**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULDAD DE ELECTROTECNIA Y COMPUTACIÓN**

**(F.E.C)**

Protocolo monográfico para optar al título de Ingeniero en computación

***“Propuesta de aplicación web para el procesamiento y gestión de datos recolectados por el Centro de Atención a la Mujer (CAM)”***

Desarrollado por:

Br. Carlos de Jesús Canales López 2001-10442

Br. Yasser Edsel López Herrera 2009-29275

Tutor:

Ing. Luis Eduardo Chávez Mairena

Abril de 2016

Contenido

[**UNIVERSIDAD NACIONAL DE INGENIERÍA** 1](#_Toc449546618)

[1. INTRODUCCIÓN 3](#_Toc449546619)

[2. ANTECEDENTES 4](#_Toc449546620)

[3. JUSTIFICACIÓN 5](#_Toc449546621)

[4. OBJETIVOS 7](#_Toc449546622)

[Objetivo General 7](#_Toc449546623)

[Objetivos Específicos 7](#_Toc449546624)

[5. MARCO TEÓRICO 8](#_Toc449546625)

[Ingeniería de software 8](#_Toc449546626)

[Modelos de desarrollo de software 8](#_Toc449546627)

[Modelo en cascada 9](#_Toc449546628)

[Modelo incremental 10](#_Toc449546629)

[Modelo en espiral 10](#_Toc449546630)

[Metodologías de desarrollo de software 11](#_Toc449546631)

[Métodos tradicionales 12](#_Toc449546632)

[Métodos ágiles 12](#_Toc449546633)

[6. DISEÑO METODOLÓGICO 13](#_Toc449546634)

[¿Por qué el modelo Incremental? 14](#_Toc449546635)

[¿Por qué el método Scrum? 15](#_Toc449546636)

[Tecnologías a aplicar 17](#_Toc449546637)

[7. CRONOGRAMA DE EJECUCIÓN 19](#_Toc449546638)

[Bibliografía 19](#_Toc449546639)

# INTRODUCCIÓN

El aborto es un tema de mucha controversia en las diversas sociedades del mundo. Muchas personas están en contra de esta práctica; otros están a favor, pero siempre y cuando, sea en una situación de riesgo de vida de las madres o mal formación congénita del feto. Pero también, hay otras personas, especialmente mujeres que, están a favor de dicha práctica cuando la madre lo desee, argumentando que es de ellas su cuerpo, y por tal razón, ellas pueden hacer y deshacer con su cuerpo y su feto.

Las leyes varían de país en país, todo depende de la cultura de sus poblaciones, de la opción religiosa de la mayoría, de las situaciones políticas, entre otras. Sin embargo, la mayoría de los seres humanos, tanto hombres, y especialmente mujeres, condenan los actos de esta índole realizados por muchas mujeres y las personas que las apoyan y las asisten en los mismos. En nuestro país – Nicaragua – este tema es muy delicado, debido a que la mayoría de las personas se identifica como cristianas o creyentes en Dios, el cual en sus mandatos, dice que no hay que matar.

Existen muchas organizaciones no gubernamentales que trabajan con niñas y mujeres que han tomado la decisión de abortar, pero la finalidad de estas organizaciones es persuadirlas a no cometer tal acto. Tal es el caso de la organización Centro de Atención a la Mujer – CAM – que trabaja de forma independiente con niñas y mujeres embarazadas en los departamentos de Managua, León y Chinandega. CAM asiste un promedio de 15 niñas y mujeres al mes, lo cual se convierte en 180 féminas asistidas al año, estas cifras solo aplican a la ciudad de Managua.

El presente protocolo monográfico pretende validar la idea del diseño de una aplicación informática que ayude a la gestión de los datos que el Centro de Atención a la Mujer – CAM - obtiene de las mujeres y niñas asistidas por esta organización.

# ANTECEDENTES

En Nicaragua no existen cifras oficiales de los abortos realizados, ya que desde el año 2006 esta práctica está penalizada en su totalidad, aún si es un caso donde la madre corre riesgo de muerte o hay malformación congénita del feto, lo cual es conocido como aborto terapéutico. Debido a esta prohibición total, las mujeres que se practican un aborto lo hacen en lugares clandestinos, con personas con pocos conocimientos médicos y en condiciones no adecuadas para tal práctica.

Según el Guttmacher Institute en su sitio web indica que, el 98% de los abortos inseguros se da en los países en desarrollo. Además, se afirma que, la probabilidad de que una mujer tenga un aborto se eleva ligeramente si vive en una región en desarrollo. Adicionalmente, este instituto señala que, en “*América Latina las tasas de aborto subregionales varían desde 29 en América Central (subregión que incluye México) a 32 en América del Sur y 39 en el Caribe. El Caribe (subregión que incluye Cuba, en donde los abortos son generalmente seguros) tiene la proporción regional más baja de abortos que son inseguros (46%), en comparación con cerca del 100% en América Central y América del Sur*”.

El Centro de Atención a la Mujer – CAM – estima que, un 5% de las mujeres atendidas por este centro abortan, y un 3% de ellas desaparecen del programa; es decir, que abandonan en su totalidad la asistencia que se les brinda en el centro. Las diferentes razones por las que las mujeres atendidas desean abortan son principalmente, razones económicas, adulterio, violación, abandono por parte del padre del bebé (madres solteras) y estudios (secundaria o universidad).

Según las experiencias de CAM les han permitido observar que el nivel de escolaridad incide grandemente en la decisión de las mujeres para abortar. Se ha estimado que un 50% de estas mujeres tiene un nivel de escolaridad bajo. Mientras que las que tienen un nivel de escolaridad medio son un 30%, y las de escolaridad alta conforman un 20%.

Adicionalmente, CAM ha estimado que un 40% de las mujeres que asisten son reincidentes en la decisión de abortar. También, el 20% del total de las féminas asistidas toman tal decisión por pura irresponsabilidad, al no tomar las precauciones respectivas al momento del acto sexual o por puro capricho.

# JUSTIFICACIÓN

Los datos que recolecta el Centro de Atención a la Mujer – CAM – provienen de las mujeres y adolescentes que llegan al centro en busca de ayuda. Cada fémina al ser atendida brinda una serie de datos, entre los cuales podemos mencionar, Datos generales, Información de contacto, Razones por las que desea abortar, entre otras. Todos estos datos son llenados a mano en un formato de nuevo expediente, del cual se extraen cada uno de los datos para brindarle la atención requerida a las solicitantes de ayuda, y para fines estadísticos.

Cada expediente llenado a mano se pasa luego a una base de datos en Excel, con el fin de tener un respaldo electrónico y facilitar un poco la manipulación de los datos. Sin embargo, esto no es suficiente para agilizar las tareas de procesado de la información. La peor situación se da cuando se debe hacer las estadísticas de los casos atendidos, donde los responsables de hacer esa tarea se pasan horas y días procesando la información, revisando caso por caso para clasificarlos, de acuerdo a la información que se desee obtener. Todos los expedientes se deben de revisar más de una vez, debido a que muchas de las mujeres asistidas cambian de opinión en muchas ocasiones, y hay que estar actualizando la información hasta que se verifica que el bebé nació o fue abortado.

Las estadísticas se deben hacer mes a mes, ya que las mismas deben ser presentadas, por medio de boletines mensuales, a los donantes que contribuyen al mantenimiento de las actividades del centro y la ayuda que se le brinda a cada mujer o adolescente embarazada. Estos boletines reportan el control mensual de atención, la calidad de la misma, total de logros, etc. Para poder realizar estas tareas se emplea una gran cantidad de tiempo en el procesado de los datos para obtener la información requerida.

La aplicación informática que se pretende diseñar, no salvará vidas de manera directa, pero sí aportará un grano de arena en el trabajo realizado por CAM, ayudándolos a hacer su trabajo de una manera más rápida, ahorrándose tiempo de procesamiento de datos, el cual podría ser utilizado para hacer trabajo de campo con las mujeres y adolescentes que acuden a la organización en busca de ayuda. Además, podrán elaborar con mayor eficiencia los boletines para los donantes, lo cual puede servir, también, para atraer más personas que apoyen al centro para salvar vidas. Todo ser humano tiene un propósito en este mundo, y puede que dentro de los bebes que se salvan de ser abortados por las mujeres que acuden a CAM, se encuentren futuros grandes ingenieros, médicos, economistas, científicos, hombres y mujeres de negocio que puedan realizar excelentes trabajos que beneficien grandemente a las sociedades del mundo.

# OBJETIVOS

## Objetivo General

Proponer una aplicación web que ayude a mejorar el procesamiento y gestión de los datos recopilados por el Centro de Atención a la Mujer – CAM – para procesar información necesaria para la organización en un menor tiempo posible.

## Objetivos Específicos

1. Realizar el análisis del modelo de negocio del Centro de Atención a la Mujer.
2. Recopilar toda la información necesaria para elaborar el diseño de una aplicación web, adecuado para el Centro de Atención a la Mujer – CAM.
3. Codificar todos los módulos e interfaces gráficas de usuarios requeridos en la aplicación web diseñada.
4. Realizar pruebas de caja blanca y caja negra para asegurar el correcto funcionamiento de la aplicación.

# MARCO TEÓRICO

Para diseñar una aplicación informática, se deben llevar a cabo una serie de pasos bien elaborados y planeados para obtener una aplicación de calidad que satisfaga las necesidades del cliente o usuario(s) y consuma los más mínimos recursos económicos y humanos al momento de su desarrollo.

## Ingeniería de software

La ingeniería de software nos brinda una serie de herramientas que nos ayudan y facilitan el proceso de planificación y desarrollo de una aplicación informática.

*“La ingeniería de software es una actividad para la solución de problemas. Se usan los modelos para buscar una solución aceptable. Esta búsqueda es conducida por la experimentación. Los ingenieros de software no tienen recursos infinitos, y están restringidos por presupuestos y tiempos de entrega”.* (Bruegge y Dutoit, 2002, p. 5).

Los modelos son muy importantes en el diseño de una aplicación informática, ya que esta es la manera en que el analista plasma su visión y conceptualización del problema y su posible solución. En la ingeniería de software, por lo general, se usa una representación gráfica para el modelado del problema y la interacción de los usuarios con la aplicación.

Según Pressman (2005), la ingeniería de software es una tecnología estratificada, cuyo enfoque debe estar comprometido con la calidad. Por tal razón, los diseñadores de aplicaciones informáticas deben de tener en cuenta estándares y normas que indican acciones a seguir para conseguir un producto que cumpla con los mínimos requisitos de calidad.

## Modelos de desarrollo de software

Un modelo de desarrollo de software es una serie de acciones que se deben llevar a cabo para obtener un producto final – el software. Existen varios modelos para desarrollar software; sin embargo, los desarrolladores deben escoger uno de ellos para realizar su trabajo. A continuación se mencionarán algunos de los modelos más conocidos y utilizados.

### Modelo en cascada

Este modelo ordena de forma rigurosa las etapas del [proceso para el desarrollo de software](http://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_software), en tal forma que, el inicio de cada etapa debe esperar a la finalización de la etapa anterior. El modelo en cascada define las siguientes etapas para el desarrollo de software.

*Ingeniería y Análisis del Sistema:* debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

*Análisis de los requisitos del software:* el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (analista) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

*Diseño:* el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

*Codificación:* el diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

*Prueba:* una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

Mantenimiento: el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

### Modelo incremental

El modelo incremental es una evolución del modelo de cascada; viene a suplir el problema de no poder retroceder en las fases de desarrollo del software. Es, por tanto, un modelo no secuencial. El funcionamiento es sencillo. Comienza con el análisis de los requisitos, tras el cual se prepara un primer diseño. La novedad de este modelo respecto del anterior, es la introducción de iteraciones para dividir diseños.

Este modelo ofrece la posibilidad de comenzar un diseño del software en caso que no convenza al cliente o a los desarrolladores, comenzando de esta manera una nueva iteración sin la necesidad de realizar un nuevo análisis de requisitos. Pueden realizarse tantas iteraciones como sean necesarias.

Con este modelo se mantiene al cliente en constante contacto con los resultados obtenidos en  cada incremento. Así, es el mismo cliente el que incluye o desecha elementos al final de cada incremento a fin de que el software se adapte mejor a sus necesidades reales. El proceso se repite hasta que se elabore el producto completo.

### Modelo en espiral

Al igual que el modelo incremental, este modelo es una evolución del modelo clásico en cascada. Este modelo equivale al de cascada, pero en él se permite el solapamiento de varias etapas con el objetivo de flexibilizar y compensar el tiempo de desarrollo total y alcanzar resultados funcionales en etapas tempranas.

Está considerado como un método de desarrollo rápido y eficiente.  
Es adecuada para proyectos en los que se tienen claros los objetivos finales pero no todos los detalles de implementación están definidos.

Normalmente el proyecto se divide en módulos pequeños y a cada uno de ellos se les aplica el siguiente proceso.

*Análisis de requerimientos*: se establecen todos los detalles funcionales deseados.

*Diseño del sistema*: con los datos de la etapa anterior se procede a diseñar las bases de datos, interfaces de usuario, etc.

*Etapas de construcción*: esta etapa de construcción comprende básicamente la codificación y test de unidades. Esta etapa es un trabajo de programación pura.  
*Test y evaluación*:en esta etapa se realiza un test del módulo completo así como su evaluación frente al estudio de requerimientos. En muchos casos en esta etapa los usuarios finales participan de manera activa aportando información decisiva para la usabilidad del sistema.

## Metodologías de desarrollo de software

Existen diversos métodos para el proceso de desarrollo de software, cada uno de ellos nos brindan una serie de herramientas, paso y técnicas para tener éxito a lo largo del ciclo de vida del software. En el desarrollo de software existen dos tipos de metodologías que definen y explican diferentes métodos del desarrollo de software; estas metodologías son, la metodología tradicional y la metodología ágil.

*“El origen de la metodología tradicional del desarrollo del software es la metodología de construcción de productos físicos (barcos, edificios, coches, etc.). Esta metodología tradicional del desarrollo del software se caracteriza por basarse en un ciclo de vida de desarrollo del software en cascada, ya que organiza los proyectos en etapas que se ejecutan secuencialmente”*. (Gordillo, 2014).

Gordillo (2014) expresa que, la metodología ágil no se basa en la metodología de construcción de productos físicos, sino que se basa en un ciclo de vida de desarrollo de software iterativo e incremental, donde se solapan las etapas de desarrollo y hay un cambio de la documentación por la interacción cara a cara con el usuario, además que, los equipos de desarrollo son multidisciplinarios.

A continuación se describirán brevemente los dos métodos más conocidos de cada una de estas metodologías.

### Métodos tradicionales

**RUP (Rational Unified Process)**

Esta metodología divide el desarrollo del software en cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada una de estas etapas es desarrollada mediante un ciclo de operaciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación.

**MSF (Microsoft Solutions Framework)**

Es un método que puede ser escalable según la magnitud del proyecto de desarrollo; además, está basado en el modelo en cascada y en el modelo en espiral. Pérez (2011) indica que, este método es adaptable, flexible y escalable, e independiente de tecnologías, lo cual significa que no se cierra a un sólo modelo de programación sino más bien queda abierto según la naturaleza del proyecto. MSF es un método con ocho principios fundamentales: Fomentar la comunicación abierta, Trabajar hacia una visión compartida, Empoderar a los miembros del equipo, Establecer la rendición de cuentas claras y la responsabilidad compartida, Centrarse en ofrecer valor empresarial, Mantenerse ágil y en espera de un cambio, Invertir en la calidad y Aprender de todas las experiencias.

### Métodos ágiles

**Método Scrum**

La metodología Scrum nos permite gestionar los productos a través de listas de productos priorizados. En consecuencia, se hace sumamente necesario saber de manera exhaustiva qué es lo que hay hacer, para tal propósito el equipo debe de mantener una comunicación constante con el cliente (Product Owner), porque es éste el que sabe realmente qué producto necesita, cómo lo necesita y para qué lo necesita. Para lograr un buen avance del proyecto con la metodología Scrum, se priorizan las cosas que aportan más y cuestan menos en el mismo; por último se dejan las cosas que aportan menos y cuestan más. Los riesgos siempre están presentes, y por eso con la metodología Scrum hay que correr primero con los riesgos más grandes, para aprender de ellos y tener el suficiente tiempo para eliminarlos de una mejor manera. También, vale mencionar que los requisitos no entendidos hay que tratar de satisfacerlos hasta que ya hayamos aprendido a cómo enfrentarlos.

**Programación extrema (XP)**

La programación extrema se puede definir como un conjunto de pasos de diversas metodologías, acopladas de manera que sean pasos flexibles a seguir utilizadas con el uso común, para realizar un desarrollo más agradable y sencillo.  
Esta metodología tiene como base la simplicidad y como objetivo principal la satisfacción del cliente; para lograrlo se deben tomar en cuenta cuatro valores fundamentales, Comunicación, Simplicidad, Retroalimentación y Coraje.

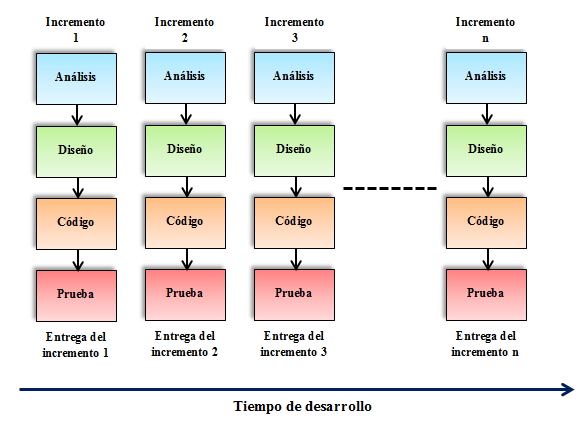
# DISEÑO METODOLÓGICO

Después de analizar los diferentes modelos y los métodos de desarrollo de software hemos llegado a la conclusión de que el modelo Incremental y el método Scrum son los más adecuados para el desarrollo de la aplicación informática que necesita el Centro de Atención a la Mujer – CAM – para la gestión y procesamiento de los datos que ellos obtienen de las mujeres y adolescentes que asisten al centro en busca de ayuda. Por tanto, a continuación, abordaremos un poco más del modelo Incremental y el método Scrum.

## ¿Por qué el modelo Incremental?

En el modelo incremental la relación entre los clientes y desarrolladores se hace estrecha, debido a que la comunicación del equipo de desarrollo con los clientes se hace de manera periódica para planificar y validar los próximos avances del proyecto. El cliente es el que decide los próximos avances que el equipo de desarrollo llevará a cabo de acuerdo a sus necesidades.

En este modelo, el software se va desarrollando poco a poco, por módulo, donde cada uno de éstos es entregado al cliente cuando se finaliza su etapa de desarrollo. Es decir, no se espera a que se finalice totalmente el proyecto de desarrollo para entregar algo funcional, sino que se van entregando partes funcionales del software hasta lograr obtener un todo funcional. Cada módulo funcional desarrollado es un ciclo o una iteración; por tanto, todo el proyecto de desarrollo del software se vuelve un conjunto de iteraciones, donde los pasos a seguir son los mismos, se repiten en cada iteración, a como se muestra en la siguiente figura.



Según Calero (2010) el modelo incremental tiene las siguientes ventajas.

* Reducción del tiempo de desarrollo inicial.
* Entregas tempranas y parciales de partes operativas del software.
* Proporciona todas las ventajas del modelo en cascada realimentado, reduciendo sus desventajas sólo al ámbito de cada incremento.
* Permite la entrega más rápida de un producto al cliente.
* Los cambios son más sencillos de manejar.

Adicionalmente calero (2010) enuncia las siguientes desventajas de este modelo de desarrollo.

* No es recomendable para sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido, y/o de alto índice de riesgos.
* Se requiere de mucha planeación, tanto administrativa como técnica.
* Se requiere de metas claras para conocer el estado del proyecto.

## ¿Por qué el método Scrum?

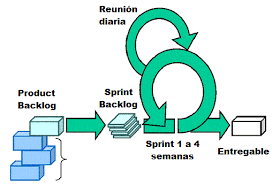
El método Scrum nos permite gestionar los productos a través de listas de productos priorizados. En consecuencia, se hace sumamente necesario saber de manera exhaustiva qué es lo que hay hacer, para tal propósito el equipo debe de mantener una comunicación constante con el cliente, porque es éste el que sabe realmente qué producto necesita, cómo lo necesita y para qué lo necesita. Para lograr un buen avance del proyecto con el método Scrum, se priorizan las cosas que aportan más y cuestan menos en el mismo; por último se dejan las cosas que aportan menos y cuestan más. Los riesgos siempre están presentes, y por eso con el método Scrum hay que correr primero con los riesgos más grandes, para aprender de ellos y tener el suficiente tiempo para eliminarlos de una mejor manera. También, vale mencionar que, los requisitos no entendidos hay que tratar de satisfacerlos hasta que ya hayamos aprendido a cómo enfrentarlos.

Scrum se basa en un proceso constructivo, iterativo e incremental donde las iteraciones tienen duración fija. Además, contiene definición de roles, prácticas y productos de trabajo estrictas en forma simple.

Sus elementos son:

1. Roles
   1. Product owner: Es el interesado en el proyecto. Es la persona que dicta los requisitos que debe poseer el sistema. Se encarga de que el retorno de la inversión sea satisfactoria para los demás interesados.
   2. Scrum master: Es el líder del equipo. Resuelve los problemas y quita los obstáculos, dirige y facilita herramientas al equipo de desarrollo.
   3. Equipo: Son las personas que se encargan del desarrollo del sistema.
2. Product backlog: Son todas las actividades y requerimientos del sistema.
3. Sprint: Periodo de tiempo en que el equipo se compromete a desarrollar una funcionalidad del sistema.
4. Sprint backlog: Trabajo o tareas determinadas por el equipo para realizar un sprint.

La siguiente imagen resume gráficamente el proceso de desarrollo con el método Scrum.



Fuente: http://agilizandoando.blogspot.com/2012/05/scrum.html

## Tecnologías a aplicar

Debido a que la aplicación informática que se pretende desarrollar será destinada a una ONG, los gastos en recursos deben de ser mínimos o de preferencia nulos. Por lo tanto, las tecnologías de desarrollo a utilizar serán Open Source o bajo licenciamiento de estudiante. En la siguiente tabla se mencionan y se describen las tecnologías que se utilizarán en el desarrollo de la aplicación.

|  |  |  |
| --- | --- | --- |
| ***Tipo*** | ***Nombre*** | ***Descripción*** |
| Servidor WEB | Microsoft Azure | Microsoft otorga a estudiantes registrados con sus cuentas de correo estudiantil licencias permanentes para uso libre de sus tecnologías; en este caso se hará uso de Azure el cual es un servicio en la nube que permite crear máquinas virtuales y poseer servidores para la puesta en marcha del sistema, se cuenta con una cuota mensual de $150 los cuales bastan para el funcionamiento del sitio. Los $150 al mes son capaces de mantener una máquina con un núcleo quadcore, 7GB de RAM y 120GB de disco duro en estado sólido. |
| Entorno de producción XAMP | Es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP. Este entorno será instalado en la máquina virtual proveída por Azure. |
| Lenguajes de programación | PHP | Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc. |
| JavaScript | Es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestra computadora, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. |
| Lenguaje de Hipertexto | HTML | Es el lenguaje que se emplea para el desarrollo de páginas de sitios WEB. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc. En combinación con los lenguajes de programación JavaScript y PHP HTML permite crear sitios dinámicos y capaces de cumplir con las necesidades de cualquier requerimiento solicitado aportando agilidad y facilidad de uso desde cualquier sitio con un navegador y acceso a internet. |
| Framework | LARAVEL | es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC. Laravel, propone una forma de desarrollar aplicaciones web de un modo mucho más ágil. Por ejemplo, en Laravel podemos usar el patrón de diseño MVC (Modelo-Vista-Controlador) tradicional, donde el controlador es programado como una clase. Por lo tanto, un Controlador es una clase PHP que dispone de métodos públicos que son el punto de entrada final de una petición HTTP (Request PHP) a nuestra aplicación. |

Framework: Un Framework es un entorno o ambiente de trabajo para desarrollo; dependiendo del lenguaje normalmente integra componentes que facilitan el desarrollo de aplicaciones como el soporte de programa, bibliotecas, plantillas y más.

# CRONOGRAMA DE EJECUCIÓN

No hay cronograma de ejecución por el momento

# Bibliografía

Bermúdez, V. (18 de junio de 2014). *http://www.confidencial.com.ni/.* Recuperado el 07 de abril de 2016, de http://www.confidencial.com.ni/: http://www.confidencial.com.ni/archivos/articulo/17968/aborto-inseguro-mata-a-47-mil-mujeres-al-ano

Bernd Bruegge, A. H. (2002). *Ingeniería de Software Orientado a Objetos.* México: Pearson Education.

Calero, W. (8 de octubre de 2010). *http://ingenieraupoliana.blogspot.com/*. Recuperado el 17 de abril de 2016, de http://ingenieraupoliana.blogspot.com/: http://ingenieraupoliana.blogspot.com/2010/10/modelo-incremental.html

Gordillo, E. (21 de octubre de 2014). *inventtatte.com*. Recuperado el 14 de abril de 2016, de inventtatte.com: https://inventtatte.com/metodologia-tradicional-vs-agil/

Institute, G. (noviembre de 2015). *www.guttmacher.org.* Recuperado el 08 de abril de 2016, de www.guttmacher.org: https://www.guttmacher.org/es/fact-sheet/datos-sobre-el-aborto-inducido-en-el-mundo

Pérez, O. (2011). Cuatroo enfoques metodológicos para el desarrollo de software RUP - MSF - XP - SCRUM. *Inventum*.

Pressman, R. (2005). *Ingeniería del Software. Un enfoque práctico.* México: McGraw-Hill.