

Ciudad de La Habana, Junio de 2009
“Año del 50 aniversario del triunfo de la Revolución”

Implementación de los módulos administración, gestión de solicitud y creación/eliminación de encuestas del sistema UCICuest.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Facultad 9

Autor: Miguel Martínez Ruiz

Tutor: Ing. Víctor Frank Molina López

DEDICATORIA

Les dedico esta tesis a todos los que de una forma u otra hicieron posible su realización, y el que hoy yo esté aquí:

A mis padres: que vean en esta tesis el fruto de sus esfuerzos, la culminación de años de espera y sueños realizados.

A mi novia: por su infinita paciencia y su inmenso amor.

A mi hermano: porque también es de él, por la solidaridad y por los consejos, aún aquellos que no sirvieron para nada pero al menos estaban ahí.

A toda mi familia: para que la sumen a todo el cariño que les tengo y sigan estando orgullosos de mí.

A mis amistades: debería incluirlos en la autoría y no en la dedicatoria, pero ocuparían demasiado espacio y “no vale la pena emborronar cuartillas” aunque sean en versión 2.0

A mis compañeros de aula: por soportarme durante 5 años.

A mis profesores: como prueba tangible de su trabajo, de su esfuerzo e infinita paciencia para conmigo.

AGRADECIMIENTOS

Toda obra, si se quiere ser realmente sincero, es el resultado de esfuerzo de muchos, aunque el mérito se le atribuya a uno solo. Es el saber aunar voluntades y esfuerzo de aquellos que confluyen en la búsqueda de un objetivo común, o que al menos han participado, de forma activa o pasiva, consciente o no, en la misma.

Ante todo, a Fidel Castro, por un millón de razones: por haber hecho una Revolución que permite a jóvenes como yo estudiar, sin importar ni origen, ni raza, ni posibilidades económicas, solo nuestro esfuerzo e inteligencia; por su idea de crear esta Universidad de Ciencias Informáticas, por el compromiso que entregó. Y por su ejemplo.

A mis padres: Si ellos no estaría aquí. No solo por haberme traído al mundo, sino por haberse sacrificado cada día para que yo pudiese estudiar. Por el ejemplo y la educación que me dieron, y dan. Por quererme y estimularme cada día, aun en los momentos más difíciles. Por estar ahí a mi lado.

A mi novia que además de amarme, me motiva a seguir estudiando, perfeccionando lo que otros me enseñan. Por las incontables horas tecleando y leyendo lo que iba haciendo. Por hacerme feliz y darme la tranquilidad que necesitaba mientras yo estudiaba.

A mi hermano Kendrick por todo el cariño y apoyo que me ha dado siempre, y por estar a mi lado.

A mis profesores: a ellos nunca podré agradecerles lo suficiente, pues el conocimiento que tan desinteresadamente me dieron y lo que ello representa en mi vida futura, no se puede expresar con simples palabras. La única forma posible es imitarles en su andar diario, en ser tan buen profesional como ellos, en servir a la Revolución y a otros de la misma manera que ellos lo hicieron conmigo.

A mis amistades por su afecto, por su apoyo, por los buenos ratos. Por ayudarme en cada momento cuando más lo necesitaba. Por ser leales.

RESUMEN

Actualmente el desarrollo de cualquier entidad o país depende en gran medida del estudio y análisis de las distintas variables que se presentan día a día, tanto humanas como materiales. Un método efectivo para conseguir información y su posterior procesamiento es la encuesta.

Se tomó como punto de partida para este trabajo una situación específica que demandaba gran cantidad de recursos humanos y materiales. A la hora de obtener datos y procesar información relacionada con el desarrollo de encuestas se originaba una situación problemática en la Universidad de las Ciencias Informáticas (UCI). Ambos procesos resultaban engorrosos y demorados en su aplicación al no contarse con un sistema automatizado capaz de realizar y publicar encuestas, motivando así la búsqueda de una solución.

Una vez definida esto se realizó un análisis minucioso de los principales conceptos y tecnologías a utilizar para cumplimentar dicha solución. Se analizó el uso de tecnologías sobre aplicaciones Web. Luego se hizo un estudio de los lenguajes de programación más apropiados para el desarrollo de los módulos, el framework y el entorno de desarrollo integrado (IDE) que se utilizarían.

Después se seleccionaron las herramientas necesarias para la obtención del producto final deseado, teniendo en cuenta también la metodología que se utilizaría para la implementación del sistema según las facilidades que pudiesen aportar al trabajo.

Se pasó entonces a la implementación de los módulos administración, gestión de solicitud y creación/eliminación de encuestas del sistema automatizado para la gestión y procesamiento dinámico de encuestas (UCICuest) usando tecnologías libres de desarrollo.

PALABRAS CLAVES

Sistema, encuestas, software libre, módulos.

ABSTRACT

The development of any entity or country nowadays depends greatly on the study and analysis of different variables that arise on day-to-day basis, both humans and materials. One effective method to gather and process information is the survey.

A specific situation was taken as starting point for this project, because it demanded the use of large amounts of human and material resources. When it was necessary to collect data and to process the information related to the implementation of surveys, a problematic situation used to arise at the Informatics Sciences University (UCI). They were demanding and slow processes when applied, since the university didn't have an automation system able to make and publish surveys, therefore a solution was needed.

Once this was defined, a throughout analysis was made on the main concepts and technologies to be used to find a solution. The use of technologies on Web applications was analyzed. Afterwards, a research was carried out on the most suitable programming languages for the development of the sub-system, the framework and the integrated development environment (IDE) to be used.

The following step was to choose the needed tools to achieve the desired final product, also considering the methodology to be used on the implementation of the system considering the advantages it could bring into the project.

The solution found was the implementation of the administration, application management and surveys creation/elimination modules of the management and dynamic processing of surveys automation system (UCICuest) using open source software and sophisticated programming methods.

Índice

Introducción	1
Capítulo 1	6
Fundamentación teórica	6
1.1. Introducción	6
1.2. Encuesta.....	6
1.2.1. Tipos de encuesta	6
1.2.2. La encuesta personal	9
1.2.3. La encuesta telefónica.....	10
1.2.4. La encuesta postal.....	12
1.2.5. La encuesta por Internet.....	14
1.3. Tipos de preguntas en las encuestas.....	15
1.3.1. Según la contestación que admitan:	15
1.3.2. Según su función en el cuestionario:	16
1.4. Informatización	16
1.4.1. Definición y tipología.....	16
1.4.2. Principios y recomendaciones para diseñar encuestas online.	22
1.5. Otras soluciones	26
1.5.1. PHPSurveyor.....	26
1.5.2. Mod_Survey	29
1.5.3. Conclusiones sobre las otras soluciones	32
1.6. Conclusiones	32
Capítulo 2	33
Tendencias y tecnologías actuales	33
2.1. Introducción	33
2.2. Aplicaciones Web	33
2.3. Lenguajes de programación del lado cliente	34
2.4.1 HTML.....	34

2.4.2	Javascript	35
2.4.	Lenguajes de Programación del lado servidor	36
2.5.1	ASP	36
2.5.2	JAVA	37
2.5.3	PHP	38
2.5.4	Fundamentación sobre la elección del lenguaje a utilizar	39
2.5.	Framework.....	40
2.5.1.	CakePHP.....	41
2.5.2.	Zend Framework.....	42
2.5.3.	Symfony	42
2.5.4.	Fundamentos para la selección del framework	44
2.6.	Arquitectura Cliente Servidor	45
2.6.1.	Patrón Arquitectónico Modelo-Vista-Controlador (MVC).....	46
2.7.	Entorno de desarrollo integrado (IDE).....	47
2.7.1.	Eclipse.....	47
2.8.2.	Zend Studio	48
2.8.3.	Fundamentación de la selección de IDE	48
2.9.	Servidor de aplicaciones Web.....	49
2.10.	Metodologías de desarrollo de software	51
2.10.1.	Extreme Programming (XP)	51
2.10.2.	Proceso Unificado de Desarrollo (RUP)	54
2.10.3.	Feature Driven Development (FDD)	59
2.10.4.	Fundamentación de elección de metodología de desarrollo	60
2.11.	Conclusiones.....	60
Capítulo 3	62
Descripción de la solución propuesta	62
3.1.	Introducción	62
3.2.	Tarjetas de Historias de Usuario.....	62
3.2.1.	Campos de la tarjeta Historia de Usuario.....	63

3.3. Tarjeta de tareas de programación	64
3.3.1. Campos de la tarjeta Tarea	64
3.4. Módulos a implementar en el sistema	65
3.5. Módulo Administración.....	65
3.5.1. Roles del sistema	66
3.5.2. Historia del Módulo	67
3.5.3. Tareas del Módulo	68
3.6. Módulo Gestión de Solicitud	71
3.6.1. Historia del Módulo	71
3.6.2. Tareas del Módulo	72
3.7. Módulo Creación/Eliminación de Encuestas	74
3.7.1. Tipos de Preguntas.....	75
3.7.2. Historia del Módulo	77
3.7.3. Tareas del Módulo	78
3.8. Conclusiones	80
Conclusiones	81
Recomendaciones	82
Referencias Bibliográficas	83

Figuras, Tablas y Gráficas

Figura 1 Diagrama de representación del Patrón MVC	47
Tabla 1 Tiempo de duración de las encuestas online.....	19
Tabla 2 Requisitos relacionados con la programación de la encuesta.	23
Tabla 3 Historia de Usuario.....	63
Tabla 4 Tarea de programación	64
Tabla 5 Historia de Usuario 01 Módulo Administración	68
Tabla 6 Tarea 01 Módulo Administración	68
Tabla 7 Tarea 02 Módulo Administración.....	69
Tabla 8 Tarea 03 Módulo Administración.....	69
Tabla 9 Tarea 04 Módulo Administración.....	70
Tabla 10 Tarea 05 Módulo Administración.....	70
Tabla 11 Tarea 06 Módulo Administración.....	71
Tabla 12 Historia de Usuario 02 Módulo Gestión de Solicitud	72
Tabla 13 Tarea 01 Módulo Gestión de Solicitud.....	72
Tabla 14 Tarea 02 Módulo Gestión de Solicitud.....	73
Tabla 15 Tarea 03 Módulo Gestión de Solicitud.....	73
Tabla 16 Tarea 04 Módulo Gestión de Solicitud.....	74
Tabla 17 Tarea 05 Módulo Gestión de Solicitud.....	74
Tabla 18 Historia de Usuario 03 Módulo Creación/Eliminación de encuestas	77
Tabla 19 Tarea 01 Módulo Creación/Eliminación de encuestas	78
Tabla 20 Tarea 02 Módulo Creación/Eliminación de encuestas	78
Tabla 21 Tarea 03 Módulo Creación/Eliminación de encuestas	79
Tabla 22 Tarea 04 Módulo Creación/Eliminación de encuestas	79
Gráfica 1 Clasificación de encuestados.	19
Gráfica 2 Patrones de conducta.....	20

Introducción

Una encuesta es un método de búsqueda sistémica de información partiendo de necesidades específicas que llevan a un investigador a la realización de preguntas a individuos en dependencia de los datos que espere obtener de ellos. Posteriormente, mediante el análisis de los resultados, obtendrá otros datos agregados. Con la encuesta se trata de obtener, de manera sistemática y ordenada, información sobre las variables que intervienen en una investigación, y esto sobre una población o muestra determinada. Esta información hace referencia a lo que las personas son, hacen, piensan, opinan, sienten, esperan, desean, quieren u odian, aprueban o desaprueban, o los motivos de sus actos, opiniones y actitudes.

A diferencia del resto de técnicas de entrevista la particularidad de la encuesta es que realizan a todos los entrevistados las mismas preguntas, en el mismo orden, y en una situación social similar; de modo que las diferencias localizadas son atribuibles a las diferencias entre las personas entrevistadas.

El desarrollo y la gran utilización de encuestas en la actualidad tiene su origen en los deseos de conocer los movimientos de la opinión pública y la predicción del voto a finales del siglo XIX, aunque no se debe olvidar las primeras utilizaciones de la encuesta en los trabajos de los reformadores ingleses del siglo XVIII, destacando entre éstas las investigaciones de Charles Booth sobre la pobreza, y las encuestas sobre condiciones sociales de trabajo en las grandes industrias realizadas por Weber a principios del siglo XX.

Los trabajos de Weber, a juicio de Lazarsfeld y Obershall, "anticipan, en todos los aspectos, la perspectiva moderna en el análisis de la conducta electoral, audiencia de radio, hábitos de compra, o cualquier otra acción realizada por grandes números de personas bajo circunstancias comparables". (1)

En la década del 30 del pasado siglo, el estadístico matemático George Gallup creó un sistema de encuestas para los medios masivos de comunicación apoyado en la noción que existía una amplia demanda periodística por las estadísticas sociales. Aunque la metodología debió ser adecuada a las circunstancias en que las encuestas se realizaban para lograr mejores resultados cualitativos de las mismas, hasta la actualidad se usan de forma generalizada, especialmente durante las campañas electorales en los Estados Unidos.

La necesidad de acceder a los datos de las encuestas en el menor tiempo posible, el lograr un procesamiento más rápido y efectivo, hizo que se implementase la automatización en los sistemas de encuestas. Esto, sumado a la creación de la red de redes, permitió el nacimiento y propagación de un nuevo tipo de encuesta: la encuesta online.

Las encuestas online se han convertido en una herramienta trascendental, con una potencialidad en ascenso, en los campos de la investigación social y de mercados que se realiza en todo el mundo en la actualidad. Por ello, es necesario tener en cuenta los factores de penetración de Internet en los hogares y empresas, el grado de avance tecnológico del país donde se vaya a realizar la investigación y el nivel de conocimiento de los internautas que se vayan a encuestar.

Teniendo en cuenta patrones básicos de eficiencia demostrados en innumerables encuestas online que se realizan a diario en todo el mundo, unido a su reducido coste, ha provocado un enorme crecimiento del sector en los últimos años, con el surgimiento de empresas de software y aplicaciones específicas. Aunque es justo señalar que este crecimiento ha prestado muy poca atención a los requisitos necesarios para crear encuestas online de calidad y reducir los errores de su diseño (muestreo, cobertura, medida y no respuesta).

Es importante puntualizar que no se puede perder de vista por aquellos que ya realicen o se planteen realizar encuestas online, que Internet hoy en día no es en su mayor parte accesible por todo tipo de dispositivos y programas destinados para ello, ni por todos los individuos que lo intentan. A su vez, el grado de complejidad es tal que gran parte de los individuos que acceden no consiguen los objetivos que pretenden por no disponer del conocimiento necesario para ello o bien se frustran por no verse cumplidas sus expectativas.

Por tanto se plantea que: “En muchos casos es necesario disponer de determinado tipo de dispositivos electrónicos, es casi imprescindible el uso del ratón, es necesario tener instalados determinados programas y conocer su uso y aplicaciones, etc. Todo ello, agravado por el inconveniente de que la gran mayoría de internautas no disponen de banda ancha ni de conexiones de alta velocidad”. (2)

No obstante y pese a todo lo anterior, Internet ofrece enormes ventajas frente a otros métodos de investigación como son la calidad de los resultados, el bajo coste y a la velocidad de análisis que no son comparables a las tecnologías tradicionales. (2)

Con esta tesis se intenta proporcionarle a los investigadores una herramienta automatizada que le permita hacer un estudio sobre la temática que le interese en ese momento, a la vez que mostrarle el alcance y las limitaciones de las encuestas online y por supuesto tener un referente normativo que indique cómo poder evitar los obstáculos inherentes a la propia Web y mejorar la calidad de las encuestas para evitar los errores derivados de su uso.

También es importante señalar que este sistema de encuesta se implementará sobre una plataforma de software libre teniendo en cuenta que esto adquiere una relevante significación en nuestro país. Entre sus principales ventajas está que representa la no utilización de productos informáticos que demanden la autorización de sus propietarios (licencias) para su explotación.

En el presente Cuba, en materia de sistemas operativos, se encuentra a merced de la empresa norteamericana Microsoft. Dicha transnacional tiene la capacidad legal de reclamar a Cuba que no siga utilizando el sistema operativo de su propiedad, amparada en leyes de propiedad industrial por las cuales también Cuba se rige. En caso de que tal eventualidad ocurriese podría provocar una interrupción del programa de informatización de la sociedad que como parte de la batalla de ideas está desarrollando el país. Otro aspecto a tener en cuenta es la posible implementación de una campaña de descrédito a la Isla, alegando el uso de la piratería informática por parte de las instituciones estatales cubanas.

También es necesario señalar que el software libre representa la alternativa para los países pobres, y es por concepción propiedad social, si se tiene en cuenta que una vez que comienza a circular, rápidamente se encuentra disponible para todos los interesados sin costo alguno o en su defecto a muy bajo costo.

Un motivo más para su utilización radica en el hecho que es desarrollado de forma colectiva y cooperativa, mostrando su carácter público y sus objetivos de beneficiar a toda la comunidad. Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por cuanto es operable en el mismo soporte de hardware con que cuenta el país.

La adquisición de cualquiera de sus distributivas puede hacerse de forma gratuita, descargándolas directamente de Internet o en algunos casos a muy bajos precios, se garantiza su explotación con un mínimo de recursos, por cuanto no hay que pagar absolutamente nada por su utilización (no requiere de licencia de uso, las cuales son generalmente muy caras), distribución y/o modificación. Permite su adaptación a los contextos de aplicación, al contar con su código fuente, lo cual garantiza un mayor por

ciento de efectividad, además la corrección de sus errores de programación y obtención de las actualizaciones y nuevas versiones. (3)

La obtención de datos y procesamiento de la información relacionada con el desarrollo de encuestas que se utiliza actualmente en la Universidad de las Ciencias Informáticas (UCI) es incompleto; por lo que constituye una **situación problemática** debido a que la Universidad no cuenta en este momento con un sistema automatizado que sea capaz de realizar y publicar encuestas. Esto da lugar a que el proceso de aplicación de las mismas sea muy engorroso y demorado.

Es necesario destinar gran cantidad de recursos humanos y materiales para poder recoger información referente a costumbres, nivel económico o cualquier otro aspecto de interés. No existe el mecanismo para guardar las estadísticas extraídas de las encuestas y los datos tienden a perderse con el tiempo.

Teniendo todos estos factores en cuenta, en el año 2006 se creó un proyecto para solucionar este problema, pero su desarrollo fue hecho sobre un software propietario, el cual no es funcional en la actualidad.

A la vez se comenzó a trabajar en el proyecto UCICUEST que además de también haber sido desarrollado sobre una plataforma de software propietario, no fue completado. Por su parte el directorio de televisión de la UCI (DTU) implantó una encuesta digital para recoger información referente a los medios y materiales publicados en la universidad. Carente de recursos esta encuesta no logró satisfacer todas las necesidades del DTU.

La experiencia adquirida por el DTU y los proyectos anteriores que recopilaron gran cantidad de información, a su vez mostrando las dificultades que para su implementación existían, motivó comenzar este proyecto, utilizando los conocimientos anteriores, y partiendo de la premisa de la necesidad de hacerlo sobre una plataforma de software libre, con todas las ventajas que el mismo trae aparejadas.

Por tanto surge la necesidad de dar solución a las situaciones antes expuestas y el **problema a resolver** es la inexistencia en la Universidad de las Ciencias Informáticas de un sistema automatizado para gestión de encuestas.

El **Objetivo General** es la implementación de los módulos administración, gestión de solicitud y creación/eliminación de encuestas que forman parte de un sistema gestión de encuestas (UCICuest) usando tecnologías libres de desarrollo.

De acuerdo a lo anterior se proponen una forma y mecanismo para desarrollar la aplicación, realizando el estudio de los siguientes **objetivos específicos**:

- Aplicar algoritmos eficientes para el procesamiento de la información.
- Implementar el algoritmo para controlar el acceso al sistema.
- Implementar el algoritmo para controlar la gestión de solicitudes de encuestas.
- Implementar el algoritmo para crear un tipo de encuesta.
- Implementar el algoritmo para gestionar un conjunto de preguntas de una encuesta.

Como **objeto de estudio** se establece el proceso de gestión de encuestas.

De ello se deriva que el **campo de acción** que abarca este trabajo, es el proceso automatizado de gestión de encuestas en la UCI.

Por tanto la **Idea a defender** que con el estudio de los sistemas de gestión de encuesta y los procesos de informatización de los mismos, se logrará la implementación de los módulos administración, gestión de solicitud y creación/eliminación de encuestas de un sistema para la gestión de encuestas sobre una plataforma de software libre en la Universidad de las Ciencias Informáticas.

Tareas de la investigación

- Identificar las distintas técnicas de programación a nivel mundial, nacional y en la Universidad.
- Valorar las tendencias, técnicas, tecnologías y metodologías relacionadas con las técnicas de programación actuales, así como de las plataformas de desarrollo que las soportan.
- Implementación del Módulo Administración
- Implementación del Módulo Gestionar Solicitud
- Implementación del Módulo Crear/Eliminar Encuesta

Capítulo 1

Fundamentación teórica.

1.1. Introducción

En el mundo actual el uso de encuestas con distintos objetivos se ha generalizado, mediante la utilización de sistemas automatizados que facilitan la gestión, análisis y procesamiento de la información recabada de forma rápida y eficiente. Este primer capítulo hará un recorrido panorámico sobre el método de encuestas, su situación actual, y los fundamentos teóricos para su implementación automatizada utilizando una plataforma de software libre, dada la factibilidad y ventajas tanto económicas como de mejoramiento que la misma ofrece.

1.2. Encuesta

La encuesta es una técnica cuantitativa utilizada en las investigaciones tanto de mercados, como de opinión pública, estudios médicos y en otra amplia gama de campos donde se necesita obtener información recurriendo a procedimientos estandarizados de interrogación con el fin de conseguir mediciones cuantitativas sobre una gran cantidad de características objetivas y subjetivas de una muestra de individuos representativa de la población o universo de forma que las conclusiones que se obtengan puedan generalizarse al conjunto de la población siguiendo los principios básicos de la inferencia estadística, ya que la encuesta se basa en el método inductivo, es decir, a partir de un número suficiente de datos se pueden obtener conclusiones a nivel general. La misma puede estar basada en aspectos objetivos (hechos, hábitos de conducta, características personales) o subjetivos (opiniones o actitudes).

1.2.1. Tipos de encuesta

Las encuestas se pueden clasificar según:

Fines científicos:

Atendiendo a los objetivos principales de la investigación es posible distinguir cinco tipos de encuestas:

- **Exploratoria:** es una investigación de reducidas dimensiones anterior a la investigación propiamente dicha, cuyo elemento definitorio es su consideración de prueba o ensayo y el carácter restringido de su planteamiento.
- **Descriptiva:** se pregunta por la naturaleza de un fenómeno social, y su objetivo es ofrecer una definición de la realidad, examinar un fenómeno para caracterizarlo del mejor modo posible o para diferenciarlo de otro.
- **Explicativa:** busca establecer las posibles causas o razones de los hechos, tratando de determinar las relaciones de causa y efecto entre los fenómenos.
- **Predictiva:** supone un "paso más" dentro de este desarrollo de tipos de investigación, puesto que es necesario conocer la explicación de los fenómenos antes de tratar de establecer una predicción de éstos.
- **Evaluativa:** cuyo fin es enjuiciar el grado de ejecución y los efectos conseguidos al aplicar un programa de acción social. (1)

Contenido:

Según su contenido se pueden definir tres tipos de encuestas:

- Encuestas referidas a hechos.
- Encuestas referidas a opiniones.
- Encuestas referidas a actitudes, motivaciones o sentimientos. (1)

Procedimiento de administración del cuestionario:

En la actualidad, existen cuatro tipos de encuesta que se dividen según el medio que se utiliza para realizarlas:

- **Personal:** Consisten en entrevistas directas o personales con cada encuestado.
- **Telefónica:** Este tipo de encuesta consiste en una entrevista vía telefónica con cada encuestado.

- **Postal:** Consiste en el envío de un cuestionario a los potenciales encuestados, pedirles que lo llenen y hacer que lo remitan a la empresa o a una casilla de correo. Para el envío del cuestionario existen dos medios:
 - El correo tradicional.
 - El correo electrónico (que ha cobrado mayor vigencia en los últimos años).
- **Internet:** Este tipo de encuesta consiste en colocar un cuestionario en una página Web o en enviarlo a los correos electrónicos de un panel predefinido. (4)

Dimensión temporal:

Según su dimensión existen cuatro tipos de encuestas:

- **Transversales:** son aquellas que analizan un momento temporal.
- **Longitudinales:** son las que analizan varios momentos temporales.
- **Retrospectivas y prospectivas:** son aquellas que se desplazan en el tiempo, antes y después.
- **Diseño de tendencias, de panel y de cohorte:** son aquellas que analizan patrones independientemente del momento.

Finalidad:

Según su finalidad se pueden clasificar en:

- **Político-sociales.**
- **Comerciales.**
- **Encuestas con fines específicos (1)**

De todas estas clasificaciones la más conocida y la que más se utiliza es la del “Procedimiento de administración del cuestionario”. Estos diferentes tipos de encuesta presentan ventajas e inconvenientes que determinan el que su aplicación sea más recomendable ante determinadas situaciones. A

continuación se resumen las principales ventajas y limitaciones de la encuesta personal, telefónica, postal y de Internet.

1.2.2. La encuesta personal

La encuesta personal es quizás el método que goza de mayor popularidad y el que se ha utilizado con mayor profusión en la captación de información primaria debido, principalmente, a las ventajas que presenta frente a los otros tipos de encuesta. Consiste en una entrevista personal que se establece entre dos personas, a iniciativa del entrevistador, para obtener información sobre unos objetivos determinados. (5)

Las principales ventajas de las encuestas realizadas personalmente son las siguientes:

- Entre los distintos métodos de encuesta, las personales son las que proporcionan un mayor índice de respuesta, ya que cuando se contacta con las personas a encuestar es poco probable que declinen responder el cuestionario o no lo concluyan una vez comenzado. (5)
- Son fiables puesto que se conoce con certeza quién contesta y se evita la influencia de terceras personas.
- Se obtienen respuestas menos evasivas e inconcretas, ya que el entrevistador puede aclarar cualquier tipo de dudas que se puedan suscitar en el cuestionario y se reducen de forma considerable las típicas respuestas de no sabe no contesta. (5)
- Permiten utilizar materiales auxiliares para profundizar en determinados temas, tales como fotografía, láminas, productos, etc.
- También pueden obtenerse datos secundarios del entrevistado como presencia, ambiente familiar, sexo., etc.

En contrapartida, las encuestas personales presentan como principales inconvenientes:

- Un coste elevado, ya que al tiempo necesario para realizar la entrevista hay que añadirle el tiempo que el entrevistador necesita para desplazarse hasta el lugar de residencia de la persona a entrevistar que, junto con el importe de los viajes, representa un gasto importante por entrevista realizada.

- Pueden originarse sesgos por influencias del entrevistador, es decir, la actuación del entrevistador en la encuesta, su presencia física, sus explicaciones, su entonación, etc. pueden representar una fuente de distorsión en la información obtenida. Para evitar estos posibles sesgos, es importante que en la selección los entrevistadores se tengan en cuenta ciertas cualidades personales, honestidad, objetividad, neutralidad y facilidad para los contactos humanos, siendo también importante una buena formación de carácter general sobre la técnica de la entrevista, formas de presentación, aclaración de respuestas, etc., además de las instrucciones concretas de cada investigación en particular que aunque traten sobre el mismo tema emplean cuestionarios y procedimientos de búsqueda de las unidades muestrales diferentes.
- Teniendo en cuenta que la base de la investigación descansa en la veracidad de la información recogida, es necesario controlar que los datos presentados por los entrevistadores son ciertos y que proceden realmente de la muestra seleccionada. Para verificar esto, en las encuestas personales es necesario un exhaustivo control de los mismos, a través de una revisión de una pequeña muestra de las encuestas realizadas por cada encuestador.
- En lo que a la realización de las encuestas personales se refiere pueden llevarse a cabo bien en el hogar de la persona encuestada o "in situ". Las encuestas efectuadas en el hogar se utilizan en investigaciones de tipo general, garantizándose la aleatoriedad de los hogares seleccionados a través de diferentes métodos de muestreo. Las encuestas "in situ" se realizan en diferentes lugares relacionados con el objetivo de la investigación (establecimientos estaciones de autobuses, etc.) con la finalidad de efectuar la encuesta en una situación más real. (5)

1.2.3. La encuesta telefónica

La encuesta telefónica es un método cuya utilización va en aumento en los últimos años a medida que se incrementa el número de hogares con teléfono y mejora, por tanto, su representatividad. Inicialmente se utilizó para realizar test de audiencia de programas y anuncios emitidos por radio y televisión, pero posteriormente se ha generalizado su uso en la captación de información.

Desde el punto de vista de la investigación de mercados la encuesta telefónica puede utilizarse en tres sentidos:

- Como único medio de realizar la investigación.
- Como una técnica combinada con otras, es decir, se puede diseñar una encuesta para realizarse personalmente, telefónicamente y por correo en función de las características y condicionantes de la investigación y de la población de la cual se pretende obtener la información.
- Como un medio auxiliar en la utilización de otras técnicas como, por ejemplo, para establecer citas en la realización de encuestas personales, informar o reclamar el envío por correo de cuestionarios, completar cuestionarios, realizados a través de encuestas postales, como medio de control de los encuestadores en entrevistas personales o telefónicas, etc. (5)

Entre las principales ventajas que presenta la encuesta telefónica se pueden destacar las siguientes:

- Es una técnica que permite obtener información con gran rapidez, ya que en pocas horas un encuestador puede realizar múltiples encuestas.
- En comparación con la encuesta personal su coste puede resultar más reducido, aunque esto depende de diversos factores tales como la duración del cuestionario, la distancia, las tarifas telefónicas, etc.
- Permite acceder más fácilmente a personas ocupadas, sobre todo para encuestas de corta duración, a las cuales suele ser difícil llegar a través de encuestas personales.
- En relación al índice de respuesta, el número de personas que aceptan contestar por teléfono una encuesta suele ser bastante elevado, aunque en los últimos años la rápida expansión de Tele-marketing "dirigido" está provocando una disminución de la disposición del público a cooperar en las encuestas telefónicas como consecuencia del aumento de llamadas telefónicas, muchas veces indeseadas, y la confusión que puede producirse en los individuos entre el tele-marketing y los estudios de investigación de mercados. Por ello, es fundamental dar una explicación clara y convincente de que se trata de un estudio de investigación, sin ninguna intención de vender nada, informarle del objetivo de la investigación, quienes realizan y de cómo esa persona ha sido seleccionada. (5)

Los principales inconvenientes o limitaciones de la encuesta telefónica son:

- Los posibles problemas de falta de representatividad muestral como consecuencia de que sólo pueden ser encuestadas las personas que disponen del teléfono, si bien en el caso de que el

universo lo constituyan las empresas no existe este problema pues prácticamente el cien por cien de ellas poseen teléfono. Con respecto a la selección de los números telefónicos es conveniente generarlos aleatoriamente a través de un directorio telefónico pues hay números que no están registrados y no incluye los nuevos abonados.

- Otro de los aspectos que supone una limitación de este método es la brevedad de la entrevista que se recomienda que no exceda de 15 minutos. Ello implica que el cuestionario también sea breve y, por tanto, no puede recabarse demasiada información.
- El no poder utilizar material auxiliar o de exhibición, como tarjetas, fotografías, bocetos, etc. limita también bastante el tipo de información que se puede obtener a través de las encuestas telefónicas.
- La brevedad de la entrevista exige que el cuestionario sea reducido y no contenga demasiadas preguntas.
- Las preguntas deben ser preferentemente cerradas o semi-abiertas y con un abanico de respuestas no muy amplio.
- Las preguntas de respuesta múltiple se deben eliminar o desglosar en sucesivas preguntas.

1.2.4. La encuesta postal

La encuesta postal consiste en el envío por correo de un cuestionario a las personas que constituyen la muestra con la esperanza de que por la misma vía lo devuelvan cumplimentado. Como técnica de entrevista en algunas ocasiones puede tener una utilización esencial cuando es el único medio de establecer contacto con los encuestados y en otras situaciones puede utilizarse como técnica opcional en función de las características de la investigación.

Entre las principales ventajas de la encuesta postal se deben destacar:

- Su reducido coste frente a los otros dos métodos de encuesta, pues es un sistema económico de recogida de información especialmente cuando el ámbito geográfico de la investigación des a nivel nacional o internacional.

- Es un sistema más flexible para el entrevistado que puede contestar el cuestionario en el momento más conveniente y empleando el tiempo que desee, lo cual puede llevar a respuestas más meditadas.
- Se evitan los posibles riesgos que puedan originarse por la influencia del encuestador en el proceso de realización de la encuesta.

En lo que se refiere a las desventajas e inconvenientes de la encuesta postal se destacan como más importantes:

- El bajo número de respuestas que se reciben en relación con los envíos realizados, que se sitúa en torno al 10% o 20%. Para tratar este problema del bajo índice de respuesta es necesario establecer una serie de mecanismos motivadores, como son, en primer lugar que junto con el cuestionario se envíe una carta de representación en la que se ponga de manifiesto la necesidad de la colaboración del encuestado y la importancia que tiene su respuesta para los fines de la investigación. En segundo lugar, otro mecanismo que incrementa el índice de respuesta es ofrecer un incentivo material, además del incentivo moral que se pretende dar con la carta. En este sentido, se han utilizado diversas formas como, por ejemplo, ofrecer una participación en sorteos a las personas que respondan, regalos, vales de descuento sobre determinados productos, facilitarlos resultados de la investigación, etc.
- La identidad del encuestado es una variable incontrolada, debido a que no se puede asegurar que realmente conteste el cuestionario el destinatario del mismo y, al mismo tiempo, también pueden existir influencias de otras personas en las respuestas del encuestado y ello, lógicamente, introduce distorsiones en la muestra.
- Las encuestas postales están limitadas a situaciones en las que se dispone de un directorio del universo a investigar, lo cual no siempre es frecuente en las investigaciones.
- Otro de los problemas que motiva la ausencia de un entrevistador hace referencia a la poca cantidad de información que se puede obtener a través del cuestionario, que necesariamente ha de ser breve, salvo que los encuestados estén emocionalmente involucrados en el tema a investigar. Si bien no existe una regla general sobre la extensión del cuestionario, normalmente la brevedad del mismo incrementa el número de respuestas.

Finalmente, se debe señalar los problemas asociados a la representatividad de la muestra, ya que la falta de respuesta puede dar lugar a que la estructura de la muestra real que se obtiene de los cuestionarios contestados no se ajuste a estructura de la muestra definida, siendo necesario o bien enviar nuevos cuestionarios a los estratos de la muestra de los que no se han recibido suficientes respuestas, o completar las respuestas necesarias mediante entrevistas telefónicas o personales o bien desechar un determinado número de encuestas de aquellos estratos de la muestra cuyas respuestas fueron más numerosas, incrementándose en este último caso el error muestral. (5)

1.2.5. La encuesta por Internet

La encuesta por Internet consiste en el envío por correo electrónico, o la inclusión en un sitio Web, a donde el encuestado pueda acceder con facilidad y responder con la misma celeridad, en caso de que sea de su interés. El incremento exponencial del número de personas con acceso a Internet, con la limitante que este se concentra en los países desarrollados, hace de este medio el más atractivo, por su coste ínfimo con respecto a los medios anteriormente expuestos, al volumen de personas al que se puede acceder y a la reducción de costos de personal que este representa. Entre las principales ventajas de la encuesta por Internet se destacan:

- Se cruzan las fronteras físicas y se llega a lugares distantes mediante el uso de Internet posibilitando un alcance global.
- Es posible llegar a todas las personas en cuestión de segundos, sin recurrir a planillas o cuestionarios impresos. En su lugar los usuarios pueden opinar mediante la Web.
- Los encuestados interactúan directamente con la encuesta automatizada, sin necesidad de la intervención de un encuestador o intermediario, lo que garantiza que las respuestas no sean alteradas.
- Disminuye sustancialmente el tiempo del proceso de recolección de datos y se elimina el proceso de transcripción de la encuesta.
- El procesamiento básico de la información es de forma inmediata.
- Se obtienen resultados parciales sin haber culminado el proceso, lo que permite la toma de decisiones anticipadas y acertadas, garantizado por la veracidad de los datos.

- El acceso a las consultas en línea vía Web es posible de acuerdo a los perfiles definidos bajo el módulo de seguridad de datos.
- A través de reportes personalizados, de una forma fácil y rápida, se realizan análisis y se llegan a conclusiones que manualmente demorarían mucho tiempo.
- Se brindan reportes de salidas de gran utilidad para la comprensión de los resultados posibilitando la toma de decisiones.

Por otra parte, presenta además desventajas e inconvenientes, entre las que sobresalen:

- Imposibilidad de conocer a ciencia cierta la identidad de los encuestados, o que estos puedan responder si apegar a sus experiencias personales.
- Individuos que carecen de la experiencia y los conocimientos sobre ordenadores no saben cómo introducir o borrar cierto tipo de respuestas como por ejemplo las casillas de respuesta única (radio buttons) a diferencia de las múltiples (checkbox) en las que para borrar una respuesta basta con pulsar encima de la introducida.
- No saber qué hacer con un menú desplegable.
- No ser capaz de ver todas las opciones de respuesta sin necesidad de utilizar la barra lateral de desplazamiento.
- Verse forzado a responder a una pregunta aunque ninguna de las categorías de respuesta fuera apropiada según su opinión.
- No saber cómo cerrar y finalizar el cuestionario que acababan de responder.
- Tener que realizar varias interacciones para responder a cada pregunta (por ejemplo, al hacer clic en la respuesta, tener que desplazar la barra lateral para poder visualizar el "Continúe con la siguiente pregunta" y poder así pinchar en el botón para que aparezca.

1.3. Tipos de preguntas en las encuestas

1.3.1. Según la contestación que admitan:

Abiertas: preguntas que sólo formulan las preguntas, sin establecer categorías de respuesta. Se deben utilizar muy poco en las encuestas porque después de la encuesta hay que cerrarlas y luego estandarizarlas.

Cerradas: Dicotónicas son las que establecen sólo 2 alternativas de respuesta, "Si o No" y a veces (Ns/Nc). Se deben utilizar sólo para temas muy bien definidos que admiten estas 2 alternativas como respuesta.

1.3.2. Según su función en el cuestionario:

Filtro: se utilizan mucho en los cuestionarios para eliminar aquellas personas que no les afectan determinadas preguntas, es decir que marcan la realización o no de preguntas posteriores.

Batería: todas las preguntas tratan sobre un mismo tema y que siempre deben ir juntas en el cuestionario en forma de batería, empezando por las más sencillas y luego las más complejas. Esto se denomina "embudo de preguntas".

De control: se utilizan para comprobar la veracidad de las respuestas de los encuestados y normalmente lo que se hace en estos casos es colocar la misma pregunta pero redactada de forma distinta en lugares separados una de la otra. (6)

1.4. Informatización

A lo largo de varios años de experiencia en el uso de las encuestas online como herramienta de investigación social y de mercado, académicos y empresas han demostrado que Internet puede ser un recurso muy valioso y que, a pesar de sus limitaciones metodológicas, constituye una fuente de datos cada vez más utilizada para todo tipo de estudios.

El período de tiempo en el que ha surgido esta metodología ha sido relativamente corto y aún se encuentra en una fase de desarrollo intensivo. Automatización.

1.4.1. Definición y tipología

Existen diversas definiciones de los términos "encuesta online" o "encuesta por Internet"; no obstante, se propone una definición alternativa que englobe y actualice las anteriores:

"Encuesta online es todo tipo de encuesta auto-administrada en la que el encuestado envía la información por Internet a través de un dispositivo electrónico preparado para tal efecto". (2)

Por supuesto, esta definición no engloba aquellas encuestas en las que el encuestado recibe la información o invitación a la encuesta por Internet y ha de enviarla por otro canal (por ejemplo por fax o por correo), pero sí aquellas en las que el encuestado recibe la invitación a participar por cualquier otro canal (por ejemplo mediante una invitación postal) y ha de enviar la información por Internet. Es importante que la información se envíe utilizando Internet como canal de transmisión, ya que, de otro modo, no podría considerarse una encuesta como online.

Los "Estándares de calidad para la realización de encuestas por Internet" redactados en Mayo de 2001 por la ADM¹ proponen una definición que es necesario actualizar, en la que hacen referencia a los diversos tipos de interacción entre el entrevistado y el cuestionario. De este modo, se definen cuatro tipos de encuestas online:

- Aquellas en las que el encuestado rellena un formulario online localizado en el servidor del instituto de investigación (al que puede acceder por diversos métodos).
- Aquellas en las que el encuestado se descarga un archivo que contiene la encuesta, la rellena y la reenvía a través de su correo electrónico.
- El cuestionario situado en un servidor remoto se responde de manera "local" y se reenvía a través del programa de correo electrónico.
- El encuestado recibe un correo electrónico con el archivo del cuestionario que ha de descargar, rellenar y reenviar en un nuevo correo como archivo adjunto.

El autor Couper propone una tipología basada en la probabilidad de selección de los encuestados, en la que distingue entre: encuestas probabilísticas aquellas en las que se conoce la probabilidad de selección de la muestra y no probabilísticas, con los siguientes ejemplos:

El aumento de la calidad de la respuesta en las encuestas online es un tema que ha preocupado siempre a las empresas e investigadores y que cuenta con escasa literatura y estudios para optimizar las formas de actuación. Algunos autores han comenzado a testar la influencia de ciertos formatos de diseño en parámetros como la tasa de respuesta, pero se limitan a mostrar aumentos o disminuciones de esta tasa, sin estudiar otros factores que puedan afectar a la calidad de la respuesta.

¹ ADM: Asociación Alemana de Institutos de Investigación de Mercado y Opinión.

Existe suficiente evidencia empírica que demuestra que los factores más significativos para aumentar la tasa de respuesta de una encuesta online enviada por email son el reenvío de un recordatorio, los incentivos y la longitud del cuestionario Deutskens No obstante, y aunque no exista evidencia empírica de ello, el diseño y la accesibilidad a la encuesta son factores tan importantes o más que los anteriormente mencionados. (2)

Respecto al envío de emails recordatorios, se ha demostrado que enviar múltiples correos con la encuesta a pesar de tener un bajo coste y ser rápido, puede provocar una reacción contraria a potenciales participantes ya que éstos considerarán el email como correo spam. Según varios autores entre ellos lleva el tiempo medio de retraso para responder a una encuesta enviada por email es de 5,59 días, a diferencia de los 12,21 días de las encuestas por correo tradicional. El estudio llevado a cabo por Deutskens demostró que un envío de recordatorios más temprano tiene una mayor tasa de respuesta (21,2%) que otro realizado posteriormente (19,5%), aunque no se observaron diferencias en la calidad de las respuestas entre ambos envíos. (2)

En cuanto a los incentivos, compararon la influencia de los siguientes incentivos en la calidad de la respuesta: incentivos monetarios (descuentos para compras) con loterías (sorteos) y donaciones a ONGs². Las donaciones tuvieron una menor tasa de respuesta que el resto de incentivos, no obstante, se demostró que aquellos sorteos en los que el premio es menos cuantioso pero hay mayores probabilidades de obtenerlo se alcanza mayor tasa de respuesta. Por otro lado no se observó ninguna influencia del incentivo en la calidad de la respuesta.

Por último, estos autores demostraron empíricamente que las versiones cortas del cuestionario obtienen una tasa de respuesta significativamente mayor. Pointer propone la siguiente tabla de recomendación para los tiempos de duración de las encuestas online en función del tipo de entrevista: (2)

Encuesta Pop-up	5 minutos
Estudio Ad Hoc	5 a 10 minutos

² ONG: Organizaciones No Gubernamentales.

Estudio Panel	10 a 20 minutos
Estudio a empleados	10 a 20 minutos
Panel de especialistas / amplio incentivo	10 a 30 minutos

Tabla 1 Tiempo de duración de las encuestas online.

En cuanto a la influencia de la frecuencia y honestidad del encuestado en relación a la calidad de la respuesta, Pointer clasifica a los encuestados en cuatro grupos:

- Ocasionales
- Infrecuentes
- Frecuentes
- Dishonestos

Y diseña la siguiente gráfica:



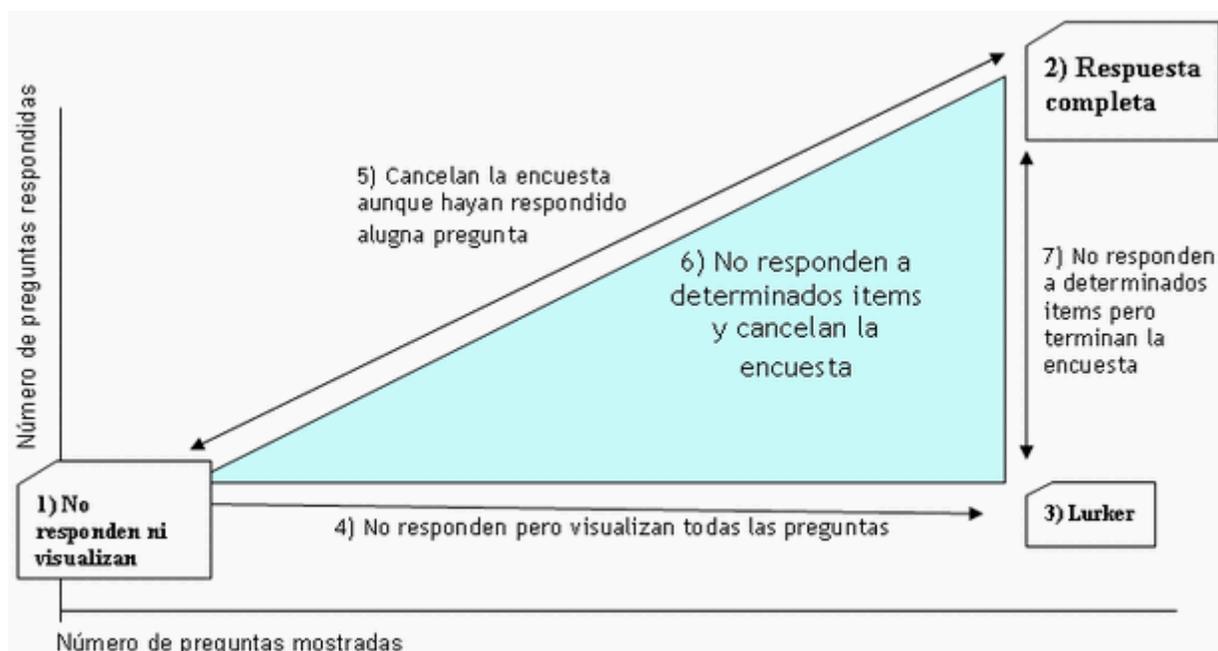
Gráfica 1 Clasificación de encuestados.

El grupo más preocupante de esta clasificación son los encuestados que se sitúan en el entorno denominado "fraudulentos" ya que completan muchas encuestas con tendencia a dar respuestas incorrectas, o bien por obtener el incentivo o bien por malicia. En cambio los voluntarios profesionales (o

también llamados caza-incentivos) son aquellos que completan entrevistas honestamente por obtener un incentivo y son mucho más frecuentes que los anteriores. (2)

Los situados en el cuadrante "rechazo" son aquellos que no quieren hacer encuestas online (o de cualquier otro tipo) por diversos motivos. En el caso de que este colectivo realice encuestas personales o telefónicas (no auto-administradas) tenderán a ser más deshonestos en las preguntas.

Bosnjak clasifica a los encuestados online en función de siete patrones de conducta con la encuesta en la siguiente gráfica: (2)



Gráfica 2 Patrones de conducta.

Como puede verse, el eje horizontal muestra el número de preguntas mostradas en la encuesta, y el eje vertical el número total de preguntas respondidas. Los encuestados se sitúan en algún punto de la gráfica en función del número de preguntas que hayan visualizado y las que han respondido, dando así lugar a las siete clasificaciones. Las tres clasificaciones de encuestados más extremas son:

- Aquellos que no responden nada ni visualizan nada del cuestionario clasificados por Pointer como "Rechazo".

- Aquellos que responden a todas las preguntas.
- Aquellos que visualizan todo pero no responden a ninguna de las preguntas (en inglés: Lurkers). En el área formada por ese triángulo se encuentran aquellos individuos que cancelan el cuestionario sin haberlo finalizado.

Al analizar el gráfico se pueden extraer tres variables que expliquen el comportamiento del encuestado: su motivación, la oportunidad de responder y la habilidad para ello. Esto significa que un encuestado puede estar motivado, tener la oportunidad, conocimientos y habilidades necesarias y contestar todo el cuestionario, en cambio, puede que esté motivado para visualizar el cuestionario pero no responder a ninguna pregunta o en el caso de que haya preguntas de respuesta obligatoria responder al azar con tal de obtener el incentivo.

Para solucionar problemas relacionados con la seguridad, los estándares de calidad establecidos por la ADM recomiendan informar al encuestado sobre las políticas de privacidad, sus derechos y el uso que se dará sobre sus datos personales de manera clara y visible. Puede que el encuestado esté motivado para responder pero que tenga dificultades técnicas o no tenga la habilidad y conocimientos necesarios para ello lo cual provoca la cancelación de la encuesta, y también puede darse el caso de que haya motivación pero no haya un sentimiento de seguridad y por tanto se deje el cuestionario en blanco.

Aquellos individuos que completan la encuesta pero dejan determinados ítems en blanco es porque no desean proporcionar determinada información por Internet, en cambio aquellos que comienzan la encuesta respondiendo a todos los ítems pero que la cancelan sin finalizarla, es o bien por dificultades técnicas, o bien porque ellos mismos decidieron su cancelación por otros motivos.

La variable oportunidad hace referencia a la situación de cada individuo frente a su encuesta online, quien en ocasiones puede no tener el tiempo, los medios, o la información necesaria para rellenarla.

La tipología presentada por Bosnjak, así como la clasificación de Pointer son muy útiles a la hora de definir patrones de conducta con el fin de mejorar la calidad de la respuesta y evitar que si hay encuestados que deciden no responder, no sea por motivos inherentes al cuestionario, por falta de información o por un diseño demasiado complejo. (2)

Un elemento importante es el diseño de proyectos en modo mixto, ofreciendo al encuestado la posibilidad de responder a la encuesta en un formato que no sea online, pero controlando a su vez que no haya duplicación de respuestas. Este modelo es más costoso, pero reduce el error de no-respuesta.

Como han demostrado varios autores, el envío de notificaciones antes de la encuesta, de emails personalizados, de recordatorios en el tiempo adecuado, así como el diseño simple, usable y no muy recargado de la encuesta o unas instrucciones claras y precisas, son factores imprescindibles para aumentar la tasa de respuesta y mejorar la calidad de la misma.

1.4.2. Principios y recomendaciones para diseñar encuestas online.

Programación de la encuesta

El grupo de trabajo NEON, dentro del gremio de Investigadores de mercado y opinión diseñó en el 2003 un documento con los requisitos que todo software debería tener para poder realizar encuestas online de calidad clasificados en "obligatorios" "necesarios" y "recomendables". Estos requisitos relacionados con la programación de la encuesta, pueden resumirse en los siguientes: (2)

Requisitos		
Obligatorios	Necesarios	Recomendables
Compatibilidad de la encuesta con todas las versiones actuales y futuras de navegadores a nivel de funcionalidades y apariencia	Compatibilidad de la encuesta con versiones antiguas de navegadores a nivel funcional	Menor dependencia del ratón en respuestas e interfaces
Recolección exhaustiva de las visitas a la página de la encuesta	Siempre que sea posible, ofrecer una versión del cuestionario que no emplee Javascript	Texto editable de protección de datos y seguridad del encuestado
Exclusión de las encuestas repetidas	Si se usa Javascript, que el	Saltos, filtros y condiciones

por medio del control de cookies, el número IP o un login al inicio	encuestado pueda controlarlo desde el comienzo de la encuesta	elaboradas
Posibilidad de incluir elementos multimedia	Posibilidad de editar y modificar el script de los cuestionarios	Protección opcional mediante contraseña
Preguntas editables en HTML y visualizables con vistas previas	Bibliotecas de preguntas para reducir tiempos de programación	Checks de consistencia
El cuestionario ha de ser editable en varias páginas html	Integración de hojas de estilo css	Botón de retroceso opcional
Posibilidad de rotar las preguntas y respuestas aleatoriamente	Es necesario poder elaborar todo tipo de preguntas	
Seguridad: Encuesta almacenada en servidores seguros, con acceso encriptado y firewall	Monitoreo de los datos externo	Separación de los datos del encuestado y sus respuestas

Tabla 2 Requisitos relacionados con la programación de la encuesta.

Puntos importantes a tener en cuenta para el investigador antes de realizar encuestas online.

Las asociaciones alemanas de institutos de investigación de mercado y opinión, redactaron en el año 2001 al mismo tiempo que los estándares de calidad, una lista de puntos importantes que el investigador tenía que tener en cuenta antes de realizar encuestas online. Estas recomendaciones son perfectamente aplicables hoy y su objetivo es facilitar al investigador las preguntas relevantes que le permitan juzgar la calidad de las distintas ofertas de investigación online de los institutos que contemple. (2)

Requisitos Generales

- ¿Posee el instituto proveedor de encuestas online la suficiente experiencia, conocimientos y fuentes técnicas necesarias para la investigación? ¿Cumplen los empleados con las aptitudes científicas requeridas, tanto en grado de formación como en experiencia? (2)

Requisitos para la investigación

- ¿Está bien determinado y es accesible por Internet el grupo de análisis o universo de la investigación?
- ¿El proceso de selección de los participantes está claramente definido y el acceso a los encuestados se podrá hacer de forma individual?
- ¿Es válido el proceso de selección de encuestados/as efectuado para obtener resultados representativos del universo de estudio?
- ¿La selección de individuos se hará mediante una captación activa (siguiendo un modelo de muestreo aleatorio o por cuotas) o por el contrario de forma pasiva, es decir por autoselección?
- En el caso de obtener una baja respuesta en una selección aleatoria, ¿es posible demostrar a posteriori mediante los correspondientes datos de referencia si la muestra obtenida reproduce adecuadamente las características y la estructura de la población objeto de estudio?
- En caso necesario, ¿es posible realizar una ponderación de los datos? ¿Se documentarán adecuadamente los procedimientos de la misma? (2)

Ejecución de la investigación

- ¿Se prestará atención a los requisitos metodológicos necesarios para el diseño, orden y formulación de las preguntas en el cuestionario? ¿Se tienen en cuenta y utilizan las posibilidades técnicas que garanticen la calidad del cuestionario online para cada tipo de investigación?
- ¿Se evitarán las distorsiones de la muestra ofreciéndole facilidades técnicas al encuestado? ¿El instituto responsable garantizará que dispondrá de la capacidad técnica suficiente para asegurar la publicación y el acceso online a la encuesta durante el período de duración de la misma?
- ¿Es el tiempo de publicación lo suficientemente amplio como para que todos los individuos de la muestra tengan la posibilidad de acceder a la encuesta?

- ¿Se proporcionarán incentivos neutros (no orientados a un colectivo concreto) en función del objeto de la investigación y del grupo de análisis, que fomenten la participación pero que al mismo tiempo no distorsionen la muestra?
- ¿Se tomarán medidas de control de campo suficientes para evitar la participación múltiple y las respuestas aleatorias?
- ¿Se documentarán adecuadamente las anteriores medidas? (2)

Disposiciones para la protección de datos

- ¿Se ofrecerá al encuestado la información metodológica pertinente que le permita conocer el objetivo de la investigación?
- ¿El participante será informado de la participación voluntaria y de la posibilidad de interrumpir y/o cancelar la encuesta en cualquier momento?
- ¿Se informará adecuadamente al encuestado en el caso de que se obtengan datos sin su conocimiento y se pedirá correctamente su permiso para dicho fin?
- ¿Se evitará con los medios técnicos adecuados el acceso o visualización de los datos personales de encuestados por terceras personas? (2)

Presentación, Interpretación y Documentación

- ¿Se presentarán los resultados de la investigación de manera completa (todos los ítems y preguntas incluidas en el cuestionario)?
- ¿La interpretación de los resultados de la investigación estará relacionada con el objeto de la misma y documentada con datos empíricos?
- ¿Contiene el informe de análisis la información básica necesaria para poder reconocer la calidad científica de la investigación así como los límites de la interpretación de los resultados? (2)

Paneles de acceso online

- ¿El panel de encuestados, se trata de un panel de "captación activa" en el cual la selección de participantes se realiza siguiendo diversas técnicas de muestreo y cada panelista recibirá una invitación individualizada?

- ¿Se publicará el número real de panelistas, es decir, de individuos que regularmente y de forma constante participan en encuestas online?
- ¿Se documentará el proceso de selección y reclutamiento de participantes así como las medidas llevadas a cabo para el filtrado de los mismos y el cuidado de la calidad del panel?
- ¿Se actualizarán de manera frecuente los datos personales de los panelistas así como sus características socio-demográficas?
- ¿Se asegurará que el participante del panel no sea encuestado con demasiada frecuencia? (2)

1.5. Otras soluciones

Existen en la actualidad otros sistemas de encuestas online desarrolladas utilizando software libre de amplia utilización en distintas partes del mundo, con similitudes y diferencias con las de este proyecto. Entre los sistemas estudiados se destacan el PHPSurveyor y el Mod_Survey, los cuales se describirán a continuación:

1.5.1. PHPSurveyor

PhpSurveyor es una colección de scripts que interactúan con una base de datos (MySQL), para desarrollar, publicar y coleccionar respuestas de encuestas. Una vez una encuesta ha sido creada, puede ser publicada en línea (mostrada como preguntas individuales, agrupadas, o todas en una sola página) o puedes usar un sistema de entrada de datos para la administración de versiones en papel de la encuesta/cuestionario.

Utilizando el PhpSurveyor se pueden establecer condiciones por las que se muestren cuestiones individuales, puede variar el aspecto de la encuesta a través de un sistema de plantillas, y puede realizar análisis estadísticos básicos de los resultados de la misma.

Este sistema incluye además la capacidad de generar “tokens” o puntos, de tal forma que si se desea invitar a una lista de personas a una encuesta, se puede proveer a cada uno de un token, de tal forma que podrán acceder a este cuestionario usando el suyo propio. Así, se puede controlar quién realiza dichas encuestas y el número de veces que lo hacen. Sin embargo, da la posibilidad de que la respuesta sea

anónima (aunque se base en tokens), es decir, posibilita que no se pueda comprobar que unos determinados resultados pertenecen a un encuestado en concreto.

En realidad, al ser un proyecto de software libre, se puede contactar con el desarrollador para colaborar con él y mejorar el producto, si se encuentran errores o se desean implementaciones nuevas. Los requisitos son PHP (4.1.0 o una versión más reciente) y MySQL. Está probado en Servidores Web Apache, pero podría funcionar en otro (Microsoft IIS), aunque se deberían realizar algunos cambios. En cuanto a la seguridad, se basa en el sistema de Apache de seguridad de directorios. Este sistema normalmente es adecuado, aunque no es seguro si se usa Microsoft IIS, al cual habría que implementarle su propio sistema de seguridad. Existen otros detalles de seguridad que deben ser tenidos en cuenta como la colocación del archivo de configuración en un lugar inaccesible vía Web. Permite el uso de plantillas para modificar la interfaz.

El PHPSurveyor tiene entre sus características la presencia de:

- Listas: permiten introducir una serie de predeterminadas respuestas desde las que los participantes pueden elegir solo una.
- Listas con comentarios: presenta una serie de respuestas predeterminadas, desde las que el participante puede elegir solo una, pero añade una caja de comentarios para que el participante explique su elección.
- Múltiples opciones: permite entrar una serie de respuestas predeterminadas desde las que los participantes pueden elegir varias.
- Múltiples opciones con comentarios: ídem anterior más comentarios.
- Numérico: crea una cuestión que solo acepta números.
- Texto corto (hasta 100 caracteres) y largo (ilimitado). El campo Otro puede aparecer en determinadas cuestiones.
- Sí/No.
- Escalas: permite entrar series de respuestas predeterminadas que los participantes pueden valorar en orden de preferencia.
- Matriz (de 5 puntos de elección): se valora del 1 al 5.
- Matriz (de 10 puntos de elección): se valora del 1 al 10.

- Matriz (Sí, No, Incierto).

Un cuestionario creado en PHSurveyor consiste en lo siguiente:

- Un título, descripción, mensajes de bienvenida y de ayuda en las cuestiones (opcional): permite explicar cómo responder dicha cuestión. Cuando se incluye, un icono aparece en el cuestionario y haciendo clic en él se muestra la ayuda. Grupos dentro de la encuesta (debe haber al menos uno).
- Cuestiones dentro de cada grupo. Las cuestiones se crean en grupos (aunque sea de una sola cuestión) y pueden ser o no obligatorias.
- Posibles respuestas predefinidas.

Algunas características importantes de gestión son:

- Introducción de datos: esta función no se encuentra disponible para los participantes, sino para los administradores.
- Comunicación automática de la actividad del usuario al administrador: Establecimiento de condiciones de enrutado o aparición de unas cuestiones frente a otras. Estas condiciones permiten establecer itinerarios diferentes de cuestionarios en función de las respuestas de los encuestados. Estas condiciones pueden ser múltiples, es decir que dentro de una misma respuesta, varias condiciones se tengan que cumplir, o que se tengan que cumplir varias condiciones entre diferentes respuestas.
- Establecimiento de fecha de expiración. Inclusión de un sistema propio de estadísticas descriptivas sencillas, y permite el filtrado de los datos y sumarios de diferentes campos.
- Vista previa de un cuestionario antes de presentarlo como definitivo
- Importar/exportar encuestas/cuestionarios en formato sql.

En cuanto al visionado de resultados, esta herramienta permite visualizar:

- El total de respuestas o las últimas 100
- Versión imprimible
- Estadísticas simples y sumarios de respuestas
- Exportación de datos a Word, Excel o CVS

- Respaldo base de datos: en archivo sql. (7)

1.5.2. Mod_Survey

El Mod_Survey es otro sistema de encuestas, en este caso utilizando XML. El principal objetivo de Mod_Survey es la construcción de una suite de encuestas que soporte todos los componentes de cuestionarios avanzados y todos los diseños que el usuario pueda concebir. De esta forma, si una característica no está soportada nativamente, entonces al menos la base del código debe ser lógica y clara para que se pueda implementar fácilmente. Asimismo, su diseño se ha basado en la facilidad de aprendizaje. Sin embargo, la riqueza en características suele entrar en conflicto con esta facilidad. Normalmente, unos resultados complejos requieren una entrada de datos compleja. La forma de entrada de Mod_Survey se produce a través de la edición de archivos basados en XML. Esto es más complejo de aprender que una GUI (Graphic User Interface o Interfaz Gráfica de Usuario), pero mucho más potente.

El usuario objetivo de esta herramienta es el investigador necesitado de realizar cuestionarios largos, con poderosas interacciones con paquetes estadísticos como SPSS y que deberán adaptar el formato y la salida de datos en dichos cuestionarios. Sin embargo, Mod_Survey permite realizar cuestionarios simples tales como una votación o una encuesta de una sola cuestión, aunque para esto es mejor usar herramientas más simples.

El Mod_Survey técnicamente funciona como un módulo añadido de Apache Web Server.

Características básicas:

- Tipos de cuestionario: la mayoría de tipos básicos, incluyendo aquellos de elección múltiple, respuestas abiertas multilínea y matrices.
- Exportación de datos: una vez los datos son recogidos, el usuario puede elegir exportarlos en una multitud de formatos de datos, importables desde la mayoría de suites estadísticas importantes.
- Los formatos de exportación incluyen campos, columnas fijas, scripts de datos y tablas. La importación se ha probado con MS Excel, MS Access, MiniTab, SPSS y R.
- Chequeo/restricciones de respuestas: las preguntas pueden ser chequeadas para asegurar que la respuesta se produce en el formato correcto.

- Estadísticos descriptivos: las características de exportación incluyen el visionado de estadísticos descriptivos y tablas de frecuencias calculadas desde los datos enviados.
- Aleatoriedad: Los bloques de preguntas de una matriz se pueden mostrar aleatoriamente tanto en filas como en columnas (o ambos a la misma vez).

Características de Recogida avanzada de datos:

- Tiempo y fecha: incluye tanto el registro de una respuesta como la medida del tiempo tomado por un encuestado para responder el cuestionario.
- Interconexión con el servidor: las variables de entorno usadas por Apache se pueden añadir a los datos. Esto permite por ejemplo guardar el número IP o el navegador del encuestado.
- Cuestiones completamente modificables: si los tipos de cuestión no son suficientes, el usuario puede diseñar cuestiones completamente adaptadas usando todas las posibilidades de HTML y JavaScript.

Interconexión de bases de datos:

- Uso de diferentes bases de datos: las respuestas se pueden guardar opcionalmente en cualquier sistema de bases de datos DBI-compatibles, como PostgreSQL, Oracle o MySQL.
- Importación de campos de bases de datos: los datos se pueden obtener de cualquier gestor de bases de datos DBI-compatible.

Modularidad:

- Lógica condicional y enrutado: los respondientes pueden ser enrutados hacia diferentes preguntas basadas en respuestas previas. El enrutado permite agregar expresiones booleanas.
- Enlazado y fragmentos: Los fragmentos, tales como el encabezado de una página, puede disgregarse del diseño de la encuesta y ser colocado en archivos separados para su inclusión en las encuestas.
- Permiso y extensión de exportaciones: los módulos de exportación de datos pueden ser individualmente habilitados o deshabilitados.

Personalización de la interfaz:

- Temas: hay diversos interfaces y temas de color dentro de la instalación base.
- Hojas de estilo: la mayoría de los aspectos de la interfaz puede ser modificada a través de hojas de estilo especiales.
- Gráficos y código HTML: etiquetas HTML pueden ser añadidas para resaltar palabras o insertar imágenes.

Otras características avanzadas:

- Contenidos dinámicos: los resultados de los respondientes pueden ser modificados automáticamente dependiendo de las anteriores respuestas (por ejemplo el tratamiento de Sr. o Sra.).
- Enviar copias de datos: se puede enviar correos electrónicos automáticos con copias de todas las respuestas de los encuestados a diversas cuentas.
- Restricción detallada de acceso: el acceso a varios aspectos de una encuesta (respuesta, exportación de datos, administración...) pueden restringirse en función del usuario.

Limitaciones:

- No tiene una GUI para editar cuestionarios. Se deben escribir los cuestionarios en el sistema de etiquetas de Mod_Survey usando un editor de texto. Esto implica una dificultad grande a la hora de elaborar los cuestionarios. Hay un proyecto, denominado Cough, que será un añadido a Mod_Survey.
- La persistencia de datos es la opción de los encuestados de guardar temporalmente las respuestas para volver en otro momento a finalizar el cuestionario.
- Modificación secundaria de datos: no se pueden modificar los datos enviados, aunque está planeado incluirlo en el futuro.
- Aleatoriedad del orden entre cuestiones: solo se permite dentro de bloques de matrices.
- Gestión de la población: no hay ninguna funcionalidad para seleccionar ejemplos de población o realizar un seguimiento. Se planea incluirlo en Cough.
- Internacionalización/localización: se encuentra traducido al alemán, sueco, italiano y francés (incompleto). Sin embargo, no hay traducción al español. (7)

1.5.3. Conclusiones sobre las otras soluciones

A la hora de enfrentarse al problema a resolver se analizaron distintos sistemas de gestión de encuestas ya existentes, entre ellos el Mod_Survey y el PHPsurveyor, para analizar si eran o no aplicables en la UCI, y determinar cuáles de sus características podrían ser útiles a la hora de crear un sistema propio.

Después de analizados se determinó que estos sistemas no se ajustaban a los requerimientos de la UCI, fundamentalmente por su incapacidad para la utilización de las ventajas que brinda un dominio y el que no definan explícitamente los roles que permiten la especialización en la gestión de información y su mejor uso.

A pesar de estas limitaciones, se determinó tomar algunas de sus características en este proyecto para la creación del sistema de gestión UCICuest incluyendo algunos de los tipos de preguntas utilizados por ellos.

1.6. Conclusiones

En este capítulo se trataron los problemas puntuales que existen en la gestión y control del procesamiento de encuestas en la Universidad de las Ciencias Informáticas, se pone de manifiesto la necesidad de crear una aplicación Web que garantice la definitiva automatización de los procesos implicados. Además de llevar a una aplicación de este sistema, basado en su efectividad y maleabilidad al estar radicado en una plataforma de software libre.

Capítulo 2

Tendencias y tecnologías actuales

2.1. Introducción

En este segundo capítulo se realiza un análisis minucioso de los principales conceptos y tecnologías que se utilizarán para cumplimentar la solución propuesta. Se abordan temas relacionados con el uso de tecnologías sobre aplicaciones Web. Se hace un estudio de los lenguajes de programación más apropiados para el desarrollo del subsistema, el framework y el entorno de desarrollo integrado (IDE) que se utilizarán. Se referencian también las herramientas necesarias para la obtención del producto final deseado. Además se considera la metodología a utilizar para la implementación del sistema teniendo en cuenta las facilidades que puede aportar al trabajo.

2.2. Aplicaciones Web

Son sistemas informáticos donde una gran cantidad de datos volátiles, altamente estructurados, van a ser consultados, procesados y analizados mediante navegadores. Una de las principales características va a ser su alto grado de interacción con el usuario, y el diseño de su interfaz debe ser claro, simple y debe estar estructurado de tal manera que sea orientativo para cada tipo de usuarios.

Las Aplicaciones Web tienen una serie de rasgos comunes que diferencia a unos tipos de aplicaciones software de otros, y que son:

Desde el punto de vista del usuario, actualmente se ha hecho muy popular el uso de este tipo de aplicaciones tanto usuarios expertos como usuarios con habilidades limitadas acceden a ellas. El número y tipo de usuario que interactúan con las Aplicaciones Web no siempre es predecible, lo que obliga a tener el concepto de accesibilidad y facilidad de uso aún más presente que en otros tipos de aplicaciones.

Desde el punto de vista de la plataforma, se realiza un uso intensivo de la red y la conexión se establece desde distintos tipos de dispositivo de acceso.

Desde el punto de vista de la información, actualmente hay disponibilidad global de fuentes heterogéneas de información, estructurada y no estructurada, perteneciente a distintos dominios y que colaboran en el cumplimiento de los objetivos de la aplicación. (8)

Ventajas de una aplicación Web

- Hoy día las aplicaciones Web presentan un gran número de ventajas, algunas de ellas se mencionan a continuación.
- No es necesario instalar una aplicación en el ordenador.
- Se puede trabajar desde cualquier lugar del mundo con conexión a Internet.
- Ahorro de costes en hardware puesto que cualquier ordenador puede utilizar un navegador.
- Las actualizaciones se incorporan en el servidor, por lo que todos los usuarios tienen las nuevas versiones a la vez e inmediatamente.
- Aumenta la seguridad frente a virus y desastres puesto que los datos están alojados en un servidor con las más altas cotas de seguridad.
- Seguridad frente a accesos indeseados mediante la protección con contraseñas y servidores seguros.
- Posibilidad de compartir información con delegaciones y otras empresas.
- Posibilidad de tener un contacto más cercano con el usuario a través de servicios de mensajería y chats.
- Tener una presencia en Internet sólida gracias a los datos de la Aplicación Web. (8)

2.3. Lenguajes de programación del lado cliente

2.4.1 HTML

Es un lenguaje estático para el desarrollo de sitios Web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos que genera pueden tener las extensiones (htm, html).

El lenguaje llamado HTML indica al navegador donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página. (9)

Ventajas:

- Permite describir hipertextos.
- Genera archivos pequeños que facilitan la navegación en Internet.
- Presenta un despliegue rápido.
- Es un lenguaje de fácil aprendizaje.
- Es compatible con todos los navegadores.

Desventajas:

- Es un lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- Las etiquetas son muy limitadas.

2.4.2 Javascript

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario. Las sentencias escritas en javascript se encapsulan entre las etiquetas `<script>` y `</script>`. (9)

Ventajas:

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código Javascript se ejecuta en el cliente.

Desventajas:

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

2.4. Lenguajes de Programación del lado servidor

Los lenguajes del lado servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

2.5.1 ASP

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitio Web dinámicos. ASP significa en inglés (Active Server Pages), fue liberado por Microsoft en 1996. Las páginas Web desarrolladas bajo este lenguaje es necesario tener instalado Internet Information Server (IIS).

ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl and Jscript (no JavaScript). El código ASP puede ser insertado junto con el código HTML. Los archivos cuentan con la extensión (asp).

Ventajas:

- Usa Visual Basic Script, siendo fácil para los usuarios.
- Comunicación óptima con SQL Server.
- Soporta el lenguaje JScript (Javascript de Microsoft).

Desventajas:

- Código desorganizado.
- Se necesita escribir mucho código para realizar funciones sencillas.
- Tecnología propietaria.
- Hospedaje de sitios Web costosos. (10)

2.5.2 JAVA

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como punteros. JavaScript, un lenguaje interpretado, comparte un nombre similar y una sintaxis similar, pero no está directamente relacionado con Java.

Sun Microsystems proporciona una implementación GNU General Public License de un compilador Java y una máquina virtual Java, conforme a las especificaciones del Java Community Process, aunque la biblioteca de clases que se requiere para ejecutar los programas Java no es software libre.

Características de Java

Orientado a Objetos:

Java soporta las características esenciales del paradigma de la programación a objetos: encapsulación, herencia y polimorfismo. Java hace uso de la definición de entidades formadas por métodos y variables que reciben el nombre de clases, la instancia de alguna clase cualquiera en Java recibe el nombre de objeto. (11)

Robusto:

Java elimina el uso de apuntadores para referenciar localidades de memoria, Java además libera al desarrollador de la necesidad de desalojar la memoria que la aplicación ya no usa, aunado a esto Java requiere la declaración explícita tanto de los tipos de datos como de métodos. Estos factores que se mencionan destacan como causas comunes de errores lo que resulta en aplicaciones poco fiables. Al

implementar una aplicación en Java se reduce el porcentaje de errores ocasionados por las causas antes mencionadas lo que da como consecuencia programas más robustos y confiables.

Es importante mencionar que Java verifica que no haya problemas tanto en tiempo de ejecución como en tiempo de compilación. Finalmente Java hace uso de excepciones para notificar al usuario acerca de las fallas que puedan encontrarse en el sistema y recuperar dicho sistema. (11)

Independiente de plataforma:

Mientras que en lenguajes de programación como C++ existe la necesidad de recompilar el código fuente cada vez que se cambia de plataforma, Java ofrece la posibilidad de que los archivos que son generados para una aplicación sean independientes de plataforma, es decir, que se compilen una vez y se ejecuten en cualquier plataforma. Esto es posible gracias a que las aplicaciones hechas en Java generan archivos conocidos como bytecode.

Estos archivos no corresponden a algún procesador o sistema operativo en particular, dígase Intel o Motorola, Unix o Windows, sino que al momento de ser ejecutados un intérprete propio de cada plataforma interpreta el bytecode al correspondiente sistema y procesador en el cual se está ejecutando. Cada plataforma (Macintosh, Windows, Linux, etc.) tiene su propio intérprete de Java, pero el archivo bytecode es el mismo para todas las plataformas. (11)

Multitarea:

A pesar de que las capacidades de multitarea que pueden ser implementadas en Java dependen en gran parte del sistema operativo en el cual se ejecuten, dígase Windows o Unix, dichas capacidades superan en gran manera a los entornos de flujo único (single-thread) que ofrecen otros lenguajes de programación. Al ser multitarea Java permite la ejecución concurrente de varios procesos ligeros o hilos de ejecución. (11)

2.5.3 PHP

Es un lenguaje de programación utilizado para la creación de sitio Web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web

dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

Ventajas:

- Muy fácil de aprender.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Seguridad:

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor Web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor Web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación. (10)

2.5.4 Fundamentación sobre la elección del lenguaje a utilizar

En el momento de elegir el lenguaje de programación del lado del servidor que se adecuase a este proyecto se decidió utilizar el PHP, por una serie de características específicas del mismo que lo hacían más factible y aconsejable. Este lenguaje garantiza la obtención de un producto que puede ser montado, ya sea en un entorno Unix o Linux, utilizando software totalmente libre, con sus posibilidades de uso sin tener que efectuar pagos por sus licencias.

Tampoco requiere definición de tipos de variables ni manejo detallado del bajo nivel. A su vez le permite agilizar la elaboración del producto al contar con librerías donde están implementadas las funciones necesarias para dar solución a gran cantidad de funcionalidades del sistema y es compatible con PostgreSQL sistema de gestión de bases de datos que se utilizará, los cuales constituyen requisitos imprescindibles para obtener una aplicación que satisfaga las necesidad del cliente.

2.5. Framework

Los frameworks son abstracciones reutilizables de código definidos en una API³. El flujo del programa no es dictado por el usuario que ejecuta determinada acción, sino por el propio framework, lo cual es su principal característica.

Un framework es una mini-arquitectura que proporciona la estructura genérica y el comportamiento para una familia de abstracciones a lo largo de un contexto de metáforas que especifican su colaboración y es usado en un dominio dado.

El framework codifica el contexto en una clase de máquina virtual mientras hace las abstracciones abiertas. Un framework no es generalmente una aplicación específica. Una aplicación, en cambio puede ser construida con uno o más frameworks insertando dicha funcionalidad. Una posible definición de framework es: conjunto de clases cooperantes que hace reusable un diseño para una clase específica de software. Un framework proporciona una guía arquitectónica para dividir el diseño en clases y definir sus responsabilidades y colaboraciones. Un framework dicta la arquitectura de la aplicación.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente,

³ API (**A**pplication **P**rogramming **I**nterface, en español, Interfaz de Programación de Aplicaciones): es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (12) El bajo costo de aprendizaje de los fundamentos de PHP, le dio al lenguaje una reputación como el "idioma de aficionados". Poco a poco la situación cambió con la llegada de los frameworks MVC y otros frameworks RAD⁴ y las bibliotecas o librerías diseñados para facilitar y acelerar el desarrollo. ROR⁵ (Ruby on Rails) trajo una nueva forma de ver el desarrollo Web: más rápido, más sencillo, más eficaz. Unos pocos años después de la aparición de ROR, salieron a la luz un gran número de frameworks de desarrollo rápido.

2.5.1. CakePHP

CakePHP es un framework (entorno de trabajo) libre y de código abierto para el desarrollo en PHP. Es una estructura de librerías, clases y una infraestructura en tiempo de ejecución para programadores de aplicaciones Web originalmente inspirado en el framework Ruby On Rails. Su principal meta es permitir su trabajo de manera estructurada y rápida, sin pérdida de flexibilidad.

Cake tiene varias características que lo hacen una gran opción como framework para el desarrollo de aplicaciones rápidas y con el menor costo de molestia. Entre las que se encuentran:

- Licencia flexible.
- Compatibilidad con PHP4 y PHP5.
- CRUD integrado para la interacción con la base de datos y las preguntas simplificadas.
- Arquitectura modelo vista controlador (MVC).
- Validación incorporada.
- Plantillas rápidas y flexibles (Sintaxis PHP, con Helpers).
- Helpers en vista para Ajax, Javascript, Formularios HTML.
- Seguridad, Sesiones y componentes para Manejo de Peticiones.
- Lista de Control y Acceso flexible.
- Desinfección de datos.

⁴ **RAD** (Rapid Application Development, en español, Desarrollo Rápido de Aplicaciones)

⁵ **ROR** (Ruby on Rails) es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC).

- Caché flexible en Vistas.
- Trabajo desde cualquier subdirectorio Web del sitio, con poca o ninguna configuración de Apache envuelta. (13)

2.5.2. Zend Framework

Zend Framework se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios Web modernos, robustos y seguros. Además es Open Source y trabaja con PHP 5 a diferencia de CakePHP que trabaja con PHP 4 y PHP 5.

Principales características:

- Trabaja con MVC (Modelo Vista Controlador).
- Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services (Amazon, Flickr, Yahoo), etc.
- El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
- Una solución para el acceso a base de datos que balancea el ORM con eficiencia y simplicidad.
- Completa documentación y tests de alta calidad.
- Un buscador compatible con Lucene.
- Robustas clases para autenticación y filtrado de entrada.
- Clientes para servicios Web, incluidos Google Data APIs y Strikelron.
- Muchas otras clases útiles para hacerlo tan productivo como sea posible. (14)

2.5.3. Symfony

Symfony está desarrollado completamente con PHP5. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Características:**Symfony se diseñó para que se ajustara a los siguientes requisitos:**

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la Web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- Symfony automatiza la mayoría de elementos comunes de los proyectos Web, como por ejemplo:
- La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- La capa de presentación utiliza plantillas y layouts que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del framework. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones.
- Los formularios incluyen validación automatizada y relleno automático de datos (“repopulation”), lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.

- La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.
- El sistema de enrutamiento y las URL limpias permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.
- El soporte de correo incluido y la gestión de APIs permiten a las aplicaciones Web interactuar más allá de los navegadores.
- Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.
- Los plugins, las factorías (patrón de diseño “Factory”) y los mixin permiten realizar extensiones a medida de Symfony.
- Las interacciones con AJAX son muy fáciles de implementar mediante los helpers que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código.
- Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones. Por defecto incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software:
- Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.
- El framework de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas (test-driven development).
- La barra de depuración Web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.
- La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.
- Es posible realizar cambios “en caliente” de la configuración (sin necesidad de reiniciar el servidor).
- El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación. (12)

2.5.4. Fundamentos para la selección del framework

Entre los distintos tipos de framework que pudiesen ser utilizados en el desarrollo de este proyecto se escogió el symfony debido a la posibilidad de personalizarlo completamente acorde a los requerimientos del cliente. Esto se logra ya que el symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Es ejecutable tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. Además es sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos. Fue desarrollado para una interacción apropiada con PHP5.

2.6. Arquitectura Cliente Servidor

La arquitectura Cliente/Servidor empezó a ser aceptado a finales de los 80's. Su funcionamiento es sencillo: se tiene una máquina cliente, que requiere un servicio de una máquina servidor, y éste realiza la función para la que está programado (nótese que no tienen que tratarse de máquinas diferentes; es decir, una computadora por sí sola puede ser ambos, cliente y servidor dependiendo del software de configuración).

Desde el punto de vista funcional, se puede definir el modelo Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

En la arquitectura Cliente/Servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son más adecuadas a sus características. Si esto se aplica tanto a clientes como servidores se entiende que la forma más estándar de aplicación y uso de sistemas Cliente/Servidor es mediante la explotación de las PCs a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo mainframe.

Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente sólo se ocupan de la interacción con el usuario (aunque esto puede variar). En otras palabras la arquitectura Cliente/Servidor es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento.

Esta arquitectura permite distribuir físicamente los procesos y los datos en forma más eficiente lo que en computación distribuida afecta directamente el tráfico de la red, reduciéndolo grandemente. (15)

2.6.1. Patrón Arquitectónico Modelo-Vista-Controlador (MVC)

Teniendo en cuenta que Symfony está basado en un patrón clásico del diseño Web conocido como arquitectura MVC, que está formado por 3 niveles:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página Web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. La Figura 1 ilustra el funcionamiento del patrón MVC.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (12)

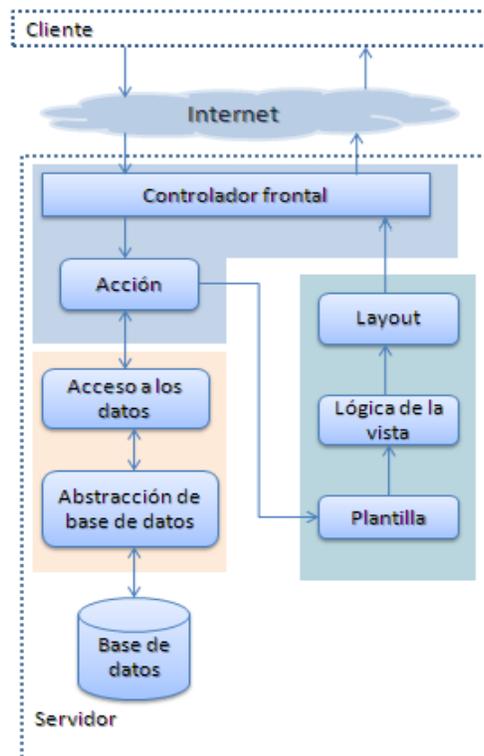


Figura 1 Diagrama de representación del Patrón MVC

El principio más importante de la arquitectura MVC es la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. (12)

2.7. Entorno de desarrollo integrado (IDE)

2.7.1. Eclipse

Eclipse es una plataforma universal para integrar herramientas de desarrollo, con una arquitectura abierta y basada en plug-ins. Además, Eclipse da soporte a todo tipo de proyectos que abarcan desde el ciclo de vida del desarrollo de aplicaciones, incluyendo soporte para modelado.

La arquitectura de plug-ins permite integrar diversos lenguajes sobre un mismo IDE e introducir otras aplicaciones accesorias. Conservan el registro de las versiones, generan y mantienen la documentación de cada etapa del proyecto.

Características:

- Editor de Texto.
- Resaltado de sintaxis.
- Compilación en tiempo real.
- Pruebas unitarias con Junit.
- Control de versiones con CVS.
- Integración con Ant.
- Asistentes (wizards).
- Plug-ins. (16)

2.8.2. Zend Studio

Zend Studio es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código. Es un IDE propietario, compatible con las plataformas Linux, MAC y Windows. Incluye todos los componentes necesarios durante el ciclo de vida de una aplicación en PHP: editor, análisis, depuración, optimizadores de código y herramientas de base de datos. Permite y agiliza el desarrollo Web y simplifica proyectos complejos. Cuenta con un buen completamiento de código, administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, PHP Documentor, manual de PHP.

Desventajas:

- Requiere Licencia de pago.
- No incluye editor visual HTML.
- Es un poco complejo. (17)

2.8.3. Fundamentación de la selección de IDE

El Entorno de Desarrollo Integrado (IDE) escogido para la implementación de este proyecto fue Eclipse debido a que el mismo incorpora un sin fin de utilidades para simplificar la labor de los programadores que están centrados en la construcción de una plataforma de desarrollo extensible. Aparte de ser un entorno de desarrollo muy completo, una de las particularidades más interesantes para la comunidad es que es de código libre y gratuito. Eclipse soporta la mayoría de los lenguajes de programación más populares, incluido el PHP, ya que para el mismo se han desarrollado varias extensiones (plugins).

2.9. Servidor de aplicaciones Web

Un servidor Web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP). (18)

Servidor Web: Es el programa que, utilizando el protocolo de comunicaciones HTTP, es capaz de recibir peticiones de información de un programa cliente (navegador), recuperar la información solicitada y enviarla al programa cliente para su visualización por el usuario. (19)

Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección.

El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Apache

Apache es un servidor HTTP, de código abierto y licenciamiento libre, que funciona en Linux, sistemas operativos derivados de Unix™, Windows, Novell NetWare y otras plataformas. Ha desempeñado un papel muy importante en el crecimiento de la red mundial, y continua siendo el servidor HTTP más utilizado, siendo además el servidor de facto contra el cual se realizan las pruebas comparativas y de

desempeño para otros productos competidores. Apache es desarrollado y mantenido por una comunidad de desarrolladores auspiciada por Apache Software Foundation.

Apache es un servidor Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP). Entre sus características se destacan:

- Multiplataforma
- Es un servidor de Web conforme al protocolo HTTP/1.1
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hebras en la versión 2.0
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor. (20)

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas del Apache.
- **Módulos Multiproceso:** Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software. (21)

2.10. Metodologías de desarrollo de software

2.10.1. Extreme Programming (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (22)

Características esenciales de XP organizadas en los tres apartados siguientes:

Las Historias de Usuario:

Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

A efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración). Las historias de usuario son descompuestas en tareas de programación (task card) y asignadas a los programadores para ser implementadas durante una iteración. (20)

Roles XP:

Programador: El programador escribe las pruebas unitarias y produce el código del sistema.

Cliente: Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

Encargado de pruebas (Tester): Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (Tracker): Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

Entrenador (Coach): Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

Gestor (Big boss): Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación. (20)

Proceso XP:

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

- El cliente define el valor de negocio a implementar.
- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- El programador construye ese valor de negocio.
- Vuelve al primer paso.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. (22)

Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas. (22)

El juego de la planificación: Hay una comunicación frecuente el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

Entregas pequeñas: Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.

Metáfora: El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).

Diseño simple: Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.

Pruebas: La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.

Refactorización (Refactoring): Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.

Programación en parejas: Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).

Propiedad colectiva del código: Cualquier programador puede cambiar cualquier parte del código en cualquier momento.

Integración continua: Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

40 horas por semana: Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.

Cliente in-situ: El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.

Estándares de programación: XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras, donde una línea entre dos de ellas significa que ambas se refuerzan entre sí. La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor ya en su ejecución. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo. (22)

2.10.2. Proceso Unificado de Desarrollo (RUP)

El ciclo de vida de RUP

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada una de ellas. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las mismas hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada una se realiza de forma planificada es por eso que se dice que son mini-proyectos.

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades. (21)

Inicio: Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto.

Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.

Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

Transición: se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

Fase de Inicio

Durante la fase de inicio las iteraciones hacen poner mayor énfasis en actividades modelado del negocio y de requisitos.

Modelado del negocio:

En esta fase el equipo se familiarizará más al funcionamiento de la empresa, sobre conocer sus procesos. (23)

- Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado.
- Entender el problema actual en la organización objetivo e identificar potenciales mejoras.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.

Requisitos:

- En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifiquen. (21)
- Establecer y mantener un acuerdo entre clientes y otros stakeholders sobre lo que el sistema podría hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

Fase de Elaboración

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura. (23)

Análisis y Diseño:

- En esta actividad se especifican los requerimientos y se describen sobre cómo se van a implementar en el sistema.
- Transformar los requisitos al diseño del sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación.

Fase de Construcción

Implementación:

- Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. El resultado final es un sistema ejecutable.
- Planificar qué subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en qué orden implementa los elementos del subsistema.
- Si encuentra errores de diseño, los notifica.
- Se integra el sistema siguiendo el plan.

Pruebas

Este flujo de trabajo es el encargado de evaluar la calidad del producto que se está desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida. (23)

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.
- Verificar que los requisitos tengan su apropiada implementación.

Despliegue

Esta actividad tiene como objetivo producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.
- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Formar a los usuarios y al cuerpo de ventas.
- Migrar el software existente o convertir bases de datos.

Fase de Transición

Gestión del proyecto:

Se vigila el cumplimiento de los objetivos, gestión de riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

- Proveer un marco de trabajo para la gestión de proyectos de software intensivos.
- Proveer guías prácticas realizar planeación, contratar personal, ejecutar y monitorear el proyecto.
- Proveer un marco de trabajo para gestionar riesgos.
- Configuración y control de cambios
- El control de cambios permite mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido. (23)

Entorno

La finalidad de esta actividad es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir. En concreto las responsabilidades de este flujo de trabajo incluyen:

- Selección y adquisición de herramientas.

- Establecer y configurar las herramientas para que se ajusten a la organización.
- Configuración del proceso.
- Mejora del proceso.
- Servicios técnicos. (23)

2.10.3. Feature Driven Development (FDD)

FDD es un proceso diseñado por Peter Coad, Erich Lefebvre y Jeff De Luca y se podría considerar a medio camino entre RUP y XP, aunque es considerado un proceso ligero.

FDD está pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas (~2 semanas) que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.

Las iteraciones se deciden en base a features (de ahí el nombre del proceso) o funcionalidades, que son pequeñas partes del software con significado para el cliente. Así, construir el sistema de ventas es algo que requiere mucho tiempo, y construir el sistema de persistencia no tiene significado para el cliente, pero si lo tiene enviar pedido por email. (24)

Un proyecto que sigue FDD se divide en 5 fases:

- Desarrollo de un modelo general.
- Construcción de la lista de funcionalidades.
- Plan de releases en base a las funcionalidades a implementar.
- Diseñar en base a las funcionalidades.
- Implementar en base a las funcionalidades.

Las primeras tres fases ocupan gran parte del tiempo en las primeras iteraciones, siendo las dos últimas las que absorben la mayor parte del tiempo según va avanzando el proyecto, limitándose las primeras a un proceso de refinamiento.

El trabajo (tanto de modelado como de desarrollo) se realiza en grupo, aunque siempre habrá un responsable último (arquitecto jefe o jefe de programadores en función de la fase en que se encuentre el

proyecto), con mayor experiencia, que tendrá la última palabra en caso de no llegar a un acuerdo. Al hacerlo en grupo se consigue que todos formen parte del proyecto y que los menos inexpertos aprendan de las discusiones de los más experimentados, y al tener un responsable último, se asignan las responsabilidades que todas las empresas exigen.

Las funcionalidades a implementar en una release se dividen entre los distintos subgrupos del equipo, y se procede a implementarlas. Las clases escritas tienen propietario (es decir, solo quién las crea puede cambiarlas), es por ello que en el equipo que implementa una funcionalidad dada deberán estar todos los dueños de las clases implicadas, pudiendo encontrarse un programador en varios grupos, implementando distintas funcionalidades. Habrá también un programador jefe (normalmente el más experimentado) que hará las funciones de líder del grupo que implementa esa funcionalidad.

En el proceso de implementar la funcionalidad también se contemplan como partes del mismo (en otros métodos se describen como actividades independientes) la preparación y ejecución de pruebas, así como revisiones del código (para distribuir el conocimiento y aumentar la calidad) e integración de las partes que componen el software.

FDD también define métricas para seguir el proceso de desarrollo de la aplicación, útiles para el cliente y la dirección de la empresa, y que pueden ayudar, además de para conocer el estado actual del desarrollo, a realizar mejores estimaciones en proyectos futuros. (24)

2.10.4. Fundamentación de elección de metodología de desarrollo

La metodología escogida para este proyecto es la XP debido a que es una metodología ágil, cuyo objetivo fundamental es la potenciación de las relaciones interpersonales para su desarrollo como tal. Disminuye el riesgo de no complacer al cliente, mediante la retroalimentación continua y entregas en cortos plazos de software funcional. Además se pueden producir rápidamente versiones del sistema que sean operativas.

También se puede diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. Permite el desarrollo del proyecto sin necesidad de redactar una especificación, análisis o diseño detallados del mismo desde un inicio. De igual forma se minimiza el trabajo innecesario, mediante la teoría de solo hacer lo que es eminentemente necesario.

2.11. Conclusiones

En el presente capítulo se realizó un análisis de las tecnologías y herramientas utilizadas a lo largo del desarrollo del sistema propuesto, profundizándose en el conocimiento de conceptos necesarios para la comprensión del trabajo. Se mencionaron las características que se tuvieron en cuenta a la hora de escoger el lenguaje de programación, el framework y la metodología a utilizar.

Capítulo 3

Descripción de la solución propuesta

3.1. Introducción

En este tercer capítulo se pasa a la descripción de los distintos módulos que se implementarán en el proyecto y que forman parte del sistema automatizado para la gestión y procesamiento dinámico de encuestas (UCICuest). Estos módulos son administración, gestión de solicitud y creación/eliminación de encuestas. La implementación de los mismos se adecua a la metodología XP que se utiliza en este proyecto.

3.2. Tarjetas de Historias de Usuario

La metodología XP para desarrollar las tareas de programación, propone el control de la documentación a través de historias de usuario. Se considera que si se hace en tarjetas individuales el manejo de éstas será más fácil, por tanto se propone utilizar el siguiente modelo.

Historia de Usuario

Número:	Usuario:
Nombre historia:	
Prioridad en negocio:	Riesgo en desarrollo:
(Alta / Media / Baja)	(Alto / Medio / Bajo)
Puntos estimados:	Iteración asignada:
Programador responsable:	
Descripción:	

Observaciones:

Tabla 3 Historia de Usuario

3.2.1. Campos de la tarjeta Historia de Usuario

Número: Índice de la historia de usuario. Es un número único que se le asigna a cada historia de usuario con el fin de lograr una mejor organización de éstas.

Nombre historia: Nombre de la historia de usuario. Debe ser descriptivo, en la medida de las posibilidades, de lo que se implementará y no muy extenso.

Descripción: Se describe el requerimiento que da origen a la historia de usuario. La descripción debe ser corta, precisa y dejar claro qué es lo que se desea hacer.

Observaciones: Aspectos que se deben tener en cuenta para implementar la historia de usuario. Precondiciones o pos-condiciones que se deben tener presentes al implementar la historia de usuario.

Puntos estimados: Tiempo estimado que demorará la implementación de la historia de usuario. Tiempo ideal en que se estima se implementará la historia de usuario.

Prioridad en negocio: Prioridad que se le asigna a la historia de usuario en el negocio. Si es importante y debe ser implementada lo antes posible, entonces la prioridad debe ser alta, si no media o baja.

Riesgo en desarrollo: Riesgo que representa para el desarrollo la historia de usuario. En dependencia del riesgo que represente se le asigna tres valores, alto, medio y bajo.

Alto: historias de usuario que no pueden ser estimadas con suficiente precisión o que implementarlas en iteraciones posteriores puede representar cambios sustanciales en el código base del proyecto.

Medio: historias de usuario que son estimadas razonablemente bien.

Bajo: historias de usuario que son estimadas sin problema alguno.

Iteración asignada: Iteración en la que se implementará la historia de usuario.

Usuario: Rol que algo o alguien juega cuando interactúa con el sistema. Puede ser cualquier individuo, grupo, organización o máquina que interactúa con el sistema. Programador responsable: Programador responsable de implementar la historia de usuario.

3.3. Tarjeta de tareas de programación

Las tareas de programación corresponden a una historia de usuario. Pero a una historia de usuario pudieran corresponderle varias tarjetas de tareas de programación, para dar respuestas a las necesidades del cliente. El tipo de tarjetas que se propone adoptan el siguiente modelo:

Tarea	
Número tarea:	Número historia de usuario:
Nombre tarea:	
Tipo de tarea :	Puntos estimados:
Fecha inicio:	Fecha fin:
Programador responsable:	
Descripción:	

Tabla 4 Tarea de programación

3.3.1. Campos de la tarjeta Tarea

Número historia de usuario: Número de la historia de usuario a la que corresponde. Índice de la historia de usuario a la que se corresponde esta tarea de programación.

Número tarea: Índice de la tarea de programación. Es un número único que se le asigna a cada tarea de programación que pertenece a una historia de usuario determinada con el fin de lograr una mejor organización de éstas.

Nombre tarea: Nombre de la tarea de programación. Debe ser descriptivo, en la medida de las posibilidades, de lo que se realizará y no muy extenso.

Tipo de tarea: Informa el tipo de tarea a realizar. Las tareas pueden ser.

- **Desarrollo:** Tarea que se realizará por primera vez.
- **Corrección:** Tarea que se realiza a partir de una anterior que no se realizó correctamente, es decir no pasó todos los casos de prueba que le corresponden correctamente.
- **Mejora:** Tarea que se realiza a partir de una anterior que se realizó correctamente pero se incorporan nuevos requerimientos para la misma, ahora tendrá que ser modificada para pasar satisfactoriamente los nuevos casos de prueba adicionales.
- **Otra:** Tarea que no corresponde con ninguna de las anteriormente descritas.

Fecha inicio: Fecha en que se inicia la implementación de la tarea de programación.

Fecha fin: Fecha en que concluye la implementación de la tarea de programación.

Programador responsable: Programador responsable de implementar la tarea de programación.

Descripción: Se describe qué es lo que se desea realizar. La descripción debe ser corta y precisa.

Puntos estimados: Tiempo estimado que demorará la implementación de la tarea de programación. Tiempo ideal en que se estima se implementará la tarea de programación.

3.4. Módulos a implementar en el sistema

Un módulo es una de las partes en las cuales se divide un programa para facilitar su ejecución. De las varias tareas que debe realizar un programa para cumplir su función u objetivo, el módulo realizará una de estas tareas o varias de ellas simultáneamente.

Atendiendo los objetivos propuestos en este proyecto se pasa a enumerar y explicar los distintos módulos que serán implementados:

3.5. Módulo Administración

En el módulo Administración, el administrador del sistema asigna distintos roles a determinados usuarios del sistema, en concordancia con las necesidades que se planteen. Estos roles le permiten a estos usuarios la navegación dentro del sistema, dependiendo del nivel de necesidad de acceso, seguridad y especialización de los mismos.

3.5.1. Roles del sistema

Los Roles son los distintos comportamientos que un individuo puede mostrar en relación con su particular posición social. Hace referencia al papel que ocupa o desempeña dentro de determinado contexto.

En el marco de este sistema, existen varios roles que pueden ser ocupados por las diferentes personas que interactúan con el sistema y que están relacionados con sus competencias profesionales y sus responsabilidades o fines dentro de la institución.

Los roles establecidos en el sistema son:

- **Cliente:** Es quien responde el cuestionario y también puede generar temas de interés de investigación por lo que puede asumir un rol de solicitante.
- **Especialista:** Encargado de realizar, crear, diseñar o montar la encuesta siempre generada con un objetivo claro y para un tema o área específica de interés a la comunidad universitaria que se somete al flujo de aprobación. Luego que se termina la encuesta se envía un aviso al editor responsable para que comience la labor de corrección. Cuando este lista la corrección la enviará al directivo para su aprobación y posterior publicación.
- **Corrector:** Corresponde en esta parte del trabajo, al editor o equipo de editores, acceder desde el sistema, a la encuesta para hacerle correcciones gramaticales, ortográficas y de redacción, y luego reinsertarlo. Después de revisar desde el sistema el cuestionario comunica al especialista que está listo.
- **Directivo:** El directivo de primer nivel que atiende la actividad o área a la que se refiere el cuestionario.
- **Administrador:** Es el encargado de la gestión de usuario, de áreas y tiene acceso a todas las funcionalidades del sistema.

Con el flujo de administración del sistema a medida que la encuesta pasa de un nivel a otro, ya se van salvando responsabilidades, es decir que en el momento que el corrector revisa la encuesta, ya el especialista no puede cambiarla ni corregirla, así mismo sucede con el corrector y el directivo.

3.5.2. Historia del Módulo

Historia de Usuario

Número: 01 **Usuario:** DTU

Nombre historia: Asignación de Roles

Prioridad en negocio: Alta **Riesgo en desarrollo:** Alta

(Alta / Media / Baja) **(Alto / Medio / Bajo)**

Puntos estimados: 45 horas **Iteración asignada:** 1

Programador responsable: Miguel Martínez Ruiz

Descripción: El sistema debe permitir que las personas se puedan autenticar en el, teniendo en cuenta las funciones que van a realizar dentro del mismo. Se debe definir el rol que jugará cada persona dentro del sistema.

Los roles deben ser:

Cliente: es quien realiza la encuesta y a su vez puede realizar solicitudes.

Especialista: es quien revisa las solicitudes (aprobándolas o denegándolas), en caso de que se deniegue la solicitud se le debe avisar al cliente el motivo y en caso de que sea aprobada se realiza la encuesta y se le muestra al corrector.

Corrector: es quien revisa la encuesta desde el punto de vista gramatical, ortográfico y de redacción, una vez revisada se le muestra al directivo.

Directivo: es quien aprueba la publicación de la encuesta.

Administrador: encargado del control y funcionamiento del sistema.

También el sistema debe permitir la creación y eliminación de áreas para agrupar las encuestas.

Observaciones: la autenticación debe ser por el usuario uci.

Tabla 5 Historia de Usuario 01 Módulo Administración

3.5.3 Tareas del Módulo

Tarea para el Módulo de Administración

Número tarea: 01

Número historia de usuario: 01

Nombre tarea: Comprobación de usuario UCI

Tipo de tarea : Desarrollo

Puntos estimados: 7 horas

Fecha inicio: 23/02/2009

Fecha fin: 27/02/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al sistema la conexión con la Base de Datos de la UCI, para verificar la existencia de los usuarios dentro del dominio .uci, y así poder pasar a la siguiente tarea de asignación de roles.

Tabla 6 Tarea 01 Módulo Administración

Tarea para el Módulo de Administración

Número tarea: 02 **Número historia de usuario:** 01

Nombre tarea: Creación de usuarios

Tipo de tarea : Desarrollo **Puntos estimados:** 7 horas

Fecha inicio: 02/03/2009 **Fecha fin:** 06/03/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al Administrador asignar los distintos roles dentro del sistema, en dependencia con las tareas a cumplir por los usuarios, determinando a su vez el nivel de libertad de acción dentro del sistema y las restricciones adecuadas.

Tabla 7 Tarea 02 Módulo Administración

Tarea para el Módulo de Administración

Número tarea: 03 **Número historia de usuario:** 01

Nombre tarea: Eliminar usuarios

Tipo de tarea : Desarrollo **Puntos estimados:** 8 horas

Fecha inicio: 09/03/2009 **Fecha fin:** 13/03/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al Administrador eliminar usuarios del sistema, privándolos de sus privilegios dentro del mismo.

Tabla 8 Tarea 03 Módulo Administración

Tarea para el Módulo de Administración

Número tarea: 04 **Número historia de usuario:** 01

Nombre tarea: Modificar usuarios

Tipo de tarea : Desarrollo **Puntos estimados:** 7 horas

Fecha inicio: 16/03/2009 **Fecha fin:** 20/03/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Considerando que surgiese la necesidad de variar las funciones de los usuarios, el Administrador puede modificar dichos privilegios cambiando los roles que tengan asignados estos usuarios.

Tabla 9 Tarea 04 Módulo Administración

Tarea para el Módulo de Administración

Número tarea: 05 **Número historia de usuario:** 01

Nombre tarea: Creación y eliminación de Áreas

Tipo de tarea : Desarrollo **Puntos estimados:** 7 horas

Fecha inicio: 23/03/2009 **Fecha fin:** 27/03/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al administrador crear y eliminar áreas dentro del sistema.

Tabla 10 Tarea 05 Módulo Administración

Tarea para el Módulo de Administración

Número tarea: 06	Número historia de usuario: 01
Nombre tarea: Seguridad del sistema	
Tipo de tarea : Desarrollo	Puntos estimados: 7 horas
Fecha inicio: 30/03/2009	Fecha fin: 03/04/2009
Programador responsable: Miguel Martínez Ruiz	
Descripción: Permite garantizar la seguridad del sistema.	

Tabla 11 Tarea 06 Módulo Administración

3.6. Módulo Gestión de Solicitud

En el módulo Gestión de Solicitud interactúan dos grupos de personas con distintos roles: los clientes y los especialistas. Los clientes del sistema pueden realizar la solicitud de la creación de una encuesta a partir de un tema determinado, donde se les permite agregar sugerencias en dependencia de sus necesidades. Los especialistas valoran entonces las características de dicha solicitud y determinan la factibilidad de la misma.

3.6.1. Historia del Módulo

Historia de Usuario	
Número: 02	Usuario: DTU
Nombre historia: Gestionar Solicitud	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
(Alta / Media / Baja)	(Alto / Medio / Bajo)
Puntos estimados: 40 horas	Iteración asignada: 1

Programador responsable: Miguel Martínez Ruiz

Descripción: Es donde los clientes pueden realizar la solicitud de la creación de una encuesta a partir de un tema suministrado por ellos, agregando sugerencias en dependencia de sus necesidades. Los especialistas valoran entonces las características de dicha solicitud y determinan la factibilidad de la misma en dependencia de factores tanto objetivos como subjetivos, y determinan si el tema suministrado permite o no la creación de una encuesta.

Observaciones: Se debe recoger el correo del cliente para una mayor comunicación con el mismo.

Tabla 12 Historia de Usuario 02 Módulo Gestión de Solicitud

3.6.2. Tareas del Módulo

Tarea para el Módulo de Gestión de Solicitud

Número tarea: 01

Número historia de usuario: 02

Nombre tarea: Crear Solicitud

Tipo de tarea : Desarrollo

Puntos estimados: 8 horas

Fecha inicio: 06/04/2009

Fecha fin: 10/04/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al usuario entrar al sistema con su usuario uci y solicitar el servicio de creación de una solicitud.

Tabla 13 Tarea 01 Módulo Gestión de Solicitud

Tarea para el Módulo de Gestión de Solicitud

Número tarea: 02 **Número historia de usuario:** 02

Nombre tarea: Revisar Solicitud

Tipo de tarea : Desarrollo **Puntos estimados:** 8 horas

Fecha inicio: 13/04/2009 **Fecha fin:** 17/04/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al especialista revisar las solicitudes y aceptarlas o denegarlas en dependencia del tema a tratar.

Tabla 14 Tarea 02 Módulo Gestión de Solicitud

Tarea para el Módulo de Gestión de Solicitud

Número tarea: 03 **Número historia de usuario:** 02

Nombre tarea: Eliminar Solicitudes Denegadas

Tipo de tarea : Desarrollo **Puntos estimados:** 7 horas

Fecha inicio: 20/04/2009 **Fecha fin:** 24/04/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al especialista eliminar las solicitudes denegadas.

Tabla 15 Tarea 03 Módulo Gestión de Solicitud

Tarea para el Módulo de Gestión de Solicitud

Número tarea: 04 **Número historia de usuario:** 02

Nombre tarea: Listar Solicitudes sin revisar

Tipo de tarea : Desarrollo **Puntos estimados:** 6 horas

Fecha inicio: 27/04/2009 **Fecha fin:** 30/04/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al especialista ver las solicitudes que no están revisadas.

Tabla 16 Tarea 04 Módulo Gestión de Solicitud

Tarea para el Módulo de Gestión de Solicitud

Número tarea: 05 **Número historia de usuario:** 02

Nombre tarea: Listar Solicitudes Aceptadas y Denegadas

Tipo de tarea : Desarrollo **Puntos estimados:** 6 horas

Fecha inicio: 04/05/2009 **Fecha fin:** 08/04/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al especialista ver las solicitudes que ya han sido revisadas.

Tabla 17 Tarea 05 Módulo Gestión de Solicitud

3.7. Módulo Creación/Eliminación de Encuestas

En el módulo Creación/Eliminación de Encuestas el especialista del sistema crea las encuestas a partir de la aceptación de la solicitud hecha por el cliente. Determinando qué tipo de preguntas debe ser utilizada en dependencia de los criterios dados por el cliente en su solicitud.

3.7.1. Tipos de Preguntas

Opciones múltiples: son aquellas preguntas donde se da un enunciado y un grupo de alternativas de respuestas para que el sujeto marque las que desee (sin límite de opciones). Ej:

¿Pregunta de opciones múltiples?

Respuesta 1

Respuesta 2

Respuesta 3

Respuesta 4

Respuesta 5

Opciones múltiples + comentario: Igual a la anterior pero con la opción de que al final se pueda añadir un comentario Ej: __otros ¿cuáles? _____.

¿Pregunta de opciones múltiples con comentario?

Respuesta 1

Respuesta 2

Respuesta 3

Respuesta 4

Respuesta 5

Otro(a)s ¿Cuáles?

Opciones múltiples con rango: Se les da al sujeto un listado de alternativas de respuesta, solo que aquí se le define la cantidad de opciones a seleccionar (Ej: 3, 5, etc.).

¿Pregunta de opciones múltiples?. Seleccione no más de 3 respuestas.

Respuesta 1

Respuesta 2

Respuesta 3

Respuesta 4

Respuesta 5

Opciones múltiples con rango + comentario: Igual a la anterior pero con la opción de que al final se pueda añadir un comentario. Ej:

¿Pregunta de opciones múltiples con comentario?
Seleccione no más de 3 respuestas.

Respuesta 1
 Respuesta 2
 Respuesta 3
 Respuesta 4
 Respuesta 5

Otro(a)s ¿Cuáles?

Opciones únicas: De un listado de alternativas de respuesta se le aclara al sujeto que solo debe marcar 1 (para ver prevalencia, importancia etc.). Ej:

¿Pregunta de opciones únicas?

Respuesta 1
 Respuesta 2
 Respuesta 3
 Respuesta 4

Solo comentario: son las preguntas abiertas. Es decir se enuncia la pregunta y aparece un cuadro de texto para que escriban abiertamente. Ej:

¿Pregunta de solo comentario?

respuesta relacionada con la pregunta en
 cuestión

Completamiento: son aquellas que luego del enunciado, asumen un formato similar a una oración que hay que completar, es decir aparece un cuadro de texto similar a “solo comentario” pero de una línea y a continuación de la frase incompleta. Ej:

¿Pregunta de completamiento?	
1- ¿Pregunta de completamiento 1?	<input type="text" value="respuesta 1"/>
2- ¿Pregunta de completamiento 2?	<input type="text" value="respuesta 2"/>
3- ¿Pregunta de completamiento 3?	<input type="text" value="respuesta 3"/>

3.7.2. Historia del Módulo

Historia de Usuario

Número: 03 **Usuario:** DTU

Nombre historia: Creación/Eliminación de encuestas

Prioridad en negocio: Alta **Riesgo en desarrollo:** Alta

(Alta / Media / Baja) **(Alto / Medio / Bajo)**

Puntos estimados: 40 horas **Iteración asignada:** 1

Programador responsable: Miguel Martínez Ruiz

Descripción: El especialista del sistema crea las encuestas teniendo en cuenta las solicitudes aceptadas y puede luego eliminarlas después de ser procesadas. Para la creación de encuestas se determinan distintos tipos de preguntas como son:

- 1- Opciones múltiples
- 2- Opciones múltiples + comentario
- 3- Opciones múltiples con rango
- 4- Opciones múltiples con rango + comentario
- 5- Opciones únicas
- 6- Solo comentario
- 7- Completamiento

Observaciones: Las encuestas deben tener título y comentario.

Tabla 18 Historia de Usuario 03 Módulo Creación/Eliminación de encuestas

3.7.3. Tareas del Módulo

Tarea para el Módulo de Creación/Eliminación de encuestas

Número tarea: 01 **Número historia de usuario:** 03

Nombre tarea: Crear Encuesta

Tipo de tarea : Desarrollo **Puntos estimados:** 8 horas

Fecha inicio: 11/05/2009 **Fecha fin:** 15/05/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al especialista del sistema crear las encuestas.

Tabla 19 Tarea 01 Módulo Creación/Eliminación de encuestas

Tarea para el Módulo de Creación/Eliminación de encuestas

Número tarea: 02 **Número historia de usuario:** 03

Nombre tarea: Determinar tipo de pregunta

Tipo de tarea : Desarrollo **Puntos estimados:** 8 horas

Fecha inicio: 18/05/2009 **Fecha fin:** 20/05/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al especialista la elección del tipo de pregunta a utilizar en la encuesta.

Tabla 20 Tarea 02 Módulo Creación/Eliminación de encuestas

Tarea para el Módulo de Creación/Eliminación de encuestas

Número tarea: 03 **Número historia de usuario:** 03

Nombre tarea: Eliminar Encuesta

Tipo de tarea : Desarrollo **Puntos estimados:** 8 horas

Fecha inicio: 21/05/2009 **Fecha fin:** 25/05/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: Permite al especialista eliminar las encuestas.

Tabla 21 Tarea 03 Módulo Creación/Eliminación de encuestas

Tarea para el Módulo de Creación/Eliminación de encuestas

Número tarea: 04 **Número historia de usuario:** 03

Nombre tarea: Listar encuestas

Tipo de tarea : Desarrollo **Puntos estimados:** 8 horas

Fecha inicio: 25/05/2009 **Fecha fin:** 29/05/2009

Programador responsable: Miguel Martínez Ruiz

Descripción: El sistema debe permitir listar todas las encuestas en dependencia del área que pertenezcan.

Tabla 22 Tarea 04 Módulo Creación/Eliminación de encuestas

3.8. Conclusiones

En este capítulo se pasó a la definición de los distintos módulos: administración, gestión de solicitud y creación/eliminación de encuestas. Para ello se realizó un profundo análisis de la solución propuesta utilizando los artefactos que propone la metodología Extreme Programming (XP). Es decir, en esta fase, una vez definidos los módulos que se implementarían, se establecieron las Historias de Usuarios correspondientes a cada uno de ellos. Por último se crearon las Tareas de programación necesarias para dar cumplimiento a las necesidades del cliente expuestas en dichas Historias de Usuarios.

Conclusiones

Este trabajo de diploma, desde su concepción inicial, estuvo orientado a la implementación de los módulos administración, gestión de solicitud y creación/eliminación de encuestas del sistema UCICuest sobre una plataforma de software libre para que en un futuro sea posible la utilización de dicho sistema por la Universidad de Ciencias Informáticas, acorde a los requerimientos funcionales de dicha institución o sus posibles clientes.

El valor fundamental de la idea de este proyecto es la gestión rápida, eficiente y segura de información recopilada a través de encuestas. Utilizando como referencia otras aplicaciones similares, pero teniendo en cuenta las características específicas que se planteaban, se pasó a la creación de los módulos necesarios.

Toda esta labor de preparación y búsqueda de información permitió la adquisición de conocimientos que posibilitaron a la postre el logro de los resultados que se muestran a continuación:

- 1- Se efectuó un detallado estudio de los diferentes tipos de encuestas.
- 2- Se realizó un análisis de los sistemas de gestión de encuesta y los procesos de informatización de los mismos.
- 3- Se reflejaron las principales características de las herramientas y tecnologías a utilizar durante el desarrollo.
- 4- Se logró la implementación de los módulos administración, gestión de solicitud y creación/eliminación de encuestas dando cumplimiento a los objetivos planteados.

Recomendaciones

Una vez concluido este trabajo de diploma, habiendo cumplimentado todos los objetivos propuestos en un inicio, y debido a que el sistema UCICuest aún no es funcional se plantean un conjunto de recomendaciones que se considera importantes a tener en cuenta para un futuro:

- 1- Continuar el desarrollo de los módulos restantes para hacer de UCICuest una herramienta útil y terminada para la Universidad de las Ciencias Informáticas.
- 2- Agregar la funcionalidad del envío de correos interna y de forma automática.
- 3- Utilizar UCICuest como base para la implementación de sistemas similares en otros escenarios.

Referencias Bibliográficas

1. Unavarra. [En línea] [Citado el: 30 de 11 de 2008.]
http://www.unavarra.es/personal/vidaldiaz/pdf/tipos_encuestas.PDF.
2. **Flores, Raúl Páramo.** solucionesnetques. [En línea] 9 de 2004. [Citado el: 1 de 12 de 2008.]
http://www.solucionesnetquest.com/papers/Rparamo_Estandares_enonline.pdf.
3. **Espinosa, Rafael A. Hernández.** monografias. [En línea] 2006. [Citado el: 1 de 12 de 2008.]
<http://www.monografias.com/trabajos40/software-cuba/software-cuba.shtml>.
4. Promonegocios. [En línea] [Citado el: 2 de 12 de 2008.]
<http://www.promonegocios.net/mercadotecnia/encuestas-tipos.html>.
5. lycos. [En línea] [Citado el: 4 de 12 de 2008.]
<http://usuarios.lycos.es/direccion/manuales/ENCUESTA.html> .
6. Gestipolis. [En línea] [Citado el: 4 de 12 de 2008.] www.Gestipolis.com.
7. lmi. [En línea] 2004. [Citado el: 26 de 03 de 2009.] <http://www.lmi.ub.es/edutec2004/pdf/903.pdf>.
8. **Daniel Mínguez Sanz, Emilio José García Morales.** eici. [En línea] [Citado el: 14 de 12 de 2008.]
http://www.eici.ucm.cl/Academicos/ygomez/descargas/Ing_Sw2/apuntes/DASBD-Methodolog-ADAsParaEIDesarrolloDeaplicacionesWeb_UWE.pdf.
9. Adelat. [En línea] [Citado el: 20 de 12 de 2008.]
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
10. Maestros del web. [En línea] 2 de 11 de 2007. [Citado el: 8 de 1 de 2009.]
<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
11. Iverclaros. [En línea] 8 de 3 de 2006. [Citado el: 11 de 1 de 2009.] <http://iverclaros.blog.galeon.com/>.
12. **Fabien Potencier, François Zaninotto.** librosweb. [En línea] [Citado el: 16 de 1 de 2009.]
<http://www.librosweb.es/symfony/>.
13. Scribd. [En línea] [Citado el: 12 de 1 de 2009.] <http://www.scribd.com/doc/2563181/Cakephp-manual-espanol-11?page=1>.
14. **Leopoldo, Carlos.** carlosleopoldo. [En línea] 27 de 11 de 2007. [Citado el: 13 de 1 de 2009.]
<http://www.carlosleopoldo.com/post/zend-framework-una-introduccion/>.

15. [En línea] [Citado el: 18 de 12 de 2008.]
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf.
16. **Laura Bermejo Sanz, Enrique Gómez Monreal.** kybele. [En línea] [Citado el: 16 de 1 de 2009.]
<http://kybele.escet.urjc.es/documentos/HC/Exposiciones/EclipseIDE.pdf>.
17. Maestros del web. [En línea] 31 de 1 de 2008. <http://www.maestrosdelweb.com/editorial/editores-web-que-facilitan-tu-trabajo/>.
18. Ocw. [En línea] 3 de 4 de 2004. [Citado el: 16 de 1 de 2009.] http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/desarrollo-de-aplicaciones-web/desarrollo-de-aplicaciones-web/XP06_M2108_01497.pdf.
19. Agpd. [En línea] 3 de 4 de 2004. [Citado el: 18 de 1 de 2009.]
<https://www.agpd.es/index.php?idSeccion=541>.
20. Barrapunto. [En línea] 2 de 3 de 2008. [Citado el: 27 de 3 de 2009.]
<http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/html/c20.html>.
21. Desarrollo web. [En línea] 2 de 3 de 2008. [Citado el: 28 de 3 de 2009.]
<http://www.desarrolloweb.com/articulos/1112.php>.
22. **José H. Canós, Patricio Letelier, Carmen Penadés.** willydev. [En línea] [Citado el: 12 de 1 de 2009.]
<http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
23. Scribd. [En línea] [Citado el: 14 de 1 de 2009.] <http://www.scribd.com/doc/297224/RUP>.
24. Java hispano. [En línea] [Citado el: 12 de 1 de 2009.]
http://www.javahispano.org/contenidos/es/procesos_de_desarrollo/?jsessionid=2A959BC68965C38A066B146BDA2A8ECC.

Bibliografía

1. **Maite Rodríguez Corbea, Meylin Ordóñez Pérez.** La metodología XP aplicable al desarrollo del Software Educativo en Cuba. Ciudad Habana. Universidad de las Ciencias Informáticas : s.n., 2007.
2. **Olivares Tamayo, Jorge Daniel y Rey Almaguer, Bernardo.** Desarrollo del Canal Informativo del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela: Subsistema de Administración. Ciudad de la Habana : s.n., 2008.
3. **Potencier, Fabien y Zaninotto, François.** Symfony la guía definitiva. 2007.
4. **Ramírez, Danay Pérez.** Investigación de la metodología ágil Extreme Programming y su aplicación en un caso de estudio. Ciudad Habana. Instituto Superior Politecnico J. A. Echeverría : s.n., 2008.
5. **Rolando Alfredo Hernández León, Sayda Coello González.** Paradigma cuantitativo de la investigación científica. Ciudad Habana : Editorial Universitaria, 2002.