**INACAP TEMUCO**

INGENIERÍA EN INFORMÁTICA

**Taller de Proyecto Informático III**

Proyecto

Migración de Datos en Excel a un Modelo Relacional (MEMRE)

**Nombres: Tracy Solange Padilla Sierra**

**Luis Alberto Lizama Bustos**

**Carlos Ignacio Héctor Andrés Herrera**

**Guillermo Soto**

Nombre Profesor: Henry José Villagra Molina

Sección: 75

Fecha: 29 de Abril de 2013

Índice

[1 Definición del problema 4](#_Toc354871228)

[1.1 Planteamiento del Problema 4](#_Toc354871229)

[1.2 Descripción del Problema 4](#_Toc354871230)

[1.3 Elementos del Problema 4](#_Toc354871231)

[1.4 Delimitación del Problema 5](#_Toc354871232)

[1.5 Formulación del Problema 5](#_Toc354871233)

[2 Objetivos 5](#_Toc354871234)

[2.1 Objetivo general 5](#_Toc354871235)

[2.2 Objetivos específicos 5](#_Toc354871236)

[3 Justificación 6](#_Toc354871237)

[4 Impacto 6](#_Toc354871238)

[4.1 Impacto Técnico 6](#_Toc354871239)

[4.2 Impacto Social 6](#_Toc354871240)

[4.3 Impacto Económico 7](#_Toc354871241)

[4.3 Impacto Legal 7](#_Toc354871242)

[5 Alcances y logros 7](#_Toc354871243)

[5.1 Alcances del proyecto 7](#_Toc354871244)

[5.2 Logros del proyecto 8](#_Toc354871245)

[6 Recolección de requerimientos 8](#_Toc354871246)

[7 Análisis de requerimientos 9](#_Toc354871247)

[8 Marco Teórico 10](#_Toc354871248)

[8.1 Marco referencial 10](#_Toc354871249)

[8.1.1 Problemática 10](#_Toc354871250)

[8.1.2 Solución 11](#_Toc354871251)

[8.1.3 Hojas de Excel como arreglos multidimensionales 12](#_Toc354871252)

[8.2 Marco conceptual 13](#_Toc354871253)

[8.2.1 Modelo relacional 13](#_Toc354871254)

[8.2.2 Descripción 13](#_Toc354871255)

[8.2.3 Esquema 14](#_Toc354871256)

[8.2.4 Instancias 15](#_Toc354871257)

[8.2.5 Dominios 15](#_Toc354871258)

[8.2.6 Claves 15](#_Toc354871259)

[8.2.7 Problemas del esquema relacional 16](#_Toc354871260)

[9 Definición de módulos 17](#_Toc354871261)

[9.1 Módulo Carga Masiva 17](#_Toc354871262)

[9.2 Módulo Mantenedores 18](#_Toc354871263)

[10 Modelo de datos 19](#_Toc354871264)

[11 Planificación 21](#_Toc354871265)

[12 Conclusión 24](#_Toc354871266)

[13 Bibliografía 25](#_Toc354871267)

# 1 Definición del problema

## 1.1 Planteamiento del Problema

El Gobierno Regional de la Araucanía GORE solicita la migración de datos de una funcionalidad del organismo, estos se encuentra en formato Excel, y se migrarán hacia un modelo relacional que cumpla con los estándares de escalabilidad para aportar una mejora a la empresa y sea éste un soporte para futuros sistemas.

## 1.2 Descripción del Problema

Actualmente los datos no se encuentran centralizados y para el organismo es un gran problema la gestión de estos. Al contener los datos en formato Excel no pueden acceder todos los usuarios al mismo tiempo y la integridad de estos se ve afectada tanto en la seguridad como en su accesibilidad. Es muy difícil generar reportes y estadísticas ya que no pueden utilizar herramientas actualizadas para el análisis de la información de su negocio.

## 1.3 Elementos del Problema

* Hojas de calculo Excel 2003
* Migración de datos
* Modelo de datos relacional
* Validación e integridad de datos

## 1.4 Delimitación del Problema

Se migrara un único formato específico y definido con los datos de la funcionalidad del organismo correspondiente al gobierno regional de la Araucanía, este se encuentra en formato Excel 2003.

## 1.5 Formulación del Problema

Se analizaran los datos contenidos en la hoja de calculo Excel para generar un modelo de datos relacional acorde a lo que los datos representen. Luego se definirán las validaciones de datos para aplicarlas sobre el proceso de migración, de manera que se pueda poblar el modelo de manera integra. Se informara el usuario del éxito del proceso de migración o de error en caso que corresponda.

# 2 Objetivos

## 2.1 Objetivo general

* Migrar hacia un modelo relacional los datos de una funcionalidad del Gobierno Regional de la Araucanía, los cuales se encuentra en formato Excel.

## 2.2 Objetivos específicos

* Centralizar la información
* Dar integridad a la información
* Actualizar tecnológicamente el uso de la información
* Agilizar la gestión de la información
* Crear un soporte escalable para futuros sistemas
* Mejorar la disponibilidad de la información

# 3 Justificación

Porque el gobierno regional de la Araucanía GORE solicito la implementación de una migración de datos aplicable a una funcionalidad del organismo, estos están en formato Excel y necesitan un sistema para poder traspasar estos datos directamente a un modelo relacional.

Los beneficios que conlleva es llegar a tener un sistema que cumpla el objetivo principal de la conversión de datos de una planilla Excel a un sistema.

# 4 Impacto

## 4.1 Impacto Técnico

El organismo cuenta con sus propios servidores y bases de datos, por lo que no existirá impacto técnico ya que se montara el sistema en estos mismos. Sera necesario realizar capacitación sobre el correcto uso del sistema para migrar los datos.

## 4.2 Impacto Social

Al migrar los datos en un modelo relacional, se logrará acelerar los procesos dentro del organismo cuando desarrollen sistemas que utilicen el nuevo modelo. Esta aceleración en los procesos a largo plazo, se verá reflejada en una atención más expedita hacia las personas que realizan trámites diarios con el GORE.

## 4.3 Impacto Económico

El desarrollo del proyecto suma en gastos de sueldos considerando a todo el equipo de trabajo por la suma $1.846.000 pesos. No se realizaran gastos de otra índole.

## 4.3 Impacto Legal

El sistema migrara información confidencial y sensible la cual no podrá ser difundida a terceros ajenos al desarrollo del proyecto. Por ende y por motivos de seguridad el organismo proveerá datos modelo para el desarrollo.

# 5 Alcances y logros

## 5.1 Alcances del proyecto

El alcance del presente proyecto consta en desarrollar el sistema que migrara los datos desde la planilla Excel hacia el modelo relacional. Instalar dicho sistema en los servidores que el GORE ya posee, probarlo y finalmente capacitar a una persona sobre cómo utilizarlo.

El GORE proveerá el formato de la planilla Excel, dicho formato no será dinámico.

Los entregables del proyecto serán las fuentes del sistema y los scripts SQL para generar el modelo relacional en base de datos.

No se consideraran modificaciones a la toma de requerimientos inicial como parte del desarrollo del sistema, cualquier modificación que estime el GORE tendrá que evaluarse como un desarrollo independiente.

El sistema está destinado específicamente para el GORE presente en la región de la Araucanía, no estando este diseñado para otras regiones u otros sistemas.

El sistema no tendrá gestión de usuarios, las consideraciones de acceso y seguridad a este serán exclusiva responsabilidad del GORE.

## 5.2 Logros del proyecto

MEMRE proporcionará las herramientas operativas que permitirán darle un valor agregado a la información el cual servirá de base para automatizar e integrar los procesos de negocios, optimizar y llevar un eficiente control de los datos de los datos que antiguamente gestionaban en planillas Excel.

# 6 Recolección de requerimientos

Se necesitan migrar datos de una funcionalidad desde una planilla Excel 2003 a un modelo relación para MySQL con PHP.

El modelo tiene que ser una base para futuros sistemas y se deben poder realizar cargas posteriores.

Se deben validar los datos que se suben en la carga y visualizar al usuario el estado de la carga, como los errores que se produzcan por concepto de validación o del éxito del proceso.

Debe existir una forma de hacer mantención sobre los datos que no representan registros de la carga masiva.

No es necesario que el sistema maneje usuarios ni registro de aquello, ya que solo personal capacitado tendrá acceso al sistema.

Se deben desarrollar las interfaces sin estilos ya que estos se definirán posteriormente por diseñadores graficos.

# 7 Análisis de requerimientos

En vista de las funcionalidades solicitadas por GORE se desglosara el desarrollo del proyecto en dos partes, una para la carga masiva como tal, otra para el mantenimiento de datos.

El lenguaje de desarrollo y el motor de base de datos ya han sido definidos por el GORE por lo que el sistema será desarrollado en PHP con MySQL.

Para la migración de datos se creara una única interfaz en la cual se seleccionara el archivo Excel, se realizara que el archivo corresponda al formato y se realizaran las validaciones de datos. El sistema responderá al usuario con la información del proceso ya sea con un mensaje de éxito si la carga resulto exitosa, o con los errores producto de la validación cuando vengan datos incorrectos.

Para la mantención de datos se desarrollara una única interfaz en la cual se podrán realizar solamente las operaciones de insertar o modificar datos, no existirá eliminación ya que impactara sobre los registros históricos.

Se creara una única interfaz la cual tendrá accesos directos a las dos interfaces descritas anteriormente, en la cual no existirán restricciones de usuarios registrados ya que el acceso a esta será exclusiva responsabilidad del GORE.

Todas las interfaces se desarrollaran en código HTML estándar sin definir estilos ni formatos por requerimiento del GORE.

# 8 Marco Teórico

## 8.1 Marco referencial

El proceso de descentralización que se ha profundizado en las últimas dos décadas ha significado un aumento de atribuciones, competencias y facultades para las regiones. La reforma constitucional y legal de 1992, que dio vida a los Gobiernos Regionales como organismos con personería jurídica y patrimonio propio, constituye el avance más significativo en la descentralización del Estado chileno, desde la creación de las regiones en el año 1976.

El Gobierno Regional es un servicio autónomo que tiene a su cargo la administración de la región, y cuya misión es: "Liderar el desarrollo de la Región de La Araucanía, implementando políticas públicas con pertinencia multicultural, equidad e identidad territorial, sustentabilidad e inclusión".

Está compuesto por el Intendente Regional en su calidad de Órgano Ejecutivo y por el Consejo Regional (CORE), que está formado por representantes de las dos provincias de la región y que son electos por los concejales de cada una de las comunas. El Consejo Regional o CORE, define y sanciona las estrategias regionales de desarrollo y sus modificaciones, las iniciativas de inversión en la región, a través de la asignación de recursos provenientes del Fondo Nacional de Desarrollo Regional.

### 8.1.1 **Problemática**

El Gore, mantiene información registrada en sus sistemas confeccionados  en planillas de cálculo Excel. Al tratarse de una institución estatal, la cantidad de información que se  maneja es bastante, puesto que se almacenan todos los documentos relacionados por los proyectos que se generan a través del FNDR (Fondo Nacional de Desarrollo Regional) y el FONDEMA (Fondo de Desarrollo para Magallanes), y por ley, estos registros se deben mantener como mínimo por cinco años.

Dentro  de las problemáticas que se presentan al mantener este tipo de sistema de archivos podemos encontrar las siguientes:

1. Lentitud generada al buscar algún tipo de registro.
2. La imposibilidad de poder generar informes basados en las planillas de cálculo.
3. Además existe una nula posibilidad de que puedan acceder dos o más personas a los registros al mismo tiempo.  Puesto que al abrir por segunda vez el archivo Excel, este se abrirá en modo solo lectura, no permitiendo realizar ningún cambio a la planilla.

Es por esto que el GORE, ha solicitado realizar la migración de estos archivos a un modelo relacional, con tal de poder solucionar las problemáticas antes mencionadas.

### 8.1.2 Solución

El lenguaje de programación que utilizaremos será PHP, por qué los sistemas de clientes de GORE Araucanía están hechos en PHP y así también lo ha determinado cliente, y debido a que nuestro proyecto es una mejora en una función en específica pero siguiendo todas las normativas y directrices existentes en la entidad mandante, continuaremos en la utilización del lenguaje PHP.

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools).

Específicamente para la implementación de nuestro proyecto haremos uso de la librería opensource (código libre) de PHP que es PhpExcel reader, la cual nos permite leer archivos de Excel de manera trasparente y rápida para el desarrollo del proyecto, sin la necesidad de reinventar la rueda.

### 8.1.3 Hojas de Excel como arreglos multidimensionales

Esta etapa es sumamente relevante para poder leer un archivo de XLS (Excel) desde un archivo PHP. La lógica es la siguiente:

1. Imagina una hoja de Excel como un arreglo “gigante” y dentro de ese arreglo tienes X cantidad de Columnas e Y cantidad de Filas.

2. Siempre piensa en arreglo de modo como si de una tabla de coordenadas se tratará es decir si tu dato en Excel se ubica en la celda A3, para PHP Excel Reader estará en la posición: $arreglo [1][3]

Lo anterior nos permite acceder a la información contenida en el archivo Excel de manera rápida y simple de entender, lo cual le da escalabilidad al sistema para futuras posibles mejoras, y también ahorra tiempo de desarrollo.

## 8.2 Marco conceptual

### **8.2.1 Modelo relacional**

Se utiliza para la gestión de una base de datos en un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postuladas sus bases en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos.

Su idea fundamental es el uso de «relaciones». Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados «tuplas». Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar, esto es, pensando en cada relación como si fuese una tabla que está compuesta por registros (cada fila de la tabla sería un registro o tupla), y columnas (también llamadas campos).

### **8.2.2 Descripción**

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que éstos se almacenen no tiene relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales, el Álgebra relacional y el Cálculo relacional. El Álgebra relacional permite describir la forma de realizar una consulta, en cambio, el Cálculo relacional sólo indica lo que se desea devolver.

En las bases de Codd se definían los objetivos de este modelo:

* **Independencia física:** La forma de almacenar los datos, no debe influir en su manipulación lógica
* **Independencia lógica:** Las aplicaciones que utilizan la base de datos no deben ser modificadas por que se modifiquen elementos de la base de datos.
* **Flexibilidad:** La base de datos ofrece fácilmente distintas vistas en función de los usuarios y aplicaciones.
* **Uniformidad:** Las estructuras lógicas siempre tienen una única forma conceptual (las tablas).

### **8.2.3 Esquema**

Un esquema contiene la definición de una estructura (generalmente relaciones o tablas de una base de datos), es decir, determina la identidad de la relación y que tipo de información podrá ser almacenada dentro de ella; en otras palabras, el esquema contiene los metadatos de la relación. Todo esquema constará de:

* Nombre de la relación (su identificador).
* Nombre de los atributos (o campos) de la relación y sus dominios; el dominio de un atributo o campo define los valores permitidos para el mismo, equivalente al tipo de dato por ejemplo character, integer, date, string, etc.

·

### **8.2.4 Instancias**

Una instancia de manera formal es la aplicación de un esquema a un conjunto finito de datos. En palabras no tan técnicas, se puede definir como el contenido de una tabla en un momento dado, pero también es válido referirnos a una instancia cuando trabajamos o mostramos únicamente un subconjunto de la información contenida en una relación o tabla, como por ejemplo:

* Ciertos caracteres y números (una sola columna de una sola fila).
* Algunas o todas las filas con todas o algunas columnas
* Cada fila es una tupla. El número de filas es llamado cardinalidad.
* El número de columnas es llamado grado.

### **8.2.5 Dominios**

Los dominios suponen una gran mejora en este modelo ya que permiten especificar los posibles valores válidos para un atributo. Cada dominio incorpora su nombre y una definición del mismo. Ejemplos de dominio:

Dirección: 50 caracteres

Nacionalidad: Español, Francés, Italiano,...

Los dominios pueden ser también compuestos a partir de otros (año, mes y día = fecha).

### **8.2.6 Claves**

* **Clave candidata:** Conjunto de atributos de una tabla que identifican unívocamente cada tupla de la tabla.
* **Clave primaria:** Clave candidata que se escoge como identificador de las tuplas.
* **Clave alternativa:** Cualquier clave candidata que no sea primaria
* **Clave externa o secundaria:** Atributo de una tabla relacionado con una clave de otra tabla.

### **8.2.7 Problemas del esquema relacional**

Una vez obtenido el esquema relacional resultante del modelo entidad relación que representaba la base de datos, normalmente tendremos una buena base de datos. Pero otras veces, debido a fallos en el diseño o a problemas indetectables en esta fase del diseño, tendremos un esquema que puede producir una base de datos que incorpore estos problemas:

* **Redundancia:** Se llama así a los datos que se repiten continua e innecesariamente por las tablas de las bases de datos.
* **Ambigüedades:** Datos que no clarifican suficientemente el registro al que representan.
* **Pérdida de restricciones de integridad:** Las relaciones no se encuentran coherentemente establecidas o están inconclusas.
* **Anomalías en operaciones de modificación de datos:** El hecho de que al insertar un solo elemento haya que repetir tuplas en una tabla para variar unos pocos datos. O que eliminar un elemento suponga eliminar varias tuplas.
* El principio fundamental reside en que las tablas deben referirse a objetos o situaciones muy concretas. Lo que ocurre es que conceptualmente es difícil obtener ese problema.
* La solución suele ser dividir la tabla con problemas en otras tablas más adecuadas.

# 9 Definición de módulos

Los distintos módulos del sistema fueron pensados de manera que se pueda separar las distintas características que posee el servicio según sus características.

Los módulos del sistema son los siguientes:

* Carga Masiva
* Mantenedores

## 9.1 Módulo Carga Masiva

En la carga masiva el sistema inicialmente presentara una interfaz en la cual el usuario podrá seleccionar un único archivo Excel, del cual se validara que corresponda a un tipo de archivo Excel 2003 con un tamaño máximo de 4 megas. Si no se cumple con esta validación inicial se informara al usuario con un mensaje de error.

Luego el sistema analizara los datos del Excel, cuyo formato contenedor no será dinámico sino que deberá cumplir con las siguientes columnas con su tipo y longitud:

* FUENTE – Caracteres, longitud máxima 4
* BIP – Numérico, mayor a 0 sin decimales, longitud máxima 10
* ETAPA – Caracteres, longitud máxima 255
* CUENTA – Numérico, mayor a 0 sin decimales, longitud máxima 10
* NOMBRE – Caracteres, longitud máxima 255
* ENERO – Numérico, mayor a 0 sin decimales, longitud máxima 10
* FEBRERO – Numérico, mayor a 0 sin decimales, longitud máxima 10
* MARZO – Numérico, mayor a 0 sin decimales, longitud máxima 10
* ABRIL – Numérico, mayor a 0 sin decimales, longitud máxima 10
* MAYO – Numérico, mayor a 0 sin decimales, longitud máxima 10
* JUNIO – Numérico, mayor a 0 sin decimales, longitud máxima 10
* JULIO – Numérico, mayor a 0 sin decimales, longitud máxima 10
* AGOSTO – Numérico, mayor a 0 sin decimales, longitud máxima 10
* SEPTIEMBRE – Numérico, mayor a 0 sin decimales, longitud máxima 10
* OCTUBRE – Numérico, mayor a 0 sin decimales, longitud máxima 10
* NOVIEMBRE – Numérico, mayor a 0 sin decimales, longitud máxima 10
* DICIEMBRE – Numérico, mayor a 0 sin decimales, longitud máxima 10
* UNIDAD TECNICA – Caracteres, longitud máxima 255

Si alguna celda de la Excel no cumple con el formato indicado se rechazara la Excel completa informando finalmente al usuario cuales filas de la Excel son las que están erróneas indicando el número de la fila con las celdas y su mensaje de error.

Los mensajes de error para cada campo serán provistos por GORE por lo que se creara un XML estándar en cual podrán ingresar dichos valores y el sistema lo leerá, por lo que no corresponde especificarlos en este documento.

En caso de que ningún dato se encuentre erróneo el sistema cargara dichos datos en el modelo relacional creado para tal efecto, informando posteriormente al usuario del éxito del proceso con un mensaje de éxito y volviendo al estado original de la interfaz.

## 9.2 Módulo Mantenedores

En el módulo de mantenedores el sistema inicialmente presentara una interfaz en la cual se podrá realizar mantención de datos sobre las tablas de ‘fuente’, ‘unidad técnica’ y ‘etapa’, dicha mantención corresponde en ejecutar acciones de inserción y modificación de datos, no se podrán eliminar ni hacer consultas.

El usuario inicialmente al acceder a la interfaz debe elegir sobre que tabla realizara mantención, luego deberá elegir si desea hacer inserción o modificación.

En el caso de la inserción se validaran los datos ingresados o que el registro no exista, si no se pasan estas validaciones se mostraras un mensaje de error permitiendo al usuario intentar nuevamente. En caso de éxito se informara el usuario con un mensaje y volviendo la interfaz al estado original. Los datos y su validación será la siguiente:

Fuente

* Nombre – Caracteres, longitud máxima 255

Unidad Técnica

* Nombre – Caracteres, longitud máxima 255

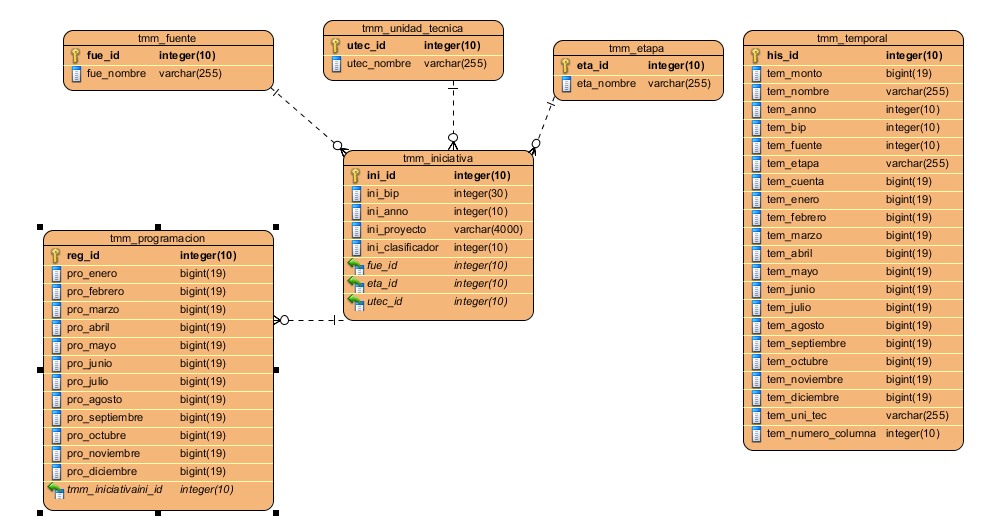
Etapa

* Nombre – Caracteres, longitud máxima 255

En el caso de la modificación el usuario deberá seleccionar de entre los datos existentes cual va a modificar, luego alterar su contenido y grabar. El sistema registrara el cambio en el modelo relacional e informara el usuario del éxito del proceso volviendo la interfaz al estado original.

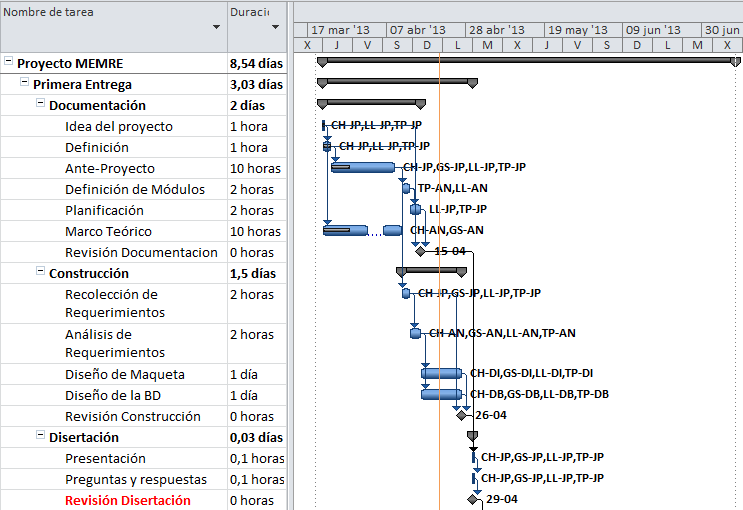
# 10 Modelo de datos

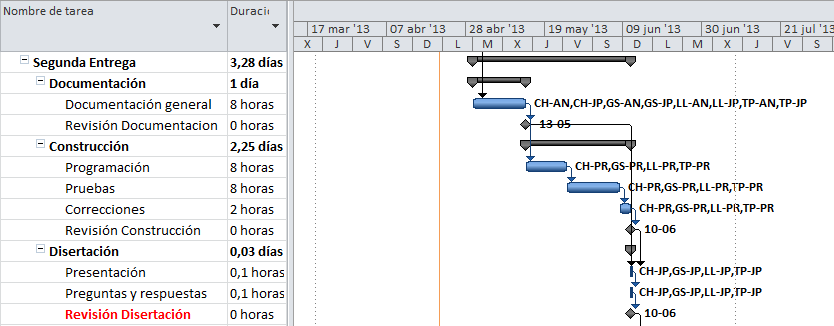
A continuación se describe la representación gráfica del modelo relacional al cual será migrada la información.

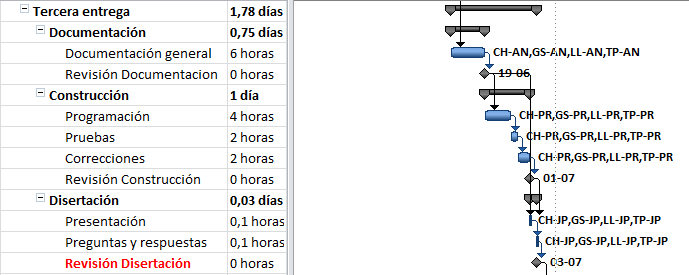


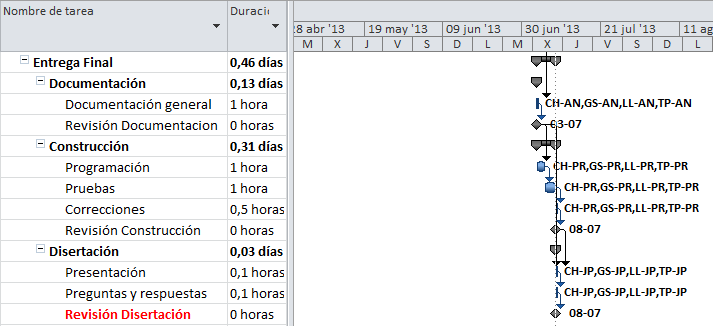
# 11 Planificación

A continuación se describe la planificación del proyecto con su respectiva representación Gantt.









# 12 Conclusión

# 13 Bibliografía

1)   Véase: R. Hernández Sampieri, Fernández Collado, C. Y Baptista Lucio, P., Metodología de la Investigación,  McGraw Hill, México, 2006. pp 64 ­65.

2)    FAQ General PHP.net.

3)   <http://blog.comunicacionweb.info>