



**UNIVERSIDAD POLITÉCNICA DE SINALOA**

**PROGRAMA ACADÉMICO DE  
INGENIERÍA EN INFORMÁTICA**

# Tesina

“Agentes de *Business Analyst* en el área de  
preventa”

Para cumplir la acreditación de las estadías profesionales y contar  
con los créditos necesarios para obtener el grado de Ingeniero en  
Informática

**Autor:**

**Liliana Guadalupe Osuna Tirado**

**Asesor: Dr. Rodolfo Ostos Robles**

**Asesor OR: Ing. Siddartha Nava Gallego**

*Mazatlán, Sinaloa 9 de Diciembre de 2016*



Zapopan, Jalisco 25 de Octubre de 2016.

LIC. NUBIA VANESSA FÉLIX VALENZUELA  
DIRECCION DE VINCULACION, DIFUSIÓN Y EXT. UNIVERSITARIA  
UNIVERSIDAD POLITÉCNICA DE SINALOA  
PRESENTE

Por medio de la presente, informo a usted que la C. Liliana Guadalupe Osuna Tirado, alumna de la Universidad Politécnica de Sinaloa de la carrera de Ingeniería en Informática, con número de matrícula 2013030470, ha sido aceptado para realizar sus Residencias Profesionales en esta empresa, participando como Analista de Sistemas para diferentes proyectos internos bajo la asesoría del Ing. Siddartha Nava Gallegos, cumpliendo un total de 500 horas comprendidas en el periodo de tiempo del 22 de Agosto del 2016 al 20 de Enero del 2017 tentativamente.

Sin más por el momento, quedo a sus órdenes para cualquier aclaración al respecto.

ATENTAMENTE

Ing. Siddartha Nava Gallegos  
Gerente General  
+52 33 1561 3626  
sdf.nava@jaguarlabs.com



Gral. Ramón Corona N. 3455 F. Col. La Mojonera, Zapopan, Jalisco, México.  
Tel. 1561 3626 y 1561 3628



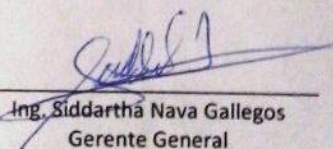
Zapopan, Jalisco a 06 de Diciembre de 2016.

LIC. NUBIA VANESSA FÉLIZ VALENZUELA.  
DIRECCION DE VINCULACION, DIFUSION Y EXT. UNIVERSITARIA  
UNIVERSIDAD POLITÉCNICA DE SINALOA  
PRESENTE

Por medio de la presente, informo a usted que la C. Lilita Guadalupe Osuna Tirado , alumna de la carrera **Ingeniería en Informática** en la Universidad Politécnica de Sinaloa , con número de matrícula **2013030470**, ha finalizados sus Prácticas profesionales en ésta empresa, participando como **"Agente de Bussines Analyst para el área de preventa."**, bajo la asesoría del Ing. **Siddartha Nava Gallegos**, cumpliendo un total de 500 horas comprendidas en el periodo de tiempo de **22 de Agosto del 2015 al 06 de Diciembre del 2015** tentativamente.

Sin más por el momento, quedo a sus órdenes para cualquier aclaración al respecto.

ATENTAMENTE

  
Ing. Siddartha Nava Gallegos  
Gerente General  
+52 33 1561 3626  
[sid.nava@jaguarlabs.com](mailto:sid.nava@jaguarlabs.com)

Gral. Ramón Corona N. 2455 F. Col. La Mojonera, Zapopan, Jalisco, México.  
Tel. 1561 3626 y 1561 3628



## Universidad Politécnica de Sinaloa



C. Lilitana Guadalupe Osuna Tirado  
Presente.-

Folio 2013030470-2016-058

Por medio de la presente me permito comunicarle que es de aceptarse el tema de tesina, el cuál se ha solicitado bajo el título:

### ***“Agentes de Business Analyst en el área de preventa”***

mismo que usted desarrollará con objeto de dar lugar a los trámites conducentes para la acreditación de la asignatura de Estadías Profesionales de la Unidad Académica de Ingeniería en Informática:

#### **Ingeniería en Informática**

Así mismo, le comunico que para el desarrollo de la citada tesina le ha sido asignado como Director Asesor de la misma al: **Dr. Rodolfo Ostos Robles** y como asesores a la **Dra. Vanessa Guadalupe Félix Aviña** y al **Dr. Luis Javier Mena Camaré**

Sin otro particular por el momento, aprovecho la ocasión para enviarle un cordial saludo.

Atentamente

Ismaylia Saucedo Ugalde

M.T.I. Ismaylia Saucedo Ugalde

Directora de la Unidad Académica de Ingeniería en Informática  
Universidad Politécnica de Sinaloa



UNIVERSIDADES POLITÉCNICAS





## Universidad Politécnica de Sinaloa



C. Liliana Guadalupe Osuna Tirado  
Ingeniero en Informática  
Presente.-

Folio: 2013030470-2016-058

Por este conducto le envío un cordial saludo y aprovecho la ocasión para notificarle que el jurado que le fue asignado para evaluar la tesina desarrollada en las estadías profesionales denominada "Agentes de Business Analyst en el área de preventiva" y que después de ser revisada en reunión de sinodales ante la Dirección de Ingeniería en Informática, integrado por:

PRESIDENTE DEL JURADO: Dr. Rodolfo Ostos Robles

SINODAL: Dra. Vanessa Guadalupe Félix Aviña

SINODAL: Dr. Luis Javier Mena Camaré

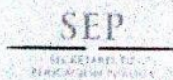
Ha decidido autorizar y aceptar la digitalización de la misma por el participante, conforme a la normatividad vigente y cumpliendo con los requisitos para tal caso.

Agradeciendo la atención a la presente, le reitero a Usted mi atenta consideración y respeto.

Atentamente

Ismaylia Saucedo Ugalde

M.T.I. Ismaylia Saucedo Ugalde  
Directora de la Unidad Académica de Ingeniería en Informática  
Universidad Politécnica de Sinaloa



DIRECCIÓN  
INFORMÁTICA  
UNIVERSIDADES POLITÉCNICAS



## **Agradecimientos**

Quiero agradecer a la universidad politécnica de Sinaloa y maestros por darme los conocimientos necesarios, como también a la empresa Jaguar Labs por aceptarme en su empresa y darme los su confianza, conocimientos, experiencia en el puesto de *Business Analyst* en el área de preventa, a mi asesor de tesina Rodolfo Ostos Robles por asesorarme en todo momento y dándome consejos y por último a mi familia que ha estado en todo momento conmigo.

## Índice

Resumen.....	5
Abstract.....	6
Introducción.....	7

## Capítulo I

1.1 Antecedentes.....	9
1.2 Localización.....	9
1.3 Objetivos y prioridades de la empresa.....	10
1.4 Organigrama de la empresa.....	10-11
1.5 Visión.....	11
1.6 Misión.....	11
1.7 Planteamiento del problema.....	12
1.8 Propuesta de investigación.....	12
1.9 Objetivos Generales.....	13
1.10 Particulares.....	13
1.11 Preguntas de investigación.....	13-14
1.12 Hipótesis.....	14
1.13 Limitaciones y supuestos.....	15
1.14 Relevancia.....	15

## Capitulo II

2.1 Introducción.....	17
-----------------------	----

2.2 La crisis del software.....	17
2.3 Ingeniería de software.....	18-23
2.4 Metodologías de desarrollo de software.....	23-31
2.5 Metodologías de desarrollo modernas.....	31-38
2.6 Análisis de metodologías clásicas y modernas.....	38-39
2.7 Conclusiones.....	39

**Capítulo 3**

3.1 Introducción.....	42
3.2 Metodología.....	42-43
3.3 Scrum y análisis de problema de desarrollo.....	43-51
3.4 Desarrollo.....	52-54
3.5 Resultados.....	54-55
3.6 Conclusión.....	56-57
Glosario.....	58-59
Referencias.....	60-67



## Índice de imágenes

Imagen 1.1 Ubicación de Jaguar Labs.....	9
Imagen 1.2 Organigrama de la empresa.....	11
Imagen 2.1 Crisis del software.....	17
Imagen 2.2 Historia.....	18
Imagen 2.3 Ingeniería de software.....	21
Imagen 2.4 Producto y proceso de la ingeniería del software.....	23
Imagen 2.5 Modelo de cascada.....	26
Imagen 2.6 Método de prototipo.....	27
Imagen 2.7 Modelo incremental.....	28
Imagen 2.8 Modelo espiral.....	29
Imagen 2.9 Modelo RDA.....	31
Imagen 2.10 Metodología ágil.....	32
Imagen 2.11 Guía Scrum.....	33
Imagen 2.12 Kanban.....	35
Imagen 2.13 Metodología xp.....	37
Imagen 3.1 Metodología Scrum.....	43
Imagen 3.2 Estructura Wbs .....	54
Imagen 3.3 Prototipo.....	55

## **Resumen**

La presente tesis realiza el desempeño de un Business Analyst en el área de preventa de la empresa Jaguar Labs, levantamiento de requerimientos y la metodología scrum.

Haciendo énfasis en el las actividades realizadas por el Business Analyst en el área de preventa se cuenta con la estrategia de cómo alcanzar los objetivos vistos y así agilizar los procesos.

El la tesis se abordan temas de metodologías modernas y tradicionales, como también procesos, wbs, prototipo y sobre todo la metodología Scrum que es la metodología implementada por Jaguar Labs

Palabras claves: Metodología, Scrum, Wbs, Levantamiento de requerimiento.

## **Abstract**

The present thesis performs the performance of a Business Analyst in the area of pre-sales of the company Jaguar Labs, survey of requirements and the scrum methodology.

Emphasizing the activities carried out by the Business analyst in the area of pre-sales, there is a strategy of how to achieve the objectives and thus streamline the processes.

The thesis addresses issues of modern and traditional methodologies, as well as processes, wbs, prototype and above all the Scrum methodology, which is the methodology implemented by Jaguar Labs

Keywords: Methodology, Scrum, Wbs, Requirements survey

## **Introducción**

La presente tesina es una investigación que tiene por objetivo dar a conocer el trabajo que tiene un *Business Analyst* en el área de preventa y la metodología usada por la empresa Jaguar Labs.

Los datos se obtuvieron gracias a una larga investigación y ayuda de algunas personas encargadas de esto.

Esta investigación cuenta con tres capítulos donde el primero se aborda los antecedentes de la empresa y el planteamiento del problema, limitaciones, alcances y los objetivos de la investigación, en pocas palabras se abordarán los aspectos metodológicos.

En el capítulo II se encontrarán los aspectos teóricos de la ingeniería de software, las metodologías tradicionales y modernas.

Y por último en el capítulo III se abordará los resultados de la investigación donde daremos como solucionamos el problema y se debe dar conocer que todos los objetivos plasmados en el capítulo I fueron logrados.

# Capítulo I

## Antecedentes y planteamiento del problema

---

En este capítulo se verá la información de la empresa y el planteamiento del problema donde se formulará preguntas sobre el proyecto y así empezar a conocer el proyecto

---

## 1.1 Antecedentes.

El Jaguar Labs es una empresa mexicana de alta tecnología innovadora con más de 175 proyectos terminados. Cuenta con una plantilla de más de 70 ingenieros, diseñadores gráficos y *testers* que están al corriente de las últimas tendencias tecnológicas.

Jaguar Labs centra su área de negocio en el desarrollo de software orientado a aplicaciones web así como aplicaciones para teléfonos móviles tipo Smartphone de acuerdo a los requerimientos de las organizaciones de sus clientes.

Jaguar Labs nació en el 2012 y visualizo el desarrollo de software para aplicaciones móviles como un factor de mercado potencial para el desarrollo del negocio y considero desde su creación el uso de la metodología Ágil con el *framework Scrum*

## 1.2 Localización.

General Ramón Corona 2455 F, Colonia La Mojonera, Zapopan, Jalisco, México.  
C.P. 45200

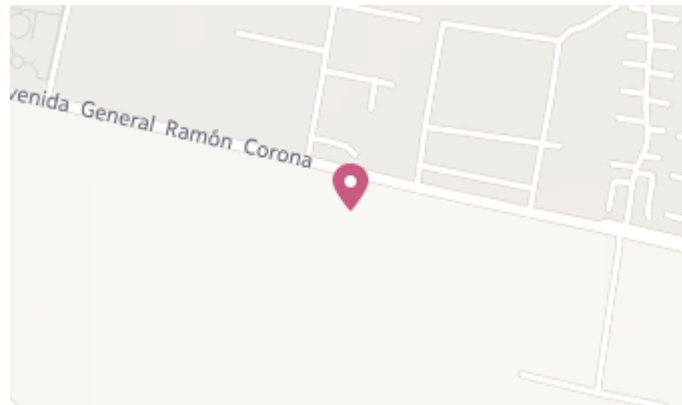


Imagen 1.1 Ubicación de Jaguar Labs

### **1.3 Objetivos y prioridades de la empresa.**

- Enfoque en satisfacción del cliente al 100%
- Enfoque en satisfacción del empleado al 85%
- Mejora de procesos, implementación CMMi 4
- Plan de crecimiento y capacitación
- Mejorar segmentación y selección de clientes
- Reforzar y crecer alianzas con socios estratégicos en México, USA y Europa
- Convertir a JaguarLabs en una marca reconocida nivel nacional.
- PROSOFT 2016
- Ser una compañía feliz

### **1.4 Organigrama de la empresa.**

La empresa se encuentra organizada de por distintos departamentos que se muestran en el diagrama.

#### 1.1. ORGANIGRAMA DE LA EMPRESA JAGUAR LABS

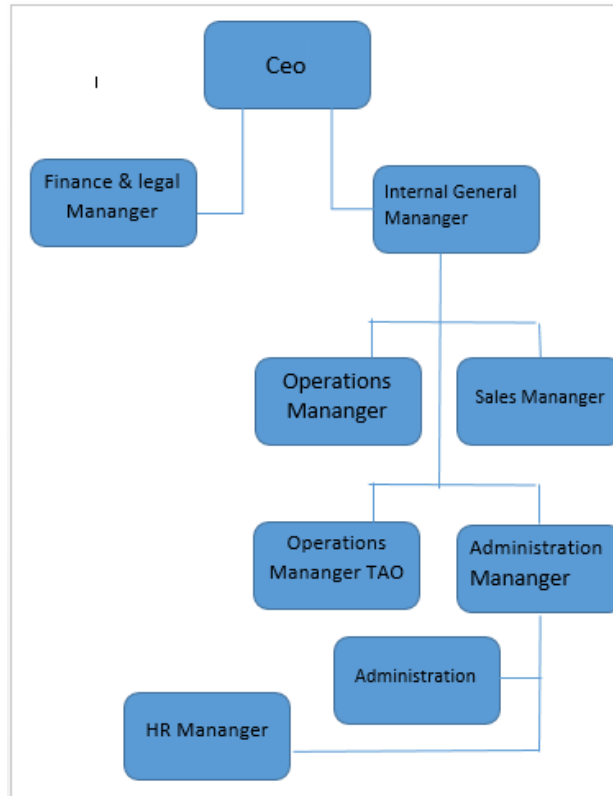


Imagen 1.2 Organización de la empresa

## 1.5 Visión.

Ser la empresa líder de México en el desarrollo de aplicaciones móviles propias o para terceros, proporcionando las mejores soluciones y experiencias a nuestros clientes y contando con el mejor equipo de asociados y socios en el mejor ambiente y formas de trabajo posibles.

## 1.6 Misión.

Proveer servicios de alta calidad, utilizando tecnologías de punta y métodos innovadores, buscando siempre superar las expectativas de nuestros clientes, además de contar con el equipo más capacitado y motivado para trabajar por el crecimiento personal, el prestigio y orgullo de formar parte de ella, y en donde finalmente, la cooperación, trabajo en equipo y respeto sean los cimientos de su éxito.



## 1.7 Planteamiento del problema.

Es muy común que en las empresas se desvíen constantemente de los proyectos ante los objetivos primordiales de los proyectos. Esto ocasiona algunos problemas como lo son: la falta de comunicación, que el proyecto no salga a tiempo, que el producto no salga como el usuario lo solicitó, que los requerimientos no estén bien definidos o simplemente la falta de los requerimientos.

Un sinnúmero de problemas pueden ser generados si no se encuentra una persona a cargo como lo son exceso de errores, falta de tiempo, mala coordinación entre los roles, requerimientos mal estructurados, pérdida de clientes y mal flujo de la aplicación.

## 1.8 Propuesta de investigación.

Debido a las diversas tareas que los *Business Analyst* (analistas de negocio) que realizan y los diversos proyectos que se obtenían se creó el puesto de *Business Analyst* en el área de preventa. Mismo que es el encargado de desempeñar tareas de análisis, levantamiento de requerimientos, creación de documentos de requerimientos, creación de estimación de costos y tiempos, prototipo e investigaciones ya sean documentadas o de mejoras entre otras tareas esto ayudando a la empresa y siguiendo la metodología scrum, las buenas prácticas de CMMI (*Capability Maturity Model Integration* y se refiere a los modelos que contienen las mejores prácticas que ayudan a las organizaciones a mejorar sus procesos) y dejando en problemas al momento del costo o del tiempo definido. [1]

Jaguar Labs no puede atender los diversos proyectos debido a la poca cantidad de empleados con la que cuenta. Esto genera un gran problema, por ello la empresa le es necesario generar más agentes de *Business Analyst* en el área de preventa para que participen en los diversos proyectos y así lograr un mejor trabajo y que todos los proyectos se cumplan en tiempo y forma para que sean exitosos. De esta forma es posible la satisfacción de las necesidades reales de los clientes.

## **1.9 Objetivos Generales.**

Esta investigación se ha planteado analizar, identificar soluciones o procesos que ayuden a las necesidades, metas y objetivos de los clientes como empresa mientras que Jaguar Labs para tener un mejor manejo hace uso de la metodología ágil SCRUM. Definiendo y sustentando iniciativas factibles que proporcioné beneficios a las empresas y a Jaguar Labs de tal manera que el desarrollo del proyecto sea exitoso con todos sus requerimientos.

## **1.10 Particulares.**

- Apoyar como agente *Business Analyst* en el área de preventa para los siguientes proyectos internos asignados:
- Prototipos de aplicaciones móviles para que el cliente pueda observar como funcionaria los módulos.
- investigaciones de nuevas tendencias y mejoras en nuestras aplicaciones.
- Generar los documentos de requerimientos para poder sacar una estimación de tiempo y costo.

## **1.11 Preguntas de investigación**

Las preguntas que principalmente se estarán contestando en esta investigación son las siguientes:

- ¿Qué es un wbs?

- ¿Para qué sirve?
- ¿En qué se beneficia el cliente con el prototipo?

## 1.12 Hipótesis.

Debido a la falta de agentes de *Business Analyst* en el área de preventa, la empresa Jaguar Labs se dio a la tarea de solicitar un agente de apoyo para atender las necesidades de los proyectos. Debido al gran número de proyectos entrantes esto impactando en los tiempos de entregas de los proyectos y los requerimientos que se presentan.

Todo esto conlleva la pérdida de clientes ya que sin una persona encargada los proyectos no están en tiempo y forma. Muchos proyectos y no cumplen las necesidades del clientes ya que como no se establecen bien los requerimientos. Mala coordinación del equipo ya que no se encuentra alguien encargado de llevar un control de los proyectos o simplemente que dicho proyecto tenga un sinfín de errores que no podrán corregir por falta de tiempo.

Por ello Jaguar Labs se dará la tarea de buscar a personas encargadas en *Business Analyst* en el área de preventa podrá tener un mejor manejo de proyectos, mejor comunicación con los clientes, entrega de proyectos en tiempo y forma.

Gracias al trabajo que se desarrolla en el área de preventa se ahorrará mucho trabajo alrededor de un 65% ya que con wbs (*Work Breakdown Structure* en español estructura detallada del trabajo) ya que con esta estructura se podrá sacar el tiempo estimado para el proyecto, los integrantes que el proyecto necesite, el tamaño del proyecto, horas que empleará cada integrante, el flujo de la aplicación y lo más importante el costo del proyecto. [2]

### **1.13 Limitaciones y supuestos.**

Para que se lleve a cabo un buen trabajo en el área de preventa se tiene que tener conocimientos sobre nuevas tendencias tecnológicas y saber todo referente a aplicaciones móviles.

Las investigaciones son de gran ayuda para el área de preventa ya que es bueno saber todo tipo de nuevas tecnologías y que cuando un cliente pida algo referente a las nuevas tecnologías no sepan que es.

El realizar wbs ayuda a tener mejor comunicación con el cliente y tener así todos los requerimientos que el cliente pide y así poder analizarlos con ellos y llevar a cabo una estimación.

### **1.14 Relevancia.**

Al cumplirse esta investigación cualquier persona que tenga acceso a ella podrá ver cómo se lleva a cabo los requerimientos de un proyecto y su prototipo al igual conocer sobre las investigaciones. Se debe destacar que al conocer más sobre lo que se realiza en el área de preventa y los beneficios que este conlleva para un proyecto sea exitoso.

El tener una persona encargada de los requerimientos se puede hacer una estimación realista, unos de los beneficios son que la arquitectura, diseño y el desarrollo del proyecto desarrollará una buena base, como también el realizar revisiones constantemente para ver el progreso del proyecto y tener un registro de las cosas que se llevan para tener una buena planeación y por ultimo lo más importante se puede controlar los requerimiento ya que muchos clientes piden cosas imposibles y pues llevando a cabo los requerimientos le puedes decir si eso está de más.

# Capítulo II

## Antecedentes teóricos

---

En este capítulo se verá las investigaciones sobre el tema, ideas y fuentes.

---

## 2.1 Introducción

Hoy en día, el software es una parte muy importante de los sistemas. Para tener buenos proyectos de software de una forma satisfactoria y construir productos de alta calidad, se necesita personal que sea profesional en software para que entiendan las características y enfoque.

El crecimiento de la informática, ha llevado a realizar software sin llevar una buena planeación así como utilizar una metodología que produzca un software de calidad, a esto se le denomina “crisis de software”.

En este capítulo se verá el software, su historia y la importancia que tiene.

## 2.2 Crisis del software

La crisis del software es el hecho de que el software que se construye no solamente no satisface los requerimientos ni las necesidades pedidas por el cliente, sino que además excede los presupuestos y los horarios de tiempos. La industria del software no ha podido satisfacer la demanda, la complejidad del software producido y demandado se incrementa constantemente, el software es solicitado para ejecutar las tareas demandantes de hoy y está presente en todos los sistemas que van desde los más sencillos hasta los de misión crítica. En estos días, las aplicaciones típicas son muy grandes y complejas para que un individuo las entienda y, por ello, lleva gran tiempo implementar software. [3]

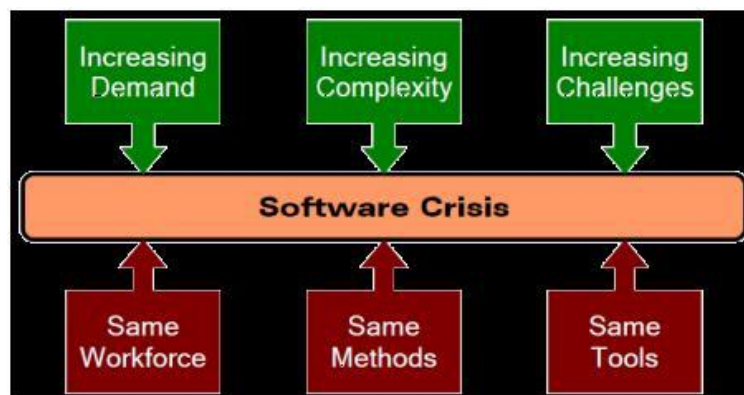


Imagen 2.1 Crisis de software

## Historia

El término "crisis del software" se recalcó en 1968, en la primera conferencia organizada por la OTAN sobre desarrollo de software y con él se mostraron los problemas que surgían en el desarrollo de sistemas de software. [5]

La crisis de software se produjo por la industria del software y los excesivos costos, retrasos en las entregas de proyectos y la insatisfacción del usuario, además por la poca experiencia en software. [25]

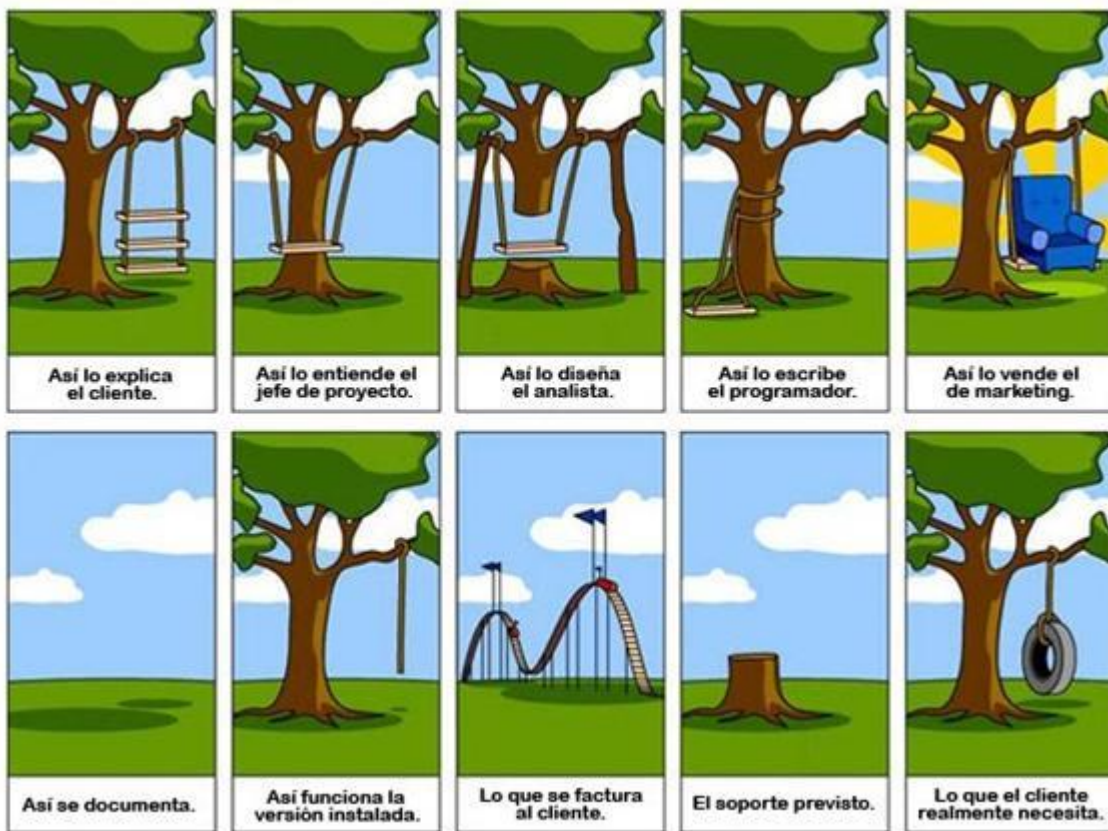


Imagen 2.2 Historia

## Síntomas

Algunos "síntomas" que puede detectarse que el software se encuentra en un periodo de crisis son:

- Baja Calidad del Software.
- Tiempo
- Precios muy altos.
- Confiabilidad.[4]

Hay algunos factores y estos son:

- Reducción del costo
- Que el hardware y software sean obsoletos.
- Que las empresas quieran usar las computadoras y el sistema.
- Incremento en el número de usuarios de los sistemas de software.

Si se habla de un proyecto muy grande puede tener muchos problema como lo es:

- Tiempo costo
- Número de desarrolladores,
- Aumento en el conocimiento del problema.
- Cambios en el entorno. [38]

### **Causas de la crisis del software**

Se encuentran muchas causas que provoca la crisis del software y estas fueron encontradas por lo difícil de entender el software enseguida se mostrarán las causas:

- Proyectos con un presupuesto elevado.
- Proyectos con un tiempo muy tardado.
- Software de baja calidad.
- El software no cumplía con los requerimientos deseados por el cliente.
- Los proyectos fueron complicados y suelen tener un código difícil de mantener.  
[39]



## 2.3 Ingeniería del Software

La Ingeniería del Software es conocida como el área de la Informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. [6]

El ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en el plazo determinado y con el presupuesto determinado [7]

La ingeniería de software, por lo tanto, incluye el análisis de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema. Cabe destacar que el proceso de desarrollo de software implica lo que se conoce como ciclo de vida del software, que está formado por cuatro etapas: concepción, elaboración, construcción y transición. La concepción fija el alcance del proyecto y desarrolla el modelo de negocio; la elaboración define el plan del proyecto, detalla las características y fundamenta la arquitectura; la construcción es el desarrollo del producto; y la transición es la transferencia del producto terminado a los usuarios. [8]

Una vez que se completa este ciclo, se entra en juego el mantenimiento del software. Se trata de una fase de esta ingeniería donde se solucionan los errores descubiertos. El proceso de mantenimiento y se muestra nuevos desarrollos, para permitir que el software pueda cumplir con una mayor cantidad de tareas. Una parte muy importante en la ingeniería de software es la arquitectura de sistemas, que consiste en determinar la estructura general del proyecto, dando a conocer los componentes que serán necesarios para llevar a cabo el proyecto.[9]

Los avances tecnológicos han afectado inevitablemente el proceso de desarrollo de software por diversos motivos, como ser el acceso de los usuarios que no se debe tener acceso a cierta información y es desconocida por completo y que no se pueden comprender, dado que no poseen el conocimiento necesario. Un consumidor bien informado es un consumidor al que muy difícil se le puede robar ya que sabe lo que

necesita y tiene la capacidad de analizar las diferentes ofertas del mercado, comparando las propuestas y prestaciones de los productos [10]

Una frase que se tiene que tener muy en cuenta es “El tiempo es dinero”, y las empresas en el área de informática lo saben muy bien. [13] Además del costo de producción necesario para realizar cada uno de las etapas del programa, la ingeniería de software debe decidir cuáles de ellas tienen sentido, son coherentes con el resto y son necesarias para comunicar claramente la esencia y los objetivos de la aplicación. [14]



Imagen 2.3 Ingeniería de software

## El producto y el proceso de la Ingeniería del Software

### El producto

El software de computadora se ha convertido en una parte muy importante ya que es la máquina que conduce a la toma de decisiones nos sirve para la investigación científica moderna y de resolución de problemas de ingeniería. [11]

El software de computadora es el producto que diseñan y construyen los ingenieros del software. Esto abarca programas que se ejecutan dentro de una computadora de cualquier tamaño, arquitectura, documentos que comprenden formularios y datos que combinan números y texto y también incluyen información de audio, vídeo e imágenes.

Los ingenieros de software lo construyen, para que cualquier persona lo pueda usar de manera directa o indirectamente.

Los pasos son construir software de computadora y otro producto satisfactorio, aplicando un proceso que conduce a un resultado de alta calidad, que satisface las necesidades de las personas que usarán el producto. Debes aplicar un enfoque de ingeniería de software.

Desde el punto de vista de un ingeniero de software, el producto obtenido son los programas, documentos y los datos que configuran el software de computadora. Pero desde el punto de vista de los usuarios el producto obtenido es la información resultante que de algún modo mejora el mundo de los usuarios. [11]

## **El proceso**

El proceso proporciona una interacción entre los usuarios y los diseñadores, entre los usuarios y las herramientas de desarrollo, y entre los diseñadores y las herramientas de desarrollo. Es un proceso donde la herramienta de desarrollo se usa como medio de comunicación, con cada iteración del diálogo se obtiene mayor conocimiento de las personas involucradas. [41]

Cuando se trabaja para construir un producto o un sistema, es importante seguir una serie de pasos predecibles para poder tener un resultado de calidad.

Es importante porque proporciona estabilidad, control y organización a una actividad que puede, si no es controlada puede volverse un caos total. [16]

Los pasos son a un nivel detallado, el proceso que adoptemos depende del software que estamos construyendo.

Desde el punto de vista de un ingeniero de software, los productos obtenidos son programas, documentos y datos que se producen como consecuencia de las actividades definidas por el proceso.

Hay una cantidad de mecanismos de evaluación del proceso de software que permiten a las organizaciones determinar la madurez de su proceso. Sin embargo, la calidad, oportunidad y viabilidad a largo plazo del producto que se está construyendo. [7]

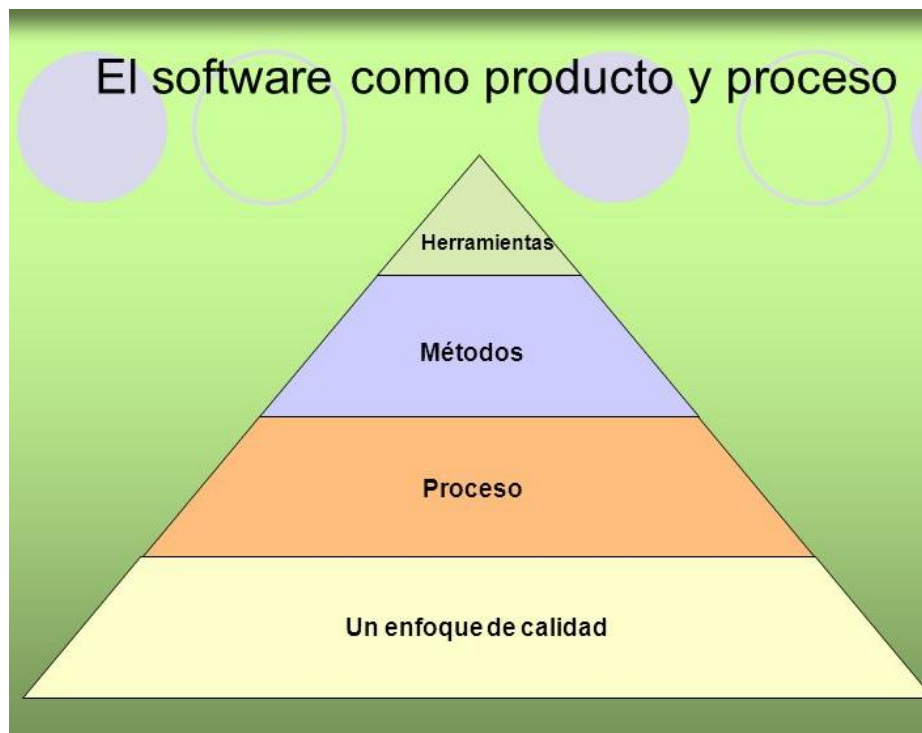


Imagen 2.4 El producto y el proceso de la Ingeniería del Software

## 2.4 Metodologías de desarrollo de software

En el desarrollo de software, una metodología hace cierto énfasis al entorno en el cuál se plantea y estructura el desarrollo de un sistema, existen una gran cantidad de metodologías de la programación que se han utilizado desde los tiempos atrás y que con el paso del tiempo han ido evolucionando. Esto se debe principalmente a que no todos los sistemas de la información, son compatibles con todas las metodologías, pues

el ciclo de vida del software puede ser variable. Por esta razón, es importante que dependiendo del tipo de software que se vaya a desarrollar, se identifique la metodología para el diseño de software y este no sea un . [12]

Una Metodología de desarrollo de software, consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo. Regularmente este tipo de metodología, tienen la necesidad de venir documentadas, para que los programadores que estarán dentro de la planeación del proyecto, comprendan perfectamente la metodología y en algunos casos el ciclo de vida del software que se pretende seguir. [15]

Aunque actualmente existen mucha variedad en metodologías de programación. La realidad es que todas están basadas en ciertos enfoques generales que se crearon hace muchos años, algunos tipos de metodologías de desarrollo de software.

Regularmente, cada metodología de desarrollo de software, tiene un enfoque bien marcado, estos enfoques no son para nada nuevos y se siguen utilizando para la planeación y desarrollo de software aún en nuestros tiempos, así que vamos a ver cuáles son cada uno de ellos y aprenderemos cómo funcionan. [15]

### **Metodología en cascada:**

El modelo de desarrollo de Software en cascada, es una metodología de la programación muy antigua. Su creador nunca lo menciona como metodología en cascada, el funcionamiento y lineamiento de los procesos de la planeación, son exactamente iguales. Básicamente, el estilo del modelo en cascada, es que no se podrá avanzar a la siguiente fase, si la anterior no se encuentra totalmente terminada, pues no tiene por qué haber vuelta atrás. Cuáles son las fases de desarrollo de software del modelo en cascada, para que te puedas dar una idea. [17]

1. **Análisis de Requisitos.** El primer nivel del modelo cascada, es el análisis de requisitos. Lo que se tiene que documentar aquí, son los objetivos del software que se debe hacer al terminar el desarrollo, sin entrar en detalles de la parte

interna. Sin embargo es importante señalar que una vez avanzado el paso de análisis, no puede haber vuelta atrás.

2. **Diseño del Sistema.** Después se elaborará lo que es la estructura del sistema y se determinarán las especificaciones para cada una de las partes del sistema que se planea desarrollar.
3. **Diseño del Programa.** En este punto, aún no ingresamos a lo que es la escritura de código, sin embargo ya se realizan los algoritmos que se van a utilizar en la programación.
4. **Codificación.** Se empieza a escribir todo el código que será necesario para el desarrollo del software.
5. **Ejecución de Pruebas.** Después que se termina la etapa de codificación es el momento de pasar a la fase 5 las pruebas donde se deberá verificar si el sistema realmente funciona correctamente recuerda intentar moverle con el objetivo de dañarlo ya que de ese modo pasará la prueba para que sea usado por el usuario final.
6. **Verificación.** Esta fase consiste en correr el software por la parte del usuario donde el verificará el software.

**Mantenimiento.** Esta es posiblemente una de las fases más tediosas porque con las actualizaciones los usuarios comienzan a comentar que la aplicación está fallando y es necesario darle mantenimiento. [17]

## **Principios básicos del modelo de cascada**

La metodología de cascada es una de las metodologías complejas y lo que busca es que sea documentada perfectamente así si se hubo algún problema al usuario se le haga más fácil encontrarlo. [51]

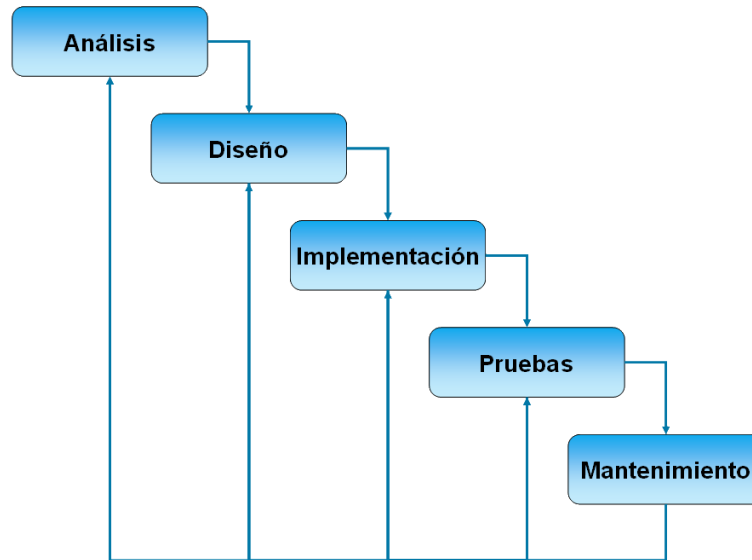


Imagen 2.5 Modelo cascada

## Método de Prototipos

Esta metodología de la programación todavía sigue siendo la favorita de muchos. Consiste que en base a los requerimientos y necesidades que tiene el cliente tenga se realice de forma rápida un prototipo como se verá el trabajo así el programador siga trabajando y llegue al código final. [40]

Mejor ver cuáles son las etapas de desarrollo de software por las cuales tendrás que pasar, en caso de utilizar la metodología de prototipos.

1. **Planeación.** En esta metodología la planeación debe ser muy rápida y debes de tomar tanto tiempo en ella ya que solo será un prototipo.
2. **Modelado.** En esta fase se trata de hacer el modelado pero deberán tener en cuenta que solo es un prototipo entonces deberá de ser rápido.
3. **Elaboración del Prototipo.** Una vez que se pasó la planeación y el modelado se deberá elaborar el prototipo esta fase tomará más tiempo que las anteriores ya que este ya será el prototipo muestra que será entregado al cliente.

4. **Desarrollo.** Posterior a contar con el prototipo elaborado y mostrado al cliente, es momento de comenzar el desarrollo. Este toma una gran cantidad de tiempo, dependiendo del tamaño del proyecto y el lenguaje de programación que se vaya a utilizar.
5. **Entrega y Retroalimentación.** Una vez entregado el proyecto, debemos darle al cliente cierta retroalimentación sobre cómo utilizarlo.
6. **Comunicación con el Cliente.** Es importante que una vez entregado el proyecto, tengamos cierta comunicación con el cliente, básicamente para que nos diga si el proyecto es correcto o si desea agregarle ciertas funciones.
7. **Entrega del Producto Final.** Por último, quedará entregar el sistema elaborado mediante esta metodología. [28]

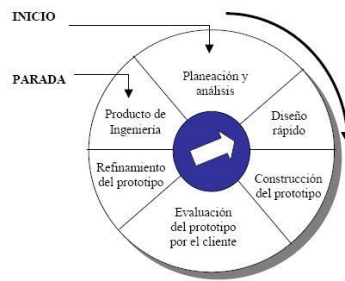


Imagen 2.6 Método prototipo.

## Principios Básicos del método de prototipos

El modelo de prototipos puede llegar a ser un poco más tedioso. Sin embargo uno de sus principios básicos que seguramente habrás notado, es que con el método de prototipos el proyecto se va dividiendo en partes cada vez más pequeñas, para evitar el peligro ante los riesgos. [29]

## Modelo Incremental o Iterativo y Creciente

El modelo Incremental, es una metodología que es de las más utilizadas hoy en día y esto es gracias a que con esta el producto suele ser más completo y un éxito. Se trata especialmente de la combinación de los modelos lineal o bien, modelo de cascada y prototipos. Básicamente consiste en completar varias iteraciones de lo que es el modelo de cascada, pero sin completar ninguna, haciendo iteraciones lo que se hace es



crear una evolución en el producto, permitiendo que se agreguen nuevas especificaciones, funcionalidades, opciones, funciones y lo que el usuario requiera después de cada iteración.

En pocas palabras, el Modelo Incremental repite el modelo de cascada una y otra vez, pero con pequeñas modificaciones o actualizaciones que se le puedan ir agregando al sistema. [18]

### Fases del Modelo Incremental

El modelo iterativo o incremental, cuenta con algunas fases de desarrollo de software que realmente no tienen mucha complejidad, vamos a verlas:

1. **Inicialización.** En esta primera fase se buscan los requisitos del proyecto solamente debemos de tener en cuenta esta metodología se basa en interacciones así que no es necesario poner todos los requisitos.
2. **Periodos de Iteración.** Durante el desarrollo del proyecto, es cuando se da inicio a las iteraciones. La primera iteración se realiza con las características iniciales y al final de la primera iteración, queda un pequeño prototipo de lo que será el proyecto.
3. **Lista de Control.** Es importante que conforme se vaya realizando cada iteración, se vaya llevando un control del mismo en una lista y tienen que estar documentadas. [19]

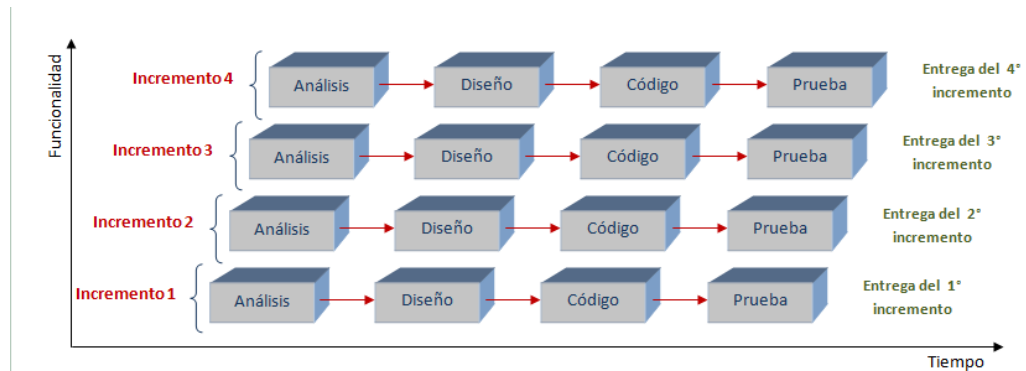


Imagen 2.7 Modelo incremental

## Principios básicos del Modelo Incremental

La idea de un modelo incremental, es utilizar una serie de mini modelos de desarrollo de software en cascada, segmentando requerimientos y permitiendo que el usuario vaya de la mano con el proyecto durante la realización. [10]

## Modelo en Espiral

El modelo de espiral se trata de una combinación entre el modelo lineal o de cascada y el modelo de prototipos a este sistema lo que debemos añadirle es la gestión de riesgos. [31]

Este modelo, consiste en ciertas fases que se van realizando en modo de espiral, utilizando procesos de la misma forma en que se utilizan en el modelo de cascada, sin embargo aquí estos no son obligatorios y no llevan precisamente el orden establecido.[31]

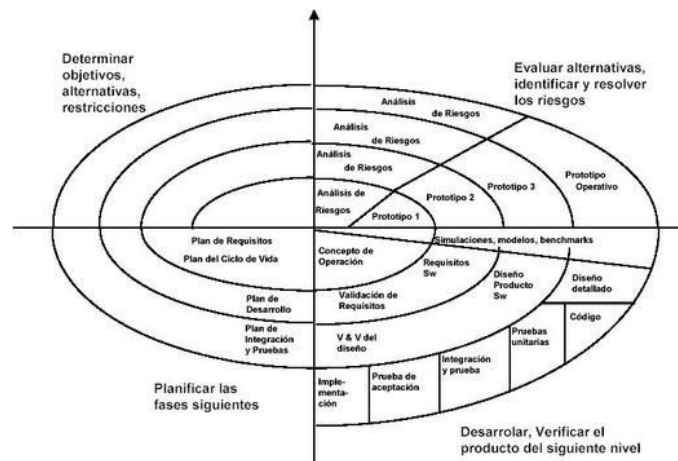


Imagen 2.8 Modelo espiral

- Determinar Objetivo.** En esta fase se deberá realizar una planeación inicial Sin embargo el proceso de determinar objetivos se hará constantemente durante cada iteración que se vaya realizando con el modelo espiral.
- Análisis de Riesgo.** En la fase de análisis de riesgos tienes que tomar en cuenta todo lo que puede dañar tu proyecto, amenazas o daños a tu proyecto así para tener en cuenta cualquier riesgo ya tener como resolverlo.

3. **Desarrollar, Validar y Probar.** Básicamente en esta fase, la forma en que se estará desarrollando el proyecto, dependerá del análisis de riesgos, pues siempre se va a ir desarrollando el proyecto enfocándose en los riesgos que podemos evitar en nuestro software, es decir, si la situación de riesgo más obvia se encuentra en la interfaz del usuario.
4. **Planificación.** Antes de proceder a realizar otra iteración o vuelta al espiral, se debe prestar atención a lo que sucedió en la vuelta anterior. Se debemos analizar detalladamente si los riesgos encontrados ya tuvieron solución para poder seguir con el desarrollo. [33]

### **Principios básicos del modelo en Espiral**

Está claro que el modelo en espiral, es muy distinto a los demás. Encontramos por fuera cuatro fases bien organizadas, las cuáles siempre deben llevar ese orden que se estipula desde el principio. Una determinación de objetivos, un análisis de riesgos, el desarrollo y las pruebas, junto con la planificación, la cual dependerá de los resultados de la iteración para saber cómo se actuará en la siguiente vuelta al espiral. [34]

### **RAD: Desarrollo Rápido de Aplicaciones (*Rapid Application Development*)**

A diferencia de otras metodologías para el desarrollo de software, la metodología RAD o desarrollo rápido de aplicaciones, no cuenta con una serie de fases

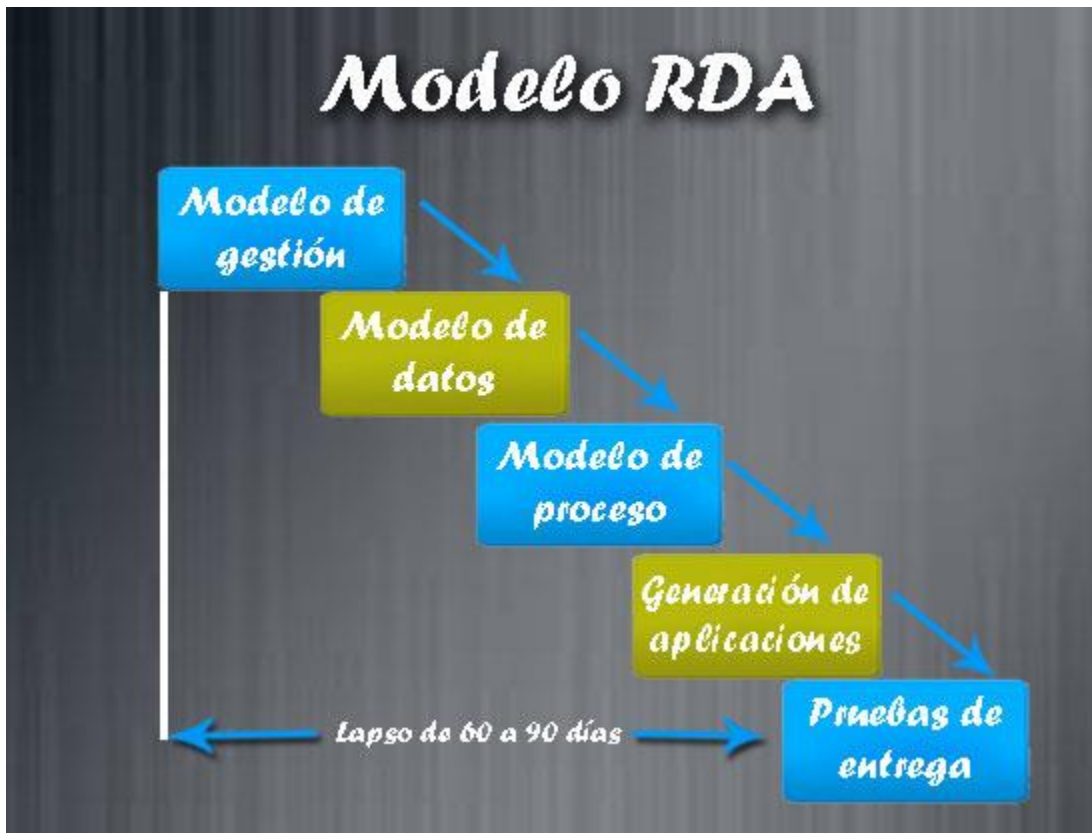


Imagen 2.9 Modelo RDA

### Principios del modelo RAD.

Del mismo modo que modelos anteriores, el Modelo RAD, está basado en el uso de las iteraciones y principalmente en el manejo de prototipos. Sin embargo a diferencia del resto, la metodología RAD hace uso de las Herramientas CASE, las cuales permitirán acelerar el proceso considerablemente. [14]

## 2.5 Metodologías Ágiles

Una metodología ágil, Se encarga principalmente en trabajar con menos documentación a diferencia de las metodologías tradicionales. [21]



Imagen 2.10 Metodología ágil

## Principales Metodologías Ágiles

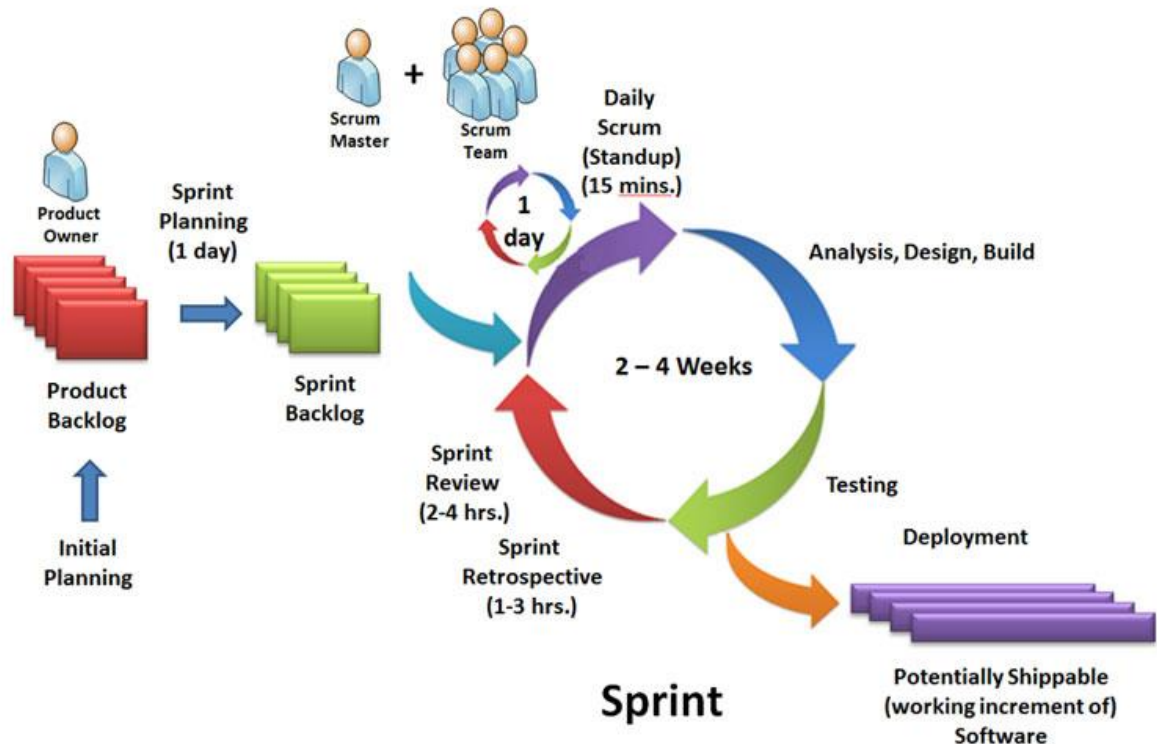
### Metodología *Scrum*

Características de la metodología *Scrum*:

1. **Desarrollo Incremental.** El desarrollo se irá incrementando poco a poco, sin importar el orden en el cual se lleven a cabo los procesos.
2. **Calidad de las personas.** La calidad del proyecto dependerá de las personas, la organización y el conocimiento de los equipos de trabajo.
3. **Adiós al secuencial y cascada.** Esto quiere decir que como anteriormente se vio que qué en las metodologías de cascada y secuencial no podían regresarse esto es todo lo contrario y gracias a esto en la metodología *Scrum* se ahorra tiempo.
4. **La comunicación es fundamental.** Aquí la comunicación es una parte muy importante ya que deberás estar en constante comunicación con el equipo. [43]

## Funcionamiento los procesos *Scrum*

La metodología *Scrum*, es bastante amigable y fomenta lo que es el trabajo en equipo en todo momento, con la finalidad de conseguir los objetivos de una forma rápida. [42]



Copyright © 2011, William B. Heys

Imagen 2.11 Guía *Scrum*

**Product backlog:** El *product backlog* no es más que una lista de las funcionalidades del producto a desarrollar. Este debe ser elaborado por el *Product Owner*. [46]

“¿Qué hay que hacer?”.

1. **Sprint backlog.** Una vez que se finalice el *product backlog*, entonces aparecerá el primer *sprint backlog*. Sin embargo en este punto el *sprint backlog* tiene como requisito marcar el tiempo en que se llevará a cabo el *Sprint*.
2. **Sprint Planning Meeting.** En este proceso del *Scrum*, es una reunión que se realiza para definir plazos y procesos a efectuarse para el proyecto.

3. **Daily Scrum o Stand-up Meeting.** Aquí lo que se hace son reuniones diarias mientras se está llevando a cabo un *Sprint*, para responder las siguientes preguntas: ¿Que hice ayer?, ¿Qué voy a hacer hoy, ¿Qué ayuda necesito? Aquí entra en función el *Scrum Master*. Pero el será el encargado de determinar la solución de los problemas y cada complicación que suceda.
4. **Sprint Review.** El *Sprint Review*, es básicamente una reseña de lo que fue el *Sprint*.
5. **Sprint Retrospective.** Permite al equipo analizar los objetivos cumplidos, si se cometieron errores, visualizarlos y tratar de no cometerlos nuevamente más adelante.
6. **Equipos que Componen los Procesos Scrum.** A continuación se verá cuáles son los equipos que conforman la metodología *Scrum*, cada quien con sus respectivas responsabilidades.
7. **Product Owner.** Si se trata de tener un líder de proyecto. Básicamente son los ojos del cliente, será la persona encargada del proyecto y de checar que se lleve a cabo de tal forma que cumpla las expectativas de lo que se espera.
8. **Scrum Master.** Ahora bien, para cada reunión realizada, siempre debe estar un líder, en este caso el *Scrum Master* será el líder de cada una de las reuniones y ayudará en los problemas que hayan surgido.
9. **Scrum Team.** Básicamente es el núcleo de la metodología *Scrum*, pues es el equipo de desarrollo, encargado de lo que es la codificación del software y de cumplir los objetivos o metas propuestas por el *Product Owner*.
10. **Cliente.** El cliente tiene la capacidad para influir en el proceso, debido a que siempre estará empapado de él, ya sea que proponga nuevas ideas o bien haciendo algún tipo de comentario. [46]

## Metodología Kanban

Siguiendo con las metodologías ágiles, nos encontramos con Kanban. Se trata de una metodología Japonesa, la cual consiste en ir etiquetando con tarjetas cada uno de los procesos que se deben llevar a cabo.

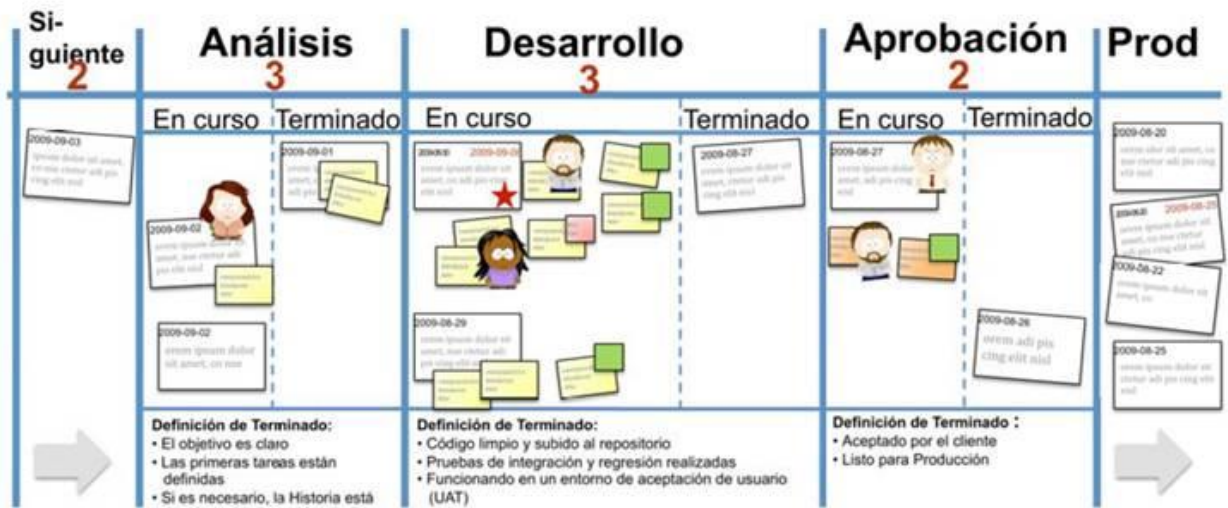


Imagen 2.12 Imagen de Kanban

Una de las principales ventajas de Kanban, es que además de ser una metodología Ágil, también es muy fácil de usar e implementar, sobretodo porque el equipo de trabajo se unirá y empezarán a trabajar a la par en diferentes aspectos del desarrollo.

### Principios Básicos de Kanban

- Garantía de Calidad.
- Desperdicios
- Mejora Continua
- Es Flexible [44]

¿Cómo configurar una estrategia Kanban? [47]

1. **Definir el Flujo de Trabajo.** Una vez que ya tienes el tablero que es requisito para esta metodología, es entonces cuando podrás empezar a seccionarlo dependiendo del número de tareas, fases o proyectos que tengas en puerta.
2. **Fases del Ciclo de Producción.** En este caso es realmente necesario que los procesos sean divididos en pequeños segmentos, para que pueda ser más rápido y no se quede estancado. Es por eso que en los post-it que vayas colocando con cada proceso, deberá ser colocado el número de horas



requeridas para completarlo, al finalizar las horas se determinará por qué no se ha terminado o bien ya se habrá avanzado a otra fase.

3. **Stop Starting, start finishing.** La filosofía de Kanban, trabaja de esta forma y su idea es “No se empieza una nueva tarea, hasta terminar la otra”. Esto se debe principalmente a que la idea es tener un alto porcentaje de tareas completadas y no como ciertos equipos de desarrollo que tienen una gran cantidad de proyectos en puerta y muchas tareas por hacer, la mayoría empezadas, pero ninguna terminada.
4. **Tener un Control.** Algo con lo que trabaja Kanban, es con el control del flujo de trabajo. Si bien la idea es que los trabajadores tengan actividad realmente constante y no se detengan aun cuando terminen sus tareas. [48]

## **Metodología XP**

Si se habla de metodologías de la programación sin mencionar a la Metodología XP, es como no hablar de nada en absoluto. Esta metodología es posiblemente la más destacada de las metodologías ágiles y esto se debe a su gran capacidad de adaptación ante cualquier tipo de imprevisto que surja. Pues la idea no es mantener ciertos requisitos desde que se está elaborando el proyecto, sino que durante el proceso, estos vayan cambiando o vayan evolucionando gradualmente sin complicaciones. [45]



Imagen 2.13 Metodología XP

### Características que componen la metodología XP

1. **Tipo de Desarrollo Iterativo e incremental.** Como hemos visto en lo que llevamos hablando de la metodología XP, el método está basado en lo que son las mejoras continuas, a base de iteraciones y por supuesto un desarrollo incremental al estilo espiral.
2. **Pruebas Unitarias.** Una de las características además son las pruebas unitarias. Se utiliza software de codificación eso sí, dependiendo del lenguaje que estemos usando es la herramienta que nos corresponde, pero de este modo se analiza el código y solucionan errores, antes de validarlo y darlo por bueno.

3. **Trabajo en Equipo.** Más específico todavía, es el trabajo en parejas, el objetivo es que el enfoque en parejas sea mayor, las distracciones son menores y el aprendizaje del uno con el otro permite que el avance del proyecto sea mucho más eficiente que cuando una persona es la encargada.
4. **Alguien del equipo trabaja con el cliente.** Es fundamental que el cliente intervenga en el desarrollo, pero obviamente el no estará en la sala de desarrollo, se debe asignar a una persona que sea le encargada de tener las reuniones con el cliente de forma constante. El será quien comunique al equipo los cambios o el seguimiento del proyecto.
5. **Corrección de Errores.** Algo importante, el hecho de que la metodología XP sea realmente rápida para el desarrollo, no significa que se pasen por alto los errores, de hecho primero se le tiene que dar corrección a los errores antes de seguir avanzando en el proyecto.
6. **Reestructuración del Código.** La idea es clara una re facturación del código siempre se debe realizar. Con esto lo que haremos es simplificar el código pero no las funciones. Pues regularmente cuando desarrollamos, agregamos algunas cosas que pueden ser innecesarias y que no afectan en el funcionamiento del sistema, estas son precisamente las que hay que re facturar.
7. **El Código es de todos.** Realmente aunque se trabaje en equipos, al final todos tendrán la posibilidad de ver el código, proponer cambios o incluso hacerlos. La idea es que si uno no detecta un error, otro lo podrá hacer, por eso el código fuente es compartido entre todos.
8. **Código simple es la clave.** Algo importante con la metodología extrema, es que la simplicidad siempre llevará la ventaja. Principalmente porque cuando se requiera hacer un cambio, si el código fuente es muy complejo, posiblemente lleve muchas horas realizar los cambios e incluso una alternativa seria ya no hacer ningún cambio para no perder tiempo. Esta es precisamente la razón por la cual el código simple, es fundamental en la metodología. [49]

Básicamente, lo que es la simplicidad y la comunicación van de la mano. Puesto que a mayor simplicidad, la comunicación necesaria será menor. Haciendo que la

eficiencia se incremente y la pérdida de tiempo en comunicación sea menor. Por eso es importante seguir al pie de la letra estas dos ideales de la metodología.

## **2.6 Análisis de metodologías clásicas y modernas**

### **¿Qué metodología debo usar?**

Hay dos tipos de metodología las clásicas y modernas, la metodología clásica con esta se pretendían ayudar a los profesionales indicando los ejemplos para realizar y documentar cada una de las tareas indicadas. Sin embargo con el paso del tiempo la mayor parte de las metodologías clásicas dejaron de funcionar y las que aún se ven su funcionalidad es mínima se limita a proyectos muy poco innovadores.

La metodología moderna es una metodología más ágil y más precisa para que todas las personas implicadas las puedan seguir y pero esta metodología debe de ser adaptable para poder usarse en los diferentes proyectos y debe de ser fácil de usar para que ninguna de las personas se les complique pero también tiene que ser muy completa.

Después de describir las diferencias de cada metodología me toca decir que la metodología que usare es *scrum* ya que en jaguar labs se maneja un equipo de desarrollo grande y se trabaja con proyectos grandes y se tiene un mejor manejo ya que se entrega *sprints* cada determinado tiempo, con equipo de trabajo disciplinado y gracias a esta metodología se puede definir por periodos de tiempo dentro del desarrollo, reuniones, roles y su principal funcionalidad es que se centra en ver que es lo que es capaz de hacer el equipo y no en crear documentación donde se tenga que seguir al pie de la letra.

### **Conclusión**

Gracias a este capítulo se puede deducir que no hay metodologías buenas o malas si no que es buscar cual se adapta mejor al proyecto que se realizará y las características que este conlleva, debemos tener en cuenta el proceso, el cliente que en cierto modo tiene su participación en las metodologías.

Una parte importante es que se debe de tener en cuenta quienes serán los implicados y su gestión dentro de la metodología porque con ellas pueden tener errores y poder corregirlo

# Capitulo III

## Resultados

---

**Se mostrará los resultados de la investigación y si trabajos futuros**

---

### 3.1 Introducción

En este capítulo se llevará a cabo la parte principal de la investigación donde se abordará los siguientes temas metodología *scrum*, wbs y prototipos.

Como anteriormente se había visto el problema que conlleva no tener el rol de *Business Analyst* en el área de preventa para resolver la problemática expuesta por el cliente y los requerimientos se debe desarrollar un buen documento donde se establezcan requerimientos para la ayuda de un mejor desarrollo de aplicaciones y sitios web pero para que todo esto termine de manera satisfactoria nos ayudamos con metodología ágil *Scrum* adoptada por la empresa. Se logró realizar un buen análisis en los proyectos de aplicación web y móvil.

Gracias al capítulo anterior fue de mucha ayuda ya que me ayudó a conocer más sobre las metodologías tradicionales y modernas como también los procesos y sobre todo sobre la metodología *scrum* ya que este es el utilizado en Jaguar Labs.

### 3.2 Metodología

*Scrum* es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto.

En *Scrum* se realizan entregas parciales y regulares del producto final. Por ello, *Scrum* está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, la competitividad, la flexibilidad.

*Scrum* también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere

trabajar utilizando un proceso especializado en el desarrollo de producto. Ver en detalle cuales son los beneficios de *Scrum*, sus fundamentos y sus requisitos.

### 3.3 Scrum y el análisis de problemas de desarrollo

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas.

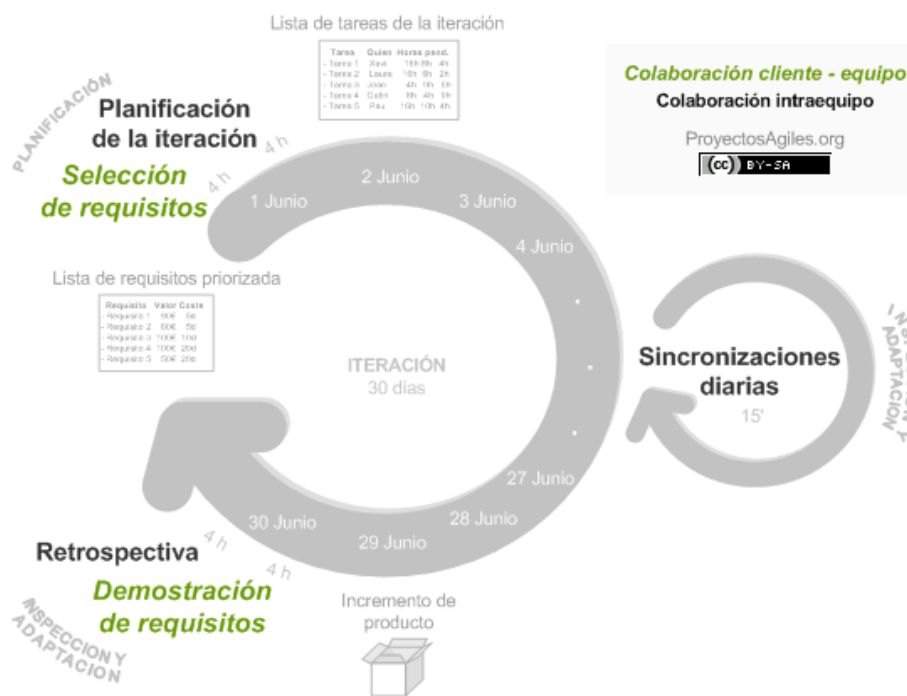


Imagen 3.1 Metodología Scrum

En la figura 3.1 se explica cómo se realiza el proceso *Scrum* ya que en un proyecto se ejecuta en bloques temporales cortos y fijos. Cada iteración tiene que proporcionar un resultado completo.

Las actividades que se llevan a cabo en *Scrum* son las siguientes:



## **Planificación de la iteración**

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

**Selección de requisitos** (4 horas máximo). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios.

**Planificación de la iteración** (4 horas máximo). El equipo elabora la iteración necesaria para desarrollar los requisitos a que se ha comprometido.

Cada día el equipo realiza una reunión de sincronización (15 minutos máximos). Cada miembro del equipo inspecciona el trabajo que el resto está realizando. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el *Scrum* Master se encarga de que el equipo pueda cumplir con su compromiso y la productividad.

## **Inspección y adaptación**

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

**Demostración (4 horas máximo)**. El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo.

**Retrospectiva** (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad.

## **Procedimiento y descripción de las actividades realizadas**

Principalmente se realiza una reuniones con el cliente así explica que es lo que quiere y con todas las especificaciones estas reuniones son de alrededor de media hora dependiendo y son grabadas al igual el cliente nos proporciona un documento con las especificaciones más detalladas por si algo faltó en la reunión una vez teniendo eso se realiza el wbs con todos los módulos y requerimientos pedidos por el cliente esta parte es una nueva implementación ya que en las metodologías vistas anteriormente no se hacen y estas reuniones Jaguar Labs las hace para tener una mejor comunicación con el cliente y así que todos los requerimientos estén bien estructurados y el cliente esté conforme con ellos.

Se realiza la estimación y se realiza una segunda reunión con el cliente para que se llegue a un acuerdo se le mostrará un documento formal mostrándole todos los requisitos, módulos, costo y personal que estará encargado, una vez que se aceptó el acuerdo el wbs pasa al *Business Analyst* para que prosiga con el proceso.

### **Wbs ( *Work breakdown Structure* )**

En esta fase el *Business Analyst* en el área de preventa tiene como papel principal realizar un documento (wbs) que es una herramienta de apoyo para los *project manager*, en el que se desglosan todos los módulos del trabajo de los proyectos en entregables más pequeños y más fáciles de trabajar representado en un documento de *Excel*.

Los pasos que realiza el *Business Analyst* en el área de preventa son los siguientes:

#### 1. Pasos a seguir para la creación de un wbs.

\*Identificar los entregables

- ¿Qué se va a hacer?

- ¿Cuál es el objetivo de hacer el wbs?

## 2. Organizar el wbs

\* Definir la estructura del wbs.

- Debe estar organizado de forma que facilite y no dificulte la administración y seguimiento.
- Los elementos de cada nivel del wbs deben ser del mismo "tipo".

## 3. Descomponer los módulos en módulos más pequeños y actividades

- Cada entregable puede estar compuesto por uno o varios modulo más pequeños entregables pequeños son más fáciles de administrar y dar seguimiento.
- El nivel más inferior del wbs deben ser actividades que deben ayudar a contestar mínimamente ¿Cuánto esfuerzo?
- 4. Asignar un identificador único a cada entregable
- Debemos ser capaces de rastrear cada entregable del wbs.
- Cada entregable es único y por lo tanto debe tener un identificador único.

## 5. Verificar el wbs

- Asegurarnos de que estamos cumpliendo el objetivo. No sobra ni falta nada.

### **Roles involucrados**

Esta fase consiste en realizar un análisis a partir del documento wbs se identifican los roles para cada integrante a realizar en el transcurso del desarrollo de los proyectos.

- BC: Ver las necesidades en proyectos, módulos, requerimientos.

- BA: Descomponer los módulos y sus requerimientos para organizar el trabajo de análisis, resolución de dudas o cambios expuestos por el cliente.
- DEV: Identificar todos los requerimientos, pantallas, tablas, etc.) que deben considerarse para dar un fechas/horas más acertadas.
- TESTING: Identificar los módulos, requerimientos, casos de prueba y ejecuciones a realizar.

## Proyectos

En base al wbs se realiza un análisis identificando lo siguiente:

- Identificar proyectos y sub-proyectos
- Identificar fases y *sprints*
- Identificar módulos y requerimientos
- Identificar tareas y actividades
- Aislar un problema e intentar solucionarlo "por partes"

Se inicia realizando el wbs que consta en mostrar un desglose de los requerimientos adquiridos por el cliente respetando las funcionalidades deseadas y establecidos , el wbs nos servirá para establecer costos del proyecto y tiempos establecidos, así como también el equipo de trabajo que estará involucrado en el desarrollo del proyecto y de base para establecer el alcance del proyecto ; en base al wbs se realiza un alcance ya que anteriormente fue establecido todos los requerimientos y sus funcionalidades, cabe mencionar que el alcance es un documento donde se le envía al cliente con una propuesta de funcionalidad y costos así como tiempo de desarrollo. Una vez establecido un acuerdo con el cliente se realiza una junta de planeación para dar a conocer el equipo de trabajo y el *Scrum* master que tomara el papel de líder del proyecto.

## **Anticipación de errores y advertencias**

El *Business Analyst* junto al equipo de desarrollo manejan la principal herramienta adoptada por Jaguar Labs que consiste en evitar el desconocimiento de los procesos del desarrollo del sistema son los llamados *Scrum* diarios que son de gran ayuda ya que se obtiene un status diario del desarrollo de los proyectos y para evitar las tendencia a cometer algún error.

## **Alcance del proyecto**

Se elabora a partir de los WBS realizado por el *Business Analyst* y de supuestos y posibles restricciones al proyecto que se han documentado en la fase de Iniciación siendo en la fase de planificación donde el alcance del proyecto se describe y se define de manera más específica, según se va obteniendo más información sobre del proyecto.

## **Documento de requerimientos**

La realización del documento de requerimientos es elaborado por el *Business Analyst* es el papel más importante en el desarrollo de los diferentes proyectos. Debe incluir tanto los requerimientos a nivel de usuario para el sistema como una especificación detallada de los requerimientos informáticos, siendo muy claros en las partes más críticas.

## **Estructura de un documento de requerimientos**

- Fecha: <Fecha en que se hace el Documento>
- Versión: <Número de Versión del Documento>
- Responsables: <Nombre de las Personas que realizan este documento>
- Índice
- Sistema a construir:

<Descripción del Sistema a construir, de sus principales funcionalidades y de los posibles usos del producto>

➤ Usuarios:

<Los principales usuarios del Sistema a Construir>

➤ Requerimientos funcionales:

<Se escriben todos los Requerimientos Funcionales del Sistema>

<Funcionalidad 1>:

<Explicación de la Funcionalidad 1 del Sistema>

➤ Requerimientos no funcionales:

<Se trata de requerimientos necesarios para el sistema.>

<Requerimiento no Funcional 1>:

<Explicación del Requerimiento no Funcional 1 del Sistema>

➤ Restricciones:

< Sobre el producto, ejemplos son proceso de software a seguir, fechas de entrega, hitos de entrega fijos, etc. >

<Restricción 1>

<Explicación de la Restricción >

➤ Interfaces:

<Se detallan las interfaces que debe proveer o utilizar la aplicación>

Interfaces de Usuario:

<Describe las interfaces de usuario que deben ser implementadas>

➤ Interfaces con Hardware:

<Incluye cualquier interfaz con el Hardware >

➤ Interfaces con Software:

<Describe las interfaces del sistema con cualquier otro sistema de software, pueden ser reusados>

### **Retrospectiva de requerimientos**

Se realizan en cada junta de planeación de *SPRINT* en la cual se evalúan las experiencias a través de la documentación de requerimientos.

Las preguntas que se realizan en la retrospectiva son:

- ¿Qué debemos de continuar haciendo?
- ¿Qué debemos de empezar a hacer?
- ¿Qué debemos dejar de hacer? Oportunidades de mejoras (procesos, plantillas, métodos de comunicación, responsabilidades)
- ¿Quedaron tareas pendientes?
- ¿Tienen bugs pendientes?

La retrospectiva nos ayuda a tener un *status* del proyecto al final de cada *Sprint*, para ayudarnos a mejorar en el siguiente *Sprint* el encargado de realizar la retrospectiva es el *Scrum Master* el cual evaluara el estatus del proyecto.

### **Showcase (Final de *sprint*)**

Esta fase del proyecto es intencionalmente una reunión muy informal que por lo general toma la forma de una demostración de las características completados al dueño del producto y asiste todo el equipo de trabajo donde están presentes para resolver dudas inmediatas o en el caso del *Business Analyst* decirle los requerimientos que se están realizando así como los cambios realizados por funcionalidad o los cambios solicitados por el mismo.

No realizar todos los objetivos en el sprint puede impactar en el tiempo de desarrollo del proyecto.

## Revisión del proyecto

Cuando se termina un proyecto se realiza un *showcase* final el cual asiste todo el equipo de trabajo y el cliente donde se da a conocer un documento llamado detalle de requerimientos realizado por el *Business Analyst* el cual contiene todos los requerimientos solicitados por el cliente y que se encuentran implementados, dándole a conocer al cliente que sus requerimientos solicitados están al 100% y no se encuentra con fallas y que la empresa cumple en tiempo y forma con lo acordado, una vez que se da a conocer el estatus final del proyecto se publica en tiendas ( app store, play store o se instalan en sus respectivos servidores ) y si no presenta ninguna falla en el periodo de pruebas se da a conocer el fin del proyecto.

## Prototipo

La realización del prototipo lleva su tiempo y este es realizado únicamente cuando el cliente lo pide y este lo realiza el *Business Analyst* en el área de preventa una vez que se realizó el wbs el *Business Analyst* en el área de preventa basándose en el wbs empieza con la realización del prototipo y ya terminado se le muestra al cliente y con esto el podrá ver que le gusta, que no, que quiere cambiar, que no y que el cliente haga sus observaciones sobre el prototipo y el *Business Analyst* en el área de preventa encargado tiene que realizar las correcciones necesarias para que una vez que sea aprobado por el cliente el prototipo pase al diseñador encargado para que este se base en el prototipo para la realización de las pantallas.

## Ventajas:

- **Mejora la velocidad de desarrollo:** Da mucho menos trabajo ya que el diseñador ya tiene una idea de lo que llevará y donde ya que es solo para que el cliente se dé una idea como estará estructurada.
- **Involucra al cliente:** La aprobación de un prototipo la tiene que dar el cliente. Como es mucho más cómodo hacer cambios sobre un esquema con poco detalle que sobre un diseño ya acabado, es fácil involucrar al cliente en esta fase de diseño.



### 3.4 Desarrollo

Al iniciar el desarrollo de la aplicación comenzamos con el levantamiento de requerimientos donde inicia el trabajo del *Business Analyst* en una junta en el cual el cliente expuso sus problemáticas para resolver y sus objetivos a lograr. Se inició una sesión de preguntas en la cual se plantearon estrategias para atender sus necesidades y realizar las planeaciones para cada impacto del proyecto. Una vez establecidos los objetivos del alcance del proyecto se realizó un wbs en el cual muestra un desglose de los requerimientos adquiridos por el cliente respetando las funcionalidades deseadas y ámbitos establecidos.

El Wbs nos servirá para establecer costos del proyecto y tiempos establecidos para cada sprint así como también el equipo de trabajo que estará involucrado en el desarrollo del proyecto y de base para establecer el alcance del proyecto; con base al bs se realiza un alcance ya que anteriormente fue establecido todos los requerimientos y sus funcionalidades. Este alcance para desarrollar el documento de requerimientos, cabe mencionar que el alcance es un documento donde se le envía al cliente con una propuesta de funcionalidad y costos así como tiempo de desarrollo. Una vez establecido un acuerdo con el cliente se realiza una junta de planeación para dar a conocer el equipo de trabajo y el *Scrum* master que tomara el papel de líder del proyecto y realizara *showcase* con el cliente para darle un estatus del proyecto en determinados tiempos establecidos llamados *sprints*.

El *Business Analyst* tiene el papel de desarrollar el documento de requerimientos que posteriormente se le otorga a cada integrante de equipo para realizar el desarrollo , la construcción del documento de requerimientos es un desglose de todos los módulos establecidos y sus componentes que contendrán , especificando de una manera muy detallada todas las funcionalidades y conteniendo *mockups* de pantallas que servirán de base para los desarrolladores además el documento de requerimientos tiene un papel fundamental ya que servirá de seguro para mostrarle al cliente en un futuro sus requerimientos establecidos y poder prevenir problemas de cambios o desacuerdos con el cliente.

El documento de requerimientos estará presente en todo el tiempo de desarrollo ya que es la guía fundamental para el desarrollador y siempre tiene que estar actualizado para cualquier duda , cambio o tecnología , para estar en constante comunicación con el equipo el *Scrum* master organiza un *Scrum* diario donde se plantean todas las dudas , comentarios , actualizaciones y avances por parte del equipo en el proyecto de esta manera el *Scrum* master junto al BA tienen un estatus actualizado del proyecto y así organizar un *showcase* al final del sprint donde se le da a conocer al cliente el estatus del proyecto y los avances o problemáticas así como necesidades para desarrollo si es que las contiene el proyecto . A final de cada sprint se realiza una junta de planeación en el cual el equipo establece horas de desarrollo dándolo a conocer al *Scrum* master tiempos estimados de desarrollo y si se realizara en tiempo y forma; el *Scrum* master realiza una retrospectiva para ver si hay problemas y cómo podemos mejorarlas al terminar.

Al terminar con todos los *sprints* establecidos se realiza un *showcase* final el cual se le da a conocer al cliente el proyecto terminado en esa junta de *showcase* se encuentra todo el equipo de trabajo y especialmente el *Business Analyst* le expone los requerimientos establecidos por el y se realiza una comparación junto el sistema o aplicación que se realizó , cuando el cliente da la aprobación final pasa a la última fase la cual se le conoce como “Tiendas” esta fase es cuando los desarrolladores suben las aplicaciones a su respectiva tienda ya sea Play store o App store y los usuarios pueden descargarlas en sus dispositivos , en caso de que no se a una aplicación para un Smartphone , los desarrolladores instalan en los servidores del cliente el sistema realizado entregando todos los recursos que se necesitan para su funcionamiento correcto y documentación realizada por el *Business Analyst* ya que este es de suma importancia para el cliente .

El papel del *Business Analyst* en el desarrollo del proyecto es crucial ya que es el encargado de saber todo el proyecto y poder resolver dudas de parte de los desarrolladores si se presentan y establecer cambios en el documento si el cliente

desea alguno y estar en constante comunicación con el cliente y *Scrum master* ya que es su mano derecha.

### 3.5 Resultados

Al realizar el papel de *Business Analysts* en el área de preventas nos dio como resultado en la aplicación la creación de un documento de requerimientos viable que cumple con todos los requerimientos que solicito el cliente o *product owner*. El cual la aplicación consiste en ser una herramienta robusta pero amigable que lleva el control de sus clientes, pedidos, deudas un panel general para saber cuánto ganan y cuanto les deben todo al alcance con un esfuerzo mínimo. La aceptación de la aplicación fue de gran impacto a la empresa ya que con su uso se pudo lograr la eficiencia y productividad de los asociados obteniendo mayores ganancias a la empresa.

Esto se logró gracias a la correcta implementación de los requerimientos y sobretodo la buena adaptación de los requerimientos para que el equipo de trabajo de diera el mejor seguimiento.

	B	C	D	E	F	G	H
1	Oe Loyalty						
2	#	Module		Requirement			
3							
4	001	Splash Page		Mostrar Logotipo de la empresa.			
5				Al iniciar la aplicación se mostrará la pantalla de Splash Page.			
6							
7	002	Login		Ingresar a la aplicación.			
8				Para que los usuarios ya registrados en el programa de Lealtad puedan ingresar a la aplicación se les solicitara los siguientes datos:			
				Número de Tarjeta: Campo de texto donde el usuario deberá de ingresar su número de tarjeta.			

### Imagen 3.2 Estructura wbs

En esta imagen se muestra una parte de lo que se conforma un wbs en la primera casilla le corresponde el número de módulo, enseguida le sigue el nombre del módulo para después los requerimientos donde se especifica que contendrá el módulo este debe de llevar un orden de importancia.

ckingbot.com/workspace#apps/pF92584ABC11472231368237

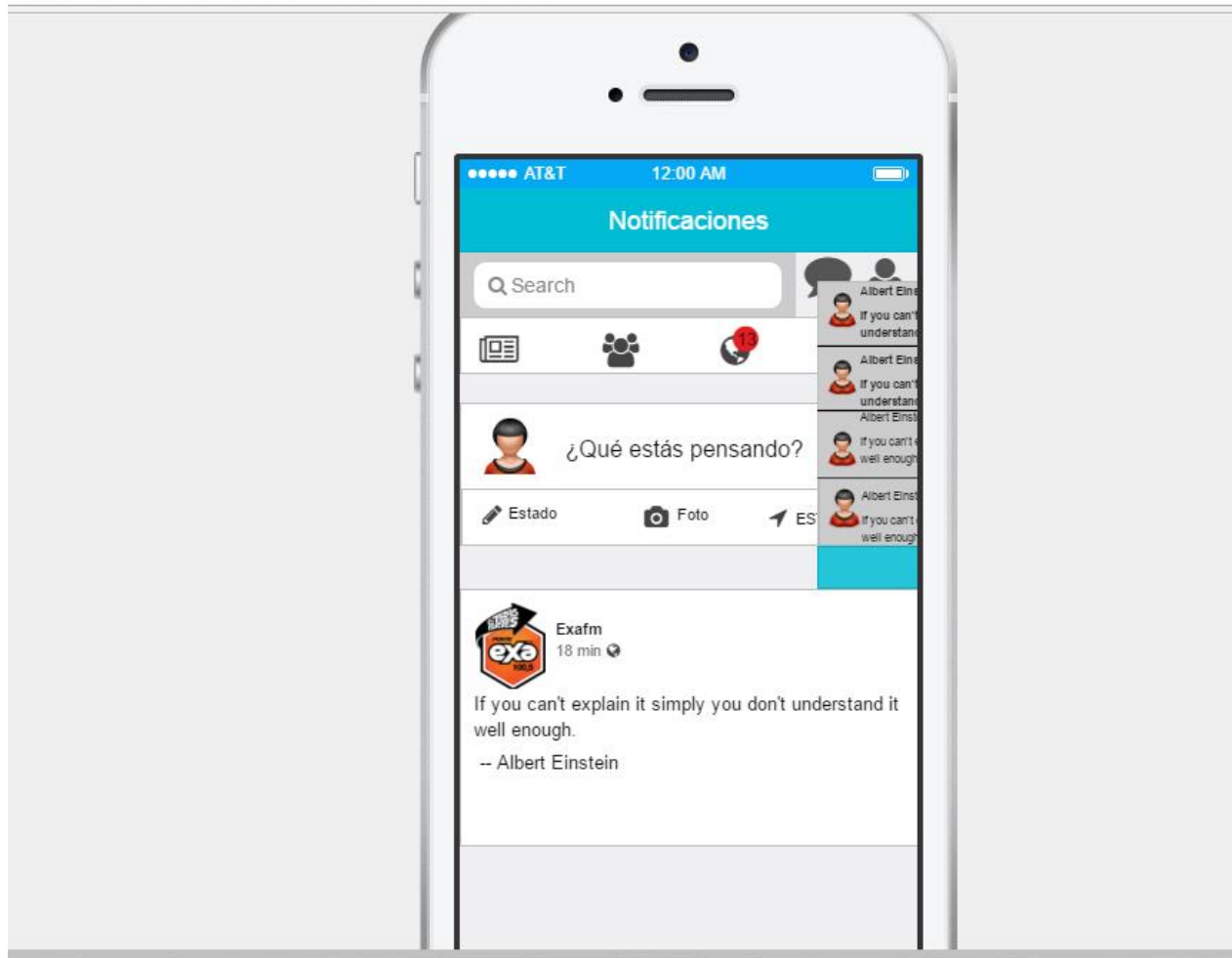


Imagen 3.3 Prototipo

Aquí se muestra un ejemplo de un prototipo que es muy parecido a Facebook se muestra las notificaciones, mensajes y publicaciones el usuario podrá ver el flujo que tendría su aplicación.

### 3.6 Conclusiones y Recomendaciones

Al realizar este proyecto se ha logrado cumplir con todos los objetivos específicos plasmados en este proyecto y esto se logró gracias a la metodología de *Scrum* y un buen planteamiento de requerimientos.

Al tener un puesto de *Business Analyst* en preventas se ha demostrado una mejoría en los proyectos con un mínimo margen de error en estos, con una entrega a tiempo y teniendo a los clientes satisfechos por el trabajo realizado, el trabajo aumentó en un 80% en los trabajos y la rapidez de estos.

En los meses que estuve trabajando en la empresa Jaguar Labs de *Business Analyst* en el área de preventa me di cuenta como los clientes llegan sin tener las ideas claras del proyecto que quieren o simplemente tienen expectativas muy altas y ahí es donde nosotros entramos para decirle que es lo mejor y darles algunas ideas y ayudarles darle una estimación de tiempo y costo.

Fue un poco difícil el estar de este lado ya que muchos clientes no entendían o pensaban que en verdad nosotros no sabíamos hacer algo pero conforme se iba pasando la reunión se iban conformando con nuestro trabajo y gracias a este análisis con wbs se puede agilizar el proceso, tener un mejor trabajo y una mejor organización con esto.

A lo largo de la estadía que se realizó en en Jaguar Labs se entendieron algunos conceptos claves para el puesto de *Business Analyst* en el área de preventa como lo es wbs y para qué sirve y se entendió que wbs es una parte muy importante en el proceso ya que con este se deriva muchas cosas como la estimación de costo y tiempo y los requerimientos bien detallados, wbs es una estructura organizada donde Jaguar Labs la usa para definir módulos, requerimientos, costo y tiempo del proyecto.

Otra cosa que es muy importante que Jaguar Labs realiza fue que usan prototipos ya que esto es una parte que beneficia al cliente porque así queda claro cómo será el flujo

de la aplicación y con eso se beneficia también los programadores y diseñadores porque ya tienen una idea clara de lo que quiere el cliente.

Este trabajo de investigación fue un gran avance para la implantación de los procesos que utilizan en Jaguar Labs, con este proyecto me han permitido contribuir con la realización de un buen análisis y un buen servicio para los clientes cumpliendo sus objetivos.

## Glosario

Aplicaciones móviles.- Es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles y que permite al usuario efectuar una tarea concreta de cualquier tipo —profesional, de ocio, educativas, de acceso a servicios, etc

Wbs.- Una estructura de descomposición del trabajo (EDT), también conocida por su nombre en inglés *Work Breakdown Structure* o WBS, es una herramienta fundamental que consiste en la descomposición jerárquica, orientada al entregable, del trabajo a ser ejecutado por el equipo de proyecto, para cumplir con los objetivos de éste y crear los entregables requeridos, donde cada nivel descendente de la EDT representa una definición con un detalle incrementado del trabajo del proyecto.

Software.- Se conoce software equipo lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

Desarrollo ágil de software .- El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

Scrum.- Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto.

Prototipo.- Los prototipos son una representación limitada de un producto, permite a las partes probarlo en situaciones reales o explorar su uso, creando así un proceso de diseño de iteración que genera calidad.

Business Analyst.- Es la persona que posee conocimientos técnicos sobre la construcción de sistemas informáticos y al mismo tiempo comprende y está al corriente

de las necesidades del usuario que requiere de esos sistemas para realizar su trabajo. Su misión es la de ser el interlocutor entre el usuario y el departamento de sistemas.

Product owner.- Ser el representante de todas las personas interesadas en los resultados del proyecto y actúa como interlocutor único ante el equipo, con autoridad para tomar decisiones.



## Referencias

- [1] C. J. P. Escobar, «blogspot,» 10 Octubre 2013. [En línea]. Available: <http://asprotech.blogspot.mx/2013/10/que-es-cmmi.html>. [Último acceso: 30 Septiembre 2016].
- [2] F. d. Rojas, «spanishpmo.com,» 16 Febrero 2011. [En línea]. Available: <http://spanishpmo.com/index.php/que-es-work-breakdown-structure-wbs/>. [Último acceso: 30 Septiembre 2016].
- [3] albemi2, «histinf.blog,» 04 Enero 2011. [En línea]. Available: <https://histinf.blogs.upv.es/2011/01/04/la-crisis-del-software/>. [Último acceso: 17 Octubre 2016].
- [4] C. MALDONADO, «blogspot,» 04 Enero 2011. [En línea]. Available: <http://cesarmaldonadotovar.blogspot.mx/>. [Último acceso: 2016 Octubre 04].
- [5] albemi2, «blogs.upv.,» 04 Enero 2011. [En línea]. Available: <https://histinf.blogs.upv.es/2011/01/04/la-crisis-del-software/>. [Último acceso: 05 Octubre 2016].
- [6] R. C. E. ISRAEL, «wordpress.com,» 23 Febrero 2010. [En línea]. Available: <https://blogadmi2.files.wordpress.com/2010/02/tarea1rcei1.pdf>. [Último acceso: 05 Octubre 2016].
- [7] R. C. E. ISRAEL, «wordpress.com/2010/02/tarea1rcei1.pdf,» 23 Febrero 2010. [En línea]. Available: <https://blogadmi2.files.wordpress.com/2010/02/tarea1rcei1.pdf>. [Último acceso: 05 Octubre 2016]

- [8] C. MALDONADO, «blogspot,» 04 Enero 2011. [En línea]. Available: <http://cesarmaldonadotovar.blogspot.mx/>. [Último acceso: 2016 Octubre 04].
- [9] J. P. P. y. A. G. Publicado, «definicion.de,» 12 Diciembre 2009. [En línea]. Available: <http://definicion.de/ingenieria-de-software/>. [Último acceso: 06 Octubre 2016].
- [10] ecured@idict.cu, «ecured.cu,» 21 Enero 2012. [En línea]. Available: [https://www.ecured.cu/Ingenier%C3%ADa\\_de\\_software](https://www.ecured.cu/Ingenier%C3%ADa_de_software). [Último acceso: 06 Octubre 2016].
- [11] U. d. andes, «codecompiling.net,» Octubre 2011. [En línea]. Available: [http://www.codecompiling.net/files/slides/IS\\_clase\\_13\\_metodos\\_y\\_procesos.pdf](http://www.codecompiling.net/files/slides/IS_clase_13_metodos_y_procesos.pdf). [Último acceso: 06 Octubre 2016].
- [12] H. Romero, «slideshare.net,» 07 Febrero 2012. [En línea]. Available: <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>. [Último acceso: 06 Octubre 2016].
- [13] J. P. P. y. A. G. Publicado, «definicion.de,» 12 Diciembre 2009. [En línea]. Available: <http://definicion.de/ingenieria-de-software/>. [Último acceso: 06 Octubre 2016].
- [14] ecured@idict.cu, «ecured.cu,» 21 Enero 2012. [En línea]. Available: [https://www.ecured.cu/Ingenier%C3%ADa\\_de\\_software](https://www.ecured.cu/Ingenier%C3%ADa_de_software). [Último acceso: 06 Octubre 2016].
- [15] H. Romero, «slideshare.net,» 07 Diciembre 2011. [En línea]. Available: <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>. [Último acceso:

06 Octubre 2016].

D. Taboas, «Proceso del Desarrollo de Software,» 30 Abril 2015. [En línea]. Available: <http://blog.nubelo.com/proceso-del-desarrollo-software/>. [Último acceso: 15 Octubre 2016].

[16]

J. E. Uchc, «Metodologías clásicas,» 11 Abril 2014. [En línea]. Available: [https://prezi.com/q\\_myjbslsn9y/21-metodologias-clasicas/](https://prezi.com/q_myjbslsn9y/21-metodologias-clasicas/). [Último acceso: 17 Octubre 2016].

[17]

B. Cristian, «Modelo incremental,» 8 Octubre 2010. [En línea]. Available: <https://procesossoftware.wikispaces.com/Modelo+Incremental>. [Último acceso: 16 Octubre 2016].

[18]

[19] M. J. Torres, «de Metodologías Modernas para el desarrollo de Software,» 30 Septiembre 2014. [En línea]. Available:

[20] <https://prezi.com/0mheugelcrjl/metodologias-modernas-para-el-desarrollo-de-software/>. [Último acceso: 16 Octubre 2016].

[21]

C. P. M. Rolando, «Metodología RAD,» 2 Septiembre 2011. [En línea]. Available: <http://metodologiarad.weebly.com/>. [Último acceso: 16 Octubre 2016].

[22] J. garzas, «Metodologías ágiles,» 25 Marzo 2013. [En línea]. Available: <http://www.javiergarzas.com/metodologias-agiles>. [Último acceso: 17 Octubre 2016]

[ 23] P. Ietelie, «Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP),» 15 Enero 2006. [En línea]. Available: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. [Último acceso: 15 Octubre 2016].

[24] U. p. d. valencia, «Metodologías Ágiles ene el desarrollo de software,» 12 Noviembre 2003. [En línea]. Available: <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>. [Último acceso: 17 Octubre 2016].

[25] C. J. P. Escobar, «blogspot,» 10 Octubre 2013. [En línea]. Available: <http://asprotech.blogspot.mx/2013/10/que-es-cmmi.html>. [Último acceso: 30 Septiembre 2016].

[26] F. d. Rojas, «spanishpmo.com,» 16 Febrero 2011. [En línea]. Available: <http://spanishpmo.com/index.php/que-es-work-breakdown-structure-wbs/>. [Último acceso: 30 Septiembre 2016].

[27] albemi2, «blogs.upv.,» 04 Enero 2011. [En línea]. Available: <https://histinf.blogs.upv.es/2011/01/04/la-crisis-del-software/>. [Último acceso: 05 Octubre 2016].

- [28] C. MALDONADO, «blogspot,» 04 Enero 2011. [En línea]. Available: <http://cesarmaldonadotovar.blogspot.mx/>. [Último acceso: 2016 Octubre 04].
- [29] C. PARDO, F. GARCÍA, F. PINO y M. PIATTINI, «redalyc.org,» Diciembre 2013. [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=47829722010>. [Último acceso: 15 Octubre 2016]
- [30] J. P. P. y. A. G. Publicado, «definicion.de,» 12 Diciembre 2009. [En línea]. Available: <http://definicion.de/ingenieria-de-software/>. [Último acceso: 06 Octubre 2016].
- [31] ecured@idict.cu, «ecured.cu,» 21 Enero 2012. [En línea]. Available: [https://www.ecured.cu/Ingenier%C3%ADa\\_de\\_software](https://www.ecured.cu/Ingenier%C3%ADa_de_software). [Último acceso: 06 Octubre 2016].
- [32] D. Gutierrez, «wikispaces.com,» Julio 2011. [En línea]. Available: <https://procesosdesoftware.wikispaces.com/file/view/ciclosdevidadelsoftware.pdf/579330701/ciclosdevidadelsoftware.pdf>. [Último acceso: 06 Octubre 2016].
- [33] U. d. andes, «codecompiling.net,» Octubre 2011. [En línea]. Available: [http://www.codecompiling.net/files/slides/IS\\_clase\\_13\\_metodos\\_y\\_procesos.pdf](http://www.codecompiling.net/files/slides/IS_clase_13_metodos_y_procesos.pdf). [Último acceso: 06 Octubre 2016].

- [34] H. Romero, «slideshare.net,» 07 Diciembre 2011. [En línea]. Available: <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>. [Último acceso: 06 Octubre 2016].
- [35] albemi2, «histinf.blog,» 04 Enero 2011. [En línea]. Available: <https://histinf.blogs.upv.es/2011/01/04/la-crisis-del-software/>. [Último acceso: 17 Octubre 2016].
- [36] H. T. ARIANNA, «blogadmi23.files.wordpress.com,» Agosto 2009. [En línea]. Available: <https://blogadmi23.files.wordpress.com/2009/08/tarea-1-crisis-del-software3.pdf>. [Último acceso: 16 Octubre 2016].
- [37] Permalink, «Tecla de escape,» 29 Julio 2015. [En línea]. Available: <http://latecladeescape.com/h/2015/07/metodologias-de-desarrollo-del-software>. [Último acceso: 15 Octubre 2016].
- [38] C. B. Ortega, «Crisis del Software,» 20 Marzo 2012. [En línea]. Available: <http://qualityandprogramming.blogspot.mx/2012/03/crisis-del-software.html>. [Último acceso: 17 Octubre 2016].
- [39] KibaXSama, «software y su crisis,» 12 Marzo 2012. [En línea]. Available: [elsoftwareysucrisis.blogspot.com/](http://elsoftwareysucrisis.blogspot.com/). [Último acceso: 17 Septiembre 2016].
- U. R. J. Carlos, «Calidad Software: Crisis del Software,» 15 Febrero 2010. [En

- [40] línea]. Available: [calidadsw2010.blogspot.com/2010/02/crisis-del-software.html](http://calidadsw2010.blogspot.com/2010/02/crisis-del-software.html). [Último acceso: 23 Septiembre 2016].
- [41] U. Nacional, «Definición de Ingeniería de Software,» 18 diciembre 2011. [En línea]. Available: [http://datateca.unad.edu.co/contenidos/301404/301404\\_ContentoEnLinea/leccin\\_6\\_\\_definicin\\_de\\_ingeniera\\_de\\_software.html](http://datateca.unad.edu.co/contenidos/301404/301404_ContentoEnLinea/leccin_6__definicin_de_ingeniera_de_software.html). [Último acceso: 14 Octubre 2016].
- [42] D. A. R. Tello, «Conceptos basicos de ingenieria de software,» 05 Septiembre 2012. [En línea]. Available: <http://www.tamps.cinvestav.mx/~ertello/swe/sesion01.pdf>. [Último acceso: 17 Septiembre 2016].
- [43] C. Gutiérrez, «¿Para qué sirve el Scrum en la Metodología Ágil?,» 2014. [En línea]. Available: <http://www.i2btech.com/blog-i2b/tech-deployment/para-que-sirve-el-scrum-en-la-metogologia-agil/>. [Último acceso: 20 Septtiembre 2016].
- [44] X. Albaladejo, «Qué es SCRUM,» 2010. [En línea]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Último acceso: 17 Octubre 2016].
- [45] «Metodologías ágiles de gestión de proyectos (Scrum, DSDM, Extreme Programming – XP...),» 7 Junio 2008. [En línea]. Available: <https://www.marblestation.com/?p=661>. [Último acceso: 16 Octubre 2016].
- [46] R. RAYA, «¿Qué son las metodologías ágiles?,» 24 Julio 2014. [En línea].

Available: <http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles/>. [Último acceso: 15 Octubre 2016].

- [47] E. Martínez, «Cómo utilizar la metodología Scrum para acometer proyectos complejos,» 30 Mayo 2013. [En línea]. Available: <http://comunidad.iebschool.com/iebs/general/metodologia-scrum/>. [Último acceso: 17 Octubre 2016].
- [48] javakreator7, «Metodología Ágil: Scrum,» 22 Noviembre 2010. [En línea]. Available: <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Rational+Team+Concert+for+Scrum+Projects/page/SCRUM+como+metodolog%C3%A9a>. [Último acceso: 17 Octubre 2016].
- [49] B. School, «Metodologías ágiles: Scrum y kanban y XP,» [En línea]. Available: <http://www.obs-edu.com/int/blog-project-management/scrum/metodologias-agiles-scrum-y-kanban-y-xp>. [Último acceso: 17 Octubre 2016].
- [50] W. Lara, «Guia Scrum,» 31 Julio 2015. [En línea]. Available: <https://platzi.com/blog/guia-scrum/>. [Último acceso: 17 Octubre 2016].
- [51] M. Soloriio, «METODOLOGIA CASCADA,» 16 Abril 2013. [En línea]. Available: <http://metodologiaencascada.blogspot.mx/>. [Último acceso: 17 Octubre 2016].



