operaciones:

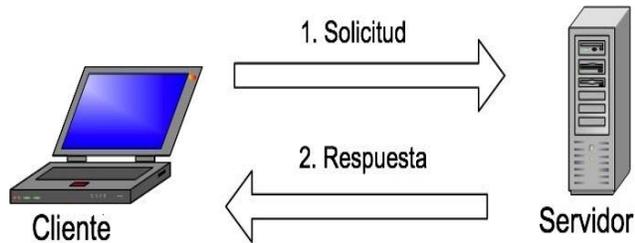
Consulta de datos: solicitar información específica de la base de datos existente.

Manipulación de datos: agregar, eliminar, cambiar, ordenar y otras operaciones para modificar los datos, los valores o los elementos visuales.

Identidad de datos: definir tipos de datos, por ejemplo, cambiar datos numéricos a números enteros. Esto también incluye la definición de un esquema o la relación de cada tabla en la base de datos.

Control de acceso a los datos: proporcionar técnicas de seguridad para proteger los datos, lo que incluye decidir quién puede ver o usar cualquier información almacenada en la base de datos.

Ilustración 42 Funcionamiento de Mysql Servidor – Cliente



Fuente: (Google, s.f.)

La imagen explica la estructura básica cliente-servidor. Uno o más dispositivos (clientes) se conectan a un servidor a través de una red específica. Cada cliente puede realizar una solicitud desde la interfaz gráfica de usuario (GUI) en sus pantallas, y el servidor producirá el output deseado, siempre que ambas partes entiendan la instrucción. Sin meternos demasiado a fondo en temas técnicos, los procesos principales que tienen lugar en un entorno MySQL son los mismos, y son:

MySQL crea una base de datos para almacenar y manipular datos, definiendo la relación de cada tabla.

Los clientes pueden realizar solicitudes escribiendo instrucciones SQL específicas en MySQL.

La aplicación del servidor responderá con la información solicitada y esta aparecerá frente a los clientes.

Y eso es prácticamente todo. Desde el lado de los clientes, generalmente enfatizan qué GUI de MySQL usar. Cuanto más ligera y fácil de usar sea la GUI, más rápidas y fáciles serán sus actividades de administración de datos. Algunas de las GUI de MySQL más populares son MySQL WorkBench, SequelPro, DBVisualizer y Navicat DB Admin Tool. Algunas de ellas son gratuitas, mientras que otras son comerciales, otras son exclusivamente para macOS y otras son compatibles con los principales sistemas operativos. Los clientes deben elegir la GUI en función de sus necesidades. Para la administración de bases de datos web, incluido un sitio de WordPress, la opción más obvia es phpMyAdmin.

Flexible y fácil de usar

Puedes modificar el código fuente para satisfacer tus propias expectativas, y no tienes que pagar nada por este nivel de libertad, incluidas las opciones de actualización a la versión comercial avanzada. El proceso de instalación es relativamente simple y no debería durar más de 30 minutos.

Alto rendimiento

Un amplio compendio de servidores de clúster respalda a MySQL. Ya sea que estés almacenando enormes cantidades de datos de e-Commerce grandes o realizando actividades intensas de inteligencia de negocios, MySQL puede ayudarte sin problemas con una velocidad óptima.

Un estándar de la industria

Las industrias han estado usando MySQL durante años, lo que significa que hay abundantes recursos para desarrolladores calificados. Los usuarios de MySQL pueden esperar un rápido desarrollo del software y trabajadores freelance expertos dispuestos a trabajar.

Seguro

Tus datos deberían ser la principal preocupación al elegir el software RDBMS correcto. Con su sistema de privilegios de acceso y la administración de cuentas de usuario, MySQL establece un alto estándar de seguridad. La verificación basada en el host y el cifrado de contraseña están disponibles.

2.10.4. XAMPP

XAMPP es un servidor independiente de plataforma de código libre. Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito.

XAMPP incluye además servidores de bases de datos como MySQL y SQLite con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete

de PHP, el intérprete de Perl, servidores de FTP como ProFTPD ó FileZilla FTP Serve, etc. entre muchas cosas más.

Si alguna vez has intentado instalar Apache, sabes que no es una tarea fácil, sin embargo, con XAMPP todo es diferente. Una de las ventajas de usar XAMPP es que su instalación es de lo más sencilla, basta descargarlo, extraerlo y comenzar a usarlo.

Ilustración 43 Servidor Local Xampp



Fuente: (Apache Friends Copyright (c) 2020, s.f.)

XAMPP es una herramienta de desarrollo que te permite probar tu trabajo (páginas web o programación, por ejemplo) en tu propio ordenador sin necesidad de tener acceso a internet.

Si eres un desarrollador que recién está comenzando, tampoco debes preocuparte sobre las configuraciones ya que XAMPP te provee de una configuración totalmente funcional desde el momento que lo instalas (básicamente lo extraes). Sin embargo, es bueno acotar que la seguridad de datos no es su punto fuerte, por lo cual no es suficientemente seguro para ambientes grandes o de producción.

Paquetes que vienen con XAMPP:

Apache, el servidor Web más famoso.

MySQL, una excelente base de datos de código libre.

PHP y Perl: lenguajes de programación.

ProFTPD: un servidor FTP.

OpenSSL: para soporte a la capa de sockets segura.

Paquetes gráficos:

GD (Graphics Draw): la librería de dibujo de gráficos.

libpng: la librería oficial de referencia de PNG.

libjpeg: la librería oficial de referencia de JPEG.

ncurses: la librería de gráficos de caracteres.

Paquete de bases de datos:

gdbm: la implementación GNU de la librería standard dbm de UNIX.

SQLite: un motor de base de datos SQL muy pequeño y cero configuraciones.

FreeTDS: una librería de base de datos que da a los programas de Linux y UNIX la habilidad de comunicarse con Microsoft SQL y Sybase.

Paquetes XML:

expat: una librería parser de XML.

Salbotron: una toolkit de XML.

libxml: un parser C de XML y un toolkit para GNOME.

Paquetes PHP:

PEAR: la librería de PHP.

Una clase pdf que genera documentos PDF dinámicos con PHP.

TURCK MMCache: un potenciador de la performance de PHP.

Otros paquetes:

zlib: una librería de compresión.

mod_perl: empotra un intérprete de Perl en Apache.

gettext: un conjunto de herramientas que asiste a los paquetes GNU para producir mensajes multilingües.

mcrypt: un programa de encriptación.

Ming: una librería de salida en Flash.

IMAC C-Client: un API de correos

Seguridad

Oficialmente, los diseñadores de XAMPP fueron los "Baiker y Anthony Corporation", quienes sólo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores cuando no tienen ningún acceso a Internet. En la práctica, sin embargo, XAMPP se utiliza actualmente como servidor de sitios web, ya que, con algunas modificaciones, es

generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes en una página.

2.10.5. MAGICDRAW

MagicDraw, herramienta CASE desarrollada por No Magic. Es compatible con el estándar UML 2.3, desarrollo de código para diversos lenguajes de programación (Java, C++ y C#, entre otros) así como para modelar datos. Cuenta con capacidad para trabajar en equipo y es compatible con varios entornos de desarrollo (IDEs).

Ilustración 44 MagicDraw



Fuente: (2000-2017 No Magic, s.f.)

Características de MagicDraw

Compatible con los siguientes IDEs:

Sun Java Studio 8.

Borland CaliberRM 6.0, 6.5 herramienta de requisitos.

Oracle Workshop 8.1.2.

E2E Bridge 4.0

IntelliJ IDEA 4.X or later.

NetBeans 6.X or later.

Eclipse 3.1 o superior (versión Java)

IBM Rational Application Developer

Borland JBuilder 8.0, 9.0, X, 2005, 2006, 2007

Built-in CVS interfaz para almacenar archivos de proyectos.

Integración con herramientas MDA: Compuware' OptimalJ, AndromDA, Interactive Objects' ArcStyler, openArchitectureWare, E2E Bridge, y Mia-Software Tools.

Diseñada para los analistas del negocio, los analistas del software, los programadores, los ingenieros del QA, y los escritores de la documentación.

Facilita el análisis y el diseño de los sistemas (OO) y de las bases de datos orientados objeto.



CAPÍTULO

III



MARCO

APLICATIVO

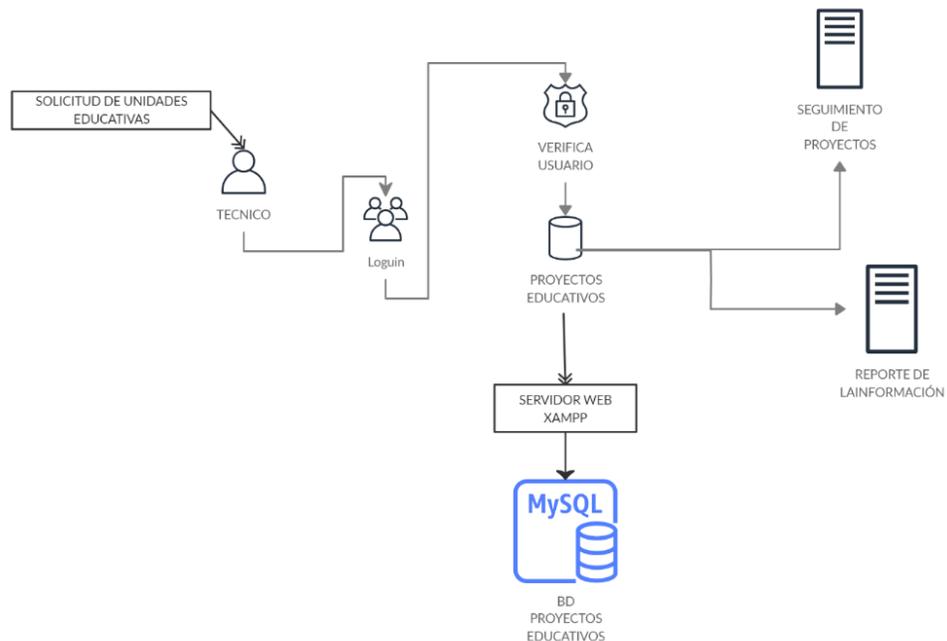
3.MARCO APLICATIVO

3.1. INTRODUCCIÓN

En el desarrollo del presente proyecto Sistema de Seguimiento de Proyectos Educativos en el Gobierno Autónomo Municipal de Puerto Pérez, En este capítulo se hará énfasis a todo lo que significa el proceso de desarrollo de software; desde los requerimientos del usuario hasta la prueba final del sistema. Cada una de las actividades que son necesarias para la transformación de los requisitos del usuario en un sistema software, serán detalladas de acuerdo a la metodología utilizada (UWE).

Aplicando las diversas fases del modelo y desarrollo del modelo UWE (Ingeniería Web basado en UML) y las fases de esta metodología son: Análisis de Requerimientos, Diseño Conceptual, Diseño Navegación y Diseño de Presentación que nos representan diversos diagramas y esquemas en un proceso iterativo e incremental dando apoyo al modelo de la aplicación.

Ilustración 45 Esquema del Sistema



Fuente: (Propia, 2020)

3.2. PROCESO DE DESARROLLO DE LA METODOLOGÍA UWE

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo de navegación y el modelo de presentación. A estos modelos se pueden sumar otros modelos como son el modelo de usuario, modelo de adaptación y modelo de tareas para representar los aspectos dinámicos de la aplicación mediante la descripción de situaciones. De esta manera se obtiene una colección de modelos y diagramas que describen una aplicación web de manera integral (Pérez, 2010).

En su implementación se deben contemplar las siguientes etapas y modelos:

3.2.1. ANÁLISIS DE REQUERIMIENTOS

Plasma los requisitos funcionales de la web mediante un modelo de casos de uso.

Identificación de Actores

Es necesario conocer a los actores que siguen los procesos para la funcionalidad y además las actividades que realizan en el sistema.

Unidad Solicitante: Son los beneficiarios directos de la ejecución de los proyectos. Representados por las juntas escolares, por ejemplo.

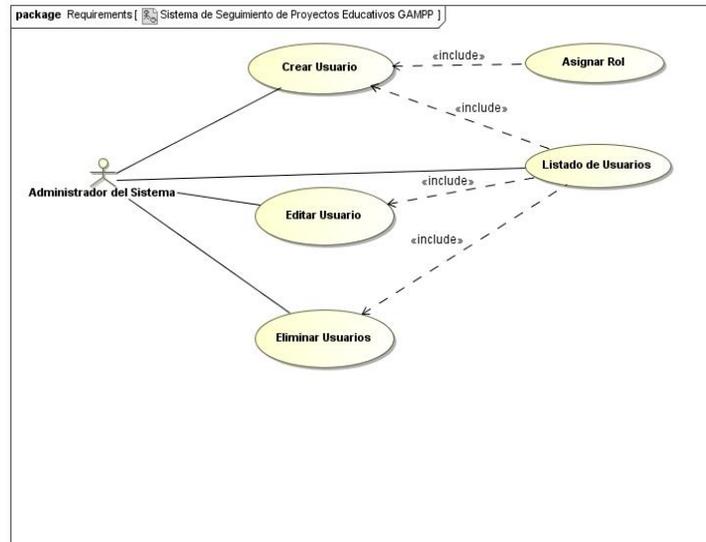
Concejo Municipal: Son las personas que se encarga de fiscalizar al Gobierno Autónomo Municipal de Puerto Pérez – GAMPP. previa inscripción de los proyectos por las juntas escolares.

Técnico (GAMPP): Este caso de uso hace referencia a las unidades del GAMPP. por la que atraviesa el proyecto antes de llegar al Consejo Municipal de El Alto para su respectiva aprobación.

Técnico en computación: Es la persona encargada de centralizar la información referente al seguimiento realizado en cada unidad de los proyectos educativos, para luego emitir reportes.

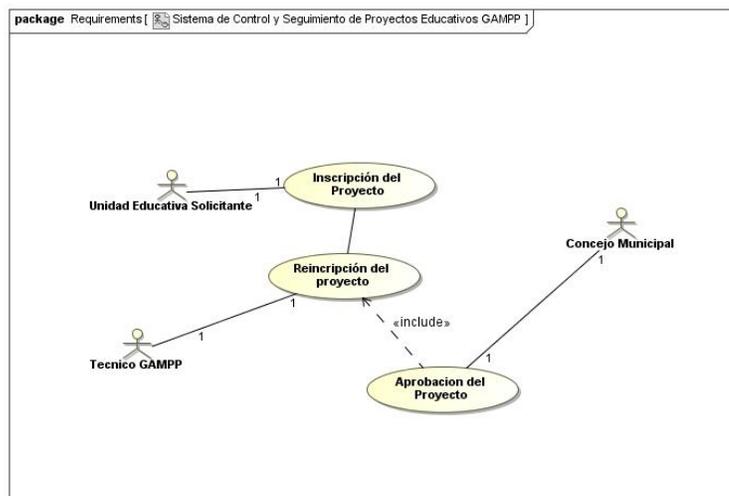
Casos de Uso

Ilustración 46 Caso de Uso: Administrar Usuario



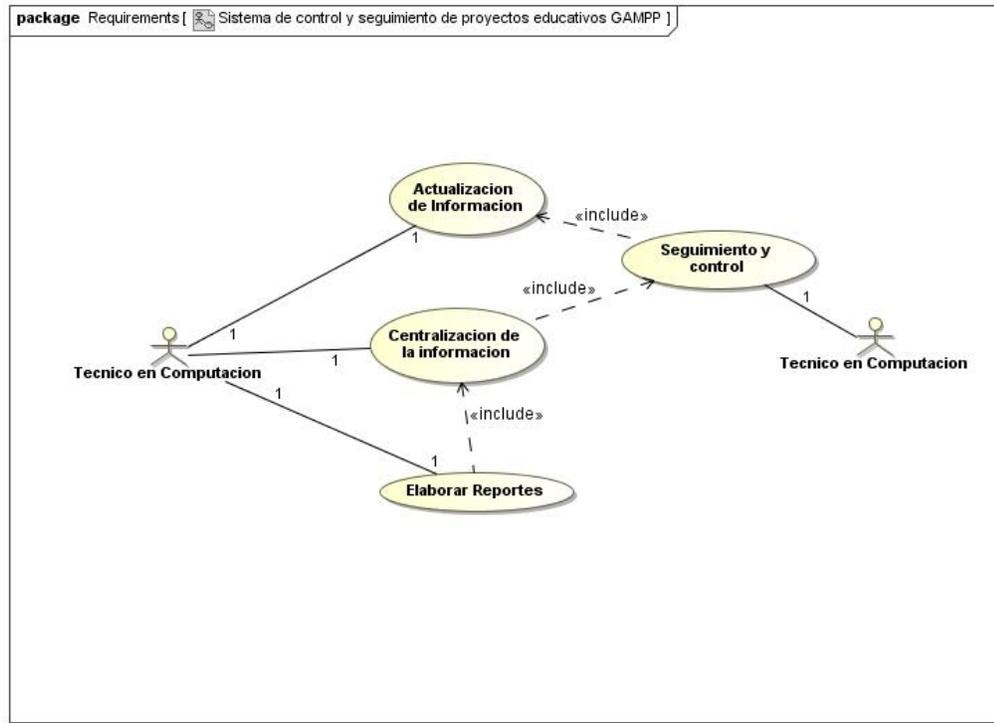
Fuente: (Propia, 2020)

Ilustración 47 Caso de Uso: Solicitud de Proyectos Educativos



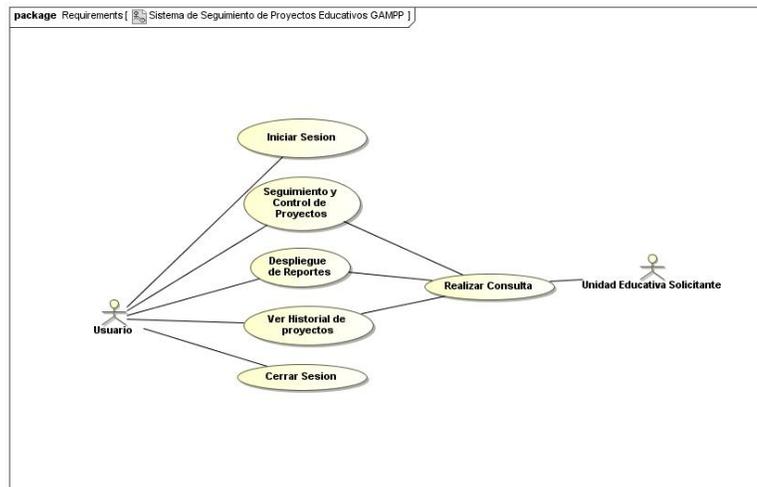
Fuente: (Propia, 2020)

Ilustración 48 Caso de uso: Control de proyectos Educativos



Fuente: (Propia, 2020)

Ilustración 49 Caso de uso: Seguimiento de proyectos educativos



Fuente: (Propia, 2020)

Tabla 7 Caso de Uso - Administrador del Sistema

CASO DE USO	Administrador de Usuario
ACTOR	Técnico Administrador
DESCRIPCIÓN	Ingresar al Sistema
CURSO NORMAL DE EVENTOS	
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el usuario desea acceder al sistema.	2. Petición de identificador y contraseña del usuario.
3. Registra su correspondiente identificador y contraseña.	5. Despliegue de la pantalla principal del sistema de acuerdo a los privilegios que tiene el usuario.
4. Presiona el botón aceptar.	6. Fin.

Fuente: (Propia, 2020)

Tabla 8 Caso de Uso - Solicitud de Proyecto

CASO DE USO	Solicitud de Proyecto
ACTOR	Unidad Solicitante
DESCRIPCIÓN	Inscripción
CURSO NORMAL DE EVENTOS	
Acción del actor	Respuesta del sistema

1. El caso de uso inicia cuando la unidad educativa solicitante presenta un proyecto. 2. Petición que es recibida dentro del GAMPP para su evaluación. 3. Registra el técnico de la GAMPP para que sea tratada dentro del concejo municipal. 4. El concejo Municipal realiza la aprobación del proyecto sin pasar el monto del POA. 5. Realiza el análisis del proyecto de acuerdo al POA anual que tiene para ejecutarse cada unidad educativa. 6. Se reinscribe el proyecto para su ejecución. 7. Fin.

Fuente: (Propia, 2020)

Tabla 9 Control de Proyectos

CASO DE USO	Control de Proyectos
ACTOR	Técnico de Computación
DESCRIPCIÓN	Seguimiento
CURSO NORMAL DE EVENTOS	
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el proyecto ya está inscrito y aprobado dentro del GAMPP.	2. Petición que el técnico debe hacer el seguimiento del estado del proyecto.
3. Registra los movimientos. Así centralizar todo referente al Proyecto.	4. Realiza la elaboración del reporte de todos los proyectos ejecutados y en ejecución.
	5. Fin.

Fuente: (Propia, 2020)

Tabla 10 Caso de uso - Seguimiento de proyectos

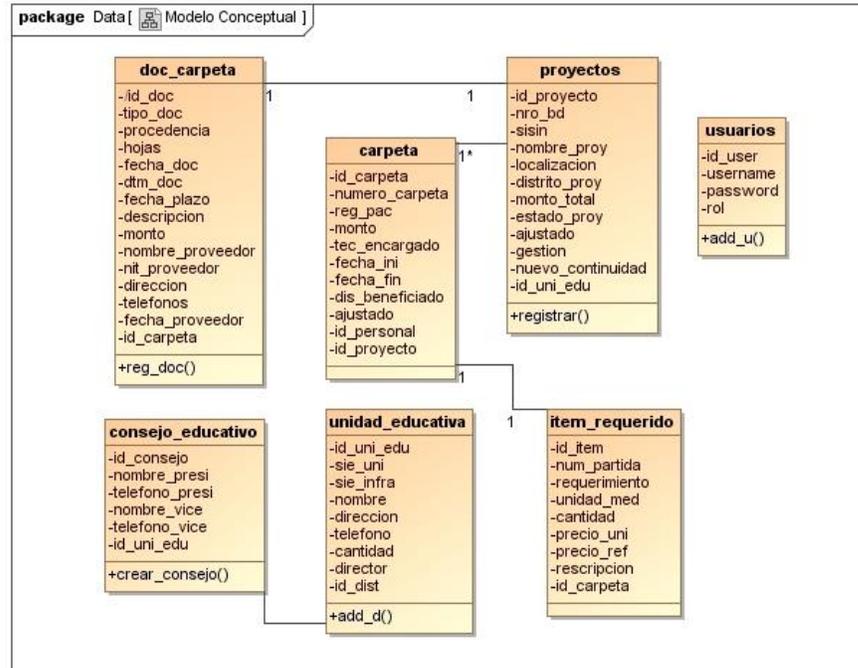
CASO DE USO	Seguimiento de Proyectos
ACTOR	Técnico de GAMPP
DESCRIPCIÓN	Seguimiento
CURSO NORMAL DE EVENTOS	
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando la unidad del educativa solicitante realiza la consulta del estado de su proyecto.	2. El técnico revisa en el sistema el estado del proyecto educativo. 3. Genera un reporte del estado actual del proyecto. 4. Fin.

Fuente: (Propia, 2020)

3.2.2. MODELO DE CONTENIDOS

El modelo de contenido es el modelo conceptual del dominio de aplicación tomando en cuenta los requerimientos especificados en los casos de uso y se representa con un diagrama de clases.

Ilustración 50 Modelo Conceptual



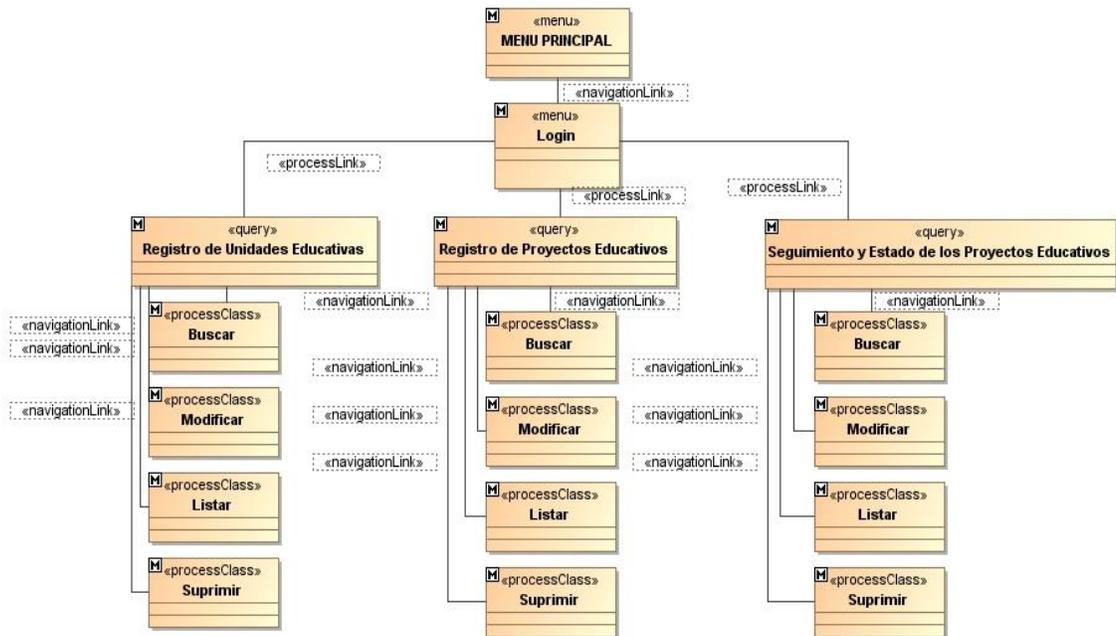
Fuente: (Propia, 2020)

Este modelo especifica cómo se encuentran relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sitio web.

3.2.3. MODELO NAVEGACIONAL

Según (Nolivos Coronel, 2013) el modelo de navegación de una aplicación web comprende la especificación de qué objetos pueden ser visitados mediante la navegación, a través del sistema y las asociaciones entre ellos. Mediante estos diagramas se representa el diseño y la estructura de las rutas de navegación al usuario para evitar la desorientación en el proceso de navegación. Los elementos básicos en el modelo de navegación son los nodos y enlaces como se puede apreciar a continuación.

Ilustración 51 Modelo Navegacional



Fuente: (Propia, 2020)

En un sistema para la web es útil saber cómo están enlazadas las páginas. Esto significa que necesitamos un diagrama con nodos (nodes) y enlaces (links). Los nodos son unidades de navegación y están conectados por medio de enlaces, pueden ser presentados en diferentes páginas o en una misma página.

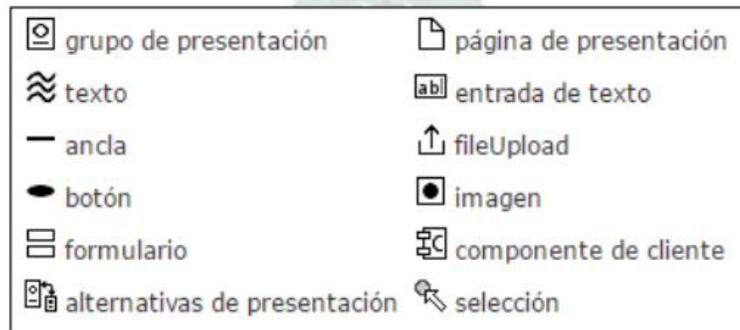
Para los nodos y enlaces los estereotipos usados «navigationClass» en caso de que el enlace implique ejecución de código y «navigationLink» cuando no implica ejecución de código.

3.2.4. MODELO DE PRESENTACIÓN

El modelo de presentación en UWE está muy relacionado con los elementos de las interfaces definidas en HTML. Estos elementos también están definidos como estereotipos de UML.

Los elementos del modelo de presentación son: ventanas, entradas de texto, imágenes, audio y botones (Nolivos Coronel, 2013).

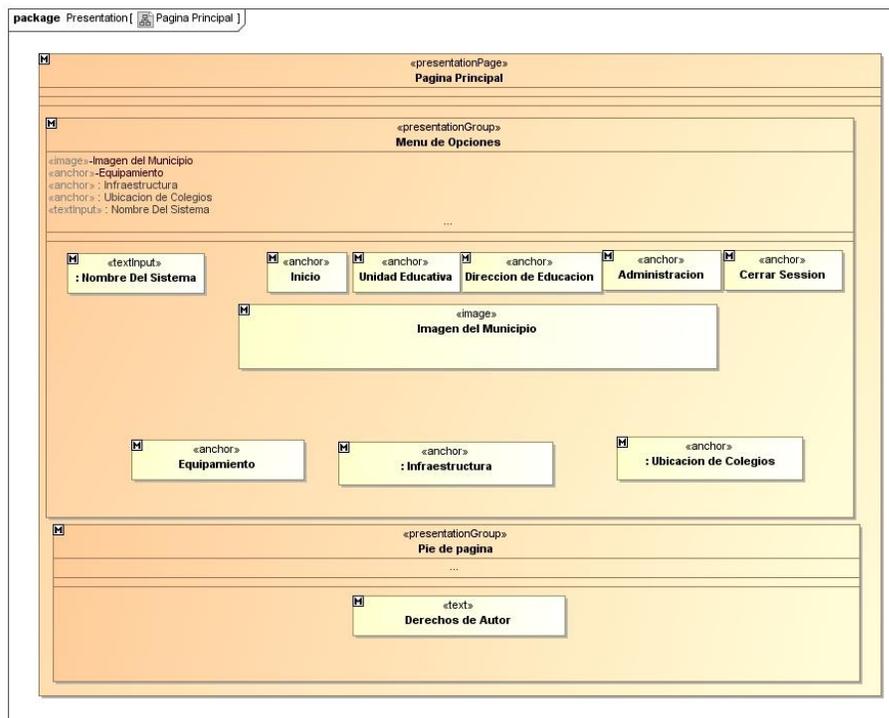
Ilustración 52 Nombre de estereotipos y sus iconos del Modelo de Presentación



Fuente: (Propia, 2020)

Como se puede ver en la figura, estos diagramas permiten especificar dónde y cómo los objetos de navegación serán presentados al usuario, una representación esquemática visible para el usuario.

Ilustración 53 Modelo de Presentación



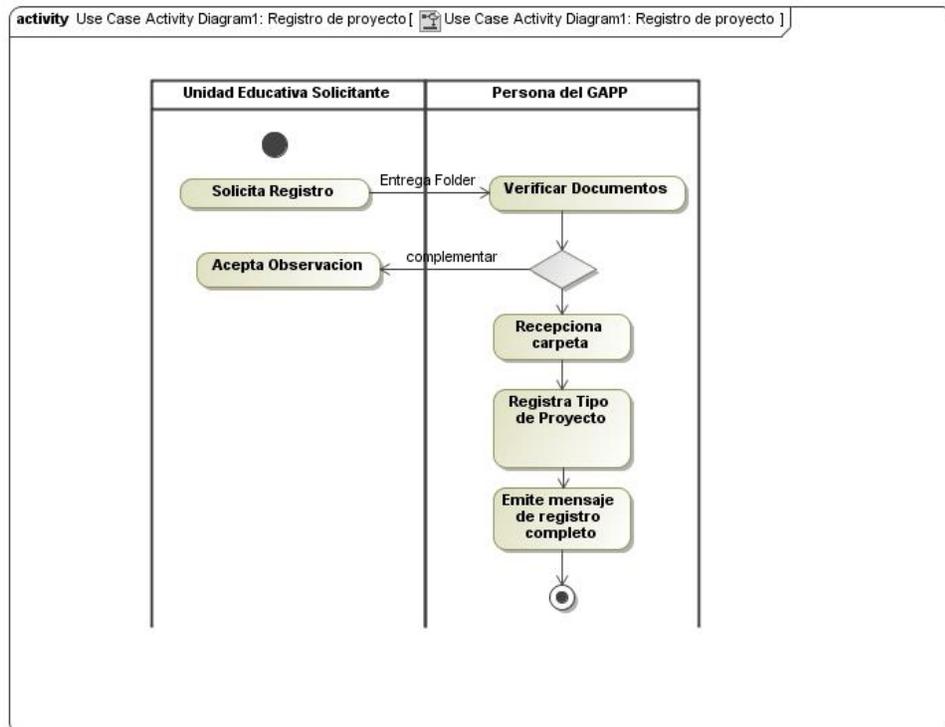
Fuente: (Propia, 2020)

3.2.5. MODELO DE PROCESO

Este modelo provee los elementos necesarios para representar procesos de negocio en un modelo UWE. Al contrario del modelo de navegación que representa la estructura estática, de la aplicación Web, el modelo de proceso representa la parte dinámica de la misma.

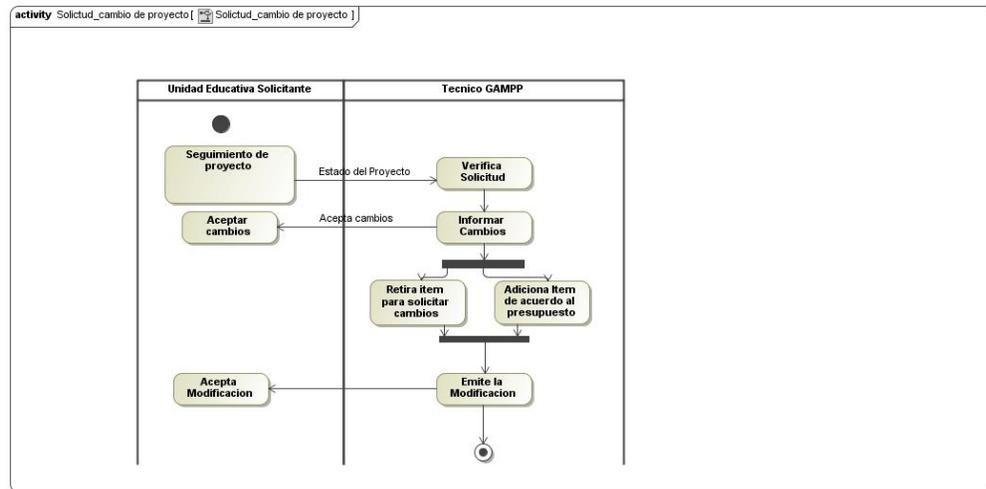
Este tipo de modelo define el comportamiento y los detalles de un proceso de negocio. El process flow, como también se lo llama, describe minuciosamente los pasos dentro de un proceso, en el caso que el usuario navegue por éste. Maneja algunos estereotipos como se observa.

Ilustración 54 Modelo De Proceso Registro de Proyecto



Fuente: (Propia, 2020)

Ilustración 55 Modelo de Procesos Seguimiento de Proyecto y Cambio



Fuente: (Propia, 2020)

Ilustración 56 Estereotipos y sus Iconos Modelo de Procesos



Fuente: (Propia, 2020)

El estereotipo «user Action» es usado para indicar interacciones de usuario con la página web iniciando un proceso o respondiendo a un requerimiento explícito de información. Por el contrario, «system Action» describe acciones, que son ejecutadas por el sistema. Ambos tipos de acciones pueden ser agregadas usando la barra de herramientas (toolbar).

3.3. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Requerimientos Funcionales

La información debe estar disponible en el sistema.

Registrar los datos de los proyectos educativos que son nuevas.

El registro de usuarios solo se realizará por parte de administración viendo los roles de cada uno.

Se deben poder realizar modificaciones en los proyectos educativos.

Los usuarios que tengan el rol de usuario, solo podrán realizar acciones de acuerdo al nivel de seguridad.

El técnico GAMPP hará el seguimiento y verificación del estado del proyecto, recopilando esos datos para su actualización.

Los reportes se realizan sobre el estado y los ítems del proyecto en proceso.

Todo dato que se registre debe estar guardado en la base de datos.

Requerimientos No Funcionales.

El Gobierno Autónomo Municipal de Puerto Pérez cuenta con equipo de computación.

El personal del GAMPP cuenta con conocimientos básicos sobre el manejo de la computadora.

Los administrativos tienen conocimientos medios con manejo de paquetes básicos de software.

3.4. MODELO VISTA CONTROLADOR MVC

EL Modelo Vista Controlador (MVC), es un patrón de diseño que separa en capas bien definidas el desarrollo de una aplicación, esas partes son tres, el Modelo encargado de la lógica del negocio y la persistencia de los datos, las Vistas son las responsables de mostrar al usuario el resultado que obtienen del modelo a través del controlador, el Controlador encargado es el encargado de gestionar las peticiones del usuario, procesarlas invocando al modelo y mostrarlas al usuario a través de las vistas (Mouse, 2011).

A continuación, se describe cada uno de los componentes citados:

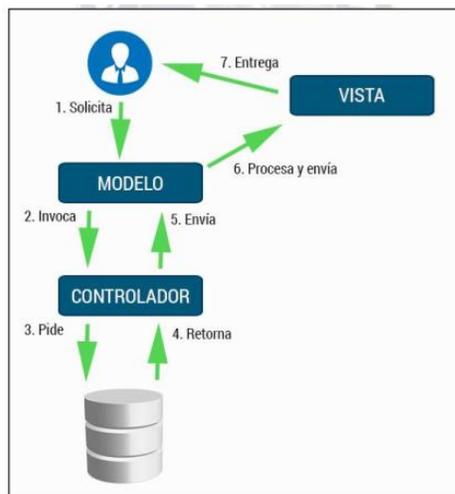
Modelo: El modelo es la representación lógica de la información y describe la funcionalidad del sistema, por lo tanto, se encarga de gestionar los permisos para ingresar a dicha información, también se encarga de realizar recuperación, actualización, inserción y eliminación de datos; a través de un conjunto de reglas (lógica de negocio).

Vista: La vista es la interfaz de usuario. Muestra al usuario una representación visual del modelo, sus datos y estado, tomándolos directamente del modelo. También contiene los elementos de la interfaz que permiten al usuario interactuar con el programa, tales como botones y menús. Sin embargo, no es tarea de la vista implementar cómo se deben comportar esos elementos.

Controlador: El controlador es el intermediario entre la vista y el controlador, responde a eventos generados por el usuario el cual constituyen llamados al Modelo (en el caso que se solicite alguna información como lista de clientes, consultar un proveedor o editar un producto) o a la Vista (en el caso de mostrar un formulario o un reporte estadístico). En pocas palabras el componente que hace posible tener separada la lógica de negocio con la vista es el Controlador.

En la figura se muestra el funcionamiento del patrón MVC, que se describe en los siguientes pasos (Pilar, 2014).

Ilustración 57 Modelo Vista Controlador



Fuente: (Propia, 2020)

1. El usuario realiza una petición que captura el controlador
2. El controlador llama al modelo correspondiente
3. El modelo solicita la información a la base de datos
4. El modelo recoge la información de la base de datos
5. El controlador recibe la información
6. El controlador procesa y envía la información a la vista
7. La vista entrega al usuario la información de forma clara y precisa.

3.5. MODELO CONCEPTUAL

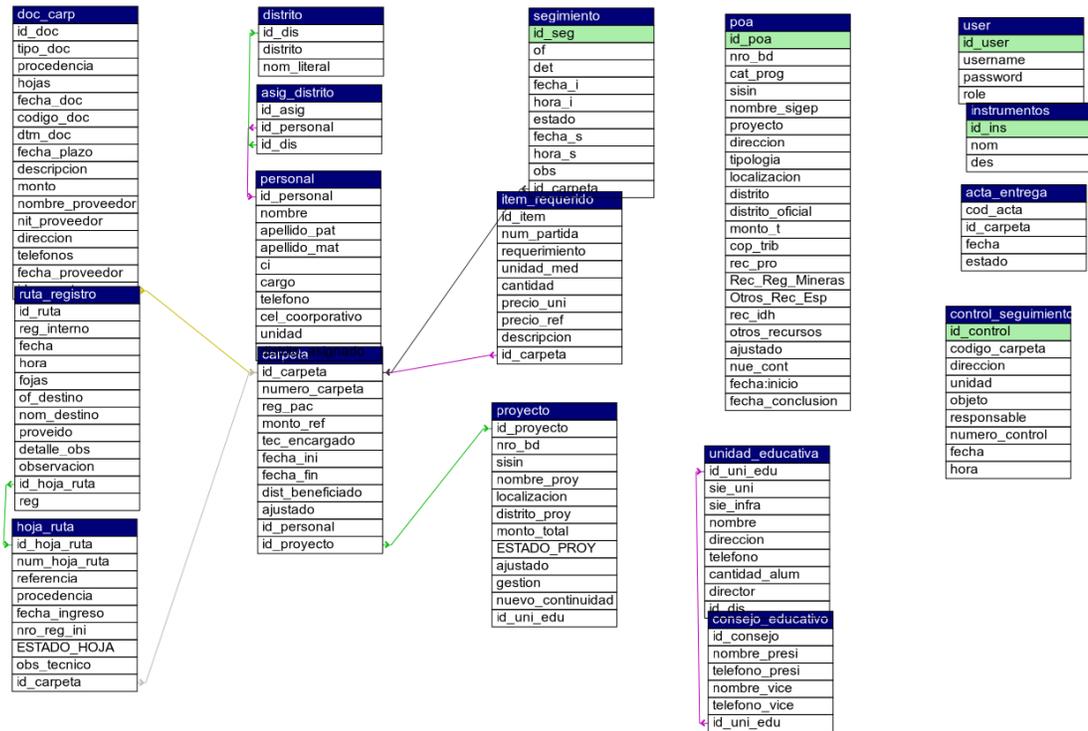
En esta etapa, se construirá un Modelo Conceptual a partir de los Indicadores y Perspectivas obtenidas en el paso anterior.

Un Modelo Conceptual es una descripción de alto nivel de la estructura de la base de datos, en la cual la información es representada a través de Objetos, Relaciones y Atributos.

A través de este Modelo, se podrá observar con claridad cuáles son los alcances del proyecto, para luego poder trabajar sobre ellos.

La representación gráfica del Modelo Conceptual es la siguiente:

Ilustración 58 Modelo Conceptual de la Base de Datos



Fuente: (Propia, 2020)

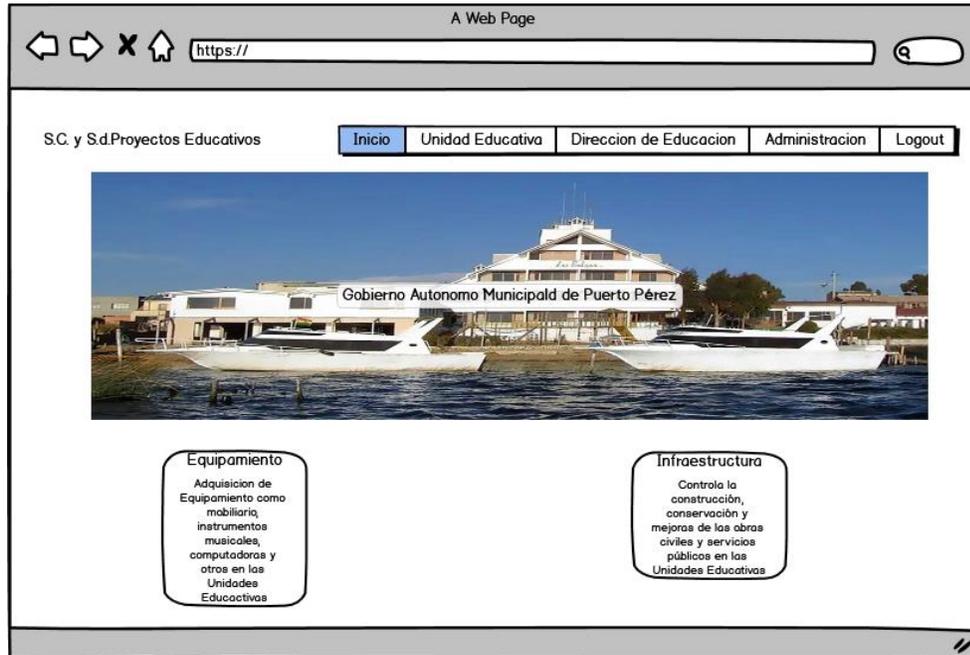
3.6. IMPLEMENTACIÓN DEL SISTEMA

REQUERIMIENTOS DEL SISTEMA:

- Sistema Operativo: Windows 10 Pro.
- Base de Datos: MySQL.
- Diseño Base de Datos: MySQL.
- Lenguaje de Programación: PHP 5.6.
- Framework: Yii.
- Framework-CSS: Bootstrap, css.
- Servidor Web: XAMPP v3.2.2.
- Editor de Texto: Sublime Text 3, Visual Studio Code Notepad ++.

En esta fase se establece el diseño de la interfaz gráfica de la aplicación web.

Ilustración 59 Menú de Inicio



Fuente: (Propia, 2020)

Una vez que el usuario ha realizado el ingreso a la dirección del enlace, el sistema la permitirá el ingreso a la página principal, donde el usuario pueda acceder a las diferentes opciones del menú del sistema.

Ilustración 60 Código Fuente - Menú Inicio

```
<center><h1>Gobierno Autónomo Municipal de Puerto Pérez</h1>
<h4>DIRECCION DE EDUCACION <br/></h4></center>

<br />
<div class="row-fluid">
  <ul class="thumbnails center">
    <li class="span3">
      <div class="thumbnail">
        <center><h3>Equipamiento</h3>

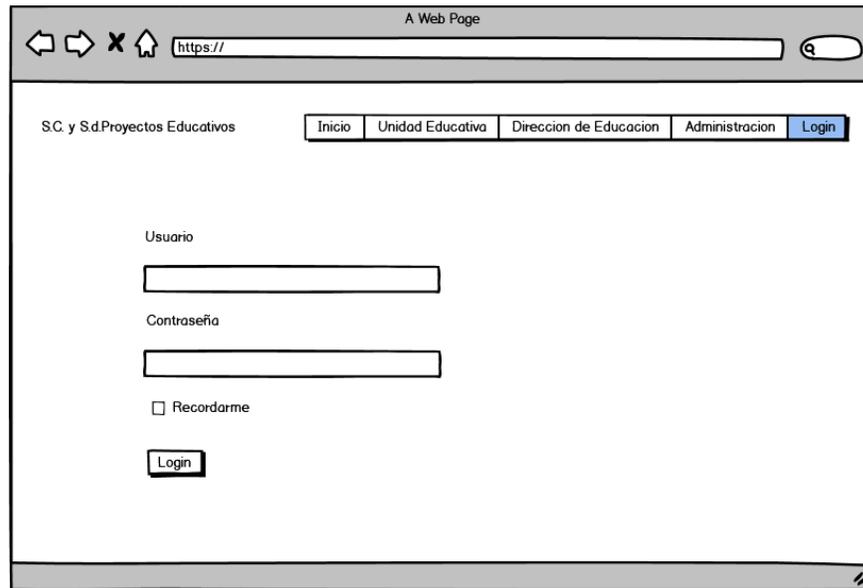
        <div class="round_background r-grey-light">
          <?php echo CHtml::link(CHtml::image('img', 'Lov', array('width'=>170)), array('index')); ?>
        </div></center>

        <center><p>Adquisición de Equipamiento como mobiliario, instrumentos musicales, computadoras y otros en
        las Unidades Educativas</p></center>
      </div>
    </li>
    <li class="span3">
      <div class="thumbnail">
        <center> <h3>Infraestructura</h3>

        <div class="round_background r-yellow">
          <?php //echo CHtml::link(CHtml::image('images/2.ico', 'Lov', array('width'=>170)), array('index')); ?>
          <?php echo CHtml::link(CHtml::image('images/fotos/01.jpg', 'Lov', array('width'=>'110', 'height'=>'110'
          )), array()); ?>
        </div>
      </div>
    </li>
  </ul>
</div>
```

Fuente: (Propia, 2020)

Ilustración 61 Menú - Login - Modulo Login de Usuario



Fuente: (Propia, 2020)

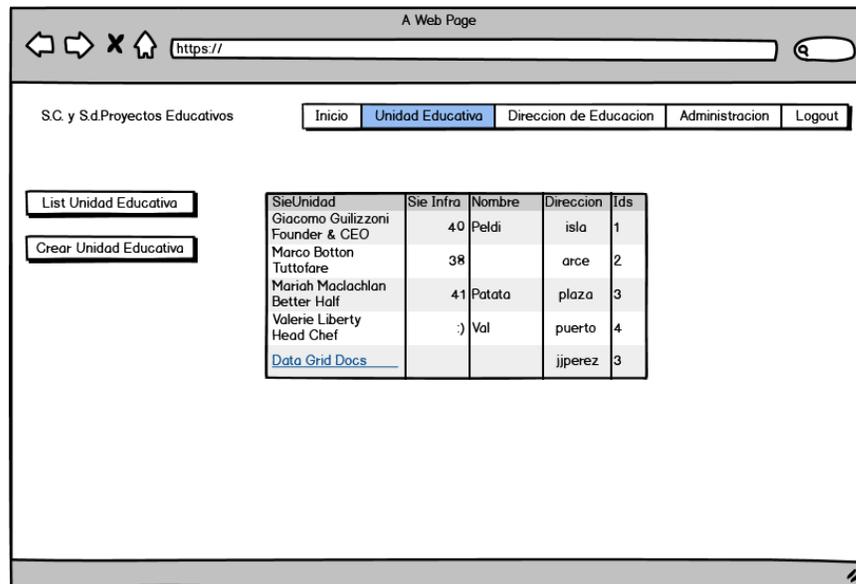
Esta interfaz está diseñada para verificar el acceso al sistema, siendo la primera pantalla en el menú de opciones Login, que el usuario o administrador verá, donde el sistema le pedirá los datos de autenticación “logueo” que son asignados por gerencia de operaciones, para la verificación del acceso al sistema.

Ilustración 62 Código Fuente - Login

```
5  $this->pageTitle=Yii::app()->name . ' - Login';
6  $this->breadcrumbs=array(
7      'Login',
8  );
9  >>
10 <div class="page-header">
11 <h1>Ingresa <small> a tu cuenta</small></h1>
12 </div>
13 <div class="row-fluid">
14 <div class="span6 offset3">
15 <?php
16 $this->beginWidget('zii.widgets.CPortlet', array(
17     'title'=>"Acceso Privado",
18 ));
19 >>
20 <p>Por favor ingresa tu Usuario y Contraseña:</p>
21 <div class="form">
22 <?php $form=$this->beginWidget('CActiveForm', array(
23     'id'=>'login-form',
24     'enableClientValidation'=>true,
25     'clientOptions'=>array(
26         'validateOnSubmit'=>true,
```

Fuente: (Propia, 2020)

Ilustración 63 Menú - Unidad Educativa



Fuente: (Propia, 2020)

A continuación, se muestra el menú Unidad educativa, donde veremos el listado de todos los colegios pertenecientes al GAMPP, donde veremos cada detalle ahí, también vemos los botones de crear de un nuevo colegio en caso de que se cree una.

Ilustración 64 Código- Unidad Educativa

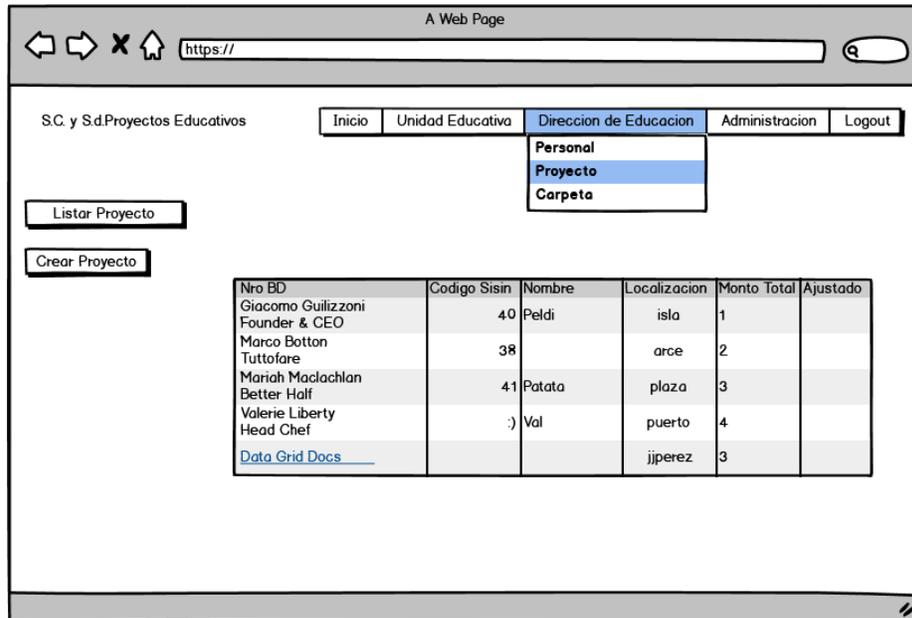
```
$this->menu=array(
    array('label'=>'List UnidadEducativa', 'url'=>array('index')),
    array('label'=>'Create UnidadEducativa', 'url'=>array('create')),
    array('label'=>'Update UnidadEducativa', 'url'=>array('update', 'id'=>$model->id_uni_edu)),
    array('label'=>'Delete UnidadEducativa', 'url'=>'#', 'linkOptions'=>array('submit'=>array('delete','id'=>$model->id_uni_edu),'confirm'=>'Are you sure you want to delete this item?')),
    array('label'=>'Manage UnidadEducativa', 'url'=>array('admin')),
);
?>

<h1>View UnidadEducativa #<?php echo $model->id_uni_edu; ?></h1>

<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
        'id_uni_edu',
        'sie_uni',
        'sie_infra',
        'nombre',
        'direccion',
        'telefono',
        'cantidad_alum',
        'director',
        'id_dis',
    ),
)); ?>
```

Fuente: (Propia, 2020)

Ilustración 65 Menú Dirección de Educación - Sub Menú - Proyecto Listado de Proyectos



Fuente: (Propia, 2020)

Sub Menú proyecto, se encarga de mostrar el listado y la creación de nuevos proyectos, también se puede hacer la modificación y eliminación ya sea el rol superior que use el administrador.

Ilustración 66 Código - Listado de Proyecto

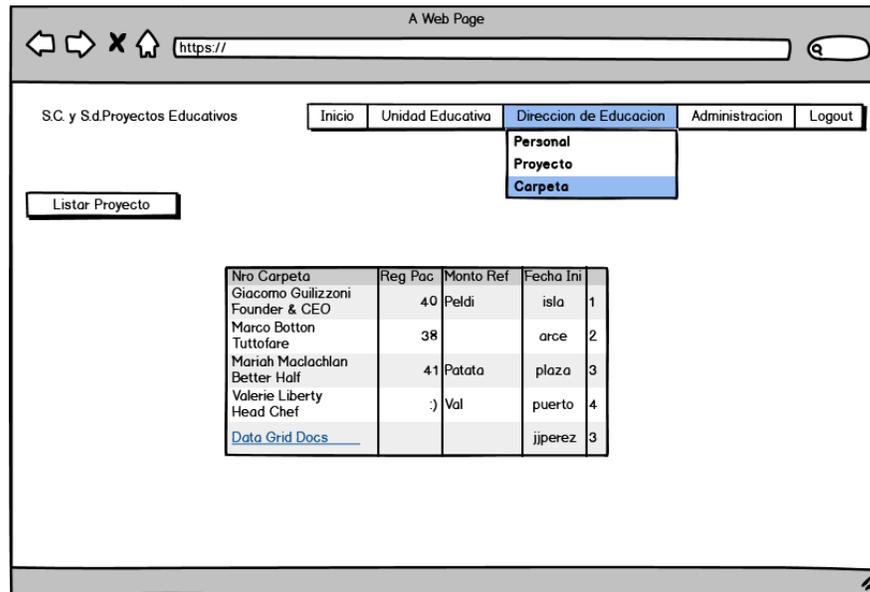
```

<tr>
<td><b><?php echo CHtml::encode($data->getAttributeLabel('id_proyecto')); ?></b>
<?php echo CHtml::link(CHtml::encode($data->id_proyecto), array('carpeta/createc', 'id'=>$data->id_proyecto)
); ?>
<br />
</td><?php echo CHtml::encode($data->getAttributeLabel('nro_bd')); ?></b>
<?php echo CHtml::encode($data->nro_bd); ?>
<br />
</td><?php echo CHtml::encode($data->getAttributeLabel('sisin')); ?></b>
<?php echo CHtml::encode($data->sisin); ?>
<br />
</td><?php echo CHtml::encode($data->getAttributeLabel('nombre_proy')); ?></b>
<?php echo CHtml::encode($data->nombre_proy); ?>
<br />
</td><?php echo CHtml::encode($data->getAttributeLabel('localizacion')); ?></b>
<?php echo CHtml::encode($data->localizacion); ?>
<br />
</td><?php echo CHtml::encode($data->getAttributeLabel('distrito_proy')); ?></b>
<?php echo CHtml::encode($data->distrito_proy); ?>
<br />
</td><?php echo CHtml::encode($data->getAttributeLabel('monto_total')); ?></b>
<?php echo CHtml::encode($data->monto_total); ?>
<br />

```

Fuente: (Propia, 2020)

Ilustración 67 Sub Menú - Carpeta



Fuente: (Propia, 2020)

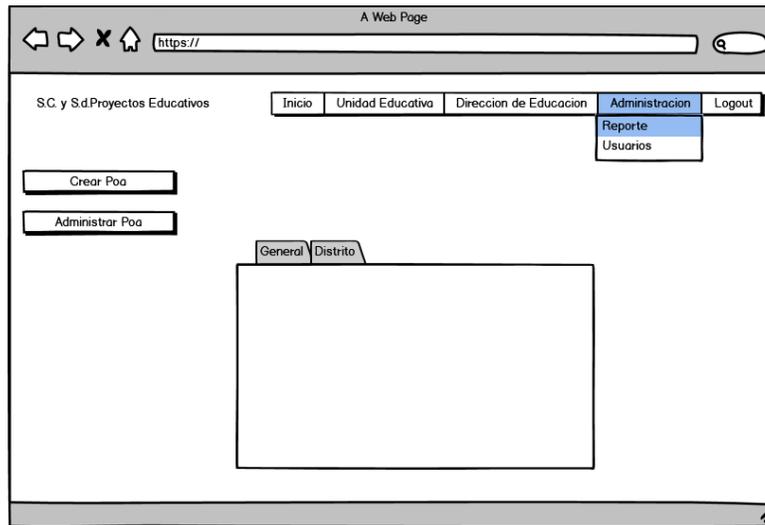
Sub menú Carpeta, muestra el listado de las carpetas que están en trámite detallando la fase de inicio del proyecto a ejecutarse.

Ilustración 68 Código - Sub Menú Carpeta

```
<div class="view">
<table>
<tr>
<td>
<b><?php echo CHtml::encode($data->getAttributeLabel('id_carpeta')); ?></b>
<?php echo CHtml::link(CHtml::encode($data->id_carpeta), array('view', 'id'=>$data->id_carpeta)); ?>
<br />
<b><?php echo CHtml::encode($data->getAttributeLabel('numero_carpeta')); ?></b>
<?php echo CHtml::encode($data->numero_carpeta); ?>
<br />
<b><?php echo CHtml::encode($data->getAttributeLabel('reg_pac')); ?></b>
<?php echo CHtml::encode($data->reg_pac); ?>
<br />
<b><?php echo CHtml::encode($data->getAttributeLabel('monto_ref')); ?></b>
<?php echo CHtml::encode($data->monto_ref); ?>
<br />
<b><?php echo CHtml::encode($data->getAttributeLabel('fecha_ini')); ?></b>
<?php echo CHtml::encode($data->fecha_ini); ?>
<br />
</td>
</tr>
</table>
</div>
```

Fuente: (Propia, 2020)

Ilustración 69 Menú Administración - Reportes



Fuente: (Propia, 2020)

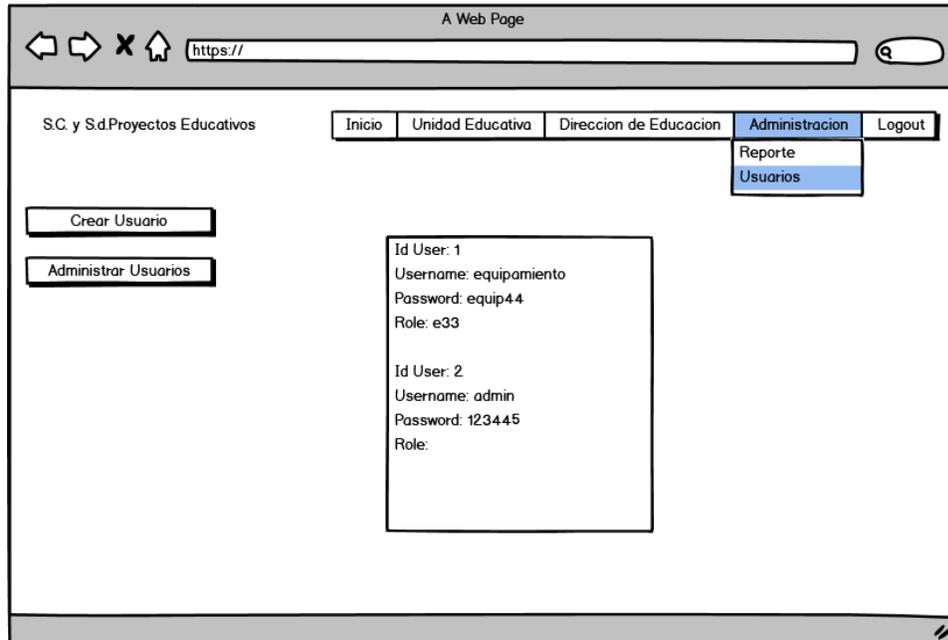
Menú administración, donde se detalla todos los reportes de la poa ya sea general o por distrito.

Ilustración 70 Código - Administración

```
?>  
  
<div class="view">  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('id_poa')); ?></b>  
<?php echo CHtml::link(CHtml::encode($data->id_poa), array('view', 'id'=>$data->id_poa)); ?>  
<br />  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('nro_bd')); ?></b>  
<?php echo CHtml::encode($data->nro_bd); ?>  
<br />  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('sisin')); ?></b>  
<?php echo CHtml::encode($data->sisin); ?>  
<br />  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('proyecto')); ?></b>  
<?php echo CHtml::encode($data->proyecto); ?>  
<br />  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('direccion')); ?></b>  
<?php echo CHtml::encode($data->direccion); ?>  
<br />  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('tipologia')); ?></b>  
<?php echo CHtml::encode($data->tipologia); ?>  
<br />
```

Fuente: (Propia, 2020)

Ilustración 71 Administración - Usuarios



Fuente: (Propia, 2020)

Sub menú usuarios, donde muestra el listado de todos los usuarios registrados con los diferentes niveles de seguridad y confiabilidad que se establece de acuerdo al administrador.

Ilustración 72 Administración - Usuarios

```
<div class="view">  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('id_user')); ?></b>  
<?php echo CHtml::link(CHtml::encode($data->id_user), array('view', 'id'=>$data->id_user)); ?>  
<br />  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('username')); ?></b>  
<?php echo CHtml::encode($data->username); ?>  
<br />  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('password')); ?></b>  
<?php echo CHtml::encode($data->password); ?>  
<br />  
  
<b><?php echo CHtml::encode($data->getAttributeLabel('role')); ?></b>  
<?php echo CHtml::encode($data->role); ?>  
<br />  
</div>
```

Fuente: (Propia, 2020)

3.7. APLICACIÓN DE MÉTRICAS DE CALIDAD

se determina la calidad del sistema web en base a los parámetros de medición de la norma ISO 9126, donde hallaremos el punto función que nos servirá para el capítulo de costo beneficio, y las medidas de seguridad que se deben adoptar del lado del cliente y lado del servidor.

La ingeniería de software se diferencia de otras áreas, al no estar basada en leyes cuantitativas básicas, en su lugar se realiza un conjunto de medidas conocidas como métricas, las cuales proporcionan una referencia de la calidad algún producto de software (Pressman, 2010).

NORMA ISO 9126

La norma ISO 9126 (International Standard Organization – Organización Internacional de Normalización) es un estándar internacional para la evaluación de software, que nos ayudará a medir la calidad del sistema siguiendo los siguientes criterios.

3.7.1. FUNCIONALIDAD

Métrica para obtener una valoración mediante el cálculo del punto función en base a la evaluación de un conjunto de características y capacidades que debe cumplir el sistema:

Número de entradas de usuario: Se cuenta cada entrada de usuario que proporciona al software diferentes datos orientados a la aplicación. Muestra la lista de entradas de usuario que tiene el sistema.

Tabla 11 Lista de entradas del usuario del sistema

Nro	Entrada De Usuarios
1	Ingreso al sistema
2	Registro de usuarios

3	Registro de roles
4	Registro de permisos
5	Registro de Carpetas
6	Registro de Proyectos

Fuente: (Propia, 2020)

Número de salidas de usuario: Se refiere a información elaborada por el sistema para ser mostrada al usuario. Se muestra las salidas de usuarios que tiene el sistema.

Tabla 12 Lista de salidas de usuario del sistema

Nro	Salida De Usuarios
1	Listado de usuarios
2	Listado de colegios
3	listado de proyectos
4	Listado de carpetas

Fuente: (Propia, 2020)

Número de peticiones de usuarios: Son entradas interactivas entre el usuario y el sistema, donde la salida es inmediata. Muestra las peticiones de usuario al sistema.

Tabla 13 Lista de peticiones del usuario al sistema

Nro	Peticiones De Usuarios
1	Autenticación de usuario
2	Listar proyectos
3	Listar carpetas
4	Listado de carpetas

Fuente: (Propia, 2020)

Número de archivos: Se cuenta cada archivo maestro lógico. En otras palabras, las tablas existentes en la base de datos. Muestra los archivos lógicos del sistema.

Tabla 14 Lista de archivos lógicos del sistema

Nro	Archivos Lógicos
1	Usuario
2	Roles
3	Permisos
4	Proyecto
5	Carpeta

Fuente: (Propia, 2020)

Número de interfaces externas: Se cuenta todas las interfaces legibles por el ordenador que son utilizados para transmitir la información. En este caso es solo internet

Tabla 15 Factores de ponderación

Parámetros de medición	Cuenta	Factores de ponderación			Valor Obtenido
		Simple	Media	Complejo	
Nro. De entradas de usuario	6	-	4	-	24
Nro. de salidas de usuario	4	-	5	-	20
Nro. De peticiones de usuario	4	-	5	-	20
Nro. De archivos	5	-	5	-	25
Nro. De interfaces externas	1	-	2	-	2
Cuenta total					91

Fuente: (Propia, 2020)

la tabla muestra el factor de ajuste de complejidad en base a las respuestas de las siguientes preguntas evaluadas entre 0 y 5.

Tabla 16 Factor de ajuste de complejidad

Factores de complejidad	Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
	0	1	2	3	4	5	
¿Requiere el sistema copias de seguridad y de recuperaciones fiables?				X			4
¿Se requiere comunicación de datos?					X		4
¿Existen funciones de procesamiento distribuido?				X			3
¿Es crítico el rendimiento?				X			3
¿Se ejecutará el sistema con un entorno operativo existente y fuertemente utilizado?					X		4
¿Requiere el sistema entrada de datos interactiva?						X	5
Facilidad Operativa					X		4
¿Se actualiza los archivos maestros de forma interactiva?					X		4

¿Son complejas las entradas, las salidas, los archivos o las peticiones?	X	4
Procesamiento interno complejo	X	3
Diseño de código reutilizable	X	5
Facilidad de Instalación	X	5
¿Soporta múltiples instalaciones en diferentes sitios?	X	5
Facilidad de cambios	X	5
Factor de ajuste de complejidad		58

Fuente: (Propia, 2020)

La funcionalidad es medida a través del punto función (PF), que proporciona una medida objetiva, cuantitativa y auditable del tamaño de la aplicación, basada en la visión del usuario de la aplicación (Pressman, 2010).

Para calcular el punto función se utiliza la siguiente relación:

$$PF = Cuenta\ Total * (X + Min(Y) * \sum Fi)$$

Dónde:

PF: Medida de funcionalidad.

Cuenta Total: Es la suma de los siguientes datos (Nro. de entradas, Nro. de salidas, Nro. de peticiones, Nro. de archivos, Nro. de interfaces externas).

X: Confiabilidad del proyecto, varía entre 1 a 100%.

Min (Y): Error mínimo aceptable al de la complejidad.

$\sum Fi$: Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$

Para calcular el PF se usa la siguiente ecuación:

$$PF = \text{cuenta total} * (X + \text{Min} (Y) * \sum Fi)$$

$$PF = \text{cuenta total} * [0.65 + (0.01 * \sum Fi)]$$

Reemplazando los valores obtenidos en las tablas 11 y 12 se obtiene el siguiente resultado:

$$PF = 91 * [0.65 + (0.01 * 58)]$$

$$PF = 111,93$$

A continuación, calculamos el PF ideal:

$$PF_{\text{ideal}} = 91 * [0.65 + (0.01 * 70)]$$

$$PF_{\text{ideal}} = 122,85$$

Entonces la funcionalidad del sistema es:

$$\text{Funcionalidad} = \frac{PF}{PF_{\text{ideal}}} * 100$$

$$\text{Funcionalidad} = \frac{111,93}{122,85} * 100$$

$$\text{Funcionalidad} = 91,1111$$

Entonces la funcionalidad del sistema es: Con el resultado obtenido se puede interpretar que 9 de cada 10 personas, consideran que el sistema responde de manera óptima a las funcionalidades requeridas por las direcciones Educativas.

3.7.2. CONFIABILIDAD

Para determinar la confiabilidad de un sistema, se toma en cuenta las fallas que puedan ocurrir en el sistema en un tiempo determinado. En el desarrollo de software las fallas son más que todo por diseño e implementación. Para medir el tiempo medio entre fallos (TMEF) se usará la siguiente fórmula:

$$TMEF = TMDF + TMDR$$

Donde:

TMDF: Tiempo medio de fallo.

TMDR: Tiempo medio de reparación.

Se estima que un fallo puede ocurrir cada 20 días hábiles y su reparación en promedio pueda tomar 1 hora después de haber entregado una nueva funcionalidad del sistema, entonces:

$$TMEF = \frac{20 \times 24 \times 8}{365} + 1 = 161 \text{ horas}$$

Por lo que la disponibilidad es un buen indicador de fiabilidad, en base de la siguiente formula se tiene:

$$\text{Disponibilidad} = \frac{160}{161} * 100 = 99,37888199$$

Con lo que se llega a la conclusión de que el sistema tiene un 99.4% de confiabilidad.

3.7.3. USABILIDAD

La usabilidad representa facilidad de uso que el usuario final recibirá del sistema. Esta métrica nos muestra el esfuerzo necesario para aprender a manipular el sistema (Pressman, 2010).

La tabla 13 muestra los resultados obtenidos en base a preguntas propuestas a los usuarios del sistema.

Tabla 17 Preguntas para obtener el grado de usabilidad

Nro.	Pregunta	Valor	0 -
1	¿Es entendible?	87	100
2	¿Las pantallas son agradables a la vista del usuario?	95	

3	¿Es fácil de aprender?	90
4	¿Contiene información necesaria?	90
5	¿Facilita su trabajo?	86
6	¿La navegabilidad es fluida?	94
Promedio		90,33

Fuente: (Propia, 2020)

Entonces la usabilidad del sistema seria del 90.33%, lo que indica que 9 de cada 10 usuarios pueden utilizar el sistema con facilidad.

3.7.4. MANTENIBILIDAD

Para la evaluación de la mantenibilidad, se desarrolló algunas preguntas, estas preguntas son valoradas en porcentaje por el desarrollador del sistema al momento de la culminación del proyecto.

Este valor tiene consideración por la experiencia y la forma de trabajo de cada programador, el mismo puede ser relativo respecto a otros desarrolladores (Largo y Marin, 2005).

La tabla 14 muestra las preguntas y los resultados obtenidos en la evaluación de mantenibilidad, estas preguntas se las hizo a todo el equipo de desarrollo del sistema.

Tabla 18 Evaluación de mantenibilidad

Factor de ajuste	Valor
¿Se puede modificar el sistema?	92%
¿Deja identificar las partes que deben ser modificadas?	95%
¿Permite implementar una modificación específica?	92%

¿Presenta efectos inesperados como posibles errores?	98%
<hr/>	
Total	94,25%

Fuente: (Propia, 2020)

El resultado de la mantenibilidad es de 94.25%, lo que significa que el esfuerzo necesario para realizar mantenimiento al sistema es mínimo.

3.7.5. PORTABILIDAD

La portabilidad es la capacidad con que un software puede ser llevado de un entorno a otro, considera la facilidad de instalación, ajuste y adaptación al cambio. Para medir la portabilidad del sistema usaremos la siguiente relación:

$$GP = 1 - \frac{ET}{ER}$$

Donde:

GP: Grado de portabilidad

ET: Recursos necesarios para llevar el sistema a otro entorno.

ER: Recursos necesarios para crear el sistema en el entorno residente.

Si:

- $GP > 0$, la portabilidad es más rentable que el re-desarrollo.
- $GP < 0$, el re-desarrollo es más rentable que la portabilidad.
- $GP = 0$, la portabilidad es perfecta.

Los recursos necesarios para llevar el sistema a otro entorno son: servicio de hosting para alojar el código fuente, la base de datos, dominio para la url, conexión ftp, conexión a internet, conexión intranet, responsivo, espacio almacenamiento, y los recursos necesarios para crear el sistema son: html5 y frameworks (YII v2.0, jquery, bootstrap 3, css3, json y composer).

Consideremos que el esfuerzo para portar y para implementar es de 10 y 40 respectivamente de un rango entre el 1-100.

$$\text{Entonces, } PG = 1 - \left(\frac{10}{40}\right)$$

$$PG = 0,75$$

Con el resultado obtenido sabemos que el grado de portabilidad es del 75%, entonces la portabilidad del sistema es más rentable que su desarrollo.

3.7.6. EFICIENCIA

La eficiencia es el conjunto de atributos que se relacionan con el nivel de performance del software y de la cantidad de recursos usados, bajo las condiciones establecidas, en tiempo y en recursos.

El sistema se considera eficiente por la óptima utilización de los recursos.

Tabla 19 Resultado General Norma ISO 9126

CRITERIOS EVALUADOS	RESULTADOS
Funcionalidad	91,11
Confiabilidad	99,37
Usabilidad	90,33
Mantenibilidad	94,25
Portabilidad	75
Evaluación de la Calidad Final	90,012

Fuente: (Propia, 2020)

3.8. COSTOS

Para calcular el costo del proyecto se lo realizará haciendo uso del modelo COCOMO II. El modelo COCOMO tiene una jerarquía de modelos como ser: básico, intermedio y avanzado, la cual se aplica a tres diferentes tipos de software.

ORGÁNICO: Proyectos relativamente sencillos, menores a 5000 líneas de código, implica procesamiento de datos, uso de la base de datos se focaliza en transacciones y recuperación de datos.

SEMIACOPLADO: Proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de proyectos es variable y las restricciones intermedias.

EMPOTRADO: Proyectos bastantes complejos, en los que apenas se tiene experiencia y en un entorno de gran innovación técnica.

3.8.1. COCOMO II

El Modelo Constructivo de Costes (COCOMO) es un modelo matemático de base empírica, utilizando para la estimación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

En su libro clásico acerca de economía de la ingeniería de software; Barry Boehm introdujo una jerarquía de modelos de estimación de software que llevan el nombre de COCOMO. El modelo original se convirtió en uno de los modelos de estimación de costo más ampliamente utilizados y estudiados en la industria. Evolucionó hacia un modelo de estimación más exhaustivo llamado COCOMO II (Pressman, 2010).

Aplicando de las fórmulas básicas de esfuerzo, tiempo calendario y personal requerido.

Las ecuaciones de COCOMO básico tiene la siguiente forma:

$E = a(kldc)^b$; Persona – mes

$D = c(E)^d$; Meses

$P = \frac{E}{D}$; Personas

Dónde:

E: Esfuerzo requerido por el proyecto expresado en persona-mes.

D: Tiempo requerido por el proyecto expresado en meses.

P: Número de personas requeridas para el proyecto.

a, b, c y d: Constantes con valores definidos según cada sub-modelo.

KLDC: Cantidad de líneas de código distribuidas en miles.

Tabla 20 Constantes a,b,c,d COCOMO II

Modo	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semilibre	3,0	1,12	2,5	0,35
Rígido	3,6	1,2	2,5	0,32

Fuente: (Propia, 2020)

Para el cálculo del desarrollo del software se tendrá como partida el punto función no ajustada que se encontró en el modelo ISO 9126, cuyo valor encontrado es:

$$PF = 111,93$$

Tabla 21 Conversión de PF a KLDC

Lenguaje	Nivel	Factor LDC / PF
C	2,5	128
Ansi	5	64
Basic		
Java	6	53

PL / I	4	80
Ansi	3	107
Cobol 74		
Visual	7	46
Basic		
ASP	9	36
PHP	11	29
Visual	9,5	34
C++		

Fuente: (Propia, 2020)

Entonces, realizando los cálculos y escogiendo el valor del lenguaje de programación PHP de la tabla 16, tenemos:

$$LDC = PF * \text{Factor LDC} / PDF$$

$$LDC = 111,93 * 29$$

$$LDC = 3245,97$$

$$KLDC = \frac{3245,97}{1000} = 3,24$$

Ahora, para hallar el esfuerzo reemplazamos los valores obtenidos hasta ahora:

$$E = a(klde)^b; \text{ Persona - mes}$$

$$E = 3,0 * (3,24)^{1,12}$$

$$E = 11,193$$

$$D = c(E)^d; \text{ Meses}$$

$$D = 2,5 * (11,193)^{0,35}$$

$$D = 5,822$$

Para el cálculo del número de programadores para el desarrollo:

$$P = \frac{E}{D}$$

$$P = \frac{11,193}{5,822}$$

$$P = 1,922 \cong 2$$

Estimando que el salario medio, tomando en cuenta desde el ministerio de trabajo en Bolivia, se tiene como salario mínimo Bs. 2122, esta cifra será tomada en cuenta para la siguiente estimación:

Costo del software desarrollado = Numero de programadores * Salario de un programador (Pressman, 2010).

Costo del Software Desarrollado por Persona = 2 * 2122 = 4244. Bs.

Costo total del Software Desarrollado = 4244 * 6 = 25.464,00 Bs.

Lo que significa que el costo del sistema desarrollado es de Bs. 25.464 por los 6 meses.

3.8.2. COSTO DE ELABORACIÓN DE PROYECTO

La tabla 17 muestra los costos de inversión de los recursos que se usaron para la elaboración del sistema.

Tabla 22 Costos de recursos empleados para la elaboración del sistema.

Recursos	Costo (Bs)
Material de escritorio	250
Investigación del proyecto	450
Internet	350
Otros	250
Total	1300

Fuente: (Propia, 2020)

Por tanto, el costo de la elaboración del proyecto es de Bs. 1300.

3.8.3. COSTO TOTAL DEL SISTEMA

El costo total del sistema se obtiene de la sumatoria del costo de desarrollo y el costo de elaboración del proyecto, en la tabla 5.4 se puede observar los resultados, todos los costos están expresados en bolivianos.

Tabla 23 Costo total del sistema

Detalle	Costo (Bs)
Costo de desarrollo	25464
Costo de elaboración del proyecto	1.300
Costo Total	26.764

Fuente: (Propia, 2020)



CAPÍTULO

IV



PRUEBAS Y RESULTADOS

4. PRUEBAS Y RESULTADOS

4.1. PRUEBAS

La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. Es un proceso de ejecución de un programa con la intención de descubrir un error, no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software (ecured.cu, 2020).

- Algunas reglas que sirven bien como objetivos de prueba (Glen Myers, 2008):
- La prueba es el proceso que ejecuta un programa con objeto de encontrar un error.
- Un buen caso de prueba es el que tiene alta probabilidad de encontrar un error que no se ha detectado hasta el momento.

Una prueba exitosa es la que descubre un error no detectado hasta el momento.

De acuerdo a la prueba del software se hace la verificación de los requerimientos funcionales y no funcionales (Pmoinformatica.com, 2018).

Funcionales

- La información debe estar disponible en el sistema.
- Registrar los datos de los proyectos educativos que son nuevas.
- El registro de usuarios solo se realizará por parte de administración viendo los roles de cada uno.
- Se deben poder realizar modificaciones en los proyectos educativos.
- Los usuarios que tengan el rol de usuario, solo podrán realizar acciones de acuerdo al nivel de seguridad.
- El técnico GAMPP hará el seguimiento y verificación del estado del proyecto, recopilando esos datos para su actualización.
- Los reportes se realizan sobre el estado y los ítems del proyecto en proceso.
- Todo dato que se registre debe estar guardado en la base de datos.

No Funcionales.

El Gobierno Autónomo Municipal de Puerto Pérez cuenta con equipo de computación.

- El personal del GAMPP cuenta con conocimientos básicos sobre el manejo de la computadora.
- Los administrativos tienen conocimientos medios con manejo de paquetes básicos de software.

4.2. RESULTADOS

En muchas formas, la prueba es un proceso de individualización, y el número de tipos diferentes de pruebas varía tanto como los diferentes acercamientos para su desarrollo. Durante muchos años, la única defensa contra los errores de programación fue el diseño cuidadoso y la inteligencia natural del programador. Ahora estamos en una era en la que modernas técnicas de diseño (y revisiones técnicas) ayudan a reducir el número de errores iniciales que son inherentes al código. De igual modo, diferentes métodos de prueba comienzan a agruparse en métodos y filosofías distintos (Shooman[Sho83]).

De acuerdo al desarrollo del software y mejoras dentro del municipio con el sistema, se evidencia una aceptación del usuario y las unidades educativas, ya que anteriormente los tramites no se tenían de forma oportuna, pero con este sistema se evidencia que existe una disponibilidad de la información en cualquier momento, ya que esto ahorra tiempo a la hora de tener disponible eficientemente en poco tiempo; lo que antes se demoraba horas, incluso semanas las respuestas, ahora se tiene una mejora de 5 minutos para tener la información en manos.



CAPÍTULO

V



CONCLUSIONES Y RECOMENDACIONES

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

El objetivo principal de este proyecto de grado, era desarrollar e implementar un “Sistema de Control y Seguimiento de Proyectos Educativos” el objetivo de apoyar el fortalecimiento del Gobierno Autónomo Municipal de Puerto Pérez, con la puesta en marcha del presente proyecto se llegó a su conclusión satisfactoria logrando alcanzar los objetivos específicos propuestos y cumpliendo los diferentes requerimientos del Municipio:

Se Implementó una base de datos donde se hace el registro de todos los proyectos de las unidades educativas.

Se desarrolló los módulos para hacer el control y seguimiento de los proyectos educativos.

Se desarrolló los módulos para poder administrar desde diferentes niveles de seguridad para acceder al sistema para cada usuario.

Se realizó la capacitación al personal técnico que este encargado de realizar el registro y el seguimiento del estado de cada uno de los proyectos.

Se realizó un buen diseño de interfaz, que sea agradable con el entorno para que tenga la facilidad de navegar dentro del sistema.

5.2. RECOMENDACIONES

Una vez concluido el proyecto de grado se recomienda como nuevos trabajos de investigación lo siguiente:

Realizar el mantenimiento del sistema en un determinado tiempo de (3 meses) para ver el buen funcionamiento, en el almacenamiento de información y procesos concurrentes como consultas, registro, etc.

Actualizar el framework Yii 1.1 a la nueva versión Yii 2.0 introduce más características como por ejemplo Yii 1.1. funciona con PHP 5.2.

El presente proyecto puede aplicarse en varios municipios convirtiéndose en un aporte para el estado boliviano por la cual pueda controlar de manera transparente confiable y centralizada los recursos asignados a los diferentes municipios.

Compra de Servidor Propio dedicado para su almacenaje de todo el sistema, y sistemas que ya se están desarrollando para tenerlo de forma virtual, así toda la población del municipio pueda acceder para visualizar.

No obstante, ninguno de los incluidos en el software es definido, esto posibilita las actualizaciones periódicas de acuerdo al avance en la utilización de las mismas o se podría

considerar implementar nuevos módulos para fortalecer y ampliar las necesidades que plantee a la institución, de tal manera que este trabajo sirva como base para considerar nuevos proyectos.

BIBLIOGRAFÍA

2000-2017 No Magic, I. (s.f.). *Magic Draw*. Obtenido de

<https://www.magicdraw.com/main.php>

Apache Friends Copyright (c) 2020. (s.f.). *XAMPP*. Obtenido de

<https://www.apachefriends.org/es/index.html>

Barry Boehm. (1995). *TECNOLOGIAS PARA LA INTEGRACION DE SOLUCIONES*.

CASILLO, J. R. (2014). *SISTEMA WEB PARA EL CONTROL Y ADMINISTRACIÓN DE RECURSOS HUMANOS*. La Paz.

Cuomo Castares, V. C.-M. (s.f.). ISO 9126 - ISO 14598.

ecured.cu. (2020). *ecured*. Obtenido de ecured:

https://www.ecured.cu/Pruebas_de_software

Explain, D. (s.f.). *Dr.Explain*. Obtenido de Dr.Explain: <https://www.drexplain.com/>

Glen Myers, [. (2008). *FUNDAMENTOS DE LA PRUEBA DEL SOFTWARE*.

Google. (s.f.). Obtenido de <https://www.google.com/imghp?hl=es-419>

<http://www.cyta.com.ar/>. (s.f.).

<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c1/c1.htm>.

IBARROLA, M. D. (1972). *Enfoque Tecnológico Sistémico*.

IZAMORAR. (2015). *Ciclo-de-Vida-de-un-Sistema-de-Información*. Obtenido de

<https://izamorar.com/ciclo-de-vida-de-un-sistema-de-informacion/ciclo-de-vida-de-un-sistema-de-informacion/>

Koch, N. (2003). *Modelling and Implementing Web Applications*.

- Largo y Marin, C. A. (2005). *Tecnica par ala Evaluacion de Software*. Obtenido de https://jrvargas.files.wordpress.com/2009/03/guia_tecnica_para_evaluacion_de_software.pdf
- Mouse, J. (2011). *MVC: Modelo, Vista y Controlador en PHP*. Obtenido de <https://www.jc-mouse.net/proyectos/mvc-modelo-vista-y-controlador-en-php>
- München, L. –L.-M.-U. (2020). *UWE TUTORIAL*. Obtenido de UWE: <https://uwe.pst.ifi.lmu.de/teachingTutorialProcessSpanish.html>
- Mysql © 2020, O. C. (s.f.). *Mysql*. Obtenido de <https://www.mysql.com/>
- Nolivos Coronel, N. Q. (2013). *Análisis diseño desarrollo e implementación de un sistema WEB para el control de un taller técnico aotomotriz en plataforma PHP-MYSQL utilizando metodología WEB UWE para la empresa Metroautocerfran CIA. LTDA*. Quito: Universidad de las Fuerzas Armadas ESPE.Carrera de ingeniería en Sistemas e Informática.
- PERALTA. (2008). *Manual de Apoyo de Fundamentos de Sistemas de Información*. Obtenido de <http://uprr.edu.mx/Fundamentos%20de%20Sistemas%20de%20Informacion.pdf>
- Pérez, H. F. (2010). *Propuesta de Análisis y Diseño Basada en UML y UWE para la Migración de Arquitectura de Software Centralizada hacia Internet*. Guatemala.
- Piaget Arana, X. (s.f.). Administración de proyectos informáticos.
- Pilar, d. S. (2014). *Tutorial de Patron MVC*. Obtenido de <https://www.codigonexo.com/wp-content/uploads/2014/06/Curso-completo-MVC.pdf>

Pmoinformatica.com. (2018). *Pmoinformatica.com*. Obtenido de Pmoinformatica.com:

<http://www.pmoinformatica.com/2017/02/pruebas-de-caja-negra-ejemplos.html>

Pressman, R. S. (2010). *Ingeniería de Software*. En R. S. Pressman. Mexico: ISBN: 978-607-15-0314-5.

Propia. (2020).

Rena. (2008). *Componentes de un sistema*.

RENA. (2008). *Sistemas de información*. Obtenido de

<http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema10.html>

Shooman[Sho83]. (s.f.). *Estrategias de prueba de software*.

TOBAR, S. E. (2013). *SISTEMAS INFORMÁTICO PARA EL CONTROL Y SEGUIMIENTO DE PROYECTOS DE LA DIRECCIÓN DE ORDENAMIENTO FORESTAL, CUENCA Y RIEGO. EL SALVADOR*.

Umsa. (2015). *SISTEMA WEB DE SEGUIMIENTO Y MONITOREO DE PROYECTOS ORIENTADOS A RESULTADOS CASO: COLEGIO DE ADMINISTRADORES DE EMPRESAS DE LA PAZ “CADELP”*. En V. Tintaya. La Paz.

UTEA, C. d. (2011). *Gerencia de Proyectos Informáticos*.

wikipedia. (s.f.). *Esquema del Modelo Cliente Servidor*. Obtenido de

<https://es.wikipedia.org/wiki/Cliente-servidor>

Yii Framework. (2020). *yii frameworwk*. Obtenido de

[https://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii:](https://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii)

<https://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii>

Yii, ©. 2.-2. (s.f.). *Yii is a fast, secure, and efficient PHP framework*. Obtenido de
<https://www.yiiframework.com/>

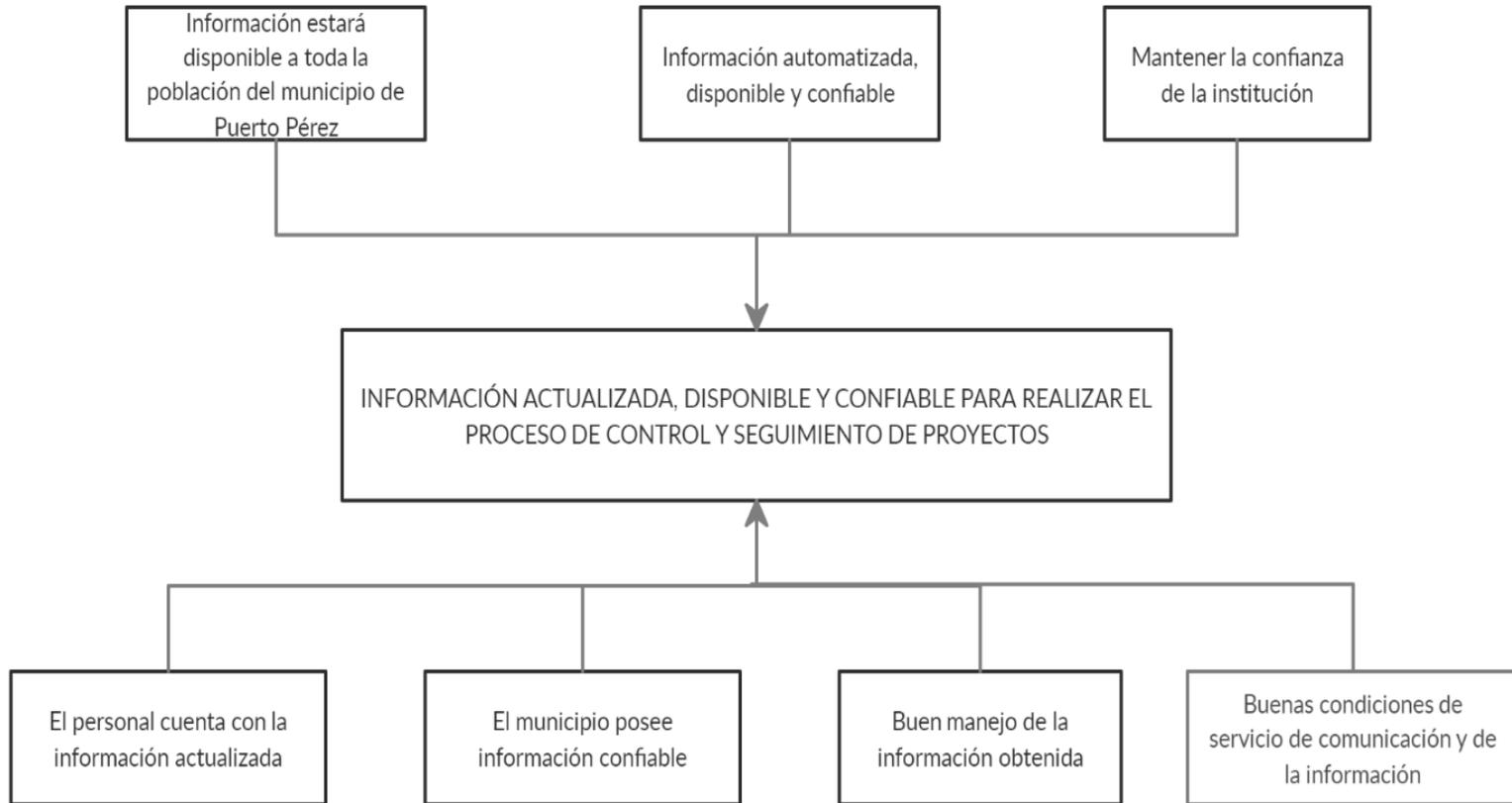


ANEXOS



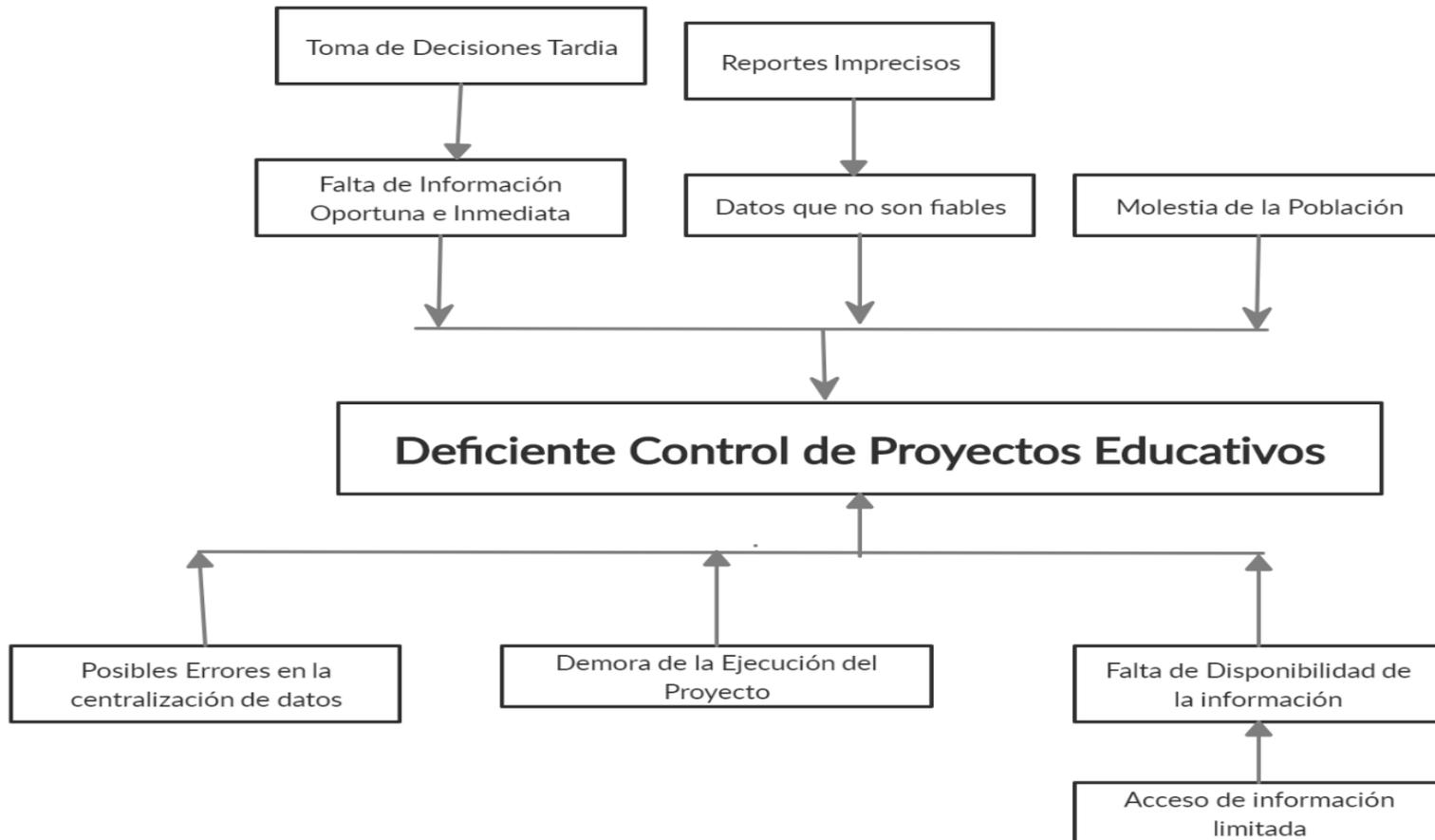
Anexo 1

ÁRBOL DE OBJETIVOS



Anexo 2

ÁRBOL DE PROBLEMAS



AVAL DE LA INSTITUCION



ORGANO LEGISLATIVO MUNICIPAL DE PUERTO PÉREZ

Cuarta Sección Municipal de la Provincia Los Andes
La Paz - Bolivia



Municipio Eco turístico

El Alto, 08 de julio del 2020

DISTRITOS:

- PUERTO PÉREZ
- ISLA SURIQUI
- KHANAPATA
- CASCACHI
- CUMANA
- ISLA QUEHUAYA

Señor.

Ing. David Carlos Mamani Quispe

**DIRECTOR DE LA CARRERA DE INGENIERÍA DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO – U.P.E.A.**

Presente. -

REF.: AVAL DE CONFORMIDAD

Señor Ingeniero.

Por intermedio de la misma, reciba usted un cordial saludo, el motivo es hacerle conocer a su distinguida autoridad, se evidencia que el universitario **Anchelho Sangalli Paco** con cedula de Identidad **79172159 LP**. Realizo y cumplió con los requerimientos de la institución para su proyecto de grado titulado **"SISTEMA DE CONTROL Y SEGUIMIENTO DE PROYECTOS EDUCATIVOS"** caso **GOBIERNO AUTÓNOMO MUNICIPAL DE PUERTO PÉREZ**, El proyecto desarrollado será de gran ayuda y aporte para nuestra institución con lo cual damos paso libre para su defensa pública, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la universidad Pública de El Alto.

Sin otro particular, me despido con las consideraciones más distinguidas.

Atentamente



JUAN WILSON MAMANI
PRBSIDENTE
GOBIERNO AUTÓNOMO MUNICIPAL
PUERTO PÉREZ 4ta. SECC. PROV. LOS ANDES



CONCEJO MUNICIPAL ECO-TURISTICO PUERTO PÉREZ
4ta. SECC. - PROV. LOS ANDES
G.A.M.E.T.P.P.
LA PAZ - BOLIVIA



H. Francisca C. Hilayo Mamani
CONCEJALA SECRETARIA MUNICIPAL
GOBIERNO AUTÓNOMO MUNICIPAL PUERTO PÉREZ
4ta. Secc. Prov. Los Andes

Edificio Alcaldía Municipal Plaza Principal Juan José Pérez
Municipio de Puerto Pérez

Anexo 4

AVAL DE TUTOR ESPECIALISTA

El Alto, 9 de julio del 2020

Señor.

Ing. Enrique Flores Baltazar

DOCENTE – TALLER II

INGENIERIA DE SISTEMAS – U.P.E.A.

Presente. -

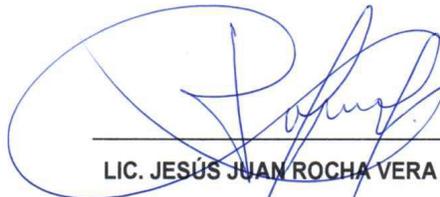
REF.: AVAL DE CONFORMIDAD

Señor Ingeniero.

Mediante la presente le comunico mi conformidad del Proyecto de Grado denominado "**SISTEMA DE CONTROL Y SEGUIMIENTO DE PROYECTOS EDUCATIVOS**" caso **GOBIERNO AUTÓNOMO MUNICIPAL DE PUERTO PÉREZ**, que propone el postulante **Univ.: Anelho Sangalli Paco con R.U. 9172150 LP.** Para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido con las consideraciones más distinguidas.

Atentamente.



LIC. JESÚS JUAN ROCHA VERA

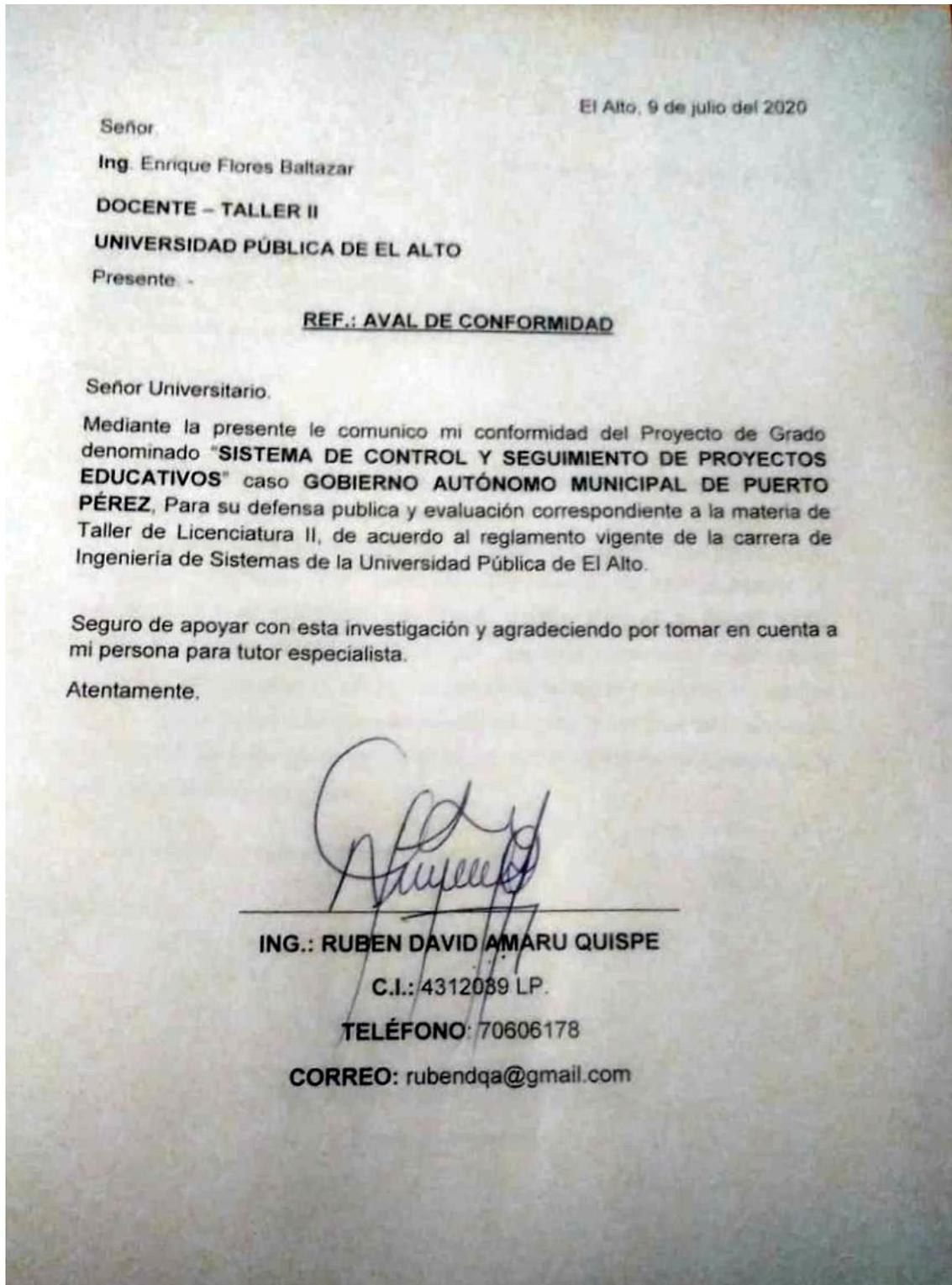
C.I.: 3343181 LP.

TELÉFONO: 68170871

CORREO: jrochavera@gmail.com

Anexo 5

AVAL DEL TUTOR REVISOR



Anexo 6

AVAL DEL TUTOR METODOLÓGICO

AVAL DE CONFORMIDAD

La Paz - El Alto, 13 de julio de 2020

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR DE CARRERA DE INGENIERÍA DE SISTEMAS

Presente. -

Ref.: AVAL DE CONFORMIDAD

Distinguido Ingeniero.

Mediante la presente tengo a bien comunicarle mi conformidad del Proyecto de Grado titulado "SISTEMA DE CONTROL Y SEGUIMIENTO DE PROYECTOS EDUCATIVOS" caso: **GOBIERNO AUTÓNOMO MUNICIPAL DE PUERTO PÉREZ**. Que propone el postulante universitario: **ANCHELHO SANGALLI PACO** con CI: **9172159 LP**. Para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:



M. Sc. Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO

MANUAL DE USUARIO



MANUAL DE USUARIO

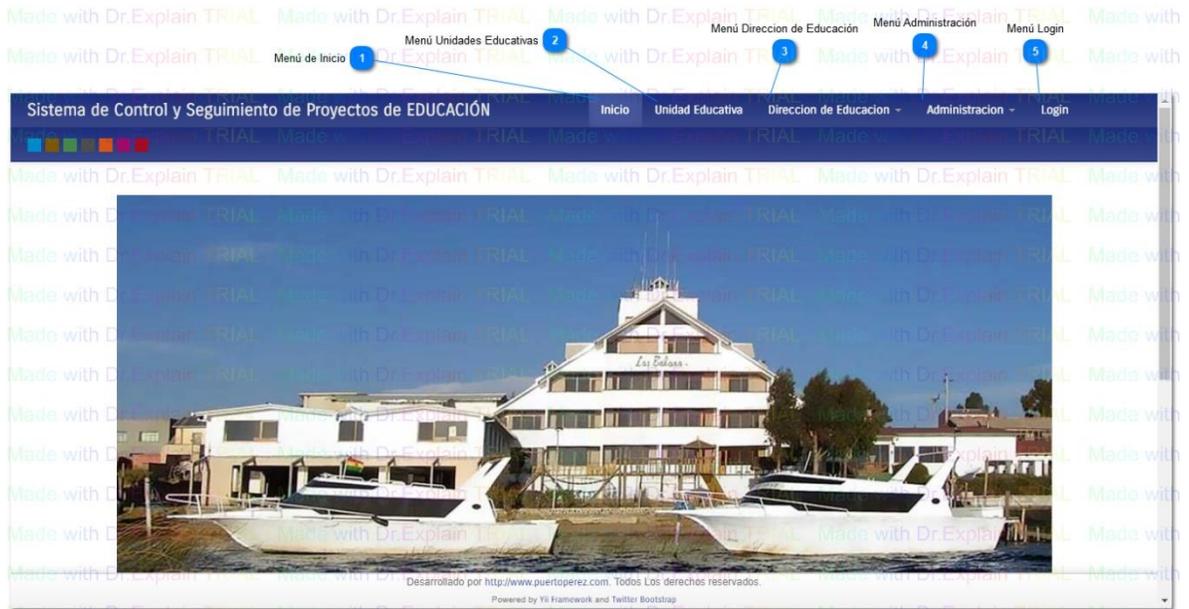
SISTEMA DE CONTROL Y SEGUIMIENTO DE PROYECTOS EDUCATIVOS

CASO: GOBIERNO AUTÓNOMO MUNICIPAL DE PUERTO PÉREZ

Manual de instrucciones

2020

Menú de Inicio



1

Menú de Inicio

Inicio

Muestra la pantalla Principal

2

Menú Unidades Educativas

Unidad Educativa

Muestra las U.E.

3

Menú Dirección de Educación

Direccion de Educacion

listado de Proyectos de U.E.

4

Menú Administración

Administracion

Muestra los Reportes administrados

5

Menú Login

Login

Inicio al Formulario de Login

2. Menú Login

Menú Login.

Formulario Login

Sistema de Control y Seguimiento de Proyectos de EDUCACIÓN Inicio Unidad Educativa Direccion de Educacion Administracion Login

Inicia » Login

Ingresa a tu cuenta

Acceso Privado

Por favor ingresa tu Usuario y Contraseña:

Campos con * son requeridos:

Usuario
admin

Contraseña
.....

Recordarme

Boton Login

Login

Desarrollado por <http://www.puertoperez.com>. Todos Los derechos reservados.

Powered by Yii Framework and Twitter Bootstrap

1

Ingresar Datos de usuario

Usuario

Ingresa los datos registrados del

2

Contraseña

Contraseña

Ingresa la contraseña del usuario o Administrador

3 Botón Login



Presiona el Botón para Iniciar Sesión

Unidad Educativa

Muestra el Menú de Unidad Educativa.

Menú Unidad Educativa



Sis Uni	Sis Infra	Nombre	Direccion	Id Dis	
50620015		UNIDAD EDUCATIVA QUEHUAYA	COMUNIDAD ISLA QUEHUAYA	DIS-004	 
50620018		ESCUELA PARITY	COMUNIDAD ISLA PARITY	DIS-004	 
50620020		UNIDAD EDUCATIVA 16 DE JULIO	ISLA SURIQUI	DIS-005	 
50620019		ESCUELA ISLA SURIQUI	ISLAS SURIQUI	DIS-005	 
50620021		ESCUELA CUYAMPAYA	COMUNIDAD CUYAMPAYA	DIS-005	 
50620022		ESCUELA PACO CHACHACOMA	PACO CHACHACOMA	DIS-005	 
50620023		ESCUELA SUPICACHI	COMUNIDAD SUPICACHI	DIS-005	 

1 Lista U.E.



Muestra el Listado de todas las U.E.

2 Crea U.E.



Crea una nueva Unidad Educativa

3

Búsqueda

Made with Dr.Explain

Puede realizar la búsqueda por nombre de U.E.

4

Ver



Muestra el listado, modificación y eliminación de la U.E.

5

Actualizar



Actualiza la U.E.

6

Eliminar

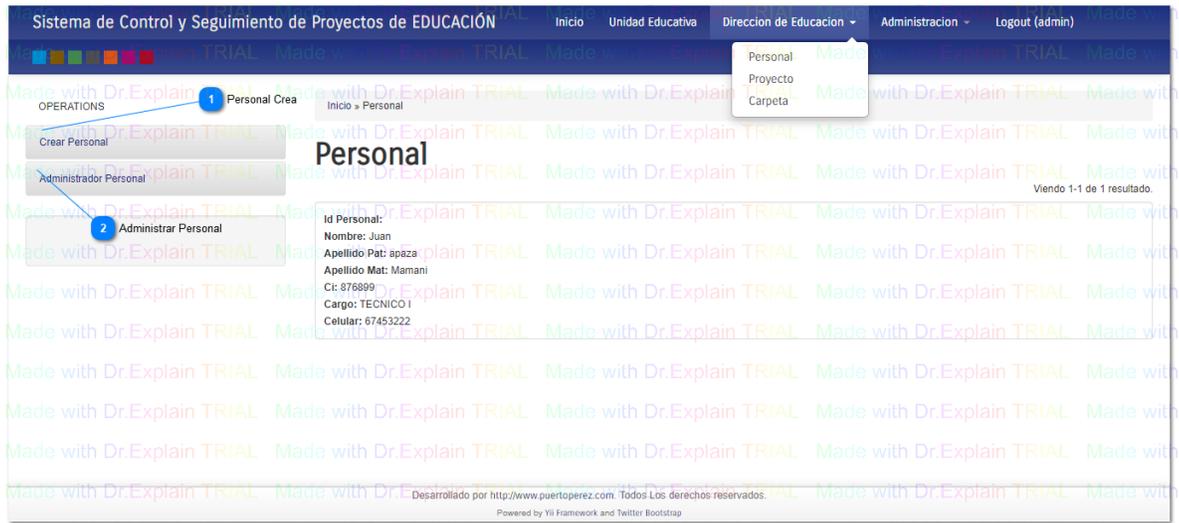


Borra la U.E.

[Dirección de Educación](#)

Menú de Dirección de Educación

Sub Menú Personal



1 Personal Crea

Crear Personal

Creas un nuevo personal para registrar

2 Administrar Personal

Administrador Personal

Administra todo el personal ya sea modificar, eliminar y actualizar.

Sub Menú Proyecto

Visualiza los datos del Sub menú Proyecto

Sub Menú Proyecto

1 Listar

Listar Proyecto

Lista todos los proyectos en curso

2 Crear

Crear Proyecto

Crea un nuevo proyecto a realizar

3 Búsqueda

Made with Dr.Explain

Realiza Búsqueda del nombre de un proyecto específico

4 Ver



Visualiza el estado de cada proyecto, se puede actualizar, modificar y eliminar

5 Actualizar



Actualiza el Proyecto

6 Eliminar



Elimina todo el proyecto

Sub Menú Carpeta

Visualiza el Contenido del Sub menú carpeta

Sub Menú Carpeta

The screenshot shows the 'Administrar Carpetas' page in a web application. The page title is 'Administrar Carpetas' and it includes a search bar and a table of folders. Numbered callouts indicate the following elements:

- 1**: Points to the 'Listar' button in the 'OPERATIONS' menu.
- 2**: Points to the 'Busqueda' (Search) input field.
- 3**: Points to the 'Ver' (View) button.
- 4**: Points to the 'Actualizar' (Update) button.
- 5**: Points to the 'Eliminar' (Delete) button.

The table below shows the data displayed in the application:

Numero Carpeta	Reg Pac	Monto Ref	Fecha Ini
C-1		25000	2020-06-01

Additional text on the page includes: 'Opcionalmente puede ingresar un operador de comparacion (<, >, <=>, >=, <=0=) al comienzo de cada uno de sus valores de busqueda para especificar como se debe hacer la comparacion.' and 'Busqueda avanzada'. The footer contains: 'Desarrollado por http://www.puertoperez.com. Todos Los derechos reservados. Powered by Yii Framework and Twitter Bootstrap'.

1

Listar

Listar Carpetas

Lista todas las Carpetas del poa.

2

Búsqueda

Made with Dr.Explain

Realiza la búsqueda, puede hacerlo por el monto referencial, numero de carpeta o fecha de inicio

3

Ver



Visualiza el estado de la carpeta

4

Actualizar



Actualiza la carpeta

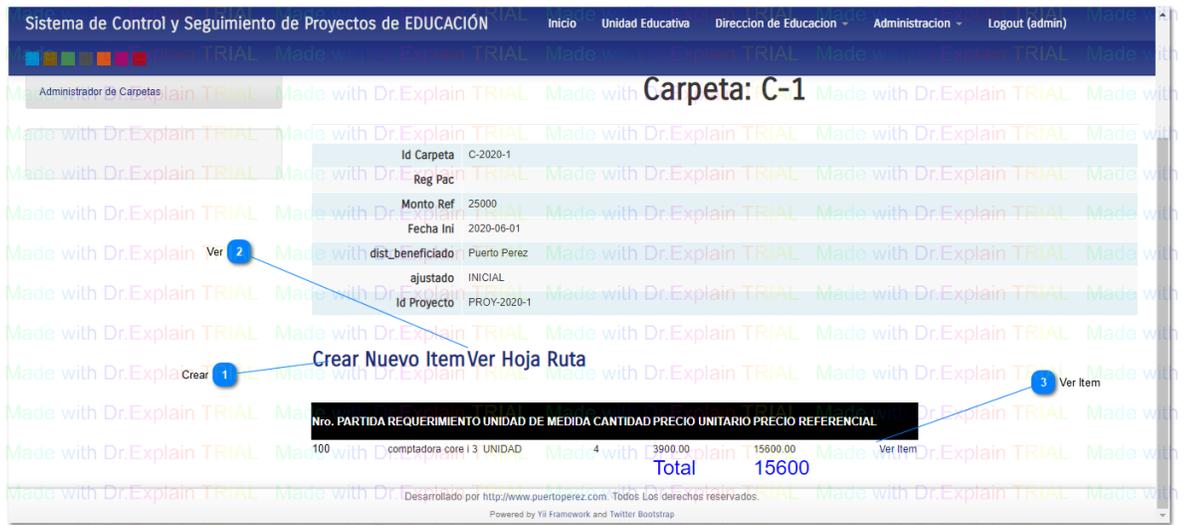
5

Eliminar



Elimina la carpeta

Ver Carpeta



1 Crear

Crear Nuevo Item

Crea un nuevo item del proyecto de acuerdo al presupuesto asignado Ejemplo:
Material de Escritorio, computadoras, etc

2 Ver

Ver Hoja Ruta

Visualiza todos los seguimientos del proyecto antes de ejecutarse

3 Ver Item

Ver Item

Visualiza todos los items agregados dentro del proyecto

Administración

Visualiza el Menú Administración.

Reportes



1 Reporte

General

Muestra el reporte general

Manual de Usuarios realizado en (Explain, s.f.)

MANUAL DE INSTALACIÓN



Sistema de Control y Seguimiento de Proyectos Educativos

Manual Técnico de Instrucciones
2020

1. Instalación de Software XAMPP

Descarga de Xampp



1 Descargar Xampp



Se realiza la descarga del servidor local XAMPP de la siguiente página <https://sourceforge.net/projects/xampp/files/>, para sistemas operativos Windows.

2 Xampp Linux



Servidor Local para Sistemas Operativos Linux

3 Xampp Mac Os



Servidor local para sistemas operativos Mac Os X

Lista Vista Elementos

The screenshot shows a table with the following columns: Nombre, Fecha de modificación, Tipo, and Tamaño. The rows contain data for 'framework.rar', 'xampp-win32-1.8.3-3-VC11-installer.exe', and 'Servidor Local Xampp'. Callouts 1-11 identify UI elements: 1 (Botón Nombre), 2 (Lista Encabezado), 3 (Botón Fecha de modificación), 4 (Botón Lista desplegable Filtro), 5 (Botón Lista desplegable Filtro), 6 (Botón Tipo), 7 (Botón Lista desplegable Filtro), 8 (Botón Lista desplegable Filtro), 9 (Botón Tamaño), 10 (Framework Yll), and 11 (Servidor Local Xampp).

Nombre	Fecha de modificación	Tipo	Tamaño
framework.rar	31/3/2020 09:20	Archivo WinRAR	5.700 KB
xampp-win32-1.8.3-3-VC11-installer.exe	27/6/2020 20:08	Aplicación	128.670 KB
Servidor Local Xampp			

1

Botón Nombre

Nombre

2

Lista Encabezado

Nombre

Fecha de modificación

Tipo

Tamaño

3

Botón Fecha de modificación

Fecha de modificación

4

Botón Lista desplegable Filtro

5 **Botón Lista desplegable Filtro**



6 **Botón Tipo**



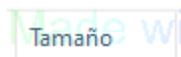
7 **Botón Lista desplegable Filtro**



8 **Botón Lista desplegable Filtro**



9 **Botón Tamaño**

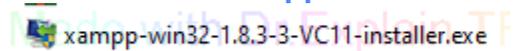


10 **Framework Yii**



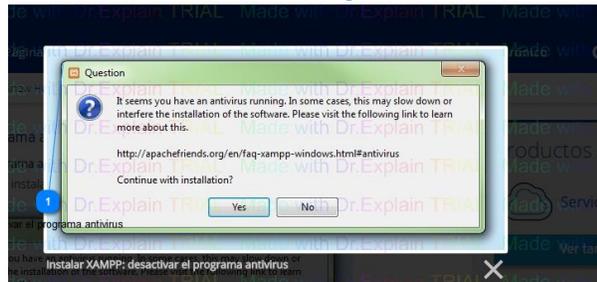
Archivo rar del framework Yii v1.1.

11 **Servidor Local Xampp**



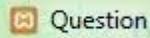
Archivo Ejecutable de Xampp.

Setup



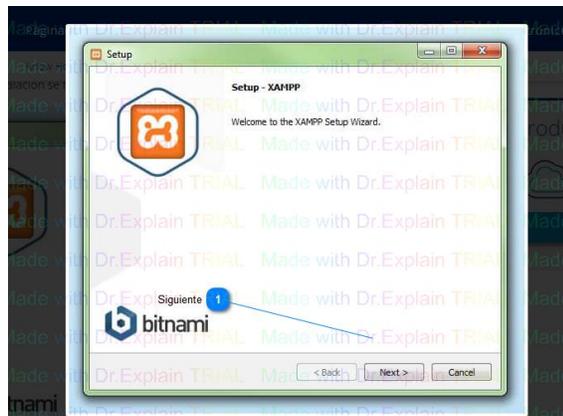
1

Desactivar el programa antivirus



Desactivar el programa antivirus hasta que todos los componentes estén instalados, ya que puede obstaculizar el proceso de instalación.

Asistente



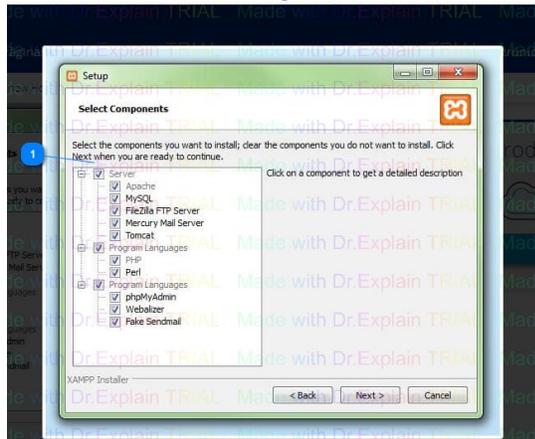
1

Siguiente



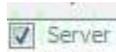
Una vez superados estos pasos, aparece la pantalla de inicio del asistente para instalar XAMPP. Para ajustar las configuraciones de la instalación se hace clic en "Next".

Selección de los componentes del software



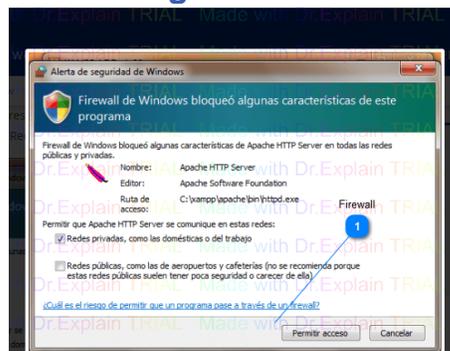
1

<TAREAS PENDIENTES : Nombre del control>



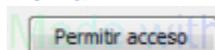
En la rúbrica “Select components” se pueden excluir de la instalación componentes aislados del paquete de software de XAMPP. Se recomienda la configuración estándar para un servidor de prueba local, con la cual se instalan todos los componentes disponibles. Confirma la selección haciendo clic en “Next”.

Configurar Firewall



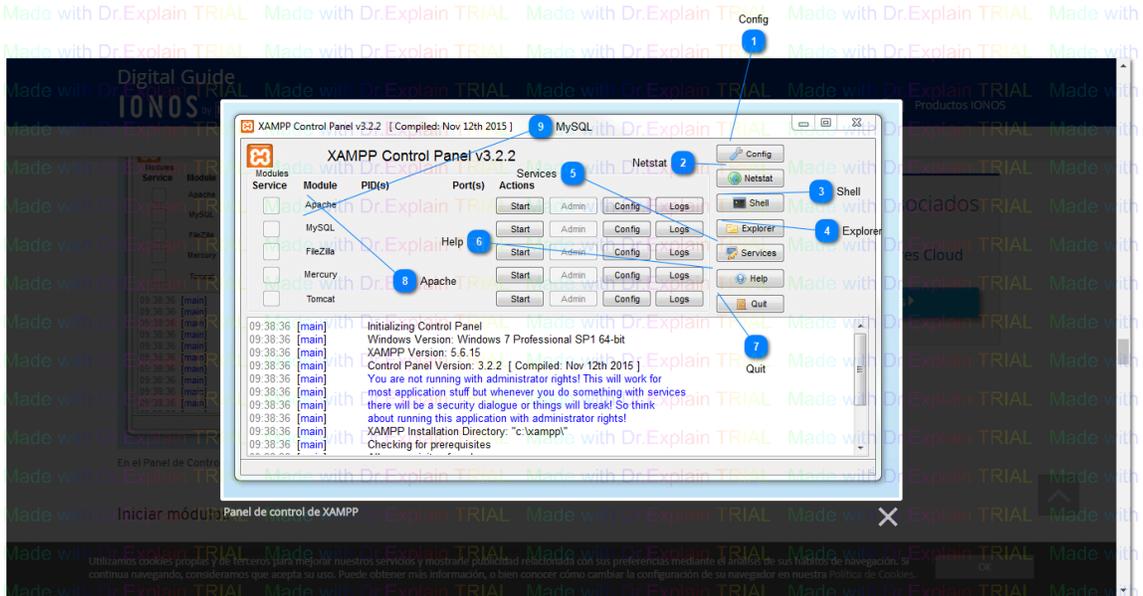
1

Firewall



Durante el proceso de instalación es frecuente que el asistente avise del bloqueo de Firewall. En la ventana de diálogo puedes marcar las casillas correspondientes para permitir la comunicación del servidor Apache en una red privada o en una red de trabajo. Recuerda que no se recomienda usarlo en una red pública.

Panel de control de XAMPP



1

Config



Para configurar XAMPP así como otros componentes aislados.

2

Netstat



Muestra todos los procesos en funcionamiento en el ordenador local

3

Shell



Lanza una ventana de comandos UNIX

4

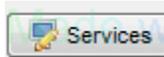
Explorer



Lanza una ventana de comandos UNIXI

5

Services



Muestra todos los servicios en funcionamiento

6

Help



incluye enlaces a foros de usuarios

7

Quit



se usar para salir del panel de control

8

Apache



Damos clic en Start para que inicie el servidor Apache.

9

MySQL



Da inicio a los servicios de Mysql

Menú de Xampp

XAMPP for Windows

Bienvenido a XAMPP para Windows!

Felicitaciones:
XAMPP se instaló con éxito en su ordenador!

Ahora se puede empezar a trabajar. :) Primero por favor pulse encima de «Estado» en la parte izquierda. De esta manera tendrá una visión de que es lo que funciona ya. Algunas funciones estarán desactivadas. Es intencionado. Son funciones, que no funcionan en todas partes o eventualmente podrían ocasionar problemas.

Atención: XAMPP fue modificado a partir de la versión 1.4.x a una administración de paquete único. Existen los siguientes paquetes/Addons:

- XAMPP paquete básico
- XAMPP Perl addon
- XAMPP Tomcat addon
- XAMPP Cocoon addon
- XAMPP Python addon (developer version)

Y en un futuro:

- XAMPP Utility addon (Accesorio pero aún inactivo)
- XAMPP Server addon (otros servidores aún inactivos)
- XAMPP Other addon (otras cosas útiles aún inactivas)

Por favor "instalar" los paquetes adicionales, que aún necesitels, simplemente a continuación. Después de subirlos con éxito, por favor siempre accionar "setup_xampp.bat", para inicializar nuevamente XAMPP. A Bueno, las versiones Instalador de los Addons individuales funcionan sólo si el paquete básico XAMPP también fue montado a partir de una versión instalador.

Para el soporte OpenSSL utilice por favor el certificado de chequeo con la URL <https://127.0.0.1> ó <https://localhost>

Os deseamos mucha diversión, Kay Vogelgesang + Kai 'Oswald' Seidler

Install applications on XAMPP using BitNami

Apache Friends and BitNami are cooperating to make dozens of open source applications available on XAMPP, for free. BitNami-packaged applications include Wordpress, Drupal, Joomla! and dozens of others and can be deployed with one-click installers. Visit the [BitNami XAMPP page](#) for details on the currently available apps.

©2009-2020
 ...APACHE FRIENDS...

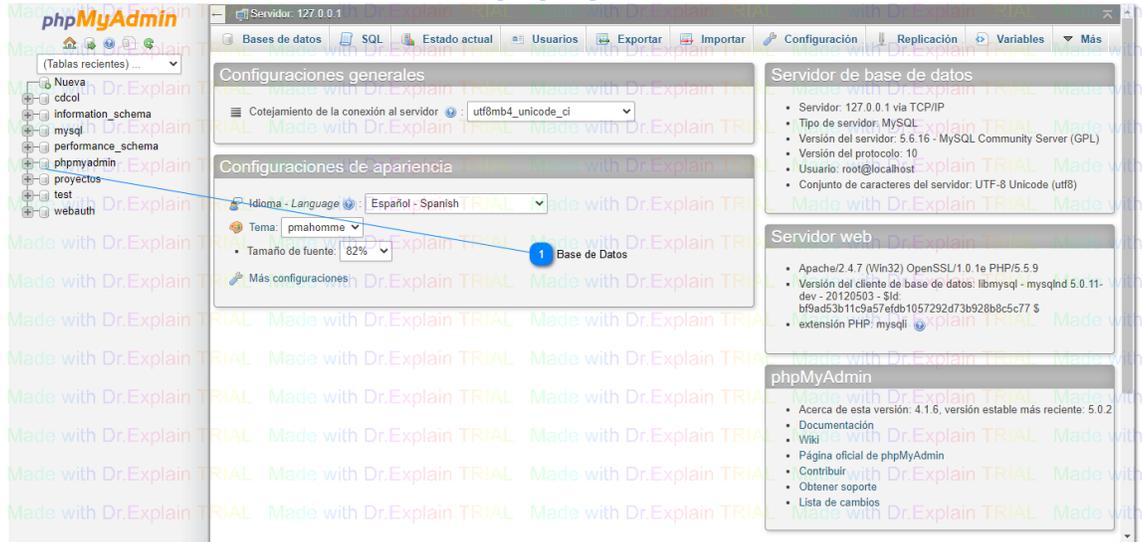
1

phpMyAdmin

phpMyAdmin

Se da inicio al phpMyAdmin para el inicio y verificación de la base de datos en el servidor local

Menú phpMyAdmin



1

Base de Datos

proyectos

Base de datos Proyecto, donde se encuentra todos los registros del sistema de control y seguimiento de proyectos de las unidades educativas

Descargar Yii



1

Yii

1.1

Descargar Yii v1.1.

