Mejorar el rendimiento del desarrollo de aplicaciones web basada en modelo base de datos orientado a objetos

Web development technique based on model of Object Oriented Data and Entities

Adolfo Vega [1,2]

[1]Universidad Cesar Vallejo Av. Alfredo Mendiola 6232, Panamericana Norte, Lima-Peru
[2]Sección de Post Grado Facultad de Ingeniería de Sistemas, Universidad Nacional de Ingeniería, Av. Túpac Amaru N° 210, Lima-Perú

Recibido : 02/02/2017 Aceptado: 04/03/2017

RESUMEN

En la actualidad los sistemas de información tienen dificultades para conseguir el éxito deseado, debido que una de los sus componentes que es el desarrollo de aplicación web tienen dificultades para ser ejecutado, esto es debido que se requiere un tiempo para la creación de la programación y otro tiempo por separado para la creación del modelo de datos, no existe una técnica que integre ambos aspectos. Estudios recientes ha demostrado que existen varias técnicas de programación tales como programación orientación a objetos, programación orientada a aspectos y programación orientado agentes, y en la creación del modelo de datos también existen técnicas tales como base de datos orientado a objetos y entidad relación, El presente trabajo de investigación tiene como objetivo formular una técnica de desarrollo web basada en modelado de base de datos orientado a objeto haciendo uso de un modelo entidad relación. En consecuencia este estudio propone dos medidas para validar el rendimiento del desarrollo de aplicación web, empleando el análisis de varianza ANOVA, primero la cantidad de líneas de programas creados y el segundo tiempo de respuesta de la base de datos, dirigido a grupo de estudiantes del curso de programación web de la Escuela de Ingeniería de Sistemas de la Universidad Nacional de Ingeniera UNI. Con los resultados obtenidos se espera alcanzar, que los profesionales de software tengan un medio para reforzar sus conocimientos en el desarrollo de aplicación web, así mismo que los proyectos de sistemas de información en la etapa de desarrollo de la programación, tenga el éxito deseado.

Palabras Clave: Técnica Programación, Datos Programación Orientado Objeto, Clase Entidad, Modelo de Datos

ABSTRACT

Nowadays information systems are struggling to achieve the desired success, because one of its components is the development of web application struggle to be done, this is it because it takes a while for the creation of programming and once separately to create the data model, there is no technique that integrates both.
Recent studies have shown that there are several programming techniques such as object oriented programming, aspect-oriented programming and programming targeted agents, and the creation of the data model are also techniques such as database connection object-oriented entity, The this research aims to develop a web development technique based on database modeling object-oriented model using entity relationship.
Consequently, this study proposes two measures to validate the performance of web application development using analysis of variance, first the number of lines of programs created and the second response time database, led a group of students of the course web programming of the School of Universidad Nacional de Ingeniera UNI.
The results obtained are expected to reach that software professionals have a means to strengthen their knowledge in the development of web application, likewise that information systems projects in the development stage of programming, has the desired success

Keywords - Technique Programming, Web, Model of Object Oriented Data, Entities.

I INTRODUCCIÓN

La aparición de los sistema de información también aparecen las tecnologías de información la cuales introducen una seria de componentes tales como lenguaje de programación, base de datos, interface gráfica, internet y comunicación, la técnica de programación que son parte de los lenguajes de programación que han permitido el desarrollo de los sistemas de información [1-18].

Para el desarrollo de aplicación web requieren de lenguaje de programación, bases de datos y herramientas de desarrollo, en las últimas décadas se han diseñados técnicas de programación basado en

Correspondencia:
adolfovega71@hotmail.com

lenguajes de programación (JAVA, PHP, C.), tales técnicas son: la programación orientada a objetos, programación orientada a agentes y la programaciones orientas a aspectos, así mismo para elaborar una base de datos se han diseñado diferentes técnicas tales como base de datos orientado a objeto OODB y el modelado entidad relación.

En las últimas tres décadas, la tecnología de base de datos ha sido objeto de cuatro generaciones de evolución en forma separada del desarrollo de la aplicación web, y la quinta generación de la tecnología de base de datos está actualmente en desarrollo. La primera generación fueron los sistemas de archivos; la segunda generación fueron los sistemas de bases de datos jerárquicas; la tercera generación fue la base de datos CODASYL sistemas [19,21]; y la cuarta generación es la base de datos relacional.

Los sistemas de información aparecieron en la década de los 60 para atender la necesidad de manejar la información de las organización, en la década de los 70 los sistemas de información aparecieron para solucionar problemas empresariales, en la década de los 70 y 80 aparecen nuevas aplicación para atender a la organización sobre el problema de la toma de decisiones y herramientas, en la década de los 90 aparecen los sistema de información con inteligencia artificial.

Los lenguajes de programación en estas últimas décadas han sufrido una serie de cambios con el objetivo de satisfacer los requerimientos de las organizaciones, en la tabla I, se muestra la evolución de los lenguajes de programación [18].

TABLA I. EVOLUCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

Generación	Lenguaje	Fecha Aproximada
Primera	Lenguaje de Maquina	1940s
Segunda	Lenguaje ensamblador	1950s
Tercera	Lenguaje de alto nivel	1960s
Cuarta	Lenguaje de consulta y de base de datos	1070s
Más allá de la cuarta	Lenguaje natural e inteligente	1980s

Los lenguajes de programación son parte de la técnica de programación, en la Figura 1 muestra las técnicas que se han desarrollado en las últimas décadas

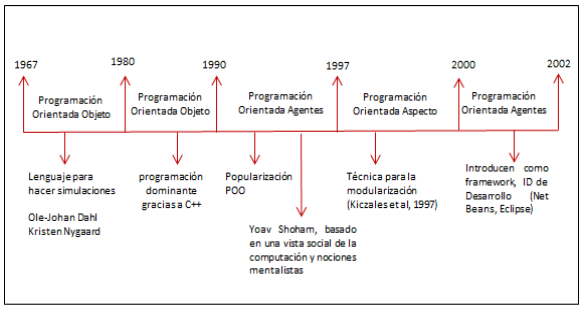


Figura 1. *Evolución de las Técnicas de Programación*

El desarrollo de aplicaciones web han presentado dificultas debido que se emplean de diferentes tiempos, un determinado tiempo para la programación web y otro tiempo por separado para la creación del modelado de datos, en la actualidad no existe un técnica que integren ambos aspectos.

Una de las dificultades para desarrollar las aplicaciones web es la compresión y diseño de los Diagramas Entidad Relación (ERD) no han permito el éxito deseado, un estudio reciente [15] indica que la compresión y diseño de los ERD es una dificultad para los ingenieros de software debido que no mejora el rendimiento de la compresión en el proceso de ERD.

EL modelo entidad-relación (ERD) y los diagramas ER (ERD) son los más utilizado en la mayoría de los desarrolladores o ingenieros de software, la cual permite la comprensión de los requerimientos de datos y modelado de bases de datos estructuras para el sistema de información antes de la etapa de implementación [15].

La determinación de la entidad en los ERD es uno de los principales impedimentos que afectan a la calidad de los datos proporcionado por los sistemas de información. La dificultad de este problema ha sido ampliamente reconocido por las comunidades de investigación [14,20] y profesionales de la industria [9,17], la cuales han impedido contar con la calidad necesaria para el desarrollo de las aplicaciones web.

El presente trabajo de investigación se ha propuesto una técnica de programación basado en la programación orientada a objeto y base de datos orientada a objetos para mejorar el rendimiento del desarrollo de aplicaciones web y que los proyectos de sistemas de información tengan el éxito deseado, así mismo servirá a los profesionales de ingeniería de software como un medio de reforzamiento de los conocimiento de desarrollo de aplicaciones web.

En un trabajo de investigación [2] aplica la técnica de programación aspecto - AspectJ (Framework), para que los desarrolladores o profesionales de ingeniería de software puedan especificar reglas para diseñar los sistemas respetando la modularidad transversal y preservar la modularidad de clase mediante el establecimiento de reglas de diseño, ayudara evitar errores y dar asistencia a los desarrolladores durante el mantenimiento del software y su evolución.

El número de lenguajes de programación es grande y cada vez mayor. Sin embargo, existe poca información estructurada y la evidencia empírica disponible para ayudar a los ingenieros de software a evaluar la idoneidad de un lenguaje de programación para un proyecto de desarrollo en particular o la arquitectura de software. En un artículo [10] presenta un modelo de características construido desde la perspectiva del programador basado en la técnica de programación por agentes, el modelo presentado servirá como una como una herramienta tanto para los profesionales e investigadores, así mismo facilitara a través de comparaciones que lenguaje de programación es el más adecuado y el uso de funciones y la eficacia en contextos específicos de desarrollo.

Un estudio reciente [4] aplica el formalismo de la Entidad-Relación (ER) y orientada a objetos (OO) para modelar los datos especiales, así mismo en este estudio indica que hace 20 años, los investigadores han propuesto diferentes métodos para adaptar ER y OO al modelo de datos espaciales. La información espacial es compleja, y el objetivo es simplificar su representación en modelos conceptuales y proponer una clasificación y una lista de trabajo con la información espacial. Numerosas referencias bibliográficas sobre el tema han sido citadas en este trabajo de investigación, como resultado del trabajo de investigación se propuso un modelo de datos con información espacial. Estos tipos de métodos son de evidente interés para representar la complejidad llena de información ambiental, se emplearon los diagramas ER y OO.

Otro estudio de investigación [15] sobre detección de defectos en los modelos de datos ERD, este estudio propone dos medidas para validar el proceso de detección de ERD con defecto. El primera variable de medición es el nivel de dificultad de detección de defectos y el segunda variable de medición es el rendimiento de detección de defectos, los resultados obtenido en este estudio son validados mediante la recolección de datos observados durante el proceso de detección de defectos, con estos resultados de este estudio se espera proporcionar información a los investigadores, empresas de software, así como a los educadores para mejorar proceso de razonamiento ERD.

II PROPUESTA DEL MODELO

En el presente trabajo de investigación se propone una nueva técnica de programación basado base de datos orientados a objetos utilizando el modelo ER, al cual consisten en los siguientes pasos:

- Paso 01: Análisis de los Objetos
- Paso 02: Elaboración del Modelo Entidad Relación
- Paso 03: Elaboración del Modelo de Clases y comportamiento
- Paso 04: Elaboración del Modelo Relacionar y Orientado a objetos
- Paso 05: Arquitectura por Capas

Para desarrollar los pasos antes indicado sea utilizado como ejemplo un Hospital que cuenta con los servicios de salud tales como Admision, Consulta Externa y Farmacia

A. Análisis de los Objetos:

Antes de determinar los objetos en el Hospital y las áreas involucradas, es necesario revisar la bibliografía de definición de objeto, desde el punto de vista filosofía un objeto es aquello que puede ser observado, estudiado y aprendido [16], otra literatura el objeto en el almacenamiento de información que consta de un estado y comportamiento [18].

En el servicio de admisión del Hospital sea identificado los siguientes objetos

- Paciente
- Cita

En el servicio de consulta externa del Hospital sea identificado los siguientes objetos

- Parte Diario

En el servicio de farmacia del Hospital sea identificado los siguientes objetos:

- Farmacia

B. Modelo Entidad Relación - ERM:

Los ERD es un esquema principal de representaciones de un modelo conceptual de datos que refleje las necesidades de datos de los usuarios en una base de datos [15].

Para la resolución de las entidad sea propuesto el método de basados en la semejanza según los estudios ampliamente reconocido por las comunidades de investigación [14,20] y profesionales de la industria.

En la actualidad existen herramientas CASE que permiten la elaboración de los diagramas entidad relación, en la Fig. 2 se muestra un ejemplo de un diagrama entidad relación para la atención de pacientes en la consulta externa de un hospital.

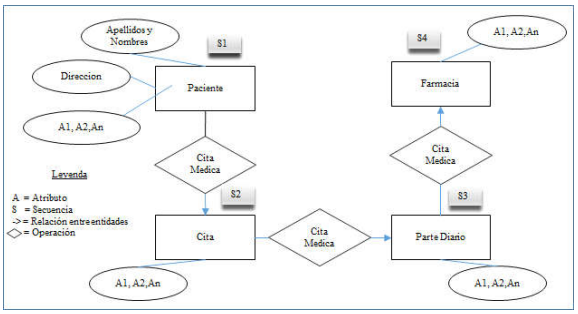


Figura 2. *Diagrama de Herencias y Jerarquías*

C. Modelo Relacional y Orientado a Objetos - OOMR

La herencia es un mecanismo para la construcción de modelado de datos orientada a objetos que permiten una clase llamada subclase heredar atributos y métodos de otra clase llamada superclase, como resultado de la herencia permite la definición de super-clases y subclases, y las clases se organizan en jerarquías de herencia en el que las definiciones de atributos y métodos se heredan entre clases, en la Fig. 3 muestra cómo definir un diagrama de herencia y jerarquía continuado con el ejemplo del hospital se plasma en dicha figura.

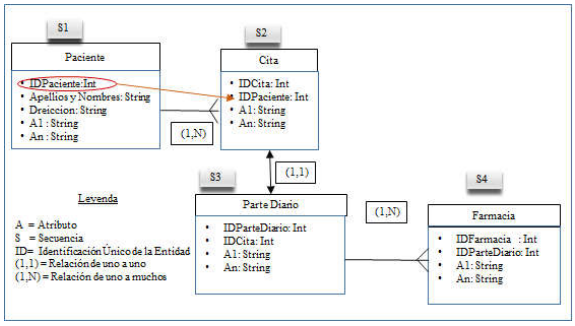


Figura 3. *Diagrama de Herencias y Jerarquías*

D. Diagrama de Clases y Comportamiento:

El UML ofrece diferente tipos de diagramas entre una de ellas es el Diagrama de Clases (Booch et al, 1999; OMG, 2009) [22].

La elaboración del diagrama de clases proviene del OOMR, donde sea definido los objetos y como se encuentran relacionados, con estos de datos se ha construido el diagrama de clases continuado con el ejemplo del hospital, en la Fig. 4 muestra el diagrama de clases donde se utilizado el lenguaje de programación Java para crear las clases llamadas entidades y su comportamiento.

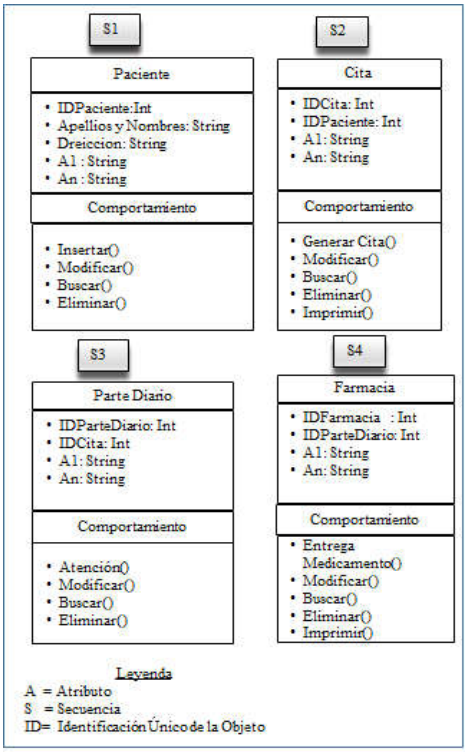
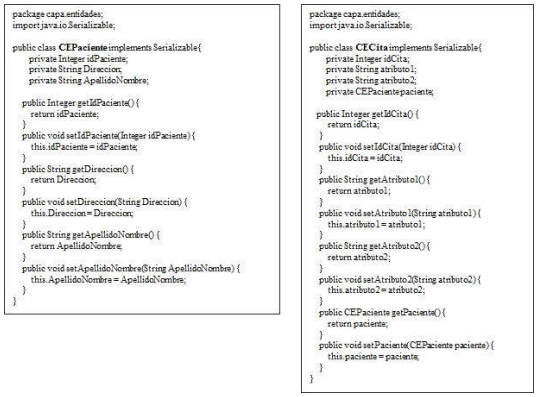


Figura 4. *Diagrama de Clases*

En la Fig. 5 muestra las sub clases son heredas de la superclase, continuado con el ejemplo del hospital.



```
package capa.entidades;
import java.io.Serializable;

public class CEPaciente implements Serializable{
    private Integer idPaciente;
    private String Direccion;
    private String ApellidoNombre;

  public Integer getIdPaciente() {
    return idPaciente;
  }
  public void setIdPaciente(Integer idPaciente) {
    this.idPaciente = idPaciente;
  }
  public String getDireccion() {
    return Direccion;
  }
  public void setDireccion(String Direccion) {
    this.Direccion = Direccion;
  }
  public String getApellidoNombre() {
    return ApellidoNombre;
  }
  public void setApellidoNombre(String ApellidoNombre) {
    this.ApellidoNombre = ApellidoNombre;
  }
}
```

```
package capa.entidades;
import java.io.Serializable;

public class CECita implements Serializable{
    private Integer idCita;
    private String atributo1;
    private String atributo2;
    private CEPaciente paciente;

  public Integer getIdCita() {
    return idCita;
  }
  public void setIdCita(Integer idCita) {
    this.idCita = idCita;
  }
  public String getAtributo1() {
    return atributo1;
  }
  public void setAtributo1(String atributo1) {
    this.atributo1 = atributo1;
  }
  public String getAtributo2() {
    return atributo2;
  }
  public void setAtributo2(String atributo2) {
    this.atributo2 = atributo2;
  }
  public CEPaciente getPaciente() {
    return paciente;
  }
  public void setPaciente(CEPaciente paciente) {
    this.paciente = paciente;
  }
}
```

Figura 5. *Clases con Herencia*

Se puede observar que la sub clase cita.java hereda de la superclase paciente, en consecuencia hereda los atributos con la información de los pacientes.

E. Arquitectura por Capas:

En el presente trabajo de investigación se propone una arquitectura de tres capas: Capa de Negocio, Capa de Entidad, Capa de Datos.

Capa de Negocio: es donde residen los programas que se ejecutan a nivel del servidor de aplicaciones y se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación que son las páginas web JSP, PHP, ASP, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. Empleado el lenguaje de programación Java se ha elaborado un ejemplo de la clase en la capa de negocio, tal como se muestra en la Fig. 6.

```
package capa.negocio;

import capa.datos.CDPaciente;
import capa.entidades.CEPaciente;
import java.sql.SQLExeception;
import javax.naming.NamingExeception;

public class CNPaciente {
      private CEPaciente epaciente = null;

      public void insertar() throws NamingException,
SQLException, Exception{
            epaciente=  new CEPaciente();
            epaciente.setIdPaciente(1);
            epaciente.setApellidoNombre("Apellido y
Nombre");
            epaciente.setDireccion("Direccion");
            CDPaciente.insertar(epaciente);
      }
}
```

```
package capa.negocio;

import capa.datos.CDCita;
import capa.entidades.*;
import java.sql.SQLExeception;
import javax.naming.NamingExeception;

public class CNCita {

    private CEPaciente paciente = null;
    private CECita cita= null;

    public void generarCita() throws NamingException,
SQLException {
          paciente= new CEPaciente();
          cita    = new CECita();
          paciente.setIdPaciente(1);
          cita.setPaciente(paciente);
          cita.setIdCita(1);
          cita.setAtributo1("Atributo1");
          cita.setAtributo2("Atributo2");
          CDCita.insertar(cita);
    }
}
```

Figura 6. *Clases que debe ser considerados en la Capa de Negocio*

Empleado el lenguaje de programación Java, sea creado la clase CNPaciente.java ubicado en la capa de de negocio, se puede observar que la clase instancia el objeto Paciente.java, la cual se almacena de información del Paciente que acude al Hospital, para luego grabar la información en la base de datos desde la clases CDPaciente.java.

Capa de Entidad: es donde residirán las clases entidades definida en el ERM, OOMR y en el diagrama clases, una clase entidad representa la tabla de la bases de datos [4]. Empleando el lenguaje de programación JAVA, sea desarrollado el código, en la Fig.7 muestra el código.

```
package capa.entidades;
import java.io.Serializable;

public class CEPaciente implements Serializable{
      private Integer idPaciente;
      private String Direccion;
      private String ApellidoNombre;

    public Integer getIdPaciente() {
       return idPaciente;
    }
    public void setIdPaciente(Integer idPaciente) {
       this.idPaciente = idPaciente;
    }
    public String getDireccion() {
       return Direccion;
    }
    public void setDireccion(String Direccion) {
       this.Direccion = Direccion;
    }
    public String getApellidoNombre() {
       return ApellidoNombre;
    }
    public void setApellidoNombre(String ApellidoNombre) {
       this.ApellidoNombre = ApellidoNombre;
    }
}
```

```
package capa.entidades;
import java.io.Serializable;

public class CECita implements Serializable{
      private Integer idCita;
      private String atributo1;
      private String atributo2;
      private CEPaciente paciente;

    public Integer getIdCita() {
       return idCita;
    }
    public void setIdCita(Integer idCita) {
       this.idCita = idCita;
    }
    public String getAtributo1() {
       return atributo1;
    }
    public void setAtributo1(String atributo1) {
       this.atributo1 = atributo1;
    }
    public String getAtributo2() {
       return atributo2;
    }
    public void setAtributo2(String atributo2) {
       this.atributo2 = atributo2;
    }
    public CEPaciente getPaciente() {
       return paciente;
    }
    public void setPaciente(CEPaciente paciente) {
       this.paciente = paciente;
    }
}
```

Figura 7. *Clases Entidad CEPaciente.java y CECita.java*

Capa de datos: es donde residen las clases que van acceder hacia la base de datos reciben todas las solicitudes de almacenamiento o recuperación de información desde la capa de negocio, en el ejemplo de la Fig. 8 muestra cómo elaborar un código para acceder a la base de dato.

```
package capa.negocio;

import capa.datos.CDPaciente;
import capa.entidades.CEPaciente;
import java.sql.SQLExeception;
import javax.naming.NamingExeception;

public class CNPaciente {
    private CEPaciente epaciente = null;

      public void insertar() throws NamingException, SQLExeception,
Exception{
            epaciente=  new CEPaciente();
            epaciente.setIdPaciente(1);
            epaciente.setApellidoNombre("Apellido y Nombre");
            epaciente.setDireccion("Direccion");
            CDPaciente.insertar(epaciente);
      }
}
```

```
package capa.datos;

import capa.entidades.CEPaciente;
import java.sql.*;
import javax.naming.NamingExeception;

public class CDPaciente {
      static ResultSet rs = null;
      static Statement stmt = null;
      static CallableStatement cstmt = null;
      static Connection cn = null;


   public static void insertar(CEPaciente epaciente) throws
NamingException, SQLException {
      cn = capa.datos.Conexion.getDataSource();
      cn.setAutoCommit(false);
      String sentenciaSQL = "{? = CALL PA_IPACIENTE(?,?,?)}";
      cstmt = cn.prepareCall(sentenciaSQL);
      cstmt.setInt(1, epaciente.getIdPaciente()) ;
      cstmt.setString(2, epaciente.getApellidoNombre()) ;
      cstmt.setString(3, epaciente.getDireccion());
      cstmt.execute();
      cn.commit();
      cstmt.close();
      cn.close();
   }
}
```

Figura 8. *Clases Datos para acceder a la Base de Datos*

III PROPUESTA DEL MODELO

En este trabajo de investigación para la validación de los resultados se ha utilizado la inferencia estadística para obtener conclusiones entre las medias de dos grupos, así mismo el tipo de diseño es el pres O_0 y post O_1 test observado grupos. Se examinarán técnicas desarrolladas para probar las diferencias en las medias de varios grupos. Esta metodología se clasifica bajo el título general de análisis de varianza o **ANOVA** [3], ampliamente reconocido por la comunidad científica.

X= Estimulación

O_0 ———— O_1

El grupo que empleara para la validación de los resultado será K= Alumnos del curso de programación web de la UNI

Las variables son:

X = cantidad de líneas de programas creados

Y = tiempo de respuesta de la base de dato

Para comenzar, consideremos la variabilidad dentro de los grupos. Para medir la variabilidad en el primer grupo, calcularemos la suma de los cuadrados de las desviaciones de las observaciones respecto de su media muestral $\overline{x}_1$, variabilidad total dentro de los grupos, que denominaremos **SCD**, será la suma de estas sumas de cuadrados para los K grupos, es decir,

$$\mathbf{SCD} = SC_1 + SC_2 + \ldots + SC_k$$

$$SCD = \sum_{i=1}^{k} \sum_{j=1}^{ni} (x_{ij} - \overline{x}_i)^2 \qquad \ldots (1)$$

Lo siguiente es medir la variabilidad entre grupos. Una medida natural será calcular las diferencias entre las medias individuales de cada grupo y la media global. De hecho, como anteriormente, estas diferencias se elevan al cuadrado, quedando.

La suma de los cuadrados de las discrepancias de todas las observaciones muéstrales respecto de su media global. Dicha suma se denomina **la suma de cuadrados total** y se expresa:

$$SCT = \sum_{i=1}^{k} \sum_{j=1}^{ni} (x_{ij} - \bar{x})^2 \qquad \ldots (2)$$

Primero, se puede demostrar que una estimación insesgada de la varianza poblacional resulta de dividir SCD por (n-k). La estimación resultante se conoce como **cuadrado medio dentro de los grupos** y se denota por CMD.

Si las medias poblacionales son iguales, otra estimación insesgada de la varianza poblacional se obtendrá dividiendo SCG por (K-1). La cantidad resultante se denomina **cuadrado medio entre grupos**.

Cuando las medias poblacionales no son iguales, el cuadrado medio dentro de los grupos no será un estimador insesgado de la varianza poblacional común. Por el contrario, el valor esperado de la correspondiente variable aleatoria excederá la varianza poblacional común, ya que incorporará información acerca de las diferencias al cuadrado de las medias poblacionales verdaderas.

Si la hipótesis nula fuese cierta, estaríamos ahora en posesión de dos estimaciones insesgadas de la misma cantidad.

Un contraste a nivel α de significación quedará determinado por la siguiente regla de decisión: Rechazar Ho_ si

$$\frac{CMG}{CMD} > F_{K-1,\, n-k;\ \alpha} \qquad \ldots (3)$$

Fo = HPo < 0.05 Existe diferencia significativa

TABLA II. TABLA DE ANALSISI VARIANZA DE UN FACTOR

FUENTEDE VARIACIÓN	SUMA DE CUADRADOS	GRADOS DE LIBERTAD	CUADRADOS MEDIOS	F RATIO
Entre grupos	SCG	K-1	$CMG = \frac{SCG}{K-1}$	
$\frac{CMG}{CMD}$				
Dentro de grupos	SCD	n-K	$CMD = \frac{SCD}{n-K}$	
Total	SCT	n-1		

III CONCLUSIONES

En el presente trabajo de investigación se han propuesto cinco pasos para desarrollar una aplicación web, se ha tomado como ejemplo un hospital para desarrollar la técnica de programación propuesta, así mismo los más resaltante ha sido la arquitectura basado en tres capas: negocio, entidad y datos, en cada capa se ha insertado el código bajo el lenguaje de programación Java, asimismo, para la validación de los resultados se ha utilizado el pre y post test con la metodología ANOVA, en la que se validaron dos variables, la cantidad de líneas de programas creadas y el tiempo de respuesta de la base de dato. Así mismo, se ha considerado como literatura de base la programación orientada a objetos, y la base de datos orientado a objetos utilizando el modelo entidad-relación para desarrollar la técnica de programación propuesta. Finalmente, con este artículo se espera que los profesionales de software tengan un medio para reforzar sus conocimientos en el desarrollo de aplicación web, así mismo que los proyectos de sistemas de información en la etapa de desarrollo de la programación, tengan el éxito deseado.

REFERENCIAS BIBLIOGRÁFICAS

[1] A. Arasu, M. Götz, R. Kaushik, On active learning of record matching packages, in: Proceedings of the 2010 International Conference on Management of Data, ACM, 2010, pp. 783-794.

[2] Alberto Costa Netoa,Rodrigo Bonifácio, Márcio Ribeiro, Carlos Eduardo Pontual,Paulo Borba, Fernando Castor, "A design rule language for aspect-oriented programming", The Journal of Systems and Software (2013) No of Pages 24.

[3] Alexandra Kuznetsova, Rune H.B. Christensen, Cecile Bavay, Per Bruun Brockhoff "Automated Mixed ANOVA Modelling of sensory and consumer data" Food Quality and Preference (2014) 1-34.

[4] A. Francois Pinet "Entity-relationship and object-oriented formalisms for modeling spatial environmental data" Environmental Modelling & Software 33 (2012) 80-91.

[5] A.Regia de M. Nieves, A. Carvalho, C. Ralha "Agent-based architecture for context - aware and personalized even Recommendation", Expert Systems with Aplicactions 41 (2014) 563 - 573

[6] D.Krzywicki, F. Byrski, M. Kisiel "Computing agents for decision support systems", Future Generation Computer Systems 37 (2014) 390-400.

[7] D. Talaba, C. Antonya, "Design evaluation and modification of mechanical systems" in virtual environments, Virtual Reality 11 (4) (2012) 275-285.

[8] D. Weidlich, L. Cser, T. Polzin, D. Cristiano, H. Zickner, "Virtual reality approaches for immersive design", in Annals of the CIRP 56 (1) (2010) 139-142.

[9] Fischer identity as a service. Architecture overview for client organizations. White paper, Fischer International Identity, 2009.

[10] H. Jordan, G. Botterweck, J. Noll, A. Butterfield "A feature model of actor, agent, functional, object, and procedural programming languages" Science of Computer Programming (2014) 167-6423

[11] I. Garcia, Magarin "A collection of method fragments automated with model transformations in agent-oriented modeling", Engineering Applications of Artificial Intelligence 26 (2013) 1131-1148.

[12] I. Garcia, C. Gutierrez "Agent-oriented modeling and development of a system for crisis management", Expert Systems with Applications 40 (2013) 6580-6592.

[13]] I. Freitas, P. da Mota, P. Oleary, E. Santana, S. Romero "Software product line scoping and requirements engineering in a small and medium-sized enterprise:An industrial case study", The Journal of Systems and Software 88 (2014) 189-206.

[14] I. Bhattacharya, L. Getoor, Collective entity resolution in relational data, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 5.

[15] Nergiz Ercil Cagiltaya, Gul Tokdemirb, Ozkan Kilic c,?, Damla Topalli "Performing and analyzing non-formal inspections of entity relationship diagram", The Journal of Systems and Software 86 (2013) 2184- 2195.

[16] S. Gunther, S. Sunkle. "RBFeatures: Feature-oriented programming with Ruby", Science of Computer Programming 77(2012) 152-173.

[17] S. Slone, The open group identity management work area. Identity management White paper, The Open Group, 2004, conference on Managemen of Data, ACM 2005. Pp 85 -96.

[18] Ralph Stair, George Reynolds, Principios de Sistemas de Información, Cenage Learning, México, Mayo de 2010, ISBN-10: 6074812675, 704 p

[19] R.W. Taylor, R.L. Frank, CODASYL data - base management systems, ACM Comput. Sur. 8 (1) (1976) 67-103.

[20] X. Dong, A. Halevy, J. Madhavan, Reference reconciliation in complex information spaces, in: Proceedings of the 2005 ACM SIGMOD International.

[21] WonKim, "Research Directions in Object-Oriented Database Systems", Microelectronics and Computer Technology Corporation 3500 West Balcones Center Drive Austin, Texas 78750.

[22] Z.M. Ma, Li Yan, Fu Zhang,, Modeling fuzzy, Modeling fuzzy information in UML class diagrams and object-oriented database models, Fuzzy Sets and Systems 186 (2012) 26-46.