



# Capítulo III

---

MARCO METODOLOGICO

## CAPÍTULO III

### MARCO METODOLÓGICO

Toda investigación debe realizar una serie de pasos, que llevan al uso de técnicas, instrumentos, métodos y estrategias destinados a facilitar el estudio del cual se quiere obtener resultados para resolver el problema por el cual esta investigación se inició. El marco metodológico se puede definir como el conjunto de acciones destinadas a describir y analizar el fondo del problema planteado, a través de procedimientos específicos que incluye las técnicas de observación y recolección de datos determinando el cómo se realizara el estudio.

#### 1. TIPO Y DISEÑO DE LA INVESTIGACIÓN

En la presente investigación se utilizó un tipo de investigación **Proyectiva**, las cuales según Hurtado (2000):

*Consiste en la elaboración de una propuesta o de un modelo, como solución a un problema o necesidad de tipo práctico, ya sea de un grupo social, o de una institución, en un área particular del conocimiento, a partir de un diagnóstico preciso de las necesidades del momento, los procesos explicativos o generadores involucrados y las tendencias futuras.*

Efectivamente, la propuesta de un Sistema CRM bajo entorno web para la gestión de los procesos operativos de la empresa Papelería y Variedades PITITA, C.A. permitirá orientar y organizar en la institución, todo el quehacer administrativo en la mencionada área, instituyendo las bases para la generación del conocimiento que dé respuesta a la problemática que esta presenta.

La investigación fue de diseño de **Campo**, Según Arias (2006):

*La investigación de campo es aquella que consiste en la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variable alguna, es decir, el investigador obtiene la información pero no altera las condiciones existentes. De allí su carácter de investigación no experimental.*

Esta investigación se encontró dentro del diseño **No Experimental**, dado que se observaron los fenómenos, hechos, situaciones o sujetos en su ambiente natural o realidad, no han sido provocados intencionalmente por el investigador. Los autores Hernández, Fernández y Baptista (1991) refieren de una investigación no experimental:

*Es aquella que se realiza sin manipular deliberadamente variables, es decir, una investigación donde no hacemos variar intencionalmente las variables independientes. Lo que hacemos en la investigación no experimental es observar fenómenos tal y como se dan en su contexto natural, para después analizarlos.*

Ahora bien, según la perspectiva temporal, la investigación fue de diseño **Transversal** ya que se midió en un solo momento y una sola oportunidad.

## 2. POBLACIÓN

En lo que concierne al espacio muestral el cual según Cerda (1997) “está constituida por todos los sujetos involucrados en la investigación. Siendo simultáneamente universo, población y muestra”. Por otra parte, se caracteriza como accesible y finita por contener un número menor a cuarenta (40) sujetos.

**Cuadro 1  
Población**

N°	Sujeto	Descripción	Observaciones
01	Administrador	Administrador y dueño de la empresa.	Sin observaciones
01	Supervisor	Supervisor de la empresa.	Sin observaciones
01	Vendedor	Empleado de la empresa.	Sin observaciones
Total: 03			

**Fuente: Leal, Padrón y Sierra (2016)**

## 3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

En cuanto a la técnica de recolección de datos, se escogió la **Entrevista** debido a que el diseño de la investigación será de **Campo** teniendo así una población sujeto con la cual se debe trabajar. Al respecto Arias (2006, p73) define

*La entrevista más que un simple interrogatorio, es una técnica basada en el dialogo o la conversación 'cara a cara', entre el entrevistador y el entrevistado acerca de un tema previamente determinado, de tal manera que el entrevistador pueda obtener la información requerida*

Además, esta fue una **Entrevista no Estructurada** ya que es la más conveniente para este tipo de población. Así mismo Arias la define de la siguiente manera “esta modalidad no se dispone de una guía de preguntas elaboradas previamente. Sin embargo, se orienta por unos objetivos preestablecidos, lo que permite definir el tema de la entrevista” (2006, p74). Es por eso, que el entrevistador debe poseer una gran habilidad para formular interrogantes sin perder la coherencia.

Aparte se utilizó un instrumento de recolección de datos llamado guía de entrevista, los autores Bogdan y Taylor (2000, p.119) definen que:

*La guía de la entrevista no es un protocolo estructurado. Se trata de una lista de áreas generales que deben cubrirse con cada informante. En la situación de entrevista el investigador decide cómo enunciar las preguntas y cuando formularlas. La guía de la entrevista sirve solamente para recordar que se deben hacer preguntas sobre ciertos temas.*

#### **4. METODOLOGIA SELECCIONADA**

Para el desarrollo de esta investigación se planteó el uso de una metodología híbrida, sustentadas en las propuestas proporcionadas por Kent Beck y Thomas A. Powell (2001), mediante la elección de las fases correspondientes para el desarrollo del proyecto según los objetivos que se

plantearon en la investigación, estas conforman un total de cuatro (4) fases y son detalladas a continuación:

- Fase I: Planificación del proyecto (Kent Beck).
- Fase II: Diseño (Kent Beck).
- Fase III: Codificación (Kent Beck).
- Fase IV. Pruebas (Powell).

#### **Fase I: Planificación del proyecto (Kent Beck).**

**Historias de usuario:** El primer paso de cualquier proyecto que siga la metodología X.P es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen.

También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

**Releaseplanning:** Después de tener ya definidas las historias de usuario es necesario crear un plan de publicaciones, en inglés "Release plan", donde se indiquen las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones. Un "Releaseplan" es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa.

Después de un "Release plan" tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado. (\*Release plan: Planificación de publicaciones).

**Iteraciones:** Todo proyecto que siga la metodología X.P. se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Releaseplanning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.

**Velocidad del proyecto:** La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que se pueden desarrollar en las distintas iteraciones.

Usando la velocidad del proyecto se controlara todas las tareas a desarrollarse en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 ó 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Release Plan".

**Programación en pareja:** La metodología X.P. aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado. El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad.

**Reuniones diarias:** Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.

## **Fase II: Diseño (Kent Beck).**

**Diseños simples:** La metodología X.P sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos



complicado posible para conseguir un diseño fácilmente entendible e impleméntable que a la larga costará menos tiempo y esfuerzo desarrollar.

**Glosarios de términos:** Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.

**Riesgos:** Si surgen problemas potenciales durante el diseño, X.P sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.

**Funcionalidad extra:** Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.

**Refactorizar:** Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos. Esto es un error porque puede generar código completamente inestable y muy mal diseñado; por este motivo, es necesario refactorizar cuando se va a utilizar código ya creado.

**Tarjetas C.R.C:** El uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo

orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.

Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

### **Fase III: Codificación (Kent Beck).**

Como ya se dijo en la introducción, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de X.P. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada.

La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

Crear test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Crear estos test antes nos ayuda a saber qué es exactamente lo que tiene que hacer el

código a implementar y sabremos que una vez implementado pasará dichos test sin problemas ya que dicho código ha sido diseñado para ese fin. Se puede dividir la funcionalidad que debe cumplir una tarea a programar en pequeñas unidades, de esta forma se crearán primero los test para cada unidad y a continuación se desarrollará dicha unidad, así poco a poco conseguiremos un desarrollo que cumpla todos los requisitos especificados.

Como ya se comentó anteriormente, X.P opta por la programación en pareja ya que permite un código más eficiente y con una gran calidad.

X.P sugiere un modelo de trabajo usando repositorios de código dónde las parejas de programadores publican cada pocas horas sus códigos implementados y corregidos junto a los test que deben pasar. De esta forma el resto de programadores que necesiten códigos ajenos trabajarán siempre con las últimas versiones. Para mantener un código consistente, publicar un código en un repositorio es una acción exclusiva para cada pareja de programadores.

X.P también propone un modelo de desarrollo colectivo en el que todos los programadores están implicados en todas las tareas; cualquiera puede modificar o ampliar una clase o método de otro programador si es necesario y subirla al repositorio de código. El permitir al resto de los programadores modificar códigos que no son suyos no supone ningún riesgo ya que para que un código pueda ser publicado en el repositorio tiene que pasar los test de funcionamiento definidos para el mismo.

La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar.

X.P afirma que la mayoría de los proyectos que necesiten más tiempo extra que el planificado para ser finalizados no podrán ser terminados a tiempo se haga lo que se haga, aunque se añadan más desarrolladores y se incrementen los recursos. La solución que plantea X.P es realizar un nuevo "Release plan" para concretar los nuevos tiempos de publicación y de velocidad del proyecto.

A la hora de codificar no seguimos la regla de X.P que aconseja crear test de funcionamiento con entornos de desarrollo antes de programar. Nuestros test los obtendremos de la especificación de requisitos ya que en ella se especifican las pruebas que deben pasar las distintas funcionalidades del programa, procurando codificar pensando en las pruebas que debe pasar cada funcionalidad.

#### **Fase IV. Pruebas (Powell).**

Esta fase en un proyecto de desarrollo Web resulta ser la clave para que el usuario final pueda valorar de forma positiva o negativa el proyecto realizado, por ello es necesario que se ejecuten tomando en cuenta al usuario haciendo uso del sistema antes de su finalización. Dichas pruebas incluyen las siguientes:

**Pruebas de Aceptación Visual:** sirven para comprobar que el aspecto del sitio es el deseado, se deben revisar cada una de las páginas contenidas en el sitio Web y asegurarse que son consistentes en disposición, color y en

estilo. Se revisa utilizando diferentes exploradores, resoluciones y entornos visuales que puedan ser empleados por los usuarios reales, realizando una exploración rápida del sitio y observar si la disposición de sus elementos presenta ligeros desplazamientos. Se observa además las páginas entornando los ojos para advertir irregularidades abstractas en la distribución. Para realizar las pruebas visuales, puede resultar necesario imprimir las páginas, aunque no se deberá centrar demasiado en la impresión de las páginas de prueba que ha sido diseñado más bien para su empleo interactivo.

**Pruebas Funcionales:** estas pruebas y las funcionales se solapan en el sentido de que la función más básica de una página es, simplemente, aparecer en la pantalla. Sin embargo, la mayor parte de los sitios disponen de otras funciones básicas, tales como la exploración. Se comprueban todos los vínculos del sitio y rectificar cualquier vínculo que no funcione adecuadamente. Comprobar todos los elementos interactivos, tales como formularios o listas.

**Revisión del Contenido:** los detalles del contenido son muy importantes, se comprueba que se ha introducido todo el contenido deseado en las páginas y que la utilización de las palabras es consistente. Además comprobar ciertos detalles, la ortografía, ya que los clientes y usuarios pueden llegar a catalogar como malo el sitio por el hecho de contener un error tipográfico. La mejor manera de realizar esta prueba es imprimir todas las páginas y leer todas y cada una de sus líneas.

**Pruebas de compatibilidad entre el sistema y el explorador:** las limitaciones del sistema y del explorador deben verificarse mediante la ejecución de pruebas, asegurándose de explorar el sitio utilizando los mismos tipos de sistemas y exploradores que emplearán los usuarios. La planificación del proyecto debe contener ciertos detalles sobre los requisitos del explorador.

**Pruebas de descarga:** se comprueba que el sitio se descarga de forma adecuada, intentando explorar el sitio bajo condiciones reales de utilización por parte del usuario. Si el sitio ha sido diseñado para usuarios conectados mediante módem, se utiliza un módem para comprobar la velocidad de descarga. Para simular el tráfico del sitio, utilice software de emulación para crear usuarios virtuales que acceden a él. Se esta forma se simulará el comportamiento del sitio bajo condiciones de utilización reales, asegurándose de ensayar el sitio en el servidor real o en un sistema equivalente.

**Pruebas de aceptación por parte del usuario:** estas pruebas son realizadas por los usuarios una vez que el sitio parezca funcionar correctamente. En software se conoce a este tipo de pruebas como pruebas beta. Las pruebas de usuario son las más importantes porque en ellas se simula el funcionamiento real de la forma más parecida posible, y si durante esta fase se consiguen problemas éstos no se podrán corregir de manera inmediata. Si no fueran problemas muy graves podría entregarse el sistema y

corregirlos después, pero si los problemas descubiertos son importantes, se debe retrasar la entrega del trabajo hasta después de corregirlos.

**Puesta en funcionamiento del sitio y fases posteriores:** una vez que el sitio se encuentra listo para la entrega, es momento de observar el sitio en funcionamiento, y verificar si ¿se satisface el sitio las expectativas del usuario?, ¿se han cumplido los objetivos del desarrollo del sitio? y si ¿son necesarias pequeñas correcciones?

## 5. CUADRO Y CRONOGRAMA, DE ACTIVIDADES Y RECURSOS

**Cuadro 2**  
**Cuadro de Actividades**

<b>Objetivo General:</b> DESARROLLAR UN SISTEMA CRM BAJO ENTORNO WEB PARA LA GESTION DE LOS PROCESOS OPERATIVOS DE LA EMPRESA PAPELERIA Y VARIEDADES PITITA C.A.			
<b>Objetivos Específicos</b>	<b>Fases Metodológicas</b>	<b>Actividades</b>	<b>Recursos</b>
Analizar los actuales procesos operativos que se realizan en la empresa, PAPELERIA Y VARIEDADES PITITA C.A.	Fase I: Planificación del proyecto. Kent Beck	<ul style="list-style-type: none"> <li>• Historias de usuario.</li> <li>• Releaseplanning</li> <li>• Iteraciones.</li> <li>• Velocidad del proyecto.</li> <li>• Programación en pareja.</li> <li>• Reuniones diarias.</li> <li>• Entrevista no estructurada</li> </ul>	<ul style="list-style-type: none"> <li>• Guía de entrevista.</li> <li>• Registro de usuario.</li> <li>• Revisión bibliográfica especializada.</li> </ul>
Determinar los requerimientos funcionales y no funcionales para la aplicación.			

**Cuadro 2  
(Cont...)**

<p>Diseñar lógica y físicamente el nuevo sistema a partir de los requerimientos obtenidos.</p>	<p>Fase II: Diseño. Kent Beck</p>	<ul style="list-style-type: none"> <li>• Diseños simples.</li> <li>• Glosarios de términos.</li> <li>• Riesgos.</li> <li>• Funcionalidad extra.</li> </ul>	<ul style="list-style-type: none"> <li>• Programas de Diseño web</li> <li>• Generadores de pantallas Componentes de Diseño</li> </ul>
<p>Diseñar lógica y físicamente el nuevo sistema a partir de los requerimientos obtenidos.</p>	<p>Fase III: Codificación. Kent Beck</p>	<ul style="list-style-type: none"> <li>• Crear test de funcionamiento</li> </ul>	<ul style="list-style-type: none"> <li>• Lenguaje de Programación(PHP)</li> <li>• Manejar de Base de Datos (MySQL)</li> </ul>
<p>Realizar las pruebas correspondientes con los empleados de la empresa.</p>	<p>Fase IV: Pruebas Powell</p>	<ul style="list-style-type: none"> <li>• Pruebas de aceptación visual.</li> <li>• Pruebas funcionales.</li> <li>• Revisión del contenido.</li> <li>• Pruebas de compatibilidad entre el sistema y el explorador.</li> <li>• Pruebas de descarga.</li> <li>• Pruebas de aceptación por parte del usuario.</li> <li>• Puesta en funcionamiento del sitio y fases.</li> </ul>	<ul style="list-style-type: none"> <li>• Plan de Pruebas</li> <li>• Software de Prueba</li> <li>• Navegador</li> <li>• MySQL</li> </ul>

Fuente: Leal, Padrón y Sierra (2016)



**Cuadro 3  
Cronograma de Actividades**

Fases	Actividades	Enero				Febrero				Marzo				Abril				Mayo				Junio				Julio				Agosto				Septiembre				Octubre				Noviembre			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Fase I	Historias de usuario	X	X	X	X																																								
	ReleasePlanning					X	X	X	X	X	X	X	X																																
	Iteraciones													X	X	X	X	X	X	X	X																								
	Velocidad del proyecto													X	X	X	X	X	X	X	X																								
	Programación en pareja																					X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								
	Reuniones diarias	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Fase II	Diseños simples																					X	X	X	X	X	X	X	X	X	X	X	X												
	Glosario de términos																									X	X	X	X	X															
	Riesgos																									X	X	X	X	X															
	Funcionalidad extra																									X	X	X	X	X															
	Refactorizar																									X	X	X	X	X															
	Tarjetas C.R.C.																									X	X	X	X	X	X	X	X												
Fase III	Codificación																					X	X	X	X	X	X	X	X	X	X	X	X	X											



Para cumplir los objetivos del proyecto fue necesaria la ejecución de diversas actividades; para ello se hace uso del cronograma de actividades que engloba la planificación desagregada y distribuida en el tiempo. Los cuales se muestran previamente en el **Cuadro 2** y **Cuadro 3** con los detalles de las actividades planificadas y las fechas previstos para su ejecución

## **6. HERRAMIENTAS Y MATERIALES UTILIZADOS**

De acuerdo con el autor Razo “define a las herramientas como “el conjunto de instrumentos de carácter manual, técnico y/o material que sirven como apoyo para la realización de una investigación.” (2001, p. 223). Así mismo Razo menciona que los materiales son “todos aquellos recursos tangibles y de consumo que será utilizados durante el desarrollo de los trabajos de investigación, tales como papelería y útiles de oficina, materiales de cómputo, como cartuchos de tinta, etcétera.” (2011, p. 116).

**Hardware:** Laptop Samsung Np300e4c con las siguientes especificaciones

- Procesador Intel(R) Pentium(r) CPU B950 2.10GHz
- Memoria RAM 4GB
- Pantalla de 17”
- Disco Duro 500GB
- Adaptador de red Qualcomm Atheros AR9485WB-EG Wireless Network Adapter

**Software:**

- Sistema Operativo Windows v.10 de 64 bits
- Navegador Google Chrome v.54.0.2840.59
- Controlador Xampp v.3.2.2
- Servidor Apache v.5.6.15
- Administrador de datos PhpMyAdmin v.4.5.1
- Herramienta de programación Sublime Text Build 3103
- Lenguajes PHP / HTML / SQL
- Estilo CSS
- PHP mail() como envío de mensajes de texto