**CAPITULO III**

**MARCO METODOLÓGICO**

**1.- TIPO Y DISEÑO DE LA INVESTIGACIÓN**

Clasificar el tipo de investigación según la finalidad, el método y la forma de obtener los datos. A continuación se presentan los conceptos de cada uno con su correspondiente contextualización.

**1.1. INVESTIGACIÓN PROYECTIVA**

Para Hurtado (2010, p. 114) este tipo de investigación propone soluciones a una actuación determinada a partir de un proceso de indagación. Implica explorar, describir, explicar y proponer alternativas de cambio, más no necesariamente ejecutar la propuesta. En esta categoría entran los “proyectos factibles”.

Balestrini (2002, p. 130) la investigación proyectiva una proposición sustentada en un modelo operativo factible orientada a resolver un problema planteado o satisfacer necesidades en una institución o campo de interés nacional. Se cumplen dos grandes fases, en la primera de ellas inicialmente se desarrollará un diagnóstico de la situación existente en la realidad objeto de estudio. En la segunda fase del proyecto y atendiendo a los resultados del diagnóstico se formulará el modelo operativo propuesto donde se intenta dar respuesta o resolver el problema planteado en la institución estudiada.

La presente investigación se considera proyectiva ya que propone una solución determinada a los procesos de gestión del estado nutricional del centro de investigaciones E-M, dicha solución consiste en el desarrollo del diagnóstico situacional del centro para comprender las necesidades existentes y posteriormente la formulación de un modelo operativo. Sin embargo, no se implementará la propuesta.

**1.2. INVESTIGACIÓN DESCRIPTIVA**

Según Bavaresco (2006, p. 27) la investigación descriptiva consiste en describir y analizar sistemáticamente características homogéneas de los fenómenos estudiados sobre la realidad. (individuos, comunidades).

De acuerdo a Arias (2006, p. 24) Consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento. Los resultados de este tipo de investigación se ubican en un nivel intermedio en cuanto a la profundidad de los conocimientos en que se refiere.

La presente investigación es descriptiva ya que tiene entre sus cometidos establecer la estructura y el comportamiento adoptados en el centro de investigaciones E-M para el posterior análisis de dichas características.

**1.3. INVESTIGACIÓN DE CAMPO**

Según Bavaresco (2006, p. 28) la investigación de campo se realiza en el propio sitio donde se encuentra el objeto de estudio. Ello permite el conocimiento más a fondo del problema por parte del investigador y puede manejar los datos con más seguridad. así podrá soportarse en diseños explorativos, descriptivos, experimentales y predictivos.

Según Arias (2006, p. 31) 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.

En la investigación es necesario la recolección de datos directos de las personas que están adscritos al CIEM y la información recolectada en el sitio permitirá el conocimiento a fondo de los problemas que son necesarios resolver.

**1.4. DISEÑO DE LA INVESTIGACIÓN**

Arias (2006, p. 26) el diseño de investigación es la estrategia general que adoptó el investigador para responder al problema planteado. En atención al diseño la investigación se clasifica en: documental de campo y experimental.

**1 .4 .1. DISEÑO NO EXPERIMENTAL TRANSECCIONAL DESCRIPTIVO**

Hernández, Fernández y Baptista (2006, p. 205) en este tipo de diseño no se construye ninguna situación sino que se observan situaciones existentes no provocadas intencionalmente en la investigación. El estudio se realiza sin la manipulación deliberada de variables y en los que sólo se observan los fenómenos en su ambiente natural para después analizarlos.

Hernández, Fernández y Baptista (2006, p. 210) el diseño no experimental transeccional descriptivo indaga la incidencia de las modalidades o niveles de una o mas variables en una población, estudio puramente descriptivo.

La investigación se considera no experimental ya que se observarán los procesos ya existentes en función de la gestión actual del estado nutricional. Por su parte, se considera transeccional descriptivo ya que las observaciones se realizarán en un periodo de tiempo único para luego describir los resultados.

**2. POBLACIÓN**

Hurtado (2010, p. 140) define la población como el conjunto de seres que poseen las características o evento a estudiar y que se enmarcan dentro de los criterios de inclusión conforman la población.

Bavaresco (2006, p. 91) Conjunto total de unidades de observación que se consideran en el estudio, es la totalidad de los elementos que forman un conjunto.

Hernández, Fernández y Baptista (2006, p. 238) Es el conjunto de todos los casos que concuerdan con una serie de especificaciones.

Para efectos de la presente investigación se considera una población conformada por veinte miembros del personal de investigación del centro, y un alrededor de dos mil pacientes actualmente involucrados con el CIEM.

**2.1. MUESTREO INTENCIONAL**

Según Namakforoosh (2000, p. 189) en el muestreo intencional todos los elementos muestrales de la población serán seleccionados bajo estricto juicio personal del investigador. En este tipo de muestreo el investigador tiene pleno conocimiento de los elementos poblacionales. Aunque el muestreo es subjetivo, requiere que el investigador conozca los elementos muestrales, lo que permite que el muestreo sea representativo.

Para la selección de la muestra se aplicará el muestreo intencional cuyos criterios de inclusión serán los siguientes:

1. Pacientes que hayan completado la historia nutricional.

2. Pacientes que puedan ser contactados vía telefónica.

**3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS**

Será indispensable la aplicación de un conjunto de técnicas que permitan la recolección de la información a partir de las unidades de observación, tales como las que se describen a continuación.

**3.1. TÉCNICAS**

Según Hurtado (2010, p. 14) las técnicas tienen que ver con los procedimientos utilizados para la recolección de los datos, es decir, él como. Estas pueden ser de revisión documental, observación, encuesta y técnicas socio métricas, entre otras.

**3.1.1. OBSERVACIÓN DIRECTA**

Arias (2006, p. 69) la observación directa es una técnica que consiste en visualizar o captar mediante la vista, en forma sistemática, cualquier hecho o fenómeno o situación que se produzca en la naturaleza o en la sociedad en función de unos objetivos de investigación preestablecidos.

**3.1.2. REVISIÓN DOCUMENTAL**

Bavaresco (2006, p. 99) esta técnica viene a brindarle al lector investigador, todo el soporte del marco teórico (bases teóricas y antecedentes de la investigación) lo que significa que se percata de todo lo escrito o que esté relacionado con el tema que escogió como investigación.

**3.1.3. ENCUESTA**

Según Parella y Martins (2006, p. 134) la encuesta es una técnica destinada a obtener datos de varias personas cuyos opiniones interesan al investigador.

Para ello a diferencia de la entrevista se utiliza un listado de preguntas escritas que se entregan a los sujetos quienes, en forma anónima, las responden por escrito, es una técnica aplicable a los sectores amplios del universo.

**3.1.4. ENTREVISTA**

Según Arias (2006, p. 73) la entrevista, más que un simple interrogatorio, es una técnica basada en un dialogo o conversación “cara a cara”, entre el entrevistador y el entrevistado acerca de un tema previamente determinado, de tal manera que el entrevistador puede obtener la información requerida.

**3.2. INSTRUMENTOS**

Según Hurtado (2010, p. 153) los instrumentos representan la herramienta con la cual se va a recoger, filtrar y codificar la información, es decir, el con que.

**3.2.1. GUÍA DE OBSERVACIÓN**

Según Vallejo, Ordoñez, Villalobos y Sánchez (2008, p. 166),la guia de observación puede definirse como el recurso físico donde el investigador registra los datos o la información obtenida por el uso temático de los sentidos, en la búsqueda de lo que necesitamos para resolver un problema de investigación. Básicamente puede mostrar aspectos observados agrupados, según la conveniencia del estudio, presencia, ausencia y observaciones respectivas.

**3.2.2. HISTORIA DE USUARIO**

Beck, diponible en linea (14/05/2012). Expone que las historias de usuarios poseen la misma finalidad que los casos de uso pero con algunas diferencias a saber. Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin enfocarse mucho en los detalles. No se debe hablar ni de posibles algoritmos para su implementación ni de diseños a base de datos adecuados.

**3.2.3. CUESTIONARIO**

Según Hurtado (2010, p. 157) Consisten en un conjunto de preguntas relacionadas con el evento de estudio. Su característica es que tales preguntas pueden ser dicotómicas, de selección, abiertas tipo escala o tipo ensayo.

**3.2.4. GUIA DE ENTREVISTA**

Hurtado (2010, p. 161) es el instrumento propio de la técnica de la entrevista. En ella el investigador señala los temas o aspectos en torno a los cuales va a preguntar. Según el grado de estructuración puede tener preguntas ya formuladas o solo enunciados temáticos.

**4. METODOLOGÍA SELECCIONADA**

Según Beck (2002, p. 96) La programación extrema es una metodología reciente en el desarrollo de software. La filosofía de X.P es satisfacer al completo las necesidades del cliente, por eso lo integra como una parte más del equipo de desarrollo.

X.P fue inicialmente creada para el desarrollo de aplicaciones dónde el cliente no sabe muy bien lo que quiere, lo que provoca un cambio constante en los requisitos que debe cumplir la aplicación. Por este motivo es necesaria una metodología ágil como X.P que se adapta a las necesidades del cliente y dónde la aplicación se va reevaluando en periodos cortos de tiempo.

X.P está diseñada para el desarrollo de aplicaciones que requieran un grupo de programadores pequeño, dónde la comunicación sea más factible que en grupos de desarrollo grandes. La comunicación es un punto importante y debe realizarse entre los programadores, los jefes de proyecto y los clientes.

**FASES DE LA METODOLOGÍA**

A continuación se explican en forma detallada las fases de planificación del proyecto, diseño, codificación y pruebas las cuales forman parte de la metología.

**1ª FASE: PLANIFICACIÓN DEL PROYECTO**

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.

Release planning: .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 "Release plan" 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 "Release planning" 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 controlaremos que todas las tareas se puedan desarrollar 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.

**2ª FASE: DISEÑO**

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 un 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, Responsabilities 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.

**3ª FASE: CODIFICACIÓN**

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.

**4ª FASE: PRUEBAS**

Uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando.

El uso de los test en X.P es el siguiente:

Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.

Hay que someter a tests las distintas clases del sistema omitiendo los métodos más triviales.

Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.

Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará. Hay que crear los test abstrayéndose del futuro código, de esta forma aseguraremos la independencia del test respecto al código que evalúa.

Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican. Ningún código puede ser publicado en el repositorio sin que haya pasado su test de funcionamiento, de esta forma, aseguramos el uso colectivo del código (explicado en el apartado anterior).

El uso de los test es adecuado para observar la refactorización. Los test permiten verificar que un cambio en la estructura de un código no tiene porqué cambiar su funcionamiento.

Test de aceptación: Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario. Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Test de aceptación"; estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido.

Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

**5. ACTIVIDADES Y RECURSOS**

La actividad según Baizán (2002, p. 7), Es la acción de intervención sobre la realidad necesaria para alcanzar los objetivos específicos de un proyecto. Y la tarea es la acción que tiene el máximo grado de concreción y especificidad. Un conjunto de tareas configura una actividad, entre las muchas que hay que realizar dentro de un proyecto.

Por otra parte, los Recursos según Nava (2002, p. 161), son los medios, materiales y personal que son necesarios para ejecutar el proyecto; deben proveerse para llevar a cabo la recolección de información y su procesamiento. Los recursos materiales son todos los medios, insumos, materia prima e instrumentos necesarios para la ejecución del proyecto y los recursos humanos se refiere a los investigadores co-investigadores y asesores, las personas fuentes de información y los recursos institucionales se refiere aquellas instituciones que prestan su colaboración al momento de ejecutar la investigación.

Las actividades y recursos referente a la presente investigación se muestran en el cuadro 1. De igual forma las actividades del proyecto serán ejecutadas en el tiempo tal como se muestra en el cuadro 2.

**5.1 CUADRO DE ACTIVIDADES Y RECURSOS**

A continuación se presenta el cuadro de actividades donde se desglosan los objetivos y las respectivas fases de la metodología en donde se cumplen, así como las actividades y recursos necesarios para lograr el objetivo.

Cuadro 1

Actividades Y Recursos

**Objetivo general:** Desarrollar un sistema de información bajo ambiente web para la gestión del estado nutricional de pacientes en el centro de investigaciones endocrino-metabólicas.

|  |  |  |  |
| --- | --- | --- | --- |
| **OBJETIVOS ESPECIFICOS** | **FASES** | **ACTIVIDADES** | **RECURSOS** |
| **Describir el proceso de gestión nutricional actual de pacientes en el centro endocrino-metabólicas.** | **FASE I:**  PLANIFICACIÓN | - Definir las historias de usuario.  - Crear plan de publicaciones (Release planning).  - Dividir en iteraciones.  - Estimar velocidad del proyecto.  - Reuniones diarias de desarrolladores.  - Programación en pareja. | - Reunión de cliente y desarrolladores.  - Prioridad de las historias de usuario.  - Definir las tareas.  - Historias de usuario.  - Desarrolladores |
| **Determinar los requerimientos funcionales para el desarrollo del sistema propuesto.** | **FASE II:** DISEÑO. | - Conseguir diseños simples y sencillos.  - Utilizar glosario de términos.  - Reducir riesgos de fallas.  - Refactorizar código.  - Realizar tarjetas C.R.C. | - Nombres de métodos y clases.  - Desarrolladores.  - Código ya creado.  - Clases y objetos del sistema. |
| **Diseñar lógica y físicamente el sistema de acuerdo a los requerimientos preestablecidos.** | **FASE III:** CODIFICACIÓN. | - Codificar mediante estándares.  - Crear pruebas de funcionamiento de código.  - Crear repositorio de código.  - Utilizar un modelo de desarrollo colectivo.  - Optimizar código. | - Estándares.  - Objetivos del código.  - Códigos implementados y correjidos.  - Repositorio de código. |
| **Demostrar la funcionalidad del sistema desarrollado utilizando las pruebas respectivas.** | **FASE IV:** PRUEBAS. | - Crear aplicaciones de prueba.  - Probar las clases del sistema.  - Crear prueba de aceptación. | - Parametros.  - Aplicación de prueba.  - Repositorio de código.  - Historia de usuario. |

**Fuente:** Chavez, Ordoñez y Saavedra (2012).

**5.2 CRONOGRAMA DE ACTIVIDADES**

Según Nava (2002, p. 162), el cronograma de actividades constituye la planificación de las actividades y tareas que se deben realizar para desarrollar el proyecto, se construye de manera ordenada y secuencialmente tomando como criterio el tiempo. Estos aspectos se muestran en el cuadro 2.

(INSERTAR CUADRO 5.2)

**6. HERRAMIENTAS Y MATERIALES**

Las herramientas son los diversos recursos informáticos que se van a utilizar para el desarrollo del sistema.

* DJANGO
* PYTHON
* MySQL
* APACHE
* SUBLIME TEXT
* GOOGLE DRIVE
* GITHUB
* TEAM SPEAK

Por otra parte, Según la RAE los materiales son el conjunto de máquinas, herramientas u objetos de cualquier clase, necesario para el desempeño de un servicio o el ejercicio de una profesión.

* COMPUTADOR
* SERVIDOR
* IMPRESORA
* INFRAESTRUCTURA DE RED
* CONECTIVIDAD WEB