

IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN ACADÉMICA (SGA) EN EL INSTITUTO TECNOLÓGICO SUPERIOR JUAN BAUTISTA AGUIRRE

Mariuxi Yomaira OLVERA MORÁN*

Coordinadora de Investigación del Instituto Tecnológico Superior Juan Bautista Aguirre, Magister en Sistemas Integrados de Gestión, Daule, Ecuador

Fernando Gadiel DOMÍNGUEZ RAMOS

Docente Investigador, Instituto Tecnológico Superior Juan Bautista Aguirre
Ingeniero en Sistemas Computacionales, Daule, Ecuador

Marcel Oswaldo MÉNDEZ MANTUANO

Departamento de Investigación, Instituto Tecnológico Superior Juan Bautista Aguirre,
Magister en Gestión Ambiental, Daule, Ecuador

John Emmanuel TOBAR LITARDO

Coordinador de Carrera, Instituto Tecnológico Superior Juan Bautista Aguirre, Magister en Sistemas de Información Gerencial, Daule, Ecuador

*Autor para correspondencia: mariuxi_olvera@hotmail.com

RESUMEN

La formación técnica y tecnológica en los últimos años en Ecuador ha tenido un alto impacto debido a la rápida inserción de profesionales que exige nuestro mercado laboral, provocando que el número de estudiantes que ingresan a las aulas de los institutos técnicos tecnológicos crezca a pasos acelerados. El presente trabajo tiene un alcance de investigación descriptiva, enfoque cualitativo, diseño no experimental y su principal objetivo es describir como se implementó el sistema de gestión académica en el Instituto Tecnológico Superior Juan Bautista del cantón Daule y como esta implementación mejoró de manera eficaz y eficiente las gestiones académicas de la institución. El sistema fue desarrollado por docentes del Instituto Tecnológico Superior Juan Bautista Aguirre y la implementación de los primeros requerimientos estuvo en 8 meses, para lograr este objetivo fue necesario la conformación de grupos en temas como: diseño, desarrollo, análisis y capacitación. El sistema cuya característica fue integral, actualmente se le continúa realizando modificaciones acordes a los estándares de calidad que exige el modelo de educación superior en el país.

Palabras claves: software, sistema, procesos, académico, diseño

ABSTRAC

The technical and technological training in recent years in Ecuador has had a high impact due to the rapid insertion of professionals required by our labor market, causing the number of students entering the classrooms of technological technical institutes to grow at an accelerated pace. The present work has a scope of descriptive research, qualitative approach, non-experimental design and its main objective is to describe how the academic management system was implemented in the Juan Bautista Superior Technological Institute of the Daule canton and how this implementation improved effectively and efficiently the academic efforts of the institution. The system was developed by teachers

of the Juan Bautista Aguirre Higher Technological Institute and the implementation of the first requirements was in 8 months, to achieve this objective it was necessary to form groups on topics such as: design, development, analysis and training. The system whose characteristic was integral, is currently continuing to make modifications according to the quality standards required by the model of higher education in the country.

Keywords: software, system, processes, academic, design

INTRODUCCIÓN

La gestión académica es un proceso complejo que involucra varios actores como los son docentes, estudiantes y personal administrativo. Algunas instituciones de educación superior gestionan estos procesos de manera manual y física, no contando con su sistema informático que les permita llevar un registro y control de sus actividades.

El acceso a la información, el tiempo de respuesta y la interacción de los usuarios con el sistema son claves al realizar un análisis de la gestión académica que actualmente realizan las instituciones de educación superior. Desde su creación hasta el año 2016 el Instituto Tecnológico Superior “Juan Bautista Aguirre” almacenaba la información de registro de estudiantes, matrícula, acta de calificaciones, reporte de notas, estadísticas, etc. en Microsoft Excel y Microsoft Word, dificultando el tiempo de acceso a la información, pues en ocasiones la información digital ya no existía debido a la presencia de virus u otros factores de riesgos informáticos y climáticos, dejando como respaldo solo los documentos físicos que se almacenan en la bodega de secretaría. El elaborar acta de calificaciones, estados de matrículas, reportes de calificaciones, listado de estudiantes desertores, repetidores, fichas técnicas etc., demoraba horas, días, semanas y meses esto en función del grado de la información que se manipulaba, en ciertos casos estos resultados no tenían coherencia.

Dado estos antecedentes, el instituto para manejar de manera eficiente su gestión académica comprendió la necesidad de implementar un sistema informático que le permitiera optimizar el tiempo de respuesta en la entrega de documentos y acceso a la información.

Fundamentación teórica

Sobre los diferentes procesos académicos y administrativos dentro de un ambiente universitario, es fundamental que la implementación práctica de diferentes aplicaciones informáticas concluidas, favorezcan una automatización enfocada en los procesos y actividades de la institución. Unas de las metas de un sistema informático está el ofrecer reportes de salida que nutren los indicadores de la planificación y evaluación institucional. Acosta *et al* (2017) determina como instrumento de planificación estratégica al cuadro de mando integral y la evaluación del desempeño de los docentes a tiempo completo y parcial, con este instrumento se espera aportar de forma significativa a la toma de decisiones estratégicas, operativas y tácticas en áreas de la gestión universitaria (Ver Figura1).

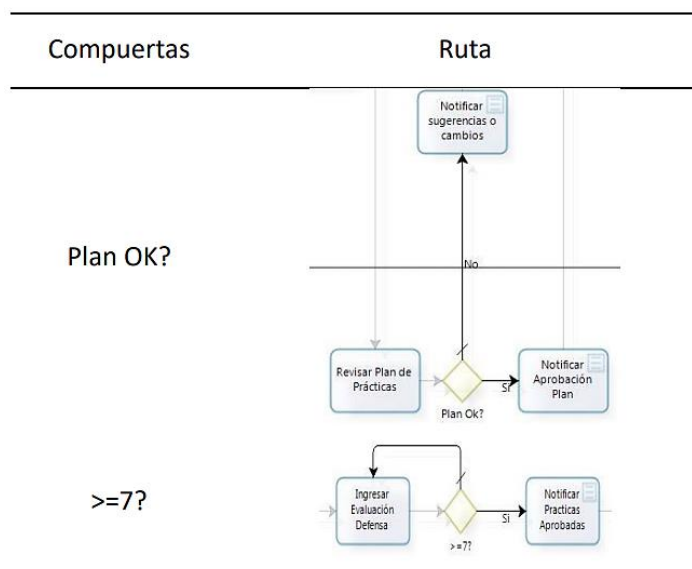
N°	Módulos del SIE-GU y aplicaciones
Academia	
1	Sistema Académico- SA
2	Sistema de Actualización de Datos- SADTS
3	Ficha Socioeconómica- FSE
4	Repositorio Digital- DSPACE
5	Sistema de Seguimiento del Silabo-SSS
6	Sistema de Homologaciones-SH
7	Sistema de Evaluación a Docentes- SED
Investigación	
8	Entorno Virtual de Aprendizaje- EVA
9	Sistema de Administración de Bibliotecas- SAB
10	Sistema de Proyectos-SP
Vinculación	
11	Sistema de Administración de Herramientas Psicológicas- SAHP
12	Sistema de Encuestas- SE
13	Sistema de Graduados- SG
14	Sistema Consultorios Jurídicos Gratuitos- SCJG
Administración	
15	Sistema de Registro de Personal- SRP
16	Sistema de Archivos e Inventarios- SAI
17	Sistema Contable-SC
18	Sistema de Gestión Documental-Help Desk (SHD)
19	Sistema de Planificación Universitaria (SPU)
20	Sistema de Evaluación Interna- SEI
21	Sistema de Escuela de Conducción (SEC)

Figura 1. Módulos del SIE-GU y sus aplicaciones
Fuente: (Acosta, Becerra, & Jaramillo, 2017)

En la evaluación de desempeño es relevante que, al contar con sistema de gestión, este desarrolle un registro y validación de las diferentes actividades académicas como administrativas, para que represente un insumo que garantice la calidad de procesos que en este rubro se desarrollan en cada una de las dependencias que integran la comunidad académica (López, 2018). Sin embargo, dicha información debe estar integrada entre cada carpeta o expediente de alojamiento, donde se pueda medir el desempeño de las actividades, este desempeño sería útil no solo para mantener un control de todas ellas, en función de los criterios de evaluación por los distintos organismos acreditadores.

Se espera que la gestión en una institución de educación superior, así como el servicio entregado a los estudiantes sea el resultado de la interacción transversal de las diferentes coordinaciones, departamentos y áreas estructuradas de forma vertical, bajo un punto de vista organizacional. En consecuencia, el desarrollo de una automatización esencialmente compleja representa una orientación a la sistematización, desarrollo y consolidación de los procesos orientados a tecnologías BPM universitarias (Aushay & García, 2017). Un aspecto relevante en el desarrollo de un sistema de gestión está en abstraer las cualidades esenciales del proceso de gestión, sintetizar, modelar y controlar la dinámica del flujo de trabajo de PPP (Ver Tabla 1) a través del uso de la BPMS en congruencia con la gestión de procesos de negocio. Bajo este esquema se podrá asegurar sustancialmente la eficiencia en el grado de cumplimiento de objetivos funcionales en la gestión de procesos académicos.

TABLA I. EJEMPLOS DE RUTAS UTILIZADAS EN LA AUTOMATIZACIÓN DEL PROCESO PPP.



Fuente: (Aushay & García, 2017)

Cortázar (2015), determina que el sistema de gestión enfocado en los procesos académicos, no busca llegar a ser como un sistema de información en gestión de la educación (EMIS), ya que el mismo está orientada a la identificación, medición y análisis que se encuentra en una materia o en una institución en particular. Por lo tanto, el diseño de una plataforma con estas características propone ahorrar tiempo, automatizar procedimientos, posibilita la utilización en la planificación estratégica, controlar materias, controlar recursos, realizar seguimiento a estudiantes y docentes, evaluar los procesos de aprendizaje y como sustento a la toma de decisiones (Ver figura1).



Figura 2. Descripción de los resultados de cada actividad

Fuente: (Cortázar, 2015)

A partir del análisis documental desarrollado en el presente apartado, se estima que la información por si sola es solo datos, es decir que son pasivos y estáticos. Dicha declaración determina que no se modifica hasta que los miembros de la institución de educación superior, le agreguen valor hasta convertirla en conocimiento, a partir de ese punto la información es activo y dinámico para la organización, tanto a nivel interno como

externo. Sin embargo, los Sistemas de Información de Gestión deben ser herramientas totalmente eficientes bajo este contexto. En consecuencia, el sistema debe ser capaz de suministrar información oportuna y relevante a las autoridades para la toma de decisiones relativas al funcionamiento de la institución, como también de apoyo a la investigación y a la extensión en función de la complejidad de todos los procesos académicos y administrativos (Garita, 2015).

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos, el desarrollo de software tiene que ver con la creación de sistemas informáticos utilizando varios lenguajes de programación cuya función es dar solución a un problema detectado. En 1958, el nombrado matemático estadístico John Tukey, acuñó el término software. El término Ingeniería de Software se utilizó por primera vez en el título de una conferencia en la OTAN celebrada en Alemania en 1968. La IEEE Computer Society publicó las primeras transacciones en Ingeniería del Software en 1972. El comité creado por la IEEE Computer Society para el desarrollo de estándares de ingeniería de software se fundó en 1976 (The Institute of Electrical and Electronics Engineers, Inc., 2004)

Para el desarrollo de software o sistemas informáticos se deben conocer las metodologías de desarrollo de software y de entre estas rescatar las más usadas. Actualmente para el desarrollo de proyectos se utilizan metodologías ágiles.

“Las metodologías ágiles son sistemas de gestión de proyectos que nos ayudan a usar el tiempo de manera efectiva y creativa. Evitan que no nos olvidemos de algunas de ellas. Permiten visualizar de manera clara las tareas a realizar.”
(Wingu, 2016, pág. 5)

De entre la gama de metodologías ágiles, la mencionada con mayor frecuencia es la Scrum, esta es una metodología que surgió para administrar de manera dinámica proyectos de desarrollo de Software, pero se puede adaptar para el trabajo en nuestras organizaciones. Al comenzar cada ciclo se definen qué tareas se van a realizar a lo largo del mismo y al finalizar se entregan resultados concretos. Además, plantea una división de roles entre el equipo de trabajo lo que fomenta la cooperación entre compañeros (Wingu, 2016, pág. 8).



Figura 3. Metodología Scrum
Fuente: (Scrum Manager, 2016)

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.). Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto de un determinado proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiante (Canós, Letelier, & Penadés, 2003).

Para el desarrollo de todo sistema informático se debe considerar la arquitectura de red, lenguaje de programación, base de datos y ciclo de vida que tendrá el sistema.

Senn (1992), afirma: “para el desarrollo de un sistema se debe considerar el ciclo del vida del sistema el mismo que tiene las siguientes fases: Investigación preliminar, determinación de requerimientos, diseño del sistema, desarrollo del sistema, prueba de sistema, implementación” (pág. 33).

Kendall & Kendall (2011), afirman:

El ciclo de vida del desarrollo de sistemas consta de siete fases 1. Identificación de los problemas, oportunidades y objetivos 2. Determinación de los requerimientos humanos de información 3. Análisis de las necesidades del sistema 4. Diseño del sistema recomendado 5. Desarrollo y documentación del software 6. Prueba y mantenimiento del sistema 7. Implementación y evaluación del sistema. (pág. 8).

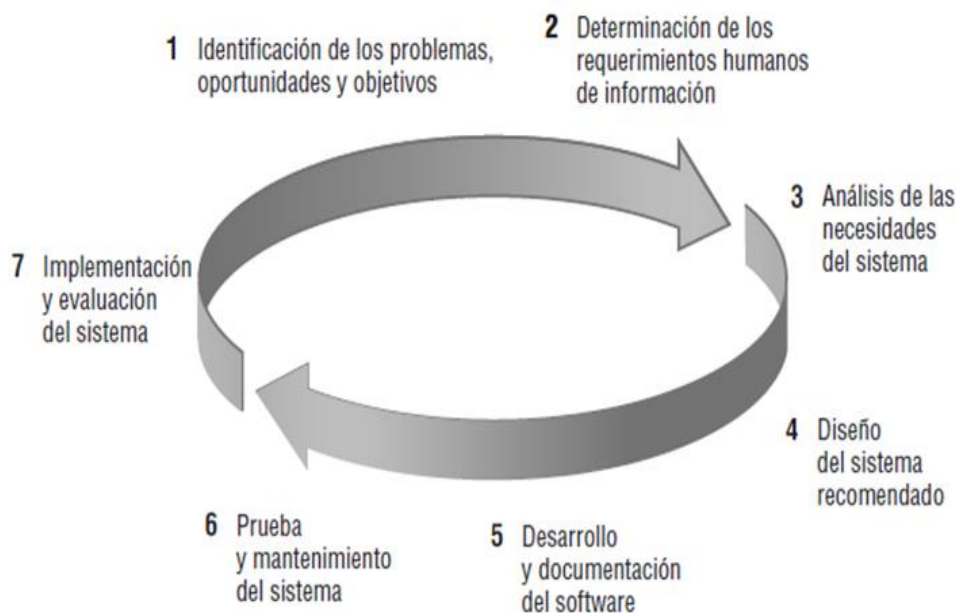


Figura 4. Las 7 fases del ciclo de desarrollo de sistemas
Fuente: (Kendall & Kendall, 2011)

MATERIALES Y MÉTODOS

Para el desarrollo de este proyecto se utilizó los siguientes materiales:

TABLA II. MATERIALES Y MÉTODOS PARA IMPLEMENTACIÓN SGA

Materiales			
Software		Hardware	Métodos
<ul style="list-style-type: none"> • Sistema Operativo Windows 7 • Base de datos MySQL Server 7 • WorkBench MySQL 6.3 	<ul style="list-style-type: none"> • 10 Laptops • Procesador Intel i3 Mínimo • Disco Duro de 500 Gb Mínimo • Memoria RAM 6 Gb Mínimo 		
<p>Herramientas de Diseño</p> <ul style="list-style-type: none"> • Dreamweaver CS6 (u otro.) • Netbeans 	<ul style="list-style-type: none"> • Servidor de Datos (Hosting) • Servidor Web (Hosting) 		<ul style="list-style-type: none"> • Arquitectura cliente servidor • Programación orientada a objetos
<p>Lenguaje de Programación</p> <ul style="list-style-type: none"> • PHP • AJAX • JS 	<p>Unidades de almacenamiento</p> <ul style="list-style-type: none"> • Pen Drive • Cd –Rom • Drive 		

Fuente: Elaboración propia

El concepto de cliente-servidor, se refiere a un modelo de comunicación que vincula a varios dispositivos informáticos a través de una red. El cliente, en este marco, realiza peticiones de servicios al servidor, que se encarga de satisfacer dichos requerimientos (Pérez Porto & Gardey, 2018).

El modelo cliente-servidor fue el método aplicado en el desarrollo del Sistema de Gestión Académica (SGA), este consiste en la distribución de las tareas, las aplicaciones “clientes” realizan peticiones a uno o varias aplicaciones “servidor”, los servidores deben estar en ejecución para poder atender los requerimientos de los clientes, el modelo permite diversificar el trabajo de la tal manera que las aplicaciones clientes no se sobrecarguen. En este modelo tanto el cliente como el servidor son consideradas como entidades abstractas que pueden en la misma maquina o en máquinas diferentes.

Desarrollar un software con una arquitectura cliente-servidor permite que los accesos, recursos e integridad de los datos sean controlados por el servidor, evitando que un cliente no autorizado cause daños al sistema, además facilita los mantenimientos.

Los Sistemas de Gestión Académica que actualmente utilizan las instituciones de educación superior, como en los Institutos Técnicos Tecnológicos y Universidades son desarrollados en arquitectura cliente-servidor, ejemplo de ello es la implementación del Sistema de Gestión Académica de la Facultad de Ingeniería en Electricidad y Computación de la Escuela Superior Politécnica del Litoral (Galio Molina, 2011).

En la arquitectura cliente-servidor el remitente de una solicitud es considerado como cliente, y recibe las repuestas del servidor, puede conectarse a varios servidores a la vez e interactúa con los usuarios finales mediante la interfaz gráfica de usuario. Al receptor

de la solicitud se los denomina servidor, este espera la solicitud, la procesa y envía una respuesta al cliente, puede recibir varias peticiones a la vez.

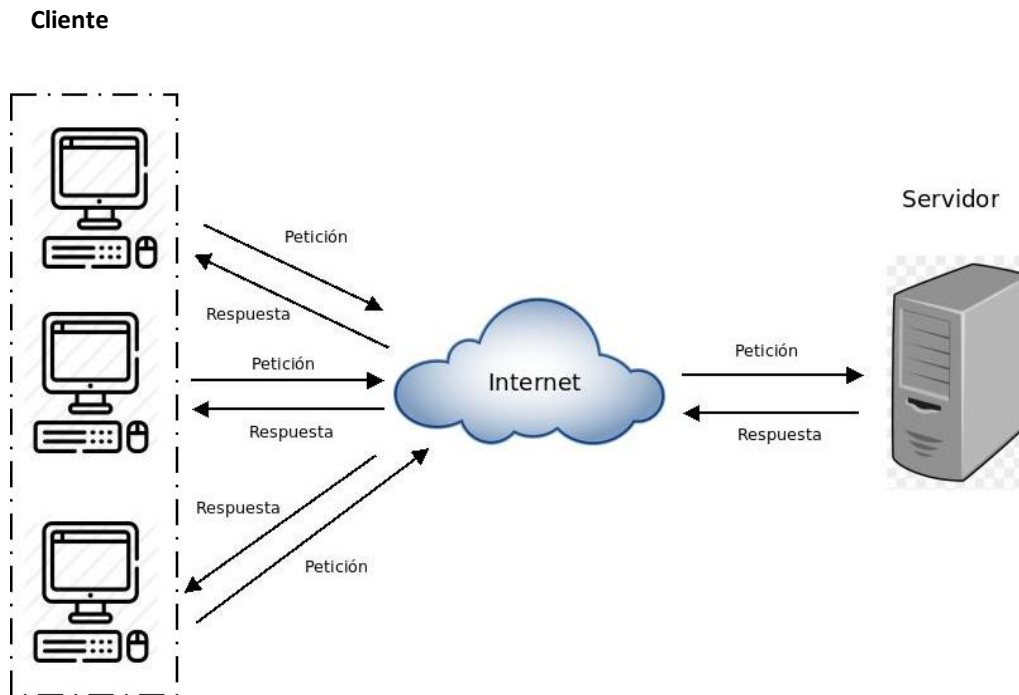


Figura 5. Modelo cliente - servidor
Fuente: (Schiaffarino, 2019)

Existen diferentes tipos de arquitecturas cliente-servidor: la arquitectura de dos capas, de tres capas y la de n capas. En la arquitectura de dos capas el cliente solicita recursos y el servidor responde directamente a la solicitud con sus propios recursos. Eso significa que el servidor no requiere de una aplicación extra para proporcionar parte del servicio. En la arquitectura de tres capas existe un nivel intermediario, la arquitectura generalmente está compartida por un cliente que solicita los recursos equipado con una interfaz de usuario o mediante un navegador web, la capa del medio es denominada software intermedio cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo. La última capa es el servidor de datos que proporciona al servidor de las aplicaciones necesarias para poder procesar y generar el servicio que solicitó el cliente en un principio. La arquitectura de n capas está compuesta por n servidores, donde cada uno de ellos brindan un servicio específico (Schiaffarino, 2019).

Cuando se desarrolla un software, sistema o aplicación informática, siempre surge la pregunta ¿Bajo qué paradigma de programación está desarrollado? ¿Por qué es importante el paradigma? ¿En qué influye? A continuación, se detallan algunas definiciones de paradigma de programación.

“Un paradigma de programación indica un método de realizar cómputos y la manera en que se deben estructurar y organizar las tareas que debe llevar a cabo un programa” (Departamento de Informática Universidad de Valladolid, 2012).

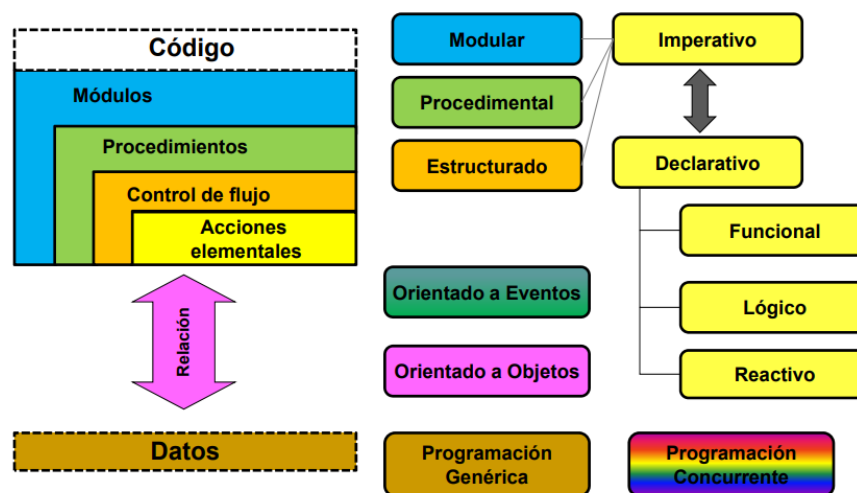


Figura 5: Tipos de paradigmas
Fuente: (Departamento de Informática Universidad de Valladolid, 2012)

“Un paradigma de programación es una colección de patrones conceptuales que moldean la forma de razonar sobre los problemas, de formular soluciones y de estructura programas. Los paradigmas de programación son: programación imperativa, funcional, lógica, orientada a objetos” (Rodríguez Salas, Santamaría Arana, Rabasa Dolado, & Martínez Bonastre, 2003, pág. 4).

En la programación imperativa, un programa es una secuencia finita de instrucciones, que se ejecutan una tras otra, los datos utilizados se almacenan en la memoria principal y se referencian utilizando variables, los lenguajes de programación que utilizan este paradigma son Pascal, Ada, Cobol, C, Modula 2 y Fortran. La programación funcional, todas las sentencias son funciones, un programa es una función que se define por composición de funciones más simples, el lenguaje de programación que utiliza este paradigma es LISP. En la programación lógica, en un programa se declaran hechos y reglas, después de pregunta por el resultado, ejemplo de un lenguaje de programación que utiliza este paradigma es Prolog. La programación orientada a objetos es un estilo de programación, este puede ser funcional o lógico, lo que lo caracteriza es la forma de manejar la información basada en los conceptos: clase, objeto, herencia. Ejemplo de lenguajes que utilizan este paradigma. Smalltalk, C++, Visual, Java, Php, etc.

Es importante estar al tanto del paradigma de programación con el que trabajó el equipo de desarrollo, de esta manera se puede conocer de qué forma está estructurado el software y que tan amigable o compatible puede ser con otros lenguajes de programación, así como la portabilidad de los datos. Entonces se puede decir que el paradigma que utiliza el equipo de desarrollo si influye en las nuevas tendencias de desarrollo de software.

La programación orientada a objetos consiste en considerar como “objetos” los sujetos de interés del programa, cada tipo de elemento diferente es definido con sus datos y los métodos de tratamiento de estos. Esto permite, básicamente, tratar estos objetos como un tipo de datos más, sobre los cuales aplicar métodos y crear construcciones que realizarán cálculos y manipulaciones. Estos nuevos tipos de datos se construyen a partir de los tipos básicos que incluye el lenguaje de programación, y las operaciones se declaran como procedimientos o funciones. (Alsina González, 2017)

La programación orientada a objetos es una técnica para diseñar y desarrollar software que se puede incorporar a una aplicación web o páginas web, el lenguaje más utilizado en este tipo de programación es PHP, lenguaje en el que fue desarrollado el SGA.

“PHP es un popular lenguaje de scripting de propósito general que es especialmente adecuado para el desarrollo web. Es rápido, flexible y pragmático, PHP impulsa todo, desde su blog hasta los sitios web más populares del mundo” (PHP, 2019).

PHP (Hypertext Preprocessor) es uno de los lenguajes de programación más utilizados en la actualidad, se utiliza mayormente para desarrollo de sitios web, pero para muchos ya es un lenguaje de propósito general, esto se debe a que es libre y gratuito, es decir, que no necesita de licencia y su código es abierto permitiendo a los programadores sacar provecho de sus beneficios, funciona en todas las versiones de Windows, posee soporte para múltiples base de datos tales como: DBase, Informix, Interbase/Firebird, Microsoft SQL Server, Msq, MySQL, Oracle, PostgreSQL, SQLite, Sybase (Minera, 2008).

Utilizar PHP y MySQL en grandes proyectos permite depurar, generar autenticidad y personalización de usuarios, así como crear un carro de compras, o un sistema de administración de contenidos, también permite crear servicios de correo electrónico basado en la web, crear foros web, generar documentos personalizado en formato pdf, conectarse a servicios web con XML y SOAP (Welling & Thomson, 2005).

MySQL es uno de los sistemas de gestión de base de datos relacional de código abierto más popular en el mundo, lo utilizan grandes empresas como Facebook, Google, Adobe, Alcatel Lucent y Zappos; gracias a que es un sistema confiable que les permite ahorrar tiempo y dinero (MySQL, 2018).

RESULTADOS Y DISCUSIÓN

Para el desarrollo del Sistema de Gestión Académico (SGA), se tomó como referencia la guía al cuerpo del conocimiento de la Ingeniería de Software SWEBOK, cuyo principal objetivo es promover una visión consistente de la ingeniería de Software en el mundo. Así como también se aplicó el ciclo de vida para el desarrollo de sistemas propuestos por Senn (1992) y Kendall & Kendall (2011).

Swelok es una guía creada por Software Engineering Coordinating Committee, promovido por la IEEE Computer Society, que proporciona a los desarrolladores de software directrices para el diseño y la creación de software, este documento recopila experiencia de experto en el área de software con la finalidad de estandarizar los procesos de desarrollo a nivel mundial.

En el diseño e implementación de Sistema de Gestión Académica (SGA), se utilizó una arquitectura cliente-servidor (programación orientada a objetos), lenguajes de programación PHP AJAX, base de datos MySQL Server 7, MySQL WorkBench 6.3, herramientas de diseño Dreamweaver CS6, Netbeans. Se aplicó la metodología Scrum, la cual propone que se establezcan los roles y tiempos de entrega al empezar el proyecto, para poder llevar un mejor control de los avances. Se conformó un equipo de trabajo de 10 personas distribuidas de la siguiente manera: 1 Jefe de proyecto, 6 Analistas, 3 Desarrolladores.

TABLA III. ROL Y FUNCIONES

Rol	# Personas	Perfil Profesional	Funciones
Jefe de Proyecto	1	Ingeniero en Sistemas Computacionales Magister en Sistemas Integrados de Gestión	Determinar los requerimientos del sistema Velar por el cumplimiento de tareas y tiempos asignados para cada una de las actividades. Presentar a Rectorado los avances del proyecto Dar solución a los problemas suscitados
Analistas de Sistemas	6	Ingeniero en Sistemas Computacionales Licenciado en Sistemas Computacionales	Realizar el levantamiento de la información Dar soporte a los usuarios Capacitar a los usuarios Elaborar manuales de usuario
Desarrolladores	3	Ingeniero en Sistemas Computacionales	Diseñar la base de datos e interfaz gráfica de usuario Administrarla base de datos Programar en los lenguajes de programación establecidos. Implementar la plataforma. Realizar las pruebas del sistemas y los test de seguridad

Fuente: Elaboración propia

La implementación del Sistema de Gestión Académica se desarrolló en las siguientes fases: 1) levantamiento de información, 2) diseño del sistema, 3) desarrollo del sistema, 4) prueba del sistema, 5) implementación.

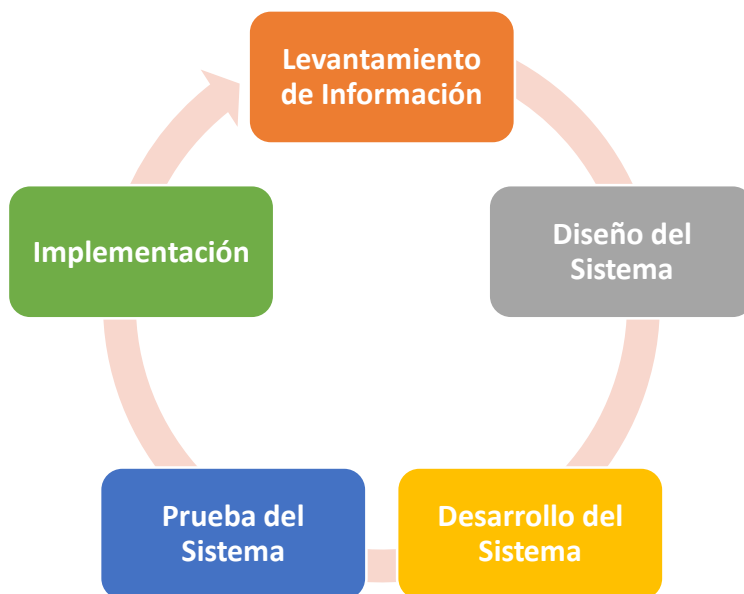


Figura 6. Implementación Sistema de Gestión Académica SGA

Fuente: Elaboración propia

1. Levantamiento de información

El levantamiento de información se realizó en dos etapas, antes de realizar el software y cuando el software ya estuvo desarrollado, el antes sirvió para determinar los requerimientos del sistema y el después sirvió para migrar todos los datos a la plataforma.

La observación directa y la entrevista a: autoridades, personal administrativo, docente, estudiantes, permitieron determinar los requerimientos del sistema y así establecer los tiempos y fases de desarrollo. El proceso de levantamiento de información tuvo una duración de 27 días, en este tiempo a más de escuchar los requerimientos propuestos por las autoridades, se verificó los archivos físicos y se trabajó en el área de matriculación que demanda mayor atención, se pudo constatar el volumen y flujos de información por día y el nivel de recurrencia a los datos, esto permitió verificar que los requerimientos estén acorde a las necesidades de la institución. Es importante que en la fase de levantamiento de información se conozca la misión, visión, políticas y reglamentos de la institución.

2. Diseño y desarrollo del Software

Después del levantamiento de información se analizó los requerimientos, determinando como prioridad para desarrollar los módulos que se detallan a continuación:

- Matriculación de estudiantes
- Generación de acta de calificaciones
- Generación de reporte record académico

En la fase de diseño, se planteó el modelo de la base de datos y los diagramas de caso de uso que permitieron a los desarrolladores tener claro el flujo de los procesos y el modelo del negocio. En la fase de desarrollo, los desarrolladores programaron en los lenguajes de programación y en base a los tiempos establecidos.

3. Implementar el software

Para la implementación del sistema, primero se capacitó al personal docente y administrativo sobre el funcionamiento del sistema, luego se pidió a los usuarios que accedan al sistema y así constatar el tiempo de respuesta, los permisos de acceso y el soporte de sistema, posteriormente se realizó el levantamiento de información de los datos de estudiantes, matriculas, notas, distributivos, horarios etc., permitiendo corregir los errores técnicos de diseño y seguridad.

RESULTADOS

El proyecto Sistema de Gestión Academia (SGA) se empezó a desarrollar en marzo del 2016, estuvo terminado en octubre del 2016 (8 meses) y se realizó de la siguiente manera.

TABLA IV: RESULTADOS DE LAS ACTIVIDADES DEL PROYECTO

Actividad	Duración	Fecha de Inicio	Fecha de Fin
Levantamiento de información	27 días	07/03/2016	12/04/2016
Soporte a matriculación	5 días	04/04/2016	08/04/2016
Diseño de base de datos	15 días	11/04/2016	29/04/2016

Rediseño y modificación de base de datos	10 días	02/05/2016	13/05/2016
Creación de procedimientos y vistas	15 días	16/05/2016	30/05/2016
Creación y definición de formularios y pantallas (Diseño y botones)	20 días	30/05/2016	24/06/2016
Desarrollo del sistemas (Programación de procesos)	45 días	27/06/2016	26/08/2016
Prueba del sistema	10 días	29/08/2016	09/09/2016
Modificación de procesos (según las pruebas)	15 días	12/09/2016	30/09/2016
Implementación del Sistema	1 día	03/10/2018	03/10/2018

Fuente: Elaboración propia

La fase levantamiento de información duro 27 días.

La fase de Diseño de software tuvo una duración de 60 días.

La fase de Desarrollo de software se realizó en 45 días.

La fase de prueba y corrección de errores tuvo una duración de 25 días.

La implementación estuvo lista octubre del 2016 y contó con los siguientes módulos:

- 1) Gestión de matriculación
- 2) Gestión de estudiantes
- 3) Gestión de carrera por periodo académico
- 4) Gestión de récord Académico
- 5) Gestión de usuarios
- 6) Gestión de perfiles y permisos (seguridad)
- 7) Reportes

El 6 de junio del 2018 en el evento Repensando la Educación Superior en Ecuador, América Latina y el Caribe: a Cien Años de la Reforma Universitaria de Córdoba organizado por la Secretaria de Educación Superior Ciencia, Tecnología e Innovación, el Instituto Tecnológico Superior Juan Bautista Aguirre muestra su sistema informático de Gestión Académica (SGA) como aporte al desarrollado de la educación superior. En este evento funcionarios de SENESCYT deciden que el sistema es óptimo para ser implementado a nivel nacional por los resultados de gestión obtenidos en el instituto y es ahí donde se emprende el proyecto SIGA (Sistema integral de Gestión Académica).

En un estudio realizado por Universidad Autónoma de Tamaulipas-México, se determinó los PYMES prefieren los sistemas informáticos que proporcionan información a tiempo, actualizada, útil, relevante, exacta, con buen nivel de detalle y fácil de interpretar (Abrego Almazán, Sánchez Tovar, & Medina Quintero, 2017).

Actualmente el Instituto Tecnológico Superior Juan Bautista Aguirre, cuenta con el Sistema de Gestión Académica (SGA), el sistema está operativo y funcionando, hasta la presente fecha, donde más de 1400 usuarios interactúan de manera dinámica sin tener ningún tipo de complicaciones. El sistema es web, por lo tanto los usuarios pueden acceder al mismo desde la comodidad de sus casas, entre los usuarios que interactúan con el sistema tenemos: estudiantes, docentes, personal administrativo y autoridades.

Se recomienda realizar un estudio que permita determinar el nivel de satisfacción de los usuarios y recoger cada una de las anécdotas y experiencias que permitan otorgar la mejora continua. Los sistemas que no se actualizan constantemente y que no satisfacen las necesidades del usuario, son propensos a ser reemplazados por otros sistemas que cumplan con estas expectativas.

CONCLUSIONES

Para el desarrollo e implementación de un Sistema de Gestión Académica es necesario: 1) conocer los procesos académicos y administrativos que maneja la institución para la que se va a desarrollar, 2) tener claro el ciclo de vida de los sistemas informáticos, 3) identificar la metodología de desarrollo de software que se ajuste a los requerimientos, 4) dominar el paradigma de programación, así como los lenguajes de programación y los gestores de base de datos a implementar.

El talento humano fue un factor importante en el desarrollo de este proyecto, quienes hicieron posible la cristalización de una idea que surgió por una necesidad de mejorar los procesos de Gestión Académica. Fueron los docentes del instituto que desarrollaron el sistema en sus horas de gestión, fines de semana y feriado. Se pudo comprobar que al definirse los roles, actividades y tiempos de ejecución, los resultados son satisfactorios, además el liderazgo y compromiso en el proyecto fueron un factor primordial en el desarrollo de este tipo de iniciativas.

La implementación del Sistema de Gestión Académica (SGA) mejoró de manera eficiente los procedimientos que desarrolla el instituto en toda su gestión académica. La información que genera el sistema permite a las autoridades tomar decisiones en el menor tiempo posible y con mejores resultados.

El sistema es integral y dinámico, debido a que cada día se incrementan módulos que mejoran la gestión académica logrando que la seguridad, permisos y perfiles de usuarios sean más restringido y acorde a las funciones que desarrolla cada funcionario.

REFERENCIAS

- Abrego Almazán, D., Sánchez Tovar, Y., & Medina Quintero, J. (2017). Influence of information systems in organizational performance. *Science Direct*, 332-334.
- Acosta, L., Becerra, F., & Jaramillo, D. (2017). Sistema de Información Estratégica para la Gestión Universitaria en la Universidad de Otavalo (Ecuador). *Formación Universitaria*. doi:<http://dx.doi.org/10.4067/S0718-50062017000200011>
- Alsina González, G. (15 de Octubre de 2017). *Programación Orientada a Objetos*. Sitio: *Definición ABC*. Obtenido de <https://www.definicionabc.com/tecnologia/programacion-orientada-objetos.php>
- Aushay, A., & García, V. (2017). Modelación, simulación y automatización de procesos en la gestión de servicios académicos universitarios. *3c Tecnología*, 32-51.
- Canós, J., Letelier, P., & Penadés, M. (2003). Metodologías Ágiles en el Desarrollo de Software. *Universidad Politécnica de Valencia*, 2-8.

- Cortázar, G. (2015). Optimización de sistemas de gestión académica. Una propuesta de gestión, medición y procesamiento de datos en un entorno virtual de aprendizaje para la toma de decisiones en instituciones educativas. *Revista Escuela de Administración de Negocios*, 80-97.
- Departamento de Informática Universidad de Valladolid. (2012). Paradigmas de Programación. España.
- Galio Molina, G. (25 de Enero de 2011). *Facultad de Ingeniería en Electricidad y Computación*. Obtenido de https://www.researchgate.net/publication/279488820_Implementacion_de_un_sistemas_de_gestion_academica_como_soporte_a_la_toma_de_decisiones_para_el_analisis_evaluacion_seguimiento_y_control_actividades_docentes_utilizando_el_sistema_abet_basado_aplicac
- Garita, W. (2015). Reflexiones e insumos sobre los Sistemas de Información Gerencial en la gestión de las instituciones de educación superior: Ventajas y desventajas. *Gestión de la educación*, 23-37.
- Kendall, K., & Kendall, J. (2011). *Análisis y Diseño de Sistemas*. México: Pearson.
- López, L. (2018). Gestión de calidad. *Encuentro Unicach*, 2.
- Minera, F. (2008). *Curso de Programación PHP* (Primera edición ed.). Lomas de Zamora: Manuel Users.
- MySQL. (09 de Noviembre de 2018). *MySQL.Com*. Obtenido de <https://www.mysql.com/why-mysql/>
- Pérez Porto, J., & Gardey, A. (01 de Noviembre de 2018). *Definición de*. Obtenido de <https://definicion.de/cliente-servidor/>
- PHP. (30 de 04 de 2019). *PHP.Net*. Obtenido de <https://www.php.net/>
- Rodríguez Salas, J., Santamaría Arana, L., Rabasa Dolado, A., & Martínez Bonastre, O. (2003). *Introducción a la programación. Teoría y práctica*. San Vicente: Club Universitario.
- Schiaffarino, A. (12 de Marzo de 2019). *Infranetworking*. Obtenido de <https://blog.infranetworking.com/modelo-cliente-servidor/>
- Scrum Manager . (Julio de 2016). Guía de Formación Versión 2.6.
- Senn, J. (1992). *Análisis y Diseño de Sistemas de Información*. Mexico: McGraw-Hill.
- The Institute of Electrical and Electronics Engineers, Inc. (2004). Guía al cuerpo de conocimiento de la Ingeniería del Software SWEBOOK. California, Estados Unidos.
- Welling, L., & Thomson, L. (2005). Desarrollo web con php y mysql php 5 y mysql 4.1 y 5: disco compacto. Madrid, España.

Wingu. (Agosto de 2016). Manual de Metodologías Ágiles. Buenos Aires, Argentina.