

INFORME FINAL DE INVESTIGACIÓN

**DESARROLLO DE UNA PLATAFORMA
INFORMÁTICA PARA FORMULACIÓN,
MONITOREO Y EVALUACIÓN DE PROYECTOS
DE INVESTIGACIÓN. CASO PRÁCTICO,
PROGRAMA DE INVESTIGACIÓN DE
ITCA FEPADE.**

DOCENTE INVESTIGADOR PRINCIPAL:
ING. EDGARDO ANTONIO CLAROS QUINTANILLA

CENTRO REGIONAL MEGATEC LA UNIÓN

ENERO 2018

INFORME FINAL DE INVESTIGACIÓN

**DESARROLLO DE UNA PLATAFORMA
INFORMÁTICA PARA FORMULACIÓN,
MONITOREO Y EVALUACIÓN DE PROYECTOS
DE INVESTIGACIÓN. CASO PRÁCTICO,
PROGRAMA DE INVESTIGACIÓN DE
ITCA FEPADE.**

DOCENTE INVESTIGADOR PRINCIPAL:
ING. EDGARDO ANTONIO CLAROS QUINTANILLA

CENTRO REGIONAL MEGATEC LA UNIÓN

ENERO 2018

Rectora

Licda. Elsy Escolar SantoDomingo

Vicerrector Académico

Ing. Carlos Alberto Arriola Martínez

Vicerrectora Técnica Administrativa

Inga. Frineé Violeta Castillo

**Dirección de Investigación
y Proyección Social**

Ing. Mario Wilfredo Montes, Director

Ing. David Emmanuel Ágreda

Sra. Edith Aracely Cardoza

Director Centro Regional La Unión

Lic. Luis Ángel Ramírez Benítez

658.404 028 5

C613d Claros Quintanilla, Edgardo Antonio, 1985-

sv Desarrollo de una plataforma informática para
formulación, monitoreo y evaluación de proyectos de
investigación : Caso práctico programa de
investigación de ITCA-FEPADE. / Edgardo Antonio
Claros Quintanilla -- 1ª ed. -- Santa Tecla, La Libertad,
El Salv. : ITCA Editores, 2018.

40 p. : 28 cm.

ISBN 978-99961-50-77-7 (impreso)

1. Administración de proyectos. 2. Programas
integrados para computador. 3. Elaboración de
proyectos. I. Título.

Autor

Ing. Edgardo Antonio Claros Quintanilla

Docente de Apoyo

Ing. Enrique Amadeo Sorto Cabrera

Tiraje: 13 ejemplares

Año 2018

Este documento técnico es una publicación de la Escuela Especializada en Ingeniería ITCA-FEPADE; tiene el propósito de difundir la Ciencia, la Tecnología y la Innovación CTI, entre la comunidad académica y el sector empresarial, como un aporte al desarrollo del país. Este informe de investigación no puede ser reproducido o publicado parcial o totalmente sin previa autorización de la Escuela Especializada en Ingeniería ITCA-FEPADE. Para referirse a este documento se debe citar al autor. El contenido de este informe es responsabilidad exclusiva de los autores.

Escuela Especializada en Ingeniería ITCA-FEPADE
Km 11.5 carretera a Santa Tecla, La Libertad, El Salvador, Centro América

Sitio web: www.itca.edu.sv

Tel: (503)2132-7423

Fax: (503)2132-7599

CONTENIDO

1.	INTRODUCCIÓN	4
2.	PLANTEAMIENTO DEL PROBLEMA.....	4
2.1.	DEFINICIÓN DEL PROBLEMA.....	5
2.2.	ANTECEDENTES	5
2.3.	JUSTIFICACIÓN.....	5
3.	OBJETIVOS	6
3.1.	OBJETIVO GENERAL	6
3.2.	OBJETIVOS ESPECÍFICOS	6
4.	HIPÓTESIS	7
5.	MARCO TEÓRICO	7
5.1.	IMPORTANCIA Y FUNCIÓN DEL MONITOREO Y EVALUACIÓN EN LOS PROYECTOS	7
5.2.	SISTEMAS DE INFORMACIÓN.....	10
5.3.	APLICACIONES WEB.....	10
5.4.	EL ENFOQUE DE GESTIÓN ÁGIL DE PROYECTOS.....	11
5.5.	EL MODELO PROGRAMACIÓN EXTREMA PARA GESTIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE.....	15
5.6.	EL LENGUAJE DE PROGRAMACIÓN PHP.....	17
5.7.	BASES DE DATOS RELACIONALES.....	19
5.8.	LA PROGRAMACIÓN ORIENTADA A OBJETOS.....	22
5.9.	EL PATRÓN DE DESARROLLO MODELO-VISTA-CONTROLADOR.....	25
6.	METODOLOGÍA DE LA INVESTIGACIÓN	26
6.1.	TIPO DE INVESTIGACIÓN.....	26
6.2.	MÉTODO DE INVESTIGACIÓN	27
	POBLACIÓN Y MUESTRA	27
	TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN	27
6.3.	FASES DE DESARROLLO DE LA INVESTIGACIÓN	27
7.	RESULTADOS.....	28
8.	CONCLUSIONES	33
9.	RECOMENDACIONES	34
10.	GLOSARIO	34
11.	BIBLIOGRAFÍA.....	35
12.	ANEXOS.....	37
12.1.	ANEXO 1: GUÍA DE ENTREVISTA UTILIZADA.....	37
12.2.	ANEXO 2: PILA DEL PRODUCTO DEL PROYECTO.....	39

1. INTRODUCCIÓN

Los beneficios que brinda la automatización de procesos por medio de un software son diversos, siendo uno de los que más impacto genera, la capacidad de optimización y centralización de las operaciones.

ITCA-FEPADE es una institución que se preocupa por el desarrollo de tecnología útil para la solución de necesidades marcadas en nuestra sociedad y también dentro de la institución misma. En esta oportunidad se ha trabajado en la elaboración de una herramienta de software que beneficiará directamente al Programa Institucional de Investigación. Se ha planteado el desarrollo de una plataforma diseñada para funcionar en un entorno web a la cual se puede tener acceso mediante una conexión de Internet y un navegador.

Este software permitirá la formulación, el monitoreo y la evaluación de los proyectos del Programa Institucional de Investigación, desde las fases tempranas, donde se realiza el proceso de presentación de propuestas, hasta la presentación del informe de cierre y costeo. Durante las actividades de monitoreo de los proyectos pone a disposición diferentes herramientas que permitirán conocer en todo momento el estado en la ejecución del proyecto; esto será efectivo mediante los reportes mensuales que los investigadores realizarán en la plataforma.

Con el desarrollo de este proyecto se está entregando un producto que servirá para tecnificar los diferentes procesos del Programa de Investigación de ITCA-FEPADE.

2. PLANTEAMIENTO DEL PROBLEMA

La Escuela Especializada en ingeniería ITCA-FEPADE como Institución de Educación Superior y en atención a la Ley de Educación Superior, impulsa de una manera sistemática el Programa Institucional de Investigación y Desarrollo en Ciencia, Tecnología e Innovación CTI. La Sede Central en Santa Tecla, así como cada uno de los centros regionales de Santa Ana, San Miguel, MEGATEC Zacatecoluca y MEGATEC La Unión, integran y participan en este Programa.

El proceso de seguimiento de los proyectos de investigación que involucra las etapas de propuestas, validación y control de desarrollo en la Sede Central y los Centros Regionales, se realiza mediante un monitoreo continuo por medio de correos institucionales, reuniones de avance y otros métodos manuales.

Ocasionalmente realizar reuniones de avance puede ser dificultoso para docentes investigadores de las regionales por la distancia, compromisos académicos, etc.

En este sentido formular, monitorear y evaluar cada uno de los proyectos es una actividad que presenta dificultades en el proceso de seguimiento y control.

2.1. DEFINICIÓN DEL PROBLEMA

¿En qué manera la implementación de una plataforma para el monitoreo y evaluación permitirá mejorar el seguimiento de los proyectos del programa de investigación en ITCA-FEPADE?.

2.2. ANTECEDENTES

La Escuela Especializada en Ingeniería ITCA-FEPADE es una Institución de Educación Superior que ofrece una educación integral; permite desarrollar en los estudiantes una actitud de servicio y compromiso hacia su entorno social, mediante la motivación de su espíritu emprendedor, su capacidad de liderazgo y aplicación de valores morales. Brinda la oportunidad de aprender-haciendo, aplicando directamente los conocimientos que se adquieren en el aula, en un entorno real, talleres y laboratorios de cada especialidad, así como prácticas en empresas.

La Escuela Especializada en Ingeniería ITCA-FEPADE tiene un compromiso con la calidad en la formación de técnicos e ingenieros integrales y competentes en áreas tecnológicas y científicas que contribuyan a mejorar la competitividad del país. Se pretende consolidar un liderazgo en la educación tecnológica a nivel nacional y regional, y además se realizan acciones para que la oferta educativa sea pertinente a la demanda de los diferentes mercados globalizados.

En el año 2002 se dieron los pasos iniciales para desarrollar una cultura de investigación en ITCA-FEPADE. En el año 2003 se impulsó de manera sistemática un Programa de Investigación Aplicada y se conformó el Departamento de Investigación y Proyección Social, cuya misión fue impulsar y desarrollar progresivamente la cultura de la investigación aplicada y la proyección social, vinculando estos programas con las actividades de docencia.

A partir del año 2009 se creó la Dirección de Investigación y Proyección Social con el objetivo de planificar, asesorar y coordinar el Programa de Investigación y Desarrollo en Ciencia, Tecnología e Innovación CTI en las 5 sedes de ITCA-FEPADE. Esta Dirección vela por la buena gestión y administración, así como por el desarrollo eficiente de los proyectos de investigación; brinda entre otros, asesoría metodológica y apoyo logístico para el desarrollo de las diferentes fases del Programa. La Dirección de Investigación y Proyección Social vela además por el fortalecimiento metodológico y técnico del cuerpo de docentes investigadores de ITCA-FEPADE.

2.3. JUSTIFICACIÓN

Para optimizar el seguimiento de los proyectos de investigación institucionales, la implementación de un software en la web es una alternativa que permitirá dar fluidez al proceso desde la etapa de propuesta hasta el cierre del mismo.

Es por lo anterior que el software incorporará elementos técnicos para el control de cada fase de los proyectos de investigación. En este sentido, para el desarrollo del proyecto se han dividido las funcionalidades esperadas en tres grupos principales de acuerdo a su naturaleza, las cuales podemos identificar de la siguiente forma:

- **Formulación y aprobación de proyectos de investigación:** Este grupo se refiere a las funcionalidades que incluyen la presentación, anulación y aprobación de propuestas de investigación, así como las observaciones pertinentes en cada documento. Es importante mencionar que en este grupo se consideran las funcionalidades que están relacionadas a la elaboración de los presupuestos financieros, cotizaciones y especificaciones técnicas.
- **Monitoreo, seguimiento y evaluación de proyectos de investigación:** En este grupo de funcionalidades se encuentran las relacionadas con el registro de horario de docentes investigadores principales y de apoyo, especificando el tiempo dedicado a las labores de investigación. De igual forma están contemplados las tareas de reportes de avance mensual, información sobre los gastos realizados, el presupuesto consumido y restante y consolidado del estado actual del proyecto en tiempo real en el momento que se solicite, detalle de las actividades realizadas de acuerdo a los objetivos del proyecto y de forma general, así mismo, la presentación y revisión de informes finales, artículos científicos para los proyectos exitosos e informes de cierre y costeo.
- **Misceláneos:** Este grupo incluye las funcionalidades relacionadas con los perfiles de usuario que tendrán acceso en el software, así mismo las funcionalidades que están vinculadas a la gestión de información multimedia también se incluyen en este grupo, de igual forma la creación de un directorio de documentos oficiales vigentes donde los investigadores puedan consultar las diferentes guías, normativos y otros elementos de interés relacionados con el programa de investigación institucional.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Desarrollar una plataforma informática para monitoreo y evaluación de los proyectos del programa de investigación en ITCA-FEPADE

3.2. OBJETIVOS ESPECÍFICOS

1. Definir los requisitos de la plataforma considerando los lineamientos del programa institucional de investigación.
2. Elaborar el diseño lógico de la plataforma y sus componentes tecnológicos.
3. Programar la plataforma utilizando tecnologías de software libre.
4. Implementar la plataforma en la institución para optimizar el proceso de monitoreo y evaluación de proyectos de investigación.

4. HIPÓTESIS

La implementación de una plataforma informática permitirá optimizar los procesos de monitoreo y evaluación de proyectos del programa de investigación en ITCA-FEPADE.

5. MARCO TEÓRICO

5.1. IMPORTANCIA Y FUNCIÓN DEL MONITOREO Y EVALUACIÓN EN LOS PROYECTOS

Es bien sabido que la implementación de un programa o proyecto social o de desarrollo, de atención humanitaria, de fortalecimiento de capacidades o de incidencia no garantiza por sí mismo la consecución de los objetivos planteados ni un impacto positivo en la vida de las personas. Las buenas intenciones, los programas o proyectos de por sí solos no garantizan resultados positivos.

Para aumentar las posibilidades de efectividad de cualquier programa o proyecto social es necesario contar con un diseño y una planeación cuidadosos, Monitoreos y Evaluaciones (MyE) apropiados. Si bien la planeación adecuada es inalterable: si nuestro proyecto no está funcionando o si las circunstancias han cambiado, los planes deben cambiar también. El MyE permite darnos cuenta si los planes están fallando o si hay oportunidad de analizar y tomar decisiones sobre el monitoreo y la evaluación genera información para mejorar la acción, reorientarla o hacer una planeación futura mucho más asertiva y efectiva.

Sin un monitoreo y una evaluación efectivos, no encaminando en la dirección correcta, si el progreso o los cambios logrados se deben a nuestra planeación a futuro (Programa de Naciones Unidas para el Desarrollo [PNUD], 2009).

El monitoreo y la evaluación sirven para:

- **Mejorar el desempeño:** permiten la detección de problemas de un proyecto en etapas iniciales de la implementación, y de esta forma reducen las posibilidades de fracaso y ahorran recursos y tiempo al posibilitar la adopción oportuna de medidas correctivas.
- **Fortalecer la toma de decisiones y la planeación:** permiten aprender de los logros y las de decisiones y al diseño de iniciativas presentes o futuras (para suspender, reformular o multiplicar un proyecto). Asimismo, permiten aproximar los proyectos a la realidad y a los resultados alcanzables.
- **Contribuir al aprendizaje:** permiten a la organización y a los involucrados en el proceso aprender de la experiencia, con lo cual no sólo se mejoran las acciones posteriores sino que se contribuye al aprendizaje institucional y al aprendizaje social.
- **Avanzar hacia el empoderamiento:** el monitoreo y la evaluación permiten fortalecer las capacidades de la población involucrada, de los socios locales, del equipo encargado de la ejecución y de la misma organización, ya que favorece el conocimiento de la problemática, de los procesos y de las estrategias.
- **Mejorar la rendición de cuentas:** permiten a las organizaciones responsables informar de los recursos y el impacto de los proyectos. Esto puede también fomentar el apoyo público y político de las iniciativas.

- **Contribuir al conocimiento general:** el análisis de la evaluación permite compartir con otros, difundir los aprendizajes y las mejores prácticas, contribuyendo al conocimiento general de los procesos y proyectos de desarrollo. Esto puede servir tanto a la misma organización como a otras organizaciones o instituciones gubernamentales y académicas.
- **Contribuir al conocimiento general:** el análisis de la evaluación permite compartir con otros, difundir los aprendizajes y las mejores prácticas, contribuyendo al conocimiento general de los procesos y proyectos de desarrollo. Esto puede servir tanto a la misma organización como a otras organizaciones o instituciones gubernamentales y académicas.

Tabla 1: Diferencias ente el monitoreo y la evaluación en los proyectos

	Monitoreo	Evaluación
Función	Información para la gestión del proyecto.	Valoración más profunda enfocada en el aprendizaje y la rendición de cuentas.
Enfoque	Revisión del progreso del proyecto.	Enfoque en los resultados, objetivos e impacto más amplio y a largo plazo.
Insumos	Levantamiento de datos/información.	Información proviene de la recolección de datos para el monitoreo, para la línea base, etc.
Tiempo	Continuo, durante toda la implementación del proyecto.	En momentos específicos del proyecto
Responsable	Equipo responsable del proyecto.	Evaluadores internos o externos al proyecto (o a la organización ejecutora).
Presentación	Presentación directa con base en formatos estandarizados.	Presentación menos estandarizada y enfocada en menos temas, pero analizados con mayor profundidad.
Responsable	Entre el equipo del proyecto y los usuarios.	Más amplia: partes involucradas, donantes, opinión pública, etc.

Monitoreo y Evaluación en el Ciclo del Proyecto

Analicemos el papel del monitoreo y la evaluación en el proceso de diseño y desarrollo. Revisamos las definiciones de y los principios básicos de la formulación y la planeación del proyecto con base principalmente en la metodología del marco lógico. Si bien este tema por sí solo conformaría un manual, es necesario mencionarlo para poder comprender todo el proceso de MyE en cada nivel de la lógica de intervención del proyecto.

El proceso de diseño y desarrollo de un proyecto puede resumirse de manera sencilla en el Ciclo del Proyecto. Este ciclo consiste en una serie de pasos estrechamente vinculados entre sí y acomodados por el orden que se desarrollan. La identificación del problema, la planeación del proyecto, su ejecución y seguimiento, y la evaluación que normalmente alimenta el análisis y la identificación de futuros proyectos (ver Figura 1).



Figura 1: MyE en el ciclo del proyecto.

En el ciclo del proyecto, el MyE se debe considerar desde la etapa de diseño y formulación, planeando los componentes del proyecto de tal forma que se tomen en cuenta, se faciliten y garanticen procesos de MyE. La etapa de planeación y diseño es crucial para todo el desarrollo del proyecto, pues sistema de MyE, estableciendo claramente la información disponible antes de la intervención (línea base) y determinando las necesidades de información (a través de la formulación de indicadores) que permitan ir midiendo avances, efectividad e impacto.

Antes de la ejecución del proyecto se define el plan de MyE, proponiendo de manera específica cómo se deberán organizar las actividades, responsabilidades y recursos para realizar el monitoreo durante el desarrollo del proyecto.

Este monitoreo se realizará periódica y sistemáticamente durante todo el tiempo de implementación del proyecto. Esto implica el seguimiento de las actividades y revisión de los objetivos, así como la recolección, sistematización y análisis de datos. Posteriormente, se lleva a cabo el proceso de evaluación, como una etapa misma del ciclo del proyecto. La evaluación producirá información que es fundamental para la rendición de cuentas, la implementación de recomendaciones y mejorar el desempeño. Estos aprendizajes pueden ser valiosos tanto para determinar las necesidades de reformulación del mismo proyecto o para ser considerados en la formulación de otros proyectos.

5.2. SISTEMAS DE INFORMACIÓN

Los sistemas de Información dan soporte a las operaciones empresariales, la gestión y la toma de decisiones, proporcionando a las personas la información que necesitan mediante el uso de las tecnologías de la información. Las empresas y, en general, cualquier organización, los utilizan como un elemento estratégico con el que innovar, competir y alcanzar sus objetivos en un entorno globalizado. Los sistemas de información integran personas, procesos, datos y tecnología, y van más allá de los umbrales de la organización, para colaborar de formas más eficientes con proveedores, distribuidores y clientes.

La especialidad de Sistemas de Información del Grado en Ingeniería Informática capacita al estudiante para comprender los procesos operativos y la gestión que llevan a cabo las personas en las organizaciones. Asimismo, lo familiariza con los diferentes tipos de herramientas tecnológicas que se han ido estandarizando recientemente para construir un sistema de información, desde la gestión de relaciones con los clientes o la cadena de suministro hasta el comercio electrónico pasando por la gestión de procesos internos, la inteligencia de negocio y gestión del conocimiento o la ayuda a la toma de decisiones.

Por lo tanto, el titulado será capaz de actuar de puente entre las necesidades de gestión y las posibilidades que la tecnología ofrece; es decir, de analizar los requisitos de la organización y diseñar soluciones eligiendo, adaptando e integrando las herramientas disponibles más adecuadas. Igualmente, será capaz de identificar las oportunidades de mejora de procesos y de introducir innovaciones, facilitando que la organización utilice sus sistemas de información para competir estratégicamente.¹

5.3. APLICACIONES WEB

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente que servía como interfaz de usuario que tenía que ser instalado por separado en cada computadora personal de cada usuario. El cliente realizaba peticiones a otro programa —el servidor— que le daba respuesta. Una mejora en el servidor, como parte de la aplicación, requería normalmente una mejora de los clientes instalados en cada computadora personal, añadiendo un coste de soporte técnico y disminuyendo la productividad.

A diferencia de lo anterior, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, soportados por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de plugins tales como JavaScript, Java, Flash, etc., para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web. Es

¹ Los Sistemas de Información: <https://www.fib.upc.edu/es/estudios/grados/grado-en-ingenieria-informatica/plan-de-estudios/especialidades/sistemas-de-informacion>, Consultado el 12-12-2017

importante conocer que, de acuerdo a la ingeniería de software, se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. En otras palabras, es un programa que se codifica en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador.²

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los correos web, wikis, blogs, tiendas en línea y la propia Wikipedia que son ejemplos bastante conocidos de aplicaciones web.

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

5.4. EL ENFOQUE DE GESTIÓN ÁGIL DE PROYECTOS

La Gestión de Proyectos de Software

El desarrollo de software es una actividad compleja y reciente, que ha generado su conocimiento en un periodo muy breve, en comparación con otras actividades profesionales: desde la aparición de máquinas que para ser útiles necesitaban ser programadas.



Figura 2: Grabador de discos Ampex, popular en la época donde surgen los primeros programas.

² Luján Mora, S. (2002) (1ra ed.). Alicante: Editorial Club Universitario. Tomado de <http://hdl.handle.net/10045/16995>

La aparición de componentes que cada dos años doblan la capacidad de sus antecesores [ley de Moore] nos ha rodeado en menos de cuatro décadas de máquinas capaces de procesar miles de millones de operaciones por segundo (MTOPS).

En 1946 ENIAC ocupaba una superficie de 160 m², pesaba 30 toneladas, y ofrecía una capacidad de proceso de 30.000 instrucciones por segundo. En 2002 El microprocesador Pentium IV a 2 Ghz ocupaba una superficie de 217 mm² y tenía una capacidad de proceso de 5.300 MTOPS (“Millions of theoretical operations per second”).

En la actualidad son cuatro los factores que imprimen un ritmo acelerado a la industria del hardware.

De ellos, tres son consecuencia de la ley de Moore:

- Incremento constante de la capacidad de operación.
- Miniaturización.
- Reducción de costes para la producción de hardware.

Y a éstos se ha sumado en la última década el avance de las comunicaciones entre sistemas.

La consecuencia es obvia: ordenadores potentes, que pueden llevarse en el bolsillo y en permanente conexión con grandes sistemas, redes de comunicación públicas, sistemas de localización GPS, etc.

Estas cuatro líneas de avance han extendido el ámbito de aplicación del hardware, e incrementado al mismo ritmo exponencial la complejidad de los sistemas en los que se integra. Los ordenadores ya no son máquinas útiles sólo para la banca o el ejército. Se encuentran presentes en todos los ámbitos, por su capacidad de proceso y de comunicación pueden ofrecer soluciones a sistemas cada vez más complejos. Este es el escenario creado por la industria del hardware, y que en las tres últimas décadas ha implicado a los desarrolladores de software en retos a los que no han respondido con solvencia.

Crisis del Software

Este problema se identificó por primera vez en 1968 (Bauer, Bolliet, & Helms, 1968), año en el que la organización OTAN celebró la primera conferencia sobre desarrollo de software, y en la que se acuñó el término “crisis del software” para definir a los problemas que surgían en el desarrollo de sistemas de software, cuyos proyectos siempre terminaban tarde, desbordando los presupuestos y con problemas de funcionamiento. También se acuñó el término “ingeniería del software” para describir el conjunto de conocimientos que existían en aquel estado inicial.

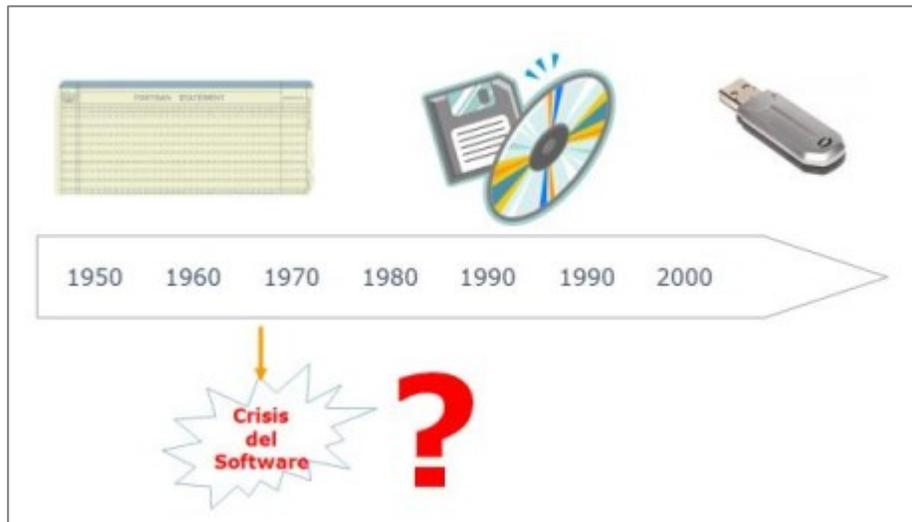


Figura 3: Cronología de los avances tecnológicos relacionados al desarrollo del software.

Cuando en los 60 y sobre todo en los 70 y 80 empezaron a hacerse habituales los ordenadores, surgieron los primeros “héroes,” que sin más información que los manuales del operativo y el lenguaje de programación, se remangaban delante del teclado para desarrollar las primeras aplicaciones.

Hasta los 70 los ordenadores fueron máquinas vanguardistas y excesivamente caras. El ejército y la banca eran los únicos sectores que se las permitían, y fueron los militares los primeros escarmentados, de proyectos en los que el software siempre llegaba tarde, mal y nunca; con demasiados errores y desbordando todas agendas previstas.

Algunas referencias útiles para comprender cuáles eran los conocimientos estables para el desarrollo de software en 1968 son:

- En 1962 se publicó el primer algoritmo para búsquedas binarias (Iverson, 1962).
- C. Böhm y G. Jacopini publicaron en 1966 el documento que creaba una fundación para la eliminación de “GoTo” y la creación de la programación estructurada (Böhm & Jacopini, 1966)
- En 1968 los programadores se debatían entre el uso de la sentencia GoTo, y la nueva idea de programación estructurada; ese era el caldo de cultivo en el que Edsger Dijkstra escribió su famosa carta “GoTo Statement Considered Harmful” en 1968 (Dijkstra, 1968).
- La primera publicación sobre programación estructurada no vio la luz hasta 1974, publicada por Larry Constantine, Glenford Myers y Wayne Stevens (Stevens, Myers, & Constantine, 1974).
- El primer libro sobre métrica de software fue publicado en 1976 por Tom Gilb (Gilb, 1976).
- El primero sobre análisis de requisitos apareció en 1976 (Bell & Thayer, 1976)

Gestión de proyectos Predictiva

Los proyectos han existido siempre. Cualquier trabajo para desarrollar algo único es un proyecto, pero la gestión de proyectos es una disciplina relativamente reciente que comenzó a forjarse en los años sesenta.

La necesidad de su profesionalización surgió en el ámbito militar. En los 50, el desarrollo de complejos sistemas militares, requería coordinar el trabajo conjunto de equipos y disciplinas diferentes, en la construcción de sistemas únicos.

Bernard Schriever, arquitecto del desarrollo de misiles balísticos Polaris, es considerado el padre de la gestión de proyectos, por la introducción del concepto de “conurrencia”, para integrar todos los elementos del plan del proyecto en un solo programa y presupuesto.

El objetivo de la conurrencia era ejecutar las diferentes actividades de forma simultánea, y no secuencialmente, y al aplicarla en los proyectos Thor, Atlas y Minuteman se redujeron considerablemente los tiempos de ejecución.

La industria del automóvil siguió los pasos de la militar, aplicando técnicas de gestión de proyectos para la coordinación del trabajo entre áreas y equipos diferentes.

Comenzaron a surgir técnicas específicas, histogramas, cronogramas, los conceptos de ciclo de vida del proyecto o descomposición en tareas (WBS Work Breakdown Structure).

“La gestión de proyectos predictiva o clásica es una disciplina formal de gestión, basada en la planificación, ejecución y seguimiento a través de procesos sistemáticos y repetibles”

La Antitesis: Agilidad

- ¿El modelo predictivo es el único posible?
- ¿Los criterios para determinar el éxito sólo pueden ser el cumplimiento de fechas y costes?
- ¿Puede haber proyectos que no tengan como finalidad realizar un trabajo previamente planificado, con un presupuesto y en un tiempo previamente calculados?
- ¿Y si el cliente no estuviera interesado en saber si el sistema tendrá 20 ó 200 funcionalidades, si estará en beta 6 meses o 2 años?
- ¿Si su interés fuera poner en el mercado antes que nadie un producto valioso para los clientes, y estar continuamente desarrollando su valor y funcionalidad?

Quizá en algunos proyectos de software el empeño en aplicar prácticas de estimación, planificación, ingeniería de requisitos sea un empeño vano. Quizá la causa de los problemas no sea tanto por una mala aplicación de las prácticas, sino por la aplicación de prácticas inapropiadas. Quizá se estén generando “fiascos” al exigir a los clientes criterios de adquisición, y al aplicar a los proyectos procesos de gestión predictivos, cuando se trata de proyectos que no necesitan tanto garantías de previsibilidad en la ejecución, como valor y flexibilidad para trabajar en un entorno cambiante.

En marzo de 2001, 17 críticos de los modelos de mejora basados en procesos, convocados por Kent Beck, que había publicado un par de años antes el libro "Extreme Programming Explained" en el que exponía una nueva metodología denominada Extreme Programming, se reunieron en Salt Lake City para discutir sobre el desarrollo de software.

En la reunión se acuñó el término “Métodos Ágiles” para definir a los que estaban surgiendo como alternativa a las metodologías formales: vCMM-SW (Paulk, B., Chrissis, & Weber, 1996) precursor del CMMI, PMI, SPICE (proyecto inicial de ISO 15504), etc. a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas, previas al desarrollo.

Los integrantes de la reunión resumieron en cuatro postulados lo que ha quedado denominado como “Manifiesto Ágil”, que son los principios sobre los que se basan estos métodos.

El manifiesto Ágil

Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar:

- ***A los individuos y su interacción***, por encima de los procesos y las herramientas.
- ***El software que funciona***, por encima de la documentación exhaustiva.
- ***La colaboración con el cliente***, por encima de la negociación contractual.
- ***La respuesta al cambio***, por encima del seguimiento de un plan.

Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda

5.5. EL MODELO PROGRAMACIÓN EXTREMA PARA GESTIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE

La programación extrema o eXtreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, “Extreme Programming Explained: Embrace Change” (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.³

Principios

- **Retroalimentación a escala fina:** El principio de pruebas: se tiene que establecer un período de pruebas de aceptación del programa (llamado también período de caja negra) donde se definirán las entradas al sistema y los resultados esperados de estas entradas. Es muy recomendable automatizar estas pruebas para poder hacer varias simulaciones del sistema en funcionamiento. Para hacer estas

³ Programación Extrema, disponible en <http://www.programacionextrema.org/>

simulaciones automatizadas, se pueden utilizar Ambientes de Prueba (Unit testing frameworks). Un buen ejemplo de un ambiente de prueba es el JUnit para Java (www.junit.org/index.htm). Otros ambientes de pruebas para otros lenguajes como C, C++, JavaScript, XML y servicios Web, pueden encontrarse en www.xprogramming.com/software.htm.

- **Proceso de planificación:** en esta fase, el usuario tendrá que escribir sus necesidades, definiendo las actividades que realizará el sistema. Se creará un documento llamado Historias del usuario (User Stories). Entre 20 y 80 historias (todo dependiendo de la complejidad del problema) se consideran suficientes para formar el llamado Plan de Liberación, el cual define de forma específica los tiempos de entrega de la aplicación para recibir retroalimentación por parte del usuario. Por regla general, cada una de las Historias del usuario suelen necesitar de una a tres semanas de desarrollo. Son muy importantes y tienen que ser una constante las reuniones periódicas durante esta fase de planificación. Estas pueden ser a diario, con todo el equipo de desarrollo para identificar problemas, proponer soluciones y señalar aquellos puntos a los que se les ha de dar más importancia por su dificultad o por su punto crítico.
- **El cliente en el sitio:** se le dará poder para determinar los requerimientos, definir la funcionalidad, señalar las prioridades y responder las preguntas de los programadores. Esta fuerte interacción cara a cara con el programador disminuye el tiempo de comunicación y la cantidad de documentación, junto con los altos costes de su creación y mantenimiento. Este representante del cliente estará con el equipo de trabajo durante toda la realización del proyecto.
- **Programación en parejas:** uno de los principios más radicales y en el que la mayoría de gerentes de desarrollo ponen sus dudas. Requiere que todos los programadores XP escriban su código en parejas, compartiendo una sola máquina. De acuerdo con los experimentos, este principio puede producir aplicaciones más buenas, de manera consistente, a iguales o menores costes.

Características:

Las características fundamentales del método son:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas:** frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación. Véase, por ejemplo, las herramientas de prueba JUnit orientada a Java, DUnit orientada a Delphi y NUnit para la plataforma.NET. Estas dos últimas inspiradas en JUnit.
- **Programación en parejas:** se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en una misma posición. Se supone que la mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- **Frecuente integración del equipo de programación con el cliente o usuario:** Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores antes de añadir nueva funcionalidad:** Hacer entregas frecuentes.

- **Refactoring del código:** es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad, pero sin modificar su comportamiento. Las pruebas han de garantizar que en el refactoring no se ha introducido ningún fallo.
- **Propiedad del código compartida:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal

Fases de la Programación Extrema:

1ª Fase: Planificación del Proyecto.

- Historias de usuario.
- Release planning.
- Iteraciones.
- Velocidad del proyecto.
- Programación en pareja.
- Reuniones diarias.

2ª Fase: Diseño.

- Diseños simples.
- Glosarios de términos.
- Riesgos.
- Funcionalidad extra.

3ª Fase: Codificación.

4ª Fase: Pruebas.

5.6. EL LENGUAJE DE PROGRAMACIÓN PHP

Como dato histórico puede decirse que, “Rasmus Lerdorf, miembro del equipo de desarrollo de Apache, creó PHP (Personal Home Page) en 1994. Su única intención fue la de crear un pequeño sistema de control para verificar el número de personas que leían su curriculum vitae en la Web. En los meses siguientes a su creación, PHP se desarrolló en torno a un grupo de programadores que comprobaban el código y sus revisiones. Para dar más potencia al sistema, Rasmus creó funciones en lenguaje C para permitir conexión a bases de datos. Este fue el comienzo de la potencia real del lenguaje.”⁴

PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de

⁴ Cabezas Granado Luis Miguel. Manual Imprescindible de PHP. Anaya Multimedia. España. 2004.

procesador de PHP que genera la página Web resultante.

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

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante dominio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. El sitio web de Wikipedia está desarrollado en PHP.

Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Aunque todo en su diseño está orientado a facilitar la creación de sitios webs, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo; a esta versión de PHP se la llama PHP-CLI (Command Line Interface).

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo, obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. Mientras que PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# y Visual Basic .NET como lenguajes), a ColdFusion de la empresa Adobe, a JSP/Java y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un entorno de desarrollo integrado comercial llamado Zend Studio. CodeGear (la división de lenguajes de programación de Borland) ha sacado al mercado un entorno de desarrollo integrado para PHP, denominado 'Delphi for PHP'. También existen al menos un par de módulos para Eclipse, uno de los entornos más populares.

Características del Lenguaje

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace

evidente en el uso de php arrays.

- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos. Incluso aplicaciones como Zend framework, empresa que desarrolla PHP, están totalmente desarrolladas mediante esta metodología.
- No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

5.7. BASES DE DATOS RELACIONALES

Por definición de los profesionales expertos y pioneros en el área de las bases de datos podemos citar que, “Una base de datos relacional es un repositorio compartido de datos. Para hacer disponibles los datos de una base de datos relacional a los usuarios hay que considerar varios aspectos. Uno es la forma en que los usuarios solicitan los datos: ¿cuáles son los diferentes lenguajes de consulta que usan, que es el lenguaje de consulta más ampliamente usado actualmente. Otro aspecto es la integridad de datos y la seguridad; las bases de datos necesitan proteger los datos del daño provocado por los usuarios, ya sean intencionados o no. El componente de mantenimiento de la integridad de una base de datos asegura que las actualizaciones no violan las restricciones de integridad que hayan especificado sobre los datos. El componente de seguridad de una base de datos incluye la autenticación de usuarios y el control de acceso para restringir las posibles acciones de cada usuario. Estos aspectos se presentan independientemente del modelo de datos, pero se estudian en el contexto del modelo de datos relacional para ejemplificarlos.

Las restricciones de integridad forman la base del diseño de bases de datos relacionales.”⁵

Para manipular la información se utiliza un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el álgebra relacional y el cálculo relacional. El álgebra relacional permite describir la forma de realizar una consulta, en cambio, el cálculo relacional sólo indica lo que se desea devolver.

El lenguaje más común para construir las consultas a bases de datos relacionales es SQL (Structured Query Language), un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

En el modelo relacional los atributos deben estar explícitamente relacionados a un nombre en todas las operaciones, en cambio, el estándar SQL permite usar columnas sin nombre en conjuntos de resultados, como el asterisco taquigráfico (*) como notación de consultas.

Al contrario del modelo relacional, el estándar SQL requiere que las columnas tengan un orden definido, lo cual es fácil de implementar en una computadora, ya que la memoria es lineal. Es de notar, sin embargo, que en SQL el orden de las columnas y los registros devueltos en cierto conjunto de resultado nunca está garantizado, a no ser que explícitamente sea especificado por el usuario.

Características:

- Una base de datos relacional se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.
- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones

Ventajas y Desventajas de las Bases de Datos Relacionales

Ventajas

- Provee herramientas que garantizan evitar la duplicidad de registros.
- Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.

Desventajas

- Presentan deficiencias con datos gráficos, multimedia, CAD y sistemas de información geográfica.

⁵ Silberschatz Abraham. Korth Henry. Suddarshan. Fundamentos de Bases de Datos, McGraw Hill. Madrid 2002.

- No se manipulan de forma manejable los bloques de texto como tipo de dato.
- Las bases de datos orientadas a objetos (BDOO) se propusieron con el objetivo de satisfacer las necesidades de las aplicaciones anteriores y así, complementar, pero no sustituir a las bases de datos relacionales.

El Gestor de Bases de Datos MySQL

Un gestor de bases de datos es un programa diseñado para administrar bases de datos, y uno de los gestores que posee una gran cuota de usuarios a nivel mundial en la actualidad es MySQL. Citando que, "MySQL es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder a un código fuente, es decir, a1 código de programación de MySQL. Todo el mundo puede contribuir para incluir elementos, arreglar problemas, realizar mejoras o sugerir optimizaciones. Y así ocurre, MySQL ha pasado de ser una "pequeiia" base de datos a una completa herramienta y ha conseguido superar a una gran cantidad de bases de datos comerciales). Por lo tanto, su rápido desarrollo se debe a la contribución de mucha gente a un proyecto, así como a la dedicación del equipo de MySQL." ⁶

De manera que MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.¹ MySQL AB desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009 desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado, se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C. Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

Características:

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

⁶ Gilfillan Ian. La Biblia de MySQL. Anaya Multimedia. Madrid. 2003

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones. MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

5.8. LA PROGRAMACIÓN ORIENTADA A OBJETOS

Las tecnologías orientadas a objetos se han convertido en uno de los motores claves de la industria del software, sin embargo, esta tecnología no es como algunos innovadores pregonan diciendo que es nueva o novedosa. Realmente se trata de una vieja y madura tecnología que se remota a los años sesenta. Este enfoque de desarrollo implica la creación de modelos del mundo real y a construcción de programas informáticos basados en esos modelos. Por lo tanto, podemos citar que, “La programación orientada a objetos es una extensión natural en la actual tecnología de programación y representa un enfoque nuevo y distinto al tradicional. Al igual que cualquier otro programa, el diseño de un programa orientado a objetos tiene lugar en la fase de diseño del ciclo de vida de desarrollo de software. El diseño de un programa orientado a objetos es único en el sentido de que se organiza en función de los objetos que manipulará.”⁷

⁷ Joyanes Aguilar Luis. Programación Orientada a Objetos. Segunda Ed. McGraw-Hill. Madrid. 2006

Es importante aclarar desde un principio la diferencia que existe entre programación orientada a objetos y un lenguaje orientado a objetos. La programación orientada a objetos es una "filosofía", un modelo de programación, con su teoría y su metodología, que conviene conocer y estudiar, antes de nada. Un lenguaje orientado a objetos es un lenguaje de programación que permite el diseño de aplicaciones orientadas a objetos. Dicho esto, lo normal es que toda persona que vaya a desarrollar aplicaciones orientadas a objetos aprenda primero la "filosofía" (o adquiera la forma de pensar) y después el lenguaje, porque "filosofía" sólo hay una y lenguajes muchos. En este se verá brevemente los conceptos básicos de la programación orientada a objetos desde un punto de vista global, sin particularizar para ningún lenguaje de programación específico.

La programación orientada a objetos surge en la historia como un intento para dominar la complejidad que, de forma innata, posee el software. Tradicionalmente, la forma de enfrentarse a esta complejidad ha sido empleando lo que se llama programación estructurada, que consiste en descomponer el problema objeto de resolución en subproblemas y más subproblemas hasta llegar a acciones muy simples y fáciles de codificar. Se trata de descomponer el problema en acciones, en verbos. En el ejemplo de un programa que resuelva ecuaciones de segundo grado, descomponíamos el problema en las siguientes acciones: primero, pedir el valor de los coeficientes a, b y c; después, calcular el valor de la discriminante; y por último, en función del signo del discriminante, calcular ninguna, una o dos raíces.

Características de Programación Orientada a Objetos

Existe un acuerdo acerca de qué características contempla la "orientación a objetos". Las características siguientes son las más importantes⁸:

Abstracción

Denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar "cómo" se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos, y, cuando lo están, una variedad de técnicas es requeridas para ampliar una abstracción. El proceso de abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

Encapsulamiento

Significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión (diseño estructurado) de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

⁸ Coad, P; Yourdon, E. (1991). Prentice-Hall International editions, ed. *Object-oriented Design*.

Polimorfismo

Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O, dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

Herencia

Las clases no se encuentran aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento, permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple; siendo de alta complejidad técnica por lo cual suele recurrirse a la herencia virtual para evitar la duplicación de datos.

Modularidad

Se denomina "modularidad" a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. Estos módulos se pueden compilar por separado, pero tienen conexiones con otros módulos. Al igual que la encapsulación, los lenguajes soportan el modularidad de diversas formas.

Principio de ocultación

Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una "interfaz" a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas; solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no puedan cambiar el estado interno de un objeto de manera inesperada, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.

Recolección de basura

La recolección de basura (garbage collection) es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente, y por tanto desvincular la memoria asociada, los objetos que hayan quedado sin ninguna referencia a ellos. Esto significa que el programador no debe preocuparse por la asignación o liberación de memoria, ya que el entorno la asignará al crear un nuevo objeto y la liberará cuando nadie lo esté usando. En la mayoría de los lenguajes híbridos que se extendieron para soportar el Paradigma de

Programación Orientada a Objetos como C++ u Object Pascal, esta característica no existe y la memoria debe desasignarse expresamente

5.9. EL PATRÓN DE DESARROLLO MODELO-VISTA-CONTROLADOR

Modelo Vista Controlador (MVC) es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. El patrón MVC fue una de las primeras ideas en el campo de las interfaces gráficas de usuario y uno de los primeros trabajos en describir e implementar aplicaciones software en términos de sus diferentes funciones.

MVC fue introducido por Trygve Reenskaug (web personal) en Smalltalk-76 durante su visita a Xerox Parc6 7 en los años 70 y, seguidamente, en los años 80, Jim Althoff y otros implementaron una versión de MVC para la biblioteca de clases de Smalltalk-80.8 Solo más tarde, en 1988, MVC se expresó como un concepto general en un artículo9 sobre Smalltalk-80.

En esta primera definición de MVC el controlador se definía como «el módulo que se ocupa de la entrada» (de forma similar a como la vista «se ocupa de la salida»). Esta definición no tiene cabida en las aplicaciones modernas en las que esta funcionalidad es asumida por una combinación de la 'vista' y algún framework moderno para desarrollo. El 'controlador', en las aplicaciones modernas de la década de 2000, es un módulo o una sección intermedia de código, que hace de intermediario de la comunicación entre el 'modelo' y la 'vista', y unifica la validación (utilizando llamadas directas o el «observer» para desacoplar el 'modelo' de la 'vista' en el 'modelo' activo).

Descripción del Modelo

De manera genérica, los componentes de MVC se podrían definir como sigue:

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

El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto, se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

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

Uso de MVC en Bases de Datos

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos que debe utilizar la aplicación; en líneas generales del MVC dicha gestión corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre la Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Uso de MVC en Aplicaciones Web

Aunque originalmente MVC fue desarrollado para aplicaciones de escritorio, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web en los principales lenguajes de programación. Se han desarrollado multitud de frameworks, comerciales y no comerciales, que implementan este patrón; estos frameworks se diferencian básicamente en la interpretación de como las funciones MVC se dividen entre cliente y servidor.

Los primeros frameworks MVC para desarrollo web planteaban un enfoque de cliente ligero el que casi todas las funciones, tanto de la vista, el modelo y el controlador recaían en el servidor. En este enfoque, el cliente manda la petición de cualquier hiperenlace o formulario al controlador y después recibe de la vista una página completa y actualizada (u otro documento); tanto el modelo como el controlador (y buena parte de la vista) están completamente alojados en el servidor. Como las tecnologías web han madurado, ahora existen frameworks como JavaScriptMVC, Backbone o jQuery14 que permiten que ciertos componentes MVC se ejecuten parcial o totalmente en el cliente.⁹

6. METODOLOGÍA DE LA INVESTIGACIÓN

6.1. TIPO DE INVESTIGACIÓN

El tipo de investigación es aplicada, bajo la premisa que el objetivo del proyecto es “Desarrollo de una Plataforma Informática”.

⁹ How to use Model-View-Controller (MVC). (1992). Web.archive.org. Revisado el 8-11-2017, disponible en <https://web.archive.org/web/20120429161935/http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>

6.2. MÉTODO DE INVESTIGACIÓN

La investigación está basada en el método científico, es decir que el desarrollo del proyecto lleva una secuencia técnica y metodológica basada en los pasos y principios científicos en la búsqueda de documentar el objeto y ofrecer una aplicación tecnológica e innovadora como aporte del mismo. En este sentido cada fase de la investigación ha llevado una serie de estudio técnico para la elaboración de la plataforma.

Población y Muestra

Población

La investigación se desarrolló tomando en cuenta todo el equipo del Programa Institucional de Investigación, el cual está conformado por vice rectoría académica, coordinadores del programa de investigación, los directores de las regionales, coordinadores del departamento de investigación de las regionales y docentes investigadores los cuales hacen un total de 33 personas.

Muestra

La muestra tomada para la realización del proyecto fue el personal de la Dirección de Investigación y Proyección Social, tal como se detalla a continuación:

Nombre	Cargo
Ing. Mario Wilfredo Montes	Director de Investigación y Proyección Social
Ing. David Emmanuel Ágreda	Coordinador de Investigación

Técnicas e Instrumentos de Investigación

Técnicas de Investigación

Entrevista: Para la formulación de la lista de requisitos de la aplicación se realizaron entrevistas dirigidas al equipo de Dirección y Coordinación del Programa de Investigación Institucional, Ing. Mario Wilfredo Montes e Ing. David Emmanuel Ágreda, respectivamente.

Instrumentos de Investigación

Guía de Entrevista¹⁰: Se diseñó con el fin de identificar elementos que permitieron realizar el levantamiento de requisitos para diseñar la plataforma digital que se implementará en el programa institucional de investigación.

6.3. FASES DE DESARROLLO DE LA INVESTIGACIÓN

Fase I: Planificación y Ejecución del Trabajo de Campo

La primera etapa consistió en la construcción y tipificación de muestra, realización de observaciones y reuniones con el equipo del programa institucional de investigación de ITCA-FEPADE, especialmente con

¹⁰ Anexo I

el personal de coordinación de esta área.

Fase II: Análisis de los Resultados

La segunda fase se realizó el filtrado necesario de los resultados que se obtuvieron de la investigación de campo, permitiendo identificar los requisitos necesarios para la elaboración técnica de la aplicación.

Fase III: Desarrollo de la Plataforma Web

La tercera fase se realizó el trabajo de programación de la aplicación utilizando las técnicas de programación orientada a objetos y el patrón Modelo-Vista-Controlador, el cual es ampliamente utilizado como estándar para el desarrollo web en la actualidad.

Fase IV: Protocolo Final del Proyecto

Se elaboró y presentó la aplicación web como producto final del proyecto y se deja a disponibilidad para el departamento coordinador del programa institucional de investigación.

7. RESULTADOS

De acuerdo a la lista de requisitos que se plantearon en la pila del producto al principio del proyecto, con la investigación se lograron los siguientes resultados en el desarrollo de la aplicación:

Vista de inicio de sesión de la plataforma



Figura 4: Para dar seguridad a la plataforma se diseñó un modelo de inicio de sesión donde se permite acceso a los usuarios mediante cuentas validadas con contraseña, estas cuentas serán previamente creadas por uno de los administradores de la plataforma

Vista principal de la aplicación

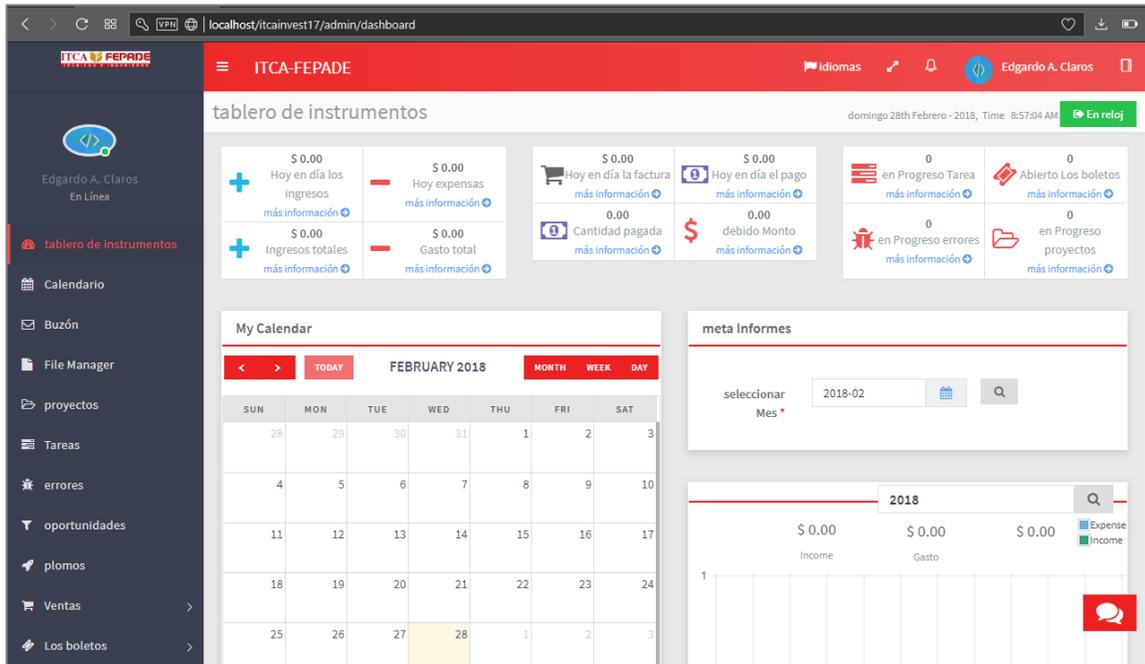


Figura 5: En el formulario principal del sistema se tiene acceso a los diferentes módulos de la plataforma, de igual forma se tiene disponible los enlaces a herramientas útiles como el calendario de eventos, vista de tareas, presupuestos, informes, entre otros.

Vista de Generalidades de Proyectos

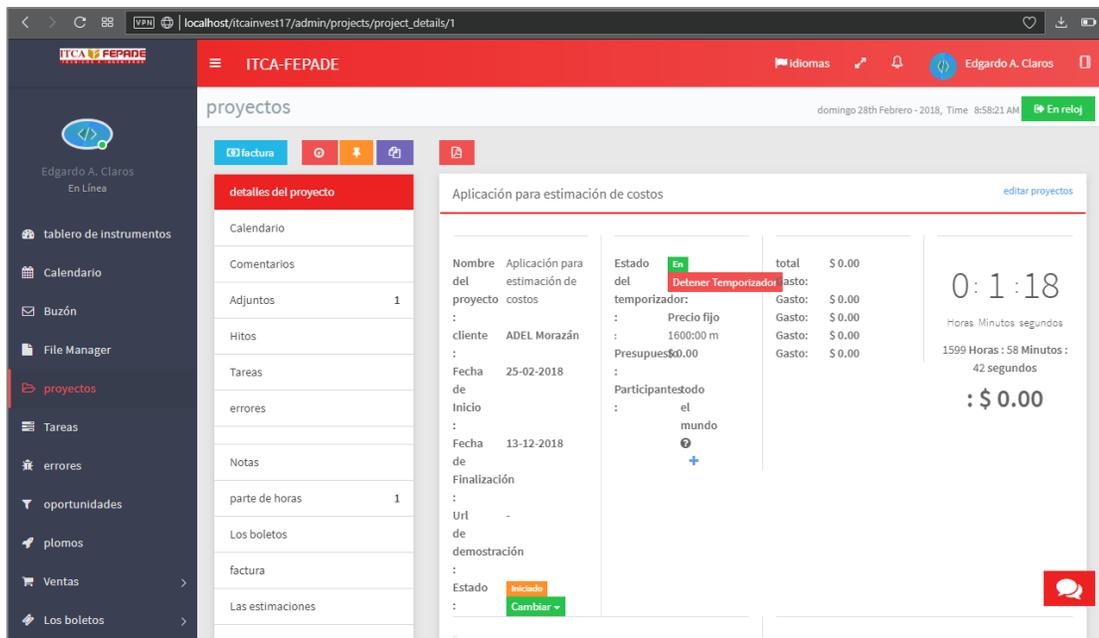


Figura 6: En la vista de proyectos se tiene acceso a toda la información sobre el mismo, como el estado, presupuesto, lista de tareas en las que se está trabajando, miembros del equipo investigador, diagramas de Gantt, tiempos de ejecución, archivos del proyecto, etc.

Vista de gestor de archivos de proyectos

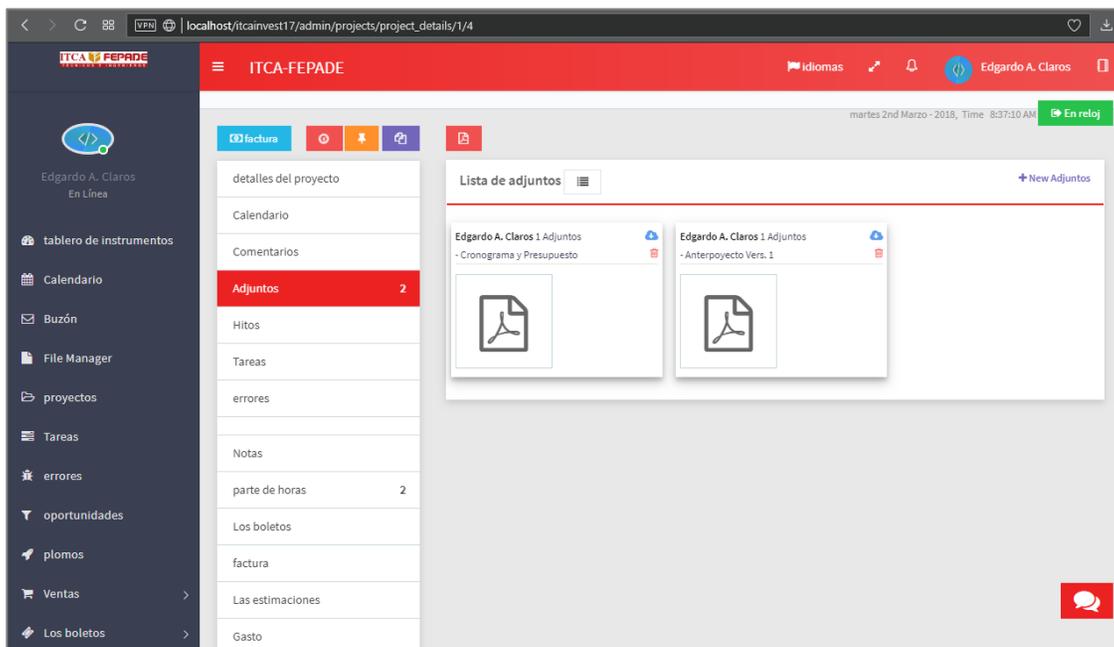


Figura 7: La plataforma cuenta con un gestor de archivos en el que a cada proyecto se puede adjuntar diversos archivos en distintos formatos, en este apartado se colocarán los documentos de propuestas, ante proyecto, reporte de avances, entre otros.

Vista de presupuesto

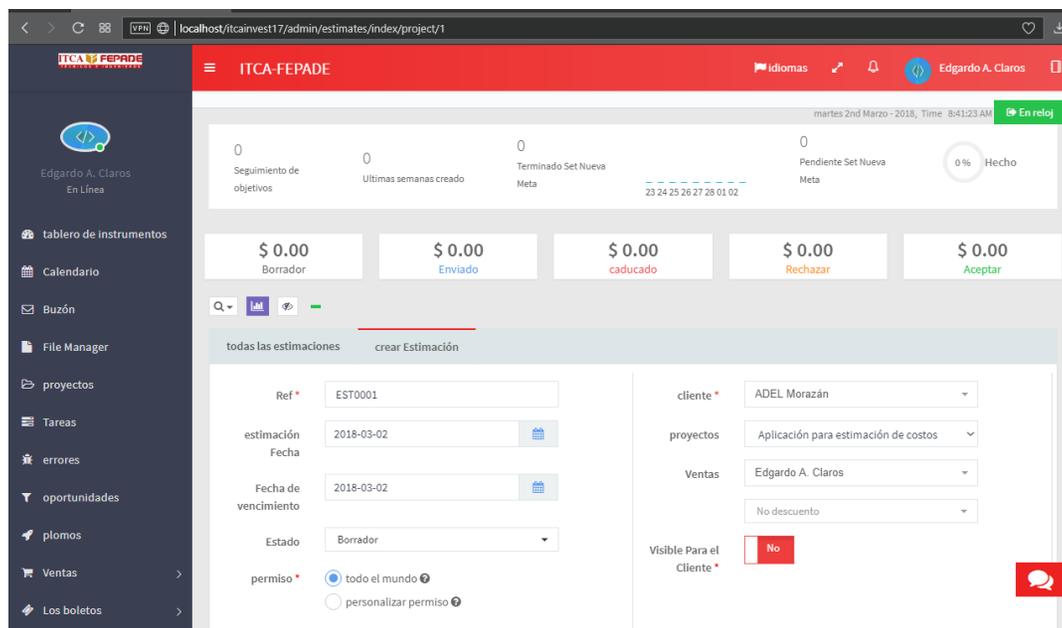


Figura 8: Una herramienta que será de utilidad para la gestión de los proyectos es el módulo de presupuesto. En este apartado se podrán crear estimaciones sobre los costos que se tendrán en un proyecto, de igual forma también se permite el ingreso de factura de gastos y compras.

Vista de la gestión de tareas de los proyectos

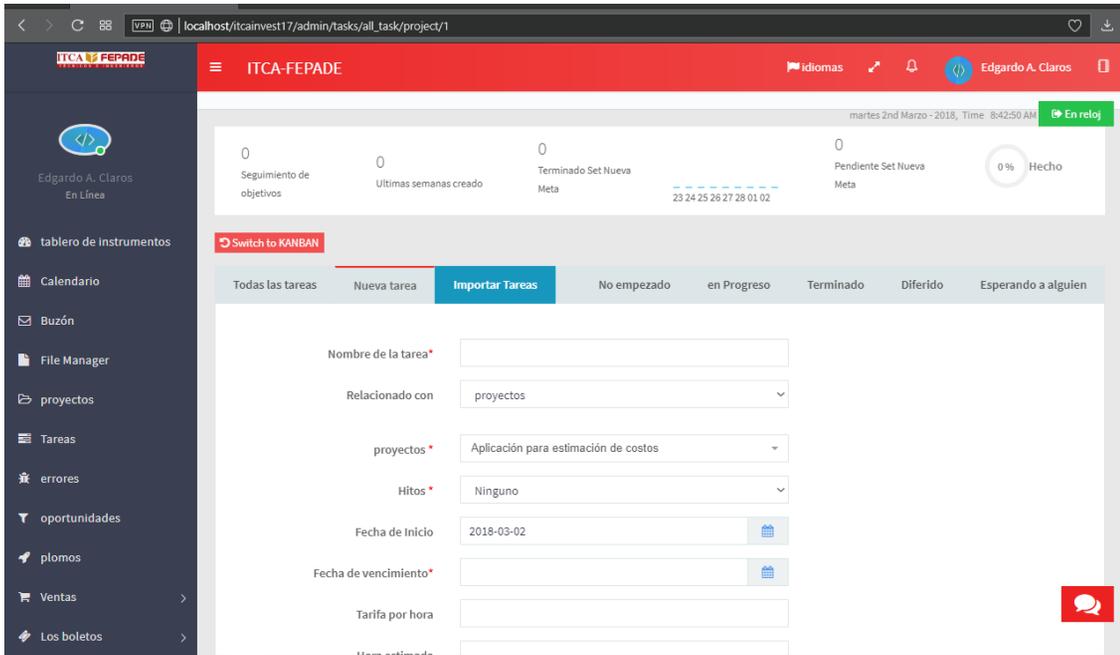


Figura 9: Los proyectos podrán mantenerse actualizados mediante el ingreso de tareas, las cuales representan las actividades que se durante el desarrollo del proyecto. Así mismo se podrá definir un estado de las tareas, como iniciado, pendiente, en proceso y terminado.

Vista de registro de gastos

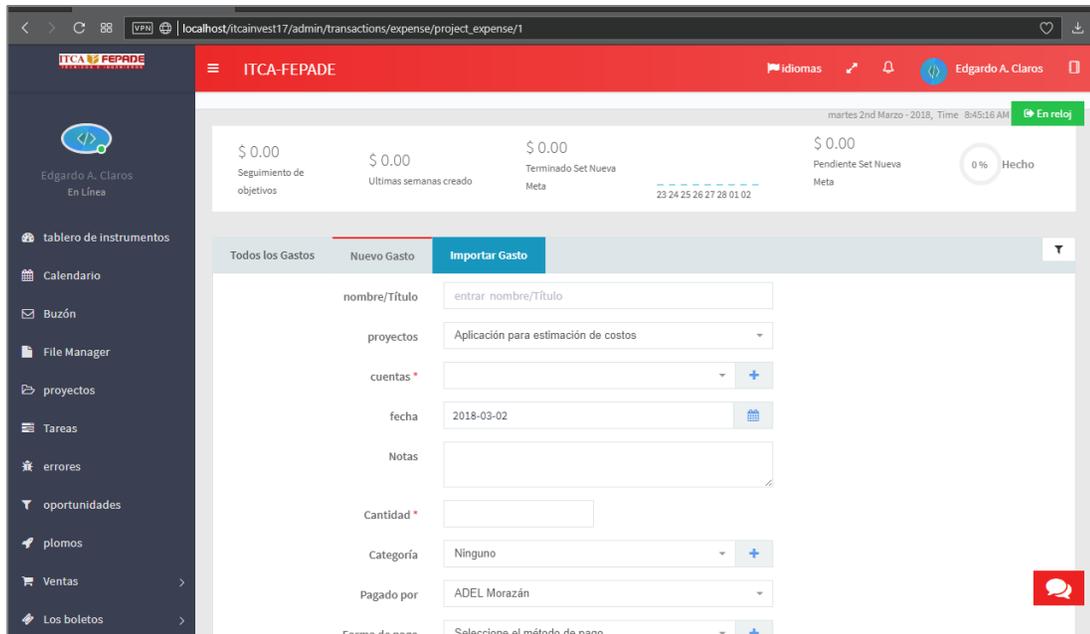


Figura 10: Este módulo es un complemento importante del módulo de presupuestos. En este apartado se pueden registrar facturas para respaldar los gastos relacionados a los proyectos, de igual forma se puede crear un registro de plan de compra para anexarlo al proyecto en cuestión.

Vista de chat entre usuarios

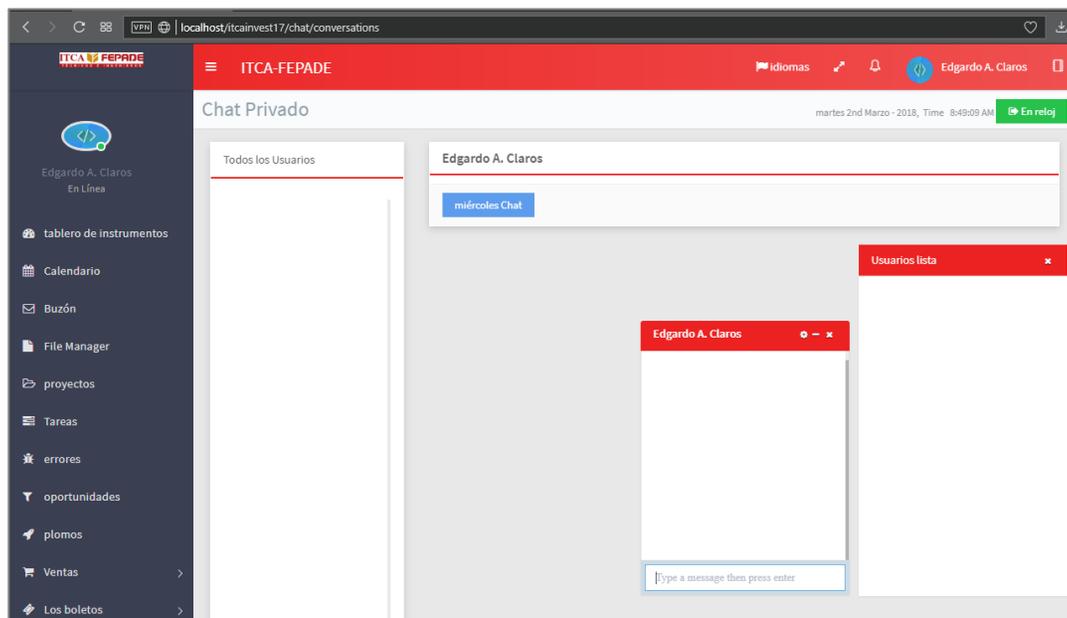


Figura 11: Este es uno de los módulos que permiten la comunicación directa entre los usuarios de la plataforma, se puede mantener un chat privado con los miembros del equipo investigador para un determinado proyecto, o se puede establecer un chat también con otro miembro de la plataforma. De igual forma se puede compartir archivos en el chat.

Vista de correo electrónico

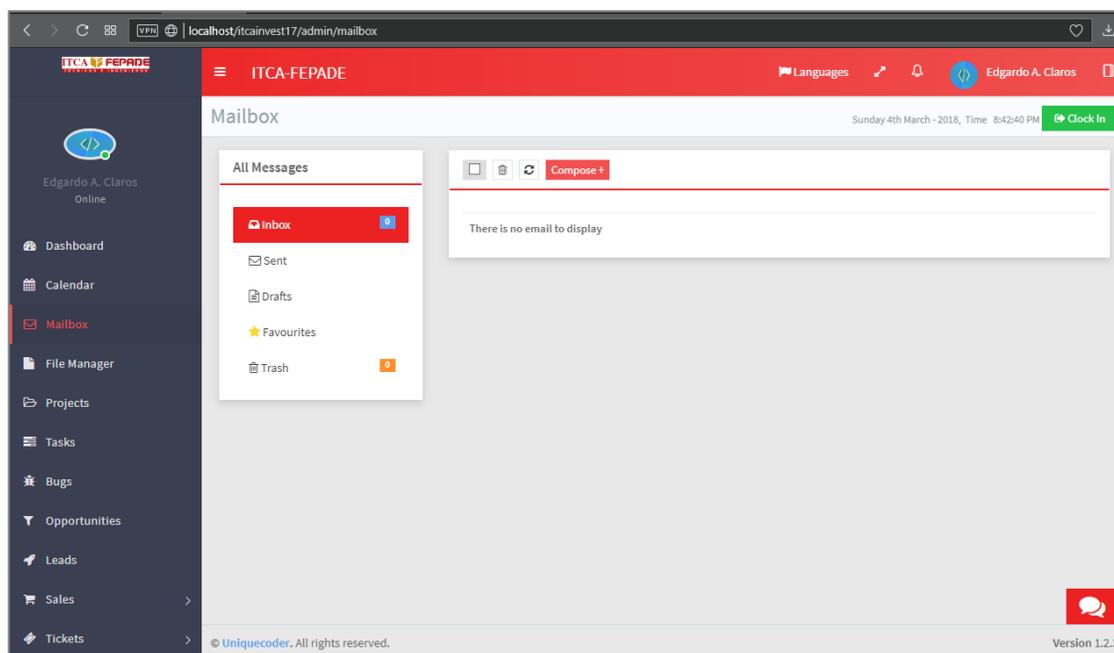


Figura 12: Otro de los medios de comunicación interno que tiene la plataforma es el módulo de correo electrónico, el cual se maneja con el usuario que se crea dentro de la plataforma. Esta herramienta brinda la posibilidad de mantener contacto con los miembros de un proyecto determinado o de todos los usuarios de la plataforma.

Vista de Informes del Sistema

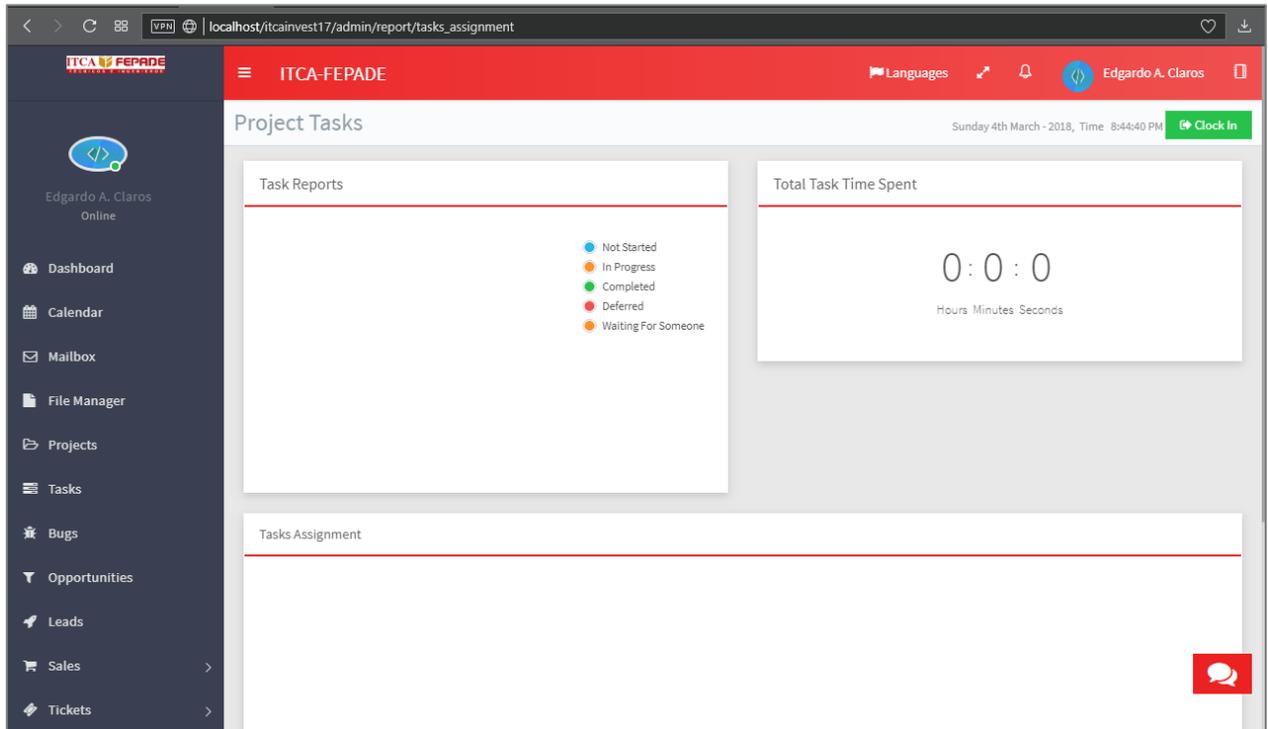


Figura 13: La plataforma puede generar reportes con información útil sobre los proyectos, como lo es: estado, información general, presupuestos, gastos, tareas trabajadas, diagrama de Gantt, etc.

8. CONCLUSIONES

1. Realizar este proyecto en la institución es uno de los esfuerzos que están permitiendo que se tecnifiquen los procesos clave dentro de las diferentes áreas. La Dirección de Investigación realiza una función clave que al implementarse esta plataforma podrá gestionar de manera digital las actividades de monitoreo y seguimiento de los diferentes proyectos en el programa de investigación institucional.
2. El software incorpora tecnologías de última generación orientadas a la web, lo cual permite el procesamiento de datos del programa de investigación de forma eficiente. Así mismo, con el avance creciente en los temas de los lenguajes de programación se ha podido crear un completo sistema sin costos adicionales con respecto al tema de licenciamiento, pues toda la base de código es libre.
3. Los investigadores ahora disponen de un software que les permita actualizar el estado de sus proyectos, así como dar seguimiento oportunamente gracias a las observaciones y monitoreo que las autoridades respectivas realicen durante todas las fases del mismo. Con ello se facilitan nuevos canales de comunicación para favorecer la fluidez de la información bilateralmente.
4. Se han incorporado elementos útiles para el monitoreo del estado de proyectos, como lo es un generador de diagramas de Gantt, módulo de estimaciones financieras, gestor de archivos, entre otros; con lo que se está entregando una herramienta integral en el control y seguimiento de proyectos.

5. El programa de investigación tiene además a disposición en la plataforma un espacio donde centralizar los documentos vigentes como guías informativas y formatos para el seguimiento de los proyectos, permitiendo de esta forma dejar a disposición de los investigadores un valioso repositorio que les servirá para respaldar y fortalecer su labor.

9. RECOMENDACIONES

1. El seguimiento de los proyectos dentro de la plataforma debe de realizarse de forma regular, a fin de mantener siempre actualizado su estado. La plataforma es una herramienta que permite automatizar muchos de los pasos que intervienen en el proceso, sin embargo, el trabajo de monitoreo debe de hacerse manualmente.
2. Los administradores de la plataforma deben de asegurarse de dar acceso a los proyectos a los usuarios idóneos para consultar detalles sobre su estado o realizar procedimientos de actualización sobre el archivo del proyecto en el sistema.
3. El sistema cuenta con diversos reportes que al generarse proveen información valiosa sobre el estado de los proyectos, sin embargo, estos no sustituyen los reportes que los investigadores realizan para presentar resultados y avances significativos sobre sus investigaciones.
4. La plataforma dispone de un módulo de gestor de archivos en cada registro de proyecto, el cual es necesario monitorear regularmente con el propósito de realizar las tareas propias de revisión, validación y aprobación cuando así fuese necesario.
5. Para facilitar la comunicación entre los usuarios se recomienda el uso de los módulos de mensajería interna, correos y chat que la plataforma provee. Así mismo, no se debe de sustituir el uso del correo electrónico institucional.
6. Es necesario que para mantener la integridad del buen funcionamiento de la plataforma los códigos fuente de la misma no sean alterados mediante intervención directa o mediante la instalación de extensiones desde la parte administrativa del software.
7. Se recomienda hacer respaldos periódicamente directamente desde el servidor web, el cual el proveedor contratado para el servicio orientará como realizarse, con esta medida se garantizará mantener siempre la información de la plataforma resguardada ante imprevistos en cualquier momento.
8. Para garantizar el correcto funcionamiento de la plataforma, se recomienda que esta esté instalada en un servidor con sistema operativo Linux o Windows Server

10. GLOSARIO

Sistema de Información: Se refiere a una aplicación informática de tipo de software cuyo objetivo es brindar soporte en la automatización de actividades y procesos de una entidad.

Protocolo: En informática, un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.

TCP/IP: Es un conjunto de protocolos que permiten la transmisión de información en redes. Consiste en cuatro capas, capa de aplicación, capa de transporte, capa de internet y capa de acceso a la red.

HTTP: Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

HTML: HTML, siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web.

POO: La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos.

MVC: Model-View-Controller. Patrón arquitectónico desarrollado para interfaces gráficas que resalta la importancia de una separación clara entre la presentación de datos y la lógica de negocio de una aplicación.

11. BIBLIOGRAFÍA

[1] Jeffrey L. Whitten

Análisis y diseño de sistemas de información
Tercera Ed. McGraw-Hill/Irwin México, 2003

[2] James A. Senn

Análisis y Diseño de Sistemas de Información
Tercera Ed. Mc Graw Hill. 2003

[3] Carles Mateu

Desarrollo de Aplicaciones Para la Web
Eureca Media SL. Barcelona, 2004.

[4] Palacio Juan, Ruata Claudia

Scrum Manager Gestión de Proyectos. Rev. 2.6. Safe Creative, 2016.
http://www.scrummanager.net/files/scrum_manager.pdf, Consultado el 08-12-2017

[5] Joyanes Aguilar Luis

Fundamentos de Programación, Tercera Ed.
McGraw Hill. Madrid. 1996.

[6] Cabezas Granada Luis Miguel

Manual Imprescindible de PHP.
Anaya Multimedia. España. 2004.

[7] Silberschatz Abraham. Korth Henry. Suddarshan
Fundamentos de Bases de Datos
McGraw Hill. Madrid 2002.

[8] Gilfillan Ian
La Biblia de MySQL
Anaya Multimedia. Madrid. 2003

[9] Joyanes Aguilar Luis
Programación Orientada a Objetos
Segunda Ed. McGraw-Hill. Madrid. 2006

[10] Fabien Potencier, Francois Zaninotto
Symfony La Guía Definitiva Rev. 1.2, Libros Web, 2008.
http://www.librosweb.es/symfony_1_2, consultado el 07-11-2017

[11] Berumen Milburn, Jaqueline
Monitoreo y Evaluación de Proyectos
Colombia, 2010

[12] Domingo Ajenjo, Alberto
Dirección y Gestión de Proyectos un Enfoque Práctico
México, 2000

12.ANEXOS

12.1. ANEXO 1: GUÍA DE ENTREVISTA UTILIZADA



ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA FEPADE MEGATEC LA UNIÓN
DEPARTAMENTO DE INVESTIGACIÓN Y PROYECCIÓN SOCIAL

Institución: _____

Dirección: _____

Entrevista dirigida a: _____

Cargo en la empresa: _____

Fecha de realización: ___ / ___ / ____

OBJETIVO: Recopilar información sobre los procesos de monitoreo y seguimiento a proyectos del programa institucional de investigación de ITCA-FEPADE

INDICACIONES: Responda a las preguntas que se le presentan a continuación de forma objetiva

1. ¿Cuál es el proceso para presentar las propuestas de proyectos de investigación?

2. ¿Cómo se procede para registrar investigadores de apoyo y estudiantes participantes en los proyectos de investigación?

3. ¿Por qué medio se presentan las propuestas y ante proyectos de investigación?

4. ¿Cómo se gestiona actualmente la elaboración de presupuestos de los proyectos de investigación?

5. ¿Actualmente, cómo se registran los horarios de los docentes investigadores principales y de apoyo?

6. ¿Cómo se procede para registrar investigadores de apoyo y estudiantes participantes en los proyectos de investigación?

7. ¿Qué herramientas utiliza para el seguimiento de los proyectos (cronogramas, diagramas de Gantt, otros)?

8. ¿Cómo entrega las observaciones a los documentos que los investigadores elaboran, tal como propuestas, ante proyectos, informes finales, etc?

9. ¿Tiene un repositorio de documentos vigentes para el programa institucional de investigación?

12.2. ANEXO 2: PILA DEL PRODUCTO DEL PROYECTO

PILA DE PRODUCTO		
PRO- YECTO	"DESARROLLO DE UNA PLATAFORMA PARA EL SEGUIMIENTO DE PROYECTOS DE INVESTIGACIÓN INSTITUCIONAL DE ITCA-FEPADE"	
FECHA	23/03/2017	
ID	DESCRIPCIÓN	PRIORIDAD
	Formulación y Aprobación de Proyectos de Investigación	
1	Ingreso de propuestas de proyectos de investigación en formulario web de la plataforma	ALTA
2	Registro de docentes investigadores principales y de apoyo en los proyectos de investigación	MEDIA
3	Registro de estudiantes participantes en los proyectos de investigación	MEDIA
4	Control aprobación, anulación y registro de observaciones a propuestas de proyectos de investigación directamente en el registro de cada propuesta	ALTA
5	Ingreso de documentos de anteproyectos de investigación en formato web y como archivo adjunto de word	ALTA
6	Control aprobación, anulación y registro de observaciones a anteproyectos de investigación directamente en el registro de cada anteproyecto	ALTA
7	Módulo administrable para la elaboración de presupuesto financiero para los proyectos de investigación	ALTA
8	Módulo básico para gestión de plan de compra de acuerdo al presupuesto del proyecto con funcionalidad de integración de cotizaciones y especificaciones técnicas	MEDIA
	Monitoreo, Seguimiento y Evaluación de Proyectos de Investigación	
9	Módulo de registro de horario de investigadores principales y de apoyo donde se pueda detallar el tiempo dedicado a actividades y trabajo de investigación	ALTA
10	Generación del resumen de estado en tiempo real de los proyectos de investigación de acuerdo a la fase, nombre del proyecto, porcentaje de avance global, centro regional e	ALTA

PILA DE PRODUCTO		
	investigador responsable. Esta información se podrá generar como reporte para impresión	
11	Control de seguimiento de tareas y actividades realizadas ligadas a los objetivos específicos del proyecto representadas por medio del porcentaje de avance de manera individual y por proyecto	MEDIA
12	Registro del reporte mensual de avance del proyecto en formulario web de la plataforma validado por el director de regional	MEDIA
13	Gestión de gastos adicionales al presupuesto realizados por cada regional para el desarrollo de los proyectos de investigación	MEDIA
14	Detalle del presupuesto consumido y el restante durante las diferentes fases de los proyectos	MEDIA
15	Módulo informativo para seguimiento del estado de registro de propiedad intelectual y/o patentes que generen los proyectos	MEDIA
16	Módulo informativo para seguimiento del estado de difusión y publicación de los proyectos de investigación	MEDIA
17	Ingreso de documentos de informe final de los proyectos de investigación	ALTA
18	Control aprobación, anulación y registro de observaciones a informes finales directamente en el registro de cada informe	ALTA
19	Ingreso de documentos de artículos científicos por proyectos exitosos	ALTA
20	Control aprobación, anulación y registro de observaciones a artículos científicos en el registro de cada artículo	ALTA
21	Gestión de informe de cierre y coste de los proyectos de investigación de acuerdo a los lineamientos del MINED	ALTA
	Misceláneos	
22	Gestión de directorio de documentos oficiales vigentes del programa de investigación institucional	MEDIA
23	El proceso de validación de los diferentes pasos en las etapas de desarrollo de los proyectos de investigación será realizado por los usuarios: investigador, director de regional, director de investigación y proyección social, entre otros según sea la necesidad particular.	ALTA
24	Gestión de información multimedia (fotos, videos, etc) para respaldo en los proyectos	MEDIA
25	Control de acceso a la plataforma mediante perfiles de usuario con diferentes niveles de privilegios de acuerdo a su tipo: investigador, director de regional, director de investigación y proyección social, entre otros según sea la necesidad particular.	MEDIA
26	Enlace al correo electrónico para facilitar la comunicación institucional	BAJA
27	Reportes de impresión de los módulos donde sea requerido	MEDIA
28	Interfaz amigable con el usuario	MEDIA
NOTAS: La pila del producto se ha elaborado de acuerdo a los requisitos identificados durante video conferencia y el normativo institucional de investigación rev. 2012		

VISIÓN

Ser una institución educativa líder en educación tecnológica a nivel nacional y regional, comprometida con la calidad, la empresarialidad y la pertinencia de nuestra oferta educativa.

MISIÓN

Formar profesionales integrales y competentes en áreas tecnológicas que tengan demanda y oportunidad en el mercado local, regional y mundial, tanto como trabajadores y como empresarios.

VALORES

EXCELENCIA: *Nuestro diario quehacer está fundamentado en hacer bien las cosas desde la primera vez.*

INTEGRIDAD: *Actuamos congruentemente con los principios de la verdad en todas las acciones que realizamos.*

ESPIRITUALIDAD: *Desarrollamos todas nuestras actividades en la filosofía de servicio, alegría, compromiso, confianza y respeto mutuo.*

COOPERACIÓN: *Actuamos basados en el buen trabajo en equipo, la buena disposición a ayudar a todas las personas.*

COMUNICACIÓN: *Respetamos las diferentes ideologías y opiniones, manteniendo y propiciando un acercamiento con todo el personal.*

SEDE Y REGIONALES EL SALVADOR



La Escuela Especializada en Ingeniería ITCA-FEPADE, fundada en 1969, es una institución estatal con administración privada, conformada actualmente por 5 campus: Sede Central Santa Tecla y cuatro Centros Regionales ubicados en Santa Ana, San Miguel, Zacatecoluca y La Unión.

1 SEDE CENTRAL SANTA TECLA

Km. 11.5 carretera a Santa Tecla, La libertad.
Tel.: (503) 2132-7400
Fax: (503) 2132-7599

2 CENTRO REGIONAL SANTA ANA

Final 10a. Av. Sur, Finca Procavia.
Tel.: (503) 2440-4348
Tel./Fax: (503) 2440-3183

3 CENTRO REGIONAL LA UNIÓN

Calle Sta. María, Col. Belén, atrás del Instituto Nacional de La Unión
Tel.: (503) 2668-4700

4 CENTRO REGIONAL ZACATECOLUCA

Km. 64.5, desvío Hacienda El Nilo sobre autopista a Zacatecoluca.
Tel.: (503) 2334-0763 y
(503) 2334-0768

5 CENTRO REGIONAL SAN MIGUEL

Km. 140 carretera a Santa Rosa de Lima.
Tel.: (503) 2669-2298
Fax: (503) 2669-0061