UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

"Trabajo de grado previo a la obtención del título de Ingeniero en sistemas y computación"

TRABAJO DE GRADUACIÓN

Título del proyecto

Análisis de las principales tecnologías multiplataforma para aplicaciones web. Aplicado al Sistema de seguimiento de graduados y egresados de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo.

Autor:

Marcela Cuello Olmedo

Director

Ing. Fernando Molina

RIOBAMBA – ECUADOR

2016

Los miembros del Tribunal de Graduación del proyecto de investigación de título:" Análisis de las principales tecnologías multiplataforma para aplicaciones web. Aplicado al Sistema de seguimiento de graduados y egresados de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo." Presentado por: Marcela Cuello Olmedo y dirigido por: Ing. Fernando Molina

Una vez escuchado la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia de la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Fernando Molina ----------

Director del proyecto **Firma**

Ing. Samuel Moreno ----------

Presidente del tribunal **Firma**

Ing. Danny Velasco ----------

Miembro del tribunal **Firma**

AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este proyecto de Graduación, nos corresponde exclusivamente a: Cuello Olmedo Blanca Marcela y del Ing. Fernando Molina; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

Blanca Marcela Cuello Olmedo
060449112-6

AGRADECIMIENTO

Agradezco sobre todo a Dios, por darme la fuerza y valentía para salir adelante, aun en los momentos más difíciles de nuestras vidas.

A mis Padres por estar siempre a mi lado en cada paso que doy en esta vida, brindándome su amor y apoyo incondicional.

A mi esposo y a mi hija por ser ese eje fundamental en mi vida que me inspira a seguir luchando cada día.

A mi tribunal de tesis por su desinteresada colaboración en el desarrollo de la presente investigación.

A la Facultad de Ingeniería por abrirnos las puertas del conocimiento, habernos inculcado los valores de ética, respeto y brindarnos la más óptima formación profesional.

A todas las personas que marcaron la diferencia durante la carrera, haciendo de esta una etapa inolvidable para mi vida a todos ellos muchas gracias.

DEDICATORIA

La presente tesis se la dedico a toda mi familia que gracias a su apoyo incondicional pude concluir mi carrera.

A mi padre Luis Cuello por su cariño y por brindarme los recursos necesarios y estar a mi lado apoyándome y aconsejándome siempre.

A mi madre Dolores Olmedo por hacer de mí una mejor persona a través de sus consejos, enseñanzas y amor.

A mi esposo Orlando Moyano por estar a mi lado siempre, ayudándome en mis momentos difíciles y brindándome su amor y comprensión.

A mi hija Naydelin Moyano Cuello por inspirarme para que mis sueños y anhelos no se queden truncados.

A mis hermanas Verónica, Magaly, Catherine por estar siempre presentes, acompañándome en los buenos y malos momentos.

A mi tío Marco Cuello que creyó en mí y supo que me convertiría en Ingeniera en Sistemas.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Investigación para aplicar la mejor tecnología multiplataforma para el desarrollo web, en la creación y operación del sistema de seguimiento de egresados y graduados de la Universidad Nacional de Chimborazo, con la finalidad de proveer una aplicación eficientemente en el procesamiento de la información recogida en entornos web. Elementos hardware (2 computadores), software (NetBeans 7.2, MySQL – Front, Power Designer, Star UML) y con la utilización de los métodos de investigación (científico) se logró plasmar el producto requerido. Además, se realizó un análisis comparación de tecnologías JSP, Python, Ruby, consiguiendo a JSP como tecnología multiplataforma ideal para el desarrollo de aplicaciones web que se adapten a ambiente heterogéneos. Las pruebas con usuarios del sistema (validar el sistema) y encuestas (recoger información de las tecnologías multiplataforma) han sido las técnicas utilizadas.

Se ha logrado crear un producto web heterogéneo, que acumula todos los requerimientos de la especificación de software y aplicando una programación estructurada se ha alcanzado un buen manejo de la tecnología JSP, lenguaje HTML tradicional y Java Script.

Mediante el uso de Chi – Cuadrado con un grado de libertad de 3 y un nivel de confianza t= 0,05 tenemos Chi – Cuadrado calculado = 9,41, mientras que el valor de Chi – Cuadrado Tabla = 7,814, de lo cual se concluye que se aprueba la Hipótesis H1 y se rechaza la Hipótesis H0.

El estudio de las tecnologías multiplataforma permitirá desarrollar eficientemente, el sistema de seguimiento de graduados y egresados de la facultad de Ingeniería de la Unach.

Se toma en consideración que el uso de las tecnologías multiplataforma JSP, Ruby o Python exige explotar de la mejor manera los beneficios que poseen, apoyarse buenas prácticas de programación, conocimientos básicos y/o profundos de lenguajes de desarrollo web (Java), HTML, Java Script, IDEs de desarrollo y editores propios de cada tecnología, facilita el buen desarrollo de aplicaciones exigentes.

UNIVERISIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CENTRO DE IDIOMAS

MsC Janneth Caisaguano 09 de Marzo de 2016

SUMMARY

Research to apply the best platform technology for development, the creation and operation of the monitoring system of graduates and National University of Chimborazo grads in order to provide an efficiently application to process information in web environments. Hardware items (2 computers), software (NetBeans 7.2, MySQL - Front, Power Designer, Star UML) and the use of research methods (scientific) managed to capture the required product. In addition, a comparison of JSP, Python, Ruby analysis technologies, getting JSP technology as a platform ideal was performed for developing web applications that adapt to heterogeneous environment. User testing system (validate the system) and surveys (gather information from multiplatform technologies) techniques have been used.
It has created a heterogeneous web product, which accumulates all the requirements of the specification of software and applying a structured programming has reached a good management of JSP technology, traditional HTML and Java Script.
By using Chi - Square with a degree of freedom of 3 and a confidence level t = 0.05 have Chi - Square calculated = 9.41, while the value of Chi - Square Table = 7,814, concluding that the hypothesis H1 is approved and the Hypothesis H0 is rejected.
The study of platform technologies will enable the efficient development, the tracking system of Engineering Faculty grads and graduates
It takes into consideration that the use of cross-platform technologies JSP, Ruby or Python required to exploit in the best way the benefits they have, support good programming practices, basic knowledge and / or depth of web development languages (Java), HTML, Java Script, development IDEs and editors themselves each technology facilitates the efficient development of demanding applications.

COORDINACION

INTRODUCCIÓN

En el presente trabajo de investigación se estudia las tecnologías multiplataforma para el desarrollo de aplicaciones web, entendido que una aplicación multiplataforma en informática se refiere al término que se usa para sistemas operativos, programas, lenguajes de programación que pueden funcionar en diferentes plataformas es decir dos o más, esa puede ser la característica principal.

El objetivo principal de la investigación consiste en analizar las tecnologías multiplataforma para el desarrollo eficiente de aplicaciones web, validar propiedades tecnológicas, especificaciones técnicas, el soporte que facilitan para el desarrollo de aplicaciones, elementos y peculiaridades en la programación que cada una utiliza en los diferentes entornos y editores de desarrollo.

En el primer capítulo se bosqueja el marco referencial que enfatiza el planteamiento del problema, su formulación, el objetivo general y los objetivos específicos.

El segundo capítulo, presenta una introducción general a las tecnologías multiplataforma Java Sever Page, Python y Ruby, que son tecnologías para el desarrollo web que va cobrando mucho interés en los programadores por las ventajas técnicas, facilidad de programar y por ser accesibles por cada usuario. Además, se hace hincapié en cada tecnología de específica definición, características de la tecnología, estructura y características del lenguaje en el que se apoya su funcionalidad, ventajas que trae el desarrollo en cada tecnología, herramientas de desarrollo. En este mismo capítulo se realiza un análisis comparativo entre las tecnologías multiplataforma JSP, Python y Ruby, para lo cual se plantea el criterio de comparación ***Características de las tecnologías multiplataforma (***Aspectos técnicos que permiten identificar información relevante de cada tecnología, de manera que facilitan su uso en la ejecución de tareas programación, configuración, implementadas en ambientes heterogéneos***);*** apoyada de un conjunto de variables que permiten valorar dichas características (***Propiedades tecnológicas, Especificaciones Técnicas, Soporte en el desarrollo de aplicaciones web, Peculiaridades de programación, Elementos del lenguaje***

Y de acuerdo al análisis y la comparación se determina que la tecnología JSP cumple mayoritariamente con el criterio establecido en un 98,3% siendo la tecnología multiplataforma aceptada para el desarrollo de sitios web adaptables a ambientes heterogéneos.

En el tercer capítulo se toma como base el análisis comparativo entre tecnologías multiplataforma, del segundo capítulo, para desarrollar el Sistema Web (SYGEGE) que servirá para el seguimiento de egresados y graduados de la Universidad Nacional de Chimborazo, bajo la metodología de desarrollo de software de Craig Larman y de la ingeniería web se detalla el desarrollo del sistema de una forma flexible, interactiva e incremental, se enfatizan casos de uso, diagramas de secuencia, de actividades, diseño de clases que permiten tener una visión de la funcionalidad que propone la solución informática.

En el cuarto y último capítulo se exterioriza la metodología de investigación aplicada en la comprobación dela hipótesis, una investigación bibliográfica, documental y con la ayuda de instrumentos para la recolección de datos (encuesta), además de la técnica estadística Chi – Cuadrado se pudo demostrar que: El estudio de las tecnologías multiplataforma **SI** permitirá desarrollar eficientemente, el sistema de seguimiento de graduados y egresados de la Universidad Nacional de Chimborazo.

Se finaliza con las debidas conclusiones (las características de las tecnologías multiplataforma permiten crear aplicaciones para diferentes ambientes tecnológicos, productos software de calidad, seguros, mantenibles, escalables, portables y funcionales) y recomendaciones (apoyarse buenas prácticas de programación, conocimientos básicos y profundos de lenguajes de desarrollo web, HTML, Java Script, IDEs y editores propios de cada tecnología JSP, Python o Ruby facilita el óptimo desarrollo de aplicaciones exigentes que tienen como ambiente de trabajo plataformas diferentes).

CAPITULO I

1. Marco referencial

1.1. Título del Proyecto

ANÁLISIS DE LAS PRINCIPALES TEGNOLOGIAS MULTIPLATAFORMA PARA APLICACIONES WEB. APLICADO AL SISTEMA DE SEGUIMIENTO DE GRADUADOS Y EGRESADOS DE LA FACULTAD DE INGENIRIA DE LA UNIVERSIDAD NACIONAL DE CHIMBORAZO.

1.2. Problematización

El avance del desarrollo tecnológico, los métodos utilizados para realizar encuestas han experimentado grandes cambios, y sin duda el medio más atractivo actualmente para llegar a los encuestados es internet.

El aprovechamiento de las ventajas que ofrecen las tecnologías web para crear un entorno uniforme que faciliten por una parte el diseño y publicación de cuestionarios y por otra parte el acceso de los egresados y profesionales de la facultad de Ingeniería de la Unach es una cuestión que requiere el uso de software libre y multiplataforma, la motivación que se tiene frente a esta necesidad es analizar que tecnología web permite un diseño de los cuestionarios personalizados a los requisitos de la facultad de Ingeniería de la Unach, la manera en que se deben publicar dichos cuestionarios y la recolección de los datos (Cuello, 2014).

JSP, Ruby y Python son tecnologías para el desarrollo de aplicaciones web para cualquier necesidad empresarial, integran entornos de diseño, que pueden configurar la publicación de encuestas, que permite la creación y manipulación de cuestionarios de forma intuitiva, y posibilita la reutilización de los diseños de encuestas pasadas para trabajos futuros. Todo ello bajo una política de control de accesos y privilegios flexible y consistente, que permite a los diseñadores proteger su trabajo de otros usuarios, a la vez que favorece la colaboración entre ellos.

Realizar un seguimiento automatizado por medio de un sistema web a los egresados y profesionales de la facultad de Ingeniería de la Unach es una necesidad a ser resuelta ante los requerimientos que exige el CEACES (Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior) para alcanzar la acreditación institucional bajo el uso de software libre y la participación de estudiantes se fortalece la presencia de la Unach en el campo educativo del Ecuador, de esta manera se constituye en una estrategia que permite mantener información actualizada de las debilidades y

fortaleza que se manifiestan en el campo laboral; y con lo cual las autoridades de la facultad estarán en la capacidad de implementar cambios a favor de la institución con los antecedentes expuestos por profesionales y egresados.

Espero que el sistema para el seguimiento de graduados y egresados permita mostrar las amplias posibilidades que las tecnologías de desarrollo web pueden ofrecer en el mundo del internet, además del uso de estándares abiertos que ofrecen al campo de las encuestas, y que sirva a la facultad de Ingeniería de la Unach como una base para mejorar sus servicios y que cubra a medida sus necesidades institucionales.

1.3. Formulación del problema

¿El estudio de tecnologías multiplataforma para la web permitirá el desarrollo de aplicaciones eficientes y eficaces para ambientes reales de ejecución?

1.4. Hipótesis

El estudio de las principales tecnologías multiplataforma permitirá desarrollar eficientemente, el sistema de seguimiento de graduados y egresados de la facultad de Ingeniería de la Universidad Nacional de Chimborazo.

1.4.1. Identificación de las variables

Variable independiente: Estudio de las tecnologías multiplataforma para aplicaciones web.

Variable dependiente: Desarrollo eficiente del sistema de seguimiento a graduados y egresados de la facultad de Ingeniería de la Universidad Nacional de Chimborazo.

1.5. Objetivos

1.5.1. Objetivo general

Analizar las tecnologías multiplataforma para aplicaciones web para el desarrollo del Sistema de seguimiento de egresados y graduados de la Facultad de Ingeniería de la Unach.

1.5.2. Objetivos específicos

- Analizar las tecnologías multiplataforma disponibles para el despliegue de aplicaciones web.

- Implementar una aplicación web multiplataforma que permita la gestión de los procesos del seguimiento de egresados y graduados de la Facultad de Ingeniería de la Unach.
- Estudiar los procesos de ingeniería de software para el desarrollo de aplicaciones web que puedan adaptarse y utilizarse en el sistema de seguimiento de egresados y graduados de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo.

1.5.3. Justificación

En el Ecuador la labor que realiza el Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior, CEAACES, es una política pública para garantizar una expansión de las Instituciones de Educación Superior con calidad académica y relevancia social. Determinar las condiciones de una institución, carrera o programa académico, mediante la recopilación sistemática de datos cuantitativos y cualitativos permitirá emitir un juicio o diagnóstico, analizando sus componentes, funciones, procesos, a fin de que sus resultados sirvan para reformar y mejorar el programa de estudios, carrera o institución. El CEAACES obliga a todos los Organismos e Instituciones que integran el Sistema de Educación Superior del Ecuador a cumplir con la Ley. Las universidades y escuelas politécnicas, 57 en total, vienen trabajando en el proceso de autoevaluación para cumplir con la acreditación que exige la nueva Ley de Educación Superior vigente en el Ecuador.

El desafío principal de la evaluación y acreditación de la Calidad de la Universidad Nacional de Chimborazo como institución de educación superior en el Ecuador es contribuir a la construcción de políticas que atiendan a: promover el desarrollo científico-tecnológico, la formación de ciudadanos y profesionales capaces de trabajar para construir una sociedad más justa e integrada y el crecimiento económico. Este desafío exige incorporar nuevas formas de gestionar los procesos institucionales, aprovechando los recursos tecnológicos se puede desarrollar aplicaciones informáticas que contribuyan al aseguramiento de la calidad Universitaria.

Si el objetivo es realizar un seguimiento a los profesionales y egresados de la facultad de Ingeniería de la Unach, entonces, se plantea desarrollar una aplicación para procesar información en cualquier plataforma, Linux o Windows, se piensa directamente en tecnologías de desarrollo y en lenguajes de programación para dichas plataformas. El contar con medios que den más sentido a las aplicaciones que se van desarrollando es dar

un paso de calidad, donde otorgar otro grado de interacción del desarrollador con las tecnologías de trabajo marca diferencias.

El despliegue de aplicaciones web, exige un grado de conocimiento del lenguaje en el que van a ser desarrolladas y de la plataforma que se utilice en su implementación y ejecución. El desarrollo web multiplataforma va cobrando más interés por sus avances acelerados en dar soluciones inmediatas a miles de clientes que cada vez exigen interacción en tiempo real, en cualquier lugar y con las mejores prestaciones del mercado. Las técnicas y herramientas que llevan a ser realidad un proyecto informático donde el trabajo de tecnologías multiplataforma para la web es un limitante que necesita orientación profesional y comprobada; se requiere conocimientos a nivel de frameworks, entornos de desarrollo, lenguajes de desarrollo, sistemas operativos, gestores de bases de datos, técnicas de programación y análisis desde el punto de vista de desarrollo. Desde una perspectiva del negocio es necesario profundizar en alcance del proyecto, tecnologías que ayudan a cumplir con el objetivo, como interrogantes más importantes.

Afortunadamente, se cuenta con tecnologías web que operan a gran nivel y en cualquier entorno, permiten la realización de trabajos en ambiente de propósito específico que se crean dinámicamente, permitiendo que se construyan aplicaciones (para gestionar información generada por su aplicación), que se ejecuten de forma colaborativa entre todos los nodos de interacción. Para hacer realidad esta colaboración es necesario el uso de técnicas interoperabilidad, de confidencialidad, de integridad y de seguridad.

Lo queda por decidir una vez que se haya definido realizar un estudio de Tecnologías web aplicado sistema web, es elegir que tecnologías son las que se van a utilizar, siendo estas las correctas para cada situación (recolección, presentación de encuestas). Existen tecnológicas a escoger una con mejores prestaciones que otras y se puede decidir por la que mejor se adapte al problema a ser resuelto.

Entonces al momento de desarrollar de una aplicación destinada a sistemas verticales (diferentes plataformas) existen factores importantes a tomar en cuenta, la ingeniería de software que no debe diferir de la ingeniería usada cuando se construye aplicaciones para la web, conocimiento del personal en el desarrollo de aplicaciones web, funcionalidades que se desea obtener, factores de mercado (Cuello, 2014).

CAPITULO II

2. Fundamentación teórico

2.1. Tecnologías multiplataforma.

JSP es una tecnología JAVA que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible en la especificación JSP 2.1. JSP's permite la actualización de código JAVA mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (Tag Librares) externas e incluso personalizadas[1].

Ruby es un lenguaje de programación orientado a objetos. Se basa sobre todo en la sintaxis de Perl y en la orientación a objetos de SmallTalk. Fue creado por el apones Yukihiro 'Matz' Matsumoto quien lo libero en 1995. Es distribuido bajo dos licencias: una propia y la GPL. Ruby es un lenguaje interpretado y no es de los más rápidos Mazt lo que quería hacer era un lenguaje divertido de programar, en el que el programador pudiera ser muy productivo en poco tiempo. Según Cuello, Matz, (2014).

Dentro de los lenguajes informáticos, Python, pertenece al grupo de los lenguajes de programación y puede ser clasificado como un lenguaje interpretado, de alto nivel, multiplataforma, de tipado dinámico y multiparadigma. A diferencia de la mayoría de los lenguajes de programación, Python nos provee de reglas de estilos, a fin de poder escribir código fuente más legible y de manera estandarizada. Estas reglas de estilo, son definidas a través de la *Python Enhancement Proposal* Nº 8 (PEP 8). (pág. 23), Según Eugenia Bahit, (2012).

En el desarrollo de aplicaciones web en distintas plataformas se utiliza entornos de desarrollo integrados, interfaz gráfica de usuarios, editores de aplicaciones; que ayudan al desarrollador a dar forma a las soluciones informáticas:[2]

IDES: Son las siglas de entorno de desarrollo integrado. Es una aplicación (entorno de programación) para desarrollar software que está compuesto normalmente por un editor

[1] Sun Microsystems. - fue una empresa informática que se dedicaba a vender estaciones de trabajo, servidores, componentes informáticos y software. Disponible en: http://www.ecured.cu/index.php/JSP

2 Solución Informática.- es una de tres herramientas informáticas o tareas diferentes: Procesos, Programas y Aplicaciones.Disponible en: http://elviajedelnavegante.blogspot.com/2010/10/herramientas-ide-gui-editor-para_18.html

de código, un compilador o un intérprete, herramientas auxiliares de desarrollo (creación de documentación, tests, empaquetado de software), un depurador y algunas veces un diseñador de interfaces gráficas incorporado.

GUI: Son las siglas de interfaz gráfica de usuario. Son herramientas para crear interfaces gráficas, esto es, componentes gráficos con los cuales el usuario interacciona con la aplicación. Hay algunos IDEs que contienen Gais integrados. Las Gais dependen de la plataforma gráfica en la que trabajemos.

Editor: Es una aplicación para escribir código de programación en el lenguaje deseado. Contiene todas las herramientas de edición necesarias, y algunas veces características propias de IDEs. Hay veces que cuesta distinguir entre un IDE y un editor por la cantidad de opciones que tiene este último.

RAD: Son las siglas de desarrollo rápido de aplicaciones. Dependiendo del autor sirve para designar a las aplicaciones de desarrollo de interfaces gráficas o a los IDEs con Gais integrados, o a las dos cosas. Es un término dado a plataformas de desarrollo como PowerBuilder, Visual Studio o Delphi, por poner ejemplos.

Shell interactivo: Es un intérprete con características especiales, que podrían incluir la completitud de código y el coloreado del mismo, navegación entre los namespaces, exportación de código, etc.

2.2. JSP

2.2.1. Definición

Java Server Pages (JSP), es una tecnología Java que permite a los desarrolladores de software generar dinámicamente HTML, XML u otros tipos de documentos, en respuesta al requerimiento de un cliente web. Esta tecnología permite que códigos Java y ciertas otras acciones predefinidas, sean integrados en contenido estático. La sintaxis JSP agrega etiquetas XML adicionales, llamadas acciones JSP, para ser usadas para invocar funcionalidades incorporadas. Esta tecnología también permite la creación de bibliotecas de etiquetas JSP, que actúan como extensiones a las etiquetas HTML y XML estándares. Los JSP's son compilados en forma de Java Servlets empleando un compilador JSP[3].

[3] *JPS. - Nuevas tecnologías en torno a sistemas de información basados en web, xml, jsp, asp y php.*

2.2.2. Características de JSP

JSP sigue la filosofía de la arquitectura JAVA de "escribe una vez ejecuta donde quieras". JSP se puede ejecutar en los sistemas operativos y servidores web más populares, como por ejemplo Apache, Netscape o Microsoft IIS.

Proceso de desarrollo abierto (Open Source). La API JSP se beneficia de la extendida comunidad JAVA existente.

Tags. La tecnología JSP permite a los desarrolladores crear nuevos tags. Así los desarrolladores pueden crear tags y no depender tanto de los scripts.

Reusabilidad entre plataformas. Los componentes JSP son reusables en distintas plataformas (Unix, Windows).

La ventaja JAVA. La tecnología JSP usa JAVA lenguaje de Script. Java es un lenguaje potente y escalable que los lenguajes de script (ASP). Las páginas JSP son compilados en Servlets por lo que actúan como una puerta a todos los servicios Java de Servidor y librerías Java para aplicaciones http. Java hace el trabajo del desarrollador más fácil ayuda a proteger al sistema de caídas, ayuda al manejo de la memoria protegiendo contra fallos de memoria y el duro trabajo de buscar los fallos de pérdida de punteros de memoria que pueden hacer más lento el funcionamiento de una aplicación. (pag. 27). Según Edelson, (2006)

Mantenimiento. Las aplicaciones que usan la tecnología JSP tienen un mantenimiento más fácil.

Java es un lenguaje estructurado y es más fácil de construir y mantenimientos grandes como aplicaciones modulares.

La tecnología JSP hace mayor énfasis en los componentes que en los Scripts, esto hace que sea más fácil de revisar el contenido sin que afecte a la lógica o revisar la lógica sin cambiar el contenido.

Debido a que la lógica JSP es abierta y multiplataforma, los servidores web, plataformas y otros componentes pueden ser fácilmente actualizados o cambiados sin que afecte a las aplicaciones basadas en la tecnología JSP. (pag. 189-192). Según I. Jacobson, (2010)

2.2.3. Características del Lenguaje

Conjunta el poder de Java en el servidor y la flexibilidad de HTML (HyperText Markup Language) en el browser.

No sólo se puede utilizar HTML, sino también XML (eXtensible MarkupLanguage) o WML (Wireless Markup Language).

Hace más fácil reusar componentes como JavaBeans y Enterprise JavaBeans los cuales realizan tareas más específicas.

Forma parte integral de Java 2 Enterprise Edition (J2EE).

Su función es saber como procesar una solicitud para crear una respuesta.

Soporta contenido dinámico que refleja las condiciones del mundo real.

Existe independencia entre la parte del diseño (interfaz) y la lógica (programa).

2.2.4. Ventajas de JSP

Se puede crear aplicaciones web que se ejecuten en varios servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. (Pag 76). Según Schmidt M. D., (2013.)

Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts ejecutables en el servidor en sintaxis Java. Por lo tanto, las JSP pueden ser escritas con un editor habitual HTML/XML. (Schmidt M. D., 2013., págs. 22-45)

JSP tiene muchas ventajas debido a sus alternativas:

a) Frente al HTML estático: el HTML normal no puede contener información dinámica, así que las páginas HTML no pueden estar basadas en la entrada del usuario a en fuentes de datos del lado del servidor. JSP es tan fácil y cómodo que es bastante razonable aumentar las páginas HTML, que sólo se benefician ligeramente por la inserción de datos dinámicos.
b) La parte dinámica de JSP está escrita en JAVA, así que es más poderoso y mejor para desarrollar aplicaciones que requieren componentes reutilizables. JSP es portable a cualquier sistema operativo y servidor web, es decir no está atado a un servidor en particular.

c) Por ser escrito en JAVA se conoce así mismo y por tal razón tiene una extensa API para el trabajo en red, acceso a bases de datos, objetos distribuidos.
d) Frente a los servlets: JSP no provee ninguna capacidad que no pueda ser en principio, llevada a cabo con un servlet. En efecto, los documentos JSP son automáticamente traducidos en servlets.

2.2.5. Herramientas para desarrollar en JSP

2.2.5.1. IDES para JSP

Bluefish: Es un editor dirigido al desarrollo web, bajo licencia GPL[4] y se enfoca en la edición de páginas dinámicas e interactivas; sus principales características son: Soporte a multiproceso para archivos remotos mediante GVFs (sistema virtual de archivos de gnome) dependiendo de la configuración definida (acceso a FTP, SFTP, HTTP, HTTPS, WebDAv, CIFS). Según (J. García, 2008, págs. 34-40)

Figura 1: IDE Bluefish
Fuente: Entornos de desarrollo de aplicaciones web integrados Bluefish.

Soporte a los siguientes lenguajes de programación: ADA, ASP.net y VBS, C/C++, CSS, CFLM, CLOJURE, HTML, XHTML, HTML 5, JAVA y JSP, JAVA SCRIPT Y JQUERY, LUA, OCTAVE/MATLAB, PASCAL, PERL, PHP, PHYTON, RUBY, SHELL, SQL, SCHEME, VALA, XML.

Múltiples codificaciones de apoyo. Bluefish trabaja internamente con UTF8, pero puede permite almacenar los documentos usando cualquier otra codificación. Según (J. García, 2008, pág. 45)

Integración de programas externos tales como: pelusa, weblint, xmllint, ordenado, javac, o cualquier programa o secuencia de comandos propios para manejar el procesamiento de texto avanzado o la detección de errores.

[4] J & Innovación. - Entornos de desarrollo de aplicaciones web integrados. Disponible en: http://unillanosgsw.wikispaces.com/J%26H-INNOVATIONS

Eclipse: Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de GNU (GNU GPL). Según (S. Brown, 2012, pág. 78)

Figura 2. IDE Eclipse
Fuente: Entornos de desarrollo de aplicaciones web integrados, Eclipse.

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización. Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversion.4 e integración con Hibernate. Según (Edelson, 2006, pág. 180)

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesado de texto como Látex, aplicaciones en red como Telnet y Sistema de gestión de base de datos. La arquitectura plugins permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse. Y no tiene por qué ser usado únicamente para soportar otros lenguajes de programación.

NetBeans: NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. Según (Schmidt M. K., 2005, págs. 99-103)

Figura 3. IDE NetBeans
Fuente: Entornos de desarrollo de aplicaciones web integrados, IDE NetBeans.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. Según (J. García, 2008, págs. 108-110)

NetBeans permite crear aplicaciones Web con PHP 5, un potente debugger integrado y además viene con soporte para Symfony un gran framework MVC escrito en php. Al tener también soporte para AJAX, cada vez más desarrolladores de aplicaciones LAMP o WAMP, están utilizando NetBeans como IDE.

Gel: Gel es un IDE para java[5] que permite ver el código coloreado de acuerdo a la sintaxis, soporta (Java, JSP, HTML, XML, C, C++, Perl, Python, etc), permite deshacer y rehacer de manera ilimitada, contiene selección de columnas identificación de bloques colorea corchetes, paréntesis, chequea ortografía automáticamente, cierra llaves de manera automática, busca expresiones regulares, apto para Java y JSP.

DrJava Java IDE: DrJava es un entorno de desarrollo integrado para java distribuido bajo licencias GNU GPL.

[5] Enyo.- Información acerca del frameworks Enyo. Disponible en: http://nodejs.org/

BlueJ Java IDE: BlueJ es un Java IDE[6] que tiene un constructor (built-in) editor, un compilador, máquina virtual y un debuffer para programas en java, además tiene interfaces gráficas para clases, soporta edición gráfica y por texto.

JIPE Java IDE: Jipe es un entorno de desarrollo gratuito para java escrito en java. Que permite escribir y testear programas java y applets. Algunas características son sintaxis highlighting, portabilidad a cualquier sistema operativo ya que se corre con una máquina virtual java, el autor declara haber testeado en linux y windows.

JCreator Java IDE LE (Light Edition): Uno de los más rápidos editores java, y una de las versiones más livianas soporta syntax highlighting, wizards, class viewer, package viewer, tabbed documents, JDK profiles (permite trabajar con múltiples herramientas java agregables), contiene una intefaz customizable. JCreator corre en Windows 95, 98, NT4, 2000, ME.

2.2.5.2. Editor para JSP

TinyMCE Javascript WYSIWYG editor: Es un excelente editor HTML WYSIWYG y uno de los más usados.

Figura 4. Editor TinyMCE Javascript WYSIWYG
Fuente: TinyMCE: Un editor WYSIWYG en JavaScript

Principales características:

- Fácil de integrar
- Configurable por medio de temas (themes) y plugins
- Salida XHTM 1.0 personalizable
- Soporte multilenguaje
- Soporta los navegadores: Mozilla, MSIE, FireFox, Opera y Safari (experimentalmente)

[6] Java. - Información sobre la tecnología web, en concreto JavaScript. Disponible en: http://www.desarrolloweb.com/javascript/

- Compresión PHP/.NET/JSP/Coldfusion GZip, haciendo de TinyMCE un 75% más pequeño
- permite usar Ajax para guardar y cargar contenido

Free Editor: Es un ambicioso programa que servirá para abrir y editar prácticamente cualquier documento e imagen.

En concreto, Free Editor abre documentos de Microsoft Office (DOCX, XLSX, PPTX), documentos genéricos (PDF, TXT, ODT), imágenes (RAW, JPG, PSD), archivos de Windows (EXE, OCX, DLL) y de código (JSP, CSS, SQL, ASP).

Dependiendo del tipo de archivo, Free Editor ofrecerá una u otra herramienta para editarlo. En el caso de las imágenes, se puede girarlas, cambiar su tamaño, recortarlas o aplicar cambios de color o luz. Con los documentos de texto, podrás cambiar el formato de texto y con las hojas de cálculo tendrás acceso a las celdas. (I. Jacobson, 2010)

Free Editor muy útil para leer documentos sin necesidad de instalar varios programas. Con uno tendrás acceso a todos los archivos. Además, también servirá para cambiar el formato de un fichero.

Free Editor soporta los siguientes formatos de Archivos de código: JSP, CS, SQL, ASPX, VB, JAVA, CSS, KML, C, PHP, ASP, MHT.

LopeEdit: Es una solución muy acertada para todos aquellos que requieran de un editor con las máximas pretensiones posibles, pero con los mínimos recursos necesarios. LopeEdit es la solución a estos problemas, ya que con la misma sencillez que el Bloc de Notas hace cosas impensables para el mismo. (S. Brown, 2012)

Características más importantes de este editor de texto son:

- Puede abrir múltiples archivos, permitiendo cambiar entre ellos de forma fácil a través de pestañas
- Realiza un resaltado de los paréntesis, corchetes y llaves opuestos
- Realiza coloreado de sintaxis para cualquier lenguaje de programación. Por defecto incorpora los siguientes lenguajes: C/C++, Java, JavaScript, Visual Basic, VBScript, HTML, ASP, JSP, SQL, Cobol, C, CSS (Cascading Style Sheets),
- Pascal, Perl y PHP. Además, se puede configurar nuevos lenguajes y el color de cada parte del código (comentarios, palabras reservadas, cadenas, números, etc.)
- La búsqueda de texto es muy cómoda, pues se hace a través de dos barras que no estorban. Una barra, la barra de búsqueda rápida, ocupa muy poco espacio. La

otra barra, la barra de búsqueda avanzada, permite buscar y reemplazar texto con más opciones

- Existen otras dos barras (una para explorar carpetas y otra para explorar archivos) que permiten examinar el sistema de ficheros para abrir los archivos rápidamente. Además, la barra de archivos tiene también unos Favoritos para poder abrir rápidamente los archivos que se utilicen con mayor frecuencia
- Tiene otra barra que permite tener múltiples portapapeles
- Barra adicional con la tabla ASCII para insertar los caracteres que no contiene el teclado
- Se puede añadir marcadores a los archivos abiertos
- Estas y otras características hacen de LopeEdit un excelente editor para los que sólo requieran de sus más que suficientes opciones.

Cambios recientes: Correcciones, Mejoras en el manejo de UTF-8.

2.2.6. Versiones de JSP

Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1 publicada en abril de 2011 como una versión estable[7].

2.2.7. Composición de JSP

Un JSP es uno de los componentes más básicos para aplicaciones de Servidor en JAVA. Su composición consta de dos grandes partes: HTML y lenguaje JAVA. Mediante HTML se especifica el contenido estático de despliegue y es mediante fragmentos del lenguaje Java que se genera contenido dinámico en efecto cumpliendo la definición de aplicación de servidor.[8]

[7] Rincón, J.M.- Estudio de la evolución Web y Lenguajes Dinámicos. Proyecto de fin de carrera, Universidad Carlos III de Madrid, España. Disponible en: htpp://www.movilwe.com

8 Osmosis Latina. - JSP (Java Server Pages): Composición. Disponible: http://javaweb.osmosislatina.com/curso/jsp.htm

2.2.8. Filosofía de JSP

JSP se basa en la filosofía de Java que dice: "Escribe una vez, ejecuta donde quieras" y permite que un mismo código fuente corra de la misma manera independientemente del sistema operativo y el dispositivo donde se ponga en marcha. Según (J. García, 2008, pág. 130)

El lenguaje Java se creó con cinco objetivos principales:

- Debería usar la metodología de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

2.2.9. Estructura y elemento del Lenguaje JSP

El código fuente de una página JSP[9] incluye:

Directivas: Dan información global de la página, por ejemplo, importación de estamentos, página que maneja los errores o cuando la página forma parte de una sesión, en el ejemplo anterior informamos del tipo de script de Java.

Declaraciones: Sirven para declarar métodos y variables.

Scripts de JSP: Es el código Java embebido en la página.

Expresiones de JSP: Formatea las expresiones como cadenas para incluirlas en la página de salida.

Estos elementos siguen una sintaxis como XML, así se obtiene un significado con una presentación totalmente separada de la lógica. Un buen ejemplo es <jsp:useBean .../> el cual busca o crea una instancia de un bean. Con el mecanismo de extensiones de tag se tiene la posibilidad de definir tags con acciones similares y poner la funcionalidad en una librería de tags. Según (I. Jacobson, 2010)

[9] Eckel B. (2002). Piensa el Java: Estructura y elementos del lenguaje JSP. Segunda Edición. Madrid. Editorial: Prentice Hall.

2.2.9.1. Directivas

Una directiva de JSP es un estamento que proporciona la información del motor de JSP para la página que la pide. Su sintaxis general es <%@ directiva {atributo ="valor"} %> dónde la directiva debe tener un número de atributos. Cada directiva tiene un XML opción al equivalente, pero esto son intentos para una futura herramienta JSP.

Posibles directivas en JSP[10] son:

Page: Información para la página.

Include: Incluye archivos completos palabra por palabra.

Taglib: La dirección de la librería de tags que se usará en la página.

Atributos y posibles valores	Descripción
language="java"	Comunica al servidor el lenguaje que va a ser utilizado en el archivo. Java es el único posible es esta especificación
extends="package.class"	La variale extends, define la clase padre del servlet generado. Normalmente no es necesario utilizar otras que no sean las clases base del proveedor.
import="package.,package.class"*	Sirve para especificar los paquetes y clases que se quieran utilizar.
session="true|false"	Por defecto session vale true, manteniendo los datos de las sesión para la página.
isThreadSafe="true|false"	Por defecto vale *true*, le hace señales al motor de JSP para que multiples pedidos del cliente puedan ser tomadas como una.
info="text"	Información en la página a la que puede accederse a través del método Servlet.getServletInfo()
errorPage="pagina_error"	Página que manejará las excepciones de errores.
isErrorPage="true|false"	Marca a la página como la página que manejará

Figura 5. *Detalles de una Directiva JSP*
Fuente: *Adoptado del Libro Piensa en Java (2015)*

2.2.9.2. Declaraciones

Una declaración de JSP[11], puede definirse como una definición de variables y métodos a nivel de clase que son usadas en la página.

Un bloque de declaraciones típico sería <%! declaración %>

Un ejemplo de declaración de script sería el siguiente:

[10] *JPS. - Nuevas tecnologías en torno a sistemas de información basados en web, xml, jsp, asp y php.*
[11] Grupo Innova: Empresas dedicadas a la automatización de los procesos empresariales. Derecho de Internet, contratación Electrónica y firma Digital, 2010. Disponible http://www.innova.com/

```
<HTML>
<HEAD>
<TITLE>Página simple JSP</TITLE>
</HEAD>
<BODY>
<%! String strCadena = "x";
int intContador = 0;
%>
</BODY>
</HTML>
```

2.2.9.3. Scripts de JSP

Los Scripts son bloques de código Java residentes entre los tags <% y %>.

Este bloque de código estará dentro del servlets generado incluidos en método _jspService ().

Los Scripts pueden acceder a cualquier variable o Beans que haya sido declarado. También hay algunos objetos implícitos disponibles para los Scripts desde entorno del Servlet. Según (I. Jacobson, 2010)

Objetos implícitos	Descripción
request	Es la petición del cliente. Es normalmente una subclase de la case HttpServletRequest.
response	Es la página JSP de respuesta y es una subclase de HttpServletResponse.
pageContext	Los atributos de la página y los objetos implicitos necesitan ser accesibles a través de API, para permitir al motro de JSP compilar la página. Pero cada servidor tiene implementaciones específicas de cada uno de esos atributos y objetos. Para solucionar este problema, el motor de JSP utilizar la clase Factory para devolver la implementación de clase PageContext del servidor. Esta clase PageContext es inicializada con los objetos response y request y algunos atributos de la directiva de la página (erropage, session,buffer and autoflush) y facilita los otros objetos implícitos para la pagina de petición.Veremos más más adelante.
session	El objeto de sesión HTTP asociado a la petición.
application	Lo que devuelve el servlet cuando se llama a getServletConfig().getContext()
out	El objeto que representa la salida de texto por pantalla.
config	El objeto ServletConfig de la página.
page	Es la forma que tiene la página para referirse a si misma. Se usa como alternativa al objeto this
exception	Es una subclase libre de Throwable que es pasada a la página que majena los errores.

***Figura 6**. Objetos implícitos para un Script JSP*
***Fuente:** Libro Piensa en Java*
***Autor:** Bruce Eckel*

El siguiente fragmento de código muestra cómo obtener el valor de un parámetro mediante el objeto request, y como pasarlo a una cadena para mostrarlo en pantalla.

```
<%
String strNombre = request.getParameter("nombre");
out.println(strNombre);
%>
```

2.2.9.4. Expresiones de JSP

[12]Las expresiones son una magnifica herramienta para insertar código embebido dentro de la página HTML. Cualquier cosa que este entre los tags <%= y %> será evaluado, convertido a cadena y posteriormente mostrado en pantalla. La conversión desde el tipo inicial a String es manejada automáticamente. Según (Schmidt M. D., 2013.)

Es importante remarcar que la expresión no termina en punto y coma (;). Esto es así porque motor de JSP, pondrá la expresión automáticamente entre out.println().

Las expresiones JSP te permiten parametrizar las páginas HTML (es parecido a cuando parametrizas una consulta SQL, pero difieren la forma de los valores). Una y otra vez, en el código de la página HTML, ser verá bucles o condiciones usando código Java, simplemente empezando y acabando las condiciones o bucles entre los tags <% y %>. Un ejemplo sería:

```
<% for (int i=0;i<5;i++) { %>
<BR>El valor del contador es <%=i%>
<% } %>
```

2.2.9.5. Variables

Los identificadores son conjuntos de letras y/o números que se utilizan para simbolizar todos los elementos que en un programa son definibles por el usuario (programador o ingeniero de software) del mismo como son las variables donde se almacenan datos, funciones (pequeños módulos con código), etiquetas, clases, objetos, etc. Según (Navenn, 2008, pág. 55)

[12] García M. (2000). Tutorial JSP Java Server Page: Expresiones JSP. Editorial: España: Editorial: Java Hispano.

En Java JSP una variable se define como un identificador que se utiliza para almacenar todos los datos generados durante la ejecución de un programa.

Existen ciertas reglas en cuanto a variables:

- Claras y con referencia directa al problema.
- No espacios en blanco, ni símbolos extraños en ellas.
- Se pueden usar abreviaturas, pero solo de carácter general.
- No deben ser palabras reservadas del lenguaje.

Ejemplos de buenas variables:

Nombre, Edad, SdoDiario, IngMensual, Perímetro, Calif1, etc.

2.2.9.6. Tipos de datos

En Java JSP a toda variable que se use en un programa, se le debe asociar (generalmente al principio del programa) un tipo de dato específico. Un tipo de dato define todo el posible rango de valores que una variable puede tomar al momento de la ejecución del programa y a lo largo de toda vida útil del propio programa. Según (I. Jacobson, 2010, pág. 234)

Los tipos de datos más comunes en Java jsp son:

Tabla 1. Tipos de datos primitivos

Palabra	Descripción	Tamaño/Formato
INTEGER		
byte	Byte – length integer	8 – bit
short	Short integer	16- bit
int	Integer	32- bit
Long	Long integer	64 bit
REAL NUMBERS		
Float	Single – precisión floating point	32 bit IEEE 754
Doublé	Double – precisión floating point	64 bit IEEE 754
OTHER TYPE		
Char	A single carácter	16 bit Unicode carácter
boolean	A boolean value	True or false

Autor: Bruce Eckel

Como se observa es muy similar a las de c o c++.

Para el caso de Sting se deberá usar la Clase String que tiene dos constructores, de momento se entiende de la siguiente manera;

String nombre= new String();

Se crea la string y se inicializa con un dato y valor.

2.2.9.7. Operadores aritméticos

Un operador es un símbolo especial que indica al compilador que debe efectuar una operación matemática y lógica. Java JSP reconoce los siguientes operadores aritméticos.

Tabla 2. Operadores aritméticos

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
%	Residuo o Modulo

Autor: Bruce Eckel

Nota: En problemas de división entre enteros, JAVA trunca la parte residual, ejemplo:

Desplegar: 13/5 -> el resultado es 2.

2.2.9.8. Operadores relacionales

Los operadores relacionales que reconocen java jsp son:

Tabla 3. Operadores Relacionales

Operador	Operación
==	Igual que
>	Mayor que
<	Menor que
>=	Mayor o igual que
< =	Menor o igual que
¡=	No es igual que o que es diferente.

Autor: Bruce Eckel

2.2.9.9. Comentarios

Existen dos tipos de comentarios que pueden ser utilizados según la finalidad deseada:

Comentarios HTML: son enviados al terminal cliente como parte de la página HTML, siendo su sintaxis:

<!-- comentario -->

Comentarios ocultos: se utilizan para documentar el código que se va escribiendo y no se envían a los navegadores de los clientes, sirviendo de utilidad al desarrollador de la página JSP. Se escriben conforme a la siguiente sintaxis:

<%-- comentario %>

2.2.9.10. Tipos de datos especiales

Cuando se necesita trabajar sobre datos de diversos tipos, en este caso ni variables escalares ni arreglos son los adecuados. Para resolver estos problemas los lenguajes de programación proveen un tipo de dato especial llamado Registros. Según (J. García, 2008)

Un registro es una variable especial que tiene la capacidad de almacenar datos de diferentes tipos.

Sin embargo, JAVA usa en su lugar una CLASE. Este método tiene la ventaja de que además de incluir los campos tradicionales de un registro, también puede incorporar una serie de métodos que permiten procesar de manera más fácil los campos o atributos clase. Según (Edelson, 2006)

2.2.9.11. Estructuras de control de flujo

Permiten alterar la secuencia normal de ejecución de un programa. Estas instrucciones se dividen en tres grandes categorías:

Instrucciones condicionales que en JAVA JSP se implementan con las instrucciones if y switch.

Instrucciones de ciclos con: for, while, do while

En java jsp muchas de ellas tienen sus correspondientes componentes visuales, derivados de html. Según (Schmidt M. K., 2005)

2.2.9.12. Estructuras de control de flujo condicionales

Una de las más poderosas características de cualquier computador es la capacidad que tiene de tomar decisiones. Es decir, al comparar dos alternativas deferentes el computador

puede tomar una decisión basándose en la evaluación que hace de alguna condición. (I. Jacobson, 2010, págs. 190,191)

En java jsp algunos ejemplos de instrucciones condicionales[13]:

a) Instrucción IF: Es la instrucción condicional más usada en los diversos lenguajes de programación, su formato completo y de trabajo en java jsp es:

Cargar o asignar la variable de condición;

If (condición)
{grupo de ciertas instrucciones;}
else
{grupo falso de instrucciones;}

Primero: Observar donde van y donde no van los puntos y comas;

Segundo: La condición va entre paréntesis;

Tercero: S un if no ocupa un grupi falso de instrucciones entonces no se pone el else y la llave antes del else terminaría con punto y coma.

Además se debe tener en cuenta que hay como tener condiciones compuestas, que es la unión de dos o más condiciones simples unidas por los llamados operadores lógicos.

b) Instrucción Switch: Existen ocasiones o programas donde se exige evaluar muchas condiciones a la vez, en estos casos se usa una condición compuesta muy grande o usando la instrucción switch.

Esta es una instrucción de decisión múltiple donde el compilador prueba o busca el valor contenido en una variable ENTERA o CHARACTER contra una lista de constantes apropiadas, es decir enteras, carácter, cuando el computador encuentra el valor de igualdad entre variable y constante entonces ejecuta el grupo de instrucciones asociados a dicha constante, si no encuentra el valor de igualdad entre variable y constante, entonces ejecuta un grupo de instrucciones asociados a un default aunque este último es opcional.

[13] JSP. - es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas web con programación en Java. Disponible en: http://www.desarrolloweb.com/articulos/831.php

El formato de esta instrucción es el siguiente:

Capturar o asignar variable de condición;

```
Switch (var OPCION)
  {
case const1: instrucción (es);
        break;
case const2: instrucción (es);
        break;
case const3: instrucción (es);
        break;
  default: instrucción (es);
  }
```

c) Control select: Existen muchas ocasiones en donde en donde el usuario del programa tiene que proporcionar datos que provienen de un conjunto finito y muy pequeño de posibles respuestas esto significa que cada vez que se ejecute el programa el usuario estará proporcionando las mismas respuestas. Por ejemplo, sexo siempre será hombre, mujer. Por situaciones como esta existen componentes html[14] que permiten programar por adelantado las posibles respuestas y el usuario solo debe seleccionar la respuesta apropiada en lugar de tener que escribirla.

Este control SELECT permite definir en primera instancia un conjunto de datos o valores respuestas asociados a una caja de edición cualesquiera así el usuario tiene la oportunidad de seleccionar un dato del conjunto de datos o respuestas ya definido. (J. García, 2008)

Este componente SELECT deberá construirse en dos partes una parte de encabezado para poner el nombre del grupo de respuestas. La segunda parte es la lista de opciones o respuestas que se debe cargar al tiempo de ejecución de la forma html. (I. Jacobson, 2010, págs. 120-125)

d) Checkbox y Exepciones: El componente CheckBox, permite seleccionar una opción al usuario del programa o tomar una decisión directamente en pantalla.

[14] JSP. - es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas web con programación en Java. Disponible en: http://www.desarrolloweb.com/articulos/831.php

Figura 7. Estructura de control Checbox
Fuente: Libro Tutorial de JSP

Se puede seleccionar dos o más checkbox.

e) **Componente Radiobutton:** Se utiliza para presentar al usuario un conjunto de opciones mutuamente excluyentes entre sí, es decir si el usuario selecciona un componente radio todos los demás componentes radiobutton en la forma, se deselecciona solos es por esta razón que se afirma que los radiobutton son excluyentes. (Schmidt M. K., 2005, pág. 223)

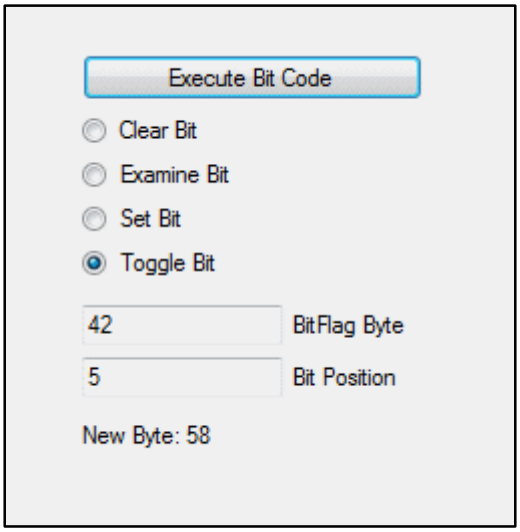


Figura 8. Estructura de control Radiobutton
Fuente: Libro Tutorial de JSP

f) **Ciclo For:** Instrucciones para ciclos resuelven el problema de repetir todo el programa o cierta parte del programa más de una vez. Este ciclo es uno de los más usados para repetir una secuencia de instrucciones sobre todo cuando se conoce la cantidad exacta de veces que se quiere que se ejecute una instrucción simple o compuesta.

Su formato general es:

for (inicialización; condición; incremento)
{Instrucción (es);};

En su forma simple la inicialización es una instrucción de asignación que carga una variable de control de ciclo con un valor inicial. La condición es una expresión relacional que evalúa la parada que determina cuando debe acabar el ciclo. El incremento define la manera en que la variable de control de ciclo debe cambiar cada vez que el computador repite un ciclo. Se debe separar esos 3 argumentos con punto y coma (;).

g) Control Ciclo While: En este ciclo el cuerpo de instrucciones se ejecuta mientras una condición permanezca como verdadera en el momento en que la condición se convierte en falsa el ciclo termina.

Su formato general es:

Cargar o inicializar variable de condición;

while (condición)
{
Grupo cierto de instrucciones;
Instrucción (es) para salir del ciclo;
};

Un error común con el while, es poner un punto y coma (;) después de la (condición) ejemplo while (condición); <= **esto es un error.**

h) Ciclo do while: Su diferencia básica con el ciclo while es que la prueba de condición es echa al finalizar el ciclo, es decir las instrucciones se ejecutan cuando menos una vez porque primero ejecuta las instrucciones y al final evalúa la condición. También se le conoce por esta razón como ciclo de condición salida.

Su formato general es: Cargar o inicializar variable de condición;

do {
grupo cierto de instrucción(es);
instrucción (es) de rompimiento de cuclo;
} while (condición);

2.2.9.13. Arreglos

Uno de los problemas más comunes en los diversos sistemas de información es el tratamiento o procesamiento de un gran volumen de datos o de información. Las variables usadas comúnmente son variables escalares, porque solo permiten almacenar o procesar un dato a la vez. En problemas que exigen manejar mucha información o datos a la vez, variables escalares no son suficientes ya que su principal problema es que solo permiten almacenar y procesar un dato a la vez. Variables tipo arreglo si permiten almacenar y procesar conjuntos de datos del mismo tipo a la vez. Según (Navenn, 2008, págs. 178-180)

Cada dato dentro del arreglo se le conoce como elemento del arreglo y se simboliza y procesa (captura, operación, despliegue) usando el nombre del arreglo respectivo y un subíndice indicando la posición relativa del elemento con respecto a los demás elementos del arreglo, en Java JSP[15] la primera posición, elemento o renglón es el creo (0).

Tipo de listas: Un arreglo tipo lista se define como una variable que permite almacenar un conjunto de datos del mismo tipo organizados en una sola columna y uno o más renglones. También recibe el nombre de vectores en algebra o arreglos unidimensionales en programación.

Los procesos normales con una lista o con sus elementos, incluyen declarar toda la lista, capturar sus elementos, desplegarlos, realizar operaciones con ellos, etc.

Para declara una lista en Java JSP se usa el siguiente formato:

tipodato nomlista[]= new tipodato[cant elementos]; En java jsp no existen arreglos o lista tradicionales lo que existe es un objeto, por tanto se deberá usar el operador new antes de empezar a procesar el arreglo, las ventajas son:

- Estas listas pueden usar el método length para conocer el tamaño de la misma.
- También se pueden crear listas con tamaño fijo o inicializado o cargadas.

Arreglos bidimensionales tipo tabla: Un arreglo tipo tabla se define como un conjunto de datos del mismo tipo organizados en dos o más columnas y uno o más renglones.

Para procesar internamente todos los elementos de la tabla se ocupan dos ciclos for(), uno externo para controlar renglón y otro interno para controlar columna. Los elementos de

[15] JSP. - es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas web con programación en Java. Disponible en: http://www.desarrolloweb.com/articulos/831.php

una tabla se deben simbolizar con el nombre de la tabla y 2 subíndices, el primer subíndice referencia al renglón y el siguiente subíndice referencia a la columna, los dos dentro del mismo corchete.

La declaración de una tabla en JSP sigue los siguientes formatos:

- tipo dato nomtabla[][] = new tipodato[reng][col];
- Clasenumerica objetotabla[][] = new constructor[ren][col];

2.2.9.14. Funciones definidas por el usuario

En Java JSP[16] una función es un módulo de un programa separado del cuerpo principal, que realiza una tarea específica y que puede regresar un valor a la parte principal del programa u otra función o procedimiento que la invoque.

La forma general de una función es:

<%! Tipodatoregresa Nom_Fun(parámetros)
{cuerpo de instrucciones ;
Instrucción return;
}; %>

El tipo especifica el tipo de valor que la función regresara utilizando la instrucción return. Si no se especifica un tipo se asume de default que el tipo regresado es int. La lista de parámetros formales es una lista de variables separadas por comas (,) que almacenaran los valores que reciba la función, estas variables actúan como locales dentro del cuerpo de la función. Aunque no se ocupen parámetros los paréntesis son requeridos. La declaración de parámetros es la especificación de cada tipo de parámetro recibido.

Instrucción return: Dentro del cuerpo de la función debe haber una instrucción return cuando menos, para regresar el valor, esta instrucción permite regresar datos.

2.2.9.15. Métodos, paquetes

Métodos en Java: Un método en Java es un conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre. Algunos de los métodos más utilizado hasta ahora: Math.pow(), Math.sqrt(), Character.isDigit(), System.out.println();

[16] JSP. - es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas web con programación en Java. Disponible en: http://www.desarrolloweb.com/articulos/831.php

Cuando se llama a un método, la ejecución del programa pasa al método y cuando éste acaba, la ejecución continúa a partir del punto donde se produjo la llamada.

Utilizando métodos:

- Se puede construir programas modulares.
- Se consigue la reutilización de código. En lugar de escribir el mismo código repetido cuando se necesite, por ejemplo, para validar una fecha, se hace una llamada al método que lo realiza.

En Java un método siempre pertenece a una clase. Todo programa java tiene un método llamado main. Este método es el punto de entrada al programa y también el punto de salida.

1) Estructura general de un método java. La estructura general de un método Java es la siguiente:

[especificadores] tipoDevuelto nombreMetodo([lista parámetros]) [throws listaExcepciones]
{
// instrucciones
[return valor;]
}

Los elementos que aparecen entre corchetes son opcionales.

Especificadores (opcional): determinan el tipo de acceso al método.

TipoDevuelto: indica el tipo del valor que devuelve el método. En Java es imprescindible que en la declaración de un método, se indique el tipo de dato que ha de devolver. El dato se devuelve mediante la instrucción return. Si el método no devuelve ningún valor este tipo será void.

nombreMetodo: es el nombre que se le da al método. Para crearlo hay que seguir las mismas normas que para crear nombres de variables.

Lista de parámetros (opcional): después del nombre del método y siempre entre paréntesis puede aparecer una lista de parámetros (también llamados argumentos) separados por comas. Estos parámetros son los datos de entrada que recibe el método para operar con ellos. Un método puede recibir cero o más argumentos. Se debe especificar

para cada argumento su tipo. Los paréntesis son obligatorios, aunque estén vacíos. Según (Edelson, 2006, págs. 156-160)

throws listaExcepciones (opcional): indica las excepciones que puede generar y manipular el método.

return: se utiliza para devolver un valor. La palabra clave return va seguida de una expresión que será evaluada para saber el valor de retorno. Esta expresión puede ser compleja o puede ser simplemente el nombre de un objeto, una variable de tipo primitivo o una constante.

El tipo del valor de retorno debe coincidir con el tipoDevuelto que se ha indicado en la declaración del método.

Si el método no devuelve nada (tipoDevuelto = void) la instrucción return es opcional.
Un método puede devolver un tipo primitivo, un array, un String o un objeto.
Un método tiene un único punto de inicio, representado por la llave de inicio {. La ejecución de un método termina cuando se llega a la llave final } o cuando se ejecuta la instrucción return. Según (I. Jacobson, 2010, pág. 120)

La instrucción return puede aparecer en cualquier lugar dentro del método, no tiene que estar necesariamente al final.

2) Implementación de métodos en Java: Pasos para implementar un método:

a) Describir lo que el método debe hacer
b) Determinar las entradas del método
c) Determinar los tipos de las entradas
d) Determinar el tipo del valor retornado
e) Escribir las instrucciones que forman el cuerpo del método
f) Prueba del método: diseñar distintos casos de prueba

Paquetes: Los paquetes son el mecanismo por el que Java permite agrupar clases, interfaces, excepciones y constantes. De esta forma, se agrupan conjuntos de estructuras[17] de datos y de clases con algún tipo de relación en común.

Con la idea de mantener la reutilización y facilidad de uso de los paquetes desarrollados es conveniente que las clases e interfaces contenidas en los mismos tengan cierta relación

[17] J & Innovación. - *Entornos de desarrollo de aplicaciones web integrados. Disponible en: http://unillanosgsw.wikispaces.com/J%26H-INNOVATIONS*

funcional. De esta manera los desarrolladores ya tendrán una idea de lo que están buscando y fácilmente sabrán qué pueden encontrar dentro de un paquete.

1) Creación de un paquete

Declaración: Para declarar un paquete se utiliza la sentencia *package* seguida del nombre del paquete que estemos creando:

package NombrePaquete;

La estructura que ha de seguir un fichero fuente en Java es:

- Una única sentencia de paquete (opcional).
- Las sentencias de importación deseadas (opcional).
- La declaración de una (y sólo una) clase pública (*public*).
- Las clases privadas del paquete (opcional).

Por lo tanto, la sentencia de declaración de paquete ha de ser la primera en un archivo fuente Java.

Nomenclatura: Para que los nombres de paquete puedan ser fácilmente reutilizados en toda una compañía o incluso en todo el mundo es conveniente darles nombres únicos. Esto puede ser una tarea realmente tediosa dentro de una gran empresa, y absolutamente imposible dentro de la comunidad de Internet. Según (Navenn, 2008, págs. 343-355)

Por eso se propone asignar como paquetes y subpaquete el nombre de dominio dentro de Internet. Se verá un ejemplo para un dominio que se llamase *japon.magic.com,* un nombre apropiado sería *com.magic.japon.paquete*.

Subpaquete: Cada paquete puede tener a su vez paquetes con contenidos parecidos, de forma que un programador probablemente estará interesado en organizar sus paquetes de forma jerárquica. Para eso se definen los *subpaquete.*

Para crear un subpaquete bastará con almacenar el paquete hijo en un directorio *Paquete/Subpaquete.*

Así una clase dentro de un *subpaquete* como *Paquete.Subpaquete. clase* estará codificada en el fichero *Paquete/Subpaquete.java*.

El JDK define una variable de entorno denominada *CLASSPATH* que gestiona las rutas en las que el JDK busca los subpaquete. El directorio actual suele estar siempre incluido en la variable de entorno *CLASSPATH*. Según (J. García, 2008)

2) Uso de un paquete

Con el fin de importar paquetes ya desarrollados se utiliza la sentencia *import* seguida del nombre de paquete o paquetes a importar. Se pueden importar todos los elementos de un paquete o sólo algunos. Para importar todas las clases e interfaces de un paquete se utiliza el metacaracter *:

import PaquetePrueba.;*

También existe la posibilidad de que se deseen importar sólo algunas de las clases de un cierto paquete o subpaquete:

import Paquete.Subpaquete1.Subpaquete2.Clase1;

Para acceder a los elementos de un paquete, no es necesario importar explícitamente el paquete en que aparecen, sino que basta con referenciar el elemento tras una especificación completa de la ruta de paquetes y subpaquetes en que se encuentra.

Paquete.Subpaquetes1.Subpaquete2.Clase_o_Interfaz.elemento

En la API de Java se incluyen un conjunto de paquetes ya desarrollados que se pueden incluir en cualquier aplicación (o *applet*) Java que se desarrolle

3) Ámbito de los elementos de un paquete

Al introducir el concepto de paquete, surge la duda de cómo proteger los elementos de una clase, qué visibilidad presentan respecto al resto de elementos del paquete, respecto a los de otros paquetes. Por defecto se considera los elementos (clases, variables y métodos) de un mismo paquete como visibles entre ellos (supliendo las denominadas *clases amigas* de C++).

Tabla 4. Ámbito de un Paquete en JAVA

Situación del elemento	*private*	*sin modificador*	*protected*	*public*
En la misma clase	Sí	Sí	Sí	Sí

En una clase en el mismo paquete	No	Sí	Sí	Sí
En una clase hija en otro paquete	No	No	Sí	Sí
En una clase no hija en otro paquete	No	No	No	Sí

Fuente: Libro Tutorial de JSP

2.2.9.16. Programación Orientación a objetos

Declaraciones o procedimientos JAVA JSP: un camino para dividir un programa en partes más pequeñas es el uso de declaraciones.

Una declaración es un grupo de instrucciones, variables, constantes, etc., que están diseñados con un propósito particular y tiene su nombre propio. Es decir una declaración es un módulo de un programa que realiza una tarea específica y que no puede regresar valores al programa principal u a otro procedimiento que lo esté vinculado.

Después de escribir una declaración se usara ese nombre propio como una sola instrucción o llamada. Las declaraciones se construyen antes del programa, es decir en la parte declarativa. Su formato es:

<%! Void Nom_decl() {instrucciones;}%>

Un programa puede tener tantos procedimientos como se deseen, para hacer una llamada o invocación al procedimiento durante la ejecución de un programa se deberá escribir el nombre de la misma y los paréntesis en blanco. Según (J. García, 2008)

Parámetros JAVA JSP: un parámetro es una variable que puede pasar su valor a un procedimiento desde el principal o desde otro procedimiento. Existen ocasiones en que es necesario mandar al procedimiento ciertos valores para que los use en algún proceso. Estos valores que se pasan del cuerpo principal del programa al procedimiento se llaman parámetros. Una declaración completa es:

<%!
Void NomProc(lista parametros)
{cuerpo de instrucciones;};
%>

Reglas para el uso de parámetros:

- Cuando se usan variables con parámetros, la variable que se manda debe ser declarada dentro del principal o del procedimiento de donde se está enviando.

- La variable que se mande tiene un nombre, la que se recibe puede tener otro nombre.
- La cantidad de variables que se envíen deben ser igual en cantidad, orden y tipo a las variables que reciben.
- La variable que se recibe tiene un ámbito local dentro del procedimiento, es decir solo la puede usar ese procedimiento.

Variables globales y locales JAVA JSP: En java jsp el lugar donde se declara una variable afectara el uso que el programa quiera hacer de esa variable.

Las reglas básicas que determinan como una variable puede ser usada dependen de 3 lugares donde se puede declarar una variable.

En primer lugar es dentro de cualquier función o procedimiento incluyendo el principal. A estas se les llama variables locales y solo pueden ser usadas por instrucciones que estén dentro de esa función o procedimiento.

En segundo lugar es como parámetro de una función o procedimiento, donde después de haber recibido el valor, podrá actuar como variable local en esa función o procedimiento.

En esencia una variable local solo es conocida por el código de esa función o procedimiento y es desconocida por otras funciones o procedimientos. Según (Schmidt M. D., 2013.)

En tercer lugar es fuera de todas las funciones pero dentro de la parte declarativa <%!...%>, a este tipo de variables de les llama variables globales y podrán ser usadas por cualquier función.

2.2.9.17. Archivos

Cuando se termina de ejecutar un programa, ante esta situación nace el concepto de archivos en JSP, que son medios que facilita el lenguaje[18] para almacenar los datos en forma permanente normalmente en los dispositivos de almacenamiento estándar.

Operaciones básicas con archivos

[18] Java. - Información sobre la tecnología web, en concreto JavaScript. Disponible en: http://www.desarrolloweb.com/javascript/

- **Escribir o grabar:** es la operación más elemental con un archivo consiste en tomar un o unos datos en variables de cualquier tipo y almacenarlas en un archivo de datos en disco.
- **Leer:** Operación que consiste en sacar los datos del archivo en disco y mandarlos a cargar en la variable respectiva.

Organización de archivos:

En general existen dos tipos de archivos (secuenciales y directos).

- **Archivos secuenciales:** En este caso los datos se almacenan en forma consecutiva y no es recorrer o accesar los n – 1 registros anteriores.

  Dentro de un **archivo secuencial plano** existen operaciones asociadas a archivos, las más elementales son:

  Creación de archivo, crear un archivo nuevo en disco.

  Apertura de archivo, abrir un archivo existente en disco.

  Cierre de archivo, operación más importante en cualquier programa que maneje archivos, cierra el archivo como ultima instrucción.

  Altas en archivo, en este proceso se carga una clase en memoria con sus datos pertinentes y se graba la clase en el archivo en disco.

  Lectura de archivo, se abre el archivo y se manda el registro de disco a una clase en memoria para su procesamiento.

  Consulta de archivos, se pretende desplegar todos los registros del archivo en disco a la pantalla.

  Búsqueda en archivos, proporciona información del algún renglón del disco al usuario, generalmente se basa en el campo clave.

  Filtros, en base a una condición conocer los elementos que pertenecen.

  Modificaciones, cuando los datos una vez que son guardados deben cambiar.

  Bajas de registros, es muy común este proceso.

- **Archivos Directos o random:** Para este caso si se puede acceder o leer un renglón n cualquiera.

Almacenamiento en archivos: existen 2 métodos.

- **Modo texto:** En este caso los datos son almacenados usando código ascii y por tanto son plenamente visibles usando cualquier editor.

- **Método binario:** En este caso los datos son almacenados en notación hexadecimal y por tanto se ocupa un editor binario es más compacto que un archivo texto.

2.3. RUBY

2.3.1. Definición

Creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995.

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos, Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia según (Navenn, 2008, págs. 4, 55) de software libre. Ruby es un lenguaje multipropósito que permite desarrollos en las siguientes áreas: aplicaciones comerciales, accesos a bases de datos, procesos de transformación de XML, aplicaciones distribuidas, aplicaciones web[19].

2.3.2. Características de Ruby

- Ruby es un lenguaje de scripts, moderno y orientado a objetos que combina una importante flexibilidad con alta productividad.
- Incorpora algunas de las mejores características de otros lenguajes como SmallTalk, Java y Perl.
- Su alcance parece ilimitado y hoy se encuentra presente en aplicaciones que van desde el desarrollo web hasta la simulación de ambientes complejos.
- Es un lenguaje multiplataforma que se integra perfectamente en gran cantidad de arquitecturas; puede correr, incluso en dispositivos móviles.
- Promueve las mejores prácticas de programación sin perder usabilidad.
- Mediante su uso se pueden complementar las características de la lógica imperativa con la lógica funcional.
- Es altamente extensible no solo mediante librerías escritas en Ruby, sino que se puede ampliar utilizando lenguaje C y actualmente de forma experimental otros lenguajes.

[19] SFML es una biblioteca de desarrollo multimedia para el lenguaje C++ de la que ya hemos hablado en Genbeta Dev. Disponible en: http://www.genbetadev.com/

- Posee una filosofía real de trabajo, que propone algunas prácticas particulares como **DRY (**Don’t repeat yourself, en español: No te repitas**),** entre otras.
- Simplifica declaraciones, estructuras y modelos sin perder potencia y permite que el programador se desarrolle de forma adecuada.
- Es un lenguaje interpretado y dinámico.
- Permite utilizar la más simple expresión para un programa o algoritmo; esto es sumando a las cuales prácticas ágiles permite desarrollar en forma amigable.

2.3.3. Características del lenguaje

- **En Ruby todo es un objeto:** esto básicamente quiere decir que, desde el más simple carácter hasta un conjunto de instrucciones, son instancias de clases y serán manipuladas como tales. Este concepto anula lo que normalmente denominamos tipos primitivos, ya que hasta el más trivial de los datos es un objeto. Según (Navenn, 2008, págs. 4,6,3,22)
- **La gran flexibilidad:** la gran flexibilidad de Ruby permite que se pueda incorporar funcionalidad en sus clases base y sus métodos. Es decir, podemos modificar absolutamente todo dentro del ambiente.
- En el lenguaje, todo tiene un valor, aunque sea nil.
- Se debe saber que, en principio, no existen diferencias entre comandos y expresiones dentro del entorno de programación.
- Ruby solo utiliza herencia simple. Esta característica habitual en muchos lenguajes facilita el trabajo de estructuras jerárquicas. Sin embargo, incorpora técnicas para poder imitar el comportamiento de la herencia múltiple de manera sencilla.
- Ruby utiliza un recolector de basura de alto nivel, por lo tanto, libera al desarrollador de estas tareas, en algunos casos triviales.
- No es de tipo estricto y no requiere declaración de variables.
- Ruby permite la programación con múltiples hilos de forma independiente al sistema operativo.

2.3.4. Herramientas para desarrollar en Ruby

2.3.4.1. IDES para Ruby

Geany: Geany es un editor de texto ligero basado en Scintilla con características básicas de entorno de desarrollo integrado (IDE). Está disponible para distintos sistemas operativos, como GNU/Linux, Mac OS X, BSD, Solaris y Microsoft Windows.

Figura 9. IDE Geany
Fuente: Adoptado del sitio de Geany. (2013)

Es distribuido como software libre bajo la Licencia Pública General de GNU. Tiene soporte para muchos lenguajes de programación distintos, como C, C++, C#, Java, JavaScript, PHP, HTML, CSS, Python, Perl, Ruby, Fortran, Pascal y Haskell. Algunas de las características más destacadas de Geany son: autocompletado, soporte multidocumento, soporte de proyectos, coloreado de sintaxis y emulador de terminal incrustado. Según (Navenn, 2008)

Bluefish: Es un editor dirigido al desarrollo web, bajo licencia GPL y se enfoca en la edición de páginas dinámicas e interactivas; sus principales características son: Soporte a multiproceso para archivos remotos mediante GVFs (sistema virtual de archivos de gnome) dependiendo de la configuración definida (acceso a FTP, SFTP, HTTP, HTTPS, WebDAv, CIFS).

Figura 10. IDE Bluefish
Fuente: Entornos de desarrollo de aplicaciones web integrados, Bluefish (2013)

Soporte a los siguientes lenguajes de programación: ADA, ASP.net y VBS, C/C++, CSS, CFLM, CLOJURE, HTML, XHTML, HTML 5, JAVA y JSP, JAVA SCRIPT Y

JQUERY, LUA, OCTAVE/MATLAB, PASCAL, PERL, PHP, PHYTON, RUBY, SHELL, SQL, SCHEME, VALA, XML.

Múltiples codificaciones de apoyo. Bluefish trabaja internamente con UTF8, pero puede permite almacenar los documentos usando cualquier otra codificación.

Integración de programas externos tales como: pelusa, weblint, xmllint, ordenado, javac, o cualquier programa o secuencia de comandos propios para manejar el procesamiento de texto avanzado o la detección de errores.

Haptana Studio: Aptana Studio es un entorno de desarrollo integrado gratuito basado en eclipse y desarrollado por Aptana, Inc., que puede funcionar bajo Windows, Mac y Linux y provee soporte para lenguajes como: Php, Python, Ruby, CSS, Ajax, HTML y Adobe AIR. Tiene la posibilidad de incluir complementos para nuevos lenguajes y funcionalidades.

Figura 11. IDE Haptana Studio
Fuente: Entornos de desarrollo de aplicaciones web integrados, Aptana Studio

Sus características principales son:

- Asistente de código para HTML y Javascript.
- Librerías ajax (jQuery, prototype, scriptaculous, Ext JS, dojo, YUI y Spry entre otras).
- Conexion vía FTP, SFTP, FTPS y Aptana Cloud.
- Herramientas para trabajo con base de datos.
- Marcado de sintaxis mediante colores.
- Compatible con extensiones para Eclipse (existen más de 1000).
- Aptana Studio es un IDE de software libre que posee la GNU General Public License o la Aptana Public License.

RubyMine: Si bien no es gratuito (salvo que sea usado para proyectos de software libre), su precio no es descomunal, partiendo de 58 euros la versión para un desarrollador individual. A cambio de este dinero, ofrece un entorno con un editor que permite el autocompletado de código, herramientas de ayuda para la refactorización, análisis de código al vuelo, visor del esquema de los modelos, soporte para Bundler o RVM, depurador, soporte para diversos sistemas de control de versione. En definitiva, toda una

navaja suiza. Está disponible para Windows, Mac OS X y Linux. Según (Schmidt M. D., 2013.)

Entre otros IDES ya conocidos también están: Eclipse, NetBeans,

2.3.4.2. Editor para Ruby

Araneae Text Editor: Araneae Text Editor es un editor de programación, de aspecto y funciones básicas, que incluye todo aquello que se necesite para crear código.

Con la simplicidad gráfica y de uso como principal característica, este editor te permite trabajar en código fuente sin temor. Además, todo desde un entorno de desarrollo limpio de características innecesarias.

Araneae Text Editor incorpora útiles funciones en un práctico menú. Estas van desde el uso de código de color en la sintaxis de comandos, uso de líneas numeradas para mayor facilidad y la interesante posibilidad de ver los resultados sin haber guardado los cambios.

Además, para los programadores más activos, Araneae es capaz de abrir y trabajar con varios documentos simultáneamente. Según (Schmidt M. K., 2005)

Araneae Text Editor soporta los siguientes formatos: HTML, XHTML, CSS, JavaScript, PHP, Ruby y más.

Komodo Edit: **(**Editor multilenguaje para programadores web**)** Komodo Edit es un editor multilenguaje especialmente pensado para programadores de sitios o aplicaciones web. Con él, se puede programar con total libertad dejando que el editor, por sí solo, se encargue de diferenciar por colores los comandos de cada una de tus líneas. Además, incorpora un sistema de auto completado para mejorar la velocidad de tu escritura.

Incluye un validador de sintaxis, funciones para trabajar con pestañas, clasificar proyectos e incluye las herramientas de edición necesarias para buscar, reemplazar, cortar y pegar cadenas de caracteres.

Como lenguajes, es capaz de identificar y colorear comandos de Ajax, CSS, HTML, JavaScripty XML; Perl, PHP, Python, Ruby y Tcl.

Ruby: (El lenguaje de programación más amistoso**)** Ruby es un lenguaje de programación orientado a objetos e interpretado, como Java o Python. Su filosofía de diseño, sin embargo, es radicalmente distinta.

Está pensado para ser un lenguaje "amistoso", agradable de usar y centrado en el programador. En Ruby todo es un objeto, y las funciones son a la vez métodos. Esto permite realizar código intuitivo, favoreciendo la legibilidad a largo plazo. La facilidad para aprender su sintaxis es notable, y al no necesitar compilación el código puede ejecutarse en varios sistemas operativos. Según (J. García, 2008)

Según (Edelson, 2006). Con una poderosa comunidad de usuarios (RubyForge) y un gestor de paquetes de librerías (RubyGems), Ruby va cobrando protagonismo como un lenguaje válido para múltiples escenarios, incluidas las aplicaciones web mediante Ruby-on-Rails.

El instalador oficial contiene el editor de código SciTE y abundante documentación, todo lo necesario para dar los primeros pasos con este magnífico lenguaje.

Cambios recientes: Nuevo núcleo de lenguaje, Nuevas librerías (como Onigmo), Mejoras en el soporte de depuración, Mejoras de rendimiento.

Ruby soporta los siguientes formatos: RB, TXT

2.3.5. Versiones de Ruby

La última versión estable de la rama 1.8 es la 1.8.7_p248, de la rama 1.9 es la 1.9.2_p180 . La versión en 1.9 que incorpora mejoras sustanciales en el rendimiento del lenguaje, que se espera queden reflejadas en la próxima versión estable de producción del lenguaje, Ruby 1.9.0.1 Diferencias en rendimiento entre la actual implementación de Ruby (1.8.6) y otros lenguajes de programación más arraigados han llevado al desarrollo de varias máquinas virtuales para Ruby. Entre ésas se encuentra JRuby, un intento de llevar Ruby a la plataforma Java, y Rubinius, un intérprete modelado basado en las máquinas virtuales de Smalltalk. Los principales desarrolladores han apoyado la máquina virtual proporcionada por el proyecto YARV, que se fusionó en el árbol de código fuente de Ruby el 31 de diciembre de 2006, y se dio a conocer como Ruby 1.9[20].

[20] Ruby. - Plataforma para el desarrollo de aplicaciones web y todo sistema referente a aplicación empresarial. Disponible en: http://es.wikipedia.org/wiki/Ruby

2.3.6. Composición de Ruby

En Ruby todos sus los elementos son objetos, clases, tipo de datos, funciones que son métodos que corresponden a un objeto. De igual manera las variables que constituyen ser parte de un programa en el lenguaje Ruby hacen referencia a objetos.

2.3.7. Filosofía de Ruby

Ruby está diseñado para la productividad y la diversión del desarrollador, siguiendo los principios de una buena interfaz de usuario. Sostiene que el diseño de sistemas necesita enfatizar las necesidades humanas más que las de la máquina[21]:

A menudo la gente, especialmente los ingenieros en computación, se centran en las máquinas. Ellos piensan, "Haciendo esto, la máquina funcionará más rápido. Haciendo esto, la máquina funcionará de manera más eficiente. Haciendo esto..." Están centrados en las máquinas, pero en realidad necesitamos centrarnos en las personas, en cómo hacen programas o cómo manejan las aplicaciones en los ordenadores. Nosotros somos los jefes. Ellos son los esclavos. Según (Alberto Espin, 2009)

Ruby sigue el "principio de la menor sorpresa", lo que significa que el lenguaje debe comportarse de tal manera que minimice la confusión de los usuarios experimentados. Matz ha dicho que su principal objetivo era hacer un lenguaje que le divirtiera a él mismo, minimizando el trabajo de programación y la posible confusión. Él ha dicho que no ha aplicado el principio de menor sorpresa al diseño de Ruby, pero sin embargo la frase se ha asociado al lenguaje de programación Ruby. La frase en sí misma ha sido fuente de controversia, ya que los no iniciados pueden tomarla como que las características de Ruby intentan ser similares a las características de otros lenguajes conocidos. En mayo de 2005 en una discusión en el grupo de noticias comp.lang.ruby, Matz trató de distanciar Ruby de la mencionada filosofía, explicando que cualquier elección de diseño será sorprendente para alguien, y que él usa un estándar personal de evaluación de la sorpresa. Si ese estándar personal se mantiene consistente habrá pocas sorpresas para aquellos familiarizados con el estándar.

Matz lo definió de esta manera en una entrevista:

[21] *Flanagan.-. Ruby Programming Language. Editorial: Oreilly & Assoc Inc.* .

Todo el mundo tiene un pasado personal. Alguien puede venir de Python, otro de Perl, y pueden verse sorprendidos por distintos aspectos del lenguaje. Entonces podrían decir 'Estoy sorprendido por esta característica del lenguaje, así que Ruby viola el principio de la menor sorpresa.' Espera, espera. El principio de la menor sorpresa no es sólo para ti. El principio de la menor sorpresa significa el principio de 'mi' menor sorpresa. Y significa el principio de la menor sorpresa después de que aprendes bien Ruby. Por ejemplo, fui programador de C++ antes de empezar a diseñar Ruby. Programé exclusivamente en C++ durante dos o tres años. Y después de dos años de programar en C++, todavía me sorprendía. Según (I. Jacobson, 2010)

Además, Ruby enfatiza en ser "lenguaje de guiones (scripts) para una programación orientada a objetos rápida y sencilla"[22].

Lenguaje de guiones interpretado:

- Posibilidad de realizar directamente llamadas al sistema operativo
- Potentes operaciones sobre cadenas de caracteres y expresiones regulares
- Retroalimentación inmediata durante el proceso de desarrollo

Rápido y sencillo:

- Son innecesarias las declaraciones de variables
- Las variables no tienen tipo
- La sintaxis es simple y consistente
- La gestión de la memoria es automática

Programación orientada a objetos:

- Todo es un objeto
- Clases, herencia, métodos
- Métodos singleton
- Mixins por módulos
- Iteradores y cierres

También:

- Enteros de precisión múltiple

[22]*Ruby. - Plataforma para el desarrollo de aplicaciones web y todo sistema referente a aplicación empresarial. Disponible en: http://es.wikipedia.org/wiki/Ruby.*

- Modelo de procesamiento de excepciones
- Carga dinámica
- Hilos

2.3.8. Estructura y elemento del Lenguaje

2.3.8.1. Variables

Las variables en Ruby[23] no necesitan declararse y no tienen tipo, puedes almacenar en ellas lo que sea. Las constantes deben empezar con una letra mayúscula y si intentas cambiar el valor de una constante durante la ejecución Ruby te lanzará un warning. Las variables globales empiezan con un símbolo del dólar y las constantes globales con un dólar seguido de una letra mayúscula. Algunos ejemplos de nombres de variables:

foo bar Pi $IVA $Precio

2.3.8.2. Tipos de datos

Números: Al principio he dicho que Ruby era un lenguaje orientado a objetos y ahora lo empezaremos a notar. Todos los tipos base en Ruby son objetos. Los números enteros pertenecen a la clase Fixnum si están comprendidos entre un rango 2−30 y 230 y Bignum para cualquier número entero fuera de ese rango. También se pueden utilizar diferentes bases de representación (binaria, octal, hexadecimal,) Los números reales pertenecen a la clase Float y también se pueden representar en notación científica. Según (Navenn, 2008, págs. 55,77,90)

```
irb> 0xFF #hexadecimal
=> 255
irb> 0b010 #binario
=> 2
irb> 020 #octal
=> 16
irb> 0.000125
=> 0.000125
irb> 1.25e-5
=> 1.25e-05
irb> 1.25e-3
=> 0.00125
irb> 0.00125
=> 0.00125
```

Figura 12. Notación Científica Tipo de datos Números
Fuente: Libro Guía del usuario de Ruby

[23] *Ruby. - Plataforma para el desarrollo de aplicaciones web y todo sistema referente a aplicación empresarial. Disponible en: http://es.wikipedia.org/wiki/Ruby*

Como todo objeto de una clase los números tienen sus propios métodos.

```
irb> 42.class
=> Fixnum
irb> -3.abs #valor absoluto
=> 3
irb> 4.succ #sucesor de 4
=> 5
irb> 3.1415.to_i
=> 3
```

Figura 13. Métodos de Un objeto tipo Número
Fuente: Libro Guía del usuario de Ruby

Cadenas: Una cadena es cualquier símbolo rodeado entre comillas dobles o simples.Las cadenas con comillas dobles permiten secuencias de escape, como "\n" o "\t", y la potente expresión de evaluación #{ }. Los operadores de suma y multiplicación permiten **concatenar** y **repetir** cadenas. Ruby tiene además el operador de **adición por el final** << para concatenar cadenas. Según (Edelson, 2006)

Podemos filetear una cadena con el **operador de corte [x..y]** donde x e y son números enteros. El operador devuelve una subcadena formada por los caracteres entre las posiciones x e y de la cadena. Combinado con el operador de asignación el operador de corte sirve para sustituir subcadenas de una cadena. Ejemplos:

```
irb> puts "\tRuby mola #{'mucho '*3}mas que Python"
        Ruby mola mucho mucho mucho mas que Python
=> nil
irb> cad1 = "chunky"
=> "chunky"
irb> cad1 += " bacon"
=> "chunky bacon"
irb> cad1 << " " + cad1
=> "chunky bacon chunky bacon"
irb> cad1[7..11]
=> "bacon"
irb> cad1[0..14] = "kevin"
=> "kevin"
irb> cad1
```

Figura 14. Operaciones en tipo de dato Cadena
Fuente: Libro Guía del usuario de Ruby

Las cadenas pertenecen a la clase String, Ruby tiene un montón de métodos que facilita, en manejo de cadenas. Entre ellos están el método print y gets.

El método print es igual que el método puts pero no añade un salto de línea al final y gets lee una cadena del tecldo.

El método strip! Remueve el \n del final de la cadena si lo hay. Los métodos que terminan con el símbolo ! indican que el objeto va a ser modificado.

Símbolos: Los símbolos empiezan por el símbolo: y pueden contener letras, dígitos o barras bajas. Para Ruby los símbolos son más fáciles de procesar que las cadenas.

Generalmente se usan cuando se necesita una cadena pero no se necesite imprimirla por pantalla, por ejemplo para pasarle un mensaje a un objeto.

:soy_un_simbolo :rojo :mil :on :off

Rangos: Los rangos en Ruby también son un tipo. Se representan por dos valores entre paréntesis separados por dos o tres puntos.

(1..5) representa los números del 1 al 5
('a'..'z') representa las letras de la 'a' a la 'z'
(1...5) representa los números del 1 al 4

El operador === sobre rangos sirve para comprobar si un valor está comprehendido en el rango, en caso afirmativo devuelve **true** y en caso contrario **false**. El rango ha de estar a la izquierda del operador:

(1..10) === 5

Expresiones regulares: En Ruby las expresiones regulares son un tipo base como las cadenas o los números. Su utilidad es la de encontrar palabras o patrones en un texto o comprobar que algo cumple un patrón. Las expresiones regulares van delimitadas por barras. La manera más simple de utilizar una expresión regular para buscar una palabra en una cadena sería la siguiente:

texto.mach(/Ruby/)

El método match compruebo si se cumplen las condiciones de la expresión regular, si se cumplen devuelve un objeto de tipo MatchData, en caso contrario devuelve nil. En nuestro ejemplo match comprueba si en la cadena está presente la palabra Ruby.
Las expresiones regulares disponen de algunos caracteres y combinaciones de caracteres con un significado especial:

Tabla 5. *Caracteres para expresiones*

Símbolo	Significado
[a-z]	Indica cualquier carácter entre la a y la z
[ab]	Indica una a o una b
\d	Un digito. Equivale a [0-9]
\w	Una letra, un dígito o una barra baja. Equivale a [A-Za –z=-9]
\s	Espacio en blanco [\t\r\n]
\b	Límite de palabra
\D	Cualquier cosa excepto dígitos. Equivale a [\d]
\W	Es la negación de \w
\S	Es la negación de \s
\B	Es la negación de \b
.	Cualquier carácter

{n}	Exactamente n veces lo que precede
{nm}	Al menos n y como máximo m de lo que precede
{n,}	N o más de lo que precede
*****	Cero o más repeticiones de lo que precede
+	Una o más repeticiones de lo que precede
?	Cero con una repetición de lo que precede
()	Agrupamiento
^	Línea empieza por lo que va a continuación
$	Línea terminada por lo que precede
i	Puesto después de la última barra delimitadora hace que no se distinga mayúsculas de minúsculas.

Autor: *Matz*

Array: Los array son colecciones de valores agrupados alrededor de corchetes y separados por comas. Un array puede contener valores de cualquier tipo y no necesariamente todos del mismo tipo. Como en la mayoría de los lenguajes de programación se puede acceder a un elemento usando un índice dentro del operador []. Los métodos de repetición, concatenación y corte de las cadenas también sirven para los array. La diferencia importante con otros lenguajes de programación es que si se intenta acceder a un elemento que no está en el array no se produce un error de índice fuera del rango sino que el array devuelve el valor nil (ausencia de valor). Según (Navenn, 2008)

Hashes: También llamados vectores asociativos o diccionarios, los hashes son como los arrays pero no están ordenados y se accede a los elementos mediante una clave. Por tanto los hashes son una colección de pares clave, valor rodeados por llaves dónde clave y valor están separados por los símbolos => y pueden ser se cualquier tipo. Por supuesto no puede haber claves repetidas. Si se intenta acceder a una clave inexistente devuelve nil. Según (Schmidt M. K., 2005)

2.3.8.3. Operadores

Como en todos los lenguajes de programación, en Ruby también existen los llamados operadores. Un operador es básicamente un símbolo que se utiliza en expresiones como "1+3″ donde 1 y 3 serían operandos y "+" es el operador[24].

Existen diferentes tipos de operadores, operadores aritméticos, de asignación, relacionales, lógicos, bitwise y de identidad, los operadores más sencillos son los aritméticos, de comparación y asignación.

[24]Villalobos, J. & Hernández, Tutoriales de Ruby: Expresiones booleanas y operadores lógicos.

Operadores aritméticos: Los operadores aritméticos son los más sencillos de todos, se utilizan para realizar operaciones aritméticas básicas, es decir sumas, restas, multiplicación división, modulo/residual, y exponenciales.

Tabla 6. Operadores aritméticos

Operador	Expresión
Suma	a + b
Resta	a – b
Multiplicación	a * b
División natural	a / b
División con floor/piso	a//b
Modulo residuo	a%b
Exponencial	a**b

Autor: Matz

Operadores de comparación: Los operadores de comparación se usan para evaluar expresiones que solo pueden tener 2 resultados, estos resultados son verdadero o falso (true o false) y son los siguientes.

Tabla 7. Operadores de Comparación

Operador	Significado
==	Evalúa como verdadero si 2 variables son iguales.
!=	Evalúa como verdadero si 2 variables son diferentes
<>	Lo mismo que !=
>	Verdadero si el operador a la izquierda es mayor que el de la derecha.
<	Verdadero si el operador a la izquierda es menor que el de la derecha
>=	Verdadero si el operador a la izquierda es mayor o igual al de la derecha.
<=	Verdadero si el operador a la izquierda es menor o igual al de la derecha.

Autor: Matz

Operadores de Asignación: Los operadores de asignación se utilizan para básicamente asignar un valor a una variable, así como cuando utilizamos el "=".

Los operadores de asignación son "=,+=,-=,*=,/=,**=,//=", a continuación algunos ejemplos.

Tabla 8. Operadores de asignación

Operador	Significado
=	Igual a, es el más simple de todos y asigna a la variable del lado Izquierdo cualquier variable o resultado del lado derecho
+=	Suma a la variable del lado izquierdo el valor del lado derecho
-=	Resta a la variable del lado izquierdo el valor del lado derecho
*=	Multiplica a la variable del lado izquierdo el valor del lado derecho.

2.3.8.4. Comentarios

Cualquier línea precedida por '#' es ignorada por el intérprete. Además, cualquier cosa que sea escrita entre las líneas '=begin' y '=end' (empezando ambas en la primera columna de su correspondiente línea), también será ignorada.

```
#Comentario de una sola línea
=begin
Esto es
un comentario
de varias
líneas
=end

=begin
Este comentario multilínea
da un error.
=end
```

MUY IMPORTANTE: este último tipo de comentarios, no puede tener espacios a su izquierda, por que daría un error. Por lo tanto, si se quiere usar, siempre van pegados al margen izquierdo de la pantalla.

2.3.8.5. Arreglos

Colección de referencias a objetos. En Ruby, las variables guardan la referencia a un objeto, y no el objeto en sí. Cada referencia a un objeto ocupa una posición en el array identificado por un índice entero.

Se indexan los elementos usando el operador, [] un método de instancia de la clase Array que puede ser sobrescrito. Llamar a una posición de un entero positivo, retorna el objeto en ese lugar o nil si no hay nada ahí. Con los índices negativos, se cuenta desde el final siendo -1 el último elemento del array, y -array.length el primero. Según (S. Brown, 2012, págs. 233,123,120)

Se pueden indexar con un par de números (inicio, cantidad). Esto devuelve un nuevo array con referencias a la cantidad de objetos comenzando en la posición inicio. También se pueden usar rangos. Ruby implementa un objeto para representar rangos como los meses de Enero a Diciembre, secuencias de números y demás. En el caso de los arrays, se usan los rangos que implementan secuencias (también implementan condiciones e intervalos).

2.3.8.6. Estructuras de control de flujo

En Ruby las sentencias de control son un poco diferentes de los demás lenguajes de programación. Según (I. Jacobson, 2010, págs. 100-120)

If y unless: En Ruby todo lo susceptible de devolver un valor lo hace. Por eso if, que en otros lenguajes se dice es una sentencia de control, en Ruby se dice que es una expresión ya que devuelve un valor. Se puede incluir la palabra then después de la condición. Puede anidar más condiciones con elsif y añadir al final un else opcionalmente. Al final has de incluir la palabra end para finalizar el bloque. La identación es opcional.

Case: Case es como switch de C, es tener varios else – if anidados, pero con una sintaxis más cómoda.

While y until: El bucle while se ejecuta mientras se cumpla una condición y el until mientras no se cumpla. Al igual que if y unless, while y until se pueden usar como modificadores.

For: El bucle **for** itera sobre colecciones al igual que hace el iterador each. Tiene una sintaxis muy parecida a Python:

```
for elemento initerable
. . .
End
```

Iterable es cualquier cosa sobre la que se pueda iterar (un rango, un array, una cadena, un diccionario), de hecho Ruby traduce eso a algo como:

```
Iterable .each{|elemento|...}
```

2.3.8.7. Estructuras de control de flujo condicionales

Declaración if: Un bloque de código dado es ejecutado solo si la condición es verdadera (true).

```
if condición
# código a ejecutar si la condición es verdadera
end
```

Declaración if…else: Se puede definir un bloque de código a ser ejecutado cuando la condición no se cumpla usando else (si no):

```
if condición
#código a ejecutar si la condición es verdadera
else
#código a ejecutar si la condición es falsa
```

end

Declaracion if…elsif…else: Mediante el uso de elsif podemos agregar condiciones con su respectivo bloque de código:

if condicion1
#código a ejecutar si condicion1 es verdadera
elsif condicion2
#código a ejecutar si condición 1 es falsa y condicion2 verdadera
else
#código si ninguna de las condiciones anteriores era verdadera
end

Declaración unless: Esta declaración que significa "a menos que" ejecuta un bloque de código si la condición es falsa o nil.

unless condición
#código a ejecutar si la condición es falsa o nil
end

Declaración unless…else: Así como if, unless también dispone de else.

unless condición

#código a ejecutar si la condición es falsa o nil
else
 # código a ejecutar si la condición es verdadera
end

Declaración case: Permite probar "casos" al comparar la expresión dada en "case" con las expresiones dadas en "when" y luego ejecutando el código correspondiente. En caso que ningún caso coincida se ejecuta el bloque de código bajo else:

case expr0
when expr1, expr2
#código a ejecutar si expr0 coindice con expr1 o expr2
when expr3
#código a ejecutar si expr0 coincide con expr3
else
 # código a ejecutar si no coindice con ningún caso
end

Repeat: Estos bucles se ejecutan siempre al menos una vez, ya que entra en el bloque de código y al final se verifica la condición de salida. La sintaxis de este tipo de bucles es:

Código while condición

O de la siguiente manera:

begin
código
end while condición

2.3.8.8. Métodos

En Ruby, todo lo que se manipula es un objeto, y el resultado de esas operaciones también son objetos. La única forma que tenemos de manipular los objetos, son los métodos:

Si los objetos (como los strings, números) son los nombres, entonces los métodos son los verbos. Todo método necesita un objeto. Es fácil decir qué objeto recibe el método: el que está a la izquierda del punto. Algunas veces, puede que no sea obvio. Por ejemplo, cuando se usa puts y gets, ¿dónde están sus objetos? Nada más iniciarse el intérprete, estamos dentro de un objeto: el objeto main. Por tanto, al usar puts y gets, se está mandando el mensaje al objeto main. Según (Schmidt M. D., 2013., págs. 288-290)

Escribiendo métodos: Un bloque de instrucciones que define un método, empieza por la palabra def y acaba por la end. Los parámetros son la lista de variables que van entre paréntesis. Aunque en Ruby, dichos paréntesis son opcionales: puts, p y gets son muy usados, y por ello que el uso de paréntesis sea opcional. Un método devuelve el valor de su última línea. Por norma, es recomendable dejar una línea en blanco entre las definiciones de métodos:

A un método se:

- Definición
- Usa
- Puede ser un argumento

Los métodos bang (!): Los métodos que acaban con una ! son métodos que modifican al objeto. Por lo tanto, estos métodos son considerados como peligrosos, y existen métodos iguales, pero sin el !. Por su peligrosidad, el nombre "bang".

Normalmente, por cada método con !, existe el mismo método sin !. Aquellos sin bang, nos dan el mismo resultado, pero sin modificar el objeto (en este caso el string). Las versiones con !, como se dijo, hacen la misma acción, pero en lugar de crear un nuevo objeto, transforman el objeto original. Según (S. Brown, 2012)

Ejemplos de esto son: upcase/upcase!, chomp/chomp!. En cada caso, si se hace uso de la versión sin !, se tiene un nuevo objeto. Si se llama el método con !, se hace los cambios en el mismo objeto al que se mande el mensaje.

Alias: Crea un nuevo nombre que se refiere a un método existente. Cuando a un método se le pone un alias, el nuevo nombre se refiere al método original: si el método se cambia, el nuevo nombre seguirá invocando el original.

```
alias nuevo_nombre nombre_original
def viejo_metodo
"viejo metodo"
end
alias nuevo_metodo viejo_metodo
def viejo_metodo
"viejo metodo mejorado"
end
puts viejo_metodo
puts nuevo_metodo
```

En el resultado, vemos como nuevo_metodo hace referencia al viejo_metodo sin modficar:

```
viejo metodo mejorado
viejo método
```

NOTA: a día de hoy Ruby no se lleva muy bien con las tildes. Esto es porque no tiene soporte para strings Unicode. Es ya implementarlo en la versión 1.9 que ya fue lanzado en Diciembre del 2007.

Métodos perdidos: Cuando se manda un mensaje a un objeto, el objeto busca en su lista de métodos, y ejecuta el primer método con el mismo nombre del mensaje que encuentre. Si no encuentra dicho método, lanza un error NoMethodError.

Una forma de solucionar esto, es mediante el método **method_missing**: si se define dicho método dentro de una clase, se ejecuta este método por defecto:

```
class Dummy
def method_missing(m, *args)
puts "No existe un metodo llamado #{m}"
end
end
Dummy.new.cualquier_cosa
```

Obtenemos: No existe un metodo llamado cualquier_cosa

Por lo tanto, method_missing es como una red de seguridad: te da una forma de manejar aquellos métodos que de otra forma darían un error en tu programa.

2.3.8.9. Orientación a objetos

Ruby es un lenguaje de programación totalmente orientado a objetos multiplataforma (lenguaje interpretado y de scripts), en el que RoR fue basado para su creación.

Desde hace tiempo el estilo de programación funcional (que se usa por ejemplo en el lenguaje C) es usado para programar. En este tipo de programación, hay que centrarse en los pasos para realizar la tarea, y no olvidarse de cómo se manejan los datos. Según (Schmidt M. K., 2005, págs. 66-70)

Sin embargo, en la programación orientada a objetos, los objetos son los agentes, el universo del programa: se presta atención a la estructura de los datos. Cuando se diseña una clase, se piensa en los objetos que serán creados por esa clase: en las cosas que podrá hacer ese objeto, y las características que lo definen. Un objeto es un contenedor de datos, que a su vez controla el acceso a dichos datos. Asociados a los objetos está una serie de variables que lo definen: sus atributos. Y también un conjunto de funciones que crean un interfaz para interactuar con el objeto: son los métodos. Un objeto es una combinación de estado y de métodos que cambian ese estado.

Una clase es usada para construir un objeto. Una clase es como un molde para objetos. Y un objeto, una instancia de la clase. Por ejemplo, se puede usar la clase Button para hacer docenas de botones, cada botón con su propio color, tamaño, forma. Según (Navenn, 2008)

El método new se usa para crear un nuevo objeto de la clase Perro. Los objetos son creados en el momento y el espacio de memoria donde se guardan, se asigna a una variable, en este caso la variable d, que se conoce como variable de referencia. Un método recién creado no es un espacio en blanco: un objeto recién creado, puede responder un montón de mensajes. Para ver la lista de esos mensajes o métodos de forma ordenada (.sort): *puts d.methods.sort*

El resultado es una lista de todos los mensajes o métodos que el objeto recién creado puede responder. De todos esos métodos, ¿los object_id yrespond_to? son importantes.

object_id, respond_to?: Cada objeto en Ruby tiene un único número asociado con él. Se puede ver dicho número mediante el método object_id.

puts "El número que identifica al objeto denotado por la variable d es #{d.object_id}."

Se puede conocer de antemano, si un objeto será capaz de responder a un mensaje; o dicho de otra forma, si un objeto posee cierto método. Para ello se usa respond_to?.

class, instance_of?: Puedes saber a qué clase pertenece un objeto mediante el método class.

instance_of?: Devuelve true si un objeto es instancia de una clase determinada.

La clase Class: Las clases en Ruby son instancias de la clase Class. Cuando se define una nueva clase (p.e. class NombreClase … end), se crea un objeto de la claseClass y es asignado a una constante (en este caso NombreClase). Cuando se usa NombreClase.new para construir un nuevo objeto, se usa el método de la clase Class para crear nuevas instancias; y después se usa el método inicializador de la propia clase NombreClase: la construcción y la inicialización de un objeto son cosas distintas, y pueden modificarse. Según (J. García, 2008)

Constructores literales: cada vez que usas uno de estos constructores, creas un nuevo objeto. Significa que se puede usar una notación especial, en vez de usar new para crear un nuevo objeto de esa clase. Las clases que un constructor literal puede crear, están en la siguiente:

Tabla 9. Constructores literales

Clase	Constructor Literal
String	“o”
Símbolo	:
Array	[ ]
Hash	{ }
Rango	… o …
Expresiones Regulares	/

Autor: Matz

Métodos de clase: La idea de los métodos de clase es mandar el mensaje a la clase, en vez de una de sus instancias. Los métodos de clase se usan porque algunas operaciones que pertenecen a una clase, no pueden ser realizadas por sus instancias. new es un buen ejemplo. La tarea de crear nuevos objetos sólo la puede hacer la clase; los objetos no pueden crearse a sí mismo.

2.4. PYTHON

2.4.1. Definición

Fue creado por Guido van Rossum y su nombre se debe a la afición de su creador a los humoristas británicos Monty Python.[25]

Python es un lenguaje de programación desarrollado como proyecto de código abierto y es administrado por la empresa Python software Foundation. Se trata de un lenguaje de programación en scripts y de un lenguaje interpretado, lo que permite ahorrar el proceso de compilado. Python permite dividir el programa en módulos reutilizables desde otros programas Python. También viene con una gran colección de módulos estándar que proporcionan E/S de ficheros, llamadas al sistema, sockets, interfaces GUI, etc.

2.4.2. Características

Python es un lenguaje de programación de propósito general, de muy alto nivel (estos son, un alto nivel de abstracción con el uso de listas, tuplas, diccionarios).

Python es un lenguaje interpretado (no es necesaria compilación), dinámico (no necesita identificar explícitamente los tipos de datos para inicializar variables de modo lo que los tipos se validan durante la ejecución del programa) y fuertemente tipado (no pueden mezclarse tipos, es necesario hacer conversiones). Según (Navenn, 2008)

Python es un lenguaje multiplataforma (Windows, Mac, Linux, etc), multiparadigma (imperativo, orientado a objetos y en menos medida funcional) y con gestión automática de memoria.

Por último, cabe destacar que Python es un lenguaje de programación con una sintaxis clara y sencilla, fácil de aprender, donde se pueden mezclar los diferentes paradigmas de programación de los que dispone, ampliamente documentado, extensible, que intenta obligar al desarrollador de software a programar de la manera correcta en el menor tiempo posible.

[25]Python. - Lenguaje de programación referente a la introducción del software de desarrollo. *García, G. Ángel L. (2011).*

2.4.3. Herramientas para desarrollar en Python

Para programar en Python lo único que se necesita es el intérprete de Python, (se puede obtener de www.python.org) y el/los ficheros/s de código fuente Python a ejecutar. Por otra parte, tenemos ciertas herramientas de desarrollo para utilizar con Python, a saber:

2.4.3.1. IDES para Python

Se pueden nombrar IDLE, Eclipse con el plugins pyDev, NetBeans, Geany, pyScripter, Ninja IDE, Stani's Python Editor, Wingware Python[26] IDE, Komodo, Pyragua, Eric, Bluefish. Aptana Studio.

2.4.3.2. GUI para Python

Por ejemplo, para wxPython tenemos wxDesigner, wxFormBuilder o wxGlade. Para GTK, tenemos Glade, Qt dispone de Monkey Studio, etc.

2.4.3.3. Editor para Python

Por ejemplo Editra, emacs, Notepad++, SciTE

2.4.3.4. RAD para Python

De los más destacados para Python tenemos BOA Constructor y SharpDevelop (para IronPython).

2.4.3.5. Shell interactivo para Python

Por ejemplo PyCrust, Dreampie, PyShell, etc.

Lo mínimo que se le debe de pedir a cualquier IDE o editor, para desarrollar en Python, es la indentación automática, coloreado y completitud de código.

2.4.4. Versiones de Python

Actualmente en Python existen dos versiones activas, la 2.XX y la 3.XX. En el sitio web de Python existe un guion de ayuda para elegir entre la 2 y la 3: http://wiki.python.org/moin/Python2orPython3

[26] Python es un lenguaje de programación desarrollado como proyecto de código abierto y es administrado por la empresaPython software Foundation. Disponible en: http://www.alegsa.com.ar/Dic/python.php

A diciembre de 2010 había estables las versiones de 2.6, 2.7 y en desarrollo la 3.2. El futuro es Python 3, el cual es incompatible con las versiones 2.XX. Entonces, se puede elegir: dependiendo de las exigencias, y de los módulos de extensión (aquellos que no están en la distribución de Python) que utilicen (frameworks y demás). La mayoría siguen siendo compatibles con 2.XX pero no con 3 (aunque esto está cambiando).

2.4.5. Composición de Python

Una aplicación Python está compuesta de la siguiente manera:
PYTHON[27]: LENGUAJE + BIBLIOTECA ESTÁNDAR + *MÓDULOS DE EXTENSIÓN*.
El lenguaje y la biblioteca estándar conforman la instalación de base. El lenguaje se compone de palabras clave (if, for, etc), funciones integradas (abs, print, etc), tipos básicos (números, secuencias, diccionarios, conjuntos, cadenas, ficheros) y las reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.
NOTA: Una función integrada es aquella no es necesaria llamarla desde ningún módulo, pues se puede acceder a ella por defecto, esto es, se carga automáticamente. La biblioteca estándar es el conjunto de módulos, paquetes y frameworks que están en la instalación base de Python y que hay que declarar de manera explícita en el código. Por ejemplo, para acceder a las funciones del sistema operativo en el que se encuentre nuestra instalación Python invocaremos al módulo **os**.
Los módulos de extensión, que pueden ser frameworks, módulos o paquetes, es software de terceros, que hay que instalar por separado.

2.4.6. Filosofía Python

Python está planteado para desarrollar software a partir de unos principios de programación. Los usuarios de Python se refieren a menudo a la **Filosofía Python** que es bastante análoga a la filosofía de Unix. El código que sigue los principios de Python de legibilidad y transparencia se dice que es "**pythonico**". Contrariamente, el código opaco u ofuscado es bautizado como "no pythonico" ("unpythonic" en inglés). Estos principios fueron famosamente descritos por el desarrollador de Python Tim Peters en El Zen de Python:

[27] Python es un lenguaje de programación desarrollado como proyecto de código abierto y es administrado por la empresa Python. Disponible en: http://www.alegsa.com.ar/Dic/python.php

- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Ralo es mejor que denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para quebrantar las reglas.
- Los errores nunca deberían dejarse pasar silenciosamente.
- A menos que hayan sido silenciados explícitamente.
- Frente a la ambigüedad, rechazar la tentación de adivinar.
- Debería haber una -y preferiblemente sólo una- manera obvia de hacerlo.
- Aunque esa manera puede no ser obvia al principio a menos que usted sea Holandés.
- Ahora es mejor que nunca.
- Aunque nunca es a menudo mejor que ya.
- Si la implementación es difícil de explicar, es una mala idea.
- Si la implementación es fácil de explicar, puede que sea una buena idea.
- Los espacios de nombres (namespaces) son una gran.

Estos principios son casi axiomáticos en Python, implementados además en el intérprete Python. Para visualizarlos ejecutar el intérprete Python y escribir: import this

2.4.7. Estructura y Elementos del Lenguaje

2.4.7.1. Elementos del Lenguaje

Como en la mayoría de los lenguajes de programación de alto nivel, en Python se compone de una serie de elementos que alimentan su estructura. Entre ellos, podremos encontrar los siguientes:

Variables: Una variable es un espacio para almacenar datos modificables, en la memoria de un ordenador. En Python, una variable se define con la sintaxis[28]:

nombre_de_la_variable = valor_de_la_variable

Cada variable, tiene un nombre y un valor, el cual define a la vez, el tipo de datos de la variable. Existe un tipo de "variable", denominada **constante**, la cual se utiliza para definir valores fijos, que no requieran ser modificados.

[28] *Bahit, E. (2012). Python para principiantes (pág. 20), N° páginas: 136*

Tipo de Datos: Una variable (o constante) puede contener valores de diversos tipos. Entre ellos:

Tabla 10. Tipo de datos

Variables
Número entero
Número entero octal
Número real
Booleano (verdadero / Falso)

Autor: Eugenia Bahit

Operadores Aritméticos: Entre los operadores aritméticos que Python[29] utiliza, podemos encontrar los siguientes:

Tabla 11. Operadores Aritméticos

Símbolo	Significado
+	Suma
-	Resta
-	Negación
*****	Multiplicación
******	Exponente
/	División
//	División Entera
%	Módulo

Autor: Eugenia Bahit

Comentarios: Un archivo, no solo puede contener código fuente. También puede incluir comentarios. Los comentarios pueden ser de dos tipos: de una sola línea o multi-línea y se expresan de la siguiente manera[30]:

```
# Esto es un comentario de una sola línea
mi_variable = 15
"""Y este es un comentario
de varias líneas"""
mi_variable = 15
mi_variable = 15 # Este comentario es de una línea también
```

En los comentarios, pueden incluirse palabras que nos ayuden a identificar, además, el subtipo de comentario:

TODO esto es algo por hacer

FIXME esto es algo que debe corregirse

[29] Python es un lenguaje de programación desarrollado como proyecto de código abierto y es administrado por la empresaPython software Foundation. Disponible en: http://www.alegsa.com.ar/Dic/python.php

XXX esto también, es algo que debe corregirse.

Tipos de Datos Complejos: Python, posee además de los tipos ya vistos, 3 tipos más complejos, que admiten una colección de datos. Estos tipos son:

- Tuplas
- Listas
- Diccionarios

Estos tres tipos, pueden almacenar colecciones de datos de diversos tipos y se diferencian por su sintaxis y por la forma en la cual los datos pueden ser manipulados.

Tuplas: Una tupla es una variable que permite almacenar varios datos inmutables (no pueden ser modificados una vez creados) de tipos diferentes:

mi_tupla = ('cadena de texto', 15, 2.8, 'otro dato', 25)

Se puede acceder a cada uno de los datos mediante su índice correspondiente, siendo 0 (cero), el índice del primer elemento:

print mi_tupla[1] # Salida: 15

También se puede acceder a una porción de la tupla, indicando (opcionalmente) desde el índice de inicio hasta el índice de fin:

print mi_tupla[1:4] # Devuelve: (15, 2.8, 'otro dato')

print mi_tupla[3:] # Devuelve: ('otro dato', 25)

print mi_tupla[:2] # Devuelve: ('cadena de texto', 15)

Otra forma de acceder a la tupla de forma inversa (de atrás hacia adelante), es colocando un índice negativo:

print mi_tupla[-1] # Salida: 25

print mi_tupla[-2] # Salida: otro dato

Listas: Una lista es similar a una tupla con la diferencia fundamental de que permite modificar los datos una vez creados mi_lista = ['cadena de texto', 15, 2.8, 'otro dato', 25]

A las listas se accede igual que a las tuplas, por su número de índice:

print mi_lista[1] # Salida: 15

print mi_lista[1:4] # Devuelve: [15, 2.8, 'otro dato']

print mi_lista[-2] # Salida: otro dato

Las lista NO son inmutables: permiten modificar los datos una vez creados:

mi_lista[2] = 3.8 # el tercer elemento ahora es 3.8

Las listas, a diferencia de las tuplas, permiten agregar nuevos valores:

mi_lista.append('Nuevo Dato')

Diccionarios: Mientras que a las listas y tuplas se accede solo y únicamente por un número de índice, los diccionarios permiten utilizar una clave para declarar y acceder a un valor: mi_diccionario = {'clave_1': valor_1, 'clave_2': valor_2, \ 'clave_7': valor_7}
print mi_diccionario['clave_2'] # Salida: valor_2
Un diccionario permite eliminar cualquier entrada: del(mi_diccionario['clave_2'])
Al igual que las listas, el diccionario permite modificar los valores mi_diccionario['clave_1'] = 'Nuevo Valor'.

2.4.7.2. Estructuras de Control de Flujo

Una estructura de control, es un bloque de código que permite agrupar instrucciones de manera controlada, entre ellas tenemos las siguientes:

- Estructuras de control condicionales
- Estructuras de control iterativas.

Para hablar de estructuras de control de flujo en Python, es imprescindible primero, hablar de identación.

La identación. En un lenguaje informático, la identación es lo que la sangría al lenguaje humano escrito (a nivel formal). Así como para el lenguaje formal, cuando uno redacta una carta, debe respetar ciertas sangrías, los lenguajes informáticos, requieren una identación.

No todos los lenguajes de programación, necesitan de una identación, aunque sí, se estila implementarla, a fin de otorgar mayor legibilidad al código fuente. Pero en el caso de Python, la dentición es obligatoria, ya que, de ella, dependerá su estructura.

Encoding: El encoding (o codificación) es otro de los elementos del lenguaje que no puede omitirse a la hora de hablar de estructuras de control.

 # -- coding: utf-8 -*-*

utf-8 podría ser cualquier codificación de caracteres. Si no se indica una codificación de caracteres, Python podría producir un error si encontrara caracteres “extraños”.

Estructuras de Control de Flujo Condicionales: Las estructuras de control condicionales, son aquellas que nos permiten evaluar si una o más condiciones se cumplen, para decir qué acción vamos a ejecutar. La evaluación de condiciones, solo puede arrojar 1 de 2 resultados: verdadero o falso (True o False). En la vida diaria, se actúa de acuerdo a la evaluación de condiciones, de manera mucho más frecuente de lo

que en realidad se cree: *Si el semáforo está en verde, cruzar la calle. Sino, esperar a que el semáforo se ponga en verde.* A veces, también se evalúa más de una condición para ejecutar una determinada acción: *Si llega la factura de la luz y tengo dinero, pagar la boleta. Según* (Schmidt M. D., 2013.)

Para describir la evaluación a realizar sobre una condición, se utilizan operadores relacionales (o de comparación):

Tabla 12. Operadores relacionales de comparación

Símbolo	Significado
==	Igual que
!=	Distinto que
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que

Fuente: Libro Python para principiantes

Y para evaluar más de una condición simultáneamente, se utilizan operadores lógicos:

Tabla 13. Operadores lógicos

Símbolo	Significado
Y	and
O	Or
x or	o excluyente

Fuente: Libro Python para principiantes

Estructuras de Control Iterativas: A diferencia de las estructuras de control condicionales, las iterativas (también llamadas cíclicas o bucles), nos permiten ejecutar un mismo código, de manera repetida, mientras se cumpla una condición. Según (J. García, 2008)

En Python se dispone de dos estructuras cíclicas:

- El bucle **while**
- El bucle **for**

Se detalle a continuación.

Bucle while: Este bucle, se encarga de ejecutar una misma acción "mientras que" una determinada condición se cumpla.

Bucle for: El bucle for, en Python, es aquel que nos permitirá iterar sobre una variable compleja, del tipo lista o tupla.

2.4.7.3. Módulos, paquetes y namespaces

Módulos Empaquetados: En Python, cada uno de nuestros archivos .py se denominan módulos. Estos módulos, a la vez, pueden formar parte de paquetes. Un paquete, es una carpeta que contiene archivos .py. Pero, para que una carpeta pueda ser considerada un paquete, debe contener un archivo de inicio llamado __init__.py. Este archivo, no necesita contener ninguna instrucción. De hecho, puede estar completamente vacío.

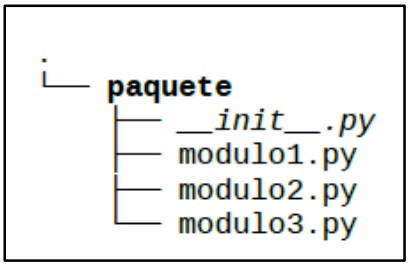


Figura 15. Paquete
Autor: Eugenia Bahit

Los paquetes, a la vez, también pueden contener otros sub-paquetes:

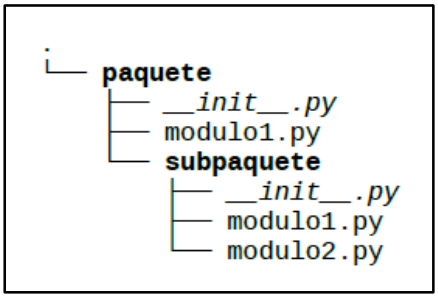


Figura 16. Subpaquete
Fuente: Libro Python para principiantes

Y los módulos, no necesariamente, deben pertenecer a un paquete:

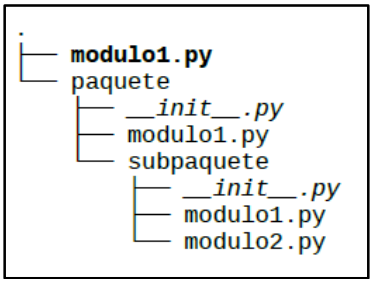


Figura 17. Módulo
Fuente: Libro Python para principiantes

Importando Módulos Enteros: El contenido de cada módulo, podrá ser utilizado a la vez, por otros módulos. Para ello, es necesario **importar los módulos** que se quieran

utilizar. Para importar un módulo, se utiliza la instrucción **import**, seguida del nombre del paquete (si aplica) más el nombre del módulo (sin el .py) que se desee importar.

 # -- coding: utf-8 *-**

import modulo # importar un módulo que no pertenece a un paquete

import paquete. modulo1 # importar un módulo que está dentro de un paquete

import paquete. subpaquete. modulo1

La instrucción import seguida de nombre_del_paquete.nombre_del_modulo, nos permite hacer uso de todo el código que dicho módulo contenga.

Namespaces: Para acceder (desde el módulo donde se realizó la importación), a cualquier elemento del módulo importado, se realiza mediante el *namespace*, seguido de un punto (.) y el nombre del elemento que se desee obtener. En Python, un namespace, es el nombre que se ha indicado luego de la palabra import, es decir la ruta (namespace) del módulo:

print modulo.CONSTANTE_1

print paquete.modulo1.CONSTANTE_1

print paquete.subpaquete.modulo1.CONSTANTE_1

Alias: Es posible también, abreviar los namespaces mediante un "alias". Para ello, durante la importación, se asigna la palabra clave as seguida del alias con el cual nos referiremos en el futuro a ese namespace importado:

import modulo as m

import paquete.modulo1 as pm

import paquete.subpaquete.modulo1 as psm

Luego, para acceder a cualquier elemento de los módulos importados, el namespace utilizado será el alias indicado durante la importación:

print **m.**CONSTANTE _1

print **pm.**CONSTANTE _1

print **psm.**CONSTANTE_1

2.4.7.4. Funciones definidas por el Usuario

Una función, es la forma de agrupar expresiones y sentencias (algoritmos) que realicen determinadas acciones, pero que éstas, solo se ejecuten cuando son llamadas. Es decir, que al colocar un algoritmo dentro de una función, al correr el archivo, el algoritmo no será ejecutado si no se ha hecho una referencia a la función que lo contiene. Según (Schmidt M. D., 2013.)

Definiendo funciones: En Python, la definición de funciones se realiza mediante la instrucción def más un nombre de función descriptivo -para el cuál, aplican las mismas reglas que para el nombre de las variables- seguido de paréntesis de apertura y cierre. Como toda estructura de control en Python, la definición de la función finaliza con dos puntos (:) y el algoritmo que la compone, irá identado con 4 espacios:

```
def mi_funcion():
# aqui el algoritmo
```

Parámetros: Un parámetro es un valor que la función espera recibir cuando sea llamada (invocada), a fin de ejecutar acciones en base al mismo. Una función puede esperar uno o más parámetros (que irán separados por una coma) o ninguno.

```
def mi_funcion(nombre, apellido):
# algoritmo
```

Los parámetros que una función espera, serán utilizados por ésta, dentro de su algoritmo, a modo de variables de ámbito local.

Llamadas de retorno: En Python, es posible (al igual que en la gran mayoría de los lenguajes de programación), llamar a una función dentro de otra, de forma fija y de la misma manera que se la llamaría, desde fuera de dicha función:

def funcion():

```
return "Hola Mundo"
def saludar(nombre, mensaje='Hola'):
print mensaje, nombre
print mi_funcion()
```

Sin embargo, es posible que se desee realizar dicha llamada, de manera dinámica, es decir, desconociendo el nombre de la función a la que se deseará llamar. A este tipo de acciones, se las denomina llamadas de retorno. Para conseguir llamar a una función de manera dinámica, Python dispone de dos funciones nativas: locals() y globals().

Ambas funciones, retornan un diccionario. En el caso de locals(), éste diccionario se compone -justamente- de todos los elementos de ámbito local, mientras que el de globals(), retorna lo propio pero a nivel global. Según (J. García, 2008).

Llamadas recursivas: Se denomina llamada recursiva (o recursividad), a aquellas funciones que en su algoritmo, hacen referencia sí misma. Las llamadas recursivas suelen ser muy útiles en casos muy puntuales, pero debido a su gran factibilidad de caer en iteraciones infinitas, deben extremarse las medidas preventivas adecuadas y, solo

utilizarse cuando sea estrictamente necesario y no exista una forma alternativa viable, que resuelva el problema evitando la recursividad. Python admite las llamadas recursivas, permitiendo a una función, llamarse a sí misma, de igual forma que lo hace cuando llama a otra función. Según (I. Jacobson, 2010)

2.4.7.5. Orientación a Objetos

En Python todo es un "objeto" y debe ser manipulado -y entendido- como tal. Pensar en objetos, puede resultar -al inicio- una tarea difícil. Sin embargo, difícil no significa complejo. Por el contrario, pensar en objetos representa la mayor simplicidad que uno podría esperar del mundo de la programación. **Pensar en objetos, es simple**... aunque lo simple, no necesariamente signifique sencillo.

2.5. Análisis comparativo de las tecnologías multiplataforma

2.5.1. Definición de criterios de análisis

Los criterios que a continuación se describe para realizar el estudio comparativo entre las tecnologías multiplataforma para el desarrollo de aplicaciones web están basados en la información bibliográfica revisada propia de cada tecnología JSP, Ruby, Python, tomando en cuenta la relación directa entre ellas y seleccionando los aspectos más relevantes para hacer el análisis comparativo.

Tabla 14. Criterios de análisis entre tecnologías multiplataforma

Criterio	Definición
Características de la tecnología multiplataforma	Aspectos técnicos que permiten identificar información relevante de cada tecnología, de manera que facilitan su uso en la ejecución de tareas programación, configuración, implementadas en ambientes heterogéneos.

Autor: Marcela Cuello Olmedo

Los criterios generales que se han tomado en cuenta para realizar el análisis comparativo se divide en variables que se describen:

Tabla 15. Variables criterio: Características de la tecnología multiplataforma

Variable	Definición
Propiedades tecnológicas	Facultades técnicas que tienen las tecnologías multiplataforma para trabajar en diferentes ambientes simultáneamente.
Especificaciones Técnicas	Propiedades que cumplen y les caracteriza a las tecnologías multiplataforma.

Soporte en el desarrollo de aplicaciones web	Técnicas soportadas por las tecnologías multiplataforma, para facilitar el trabajo en el desarrollo y configuración de sistemas informáticos.
Peculiaridades de programación	Características funcionales que posee la tecnología multiplataforma para ser usada en el desarrollo de aplicaciones web.
Elementos del lenguaje	Componentes que posee un lenguaje de programación para facilitar la implementación de aplicaciones orientadas a la web, todo esto en un ambiente multiplataforma.

Autor: Marcela Cuello Olmedo

2.5.2. Pesos para variables de análisis

Tabla 16. Pesos para variables

Cualitativa	Cuantitativa
Cumple totalmente	2
Cumple parcialmente	1
No cumple	0

Autor: Marcela Cuello Olmedo

2.5.3. Análisis comparativo de tecnologías multiplataforma

2.5.3.1. Criterio Características de las tecnologías multiplataforma

Tabla 17. Criterio Características de las tecnologías multiplataforma

Variables	Tecnologías Multiplataforma		
	JSP	**RUBY**	**PYTHON**
Propiedades tecnológicas	2	1,33	1,50
Especificaciones Técnicas	1,83	2	2
Soporte en el desarrollo de aplicaciones web	2	1,63	1,63
Peculiaridades de programación	2	1,77	1,54
Elementos del lenguaje	2	1,55	1,73
Total	**9,83**	**8,28**	**8,40**
Porcentaje	98,3%	82,8%	84%

Autor: Marcela Cuello Olmedo

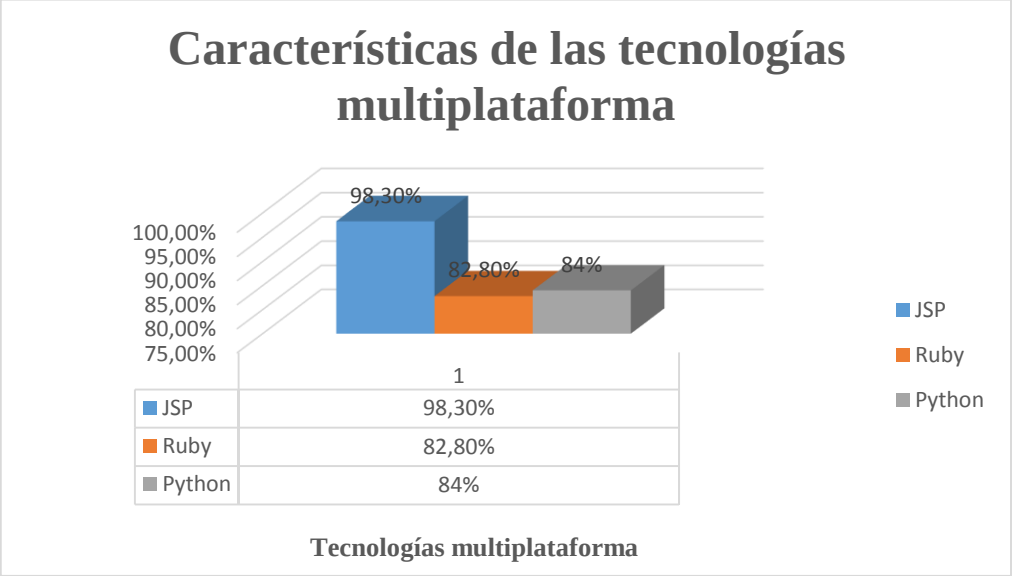


Figura 18. Características de las tecnologías multiplataforma
Autor: Marcela Cuello Olmedo

2.5.3.2. Análisis de resultados

A lo largo de la investigación he logrado encontrar, validar y determinar una serie de elementos importantes para seleccionar la tecnología multiplataforma apropiada para el desarrollo de un sistema que trabaje en cualquier ambiente tecnológico, catalogar una serie de características que permitan elegir una tecnología propia para ambiente heterogéneos.

Como resultado del análisis y de acuerdo al puntaje obtenido por cada una de las tecnologías analizadas se ha determinado que la Tecnología JSP alcanza un puntaje superior del (98,30%) que equivale a Bueno con relación a las otras tecnologías Ruby (82,80%), Python (84%).

2.5.3.3. Conclusiones de resultados

JSP: (98,30%) Es una tecnología multiplataforma soportada en Solaris, Windows, Mac OS, Linux trabajar con servidores web Apache, Netscape, IIS, integración bases de datos es mediante JDBC y ODBC, además de trabajar con componentes JavaBeans, Enterprise JavaBeans y extensiones JSP. Solo se ve limitado por la dificultad de aprendizaje que posee debido a que Java es un lenguaje muy potente, pero un poco más complicado de usar porque es orientado a objetos y la manera de escribir los programas es más rígida. Trabaja con versiones estables y actualizadas con el propósito de escribir aplicaciones y ejecutarlas en cualquier ambiente tecnológico, la capacidad de uso mayor cada vez

sorprendente ya que permite escribir programas prácticamente para todo lo que se pueda necesitar. Sus componentes JSP son reusables en distintas plataformas, su flexibilidad es notable al ser Java el lenguaje de desarrollo que tiende a ser potente y escalable. Soporta contenido dinámico que refleja las condiciones del mundo real y existe independencia entre la parte del diseño y la lógica. Existen IDES de gran uso, Ejemplo: **NetBeans** (un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java). JSP proveer técnicas de desarrollo, un lenguaje estructurado, orientado a objetos y respalda su trabajo al usar etiquetas HTML, plantillas con una sintaxis limpia y además se apoya de otros lenguajes de programación, de esta manera permiten la construcción de aplicaciones web de forma simplificada y acelerada, no descuida elementos de programación que favorece eficientemente un desarrollo completo y respaldado técnicamente, declaraciones, variables, operadores, documentación a nivel de código, estructuras de control, librerías, paquetes y métodos actúan eficientemente en la producciones de nuevos sistemas software.

RUBY: (82,80%) Una tecnología con licencia de software libre, multiplataforma que se integran en gran cantidad de arquitecturas, incluso en dispositivos móviles, se ve aun limitada en: la reusabilidad de componentes, la protección me memoria (gestión de memoria automática) y la integración de bases de datos. Ofrece versiones actualizadas de IDES, editores tiene como propósito hacer que la programación sea divertida y flexible para el programador, e influenciada por los lenguajes de programación Ada, C++, CLU, DYLAN, EIFEEL, LISP, PERL, PYTHON, sitios famosos como TWITTER, HULU y GROUPON han logrado posesionarse con éxito en la red mundial del internet. La usabilidad de la tecnología es satisfactoria ya que el código Ruby elegante, potente y expresiva. Ruby es mejor para un programador que ya sabe más lenguajes de programación. Se puede modificar absolutamente todo dentro del ambiente Ruby. Limitada por la no existencia de independencia entre la parte de diseño y la lógica de programación. Esta tecnología tiene limitaciones al no ser un lenguaje de scripting, no se compila, es por eso que es un lenguaje de guiones o Scripts, mayormente trabaja con etiquetas especiales además del manejo de datos complejos, declaración de variables y manejo de archivos, aunque es lo que le caracteriza.

PYTHON: (84%) También considerado multiplataforma (Windows, Mac, Linux) pero limitada al uso de componentes propios que al ser propios de la tecnología en algunos casos un módulo, paquete o componente requiere una instalación por separado, tiene versiones actualizadas de IDES, editores para el desarrollo, su propósito es enfatizar la

productividad y legibilidad del código; sitios famosos como YOUTUBE, GOOGLE han logrado posesionarse con éxito en la red mundial del internet. La usabilidad de Python es sorprendente la legibilidad que ofrece el lenguaje de programación; Python es ideal para principiantes, a menudo recomendado por los programadores debido a la simplicidad de su sintaxis. La libertad y facilidad de reusar componentes aun limita a esta tecnología. Además de la independencia entre la parte de diseño y lógica. Es una tecnología que permite el desarrollo de aplicaciones web con soporte para un buen procesamiento de trabajo, dinámico y flexible a nuevos requerimientos.

La productividad, la creación de aplicaciones web de manera simplificada y acelerada limitan a esta tecnología multiplataforma, lo que se resume a que un programador desarrolle aplicaciones rigurosamente de manera correcta y tiene características favorables a nivel de lenguajes de programación, permite el uso de variables, tipo de datos simples y complejos, manejo de estructuras de control, arreglos, métodos, archivos, aunque al ser una tecnologías que va cobrando fuerza en el mercado de desarrollo de software aún se limita el uso adecuado de tipos de datos complejos y el manejo de archivos.

2.5.3.4. Conclusiones generales

De acuerdo al análisis realizado y por los resultados obtenidos por cada una de las tecnologías multiplataforma, se concluye que: JSP (Java Server Page) brinda las mejores prestaciones en el desarrollo de aplicaciones web multiplataforma para el seguimiento de egresados y graduados de la Universidad Nacional de Chimborazo

CAPITULO III

3. Desarrollo del sistema web para el seguimiento a egresados y graduados de la UNACH.

3.1. Metodología CRAIG LARMAN

Se realiza en base a la metodología de Craig Larman, define una serie de actividades que son adaptables a las condiciones de este proyecto, el mismo que no fija metodología estricta y posibilita omitir algunas actividades de dicha metodología. Larman propone un ciclo de vida interactivo e incremental (el más usado) haciéndole suficientemente flexible y adaptable a las necesidades de los desarrolladores, esta metodología va hacer aplicada siguiendo el siguiente esquema:

- Planificación y especificación de requisitos (Definición de requisitos, Definición de casos de uso en formato de alto nivel.)
- Análisis (Definición de casos de uso en formato extendido, Definición de los diagramas de secuencia del sistema, Diagramas de estado, Modelo conceptual, Diagrama de actividades.)
- Diseño (Definición de la arquitectura del sistema, Definición de interfaz de usuario, Diagramas de interacción, Diagramas de clases de diseño, Esquema de la base de datos, Diagramas de componentes, Diagramas de despliegue.)
- Implementación

3.2. Planificación y especificación de requisitos

3.2.1. Definición del ámbito del software

La Universidad Nacional de Chimborazo tiene como desafío lograr la evaluación y acreditación como institución de Educación Superior de calidad, este desafío exige incorporar nuevas formas de gestionar los procesos institucionales, aprovechando los recursos tecnológicos se puede desarrollar aplicaciones informáticas que contribuyan al aseguramiento de la calidad Universitaria. Entre los procesos exigidos esta realizar un seguimiento a los profesionales y egresados de la facultad de Ingeniería de la Unach, entonces, se plantea desarrollar una aplicación para procesar información en cualquier plataforma, Linux o Windows, se piensa directamente en tecnologías de desarrollo y en lenguajes de programación para dichas plataformas. El contar con medios que den más sentido a las aplicaciones que se van desarrollando es dar un paso de calidad, donde otorgar otro grado de interacción del desarrollador con las tecnologías de trabajo marca diferencias.

Con el objetivo de mejorar la gestión académica de egresados y profesionales se desarrollara un sistema software Web de Seguimiento Universitario (**SYGEGE**) que permitirá conocer la situación actual de los profesionales y egresados de la facultad, el sistema proporcionara los requisitos técnicos para recoger información de ámbito laboral y educativo por parte de los encuestados, los resultados que arroje una debida tabulación de datos ayudara a la toma de decisiones por parte de la autoridades respectivas. El sistema **SYGEGE** gestionara módulos de encuestas, tabulación, reportes.

El sitio Web será puesto en producción en la plataforma tecnológica de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo. De esta manera se aumenta el nivel de calidad institucional, ya que proporcionar soluciones tecnológicas en procesos académicos contribuye a lograr la excelencia universitaria.

3.2.2. Antecedentes tecnológicos

Actualmente la Facultad de Ingeniería de la Universidad nacional de Chimborazo cuenta con los siguientes recursos:

Tabla 18: Recurso Humano

Nombre	Cargo
Lic. Lorena Ortega	**Secretaria**
Ing. Samuel Moreno	**Técnico**

Autor: Marcela Cuello Olmedo

Tabla 19: Recurso Hardware

Cantidad	Hardware	Especificación
10	Computador de escritorio	Procesador INTEL Pentium IV 2.0 GHz. Memoria 1GB Disco duro 500 GB CD – ROM Teclado Mouse Monitor Samsung SyncMaster S58 de 17”

Autor: Marcela Cuello Olmedo

Tabla 20: Recurso Software

Cantidad	Software	Especificación
1	Windows 7	Sistema Operativo

Autor: Marcela Cuello Olmedo

3.2.3. Definición de la alternativa de solución

El sistema **SYGEGE** presenta una interfaz de usuario que permite la gestión de encuestas, tabulación de encuestas y reportes de forma ordenada. El sistema está compuesto por los siguientes módulos

- Módulo de Administración
- Módulo de Encuestas

El sistema software simula el comportamiento de una encuesta informativa, este sistema permite a sus usuarios proporcionar información de la situación laboral, profesional y personal de los profesionales y egresados de la facultad de Ingeniería de la Unach, además de otras funciones como las de consultar sus datos personales, generar reportes de la información, etc.

La funcionalidad de este sistema básicamente es la siguiente: un usuario debe darse de alta en el sistema mediante el proceso de ingreso de su información personal, profesional y laboral. Una vez que tiene constancia que ya es parte del sistema, puede empezar a realizar la encuesta electrónica, cuando ya haya terminado puede consultar su información básica. El sistema en cada momento tendrá constancia de los usuarios que hayan ingresado su información y de los que hayan terminado las encuestas.

Los administradores del sistema en cualquier momento podrán solicitar información estadística de las preguntas contestadas en la encuesta y además de obtener información de los encuestados. Este proceso se encuentra representado en la siguiente figura.

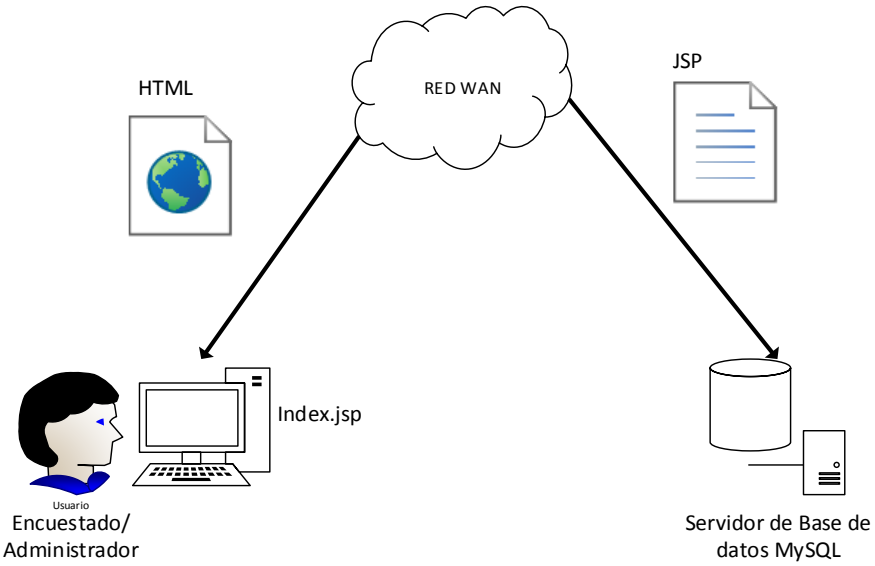


Figura 19: Arquitectura SYGEGE
Autor: Marcela Cuello Olmed

3.2.4. Características de los usuarios

Los usuarios del producto deberán desenvolverse con soltura en el uso de las nuevas tecnologías, en concreto deberán tener unas nociones básicas sobre navegadores web y encuestas electrónicas.

Los usuarios que hagan uso del sistema **SYGEGE** deben cumplir los requisitos siguientes: ser profesionales o egresados de la facultad de ingeniería de la Unach, proporcionar información personal, profesional y laboral verídica, conocer procesos para realizar encuestas en internet. Por otra parte, están los usuarios administrativos, que deben conocer los procedimientos para generar reportes de las encuestas ingresadas.

3.2.5. Requisitos funcionales

Tabla 21: Requisitos funcionales

Nº	Requisitos	Responsable
RQ1	El sistema software SYGEGE permitirá el Ingreso la información personal, laboral y profesional del encuestado.	Encuestado
RQ2	El sistema software SYGEGE permitirá registrar datos de encuestas	Encuestado
RQ3	El sistema software SYGEGE permitirá loguearse en el sistema.	Encuestado
RQ4	El sistema software SYGEGE permitirá consultar información	Encuestado
RQ5	El sistema software SYGEGE permitirá modificar la información personal, laboral y profesional de los encuestados.	Encuestado
RQ6	El sistema software SYGEGE permitirá loguearse en el sistema.	Administrador
RQ7	El sistema software SYGEGE permitirá eliminar encuestado	Administrado
RQ8	El sistema software SYGEGE permitirá tabular la información proporcionada en las preguntas de la encuesta.	Administrador
RQ9	El sistema software SYGEGE permitirá generar reportes por cada pregunta del formulario de la encuesta y de acuerdo a la facultad y escuela seleccionada.	Administrador

Autor: Marcela Cuello Olmedo

3.2.6. Requisitos de interfaz

Interfaz de usuario: Las salidas del sistema van destinadas al usuario final administrador y encuestado serán ofrecidas a través de la pantalla en las cuales se realizarán las operaciones debidas y permitidas, además complementadas con mensajes de error, advertencia e información. Además, el usuario tendrá la posibilidad de elegir comunicarse con el sistema, sea con el teclado o botones como se hace normalmente.

Interfaz con otros sistemas: El sistema SYGEGE no interactúa con ningún otro sistema.

3.2.7. Requisitos no funcionales

Requisitos de hardware: Con el fin de que el sistema SYGEGE sea accesible y evitar errores de comunicación en su fase de producción, se sugiere que todo el equipo hardware del usuario administrador y encuestado sea un ordenador de escritorio con altas capacidades de soporte técnico. Para lo cual se recomienda equipo hardware de tecnología actual, con las características siguientes:

Tabla 22: Requisitos de hardware

Cantidad	Hardware	Especificaciones
1	Computador personal	Procesador Core i3, 2.5 Ghz, Memoria 2 GB, Disco Duro 500 GB, Teclado, Monitor Samsung SyncMaster S58 de 17'', Mouse.

Autor: Marcela Cuello Olmedo

Requisitos de software: El sistema SYGEGE puede ser ejecutado en cualquier ordenador, con un sistema operativo de cualquier plataforma (Windows, Linux), se exige que debe poseer conexión a internet y tener instalado un navegador de páginas web (Mozilla Firefox, Google Chrome, Internet Explorer). Los siguientes atributos del sistema, definen al *sistema* tal como lo conocemos u observamos, entre los atributos tenemos:

Tabla 23: Requisitos de software

Atributo	Descripción
Portabilidad	La aplicación será portable a otros sistemas operativos, se diseñará con tal objetivo
Fácil de usar	Fácil de aprender, agradable para el usuario.
Disponibilidad	El sistema debe estar disponible cuando se lo requiera
Flexible	El sistema debe estar en capacidad de ser modificado.
Fiabilidad	La aplicación debería tener máxima fiabilidad ya que se va a ejecutar en un entorno productivo, aun así, en caso de fallo, no correrán peligro vidas humanas o animales, con lo que la fiabilidad no es crítica.
Mantenibilidad	La aplicación se dejará estructurada para poder acometer en el futuro ampliaciones en la funcionalidad, la aplicación no necesitará mantenimiento, aunque si requerirá que el servidor esté operativo y un mínimo mantenimiento de la BDD en caso de un número elevado de clientes.
Seguridad	Al ser una aplicación en la que se registrara información y que podrían leer terceras personas, es necesario idear un sistema que evite la suplantación de identidad, y ataques.

Autor: Marcela Cuello Olmedo

3.2.8. Estimación de Costos

La estimación de costos para el proyecto se lo realiza con la finalidad de obtener una valoración del esfuerzo y recursos necesarios para el desarrollo de sistema software. SIGEGE

3.2.8.1. Costos complementarios

Costo de hardware: En la definición del ámbito de software, en las secciones antecedentes tecnológicos se ha definido que la Facultad de Ingeniería cuenta con los recursos hardware necesario para el funcionamiento del sistema software SYGEGE.

Costo de software: No existe costo a pagarse por motivo de licencias, porque se utiliza software libre para el desarrollo del sistema software SIGEGE.

3.2.9. Factibilidad

El detalle de los tipos de factibilidad que se considera para el desarrollo del sistema SYGEGE es el siguiente:

a) Factibilidad técnica

La disponibilidad de la tecnología (técnica - humana) que satisface las necesidades del usuario, determina si la solución propuesta puede ser implantada con el hardware, software y recurso técnico (recurso humano) que se dispone. En el desarrollo del sistema software SYGEGE se cuenta con los recursos hardware y software necesarios. Las características del hardware y software requerido, así como el personal técnico y profesional requerido para el desarrollo del sistema, es el siguiente:

Hardware requerido: Como recursos hardware adicional recomendado se considerarán los siguientes:

Tabla 24: Hardware requerido

Cantidad	Descripción	Observación
1	Impresora	Impresión de los reportes
1	Módem de 128 Kbps	Para realizar las pruebas con conexión a la red

Autor: Marcela Cuello Olmedo

Software requerido: Los recursos software adicionales que se requieren para el desarrollo de nuestro sistema software:

Tabla 25: Software requerido

Nombre	Descripción	Estado	Observación
IReport	Programa para generar reportes de la aplicación.	Legal	Ninguna.
IDE NetBeans	Herramienta de desarrollo	Libre	Soporte JSP.
MySQLFront	DBMS	Libre	Ninguna

	Software de conexión a internet para el módem		Ninguna

Autor: Marcela Cuello Olmedo

MySQL es uno de los gestores de bases de datos más popular que existe. Actualmente, y tras la operación de compra de Sun Microsystems, pertenece a Oracle Corporation. MySQL ha sido elegido para este proyecto frente al resto de alternativas dada la familiaridad del equipo con ella así como la cantidad de recursos y documentación existente en la red (aunque tampoco las alternativas planteadas escasean en documentación y recursos, está claro que es MySQL el gestor de base de datos más extensamente utilizado)[31].

Recurso humano requerido: Como recurso humano necesario para el desarrollo del sistema está el siguiente:

Tabla 26: Recurso humano requerido

Función	Formación	Experiencia
Desarrollador	Estudiante de la Escuela de Ingeniería en Sistemas.	Desarrollador web, html, jsp, java script.
Administrador de Bases de Datos	Estudiante de la Escuela de Ingeniería en Sistemas.	Transact SQL.

Autor: Marcela Cuello Olmedo

b) Factibilidad operativa

Recurso Humano: Usuarios directos (Quienes van a manejar el sistema).

Tabla 27: Recurso Humano

Nombre	Función
Administrador	Administrador de la plataforma tecnológica Unach
Encuestado	Profesional o egresado que proporciona información en la encuesta

Autor: Marcela Cuello Olmedo

c) Factibilidad legal

Para el seguimiento de Profesionales y egresados de la Facultad de Ingeniería, la Universidad Nacional de tiene reservado todos los derechos de autor, cualquier copia parcial o total debe ser autorizada por los autores según la Ley de Propiedad Intelectual. Según lo estipula en la Legislación Ecuatoriana que consta en el código penal con la Ley

[31] *Romeu, J, A., & Santabárbara R, I., & Jiménez, G., R. Proyecto de Sistemas Informáticos: SSII TunnelBroker. Págs. 62*

de Comercio Electrónico, firmas electrónicas y mensajes de datos. Esto asegura que el sistema SIGEGE que se está proponiendo es legalmente factible, ya que no existen impedimentos para que se lleve a cabo.

d) Factibilidad económica

Tabla 28: Factibilidad económica

Costos	Valor
Costos	**$1771,00**
Costos de desarrollo	$1771,00
Costo de personal técnico	$00,00
Costo de hardware y software	$1140.00
Costo de suministros	$631.50
Costo de instalación	**$ 00,00**
Costos de personal para instalación	$ 00,00
Costos de operación	**$ 00,00**
Costos de personal de operación	$ 00,00
Costos de mantenimiento hardware	$ 00,00
Costos de suministros de operación	$ 00,00

Autor: Marcela Coello Olmedo

3.2.10. Planificación y análisis de riesgos

El análisis de riesgo, también conocido como evaluación de riesgo o PHA por sus siglas en inglés Process Hazards Analysis, es el estudio de las causas de las posibles amenazas y probables eventos no deseados y los daños y consecuencias que éstas puedan producir.

El principal factor crítico en el desarrollo de un sistema software es el análisis de riesgos, esta es una estrategia que se utiliza para gestionar los riesgos de una manera efectiva y de esta forma evitar que dichos riesgos se transformen en problemas.

Tabla 29: Nomenclatura Riesgos

Siglas	Descripción
RP	Riesgo del proyecto
RT	Riesgo técnico
RN	Riesgo del negocio

Autor: Marcela Cuello Olmedo

3.2.10.1. Identificación de riesgos

Tabla 30: Identificación de riesgos

Riesgo	Descripción del riesgo	Categoría	Consecuencia
R1	Cambio de asesor técnico del proyecto (director de tesis).	RP	Cambio de proyecto. Pérdida de tiempo

R2	Inexistencia del hardware requerido para la implementación del proyecto en la escuela de Posgrado.	RP,RT	Retraso del proyecto Riesgo de no poder implementar el sistema.
R3	El presupuesto asignado no fue suficiente para culminar el proyecto.	RN	Retraso del proyecto hasta que se obtengan los nuevos recursos.
R4	Falta de formación del equipo de desarrollo en el manejo de las herramientas.	RP	Retraso en la realización del proyecto.
R5	Pérdida de apoyo del nivel estratégico de la institución.	RN	Desconfianza en el nivel estratégico. No poder continuar con el proyecto.
R6	Cambio del responsable del proyecto.	RN	Implementación desordenada de la aplicación.

Autor: Marcela Coello Olmedo

3.2.10.2. Categorizar riesgos

Tabla 31: Probabilidad de riesgos

Porcentaje	Descripción	Valor
1% - 33%	Baja	1
34%- 67%	Media	2
68%- 99%	Alta	3

Autor: Marcela Coello Olmedo

Tabla 32: Impacto del riesgo

Impacto	Costo	Retraso	Impacto Técnico	Valor
Bajo	< 1 %	1 semana	Ligero efecto en el desarrollo del proyecto.	1
Moderados	< 5%	2 semanas	Moderado efecto en el desarrollo del proyecto.	2
Alto	< 10%	1 mes	Severo efecto en el desarrollo del proyecto	3
Crítico	> 10%	> 1 mes	El proyecto no puede ser culminado.	4

Tabla 33: Exposición de riesgos

Impacto / Probabilidad	Baja=1	Moderado=2	Alto=3	Crítico=4
Alta=3	3	6	9	12
Media=2	2	4	6	8
Baja=1	1	2	3	4

Autor: Marcela Coello Olmedo

Tabla 34: Código de colores según la exposición del riesgo

Exposición del Riesgo	Valor	Color
Baja	1 o 2	Verde
Media	3 o 4	Amarillo
Alta	>= 6	Rojo

Tabla 35: Determinación de la Prioridad de Riesgos

Riesgo	Probabilidad			Impacto		Exposición al riesgo	
	%	Valor	Proba.	Valor	Impacto	Valor	Exposición
R2	40%	2	Media	3	Alto	6	Alta
R5	30%	2	Media	3	Alto	6	Alta
R3	40%	2	Media	3	Alto	6	Alta
R1	70%	1	Baja	3	Alto	3	Media
R6	20%	1	Baja	3	Alto	3	Media
R4	10%	1	Baja	2	Media	2	Baja

Autor: Marcela Coello Olmedo

3.2.10.3. Hojas de riesgo

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R2		FECHA: 06-10-2013	
ID. DEL RIESGO: R1		FECHA: 05-10-2013	
Probabilidad: Media Probabilidad: Baja	Impacto: Alto Impacto: Alto	Exposición: Alta Exposición: Media	Prioridad: 1 Prioridad: 1
Valor: 2 Valor: 1	Valor: 3 Valor: 3	Valor: 6 Valor: 3	
DESCRIPCIÓN: Inexistencia del hardware requerido para la implementación del proyecto en la escuela de Posgrado. DESCRIPCIÓN: Cambio de asesor técnico del proyecto (director de tesis).			
REFINAMIENTO: REFINAMIENTO:			
Causas: La tecnología que se usara requiere un equipo de cómputo adecuado para un buen rendimiento. **Causas:** Solicitud de renuncia, por parte del profesor, enfermedad o disposición de las autoridades de la escuela.			
Consecuencias: Retraso del proyecto y riesgo de no poder implementar el sistema. **Consecuencias:** Cambio de proyecto, pérdida de tiempo.			
REDUCCIÓN: REDUCCIÓN:			
Adquirir equipos de excelente tecnología para la implementación del sistema. Cooperar entre los desarrolladores del proyecto de tesis. Informar a las autoridades de la Facultad de Ingeniería de la situación del proyecto.			
SUPERVISIÓN: SUPERVISIÓN:			
Solicitar al director de la escuela de Posgrado la adquisición de los equipos para la implementación del sistema. Desde el inicio del proyecto se debe haber acuerdos en la planificación del proyecto, para prevenir problemas a futuro.			
GESTIÓN: GESTIÓN:			
Lograr obtener las características necesarias que deben tener los equipos para el correcto funcionamiento del sistema. Agotar todos los medios necesarios para lograr la permanencia del asesor dentro del proyecto.			
ESTADO ACTUAL:			
X			

ESTADO ACTUAL: Fase de reducción iniciada Fase de Supervisión iniciada Gestionando el riesgo	 X ☐
RESPONSABLES: Marcela Cuello Olmedo	
RESPONSABLES: Marcela Cuello Olmedo	

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R3		FECHA: 08-10-2013	
Probabilidad: Media Valor: 2	Impacto: Alta Valor: 3	Exposición: alta Valor: 6	Prioridad: 1
DESCRIPCIÓN: El presupuesto asignado no fue suficiente para culminar el proyecto.			
REFINAMIENTO: **Causas:** Planificación incorrecta de los recursos disponibles para la ejecución del proyecto o mala inversión de los mismos para llevar a cabo tareas no previstas. **Consecuencias:** Retraso del proyecto hasta que se obtengan los nuevos recursos.			
REDUCCIÓN: Emplear los recursos económicos solo en lo que necesariamente fue planificado No invertir recursos en tareas no planificadas evitando que se agoten en la marcha del proyecto.			
SUPERVISIÓN: Controlar gastos efectuados que sean únicamente los planificados.			
GESTIÓN: Llevar un gasto controlado para evitar el desperdicio del recurso económico.			
ESTADO ACTUAL: Fase de reducción iniciada Fase de Supervisión iniciada Gestionando el riesgo	 X ☐		
RESPONSABLES: Marcela Cuello Olmedo.			

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R4		FECHA: 15-10-2013	
Probabilidad: Baja Valor: 1	Impacto: Media Valor: 2	Exposición: Media Valor: 2	Prioridad: 1
DESCRIPCIÓN: Falta de formación del equipo de desarrollo en el manejo de las herramientas.			
REFINAMIENTO: **Causas:** Desconocimiento en el de las herramientas, utilización de nuevas tectologías **Consecuencias:** Retraso en la realización del proyecto			
REDUCCIÓN: Realizar cursos, recopilar información sobre las herramientas, tener un asesoramiento continuo.			
SUPERVISIÓN: Al inicio del proyecto hay que conocer las herramientas a emplear para tener la posibilidad de investigar sus funcionalidades y opciones de trabajo.			
GESTIÓN: Preparar al equipo de desarrollo mediante cursos, bibliografía y consultas con personas que dominen las herramientas.			
ESTADO ACTUAL: Fase de reducción iniciada [X] Fase de Supervisión iniciada [ ] Gestionando el riesgo [ ]			
RESPONSABLES: Marcela Cuello Olmedo.			

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R5		FECHA: 18-10-2013	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alto Valor: 6	Prioridad: 1
DESCRIPCIÓN: Pérdida de apoyo del nivel estratégico de la institución.			
REFINAMIENTO: **Causas:** Falta de comunicación por parte del Jefe del proyecto con el nivel estratégico acerca del avance del proyecto. **Consecuencias:** Desconfianza en el nivel estratégico con la realización del proyecto, No poder continuar con el proyecto.			
REDUCCIÓN: Mantener una comunicación constante con el nivel estratégico de la empresa.			
SUPERVISIÓN: Presentar los reportes acerca de los beneficios que el proyecto que se encuentra en ejecución traerá a la Facultad de Ingeniería de la Unach una vez que este sea terminado.			
GESTIÓN: Informar periódicamente al nivel estratégico sobre el avance del proyecto.			
ESTADO ACTUAL: Fase de reducción iniciada [X] Fase de Supervisión iniciada [ ] Gestionando el riesgo [ ]			
RESPONSABLES: Marcela Cuello Olmedo.			

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R6		FECHA: 22-10-2014	
Probabilidad: Baja Valor: 1	Impacto: Alto Valor: 3	Exposición: Media Valor: 3	Prioridad: 1
DESCRIPCIÓN: Cambio del responsable del proyecto.			
REFINAMIENTO: **Causas:** Puede suceder en casos extremos, como fallecimiento, viaje obligado o desacuerdo con otro miembro del proyecto. **Consecuencias:** Retraso en la entrega, o que no se realice el proyecto			
REDUCCIÓN: Mantener una relación aceptable entre los miembros del equipo de trabajo, en caso de desacuerdo.			
SUPERVISIÓN: Ser más responsables con las tareas encomendadas. Cumplir con el trabajo.			
GESTIÓN: Tener comunicación permanente con el equipo de trabajo, y comunicar de todos los avances del proyecto que se realice.			
ESTADO ACTUAL: Fase de reducción iniciada [X] Fase de Supervisión iniciada [ ] Gestionando el riesgo [ ]			
RESPONSABLES: Marcela Cuello Olmedo.			

3.3. Definición de casos de usos

Un caso de uso se puede describir como la historia de un sistema, por lo cual es una excelente técnica para entender y describir requerimientos.[32] (Cockburn, 2006)

[32] *Cockburn, Alistar. Use Case Fundamentals. Pág. 1.*

La facultad de Ingeniería de la Universidad Nacional de Chimborazo conjuntamente con el sistema SYGEGE tiene como objetivo realizar un seguimiento a los profesionales y egresados de la facultad. Para dicho seguimiento se requiere proporcionar un ambiente web que esté al alcance de los actores mencionados, utilizar la información más relevante proporcionada en las encuestas aplicadas para obtener resultados que ayuden en la toma de decisiones en mejora de oportunidades de estudio, capacidades laborales y mejoramiento profesional de la Unach. Se cuenta con 2 módulos de trabajo para los actores, administrador y encuestado, que contemplan las actividades siguientes:

Modulo Encuestado: Ingresar Información, Registrar datos de encuestas, Loguearse en el sistema, Consultar información, Modificar Información.

Modulo Administrador: Loguearse en el sistema, Eliminar encuestado, Generar reportes.

SYGEGE: El sistema de seguimiento genera formularios de encuestas, permite terminar una encuesta y tabula los datos proporcionados por el encuestado (Informante).

3.3.1. Diagramas de casos de usos

3.3.1.1. Módulo Encuestado

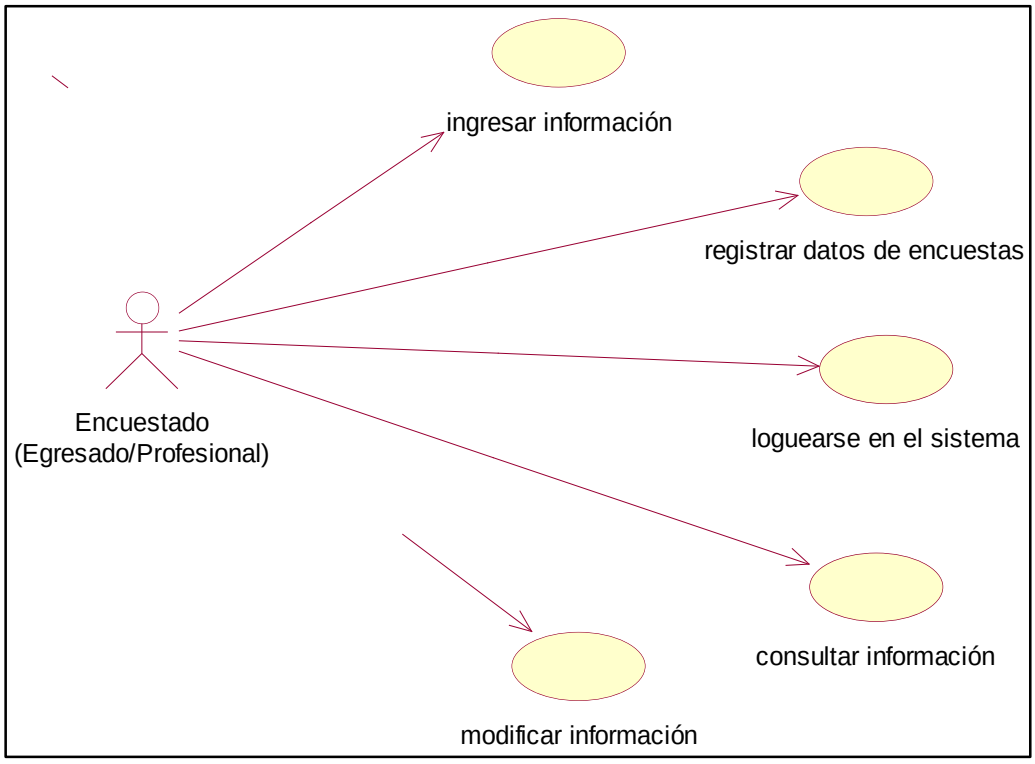


Figura 20: Casos de uso Módulo Encuestado
Autor: Marcela Cuello Olmedo

Especificación de casos de uso

GE- CU- 001: Ingresar Información

DESCRIPCIÓN:	Permite ingresar un encuestado
ACTORES:	Encuestado
DISPARADOR:	Ingreso a la consola del Encuestado.
PRECONDICIÓN:	El actor se encuentra en la página de inicio.
POSTCONDICIÓN:	Datos de encuestado registrados.

FLUJO BÁSICO:

1. El caso de uso comienza con el ingreso por parte del actor: Encuestado a la opción Encuestas.
2. El sistema muestra la opción para Ingresar información personal, profesional y laboral.
3. El actor: Encuestado, ingresa: nombre, edad, cédula de identidad, teléfono de domicilio, teléfono celular, empresa o institución que trabaja, ciudad de trabajo, dirección trabajo, teléfono de trabajo, trabajo que desempeña, correo electrónico, dirección padre (familiares), teléfono padre (familiares).
4. El actor: Encuestado, debe marcar su género (femenino, masculino).
5. El actor: Encuestado, debe indicar su estado civil (soltero, casado, divorciado, otro).
6. El actor: Encuestado, debe indicar su año de graduación.
7. El actor: Encuestado, debe indicar su nota de graduación (entre 7 y 8, entre 8 y 9, entre 9 y 10).
8. El sistema guarda la información.
9. El caso de uso termina con el ingreso del encuestado.

FLUJO ALTERNATIVO 1:

1. El caso de uso comienza con el ingreso por parte del actor: Encuestado a la opción Encuestas.
2. El sistema muestra la opción para Ingresar información personal, profesional y laboral.
3. El actor: Encuestado, ingresa: nombre, edad, cédula de identidad, teléfono de domicilio, teléfono celular, empresa o institución que trabaja, ciudad de trabajo, dirección trabajo, teléfono de trabajo, trabajo que desempeña, correo electrónico, dirección padre (familiares), teléfono padre (familiares).
4. SI el actor no registra información ENTONCES
5. El caso de uso termina sin el ingreso del encuestado.

FLUJO ALTERNATIVO 2:

1. El caso de uso comienza con el ingreso por parte del actor: Encuestado a la opción Encuestas.
2. El sistema muestra la opción para Ingresar información personal, profesional y laboral.
3. El actor: Encuestado, ingresa: nombre, edad, cédula de identidad, teléfono de domicilio, teléfono celular, empresa o institución que trabaja, ciudad de trabajo, dirección trabajo, teléfono de trabajo, trabajo que desempeña, correo electrónico, dirección padre (familiares), teléfono padre (familiares).
   3.1 SI no registra datos completos ENTONCES
   Mensaje: "Falta información registrar"
4. El actor: Encuestado, debe marcar su género (femenino, masculino).
   4.1 SI no marca su genere ENTONCES
   Mensaje: "Debe marcar género"

5. El actor: Encuestado, debe indicar su estado civil (soltero, casado, divorciado, otro).
   5.1 SI no marca su estado civil ENTONCES
   Mensaje: "Debe marcar su estado civil"
6. El actor: Encuestado, debe indicar su año de graduación.
   6.1 SI no registra su año de graduación ENTONCES
   Mensaje: "Debe registrar año de graduación"

7. El actor: Encuestado, debe marcar su nota de graduación (entre 7 y 8, entre 8 y 9, entre 9 y 10).
   7.1 SI no marca su nota de graduación ENTONCES
   Mensaje: "Debe marcar nota de graduación"
8. El sistema guarda la información.

El caso de uso termina con el ingreso del encuestado.

Autor: *Marcela Cuello Olmedo*

GE- CU- 002: Registrar datos de encuestas

DESCRIPCIÓN:	Permite registrar información de la encuesta
ACTORES:	Encuestado
DISPARADOR:	Ingreso a la consola del Encuestado.
PRECONDICIÓN:	El actor ha ingresado satisfactoriamente al sistema. Ha ingresado su información personal, profesional y laboral.
POSTCONDICIÓN:	Encuesta realizada

FLUJO BÁSICO:

1. El caso de uso comienza con la generación del formulario de la encuesta a partir de haber ingresado la información personal, profesional y laboral del encuestado.
2. El sistema muestra el formulario de encuestas "Conocimientos generales, habilidades y destrezas."
3. El actor selecciona la respuesta a la pregunta planteada.
4. El sistema muestra el formulario de encuesta "Grado de satisfacción con su experiencia formativa en la Unach".
5. El actor selecciona la respuesta a la pregunta planteada.
6. El sistema muestra el formulario de encuesta "Postgrado y desarrollo profesional".
7. El actor selecciona la respuesta a la pregunta planteada.
8. El sistema muestra el formulario de encuesta "Preguntas globales"
9. El actor selecciona la respuesta a la pregunta planteada.
10. El caso de uso termina con la encuesta realizada.

FLUJO ALTERNATIVO:

1. El caso de uso comienza con la generación del formulario de la encuesta a partir de haber ingresado la información personal, profesional y laboral del encuestado.
2. El sistema muestra el formularios de encuestas
   SI el actor no contesta a la encuesta ENTONCES
3. El caso de uso termina sin la encuesta realizada.

FLUJO ALTERNATIVO 1:

1. El caso de uso comienza con la generación del formulario de la encuesta a partir de haber ingresado la información personal, profesional y laboral del encuestado.

2. El sistema muestra el formulario de encuestas “Conocimientos generales, habilidades y destrezas.”
3. El actor selecciona la respuesta a la pregunta planteada.
   3.1 SI el actor no selecciona respuesta ENTONCES
   Mensaje: “Debe seleccionar una respuesta”
4. El sistema muestra el formulario de encuesta “Grado de satisfacción con su experiencia formativa en la Unach”.
5. El actor selecciona la respuesta a la pregunta planteada.
   5.1 SI el actor no selecciona respuesta ENTONCES
   Mensaje: “Debe seleccionar una respuesta”
6. El sistema muestra el formulario de encuesta “Postgrado y desarrollo profesional”.
7. El actor selecciona la respuesta a la pregunta planteada.
   7.1 SI el actor no selecciona respuesta ENTONCES
   Mensaje: “Debe seleccionar una respuesta”
8. El sistema muestra el formulario de encuesta “Preguntas globales”
9. El actor selecciona la respuesta a la pregunta planteada.
   9.1 SI el actor no selecciona respuesta ENTONCES
   Mensaje: “Debe seleccionar una respuesta”
10. El caso de uso termina con la encuesta realizada.

Autor: *Marcela Cuello Olmedo*

GE- CU- 003: Loguearse en el sistema

DESCRIPCIÓN:	Permite validar el ingreso de un usuario del sistema
ACTORES:	Encuestado
DISPARADOR:	Ingreso a la consola del Encuestado, opción ingresar al sistema.
PRECONDICIÓN:	El actor existe en el sistema.
POSTCONDICIÓN:	El actor ha ingresado satisfactoriamente al sistema.

FLUJO BÁSICO:

1. El caso de uso comienza con el ingreso por parte del actor: Encuestado al sistema de SYGEGE, opción Consultar Información.
2. El sistema muestra la opción para Loguearse en el sistema.
3. El actor: Encuestado ingresa el usuario y la clave.
4. El sistema devuelve información personal, profesional y laboral del actor.
5. El caso de uso termina con el ingreso al sistema.

FLUJO ALTERNATIVO:

1. El caso de uso comienza con el ingreso por parte del actor: Encuestado al sistema de SYGEGE, opción Consultar Información.
2. El sistema muestra la opción para Loguearse en el sistema.
3. El actor: Encuestado ingresa el usuario y la clave.
4. SI no existe el encuestado ENTONCES
5. El caso de uso termina sin el ingreso del encuestado al sistema.

Autor: *Marcela Cuello Olmedo*

GE- CU- 004: Consultar información

DESCRIPCIÓN:	Permite consultar información personal, profesional y laboral.

ACTORES:	Encuestado
DISPARADOR:	Consola de encuestado.
PRECONDICIÓN:	El actor ha ingresado satisfactoriamente al sistema. Se encuentra ingresado el encuestado.
POSTCONDICIÓN:	Se ha consultado la información del encuestado

FLUJO BÁSICO:

1. El caso de uso comienza con el ingreso por parte del actor: Encuestado a la opción consultar información.
2. El sistema muestra formulario de información.
3. El actor: Encuestado consulta información personal, profesional y laboral.
4. El caso de uso termina con la consulta realizada.

Autor: *Marcela Cuello Olmedo*

GE- CU- 005: Modificar Información

DESCRIPCIÓN:	Permite modificar información del encuestado
ACTORES:	Encuestado
DISPARADOR:	Ingreso a la consola del Encuestado, opción Modificar.
PRECONDICIÓN:	El actor ha ingresado satisfactoriamente al sistema. Se encuentra registrado el encuestado.
POSTCONDICIÓN:	Se ha actualizado información del encuestado

FLUJO BÁSICO:

1. El caso de uso comienza con el ingreso por parte del actor: Encuestado a la opción Modificar.
2. El sistema muestra la opción para Modificar información.
3. El sistema permite la actualización de datos permitidos.
4. El actor Encuestado modifica información: personal, profesional y laboral.
5. SI el estudiante registra información ENTONCES; se modifica: nombre, edad, cédula de identidad, teléfono de domicilio, teléfono celular, empresa o institución que trabaja, ciudad de trabajo, dirección trabajo, teléfono de trabajo, trabajo que desempeña, correo electrónico, dirección padres (familiares), teléfono padres (familiares), usuario y clave.
6. El actor: Encuestado, actualiza género (femenino, masculino).
7. El actor: Encuestado, actualiza su estado civil (soltero, casado, divorciado, otro).
8. El actor: Encuestado, actualiza su año de graduación.
9. El actor: Encuestado, actualiza nota de graduación (entre 7 y 8, entre 8 y 9, entre 9 y 10).
10. El sistema guarda la información.
    El caso de uso termina con el ingreso del encuestado.
11. El sistema guarda cambios realizados.
12. El caso de uso termina con la actualización de la información del encuestado.

FLUJO ALTERNATIVO:

1. El caso de uso comienza con el ingreso por parte del actor: Encuestado a la opción Modificar.
2. El sistema muestra la opción para Modificar información.
3. El sistema permite la actualización de datos permitidos.
4. El actor Encuestado modifica información: personal, profesional y laboral.
5. SI el estudiante registra información ENTONCES; se modifica: nombre, edad, cédula de identidad, teléfono de domicilio, teléfono celular, empresa o institución que trabaja, ciudad de trabajo, dirección trabajo, teléfono de trabajo,

trabajo que desempeña, correo electrónico, dirección padres (familiares), teléfono padres (familiares), usuario y clave.
6. El actor: Encuestado, actualiza género (femenino, masculino).
7. El actor: Encuestado, actualiza su estado civil (soltero, casado, divorciado, otro).
8. El actor: Encuestado, actualiza su año de graduación.
9. El actor: Encuestado, actualiza nota de graduación (entre 7 y 8, entre 8 y 9, entre 9 y 10).
10. SI el encuestado no modifica información ENTONCES
11. El caso de uso termina sin la modificación de la información.

Autor: *Marcela Cuello Olmedo*

3.3.1.2. Módulo Administrador

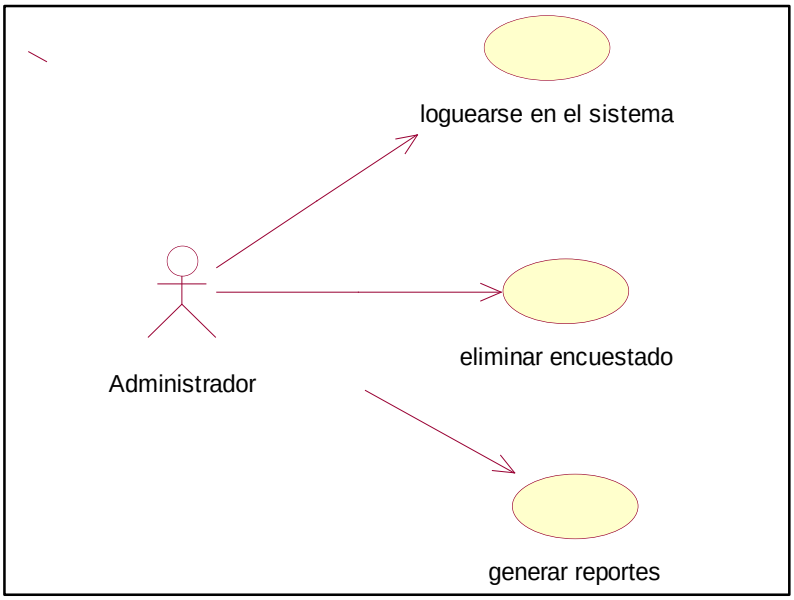


Figura 21: Casos de uso Módulo Administrador
Autor: Marcela Cuello Olmedo

Especificación de casos de uso

GE- CU- 006: Loguearse en el sistema

DESCRIPCIÓN:	Permite validar el ingreso de un usuario del sistema
ACTORES:	Administrador
DISPARADOR:	Ingreso a la consola del Administrador, opción tabulación
PRECONDICIÓN:	El actor existe en el sistema.
POSTCONDICIÓN:	El actor ha ingresado satisfactoriamente al sistema.
FLUJO BÁSICO: 6. El caso de uso comienza con el ingreso por parte del actor: Administrador al sistema de SYGEGE, opción Tabulación. 7. El sistema muestra la opción para Loguearse en el sistema. 8. El actor: Administrador ingresa el usuario y la clave. 9. El caso de uso termina con el ingreso al sistema.	
FLUJO ALTERNATIVO:	

6. El caso de uso comienza con el ingreso por parte del actor: Administrador al sistema de SYGEGE, opción Tabulación. 7. El sistema muestra la opción para Loguearse en el sistema. 8. El actor: Administrador ingresa el usuario y la clave. 9. SI no existe el actor ENTONCES 10. El caso de uso termina sin el ingreso del actor al sistema.

Autor: Marcela Cuello Olmedo

GE- CU- 007: Eliminar encuestado

DESCRIPCIÓN:	Permite eliminar un informante encuestado del sistema, siempre y cuando no haya registrado la encuesta.
ACTORES:	Administrador
DISPARADOR:	Ingreso a la consola del Administrador, opción Eliminar Encuestado.
PRECONDICIÓN:	El encuestado existe en el sistema, el actor ha ingresado al sistema satisfactoriamente.
POSTCONDICIÓN:	El encuestado ha sido eliminado del sistema.
FLUJO BÁSICO: 1. El caso de uso comienza con el ingreso por parte del actor Administrador a la opción Eliminar Encuestado " 2. El sistema lista a los encuestados que no han registrado encuesta. 3. El actor Administrador selecciona el usuario a eliminar. 4. El actor confirma la acción eliminar. 4. El caso de uso termina cuando el sistema elimina a un informante encuestado.	
FLUJO ALTERNATIVO 1. El caso de uso comienza con el ingreso por parte del actor Administrador a la opción Eliminar Encuestado " 2. El sistema listo a los encuestados que no han registrado encuesta. 3. SI el actor Administrador no selecciona el usuario a eliminar. 4. El caso de uso termina sin la eliminación de encuestado.	

Autor: Marcela Cuello Olmedo

GE- CU- 008: Generar reportes

DESCRIPCIÓN:	Permite generar información procesada de cada una de las preguntas de la encuesta realizadas a egresados y estudiantes de la Facultad de Ingeniería de la Unach
ACTORES:	Administrador
DISPARADOR:	Ingreso a la consola del Administrador, Reportes.
PRECONDICIÓN:	El administrador existe en el sistema, el actor ha ingresado al sistema satisfactoriamente.
POSTCONDICIÓN:	Se ha generado un reporte
FLUJO BÁSICO: 1. El caso de uso comienza con el ingreso por parte del actor Administrador a la opción “Reportes " 2. El sistema lista los reportes disponibles por pregunta de encuesta. 3. El actor Administrador selecciona reporte a generar. 4. El sistema genera un documento PDF con la información procesada. 4. El caso de uso termina cuando el actor visualiza la información.	
FLUJO ALTERNATIVO	

1. El caso de uso comienza con el ingreso por parte del actor Administrador a la opción “Generar Reportes "
2. El sistema lista los reportes disponibles por pregunta de encuesta.
3. El actor Administrador selecciona reporte a generar.
4. SI el actor no selecciona algún reporte
5. El caso de uso termina sin la generación del reporte.

Autor: Marcela Cuello Olmedo

3.4. Diagrama de secuencias del sistema

3.4.1. Modulo Encuestado

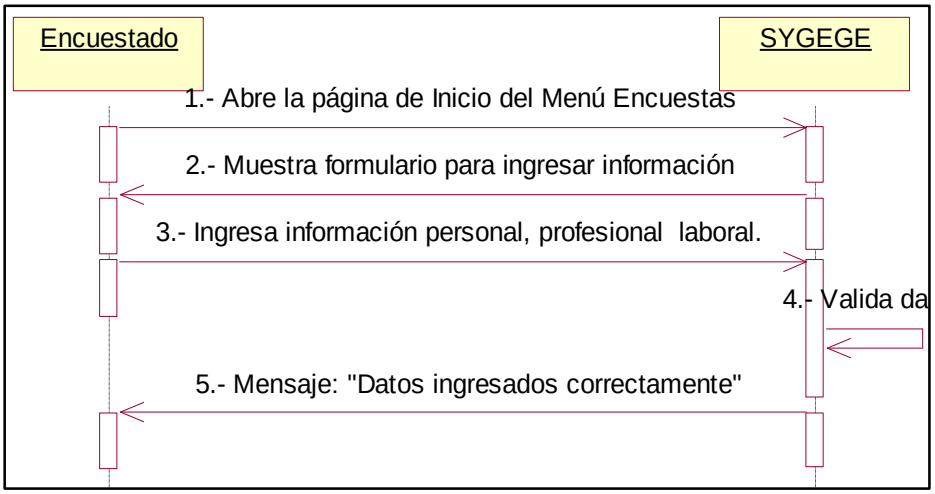


Figura 22: Ingresar Información
Autor: Marcela Cuello Olmedo

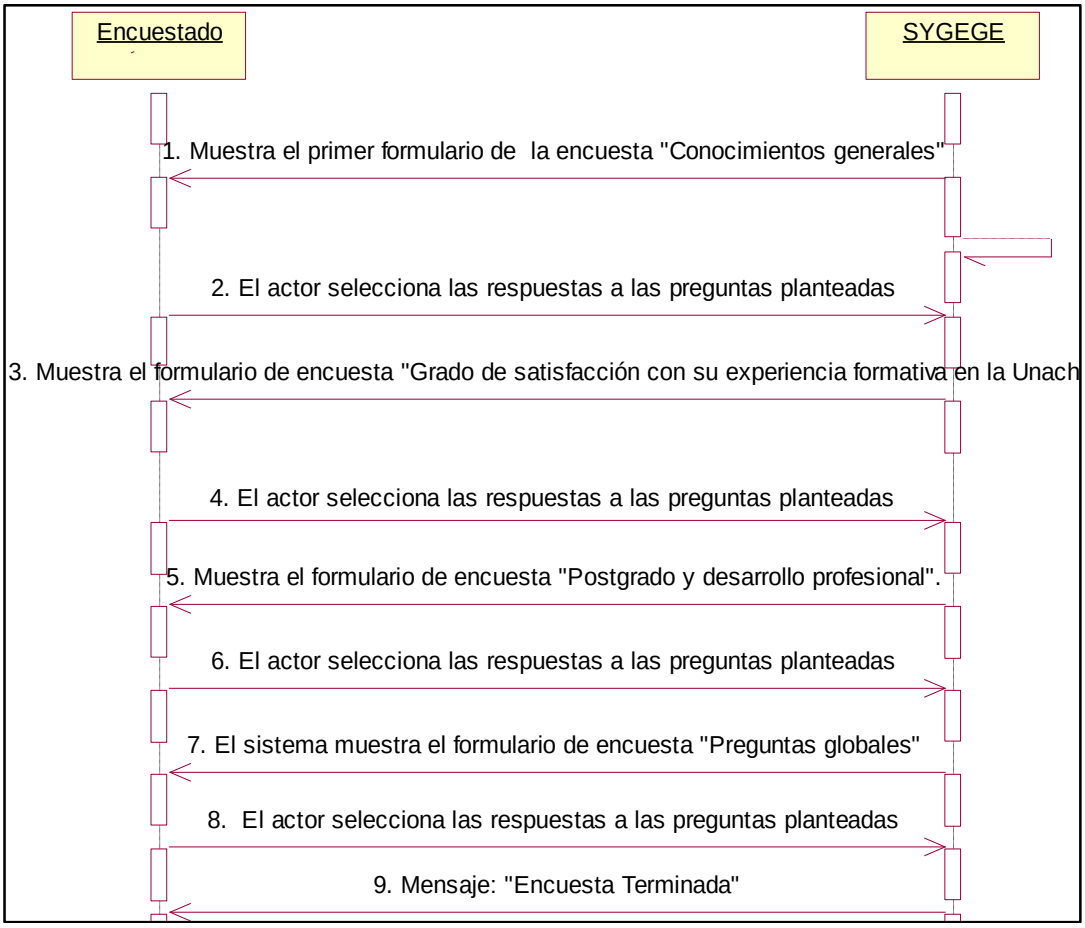


Figura 23: Registrar datos de encuestas
Autor: Marcela Cuello Olmedo

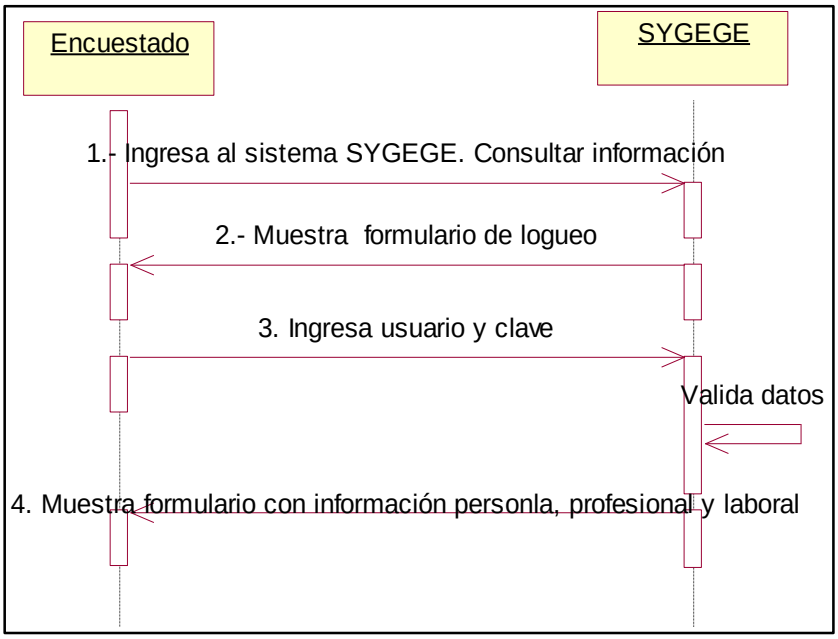


Figura 24: Loguearse en el sistema
Autor: Marcela Cuello Olmedo

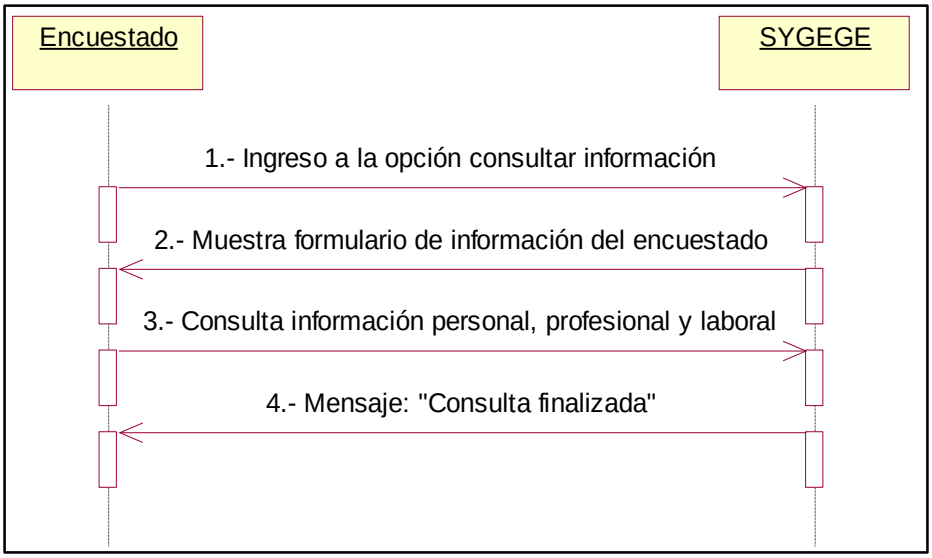


Figura 25: Consultar Información
Autor: Marcela Cuello Olmedo

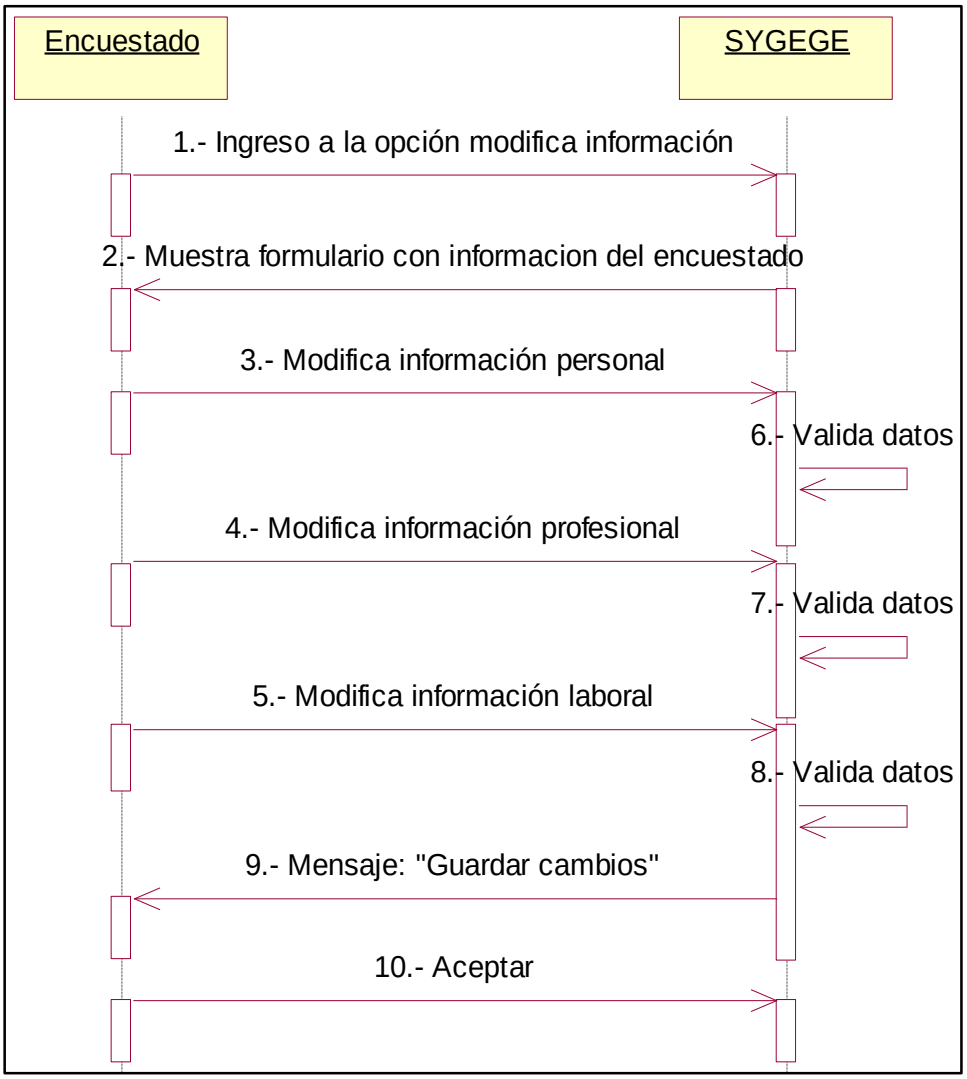


Figura 26: Modificar Información
Autor: Marcela Cuello Olmedo

3.4.2. Modulo Administrador

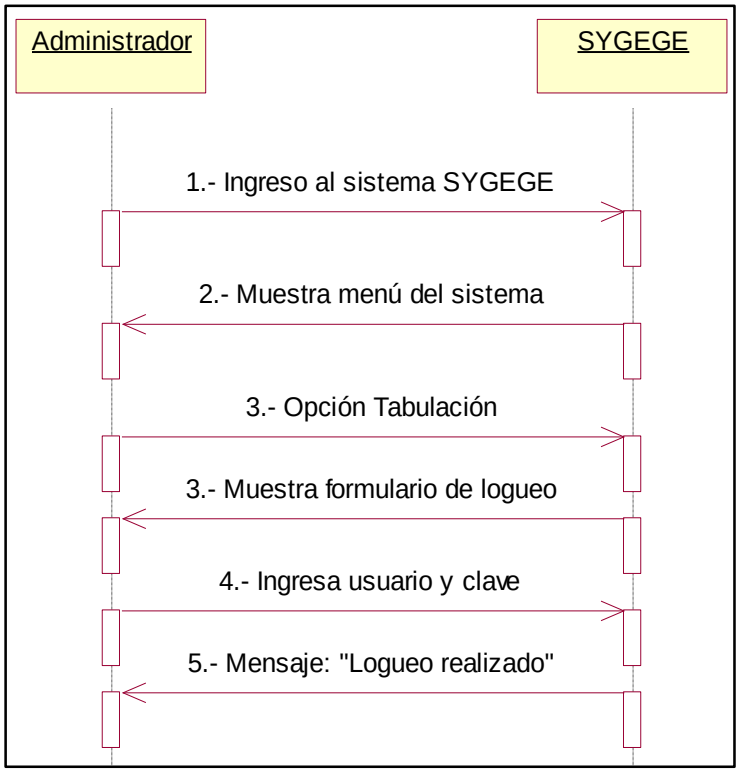


Figura 27: Loguearse en el sistema
Autor: Marcela Cuello Olmedo

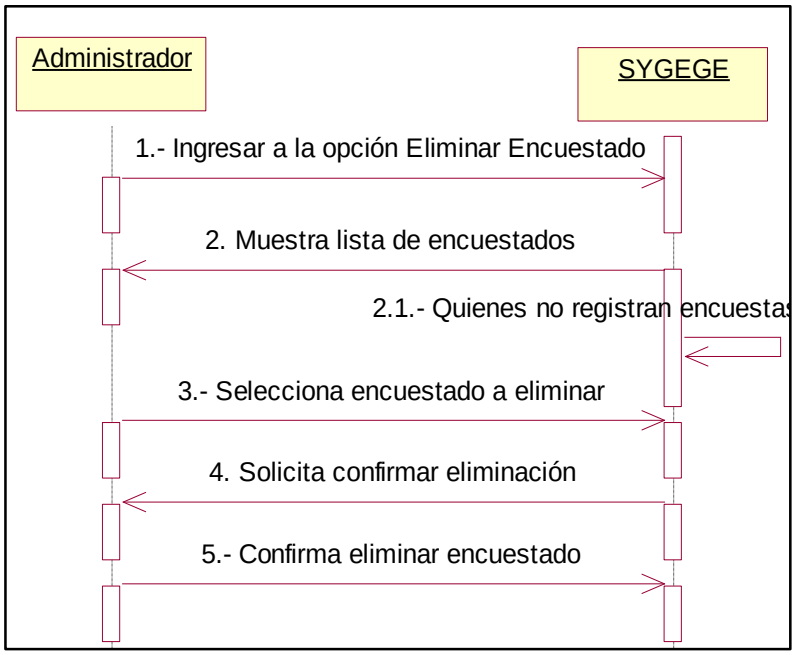


Figura 28: Eliminar encuestado
Autor: Marcela Cuello Olmedo

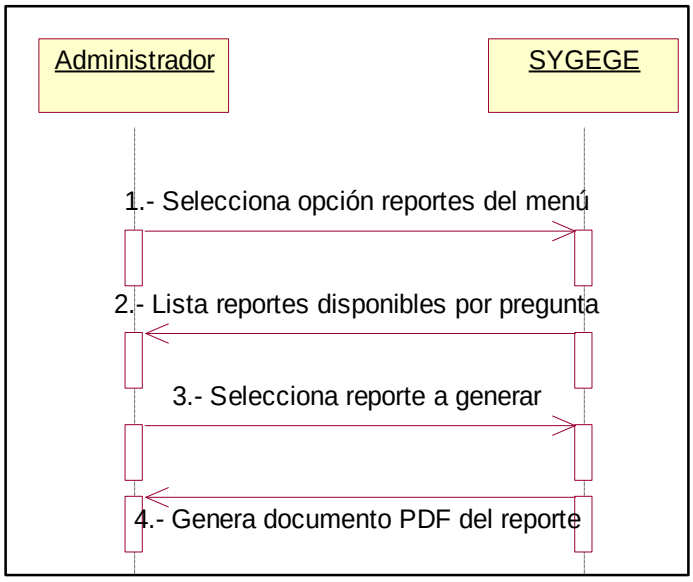


Figura 29: Generar reportes
Autor: Marcela Cuello Olmedo

3.5. Diagrama de actividades

3.5.1. Módulo Encuestado

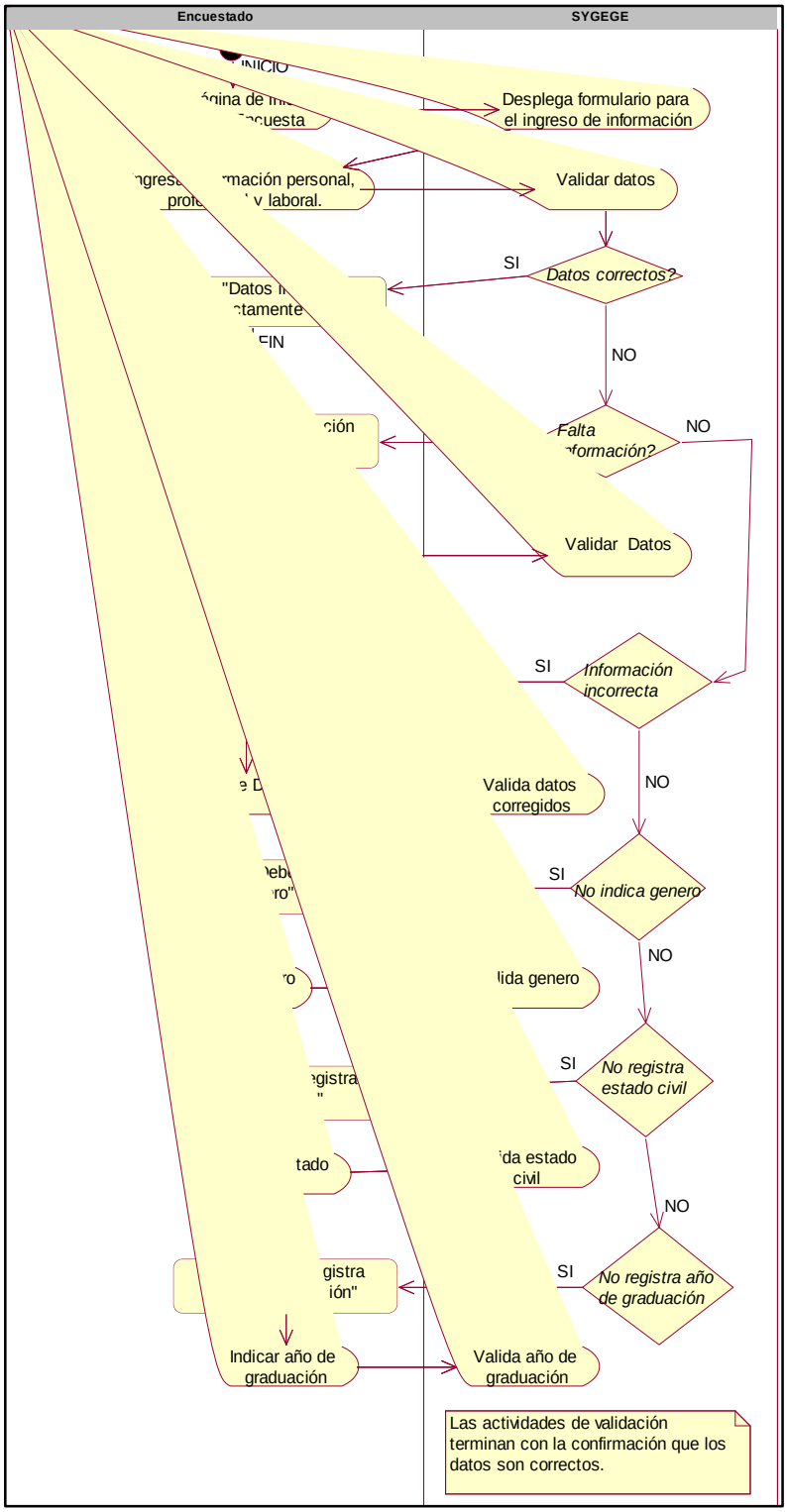


Figura 30: Ingresar información
Autor: Marcela Cuello Olmedo

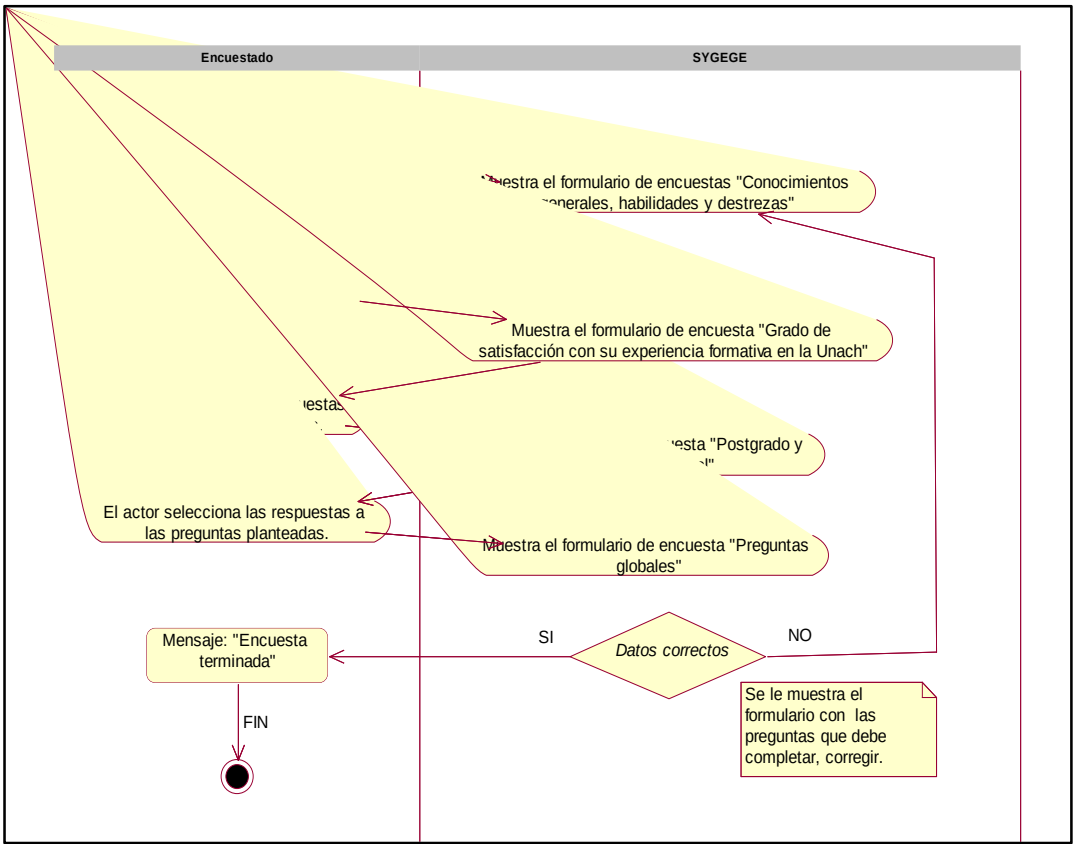


Figura 31: Registrar datos de encuestas
Autor: Marcela Cuello Olmedo

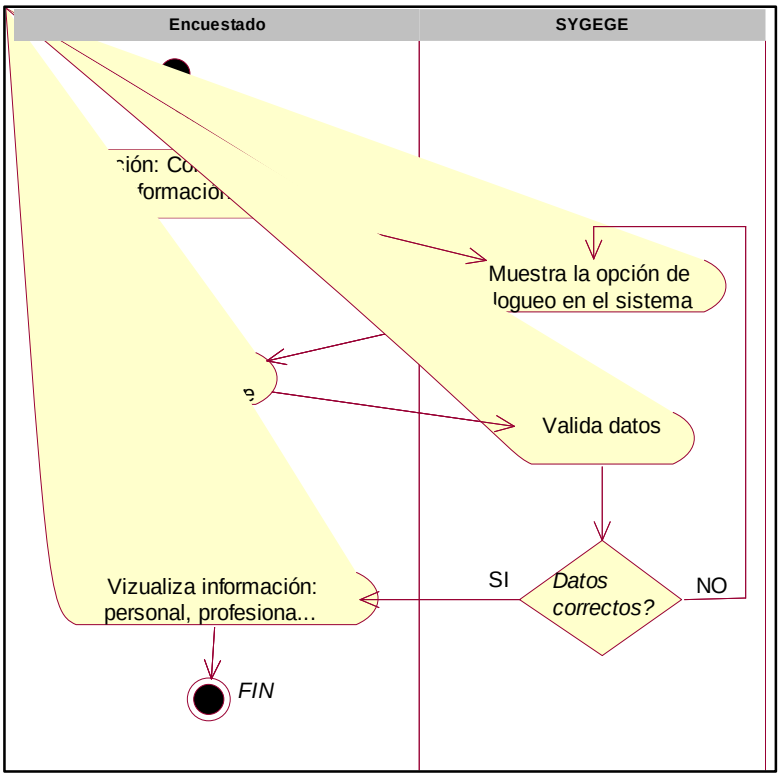


Figura 32: Loguearse en el sistema
Autor: Marcela Cuello Olmedo

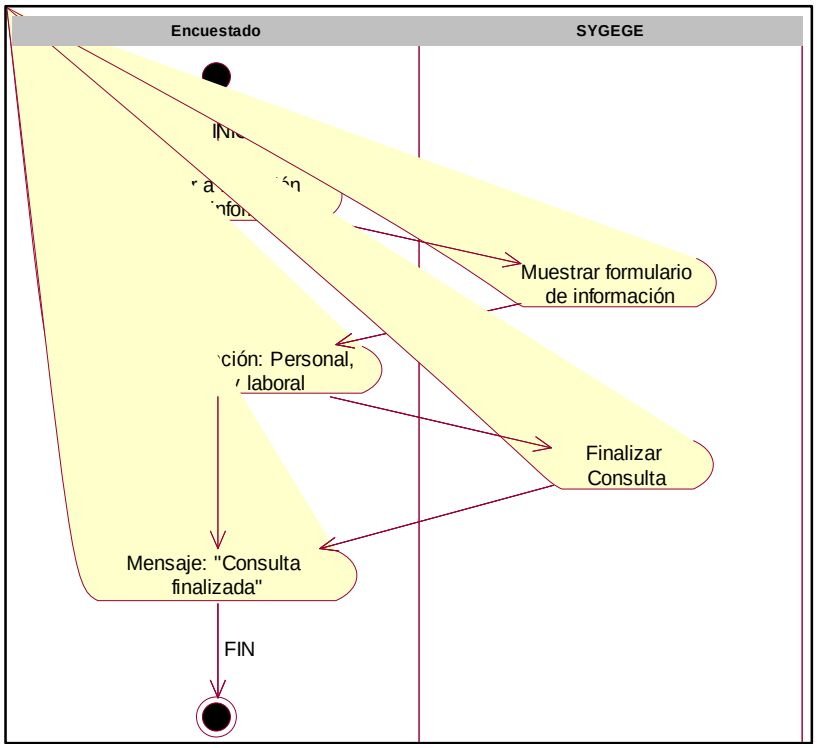


Figura 33: Consultar Información
Autor: Marcela Cuello Olmedo

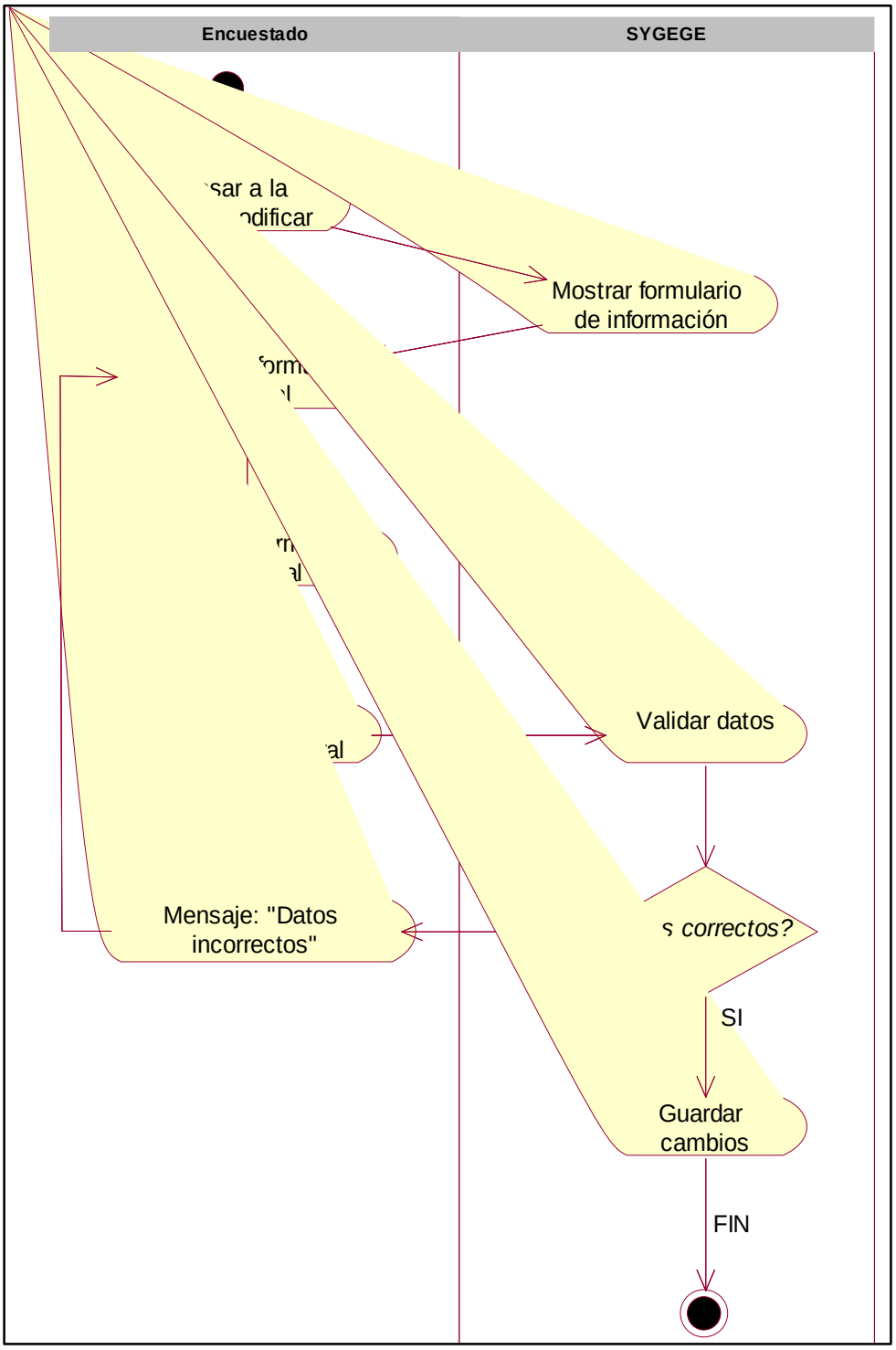


Figura 34: Modificar Información
Autor: Marcela Cuello Olmedo

3.5.2. Módulo Administración

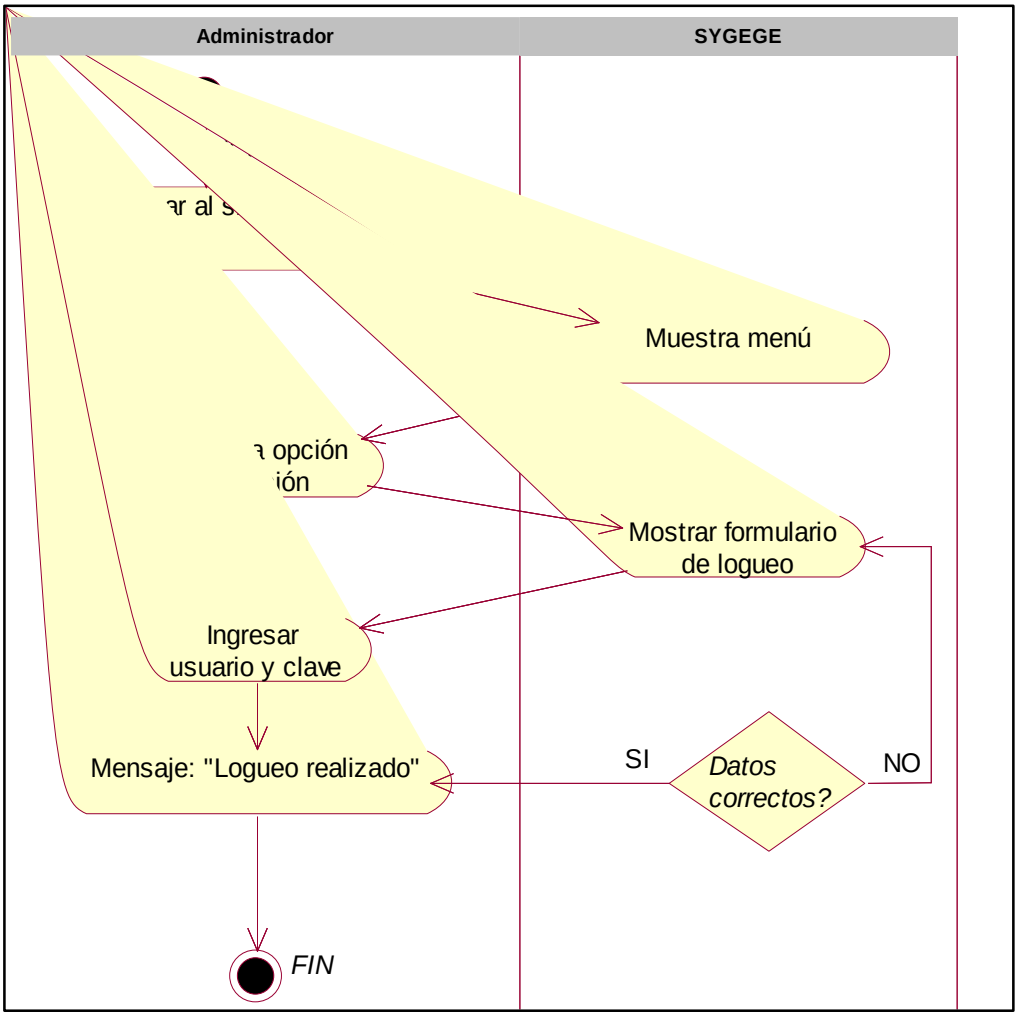


Figura 35: Loguearse en el sistema
Autor: Marcela Cuello Olmedo

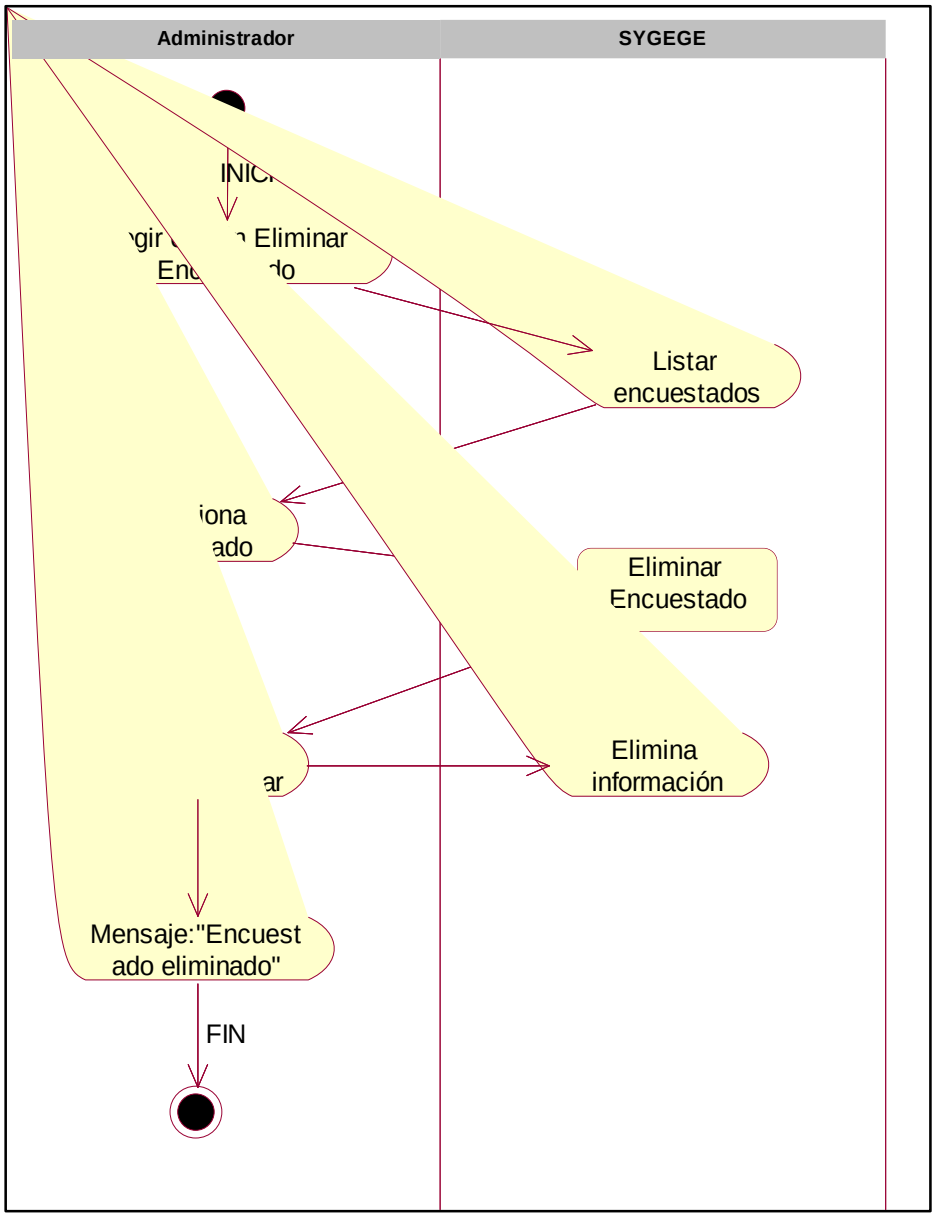


Figura 36: Eliminar encuestado
Autor: Marcela Cuello Olmedo

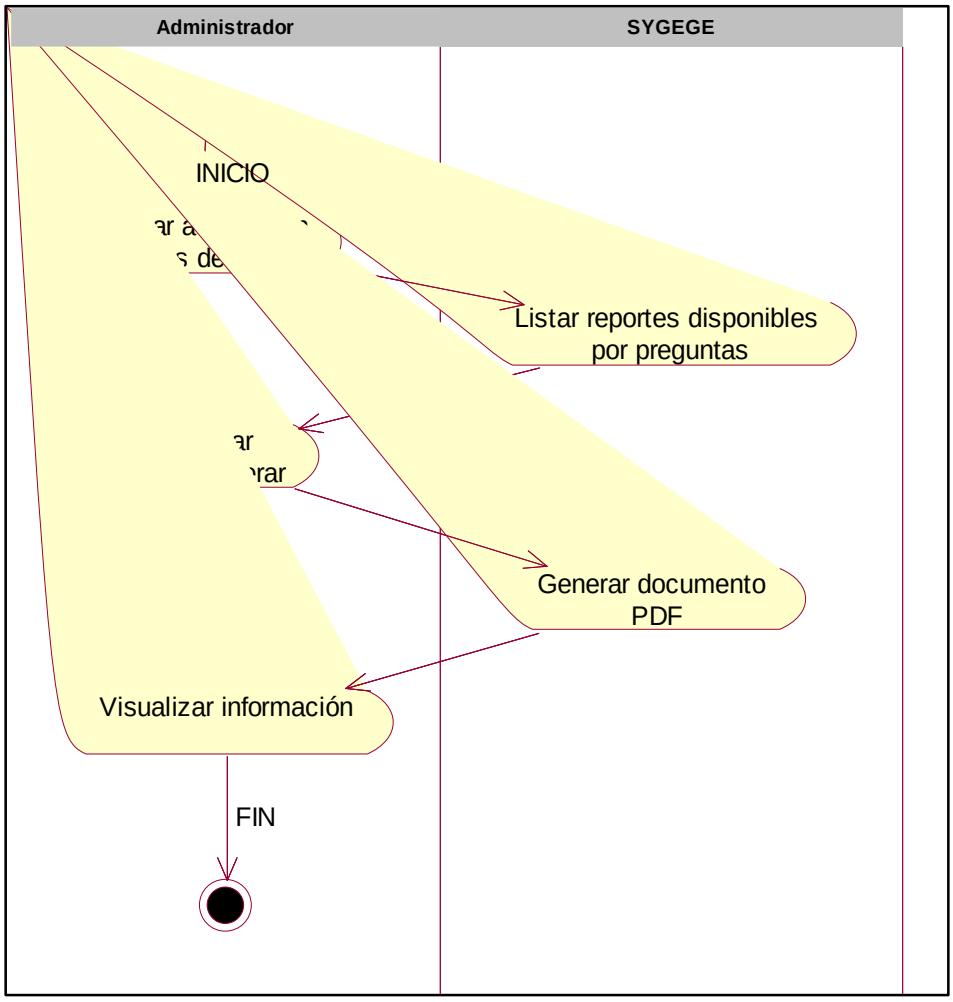


Figura 37: Generar reportes
Autor: Marcela Cuello Olmedo

3.6. Diagrama de clases

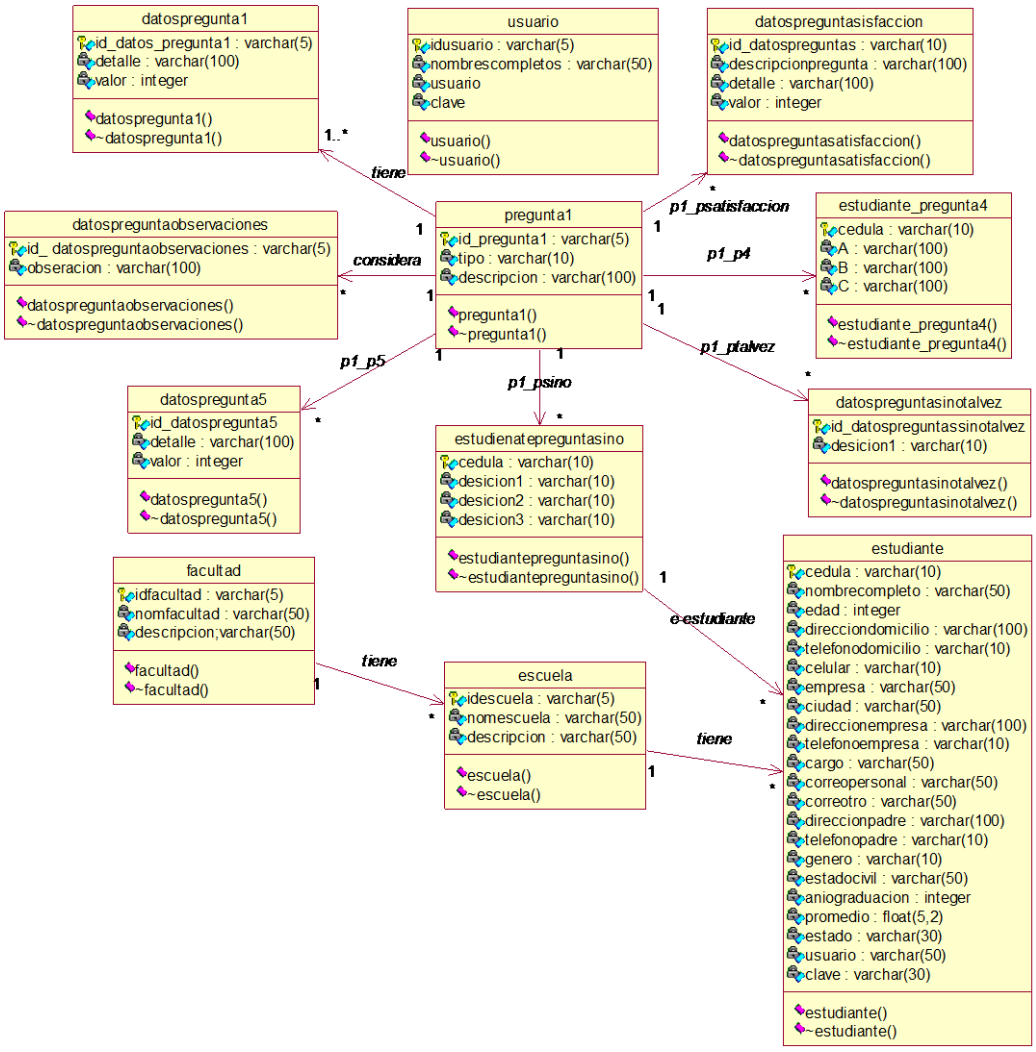


Figura 38: Diagrama de Clases SYGEGE
Autor: Marcela Cuello Olmedo

3.7. Diseño

Según Pressman, el diseño del software es realmente un proceso de muchos pasos, pero que se clasifican dentro de un mismo. En general, la actividad del diseño se refiere al establecimiento de la estructura de datos, la arquitectura general del sistema, representación de interfaz de usuario. Las interfaces del usuario, el cual es el medio con que el usuario puede comunicarse con el ordenador, las interacciones del usuario con el sistema SYGEGE reflejada con los diagramas de interacción. Además, se elabora un diagrama de clases que representará las clases que serán utilizadas dentro del sistema y sus relaciones y a partir de ellas tener el esquema de la base de datos. Los diagramas de

componentes y despliegue permiten representar como el sistema está dividido en componentes y el hardware en su implementación.

3.7.1. Definición de la arquitectura del sistema

El diseño de la arquitectura del sistema se refiere a la estructura global del software y la manera en que esa estructura proporciona integridad conceptual a un sistema. La arquitectura del software que este proyecto propone como una herramienta aplicativa de un contexto real, que describe las partes de un buen diseño. Se definen los componentes del sistema y la manera en que se empaquetan y cómo interactúan con otros. La especificación de la arquitectura del software tiene que representar fuertemente el concepto de modularidad del software, ya que es necesario poder identificar los componentes que forman parte de un ambiente virtual. La manera en que este proyecto de software se encuentra estructurado tiene que ser especificada desde su componente básico hasta los usuarios. La siguiente figura ilustra la arquitectura del sistema SYGEGE.

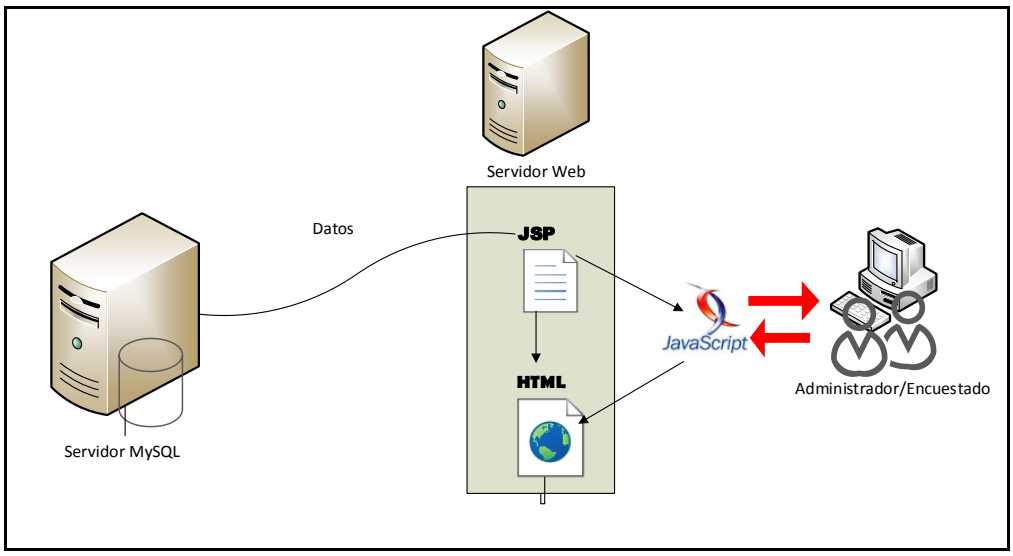


Figura 39: Arquitectura del sistema
Autor: Marcela Cuello Olmedo

3.7.2. Definición de la interfaz de usuario

Módulo Encuestas

Figura 40. Formulario de inicio
Autor: Marcela Cuello Olmedo

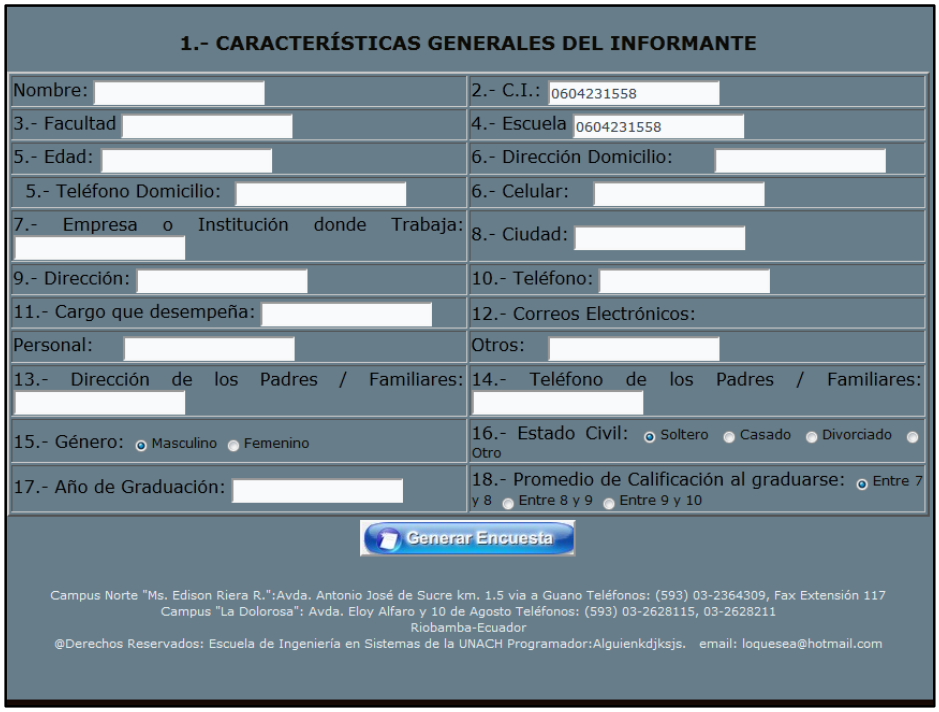


Figura 41: Página Características del Informante
Autor: Marcela Cuello Olmedo

Figura 42: Registro de Encuesta
Autor: Marcela Cuello Olmedo

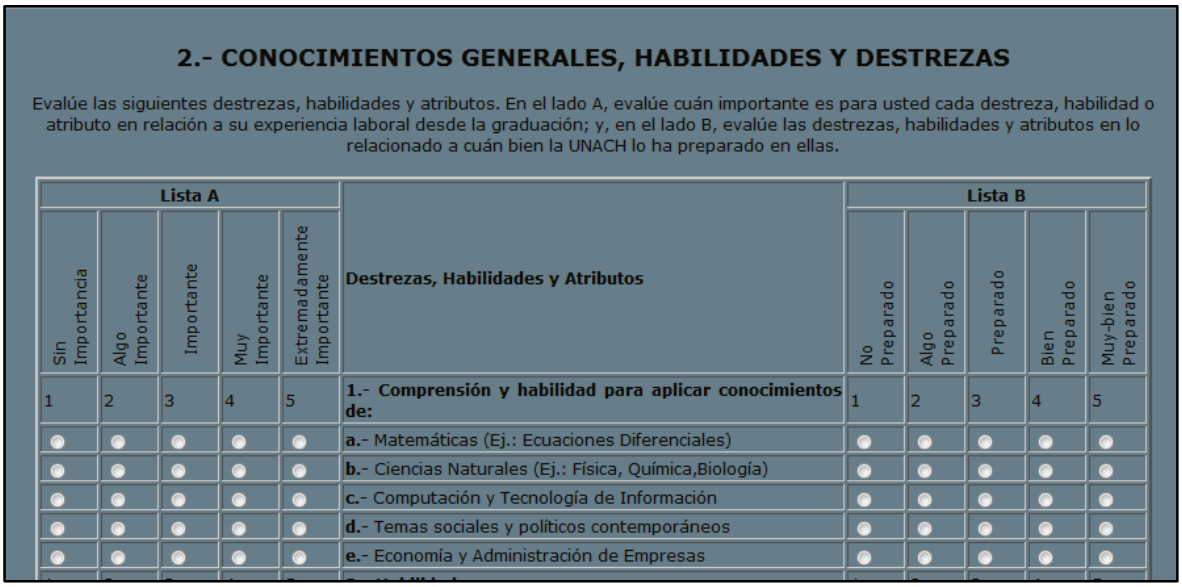
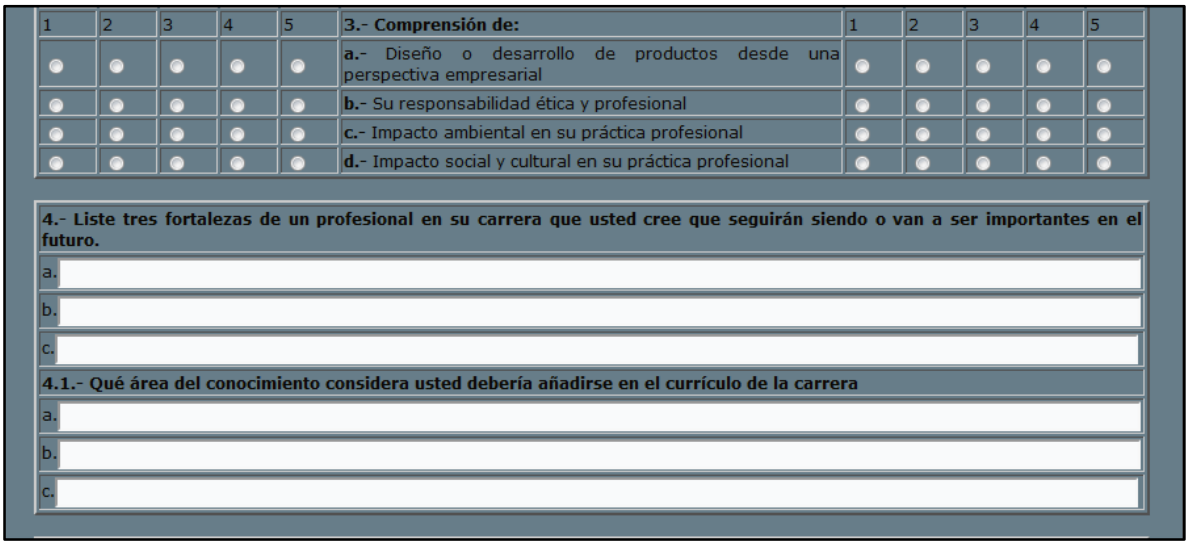


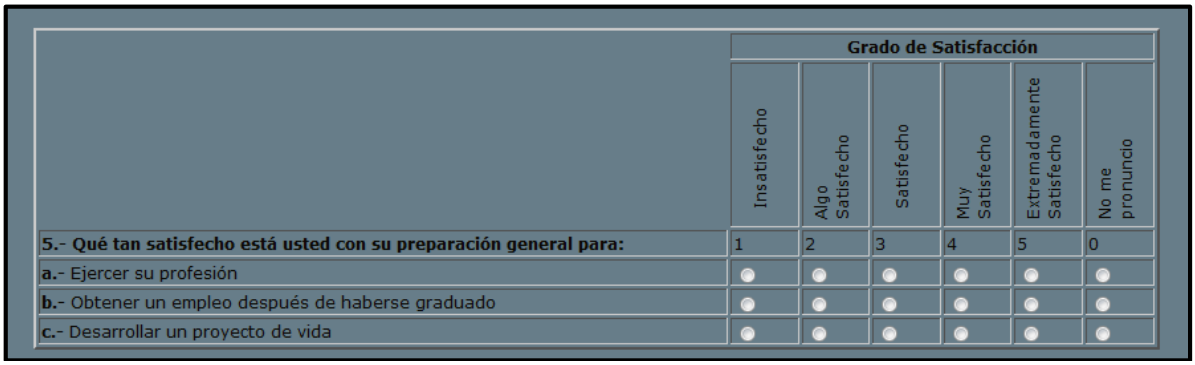
Figura 43: Encuesta. Conocimientos Generales, Habilidades y Destrezas (1)
Autor: Marcela Cuello Olmedo



1	2	3	4	5	3.- Comprensión de:	1	2	3	4	5
○	○	○	○	○	a.- Diseño o desarrollo de productos desde una perspectiva empresarial	○	○	○	○	○
○	○	○	○	○	b.- Su responsabilidad ética y profesional	○	○	○	○	○
○	○	○	○	○	c.- Impacto ambiental en su práctica profesional	○	○	○	○	○
○	○	○	○	○	d.- Impacto social y cultural en su práctica profesional	○	○	○	○	○

4.- Liste tres fortalezas de un profesional en su carrera que usted cree que seguirán siendo o van a ser importantes en el futuro.
a.
b.
c.
4.1.- Qué área del conocimiento considera usted debería añadirse en el currículo de la carrera
a.
b.
c.

Figura 44: Encuesta. Conocimientos Generales, Habilidades y Destrezas (2)
Autor: Marcela Cuello Olmedo



	Grado de Satisfacción					
	Insatisfecho	Algo Satisfecho	Satisfecho	Muy Satisfecho	Extremadamente Satisfecho	No me pronuncio
5.- Qué tan satisfecho está usted con su preparación general para:	1	2	3	4	5	0
a.- Ejercer su profesión	○	○	○	○	○	○
b.- Obtener un empleo después de haberse graduado	○	○	○	○	○	○
c.- Desarrollar un proyecto de vida	○	○	○	○	○	○

Figura 45: Encuesta. Conocimientos Generales, Habilidades y Destrezas (3)
Autor: Marcela Cuello Olmedo

3.- GRADO DE SATISFACCIÓN CON SU EXPERIENCIA FORMATIVA EN LA UNACH

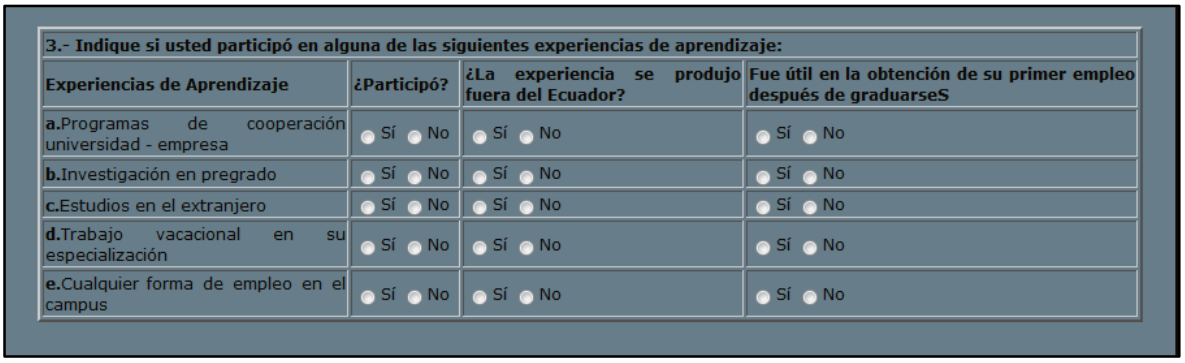
Indique su satisfacción en cada uno de los siguientes aspectos de acuerdo a su experiencia en la UNACH. Si no ha tenido experiencia en algún área particular, marque en la casilla "No me pronuncio".

	Calificación					
	Insatisfecho	Algo Satisfecho	Satisfecho	Muy Satisfecho	Extremadamente Satisfecho	No me pronuncio
1.- Satisfacción con el trato recibido por:	1	2	3	4	5	0
a.- Directivos	○	○	○	○	○	○
b.- Profesores	○	○	○	○	○	○
c.- Tutores de Tesis/Prácticas pre profesionales/otros	○	○	○	○	○	○
d.- Personal Administrativo y de Servicios	○	○	○	○	○	○
e.- Técnicas de Laboratorio	○	○	○	○	○	○
f.- Personal de biblioteca	○	○	○	○	○	○

2.- En qué actividades extra-curriculares participó usted durante su período como estudiante de la UNACH (Puede marcar más de una opción).

○ Representación estudiantil	○ Servicio comunitario
○ Desarrollo de Liderazgo	○ Grupo de teatro, coros, etc.
○ Publicaciones Estudiantiles	○ Otro:
○ Sociedades profesionales	
○ Clubes deportivos	○ Ninguna

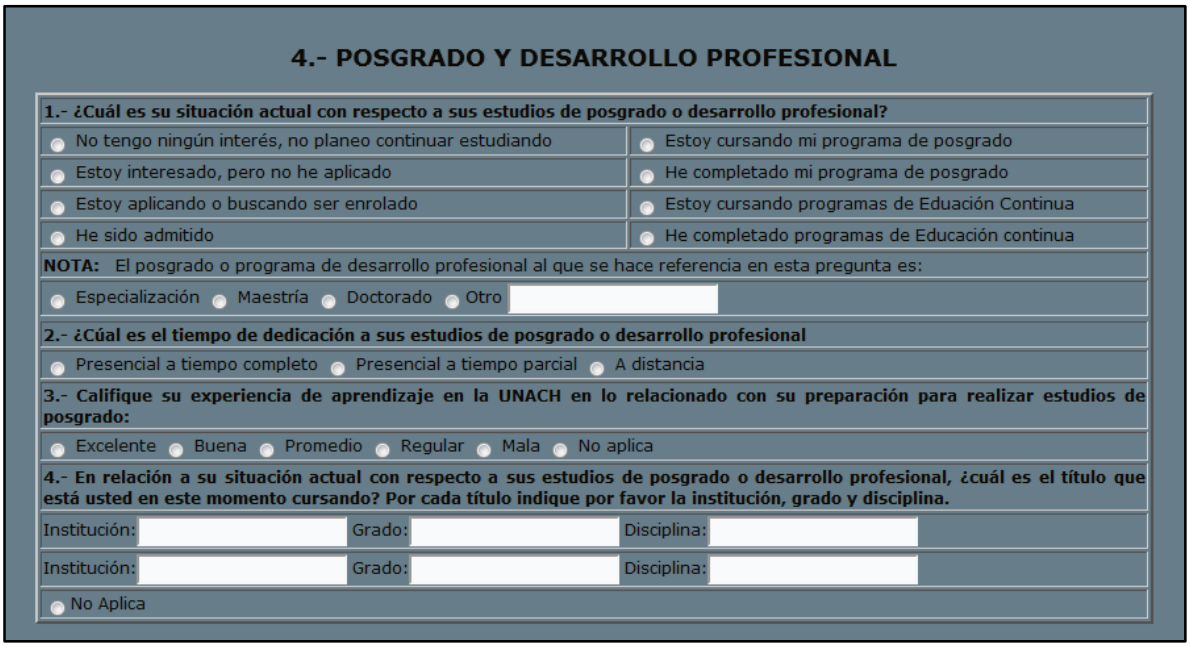
Figura 46: Encuesta. Grado de satisfacción con su experiencia formativa en la Unach (1)
Autor: Marcela Cuello Olmedo



3.- Indique si usted participó en alguna de las siguientes experiencias de aprendizaje:

Experiencias de Aprendizaje	¿Participó?	¿La experiencia se produjo fuera del Ecuador?	Fue útil en la obtención de su primer empleo después de graduarseS
a.Programas de cooperación universidad - empresa	○ Sí ○ No	○ Sí ○ No	○ Sí ○ No
b.Investigación en pregrado	○ Sí ○ No	○ Sí ○ No	○ Sí ○ No
c.Estudios en el extranjero	○ Sí ○ No	○ Sí ○ No	○ Sí ○ No
d.Trabajo vacacional en su especialización	○ Sí ○ No	○ Sí ○ No	○ Sí ○ No
e.Cualquier forma de empleo en el campus	○ Sí ○ No	○ Sí ○ No	○ Sí ○ No

Figura 47: Encuesta. Grado de satisfacción con su experiencia formativa en la Unach (2)
Autor: Marcela Cuello Olmedo



4.- POSGRADO Y DESARROLLO PROFESIONAL

1.- ¿Cuál es su situación actual con respecto a sus estudios de posgrado o desarrollo profesional?

No tengo ningún interés, no planeo continuar estudiando	Estoy cursando mi programa de posgrado
Estoy interesado, pero no he aplicado	He completado mi programa de posgrado
Estoy aplicando o buscando ser enrolado	Estoy cursando programas de Eduación Continua
He sido admitido	He completado programas de Educación continua

NOTA: El posgrado o programa de desarrollo profesional al que se hace referencia en esta pregunta es:

Especialización Maestría Doctorado Otro

2.- ¿Cúal es el tiempo de dedicación a sus estudios de posgrado o desarrollo profesional

Presencial a tiempo completo Presencial a tiempo parcial A distancia

3.- Califique su experiencia de aprendizaje en la UNACH en lo relacionado con su preparación para realizar estudios de posgrado:

Excelente Buena Promedio Regular Mala No aplica

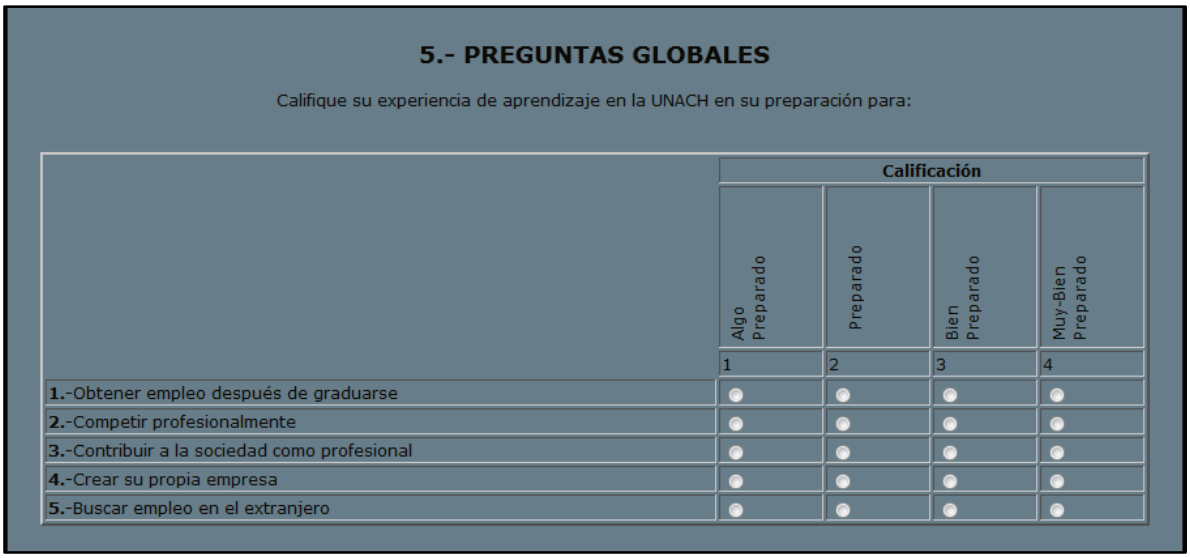
4.- En relación a su situación actual con respecto a sus estudios de posgrado o desarrollo profesional, ¿cuál es el título que está usted en este momento cursando? Por cada título indique por favor la institución, grado y disciplina.

Institución: Grado: Disciplina:

Institución: Grado: Disciplina:

No Aplica

Figura 48: Encuesta. Posgrado y desarrollo profesional
Autor: Marcela Cuello Olmedo



5.- PREGUNTAS GLOBALES

Califique su experiencia de aprendizaje en la UNACH en su preparación para:

	Calificación			
	Algo Preparado	Preparado	Bien Preparado	Muy-Bien Preparado
	1	2	3	4
1.-Obtener empleo después de graduarse				
2.-Competir profesionalmente				
3.-Contribuir a la sociedad como profesional				
4.-Crear su propia empresa				
5.-Buscar empleo en el extranjero				

Figura 49: Encuesta. Preguntas Globales
Autor: Marcela Cuello Olmedo

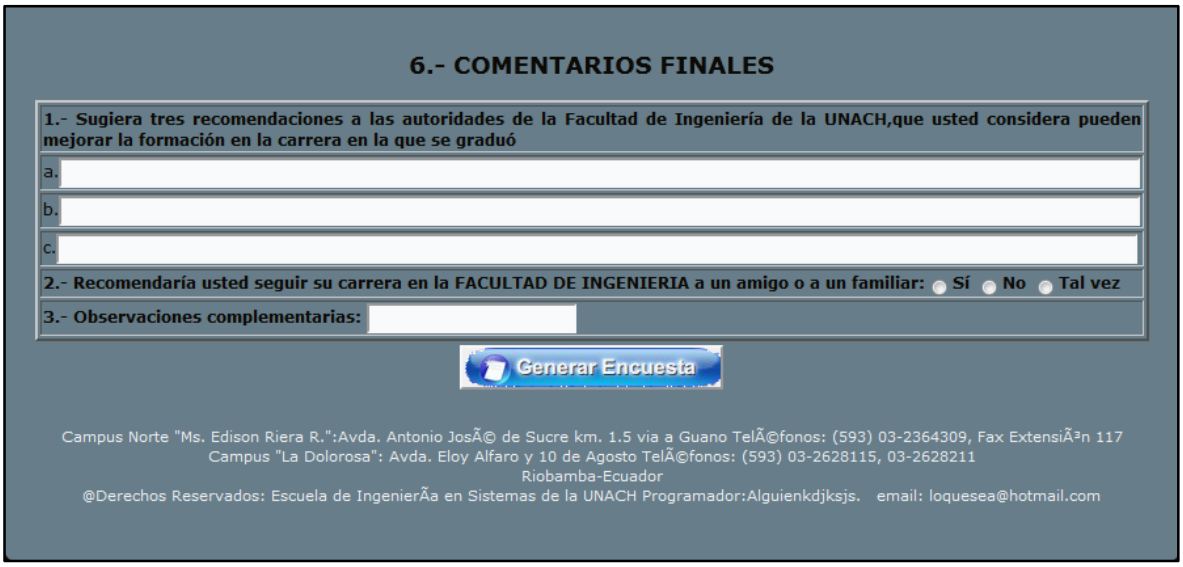
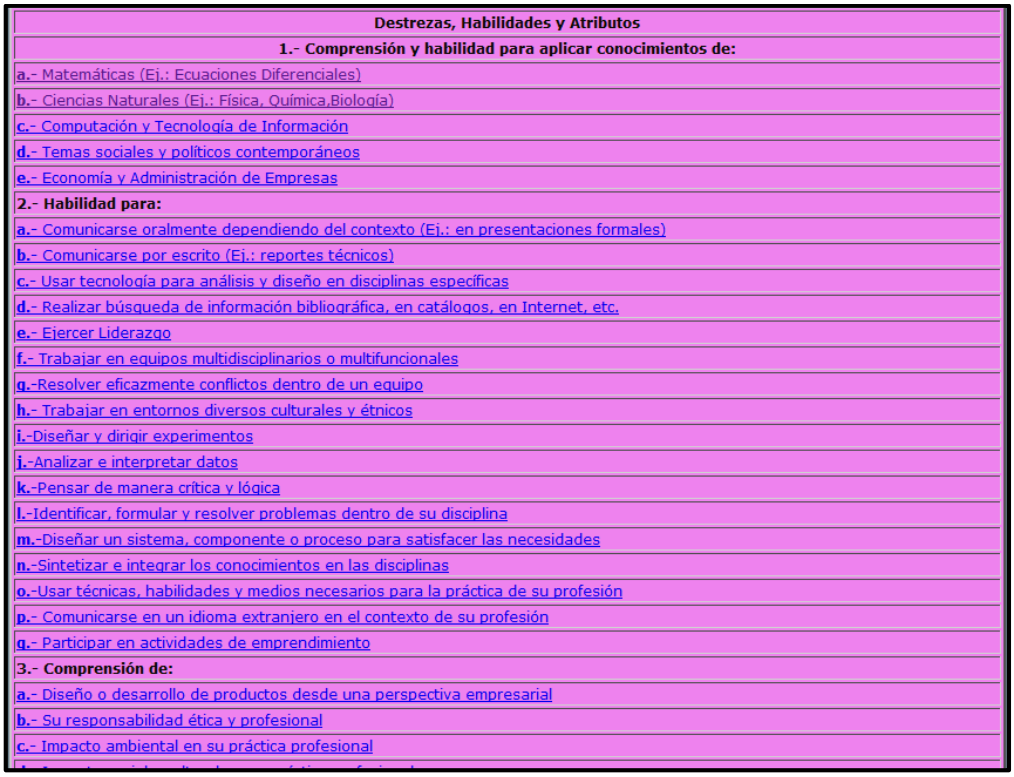


Figura 50: Encuesta. Comentarios Finales
Autor: Marcela Cuello Olmedo

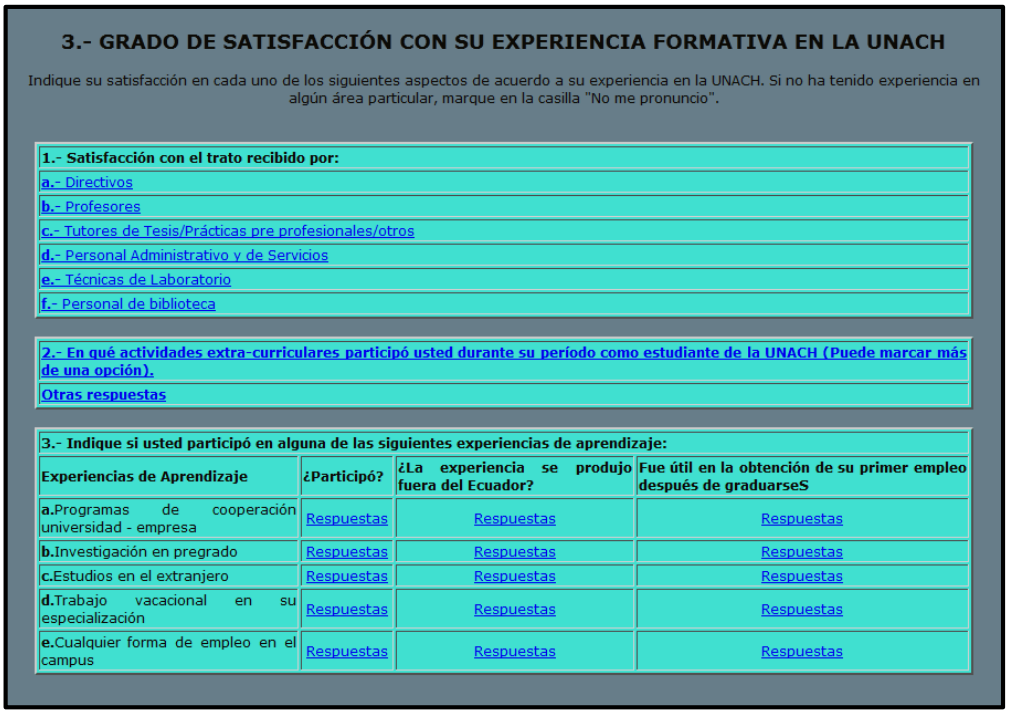
Figura 51: Página de Menú. Tabulación
Autor: Marcela Cuello Olmedo



Destrezas, Habilidades y Atributos
1.- Comprensión y habilidad para aplicar conocimientos de:
a.- Matemáticas (Ej.: Ecuaciones Diferenciales)
b.- Ciencias Naturales (Ej.: Física, Química,Biología)
c.- Computación y Tecnología de Información
d.- Temas sociales y políticos contemporáneos
e.- Economía y Administración de Empresas
2.- Habilidad para:
a.- Comunicarse oralmente dependiendo del contexto (Ej.: en presentaciones formales)
b.- Comunicarse por escrito (Ej.: reportes técnicos)
c.- Usar tecnología para análisis y diseño en disciplinas específicas
d.- Realizar búsqueda de información bibliográfica, en catálogos, en Internet, etc.
e.- Ejercer Liderazgo
f.- Trabajar en equipos multidisciplinarios o multifuncionales
g.-Resolver eficazmente conflictos dentro de un equipo
h.- Trabajar en entornos diversos culturales y étnicos
i.-Diseñar y dirigir experimentos
j.-Analizar e interpretar datos
k.-Pensar de manera crítica y lógica
l.-Identificar, formular y resolver problemas dentro de su disciplina
m.-Diseñar un sistema, componente o proceso para satisfacer las necesidades
n.-Sintetizar e integrar los conocimientos en las disciplinas
o.-Usar técnicas, habilidades y medios necesarios para la práctica de su profesión
p.- Comunicarse en un idioma extranjero en el contexto de su profesión
q.- Participar en actividades de emprendimiento
3.- Comprensión de:
a.- Diseño o desarrollo de productos desde una perspectiva empresarial
b.- Su responsabilidad ética y profesional
c.- Impacto ambiental en su práctica profesional

Figura 52: Reportes: Conocimientos por área

Autor: Marcela Cuello Olmedo



3.- GRADO DE SATISFACCIÓN CON SU EXPERIENCIA FORMATIVA EN LA UNACH

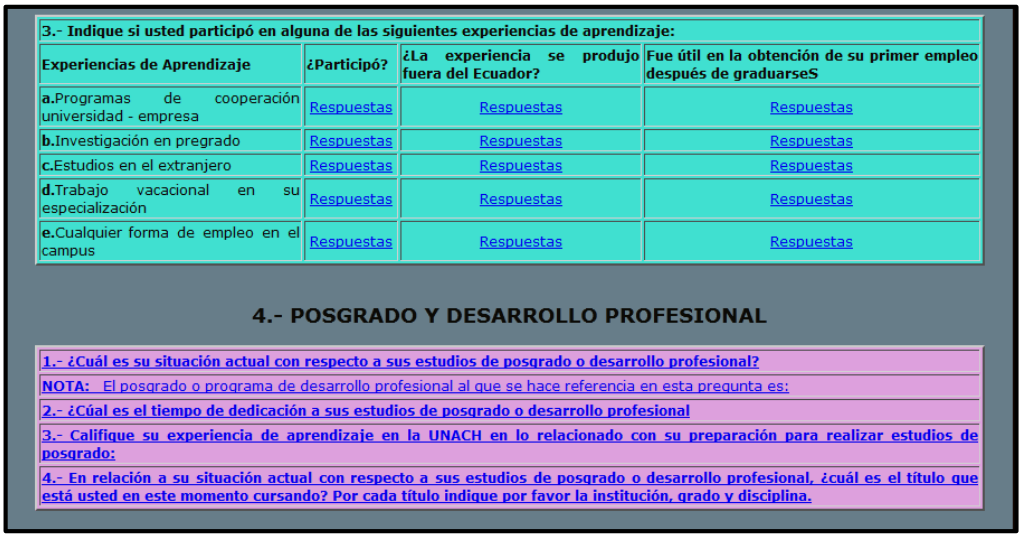
Indique su satisfacción en cada uno de los siguientes aspectos de acuerdo a su experiencia en la UNACH. Si no ha tenido experiencia en algún área particular, marque en la casilla "No me pronuncio".

1.- Satisfacción con el trato recibido por:
a.- Directivos
b.- Profesores
c.- Tutores de Tesis/Prácticas pre profesionales/otros
d.- Personal Administrativo y de Servicios
e.- Técnicas de Laboratorio
f.- Personal de biblioteca

2.- En qué actividades extra-curriculares participó usted durante su período como estudiante de la UNACH (Puede marcar más de una opción).
Otras respuestas

3.- Indique si usted participó en alguna de las siguientes experiencias de aprendizaje:			
Experiencias de Aprendizaje	**¿Participó?**	**¿La experiencia se produjo fuera del Ecuador?**	**Fue útil en la obtención de su primer empleo después de graduarseS**
a.Programas de cooperación universidad - empresa	Respuestas	Respuestas	Respuestas
b.Investigación en pregrado	Respuestas	Respuestas	Respuestas
c.Estudios en el extranjero	Respuestas	Respuestas	Respuestas
d.Trabajo vacacional en su especialización	Respuestas	Respuestas	Respuestas
e.Cualquier forma de empleo en el campus	Respuestas	Respuestas	Respuestas

Figura 53: Reportes: Habilidad
Autor: Marcela Cuello Olmedo



3.- Indique si usted participó en alguna de las siguientes experiencias de aprendizaje:			
Experiencias de Aprendizaje	**¿Participó?**	**¿La experiencia se produjo fuera del Ecuador?**	**Fue útil en la obtención de su primer empleo después de graduarseS**
a.Programas de cooperación universidad - empresa	Respuestas	Respuestas	Respuestas
b.Investigación en pregrado	Respuestas	Respuestas	Respuestas
c.Estudios en el extranjero	Respuestas	Respuestas	Respuestas
d.Trabajo vacacional en su especialización	Respuestas	Respuestas	Respuestas
e.Cualquier forma de empleo en el campus	Respuestas	Respuestas	Respuestas

4.- POSGRADO Y DESARROLLO PROFESIONAL

1.- ¿Cuál es su situación actual con respecto a sus estudios de posgrado o desarrollo profesional?
NOTA: El posgrado o programa de desarrollo profesional al que se hace referencia en esta pregunta es:
2.- ¿Cúal es el tiempo de dedicación a sus estudios de posgrado o desarrollo profesional
3.- Califique su experiencia de aprendizaje en la UNACH en lo relacionado con su preparación para realizar estudios de posgrado:
4.- En relación a su situación actual con respecto a sus estudios de posgrado o desarrollo profesional, ¿cuál es el título que está usted en este momento cursando? Por cada título indique por favor la institución, grado y disciplina.

Figura 54: Reportes: Posgrado y Desarrollo Profesional
Autor: Marcela Cuello Olmedo

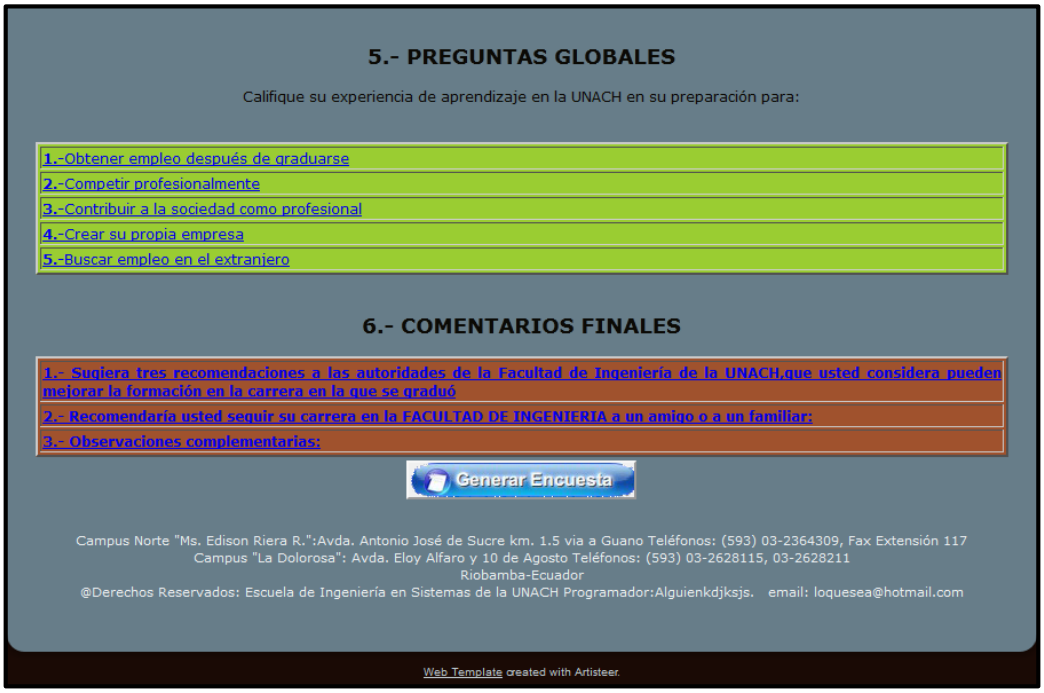


Figura 55: Reportes: Preguntas globales y Comentarios finales
Autor: Marcela Cuello Olmedo

Figura 56: Reporte Tabulación encuesta
Autor: Marcela Cuello Olmedo

3.7.3. Esquema de la base de datos

El esquema de una base de datos (en inglés, Database Schema) describe la estructura de una Base de datos, en un lenguaje formal soportado por un Sistema administrador de Base de datos (DBMS). En una Base de datos Relacional, el Esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla. El esquema es generalmente almacenado en un Diccionario de Datos. Aunque es común que el esquema sea definido en un lenguaje de Base de datos, el término se usa a menudo para referirse a una representación gráfica de la estructura de base de datos.

El sistema SYGEGE utiliza una base de datos relacional, elaborada en MYSQL que enfatiza que existen datos almacenados en las tablas para convertirse en información luego de un modelamiento completo, el esquema se detalla de acuerdo a los módulos que se manejan en el sistema software.

Base de datos: Egresados

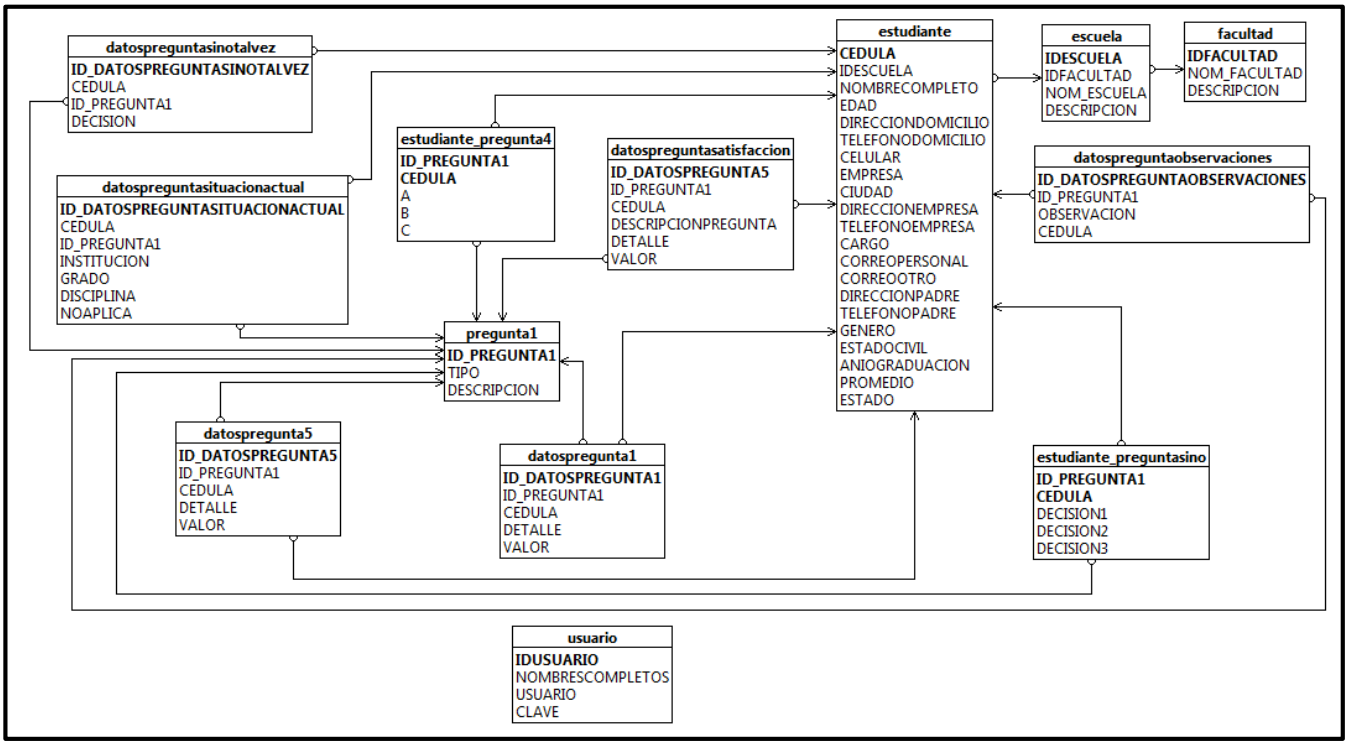


Figura 57: Esquema Base de Datos SYGEGE
Autor: Marcela Cuello Olmedo

3.7.4. Diagrama de componentes

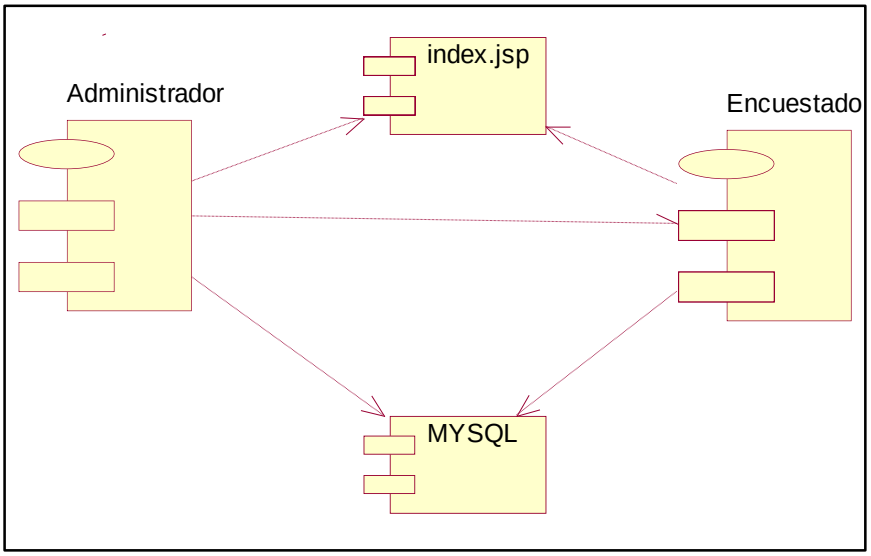


Figura 58: Diagrama de componentes SYGEGE
Autor: Marcela Cuello Olmedo

3.7.5. Diagrama de despliegue

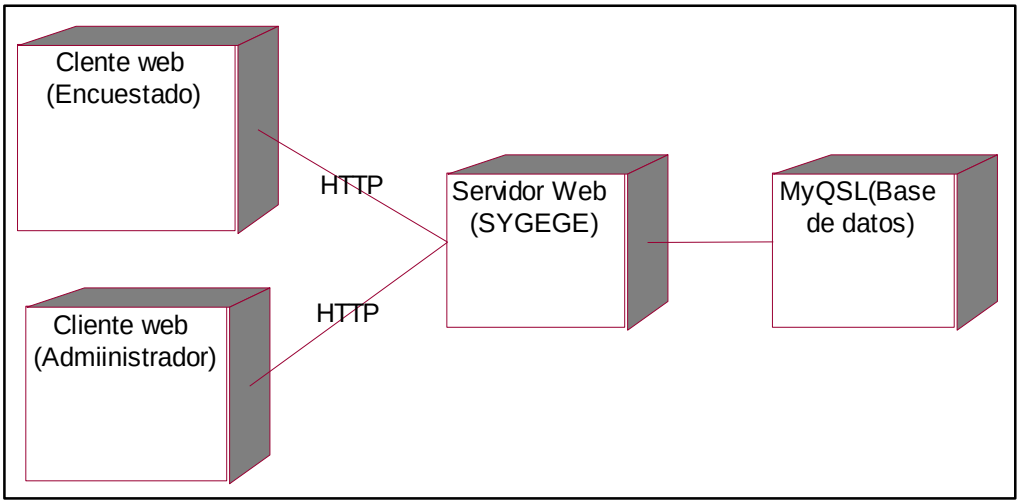


Figura 59: Diagrama de despliegue SYGEGE
Autor: Marcela Cuello Olmedo

3.8. Implementación del sistema SYGEGE

Se procede a programar o implementar los diseños especificados en el modelo de diseño, en esta fase se centra en crear una aplicación que permita la gestión de encuestas de seguimiento a egresados y profesionales, se emplea MYSQL 5.0, para la gestión de datos de los encuestados y de las encuestas, para la programación del sistema se utiliza el IDE NetBeans 7.0.0 que es una herramienta de desarrollo web y el código fuente es generado en lenguaje JSP y Java Script.

En el desarrollo de SYGEGE (Sistema de seguimiento de egresados y profesionales, la base de datos SYGEGE diseñada es muy importante para registrar información de encuestados y encuestas. El módulo de Administración gestiona encuestados, tabulación de datos y reportes; el módulo Encuestados gestiona: información de encuestados, registro de encuestas.

CAPITULO IV

4. Metodología

4.1.Tipo de estudio

En el siguiente apartado se detalla el estudio y las técnicas de investigación tales como: investigación de campo, investigación bibliográfica, las cuales se detallan a continuación:

4.1.1. Según el objeto de estudio

- ✓ **Investigación de Campo**. - Recolección de datos e identificación de requerimientos funcionales y no funcionales para el desarrollo del sistema SYGEGE para el seguimiento de graduados y egresados de la UNACH.

4.1.2. Según la fuente de investigación

a. **Investigación bibliográfica**. - Recolección de la información, utilizando técnicas y estrategias para acceder a documentos como: tesis, paper y libros para la investigación.

4.1.3. Según las variables

b. **Investigación Descriptiva. -** Con la medición y evaluación de varios aspectos como, dimensiones, funcionalidades, requerimientos, se desarrolla el sistema antes mencionado.

4.2. Población y muestra

4.2.1. Población

La población está comprendida por las tecnólogas multiplaformas como JSP, Ruby, y Python escogidos para el análisis comparativa de nuestro estudio.

Muestra

Para la muestra se considerará a toda la población, a la persona encargada de utilizar el sistema que esta totalizada por dos usuarios, los cuales son usuarios directos del sistema SYGEGE

4.3.Operacionalización de las variables

Mediante el uso de las variables definidas en el capítulo 2 sección 2.5 se precisan los siguientes indicadores: técnica, desarrollo, multiplataforma, lenguaje. Los cuales permitirán hacer un análisis de las tecnologías multiplataforma para seleccionar, el que mejor se acople a los requerimientos para el desarrollo del sistema SYGEGE.

Tabla 36 Operación de las variables

Variable	Tipo	Definición conceptual	Dimensión	Indicadores
Estudio de las tecnologías multiplataforma para aplicaciones web	Independiente	Estas tecnologías permiten la integración y optimización de los procesos y recursos de la institución.	Se utilizan las tecnologías multiplataforma para la UNACH: JSP, RUBY Y PYTHON.	Eficiencia Costo Calidad
Permitirá desarrollar de forma eficiente el sistema de seguimiento a graduados y egresados de la facultad de Ingeniería de la Universidad Nacional de Chimborazo	Dependiente	Mediante el desarrollo de este sistema permitirá desarrollar de forma eficiente los procesos de seguimientos a graduados y egresados de la UNACH	Departamentos de: Seguimiento a graduados y egresados.	Técnica, Desarrollo, Multiplataforma, Lenguaje

Autor: Marcela Cuello

4.1. Comprobación de la hipótesis

Para verificar la hipótesis se utiliza, un estadígrafo conocido como Chi – Cuadrado, el cual es un método útil para probar las hipótesis relacionadas con la diferencia entre el conjunto de frecuencias observadas en una muestra y el conjunto de frecuencias teóricas y esperadas de la misma muestra.

En este tipo de problemas el estadístico de prueba es:

$$(fo - Fe)^2$$

$$X = \Sigma \frac{}{Fe}$$

En donde:

X^2 = Chi-cuadrado

Σ= Sumatoria

Fo = Frecuencia observada de realización de un acontecimiento determinado. (Se puede medir físicamente)

Fe = Frecuencia esperada o teórica. (El resultado que se desea obtener de un experimento)

La aplicación de esta ecuación requiere lo siguiente:

- ✓ Encontrar la diferencia entre cada frecuencia observada y la correspondiente frecuencia esperada.
- ✓ Elevar al cuadrado estas diferencias.
- ✓ Dividir cada diferencia elevada al cuadrado entre la correspondiente frecuencia esperada.
- ✓ Sumar los cocientes restantes.

4.1.1. Nivel de significancia

Además, se hizo uso de un margen de error del 5% el cual se convierte en un nivel de confianza de 0.05 con el que se buscan los datos en la tabla chi-cuadrado.

El grado de libertad se obtendrá a través de la formula.

G1=(f-1) (c-1)

Dónde:

G1= Grado de libertad

F=Filas

C= Columnas.

Para obtener el chi-cuadrado según la tabla se buscó el grado de libertad y el nivel de confianza y así se obtuvo la chi-cuadrado tabla (Xt) que se compara con el chi-cuadrado calculado(Xc).

De acuerdo a este criterio se determinó si el Xc es mayor o igual que el Xt se aceptó la hipótesis de trabajo y se rechazó la hipótesis nula.

Si el Xt es mayor que el Xc se rechaza la hipótesis de trabajo y se acepta la hipótesis nula

Tabla 37 Encuesta sobre Tecnologías

N°	Pregunta	Alternativas		TOTAL
		SI	NO	
2	*¿Las Tecnologías multiplataforma permite el desarrollo de sistemas web,* que sean sencillas de administrar y mantener por las organizaciones?	16	12	28
4	*¿Se puede mejorar la eficiencia de un sistema web, si se utiliza la mejor tecnología orientada a la creación de sistemas web?*	25	4	29
7	*¿Las tecnologías Multiplataforma son la mejor alternativa para la creación de sistemas dinámicos de encuestas?*	16	13	29
10	*¿La tecnología JSP, facilita la presentación de la información en forma tabulada además de graficas estadísticas obtenida de la base de datos?*	17	11	28

Autor: Marcela Cuello

Cálculo de las frecuencias Esperadas

Cuando tenemos un solo criterio de clasificación dividido en varias categorías el cálculo de las frecuencias teóricas o esperadas es sencillo:

$$Fe = N/K$$

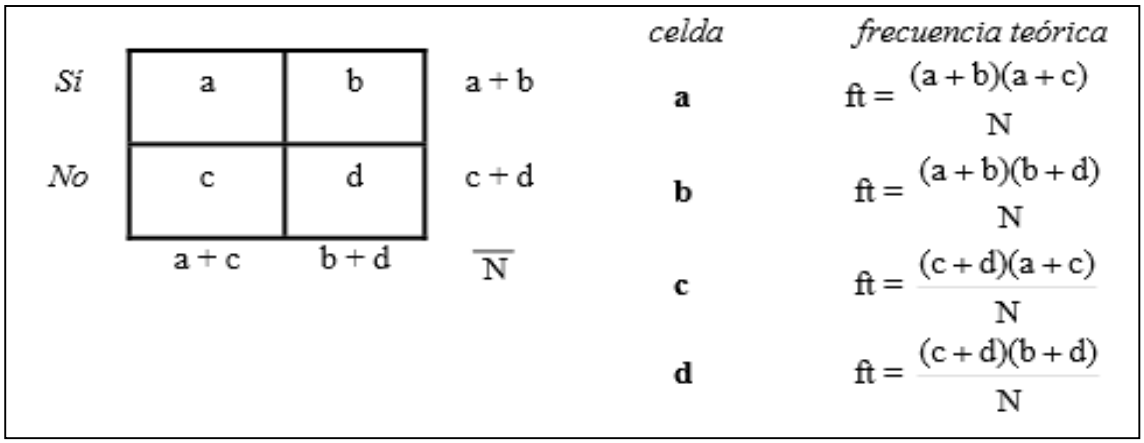
N= número de eventos

K=número de oportunidades

En este caso todos tienen la misma oportunidad

Cuando hay dos criterios de clasificación como en nuestro caso (cuadros de doble entrada). Las frecuencias teóricas de cada casilla son iguales al producto de las sumas marginales dividido por el número total de sujetos.

En el caso de dos categorías con dos niveles de clasificación (podrían ser más) tendríamos:



Autor: Morales Vallejo, Pedro (2008) Estadística aplicada a las Ciencias Sociales. Madrid: Universidad Pontificia Comillas (edit@pub.upcomillas.es)

Calculo de frecuencias Esperadas con respuesta afirmativa

Tabla 38 Calculo de frecuencias esperadas

N°	Pregunta	Alternativas		Frecuencia Esperada (Fe) Positivas
		SI	NO	
2	*¿Las Tecnologías multiplataforma permite el desarrollo de sistemas web,* que sean sencillas de administrar y mantener por las organizaciones?	16	12	20.140
4	*¿Se puede mejorar la eficiencia de un sistema web, si se utiliza la mejor tecnología orientada a la creación de sistemas web?*	25	4	20.859
7	*¿Las tecnologías Multiplataforma son la mejor alternativa para la creación de sistemas dinámicos de encuestas?*	16	13	20.5
10	*¿La tecnología JSP, facilita la presentación de la información en forma tabulada además de graficas estadísticas obtenida de la base de datos?*	17	11	16.210

Autor: Marcela Cuello

Calculo de frecuencias Esperadas con respuesta negativa

Tabla 39 Cálculo de Frecuencia esperada negativa

N°	Pregunta	Alternativas		Frecuencia Esperada (Fe) Negativas
		SI	NO	
2	*¿Las Tecnologías multiplataforma permite el desarrollo de sistemas web,* que sean sencillas de administrar y mantener por las organizaciones?	16	12	7.859
4	*¿Se puede mejorar la eficiencia de un sistema web, si se utiliza la mejor tecnología orientada a la creación de sistemas web?*	25	4	8.140
7	*¿Las tecnologías Multiplataforma son la mejor alternativa para la creación de sistemas dinámicos de encuestas?*	16	13	8.5
10	*¿La tecnología JSP, facilita la presentación de la información en forma tabulada además de graficas estadísticas obtenida de la base de datos?*	17	11	11.789

Autor: Marcela Cuello

Tabla de Contingencia

Tabla 40 Cálculo de Chi Cuadrado calculado

Pregunta	Fo	Fe	Fo-Fe	(Fo-Fe)²	(Fo-Fe)² / Fe
¿Las Tecnologías multiplataforma permite el desarrollo de sistemas web, que sean sencillas de administrar y mantener por las organizaciones?	16	20.140	*-4.14*	*17.139*	*0.851*
¿Se puede mejorar la eficiencia de un sistema web, si se utiliza la mejor tecnología orientada a la creación de sistemas web?	25	20.859	*4.141*	*17.147*	*0.822*
¿Las tecnologías Multiplataforma son la mejor alternativa para la creación de sistemas dinámicos de encuestas?	16	20.5	*-4.5*	*20.25*	*0.987*
¿La tecnología JSP, facilita la presentación de la información en forma tabulada además de graficas estadísticas obtenida de la base de datos?	17	16.210	*0.79*	*0.624*	*0.038*

¿Las Tecnologías multiplataforma no permite el desarrollo de sistemas web, que sean sencillas de administrar y mantener por las organizaciones?	12	7.859	*4.141*	*17.147*	*2.181*
¿No se puede mejorar la eficiencia de un sistema web, si se utiliza la mejor tecnología orientada a la creación de sistemas web?	4	8.140	*-4.14*	*17.139*	*2.105*
¿Las tecnologías Multiplataforma no son la mejor alternativa para la creación de sistemas dinámicos de encuestas?	13	8.5	*4.5*	*20.25*	*2.382*
¿La tecnología JSP, no facilita la presentación de la información en forma tabulada además de graficas estadísticas obtenida de la base de datos?	11	11.789	*-0.789*	*0.622*	*0.052*
					9.418

Autor: Marcela Cuello

Chi-Cuadrado Calculado $X^2c = 9.418$

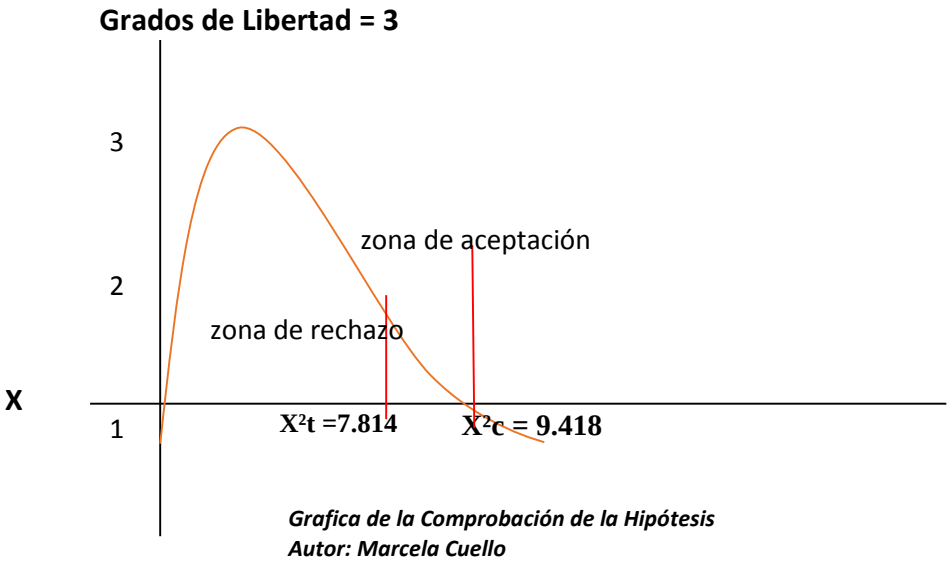
Ahora Calculamos Chi-Cuadrado Tabla

Grado de libertad
Gl: (f-1) (c-1) **f**= número de filas y **c** = número de columnas
(4-1) (2-1)
(3) (1) = 3
Gl: 3
Nivel de confianza=0.05
Chi-Cuadrado Tabla.

X^2t = **7.814**Valor encontrado en la tabla de probabilidad Chi Cuadrado (Anexo)

$$X^2c = 9.418 > X^2t = 7.814$$

De acuerdo a estos resultados pudo comprobarse que el chi-cuadrado calculado es mayor que el chi-cuadrado tabla, por lo cual se acepta la Hipótesis de trabajo y se rechaza la Hipótesis nula, Es decir **"El estudio de las principales tecnologías multiplataforma permitirá desarrollar eficientemente, el sistema de seguimiento de graduados y egresados de la facultad de Ingeniería de la Universidad Nacional de Chimborazo"**



Grafica de la Comprobación de la Hipótesis
Autor: Marcela Cuello

DISTRIBUCION DE χ^2

Grados de libertad	Probabilidad										
	0,95	0,90	0,80	0,70	0,50	0,30	0,20	0,10	0,05	0,01	0,001
1	0,004	0,02	0,06	0,15	0,46	1,07	1,64	2,71	3,84	6,64	10,83
2	0,10	0,21	0,45	0,71	1,39	2,41	3,22	4,60	5,99	9,21	13,82
3	0,35	0,58	1,01	1,42	2,37	3,66	4,64	6,25	7,82	11,34	16,27
4	0,71	1,06	1,65	2,20	3,36	4,88	5,99	7,78	9,49	13,28	18,47
5	1,14	1,61	2,34	3,00	4,35	6,06	7,29	9,24	11,07	15,09	20,52
6	1,63	2,20	3,07	3,83	5,35	7,23	8,56	10,64	12,59	16,81	22,46
7	2,17	2,83	3,82	4,67	6,35	8,38	9,80	12,02	14,07	18,48	24,32
8	2,73	3,49	4,59	5.53	7,34	9,52	11,03	13,36	15,51	20,09	26,12
9	3,32	4,17	5,38	6,39	8,34	10,66	12,24	14,68	16,92	21,67	27,88
10	3,94	4,86	6,18	7,27	9,34	11,78	13,44	15,99	18,31	23,21	29,59
	No significativo								Significativo		

CONCLUSIONES

- Se analiza las tecnologías multiplataforma JSP, Ruby y Python para el despliegue de aplicaciones web, la sintaxis limpia, la legibilidad de código y la programación funcional que poseen conducen a obtener resultados en productos software de calidad, seguros, mantenibles, escalables, portables y funcionales.

- Para el desarrollo de una aplicación web de propósito general, JSP y el ambiente Windows crean un escenario que acelera y facilita el uso de técnicas de programación (orientación a objetos, estructuras de control, manejo de módulos, paquetes, etc.). El sistema de gestión de los procesos del seguimiento de egresados y graduados de la Unach, se apoya de las mejores propiedades del lenguaje de programación JAVA y de las capacidades técnicas de la tecnología JSP como es brindar seguridad, madurez y portabilidad a las aplicaciones web.

- Los procesos de ingeniería de software que comúnmente se conoce (desarrollo, operación y mantenimiento), han sido adoptados por la ingeniería web, en ambiente multidisciplinarios (como la web) y donde los requerimientos son cambiantes. La funcionalidad, mantenibilidad, escalabilidad y seguridad de una aplicación web son características que se cumple y son cubiertos en dichas fases, estas consideraciones técnicas y los procesos de ingeniería de software son aptos para el desarrollo del sistema de seguimiento de egresados y graduados de la Universidad Nacional de Chimborazo.

- La existencia de una nueva plataforma tecnológica que permitirá satisfacer las actuales necesidades de negocios cambiará la manera en que se desarrollan las aplicaciones Web. Esto representa una evolución, no una revolución, al incorporar conceptos existentes que han sido de gran utilidad. Sea en la tecnología que fuera (JSP, Ruby o Python), iremos viendo cómo, progresivamente, las tecnologías multiplataforma irán tomando fuerza en el desarrollo de aplicaciones con base en la Web.

RECOMENDACIONES

- El uso de las tecnologías multiplataforma JSP, Ruby o Python obliga a explotar de mejor manera los beneficios que estas poseen, apoyarse buenas prácticas de programación, conocimientos básicos y/o profundos de lenguajes de desarrollo web, HTML, Java Script, IDEs de desarrollo y editores propios de cada tecnología, facilita el buen desarrollo de aplicaciones exigentes.

- Con respecto al diseño y desarrollo de una aplicación web (procesos de ingeniería de software), específicamente en sitios web dinámicos, es aconsejable seguir una metodología de selección, la misma que se detalla a continuación: Analizar el nivel de complejidad de la aplicación a desarrollar, determinar el costo que se puede invertir en el proyecto, determinar el nivel de seguridad que se manejara en la aplicación, mirar la portabilidad de la aplicación, la estabilidad, la seguridad en ambientes heterogéneos.

- En lo que respecta al éxito del sitio web de seguimiento a egresados y graduados de la Unach, como cualquier otro sitio web publicado en Internet es aconsejable su constante actualización de contenidos, como también responder oportunamente a los requerimientos de sus visitantes, en este caso en especial a los profesionales y egresados.

- Finalmente entender que cada minuto que pasa se desarrollan con gran facilidad las diferentes tecnologías web existentes y aparecen otras nuevas que ofrecen mejoras para los desarrolladores o programadores web, por lo cual es fundamental estar en una constante actualización de conocimientos.

BIBLIOGRAFÍA

1. Alberto Espin, A. M. (2009). Nuevas tecnologías en torno a sistemas de información basados en web, xml, jsp, asp y php. Perú: Advertur.
2. Bahit, E. (2010). Curso Python para Principiantes. Mexico.
3. Edelson, B. M. (2006). "Java and XML". . O'Reilly. USA. .
4. García, Á. L. (2010). Introducción a Python. Colombia: Adverture.
5. I. Jacobson, G. B. (2010). "El Proceso Unificado de desarrollo de software. España.
6. J. García, J. R. (2008). Java Construction. San Sebastian.
7. Navenn, S. B. (2008). "Enhanced NMS Tool Architecture for Discovery and Monitoring of Nodes". Mexico: Blekinge Institute of Technology.
8. S. Brown, S. D. (2012). "Pro JSP 2". Cuarta edición. . Usa: Appres.
9. Schmidt, M. D. (2013.). "Architecture for SNMP Management Frameworks". Colombia.
10. Schmidt, M. K. (2005). Essential SN MP Segunda Edicion. USA.
11. Costa, D. C. (2009). Introducción a la Base de Datos. México: FUOC.
12. Fernández Alarcón, V. (2006). Desarrollo de sistemas: Una metodología basada en el modelado. (Primera ed.). Cataluña, España: Edicions UPC.
13. Fernández Alarcón, V. (2010). Desarrollo de sistemas de información.Una metodología basada en el modelado. Cataluña: Edicions UPC SL.
14. Gil, F. (2012). Experto en Drupal 7: Nivel Inicial (Primera ed.). Lima, Peru: Forcontu S.L.
15. GONZALEZ LORCA, J. (2005). SISTEMAS WORKFLOW: FUNCIONAMIENTO Y METODOLOGIA DE IMPLANTACION. Madrid, España: TREA.
16. A.Simon, H. (2010). El comportamiento Administrativo. Chile: Aguilar.
17. ANDREU, E., RICART, J., & VALOR, J. (1997). Estrategia y Sistemas de Información (Segunda ed.). Barcelona, España: Ediciones Barcelona.

ANEXOS

ANEXO A

ANALISIS DE TECNOLOGÍAS MULTIPLATAFORMAS

Criterio: Características de la tecnología multiplataforma

Sección A: Propiedades tecnológicas

Características	Tecnologías de Desarrollo Web		
	JSP	**RUBY**	**PYTHON**
Es una tecnología multiplataforma	2	2	2
Trabaja con servidores web	2	2	2
Tiene componentes reusables la plataforma	2	1	2
Permite la protección de memoria de escape	2	1	1
Permite la integración de Base de Datos	2	1	1
Trabaja con componentes propios	2	2	1
Total	**12**	**9**	**9**
Resultado	**2/2**	**1,50**	**1,50**
Equivalencia	**100%**	**75%**	**75%**

Autor: *Marcela Cuello Olmedo*

Conclusión Parcial Sección A:

JSP: Con propiedades tecnológicas al 100%. Una tecnología multiplataforma que es soportada en Solaris, Windows, Mac OS, Linux, además de trabajar con servidores web Apache, Netscape, IIS. La integración bases de datos es mediante JDBC y ODBC, además de trabajar con componentes propios como son: Componentes propios y reusables como son: JavaBeans, Enterprise JavaBeans y extensiones JSP.

RUBY: Al ser una tecnología con licencia de software libre, es también multiplataforma que se integran en gran cantidad de arquitecturas, incluso en dispositivos móviles, cumple con un 75% de las propiedades tecnológicas señaladas, se ve aun limitada en: la reusabilidad de componentes, la protección de memoria (gestión de memoria automática) y la integración de bases de datos.

PYTHON: Esta tecnología cumple con el 75% de las características tecnológicas, también considerado multiplataforma (Windows, Mac, Linux). Limitada al uso de componentes propios que al ser propios de la tecnología en algunos casos un módulo, paquete o componente requiere una instalación por separado.

Sección B: Especificaciones Técnicas

Entre las características que se valora en cada una de estas tecnologías, se establecen: versión actual, propósito, año de creación, influencias de otros lenguajes, sitios desarrollados con estas tecnologías, usabilidad, curva de aprendizaje, popularidad, velocidad de ejecución y demanda laboral de profesionales.

Características	Tecnologías de Desarrollo Web		
	JSP	**RUBY**	**PYTHON**
Trabaja con versiones actualizadas	2	2	2
Trabaja para un propósito general	2	2	2
Se conoce quien lo desarrollo	2	2	2
Utilizado para construir sitios web reconocidos.	2	2	2
Capacidad de usabilidad	2	2	2
Facilidad de aprendizaje	1	2	2
Total	**11**	**12**	**12**
Resultado	**1,83**	**2/2**	**2/2**
Equivalencia	**91,67%**	**100%**	**100%**

Autor: *Marcela Cuello Olmedo*

Conclusión Parcial Sección B:

JSP: Con un 91,67% esta tecnología solo se ve limitada por la dificultad de aprendizaje que posee debido a que exige tener una experiencia previa en programación. Esto es debido a que Java es un lenguaje muy potente, pero un poco más complicado de usar porque es orientado a objetos y la manera de escribir los programas es más rígida. Trabaja con versiones estables y actualizadas (JSP 2.1) desarrollo de la compañía Sun Microsystems, el propósito de JSP es escribir aplicaciones y ejecutarlas en cualquier ambiente tecnológico (dispositivos y sistema operativo), la capacidad de uso mayor cada vez sorprendente ya que permite escribir programas prácticamente para todo lo que se pueda necesitar: Software para equipos inteligentes, control, navegación, industria, red, telefonía celular y por supuesto, la Web.

RUBY: En un 100% esta tecnología ofrece versiones actualizadas de IDES, editores para el desarrollo de software (RUBY: 1.9.3 liberada en octubre 31, 2011), tiene como propósito hacer que la programación sea divertida y flexible para el programador. Creada en 1995 por Yukihiro Matz Matsumoto, e influenciada por los lenguajes de programación Ada, C++, CLU, DYLAN, EIFEEL, LISP, PERL, PYTHON, sitios famosos como TWITTER, HULU y GROUPON han logrado posesionarse con éxito en la red mundial del internet. Además, por experiencia propia de programadores reconocen que la usabilidad de la tecnología Ruby es satisfactoria ya que la describen al código Ruby

elegante, potente y expresiva. Es muy fácil de utilizar gracias a su principio que es minimizar la confusión de los usuarios. Ruby es mejor para un programador que ya sabe un lenguaje o dos.

PYTHON: En un 100% esta tecnología ofrece versiones actualizadas de IDES, editores para el desarrollo de software (PYTHON: 3.2.2 liberada en septiembre 4, 2011), tiene como propósito enfatizar la productividad y legibilidad del código. Creada en 1995 por Guido Van Rossum, e influenciada por los lenguajes de programación ABC, ALGOL 68, C, C++, ICON, JAVA, LISP, PERL, sitios famosos como YOUTUBE, GOOGLE han logrado posesionarse con éxito en la red mundial del internet. Además, la usabilidad de Python es sorprendente la legibilidad que ofrece el lenguaje de programación. Python es ideal para principiantes, a menudo recomendado por los programadores debido a la simplicidad de su sintaxis.

Sección C: Soporte en el desarrollo de aplicaciones web

Características	Tecnologías de Desarrollo Web		
	JSP	**RUBY**	**PYTHON**
Reúsa componentes	2	2	1
Tiene flexibilidad	2	2	2
Soporta procesamiento	2	2	2
Soporta contenido dinámico	2	1	2
Independencia de la parte de diseño	2	1	1
Independencia de la parte lógica	2	1	1
Existen IDES para la tecnología	2	2	2
Existen editores para la tecnología	2	2	2
Total	**16**	**13**	**13**
Resultado	**2/2**	**1,63**	**1,63**
Equivalencia	**100%**	**81,5%**	**81,5%**

Autor: *Marcela Cuello Olmedo*

Conclusión Parcial Sección C:

JSP: Al 100% los componentes JSP son reusables en distintas plataformas (Unix, Windows), su flexibilidad es notable al ser Java el lenguaje de desarrollo que tiende a ser potente y escalable. Soporta contenido dinámico que refleja las condiciones del mundo real y existe independencia entre la parte del diseño (interfaz) y la lógica (programa). Para la tecnología JSP existe una gran cantidad de IDES disponibles y de gran uso, Ejemplo: **NetBeans** (un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo).

RUBY: Con un 81,5% de cumplimiento en soporte para el desarrollo, en esta tecnología la flexibilidad se ve favorecida ya que se puede modificar absolutamente todo dentro del ambiente. Limitada por la no existencia de independencia entre la parte de diseño y la lógica de programación.

PYTHON: La libertad y facilidad de reusar componentes aun limita a esta tecnología. Además de la independencia entre la parte de diseño y lógica. Con un 81,5% es una tecnología que permite el desarrollo de aplicaciones web con soporte para un buen procesamiento de trabajo, dinámico y flexible a nuevos requerimientos. Además, se apoya de la existencia de IDES y editores que facilitan el trabajo de los programadores.

Sección D: Peculiaridades de programación

Características	Tecnologías de Desarrollo Web		
	JSP	**RUBY**	**PYTHON**
No se compila (es un lenguaje de Scripting)	2	0	2
Tiene influencia de lenguajes de programación.	2	2	2
Lenguaje dinámico	2	2	1
Lenguaje Interpretado	2	2	2
Orientado a objetos	2	2	2
Sintaxis limpia	2	2	2
Trabaja con etiquetas especiales	2	1	1
Productivo	2	2	1
Código Expresivo	2	2	2
Persistencia de datos	2	2	2
Usa plantillas	2	2	1
Simplifica la construcción de las aplicaciones	2	2	1
Acelera la construcción de las aplicaciones	2	2	1
Total	**26**	**23**	**20**
Resultado	**2/2**	**1,77**	**1,54**
Equivalencia	**100%**	**88,50%**	**77%**

Autor: *Marcela Cuello Olmedo*

Conclusión Parcial Sección D:

JSP: Las peculiaridades de la programación en la tecnología JSP son notorias al proveer técnicas de desarrollo que sobresalen con respecto a otras. Es un lenguaje estructurado por tanto completamente orientado a objetos y respalda su trabajo al usar etiquetas HTML, plantillas con una sintaxis limpia y además se apoya de otros lenguajes de

programación, de esta manera permiten la construcción de aplicaciones web de forma simplificada y acelerada.

RUBY: Con un 88,5% de cumplimiento, esta tecnología tiene limitaciones al no ser un lenguaje de scripting, no se compila, es por eso que es un lenguaje de guiones o Scripts que es interpretado por un intérprete (en este caso el intérprete de Ruby) para ser ejecutado en tu computadora, mayormente trabaja con etiquetas especiales.

PYTHON: Con un 77% la productividad, la creación de aplicaciones web de manera simplificada y acelerada limitan a esta tecnología multiplataforma, lo que se resume a que un programador desarrolle aplicaciones rigurosamente de manera correcta.

Sección E: Elementos del lenguaje

Características	Tecnologías de Desarrollo Web		
	JSP	**RUBY**	**PYTHON**
Permite la declaración y el uso de variables	2	0	2
Define Tipo de datos	2	2	2
Define Operadores aritméticos	2	2	2
Define Operadores relacionales	2	2	2
Soporta comentarios	2	2	2
Define Tipos de datos complejos o especiales	2	1	1
Trabaja con estructuras de control	2	2	2
Trabaja con arreglos	2	2	1
Permite funciones definidas por el usuario.	2	1	2
Trabaja con métodos y paquetes	2	2	2
Trabaja con archivos	2	1	1
Total	**22**	**17**	**19**
Resultado	**2/2**	**1,55**	**1,73**
Equivalencia	**100%**	**77,5%**	**86,5%**

Autor: *Marcela Cuello Olmedo*

Conclusión Parcial Sección E:

JSP: Esta tecnología cumple con un 100% todas las características que debe poseer un lenguaje de desarrollo web multiplataforma del que se apoya (JAVA), no descuida elementos de programación que favorece eficientemente un desarrollo completo y respaldado técnicamente, declaraciones, variables, operadores, documentación a nivel de código, estructuras de control, librerías, paquetes y métodos actúan eficientemente en la producción de nuevos sistemas software.

RUBY: El uso de datos complejos, declaración de variables y manejo de archivos limitan a esta tecnología multiplataforma, aunque es lo que le caracteriza no se debe olvidar que un programador conoce de variables, datos complejos y archivos a nivel básico y estas características pueden desviar el uso adecuado de IDES o editores de Ruby, esta tecnología cumple con el 77,55 % de las características señaladas.

PYTHON: Con un 86,55% esta tecnología tiene características favorables a nivel de lenguajes de programación, los IDES o editores para Python permiten el uso de variables, tipo de datos simples y complejos, manejo de estructuras de control, arreglos, métodos, archivos, aunque al ser una tecnología que va cobrando fuerza en el mercado de desarrollo de software aún se limita el uso adecuado de tipos de datos complejos y el manejo de archivos.

ANEXO B

REGLAMENTO CEAACEESS

Reglamento CEAACES

El CEAACES es vital en la transformación de la educación superior

La labor que realiza el Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior, CEAACES, hay que entenderla como una política pública para garantizar una expansión de las IES con calidad académica y relevancia social.

A partir de la expedición de la nueva Ley Orgánica de Educación Superior publicada en el Registro Oficial el 12 de octubre de 2010 , en el Ecuador se vive la construcción de un nuevo modelo de la educación superior que tiende al mejoramiento significativo de las estructuras académicas y jurídico-administrativas de las instituciones de este nivel; pero, sobre todo, al incremento radical de la calidad de las carreras y de los programas de postgrado, en directa relación con su aporte a la solución de los problemas del país, en especial de los sectores que requieren urgentemente de estas soluciones.

El artículo 93 de la LOES vigente establece que el principio de la Calidad consiste en la búsqueda constante y sistemática de la excelencia, la pertinencia, producción óptima, transmisión del conocimiento y desarrollo del pensamiento mediante la autocrítica, la crítica externa y el mejoramiento permanente.

La Evaluación de la Calidad es el proceso para determinar las condiciones de la institución, carrera o programa académico, mediante la recopilación sistemática de datos cuantitativos y cualitativos que permitan emitir un juicio o diagnóstico, analizando sus componentes, funciones, procesos, a fin de que sus resultados sirvan para reformar y mejorar el programa de estudios, carrera o institución. La Evaluación de la Calidad es un proceso permanente y supone un seguimiento continuo.

La acreditación es una validación de vigencia quinquenal realizada por el Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior, CEAACES, para certificar la calidad de las instituciones de educación superior, de una carrera o programa educativo, sobre la base de una evaluación previa. La Acreditación es el producto de una evaluación rigurosa sobre el cumplimiento de lineamientos, estándares

y criterios de calidad de nivel internacional, a las carreras, programas, postgrados e instituciones, obligatoria e independiente, que define el CEAACES.

El procedimiento incluye una autoevaluación de la propia Universidad, así como una evaluación externa realizada por un equipo de pares académicos expertos, quienes a su vez deben ser acreditados periódicamente. El CEAACES es el organismo responsable del aseguramiento de la Calidad de la Educación Superior, sus decisiones en esta materia obligan a todos los Organismos e Instituciones que integran el Sistema de Educación Superior del Ecuador a cumplir con la Ley.

El Aseguramiento de la Calidad de la Educación Superior está constituido por el conjunto de acciones que llevan a cabo el CEAACES y las propias IES con el fin de garantizar la eficiente y eficaz gestión, aplicables a las carreras, programas académicos, a las instituciones de educación superior y también a los consejos u organismos evaluadores y acreditadores.

La clasificación académica o categorización de las instituciones, carreras y programas será el resultado de la evaluación. Hará referencia a un ordenamiento de las instituciones, carreras y programas de acuerdo a una metodología que incluye 46 indicadores y ponderaciones internacionales.

La planificación y ejecución de la autoevaluación está a cargo de cada una de las instituciones de educación superior, en coordinación con el CEAACES.

La Autoevaluación es un riguroso proceso de análisis que una institución realiza sobre la totalidad de sus actividades institucionales o de una carrera, programa o postgrado específico, con amplia participación de sus integrantes, a través de un análisis crítico y un diálogo reflexivo, a fin de superar los obstáculos existentes y considerar los logros alcanzados, para mejorar la eficiencia institucional y mejorar la calidad académica.

La evaluación externa es el proceso de verificación que el CEAACES realiza a través de pares académicos de la totalidad o de las actividades institucionales o de una carrera o programa para determinar que su desempeño cumple con las características y estándares de calidad de las instituciones de educación superior y que sus actividades se realizan en concordancia con la misión, visión, propósitos y objetivos institucionales o de carrera, de tal manera que pueda certificar ante la sociedad la calidad académica y la integridad institucional.

Para la emisión de informes de evaluación externa el CEAACES observará absoluta rigurosidad técnica y académica.

Las universidades y escuelas politécnicas, 57 en total, vienen trabajando en el proceso de autoevaluación para cumplir con la acreditación que exige la nueva Ley de Educación Superior vigente en el Ecuador.

El desafío principal de la evaluación y acreditación de la Calidad de la educación superior en el Ecuador es contribuir a la construcción de políticas de Estado en materia de ES que atiendan a: promover el desarrollo científico-tecnológico y el crecimiento económico; la formación de ciudadanos y profesionales capaces de trabajar para construir una sociedad más justa e integrada; que la educación superior se asuma efectivamente como tercer nivel del sistema educativo, contribuyendo al mejoramiento de la calidad y la pertinencia del sistema en su conjunto.

LEY ORGÁNICA DE EDUCACIÓN SUPERIOR (Ecuador)

Las instituciones de Educación Superior en el Ecuador, están bajo control de "El Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior, CEAACES, organismo público técnico, con personería jurídica y patrimonio propio, con independencia administrativa, financiera y operativa.

Funciona en coordinación con el Consejo de Educación Superior. Tiene facultad regulatoria y de gestión.

Misión del CEAACES: Ejercer la rectoría de la política pública para el aseguramiento de la calidad de la educación superior del Ecuador, a través de los procesos de evaluación, acreditación y categorización en las Instituciones de Educación Superior (R.O. 733 miércoles 27 de junio 2012, Pág. 7).

Visión del CEAACES: El CEAACES será un referente nacional y regional en la creación e implementación de metodologías integrales, articuladas y transparentes, de evaluación, acreditación y aseguramiento de la calidad de la educación superior (R.O. 733 miércoles 27 de junio 2012, Pág. 7).

Las instituciones de Educación Superior en el ecuador deben cumplir con algunos reglamentos que aseguren su participación en programas de educativos, que justifiquen sus servicios de calidad. Entre los cuales está el seguimiento a profesionales y graduados,

el uso de tecnologías libres y la participación continua de los estudiantes, en cada proceso universitarios.

Artículo de Seguimiento a graduados y egresados

TÍTULO VII

INTEGRALIDAD

CAPITULO 2

DE LA TIPOLOGÍA DE INSTITUCION

Sección Tercera

Del Funcionamiento de las Instituciones de Educación Superior

Art. 142.- Sistema de seguimiento a graduados. - Todas las instituciones del sistema de educación superior, públicas y particulares, deberán instrumentar un sistema de seguimiento a sus graduados y sus resultados serán remitidos para conocimiento del Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior.

Artículo del uso de programas informáticos

TÍTULO II

AUTONOMÍA RESPONSABLE DE LAS UNIVERSIDADES Y ESCUELAS POLITÉCNICAS

CAPÍTULO 2

PATRIMONIO Y FINANCIAMIENTO DE LAS INSTITUCIONES DE EDUCACIÓN SUPERIOR

AUTONOMÍA RESPONSABLE DE LAS UNIVERSIDADES Y ESCUELAS POLITÉCNICAS

Art. 32.- Programas informáticos. - Las empresas que distribuyan programas informáticos tienen la obligación de conceder tarifas preferenciales para el uso de las licencias

obligatorias de los respectivos programas, a favor de las instituciones de educación superior, para fines académicos.

Las instituciones de educación superior obligatoriamente incorporarán el uso de programas informáticos con software libre.

Artículo participación de estudiantes y graduados

TÍTULO III

EL COGOBIERNO

CAPÍTULO 2

DEL COGOBIERNO DE LAS UNIVERSIDADES Y ESCUELAS POLITÉCNICAS

Sección Tercera

De la participación de las y los estudiantes, las y los graduados, las y los servidores y las y los trabajadores en el cogobierno.

Art. 60.- Participación de las y los estudiantes. - La participación de las y los estudiantes en los organismos colegiados de cogobierno de las universidades y escuelas politécnicas públicas y privadas, en ejercicio de su autonomía responsable, será del 10% al 25% por ciento total del personal académico con derecho a voto, exceptuándose al rector o rectora, vicerrector o vicerrectora y vicerrectores o vicerrectoras de esta contabilización.

La participación de los graduados en los organismos colegiados de cogobierno de las universidades y escuelas politécnicas públicas y privadas, en ejercicio de su autonomía responsable, será del 1% al 5% del personal académico con derecho a voto, exceptuándose al rector o rectora, vicerrector o vicerrectora y vicerrectores o vicerrectoras de esta contabilización. Los graduados deberán tener como requisito haber egresado por lo menos cinco años antes de ejercer la mencionada participación. La elección de representantes estudiantiles y de los graduados ante los órganos colegiados se realizará por votación universal, directa y secreta. Su renovación se realizará con la periodicidad establecida en los estatutos de cada institución; de no hacerlo perderán su representación.

Para estas representaciones, procederá la reelección, consecutivamente o no, por una sola vez.

ANEXO C

ENCUESTA COMPROBACIÓN DE HIPÓTESIS

UNIVERSIDAD DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

ENCUESTA ACERCA DE LAS TECNOLOGÍAS MULTIPLATAFORMA

Objetivo: *Recolectar información técnica acerca de las tecnologías multiplataforma para el desarrollo web dinámicos.*

Dirigido a: *Expertos y desarrolladores de software. Estudiantes y profesionales de sistemas de la Unach.*

CUESTIONARIO

1. ***¿Desarrollar, implantar, documentar y mantener aplicaciones informáticas multiplataforma, implica utilizar tecnologías y entornos de desarrollo específicos que garantizan el acceso a los datos de forma segura y cumplir criterios de usabilidad y calidad exigidos en estándares establecidos?***

   SI ( ) *NO ( )*

2. ***¿Las Tecnologías multiplataforma permite el desarrollo de sistemas web, que sean sencillas de administrar y mantener por las organizaciones?***

   SI ( ) *NO ( )*

3. ***¿Desarrollar aplicaciones que se pueden ejecutar en cualquier ámbito de la web, es posible, por los avances de los lenguajes de programación para cada plataforma, sus técnicas de programación, métodos y procesos de desarrollo garantizan la creación de productos con alta eficacia comunicativa?***

   SI ( ) *NO ( )*

4. ***¿Se puede mejorar la eficiencia de un sistema web, si se utiliza la mejor tecnología orientada a la creación de sistemas web?***

   SI ( ) *NO ( )*

5. ***¿La flexibilidad de JSP radica en que es una tecnología que se apoya de un lenguaje (Java) potente y escalable que permite crear cualquier tipo de aplicaciones para cualquier plataforma tecnológica?***

   SI ( ) *NO ( )*

6. ***¿La tecnología JSP y el lenguaje de programación JAVA se complementan para el desarrollo completo y acelerado de aplicaciones web, elementos y técnicas de programación usadas en cualquier entorno de desarrollo multiplataforma garantizan la construcción de software web?***

   SI ( ) *NO ( )*

7. ***¿Las tecnologías Multiplataforma son la mejor alternativa para la creación de sistemas dinámicos de encuestas?***

   SI ( ) *NO ( )*

8. ***¿Se puede mejorar la eficiencia de un sistema web, si se utiliza la mejor tecnología orientada a la creación de sistemas web?***

   SI ( ) *NO ( )*

9. ***¿Las tecnologías Multiplataforma son la mejor alternativa para la creación de sistemas dinámicos de encuestas?***

   SI ( ) *NO ( )*

10. ***¿La tecnología JSP, facilita la presentación de la información en forma tabulada además de graficas estadísticas obtenida de la base de datos?***

    SI ( ) *NO ( )*

Gracias por su colaboración