UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO

DE INGENIERO EN SISTEMAS COMPUTACIONALES

TEMA:

DISEÑO E IMPLEMENTACIÓN DE LA ARQUITECTURA DE SOFTWARE EMPRESARIAL PARA LA EMPRESA PÚBLICA LA UEMPRENDE CON EL FIN DE MEJORAR EL RENDIMIENTO ORGANIZACIONAL A TRAVÉS DE LA TECNOLOGÍA.

AUTOR:

EDUARDO STALIN TUSA VITAR

DIRECTOR:

Msc. MAURICIO REA

Ibarra – Ecuador

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto de Repositorio Digital Institucional, determina la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1003869730
NOMBRES:	EDUARDO STALIN TUSA VITAR
DIRECCIÓN:	IBARRA, Lajas 2-178 y San Luis
EMAIL:	steduardito4@gmail.com estusa@utn.edu.ec
TELÉFONO FIJO:	062631431
TELEFONO MOVIL:	0988337648

DATOS DE LA OBRA	
TÍTULO:	DISEÑO E IMPLEMENTACIÓN DE LA ARQUITECTURA DE SOFTWARE EMPRESARIAL PARA LA EMPRESA PÚBLICA LA UEMPRENDE CON EL FIN DE MEJORAR EL RENDIMIENTO ORGANIZACIONAL A TRAVÉS DE LA TECNOLOGÍA.
AUTOR (ES):	EDUARDO STALIN TUSA VITAR
FECHA:AAAAMMDD	
SÓLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☐ PREGRADO ☐ POSTGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
ASESOR/DIRECTOR:	MSC. MAURICIO REA

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Eduardo Stalin Tusa Vitar, con cédula de identidad Nro. 1003869730, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación del trabajo en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

EDUARDO STALIN TUSA VITAR

1003869730

CESIÓN DE DERECHOS DE AUTOR

A FAVOR DE LA UNIVERSIDAD TECNICA DEL NORTE

Yo, Eduardo Stalin Tusa Vitar con cédula de identidad Nro. 1003869730, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5, 6, en calidad de autor del trabajo de grado denominado **"DISEÑO E IMPLEMENTACIÓN DE LA ARQUITECTURA DE SOFTWARE EMPRESARIAL PARA LA EMPRESA PÚBLICA LA UEMPRENDE CON EL FIN DE MEJORAR EL RENDIMIENTO ORGANIZACIONAL A TRAVÉS DE LA TECNOLOGÍA."**, que ha sido desarrollado para optar por el título de Ingeniero en Sistemas Computacionales, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

(Firma):.....

Nombre: Eduardo Stalin Tusa Vitar

Cédula: 1003869730

Ibarra, 12 de octubre de 2015

CERTIFICACIÓN DEL ASESOR

El señor Eduardo Stalin Tusa Vitar, ha trabajado en el desarrollo del proyecto de tesis "Diseño e implementación de la arquitectura de software empresarial para la empresa pública la Uemprende con el fin de mejorar el rendimiento organizacional a través de la tecnologia.", previo a la obtención del Título de Ingeniero en Sistemas Computacionales, realizándola con interés profesional y responsabilidad, lo cual certifico en honor a la verdad.

Msc. Mauricio Rea

DIRECTOR DE TESIS

CONSTANCIA.

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 08 días del mes de julio del 2015.

AUTOR	**ACEPTACIÓN**
.....	.....
Eduardo Stalin Tusa Vitar	Ing. Ana Ximena Quinteros
C.C.: 1003869730	**Gerente Subrogante de la Empresa Pública La Uemprende EP**

Facultado por resolución de Consejo Universitario_____

DEDICATORIA

A Dios, a mi mamá: Mariana Vitar, a mi papá: Eduardo Tusa, y a mi familia por el apoyo incondicional que me brindaron y me siguen entregando en cada paso que me llevó a culminar este proyecto.

AGRADECIMIENTO

“La vida es un reto que todos debemos vencer”, es el lema que siempre he llevado desde los primeros días que ingresé a la universidad, una carrera como lo es Ingeniería en Sistemas Computacionales, resulta un poco difícil, pero no imposible terminarla, de allí que agradezco en primera instancia a Dios, a mi Mamá: Mariana Vitar y a mi Papá: Eduardo Tusa, quienes a través del tiempo fueron mi soporte en aquellos momentos de adversidad y dificultad, entregándome su apoyo incondicional en todos los sentidos, siempre estaré muy agradecido con ellos por el resto de mi vida.

A mi hermana mayor: Viviana Tusa, quien con sus consejos me daba aliento en cosas que al principio parecían no ser tan importantes, pero que luego se plasmaron en mi mente para salir adelante en todo.

A mis docentes de aula, coordinadores de carrera, decanos y personal administrativo, con quienes compartí esta vida universitaria, como si fueran compañeros de aula, llevándome gratas experiencias de aprendizaje, así como también siendo buenos amigos de cada uno de ellos

Finalmente a mi director de trabajo de grado, mi estimado docente y amigo Msc. Mauricio Rea, con quien desde un principio me he sentido apoyado y guiado, siendo el pilar fundamental en este proceso de investigación, ayudándome con su extensa experiencia y paciencia para culminar con éxito el presente proyecto.

RESUMEN

Dentro del contexto de una infraestructura tecnológica en cualquier empresa siempre surge la necesidad de realizarse el cuestionamiento de ¿Cómo empiezo a automatizar mis procesos?, ya que es indudable que si una empresa se aleja del plano tecnológico, se aleja no solo de nuevas oportunidades en el mercado que lo rodea, sino que también se minimiza a un pequeño negocio sin metas ni objetivos.

De allí que surge la necesidad de que toda empresa debe tener su propia plataforma tecnológica, con el único fin de generar un mayor rendimiento de la misma, puede ser en diferentes ámbitos específicos y diferentes para cada una, por ello contar con una "Arquitectura de Software (AS)" es fundamental.

Una AS, es la esencia del ramal tecnológico que acoge a todos los procesos existentes, ya que si no sabemos lo que vamos a hacer, entonces corremos el riesgo de que nuestros proyectos fracasen y por resultado simple, la misma empresa será afectada, es por ello que para estar al día en el mundo globalizado, se debe tener bien diseñada, desarrollada y ejecutada nuestra AS, con procesos prioritarios, operativos y estratégicos, que conlleven a una armonía tecnológica entre la lógica de negocio y un sistema informático desarrollado con cada uno de los lineamientos como lenguaje de programación, sistemas gestores de bases de datos, frameworks de desarrollo, en caso de ser usados, y como prioridad el usuario final el cual será quien determine la funcionalidad del sistema implementado y desarrollado.

Las inversiones pueden ser un tanto extensas, pero debemos verlas como eso, como inversiones, no como un gasto que no conlleve ninguna ganancia para la institución.

ABSTRACT

Within the context of a technological infrastructure in any company always the need arises the question: how can I beginning to automate processes of my enterprise?, as it is clear that if a company moves away from the technological level, departs not only of new opportunities of the market, it also minimized to a small business without goals or objectives.

Hence it arises the need for every company should have its own technology platform, with the sole purpose of generating a higher performance on it, it can be in different specific and different fields for each one, thus having a software architecture it's fundamental.

A AS, is the essence of technological branch that assemble all existing processes, because if we do not know what we will do, then we risk our projects fail and the same company will be affected, this is why to keep up in a globalized world, our AS should be well designed, developed and executed, with priority, operational and strategic processes that lead to technological harmony between business logic and a computer system developed with each one of guidelines, as a programming language, databases management systems , development frameworks, should be used, and as a priority the end user who will determine the functionality of the system implemented and developed.

Investments can be somewhat high, but we should see them like that, as investments, and not as an expense that does not involve any profit for the institution.

Índice de contenidos

Índice de ilustraciones

Índice de gráficos

Índice de tablas

CAPITULO I

- ANTECEDENTES
- SITUACIÓN ACTUAL
- JUSTIFICACIÓN
- DESCRIPCIÓN DEL PROBLEMA
- OBJETIVOS
- ALCANCE

1. Introducción – Empresa La Uemprende EP

1.1. Antecedentes

La empresa pública La Uemprende E.P. es una empresa anexa a la Universidad Técnica del Norte, desde su creación se encuentra inmersa en proyectos macro generando utilidades para la UTN[1].

La empresa se constituye legalmente el martes 10 de septiembre de 2013, mediante el oficio No. 185-HCU-UTN (Universitario, 2014), donde se realizan todas las consideraciones pertinentes.

Misión

"Contribuir al desarrollo del país a través de la prestación de servicios, diseño de proyectos, consultorías, construcción e infraestructura y provisión de bienes a los campos social-cultural, económico productivo, ambiental y político-institucional como aporte al bienestar de los Ecuatorianos" (Universitario, 2014).

Visión

"En el 2016 será una empresa exitosa, competitiva a nivel nacional que contribuya al desarrollo del país, con servicios de calidad y que genere bienestar económico social y ambiental a sus clientes" (Universitario, 2014).

[1] Universidad Técnica del Norte.

Valores Institucionales

Dentro de mismo contexto la empresa se maneja por los valores institucionales, con los cuales fomenta la competitividad y el correcto desempeño de cada una de sus dependencias, entre ellos encontramos, la responsabilidad económica, social y ambiental, el compromiso que tiene con la sociedad, la transparencia en la gestión realizada y la pertinencia. (Universitario, 2014)

Al ser una entidad pública relativamente nueva no posee una infraestructura tecnológica debidamente investigada e implementada, lo que conlleva a no obtener los resultados deseados en la ejecución de los procesos administrativos que a diario se ejecutan dentro y fuera de la provincia.

Desde el Directorio ejecutivo conformado por el Sr. Rector de la UTN, un delegado del HCU[2] y un representante de los decanos. (Universitario, 2014), conjuntamente con la gerencia se ha optado por definir los parámetros específicos a seguir para el mejoramiento continuo de la empresa, esquematizando que las tecnologías de la información son indispensables para su correcto desarrollo.

Tomando en cuenta los diferentes aspectos que la componen y sus necesidades, además teniendo en cuenta su misión y visión institucional es imperativamente importante un documento guía que maneje la estructura de los procesos administrativo – tecnológicos, específicamente en el área de desarrollo de sistemas informáticos a corto y largo plazo.

[2] Honorable Consejo Universitario

Servicios que ofrece la Empresa Pública La Uemprende EP.

Capacitación

Dirigida a incrementar capacidades individuales e institucionales al sector público y/o privado en las diversas áreas del conocimiento.

Consultorías.

Diseño y Ejecución.

Construcción e infraestructura.

Almacén Universitario.

El Almacén Universitario brinda el servicio de venta de libros, materiales, papelería, bazar, para ofrecerlos a Docentes, Empleados, Estudiantes y público en general que estén relacionados con los diferentes programas de estudio de cada una de las Facultades de la Universidad.

1.2. Situación Actual

En la actualidad la Empresa Pública "La Uemprende EP[3]" tiene el reto de convertirse en la empresa pionera en el ámbito de prestación de servicios en la región uno de país (Universitario, 2014), lo cual genera cierta incertidumbre al ser una entidad relativamente nueva que no cuenta con la infraestructura tecnológica para afrontar dicha demanda de servicios, es por ello que todavía no se tiene una plataforma tecnológica debidamente estudiada y probada que coadyuve a afrontar los retos que posee.

Mencionaré algunos casos; en este momento la implementación de sistemas informáticos no posee una arquitectura específica para nuestra empresa, por lo que su desarrollo es poco

[3] Empresa Pública

confiable, también no se cuenta con un sistema de registro para las capacitaciones que a diario se realizan en los diferentes cursos que se imparten, obteniendo así un resultado poco eficiente con la información de los participantes y docentes que a corto y largo plazo podrían ser nuevamente parte de nuestros cursos.

Actualmente entre los objetivos primordiales de la empresa pública La Uemprende EP se encuentra la necesidad de sistematizar los procesos que se generan a diario en las diferentes áreas como adquisiciones, contratos, cursos, talleres, seminarios, estudios y de los ejemplos antes expuestos.

Entonces utilizando el análisis de situación actual se puede saber que al no contar con una arquitectura de software ágil en la creación y desarrollo de aplicaciones informáticas que optimicen sus procesos, conlleva a la pérdida de tiempo, clientes e información de mucha importancia para sus autoridades, generando retraso en los compromisos adquiridos con sus clientes.

1.3. Organización de la empresa

La empresa cuenta actualmente con dos departamentos principales, que son:

1.1.1.1 Departamento Administrativo Financiero

1.1.1.2 Departamento de Tecnologías de la Información y Comunicación

Dentro de los cuales se derivan las siguientes dependencias:

- Talento Humano
- Financiero
- Administrativo

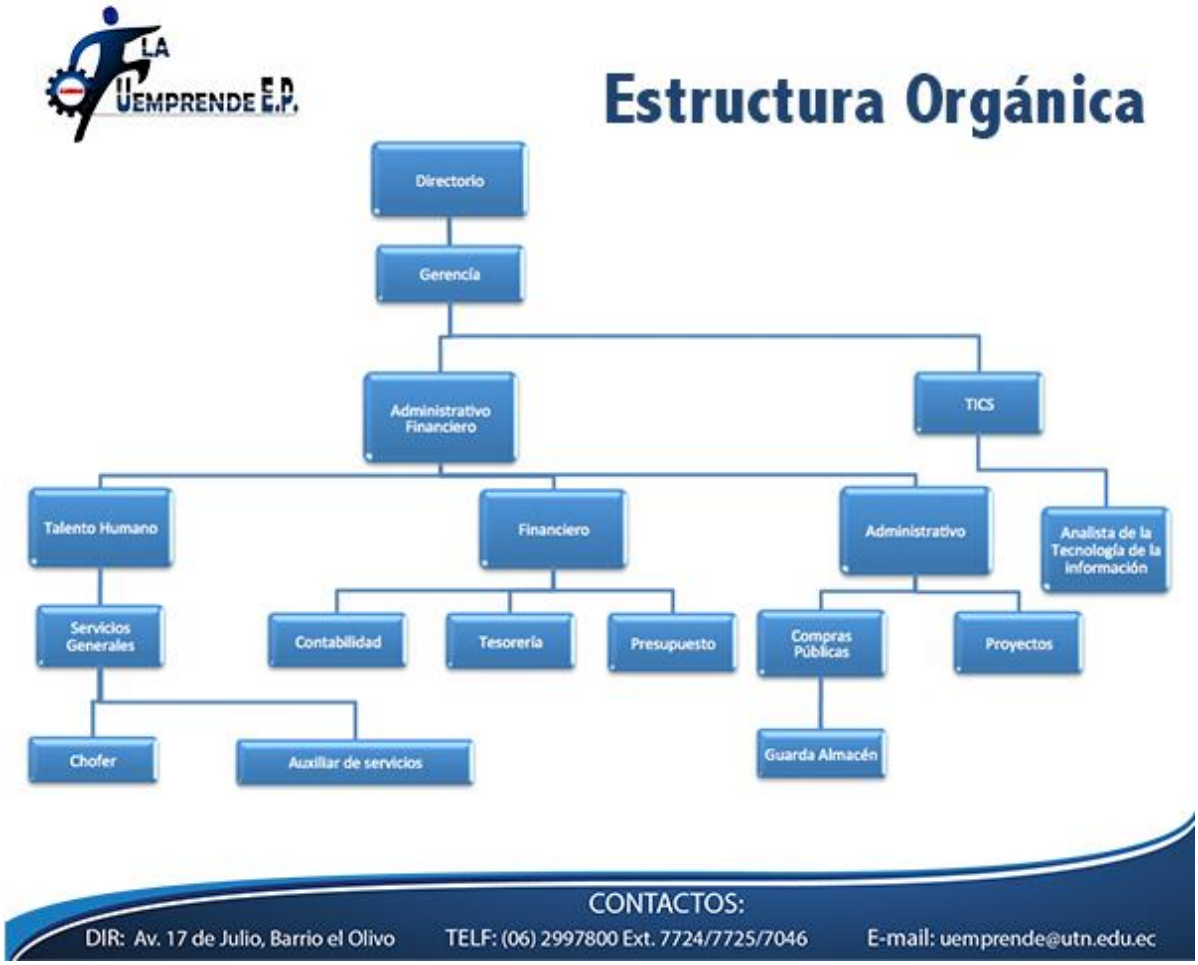

Ilustración 1: Organigrama Uemprende EP

Fuente: (Universitario, 2014)

1.4. Justificación

Se plantea el presente tema de investigación ya que la empresa no cuenta con una arquitectura empresarial de software que le permita llevar con eficiencia los procesos administrativos – tecnológicos, dificultando así el correcto desarrollo y crecimiento a corto y largo plazo en áreas específicas como el desarrollo de software e implementación y adquisición de hardware, generando atributos bajos de calidad de software sin tener la certeza de su correcto funcionamiento ya que al no contar con una arquitectura probada se corre el riesgo del fracaso e ineficiencia en desarrollo y ejecución, es por ello que la presente arquitectura pretende resolver dichos problemas generados mejorando la productividad de desarrollo y gestión de la empresa.

Por lo tanto será de gran beneficio para la entidad pública contar con un documento técnico debidamente desarrollado, que será de gran utilidad a corto y largo plazo, beneficiando a sus autoridades y también a sus clientes, ya que se agilizarán todos los procesos tanto de desarrollo como de ejecución de proyectos cuando esta arquitectura sea desarrollada en los sistemas informáticos.

1.5. Formulación del Problema

Dificultad de diseñar una arquitectura de software para una empresa debido a la diversidad de arquitecturas y tecnologías existentes en el mercado, lo que ha generado una brecha entre los procesos administrativos con la tecnología. Además que los procesos todavía se los lleva en documentos de texto, hojas de Excel, e inclusive a mano en hojas de papel, siendo otro gran inconveniente el no contar con el talento humano necesario en el área de TIC[4].

[4] Tecnologías de la Información y Comunicación.

1.6. Objetivos

1.6.1. Objetivo General

Desarrollar una arquitectura de software empresarial que permita automatizar los procesos de la Uemprende E.P. de manera integral enfocándose al mejoramiento de los procesos existentes.

1.6.2. Objetivos Específicos

- Recolección de información y análisis de situación inicial de la empresa La Uemprende Empresa Pública.
- Creación del mapa de procesos mediante la caracterización de la información.
- Diseñar la arquitectura de software para la empresa La Uemprende Empresa Pública.
- Implementación del prototipo con la arquitectura diseñada.
- Realizar el análisis costo beneficio conjuntamente con las conclusiones y recomendaciones.

1.7. Alcance

El presente trabajo de investigación se encargará de realizar la arquitectura de software para la empresa pública La Uemprende, con el fin de proveer una herramienta de TIC para el correcto desarrollo y funcionamiento de los sistemas informáticos, entregando la documentación para su posterior implantación, teniendo en cuenta que la investigación contempla la caracterización de procesos de la empresa, ya que el levantamiento formal de procesos conlleva un esfuerzo que sale del alcance de este proyecto.

Cabe destacar que el tema principal de la investigación es el diseño de la arquitectura, ya que la empresa cuenta con 6 servicios específicos, siendo el punto central de mi investigación el servicio de capacitaciones, para demostrar el presente trabajo se realizarán pruebas con el prototipo del módulo de capacitaciones, entendiéndose por prototipo a la representación del sistema de capacitaciones, aunque no es un sistema completo, posee parte de las características del sistema final, entre ellas constará el módulo de inscripciones, usuarios, auditoría, quedando a consideración de la empresa eliminarlo, reutilizarlo o reciclarlo.

Para el desarrollo del prototipo se utilizará como metodología de desarrollo XP[5] por la facilidad y rapidez de culminación del sistema – prototipo, como lenguaje de programación PHP 5.x, utilizando la base de datos PostgreSql versión 9.

[5] Extreme Programming

1.8. Diagrama de bloques de subsistemas.

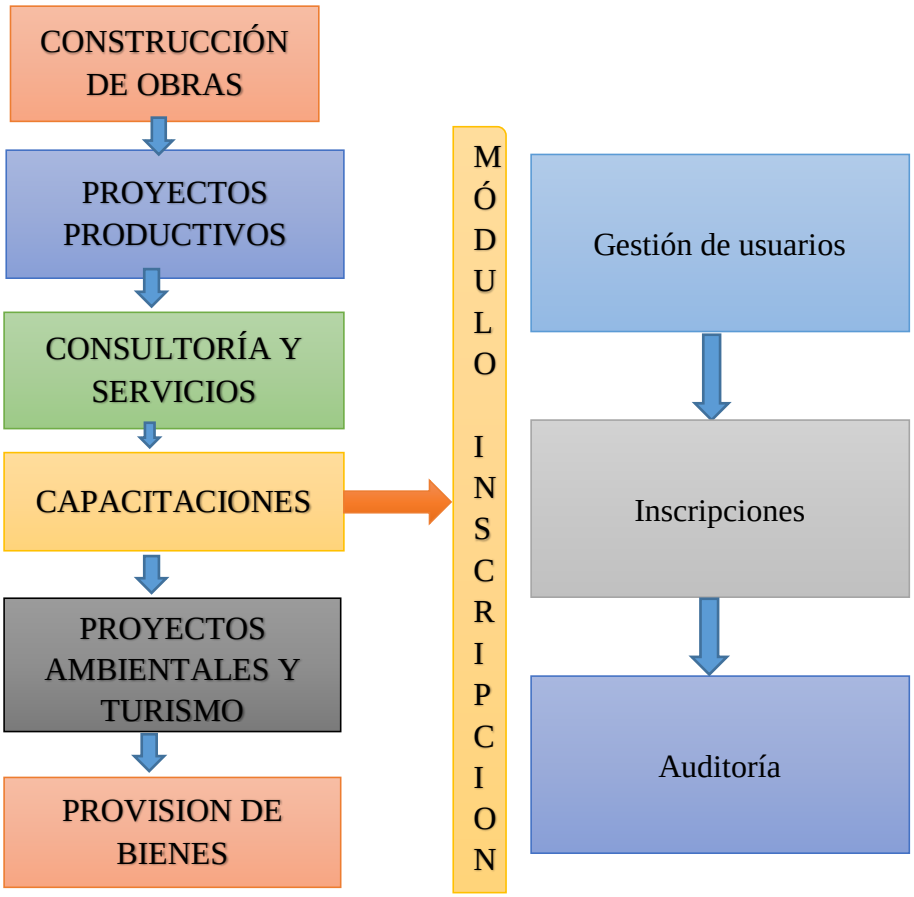

Ilustración 2: Diagrama de sistemas

Fuente: El Autor

1.9. ¿Qué son las empresas públicas?

Conceptualmente el término de empresa pública comprende dos dimensiones importantes, lo público y lo empresarial.

La dimensión pública se refiere a la pertenencia o propiedad y control por parte de autoridades públicas, las cuales deben depender al interés público y a las políticas públicas.

En cambio la dimensión empresarial es específicamente los campos en los cuales se va a desenvolver la empresa, como por ejemplo mercantiles, de construcción e infraestructura, finanzas, servicios, etcétera. (jurídicas, 2012).

1.10. Principios de las Empresas Públicas

En el Ecuador, de acuerdo a la constitución del 2008, las EP[1] se rigen de acuerdo a los siguientes principios:

- Ejecutar el buen vivir en todo su sentido enfocándose en el desarrollo humano.
- Promover y practicar el desarrollo sustentable, integral, descentralizado y desconectado del estado, así como también de las actividades económicas asumidas (Auditores, 2012)

1.11. Tipos de Empresas Públicas

En Realidad en el Ecuador las empresas públicas no tienen una clasificación ya que todas comprenden un mismo fin, que es ser parte del estado, descentralizando los servicios y generar utilidades para las entidades en donde fueron conformadas, sobrellevando así un correcto funcionamiento.

[1] Empresas Públicas

CAPITULO II

Marco Teórico

- ¿Qué es Arquitectura de Software?
- ¿Qué es un Arquitecto de Software?
- Patrones de Diseño
- Patrones de Arquitectura
- Características de los patrones de diseño y patrones de arquitectura.
- Metodologías ágiles de desarrollo de software
- Metodología XP
- Metodologías para diseño de arquitecturas de software.
- Planificador de Recursos Empresariales ERP

2.1. ¿Qué es Arquitectura de Software?

Dentro del marco del desarrollo de software empresarial e incluso en sistemas informáticos de bolsillo, cada informático se ha encontrado con un millar de conflictos en las diferentes etapas del ciclo de vida de los sistemas informáticos.

En gran medida las "complicaciones" existentes se debe esencialmente a la carencia o poco conocimiento de un correcto uso de la Arquitectura de Software, ya que en los últimos años cada profesional en desarrollo de software opta por realizar su propia arquitectura al momento de empezar un proyecto, sin tomar en cuenta los factores tanto internos como externos que siempre serán adyacentes al desarrollo de la programación.

Ahora entonces que implica un SI[2] ejecutado con una correcta Arquitectura de Software, en contraste una AS[3] nos permite tener flexibilidad y con una prospectiva amplia también tener adaptabilidad, especialmente en el mercado de software donde nada es estático.

2.2. Recursos económicos empresariales.

En cuanto a los recursos económicos empresariales nos permite reducir en cuanto a hardware y software los costos de mantenimiento, también amortiza el costo en tempranas fases de desarrollo y lo que es mejor, una AS nos sirve como una guía muy clara y entendible para el desarrollo lo que favorece al control de los proyectos, estableciendo un lenguaje común entre

[2] Sistema Informático

[3] Arquitectura de Software

los participantes de una área de TIC[4] ya sea empresarial o a nivel personal. (Facultad de Ingeniería - Universidad de Uruguay, 2013)

Para contrastar con la calidad de software final, se tomará el concepto de la definición "oficial", de la AS[5] que brinda el documento de la IEEE Std 1471-2000.

"*La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución."*

De dicho concepto podemos decir entonces que la AS es la organización de los componentes de un sistema que le otorgan su misión y visión. (Reynoso, 2014)

2.3. ¿Qué es un Arquitecto de Software?

Dentro del marco de la AS se encuentran definiciones muy parecidas a lo que es un Arquitecto de Software, conozcamos entonces sus características.

2.3.1. Características de un Arquitecto de Software

- Debe ser una persona llena de habilidades.
- Debe tener experiencia en desarrollo de software y dominar todas o parcialmente la mayoría de herramientas de tecnología.
- También debe poseer una excelente habilidad de comunicación.
- Poseer un alto espíritu de liderazgo.
- Debe ser proactivo.

---

[4] Tecnología de la Información y Comunicación

- Debe estar siempre orientado a los objetivos tanto de la empresa como personales. (Kruchten, 2010)

Definición

Atreverse a dar una definición de lo que es un arquitecto de software de manera generalizada deben tomarse diferentes aspectos, ya que existen varios tipos de arquitectos de software, encargados de realizar diferentes tareas, aunque se parezcan, nunca serán las mismas actividades.

Por ejemplo un arquitecto de software de aplicaciones o sistemas informáticos bancarios debe ser más minucioso en el control en las diferentes etapas del desarrollo del software, lo que no sucede en el diseño e implementación de un sistema informático para una pequeña empresa, como un supermercado local.

Teniendo en cuenta todas las especificaciones descritas anteriormente procedemos a definir los tipos de arquitectos de software.

2.3.2. Tipos de Arquitecto de Software

Arquitecto de Soluciones.

Cuando el tamaño del proyecto es demasiado amplio, es posible que exista más de un Arquitecto de Software, allí es donde el arquitecto de soluciones ejerce su papel, el cual es quien participa del proyecto desde su concepción, es decir la parte inicial del sistema.

Arquitecto de Software o de Sistemas.

Es el encargado de las decisiones en el diseño del software, generalmente también puede tomar las decisiones en el ámbito de hardware.

2.3.2.1. Arquitecto de Empresarial

Es el que generalmente abarca la mayoría de funciones, es decir puede llegar a desempeñar las funciones de Arquitecto de Soluciones, Arquitecto de Software e inclusive se lo puede llamar como un Arquitecto SOA.

2.3.2.2. Arquitecto SOA

Generalmente es parecido al Arquitecto Empresarial ya que sus funciones son netamente para empresas y proyectos a gran escala, abarcando a los diferentes tipos de arquitectos de software.

Finalmente cualquiera que sea el tipo de arquitecto de software, de manera general su rol dentro del proyecto va a ser siempre de tomar decisiones que van a afectar siempre de manera radical el desarrollo.

No obstante cada tipo de arquitecto es de suma importancia ya que no siempre en todos los casos se estará en un ambiente amplio de desarrollo, ya que generalmente son utilizados en pequeños de grande, mediana y pequeña escala. (Cervantes, 2011)

2.3.3. Actividades del Arquitecto de Software.

2.3.3.1. Concepción del Proyecto

Generalmente se lo conoce como etapa inicial en donde se debe generar una propuesta técnica y económica, se podría decir que una de las partes principales del desarrollo del sistema ya que se recoge los requerimientos iniciales que el cliente solicita, además de proporcionar una estimación de los tiempos de ejecución de todo el proyecto en general.

2.3.3.2. Requerimientos

Específicamente este punto concierne a la calidad del software, teniendo en cuenta las diferentes normas que deben ser tomadas en cuenta además deben estar debidamente justificadas y alineadas a los objetivos de negocio.

En caso de que el cliente solicite atributos de calidad muy exigentes, como por ejemplo alta disponibilidad 24/7 el Arquitecto de software se convierte en un negociador ya que debe tratar de negociar y establecer las métricas adecuadas para la AS.

2.3.3.3. Diseño del Sistema

En esta etapa el arquitecto de software debe hacer uso de todas las capacidades y aptitudes que posee ya que se requieren conocimientos técnicos avanzados al fin de establecer una solución técnica pertinente que satisfaga en la medida de lo posible todas las necesidades del cliente, ya que luego deberá presentárselas y saber explicar su diseño lo más comprensible que se pueda, responder preguntas acerca de su diseño y aceptar todas las observaciones en la medida que sea posible.

2.3.3.4. Construcción y pruebas del sistema

En esta etapa las actividades se reducen circunstancialmente hablando técnicamente, aunque esto no significa que las actividades se hayan terminado, todo lo contrario se debe emplear el tiempo en realizar las respectivas correcciones solicitadas o que tal vez se pasaron por alto.

Desde el punto de vista no-técnico las actividades se incrementan de manera paulatina a medida que los programadores realizan el sistema, ya que se debe tener una estrecha relación con cada participante del proyecto, y en efecto, es contraproducente llevar un desarrollo aislado, la razón es simple; el Arquitecto de Software debe supervisar que el sistema se desarrolle de acuerdo a la AS diseñada.

En lo referente a las pruebas, es de vital importancia que el Arquitecto de Software supervise las normas implementadas y la calidad del software.

2.3.3.5. Liberación

Una vez entrado en producción tanto la AS como el sistema informático y en un ambiente definitivo se realizan los ajustes finales necesarios para obtener un funcionamiento óptimo de la AS diseñada. (Cervantes, 2011)

2.4. Patrones de Diseño

De manera general los patrones de diseño son soluciones a los problemas que diariamente se presentan en el desarrollo de una AS y un sistema informático, es indispensable que todo Ingeniero en Sistemas y/o programadores sepa muy bien su utilización, ya que de ello depende que todo el trabajo realizado esté desarrollado en lo posible de la mejor manera.

Existen varios tipos de patrones de diseño que podemos o no tomarlos como solución en nuestro desarrollo, pero lo mejor es usar el sentido común y saber cuándo y cómo utilizarlos ya que se pueden usar prácticamente para cualquier sistema informático.

2.4.1. Tipos de patrones de diseño

Los patrones de diseño es una rama extensa de la ingeniería de software, su clasificación es muy extensa, dependiendo del tipo de tecnologías que se vaya a utilizar en el desarrollo de un SI[6], lo que conlleva a clasificarlos por el ambiente para cual son desarrollados, ya sea web o de escritorio.

Otra clasificación que también se puede citar de manera general es dependiendo del lenguaje de programación que se utilice, es decir tenemos patrones de diseño distintos para PHP[7], java, C# o también los famosos patrones de diseño GoF[8] que están destinados y enfocados directamente a la web.

A continuación vamos a explicar cada uno de los patrones más utilizados

[6] Sistema Informático

[7] Hypertext Pre-processor

[8] The Gang of Four

2.4.1.1. Patrones Sencillos

Facade.

Es en el cual se dispone de una sola interfaz para acceder al sistema, actuando como un único punto de ingreso, de una u otra forma sirve para facilitar su uso con el usuario. (Mondaray, 2012)

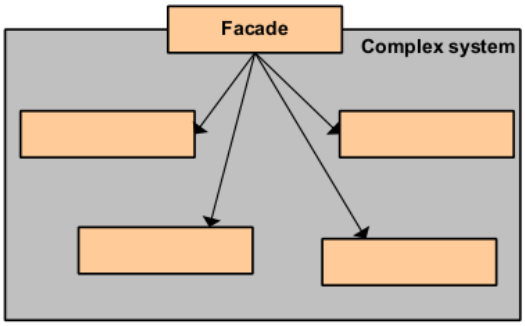

Gráfico 1: Facade

Fuente: ***(Mondaray, 2012)***

Singleton

Es uno de los patrones con los cuales es muy difícil la creación de nuevos objetos ya que una vez definido este patrón es imposible añadir objetos a las clases.

Como ejemplo de este patrón se expresa que Dios es un singleton. (Mondaray, 2012)

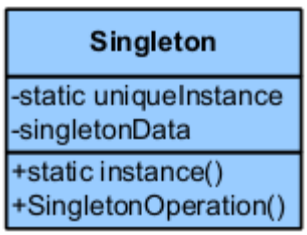

Gráfico 2: Singleton

Fuente: (Mondaray, 2012)

Mediator

Se esquematiza como su nombre lo indica, como un mediador con el cual los objetos que forman este patrón no se refieren entre ellos, de manera técnica es el uso de encapsulación en su estado más simple. (Mondaray, 2012)

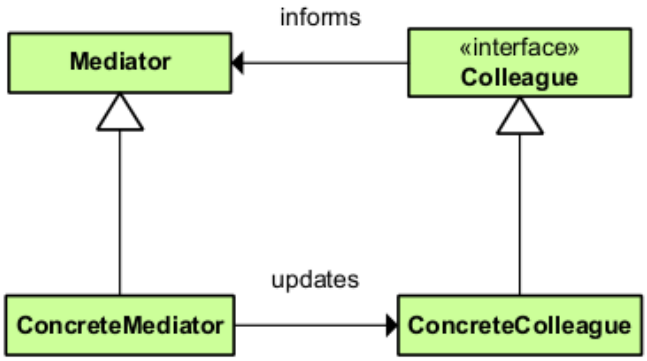

Gráfico 3: Mediator

Fuente: (Mondaray, 2012)

Método Factory

Es simplemente un método público creado en la misma clase para poder definir objetos de la misma clase.

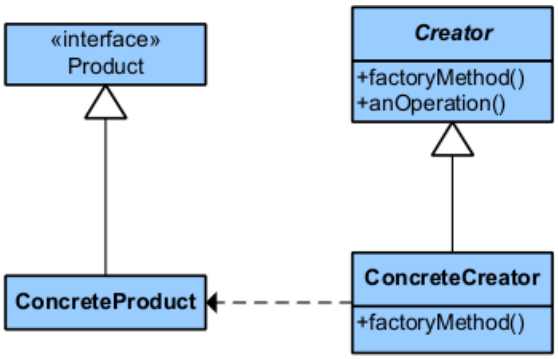

Gráfico 4: Factory

Fuente: (Mondaray, 2012)

Prototype

Es el patrón más simple que se pueda implementar, se trata básicamente de la creación de objetos, clonando otros, es como copiar un texto tal cual desde internet en un bloc de notas. (Mondaray, 2012)

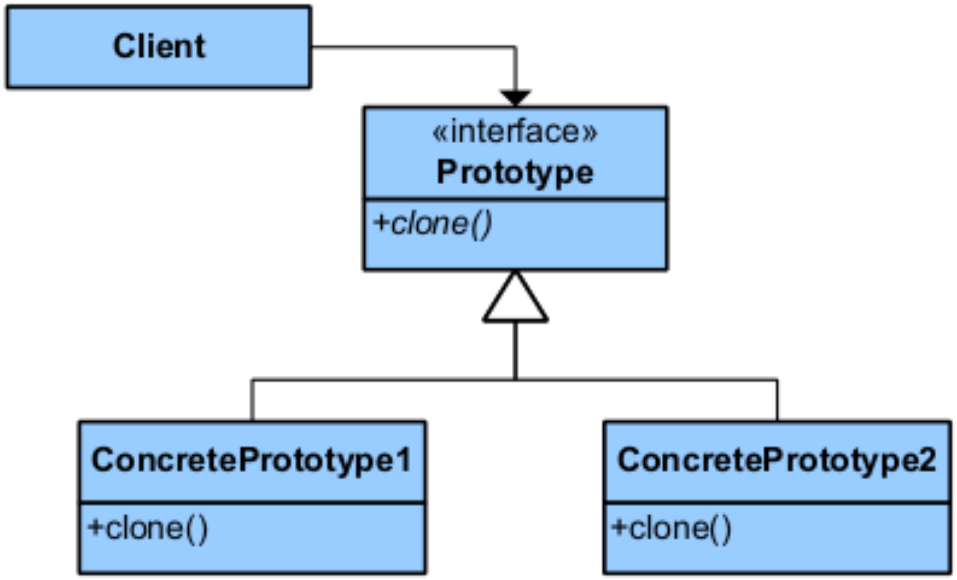

Gráfico 5: Prototype

Fuente: (Mondaray, 2012)

Como hemos podido apreciar, se ha procedido a nombrar los patrones más utilizados dentro de diferentes ámbitos, consecuentemente avanzaremos revisando los Patrones Intermedios.

2.4.1.2. Patrones Intermedios.

Proxy

Tal vez es uno de los más conocidos para ambientes de desarrollo web, su función principal es la de generar recursos.

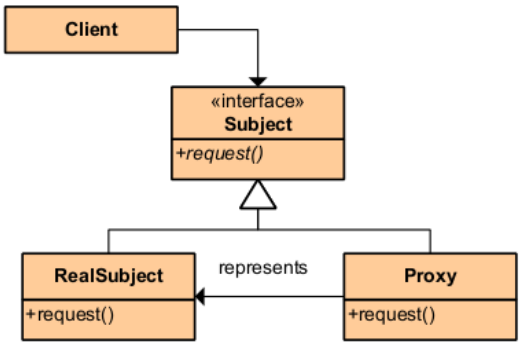

Gráfico 6: Proxy

Fuente: (Mondaray, 2012)

Adaptador (Adapter)

Se basa en la herencia, es decir mediante la creación de interfaces con el fin de poder utilizar las clases que no se pueden usar con una sola interfaz. (Mondaray, 2012)

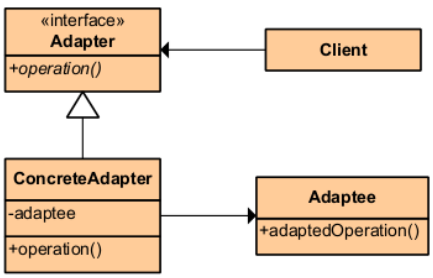

Gráfico 7: Adapter

Fuente: (Mondaray, 2012)

Bridge (Puente)

Basada en el principio del patrón anterior, Adapter, mientras este evita modificar las clases existentes, bridge relaciona la estructura de las interfaces que se desean implementar. (Mondaray, 2012)

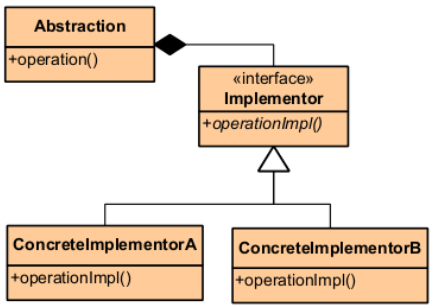

Gráfico 8: Bridge

Fuente: (Mondaray, 2012)

2.4.1.3. Patrones Avanzados.

Abstract Factory (De factor abstracto)

De manera relativa es uno de los más importantes ya que posee un sistema bastante útil, es decir procede a la creación de familias de clases, entendiendo por familias de clases a un sistema operativo diferente. (Mondaray, 2012)

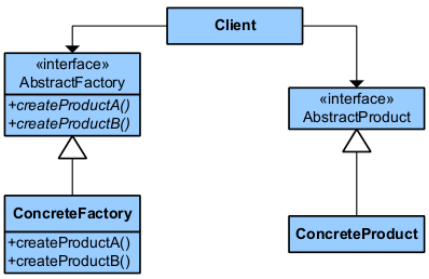

Gráfico 9: Abstract Factory

Fuente: (Mondaray, 2012)

2.5. Patrones de Arquitectura

Los patrones de arquitectura de manera resumida y comprensible son soluciones a problemas de arquitectura de software en la ingeniería de software, también se los conoce como "arquetipos"[9].

Un patrón arquitectónico es esencial al momento de realizar un sistema informático ya que contiene la organización estructural, si los comparamos con los patrones de diseño siempre, se tendrá mayor abstracción en los PDA[10] .

De manera recurrente siempre se confunde el término de PDA, con una AS como tal, ya que ambas ofrecen una imagen del sistema, la diferencia es que un PDA comunica una imagen solamente de ciertos elementos esenciales, más no de la arquitectura como tal.

Entonces podemos concluir que un PDA toma características esenciales de la AS, como por ejemplo, ciertos elementos de calidad, alta disponibilidad, esquemas que definen principalmente el rendimiento.

A continuación se procederá a describir algunos de los PDA más importantes y usados en los diferentes sistemas informáticos.

---

[9] Nombre resumido con el que generalmente se conoce a los Patrones de Arquitectura

[10] Patrones de Arquitectura

2.5.1. Patrón arquitectónico en Pizarra

Es un modelo de PDA que se utiliza generalmente en la inteligencia artificial, es decir en la elaboración de sistemas expertos, o en software multiagente, usando agentes lógicos, agentes basados en bases de conocimiento, etc. (Wikipedia, La enciclopedia libre, 2013)

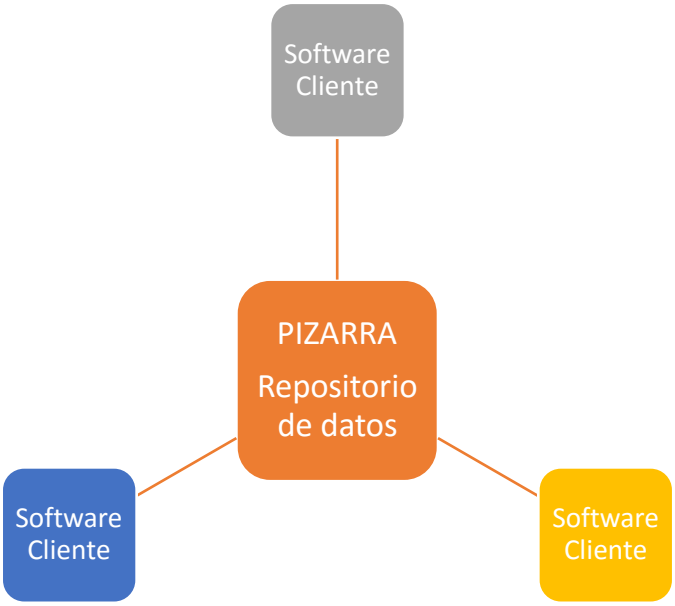

Ilustración 3: Pizarra

Fuente: El Autor

2.5.2. Patrón arquitectónico Cliente – Servidor.

Es uno de los más conocidos y aplicados por su versatilidad al momento de realizar proyectos pequeños y grandes, su entorno de desarrollo generalmente está destinado a ambientes de escritorio, donde los usuarios interactúan directamente con los datos que el servidor proporciona.

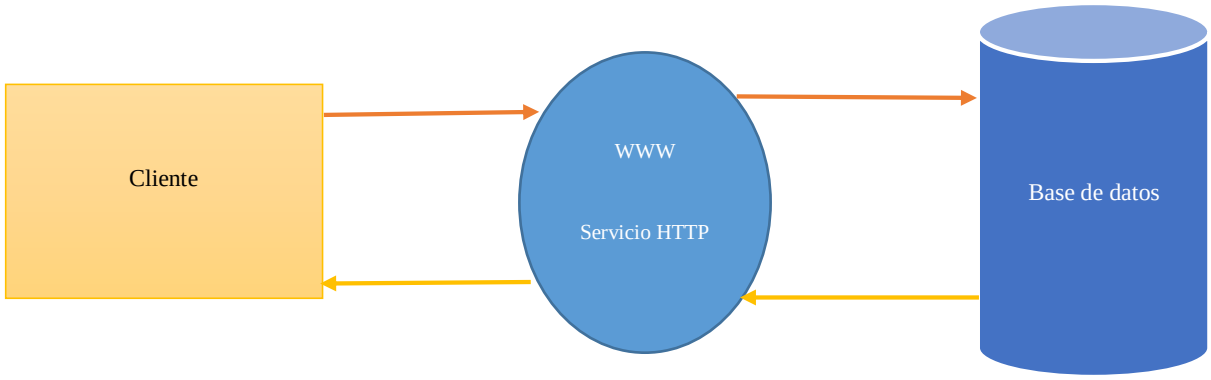

Ilustración 4: Cliente - Servidor

Fuente: El autor

2.5.3. Patrón arquitectónico Modelo – Vista – Controlador (MVC)

Aunque no es tan versátil como el Cliente – Servidor, es también uno de los más usados, tanto para entornos web, como de escritorio, donde se podría decir que usamos los llamados sistemas por capas, ya que la capa del modelo contiene la lógica de negocio con la cual se implementarán todos los llamamientos que realice el controlador, mientras que la vista únicamente muestra en un formato amigable al usuario la información y lógica de negocio.

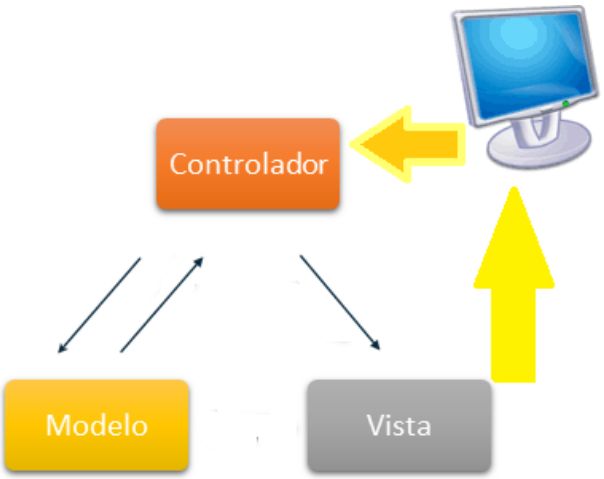

Ilustración 5: Modelo – Vista - Controlador

Fuente: El autor.

2.5.4. Patrón arquitectónico orientado a servicios (SOA[11])

En la actualidad es uno de los PDA que empieza a ser implementado en varias organizaciones, no sólo por su alta escalabilidad, sino por su alta reducción de costo al momento de implementación, además de ser muy flexible con la interacción de otros sistemas legados, e inclusive proporciona una alineación directa al proceso de negocio.

Dentro del marco del patrón arquitectónico SOA, se encuentran diversos patrones adicionales que lo conforman, inclusive se lo ha denominado como una AS de manera totalmente independiente, actualmente se la ha adoptado como una Arquitectura Empresarial (AE)[12] para grandes ambientes de producción y desarrollo de sistemas informáticos.

---

[11] Service Oriented Architecture

[12] Arquitectura Empresarial

Cada arquitecto de software tiene en sus manos la forma como se implementará la arquitectura SOA, ya que depende de manera específica de los requerimientos que realicen los funcionarios de la institución donde se la utilizará, por ello definir una ilustración se hace un poco relativo ya que sería un error mostrar una esquema generalizado de lo que es una arquitectura de software SOA.

Sin embargo existen factores indispensables que toda Arquitectura SOA debe poseer, como su principal componente, llamado Bus de Servicios el cual se encarga prácticamente de todas las transacciones que se realicen en esta AS.

A continuación procedemos a mostrar un pequeño esquema de los componentes básicos de una arquitectura SOA.

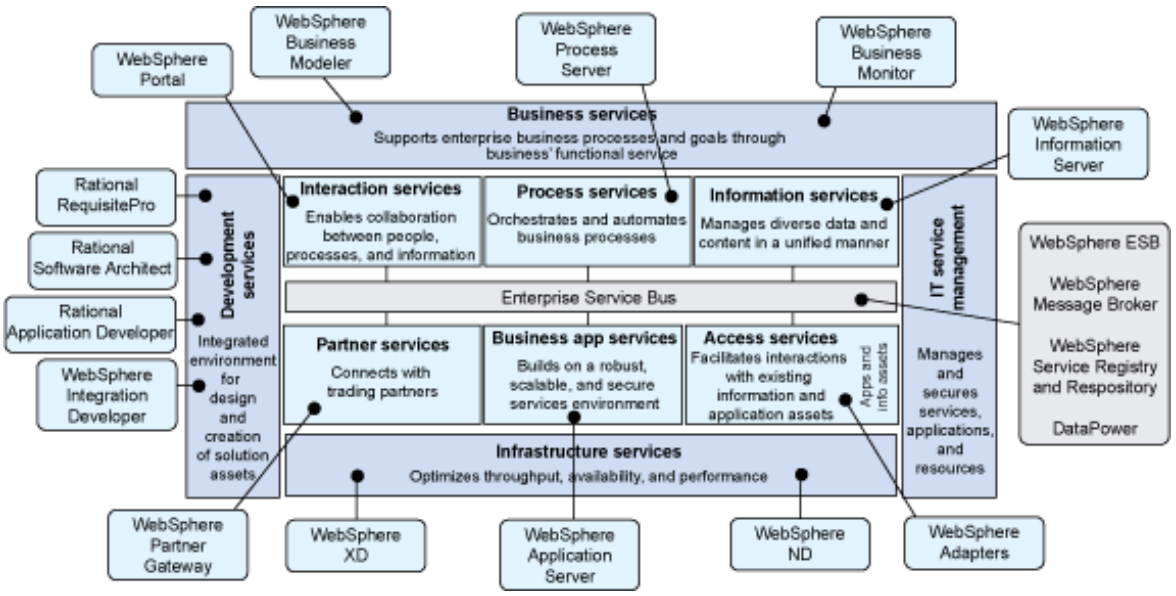

Ilustración 6: SOA

Fuente: (IBM, 2015)

2.6. Características de los patrones de diseño y patrones de arquitectura.

Una manera resumida de las características de estos elementos esenciales para una AS, es que los patrones de diseño sirven específicamente para enfocarnos en el problema de vamos a resolver, de modo que no impacta significativamente en la AS, ya que no nos preocuparemos en la interacción entre todos los patrones entre sí, como un todo. (Victor Hugo Jimenez Torres, 2014)

2.7. Metodologías ágiles de desarrollo

Dentro de los parámetros que conlleva una arquitectura de software, las metodologías de desarrollo son esenciales al momento de realizar desde un prototipo hasta un software a gran escala, por ello un ingeniero en sistemas es de manera relativa un arquitecto de software, ya que posee todos los conocimientos para poder utilizar todas las metodologías existentes, o en su defecto las más utilizadas o conocidas.

Definición de metodologías ágiles de desarrollo.

Son propuestas metodológicas que posibilitan un mejor control en los procesos, estableciendo actividades rigurosas dando mayor seguridad al desarrollo y control del software.

Aunque en la mayoría de los casos las MAD[13] siempre ha demostrado ser efectivas en muchos proyectos, también han generado ciertas eventualidades o problemas directos, ya sea por su incorrecta utilización o la mayoría de veces por la falta de experiencia de los técnicos o ingenieros a cargo de un proyecto.

[13] Metodologías ágiles de desarrollo.

Pero sin lugar a dudas las MAD han revolucionado la forma como desarrollar software, además ha generado un gran debate entre quienes mantienen el uso de las metodologías tradicionales generando resistencia al cambio y escepticismo, porque no las ven como una alternativa a las metodologías tradicionales.

Sin embargo pese al poco convencimiento que les otorgan a las MAD, los usuarios de las metodologías tradicionales han logrado centrarse en una específica, que relativamente según su criterio la muestran como una posible metodología que pudiese convertirse en una alternativa a las que ya se conoce, hablamos claro de la metodología XP[14](Xtreme Programming), siendo, claro; una de las más populares y utilizadas, de entre las metodologías ágiles.

Principios de Agilidad

Una vez que tenemos claro el contexto de una metodología ágil de desarrollo, es tiempo de esquematizar los principios de agilidad, para que una MAD sea llamada así como tal, ya que para que alguna metodología pueda ser "ágil" debe cumplir los siguientes principios:

Rapidez

Es una de las prioridades con mayor grado de importancia, ya que se debe entregar al cliente un software de calidad en el menor tiempo posible.

[14] Xtreme Programming

Cambio

Los nuevos requerimientos siempre deben ser tomados en cuenta, inclusive si el desarrollo del proyecto está ya en la fase final, ya que un proceso ágil, al contrario de lo que se piensa un proceso ágil toma ventaja del requerimiento.

Eficiencia

Usando los recursos que se obtengan a disposición, las fechas de entrega de software deben ser como mínimo 3 semanas y un mes (30 días).

Trabajo en grupo

Aunque ciertos programadores pueden realizar sus tareas desde sus estaciones de trabajo en sus hogares o indistintamente en otro lugar, el principio de agilidad menciona estrictamente que tanto los desarrolladores como como las personas de negocio deben trabajar juntas diariamente, en el mismo lugar durante toda la ejecución del proyecto.

Motivación y buen ambiente de trabajo

Este principio es también uno de los esenciales, ya que sin lugar a dudas, la motivación que el jefe del proyecto ejerza en sus colaboradores se verá reflejado en el resultado deseado, sin dejar a un lado el ambiente de trabajo donde se lleva a cabo el desarrollo del proyecto, generando así individuos motivados y resultados con mayor compromiso.

Transmisión de información

Para que la claridad del contenido del o los requerimientos sea comprendido en su esencia, es necesario la transmisión de información cara a cara, ya que una mala comprensión de algún proceso genera retraso y pérdida de tiempo, afectando a todos los colaboradores del proyecto.

Avance

Una clara forma de demostrar el avance el proyecto es que el software esté funcional desde su etapa media de avance. (Pressman, 2014)

Aunque ciertos MAD en sus proyectos pueden llegar a cumplir con todos estos principios, algunas pueden obviarlos parcialmente, ya que si los ignorasen en su totalidad no serían MAD como tal.

Metodologías ágiles de desarrollo de software

Dentro del contexto de metodologías ágiles de desarrollo (MAD), tenemos muchas que actualmente están generando proyectos mediante su uso, aunque una de las tareas a las que se enfrentan estas nuevas tendencias, es por supuesto, la estandarización, siendo un pequeño problema pero no tan grave al momento de su implementación.

Entre las MAD más utilizadas podemos encontrar las siguientes:

- Adaptive Software Development (ASD)
- Agile Unified Process (AUP)
- Crystal Clear
- Essencial Unified Process (EssUP)
- Feature Driven Development (FDD)
- Kanban
- Open Unified Process (OpenUP)
- Programación Extrema (XP)
- Método de desarrollo de sistemas dinámicos (DSDM)
- Scrum.

(Pressman, 2014)

2.8. La Metodología XP[15]

Selección

Decir por qué se escogió esta metodología de entre el resto de las antes nombradas.

Historia

Por motivo de análisis de una de las MAD más usadas actualmente, es indispensable conocer un poco de la historia y como se sobrellevó esta metodología cuando tuvo su primera aparición en el mundo de la ingeniería de software. (Wells, 2013)

[15] Xtreme Programming

Aunque los primeros principios de XP se remontan a la década de 1980, en un primer bosquejo de la metodología realizado por Kent Beck, debemos destacar que el primer proyecto generado usando XP ya enteramente usándolo con su nombre como tal fue en el año 1996, exactamente el 6 de marzo, desde allí se demostró su efectividad tanto para pequeños como grandes proyectos, su primera carta de presentación fue la satisfacción generada en los clientes que cada vez más optaban por implementar sus sistemas con esta metodología. (Pressman, 2014)

Inclusive XP ya no es la única MAD de su tipo, ya que recientemente el autor de los principios de XP, ha presentado una MAD llamada IXP[16], la cual está dirigida específicamente a grandes empresas.

Valores de XP

Para poder comprender y poner en práctica esta MAD, se debe seguir los siguientes 5 valores de XP, en la primera edición de Extreme Programming Explained de Kent Beck eran sólo 4, el quinto se añadió en la segunda edición del mismo, quedando finalmente definidos los 5 que a continuación se procede a describir.

Comunicación

Es una parte que con certeza sino se la toma en serio, puede conducir al fracaso del proyecto, la comunicación se refiere a las diferentes interacciones que haga el cliente con los programadores o los programadores entre sí, por lo que para un programador mientras más

[16] Industrial Xtreme Programming

simple sea es mucho más fácil entenderlo, inclusive los comentarios no son de gran ayuda en este caso, ya que dejan de ser útiles conforme se va modificando el proyecto, para los comentarios es preferible que únicamente esté comentado la finalidad de clase y la función del método.

En lo referente al talento humano, se utiliza la programación por parejas, donde cada programador comunica a su co-programador toda la información necesaria, finalmente el cliente, quien debe estar en toda la fase del proyecto definiendo características prioritarias y aclarando todas las dudas que se presenten.

Simplicidad

Una forma de simplicidad que se aplica en XP, es la de impedir a los programadores a rediseñar el proyecto, ya que el objetivo principal es crear un diseño fácil que se implemente con facilidad en código, en caso de que se necesite rediseñar el sistema, se lo hará posteriormente.

Retroalimentación o Feedback

Como su nombre, explica, retroalimentación, en este valor, actúa directamente el cliente, ya que como está presente en el desarrollo total del sistema, la información se conoce en tiempo real, lo que genera un rápido desarrollo y resultados rápidos.

Por ejemplo, en ciertas metodologías el desarrollo se toma los requerimientos en la parte inicial al cliente, y se vuelve a entrevistar con él una vez terminado total o parcialmente, y lo que ocurre es que al estar insatisfecho el cliente o por algún malentendido de los desarrolladores, genera meses de trabajo tirados por la borda.

En lo referente al código, las pruebas unitarias ayudan a controlar más rápidamente los fallos debido al cambio frecuente del código. (Pressman, 2014)

Respeto

Una forma de las muchas que hay es el respeto de los programadores hacia ellos mismo y sus compañeros, es decir, no permitirse modificar el código con fines maliciosos, para generar fallas a propósito en las pruebas unitarias.

También el respeto se demuestra en el trabajo diario, ya que el fin de todo el equipo es alcanzar un software de calidad, esto se logra no haciendo de menos a ninguno de los integrantes del grupo de desarrollo.

Coraje y Valentía

Alguna vez escucharemos en ofertas laborales frases como “Trabajo bajo presión”, desde ese momento no damos por hecho que se planee usar XP como MAD, sino que un programador debe estar motivado siempre, un ejemplo sería la revisión de varias veces el código con el fin de buscar una salida más conveniente que la anterior ya generada, lo que comúnmente se llama reconstrucción de código.

La valentía es un semejante de la persistencia, ya que debemos ser valientes para desechar código obsoleto, no importa el esfuerzo que nos haya llevado implementarlo, además como dato adicional, un programador está en la capacidad de permanecer en un problema todo un día completo, pero si es persistente podrá resolverlo con mucha facilidad al día siguiente. (Pressman, 2014)

2.8.1. Fases de XP

Aunque XP sea considerada una MAD posee una cierta estructura con pasos a seguir para poder implementarla de manera adecuada, además es necesario tomar en consideración que esta metodología usa un enfoque orientado a objetos, lo cual engloba un conjunto de reglas y prácticas que se debe tomar en cuenta, a continuación se procederá a nombrar y explicar cada una de ellas.

Planeación

Esta es una de las actividades que podemos decir es una de las más importantes, ya que los involucrados en el proyecto de software a desarrollarse deben prestar mucha atención, entonces los técnicos a cargo del desarrollo deben entender el contexto de negocio así como las características principales de entrada - salida y las funcionalidades que se requieran.

Generalmente en términos técnicos del área de desarrollo se conoce como fase de recopilación de requerimientos los cuales deben ser anotados por escrito en las famosas historias de usuario, que desde este momento las llamaremos "Historias XP" para describirlas en posteriores capítulos. (Pressman, 2014)

Historias XP

Las historias XP son básicamente semejantes a los casos de uso que se utilizan en metodologías de amplia documentación, como RUP[17] por sus siglas en Inglés (Rational Unified Process) deben ser la llave maestra con la cual se empezará el desarrollo, ya que en ellas se encuentra descrito lo que se debe hacer es decir, las características y funcionalidad del software,

[17] Rational Unified Process

esta es escrita por el cliente e indizada donde el mismo cliente le asigna la prioridad, para que luego consecuentemente el equipo de desarrollo la evalúe y asigne su costo en días o semanas desarrollo, como un estimado tiempo para cada historia se puede tomar el valor de 3 semanas, en caso de que el trabajo descrito en la historia XP indizada sea demasiado extenso y sobrepase el costo de 3 semanas, el equipo de desarrollo debe pedir al cliente que la historia con un coste superior se descomponga en historias más pequeñas (Pressman, 2014)

En ciertos casos, principalmente en desarrolladores junior es muy común no saber cuándo realizar estas historias XP, por lo que como un dato importante resalto: “Las historias de Usuario XP pueden ser realizadas en cualquier momento.”

Implementación de una Historia XP

Cuando se trata de proyectos a gran escala y con alta sensibilidad en sus requerimientos y proceso de negocio se debe tener claro que para realizar la evaluación y costes de cada historia indizada debemos conocer en qué orden se deben implementar cada una.

Existen 3 métodos de implementación.

Implementación inmediata

Como su nombre lo expresa son las historias XP que por su bajo costo de tiempo se las ejecuta de manera inmediata.

Programación de actividades

Son aquellas historias que tienen un alto coste, las cuales entrarán a la programación de actividades para posteriormente implementarse de manera inmediata.

Implementación por riesgo

Son aquellas que por su alto riesgo deben entrar a la programación de actividades para ser implementadas primero, una historia XP de alto riesgo no necesariamente puede tener un alto costo.

Para finalizar con las historias de usuario, hay que destacar que a medida que avance el proyecto de software, el cliente podrá realizar un CAB[18] de historias XP, así como de las que ya estén implementadas.

Diseño

Una vez obtenidas las historias de usuario XP en la fase de Planeación ahora es indiscutiblemente necesario la realización del diseño, usando las características y requerimientos obtenidos.

Para ello la Metodología XP usa un término que suelen usar ciertos autores, conocido como MS (Mantenlo Sencillo), por los que al ser una MAD (Metodología Ágil de Desarrollo) busca siempre la manera de presentar el proyecto de forma sencilla para ello se toma en consideración una forma bastante sencilla que ayuda en la programación orientada a objetos, como lo son las tarjetas CRC[19].

Tarjetas CRC

Una manera de saber cuáles son las clases esenciales del sistema, es decir *el meollo del asunto* son las tarjetas CRC, las cuales proporcionan de manera fácil y ordenada la

[18] Crear, Actualizar, Borrar

[19] Clase, Responsabilidad, Colaborador.

identificación y organización ayudando al grupo de técnicos de desarrollo a poner un nombre a la clase, las responsabilidades de la clase, y los colaboradores de dicha clase. (Pressman, 2014)

Prototipos XP

En casos muy particulares ciertas veces el desarrollador se enfrentará a diversas situaciones de las que generalmente no tiene conocimiento, como por ejemplo cuando es necesario realizar un prototipo del prototipo del sistema creado con XP, un caso especial se da cuando una historia XP presenta un problema muy complejo de diseño, lo que se conoce como *solución en punta,* con el único fin de evitar a futuro en el proyecto verdadero un impacto considerable de la "Historia XP Problema"

Codificación

Al contrario de lo que la mayoría piensa, en esta etapa de XP, se pensaría que al leer el nombre *codificación*, pensamos enseguida que se refiere a la construcción y programación del código total que se empleará para la construcción del sistema, pero no es así, en esta etapa los desarrolladores realizan pruebas unitarias a cada una de las historias XP obtenidas, de las cuales se obtendrá errores locales o de componente de la interfaz o las estructuras de datos.

Una manera de realizar la codificación de las PU[20] es mediante la retroalimentación, y actualización constante del código que se genera, así como su pronta puesta a prueba, para ello es necesario que las responsabilidades de cada historia XP para cada PU recaiga en una pareja de colaboradores, ya que como se dice *dos cabezas piensan más que una* y de esta manera a

[20] Pruebas Unitarias

medida que las parejas de desarrolladores entreguen el código, este se integra al del resto de parejas, generalmente para la integración existe un equipo dedicado específicamente para dicho proceso, ya que receptar código y actualizarlo a diario requiere de concentración y sobre todo trabajo en equipo, que era uno de los valores de XP mencionados anteriormente. (Pressman, 2014)

Pruebas

Una vez que se ha completado total o parcialmente las 3 fases anteriores, es momento de obtener los resultados preliminares de las pruebas unitarias de código y de las integraciones realizadas por el equipo de integración.

En esta etapa es indispensable que las pruebas unitarias se estructuren de manera que puedan ser automatizadas para los diferentes cambios que es seguro habrá en el proyecto.

Pruebas de Aceptación de XP

Son derivadas de las historias de usuario hechas por el cliente, y al ser realizadas por este, es el cliente quien se encarga de revisar las características y funcionalidad del sistema, es decir las que el cliente puede observar.

XP Industrial

Como un adicional ante las nuevas tecnologías derivadas de XP, se ha empezado a implementar una no tan nueva derivada de XP, es el uso industrial que se ha procedido a dar a esta tecnología, sus siglas oficiales son IXP[21]

[21] Industrial Xtreme Programming

Desde la misma creación ce XP, esta MAD ha revolucionado el ámbito de la programación de aplicaciones, por ello, en algunos casos se ha tenido la necesidad de optar su uso a algo mucho más extenso que una pequeña aplicación empresarial o una PYME[22], de allí que surge IXP, con un enfoque mucho más amplio, sugiriendo así que una MAD puede ser usada perfectamente para aplicaciones muy complejas, no solo a nivel del sistemas, sino orientado más al desarrollo de una arquitectura de software completa. (Kent, 2011)

A continuación una ilustración de manera general de IXP.

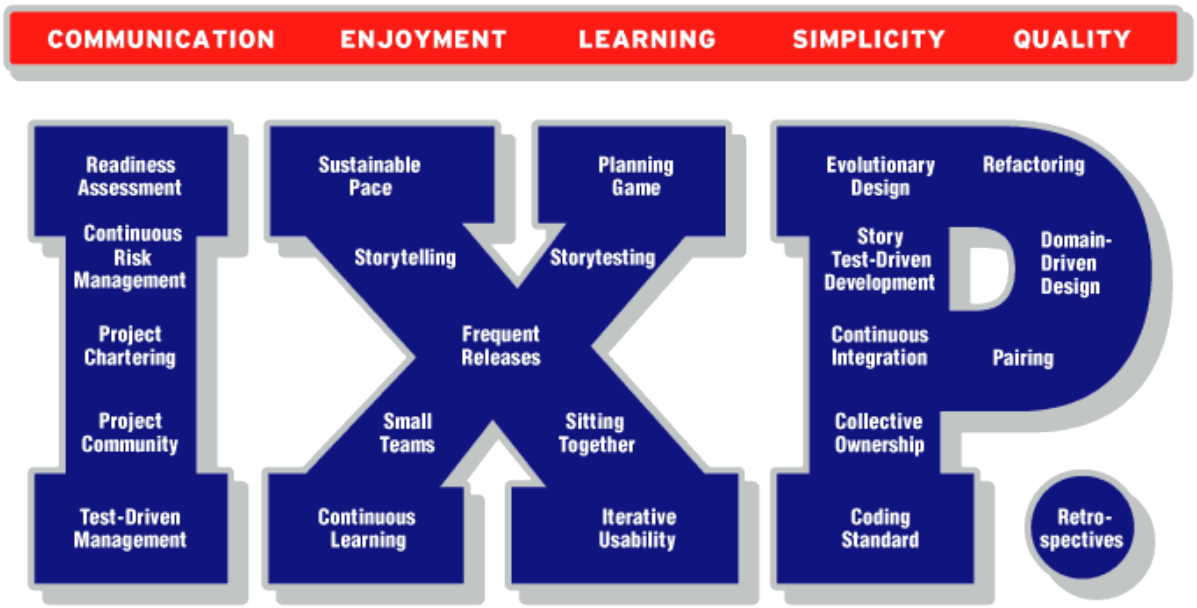

Ilustración 7: XP

Fuente: (industrialxp, 2011)

[22] Pequeñas y medianas empresas

2.9. Metodologías para el desarrollo de arquitecturas de software.

Una AS es esencial para el correcto funcionamiento de la empresa para la cual se ha desarrollado, por ende una AS no podrá en la mayoría de los casos ser la misma para una empresa, como para otra, ya que tienen diferentes necesidades, tal vez tengan metas semejantes pero la forma de alcanzarlas será distinta en todos los casos, o en la mayoría de ellos.

Es por ello que al ser la AS una ciencia relativamente contemporánea, no tiene estructurado un estándar con los procedimientos necesarios para cada ambiente donde se desea implementarla, por ello decir que una MDAS[23] es universal para todos los casos sería un error, ya que cada Arquitecto de Software debe reunir todas las necesidades de su empresa para poder implementarlas.

En lo que sí podría coincidir, en ciertos casos, serían los primeros procedimientos a tomar en cuenta, como se detalla en la ilustración 8.

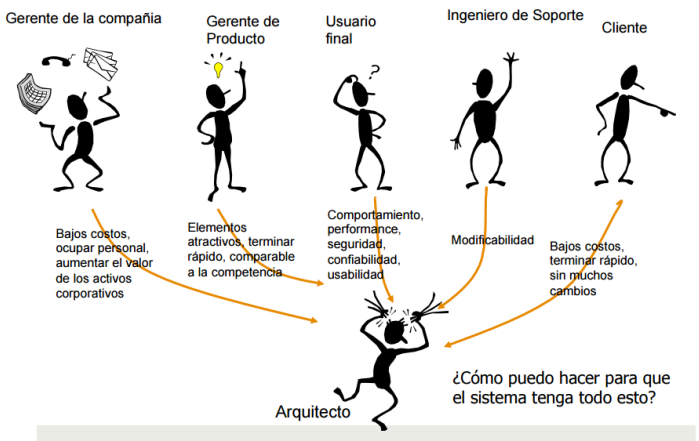

Ilustración 8: Procedimientos

Fuente: (Cali, 2012)

[23] Metodología para el desarrollo de arquitecturas de software

Ahora la integración de todos los requerimientos, tal como se describe en la ***ilustración 15*** , deben ser enfocados en un modelo entendible realizado por y para el arquitecto de software, por lo que es recomendable usar un Framework de desarrollo de una AS.

2.9.1. Framework de Orientación del desarrollo de una AS.

Para el desarrollo de una AS de manera general se debe tomar en cuenta los siguientes puntos:

- Motivación
- Revisión del Framework
- Arquitecturas y disciplinas de la Arquitectura (¿Qué es lo que se va a hacer?)
- Perspectiva de la AS (¿Dónde?)
- Requerimientos de la AS (¿Por qué?)
- Referencia Arquitectónica (¿Con qué?)
- Organización, grupo de trabajo, individuos. (¿Quienes?)
- Metodologías a usar (¿Cómo?)

Fuente: (Duggan, 2012)

El presente framework se lo ha realizado en base a investigaciones de varias AS conformadas y probadas, las cuales muestran un grado de complejidad aceptable como para un tema de estudio en el futuro.

Al no contar con una metodología estandarizada, se hace necesario para este tema de investigación la recopilación de varias abstracciones ya implementadas que han sido, estables y han dado buenos resultados.

Así mismo existe un framework adicional para la implementación correcta de una AS.

2.9.2. The Open Group Architecture Framework [24](TOGAF)

Una verdadera metodología probada, desarrollada e implementada es el TOGAF, el cual fue creado por el Open Group en 2010, fue basado en el framework de arquitectura técnica para el manejo de información (TAFIM)[25] por sus siglas en Inglés, el mismo que fue puesto en producción por el departamento de defensa de los Estados Unidos de América, aunque estuvo disponible desde el año 1995, en la actualidad el TOGAF ha sido objeto de múltiples modificaciones y actualizaciones, actualmente se encuentra disponible en la versión 9.

Los componentes del TOGAF son esencialmente los siguientes:

- Método de desarrollo de Arquitectura.(ADM)[26]
- Un Framework para definir el contenido estructural de la Arquitectura.(ACF)[27]

Además de numerosas herramientas, modelos referenciales, taxonomías, buenas prácticas, principios, guías y tecnologías.

Al proveer TOGAF diferentes herramientas para un desarrollo de alto rendimiento empresarial, se hacen necesarias las vistas de desarrollo empresarial, las cuales definen las TI[28] del sistema. (Vogel, Arnold, & Chughtai, 2011)

---

[24] The Open Group Architecture Framework

[25] Technical Architecture Framework for Information Management

[26] Architecture Development Method

[27] Architecture Content Framework

[28] Tecnologías de Información

Tal como se muestra en la ilustración los diferentes componentes de este Framework, tienen como común denominador la complejidad necesaria para una AS Enterprise.

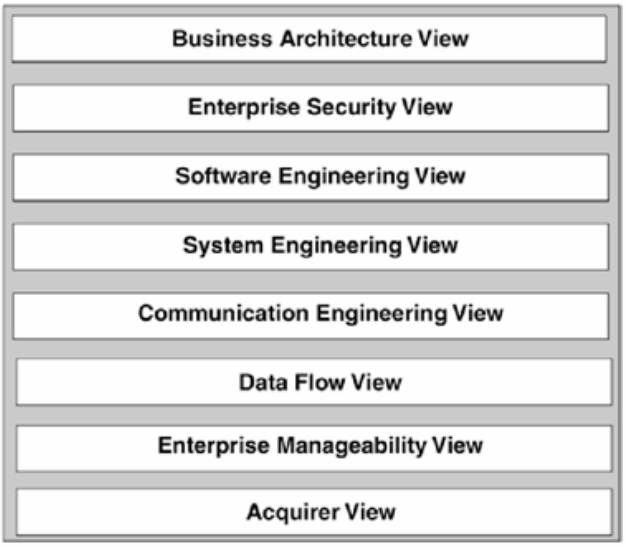

Ilustración 9: Componentes de TOGAF

Fuente: (Vogel, Arnold, & Chughtai, 2011)

2.10 Planificador de Recursos Empresariales ERP

Al momento de escuchar el término "planificador", se nos viene a la mente talvez una especie de framework que permita resolver las diferentes actividades del ciclo de negocio de una AS.

Pero no conlleva mucho su nombre con la verdadera función en la vida real, por lo que cada PRE-ERP[29] tiene tanto su metodología, funcionalidad y sus propias tecnologías específicas o ajustadas para cada empresa.

Componentes de una PRE-ERP

Básicamente se componen de los siguientes ítems.

- La funcionalidad para la cual van a ser utilizados.
- El soporte en cuanto a el idioma, si será implementado en un grupo empresarial o en una sola empresa.
- Tecnologías a usar, como por ejemplo, sus bases de datos, lenguaje de programación, y el diccionario de datos.
- Framework base, es decir si la plataforma tecnológica estará abierta a nuevas actualizaciones, plugins, etc, siempre y cuando el rendimiento mejore sin aumentar en gran escala su tamaño. (homenet, 2015)

[29] Planificador de Recursos Empresariales ERP

Ilustración 10: PRE - ERP

Fuente: (tinetworksolutions, 2012)

Entonces como podemos apreciar en la ilustración, definimos que un PRE-ERP es una software que depende del uso que se lo vaya a emplear en cada empresa, por lo que no debe ser precisamente desarrollado, más bien debe ser ajustado a las tecnologías con las que dispone una empresa para su implementación, independientemente de cuantos PRE-ERP existan es decisión de un Arquitecto de Software viabilizar cuál de todos los existentes tienen una correlación de su empresa con el PRE-ERP.

CAPITULO III

Mapa de Procesos de la Empresa Pública La Uemprende EP.

- Estado de situación inicial de la empresa.
- Investigación de procesos que posee la empresa pública La Uemprende E.P.
- Caracterización de la información de los procesos.
- Realización del mapa de procesos

3.1. Estado de Situación Inicial de la Empresa.

Una vez revisada la teoría necesaria para poder comprender el desarrollo de una AS, empezaremos por definir el ESI[30] de la Uemprende EP.

Dentro del contexto de un ESI debemos enfocarnos en la parte técnica de la empresa, como que sistemas posee, procesos, macro procesos, subprocesos y la prioridad que cada uno tiene dentro de nuestra empresa, siguiendo el modelo de negocio.

La Empresa Pública La Uemprende EP, consta de 3 sistemas que actualmente son de utilidad para la empresa, los tres sistemas están separados entre sí, sin ningún enlace, ya que los 3 están desarrollados con diferentes tecnologías.

- Sistema de Contabilidad
- Sistema de Inscripciones Online
- Plataforma del Curso de Formación Continua.

Además de contar con su sitio web v 1.0, y su servidor compartido fuera de su institución.

En el ámbito tecnológico, cuenta con una considerable cantidad de computadores y equipos informáticos necesarios para el funcionamiento de los procesos administrativos, estos equipos no poseen una conexión de red adecuada, es decir la topología de red no se ajusta a las necesidades de las instalaciones.

No posee de un plan informático, extremadamente necesario para ejecutar sus diferentes actividades durante el año, por lo que la improvisación es una diaria tarea. (Directorio La Uemprende EP, 2014)

---

[30] Estado de Situación Inicial

Por todo lo mencionado anteriormente al no poseer los requisitos necesarios para un buen funcionamiento, es imposible que cuente con una Arquitectura de Software, por lo que se hace necesario su pronto diseño e implementación.

Equipos Tecnológicos

Dentro de la Uemprende EP, los equipos tecnológicos son reducidos por lo que se cuenta con los siguientes ítems:

- Servidor Web
- Computadores de Escritorio, portátiles, router inalámbrico
- Laboratorio de Computación con una capacidad para 21 personas.
- 5 proyectores.

- .

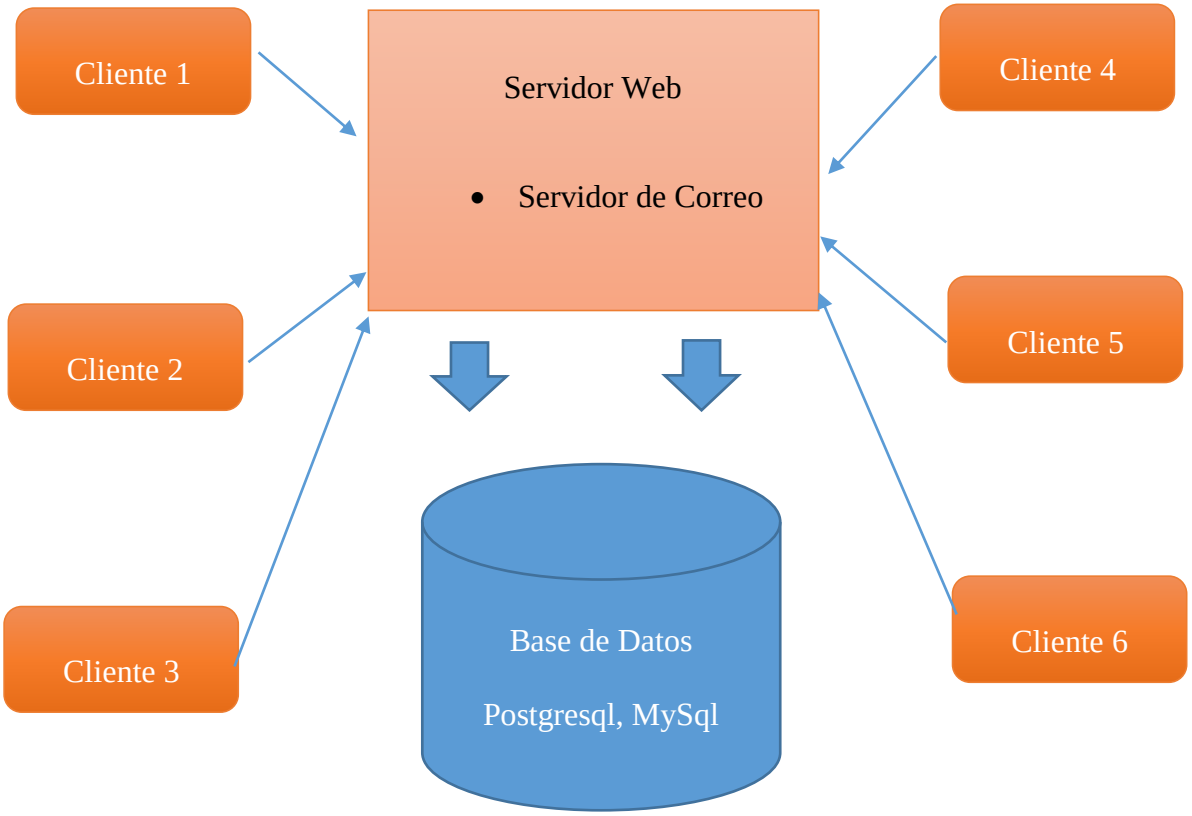

Ilustración 11: Diagrama de Arquitectura Actual Uemprende EP

Fuente: El Autor

3.2. Procesos de la Empresa Pública La Uemprende EP.

Los procesos ejecutados por la Uemprende, los cuales necesitan ser automatizados son:

- Gestión de usuarios en Inscripciones Online, capacitaciones de cursos.
- Gestión de Órdenes de salida y control de vehículos, kilometraje, ruta, combustible.
- Gestión contable, balances, inventario.
- Gestión Financiera, solicitudes de pago, viáticos, subsistencias.
- Gestión de Talento Humano, cargos, custodios de bienes, responsables de proyectos. (Directorio La Uemprende EP, 2014)

Proyectos a ejecutarse a largo plazo.

- ❖ Implementación del sistema informático integrado Uemprende EP que constará de los siguientes módulos:
- ❖ Módulo Contable.
- ❖ Módulo Financiero
- ❖ Módulo Recursos Humanos
- ❖ **Módulo de Capacitaciones.**
- ❖ Módulo de Control de Bienes – Inventario
- ❖ Módulo de Control de vehículos. (Uemprende, 2013)

Proyectos a ejecutarse en corto plazo.

- ❖ Arquitectura de Software de toda la empresa.
- ❖ Cableado Estructurado de su edificio.
- ❖ Adquisición y Licenciamiento de Software.
- ❖ Capacitación del uso de la infraestructura tecnológica.
- ❖ Implementación del prototipo del módulo de inscripciones.

3.3. Mapa de Procesos.

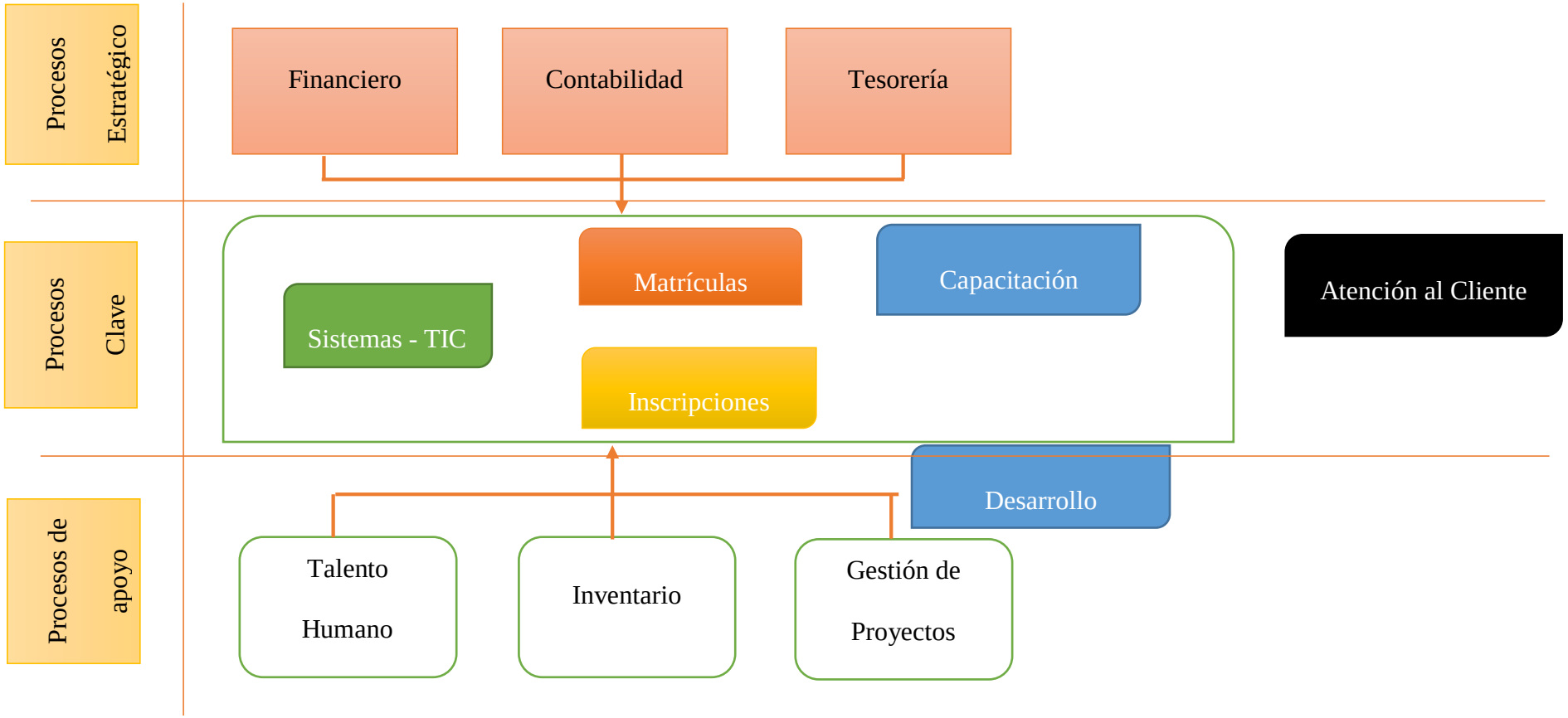

Ilustración 12: Mapa de procesos

Fuente: El Autor

3.4. Caracterización de Principales Procesos.

Tabla 1: Caracterización de procesos

Fuente: (Directorio La Uemprende EP, 2014)

	Entradas	Proceso	Documentos y registros	Salida	Usuario Final
	• Normativa Vigente • Realizar presupuesto. • Reporte de novedades	• Análisis total de presupuesto según declaración de sueldos. • Revisión de cláusulas pautadas entre	• Manual de funciones. • Evaluación de puestos de trabajo (Matrices mensuales).	• Estructura de salarios. • Distribución de sueldos a empleados. • Liquidación de la nómina. • Pago de la nómina.	Empleados internos de la empresa, clientes, empresas públicas o privadas con las cuales se haya contratado servicios o bienes.

Financiero	• Modificaciones contractuales y legales. • Recibos de pago de consolidado de nómina. • Recibe archiva, copia. • Emisión de facturas y retenciones.	empresa – empleado. • Análisis y verificación de reporte y tarifas correspondientes. • Liquidación de prestaciones sociales y pagos a terceros. • Consignación respectiva en el banco.	• Tablas de retención en la fuente. • Comprobante de pago. • Planilla de nómina.	• Entrega de comprobantes de pago. • Total deducido a consignar.	

Capacitación	• Normativa vigente. • Solicitud de convenios. • Propuestas y proyectos. • Información de usuarios. • Bases de datos de personal (Facilitadores – Estudiantes) • Inscripciones. • Matrículas. • Notas por curso.	• Planeación de personal Académico y administrativo. • Planeación de capacitaciones. • Registro y control de usuarios. • Elaboración de certificados. • Medición de la satisfacción del usuario. • Reportes de notas.	• Manual de funciones. • Evaluación de ciclos académicos por curso. • Reporte de novedades, horas de recuperación, horas extras, permisos. • Estructura de Cursos. • Contratos.	• Estructura de ciclos académicos. • Distribución de aulas, docentes. • Entrega de certificados. • Consolidación de notas por materia. • Informe • Cierre de circo académico	• Comunidad Universitaria. • Instituciones públicas o privadas. • Comunidad Externa. • Población de la zona 1 del Ecuador.

	• Nro. Aprobados. • Nro. Reprobados.	• Consolidación de base de datos por ciclo académico. • Diccionario de datos. • Cierre de ciclo de capacitación.	• Recibe, archiva copia.		

Talento Humano	• Políticas y marco de referencia para la administración de talento humano. • Modificaciones en la estructura jerárquica de cargos. • Valoración de cargos. • Normativa vigente de trabajo, código de trabajo.	• Administración de personal. • Control de asistencia. • Permisos por enfermedad. • Permisos por situaciones ajenas a la empresa. • Uso del sistema de gestión documental Quipux	• Manual de funciones. • Trámites administrativos. • Coordinación administrativa. • Autorizaciones. • Informes y asesorías especializadas. • Quejas y reclamos. • Acciones de mejora. • Planes de mejoramiento	• Proceso en mejoramiento continuo. • Asesorías. • Estructura organizacional con valor de cargo. • Informes de aplicación de régimen disciplinario. • Reporte de novedades, incapacidades, vacaciones.	• Empleados internos y externos de la empresa.

	• Sistema de servidores públicos. • Apoyo operativo en el proceso de selección y vinculación desde la solicitud de cubrimiento de vacantes. • Informe de selección.		• Planes de manejo de riesgos en el trabajo (Seguridad Laboral)	• Informe de seguimiento a procesos de selección y contratación. • Programa de inducción.	

CAPITULO IV

Diseño de la Arquitectura de Software para la Empresa Pública La Uemprende EP.

- Ciclo de negocio de arquitectura de software ABC
- Plan de desarrollo informático La Uemprende E.P.
- Estudio de la estructura orgánica funcional.
- Análisis de macro procesos y Subsistemas de La Empresa Pública La Uemprende E.P.
- Arquitectura de despliegue de aplicaciones.
- Diagrama de Macro Procesos
- Diseño de Capas de la Arquitectura de Software.
- Implementación

4. Ciclo de negocio de arquitectura de software ABC

Un tema específico es que el ciclo de negocio de una AS no puede reducirse únicamente a aspectos de manera general de los procesos que se manejan en una empresa, ya que para ello se debe analizar detenidamente cada uno de ellos antes de realizar cualquier implementación, por ende se deben cumplir ciertos pasos para el análisis, los cuales se detallan a continuación.

1. Plan de desarrollo Informático de la empresa.
2. Estudiar la estructura orgánica funcional.
3. Analizar los procesos y subsistemas con los que se cuenta.
4. Mostrar una arquitectura que despliegue las aplicaciones usadas.
5. Finalmente la Implementación.

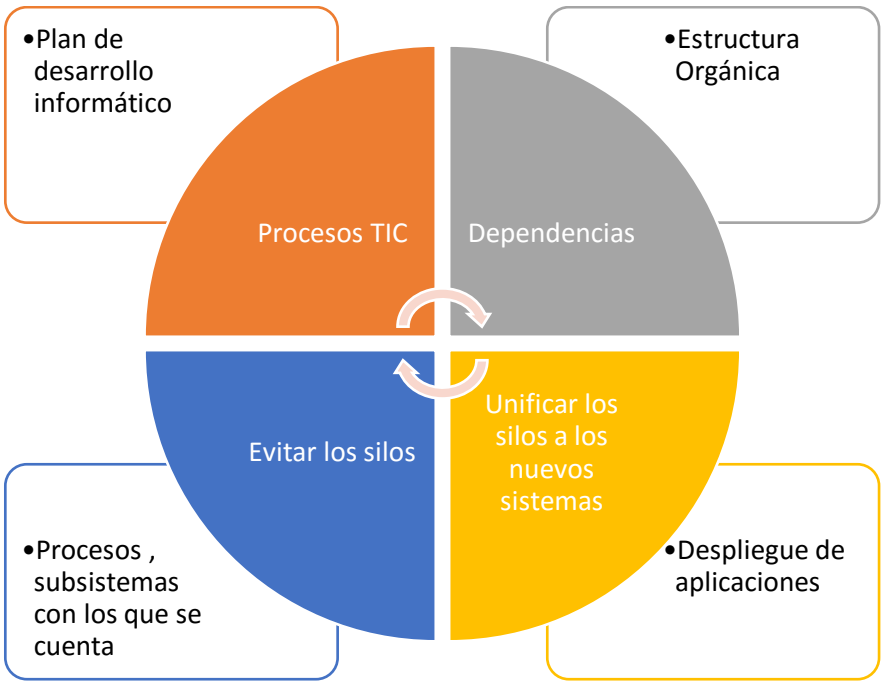

Ilustración 13: Ciclo de negocio AS

Fuente: El Autor

4.2 Plan de desarrollo informático La Uemprende E.P.

Un plan de desarrollo informático comprende la cobertura que se desea obtener de los procesos en el área de TIC, por lo cual debe ser elaborado al inicio del año, y con una duración no menor a los 2 años, cumpliendo las reglas específicas de la empresa, ajustándose en el transcurso del tiempo a las necesidades de cada una de las dependencias, en las cuales se requiere su intervención, que hay que destacar que son todas hoy en día.

Por ello es que la política de la Uemprende EP, en su Plan estratégico señala claramente que uno de los proyectos es el "Implementación del sistema de gestión documental QUIPUX y diseño y utilización de formatos estandarizados para optimizar el uso del papel", (Directorio Uemprende EP, 2014)

4.3 Estudio de la Estructura Orgánico Funcional

Una empresa en crecimiento necesita regirse a cierta estructura organizacional para alcanzar los fines de equipo y cumplir con la misión y visión generadas en su plan estratégico, atacando los puntos primordiales de su análisis FODA así como los procesos esenciales en la lógica de negocio. Por ello una estructura jerárquica analizada y debidamente aprobada por el directorio de la Uemprende EP es el siguiente:

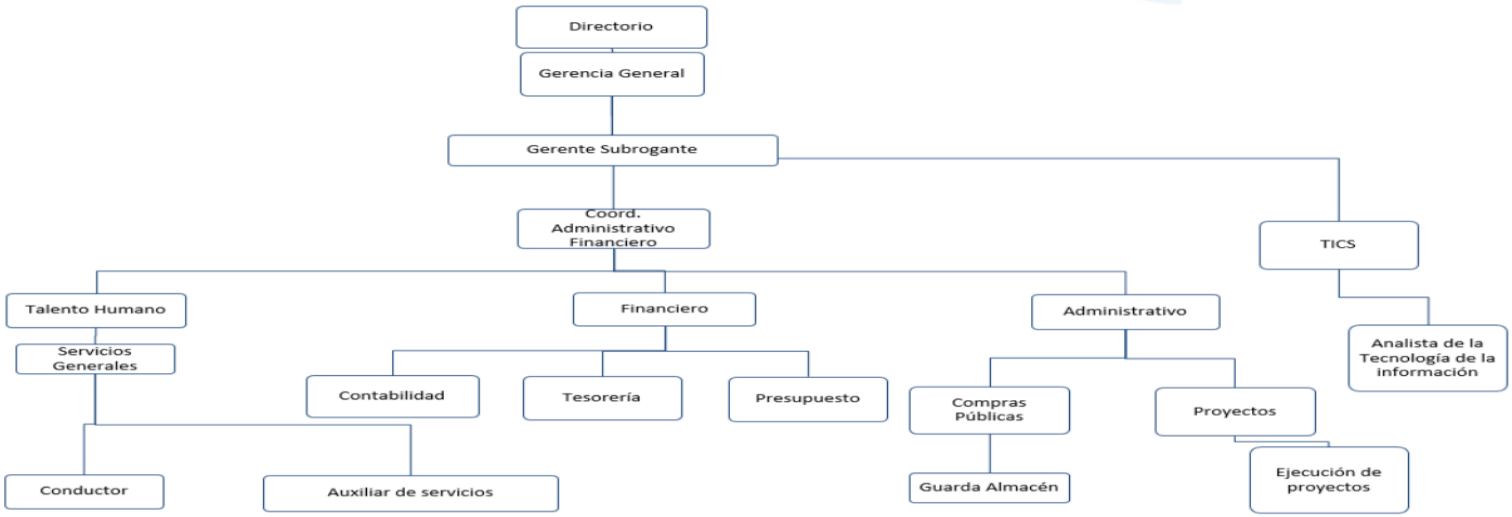

Ilustración 14: Estructura Organizacional

Fuente: (Uemprende, 2013)

4.4 Análisis de Macro Procesos y Subsistemas de la Uemprende EP

Tabla 2: Macro Procesos

Fuente: El Autor

MACROPROCESO	PROCESO	SUBPROCESO	PRIORIDAD	PROCESO ACADÉMICO	GESTIÓN DE FACILITADORES	PLANIFICACIÓN ESTRATÉGICA	TALENTO HUMANO TRANSPORTES	CONTROL DE ASISTENCIA	NÓMINA Y ROLES DE PAGO	GESTIÓN DE PROCESOS INSTITUCIONALES	TRÁMITES INSTITUCIONALES	PRESUPUESTO	ADQUISICIONES	RECAUDACIÓN	ACTIVOS FIJOS	INVENTARIOS	CLIENTES	PROVEEDORES	TESORERÍA Y BANCOS	ANEXOS SRI	CONTABILIDAD ADMINISTRATIVA	CUENTAS POR COBRAR	SEGURIDADES INFORMÁTICAS	AUDITORÍA DE BASES DE DATOS
DIRECCIÓN Y GOBIERNO	ANÁLISIS DE ENTORNO		3			x				x	x										x			
	DIRECCIÓN EN GOBIERNO		3							x	x										x			
	GESTIÓN DE LA COMUNICACIÓN		3			x				x	x				x						x			
	GESTIÓN DEL VALOR		3			x					x													
	GESTIÓN DEL RIESGO		3			x					x													
	GESTIÓN DE LA MEJORA Y EL CAMBIO		3			x				x	x				x		x				x		x	
GENERACIÓN DE VALOR	INVESTIGACIÓN	GESTIÓN DE DOCENTES INVESTIGADORES	3		x	x																		
		GESTIÓN DE CONVOCATORIAS	3		x																			
		GESTIÓN DE EVALUADORES	3		x			x																
		GESTIÓN DE PROYECTOS	3		x					x														
	DOCENCIA	PROCESO ACADÉMICO	1	x		x																		
		GENERACIÓN DE NUEVOS CONVENIOS	2	x																				
		DIFUSION DE ACTIVIDADES DE VINCULACIÓN	2	x	x																			
ASESORAMIENTO Y APOYO	AUDITORIA INTERNA		1		x			x	x	x		x				x		x		x	x		x	
	ASESORIA JURIDICA		1		x			x	x							x				x				
	SECRETARIA GENERAL		2		x						x													
	PLANIFICACION INSTITUCIONAL	PLANIFICACIÓN ESTRATÉGICA Y OPERATIVA	2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	INFORMACIÓN Y COMUNICACION INSTITUCIONAL		1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	ADMINISTRACIÓN DEL CAPITAL HUMANO	ADMINISTRACION DE PERSONAL ACADEMICO	2		x			x	x											x	x			
	ADMINISTRACION FINANCIERA	GESTIÓN PRESUPUESTARIA	2			x	x	x	x	x	x	x	x	x				x	x	x	x	x		
		GESTION ADQUISICIONES	2							x	x	x	x		x			x				x		
		GESTION Y CONTROL DE ACTIVOS FIJOS	2								x		x		x	x							x	
	ADMINISTRACIÓN Y CONTROL DE BIENES		2				x						x		x	x								
	ADMINISTRACIÓN DE PROYECTOS																							
	GOBERNANZA DE LAS TICS	CONTROL DE DOCUMENTOS DE TIC	1																					
		PLANEACIÓN ESTRATEGICA DE TIC	1			x				x		x												
		GESTIÓN DE PROYECTOS TECNOLÓGICOS	1								x		x											
		DESARROLLO DE SOFTWARE	1	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
		MANTENIMIENTO DE SOFTWARE DE DESARROLLADO	1	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
		GESTIÓN DE ACCESO A APLICACIONES DE SOFTWARE E INTERNET	1		x								x				x						x	
		RESPALDO Y RESTAURACION DE DATOS ELECTRÓNICOS	1		x																		x	x
		ATENCIÓN AL USUARIO	1		x							x		x			x			x				
		APLICACIÓN DE LA RED DE VOZ Y DATOS	1																				x	
		ADMINISTRACIÓN DEL RENDIMIENTO Y FALLAS DE RED	1																				x	
		ADMINISTRACIÓN DE SERVIDORES	1																				x	
		GESTION DE LICENCIAS DE SOFTWARE ADQUIRIDO	1								x		x										x	
		CONTROL DE BIENES INFORMÁTICOS	1												x	x								
		FORMULACIÓN Y MONITOREO DEL PLAN DE SEGURIDAD Y CONTINGENCIA	1																				x	x
		CAPACITACIÓN EN TECNOLOGIA DE LA INFORMACIÓN Y COMUNICACIÓN	1		x						x	x		x									x	
	CONTROL DE GESTIÓN				x	x				x	x		x	x	x				x					
	CONTROL INTERNO		2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

4.5 Arquitectura de despliegue de aplicaciones

Tabla 3: Despliegue de aplicaciones
Fuente: El Autor

	SISTEMA	PLATAFORMA	ESTADO	BDD
GESTIÓN ACADÉMICA	PROCESO ACADÉMICO	PHP 5.X	ESTUDIO	POSTGRESQL
	GESTIÓN DE FACILITADORES	PHP 5.X	ESTUDIO	POSTGRESQL
	TALENTO HUMANO TRANSPORTES	PHP 5.X	ESTUDIO	POSTGRESQL
	CONTROL DE ASISTENCIA	PHP 5.X	ESTUDIO	POSTGRESQL
	NÓMINA Y ROLES DE PAGO	PHP 5.X	ESTUDIO	POSTGRESQL
GESTIÓN ADMINISTRATIVA	CONVENIOS	PHP 5.X	ESTUDIO	POSTGRESQL
	GESTIÓN DE PROCESOS INSTITUCIONALES	PHP 5.X	ESTUDIO	POSTGRESQL
	TRÁMITES INSTITUCIONALES	PHP 5.X	ESTUDIO	POSTGRESQL
GESTIÓN FINANCIERA	PRESUPUESTO	PHP 5.X	ESTUDIO	POSTGRESQL
	ADQUISICIONES	PHP 5.X	ESTUDIO	POSTGRESQL
	RECAUDACIÓN	PHP 5.X	ESTUDIO	POSTGRESQL
	ACTIVOS FIJOS	PHP 5.X	ESTUDIO	POSTGRESQL
	INVENTARIOS	PHP 5.X	ESTUDIO	POSTGRESQL
	CLIENTES	PHP 5.X	ESTUDIO	POSTGRESQL
	PROVEEDORES	PHP 5.X	ESTUDIO	POSTGRESQL
	TESORERÍA Y BANCOS	PHP 5.X	ESTUDIO	POSTGRESQL
	ANEXOS SRI	PHP 5.X	ESTUDIO	POSTGRESQL
	CONTABILIDAD ADMINISTRATIVA	PHP 5.X	ESTUDIO	POSTGRESQL
	CUENTAS POR COBRAR	PHP 5.X	ESTUDIO	POSTGRESQL
	COSTEO BASADO EN ACTIVIDADES	PHP 5.X	ESTUDIO	POSTGRESQL
	GESTIÓN BASADA EN ACTIVIDADES	PHP 5.X	ESTUDIO	POSTGRESQL
GESTIÓN DE SEGURIDAD	SEGURIDADES INFORMÁTICAS	PHP 5.X	ESTUDIO	POSTGRESQL
	AUDITORÍA DE BASES DE DATOS	PHP 5.X	ESTUDIO	POSTGRESQL
PORTAFOLIOS	PORTAFOLIO ESTUDIANTIL	PHP 5.X	ESTUDIO	POSTGRESQL
	PORTAFOPIO FACILITADORES	PHP 5.X	ESTUDIO	POSTGRESQL
	PORTAFOLIO ADMINISTRATIVOS	PHP 5.X	ESTUDIO	POSTGRESQL

4.6. Metodología de Capas de la Arquitectura de Software.

La presente metodología de AS, es específicamente orientada a ser una AS Empresarial por lo que se ha buscado un gestor de base de datos que sea opensource, al ser una empresa pública, y segundo que cumpla con características extremadamente altas en lo referente a seguridades, replicaciones, backups, etc, por nombrar algunos.

La Empresa Pública La Uemprende EP, cuenta con un servidor web donde posee alojados 2 gestores de bases de datos, MySql y PostgreSQL, cada uno de estos gestores trabaja indistintamente en el ambiente de producción con varios registros y peticiones realizadas desde los pequeños sub sistemas con los que actualmente cuenta.

4.6.1. Capa de acceso de datos

Como se define en el capítulo 1, en el alcance, se usará un framework de desarrollo para el sistema – prototipo, siendo uno de los más ideales para aplicaciones robustas y empresariales, Yii2 Framework, el cual mediante la API que posee, será el manejador de los accesos a la base de datos Empresarial, de manera resumida sería el que traduzca la lógica de los objetos a la lógica relacional.

En definitiva gracias a la estructura que posee con el Frontend y el Backend, que es un componente una de las formas de controlar el acceso al sistema, en esta capa podemos con una sola línea de configuración pasar de MySQL a PostgreSQL u Oracle y la aplicación seguirá funcionando correctamente. (Fabien Potencier, 2011)

Además una forma eficiente de evitar el acceso a los datos directamente desde la aplicación es con el uso de las diferentes capas que posee Yii 2, creando una fuerte aplicación MVC[31].

[31] Modelo Vista Controlador

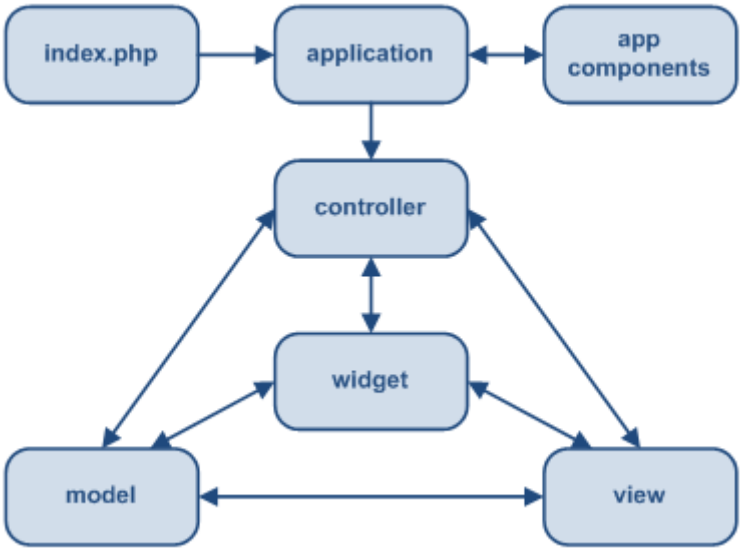

Ilustración 15: Modelo Vista Controlador

Fuente: (Yii Framework, 2015)

4.6.2 Capa de Autentificación y acceso.

La autentificación en sí, trata de verificar si un usuario es quien dice ser, generalmente implica en un nombre de usuario y contraseña, aunque puede existir otros métodos como tarjetas inteligentes, huellas dactilares, entre otras.

Entonces la capa de acceso y autentificación es la encargada de, una vez identificado al usuario, es decir (autenticado) permitirle manipular recursos específicos del sistema.

Ahora Yii tiene un Build de autenticación/autorización (auth) framework que es muy sencillo y fácil de usar y puede ser personalizado según las necesidades del programador del sistema.

La parte central del Auth Framework es que es una aplicación pre declarada del componente de usuario, que posteriormente es implementado en el IWeb User de Yii2, además podemos realizar el monitoreo de usuario, es decir saber si esta un cierto usuario conectado o no, esto se lo logra a través de CWeb. (Yii Framework, 2015)

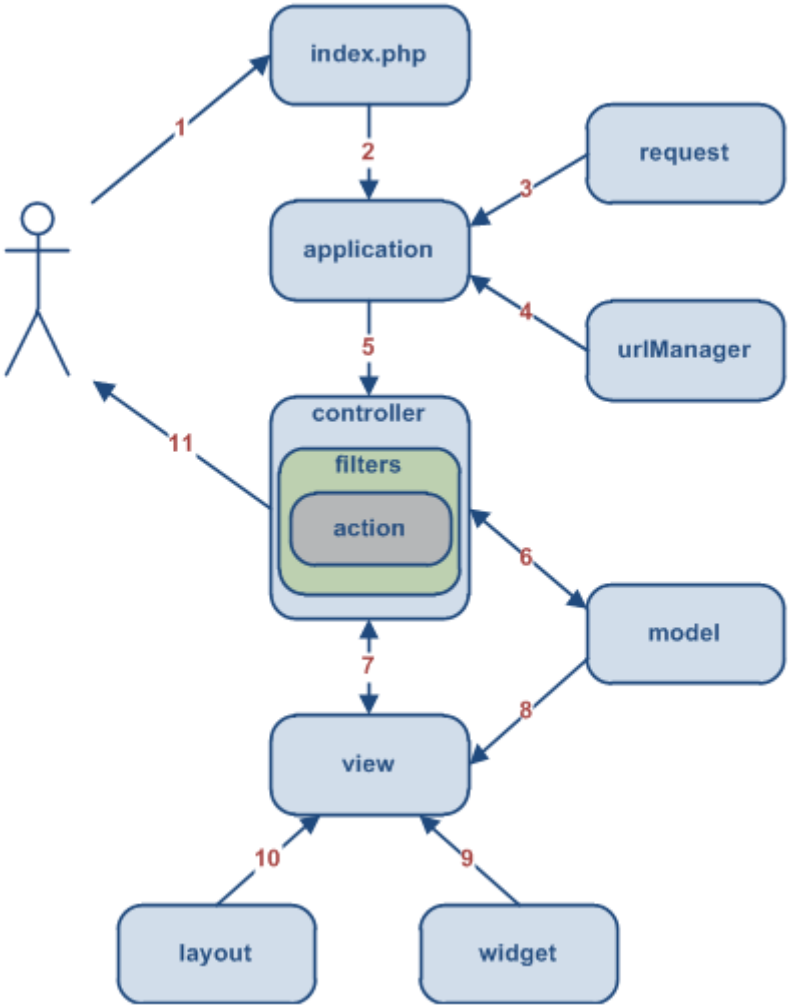

Ilustración 16: Autentificación

Fuente: (Yii Framework, 2015)

4.6.3. Capa de auditoría y control

En síntesis la auditoría se refiere a un log de los eventos que se generan dentro del sistema, dejando pistas de cada transacción que se ejecuta.

Además de las pistas que se generan, esta capa sirve para que el programador sepa cuáles son los puntos donde se generan conflictos o problemas.

Yii2 en definitiva es muy versátil ya que para esta capa nos ofrece una extensión llamada **yii-audit-module,** ayudándonos en la gestión de errores de ejecución en el sistema.

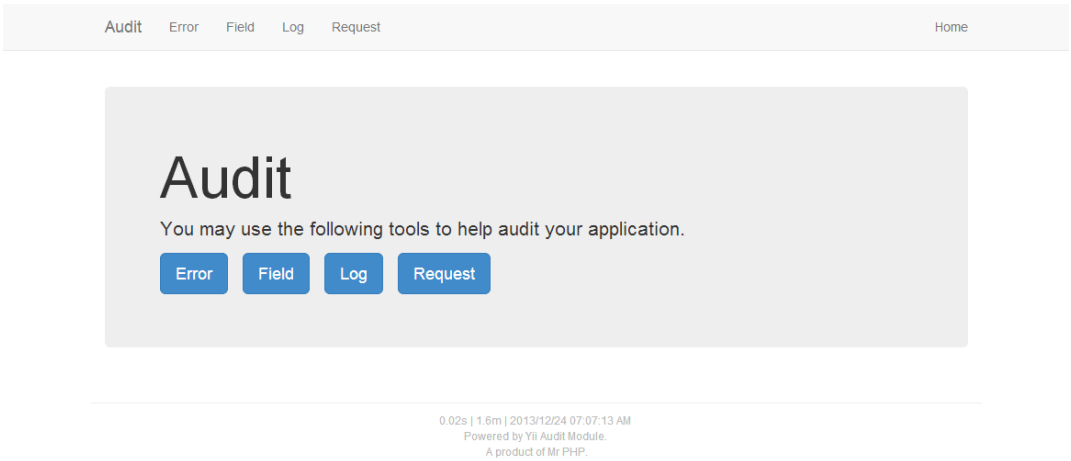

Ilustración 17: Auditoría Yii2

Fuente: (Yii2, 2014)

4.6.4. Capa de lógica de negocio.

Es fundamental a la hora de la implementación de clases y métodos que el programador debe mantener, la forma ordenada y con la correcta nomenclatura garantiza que en modificaciones posteriores o migraciones se no exista inconvenientes generados por código mal estructurado.

Por ello en la capa de modelo, yii2 al ser orientado a objetos en su totalidad, y al representar un modelo a una entidad, generalmente se mapea a una tabla de nuestra base de datos.

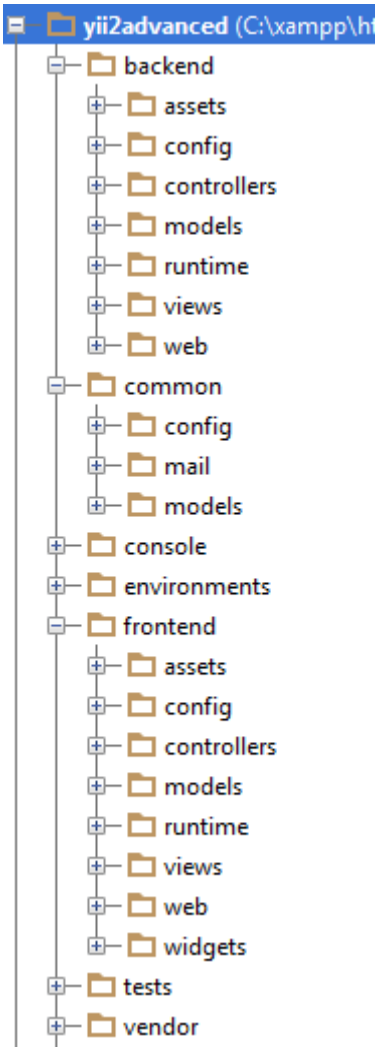

Ilustración 18: Lógica de negocio

Fuente: (Yii Framework, 2015)

4.6.4. Capa de presentación

Es una de las partes más importantes ya que es la que va interactuar directamente con el usuario final, debe tener una usabilidad extremadamente bien diseñada, ya que de ello depende la adaptabilidad del usuario al sistema.

En este punto Yii2 es bastante funcional ya que la presentación se la realiza usando el layout que ofrece Yii2, donde contiene la porción de header y footer y embeber.

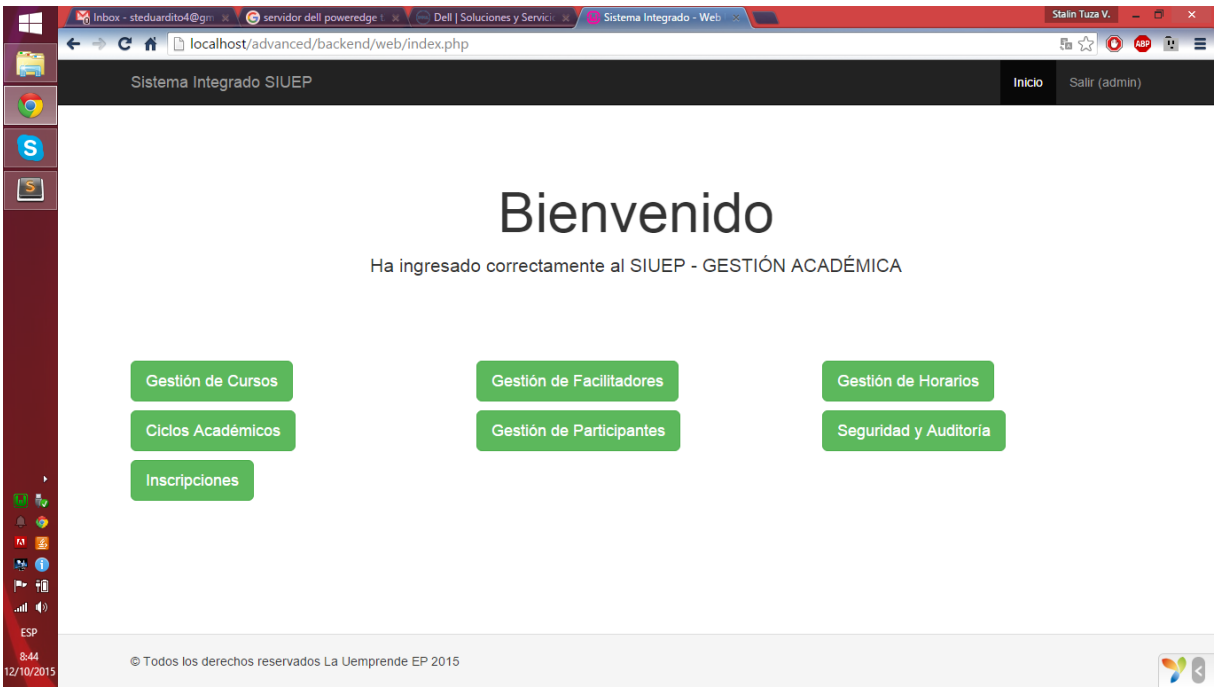

Ilustración 19: Menú Inicio

Fuente: El Autor

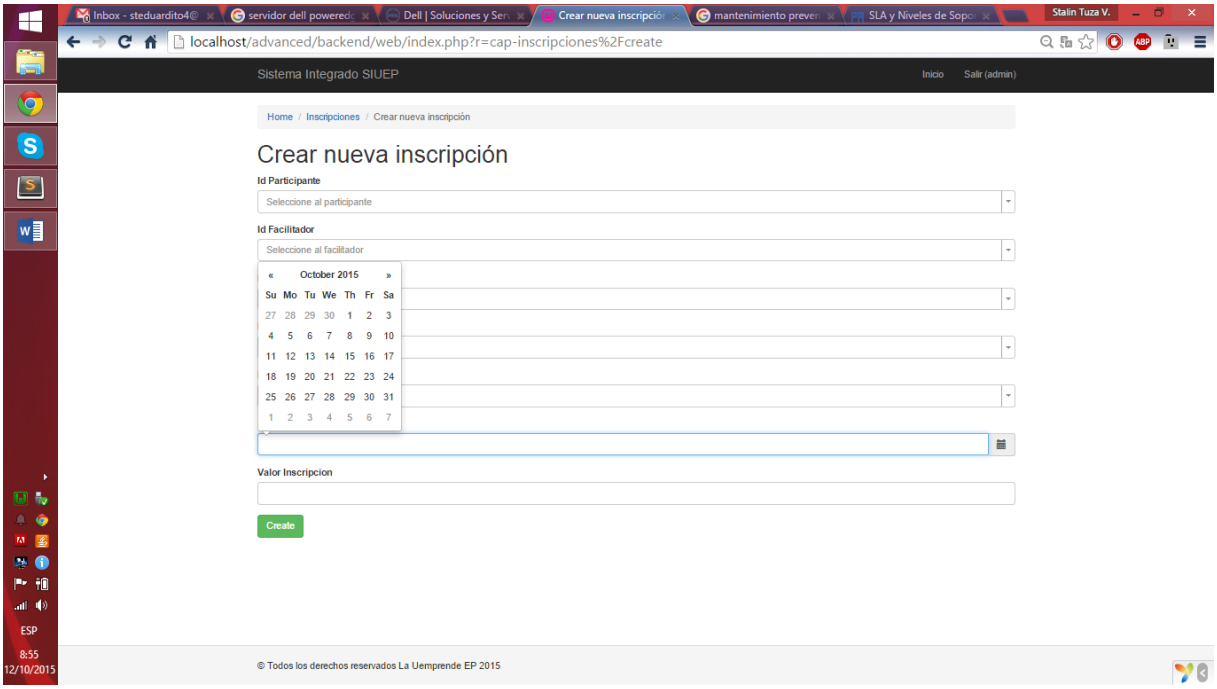

Ilustración 20: Formulario Inscripciones

Fuente: El Autor

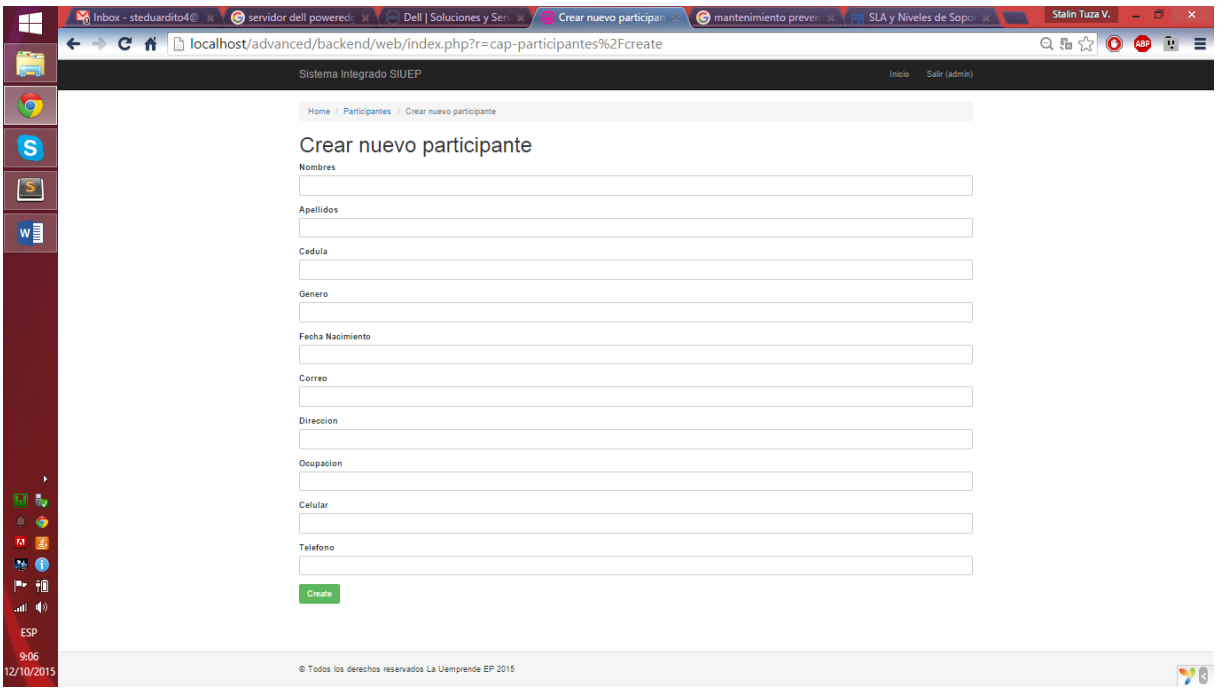

Ilustración 21: Ingreso de particpantes

Fuente: El Autor

4.7 Implementación

4.7.1. Políticas de programación.

Dentro de la AS, es fundamental definir los parámetros que deben regir los procedimientos indicados en los capítulos anteriores, implementando correctas y funcionales políticas de programación, que ayudarán a que en el desarrollo de módulos consecuentes, no exista confusiones, y se mantenga los lineamientos y unificación, tanto de la base de datos, como de la capa del modelo de negocio.

Por ello se establece las siguientes políticas:

Modelado de base de datos.

En el modelado de un nuevo módulo del sistema debe establecerse el nombre del módulo, el cual debe ser conciso, entendible, y que refleje toda la estructura que será implementada.

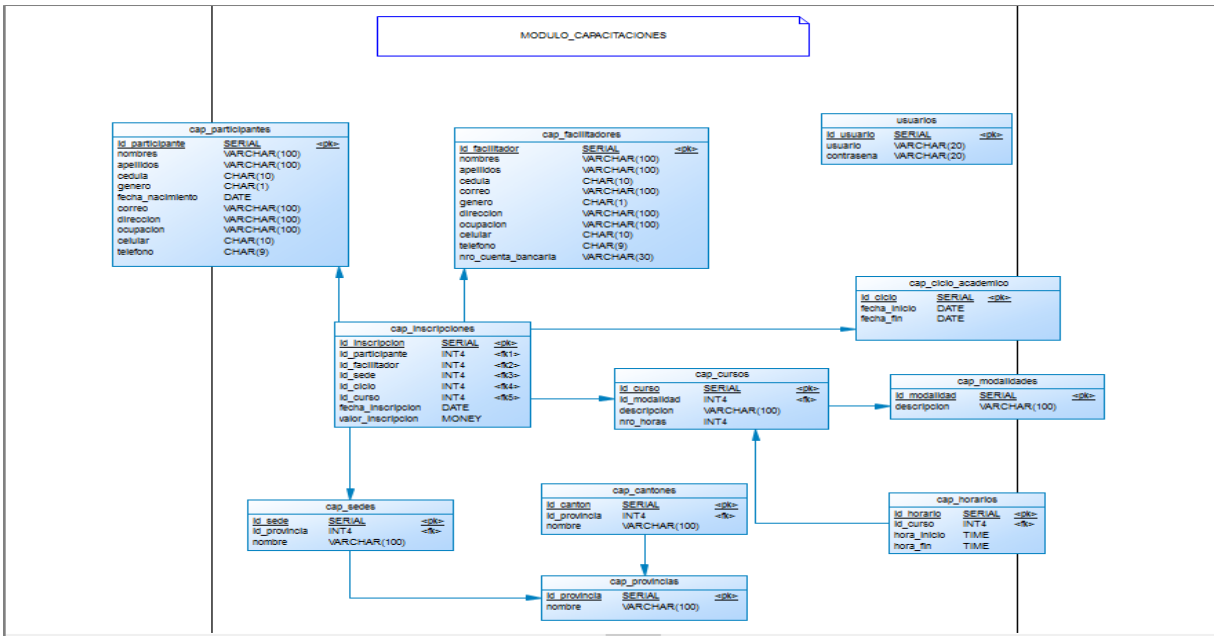

Gráfico 10: Diagrama de base de datos

Fuente: El Autor

Como se refleja en el diagrama 10: Diagrama de base de datos, el nombre “Módulo Capacitaciones” enseguida deja entrever que se trata específicamente de las capacitaciones que ofrece la institución, indistintamente de cual sea el curso

Nomenclatura de tablas

A la hora de modelar una base de datos, una forma eficiente y eficaz a la hora de poner nombres, y la establecida para la presente AS, es la siguiente:

- Tomar las iniciales en mayúscula del módulo como identificador de la tabla.
- Tomar las 3 primeras letras en minúscula del nombre del módulo
- Para la separación o espacios usar el guion bajo, seguido del nombre en minúsculas de la tabla.
- El nombre de la tabla debe reflejar el contenido específico de lo que va a contener, debe estar en plural. (Zend Technologies, 2015)

Ejemplo:

Paso 1: Tomar las 3 primeras letras en minúscula del nombre del módulo.

cap

Paso 2: Separar los espacios con guion bajo.

cap_

Paso 3: Nombre acorde al contenido.

cap_participantes

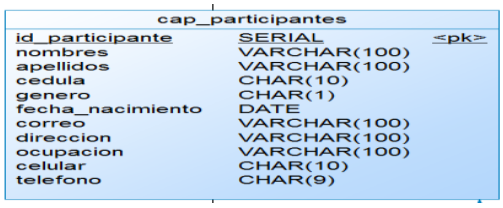

Gráfico 11: Tabla participantes

Fuente: El Autor

Nomenclatura de las clases, en el modelo de negocio.

Los nombres de las clases podrán contener únicamente caracteres alfanuméricos, los números están permitidos pero no son recomendados.

Si un nombre de una clase se compone de más de una palabra, la primera letra de cada palabra debe ser en mayúscula.

Las letras mayúsculas sucesivas no están permitidas, en ningún caso (Zend Technologies, 2015)

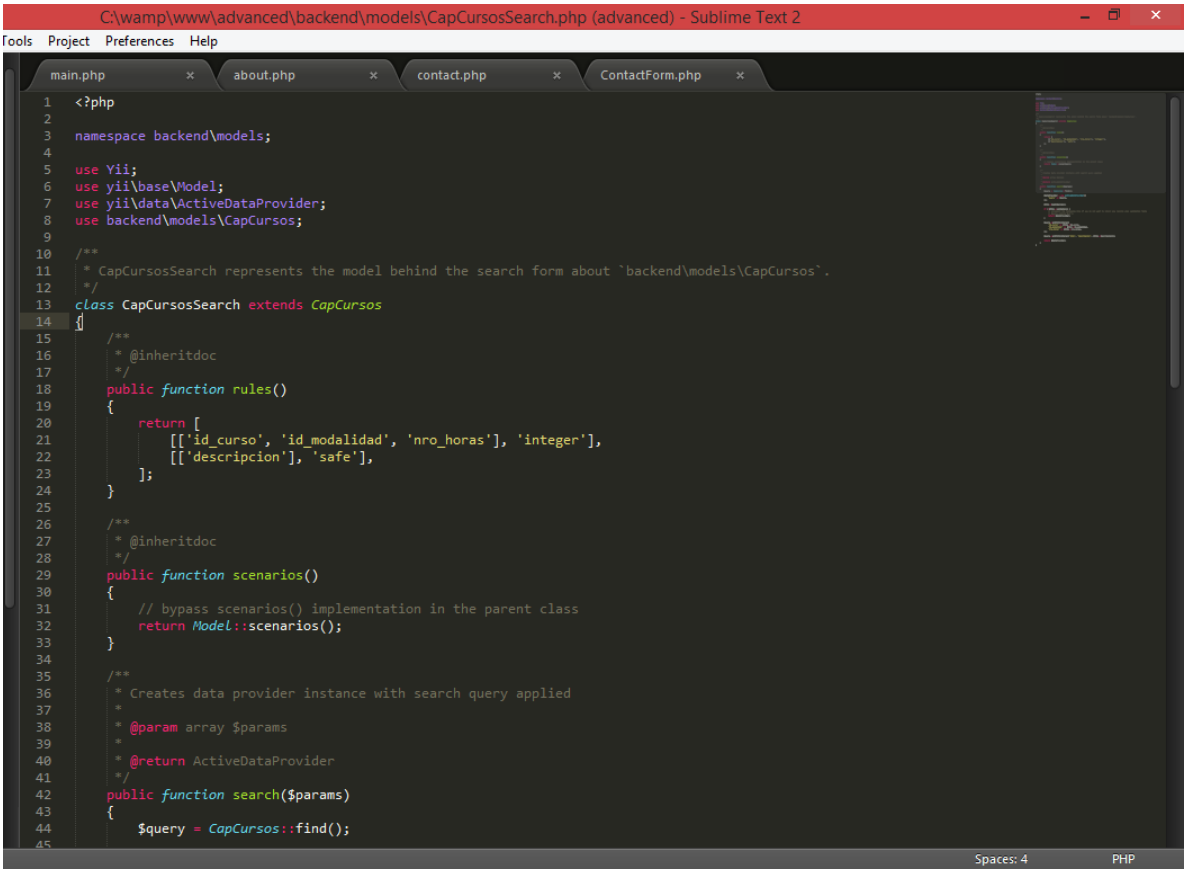

Ilustración 22: Modelo de negocios - Nomenclatura

Fuente: El Autor

Nomenclatura de archivos

Cualquier archivo que contenga código php, debe terminar con la extensión ".php", con la excepción de los scripts de vista. (Zend Technologies, 2015)

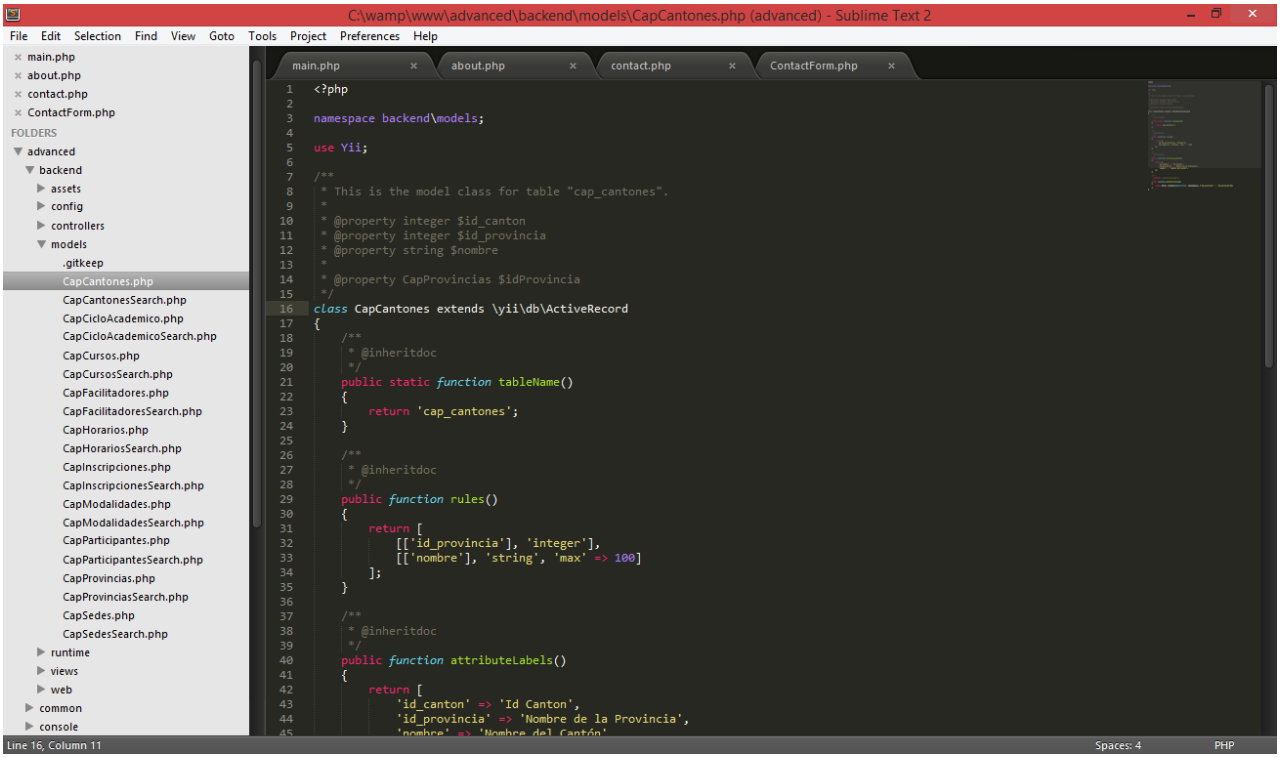

Ilustración 23: Nomenclatura de archivos

Fuente: El Autor

Nomenclatura de funciones y métodos.

De la misma manera que sucede con la nomenclatura de clases se permiten únicamente caracteres alfanuméricos, con la misma recomendación de no usar números en la manera de lo posible, adicionalmente no se permiten en ningún caso los guiones.

Los nombres de las funciones deben empezar con letras minúsculas, en el caso de que un método o función contenga dos o más palabras, la primera letra de cada palabra nueva debe empezar con letra mayúscula, lo que generalmente se conoce como "camelCase".

Los nombres de las funciones deben ser lo más detallados posible, ya que a futuro el programador debe saber a detalle, o lo más claro posible, cual es el propósito para el que se

creó y su comportamiento, ya que no necesariamente el programador será el mismo que desarrolló el sistema.

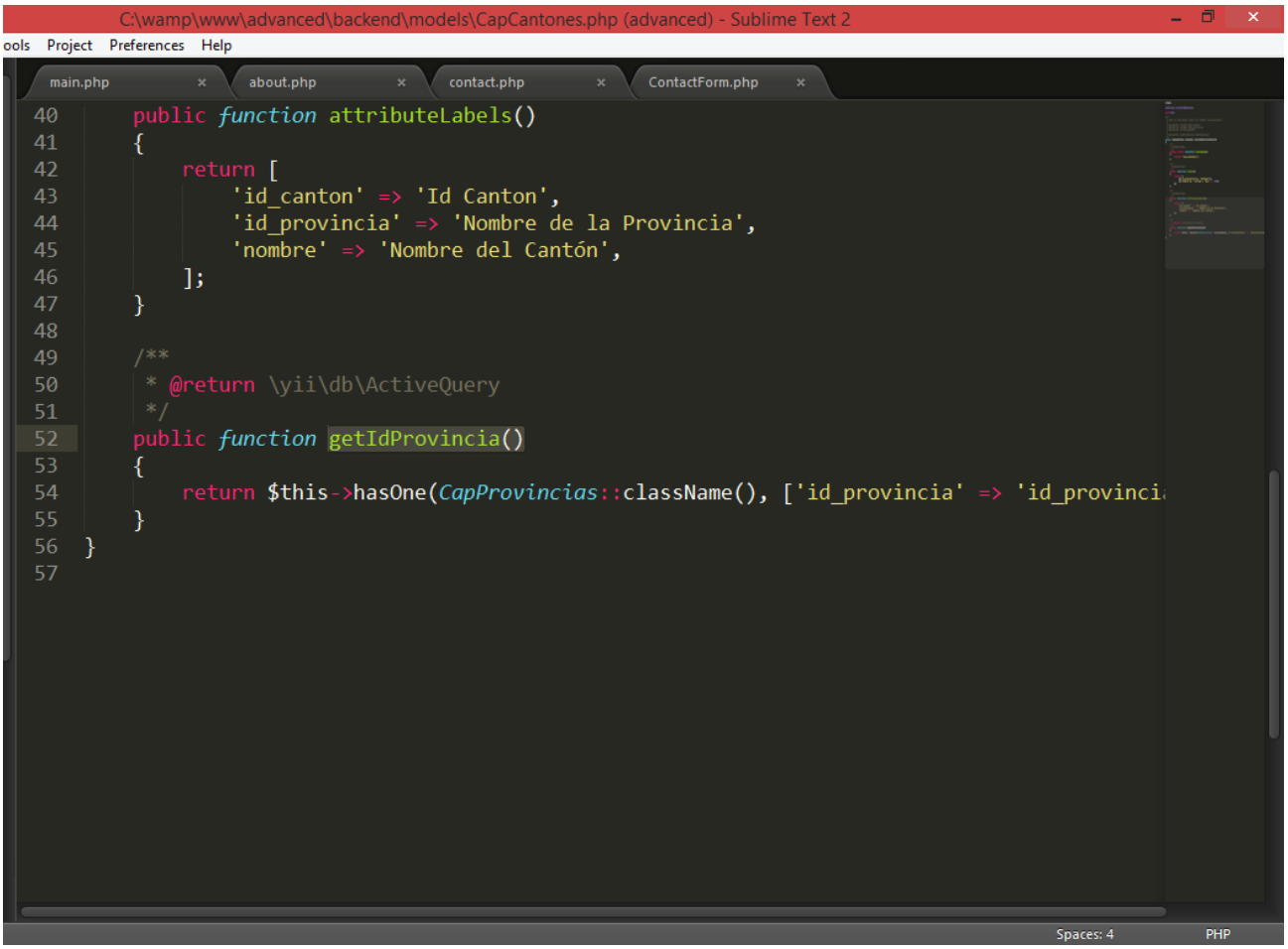

Ilustración 24: Nomenclatura de funciones

Fuente: El Autor

Como se puede apreciar, el nombre getIdProvincia, cumple con todos los requisitos mencionados anteriormente, el cual es obtener el id de la provincia en el modelo, además expone claramente cuál es el objetivo para el que fue implementado.

Nomenclatura de variables.

Nuevamente el estándar es usar solo caracteres alfanuméricos, en los casos que una variable sea declarada como protegida, o privadas, el primer caracter puede ser un subrayado, conocido generalmente por signo de dólar $, en el uso común.

Al igual que los nombres de las funciones, estos deben empezar con una letra minúscula y seguir el camelCaps.

Nomenclatura de constantes

En PHP, generalmente las constantes las define el programador, tanto para facilitar el entendimiento o para simplificar el código, optimizando recursos.

Ahora la nomenclatura sugerida es que todas las constantes sean escritas en mayúscula, en caso de ser más de una palabra, deben ser separadas con guion bajo, seguido de la palabra siguiente en mayúsculas todo. (Zend Technologies, 2015)

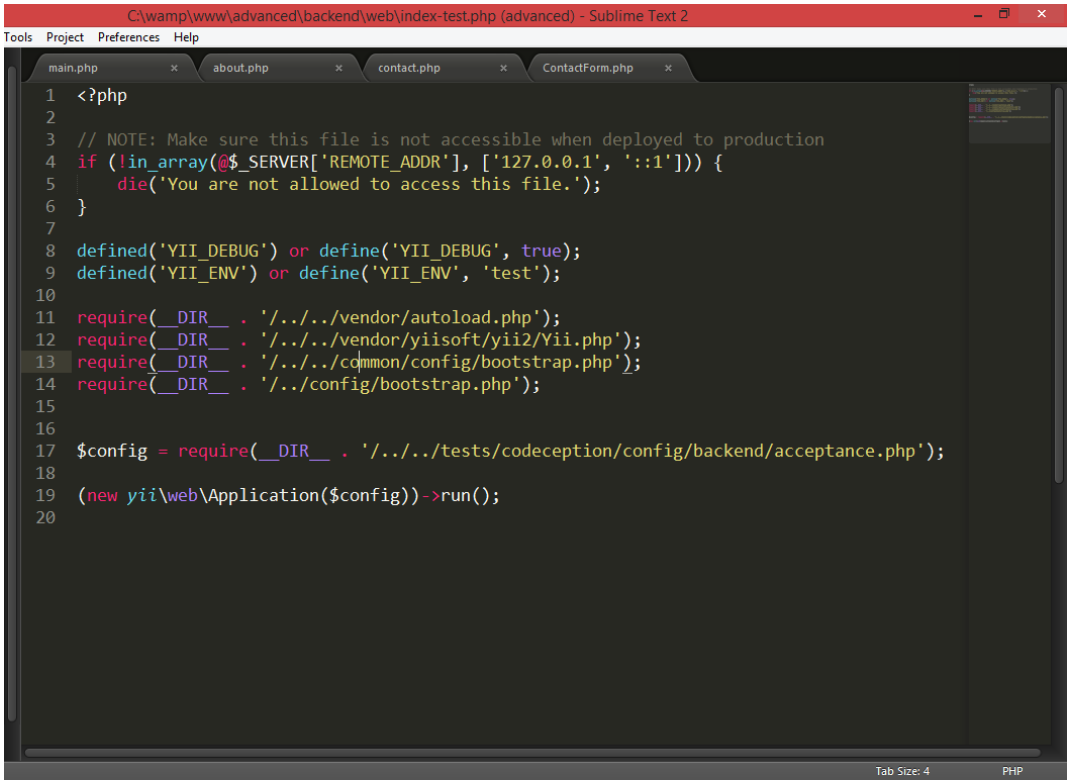

Ilustración 25: Nomenclatura de constantes

Fuente: El Autor

4.7. Diagrama de base de datos. (Módulo Capacitaciones)

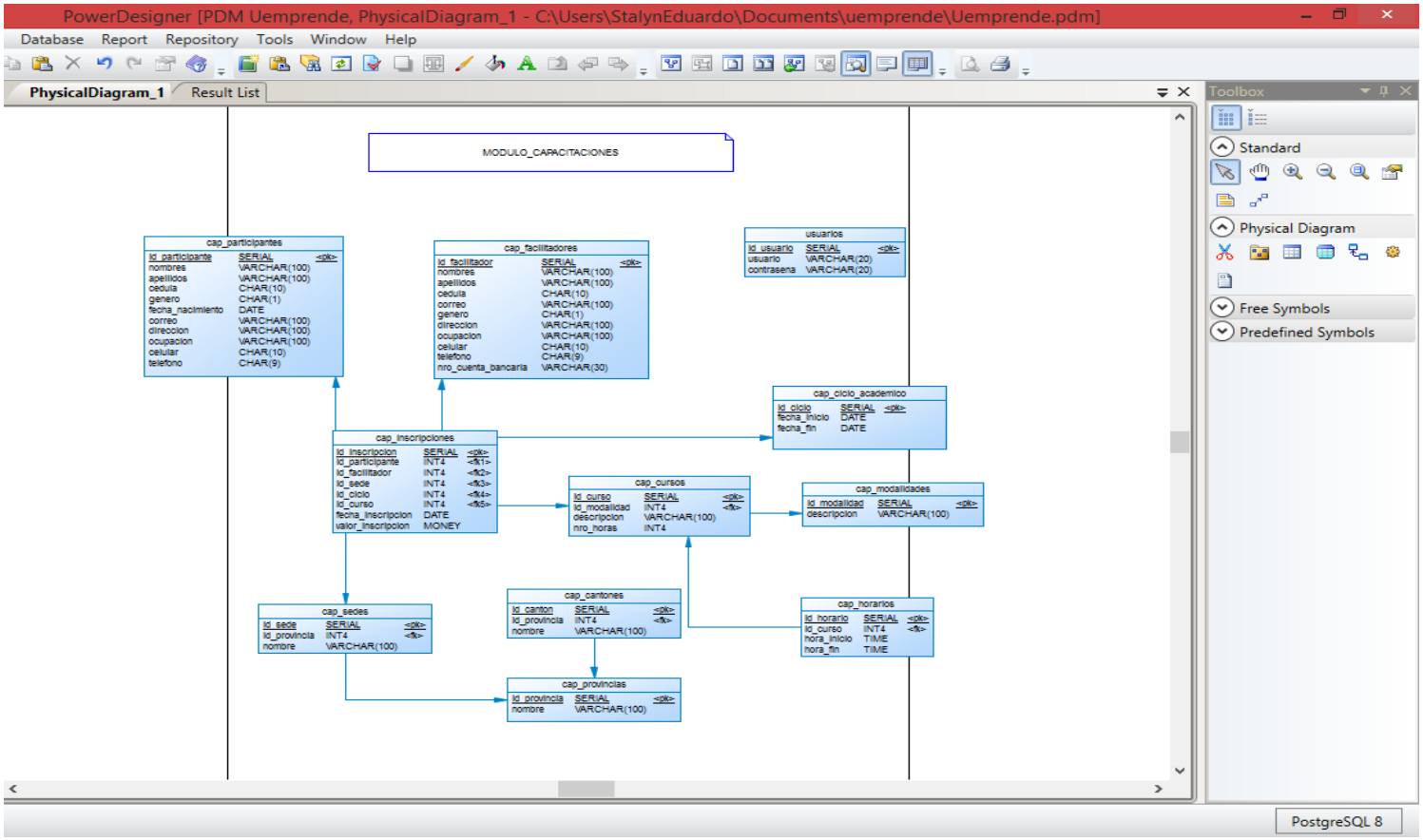

Ilustración 26: Diagrama de base de datos – Módulo Capacitaciones

Fuente: El Autor

CAPITULO V

- Conclusiones
- Recomendaciones.

5.1. Conclusiones

Al finalizar el presente proyecto de investigación se puede concluir lo siguiente:

- Mediante la recopilación correcta de la información de la Uemprende EP y de cualquier empresa, se puede diseñar una Arquitectura de Software, robusta y con estándares definidos.
- Para poder definir correctamente una AS, se debe contar con la información de cada proceso que se maneja de manera interna, y un mapa de procesos es lo ideal para empezar.
- Una empresa que no cuente con una AS definida como la Uemprende EP, difícilmente podrá iniciar la implementación de sistemas informáticos en base a sus necesidades, a corto plazo, ya que para cada sistema deberá hacer todos los pasos que una AS lo puede definir para todos los sistemas que vayan a desarrollarse.
- El uso de tecnologías definidas en la AS, como lenguaje de programación determina la escalabilidad que se obtendrá en futuros desarrollos.
- Los frameworks de desarrollo no solo permiten un desarrollo rápido de sistemas robustos, sino que también se acoplan a la AS planteada, como Yii V 2.x.x.
- La AS basada en PHP, con una base de datos Enterprise, Postgresql v 9.x, lenguaje de programación orientado a objetos PHP v 5.x, y como framework de desarrollo Yii Framework, se ajusta a las necesidades de escalabilidad de la Uemprende EP.

5.2. Recomendaciones

- Al estar incluida la información de la empresa en el presente proyecto se debe empezar rápidamente con el desarrollo de los sistemas planteados en la AS, para poder conformar el Sistema Integrado Uemprende.
- Poner en conocimiento a las empresas que el área de TIC, no solo es un centro de programación, sino un área donde las necesidades de la empresa pueden automatizarse y generar utilidad, utilizando el mapa de procesos o en sí, los procesos y automatizarlos.
- Realizar temas de tesis consiguientes para implementar cada módulo presentado en la presente arquitectura de software, tomando en cuenta las diferentes observaciones y características que constan en el documento.
- Utilizar el presente proyecto como una guía para estudiantes de carreras afines al desarrollo de AS, ya que muchas veces un programador no siempre puede ser un Arquitecto de Software.

Referencias

Auditores, C. y. (Agosto de 2012). *auditoresycontadores*. Obtenido de http://www.auditoresycontadores.com/articulos/contabilidad/impuestos/89-tipos-de-empresas-en-ecuador

Cali, U. J. (9 de 2012). *Ambientes Visuales de Programación Aplicativa*. Obtenido de http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:s2_conceptosdemodelado.pdf

Cervantes, H. (2011). *SG Buzz.* Obtenido de http://sg.com.mx

Directorio La Uemprende EP. (2014). Plan Estratégico Institucional Empresa Pública La Uemprende EP. *Plan Estratégico Institucional Empresa Pública La Uemprende EP*. Ibarra, Imbabura, Ecuador.

Directorio Uemprende EP. (2014). Plan Estratégico Institucional Empresa Pública La Uemprende EP . Ibarra.

Duggan, D. (2012). *Enterprise Software Architecture and Design : Entities, Services, and Resources.* Wiley.

Fabien Potencier, F. Z. (2011). *Symfony 1.4, la guía definitiva.* LibrosWeb.

Facultad de Ingeniería - Universidad de Uruguay. (2013). *Facultad de Ingeniería*. Obtenido de http://www.fing.edu.uy

homenet. (2015). *Homenet*. Obtenido de http://www.homenet.net.ec/sitio/index.php?option=com_content&view=article&id=80&Itemid=480&lang=es

IBM. (2015). *IBM*. Obtenido de http://www.ibm.com/developerworks/ssa/websphere/techjournal/0904_badawi/images/image002.gif

industrialxp. (2011). *industrialxp.org*. Obtenido de http://www.industrialxp.org/

jurídicas, I. d. (2012). *Biblioteca Jurídica Virtual*. Obtenido de http://biblio.juridicas.unam.mx/

Kent, B. (2011). *Extreme Programming Explained.* NJ: Pearson Education.

Kruchten, P. (2010). *Springer*. Obtenido de http://link.springer.com/chapter/10.1007/978-0-387-35563-4_33

Mondaray, S. G. (2012). *GodTIC*. Obtenido de http://www.godtic.com/

Pressman, R. (2014). *Ingeniería del software: Un enfoque práctico.* Mexico: McGraw-Hill.

Reynoso, C. (1 de 06 de 2014). *Carlos Reynoso* . Obtenido de http://carlosreynoso.com.ar/archivos/arquitectura/Arquitectura-software.pdf

tinetworksolutions. (2012). Obtenido de http://www.tinetworksolutions.com/images/graf_ctnet3.png

Uemprende. (2013). *Uemprende*. Obtenido de Uemprende: http://www.lauemprende.com/site/organigrama.html

Universitario, H. C. (20 de Julio de 2014). *La Uemprende*. Obtenido de La Uemprende: http://www.lauemprende.com

Victor Hugo Jimenez Torres, W. T. (4 de 12 de 2014). Lenguajes de Patrones de Arquitectura de Software: Una Aproximación Al Estado del Arte. Pereira, Colombia .

Vogel, O., Arnold, I., & Chughtai, A. (2011). *Software Architecture.* Springer.

Wells, D. (8 de 10 de 2013). *Extreme Programming*. Obtenido de http://www.extremeprogramming.org/

Wikipedia, La enciclopedia libre. (3 de 14 de 2013). *Wikipedia, La enciclopedia libre*. Obtenido de http://es.wikipedia.org/w/index.php?title=Arquitectura_en_pizarra_(inform%C3%A1tica)&oldid=65195044.

Yii Framework. (2015). *Fundamentos Yii*. Obtenido de http://www.yiiframework.com/doc/guide/1.1/es/basics.mvc

Yii2. (2014). *Yii Audith Module*. Obtenido de http://cornernote.github.io/yii-audit-module/#installation

Zend Technologies. (2015). *PHP File Formating*. Obtenido de http://framework.zend.com/manual/1.12/en/coding-standard.php-file-formatting.html