

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**DESARROLLO E IMPLEMENTACIÓN DE APLICACIONES
DE SUPERVISIÓN Y CONTROL DE GENERACIÓN
ELÉCTRICA Y ANÁLISIS DE DATOS HISTÓRICOS PARA
LA EMPRESA ELÉCTRICA PROVINCIAL COTOPAXI S.A.**

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS

POR:

**MARCOS PATRICIO ARMAS VILLAMARÍN
SANTIAGO DAVID LUCIO CRUZ**

SANGOLQUÍ, Noviembre del 2012

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por los Srs. MARCOS PATRICIO ARMAS VILLAMARIN Y SANTIAGO DAVID LUCIO CRUZ como requerimiento parcial a la obtención del título de INGENIEROS EN SISTEMAS.

Sangolquí, noviembre del 2012

Ing. Mario Ron

DEDICATORIA

A Dios.

Por permitirme llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mi madre Nelly.

Por apoyarme en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A mi padre Miguel.

Por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

Santiago Lucio

DEDICATORIA

A Dios y a mis padres Marcos y María, por ser quienes han estado presentes en todos los momentos buenos y malos de mi vida.

Marcos Armas

AGRADECIMIENTOS

La presente Tesis es un esfuerzo en el cual, directa o indirectamente, participaron varias personas leyendo, opinando, corrigiendo, teniéndome paciencia, dando ánimo, acompañando en los momentos de crisis y en los momentos de felicidad.

Agradezco al Ing. Mario Ron por haber confiado en nosotros, por la paciencia y por la dirección de este trabajo. Al Ing. Diego Marcillo por los consejos, el apoyo y el ánimo que nos brindó, por sus comentarios en todo el proceso de elaboración de la Tesis y sus atinadas correcciones.

A mi Madre y hermanos que me acompañaron en esta aventura que significó la Ingeniería y que, de forma incondicional, entendieron mis ausencias y mis malos momentos. A mi padre, que a pesar de la distancia siempre estuvo atento para saber cómo iba mi proceso. A ti Arita, que desde un principio hasta el día hoy sigues dándome ánimo para terminar este proceso.

Santiago Lucio

AGRADECIMIENTOS

A la Empresa Eléctrica Provincial de Cotopaxi S.A. por su apoyo al presente proyecto y confianza.

Marcos Armas

Índice de Contenidos

Índice de Contenidos.....	7
Índice de Tablas	14
Índice de Imágenes	16
Índice de Gráficos	19
RESUMEN	1
CAPÍTULO 1	2
INTRODUCCIÓN	2
1.1.- Planteamiento del problema.....	3
1.2.- Descripción de la solución Propuesta	5
1.3.- Justificación.....	7
1.4.- Objetivos	8
1.4.1.- Objetivo general.....	8
1.4.2.- Objetivos específicos	8
1.5.- Alcance	8
1.6.- Metodología	10
CAPÍTULO 2	11
MARCO TEÓRICO REFERENCIAL.....	11
2.1.- Herramientas de desarrollo	11
2.1.1.- Introducción	11
2.1.2.- Herramienta de Instrumentación Virtual.....	11
2.1.3.- Herramienta de Desarrollo Web.....	16
2.1.3.1.- Fases de desarrollo de una aplicación web	17
2.1.3.2.- PHP.	19
2.1.4.- Herramienta de Gestión de Base de Datos.....	27
2.1.4.1.- MySQL 1.2.17	29
2.1.4.2.- Conectores y Drivers.	31
2.1.4.3.- Administración de Almacenamiento	32
2.1.4.4.- Dimensiones máximas de las tablas:.....	34
2.1.4.5.- Ventajas:	34
2.1.4.6.- Inconvenientes:.....	35
2.1.4.7.- Tipos de compilación del servidor	36
2.1.4.8.- Licencia	37
2.2.- Ambientes de desarrollo.....	37
2.2.1.- NetBeans	38

2.2.1.1.-	Auto-completado y documentación de funciones PHP:.....	38
2.2.1.2.-	Generador de PHPDoc:.....	39
2.2.1.3.-	Auto-completado de código propio:.....	39
2.2.1.4.-	Soporte para CVS, Mercurial, Subversión:	40
2.2.1.5.-	Atajos de teclado:	40
2.2.1.6.-	Características Adicionales.....	41
2.3.-	Herramientas de software adicionales	41
2.3.1.-	Propel ORM.	42
2.3.2.-	JavaScript	44
2.3.3.-	AJAX.....	44
2.3.4.-	JQuery	45
2.3.5.-	ThickBox	45
2.4.-	Patrón Modelo Vista Controlador.	46
2.4.1.-	Modelo.	46
2.4.2.-	Vista.	47
2.4.3.-	Controlador.	47
2.5.-	Sistemas de Monitoreo y control (SCADA).	47
2.5.1.-	Características.	48
2.5.2.-	Prestaciones	49
2.5.3.-	Componentes de Hardware	50
2.5.4.-	Unidad central (MTU, Master Terminal Unit).....	51
2.5.4.1.-	Estaciones Remotas (RTU, Remote Terminal Unit)	51
2.5.4.2.-	Sistema de comunicación	53
2.5.4.3.-	Instrumentos de campo	54
2.5.5.-	Estructura de un sistema SCADA	54
2.5.6.-	Arquitectura de un sistema SCADA.....	55
2.5.6.1.-	Escalabilidad	55
2.5.6.2.-	Disponibilidad.....	55
2.5.6.3.-	Robustez.....	56
2.5.6.4.-	Prestaciones.....	57
2.5.6.5.-	Seguridad.....	57
2.5.6.6.-	Mantenibilidad.....	57
2.5.7.-	Estructura y componentes de un Software SCADA.	57
2.6.-	Metodología de desarrollo de Software.....	58
2.6.1.-	Metodologías Tradicionales.	59
2.6.2.-	Metodologías Ágiles.....	59

2.6.3.-	Selección de la metodología de desarrollo	61
2.6.3.1.-	Programación Extrema.....	61
	<i>Valores</i>	62
	<i>Principios</i>	62
	<i>Roles</i>	65
	<i>Proceso XP</i>	66
2.6.3.2.-	Metodología de desarrollo Scrum.....	67
	<i>Roles</i>	69
	<i>Estructura</i>	72
	<i>Reuniones</i>	74
	<i>Artefactos</i>	76
2.6.3.3.- Comparación entre XP y SCRUM y elección de la metodología a usar.	78
2.7.-	Sistemas de inteligencia de negocios.	79
2.7.1.-	Introducción	79
2.7.2.-	Componentes de Inteligencia de Negocios.....	82
2.7.2.1.-	DataWareHouse	82
2.7.2.2.-	OLAP Procesos de Análisis en Línea	88
2.7.2.3.-	DATAMINING (minería de datos).....	93
2.7.3.-	Beneficios de Inteligencia de Negocios	94
CAPÍTULO 3		99
SELECCIÓN DE LA SUITE DE INTELIGENCIA DE NEGOCIOS.....		99
3.1.-	Herramientas utilizadas en una Suite de Inteligencia de Negocios.....	99
3.1.1.-	OLAP	99
3.1.2.-	Scorecarding (marcador)	99
3.1.3.-	Dashboarding (Tablero de instrumentos).....	99
3.1.4.-	Información de visualización	100
3.1.5.-	Data Modeling.....	100
3.1.6.-	Data Warehouse (Inventario de datos)	101
3.1.7.-	Document Warehouse (almacén de documentos)	101
3.1.8.-	Datamining (Extracción de datos)	101
3.1.9.-	Text Mining (Extracción de texto).....	102
3.1.10.-	EIS – Sistemas de información ejecutiva.....	102
3.1.11.-	DSS – Decision Support Systems.....	102
3.2.-	Lista de criterios a considerar.	102
3.3.-	Suites de Inteligencia de Negocios.	103

3.3.1.-	Pentaho BI Suite	103
3.3.2.-	Jasper, de Jaspersoft.....	106
3.3.3.-	BIRT.....	109
3.3.4.-	Palo Suite de Jedox	111
3.4.-	Análisis.....	114
3.5.-	Resultados.	121
CAPÍTULO 4		122
DESARROLLO DEL SISTEMA		122
4.1.-	Introducción.....	122
4.2.-	Desarrollo del Subsistema de obtención de Datos.....	122
4.2.1.-	Visión general	122
4.2.2.-	Análisis de requerimientos.....	123
4.2.3.-	Requerimientos Funcionales.....	123
4.2.4.-	Requerimientos no funcionales.....	123
4.2.5.-	Análisis.....	123
4.2.6.-	Duración del sprint	125
4.2.7.-	Primera iteración.....	126
4.2.8.-	Exploración	126
4.2.8.1.-	Capturar datos del sistema principal.....	126
4.2.8.2.-	Acondicionar valores y calcular potencias generadas.	130
	<i>Acondicionamiento del valor de corrientes.</i>	130
	<i>Acondicionamiento del valor de los voltajes.</i>	131
	<i>Calculo de las potencias.</i>	131
	<i>Acondicionamiento de niveles de agua.</i>	132
4.2.9.-	Carta de Burndown.	133
4.2.10.-	Revisión	134
4.2.11.-	Segunda iteración.	135
4.2.12.-	Almacenar en base de datos	136
4.2.12.1.-	Creación de la base de datos.	136
4.2.12.2.-	Desarrollo de la función Base de datos	137
4.2.13.-	Carta de Burndown.	140
4.2.14.-	Revisión	141
4.3.-	Desarrollo del Subsistema Monitor	141
4.3.1.-	Visión general	141
4.3.2.-	Análisis de requerimientos.....	142
4.3.2.1.-	Requerimientos Funcionales.	142

4.3.2.2.-	Requerimientos no funcionales.....	143
4.3.3.-	Análisis.....	143
4.3.4.-	Duración del Sprint	145
4.3.5.-	Primer Sprint.....	145
4.3.5.1.-	Planificación.....	145
4.3.5.2.-	Arquitectura	146
	<i>Interfaz Gráfica</i>	146
	<i>Estilo del Texto</i>	149
	<i>Estructura de archivos</i>	149
	<i>Nomenclaturas</i>	150
4.3.5.3.-	Administración de menús y opciones.	152
	<i>Página contenedora del subsistema Monitor</i>	154
	<i>Página de administración de Menús</i>	154
	<i>Página de administración de Opciones</i>	157
4.3.5.4.-	Administración de usuarios y grupos de usuarios.....	158
	<i>Página de administración de Grupos de Usuarios</i>	159
	<i>Página de administración de Usuarios</i>	160
	<i>Administración de permisos de grupos de usuarios a menús y opciones</i>	161
	<i>Página de administración de permisos de grupos de usuarios a menús y opciones</i>	162
	<i>Página de inicio de sesión</i>	163
4.3.5.5.-	Administración de estructura geográfica.....	163
	<i>Página de administración de Regiones</i>	164
	<i>Página de administración de Provincias</i>	165
	<i>Página de administración de Ciudades</i>	165
	<i>Página de administración de Centrales</i>	165
4.3.5.6.-	Carta de Burndown.....	166
4.3.5.7.-	Revisión.....	168
4.3.6.-	Segundo Sprint.	168
4.3.6.1.-	Planificación.....	168
4.3.6.2.-	Administración de Generadores (GPU's).....	171
4.3.6.3.-	Administración de Tanques.	172
4.3.6.4.-	Administración de la Configuración de Memoria.	174
4.3.6.5.-	Administración de la planificación de mantenimiento de generadores. 175	
4.3.6.6.-	Alarmas de aviso de eventos importantes.	177

4.3.6.7.-	Carta de Burndown.....	177
4.3.6.8.-	Revisión.....	179
4.3.7.-	Tercer Sprint.....	179
4.3.7.1.-	Planificación.....	179
4.3.7.2.-	Generar gráficos dinámicos actualizables durante el transcurso del tiempo del estado de generación eléctrica por grupo de generación.....	181
4.3.7.3.-	Generar gráficos históricos de generación eléctrica con filtros.....	187
4.3.7.4.-	Generar un gráfico de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.....	189
4.3.7.5.-	Generar gráficos dinámicos de niveles de agua por tanque actualizables durante el transcurso del tiempo.....	190
4.3.7.6.-	Carta de Burndown.....	193
4.3.7.7.-	Revisión.....	194
4.4.-	Implementación del subsistema de Análisis de Datos.....	195
4.4.1.-	Visión general.....	195
4.4.2.-	Análisis de requerimientos.....	195
4.4.2.1.-	Requerimientos Funcionales.....	196
4.4.2.2.-	Análisis.....	196
4.4.2.3.-	Duración del Sprint.....	197
4.4.2.4.-	Primer Sprint.....	197
	<i>Planificación.....</i>	197
	<i>Desarrollo de la base de datos para el cubo.....</i>	198
	<i>Instalación y configuración de la Suite Pentaho.....</i>	200
	<i>Creación de un proceso de ETL para la carga de datos.....</i>	203
	<i>Crear transformación para la tabla dim_central.....</i>	204
	<i>Crear Transformación para la tabla dim_generador.....</i>	205
	<i>Crear Transformación para la tabla dim_gografica.....</i>	205
	<i>Crear Transformación para la tabla dim_variable.....</i>	207
	<i>Crear Transformación para la tabla dim_tiempo.....</i>	207
	<i>Crear Transformación para la tabla fact_generacion.....</i>	210
	<i>Crear el trabajo para ejecutar todas las transformaciones.....</i>	211
	<i>Carta de Burndown.....</i>	212
	<i>Revisión.....</i>	213
4.4.2.5.-	Segundo Sprint.....	214
	<i>Planificación.....</i>	214
	<i>Creación de Metadatos.....</i>	214

<i>Creación del cubo.</i>	217
<i>Desarrollo de reportes.</i>	222
<i>Desarrollar reporte de curvas de potencia generada hora a hora por cada día de cada Generador (cuatro en total) en kw por cada fase.</i>	222
<i>Desarrollar reporte de curvas de potencia generada hora a hora por cada día de cada generador (cuatro en total) en kw por todas las fases.</i>	224
<i>Desarrollar reporte de curvas de la potencia total de todos los generadores en kw/h.</i>	225
<i>Desarrollar reporte de curvas de variación de voltaje de barras por cada grupo y fase.</i>	226
<i>Desarrollar reporte de Curvas de variación de corriente en función del tiempo de cada grupo y fase.</i>	227
<i>Carta de Burndown</i>	228
<i>Revisión</i>	229
CAPÍTULO 5	230
CONCLUSIONES Y RECOMENDACIONES.	230
5.1.- Conclusiones.....	230
5.2.- Recomendaciones.....	231
BIBLIOGRAFÍA.	233
ANEXO A	236
APÉNDICE A.....	248
APÉNDICE B.....	249

Índice de Tablas

Tabla 2.1: Comparativa entre LabView y HP-EE.....	13
Tabla 2.2 Comparativa entre MySql y PostgreSQL.....	27
Tabla 2.3: Comparativa entre Metodologías Ágiles y Tradicionales	60
Tabla 3.1: Comparativa entre las características no funcionales	114
Tabla 3.2: comparativa de características de diferentes suites de BI	117
Tabla 3.3 Características importantes en versión community open source.....	120
Tabla 4.1: Pila del producto a desarrollar	125
Tabla 4.2 Pila del sprint con las actividades a realizar	126
Tabla 4.3: Información a utilizarse para elaborar la carta de Burndown	133
Tabla 4.4 Pila del producto después de la primera iteración	135
Tabla 4.5 Pila del sprint.....	135
Tabla 4.6 Pila del producto después de la primera iteración.....	140
Tabla 4.7 Pila de producto para el desarrollo del subsistema.....	144
Tabla 4.8 Pila del sprint con las actividades a realizar	145
Tabla 4.9 Archivos CSS usados en el proyecto	149
Tabla 4.10 Nomenclatura usada en Controles Visuales de las páginas HTML.....	151
Tabla 4.11 Nomenclatura de clases, ejemplo tabla PERSONA	152
Tabla 4.12: Información a utilizarse para elaborar la carta de Burndown	167
Tabla 4.13 Pila del producto después del primer sprint	169
Tabla 4.14 Pila del Segundo Sprint	170
Tabla 4.15: Información a utilizarse para elaborar la carta de Burndown	178
Tabla 4.16 Pila del Producto	180
Tabla 4.17 Pila del sprint	181

Tabla 4.18 Carta de Burndown del Último sprint.....	193
Tabla 4.19 Pila del Producto	197
Tabla 4.20 Pila del sprint con las actividades a realizar	198
Tabla 4.21 Rutas de extracción de herramientas	200
Tabla 4.22 Descripción de los módulos de administration-console y bi-server	201
Tabla 4.23 Registros de Ejemplo.....	208
Tabla 4.24: Información a utilizarse para elaborar la carta de Burndown	213
Tabla 4.25 Pila del Sprint	214
Tabla 4.26 Significado de cada icono de la estructura.	219
Tabla 4.27: Información a utilizarse para elaborar la carta de Burndown.....	228

Índice de Imágenes

Imagen 2.1: Pantalla Frontal del programa “Grabar.VI”	14
Imagen 2.2: Diagrama de Bloques del programa “Grabar.VI”	15
Imagen 3.1: Ejemplo de presentación de datos con Pentaho	100
Imagen 3.2: Pentaho Data Integration	105
Imagen 3.3: Pantalla de trabajo de JasperETL	107
Imagen 4.1 Pantalla Principal del Sistema	124
Imagen 4.2 Parámetros de entrada	127
Imagen 4.3 Código desarrollado para el programa SubVI	128
Imagen 4.4: Programación del sistema principal	129
Imagen 4.5: Diagrama de Bloque de Cálculo de Potencias	132
Imagen 4.6 Sub VI “Graba Base”	132
Imagen 4.7 Diagrama de Bloques de “grabar en mysql.vi”	138
Imagen 4.8: Diagrama de bloques de “Generador de comandos SQL.vi”	139
Imagen 4.9 “Base de datos.vi”	139
Imagen 4.10 Diseño base del sistema	154
Imagen 4.11 Administración de Menús	155
Imagen 4.12 Formulario de edición para agregar o editar un registro	156
Imagen 4.13 Formulario de ingreso de datos	157
Imagen 4.14 Formulario del área de despliegue de contenido	159
Imagen 4.15 Formulario del área de edición	160
Imagen 4.16 Formulario de Administración de usuarios	160
Imagen 4.17 Formulario de Administración usuarios, opción “Nuevo Usuario”.....	161
Imagen 4.18 Página de asignación de grupos de usuarios a menús y opciones	162
Imagen 4.19 Área de despliegue de información	166
Imagen 4.20 Área de edición	166

Imagen 4.21 Formulario de Administración de generadores	171
Imagen 4.22 Administración de generadores, opción “Nuevo Generador”	172
Imagen 4.23 Formulario de Administración de Tanques	173
Imagen 4.24 Administración de Tanques, opción “Nuevo Tanque”	173
Imagen 4.25 Formulario de Configuración Posición de Memorias GPU	174
Imagen 4.26 Configuración Posición de Memorias GPU, “Nueva Posición”	175
Imagen 4.27 Formulario de Mantenimiento de Generadores	176
Imagen 4.28 Formulario de Mantenimiento Generadores, “Nuevo Generador”	176
Imagen 4.29 Alarmas de Aviso de eventos importantes	177
Imagen 4.30 Muestra de niveles de las tres corrientes de un GPU	184
Imagen 4.30 Estado del Grupo Uno (GPU 1) desde la página principal	186
Imagen 4.31 Estado del Grupo Uno (GPU 1), link “Abrir en otra página”	187
Imagen 4.32 Formulario de búsqueda con varios filtros	188
Imagen 4.33 Resultado del gráfico	188
Imagen 4.33 Resultado del Punto	190
Imagen 4.34 Chart devuelto por la función	192
Imagen 4.35 Gráfico Final del Desarrollo	193
Imagen 4.36 Nueva conexión de a la base de datos “cubo_central”	201
Imagen 4.37 Consola de usuario	203
Imagen 4.38 Programación desarrollada para realizar esta transformación	204
Imagen 4.39 Programación tabla dim_central	205
Imagen 4.40 Programación tabla dim_geográfica	206
Imagen 4.41 Programación tabla dim_variable	207
Imagen 4.42 Programación tabla dim_tiempo	209
Imagen 4.43 Programación Fact_generacion	210
Imagen 4.44 Borrado de tabla Fact_generacion	211

Imagen 4.45 Modelamiento obtenido por Pentaho Metadata Editor	216
Imagen 4.46 Configuración usada para la publicación	216
Imagen 4.47 Estructura para realizar reportes	217
Imagen 4.48 Ventana de conexión a la base de datos.....	218
Imagen 4.49 Definición del cubo.	219
Imagen 4.50 Conexión al repositorio de Pentaho	220
Imagen 4.51 Solución del repositorio en donde se debe almacenar	220
Imagen 4.52 Consola del Usuario, esquema del Cubo	221
Imagen 4.53 Pivot Table	221
Imagen 4.54 Área de parámetros del reporte	223
Imagen 4.55 Área del reporte	224
Imagen 4.56 Área del reporte	225
Imagen 4.57 Área del reporte.....	226
Imagen 4.58 Reporte Voltaje por Generador	227
Imagen 4.59 Reporte Corriente por Generador	228

Índice de Gráficos

Gráfico 2.2: Código de una página HTML con PHP	22
Gráfico 2.3: Código recibido por el navegador	23
Gráfico 2.4 Visualización el navegador	23
Gráfico 2.5: Diagrama general de un Sistema y sus componentes	31
Gráfico 2.6: Autocompletado PHP	38
Gráfico 2.7: Generador de PHPDoc	39
Gráfico 2.8: Auto-Completado de código propio	39
Gráfico 2.9: Código PHP con su historial de cambios	40
Gráfico2.10: Flujo de actividad de la arquitectura MVC	46
Gráfico 2.11: Estructura básica de un sistema SCADA a nivel de hardware	50
Gráfico 2.12: Estructura básica de un sistema SCADA	54
Gráfico 2.13: Redundancia de Dispositivos en un sistema SCADA	56
Gráfico 2.14: Prácticas en Programación Extrema	62
Gráfico 2.15: Fases de la estructura Scrum	72
Gráfico 2.16: Reuniones dentro de SCRUM, Fases de los Sprints	74
Gráfico 2.17: Modelo Integral de una Solución BI	82
Gráfico 2.18: Esquema De Funcionamiento De Un DataWarehouse	87
Gráfico 2.19: Modelo de un Ambiente OLAP	89
Gráfico 2.20: Arquitecturas OLAP	90
Gráfico 3.1: Módulos de Palo Suite de Jedox	111
Gráfico 4.1: Estructura del Arreglo	128
Gráfico 4.2: Avance del Proyecto	132
Gráfico 4.3 Avances en la Implementación del Sistema	134
Gráfico 4.4 Diagrama de tablas.....	137

Gráfico 4.5 Arquitectura de la conexión a la base de datos	138
Gráfico 4.6 Estructura de implementación del subsistema monitor.....	140
Gráfico 4.7: Avance del Proyecto	141
Gráfico 4.8: Diseño para la página principal.....	147
Gráfico 4.9: Prototipo del área de edición	148
Gráfico 4.10 Representación del Subsistema	153
Gráfico 4.11 Representación de las clases grupo de usuario y usuario	158
Gráfico 4.12 Relación entre la clase Grupo de Usuario y Permisos	161
Gráfico 4.13 El proceso de autenticación	163
Gráfico 4.14 Clases de la función es administrar estructura geográfica	164
Gráfico 4.15: Avance del Proyecto	167
Gráfico 4.16 Administración de generadores, GPU's	171
Gráfico 4.17 Administración de tanques	172
Gráfico 4.18 Configuración de Memoria, relación con GPU's	174
Gráfico 4.19 Diagrama de objetos a implementar en Generador o Tanque	175
Gráfico 4.20: Avance del Proyecto	178
Gráfico 4.21: Clase LecturaControlador	182
Gráfico 4.22: Clase LecturaTanqueControlador	191
Gráfico 4.23 Carta de Burndown final para el subsistema Monitor	194
Gráfico 4.24 Diagrama de implementación del subsistema monitor.....	195
Gráfico 4.25 Diagrama Entidad Relación base datos del cubo central	199
Gráfico 4.26 Carta Burndown final del primer sprint en el tiempo planeado	213
Gráfico 4.27 Estructura planteada “Desarrollo de base de datos para el cubo”	218

RESUMEN

El proyecto desarrollado brinda una solución a una problemática presentada en la Empresa Eléctrica Provincial de Cotopaxi (ELEPCO) en cuanto al monitoreo y control de sus centrales hidroeléctricas, ELEPCO contaba con un sistema pero este no cubría sus necesidades, se tenía poco aprovechamiento de los datos históricos, se presentaban reportes de generación no muy fiables puesto que los hacían de forma manual en una hoja de cálculo, la solución propuesta fue desarrollar tres subsistemas o programas que agilizan el monitoreo de los equipos de generación, el primer subsistema se integra a un sistema SCADA de la empresa y acondiciona los datos obtenidos para luego almacenarlos en una base de datos, el segundo subsistema es una interface web donde el usuario puede revisar los valores de generación eléctrica y niveles de agua, y por último un subsistema de análisis de datos históricos que en cambio ayuda a que el usuario pueda obtener reportes históricos de generación, para este subsistema se utilizó una herramienta de inteligencia de negocios por su capacidad de manejo y consolidación de grandes cantidades de datos, la metodología utilizada en este proyecto fue SCRUM y fue necesario tres ciclos completos del mismo, uno para cada subsistema. El resultado de este proyecto fue apreciado por ELEPCO porque facilitó el monitoreo de los equipos de campo, inclusive lo pueden realizar desde una ubicación remota a la central hidroeléctrica, aumentó la fiabilidad de los reportes de generación, y aprovechó los datos históricos.

CAPÍTULO 1

INTRODUCCIÓN

La Empresa Eléctrica Provincial Cotopaxi ELEPCO S.A.-, según la Ley de Régimen del Sector Eléctrico, es una institución dependiente del Ministerio de Electricidad y Energías Renovables, que tiene como funciones principales la generación, distribución y comercialización de energía eléctrica, sin embargo, desde antes de la promulgación de la mencionada Ley, tenía como parte constitutiva de su sistema las centrales de generación de Illuchi, cuya área operativa debe separarse tanto administrativa como técnicamente de la empresa de distribución, por disposición de la Ley.

La Empresa Eléctrica Provincial Cotopaxi S.A. tiene una planta generadora de electricidad ubicada en el sector de Illuchi, sitio localizado a 30 Km. al Este de la ciudad de Latacunga, esta Central generadora posee 4 grupos hidráulicos, cada uno está conformado por una turbina tipo Pelton, el alternador y los equipos de control y protección, entre los cuales consta un GENERATOR PROTECTION UNIT (G. P. U.), que es una unidad que en base de electrónica de estado sólido, cumple con las siguientes funciones:

- Protección contra sobrecorrientes.
- Protección contra sobre y bajos voltajes.
- Protección contra motorización del generador.
- Almacenamiento en memoria de corta duración de los parámetros eléctricos producidos por los eventos anormales.

- Medición de potencia generada (Kilovatios - kW).
- Medición de energía generada (Kilovatios hora - kWh).
- Voltaje generado (Kilovoltios – kV).
- Corriente generada (Amperios – A).

Esta central se encuentra dentro del sistema interconectado nacional, que es a donde se dirige toda la energía que generan las plantas de todo el país, por lo tanto la energía generada por la planta alimenta a este sistema. Este sistema interconectado es controlado por el CENACE (Centro Nacional de Control de Energía), el cual posee medidores con acceso a internet en cada una de las empresas generadoras, por medio de los cuales toman datos de la cantidad que está generando cada planta en GW/h (Giga Wattios por Hora), para posteriormente depositar en las cuentas de cada empresa el valor en dinero de estos Giga Wattios.

La Dirección de Generación tiene a su cargo el control, supervisión y monitoreo de todos los parámetros eléctricos antes mencionados, que tienen relación con la producción de energía eléctrica y otros que son necesarios para la operación del sistema, como el acopio de agua en los reservorios hasta su utilización en el funcionamiento de la maquinaria para producir la electricidad.

1.1.- Planteamiento del problema

Tal como se indicó anteriormente, cada grupo de generación tiene un puerto RS232, y los operadores se conectan mediante un cable serial y un

software de ABB, empresa fabricante de los GPU's; para tomar datos del estado actual del generador. Este software no posee una función de captura automática o programada, ni de almacenamiento de los datos extraídos por lo que los operadores prefieren tomar estos datos manualmente desde los displays situados en la parte frontal de los GPU's, y luego lo ingresan en una hoja de Excel a manera de bitácora, estos archivos son entregados al Jefe de la planta para que los almacene en su computador y envíe a la gerencia ubicada en Latacunga por medio físico sea en un CD o en un Pen Drive.

Con esta forma de trabajo, explica el Jefe de la planta; se tiene algunos problemas, entre ellos:

- Distorsión de la información a causa de la toma errada de los datos de parte de los operadores.
- Poco aprovechamiento de los datos históricos.
- La falta de un sistema informático para procesamiento de la información recolectada y para el análisis de los datos.
- Dificultad y tardanza de la obtención de los datos actuales desde la gerencia.
- El lugar donde el operador debe tomar las lecturas de los generadores es una zona de alto riesgo auditivo y de alto voltaje, por lo que se desea disminuir el tiempo que el operador debe permanecer en este sitio.

Los archivos de Excel que llegan a la gerencia son usados para comparar el total generado computado por la empresa contra el cómputo del CENACE para el pago mensual de la energía generada. Además, se utilizan para hacer una

planificación de cuanta energía se debe comprar en un determinado mes al sistema interconectado porque la cantidad generada no es constante durante todo el año.

Por otro lado, al ser un sistema hidráulico de generación, es necesario también tener información de los niveles de agua en el tanque de presión (tanque reservorio de agua), en este se ha instalado un sensor ultrasónico que por vía de radio envía este parámetro hasta la casa de máquinas en donde se procesa en otra hoja de Excel. Cabe destacar que los reservorios de agua se encuentran a una distancia de 2 Km cuesta arriba de la planta de generación Illuchi en donde no existe un operario fijo.

El requerimiento de la Dirección es que se pueda acceder a los datos tanto de generación como de niveles de agua de manera remota y desde cualquier lugar, para poder tomar decisiones ante variaciones peligrosas de estos niveles, además que este sistema ayude al monitoreo de la central de generación y ayude a la planificación de la compra y venta de energía.

1.2.- Descripción de la solución Propuesta

Para tomar los datos transferidos tanto de generación eléctrica como de niveles de agua se realizará una interface en Labview, donde se los almacenará en una base de datos. Esta interface va a permanecer siempre prendida y configurada para que realice una lectura periódica en cada GPU.

Una vez realizada la interface para obtener los datos, se desarrollará un sistema de supervisión de niveles de generación eléctrica con el lenguaje de programación PHP junto con FLEX en donde se desplegarán los datos actuales de generación con una interfaz agradable para el usuario tanto en texto como gráficamente. Además de otras herramientas administración de eventos importantes que deben ser atendidos por el operador, entre estos eventos se tiene:

- Una baja importante en los niveles de agua de los reservorios.
- Necesidad de mantenimiento.
- Bajas en la generación de energía por evento desconocido.

Para resolver el problema de la baja utilización de los datos históricos se construirá un cubo de información junto a una colección de reportes predefinidos que mostrarán el comportamiento que ha venido teniendo la generación, esta solución permitirá al usuario analizar y tomar decisiones para planificar la compra de mayor o menor cantidad de energía al sistema interconectado. En el portal se mostrarán estos reportes mediante niveles de acceso por usuarios, entre los principales reportes se tienen:

- Curva de potencia generada hora a hora por cada día de cada grupo (cuatro en total) en kw.
- Curva de la energía generada en kw/h.
- Curvas de variación de voltaje de barras.

- Curvas de variación de corriente en función del tiempo de cada grupo.
- Hora de conexión y desconexión de los generadores.
- Reportes de fallas de cada grupo.

Los datos históricos que se tienen en las hojas de Excel serán ingresados a la base de datos por una interface web desarrollada en PHP donde se subirán los archivos de Excel y se migrará esa información a la base de datos mediante un proceso automático.

1.3.- Justificación

El proyecto tiene gran importancia por tener varios temas que ameritan investigación como por ejemplo el protocolo de conexión entre los GPU's y el computador, la forma de conexión entre estos, de igual manera la captura de datos de niveles de agua.

También se analizará diferentes plataformas de Inteligencia de Negocios para la construcción del cubo y la publicación de reportes con tablas dinámicas y gráficos dinámicos.

La solución que se propone es una innovación que no solo va a ser utilizado en esta planta sino que después se instalará en dos plantas más que son propiedad de ELEPCO S.A. Además por ser un tema donde se desarrollará una solución integral para el monitoreo, supervisión y análisis histórico de datos de generación eléctrica puede llegar a interesar a otras empresas del mismo medio en todo el país.

1.4.- Objetivos

1.4.1.-Objetivo general

Desarrollo e implementación de aplicaciones de supervisión y control de generación eléctrica y análisis de datos históricos para la Empresa Eléctrica Provincial Cotopaxi S.A. para disminuir el tiempo del proceso de análisis de datos de generación, desde la captura de datos hasta el análisis de los mismos, proveyendo herramientas y reportes que permitan minimizar las pérdidas económicas por fallas en los reportes que sirven para el análisis.

1.4.2.-Objetivos específicos

- Agilizar el proceso de análisis de datos de generación eléctrica.
- Disminuir el riesgo de error en la captura de los datos de generación.
- Aumentar la autenticidad de los reportes evitando la intervención humana en los procesos de cálculo.
- Documentar los requerimientos usando la norma IEEE 830.
- Usar la metodología de desarrollo ágil SCRUM.
- Desarrollar el sistema de supervisión y monitoreo con PHP.

1.5.- Alcance

Se comenzará construyendo la interface para tomar datos de los GPU's con Labview, se deberá investigar el protocolo de conexión a los GPU's. El programa tendrá un temporizador para que cada cierto tiempo se tomen las lecturas que cada GPU este realizando en ese momento del generador y las almacene en una base de datos MySql.

El sistema de supervisión y monitoreo de generación eléctrica será desarrollado en PHP con las siguientes partes:

- Página de inicio de sesión.
 - o Página principal.- Contendrá un resumen del estado actual de los generadores y el nivel de agua.
 - o Detalles de los generadores.- Contendrá un informe completo del estado actual de un generador seleccionado.
 - o Detalle del nivel de agua.- Incluirá un informe completo del nivel actual de agua.
 - o Alarmas.- Se mostrarán alarmas de eventos importantes.
 - o Administración de usuarios.
 - o Administración de permisos
 - o Administración de Configuraciones.

Se investigará diferentes suites de Inteligencia de Negocios para la construcción del sistema de análisis de datos históricos y se seleccionará al que mejores beneficios brinde para el proyecto. Para el análisis se desarrollará una matriz de comparación ponderando varias cualidades entre las cuales se tiene:

- Proveedor
- Tecnología.
- Funcionalidades.
- Funcionalidad adicional.
- Costo.

A partir de la selección de la suite de Inteligencia de negocios, se construirá un cubo de información con los datos almacenados en la base de datos de

generación eléctrica. Se diseñará una colección de reportes, tablas cruzadas, reportes dinámicos según los requerimientos de ELEPCO y a los que el usuario podrá acceder dependiendo el nivel de acceso que tenga.

Se realizará la implementación del sistema en las oficinas de ELEPCO S.A. y se capacitará al personal cuando el desarrollo haya concluido.

1.6.- Metodología

Para el desarrollo de este proyecto se usará “Scrum” porque se tienen tres productos que pueden ser construidos en forma secuencial y pueden ser puestos en producción en diferentes puntos en el tiempo siguiendo el orden de dependencia entre estos. Y porque el proyecto debe ser realizado en el menor tiempo posible por requerimiento de ELEPCO S.A.

Scrum es una metodología ágil que consiste en formar un equipo multidisciplinario, los miembros del equipo deben cubrir todo el conocimiento para el desarrollo. El equipo intervendrá en todo el desarrollo, pero a diferencia de otras metodologías el equipo desarrollará el sistema módulo por módulo en el momento que se requiere, es decir; todas las fases como requisitos, análisis, codificación, pruebas, integración se realizan en el momento que se necesite en el proyecto.

CAPÍTULO 2

MARCO TEÓRICO REFERENCIAL.

2.1.- Herramientas de desarrollo

2.1.1.-Introducción

Para este proyecto se definen 3 módulos que deben ser desarrollados en herramientas distintas por la utilidad final que van a tener, los cuales son los siguientes: Subsistema Obtención de Datos, Subsistema Monitor y Subsistema Analizador de Datos Históricos y en conjunto estos tres sistemas integrarán el Sistema De Monitoreo y Control ELEPCO.

2.1.2.-Herramienta de Instrumentación Virtual

Subsistema de Obtención de datos de los GPU's: Para el desarrollo de este subsistema se necesita una herramienta de instrumentación virtual que es la parte de la electrónica, principalmente analógica, que se encarga del diseño y manejo de los aparatos electrónicos y eléctricos, sobre todo para su uso en mediciones.

La instrumentación electrónica se aplica entre otras cosas para el censado y procesamiento de la información proveniente de variables físicas y químicas, en el caso del sistema de monitoreo esta información proviene de los GPU's, a partir de las cuales realiza el monitoreo de procesos, empleando dispositivos y tecnologías electrónicas.

Una de las tendencias más utilizadas en la instrumentación es la instrumentación virtual, cuya idea es sustituir elementos físicos (hardware) por

otros Virtuales (software), que ejecute un programa específico que simule al elemento físico, este programa se comunica con los dispositivos para configurarlos y leer sus medidas.

Las ventajas de la instrumentación virtual son: es capaz de automatizar las medidas, procesado de la información, visualización y actuación remotamente, etc. En el mercado existen varios programas especializados en este campo, de una amplia gama los comúnmente utilizados LabVIEW de National Instruments y HP-VEE de Agilent Technologies ¹.

Realizando una comparación entre estas dos herramientas de instrumentación virtual se puede decir que son de similares características, las dos trabajan con interfaz gráfica muy amigable para el usuario, su entorno dispone de librerías matemáticas, para el análisis de datos y de los drivers de control de la mayoría de equipos existentes, pero si se considera el tamaño de la aplicación final, la mejor opción es HP-VEE, si en cambio se considera la velocidad LabVIEW es el ganador, si de precisión se trata, las dos opciones son sobresalientes pues presentan un mínimo margen de error, si se considera las capacidades de control de dispositivos, HP-VEE no permite la gestión de tarjetas de adquisición pues necesita paquetes adicionales, finalmente las librerías de adquisición y procesado de LabVIEW son mucho más extensas y potentes, permitiendo también conexiones con bases de datos específicas. ²

¹ http://es.wikipedia.org/wiki/Instrumentaci%C3%B3n_electr%C3%B3nica

² http://books.google.com.ec/books?id=7Poo_SxtYKEC&pg=PA11&lpg=PA11&dq=caracteristicas+de++hp+vee&source=bl&ots=Wkslxc4MmA&sig=_y3Bkvvys-BbHomnhJBcRGsGkfc&hl=es&ei=Df1UTNzyPML38Aa2yLjhCA&sa=X&oi=book_result&ct=result&resnum=4&ved=0CDQQ6AEwAw#v=onepage&q&f=false

Tabla 2.1: Comparativa entre LabView y HP-VEE.

Software	LabVIEW	HP-VEE
Tiempo de ejecución	Tiempo de bus: 112.37 s Tiempo de Procesado: 12.36 s TOTAL: 125 s	Tiempo de bus: 101 s Tiempo de Procesado: 70 s TOTAL: 171 s
Error cuadrático medio	De módulo: 0.0185 De fase: 0.3171	De módulo: 0.0035 De fase: 0.2422
Tamaño de la aplicación	Fichero .VI 375.381 bytes	Fichero .vee 92.070 bytes
Tipo de programación	Gráfica	Gráfica
Programas .EXE	SI	NO
Ampliación de librerías	CIN (Code Interfase Node)	HP-BASIC
Librerías de procesado	Amplia	Limitada

Considerando todos los puntos analizados, se podría utilizar cualquiera de las dos opciones dado que no se van a explotar todas las bondades de estos programas, únicamente se utilizará su capacidad de obtener datos, en este caso se va a utilizar LabVIEW 8.5 puesto que ELEPCO tiene compradas las licencias del software y como un punto a favor posee personal que se encuentra capacitado en el manejo de este software, lo cual puede ser un apoyo en el momento del desarrollo y resultaría más fácil de capacitarlo para que realice mantenimientos o actualizaciones al módulo final.

2.1.2.1.- LABVIEW 8.5

Sus iniciales significan Laboratory Virtual Instrument Engineering Workbench, Es un entorno gráfico utilizado entre otras cosas para el desarrollo de aplicaciones sofisticadas de medida, pruebas y control usando íconos gráficos e

intuitivos y cables que parecen un diagrama de flujo, desde la adquisición de datos hasta el control remoto de equipos electrónicos, su entorno dispone de cientos de bibliotecas integradas para análisis avanzado y visualización de datos, para su análisis y de los drivers de control de la mayoría de equipos existentes, todo para crear instrumentación virtual (VI).

En general, un programa de VI consta de dos partes bien diferenciadas, el Panel Frontal (Front Panel) y el Diagrama de Bloques (Block Diagram).

El panel frontal es la interface del programa con el usuario, en él están representadas todas las entradas y salidas del programa. Por analogía a un instrumento real, las entradas del panel frontal se llaman controles y las salidas indicadores.

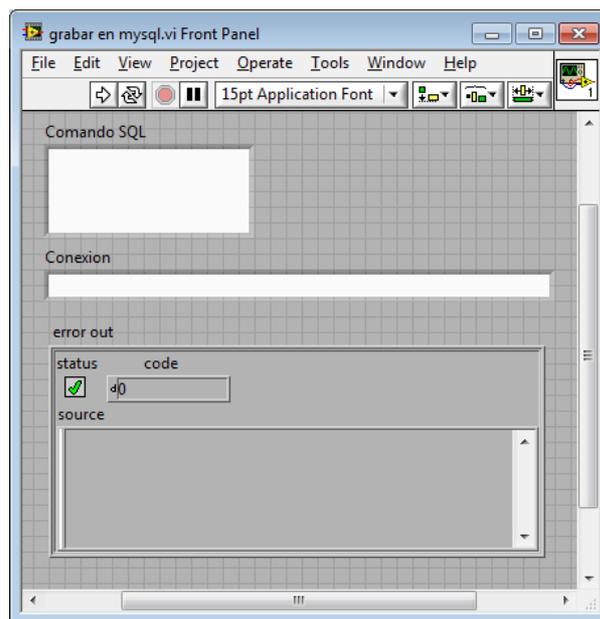


Imagen 2.1: Pantalla frontal del programa “Grabar en mysql.VI”.

El diagrama de bloques es el código de programación escrito en lenguaje gráfico. Los distintos componentes del diagrama de bloques son los nodos del programa. Los componentes están interconectados unos con otros. Estas interconexiones definen el flujo de datos en el diagrama de bloques.

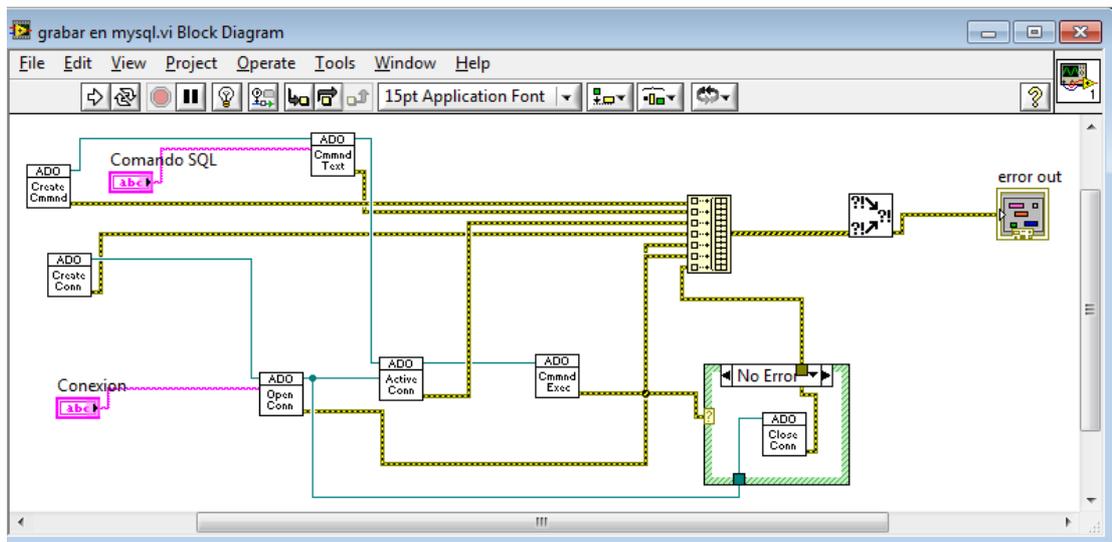


Imagen 2.2: Diagrama de bloques del programa “Grabar.VI”

2.1.2.2.- Características:

- **Programación Más Rápida** : Programación gráfica, LabVIEW posee almacenado un sinnúmero de controles que se encuentran programados previamente, lo cual permite programar con bloques de función gráficos, es decir únicamente arrastrar el control previamente programado y unirlo con el resto de botones programados en lugar de escribir líneas de texto.
- **Representación de Flujo de Datos**. Permite desarrollar, dar mantenimiento y comprender el código fácilmente con una representación intuitiva en diagramas de flujo.
- **Integración de Hardware con LabVIEW**: Permite conexiones a cualquier instrumento o sensor con bibliotecas integradas y miles de controladores de instrumentos, lo cual es la principal función que se busca en este proyecto, puesto que esta característica del LabVIEW es la que se utilizará para permitir

la conexión de los radios (Spread Spectrum) con el sistema de monitoreo para que nos provea de la información de generación necesaria para que este sistema funcione.

- **Reutilización de Código:** Permite integrar código basado en texto y DLLs o incorporar fácilmente archivos .m originales y de otras empresas, es decir, se pueden crear subprogramas en forma de DLLs y reutilizarlos en otros proyectos, ya que para adjuntándolos únicamente basta arrastrarlos dentro del proyecto y conectarlos al resto de botones programados previamente.
- **LabSQL:** LabSQL es un conjunto de VIs que utiliza la colección de objetos de ADO en LabVIEW para permitir la conexión a casi cualquier base de datos, realizar consultas SQL, manipular los registros, etc. Básicamente se trata de una colección de VIs que actúa como contenedores para las propiedades y métodos ADO.³ Soporta todas las bases de datos que tengan driver ODBC.

2.1.3.- Herramienta de Desarrollo Web

Módulo de presentación de la información (sitio web)

En este módulo se va a realizar todo el diseño del portal web que va a contener la información ya procesada de todos los módulos de este proyecto, para esto se debe tener en claro las fases de un desarrollo web, así como los lenguajes de programación usados, que son muy extensos y variados, y por ello es necesario tener herramientas específicas para cada una de ellas. A

³ <http://jeffreytravis.com/lost/labsql.html>

continuación las principales herramientas existentes para poder desarrollar un proyecto web.

2.1.3.1.- Fases de desarrollo de una aplicación web

- *Diseño:*

Esta fase consiste en crear esbozos sobre cómo va a quedar el sitio web final, es únicamente un esqueleto o una demo del producto final que se lo pudiera realizar en cualquier herramienta gráfica, como Photoshop, GIMP o Inkscape, o inclusive Paint esta fase es más utilizada en páginas web que van a ser diseñadas para ventas.

- *Maquetación HTML/CSS /Programación servidor:*

Es la fase en donde se convierten los esbozos creados en la fase anterior en plantillas HTML, su respectiva hoja de estilos, y las imágenes usadas. En este proyecto no van a existir únicamente páginas HTML sino una combinación de HTML con programación que puede ser: PHP, ASP .NET, Perl, etc.⁴

En vista de que el proyecto se encuentra en un lineamiento de la utilización de software libre, se van a analizar dos de los mejores software existentes para programación web: PHP y Perl.

Tanto PHP como Perl son dos lenguajes que pueden ser ejecutados como scripts CGI que es una interfaz para hacer posible la interacción entre una página y el servidor web, mediante la invocación de aplicaciones a las que se les pasan

⁴ <http://www.maestrosdelweb.com/editorial/herramientas-adecuadas-para-el-diseno-y-desarrollo-de-un-sitio-web/>

ciertos parámetros (en la propia URL o en las cabeceras HTTP), y de las que se obtiene una página web que el servidor devuelve como resultado de la invocación.

Ambos lenguajes permiten tener código embebido en el código HTML sea en la propia URL o en las cabeceras HTTP, esta es una ventaja porque a más de que va todo junto se evita la generación de scripts por separado y hace que el código sea más portable.

La primera característica favorable para la elección de PHP para el desarrollo del sistema de monitoreo es el acceso a las bases de datos el cual viene proporcionado por las librerías específicas de cada base de datos, mientras tanto, en Perl, existe una librería genérica independiente de la base de datos (DBD) que luego usa un conector (driver) para cada base de datos.⁵

PHP posee un formato menos confuso y más estricto, sin perder flexibilidad, esto lo hace de fácil integración con HTML que con Perl.

Una muy buena ventaja de Perl es que no tiene que compilar su aplicación, Perl es interpretado y por ello, sus programas pueden ser corridos en muchas plataformas sin necesidad de ser recompilado, esto reduce el ciclo de programación y lo hace portable, ya que existe un interpretador de PERL embebido en cada variedad de Unix y Windows, por lo que los cambios que debe hacer a su aplicación son mínimos o nulos.⁶

Otra ventaja de PHP sobre Perl es que PHP fue diseñado para desarrollo de scripts orientados a web, esto no quiere decir que Perl no fue desarrollado para esto, sino que Perl fue diseñado para hacer muchas cosas más y debido a

⁵ <http://davidasorey.net/static/perl-php-bbdd-tutorial/>

⁶ <http://www.enterate.unam.mx/Articulos/2006/abril/perl.htm>

esto, se hace muy complicado. La flexibilidad y complejidad de Perl dificultan la escritura de código que otro programador mantener con facilidad.⁷

Del análisis realizado se puede dar por elegido a PHP resumiendo que éste cuenta con prácticamente toda la funcionalidad favorable para el desarrollo, tales como construcciones del lenguaje, sintaxis y demás, sin hacerlo tan complicado como Perl puede llegar a ser. Perl es un lenguaje muy usado y auténtico, ha estado vigente desde finales de los ochenta, pero PHP está madurando muy rápidamente, y cuenta con una mayor comunidad de usuarios del lenguaje. Hoy en día es el principal lenguaje de script para el desarrollo de páginas web.

2.1.3.2.- PHP.⁸

PHP (Personal Home Page) es un lenguaje de programación interpretado que hizo su aparición en 1994, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz, es rápido a pesar de ser interpretado, multiplataforma y dispone de una gran cantidad de librerías que facilitan el desarrollo de las aplicaciones.

PHP permite embeber su código dentro de una página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas desarrollados íntegramente en un lenguaje distinto al HTML, además

⁷ <http://www.php-es.com/faq.languages.html>

⁸ <http://es.wikipedia.org/wiki/PHP>

ofrece un sinfín de funciones para la explotación de bases de datos de una manera llana, sin complicaciones.⁹

Entre su soporte pueden mencionarse InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML.¹⁰

PHP funciona tanto en sistemas Unix o Linux con servidor web Apache como en sistemas Windows con Microsoft Internet Information Server, de forma que el código generado por cualquiera de estas plataformas no debe ser modificado al pasar a la otra.¹¹

PHP tiene soporte para la programación orientada a objetos, es decir, es posible crear clases para la construcción de objetos, con sus constructores.

Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. Es utilizado en aplicaciones Web-relacionadas por algunas de las organizaciones más prominentes como la NASA.

⁹ <http://www.superhosting.cl/manuales/introduccion-a-la-programacion-en-php.html>

¹⁰ <http://www.maestrosdelweb.com/editorial/phpintro/>

¹¹ <http://www.adrformacion.com/cursos/php/leccion1/tutorial1.html>

INTRODUCCION A PHP¹²

PHP es un lenguaje de scripting que permite la generación dinámica de contenidos que se ejecuta en el lado del servidor web, multiplataforma e independiente del navegador. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos.

Al ser un lenguaje de script, se programa mediante simples sentencias sin compilar, que son interpretadas por el motor PHP instalado en el servidor, devolviendo al usuario de la página el texto resultante. Es decir, la presentación resultante será una simple página web sin mostrarse al usuario nada de php.

Es multiplataforma, en el sentido de que PHP puede ejecutarse en entorno unix y windows, y es independiente del navegador que se utilice, en el sentido de que el código php se ejecuta en el servidor, que devuelve al navegador solo texto

Para delimitar la sección de código PHP se puede realizar de varias formas:

- Usando las etiquetas `<?php y ?>`
- Usando las etiquetas `<? y ?>`
- Mediante `<script lenguaje="php"> </script>`

El funcionamiento de las páginas en PHP alojadas en un servidor es el siguiente:

- El navegador del usuario envía la petición del documento PHP.
- La petición llega al servidor el cual localiza el documento, lanza el intérprete de PHP y ejecuta todo su código.

¹² http://www.webtaller.com/manual-php/introduccion_php.php

- Una vez ejecutado el código se genera el resultado en HTML y lo devuelve al servidor para que lo transfiera al usuario.
- El servidor transfiere el resultado en HTML y es mostrado en el navegador del usuario

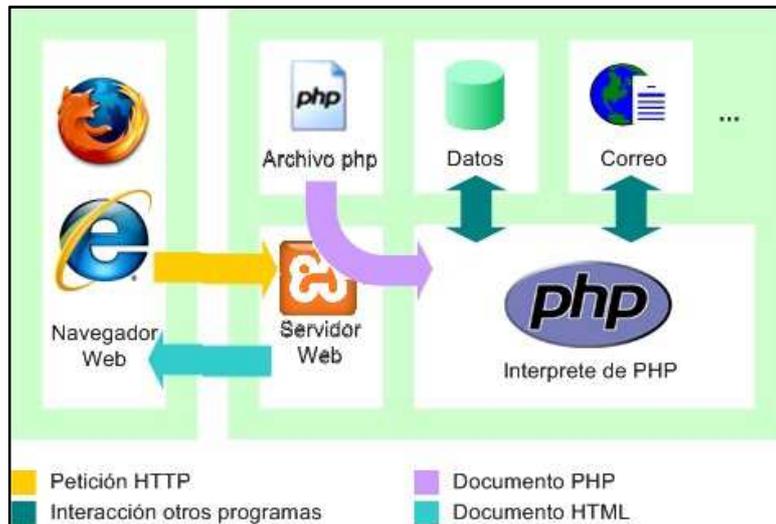


Gráfico 2.1: El funcionamiento de las páginas en PHP¹³.

```

<html>
<head>
<title> Ejemplo PHP </title>
</head>
<body>
<?php
$navigator = getenv("HTTP_USER_AGENT");
?>
<P>Este es un ejemplo
<b>
<?php
echo($navigator);
?>
</b>.
</P>
</body>
</html>

```

Gráfico 2.2: Código de una página HTML con PHP.

¹³ <http://www.kamlov.site90.net/?p=117>

```
<html>
<head>
<title> Ejemplo PHP </title>
</head>
<body>
<P>Este es un ejemplo
<b>del uso de PHPb>.
</P>
</body>
</html>
```

Gráfico 2.3: Código recibido por el navegador.

Este es un ejemplo del uso de PHP

Gráfico 2.4 Visualización el navegador.

El código generado es mantenible, se lee muy bien y se puede programar utilizando objetos. PHP es un lenguaje ideal tanto para el que comienza a desarrollar aplicaciones web como para el desarrollador experimentado.

Una de las grandes ventajas que tiene PHP es su gran versatilidad a la hora de trabajar con bases de datos, permitiendo realizar páginas con información almacenada en forma persistente en bases de datos y mostrarla según la petición de los usuarios.

Además PHP soporta herencia, aunque no múltiple. Y no, todas las reglas de encapsulación pública y privada están implementadas.

Una potente característica de PHP es la conectividad ya que dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite a PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras.

Para el diseño de un portal utilizando PHP se deben tener en cuenta las cuatro grandes características: Velocidad, estabilidad, seguridad y simplicidad.¹⁴

Velocidad: No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.

Estabilidad: La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es más difícil la existencia de bugs. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

Simplicidad: Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

VENTAJAS

- Es OpenSource, permite el acceso a cualquier persona.
- Es un lenguaje multiplataforma.

¹⁴ http://www.programacion.com/articulo/por_que_elegir_php_143

- PHP corre en la gran mayoría de plataformas utilizando el mismo código fuente.
- Es de fácil entendimiento y completamente expandible.
- Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como módulo CGI.
- PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- Programación segura y confiable, ya que el código fuente escrito en PHP es invisible al navegador y al cliente puesto que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Amplia gama de conexiones a motores de bases de Datos
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Soporta Programación Orientada a Objetos
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).

INCONVENIENTES

- La ofuscación (cambio no destructivo) de código es la única forma de ocultar los códigos fuentes, ya que al ser de lenguaje libre lo hace accesible a cualquier persona.
- El manejo de errores no es tan sofisticado como Cold Fusion o ASP.
- No existe IDE propio que provea las herramientas ideales para la programación como un Debugger, visor de variables, compilación de errores, todos los errores se los revisa desde el programa en ejecución.

SEGURIDAD¹⁵

PHP es un potente lenguaje y el intérprete, tanto incluido en el servidor Web como módulo o ejecutado como un binario CGI(Common Gateway Interface), puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea seguro por defecto.

PHP ha sido diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI, Perl o C y con la correcta selección de las opciones de configuración de tiempo de compilación y ejecución se consigue la exacta combinación de libertad y seguridad que se necesita. Ya que existen diferentes modos de utilizar PHP, existe también una multitud de opciones de configuración que permiten controlar su funcionamiento. Una gran selección de opciones garantiza que se pueda usar PHP para diferentes aplicaciones, pero también

¹⁵ <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>

significa que existen combinaciones de estas opciones y configuraciones del servidor que producen instalaciones inseguras.

2.1.4.- Herramienta de Gestión de Base de Datos

Un punto clave en este desarrollo es la base de datos, puesto que dependiendo de la información que allí se almacene depende todo el funcionamiento del sistema en general, para esto se debe realizar un análisis para determinar la base de datos le conviene al proyecto, siguiendo con el lineamiento de la utilización de software libre, existen varias alternativas de bases de datos que se las puede considerar, PostgreSQL y MySQL son los Sistema de gestión de base de datos (SGBD) de mayor representación dentro del mundo de Software Libre, principalmente por su alta integración con herramientas de programación WEB como PHP o JAVA.

Pues bien, desde un punto de vista objetivo, hacer la comparación de PostgreSQL y MySQL se podría considerar inadecuado; ya que si se realiza una pequeña analogía, sería como comparar un camión, frente a una tractomula, cada uno fue desarrollado con un propósito y un tipo de servicio diferente.

Tabla 2.2 Comparativa entre MySql y PostgreSQL

Características	MySql	PostgreSQL
Definición del esquema	Relacional	Objeto / Relacional
SQL para consultas	SQL92	SQL92 y SQL99
Herramientas de administración gráfica	PhpMyAdmin, MySql Administrator entre otros.	PgAccess, PhpPgAdmin, entre otros.
Tipos de datos	Entre numéricos, tipos de	En la versión 7.4 soporta 62

soportados	fecha y hora, y tipos de cadena de caracteres en total tiene 27 tipos de columna	tipos de datos entre carácter, numéricos, fecha y otros
Procedimientos almacenados	A partir de la versión 5 es posible crear Procedimientos almacenados y triggers	Triggers y funciones.
Elementos adicionales (vistas, secuencias, sinónimos)	Soporta vistas, no soporta sinónimos, y soporta algunas secuencias.	Soporta vistas, dominios, tipos de objetos nuevos, secuencias, reglas, no soporta sinónimos.
Backup en Caliente	Mysql_dump	Pg_dump
Integridad de Datos	Tiene soporte de constraint usando el motor de almacenamiento InnoDB	Buen manejo de constraint
Joins	Soporte de Joins, inner join. Natural join, cross join.	Soporta, natural join, inner join, cross join sin ningún inconveniente, además permite manejar subconsultas con unión, intersección y excepciones.
Recursos	Consume bajos recursos de RAM y hardware	El consumo de recursos es un 25% mayor al de MySql
Interfaz gráfica de comandos	MySQL	Psql
PHP	Excelente integración con PHP	Buena integración con php a partir de sus últimas versiones.
Rendimiento	Alta velocidad	Alta velocidad pero menor a MySql.

En resumen, PostgreSQL y MySQL tienen muy buen soporte web, actualmente es para lo que más se les utiliza, en el sistema de monitoreo a desarrollarse un factor principal es la velocidad de procesamiento de la información almacenada por LabVIEW en la base de datos, esto le da un punto a favor de MySQL ya que este es más rápido debido a que no hace ningún tipo de

control y/o auditoría, si se toma en cuenta que el sistema de monitoreo a parte de su rapidez también tiene que poseer estabilidad, integridad y seguridad del sistema, el punto a favor es PostgreSQL, esto no quiere decir que MySQL tenga estas características, sino que PostgreSQL supera a MySQL en un porcentaje que podría considerarse. Se debe tener en cuenta que PostgreSQL es un SGBD de alto desempeño orientado a la construcción de Minerías de información, es decir está orientado al manejo de grandes volúmenes de información, y MySQL está orientado a la realización de transacciones rápidas con el fin de agilizar la respuesta de una WEB.

Un excelente punto a favor para MySQL es su fácil integración a LabVIEW que es la herramienta de desarrollo elegida para uno de los módulos del sistema a desarrollarse, permitiendo inclusive realizar una conexión a herramientas para generar cubos de información a partir la información almacenada en ella.

Evaluando estas alternativas de bases de datos, se puede llegar a la conclusión de que el SGBD a utilizar es MySQL, ya que a parte de las características mencionadas que la hacen mejor a PostgreSQL, su facilidad de conexión a las herramientas de Desarrollo escogidas facilitaría el desarrollo de este proyecto.¹⁶

2.1.4.1.- MySQL 1.2.17

SQL, Structure Query Language (Lenguaje de Consulta Estructurado) es un lenguaje de programación para trabajar con base de datos relacionales como MySQL

¹⁶ <http://www.umanizales.edu.co/programs/ingenieria/Ventana/ventana12/articulo25.pdf>

MySQL es software Open Source (fuente abierta), esto significa que es posible para cualquier persona usarlo y modificarlo. MySQL usa el GPL (GNU General Public License) para definir lo que se puede y no se puede hacer con el software en diferentes situaciones.

MySQL es un interpretador de SQL, es un servidor de base de datos. MySQL permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas operaciones.

Ingresando instrucciones en la línea de comandos o embebidas en un lenguaje como PHP permite la comunicación con el servidor. Cada sentencia debe acabar con punto y coma (;).

MySQL posee sensibilidad a mayúsculas y minúsculas, es decir, si hace diferencia entre ellas, depende del sistema operativo, en Windows no es sensible, pero en Linux si lo es. Por ejemplo Windows interpreta igualmente las siguientes sentencias:

```
create database Central;
```

```
Create DataBase Central;
```

MySQL es un sistema de base de datos relacional, multihilo y multiusuario. Es desarrollado por MySQL AB desde 1995 (actualmente subsidiaria de Oracle Corporation) como software libre, con el objetivo de que cumpla el standard SQL sin sacrificar velocidad, fiabilidad o usabilidad.

Su principal objetivo de diseño fue la VELOCIDAD. MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran

archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

Una característica importante es que consume muy pocos recursos, tanto de CPU como de memoria.

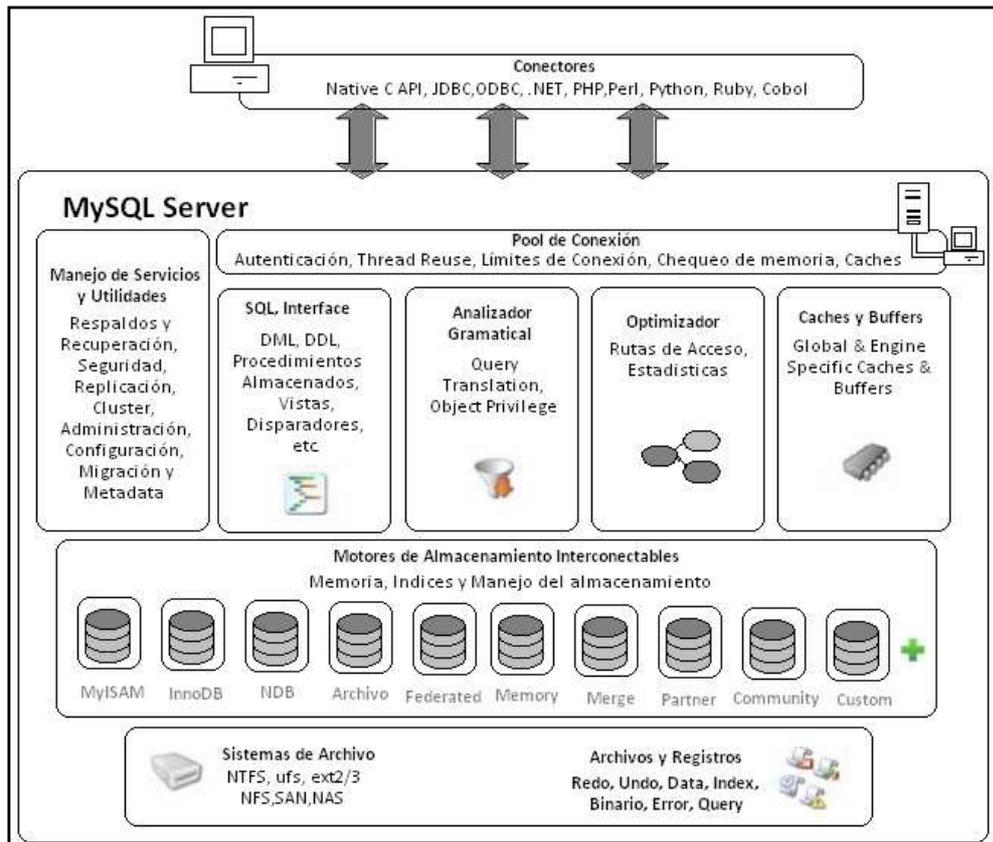


Gráfico 2.5: Diagrama de arquitectura de MySQL.

2.1.4.2.- Conectores y Drivers.

En la página web de MySQL se encuentra todos los drivers disponibles para la utilización de la base de datos con el lenguaje que se desea usar, aquí se tiene una lista de los drivers disponibles para cada lenguaje:

- **Connector/ODBC:** es un controlador genérico que puede funcionar en cualquier plataforma (Windows, Linux, Mac Os X, y UNIX), es

estandarizado y se lo puede usar con cualquier lenguaje con soporte para conexiones ODBC.

- Connector/J: Para aplicaciones Java.
- Conector/Net: Para aplicaciones .Net
- Connector/MXJ: MBean para integrar el servidor MySQL en aplicaciones Java.
- Connector/C++: Para aplicaciones en C++.
- Connector/C: Para aplicaciones en C.

Además existen utilitarios desarrollados para integrar al desarrollo, los mismos que facilitan la conexión y la administración de transacciones entre la aplicación y la base de datos como son los ORM (Object Relational Model) y librerías de abstracción de base de datos entre algunos ejemplos de ORM para php están Propel ORM y Doctrine que automáticamente generan el código de la capa de acceso a datos, Un ejemplo de una librería de abstracción de base de datos se tiene ADODB para php que provee una colección de clases que facilitan el trabajo con la base de datos.

2.1.4.3.- Administración de Almacenamiento

MySQL posee varios tipos de Administración de Almacenamiento, permite tener tablas tanto no transaccionales como transaccionales, actualmente permite el uso de clustering que es tener la base de datos distribuida en varias máquinas. Aquí se detallará los principales tipos de administración de almacenamiento de MySQL.

- **MyISAM**

MySQL por ser muy utilizado en aplicaciones web, en plataformas y por herramientas de seguimiento de errores como Bugzilla, está muy ligada a PHP. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para el sistema de monitoreo que se desarrollará.

- **InnoDB**

InnoDB es una seguridad para las transacciones (conforme a ACID) este motor de almacenamiento de MySQL permite las funciones de rollback, commit, brinda capacidades de recuperación de fallos para proteger los datos del usuario. InnoDB aumenta la concurrencia multi-usuario y el rendimiento. Almacena los datos del usuario en índices agrupados para reducir las operaciones de entrada y salida para las consultas comunes sobre la base de las claves principales. Para mantener la integridad de los datos, InnoDB también soporta restricciones FOREIGN KEY que permiten la integridad referencial. Se puede mezclar libremente tablas InnoDB con tablas de otros motores de almacenamiento de MySQL, incluso dentro de la misma declaración.

- **NDB** ¹⁷

MySQL clúster es una tecnología que permite el clustering de bases de datos en memoria en un ambiente de no compartición. La arquitectura de no compartición permite que el sistema gestor de base de datos (SGBD) funcione utilizando hardware no muy costoso y con requerimientos mínimos tanto de software como de hardware.

Como todo sistema de clustering, está diseñado para no tener un sólo punto de falla, cada componente tiene su propia porción de disco y memoria para trabajar. Bajo este esquema no se recomienda el uso de mecanismos de almacenamiento compartido como carpetas compartidas por red, sistemas de archivos de red, etc.

2.1.4.4.- Dimensiones máximas de las tablas:

Con el motor de almacenamiento MyISAM el tamaño de la base de datos puede crecer hasta 65536 TB (256^{17-2} Bytes) prácticamente el límite del tamaño de una base de datos esta dado por el sistema de ficheros del servidor. En cambio con el motor InnoDB que mantiene las tablas en un espacio que puede ser creado a partir de varios ficheros, el tamaño de este espacio puede ser hasta de 64 TB, donde pueden incluir particiones de disco.

2.1.4.5.- Ventajas:

- Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.

¹⁷ http://es.wikipedia.org/wiki/MySQL_Cluster

- Mejores utilidades de administración (backup, recuperación de errores, etc.).
- Integridad de datos al no responder el servicio, no suele perder información ni corromper los datos.
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.
- MySQL se comporta mejor que PostgreSQL a la hora de modificar o añadir campos a una tabla "en caliente".

2.1.4.6.- Inconvenientes:

- No tiene las mismas capacidades como los otros gestores profesionales.
- No es intuitivo.
- No hay documentación para algunas utilidades.

Características adicionales

- Usa GNU Automake, Autoconf, y Libtool para portabilidad
- Uso de multihilos mediante hilos del kernel.
- Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice
- Tablas hash en memoria temporales
- El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL
- Completo soporte para operadores y funciones en cláusulas select y where.

- Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación
- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).
- Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.

2.1.4.7.- Tipos de compilación del servidor

Existen tres tipos de compilación del servidor MySQL: Estándar, Max (No se trata de MaxDB, que es una cooperación con SAP), MySQL-Debug.

MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porque el código de depuración puede reducir el rendimiento.

Estándar: Los binarios estándar de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.

Max: Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.

2.1.4.8.- Licencia

La licencia GNU GPL de MySQL obliga a que la distribución de cualquier producto derivado (aplicación) se haga bajo esa misma licencia. Si un desarrollador desea incorporar MySQL en su producto pero desea distribuirlo bajo otra licencia que no sea la GNU GPL, puede adquirir una licencia comercial de MySQL que le permite hacer justamente eso.¹⁸

2.2.- Ambientes de desarrollo

Hablar de ambientes de desarrollo de software es hablar de varias herramientas contenidas en una sola, estas herramientas ayudan a los programadores a desarrollar software sobre entornos más amigables. Es decir, aquellos en los que el programador puede acceder con el menor esfuerzo a diferentes recursos como editores, compiladores, herramientas de análisis, etc. Para este proyecto se han evaluado y elegido herramientas de desarrollo como PHP, MySQL, así como un servidor web Apache por lo que los ambientes de desarrollo que se pueden adaptar a este proyecto son “Eclipse” y “NetBeans”, en este punto no caben las comparaciones porque estos dos ambientes de desarrollo son prácticamente iguales, los dos ambientes son OpenSource, constan con profilers, debuggers, coloreado en el código, conexiones con bases de datos, en el caso de este proyecto conexión con MySQL, de hecho es posible importar los trabajos realizados en Eclipse a NetBeans y viceversa, una única diferencia entre estos dos ambientes se la puede citar en que Eclipse tiene más usuarios y por lo tanto se va a encontrar más información sobre este ambiente

¹⁸ <http://es.wikipedia.org/wiki/MySQL>

2.2.1.- NetBeans

Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso y está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio.¹⁹

Entre las características de la plataforma están:

2.2.1.1.- Auto-completado y documentación de funciones PHP:

Rápido acceso a la documentación de PHP, y si se necesita más información se provee el link directo a la función.

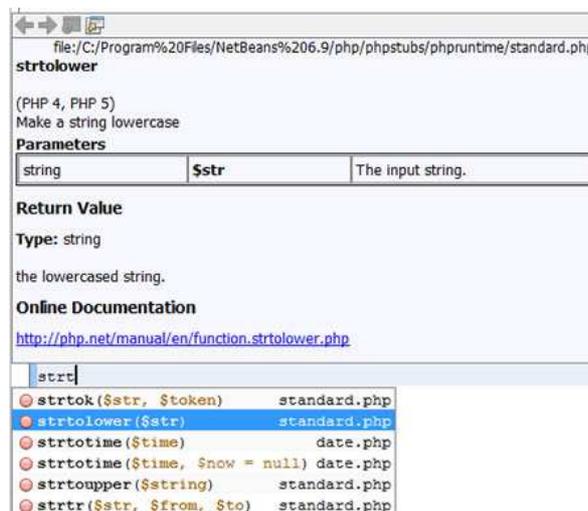


Gráfico 2.6: Autocompletado PHP.

¹⁹ http://netbeans.org/index_es.html

2.2.1.2.- Generador de PHPDoc:

La documentación es muy importante, pero difícil de realizar. Con NetBeans esto se hace fácil: se debe posicionar el cursor en la línea anterior a la declaración de la función, luego escribir `/**` y presionar [ENTER].

```
/**
 *
 * @param <type> $parametro1
 * @param <type> $parametro2
 * @return <type>
 */
function miFuncion($parametro1, $parametro2) {
    return $parametro1+$parametro2;
}
```

Gráfico 2.7: Generador de PHPDoc.

2.2.1.3.- Auto-completado de código propio:

Esto es una consecuencia del punto anterior, al documentar código con el formato esperado, estos serán mostrados como se observa a continuación.

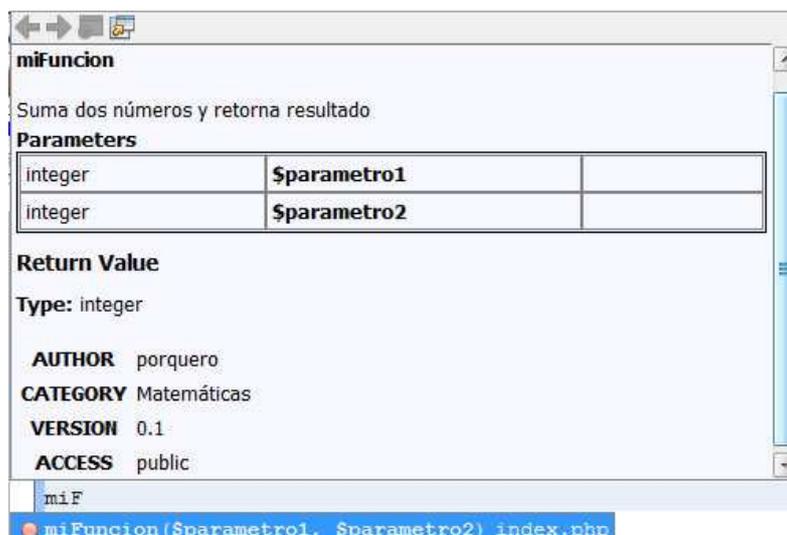
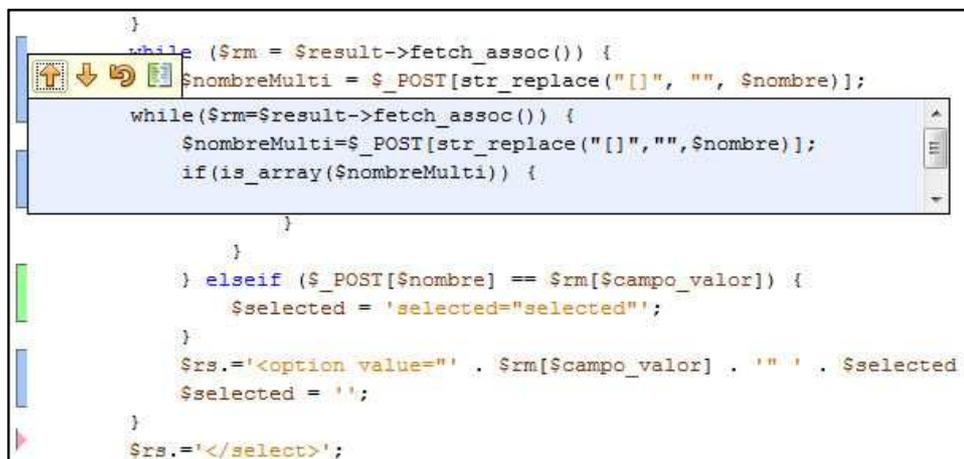


Gráfico 2.8: Auto-Completado de código propio.

2.2.1.4.- Soporte para CVS, Mercurial, Subversión:

Notar la diferencia de colores:

- Verde: Código agregado
- Azul: Código modificado
- Rojo: Código eliminado
- Además si se hace click en el cualquiera de estos colores mostrará el código reemplazado. Una gran extensión al [CTRL]+Z.



```
while ($rm = $result->fetch_assoc()) {
    $nombreMulti = $_POST[str_replace("[]", "", $nombre)];
    while($rm=$result->fetch_assoc()) {
        $nombreMulti=$_POST[str_replace("[]", "", $nombre)];
        if(is_array($nombreMulti)) {
            }
        }
    } elseif ($_POST[$nombre] == $rm[$campo_valor]) {
        $selected = 'selected="selected"';
    }
    $rs.='<option value="' . $rm[$campo_valor] . "' . $selected
    $selected = '';
}
$rs.='</select>';
```

Gráfico 2.9: Código PHP con su historial de cambios.

2.2.1.5.- Atajos de teclado:

- Ordenar y tabular código: [ALT]+[SHIFT]+F
- Seleccionar el archivo editando en el proyecto: [ALT]+[SHIFT]+1
- Alternar maximizar ventana: [SHIFT]+[ESC]
- Ir a siguiente ocurrencia de variable/función/método marcado (el cursor debe estar dentro del nombre)
- Ir a declaración de variable/función/clase: [CTRL]+B ó [CTRL]+(click)

- Lista de archivos abiertos: [CTRL]+[TAB]
- Duplicar línea: [CTRL]+[SHIFT]+UP ó [CTRL]+[SHIFT]+DOWN
- Mover línea: [ALT]+[SHIFT]+UP ó [ALT]+[SHIFT]+DOWN
- Eliminar línea: [CTRL]+E
- Comentar: [CTRL]+[SHIFT]+C
- Renombrar variable/función/clase: [CTRL]+R.²⁰

2.2.1.6.- Características Adicionales

- Administración de las interfaces de usuario (ej. menús y barras de herramientas)
- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- Administración de ventanas
- Framework basado en asistentes (diálogos paso a paso).²¹

2.3.- Herramientas de software adicionales

Además del software que se ha visto anteriormente se usarán librerías para agilizar el desarrollo y que serán de gran utilidad para el proyecto.

²⁰ <http://playa-soluciones.cl/blog/7-investigacion-y-desarrollo/31-razones-por-las-cuales-usar-netbeans-para-proyectos-php>

²¹ <http://es.wikipedia.org/wiki/NetBeans>

2.3.1.-Propel ORM.

Hoy en día existen varias formas de conexión entre la base de datos MySQL y PHP, una de ellas es usar las funciones que vienen integradas en el lenguaje pero su alcance es limitado y se necesita codificar más para lograr un buen control de la transaccionalidad de la base de datos, existen también librerías de software libre que tienen un alcance mayor y brindan funciones de ejecución de comandos y búsqueda más avanzados.

Existen también utilitarios de mapeo objeto relacional que traen ya embebido la conexión a múltiples bases de datos, además que generan la capa de modelo o acceso a datos para un proyecto de desarrollo.

“Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia”²².

Para este proyecto se usará Propel ORM²³ que es una herramienta de mapeo objeto relacional de software libre la principal ventaja de esta herramienta es que puede ser usada con la mayoría de motores de bases de datos, la utilización de funciones amigables como por ejemplo para buscar dentro de una tabla de personas todas las personas con nombre “Marcos” se usaría la siguiente sentencia:

```
$personas = LecturaGpuQuery::create()  
->filterByNombre(“Marcos”)
```

²² http://es.wikipedia.org/wiki/Mapeo_objeto-relacional

²³ <http://www.propelorm.org/>

->orderByFecha()

->find();

Para generar la capa de acceso a datos con Propel se debe primero generar un archivo con la información de la base de datos a usar y se lo guarda en dos archivos "runtime-conf.xml" y "buil.properties" en la raíz del directorio del proyecto. Los archivos tienen la siguiente estructura:

Runtime-conf.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <propel>
    <datasources default="central">
      <datasource id="central">
        <adapter>mysql</adapter> <!-- sqlite, mysql, myssql, oracle, or pgsql -->
        <connection>
          <dsn>mysql:host=localhost;dbname=central</dsn>
          <user>root</user>
          <password>marcos</password>
        </connection>
      </datasource>
    </datasources>
  </propel>
</config>
```

Build.properties:

```
propel.project = central
```

```
# The Propel driver to use for generating SQL, etc.
```

```
propel.database = mysql
```

```
# This must be a PDO DSN
```

```
propel.database.url = mysql:dbname=central
```

```
propel.database.user = root
```

```
propel.database.password = marcos
```

Una vez configurado estos archivos se procede a ejecutar los siguientes comandos en la consola de Windows para que se comience a generar el código:

- Para generar el archivo schema.xml que contiene un mapeo de la base de datos:

```
propel-gen ..\htdocs\central reverse
```

- Para generar el código de objeto relacional:

```
propel-gen ..\htdocs\central om
```

- Y por último para generar la configuración se ejecuta:

```
propel-gen ..\htdocs\central convert-conf
```

2.3.2.- JavaScript

Es un lenguaje de programación interpretado, es un lenguaje orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.²⁴

Este es un lenguaje que se ejecuta del lado del cliente, aunque existe forma de que pueda ser ejecutado del lado del servidor esta mayormente orientado al lado del cliente, su ventaja principal es que permite agregar funcionalidad a una página web y aumentar la interacción de la interface con el usuario evitando refrescar la página en cada acción.

2.3.3.- AJAX

Es el acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo web que permite que la interface web no deba refrescarse en cada acción que ejecute el usuario, las aplicaciones que son desarrolladas mediante

²⁴ <http://es.wikipedia.org/wiki/JavaScript>

AJAX se llaman RIA (Rich Internet Applications) aplicaciones de internet enriquecidas, la página web mantiene una comunicación asíncrona con el servidor en segundo plano.²⁵

Las llamadas a las funciones AJAX se las hace mediante JavaScript y la transmisión de datos se las hace mediante XMLHttpRequest (interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web) para los datos transferidos se usa cualquier codificación basada en texto, incluyendo: texto plano, XML, JSON, HTML y codificaciones particulares específicas.

2.3.4.- JQuery

Es una librería para JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.²⁶

Su licenciamiento es libre y se puede ocupar tanto en proyectos libres como privativos, es muy importante destacar que es soportado por todos los navegadores hoy en día.

2.3.5.- ThickBox

Es una herramienta desarrollada mediante JavaScript utiliza también la librería de JQuery y sirve para desarrollar ventanas pop up que se despliegan dentro de una página web permite mostrar todo tipo de contenido como: imágenes, videos, HTML (iFrames).

²⁵ <http://es.wikipedia.org/wiki/AJAX>

²⁶ <http://es.wikipedia.org/wiki/JQuery>

Para usar esta herramienta se debe agregar los siguientes archivos al proyecto que se esté desarrollando:

Thickbox.css

Thickbox.js

2.4.- Patrón Modelo Vista Controlador.

Para el desarrollo de sitios web es necesario usar una arquitectura adecuada, el patrón Modelo Vista Controlador (MVC) ayuda a construir una arquitectura separando el proyecto en tres componentes distintos: la interfaz de usuario, los datos de la aplicación, y la lógica del negocio.

En gráfico 2.10 se puede apreciar el flujo de actividad entre los componentes de la arquitectura MVC.

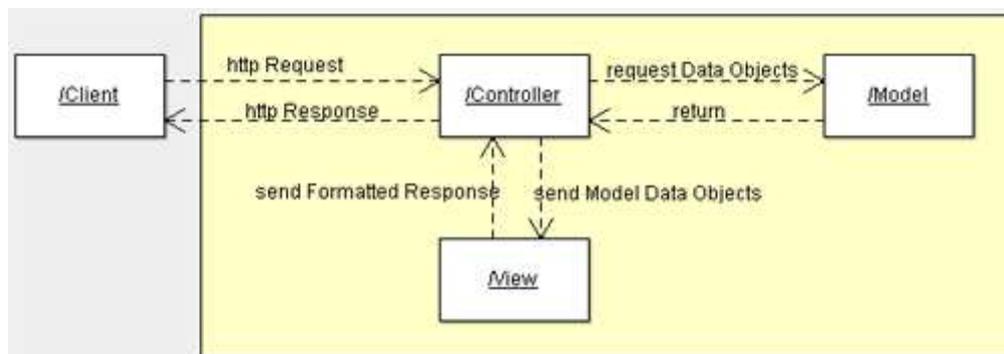


Gráfico 2.10: Flujo de actividad de la arquitectura MVC²⁷.

2.4.1.- Modelo.

Es una abstracción del modelo relacional de la base de datos, es decir; que se encarga de consultar, modificar y eliminar los datos.

²⁷ <http://php-html.net/tutorials/model-view-controller-in-php/>

2.4.2.- Vista.

Es una representación grafica de los datos para que el usuario pueda interactuar con ellos.

2.4.3.- Controlador.

Es el encargado de gestionar las peticiones del usuario usa el modelo para traer datos y devuelve una vista.

Con el siguiente ejemplo se explica el funcionamiento del patrón MVC:

- El usuario activa un evento en la interfaz (botón, enlace, etc.)
- El controlador recibe el evento y lo gestiona.
- El controlador consulta o modifica el modelo.
- El controlador manda la respuesta a la interfaz y esta reacciona en función de esta (cambia de pantalla, abre un enlace, etc.)
- La interfaz espera una nueva acción del usuario.

2.5.- Sistemas de Monitoreo y control (SCADA).

Los sistemas de monitoreo y control para equipamiento electrónico son conocidos como sistemas SCADA (Supervisory Control and Data Acquisition). Un sistema SCADA “es una aplicación de software especialmente diseñada para funcionar sobre ordenadores (computadores) en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos) y controlando el proceso de forma automática desde la pantalla del ordenador.”²⁸

²⁸ <http://es.wikipedia.org/wiki/SCADA>

Un SCADA es un sistema centralizado, jerárquico cuyo objetivo es controlar el proceso de una planta de producción desde un solo computador centralizado, este computador recibe la información recolectada de los diferentes equipos de campo mediante una estructura de componentes entre ellos sensores, actuadores, PLC's, DCS's los mismos que están interconectados por una red de comunicaciones.

2.5.1.- Características.

- La **Monitorización** es la presentación de las variables de control de cada equipo en una pantalla, la representación de estas variables deben ser mediante gráficos avanzados que hagan referencia al equipo que se está monitorizando, por ejemplo si se monitorea la temperatura de un equipo se podría hacer un gráfico de un termómetro, del nivel de agua de un tanque con un gráfico de un tanque.
- **Supervisión** mediante herramientas de gestión que nos permitan controlar el proceso industrial y tomar decisiones, inclusive que tenga rutinas de software que realicen acciones automáticas como anular tareas o modificar aquellas que están asociadas a los autómatas.
- **Adquisición de datos y almacenamiento** en bases de datos relacionadas o archivos planos dependiendo el uso que se quiera dar a la información histórica, con una base de datos relacionada se va a obtener mayores ventajas por que la manipulación de la información es más fácil por el uso del SQL y de bases transaccionales.

- **El mando** o la facilidad que tiene el operador o el supervisor para modificar el proceso (parar el sistema, encenderlo, cambiar parámetros de configuración). El sistema debe permitir hacer estas acciones en caliente, es decir que no requieran un paro del sistema para realizarlas.
- **Programación de acciones** ante eventualidades o por requerimientos de la empresa, como por ejemplo manipular la regulación de una válvula de agua para que cada cierto tiempo se abra y se cierre, o al recibir un mensaje de alarma el sistema automáticamente realice una acción de protección.
- **Seguridad de la información** obtenida, los datos recolectados deben ser restringidos a personal autorizado para evitar la mala utilización o la fuga de información, además la red de comunicación también debe ser protegida ante interceptaciones o intrusos.
- **Control de usuarios** con acceso a la información y a los controles que cada uno puede realizar o ver.
- **Arquitectura abierta** y la posibilidad que se pueda extender con la menor dificultad.
- **Presentación gráfica** de los elementos supervisados, interfaz del operador o HMI (Human Machine Interface).

2.5.2.- Prestaciones

- Paneles de alarmas que muestren las eventualidades de ser posible con sonidos y ventanas emergentes que llamen la atención al supervisor.

- Análisis de datos históricos, explotación de la información recolectada mediante sistemas de análisis de información histórica para el mejor aprovechamiento, la posibilidad de volcar los datos en hojas de cálculo, o la obtención de reportes.
- Reportes e informes de datos generales.
- Cambiar el proceso del autómeta de forma parcial o total.
- Cálculos avanzados sobre el CPU y no sobre el autómeta menos especializado.

2.5.3.- Componentes de Hardware

Un sistema SCADA es un software que necesita de todo un equipamiento de hardware para cumplir con las tareas de supervisión y control.

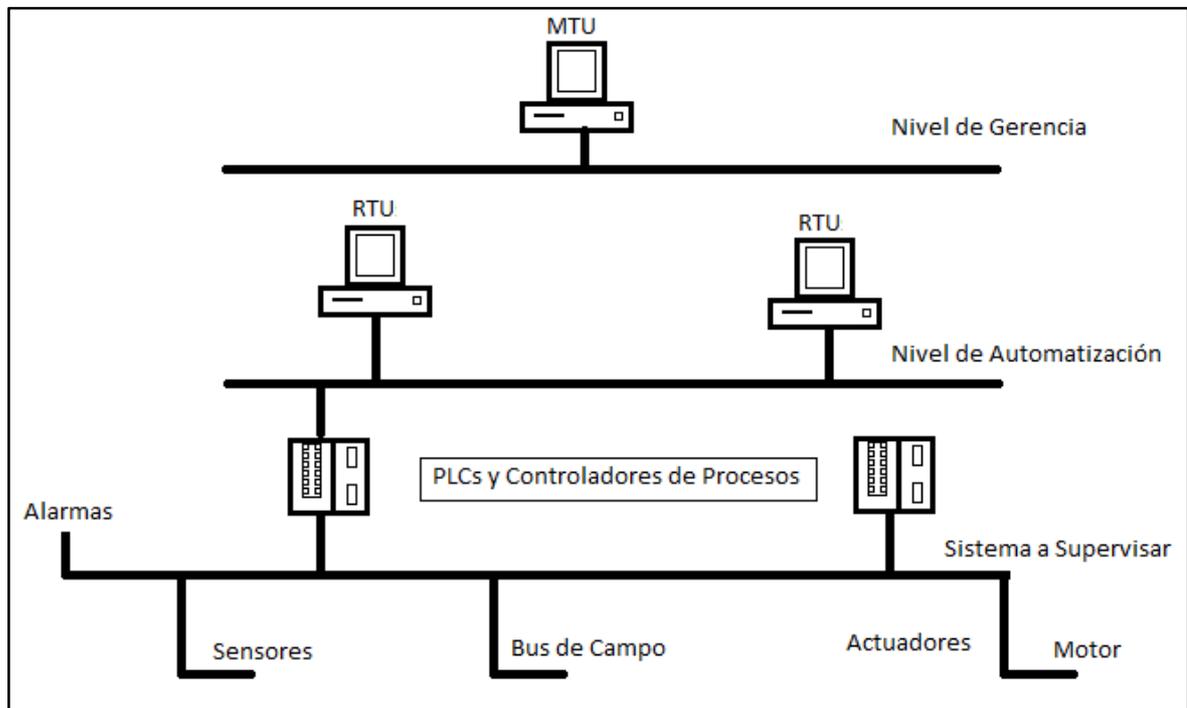


Gráfico 2.11: Estructura básica de un sistema SCADA a nivel de hardware.

2.5.4.-Unidad central (MTU, Master Terminal Unit)

La unidad central es un computador que recopila todos los datos recolectados por las subestaciones debe proveer un sistema sencillo y barato de interoperabilidad con multiplataforma y multisistemas para que puedan intercambiar información sin inconvenientes.

Su función principal es la recolección de datos y el almacenamiento de los mismos, entre otras funciones se tiene:

- Gestionar las comunicaciones.
- Envío de información.
- Comunicación con los Operadores.
- Análisis.
- Impresión.
- Visualización de datos.
- Mando.
- Seguridad.

El MTU es la interface del Operador, aquí el podrá controlar el proceso, modificarlo si es necesario y llevar a cabo funciones de supervisión.

2.5.4.1.- Estaciones Remotas (RTU, Remote Terminal Unit)

Las estaciones remotas son aquellos equipos que están alejados de la unidad central y cumplen funciones específicas de control, supervisión y/o comunicación, estas unidades no solo son computadores sino que pueden ser:

- RTU's, son equipos especializados en comunicación, es decir; que son usados para lectura de datos y replicación, aunque hoy en día también son usados para controlar procesos mediante tarjetas convertidoras adecuadas o se comunican con los elementos de control (PLC, reguladores) mediante protocolos de comunicación adecuados.
- PLC's (Programmable Logic Controller) Controlador lógico programable, son equipos dedicados al control de un proceso en general, sus funciones son de control, pero se han venido incorporando nuevas funciones que ya no existe mucha diferencia entre PLC y RTU.
- IED's (Intelligent Electronic Devices) periféricos inteligentes, son dispositivos que pueden controlar un proceso sin la necesidad de la intervención del operador, pueden ser programados para que ejecuten alguna acción después de un evento.

Las funciones de los IED's también han sido incluidas en los PLC así que el PLC es el elemento más avanzado para control y comunicación en sistemas SCADA.

- Sistemas Remotos: son subsistemas que pueden controlar independientemente una área del proceso, por ejemplo se puede tener un sistema general de la producción eléctrica y subsistemas que controlen la generación de una planta.

2.5.4.2.- Sistema de comunicación

Para la comunicación entre servidores (unidades centrales) y clientes (unidades remotas) se puede decir que se usan todos los medios existentes para comunicación por la actual estandarización de las comunicaciones, se pueden encontrar sistemas SCADAS sobre formatos estándares como los RS-232, RS-422 y RS-485 a partir de los cuales y mediante una red TCP/IP se puede generalizar el uso de los datos recolectados pasándolos a otro sistema o subsistema que lo requiera, además se puede usar formas más modernas de comunicación como Bluetooth, Microondas, satélite, cable, etc.

Para comunicar los elementos de campo con las estaciones de control se usan protocolos industriales, para la comunicación entre estas muchas veces se requiere interfaces especiales como módems.

Es necesario implementar un plan de seguridad para proteger el sistema de comunicación puesto que los datos que son recolectados a mas que pueden ser perdidos o alterados por causas como la distancia entre el RTU y el MTU, los equipos electrónicos que generan campos electroestáticos también pueden ser vulnerables a atentados por terceros que quieran hacer daño a la empresa o interceptar los datos para otro fin.

Para evitar la pérdida de señal y consecutivamente de datos, se debe usar medios de transmisión seguros como cables SPT si es que se realiza una red LAN, o con radios de alta frecuencia tomando en cuenta el tipo del equipo de campo que se va a controlar. Para proteger los datos contra terceros, hay que implementar seguridades en la red como Firewalls, Gateways para así tener una comunicación confiable y segura.

2.5.4.3.- Instrumentos de campo

Son todos aquellos que permiten tanto realizar la automatización o control del sistema (PLC's, controladores de procesos industriales, y actuadores en general) como los que se encargan de la captación de información del sistema (sensores y alarmas). A pesar de que todos los componentes pueden ser de diferentes proveedores, es más; aun el software de cada componente puede ser de diferente proveedor, pero el sistema SCADA los integra y se crea un solo Software de control para todos los equipos.

2.5.5.- Estructura de un sistema SCADA

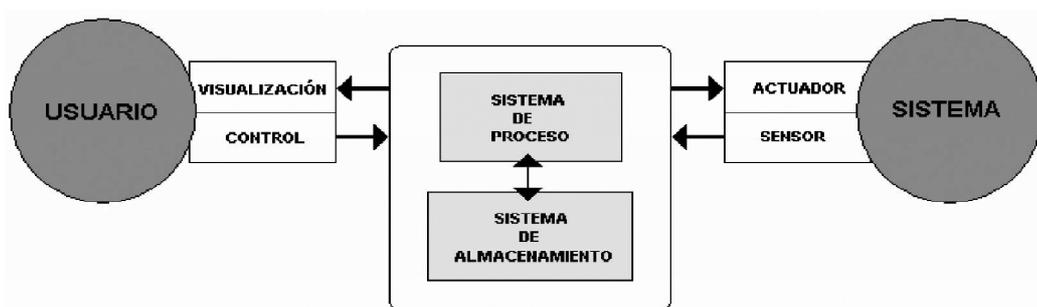


Gráfico 2.12: Estructura básica de un sistema SCADA.

El usuario mediante un sistema de visualización y de control se comunica con el sistema de proceso, este sistema está a su vez se comunica con los actuadores, sensores, PLC, entre otros y estos se conectan a los equipos de campo que son los equipos que se está monitoreando (Generadoras de electricidad, válvulas, compuertas, motores). El sistema de proceso está leyendo los datos de los sensores y los almacena en un sistema de almacenamiento sea este tan básico como un archivo plano como más avanzado como una base de datos.

2.5.6.- Arquitectura de un sistema SCADA

Con la estructura básica que se revisó en el tema anterior se pueden diseñar varias arquitecturas con servidores, computadoras y equipos de campo dependiendo de los recursos y de los requerimientos del cliente.

Una arquitectura debe tener las siguientes características:

2.5.6.1.- Escalabilidad

Quiere decir que el sistema debe admitir el incremento no solo de equipos a controlar, sino de servidores, equipos clientes, y de aplicaciones, esto dependiendo de:

- Espacio disponible
- Capacidad del equipo informático (memoria, procesadores, alimentaciones)
- Capacidad del sistema de comunicaciones (limitaciones físicas, protocolos, tiempos de respuesta)

2.5.6.2.- Disponibilidad

El sistema debe tener una “llanta de emergencia” ante cualquier fallo que pueda presentarse, esta llanta de emergencia debe ser remplazada de inmediato (si es automáticamente es mejor) para así tener el sistema completo en funcionamiento, como por ejemplo cuando exista una falla de hardware en el servidor principal se puede solucionar con un servidor espejo que trabaje en paralelo al servidor principal, así cuando el servidor principal presente alguna novedad, automáticamente funcione el servidor espejo.

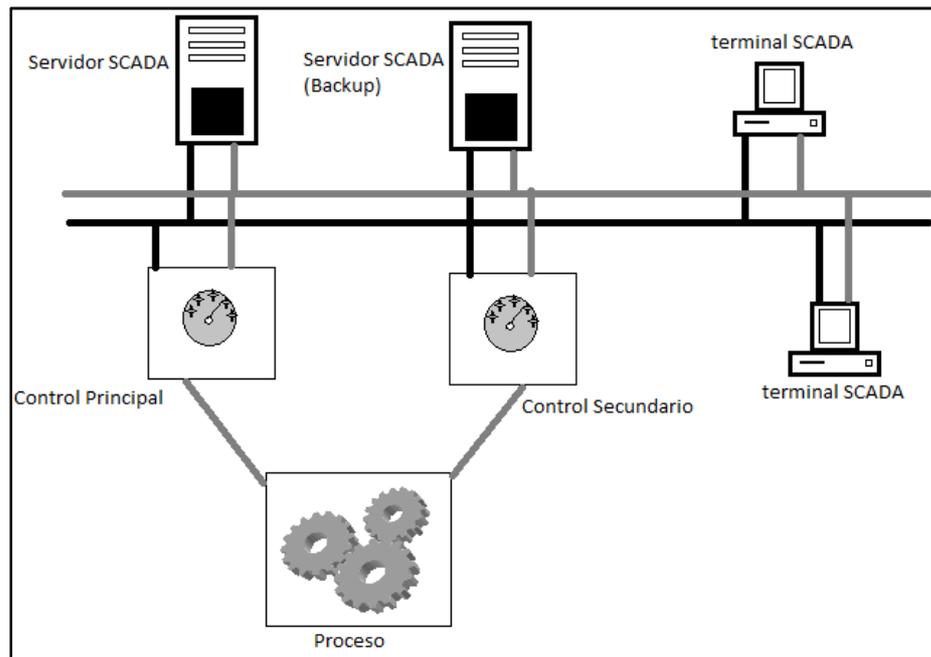


Gráfico 2.13: Redundancia de dispositivos en un sistema SCADA.

Como se observa en el gráfico 2.13 el sistema contiene 2 servidores SCADA un principal y uno de backup, y así mismo se tiene 2 controladores que están actuando en el mismo proceso de esta manera se puede tener una mayor disponibilidad mediante la redundancia de los dispositivos y de la información.

2.5.6.3.- Robustez

La robustez a diferencia de la disponibilidad es que al presentarse un fallo que no pudo ser solucionado con un sistema de backup, por lo menos se pueda usar el sistema por partes, es decir; que el control de los procesos estén en diferentes sistemas que puedan actuar independientemente, y aun así falle uno el resto pueda seguir trabajando, o si falla el aplicativo del MTU se pueda seguir controlando desde los RTU.

2.5.6.4.- Prestaciones

Es el tiempo de respuesta del sistema, se debe tomar en cuenta la cantidad de información que se está recolectando, y se está analizando, cuando la carga de datos es demasiado extenso se debe repartir el trabajo entre varias máquinas para dividir el trabajo y disminuir el tiempo de trabajo, así mismo el usuario debe estar consciente de que si la carga de datos que está consultando es muy grande, obviamente el sistema va a demorarse más.

2.5.6.5.- Seguridad

Un sistema SCADA está diseñado para funcionar en línea y continuamente sin parar, una falla en la arquitectura, un usuario malintencionado, o una situación imprevista, pueden hacer que el sistema deje de funcionar bien.

Se deben tomar en cuenta políticas de seguridad, como control de usuarios, accesos de usuarios a funciones específicas, encriptado de datos importantes, y validación de la información ingresada.

2.5.6.6.- Mantenibilidad.

Un módulo que permita planificar el mantenimiento preventivo de cada parte del sistema, desde los equipos de campo hasta el software de los equipos podría disminuir el tiempo de paro del proceso por motivos de mantenimiento.

2.5.7.- Estructura y componentes de un Software SCADA.

El software SCADA es denominado también HMI (Human Machine Interface), este software tiene como característica principal una gráfica avanzada

que permite visualizar todo el proceso con gráficos representativos al equipo que se está monitoreando.

En nivel de automatización por lo general son programas que se comunican con los equipos de campo (PLC, sensores, actuadores, etc.) recolectando los datos de las lecturas y mandando señales de control para que estos equipos de campo a su vez interactúen con el equipo al que se está monitoreando (Generadora de Electricidad, Motor, Válvula), el software de este nivel puede ser totalmente independiente, es decir; que no depende de otro programa del nivel de automatización o de gerencia para que funcione, suelen usarlo los operadores y supervisores.

En el nivel de gerencia en cambio son aplicativos que dependen de todos los programas hechos en el nivel de automatización porque en esta capa están integrados todos esos programas, en esta aplicación se puede divisar y controlar todos los equipos de campo, es decir; todo el proceso completo. Estos sistemas se conectan también con los sistemas de gestión de la empresa como son los ERP (Enterprise Resource Planning o Planificación de Recursos Empresariales).

2.6.- Metodología de desarrollo de Software.

“Metodología de desarrollo de software en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información”²⁹.

²⁹ http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software

2.6.1.-Metodologías Tradicionales.

Las son aquellas que se basan en un proceso dividido en etapas consecutivas, en cada etapa se realiza una actividad específica y en algunos casos con gente especializada para dicha actividad, se caracteriza también por la documentación exhaustiva, grandes equipos de desarrollo, tiempos de implementación largos y costos altos. Entre las principales metodologías tradicionales se encuentran RUP, MSF

2.6.2.-Metodologías Ágiles.

Las metodologías ágiles al contrario de las tradicionales están basadas en dos aspectos puntuales, el retrasar las decisiones y la planificación adaptativa. El retrasar las decisiones lo máximo permite que el cliente defina su requerimiento y desarrollarlo hasta cuando el este seguro. Y la planificación adaptativa es que el producto en desarrollo soporte cualquier cambio a cualquier fecha con el menor esfuerzo posible, el objetivo de las metodologías ágiles esta en obtener software funcional y desarrollar los documentos que den mayor valor al producto.

Diferencias entre metodologías tradicionales y ágiles³⁰.

El modelo tradicional está basado en el desarrollo en cascada y por procesos en donde cada fase del proyecto tiene que pasar por un equipo especializado para esa fase, cada equipo realiza su trabajo en su tiempo y al final de cada fase se traspasa el proyecto de equipo a equipo como una carrera de postas.

³⁰ <http://www.slideshare.net/fmmeson/metogologias-de-desarrollo-de-software-tradicionales-vs-agiles>

Las metodologías ágiles en cambio tienen un proceso iterativo e incremental conformado por equipos multidisciplinarios que tienen el conocimiento necesario para abarcar todo el proyecto, el número de integrantes de los equipos son más pequeños pero en cambio tienen un mejor rendimiento intelectual.

Los modelos tradicionales se basan mucho en la documentación que se va a generar, es decir, que obligan al desarrollo a que todo quede documentado a pesar de que alguno de estos no vaya a ser útil en el futuro, esto hace que el desarrollo se haga muy burocrático y pesado, al contrario las metodologías ágiles disminuyen la cantidad de documentación pero las sustituyen con la comunicación entre los miembros del equipo.

A continuación se muestra una tabla con las principales diferencias entre metodologías ágiles y tradicionales.

Tabla 2.3: Comparativa entre Metodologías Ágiles y Tradicionales

Metodologías Tradicionales	Metodologías Ágiles
Cambios lentos a moderados.	Cambios moderados a rápidos.
Productos medianos a grandes.	Productos medianos a chicos.
Desarrolladores con conocimientos medianos.	Desarrolladores con altos conocimientos.
Alto número de personal.	Solo el personal adecuado.
Más roles y Equipos especializados para el desarrollo.	Menos roles y equipos multifuncionales.
Comunicación con el cliente mediante reuniones.	El cliente es parte del equipo de desarrollo.
Mayor número de artefactos.	Menor número de artefactos, solo se desarrollan aquellos que den valor.

En conclusión se ha decidido usar una metodología ágil porque a diferencia de las metodologías por procesos se desarrollan solo los artefactos que son de valor para el producto final, no se basan en la generación de documentos sino en la comunicación entre los miembros del equipo, proponen mejores prácticas para el desarrollo de software como la programación en parejas, equipos de desarrollo multidisciplinarios, entregas continuas, desarrollo por medio de pruebas, tiempos de desarrollo cortos y lo más importante que se obtienen resultados rápidos.

2.6.3.- Selección de la metodología de desarrollo

Entre las principales metodologías ágiles se tiene a Programación Extrema (XP) y SCRUM. Ambas metodologías han sido desarrolladas por expertos en ingeniería de software miembros desarrolladores del manifiesto ágil (documento donde se describe el fin y la búsqueda del desarrollo ágil), al tratar de comparar estas dos metodologías se podría llegar a la conclusión de que deberían ser usadas conjuntamente puesto que son consideradas como complementarias, Scrum para la administración del proyecto y XP para el desarrollo.

2.6.3.1.- Programación Extrema

“La programación extrema es una metodología de desarrollo ligera (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas.”³¹

³¹ <http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>

Valores

- Simplicidad.
- Comunicación.
- Retroalimentación
- Coraje.

Principios

Los principios son las mejores prácticas que propone la metodología para el desarrollo de cualquier producto, son doce y se dividen en cuatro grupos.

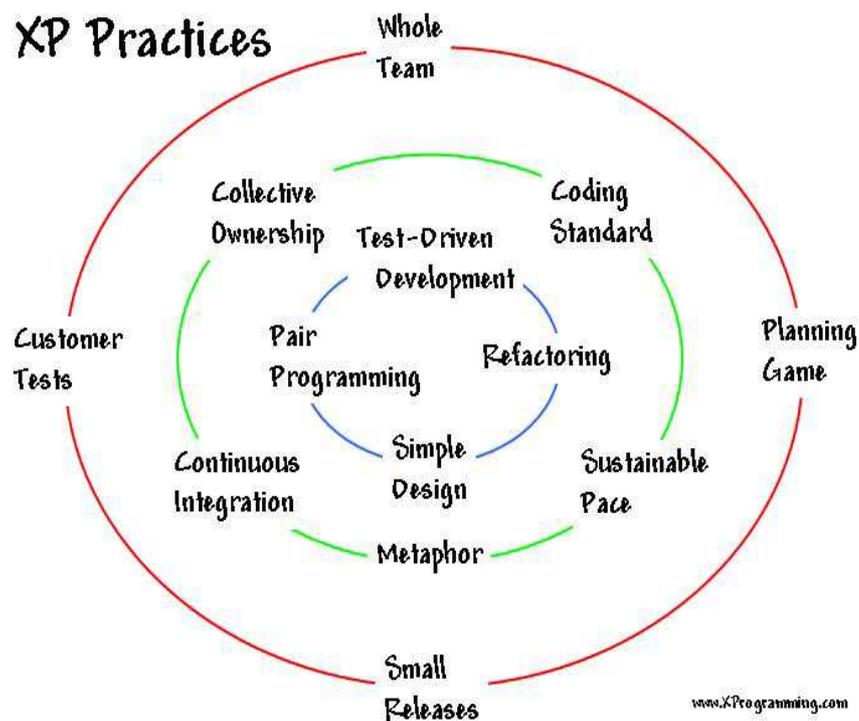


Gráfico 2.14: Mejores prácticas en Programación Extrema³².

- Retroalimentación a escala fina

³² <http://procesosdesoftware.wikispaces.com/METODOLOGIA+XP>

El principio de pruebas.

Es más fácil y comprensivo para el programador que se comience un desarrollo por las pruebas es decir que se desarrolle una parte básica como pueden ser las interfaces y junto al usuario definir el producto final.

Procesos de planificación.

Es la parte de levantamiento de requerimientos, aquí el usuario escribe un documento llamado historias de usuario; uno por cada requerimiento, en un lenguaje natural que es lo que él necesita que haga el sistema, se recomienda que sean entre 20 y 80 historias de usuario dependiendo la complejidad. Junto a estas historias se desarrolla un plan de liberación con tiempos de entrega y prioridades, se recomienda bastante comunicación con el cliente para aclarar todos los puntos que son difíciles o que se consideren puntos críticos del proyecto.

El cliente en el sitio.

Representante del cliente que va a estar en el equipo de desarrollo él será quien revise y pruebe el sistema y hará que vaya acorde al negocio.

Programación en parejas

Escribir código entre dos personas y una computadora según algunos expertos es la mejor manera de hacer buenos sistemas porque mientras uno codifica el otro corrige o ayuda a pensar, o como dice el dicho dos cabezas piensan mejor que una.

- Proceso continuo en lugar de por lotes

Integración continua

La instalación de los incrementos del software deben ser integrados rápidamente.

Refactorización

Los programadores deben analizar el diseño del sistema y codificar estrictamente lo necesario y reusar el máximo posible de código generado.

Entregas pequeñas

Cada 2 o 3 semanas se deben hacer entregas pequeñas para que el usuario pruebe el sistema y se corrijan los defectos rápidamente.

- Entendimiento compartido

Diseño simple

El diseño simple se basa en que se deben enfocar en entregar programas sencillos que cumplan todos los requerimientos, evitar las redundancias y concentrarse en los requerimientos que den mayor valor al programa.

Metáfora.

Este punto reemplaza al UML, consiste en que los programadores al inicio del proyecto definen en su lenguaje cómo funciona el sistema completo. Aquí se expresa la visión, el propósito y el alcance del sistema. También se desarrollan unas tarjetas CRC (Clase, Responsabilidad,

Colaboración) cada una representa una clase de la programación orientada a objetos se define sus responsabilidades (lo que ha a hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas).

Propiedad colectiva del código.

Esto es que un miembro del equipo tiene acceso a todo el código que ha desarrollado todo el equipo, algunos defensores de XP defienden que mientras más gente revise el código menos errores aparecerán

Estándar de codificación

Ya que XP define que el código debe ser de propiedad del equipo, también define que el código debe tener ciertas reglas de escritura, documentación, de tal manera que parezca que el sistema haya sido escrito por una sola persona.

- Bienestar del programador

Semana de 40 horas

Un programador que trabaje demasiado puede bajar la calidad del código que genere, es recomendable que el programador trabaje solo 40 horas semanales y permitir las horas extras solo en casos de extrema necesidad.

Roles

- Programador, es quien desarrolla los prototipos y genera el código del sistema.

- Cliente, escribe las historias de usuario, hace las pruebas funcionales del sistema, define las prioridades del desarrollo y que requerimientos se van a desarrollar en la iteración.
- Encargado de pruebas. Ayuda al cliente con las pruebas funcionales, realiza pruebas regularmente y entrega los resultados al equipo.
- Encargado del seguimiento. Es el encargado de revisar en cada iteración del proyecto el acierto de las estimaciones realizadas, mide los resultados y proporciona una retroalimentación al equipo.
- Consultor. Es una persona externa al equipo tiene un conocimiento especializado en algún tema que es necesario para el proyecto.

Proceso XP

El proceso XP es sencillo, consta de 3 fases iterativas que son:

- Interacción con el cliente.

Es la fase inicial del proyecto donde el cliente describe sus requerimientos, se usan las historias de usuario que son unas tarjetas donde el cliente escribe en su propio lenguaje el requerimiento de cómo él quiere que funcione el software, una vez que termina, el equipo va a calcular el esfuerzo que representa cada requerimiento, por último el cliente va a dar prioridades a cada requerimiento dependiendo cuales son los más importantes y el orden lógico de desarrollo.

- Planificación del proyecto.

Se debe planificar de manera iterativa aquellos puntos importantes para el desarrollo como cada qué tiempo se va a entregar un incremento,

las entrevistas con el cliente, el esfuerzo, la cantidad de horas de trabajo para los programadores, y también hay tener en cuenta que la planificación no siempre va a ser exacta sino que debe estar previsto los cambios que se pueden a los requerimientos y que afecten a la planificación.

- Diseño, desarrollo y pruebas.

Dentro de las mejores prácticas de XP se encuentra las pruebas unitarias, es decir que el programador desarrolla una especie de prototipo con la funcionalidad básica pero que ya pueda ser entendible para el cliente, el programador junto con el cliente prueban y revisan el pequeño avance y luego el programador va a completar la función hasta que esté totalmente aprobado.

La comunicación entre el equipo durante el desarrollo es fundamental para XP, se fomenta el uso de metáforas que son pequeñas historias que los programadores escriben describiendo cual es su visión del producto, como va a funcionar y cuáles son sus componentes.

2.6.3.2.- Metodología de desarrollo Scrum

Scrum más que una metodología de desarrollo es un marco de trabajo que proporciona las mejores prácticas para el desarrollo de software de calidad, donde el gestor del proyecto puede emplear diferentes procesos y técnicas que sean las más apropiadas para el proyecto, se basa más en el resultado final del producto que la documentación generada. A pesar de que Scrum nace para desarrollar nuevos productos tecnológicos se acopla a ambientes con requerimientos muy inestables y que necesiten resultados rápidos como los proyectos de software.

Está basado en la teoría empírica de control de procesos, el proceso va mejorando según la experiencia que se va teniendo en las pruebas. Como todas las metodologías ágiles su desarrollo es iterativo e incremental, se va enfocando gradualmente hacia el objetivo que es el producto final con el fin de pre visualizar los posibles riesgos para el proyecto. La teoría empírica de control de procesos tiene sus bases en 3 pilares³³:

Transparencia.- Todos los procesos que afecten al producto final deben ser comunicados al gestor del proyecto para que tome sus decisiones, siga con su control de resultados y por último establezca las prioridades para el proyecto. Así mismo cuando el gestor menciona que algo está terminado o hecho esto debe ser equivalente a su definición de lo que quiere decir terminado o hecho.

Inspección.- Se inspeccionan los aspectos del proceso para disminuir el riesgo de encontrar diferencias inaceptables. El exceso de inspección puede traer malestar al desarrollo puesto que cada inspección modifica el proceso. La tarea de inspección depende de la habilidad y la diligencia con que las personas realizan esta función.

Adaptación.- Cuando una inspección detecta que el avance o el resultado es inaceptable, el gestor del proyecto debe lo más rápido posible minimizar el impacto que podría tener y tomar una decisión para adaptar el proceso ya

³³http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf

desarrollado y reprogramarlo para obtener el resultado deseado, así se disminuye la desviación del proyecto antes que avance más y el impacto sea mayor.

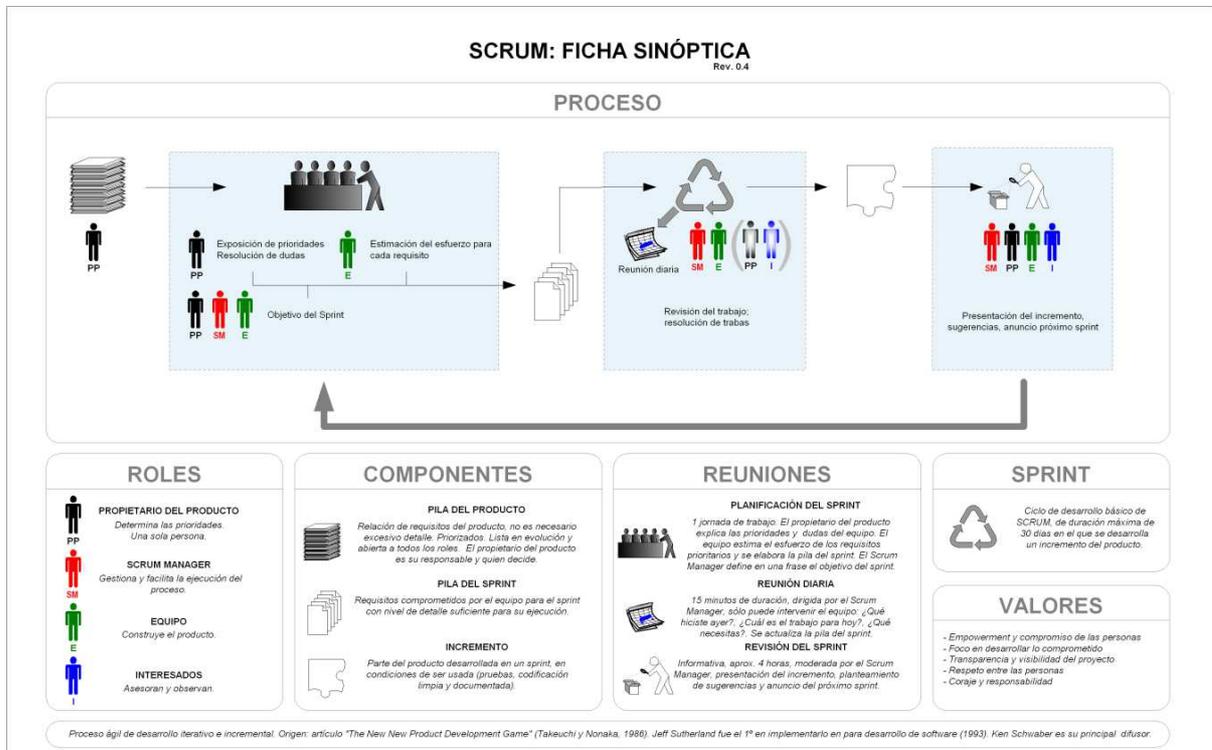


Gráfico 2.15: Proceso completo de Scrum³⁴.

Como se observa en el gráfico 2.15, el proceso consta de roles, componentes y reuniones los mismos que se va a detallar a continuación

Roles^{35 36}

Existen 3 roles importantes que son el Dueño del Producto (Product Owner), Facilitador (Scrum Master) y el Equipo, estas personas van a estar directamente involucradas en el proyecto.

³⁴ http://www.omitsis.com/sites/default/files/images/Ficha_scrum.png

³⁵ www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf

³⁶ <http://sites.google.com/site/oeguzman/losrolesdescrum>

Dueño del Producto: Es quien representa los intereses del cliente, debe estar al tanto del avance del proyecto, y asegurarse que el equipo va desarrollando acorde a la perspectiva del negocio.

Está a cargo de hacer una lista de prioridades de los requerimientos a desarrollar esta lista toma el nombre de Pila del producto (Product Backlog), el Propietario del Producto es el único que puede modificar estas prioridades.

Facilitador (Scrum Master): Su función es limpiar el camino para que el equipo de desarrollo logre las metas del sprint. También está encargado de controlar que el proceso Scrum se lo utilice como es debido. Capacita al equipo a ser más productivos y hacer productos de mayor calidad, ayuda al equipo a comprender y usar la autogestión y la inter-funcionalidad. En conclusión es quien hace cumplir las reglas y hace que el equipo no se desvíe de su objetivo. No es el líder del equipo ya que el equipo Scrum es auto-organizado.

Equipo: Es un grupo pequeño de 5 a 9 personas multidisciplinarias; es decir, que posean el conocimiento y las habilidades necesarias para desarrollar el trabajo, son quienes van a entregar el producto. Cada miembro del equipo debe ser multifuncional para que sea capaz de no solo intervenir en una función sino en varias; por ejemplo, todos deben ser capaces de codificar, así el arquitecto o diseñador no están especializados solo a su función sino que ayuda a la codificación, a las pruebas, al diseño o a lo que sea necesario, de igual manera en Scrum no existen subequipos dentro de los equipos que se dediquen a ciertas

funciones específicas o especializadas como diseño, análisis, codificación, pruebas.

El equipo no tiene un líder definido sino que es auto-organizado, esto es que entre los miembros definen como se va a convertir un requerimiento de la Pila del Producto en un incremento de software funcional y entregable, para esto definen el esfuerzo necesario que van a requerir.

Es recomendable que los miembros del equipo se mantengan durante todo el proyecto, o por lo menos en un sprint, al final de un sprint se puede cambiar un miembro o la composición del equipo, se debe tener cuidado porque la auto-organización del equipo que se ha venido puliendo durante el proyecto puede disminuir hasta que los nuevos integrantes se adapten al modo de trabajo.

A parte de los roles anteriormente mencionados existen cuatro más que no intervienen directamente en el proyecto pero deben ser tomadas en cuenta para revisiones o pruebas entre ellos están el usuario final, expertos del negocio, clientes, proveedores, ellos son importantes porque van a revisar y dar sus observaciones con respecto al resultado del producto y van a ser útiles para corregir errores o cambiar requerimientos.

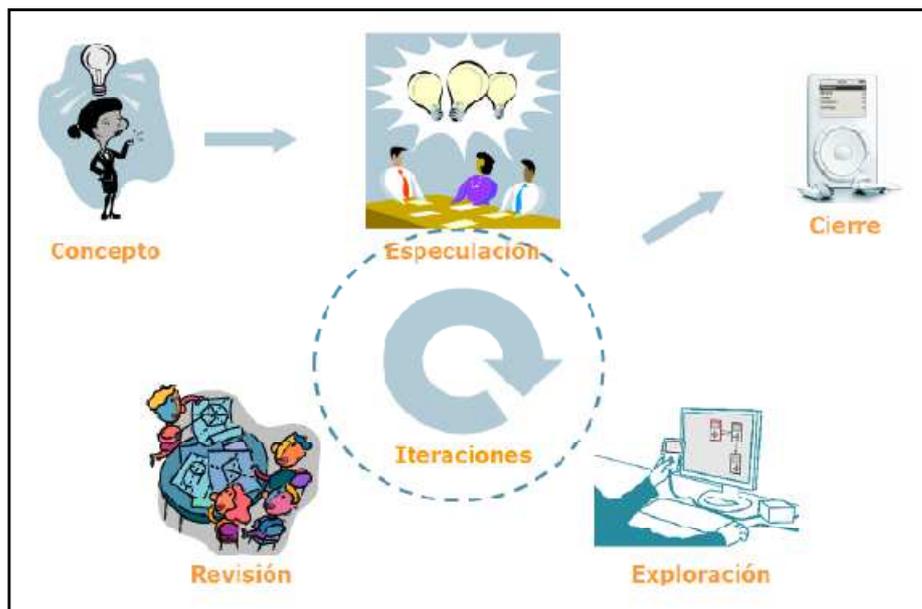


Gráfico 2.15: Fases de la estructura Scrum.

La estructura de Scrum sigue un patrón que consta de 5 fases que son:

- **Concepto**

En esta fase se comienza el proyecto obteniendo la visión y el alcance, es muy importante saber cuál es la visión para saber qué es lo que se quiere obtener y que nomas se necesitaría para hacerlo, además se designan la persona responsable para dirigir el proyecto y quienes van a participar en él. Esta información debe estar en conocimiento de todos los integrantes del equipo

- **Especulación**

Es la primera fase de la iteración del desarrollo, parte con la visión del producto obtenido de la fase del concepto, se especula acerca de la visión, se sacan hipótesis que obviamente van a ser muy generales puesto que la

³⁷ www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf

información no es suficiente para cubrir todas las etapas del desarrollo (requisitos, diseño, costes...). Aquí se valora el proyecto, se arman las agendas y se determina la primera aproximación de lo que se podrá producir, en las siguientes iteraciones se va a ir aproximando al requerimiento.

El alcance de esta fase es:

- Desarrollo / revisión de los requisitos generales del producto.
- Desarrollo de una lista con las funcionalidades esperadas.
- Construcción de un plan de entrega: Fechas en las que se necesitan las versiones, hitos e iteraciones del desarrollo. Este plan refleja ya el esfuerzo que consumirá el proyecto durante el tiempo
- Gestión de riesgos si es que el modelo de gestión lo contempla.

- **Exploración**

Se desarrolla un incremento del producto el mismo que es posible ser entregado y puesto a funcionar, el incremento se define en la etapa de especulación

- **Revisión**

Aquí se revisa el avance del producto hasta donde este, se trabaja con el producto final para revisar que este acorde de los requerimientos.

- **Cierre**

El cierre se lleva a cabo a la fecha que se planteo entregar una versión de producto, a ese punto el producto deseado está listo y saldrá al mercado pero como en todo desarrollo ágil este producto no quedará así sino que se desarrollarán mejoras continuas para evitar que quede obsoleto en poco tiempo.

Los avances incrementales del producto son continuos hasta alcanzar la visión, esto hace que el proyecto sea continuo y no se estanque en una versión única sino que se vaya mejorando en el menor tiempo posible.

Reuniones

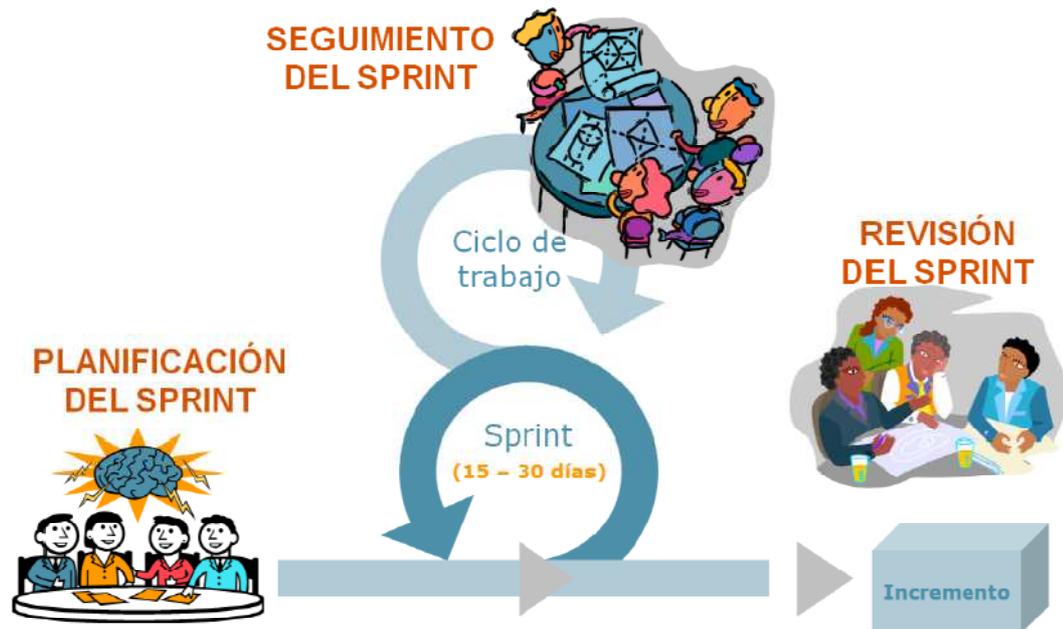


Gráfico 2.16: Reuniones dentro de SCRMS, fases de los sprints.

- Planificación del sprint

Un Sprint es una iteración del desarrollo, en esta reunión se desarrolla antes de cada sprint y se definen los objetivos, se genera una lista de tareas a desarrollar que se llama la Pila del Sprint (Sprint Backlog). Se define también el objetivo del Sprint que viene a ser los intereses del negocio que se van a alcanzar.

- **Sprint**

Un sprint es una serie de actividades iterativas, el objetivo es desarrollar una parte funcional y entregable del producto, tiene una duración máxima de 30 días. Durante un sprint el Facilitador se encarga de que no se realicen cambios que afecten al objetivo del Sprint.

Un sprint muy rara vez se cancela, solo el Propietario del Producto tiene la autoridad de hacerlo, y se lo hace cuando el sprint ha perdido su sentido. Si el sprint ya no tiene sentido puede ser por el cambio de tecnología, cambio de objetivos de mercado. Cuando se cancela un sprint hay que modificar los requerimientos de la Pila del Producto puesto que el sprint pudo haber sido un incremento importante y debe reorganizarse el resto de requerimientos.

- **Seguimiento del sprint**

Diariamente es necesario realizar una reunión para revisar lo desarrollado hasta el momento y definir qué es lo que se va a hacer en la jornada de trabajo y ver cuáles son los problemas que interrumpen el trabajo y deben ser resueltos lo más antes posible. Solo intervienen los miembros del equipo y cada uno responde estas tres preguntas:

- ¿Qué ha realizado desde la reunión anterior?
- ¿Qué va a desarrollar hasta la próxima reunión de seguimiento?
- ¿Cuáles son los impedimentos que se deben solventar para que pueda realizar su trabajo?

- **Revisión del sprint**

En esta reunión se presenta los resultados finales del sprint. Es una reunión informal de cuatro horas para los sprints de 30 días, para sprints menores la reunión no debe ser mayor al 5% del total de tiempo del sprint. Se analizan además del resultado del producto la organización del equipo, las dificultades que hubo y se proponen mejoras a los procesos que se van realizando. El Propietario del Producto discute acerca de los requerimientos faltantes de la Pila del Producto, da una estimación de cuanto falta para concluir con los requerimientos con la experiencia ganada en los sprints concluidos, por último entre todos se ponen de acuerdo en que se debe hacer a continuación, esta reunión es de gran ayuda para la reunión de planificación del sprint.

Artefactos

SCRUM no obliga a generar cierta cantidad de documentos o artefactos si no que pone a decisión del Propietario del Producto elegir los documentos necesarios para el desarrollo del proyecto y que darán mayor valor al mismo. Los siguientes artefactos son los básicos que se necesitan para el desarrollo con SCRUM.

- **Pila del Producto(Product backlog)**

Es una lista que contiene todos los requerimientos que van a ser desarrollados en los Sprints. El Propietario del Producto es el encargado de organizar e ir dando prioridades a la lista de requerimientos. Se listan las características, funciones, tecnologías, mejoras y correcciones de errores que son

los cambios que se harán al proyecto para futuras versiones. La lista de requerimientos nunca va a estar completa desde el inicio, sino que según se avance el proyecto se van a ir puliendo hasta alcanzar los requerimientos definitivos.

- **Product Backlog Burndown.**

Es un gráfico de seguimiento del proyecto, en función del esfuerzo estimado versus el esfuerzo real, su objetivo es mostrar el esfuerzo gastado y la disminución del trabajo en un punto del proyecto.

- **Pila del Sprint.**

Es una lista de actividades que el Equipo define como necesarias para alcanzar el objetivo del Sprint. Al igual que la Pila del Producto se tiene un orden de prioridades asignados por el Equipo.

- **Incremento.**

En un sprint se encarga al Equipo en entregar un incremento funcional del sistema, este incremento debe ser definido como “hecho” o “completado” basado en una definición propia de hecho, es decir, que elementos se necesitan que estén generados en su totalidad (Ejecutables, manuales técnicos, manuales de usuario, diagramas, etc.).

2.6.3.3.- Comparación entre XP y SCRUM y elección de la metodología a usar.

Se ha revisado a XP y SCRUM como las posibles metodologías a usar en este desarrollo, siendo las más aceptables dentro de las comunidades ágiles por la documentación que se puede encontrar tanto en el internet como en libros, pero, una vez que ya se ha revisado cada una de estas metodologías se debe compararlas para analizar cuál es la que más se adapta al ambiente de desarrollo de este proyecto.

En primer lugar se puede decir que SCRUM es un proceso más completo en cuanto a la administración del proyecto porque es una metodología de administración ágil, mientras que XP se enfoca más en el proceso de ingeniería por lo que se deduce que podrían ser metodologías complementarias.

Se puede notar en SCRUM que su enfoque principal está en la transparencia, la inspección y la adaptabilidad a los cambios, mientras que XP tiene su enfoque en el desarrollo del producto proveyendo mejores prácticas para ello.

Al ser ambas metodologías ágiles que cumplen con el manifiesto ágil en cuanto a las entregas de producto, a desarrollar software funcional y disminuir la documentación, pero a pesar de ello XP provee un mecanismo para remplazar el uso de UML mientras que SCRUM deja abierta la posibilidad de desarrollar los artefactos que sean necesarios para el proyecto el único requisito es que agreguen valor al proyecto. El número de roles que usa cada metodología en fin es un número menor a una metodología tradicional como RUP, sin embargo SCRUM ocupa 3 roles principales más 4 roles de menor importancia, XP por su

lado ocupa 6 roles donde todos están comprometidos y son fundamentales en el proyecto. La forma de comunicación con el cliente difiere entre SCRUM y XP. En SCRUM el Propietario del Producto es el mediador entre el cliente y el equipo, puede ser o no parte del equipo, en XP en cambio el cliente es parte del equipo y esta siempre disponible para pruebas del sistema. En las dos metodologías el Propietario del producto en SCRUM o el Cliente en XP son quienes van indicando el orden de desarrollo de los requerimientos y son responsables de que no se pierda la visión del negocio del producto final.

Como conclusión se va a usar la metodología de desarrollo SCRUM porque se enfoca más en la administración del proyecto y no impone que documentos deben ser desarrollados sino que deja al criterio del equipo el análisis de que artefactos van a ser de mayor importancia y cuáles van a proveer mayor valor, además permite el enfoque gradual hacia el producto final, la adaptabilidad de los cambios y el control del riesgo mediante la comunicación continua entre el equipo y el cliente.

2.7.- Sistemas de inteligencia de negocios.

2.7.1.-Introducción

Con el crecimiento de las empresas en forma simultánea crece su información y muchas de las interrogantes que se hacen en ellas tienen sus respuestas ocultas en tablas de Excel, bases de datos, e informes atrasados. Para encontrarlas, hay que tener la voluntad y tiempo de buscar los datos, la capacidad de analizarlos, y tomar las decisiones, el acceso a los datos no ha dejado de ser dificultoso, e incluso puede ser generador de conflictos.

El origen de los datos es muchas veces lejano para un mando medio, y más aún para los directivos de la empresa. Los registros, aún en los actuales tiempos, siguen redactándose a mano la mayoría de las veces y su traspaso a un sistema informatizado suele ser lento.

La información disponible en los sistemas por lo general es notablemente fragmentaria y su acceso depende de la estrecha colaboración entre departamentos, lo que dificulta todo intento de tener una vista global de la empresa.

A esta problemática se le ha encontrado una solución que es el acceder a la información presentada en un formato fácil de interpretar y que permite la toma de decisiones basada en datos. La solución se ha ido desarrollando a la par de los avances que se han hecho posibles a nivel de herramientas de software y las capacidades de procesamiento de las bases de datos. Esta solución es la denominada “Inteligencia de Negocios (BI)”.

En un primer nivel, se trata de acceder a la información contenida en diferentes bases de datos dentro de la empresa, que se agrupan para presentar indicadores claves de desempeño en las áreas críticas de cada uno de sus departamentos.

A esta capacidad de trabajar con múltiples bases de datos para poder realizar un análisis en tiempo real tanto históricos como proyectivos, se denomina “Cubos de Información”; y se ha convertido en una necesidad cada vez en mayor en empresas medianas y grandes.³⁸

³⁸ <http://www.gestiopolis.com/canales2/gerencia/1/busint.htm>

Cuando ya se cuentan con años de registros de datos, se vuelve clave el realizar un análisis de la información para mejorar la toma de decisiones. Si bien la tecnología que sustenta estas capacidades es relativamente compleja, sus beneficios son visibles y pueden ser presentados de manera rápida y muy simple para las directivas de las Empresas.

La Inteligencia de Negocios o Business Intelligence (BI) se puede definir como el proceso de analizar los bienes o datos acumulados en la empresa y extraer una cierta inteligencia o conocimiento de ellos. BI apoya a las Directivas con la información correcta, en el momento y lugar correcto, lo que les permite tomar mejores decisiones de negocios. La información adecuada en el lugar y momento adecuado incrementa efectividad de cualquier empresa.

Con Inteligencia de Negocios (BI) las empresas pueden generar bases de datos de clientes, crear reportes, generar escenarios que den una mejor perspectiva con respecto a una decisión, hacer pronósticos de ventas y devoluciones, compartir información entre departamentos, análisis multidimensionales, generar y procesar datos, cambiar la estructura de toma de decisiones, mejorar el servicio al cliente.³⁹

BI comprende un conjunto de aplicaciones, tecnologías y plataformas de software que permiten a las empresas obtener información precisa y oportuna de sus operaciones diarias, para así poder tomar decisiones racionales que llevan a acciones estratégicas más eficientes y acciones operativas más eficaces.

Las aplicaciones de inteligencia de Negocios incluyen actividades como:

³⁹ <http://www.monografias.com/trabajos14/bi/bi.shtml>

- **CONSOLIDACIÓN DE DATOS:** DATA WAREHOUSE
- **CONSULTAS DINÁMICAS:** OLAP
- **VISIÓN ESTRATÉGICA:** TABLERO DE COMANDOS Y PLANIFICACIÓN
- **EXPLORACIÓN AVANZADA:** MINERÍA DE DATOS

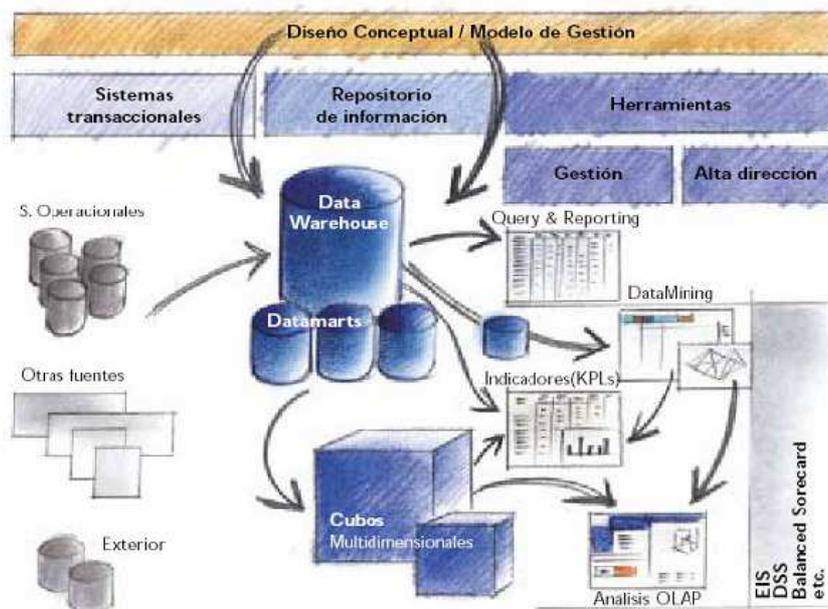


Gráfico 2.17: Modelo Integral de una Solución BI⁴⁰.

2.7.2.- Componentes de Inteligencia de Negocios

2.7.2.1.- DataWareHouse

En tecnologías de la información, un almacén de datos (datawarehouse) es una colección de datos orientada a un determinado ámbito, integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza, es una base de datos corporativa que se caracteriza por integrar y depurar información de una o más fuentes distintas, para luego procesarla

⁴⁰ <http://www.eoi.es/blogs/alexisfedericoreyes/2012/04/24/business-intelligence/>

permitiendo su análisis desde infinidad de perspectivas y con grandes velocidades de respuesta.

Se lo puede definir como un expediente completo de una organización, más allá de la información transaccional y operacional, almacenado en una base de datos diseñada para favorecer el análisis y la divulgación eficiente de datos. Un datawarehouse representa en la mayoría de las ocasiones el primer paso, desde el punto de vista técnico, para implantar una solución completa y fiable de Business Intelligence.

La ventaja principal de este tipo de bases de datos radica en las estructuras en las que se almacena la información. Este tipo de persistencia de la información es homogénea y fiable, y permite la consulta y el tratamiento jerarquizado de la misma (siempre en un entorno diferente a los sistemas operacionales).

La finalidad del DataWarehouse o Almacén de datos es reunir y consolidar las bases de datos diferentes, que se mantienen en los diferentes departamentos o áreas funcionales de la empresa, como subsistemas de información independientes, en una gran base de datos, recogiendo datos muy dispares y, muchas veces infrautilizados, procedentes de fuentes internas, repartidas por toda la organización. También, recolecta datos o informaciones externas que, rutinariamente, se reciben sobre las diferentes entidades u objetos de información, es decir, clientes, proveedores, productos y servicios, canales, estructura organizativa, competencia, mercado, coyuntura económica, etc., en resumen, los derivados de las relaciones de la empresa con su entorno.

A diferencia de las bases de datos tradicionales, las cuales son vistas en dos dimensiones, los DataWarehouse tienen una estructura tridimensional que le da forma de cubo, lo que permite agrupar la información en base a más variables, lo que la hace más útil en la búsqueda, así se puede por ejemplo, obtener la cantidad referente a generación eléctrica en un tiempo determinado y en una hora determinada, o la generación eléctrica dimensionada por generador y hora.⁴¹

Un datawarehouse se caracteriza por ser:

- **Integrado:** Realiza una integración de los datos provenientes de bases de datos distribuidas por las diferentes unidades de la organización y que con frecuencia tendrán diferentes estructuras. Se debe facilitar una descripción global y un análisis comprensivo de toda la organización en el almacén de datos. La información suele estructurarse también en distintos niveles de detalle para adecuarse a las distintas necesidades de los usuarios.
- **Temático:** sólo los datos necesarios para el proceso de generación del conocimiento del negocio se integran desde el entorno operacional. Los datos se organizan por temas para facilitar su acceso y entendimiento por parte de los usuarios finales. Por ejemplo, todos los datos sobre clientes pueden ser consolidados en una única tabla del datawarehouse. De esta forma, las peticiones de información sobre clientes serán más fáciles de responder dado que toda la información reside en el mismo lugar.
- **Histórico:** el tiempo es parte implícita de la información contenida en un datawarehouse. En los sistemas operacionales, los datos siempre reflejan

⁴¹ <http://www.monografias.com/trabajos75/inteligencia-negocios/inteligencia-negocios.shtml>

el estado de la actividad del negocio en el momento presente. Por el contrario, la información almacenada en el datawarehouse sirve, entre otras cosas, para realizar análisis de tendencias. Por lo tanto, el datawarehouse se carga con los distintos valores que toma una variable en el tiempo para permitir comparaciones.

- **No volátil:** La información no se modifica ni se elimina, una vez almacenado un dato, éste se convierte en información de sólo lectura, y se mantiene para futuras consultas. La información es por tanto permanente, significando la actualización del datawarehouse la incorporación de los últimos valores que tomaron las distintas variables contenidas en él sin ningún tipo de acción sobre lo que ya existía.

Otra característica del datawarehouse es que contiene metadatos, es decir, datos sobre los datos. Los metadatos permiten saber la procedencia de la información, su periodicidad de refresco, su fiabilidad, forma de cálculo... etc.

Los metadatos serán los que permiten simplificar y automatizar la obtención de la información desde los sistemas operacionales a los sistemas informacionales.

El metadato documenta, entre otras cosas, qué tablas existen en una base de datos, qué columnas posee cada una de las tablas y qué tipo de datos se pueden almacenar. Los datos son de interés para el usuario final, el metadato es de interés para los programas que tienen que manejar estos datos. Sin embargo, el rol que cumple el metadato en un entorno de almacén de datos es muy diferente al rol que cumple en los ambientes operacionales.⁴²

⁴² http://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos

Los objetivos que deben cumplir los metadatos, según el colectivo al que va dirigido, son:

- **Dar soporte al usuario final**, ayudándole a acceder al datawarehouse con su propio lenguaje de negocio, indicando qué información hay y qué significado tiene. Ayudar a construir consultas, informes y análisis, mediante herramientas de BI.
- **Dar soporte a los responsables técnicos del datawarehouse en aspectos de auditoría**, gestión de la información histórica, administración del datawarehouse, elaboración de programas de extracción de la información, especificación de las interfaces para la realimentación a los sistemas operacionales de los resultados obtenidos.

Es importante entender cuál es el proceso de construcción del datawarehouse, denominado ETL (Extracción, Transformación y Carga):

- **Extracción:** Este proceso recopila la información buscando en las distintas fuentes internas y externas.
- **Transformación:** Procesa la información obtenida en el proceso de extracción filtrando, limpiando, depurando, homogeneizando y agrupando la información.
- **Carga:** Organiza y actualiza la información dentro de la base de datos.

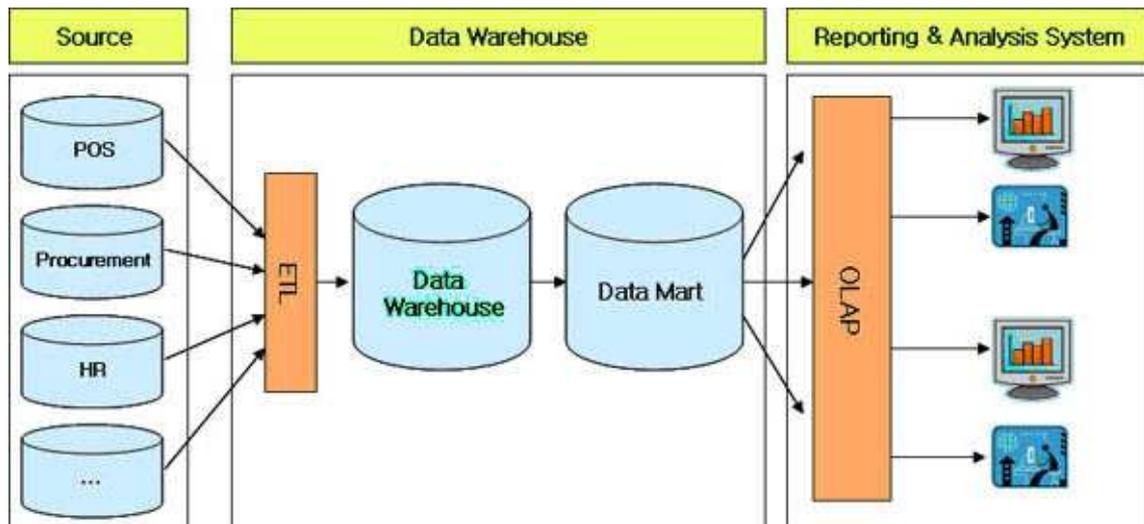


Gráfico 2.18: Esquema de funcionamiento de un Data Warehouse⁴³

Los datos de origen están en los diferentes Sistemas Operacionales, a la izquierda del esquema. Mediante ciertos programas, se Extraen de allí, se Transforman para hacerlos coherentes unos con otros, y por fin se Cargan en las Bases de Datos que componen el Data Warehouse.

Una vez éste cargado, se puede replicar parte de la información para cargarla en otras tablas más pequeñas y especializadas, orientadas generalmente a un Departamento o Área concreto de la empresa, y que proveen ni más ni menos que una vista parcial del Data Warehouse. A estos “mini-Data-Warehouses” se les conoce como Data Marts, y suelen ser más pequeños en tamaño, y más manejables, que los Data Warehouses de los que proceden.

Posteriormente los usuarios, acceden a los datos del uno o de los otros, en función de su conveniencia, bien para obtener información de algo que preguntan

⁴³ <http://www.monografias.com/trabajos75/inteligencia-negocios/inteligencia-negocios2.shtml>

(Query&Reporting), donde se consigue como salida un informe o gráfico con la información solicitada, bien para navegar por la información contenida allí, mediante la técnica conocida como OLAP, de Online Analytical Processing, que es como el online de toda la vida (OLTP), pero sobre la información de negocio contenida en los Data Warehouses o Data Marts. La idea es que, una vez hecho cierto informe en el que se detecta que pasa algo interesante en una o varias líneas del mismo, se pueda navegar jerárquicamente, para obtener más detalles de ese interesante hecho encontrado, hasta poder determinar sus causas últimas.⁴⁴

2.7.2.2.- OLAP Procesos de Análisis en Línea

Los servicios OLAP son utilizados para agilizar la consulta de grandes cantidades de datos, constituyen un ambiente de proceso equivalente a las bases de datos y almacenamiento de datos, se encarga de recolectar, organizar y presentar la información de acuerdo con un modelo que facilite su explotación, requiriendo de un motor semejante a los manejadores de base de datos denominado “servidor OLAP”, encargado de brindar una función de sistema de soporte a las decisiones (SSD) a partir del análisis multidimensional de datos que satisfacen los requerimientos de una amplia variedad de usuarios. La razón de usar OLAP para las consultas es la velocidad de respuesta.

El proceso analítico en línea es un método ágil y flexible para organizar datos, especialmente metadatos, sobre un objeto o jerarquía de objetos como en un sistema o una organización multidimensional para poder ser manejado y

⁴⁴ http://www.sinnexus.com/business_intelligence/datawarehouse.aspx

recuperar recíprocamente los datos en cualquier combinación de dimensiones y/o de atributos para reporting o queries producto de la integración de las dimensiones seleccionadas sobre el objeto

Los sistemas OLAP son bases de datos orientadas al procesamiento analítico. Los objetivos fundamentales en este entorno son el análisis de la información, es decir, el sistema en que se almacene la información ha de proveer de funciones de consulta analítica y de apoyo a la toma de decisiones. Este análisis suele implicar, generalmente, la lectura de grandes cantidades de datos para llegar a extraer algún tipo de información útil se enmarcan en lo que se podría llamar bases de datos corporativas.

Se compone de estructuras de datos y motores encargados de su administración y explotación. El flujo de operación inicia con la extracción de la información registrada en el depósito para organizarla en una estructura denominada almacén OLAP, a partir de ella se producen las “tablas pivote” cuyo contenido y formato está adaptado a facilitar los tipos de consulta que el usuario espera realizar.

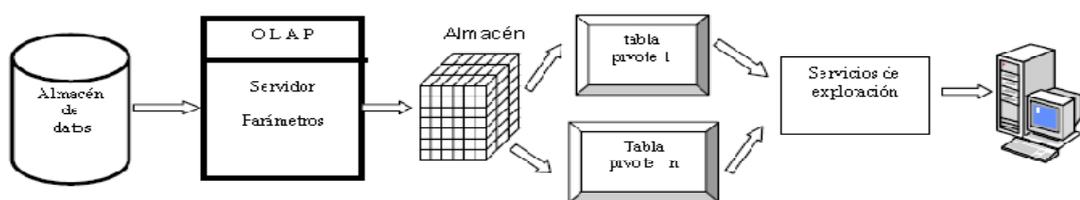


Gráfico 2.19: Modelo de un ambiente OLAP⁴⁵

Es el programa que provee los servicios de análisis de información a través de la definición de una serie de parámetros que especifican el modelo de almacén

⁴⁵ <http://www.monografias.com/trabajos75/inteligencia-negocios/inteligencia-negocios2.shtml>

de datos OLAP. En los sistemas OLAP la información ya no se almacena en tablas, sino en cubos de n dimensiones. Se construye a partir de una tabla principal, llamada tabla de hechos o tabla pivote, que enumera los hechos ciertos en el sistema y los procesos que responden a los planteamientos del tomador de decisiones. A ella se va añadiendo una dimensión por relación cuya información que se quiera almacenar, de manera que todas están pre calculadas en el sistema. En la base de cualquier sistema OLAP se encuentra el concepto de cubo OLAP (cubo multidimensional o hipercubo). Se compone de hechos numéricos llamados medidas que se clasifican por dimensiones. El cubo de metadatos es típicamente creado a partir de un esquema en estrella o copo de nieve, esquema de las tablas en una base de datos relacional. Las medidas se obtienen de los registros de una tabla de hechos y las dimensiones se derivan de la dimensión de los cuadros.

El modelo de almacenamiento OLAP el área donde se organiza la información de acuerdo a un modelo cúbico de diversas dimensiones, que procura optimizar la respuesta a cierto tipo de requerimientos de explotación. Los usuarios y aplicaciones reciben una “vista” (son perspectivas de información que se ponen al alcance del interesado) del cubo, a pesar de que internamente cuente con alguna de las tres clases de arquitectura OLAP.

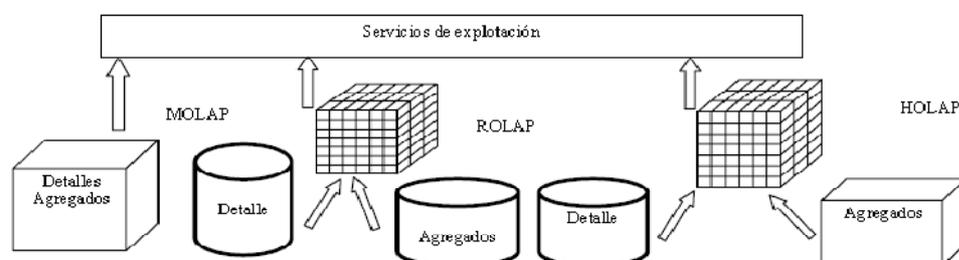


Gráfico 2.20: Arquitecturas OLAP

- **Proceso analítico en línea multidimensional (MOLAP).**

Este proceso almacena los datos en una base de datos multidimensional, el modelo está especializado en tareas de consulta sin alteración de la información original, emplea una estructura multidimensional con los datos agregados y los de detalle. Para optimizar los tiempos de respuesta, el resumen de la información es usualmente calculado por adelantado. Estos valores pre calculados o agregaciones son la base de las ganancias de desempeño de este sistema. Algunos sistemas utilizan técnicas de compresión de datos para disminuir el espacio de almacenamiento en disco debido a los valores pre calculados.

- **Proceso analítico en línea relacional (ROLAP).**

Almacena los datos en un motor relacional, esta arquitectura separa los datos agregados en una estructura relacional y los de detalle se mantienen en el depósito original para efectos de registrar actualizaciones, típicamente, los datos son detallados, evitando las agregaciones y las tablas se encuentran normalizadas. La arquitectura está compuesta por un servidor de banco de datos relacional y el motor OLAP se encuentra en un servidor dedicado. La principal ventaja de esta arquitectura es que permite el análisis de una enorme cantidad de datos.

- **Proceso analítico en línea híbrido (HOLAP).**

Combina las ventajas de los modelos multidimensional y relacional, al mantener los datos de detalle en una estructura relacional y los agregados en MOLAP, ofreciendo diversas vistas virtuales que enlazan los dos modelos.

Es un marco de servicio compuesta por un área temporal de almacenamiento de datos para un tipo de consulta específica – conocida como “cache” – y un motor responsable de su carga, organización, acceso y mantenimiento, con el propósito de reducir la carga de transferencia de la información durante la explotación y aumentar la rapidez de los procedimientos de análisis que el operador ejecuta aún sin estar conectado en línea al ambiente de trabajo.⁴⁶

Por lo general MOLAP ofrece mejor rendimiento debido a la especializada indexación y a las optimizaciones de almacenamiento. MOLAP también necesita menos espacio de almacenamiento en comparación con los especializados ROLAP porque su almacenamiento especializado normalmente incluye técnicas de compresión.

ROLAP es generalmente más escalable. Sin embargo, el gran volumen de pre-procesamiento es difícil de implementar eficientemente por lo que con frecuencia se omite; por tanto, el rendimiento de una consulta ROLAP puede verse afectado.

Desde la aparición de ROLAP van apareciendo nuevas versiones de bases de datos preparadas para realizar cálculos, las funciones especializadas que se pueden utilizar tienen más limitaciones.

HOLAP (OLAP Híbrido) engloba un conjunto de técnicas que tratan de combinar MOLAP y ROLAP de la mejor forma posible. Generalmente puede pre-procesar rápidamente, escala bien, y proporciona una buena función de apoyo.⁴⁷

⁴⁶ <http://www.monografias.com/trabajos75/inteligencia-negocios/inteligencia-negocios2.shtml>

⁴⁷ <http://es.wikipedia.org/wiki/OLAP>

2.7.2.3.- DATAMINING (minería de datos)

Es una de las principales herramientas que se utilizan dentro de los programas de gestión del conocimiento como soporte a la toma de decisiones, son auténticas herramientas de extracción de conocimiento útil, a partir de la información contenida en las bases de datos de cualquier empresa. El fin es la extracción de información oculta, tendencias y correlaciones o análisis de datos mediante técnicas estadísticas de grandes bases de datos de forma sencilla y accesible a los usuarios finales, para solucionar, prever y simular problemas del negocio. El datamining incorpora la utilización de tecnologías basadas en redes neuronales, árboles de decisión, reglas de inducción, análisis de series temporales y visualización de datos. Las herramientas de datamining o minería de datos pueden responder a preguntas de negocios empresariales a priori no planteadas o que pueden consumir demasiado tiempo para ser resueltas.

Es el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, que en una primera selección pueden ser pertinentes, pero que la aplicación de técnicas de selección ceñida a una determinada demanda, reduce el tamaño de los datos eligiendo las variables más influyentes en el problema, básicamente para intentar ayudar a comprender el contenido de un repositorio de datos.

De forma general, los datos son la materia prima bruta. En el momento que el usuario les atribuye algún significado especial pasan a convertirse en información. Cuando los especialistas elaboran o encuentran un modelo, haciendo que la interpretación que surge entre la información y ese modelo represente un valor agregado, entonces se refiere al conocimiento.

Aunque en datamining cada caso concreto puede ser radicalmente distinto al anterior, el proceso común a todos ellos se suele componer de cuatro etapas principales:

- **Determinación de los objetivos.** Objetivos que el cliente desea bajo la orientación del especialista en datamining.
- **Pre procesamiento de los datos.** Se refiere a la selección, la limpieza, el enriquecimiento, la reducción y la transformación de las bases de datos. Esta etapa consume generalmente alrededor del 70% del tiempo total de un proyecto de datamining.
- **Determinación del modelo.** Se comienza realizando unos análisis estadísticos de los datos, y después se lleva a cabo una visualización gráfica de los mismos para tener una primera aproximación. Según los objetivos planteados y la tarea que debe llevarse a cabo, pueden utilizarse algoritmos desarrollados en diferentes áreas de la Inteligencia Artificial.
- **Análisis de los resultados.** Verifica si los resultados obtenidos son coherentes y los coteja con los obtenidos por los análisis estadísticos y de visualización gráfica. El cliente determina si son novedosos y si le aportan un nuevo conocimiento que le permita considerar sus decisiones.⁴⁸

2.7.3.- Beneficios de Inteligencia de Negocios

Las aplicaciones de BI han evolucionado drásticamente en muchas direcciones, debido al crecimiento exponencial de la información. Desde reportes

⁴⁸ <http://www.monografias.com/trabajos75/inteligencia-negocios/inteligencia-negocios.shtml>

operacionales generados por mainframes, modelación estadística de campañas publicitarias, ambientes OLAP multidimensionales para analistas, los beneficios obtenidos de implementar esta solución tecnológica son muchos, entre ellos se pueden citar los siguientes:

- **Mejor calidad y precisión de la información**, por lo tanto crea una lealtad de los clientes puesto que las aplicaciones que buscan tendencias, ayudan a identificar y entender “qué” son los mejores o más rentables clientes y el “por qué” los clientes se quedan o abandonan la empresa por lo tanto genera una mejor explotación y aprovechamiento de toda la información de la empresa.
- **Administración del Riesgo**. Especialmente las entidades financieras y las aseguradoras, pero también otras empresas, se requiere calcular riesgos y simular escenarios en base a las cifras con que se cuentan. Las aplicaciones BI ayudan a encontrar patrones de comportamiento en los datos y tomar decisiones más adecuadas para disminuir el riesgo, por lo tanto mejora la relación con sus clientes, socios y equipo de trabajo ayudando a descubrir tendencias escondidas entre grandes volúmenes de datos, desde la detección de fraude hasta el análisis de la cartera de clientes.
- **Cultura de Comunicación**. Crea una cultura de comunicación entre los departamentos de la empresa al tener que compartir información entre sus distintas áreas, fortaleciendo sus capacidades analíticas y de planificación puesto que se le proporciona Facilidad, flexibilidad, y poder de análisis en la

navegación de información obtenida al instante reduciendo así sus costos administrativos y de preparación de informes elevando su competitividad al ser más eficiente en sus procesos

- **Análisis de Costos según Actividad.** Descubrir los costos reales de los productos, servicios o clientes más allá de la contabilidad tradicional y descubre por ejemplo que 25% de la actividad de la empresa causa 80% de los costos.
- **Avisos y Alertas.** Se podrían vigilar automáticamente los indicadores de la operación y generar avisos por email o SMS si un indicador llama la atención, a cualquier momento. Ya no es necesario leer cada mañana muchas páginas de informes o mirar cada hora una pantalla causando una considerable reducción en los tiempos de espera y por consiguiente un retorno de inversión a corto y mediano plazo.⁴⁹

Las aplicaciones de análisis de negocio, siguiendo en modelo de Cuadro de Mando Integral (CMI), pueden estar presentes en todas las áreas de la empresa, analizando diferentes aspectos:

- **Marketing:** El BI permite identificar de forma más precisa los segmentos de clientes y estudiar con mayor detalle su comportamiento. Para ello se

⁴⁹ http://www.mextisa.com.mx/site/index.php?option=com_content&view=article&id=7&Itemid=7&limitstart=1

pueden incluir análisis capaces de medir, por ejemplo, el impacto de los precios y las promociones en cada segmento.

- **Compras:** El BI permite acceder a los datos del mercado, vinculándolos con la información básica necesaria para hallar las relaciones entre coste y beneficio. Al mismo tiempo, permite monitorizar la información de cada línea o cadena de producción, lo que puede ayudar a optimizar el volumen de las compras.
- **Producción:** El BI proporciona un mecanismo que permite analizar el rendimiento de cualquier tipo de proceso operativo, ya que comprende desde el control de calidad y la administración de inventarios hasta la planificación de la producción.
- **Ventas:** El BI facilita la comprensión de las necesidades del cliente, así como responder a las nuevas oportunidades del mercado. También son posibles análisis de patrones de compra para aprovechar coyunturas de ventas con productos asociados.
- **Económico-financiero:** El BI permite acceder a los datos de forma inmediata y en tiempo real, mejorando así ciertas operaciones, que suelen incluir presupuestos, proyecciones, control de gestión, tesorería, balances y cuentas de resultados.

- **Atención al cliente:** Aplicado a este ámbito, el BI permite evaluar con exactitud el valor de los segmentos del mercado y de los clientes individuales, además de ayudar a retener a los clientes más rentables.
- **Recursos humanos:** Obteniendo los datos precisos de la fuente adecuada, el BI permite analizar los parámetros que más pueden afectar al departamento: satisfacción de los empleados, absentismo laboral, beneficio-hora/hombre... etc.⁵⁰

⁵⁰ <http://www.trevenque.es/index.php/BI-en-las-diferentes-areas-empresariales/471/0/>

CAPÍTULO 3

SELECCIÓN DE LA SUITE DE INTELIGENCIA DE NEGOCIOS

3.1.- Herramientas utilizadas en una Suite de Inteligencia de Negocios.

Una suite de inteligencia de negocios es muy extensa y tiene una integración de varias herramientas que facilitan el trabajo de BI, entre las diversas herramientas se tiene:

3.1.1.- OLAP

Son arreglos de datos que permiten un análisis más rápido cuando se tiene gran cantidad de datos, se basan en el análisis dimensional y el concepto de hipercubo.

3.1.2.- Scorecarding (marcador)

El "scorecarding" es un sistema de gestión estratégico que obliga a los gestores a centrar su atención en las cifras de rendimiento que conducen al éxito y equilibra las perspectivas financieras con el cliente, proceso interno y las perspectivas de aprendizaje y crecimiento. El sistema consta de cuatro procesos:

3.1.3.- Dashboarding (Tablero de instrumentos)

Es la presentación de los datos a través de interfaces gráficas de fácil acceso y comprensión al usuario.

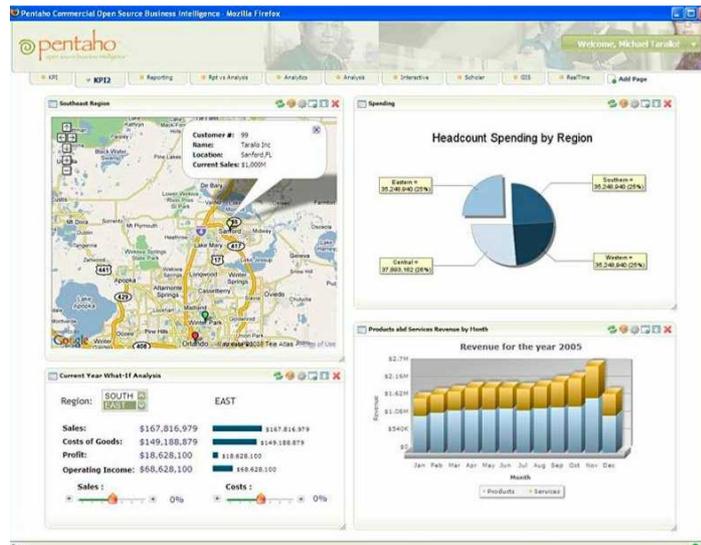


Imagen 3.1: Ejemplo de presentación de datos con Pentaho

3.1.4.- Información de visualización

Se refiere a la aplicación práctica de visualización de datos, es decir la facilidad de obtener datos, esto implica la clasificación, el procesamiento, y la representación abstracta de datos facilitando la interacción con el usuario.

3.1.5.- Data Modeling

Es armar la estructura de datos conteniendo las reglas del negocio y la lógica del negocio, aquí se unen las estructuras de datos de interés para definir un “todo” coherente, inseparables y eliminando la duplicidad de información.

Herramientas para definir la lógica de negocios

Es la lógica que representa las reglas del negocio, es una herramienta que realiza operaciones especiales sobre los datos transmitidos a través él.

Herramientas para definir las reglas del negocio

Las reglas de negocio, sencillamente, describir las operaciones, las definiciones y las limitaciones que se aplican a una organización en el logro de sus objetivos.

3.1.6.-Data Warehouse (Inventario de datos)

Contiene toda la información de la empresa, almacenada en una base de datos diseñada para realizar un eficiente análisis de datos, por lo general se realiza una consolidación de bases de datos distribuidas que contienen los datos necesarios y poder hacer un análisis global de todo.

3.1.7.-Document Warehouse (almacén de documentos)

A diferencia del Datawarehouse aquí se almacena la información de las razones (“¿porque las cosas han sucedido?”), una vez que se ha determinado en el Data Warehouse una variación en los análisis se ingresa la razón por la que ha sucedido.

3.1.8.-Datamining (Extracción de datos)

Es una técnica que mediante el uso de técnicas computacionales derivados de la estadística y de reconocimiento de patrones realiza búsquedas automáticas de muestras en grandes almacenes de datos.

3.1.9.-Text Mining (Extracción de texto)

Es la extracción o análisis inteligente de texto, clasifica la información y separa lo importante de lo trivial, un claro ejemplo de un sistema de Text Mining es Echelon (ECHELON es considerada la mayor red de espionaje y análisis para interceptar comunicaciones electrónicas de la historia).⁵¹

3.1.10.- EIS – Sistemas de información ejecutiva.

Son sistemas que ayudan a la toma de decisiones por la parte gerencial de la empresa, estos sistemas contienen gráficos, informes de tablas, reportes para que el gerente o el encargado de tomar decisiones lo realicen con mayor precisión.

3.1.11.- DSS – Decision Support Systems

Similar a los sistemas EIS son sistemas que ayudan a la toma de decisiones.

SIG – Sistemas de información geográfica.

Es un sistema donde se puede relacionar un dato con una ubicación geográfica.

3.2.- Lista de criterios a considerar.

Existe una larga lista de criterios que se podrían considerar para el análisis de cada suite pero se debe tomar en cuenta los requerimientos realizados por ELEPCO S.A. para la construcción del Cubo. Entre los requerimientos se tiene:

⁵¹ <http://es.wikipedia.org/wiki/Echelon>

- Licencia open source.
- Interface web
- Multiusuario.
- Que contenga las siguientes herramientas:
 - o Servidor OLAP
 - o Editor de reportes
 - o ETL (Extracción, transformación y carga)
 - o Data Modeling
 - o Editor de metadatos.

3.3.- Suites de Inteligencia de Negocios.

Actualmente existe una gran variedad de productos tanto open source como software propietario para inteligencia de negocios, se eligió 4 suites que poseen las herramientas necesarias solicitadas en los requerimientos del proyecto y que actualmente están entre las suites open source más conocidas y recomendadas, entre las que están:

3.3.1.-Pentaho BI Suite

Pentaho es una herramienta de Business Intelligence desarrollada bajo la filosofía del software libre para la gestión y toma de decisiones empresariales. Es una plataforma compuesta de diferentes programas que satisfacen los requisitos de BI. Ofreciendo soluciones para la gestión y análisis de la información,

incluyendo el análisis multidimensional OLAP, presentación de informes, minería de datos y creación de cuadros de mando para el usuario.⁵²

Pentaho al igual que otras soluciones Open Source ofrecen dos tipos de licenciamiento: Enterprise y Community cuya diferencia radica en que la versión Enterprise tiene un costo y agrega código certificado, software especial para el desarrollo y soporte técnico, mientras que la versión community es totalmente gratuita y se compone por las herramientas básicas.

Entre los programas que integran la solución de Pentaho están:

- **Pentaho Análisis:** en esta herramienta se crean los cubos multidimensionales OLAP, soporta lenguaje de consulta MDX y almacena las estructuras de datos en archivos XML, gestiona comunicación entre una aplicación OLAP (escrita en Java) y la base de datos con los datos fuente⁵³.
- **Pentaho Data Integration:** es la herramienta ETL de Pentaho, también llamada Kettle, gestiona toda la parte de migración de datos y transformaciones, se conecta a las bases de datos mediante JDBC, ODBC, OCI, JNDI lo que le permite tener conexión a la mayoría de servidores de bases de datos. Actualmente Pentaho está mejorando este producto intentando integrar varias herramientas como Pentaho Análisis, Report Designer y Pentaho Metadata Editor y no tener varios programas por separado. En Kettle

⁵² <http://ignaciobustillo.com/blog/blog/63-ique-es-pentaho>

⁵³ <http://pentaho.almacen-datos.com/mondrian.html>

se crean Transformaciones que son procedimientos ETL básicamente es decir; de lectura, transformación y carga de datos, también se crean Trabajos que es un procedimiento que puede tener varias transformaciones además se puede controlar el estado del trabajo si tuvo éxito o no, el diseño de estos trabajos y transformaciones son netamente gráficos, es decir no hace falta conocer mucha programación para realizarlos.

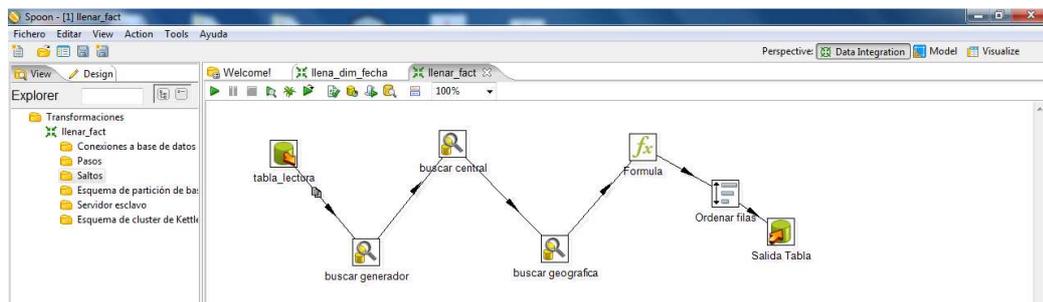


Imagen 3.2: Pantalla de trabajo de Pentaho Data Integration.

- **Pentaho Metadata Editor:** Permite crear una capa de abstracción que presenta la información de base de datos. Se crea un mapeo de la base de datos según las reglas del negocio y se presenta al usuario de forma que sea más fácil para entender y el usuario final sea quien pueda armar sus propios reportes.
- **Report Designer:** Es un programa para crear reportes, la ventaja de este reporteador es que puede conectarse tanto a una base de datos relacional como a un modelo de datos de Pentaho Metadata Editor, y además permite publicar los reportes creados directo en el servidor BI de Pentaho, no tiene un alcance tan bondadoso como otros reporteadores pero tiene varias funciones que son muy útiles entre las que se tiene:

- Gráficos de pasteles, líneas, barras etc.
 - Permite usar parámetros de entrada.
 - Uso de subreportes.
 - Conexión con todas las bases de datos que usen JDBC.
 - Conexión con el servidor OLAP de Pentaho.
 - Conexión con archivos de modelo de datos.
- **Pentaho BI Server:** Herramienta que proporciona el servidor y plataforma web del usuario final. Aquí este podrá interactuar con la solución Business intelligence previamente creada con las herramientas anteriormente comentadas, posee administración de usuarios, roles y conexiones con las bases de datos a usar, además provee un espacio de trabajo para que el usuario pueda crear reportes add hoc y almacenarlos.

3.3.2.-Jasper, de Jaspersoft

Al igual que Pentaho es una herramienta integrada por varios programas, pero con la diferencia que Jasper no los absorbió esto le hace depender de los desarrolladores en cuanto a solución ETL y MONDRIAN, al tener igual acceso al código fuente de Mondrian, Jasper puede adaptar y continuar los desarrollos en cualquier punto de Mondrian.

Entre los programas que integran Jasper se tiene:

- **JasperETL:** es un programa ETL de la empresa Talend, originalmente se llama Talend Studio, este programa no ha sido absorbido por Jasper sigue perteneciendo a Talend a diferencia de Pentaho con Kettle.

Usa una interface visual agradable y muy intuitiva pero es muy técnico por ser una herramienta generadora de código java o perl, es decir; que el código generado se puede compilar y crear un “war” en el caso de Java, está orientado a un usuario de tipo programador con conocimientos técnicos superiores a los de Kettle pero ofrece una mayor flexibilidad.

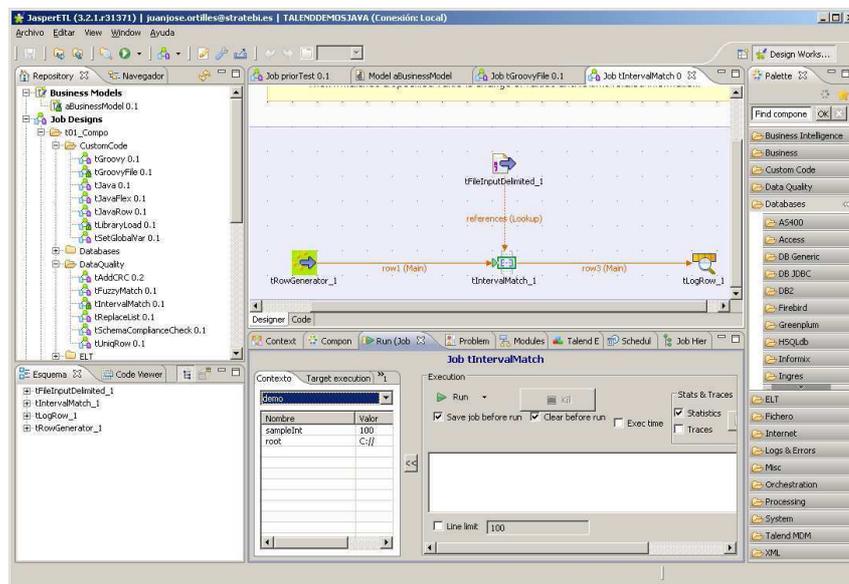


Imagen 3.3: Pantalla de trabajo de JasperETL.

- **Jasper Server:** Es una aplicación web 100% J2EE que permite gestionar todos los recursos BI, es una aplicación totalmente funcional y dispone de todos los recursos para el análisis BI.

Permite crear reportes Add hoc, también gestiona los reportes desarrollados en iReports (reporteador de JasperReports), mantiene la información actualizada y es muy agradable para el usuario final.

- **Informes:** La mejor característica de Jasper son sus generadores de reportes e informes, el entorno de desarrollo de los informes es Ireports el mismo que está basado en NetBeans.

Tiene una interfaz web para creación de reportes ad-hoc, y su motor de ejecución es JasperReports que es usado en muchas aplicaciones Java hoy en día.

- **IReport:** Es el entorno de trabajo para desarrollar reportes, tiene una gran cantidad de funcionalidades como:⁵⁴

- Soporte para TrueType (Formato estándar de tipos de letra).
- Manejo de múltiples fuentes de datos como: todas las bases de datos soportadas por JDBC, archivos XML, CSV, Hibernate entre otros...
- Soporta SQL, HQL, EJBQL, MDX y Xpath.
- Wizard para la creación de reportes y sub-reportes
- Más de 30 elementos para formatear el reporte (líneas, elipses, TextFields, charts, código de barras, etc.)
- Los reportes generados se puede pre visualizar y/o exportar en PDF, HTML, RTF, XLS, CSV, TXT.

⁵⁴ <http://mistock.lcompras.biz/paradigmas23/1812-ireport>

- **Ad-Hoc:** Es un punto fuerte de Jasper si se desea tener acceso a informes rápidos, tiene entre sus opciones:
 - o Selección de diferentes tipos de plantillas y formatos.
 - o Selección de diferentes orígenes de datos.
 - o Validación de la consulta al vuelo.
 - o Creación de los informes arrastrando los campos al lugar deseado.
 - Tablas
 - Gráficos
 - Tablas cruzadas
 - o Edición de todos los aspectos de los informes.
- **OLAP:** El servidor OLAP de Jasper es Mondrian el mismo motor de Pentaho pero Jasper ha realizado unas mejoras en el editor ad-hoc para reportes rápidos.

3.3.3.-BIRT

Birt es la solución BI que comenzó siendo desarrollado por Actuate que luego se fusionó a la Fundación Eclipse al ser un proyecto que le interesó a Eclipse, Actuate donó el proyecto y se convirtió en un proyecto de alto nivel dentro de la comunidad Eclipse. BIRT significa Business Intelligence Reporting Tools, es un plug in para eclipse que permite realizar reportes.

Fue presentado en el año 2005 y tiene las siguientes funciones:

- Informes dinámicos
- Motor OLAP interno

- Tablas cruzadas
- Scripting en diferentes momentos de ejecución de los informes
- Exportación a diferentes tipos de documentos.
- La potencia ilimitada que proporciona poder invocar clases Java.
- Integración con eclipse

Contiene los siguientes módulos:

- **IServer:** Es una aplicación web disponible solo para la versión Enterprise, esta aplicación tiene un aspecto visual menos atractiva que Pentaho y Jasper, y tiene un manejo muy técnico, contiene un organizador de archivos mediante arboles de carpetas y tiene una interfaz para crear nuevos reportes.
- **Eclipse BIRT:** es un generador de reportes muy dinámicos, que pueden cambiar sus valores o aspectos según los parámetros de entrada, además permite inyectar valores en una tabla cruzada, tiene dos componentes que son: un editor visual dentro del IDE Eclipse para diseñar reportes y un componente de rutina para integrar los reportes a otro sistema basado en el entorno de java.
- **OLAP:** Eclipse tiene un novedoso motor OLAP interno que le permite generar sus propios reportes de tablas cruzadas.
- No posee herramientas ETL y tampoco permiten la creación de cuadros de mando.

3.3.4.- Palo Suite de Jedox

Jedox por su lado desarrolló un motor MOLAP que a diferencia de los motores OLAP tradicionales este permite trabajar directamente con los datos del cubo, el usuario puede manipular los datos de tal manera que puede plantear supuestos escenarios y hacer análisis “y si”.

Tiene una estructura distinta al resto de soluciones BI, primero que su motor es MOLAP y contiene varias herramientas para llenarlo y explotarlo, todo ello condicionado por el motor.

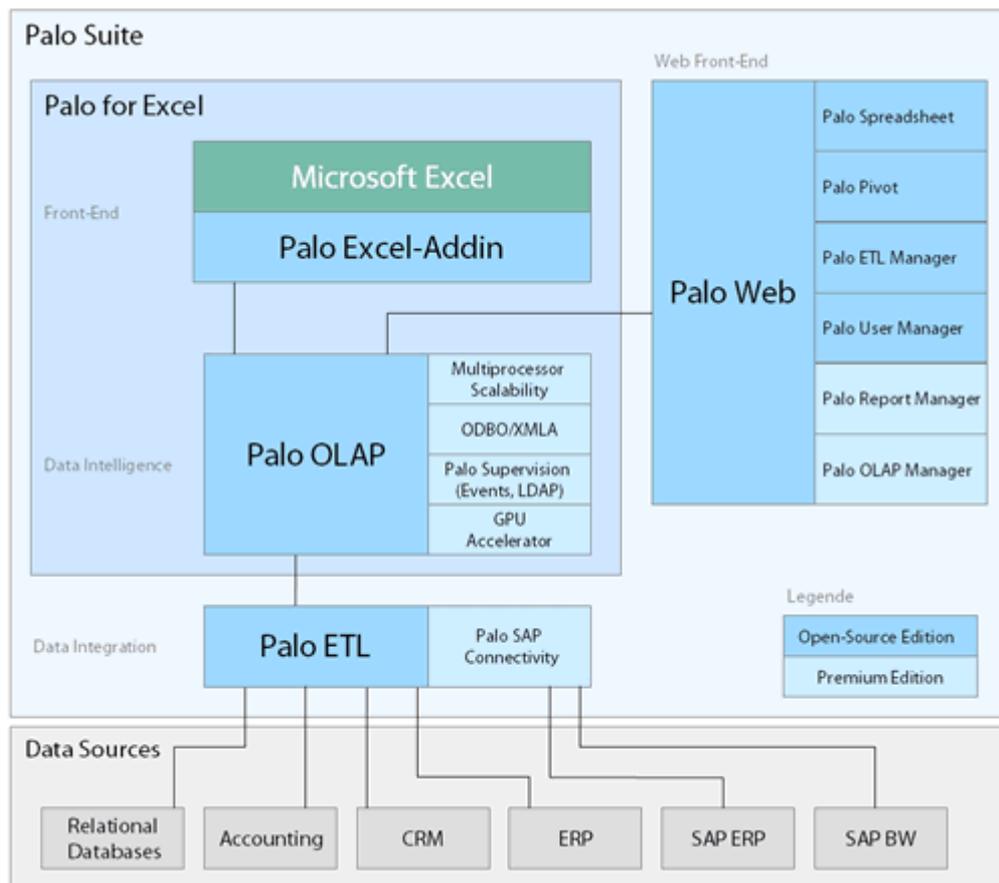


Gráfico 3.1: Módulos de Palo Suite de Jedox

PALO ETL:

Posee todas las transformaciones existentes en las herramientas ETL con la diferencia que PALO ETL contiene varias acciones que son hechas por el servidor automáticamente, es de fácil uso y no requiere de conocimientos avanzados, para el desarrollo se basa en estándares como:

- Encapsulación de ETL en proyectos.
- ETL secuenciales
 - o Todos los trabajos deben mostrar su secuencialidad.
 - o Se invoca el último paso para correr el proyecto y este se encarga de invocar a sus pasos predecesores.

No posee una interface gráfica como el resto de herramientas ETL, sino que tiene formularios por cada paso que se desarrolla, es una herramienta estable pero muy lenta.

PALO WEB:

Palo web es una interface web de administración, tiene dos versiones según su licenciamiento y su diferencia está en el número de herramientas que contiene cada uno:

- Versión Gratuita:
 - o Gestión de usuarios.
 - o Hojas de cálculo.
 - o Vistas OLAP

- Para la versión Enterprise se agregan:
 - o Informes.
 - o Manejo del servidor.

Reportes: PALO Suit dentro de sus aplicaciones contiene un editor de reportes y un organizador de reportes en secciones diferentes, la ventaja principal de PALO en reportes es que tiene todo integrado en una única aplicación lo que hace más fácil integrar, por ejemplo, se puede crear un reporte que ejecute una acción de lectura de cubos, o se puede crear un reporte para ejecutar a un proceso ETL, es muy versátil, se puede agregar programación a los reportes mediante macros, el diseñador de reportes es muy parecido a una hoja de Excel, es muy técnico por lo que se requiere conocimientos de programación para su uso.

PALO for Excel:

Es un plugin para Excel que permite conectar un cubo de PALO y explotar los datos desde Excel.

OLAP:

A diferencia de las otras soluciones BI PALO tiene un motor MOLAP, lo que le permite hacer búsquedas más rápidas y ocupar menor espacio en disco. El motor MOLAP carga todos los datos y pre-calcula todas las intersecciones por lo que no se fundamenta en ninguna base de datos sino que crea sus propios cubos.

Sus principales características son:

- Multidimensional in-memory.
- Agregación en tiempo real
- Write-back
- Gestión de permisos de usuarios

3.4.- Análisis.

En este apartado se realizarán varias comparaciones entre las suites detalladas en el punto anterior. La siguiente tabla es una comparación entre las características no funcionales, como proveedor, versión, dificultad de instalación, etc.

Tabla 3.1: Comparativa entre las características no funcionales de las Suites BI.

Característica	Pentaho	Jasper	Birt	Palo
Fabricante	Pentaho	JasperSoft	Actuate - Eclipse Foundation	Jedox
Resumen	Es una herramienta integrada por varios proyectos open source que han sido adaptados para trabajar juntos, provee todas herramientas para completar una solución de Business intelligence.	Al igual que Pentaho es una herramienta integrada por varios programas a diferencia que hay ciertas herramientas que no han sido adoptadas por Jasper por lo que se depende de otras compañías como es el caso de Talend con la herramienta ETL	Es un plugin de Eclipse, y consta actualmente como un proyecto de alto nivel dentro de la Fundación Eclipse, no posee herramientas como ETL y tampoco tiene un generador de reportes Ad-hoc	Es un sistema integrado todo en uno, es decir que a diferencia de Pentaho y Jasper tiene todos los programas necesarios integrados en una sola herramienta.
Dificultad de instalación	Alta	Alta	Alta	Media

<p>Versión actual</p>	<p>Pentaho BI Suite: 4.1 Pentaho Data Integration: 4.2.1 Pentaho Analysis: 3.3 Pentaho BI Suite: 4.0 Pentaho Report Designer: 3.8.3 ⁵⁵</p>	<p>JasperRepot Server: 4.2.1 IReport Designer: 4.5.0 JasperETL: 4.1.2 ⁵⁶</p>	<p>Eclipse BIRT Report Designer all in one: 2.6 ⁵⁷</p>	<p>Jedox Suite 3.3 ⁵⁸</p>
<p>Diferencia entre versiones de comunidad y Enterprise</p>	<ul style="list-style-type: none"> - En cuanto a la herramienta ETL Pentaho agrega Agile BI que permite en tiempo de ETL diseñar las vistas OLAP y dashboards (cuadros de mando). - Cuadros de mando CDF con mejores gráficas. - Visor OLAP. -Código certificado y soporte 	<ul style="list-style-type: none"> - Existe una gran diferencia en cuanto a funcionalidades en la herramienta ETL como: Data Quelite, Wizard, Auto documentación, etc. - Informes Ad-hoc - Metadata. -OLAP schema workbench. - Soporte 	<ul style="list-style-type: none"> - No hay versión community del Servidor IServer. - Soporte 	<ul style="list-style-type: none"> - OLAP - Report Manager.
<p>Comunidad de soporte.</p>	<p>Posee una comunidad de soporte en varios idiomas en la red.</p>	<p>Al igual que Pentaho posee una comunidad de soporte en varios idiomas.</p>	<p>Birt posee una comunidad activa por cada producto.</p>	<p>Palo si posee una comunidad pero no es muy activa.</p>
<p>Documentación</p>	<p>Por ser un proyecto open source muy grande la mayor parte documentada pero hay partes desactualizadas, posee también una documentación en línea.</p>	<p>Jasper posee documentación actualizada en línea y en la instalación se provee los manuales</p>	<p>Bird tiene su documentación pero hay partes que se debe pagar el contrato de soporte.</p>	<p>Tiene una muy buena documentación.</p>

⁵⁵ <http://community.pentaho.com/>

⁵⁶ <http://jasperforge.org/projects/>

⁵⁷ <http://www.birt-exchange.com/be/downloads/>

⁵⁸ <http://www.jedox.com/en/jedox-downloads/>

Como conclusión de esta comparación se puede decir que todas las herramientas poseen un grado alto de instalación a excepción de Palo que su instalación no es tan complicada, pero si se necesita conocimientos técnicos para su instalación.

En cuanto a las versiones de los productos se puede decir que tanto Pentaho y Jasper son empresas que lanzan versiones más a menudo, por lo que las versiones anteriores son remplazadas continuamente.

Palo es el software que menos diferencia tiene entre su versión community con la Enterprise, pero hay que notar que los componentes OLAP y Report Manager son indispensables en una solución de Bussines Intelligence, en cambio Pentaho provee la mayor cantidad de herramientas pero sus reportes en la versión community son básicos, se pueden crear reportes más avanzados pero se requiere mayores conocimientos técnicos en varias tecnologías.

La documentación es una parte esencial, la mayoría de proyectos tiene una documentación tanto integrado al sistema como en internet, pero el mayor problema es que muchas veces lanzan nuevas versiones del producto pero la documentación no la actualizan, por eso es necesario conocer las comunidades en internet que se dedican a compartir sus experiencias con el manejo del software, es el caso de los foros, en los que Pentaho, Birt y Jasper tienen una comunidad activa de usuarios que están pendientes en ayudar.

La tabla 3.2 contiene una comparación de características de cada herramienta de las diferentes suites de Bussines Intelligence.

Tabla 3.2: Comparativa de características de diferentes suites de BI

Característica	Pentaho	Jasper	Birt	Palo
OLAP	Su motor es Pentaho Análisis que es basado en Mondrian, fácil de usar, genera esquemas XML que luego serán interpretados por Pentaho BI Server	Al igual que Pentaho, su motor OLAP es Mondrian, pero Jasper ha realizado mejoras en cuanto los reportes rápidos (pívor) desde el editor OLAP.	Posee un motor OLAP interno que permite generar reportes de tablas cruzadas.	Tiene un motor MOLAP Interno que a diferencia de las soluciones BI carga todos los datos y pre-calcula todas las intersecciones por lo que no se fundamenta en ninguna base de datos sino que crea sus propios cubos.
Reportes	Su editor es Report Designer, para agregar programación se usa Design Studio y se almacena los reportes en un repositorio de Pentaho, se pueden crear reportes en base al esquema OLAP de Mondrian o a un archivo de metadatos realizado en Pentaho Metadata Editor	Su editor es JasperReports y son ejecutados mediante IReports, es el reporteador más potente en esta comparación, permite la conexión a todas las bases de datos además de archivos planos, XML, Hibernate y otros.	Eclipse BIRD es un plug in para Eclipse que genera de reportes, su ventaja está en que se puede inyectar valores en una tabla cruzada para optimizar el análisis de datos, al igual que Jasper contiene un editor y un componente de rutina para su ejecución e integración con otros sistemas.	PALO tiene integrado un reporteador cuya ventaja es que se puede agregar programación en cualquier punto del diseño, y se puede ocupar todas las acciones creadas en el servidor. Se puede integrar con Excel, y se pueden crear reportes estilo Excel.
Herramienta Web	Pentaho BI server provee una interface web donde se organiza los archivos del repositorio, y permite la creación de reportes	Jasper Server es una aplicación java permite organizar el repositorio de reportes y la creación de	Su herramienta web es IServer que es una herramienta disponible solo para la versión Enterprise es	Con PALO Web se obtiene el acceso a toda la plataforma BI, es decir que desde aquí se pueden crear cubos reportes, usuarios, roles.

	rápidos.	reportes ad-hoc o reportes rápidos, mantiene la información actualizada.	muy técnico y más limitado que Pentaho y Jasper en reportes ad-hoc.	
ETL	Pentaho Data Integration basado en Kettle provee el ETL para Pentaho, es una herramienta de programación visual, intuitiva y fácil de usar, trabaja con transformaciones y trabajos (Jobs).	JasperETL es una herramienta muy técnica pero muy versátil, genera código java o perl los mismos que pueden ser compilados.	No posee herramienta ETL	A diferencia de Pentaho y Jasper PALO ETL gestiona los trabajos en proyectos y cada proyecto tiene sus trabajos, transformaciones, extracción y carga de datos, es muy organizado y hay que seguir su estándar de desarrollo, no es muy técnico y tiene varias tareas preprogramadas.

Como se puede observar en la tabla 3.2, cada Suite de BI propone diferentes soluciones, cada Suite tiene sus propias herramientas tanto de servidor OLAP, ETL, reportadores, Herramientas web, pero el valor agregado que pone cada marca es lo que diferencia y lo que a la final se valor al momento de elegir.

Según la matriz anterior se puede revisar que BIRD es la suite con menor número de componentes, es decir; es el más limitado, pero provee un poderoso generador de reportes que se pueden integrar a otros sistemas lo que lo hace muy útil al momento de desarrollar una aplicación Java, en cambio Pentaho provee una mayor cantidad de aplicativos gratuitos pero su reporteador no es el mejor de entre los tres, Jasper por su lado provee las herramientas necesarias, pero posee bastantes limites en cuanto a las funciones de Bussines Intelligence

porque la mayoría de estas funciones están disponibles a partir de su versión Enterprise, en cuanto a Palo la ventaja mayor es su facilidad de uso, tener todo integrado en un solo programa, y su motor MOLAP lo que le hace más rápido y mejor para el análisis de datos, además de la conexión con Excel, CALC de open office lo hace la solución con mayor número de herramientas.´

En cuanto a herramientas ETL se puede ver que existe una gran diferencia entre cada Suite, Pentaho provee una programación visual, es un sistema estable, permite realizar Jobs (trabajos), transformaciones tiene una gran paleta de componentes para utilizar y se basa en la generación de scripts SQL para la ejecución de las tareas no así Jasper que genera código java o perl para luego compilarlos y ejecutarlos, pero en cambio Palo no provee una programación visual de drag and drop, al contrario cada paso tiene un formulario para programarlo, se debe almacenar todo según la estructura que provee Palo ETL que es guardar todos los trabajos en proyectos, guarda una lógica de creación de extracciones, cargas, transformaciones, así se organiza el proyecto y queda totalmente documentado. En si Palo provee una herramienta muy estable pero es muy lenta en el procesamiento no así Pentaho data Integration que demuestra ser el más ligero y de rápido procesamiento.

Cada suite trata de mejor manera sobresalir en la generación de reportes, en este aspecto Jasper y Palo son los mejores, Jasper tiene un motor potente iReports que visualmente es muy agradable, Palo a su vez la posibilidad de crear reportes en Excel lo hace más versátil, pero Pentaho y Birt no se quedan atrás,

Pentaho a pesar de no tener un fuerte motor de reportes, no quiere decir que sus reportes no sirvan, son muy útiles y es una solución muy económica que provee lo básico para desarrollar un sistema de BI, en cuanto a Birt tiene un buen motor de reportes pero sus funciones internas son muy limitadas en cuanto a Business Intelligence.

Los motores OLAP son muy parecidos, partiendo de que Pentaho y Jasper utilizan el mismo motor Mondrian, un poderoso motor OLAP que exporta esquemas en archivos XML, provee un motor Pívor, y lenguaje MDX para la exploración de datos, Birt tiene un motor interno que permite crear reportes de tablas cruzadas, y la gran diferencia la pone Palo que tiene un motor MOLAP que según se ha hablado antes es mejor para el análisis de datos.

Tabla 3.3 Características importantes de Suites BI en versiones Community Open

Source.

Característica	Pentaho	Jasper	Birt	Palo
Reportes Ad-hoc	Posee en la versión community pero son muy básicos.	No	No	La mayoría de reportes son prediseñados, no posee un reporteador ad-hoc
Visor OLAP	Posee un visor OLAP.	Tiene un visor OLAP básico.	No	Si posee un visor OLAP
Cuadros de mando	Posee cuadros de mando básicos en la versión community con gráficas limitadas y su desarrollo es complicado.	Tienen un diseñador de paneles que reemplaza la necesidad de cuadros de mando.	Posee pero son muy básicos,	No tiene un apartado especial para esta opción, pero se pueden desarrollar cuadros de mando.
Cuadros de mando ad-hoc	No posee en la versión community	No	N	No
Interconexión con Excel	No	No	No	Si
Reportes Spreadsheet	No	No	Si	Si

3.5.- Resultados.

Después de haber analizado cada producto se ha llegado a la conclusión que la suite de Bussines Intelligence que mejor se adaptaría al proyecto es Pentaho, porque provee la mayor cantidad de herramientas gratuitas que cubriría las necesidades para el proyecto en curso por tener un motor OLAP, Reporteador, ETL, Herramienta Web, editor de metadatos además de funciones extras como reportes ad-hoc y la posibilidad de crear cuadros de mando, también por poseer una comunidad activa de usuarios en idioma Inglés y Español.

Pentaho también es un proyecto que continuamente se sigue desarrollando lo que es una ventaja porque se siguen obteniendo actualizaciones para defectos en el producto y nuevas funcionalidades que se podrían utilizar.

CAPÍTULO 4

DESARROLLO DEL SISTEMA

4.1.- Introducción.

En este capítulo se explicará la construcción de la solución propuesta para ELEPCO, está documentado mediante las mejores prácticas que propone SCRUM para el desarrollo de software, se ha dividido el desarrollo tres partes, en cada una se desarrolla un sub sistema, está dividido así porque cada uno es independiente y se lo desarrolla en tecnologías diferentes.

La recolección de requerimientos se lo realizó previamente a y están documentados según la norma IEEE830 y esta adjuntado al ANEXO A, este documento contiene los requerimientos por cada sistema. En el desarrollo de cada sistema se tomarán los requerimientos funcionales y no funcionales como historias de usuario y se armará la pila del producto.

4.2.- Desarrollo del Subsistema de obtención de Datos

4.2.1.-Visión general

Este módulo va a constar como parte de un sistema ya desarrollado por ELEPCO; en el sistema existente ya están desarrolladas las funciones de lectura de los equipos de campo: GPU's y sensores de nivel de agua. El objetivo de este subsistema es almacenar los datos obtenidos en una base de datos relacional para ser utilizados en las siguientes etapas del desarrollo.

4.2.2.- Análisis de requerimientos.

Del documento ANEXO A donde está la especificación de requerimientos del sistema, se extraen los requerimientos funcionales y no funcionales para este subsistema.

4.2.3.- Requerimientos Funcionales.

- Capturar datos del sistema principal
- Acondicionar los valores.
- Calcular potencia generada en Watts.
- Almacenar los datos recolectados en una base de datos.

4.2.4.- Requerimientos no funcionales

- Desarrollo del sistema en LabVIEW
- Trabaja en sistema operativo Windows.

4.2.5.- Análisis

En este apartado se analizará el sistema a desarrollar, se requiere realizar una adaptación a un sistema ya construido, una función de almacenamiento en una base de datos, para lo que se realizará un programa que recibirá una matriz de datos con los valores de lectura y fecha de lectura de los equipos y el programa se encargará de almacenarlos. Aprovechando la ventaja de extensibilidad de LabVIEW se puede realizar un desarrollo totalmente separado del programa principal evitando así interferir con el funcionamiento principal del sistema existente.

Este programa será un servicio silencioso del sistema existente en ELEPCO, el usuario no podrá realizar ninguna actividad sobre este, es decir; no existe un caso de uso sino que será una extensión a un caso de uso ya existente.

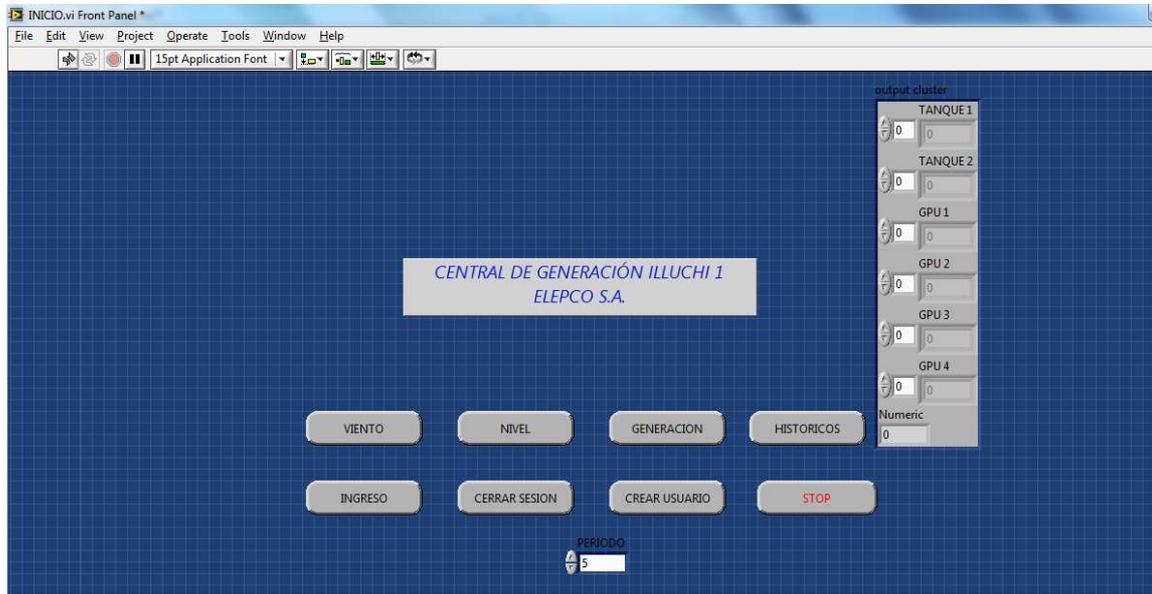


Imagen 4.1 Pantalla principal del sistema principal.

Como se puede apreciar en la imagen 4.1 en el sistema se puede tener acceso a las siguientes opciones:

- Viento.
- Nivel de agua.
- Generación eléctrica.
- Datos históricos.
- Ingreso (función de inicio de sesión)
- Cerrar sesión
- Crear Usuario.
- Stop (Paro del sistema).
- Periodo (período en minutos para grabar en archivos planos)

LabVIEW es un lenguaje de desarrollo visual, no usa el paradigma de programación orientada a objetos, es decir que no se pueden crear diagramas UML de los programas desarrollados en este, por eso la forma de documentación será mediante diagramas de flujo.

En la tabla 4.1 se muestra la pila del producto a desarrollar con sus actividades, importancia, la estimación en puntos (cada punto significa una hora hombre), tomando una buena práctica de la metodología de desarrollo XP, se toma cada requerimiento como una historia de usuario.

Tabla 4.1: Pila del producto para el desarrollo del subsistema de Obtención de Datos

Historia de usuario.	Categoría (Módulo)	Importancia	Estimación Inicial (puntos)
Capturar datos del sistema principal.	Capturador de datos	10	8
Acondicionar valores	Capturador de datos	10	8
Calcular potencia generada.	Capturador de datos	20	8
Almacenar en base de datos	Capturador de datos	30	24

4.2.6.- Duración del sprint

SCRUM recomienda que la duración de cada sprint sea de máximo 30 días dependiendo las actividades a realizarse, para la estimación del número de horas de este proyecto se tomó en cuenta la dificultad de cada actividad, por lo que para culminar el proyecto se necesitará de dos sprints de 10 días cada uno.

4.2.7.- Primera iteración.

Tabla 4.2 Pila del primer sprint, subsistema Obtención de Datos

Historia de usuario.	Categoría (Módulo)	Importancia	Estimación Inicial (puntos)
Capturar datos del sistema principal.	Capturador de datos	10	8
Acondicionar valores	Capturador de datos	10	8
Calcular potencia generada.	Capturador de datos	20	8

4.2.8.- Exploración

En esta sección se detalla el desarrollo en sí, describiendo la tecnología utilizada y la arquitectura implementada.

4.2.8.1.- Capturar datos del sistema principal.

Esta historia de usuario consiste en armar un subprograma VI en LabVIEW, va a ser el programa principal del subsistema, este recibirá los datos del sistema existente en ELEPCO, y entregará los datos a los siguientes subprogramas que se creen, en el desarrollo de los siguientes subprogramas se seguirá incrementando código en este.

Recibirá los datos del sistema existente en un arreglo de datos conteniendo la información de cada generador y tanque, además recibirá un parámetro tipo boolean (verdadero o falso) para activar o desactivar el subsistema.

En LabVIEW todo control que se ocupe debe ser representado en el formulario, como se puede ver en la Imagen 4.2 todos los parámetros de entrada se representan gráficamente.

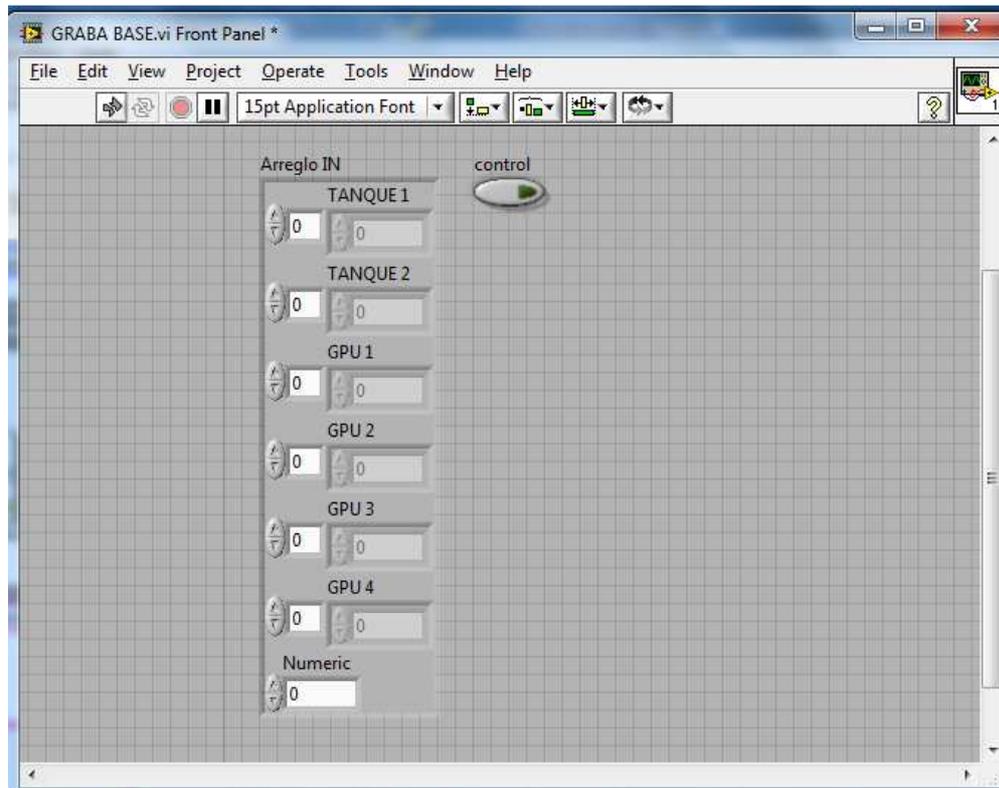


Imagen 4.2 Parámetros de entrada.

El arreglo que llega al sub sistema está integrado por seis campos los mismos que tienen la información de los niveles de agua y los datos de generación, los datos de generación a su vez son un arreglo de seis posiciones correspondientes a los seis valores de generación de corrientes y voltajes por cada fase que se obtienen de los medidores, el arreglo tiene la estructura que se muestra en el gráfico 4.1.

VALOR NIVEL DE AGUA TANQUE 1					
VALOR NIVEL DE AGUA TANQUE 2					
VALORES DE GENERACION GPU1					
1	2	3	1	2	3
VALORES DE GENERACION GPU2					
1	2	3	1	2	3
VALORES DE GENERACION GPU3					
1	2	3	1	2	3
VALORES DE GENERACION GPU4					
1	2	3	1	2	3

Gráfico 4.1: Estructura del arreglo de datos entregado por el sistema principal.

El código de este subsistema es sencillo solo debe recibir el arreglo del sistema principal y luego mediante el operador Unbundle by Name se separa el arreglo en diferentes variables.

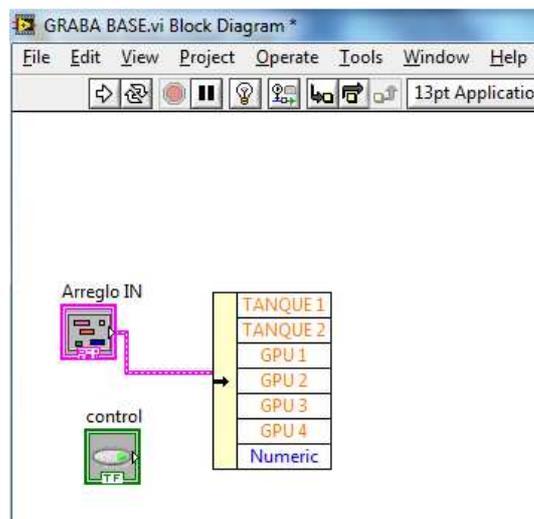


Imagen 4.3 Código desarrollado para el SubVI "GRABA BASE.vi".

Una vez incluido en el programa las variables se especifica que estas van a ser los parámetros de entrada, y se lo integra al programa principal quedando la programación del sistema principal de la siguiente forma.

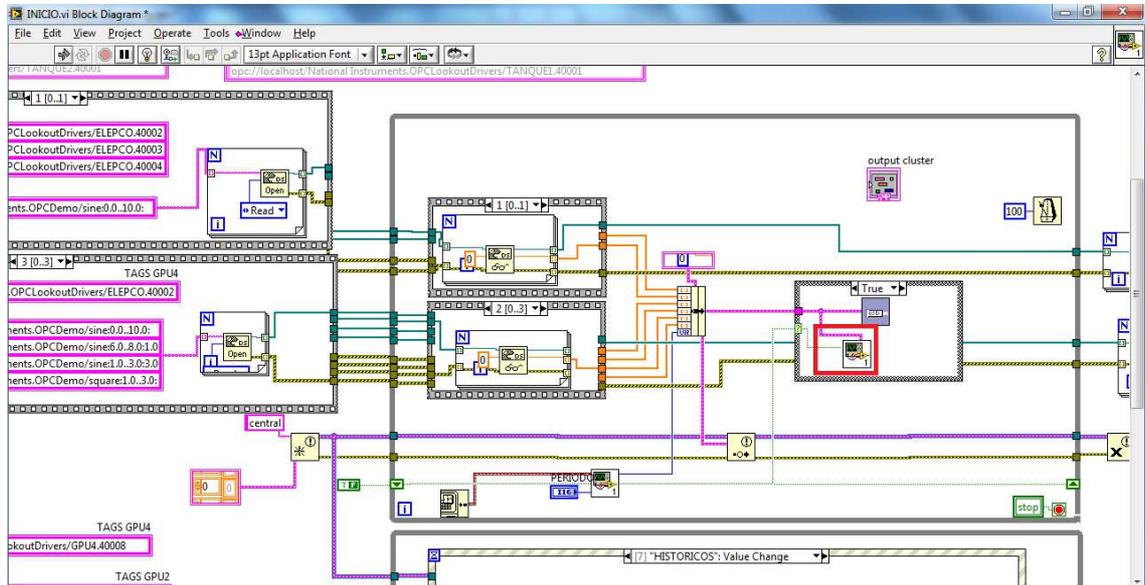


Imagen 4.4: Programación del sistema principal.

Como se ve en la imagen 4.4, la adaptación del subprograma al sistema existente es muy sencilla puesto que solo consiste en incluir el subprograma y entregar los parámetros de entrada, en la imagen el recuadro rojo marca la posición de la llamada al subprograma.

Los requerimientos de acondicionar valores y calcular potencia de generación se los va a realizar en un solo SubVI para que este entregue un solo arreglo con los valores de generación (corrientes, voltajes y potencias).

4.2.8.2.- Acondicionar valores y calcular potencias generadas.

Este subprograma recibirá como parámetro un arreglo de datos de lectura de los GPU's de ocho posiciones correspondientes a tres variables de corrientes, tres variables de voltajes más los valores de nivel de agua de los dos tanques. Los GPU's devuelven las lecturas en valores entre 0 y 4096 (2^{12}) para cada variable, estos valores son programados para que representen una escala y mediante una ecuación lineal se obtiene el valor correspondiente.

Acondicionamiento del valor de corrientes.

Los GPU's 3 y 4 tienen una mayor capacidad y generan hasta 800 Amperios, mientras que los GPU's 1 y 2 solo 400 amperios, cabe destacar que la corriente puede tomar valores negativos, si esto sucediera entonces quisiera decir que el generador en vez de entregar energía está consumiendo energía lo cual nunca va a suceder:

La escala de valores está entre -800 Amperios y 800 Amperios en el caso de los GPU's 3 y 4, para los GPU's 1 y 2 está entre -400 y 400

La siguiente fórmula se usa para calcular el valor de las corrientes para los GPU's 3 y 4

$$i = (0.09768 * x - 199.98372) * 4$$

Como los GPU's 1 y 2 tienen la mitad de la capacidad de los GPU's la fórmula es la misma dividida para dos.

$$i = (0.09768 * x - 199.98372) * 2$$

Donde x representa el valor de la lectura, la unidad devuelta es amperios.

Acondicionamiento del valor de los voltajes.

Para los voltajes es muy similar, se tiene la misma escalada de 0 a 4096 pero sin valores negativos, el valor va de 0V a 13800V, por lo que la ecuación sería la siguiente:

$$V = 3.369x$$

Donde x representa el valor de la lectura, la unidad devuelta son voltios:

$$(((1.17216*x)-(2400))*(23/24))*150$$

Calculo de las potencias.

Una vez obtenidos los valores de voltajes de línea se obtienen los valores de potencia con la siguiente fórmula.

$$P = V/\sqrt{3}$$

Por cada valor de voltaje recibido se calcula una potencia, en este caso es la potencia de línea, para calcular la potencia trifásica o total generado se usa la siguiente fórmula.

$$Potencia Trifásica = \frac{\frac{C1 + C2 + C3}{3} * \frac{V1 + V2 + V3}{3} * 1.73}{1000}$$

La imagen 4.5 muestra el funcionamiento de este subprograma.

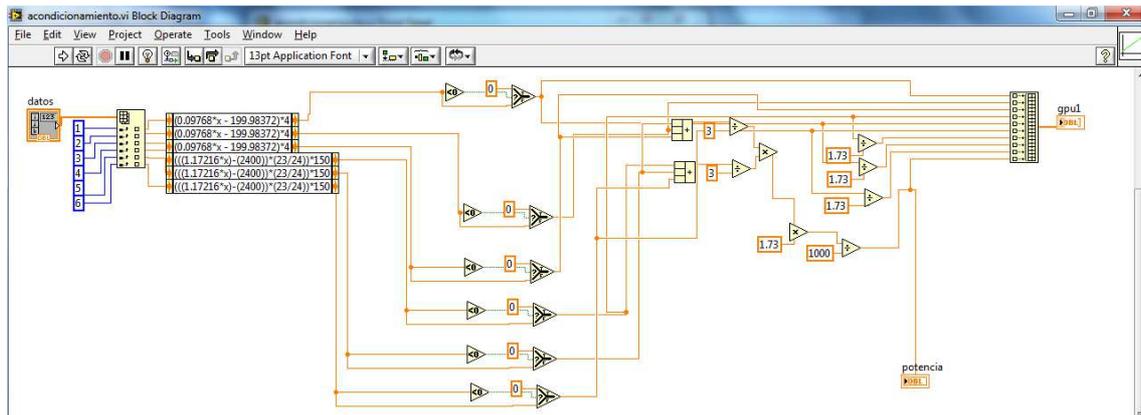


Imagen 4.5: Diagrama de bloque de cálculo de potencias.

Acondicionamiento de niveles de agua.

De la misma forma se crea un nuevo sub VI para acondicionar los valores de los tanques pero en este caso se utilizó la siguiente fórmula de cálculo:

$$\text{Nivel de Agua} = -5 + \left(\frac{X}{819}\right)$$

Una vez terminado de crear los dos sub VI, se los agrega al programa “Graba Base.vi” tal como lo muestra la imagen 4.6.

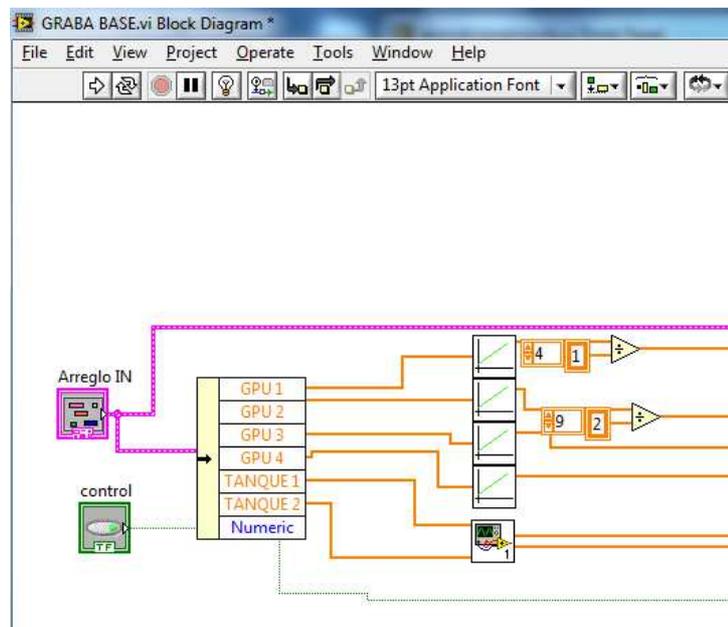


Imagen 4.6 Programación subVI “Graba Base”

4.2.9.- Carta de Burndown.

Como se explicó en la metodología Scrum en el capítulo 2, la carta de Burndown muestra la cantidad de trabajo restante a través del tiempo; es una manera de visualizar la relación entre la cantidad de trabajo restante en cualquier punto. Para elaborar la carta de Burndown se emplea la fecha en el que se finaliza cada elemento de la pila del sprint, y los puntos restantes para terminar el proyecto. En el presente proyecto, no se emplean fechas para indicar la finalización de cada elemento de la pila, sino los días transcurridos desde el inicio del proyecto.

Tabla 4.3: Información para la carta de Burndown al finalizar el primer Sprint

Historia de usuario.	Categoría (Módulo)	Estimación Inicial (puntos)	Finalización	Puntos restantes
Capturar datos del sistema principal.	Capturador de datos	8	Día 3	40
Acondicionar valores	Capturador de datos	8	Día 7	32
Calcular potencia generada.	Capturador de datos	8	Día 10	24

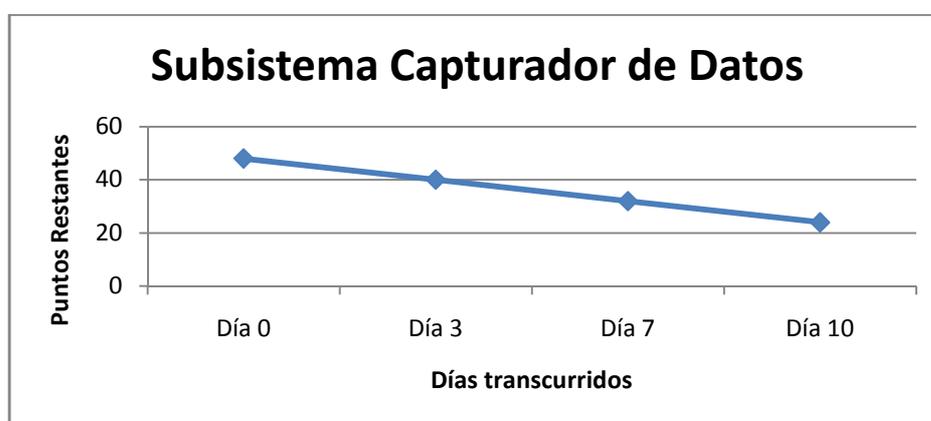


Gráfico 4.2: Avance del Proyecto, subsistema Capturador de Datos.

Como se ve en el gráfico 4.2 se ha avanzado la mitad de la estimación inicial del proyecto en los primeros diez días de desarrollo, esto se debe a que es un subsistema pequeño con pocas funciones que no presentan mayor dificultad.

4.2.10.- Revisión

Una vez terminado el desarrollo de las actividades planificadas en este sprint se presenta el avance al dueño del producto. El objetivo es presentar la funcionalidad del sistema, y reunir ideas y sugerencias de lo que debería hacerse a continuación.

El dueño del producto no presenta novedades de cambios para el avance actual por lo que se continúa con el siguiente sprint.

El gráfico 4.3 diagrama muestra como esta implementado el sistema hasta el punto actual.

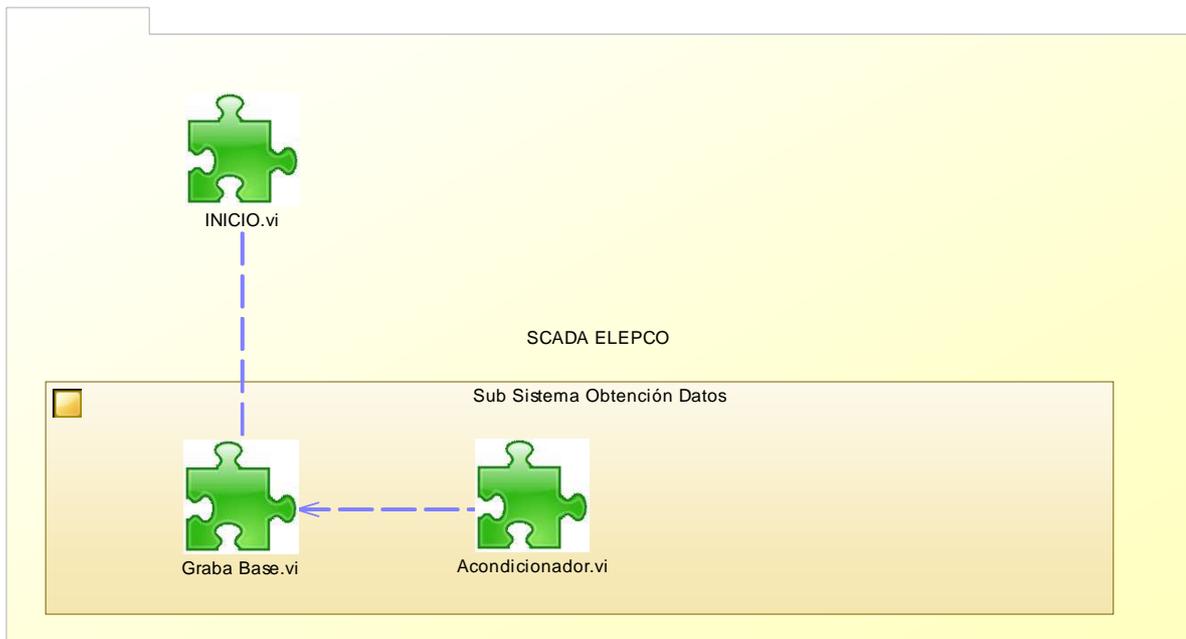


Gráfico 4.3 Avances en la implementación del Sistema.

4.2.11.- Segunda iteración.

El objetivo de este sprint es desarrollar la función de almacenamiento de los datos recolectados y acondicionados a una base de datos relacional en MySQL.

Tabla 4.4 Pila del producto después de la primera iteración.

Historia de usuario.	Categoría (Módulo)	Importancia	Estimación Inicial (puntos)	Completado
Capturar datos del sistema principal.	Capturador de datos	10	8	Si
Acondicionar valores	Capturador de datos	10	8	Si
Calcular potencia generada.	Capturador de datos	20	8	Si
Almacenar en base de datos	Capturador de datos	30	24	No

Como se puede apreciar solo resta desarrollar la función de almacenamiento en la base de datos, esta función se la consideró la más complicada de todas porque no existe una función genérica integrada en LabVIEW sino que se debe incluir unas librerías de conexión a bases de datos esta librería se llama LabSQL.

Tabla 4.5 Pila del segundo sprint, subsistema Obtención de Datos.

Historia de usuario.	Categoría (Módulo)	Importancia	Estimación Inicial (puntos)	Completado
Almacenar en base de datos	Capturador de datos	30	24	No

4.2.12.- Almacenar en base de datos

Para realizar esta historia de usuario se van a realizar las siguientes actividades.

- Creación de la base de datos.
- Desarrollo de la función de almacenar en la base de datos.

4.2.12.1.- Creación de la base de datos.

La base de datos a crear va a ser compartida por dos subsistemas del proyecto, por el Capturador de Datos y el Monitor, por lo que se debe considerar que el diseño debe permitir el funcionamiento de los dos módulos sin que ellos se interfieran, el nombre de esta base de datos será “central”.

Para la creación de las primeras tablas de la base de datos se debe tener en cuenta primero que el subsistema capturador ingresara la información en estas tablas, luego el módulo monitor leerá la información para la generación de reportes y luego se migrarán a otra base de datos para generar un cubo de información.

Como se vio en el sprint anterior son nueve valores de generación a almacenar entre corrientes, voltajes y potencias. Los valores de corrientes y voltajes tienen un identificador en la memoria de los GPU's según su posición de memoria este valor es un número que comienza a partir del 40001 hasta el 40006 los mismos que se tomarán como identificadores para las tablas de generadores. Para los valores de potencias se van a grabar como que si vinieran de las posiciones del 40007 al 40009, además se necesita ingresar en una tabla aparte la información de los GPU's que se van a leer.

De la misma manera que los GPU's se necesita almacenar la información obtenida desde los tanques de agua, en este caso se obtiene un solo valor por tanque, este valor es el número en metros desde el borde superior del tanque hasta la superficie del agua, por lo que la tabla quedaría de la siguiente manera:

Cada lectura que realiza el Sistema actual genera un registro por cada equipo (GPU o Tanque) y por cada posición de memoria a eso se le agregará la fecha de la lectura y un identificador de registro, el gráfico 4.4 muestra el diagrama de tablas para esta parte del desarrollo.

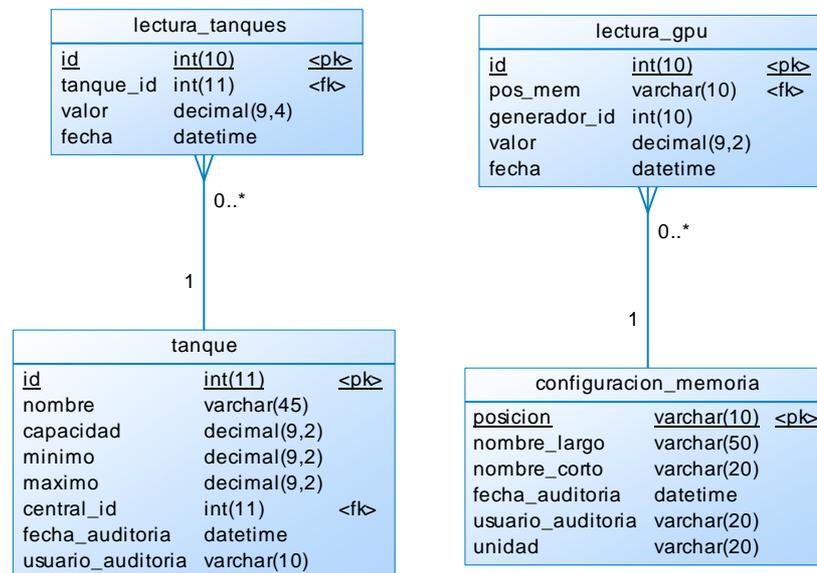


Gráfico 4.4 Diagrama de base de datos "central".

4.2.12.2.- Desarrollo de la función Base de datos

Esta función es un programa SubVi que va a ser llamado desde el programa principal (Graba Base Datos.vi), va a recibir los datos de generación ya acondicionados y los datos de niveles de agua en un arreglo de datos.

El motor de base de datos a utilizar será MySQL y la forma conexión entre la base de datos con el sistema a desarrollar será mediante ODBC para lo cual se instaló los siguientes programas.

- MySQL server 5.1
- MySQL connector ODBC 5.1.6
- MySQL GUI tools.

MySQL tiene una arquitectura Cliente – Servidor.

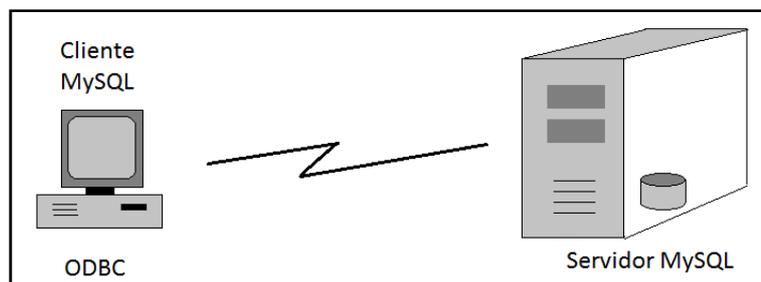


Gráfico 4.5 Arquitectura de la conexión a la base de datos.

Mediante la librería LabSQL para LabVIEW se crea un subVI que lo que realizará será tomar como parámetro de entrada una sentencia SQL y ejecutarla en la base de datos este sub programa se lo nombrará como “grabar en mysql.vi”.

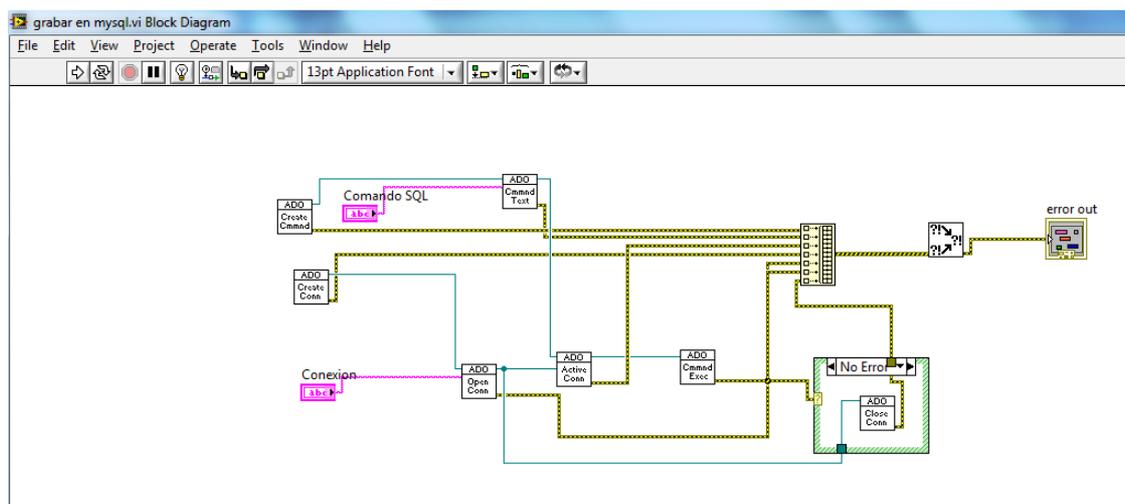


Imagen 4.7 Diagrama de bloques de “grabar en mysql.vi”

Se creó otro programa subVI que recibe los valores de generación y arma las sentencias Insert se lo llamará “Generador de comandos SQL.vi”.

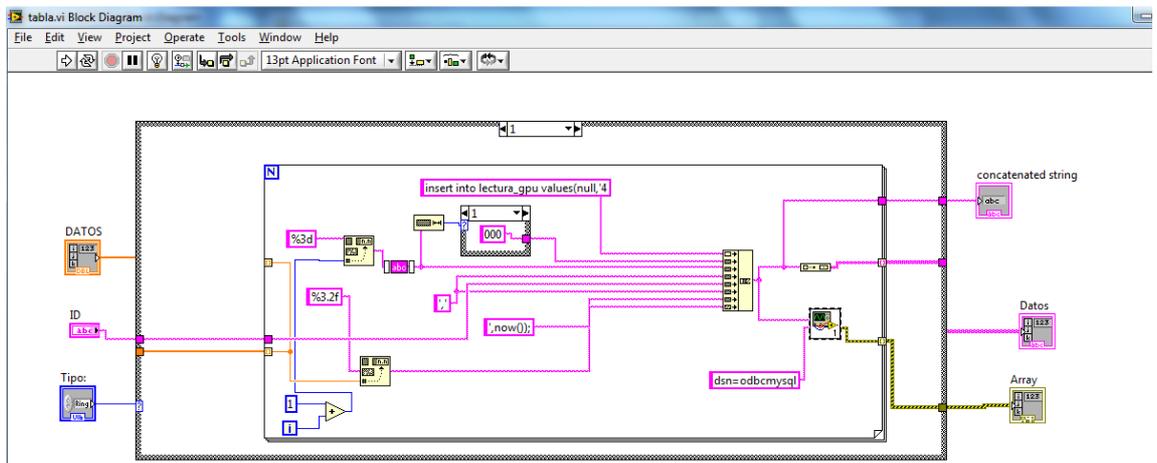


Imagen 4.8: Diagrama de bloques de “Generador de comandos SQL.vi”.

Por último se creó un SubVI adicional que tome las matrices de datos de generación y niveles de agua y envía matriz por matriz a la función anterior y su nombre será “Base de datos.vi”.

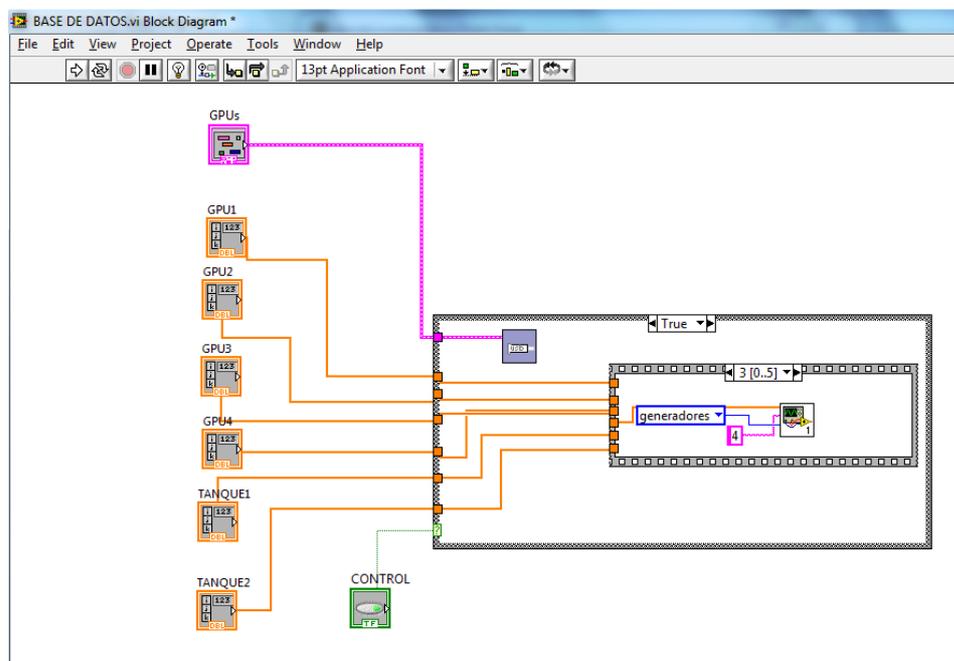


Imagen 4.9 Programación del SubVI “Base de datos.vi”.

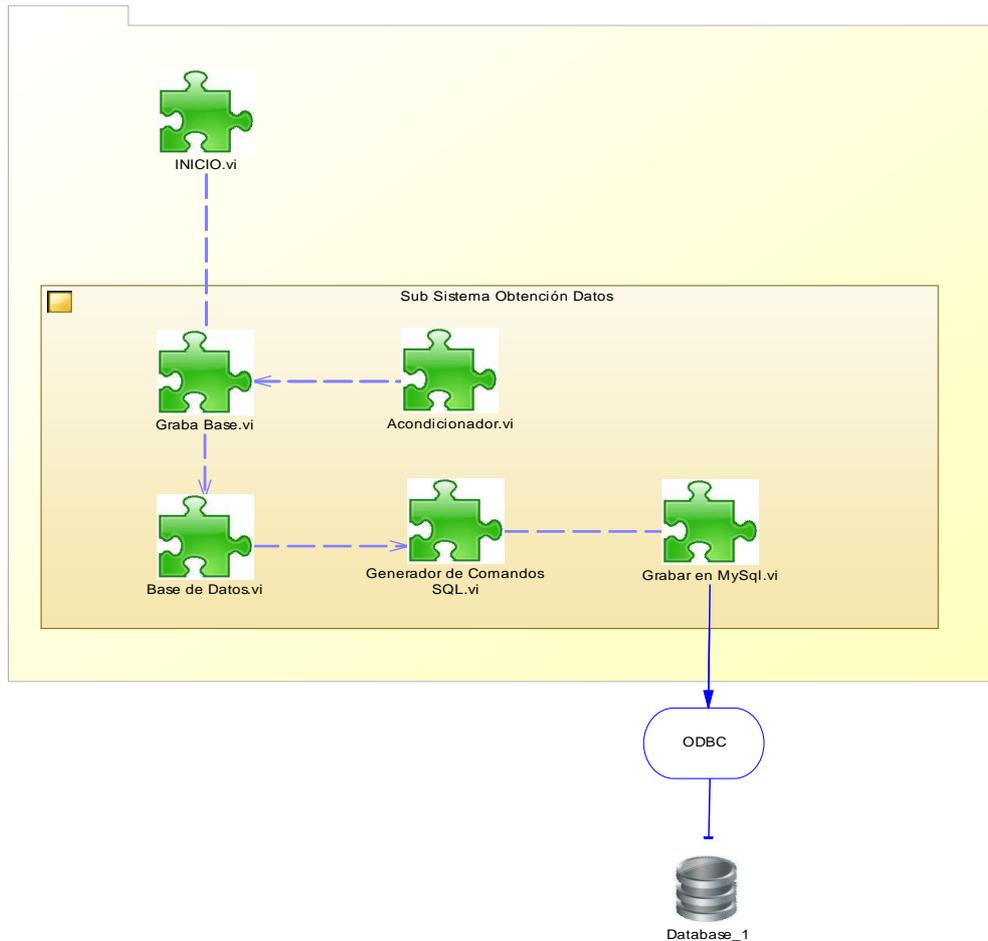


Gráfico 4.6 Estructura de implementación del subsistema monitor

4.2.13.- Carta de Burndown.

Una vez terminado el segundo sprint se puede decir que el desarrollo ha finalizado para ello se completa la pila del sprint actualizando el día de finalización y los puntos restantes y se genera el gráfico del Burndown.

Tabla 4.6 Pila del producto después de la segunda iteración.

Historia de usuario.	Categoría (Módulo)	Estimación Inicial (puntos)	Finalización	Puntos restantes
Almacenar en base de datos.	Capturador de datos	24	Día 24	0

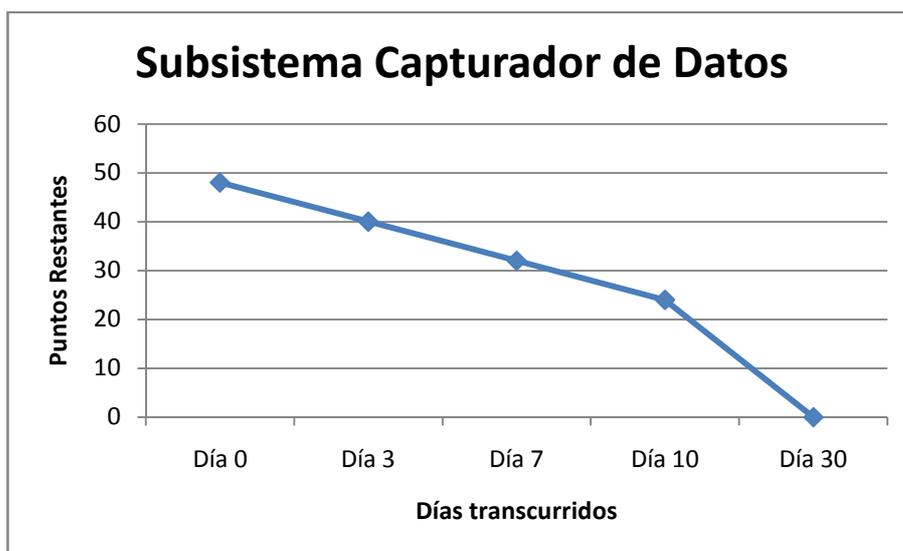


Gráfico 4.7: Avance del Proyecto

4.2.14.- Revisión

Una vez concluido el desarrollo del subsistema se lo presenta al dueño del producto, el mismo que no presenta inconformidad por los resultados del sistema, por lo que se da por finalizado el desarrollo del subsistema de obtención de datos y se sigue con el desarrollo del subsistema Monitor.

4.3.- Desarrollo del Subsistema Monitor

4.3.1.-Visión general

El subsistema Monitor será una interface web y su función principal será mostrar los valores actuales de los generadores y de los tanques para que el usuario pueda monitorear los equipos de campo desde un computador local o remoto a la central hidroeléctrica, este subsistema no tendrá funciones de control

de procesos como encendido y apagado de equipos, apertura o cierre de compuertas, etc.

Entre las alarmas que se monitorearán están: encendido de equipos, apagado de equipos, nivel de agua, otras alarmas se encuentran restringidas por los GPU's.

4.3.2.- Análisis de requerimientos.

Del ANEXO A donde se encuentran detallados los requerimientos para este subsistema se toman los requerimientos funcionales y no funcionales para su análisis.

4.3.2.1.- Requerimientos Funcionales.

- Administración de usuarios y grupos de usuarios.
- Administración de estructura geográfica (Región, Provincia, Ciudad, Central de Generación) de los equipos.
- Administración de Generadores (GPU's).
- Administración de tanques.
- Administración de permisos de grupos de usuarios a menús y opciones.
- Administración de menús y opciones.
- Administración de la Configuración de Memoria.
- Alarmas de aviso de eventos importantes.
- Generar gráficos dinámicos actualizables durante el transcurso del tiempo del estado de generación eléctrica por grupo de generación.
- Generar gráficos históricos de generación eléctrica con filtros.

- Generar gráficos de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.
- Generar un gráfico de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.
- Administración de la planificación de mantenimiento de generadores.

4.3.2.2.- Requerimientos no funcionales

- Interface web.
- Los colores principales a aplicar en el diseño deben ser celeste y negro con escalas de grises.
- El estado de los generadores (Encendido o Apagado) deben estar en la página principal y se deben mostrar en una sección de fácil visualización.
- Los gráficos de las variables a monitorear deben poder mostrarse en ventanas independientes, para que el usuario pueda seguir revisando otras secciones del sistema.
- Tener información de auditoría en las tablas con el nombre de usuario quien realizo el último cambio y la fecha del cambio.

4.3.3.- Análisis

Para la construcción de este sistema se partirá con el desarrollo de la estructura de base de datos, lo que se hará es diseñar y desarrollar un sistema basado en tecnología web usando el lenguaje de programación PHP que permita monitorear el estado de generación de los equipos de campo de la central hidroeléctrica.

Tabla 4.7 Pila de producto para el desarrollo del subsistema Monitor.

No.	Historia de usuario	Categoría	Importancia	Estimación inicial en puntos.
1	Administración de usuarios y grupos de usuarios.	Monitor	5	6
2	Administración de estructura geográfica (Región, Provincia, Ciudad, Central) de los equipos.	Monitor	5	13
3	Administración de Generadores (GPU's).	Monitor	5	6
4	Administración de tanques.	Monitor	5	4
5	Administración de permisos de grupos de usuarios a menús y opciones.	Monitor	8	10
6	Administración de menús y opciones.	Monitor	6	8
7	Administración de la Configuración de Memoria.	Monitor	6	4
8	Alarmas de aviso de eventos importantes.	Monitor	4	6
9	Generar gráficos dinámicos actualizables durante el transcurso del tiempo del estado de generación eléctrica por grupo de generación.	Monitor	6	24
10	Generar gráficos históricos de generación eléctrica con filtros.	Monitor	6	48
11	Generar gráficos de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.	Monitor	5	20
12	Generar gráficos dinámicos de niveles de agua por tanque actualizables durante el transcurso del tiempo.	Monitor	6	30
13	Administración de la planificación de mantenimiento de generadores.	Monitor	8	6
				185

4.3.4.- Duración del Sprint

Como se ha visto anteriormente Scrum recomienda Sprints de 15 a 30 días, para establecer la duración de cada sprint en el presente proyecto se toma como base el tiempo necesario para implementar y probar el primer sprint. Las historias de usuario que incluye el primer sprint se encuentran en el apartado correspondiente a este. Se estima que 30 días son suficientes para implementar el primer sprint, por lo que se establece en 30 días la duración de cada sprint para el presente subsistema.

4.3.5.- Primer Sprint.

4.3.5.1.- Planificación.

El objetivo planteado para el primer sprint es definir la arquitectura del prototipo. Tomando una buena práctica de XP se seleccionan las historias de usuario de mayor importancia para el desarrollo de la arquitectura; en el caso de Scrum las historias de usuario son los elementos de la pila del producto.

Tabla 4.8 Pila del primer sprint, subsistema Monitor.

No	Historia de usuario	Categoría	Importancia	Estimación inicial en puntos.
6	Administración de menús y opciones.	Monitor	6	8
1	Administración de usuarios y grupos de usuarios.	Monitor	5	6
5	Administración de permisos de grupos de usuarios a menús y opciones.	Monitor	8	10
2	Administración de estructura geográfica (Región, Provincia, Ciudad, Central) de los equipos.	Monitor	7	13

4.3.5.2.- Arquitectura

El apartado de Arquitectura tiene por objetivo definir varios lineamientos que se siguen mientras se desarrolla el producto. Estos lineamientos son fruto de la experiencia tanto del autor del proyecto, como del director del mismo. Los lineamientos que se definen en este apartado se irán desarrollando conforme se requiera durante la evolución del proyecto.

La arquitectura a utilizar en este proyecto es Modelo Vista Controlador (MVC), como se vio en el punto 2.4 este patrón permite desarrollar aplicaciones dividiéndolas en tres partes esenciales. Para la aplicación de este patrón en PHP se va a crear una estructura de archivos que permita tal división del software que se lo detallará más adelante.

Es importante para la aplicación de este patrón la utilización de un solo archivo index.php que tendrá una función de enrutador se lo conoce como un controlador frontal (front controller), su función es proveer un punto de entrada centralizada para las peticiones de usuario.

El archivo index.php contendrá el diseño gráfico base de todo el sistema, desde aquí el usuario accederá a los menús que a su vez llamarán a un método de una clase de control el mismo que solicitará a la capa Modelo los datos necesarios para mostrar y utilizará una vista para formatear los datos y mostrar en la página index.php.

Interfaz Gráfica

Para el desarrollo de este proyecto se va a utilizar un patrón de diseño gráfico, una estructura de pantallas que tendrán las mismas características y

carguen el contenido de la sección seleccionada por el usuario, la estructura de la página principal será la siguiente:

- Título.
 - o Logo de la empresa.
 - o Estado GPU's.
 - o Datos de usuario y datos de central actuales.
- Área de menús.
- Área de Opciones.
- Área de despliegue de contenido.
- Área de Edición.

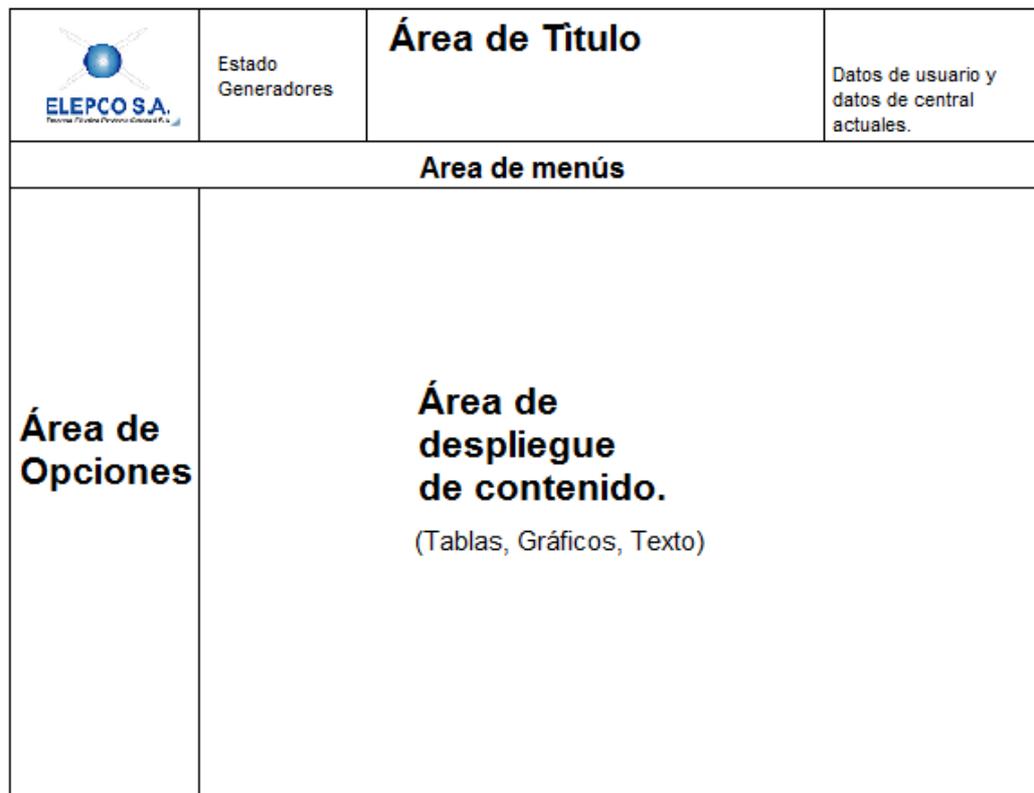


Gráfico 4.8: Diseño para la página principal.

En el gráfico 4.11 se puede ver el diseño básico de la página principal el mismo que se usará para todas las opciones para desplegar contenido, es decir; tablas, gráficos o texto. El área de edición de registros se desplegará en una ventana auxiliar en forma de pop-up, en esta ventana se mostrará solo la información a editar y los botones para las acciones:

- Guardar.
- Nuevo.
- Limpiar.
- Salir.

Para desarrollar esta ventana se usará el utilitario ThickBox⁵⁹ que es una librería para JQuery lo que hace es presentar en un IFrame que muestra el contenido de otra página en este caso los formularios de edición.

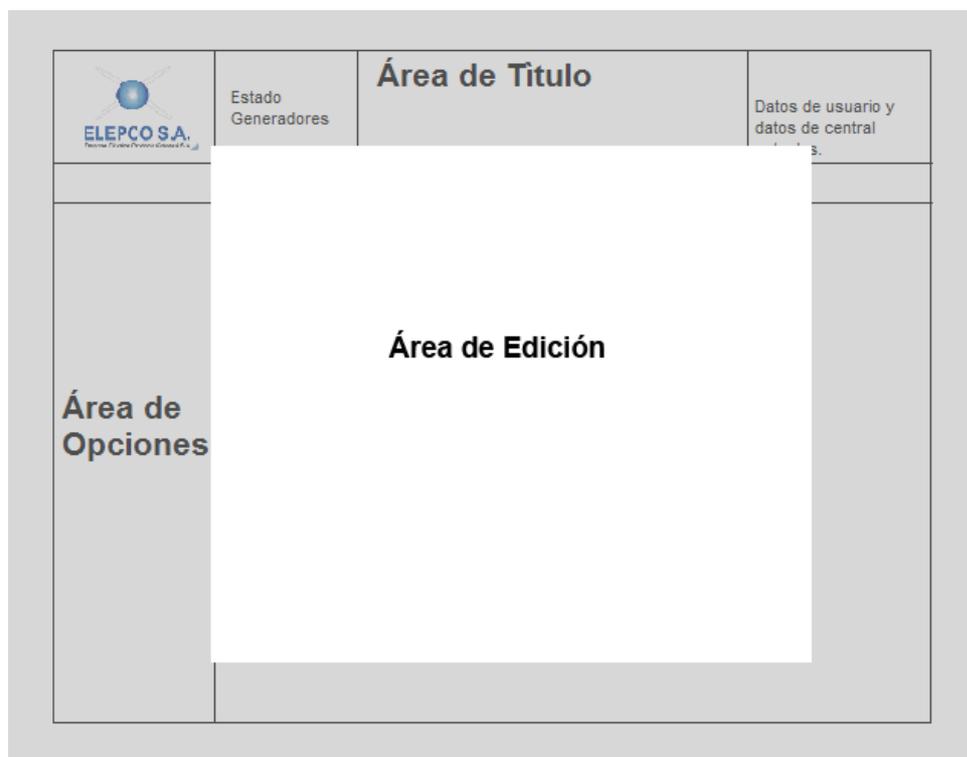


Gráfico 4.9: Prototipo del área de edición.

⁵⁹ <http://jquery.com/demo/thickbox/>

Estilo del Texto

Para controlar el estilo del contenido a desplegar en las páginas (Tipo de letra, color, tamaño, efectos) se usará las Hojas de Estilo en Cascada (CSS, Cascading style sheet) por defecto se utilizará la siguiente configuración para el texto:

- Fuente: Arial
- Tamaño: 13 px (pixels)
- Color: Negro (#000000)

Tabla 4.9 Archivos CSS usados en el proyecto.

Archivo	Descripción
Styles	Estilos generales para todas las páginas.
Menu_styles	Estilos específicamente para menús
ThickBox	Estilos usados por el utilitario ThickBox para el área de edición.

Estructura de archivos.

Es importante definir la estructura de archivos que se utilizará para organizar el proyecto, esta estructura está relacionada con la arquitectura del sistema que se construirá.

Para el desarrollo de este subsistema se usará una arquitectura de 3 capas, la primera capa para el Modelo o el acceso a datos que ha sido generada mediante Propel, la segunda capa para las Vistas o elementos q serán las interfaces visuales para el usuario y la tercera capa para los Controladores o la capa de la lógica del negocio que se encargará de procesar la información. Con esto se tendrá el sistema de archivos principal que será:

- **Monitor (Directorio del proyecto Monitor)**

- Build.

En este directorio estarán todos los archivos generados por Propel ORM, toda la estructura que se vio según el gráfico 4.9, tiene los subdirectorios:

- Classes
- Conf

- **Vistas**

Aquí se encuentran todas las interfaces gráficas que se mostrarán al usuario y están agrupadas en función de la navegabilidad del sistema, es decir; una opción tendrá asignada una vista del sistema, las vistas estarán agrupadas en diferentes subdirectorios dependiendo el control mediante el cual serán usados.

- **Controlador**

Este directorio será donde se almacenarán las clases de control o lógica de negocio del subsistema. Se creará una clase controlador por cada clase de persistencia.

Nomenclaturas

En un desarrollo donde se generan múltiples objetos y mucho código se debe nombrar cada elemento (control, tabla, clase, página) de forma que sea entendible para el programador y ayude al mantenimiento del software, por lo

general se utiliza un prefijo o posfijo por el tipo de elemento o por un código que identifique a un grupo al que pertenece el objeto.

En este proyecto se usó la siguiente nomenclatura:

- Controles Visuales de las páginas HTML:

Tabla 4.10 Nomenclatura usada en Controles Visuales de las páginas HTML

Prefijo	Etiqueta	Descripción
Txt	Text	Campos de ingreso de texto
Btn	Button	Botones que ejecutan una acción.
Frm	Form	Formularios
Hid	Hidden	Campos ocultos que contienen información.
cmb	Combo	Listas desplegables

- Tablas de la base de datos, se considero no utilizar prefijos o posfijos por la razón que para que Propel genere las clases de acceso de persistencia con nombres más claros, entonces los nombres de las tablas deberán ser los nombres completos de la entidad a la que corresponden.
- Para la capa de acceso a datos Propel genera su propia nomenclatura con prefijos y posfijos y añadiendo el nombre de la tabla, se generan 7 archivos por tabla, a continuación se detalla la nomenclatura poniendo como ejemplo que las clases pertenecen a la tabla PERSONA:

Tabla 4.11 Nomenclatura de clases, ejemplo tabla PERSONA

Prefijo	Posfijo	Ejemplo	Descripción
	TableMap	PersonaTableMap	Define la estructura de la tabla PERSONA
Base		BasePersona	Esta clase representa una fila de la tabla PERSONA
Base	Peer	BasePersonaPeer	Clase estática para mejorar las consultas y operaciones de actualización de la tabla PERSONA
Base	Query	BasePersonaQuery	Representa un query (consulta) para la tabla PERSONA
		Persona	Subclase heredada de BasePersona que representa una fila de la tabla PERSONA.
	Peer	PersonaPeer	Subclase heredada de BasePersonaPeer
	Query	PersonaQuery	Subclase heredada de BasePersonaQuery

- Para la capa de lógica de negocio o control se van a nombrar las clases colocando el nombre de la entidad más la palabra Controlador; por ejemplo, el controlador de la entidad PERSONA tendrá el nombre de PersonaControlador.

4.3.5.3.- Administración de menús y opciones.

Esta funcionalidad permitirá administrar la navegación del subsistema, se ha definido dos niveles que son los Menús y las Opciones, esta navegación va a estar relacionado con la capa de Vistas del subsistema. Las opciones tienen

asignadas una vista del subsistema y poseen un campo donde se indica la clase controlador y la función controlador a llamar, los Menús en cambio agruparán varias opciones. Por defecto se creará un subdirectorio por clase de persistencia para agrupar en estas carpetas las vistas y pequeños despliegues o algún servicio que se requiera como las acciones de editar, guardar, eliminar que estén relacionados con esta clase.

La navegación del sistema es la siguiente:

Al momento que el usuario Inicie Sesión en el área de menús se desplegará los menús a los que este posea permisos, una vez que el usuario seleccione un menú, en el área de opciones se deben desplegar las opciones a las que tiene derechos.

Con esta información se puede generar un diagrama UML de objetos de cómo está representada esta parte del subsistema.

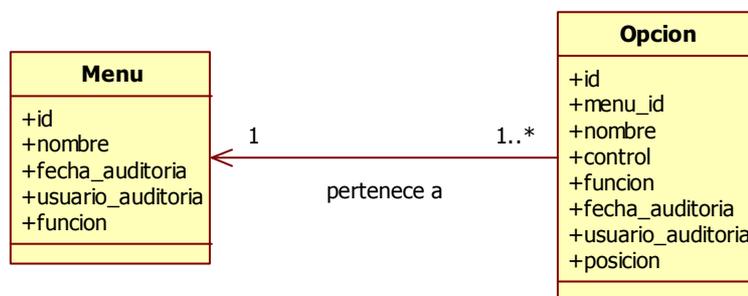


Gráfico 4.10 Diagrama UML para administración de menús y opciones.

Esta historia de usuario se divide en 3 tareas:

1. Página contenedora del subsistema Monitor
2. Página de administración de Menús
3. Página de administración de Opciones.

Página contenedora del subsistema Monitor

Como se vio en el apartado de arquitectura del sistema, la página base del subsistema tiene la forma del gráfico 4.9, la tarea de este apartado será armar el código HTML para diseñar la página con las secciones indicadas, para luego en las tareas siguientes ir agregando la funcionalidad para cada sección.

Se ingresan datos en las tablas de Menú y Opciones para poder desplegar datos en las áreas correspondientes y una vez hecho esto se logra obtener el diseño base del sistema que es el siguiente:

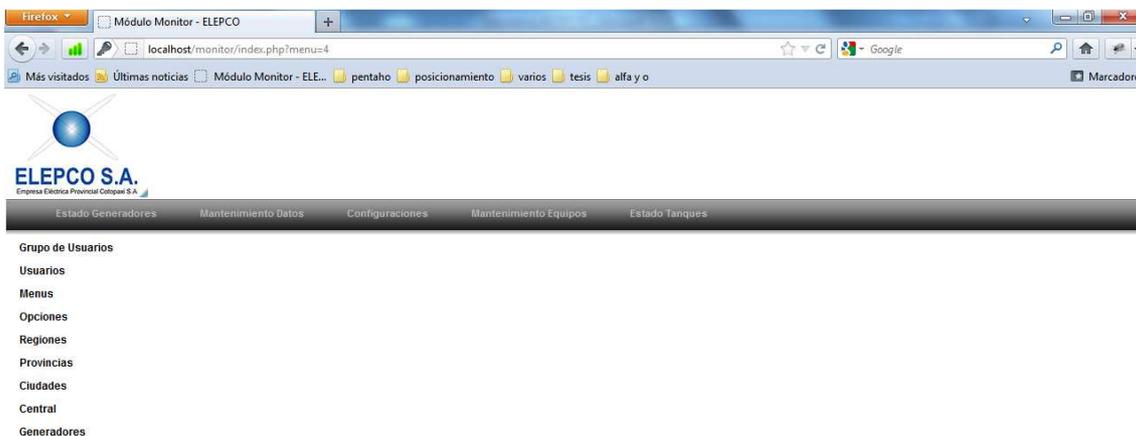


Imagen 4.10 Diseño base del sub sistema Monitor.

Página de administración de Menús

Un menú consta de varios atributos, entre los cuales se tiene:

- id: Identificador del menú
- nombre: Corresponde al nombre del menú (este nombre se desplegará en pantalla).
- fecha_auditoria: es la fecha en la que se realizó el último cambio.
- usuario_auditoria: es el usuario quien modificó el registro por última vez.

Siguiendo el patrón de diseño que se indicó en la parte de arquitectura del sistema, se desarrolla una vista “listar.php” que se agregó en una carpeta de nombre “menu” en la capa de vistas y se crean tres archivos en un subdirectorio en la capa de vistas con nombre “menú” estos tres archivos corresponden para las acciones de editar, eliminar, guardar.

Además se crea un control en el directorio Controlador con nombre “MenuControlador.php”.

El diseño de esta página es el patrón con el que se desarrolló el resto de páginas de administración de datos, se diseñó un estilo de tabla para desplegar los registros, junto a cada registro existe una opción de editar y eliminar, y se tiene una barra de herramientas donde se despliega un botón para agregar un nuevo registro, y otros para desplegarse entre los registros, por defecto se mostrarán 10 registros por página.

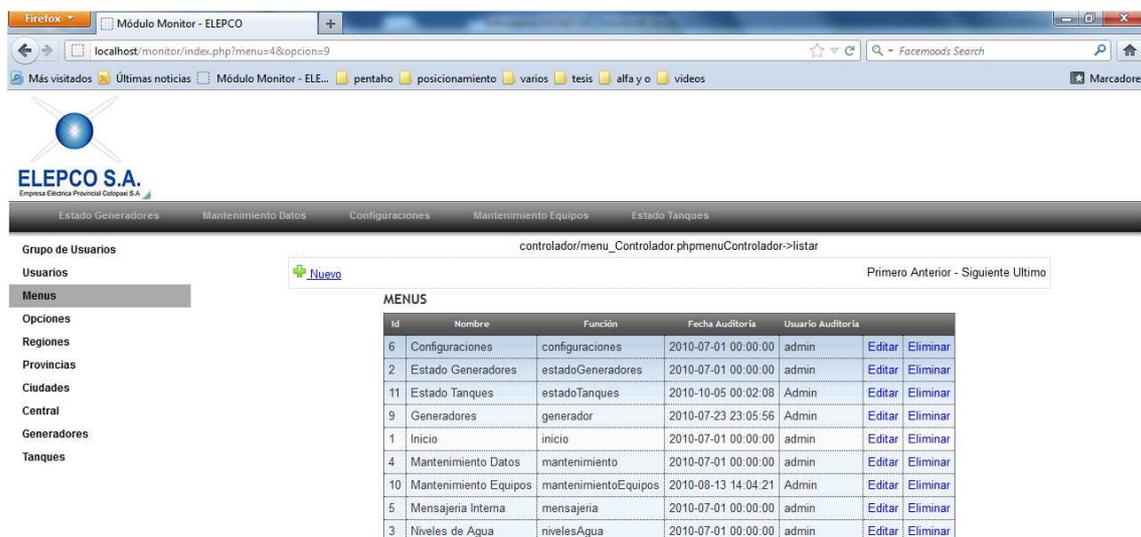


Imagen 4.11 Administración de Menús.

El formulario de edición para agregar o editar un registro de la tabla se presenta en el área de edición y tiene el siguiente aspecto:

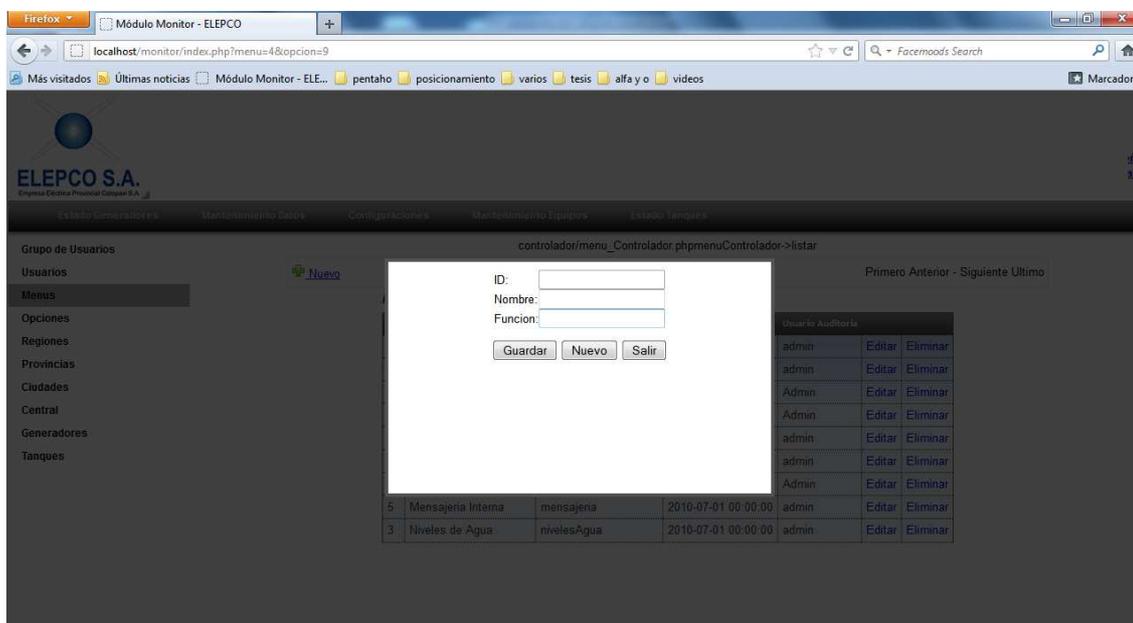


Imagen 4.12 Formulario de edición para agregar o editar un registro.

Para la validación del formulario se usará código en JavaScript; es decir, que la máquina cliente será la encargada de realizar el trabajo de validación, con el objetivo de evitar el envío de datos al servidor y disminuir el tiempo que se demora en esta acción, así el servidor solo se encargará de recibir los datos y ejecutar la acción de inserción, edición o eliminación.

Para ejecutar las acciones de guardar y eliminar en cambio se usará una función Ajax que llamará mediante el método GET (método para pasar valores entre páginas) a la vista guardar.php o eliminar.php que ejecutará la acción en el objeto de persistencia y devolverá el resultado de la ejecución.

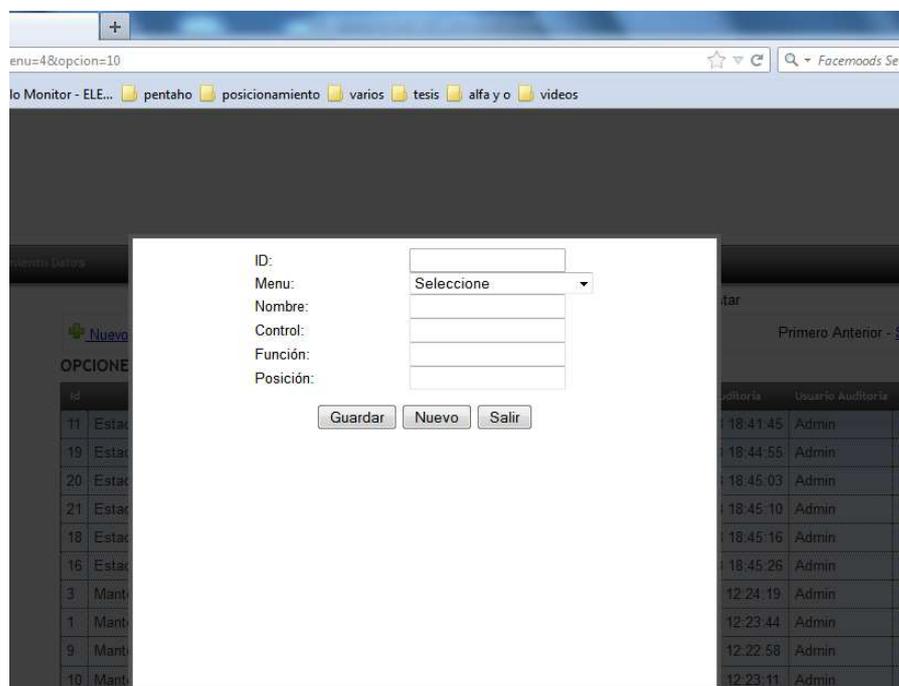
Página de administración de Opciones.

Como se habló al inicio de este apartado una opción representa una vista del proyecto, al momento que el usuario ingresa en una opción se ejecuta un método de una clase control y este a su vez llama a la vista correspondiente.

Para esto los campos necesarios para la tabla de opciones son:

- Id: código único del registro.
- menu_id: código del menú contendor.
- Nombre: nombre de la opción.
- Control: nombre de la clase controlador.
- Función: nombre del método de la clase controlador.
- usuario_auditoria: usuario quien registro el último cambio.
- fecha_auditoria: fecha del último cambio

Se mantiene el mismo patrón de diseño que se vio en el punto anterior. El formulario de ingreso de datos queda de la siguiente forma.



The image shows a web browser window displaying a data entry form for 'OPCIONE'. The form is centered on the screen and contains the following fields and controls:

- ID: A text input field.
- Menu: A dropdown menu with the text 'Seleccione'.
- Nombre: A text input field.
- Control: A text input field.
- Función: A text input field.
- Posición: A text input field.

Below the form fields are three buttons: 'Guardar', 'Nuevo', and 'Salir'. The background of the browser shows a table with columns for 'ID', 'Estat', and 'Mant', and another table with columns for 'Fecha Auditoria' and 'Usuario Auditoria'.

Imagen 4.13 Formulario de ingreso de datos.

4.3.5.4.- Administración de usuarios y grupos de usuarios.

Esta funcionalidad del subsistema de administración tiene por objetivo permitir la administración de cuentas de usuarios. Los usuarios son clasificados en grupos de usuarios dependiendo el perfil que tenga el usuario, por defecto se crearán tres grupos de usuario que serán administradores, operadores y jefes.

Para evitar accesos no autorizados desde la aplicación se deben establecer contraseñas a las cuentas de usuario.

El siguiente diagrama UML corresponde a la representación de las clases grupo de usuario y usuario

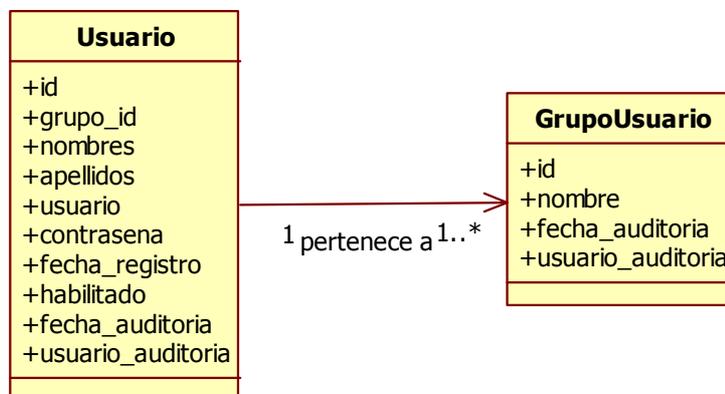


Gráfico 4.11 Diagrama UML para grupo de usuario y usuario

Esta historia de usuario se divide en 2 tareas:

1. Página de administración de Grupos de Usuarios
2. Página de administración de Usuarios

Página de administración de Grupos de Usuarios.

El objetivo de esta página es gestionar los grupos de usuarios que tendrá el subsistema, un grupo de usuario corresponde a un perfil de usuario, es decir; que los permisos a las opciones del sistema serán asignados a nivel de grupo de usuario. Las acciones que un usuario podrá realizar en esta página serán:

- Creación de grupos de usuario.
- Eliminación de grupos de usuario.
- Modificación de grupos de usuario.

Los campos necesarios para crear la tabla son:

- Id: código único del registro.
- Nombre: nombre del grupo de usuario.
- Fecha_auditoria: fecha del último cambio.
- Usuario_auditoria: usuario quien realizó el último cambio.

Las siguientes capturas de pantallas son el diseño de la página de administración de grupos de usuarios, el diseño sigue el patrón detallado en los puntos anteriores.

Imagen 4.14 Formulario del área de despliegue de contenido.



Logo: ELEPCO S.A. Empresa Eléctrica Provincial Cotacachi S.A.

Estado Generadores | Mantenimiento Datos | Configuraciones | Mantenimiento Equipos | Estado Tanques

Grupo de Usuarios

Usuarios

Menus

Opciones

Regiones

Provincias

Ciudades

Central

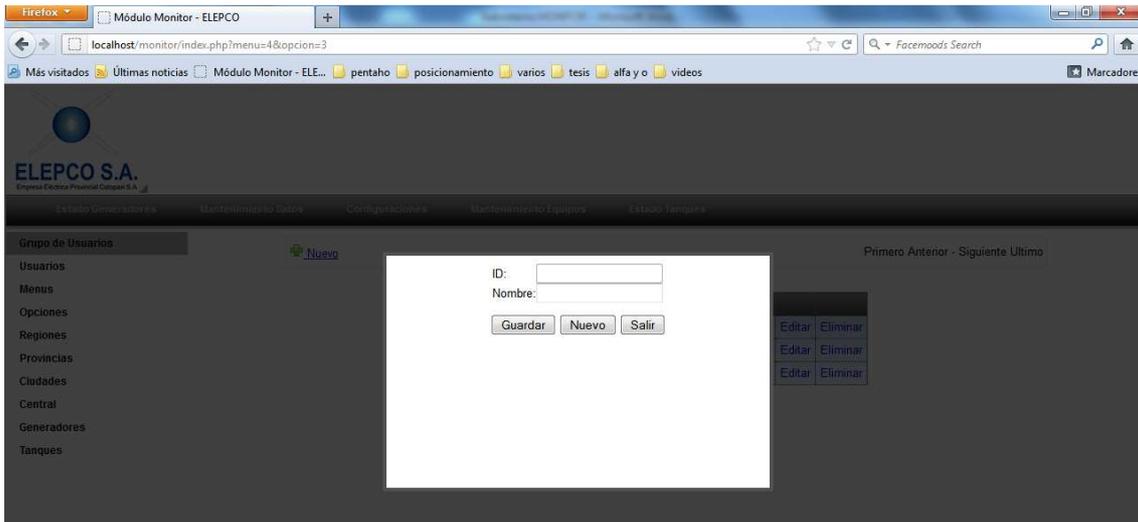
Generadores

Tanques

[Nuevo](#) Primero Anterior - Siguiente Ultimo

GRUPOS DE USUARIOS						
Id	Nombre	Fecha Auditoria	Usuario Auditoria			
1	Administradores	2012-03-04 12:28:44	Admin	Editar	Eliminar	
4	Jefes	2012-05-06 20:38:00	Admin	Editar	Eliminar	
2	Operadores	2010-07-24 17:56:11	Admin	Editar	Eliminar	

Imagen 4.15 Formulario del área de edición



Página de administración de Usuarios.

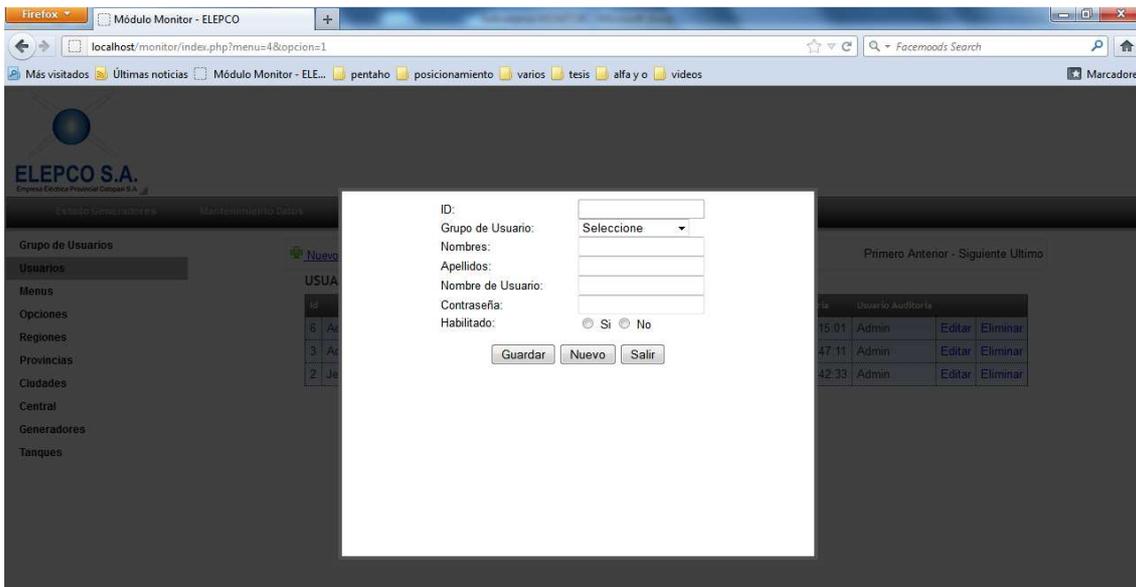
El principal objetivo de esta página es administrar los usuarios quienes tendrán acceso al sistema, las funciones de esta página son:

- Creación de usuarios, asignando un grupo de usuario y contraseña.
- Eliminación de usuarios.
- Modificación de datos de usuario.

Imagen 4.16 Formulario de administración de usuarios



Imagen 4.17 Formulario de administración de usuarios, opción “Nuevo Usuario”



Administración de permisos de grupos de usuarios a menús y opciones.

Esta función del subsistema permitirá asignar permisos a los grupos de usuarios a las diferentes opciones del sistema, además de controlar el acceso de los usuarios al sistema. El diagrama UML a continuación muestra la relación entre la clase Grupo de Usuario y Permisos.

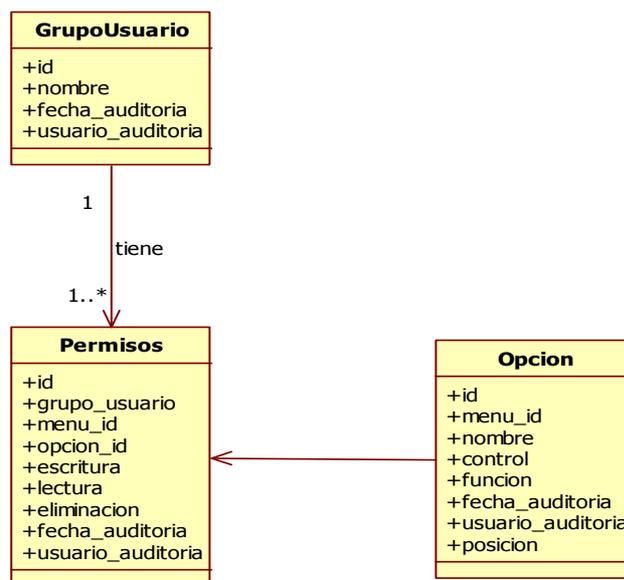


Gráfico 4.12 Relación entre la clase Grupo de Usuario y Permisos

Las tareas que se realizarán en esta historia de usuario serán:

- Página de administración de permisos de grupos de usuarios a menús y opciones.
- Página de inicio de sesión.

Página de administración de permisos de grupos de usuarios a menús y opciones.

En esta página el usuario podrá realizar las asignaciones de permisos de los usuarios a las diferentes opciones del sistema.

Para los grupos de usuario por default que tendrá el sistema se asignarán permisos a las opciones dependiendo el perfil que representan, pero el sistema permitirá la creación de más grupos de usuarios y la personalización de sus derechos.

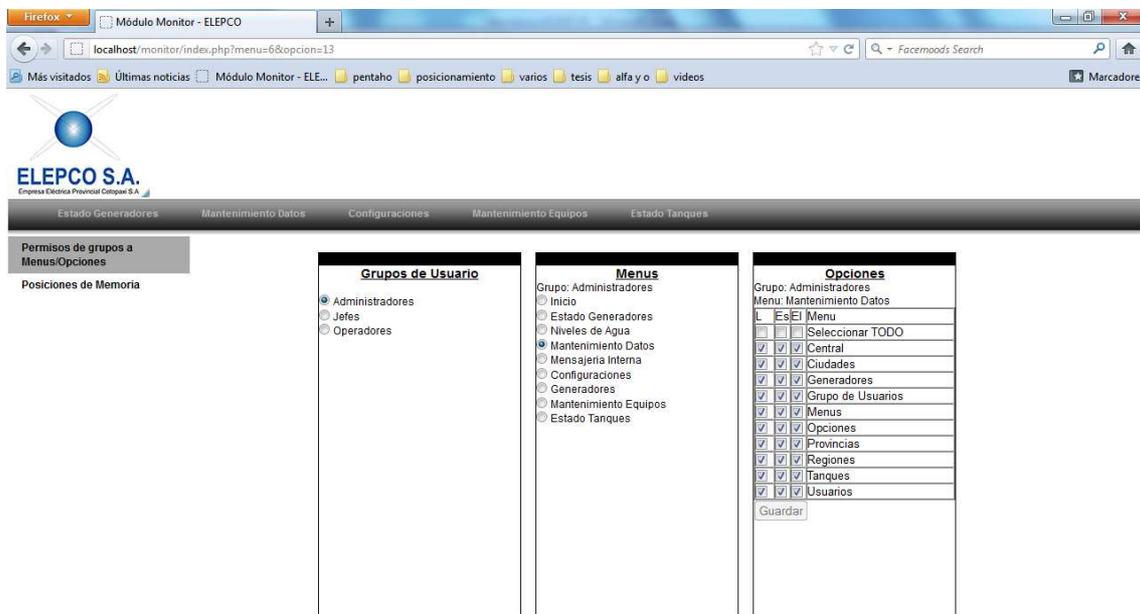


Imagen 4.18 Página de asignación de grupos de usuarios a menús y opciones

El usuario deberá elegir el grupo de usuario y automáticamente mediante una función Ajax el sistema desplegará los menús disponibles del sistema, luego al escoger un menú así mismo se desplegarán las opciones que contiene el menú, los permisos a cada opción pueden ser de lectura, escritura y eliminación.

Página de inicio de sesión.

La página de Inicio de sesión, o de Login, es el encargado de autenticar al usuario antes de permitir acceso a los diferentes módulos del subsistema de administración. Este formulario debe permitir el acceso únicamente a usuarios que están habilitados, es decir, para que un usuario pueda lograr acceso al sistema debe contar con un nombre y contraseña válidos y tener habilitada su cuenta.

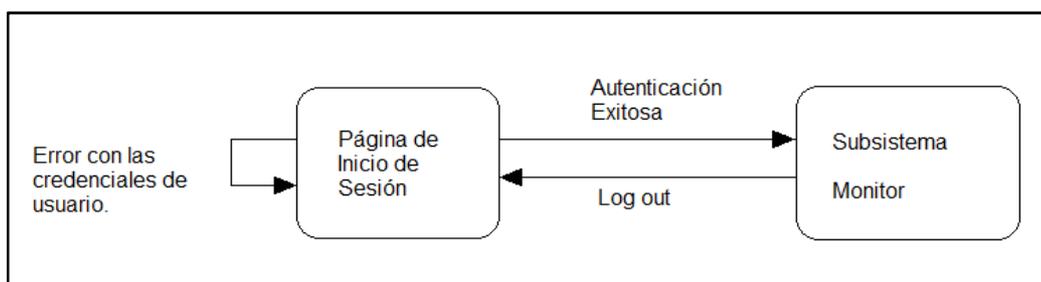


Gráfico 4.13 El proceso de autenticación

4.3.5.5.- Administración de estructura geográfica.

El objetivo de esta función es administrar la estructura geográfica de las centrales hidroeléctricas para poder agrupar los valores generados por cada nivel de la estructura. Como se vio anteriormente ELEPCO cuenta con varias centrales distribuidas en toda la provincia de Cotopaxi en diferentes cantones. La estructura

constará de tres niveles: Región, Provincia y Ciudad más un registro de la Central de Generación que se relaciona con los tres niveles, a continuación se muestra el diagrama UML con las respectivas clases para esta función:

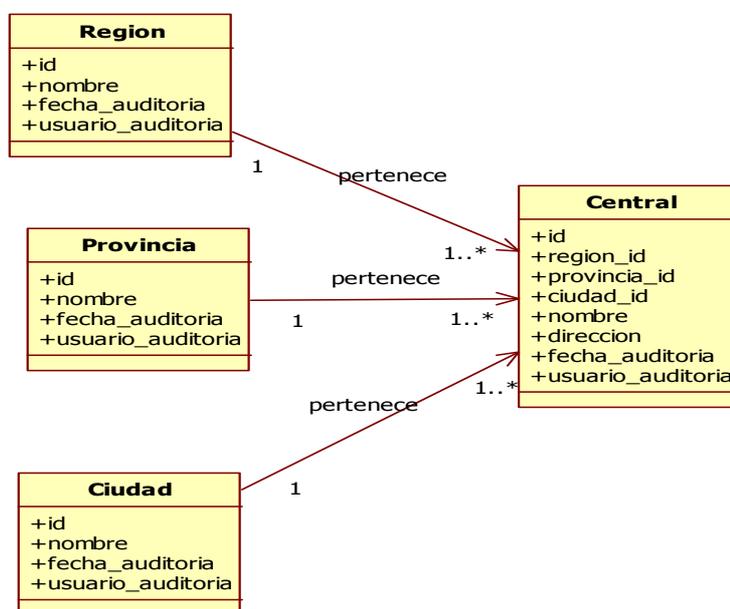


Gráfico 4.14 Clases de la función administrar estructura geográfica

Las tareas a realizar en esta historia de usuario serán las siguientes:

- Página de administración de Regiones
- Página de administración de Provincias
- Página de administración de Ciudades
- Página de administración de Centrales

Página de administración de Regiones.

Aquí se administrarán todos los datos de las regiones donde ELEPCO posee centrales de generación, el desarrollo de esta página será siguiendo el patrón de las paginas anteriores, se tendrán las mismas 2 áreas de contenido y

de edición, las capturas a continuación muestra el diseño de cada una de estas áreas.

Página de administración de Provincias.

Esta página se refiere al mantenimiento de datos de las provincias donde se encuentran las centrales de generación.

Página de administración de Ciudades.

En este punto se desarrollará la administración de Ciudades, que es un nivel más de la estructura de una de una central hidroeléctrica, aquí se ingresarán todas las ciudades donde se encuentren las Centrales Hidroeléctricas.

Página de administración de Centrales.

Una vez concluido las administraciones de Región, Provincia y Ciudad se continúa con el desarrollo de la página de administración de centrales hidroeléctricas, se han considerado necesarias los siguientes campos para esta opción:

- Id: es el código único con el que se identificará la central hidroeléctrica
- Region_id: código de la región
- Provincia_id: código de la provincia
- Ciudad_id: código de la ciudad
- Nombre: nombre de la central hidroeléctrica
- Direccion: dirección de la central
- Fecha_auditoria: fecha del último cambio del registro

- Usuario_auditoria: usuario quien realizó el último cambio



Imagen 4.19 Administración de centrales, área de despliegue de información.

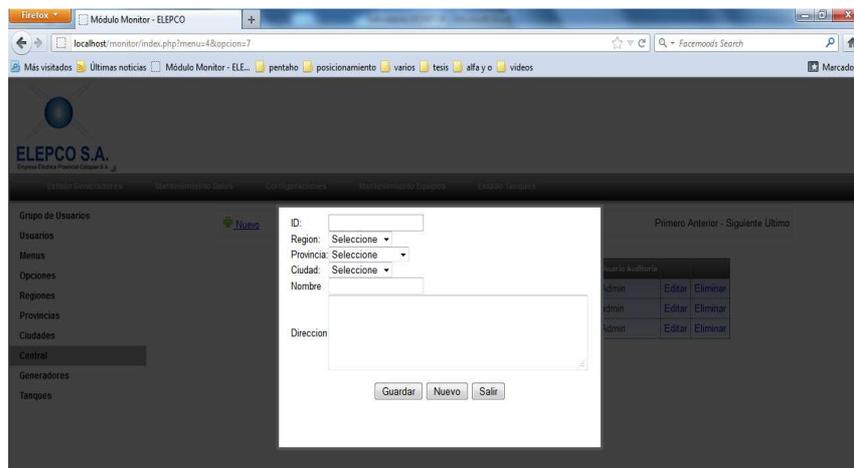


Imagen 4.20 Administración de centrales, área de edición.

4.3.5.6.- Carta de Burndown

Una vez finalizado el desarrollo del primer sprint, se completa la pila del sprint actualizando el día de finalización y los puntos restantes y se genera el gráfico del Burndown.

Tabla 4.12: Información para la carta de Burndown al finalizar el primer Sprint.

No	Historia de usuario	Categoría	Estimación inicial en puntos.	Finalización	Puntos Restantes
6	Administración de menús y opciones.	Monitor	8	Día 10	177
1	Administración de usuarios y grupos de usuarios.	Monitor	6	Día 13	171
5	Asignación de permisos de grupos de usuarios a menús y opciones.	Monitor	10	Día 20	161
7	Administración de estructura geográfica (Región, Provincia, Ciudad, Central) de los equipos.	Monitor	13	Día 24	148

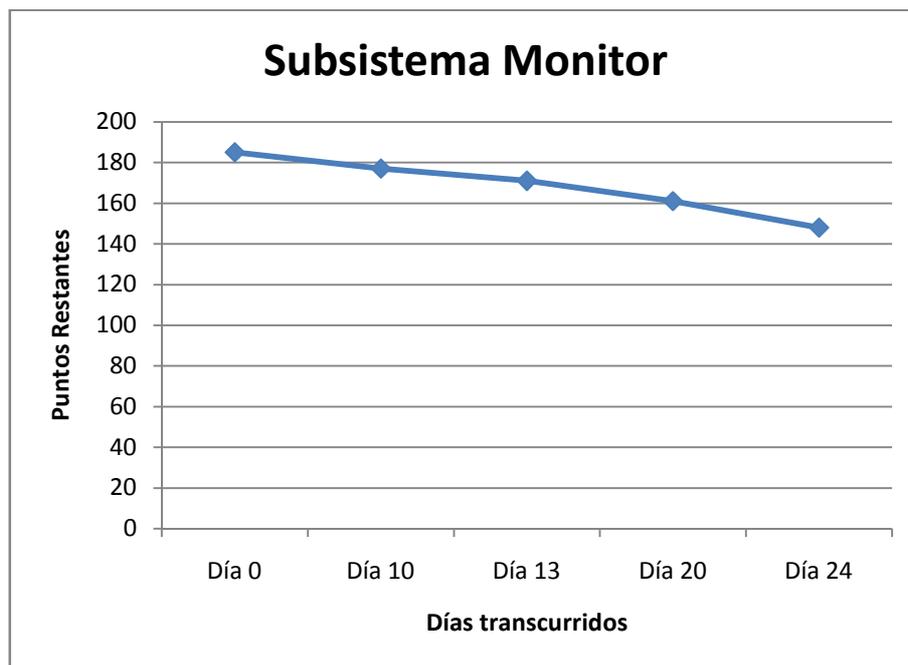


Gráfico 4.15: Avance del proyecto.

El gráfico anterior corresponde al gráfico Burndown de esta iteración, como se puede apreciar se ha logrado avanzar una buena parte del proyecto debido a que se estableció un patrón de desarrollo lo que agilizó en gran manera la velocidad del desarrollo, además porque organizó el proyecto dividiéndole en capas y la utilización de un ORM que ayudo con la generación de la capa de acceso a datos.

4.3.5.7.- Revisión.

Una vez terminado el desarrollo de las actividades planificadas en este sprint se presenta el avance al dueño del producto. El objetivo es presentar la funcionalidad del sistema, y reunir ideas y sugerencias de lo que debería hacerse a continuación.

El dueño del producto pide que se agregue un campo que indique el orden de las opciones para mostrar en el menú, este requerimiento se agrega a la pila del producto para en el siguiente sprint procesar el requerimiento.

4.3.6.- Segundo Sprint.

4.3.6.1.- Planificación.

Para este segundo sprint el objetivo planteado es continuar desarrollando las opciones que son importantes para el funcionamiento del sistema en base a la arquitectura planteada en el primer sprint, antes de seleccionar las historias de usuario a desarrollar se debe aumentar una, agregar una columna que especifique la posición a aparecer una opción en el menú.

Tabla 4.13 Pila del producto después del primer sprint.

No	Historia de usuario	Categoría	Importancia	Estimación inicial en puntos.	Completado
1	Administración de usuarios y grupos de usuarios.	Monitor	5	6	✓
2	Administración de estructura geográfica (Región, Provincia, Ciudad, Central) de los equipos.	Monitor	5	13	✓
3	Administración de Generadores (GPU's).	Monitor	5	6	✗
4	Administración de tanques.	Monitor	5	4	✗
5	Administración de permisos de grupos de usuarios a menús y opciones.	Monitor	8	10	✓
6	Administración de menús y opciones.	Monitor	6	8	✓
7	Administración de la Configuración de Memoria.	Monitor	6	4	✗
8	Alarmas de aviso de eventos importantes.	Monitor	4	6	✗
9	Generar gráficos dinámicos actualizables durante el transcurso del tiempo del estado de generación eléctrica por grupo de generación.	Monitor	6	24	✗
10	Generar gráficos históricos de generación eléctrica con filtros.	Monitor	6	48	✗
11	Generar un gráfico de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.	Monitor	5	20	✗

12	Generar gráficos dinámicos de niveles de agua por tanque actualizables durante el transcurso del tiempo.	Monitor	6	30	x
13	Administración de la planificación de mantenimiento de generadores.	Monitor	8	6	x
				185	

Para la pila del segundo sprint se considera las revisiones realizadas en el primer sprint y así mismo se toman las historias de usuario que deben ser de mayor importancia para el avance del proyecto, se espera que en este sprint se agilice el desarrollo puesto que ya existe una arquitectura definida.

Tabla 4.14 Pila del Segundo Sprint, subsistema Monitor

No	Historia de usuario	Categoría	Importancia	Estimación inicial en puntos.
14	Añadir campo de orden en Opciones	Monitor	3	2
3	Administración de Generadores (GPU's.)	Monitor	5	6
4	Administración de tanques.	Monitor	5	4
7	Administración de la Configuración de Memoria.	Monitor	6	4
13	Administración de la planificación de mantenimiento de generadores.	Monitor	8	6
8	Alarmas de aviso de eventos importantes.	Monitor	4	6

Debido a que la historia de usuario “Añadir campo de orden en opciones” es muy pequeña no se incluirá documentación de este desarrollo.

4.3.6.2.- Administración de Generadores (GPU's).

El objetivo de esta parte del sistema es registrar los generados GPU de la central hidroeléctrica, con la finalidad de poder identificar los valores que entrega cada equipo. La tabla que se ocupará en este desarrollo ya fue creada en el desarrollo del subsistema de lectura de datos por su relación con la tabla donde se registran las lecturas de los GPU's. Se deberá también registrar la Central de Generación a la que pertenece cada equipo.

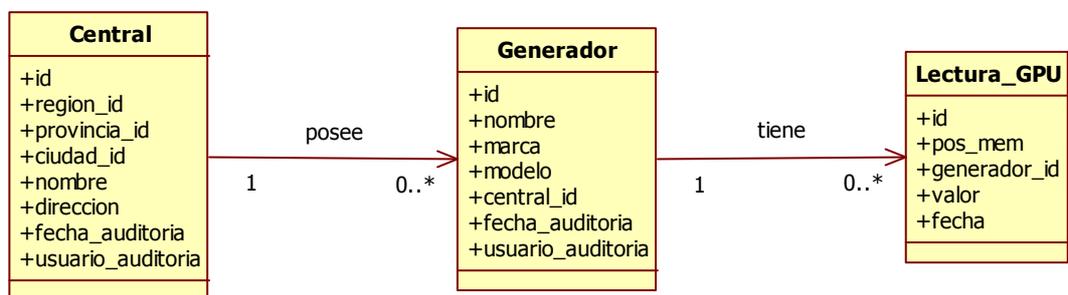


Gráfico 4.16 Diagrama UML para administración de generadores (GPU's)

Como se puede apreciar la única tarea a realizar será la página de administración de Generadores. Por lo que se procede directamente a diseñar las páginas y a su inmediato desarrollo.



Imagen 4.21 Formulario de administración de generadores.

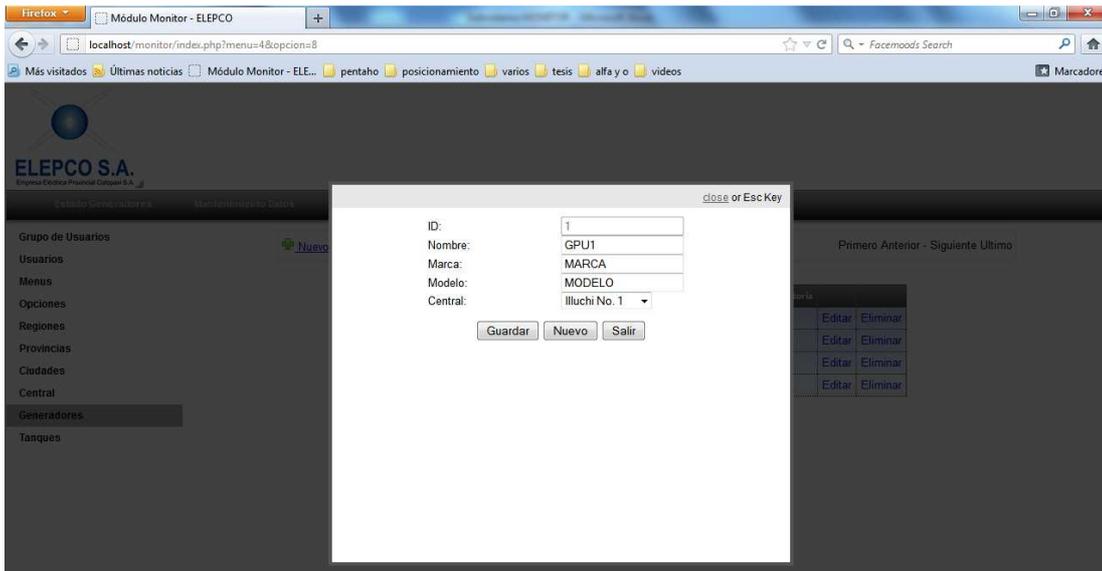


Imagen 4.22 Administración de generadores, opción “Nuevo Generador”.

4.3.6.3.- Administración de Tanques.

De la misma forma que la administración de GPU's, la administración de tanques tiene como objetivo registrar datos de información de los tanques de agua, está relacionado también con el subsistema de lectura de datos por lo que en esta sección se creará una clase de lectura de tanques.

Se debe registrar la capacidad total del tanque, el nivel máximo que debe alcanzar el agua, el nivel mínimo y la central en la que se encuentra.

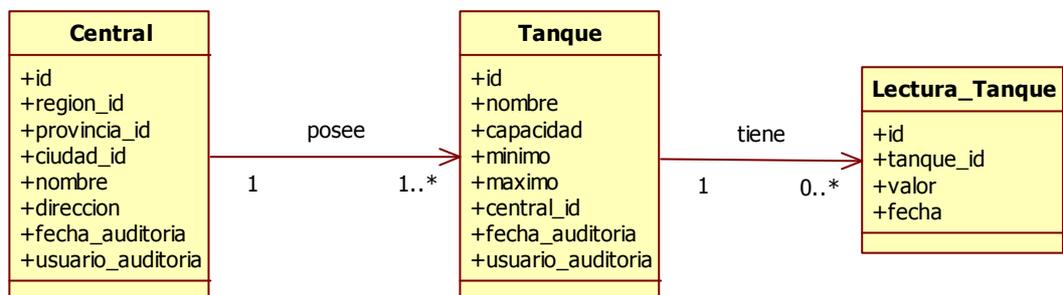


Gráfico 4.17 Diagrama UML para administración de tanques

Se procede con el desarrollo de la única tarea para esta historia de usuario que es desarrollar la Página de Administración de Tanques.



Imagen 4.23 Formulario de administración de tanques.

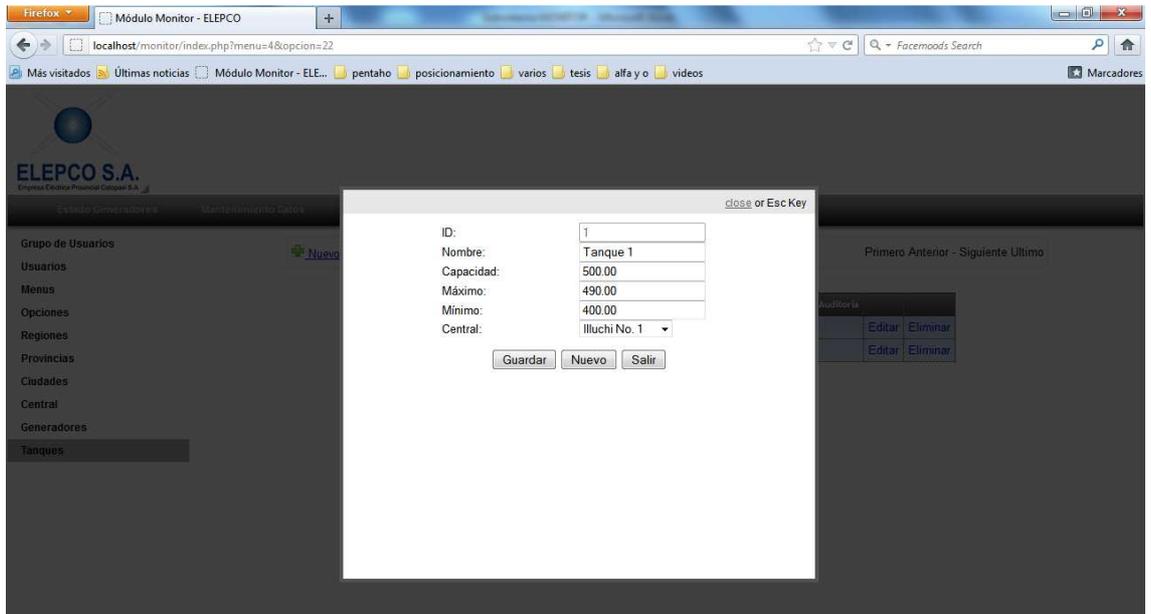


Imagen 4.24 Administración de Tanques, opción "Nuevo Tanque".

4.3.6.4.- Administración de la Configuración de Memoria.

Cada GPU contiene posiciones de memoria donde almacena los valores de las mediciones, cada lectura que se almacena contiene esta posición de memoria, el objetivo de esta página es registrar estas posiciones de memoria identificando el valor que representan y la unidad de medida.

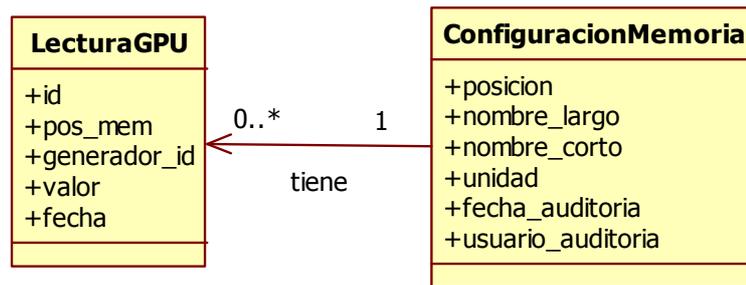


Gráfico 4.18 Relación de Configuración de Memoria con GPU's

Posición	Nombre Largo	Nombre Corto	Unidad	Fecha Auditoria	Usuario Auditoria	Editar	Eliminar
40001	Corriente A	Corriente A	W	2011-03-02 22:27:42	Admin	Editar	Eliminar
40002	Corriente B	Corriente B	W	2011-03-02 22:27:56	Admin	Editar	Eliminar
40003	Corriente C	Corriente C	W	2011-03-02 22:28:13	Admin	Editar	Eliminar
40004	Volataje de Linea A	Volataje de Linea A	V	2011-03-02 22:37:12	Admin	Editar	Eliminar
40005	Volataje de Linea B	Volataje de Linea B	V	2011-03-02 22:37:33	Admin	Editar	Eliminar
40006	Volataje de Linea C	Volataje de Linea C	V	2011-03-02 22:37:52	Admin	Editar	Eliminar
40007	Potencia A	Potencia A	P	2011-03-02 22:41:21	Admin	Editar	Eliminar
40008	Potencia B	Potencia B	P	2011-03-02 22:41:14	Admin	Editar	Eliminar
40009	Potencia C	Potencia C	P	2011-03-02 22:41:04	Admin	Editar	Eliminar
40010	posicion 40010		SU	2010-08-25 23:57:25	admin	Editar	Eliminar

Imagen 4.25 Formulario de configuración posición de memorias GPU.

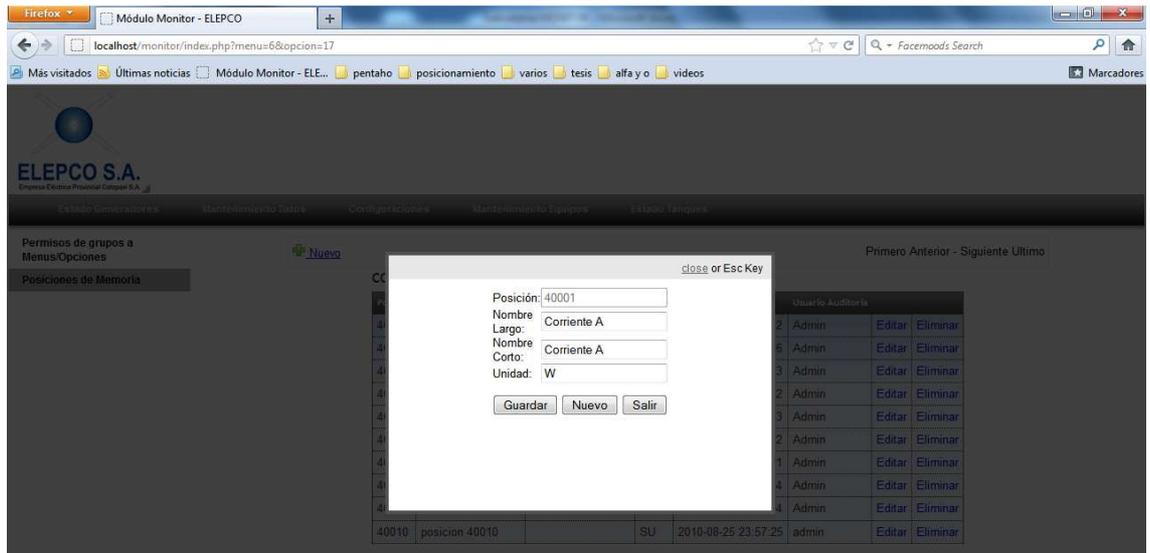


Imagen 4.26 Configuración posición de memorias GPU, “Nueva Posición”.

4.3.6.5.- Administración de la planificación de mantenimiento de generadores.

El objetivo de esta opción es registrar el mantenimiento de los equipos de campo (Generador o Tanque) con el fin de llevar un control de paros de la generación eléctrica a causa de mantenimientos. A continuación el diagrama de clases a implementar para esta funcionalidad.

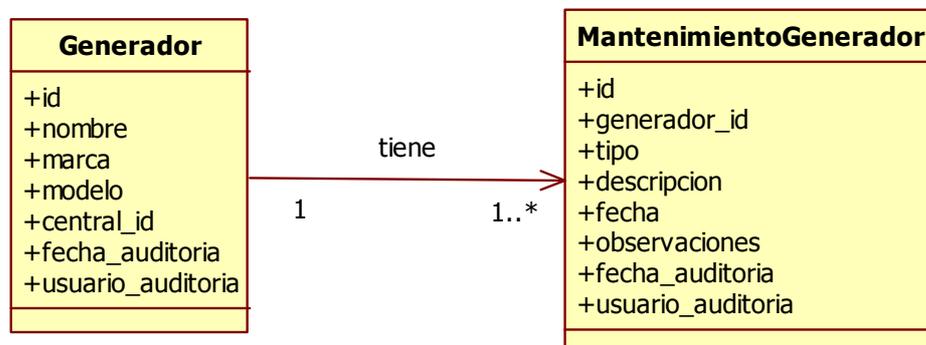


Gráfico 4.19 Diagrama de clases a implementar en Generador o Tanque

Como se puede ver, esta función es un formulario sencillo para registrar los mantenimientos, se usará el mismo patrón que se ha venido usando.

Id	Generador	Tipo	Descripción	Fecha	Observaciones	Estado	Fecha Auditoria	Usuario Auditoria
2	GPU1	Preventivo	Calibración de los ejes.	2010-08-20 15:20:00	No existen observaciones.	T	2012-05-26 12:13:41	Admin
3	GPU1	Preventivo	Engrasar ejes.	2010-08-01 00:00:00	No existieron novedades.	T	2012-05-26 12:14:29	Admin
4	GPU2	Correctivo	Cambio de bandas.	2010-08-01 00:00:00	La banda tuvo que reemplazarse porque la anterior se rompió inesperadamente.	T	2012-05-26 12:15:57	Admin
5	GPU3	Preventivo	Engrasar ejes	2010-08-18 18:00:00		T	2012-05-26 12:16:22	Admin
6	GPU2	Preventivo	Engrasar Ejes	2010-08-01 00:00:00		T	2012-05-26 12:16:45	Admin
7	GPU1	Preventivo	Cambio de resistencias	2010-08-01 00:00:00		T	2012-05-26 12:17:19	Admin
8	GPU2	Preventivo	Colocacion de nuevos empaques.	2010-08-01 00:00:00		T	2012-05-26 12:18:07	Admin
9	GPU2	Preventivo	Cambio de bandas	2010-08-01 00:00:00	Bandas desgastadas.	T	2012-05-26 12:18:45	Admin
12	GPU4	Preventivo	Engrasar ejes.	2008-03-01 15:30:00		T	2012-05-26 12:19:51	Admin
13	GPU4	Preventivo	Cambio de bandas	2010-09-14 17:24:13	Bandas desgastadas.	T	2012-05-26 12:20:35	Admin

Imagen 4.27 Formulario de mantenimiento de generadores

close or Esc Key

ID:

Generador:

Tipo:

Descripción:

Fecha:

Observaciones:

Estado:

Imagen 4.28 Formulario de mantenimiento de generadores, "Nuevo Generador".

4.3.6.6.- Alarmas de aviso de eventos importantes.

El objetivo de esta funcionalidad será presentar en una área de la página principal el estado (encendido o apagado) de cada equipo generador de energía, para lo que se ha dispuesto en la parte superior, alado del logo de la empresa colocar esta funcionalidad, tal como se indico en la plantilla inicial en el primer sprint en el Gráfico 4.8.

Para esto se tomará el último valor de la tabla de lecturas de generación y si este es mayor a cero (0) entonces el generador esta encendido, caso contrario se encuentra apagado.

En la plantilla index.php del proyecto se aumenta una sección donde se desplegarán los datos, la siguiente captura de pantalla muestra el diseño desarrollado.



Imagen 4.29 Alarmas de aviso de eventos importantes

4.3.6.7.- Carta de Burndown

Se actualiza la pila del sprint colocando el día de finalización y los puntos restantes, con estos datos se generan también el gráfico de Burndown.

Tabla 4.15: Información para la carta de Burndown al finalizar el segundo Sprint

No.	Historia de usuario	Categoría	Estimación inicial en puntos.	Finalización	Puntos restantes
3	Administración de Generadores (GPU's.)	Monitor	6	Día 8	142
4	Administración de tanques.	Monitor	4	Día 11	138
7	Administración de la Configuración de Memoria.	Monitor	4	Día 15	134
13	Administración de la planificación de mantenimiento de generadores.	Monitor	6	Día 17	128
8	Alarmas de aviso de eventos importantes.	Monitor	6	Día 20	122

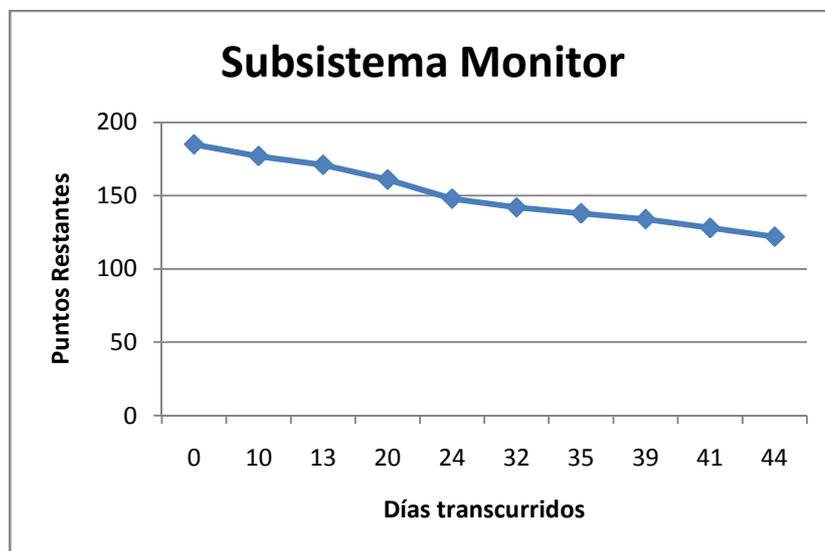


Gráfico 4.20: Avance del Proyecto

Como se puede apreciar este sprint se ha completado en un tiempo menor al estimado porque existió un incremento en la velocidad del desarrollo, esto se debe fundamentalmente a tres aspectos:

- Se incrementó el factor de dedicación del equipo Scrum
- Se trabajó sobre una arquitectura definida.
- No se desperdició tiempo en el diseño de formularios, tablas ya que en el primer sprint se definió un estilo para las páginas.

4.3.6.8.- Revisión.

En la revisión con el dueño del producto supo mencionar un cambio importante para la historia de usuario de “Mantenimiento de generadores”, solicita aumentar un campo que indique el estado del mantenimiento, este campo debe permitir al usuario elegir las siguientes opciones:

- Pendiente.
- Terminado.
- En curso.
- Cancelado.

4.3.7.- Tercer Sprint.

4.3.7.1.- Planificación.

El objetivo de este sprint es culminar el proyecto desarrollando las partes más complejas del sistema, las mismas que son la generación de gráficos de estado de los equipos. Antes de seleccionar las historias de usuario a desarrollar se debe aumentar una “Agregar un campo que el estado del Mantenimiento de

Equipos”. La tabla a continuación muestra la pila del producto después del primer sprint.

Tabla 4.16 Pila del Producto después del segundo sprint.

No	Historia de usuario	Categoría	Importancia	Estimación inicial en puntos.	Completado
1	Administración de usuarios y grupos de usuarios.	Monitor	5	6	✓
2	Administración de estructura geográfica (Región, Provincia, Ciudad, Central) de los equipos.	Monitor	5	13	✓
3	Administración de Generadores (GPU's).	Monitor	5	6	✓
4	Administración de tanques.	Monitor	5	4	✓
5	Administración de permisos de grupos de usuarios a menús y opciones.	Monitor	8	10	✓
6	Administración de menús y opciones.	Monitor	6	8	✓
7	Administración de la Configuración de Memoria.	Monitor	6	4	✓
8	Alarmas de aviso de eventos importantes.	Monitor	4	6	✓
9	Generar gráficos dinámicos actualizables durante el transcurso del tiempo del estado de generación eléctrica por grupo de generación.	Monitor	6	24	✗
10	Generar gráficos históricos de generación eléctrica con filtros.	Monitor	6	48	✗
11	Generar un gráfico de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.	Monitor	5	20	✗
12	Generar gráficos dinámicos de niveles de agua por tanque actualizables durante el transcurso del tiempo.	Monitor	6	30	✗
13	Administración de la planificación de mantenimiento de generadores.	Monitor	8	6	✓
				185	

Tabla 4.17 Pila del Tercer Sprint, subsistema Monitor

No.	Historia de usuario	Categoría	Importancia	Estimación inicial en puntos.
15	Agregar un campo que el estado del Mantenimiento de Equipos	Monitor	1	1
9	Generar gráficos dinámicos actualizables durante el transcurso del tiempo del estado de generación eléctrica por grupo de generación.	Monitor	6	24
10	Generar gráficos históricos de generación eléctrica con filtros.	Monitor	6	48
11	Generar un gráfico de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.	Monitor	5	20
12	Generar gráficos dinámicos de niveles de agua por tanque actualizables durante el transcurso del tiempo.	Monitor	6	30

La tabla 4.17 es la pila para este sprint como se puede ver las historias de usuario tienen una estimación alta, pero se va a agilizar el desarrollo construyendo en primer lugar un diseño base para las imágenes dinámicas por lo que se espera terminar en un tiempo menor al estimado.

4.3.7.2.- Generar gráficos dinámicos actualizables durante el transcurso del tiempo del estado de generación eléctrica por grupo de generación.

El objetivo de esta funcionalidad es presentar una línea de tiempo en un gráfico bidimensional, se tendrán 2 ejes (x, y) donde que el eje de las x representará la hora de lectura y el eje de las y el valor de la lectura.

Para comenzar esta historia de usuario se creará primero una página independiente de todo el proyecto (grafico_gpu.php) y se incluye la librería Open

Flash Chart al proyecto que permitirá poner gráficos Flash de varios tipos (gráficos de líneas, barras, pasteles, radares, etc.)

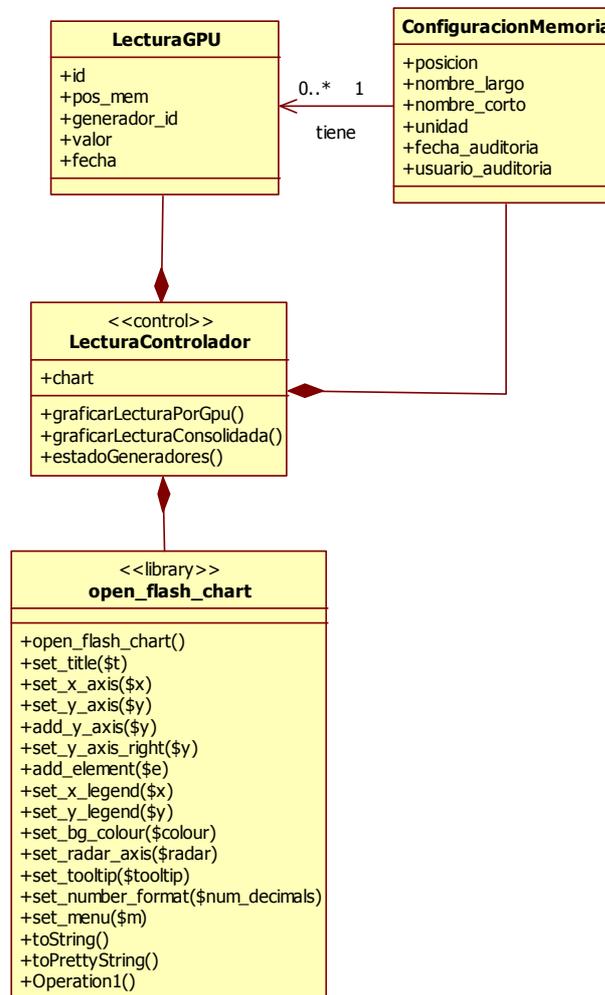


Gráfico 4.21: Diagrama UML para generar gráficos de Generación Eléctrica.

En la clase LecturaControlador se desarrolla con el método graficarLecturaPorGpu() que construye un objeto de la clase Open_flash_chart con los datos según los parámetros enviados:

- GPU,
- Posiciones de memoria,
- Fecha y hora inicial,
- Fecha y hora final,

Se ocupará la clase de persistencia LecturaGPU que es la tabla donde está almacenado las lecturas de los GPU's, también ocupará la clase de ConfiguracionMemoria para consultar los nombres de las posiciones de memoria, el resultado de la función graficarLecturaPorGpu() es un objeto de la clase open_flash_chart.

Una vez que el desarrollo de la clase LecturaControlador haya concluido se llama a la función graficarLecturaPorGpu() desde la página grafico_gpu.php, aquí se debe convertir el objeto de open_flash_chart al formato JSON para con esta información enviar los datos al control de open_flash_chart y que este dibuje los valores en el gráfico.

Los parámetros que se necesitan para llamar a la función graficarLecturaPorGpu() se recibirán mediante la URL de la página que es una técnica para pasar valores entre páginas web más conocido como método GET, también se recibirán otras variables que se utilizarán para el comportamiento del gráfico, a continuación se detalla la lista de variables que recibirá la página.

- GPU (gpu): Id del GPU.
- Posiciones de memoria (pos): lista de posiciones de memoria que se van a graficar separadas por punto y comas (;).
- Fecha y hora inicial (fi): hora de inicio a graficar.
- Fecha y hora final (ff): hora fin a graficar,
- Número de horas (for_ff): En el caso de querer graficar de la hora actual a un número de horas atrás se debe ingresar un número en formato: -3 hour, este parámetro reemplaza a los parámetros de fecha y hora inicial y fecha y hora final.

- Tiempo de refrescamiento (tr): es un número en milisegundos para refrescar la página y se actualice el gráfico.
- Alto del cuadro del gráfico (h).
- Ancho del cuadro del gráfico (w).

Por ejemplo para graficar las lecturas de las tres corrientes del GPU 1 de las últimas 3 horas, la URL que se debe llamar será:

http://localhost/monitor/grafico_gpu.php?gpu=1&pos=40001;40002;40003&for_ff=-3%20hour&tr=100&h=150

Además se implementará un botón para mostrar el gráfico mostrado en un formato que permita la impresión del mismo, para ello se utiliza la función OFC.jquery.popup de Open Flash Chart, lo que hace es convertir el gráfico que está en Flash a un formato de imagen el mismo que se puede imprimir, se muestra en una nueva página web.



Imagen 4.30 Muestra de niveles de las tres corrientes de un GPU.

Luego que esté desarrollada esta página web se debe integrarla a la página general, en el sistema Monitor se crea el menú Estado de Generadores, y cuatro opciones:

- Grupo Uno (GPU 1)
- Grupo Dos (GPU 2)
- Grupo Tres (GPU 3)
- Grupo Cuatro (GPU 4)

En cada opción se va a mostrar tres gráficos, uno para las corrientes, el segundo para los voltajes, y el tercero para las potencias. Para incrustar estos gráficos en la página de la opción se va a usar IFrames que es una etiqueta especial de HTML que permite incluir una página web dentro de otra página web, por ejemplo para mostrar el gráfico de lecturas de corrientes del GPU 1 el código HTML es el siguiente.

```
<iframe src = grafico_gpu.php?gpu=1&pos=40001;40002;40003&for_ff=-  
3%20hour&tr=100&h=150" align="left" width="100%" height="200" >
```

Por último se agrega un link que abre cada gráfico en una página nueva, la URL será la misma que se muestra en el IFrame, el cambio es que estará atado a un link que mediante una función JavaScript abrirá el mismo gráfico en una nueva página.

La función JavaScript es la siguiente:

```
function abrir(pagina, nombre, width, height)
{
    var w = window.open(pagina, nombre, "width="+width+",
height="+height+", screenX=0,screenY=0, top=0, left=0, status=yes ,
resizable=yes, scrollbars=yes" );
    w.focus();
}
```

Y el código del link el siguiente:

```
<a href = "JavaScript:abrir(
'grafico_gpu.php?gpu=1&pos=40001;40002;40003&for_ff=-
3%20hour&tr=100&h=150', 'G2', '1000', '200')" > Abrir en otra página </a>
```



Imagen 4.30 Estado del Grupo Uno (GPU 1) desde la página principal.

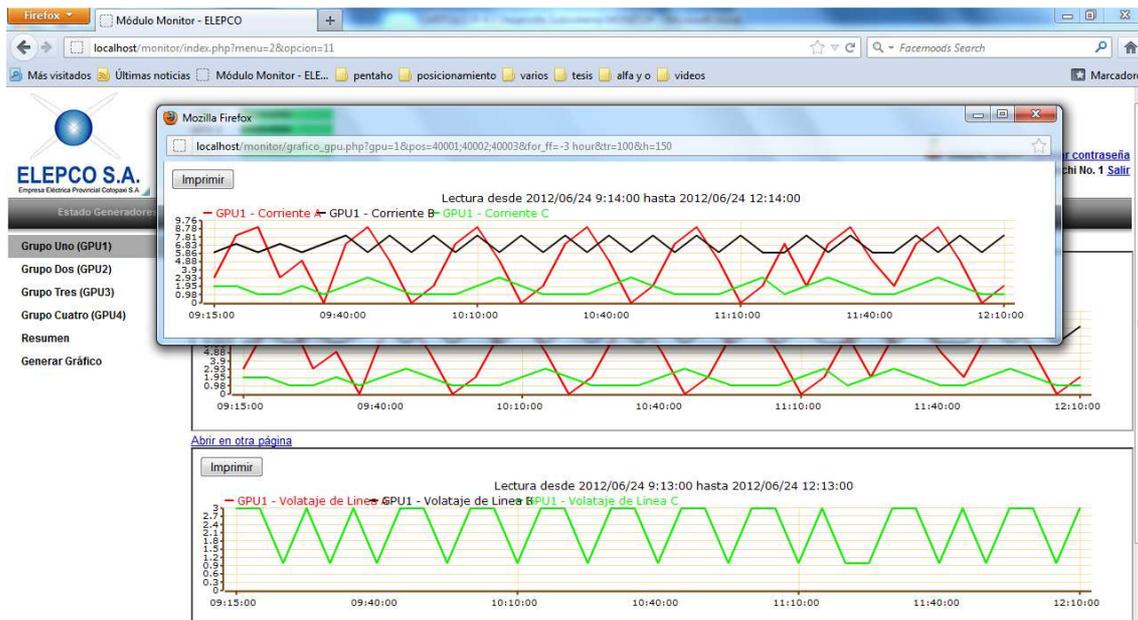


Imagen 4.31 Estado del Grupo Uno (GPU 1), link “Abrir en otra página”.

4.3.7.3.- Generar gráficos históricos de generación eléctrica con filtros.

El objetivo de esta función es tener un formulario de búsqueda con varios filtros entre los que están: la central de generación, el generador, el valor (corriente, voltaje, potencia), la fecha inicial, fecha final, el tiempo de recarga, el alto y ancho del gráfico. Para el desarrollo de esta historia se utilizará la página ya realizada en el punto anterior grafica_gpu.php.

Se crea una opción con nombre “Generar Gráfico” perteneciente al menú “Estado Generadores”, se crea la vista “generar_grafico.php” y dentro de esta página estará el código para graficar el formulario y mediante funciones JavaScript se incrustará el gráfico en un IFrame como se lo hizo en el punto anterior

El modelo del formulario desarrollado queda según la siguiente captura de pantalla.



Imagen 4.32 Formulario de búsqueda con varios filtros.

El botón generar del formulario, lo que hace es llamar a una función JavaScript que arma una URL para llamar a la página graficar_gpu.php para ello incluye los filtros especificados por el usuario.

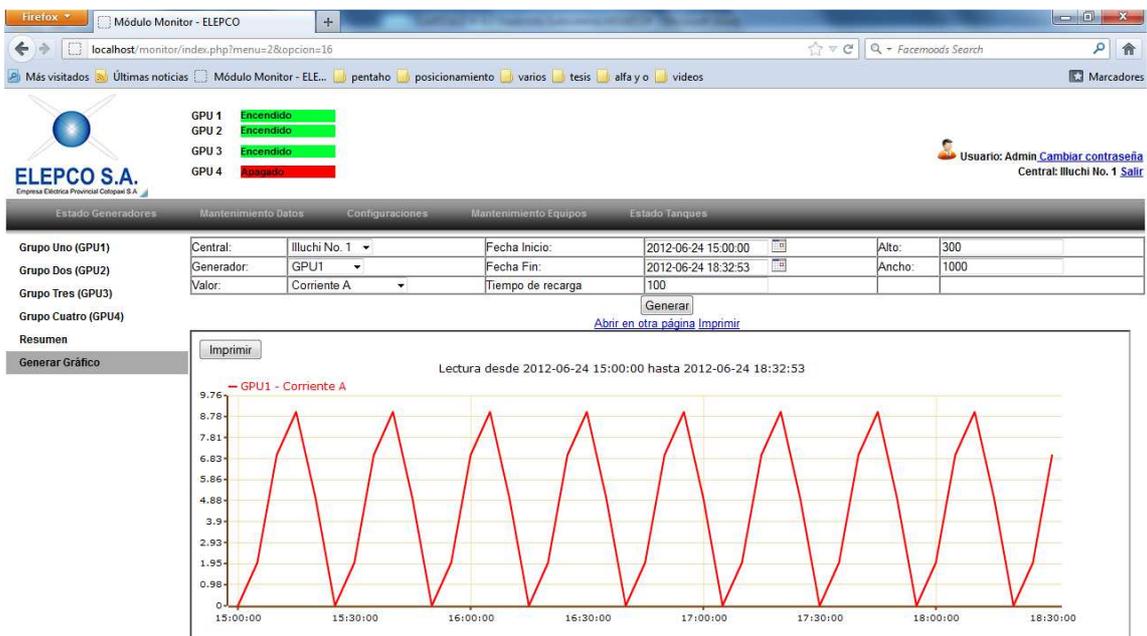


Imagen 4.33 Resultado del gráfico.

4.3.7.4.- Generar un gráfico de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.

El valor de las potencias indica el total de energía eléctrica generada por el GPU, en total la suma de estas variables sería el total generado por la compañía, para esta función se va a ocupar la página grafica_gpu.php pero se debe realizar un cambio en su funcionalidad, para que además de las líneas de potencias cada GPU a mostrar se grafique una adicional que sea la suma total de las potencias generadas, para ello se aumenta una variable opcional para ser pasada mediante la URL o método GET, cuando esta variable esta con valor "s" entonces se sumaría y la variable de GPU ya no sería necesaria puesto que se sobrentendería que se debe tomar en cuenta todos los generadores de la central hidroeléctrica.

Además se crea una nueva función en la clase "LecturaControlador" para que realice esta operación devolviendo ya un objeto de la clase OpenFlashChart, lista para ser graficada.

Se crea una opción con nombre "Resumen" y la vista "resumen.php", sobre esta vista al igual que las opciones de Estado de los GPU's se incrustará el gráfico en un IFrame, el gráfico tendrá las potencias de cada GPU y una línea más indicando la suma total.

La URL que se utilizará para graficar el total de las potencias es:

http://localhost/monitor/grafico_gpu.php?pos=40010&for_ff=-3%20hour&h=150&tr=100&gr=s

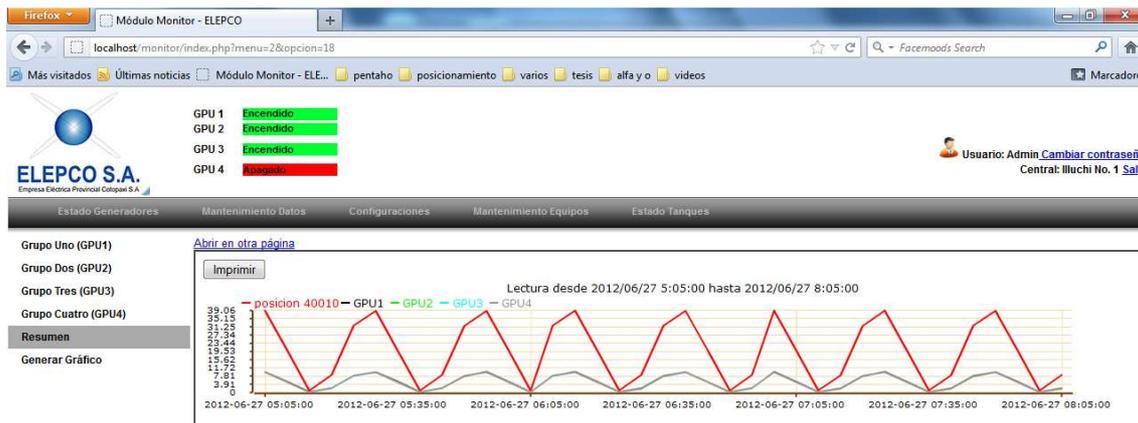


Imagen 4.33 Resumen de potencias.

Como se puede apreciar en la captura de pantalla las opciones de “Abrir en otra página” e “Imprimir” se mantienen como estándar de diseño en la página.

4.3.7.5.- Generar gráficos dinámicos de niveles de agua por tanque actualizables durante el transcurso del tiempo.

Esta opción tiene como objetivo mostrar un gráfico indicando el nivel de agua de los tanques, se debe tener en cuenta que el subsistema de Lectura de Datos almacena estos valores en metros y representa la distancia que existe entre la superficie del tanque y la superficie de agua, por esta razón en la tabla de Tanque se incluyó una columna con nombre Capacidad que indica la profundidad total del tanque, un valor máximo y una columna que indica el valor mínimo que debe tener el tanque para cerrar la compuerta de paso de agua, en el gráfico se deberá mostrar el nivel total de agua para esto se debe restar la capacidad total del tanque menos el valor guardado por el sub sistema de Lectura.

Se generará un gráfico mediante Open Flash Chart, se utilizará un gráfico de barras que se asemeja a la forma de un tanque, se colocará un diseño que

muestre un tanque a la vez, con esto se puede tener el monitoreo separado por cada tanque pero la opción que se creará en el sistema se podrán ver el estado de los dos tanques.

El siguiente diagrama muestra las clases a desarrollar en este apartado.

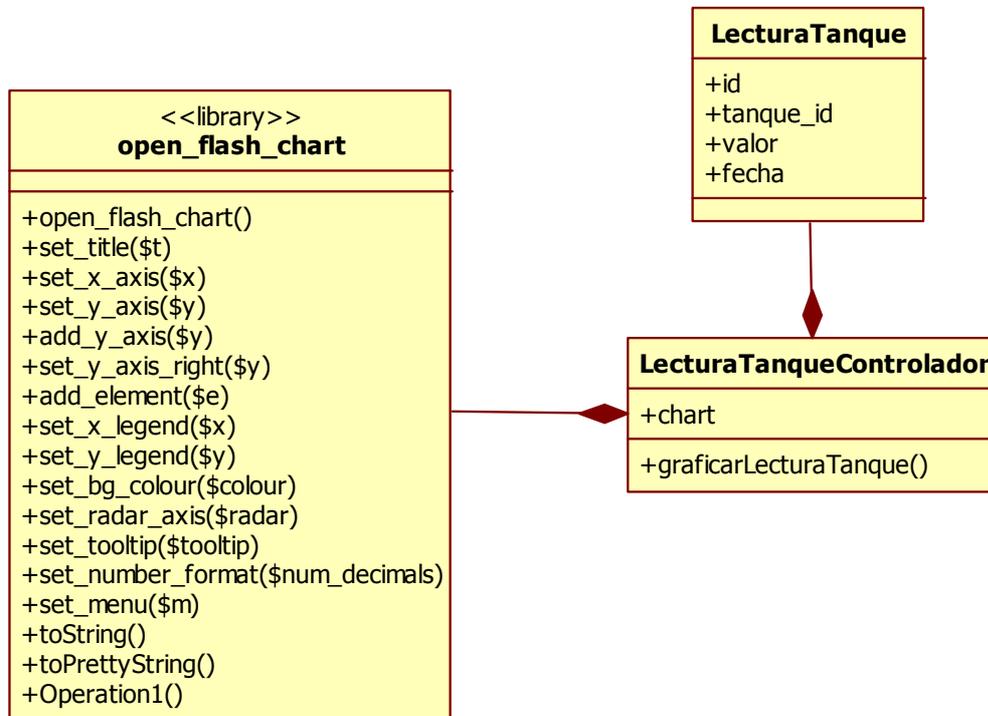


Gráfico 4.22: Diagrama UML para generar gráficas de niveles de agua.

Se crea la clase LecturaTanqueControlador que ocupa las clases de Open_flash_chart y LecturaTanque que ya están creadas, y se desarrolla la función graficarLecturaTanque, este método tomará como parámetro el id del tanque y devolverá un objeto de open_flash_chart con el valor de la última lectura del tanque. En esta función hay que tomar en cuenta un requerimiento no funcional realizado por el usuario, si el valor del nivel de agua se encuentra por debajo del valor mínimo especificado en el tanque entonces se debe mostrar el gráfico de color Rojo caso contrario si el valor es mayor que el valor mínimo y

menor que el valor máximo entonces se debe poner del color Naranja, y por último si es mayor que el valor máximo se debe mostrar de color Verde

Una vez que esté desarrollada la clase `LecturaTanqueControlador`, se procede a crear la página `grafico_tanque.php`, aquí se llamará a la función `graficarLecturaTanque` y se graficará el chart devuelto por la función, esta página recibirá como parámetros el id del tanque el alto y el ancho del recuadro a graficar los mismos que serán ingresados en la URL de la página.

El gráfico queda de la siguiente forma.

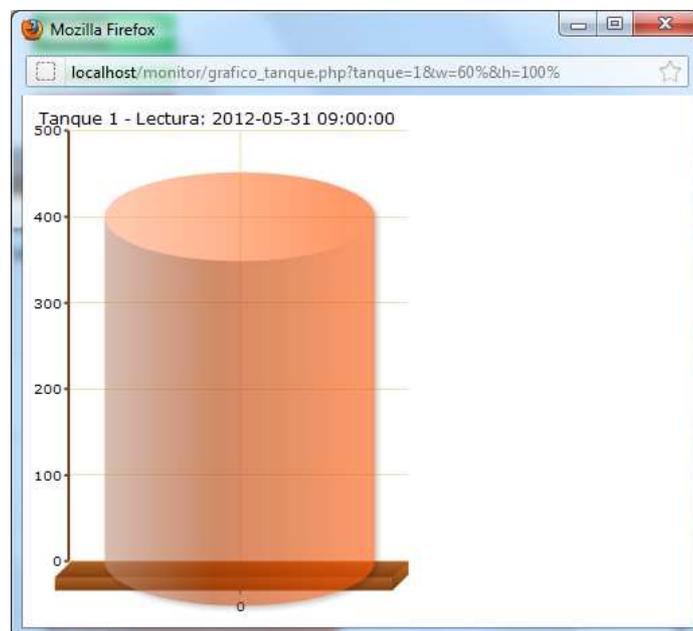


Imagen 4.34 Chart devuelto por la función,

Se crea un nuevo menú en el sistema con nombre “Estado Tanques” y una opción con nombre “Nivel de agua” atada a la vista “estado tanques.php” en esta opción se mostrarán el nivel de agua de los dos tanques y el usuario también tiene la posibilidad de ver cada gráfico en una página nueva como fue uno de los requerimientos no funcionales.



Imagen 4.35 Gráfico final de representación de tanques.

4.3.7.6.- Carta de Burndown

Tabla 4.18 Carta de Burndown del Último sprint.

No	Historia de usuario	Categoría	Estimación inicial en puntos.	Finalización	Puntos Restantes
9	Generar gráficos dinámicos actualizables durante el transcurso del tiempo del estado de generación eléctrica por grupo de generación.	Monitor	24	Día 15	98
10	Generar gráficos históricos de generación eléctrica con filtros.	Monitor	48	Día 20	50
11	Generar un gráfico de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.	Monitor	20	Día 22	30
12	Generar gráficos dinámicos de niveles de agua por tanque actualizables durante el transcurso del tiempo.	Monitor	30	Día 30	0

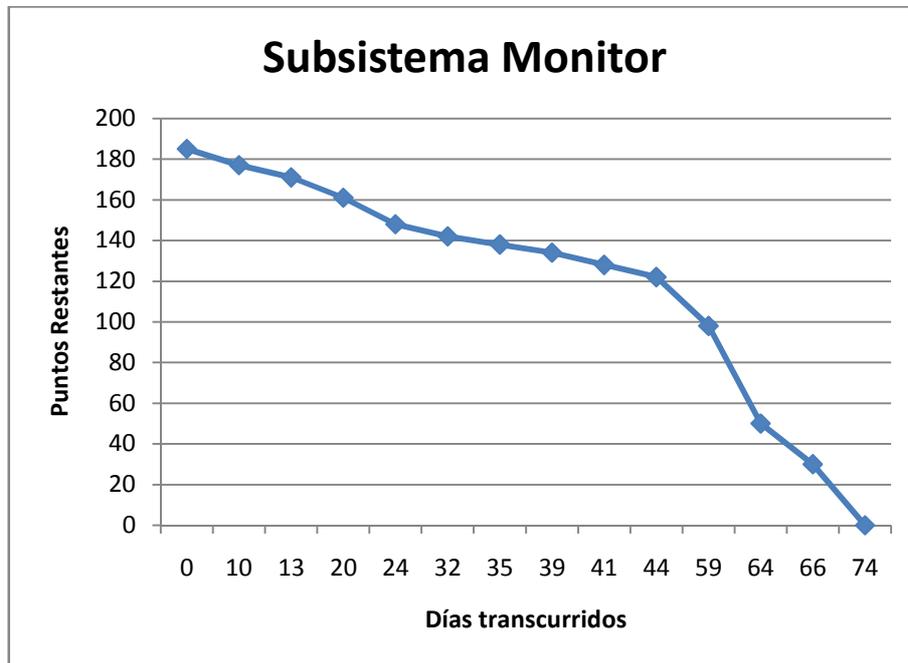


Gráfico 4.23 Carta de Burndown final para el subsistema Monitor

El gráfico anterior es la carta de Burndown final para el subsistema Monitor, se ha concluido el proyecto y se observa que en el último Sprint a pesar de tener las historias de usuario con mayor estimación de puntos se logró disminuir el tiempo de desarrollo esto se debe a que se utilizó librerías de software que fueron de ayuda para el proyecto, la creación de componentes reutilizables para las opciones y la constante comunicación con el dueño del producto.

4.3.7.7.- Revisión.

El dueño del producto muestra conformidad con el producto obtenido, y se da por finalizado el desarrollo del subsistema Monitor.

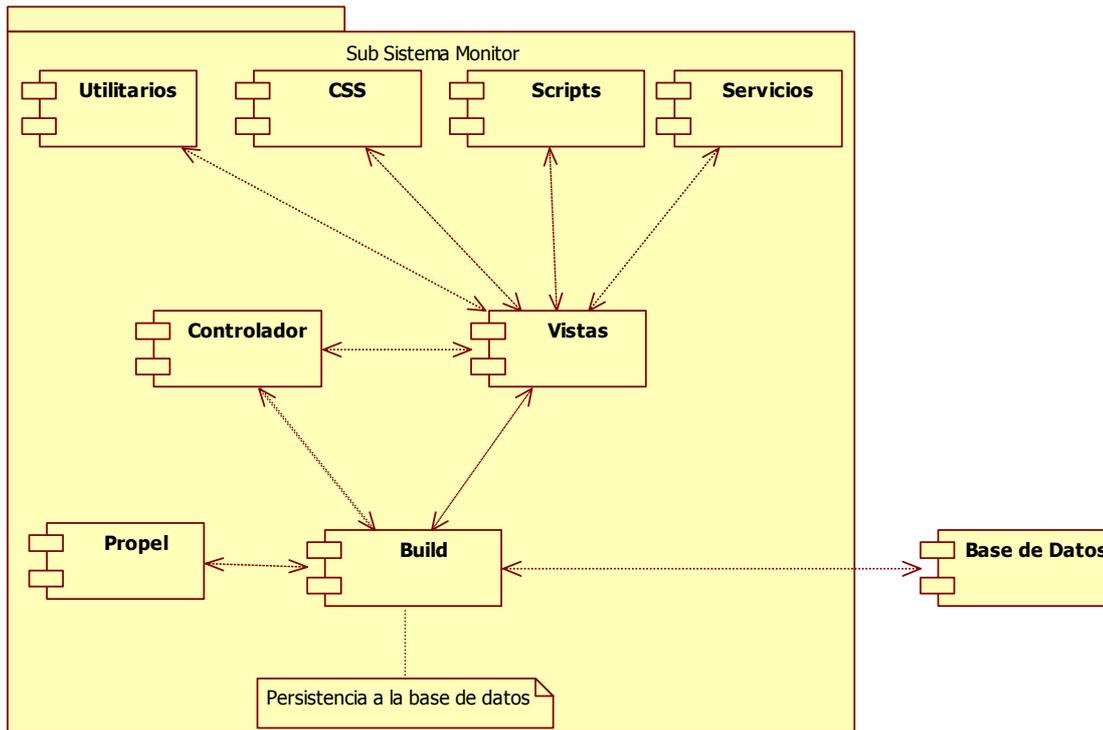


Gráfico 4.24 Diagrama de implementación del subsistema monitor

4.4.- Implementación del subsistema de Análisis de Datos.

4.4.1.- Visión general.

El objetivo de este capítulo es implementar la suite de inteligencia de negocios Pentaho y desarrollar varios reportes según los requerimientos de ELEPCO.

4.4.2.- Análisis de requerimientos.

Los siguientes requerimientos fueron tomados del Anexo A (Especificación de Requerimientos del Sistema) y corresponden a los reportes que se deben desarrollar.

4.4.2.1.- Requerimientos Funcionales.

- Curva de potencia generada hora a hora por cada día de cada grupo (cuatro en total) en kw.
- Curva de la energía generada en kw/h.
- Curvas de variación de voltaje de barras.
- Curvas de variación de corriente en función del tiempo de cada grupo.

4.4.2.2.- Análisis.

Para la implementación de este subsistema además de los requerimientos del cliente (reportes) se debe tomar en cuenta las tareas que son parte de la implementación, puesto que son varios pasos para instalar y configurar la aplicación, a continuación se listan estas tareas:

- Desarrollo de la base de datos para el cubo.
- Instalación y configuración inicial de la Suite Pentaho.
- Creación de los procesos ETL para migración de datos.
- Creación de Metadatos.
- Creación del Cubo

Para armar la pila del producto se agregarán los requerimientos funcionales tomados del Anexo A, a demás las tareas de implementación, los requerimientos funcionales se resume en una sola historia de usuario.

Tabla 4.19 Pila del Producto, subsistema de Análisis de Datos

No	Historia de usuario	Categoría	Importancia	Estimación inicial en puntos.
1	Desarrollo de la base de datos para el cubo.	Análisis de datos	5	6
2	Instalación y configuración de la Suite Pentaho.	Análisis de datos	10	24
3	Creación de un proceso de ETL para la carga de datos.	Análisis de datos	6	24
4	Creación de Metadatos.	Análisis de datos	5	12
5	Creación del cubo.	Análisis de datos	5	12
6	Desarrollo de reportes.	Análisis de datos	10	24
				102

4.4.2.3.- Duración del Sprint

Para este subsistema se necesitarán de dos sprints de 15 días cada uno, las tareas de implementación y configuración requieren mayor análisis en su construcción por esta razón se alarga los tiempos.

4.4.2.4.- Primer Sprint

Planificación

El objetivo de este sprint es desarrollar la base de datos para el cubo y un proceso de migración de datos y comenzar con la implementación y configuración de la plataforma Pentaho.

Tabla 4.20 Pila del primer sprint, subsistema de Análisis de Datos.

No.	Historia de usuario	Categoría	Importancia	Estimación inicial en puntos.
1	Desarrollo de la base de datos para el cubo.	Análisis de datos	5	6
2	Instalación y configuración de la Suite Pentaho.	Análisis de datos	10	24
3	Creación de un proceso de ETL para la carga de datos.	Análisis de datos	6	24

Desarrollo de la base de datos para el cubo.

Para comenzar con la implementación de la Suite se desarrollará la base de datos para el cubo, el diseño de esta base debe partir desde la base de datos del “central” para a partir de sus tablas crear otro diseño OLAP en estrella según la siguiente estructura de jerarquías necesarias para construir los reportes solicitados:

- Dimensiones
 - o Central hidroeléctrica.
 - Nombre central
 - o Generador
 - Nombre Generador
 - o Geográfica
 - Región
 - Provincia
 - Ciudad
 - o Variable
 - Nombre Variable

- Tiempo
 - Año
 - Mes
 - Día
 - Hora
- Métricas:
 - Valor
 - Valor Potencia

Cada dimensión de la estructura será una tabla y los atributos de las jerarquías serán columnas de las tablas dimensión, cada tabla dimensión tiene un campo código, las tablas dimensión están relacionadas con este código con la tabla de hechos fact_generacion.

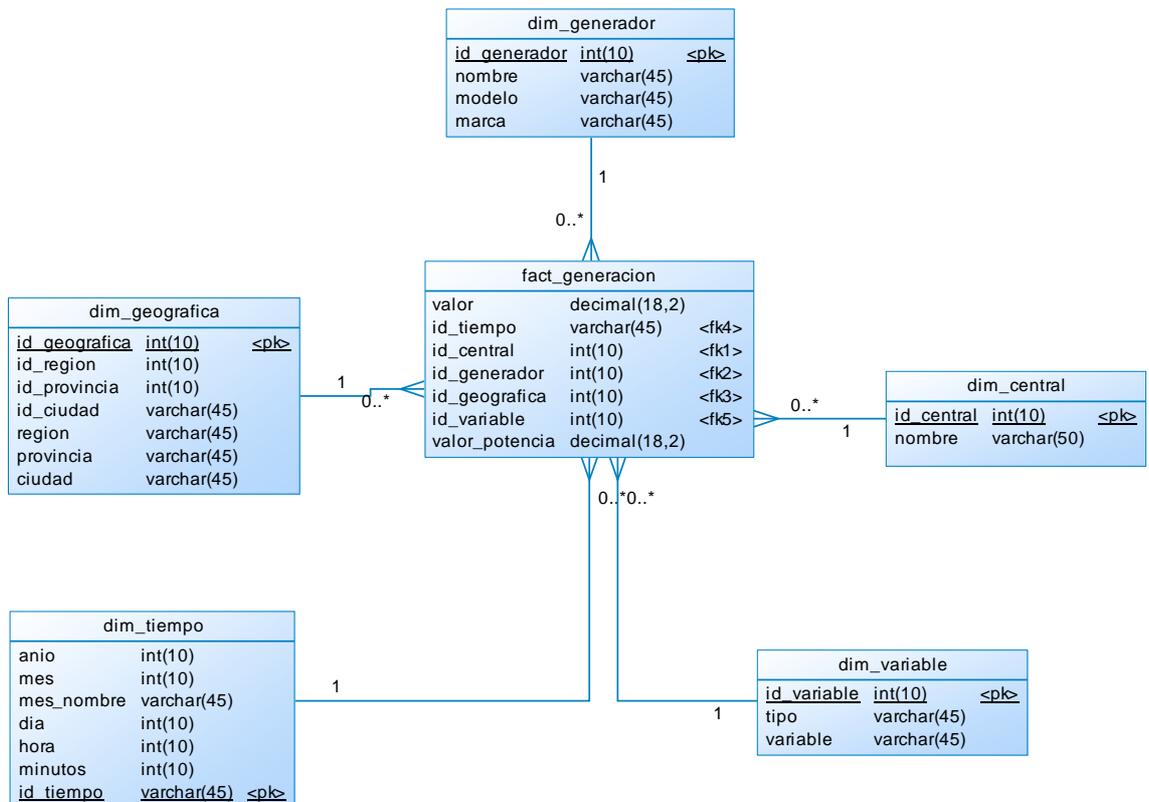


Gráfico 4.25 Diagrama de Entidad Relación de la base de datos “cubo_central”

Instalación y configuración de la Suite Pentaho.

Como se observó en el punto 3.3.1 Pentaho es un conjunto de herramientas que satisfacen los requisitos de BI; para obtener los módulos se procede a descargar los programas de la página web oficial de Pentaho, cada módulo viene por separado y están comprimidos en formato zip, se las extrae en una carpeta en el disco C:\ con nombre Pentaho, y será ahí donde se implementará todo el subsistema quedando de la siguiente Forma:

Tabla 4.21 Rutas de extracción de herramientas

RUTA	DESCRIPCIÓN
C:\Pentaho	Raíz de la implementación.
C:\Pentaho\administration-console	Módulo de administración y configuración, se encarga de gestionar usuarios, conexiones a bases de datos y gestiona trabajos programados para actualizar configuraciones
C:\pentaho\Biserver-ce	Más conocido como Pentaho Análisis, es el servidor OLAP, también tiene toda la aplicación de usuario (Consola de Usuario) y los repositorios de archivos (Reportes, metadatos, cubos).
C:\Pentaho\Data-integration	Módulo ETL de Pentaho.
C:\Pentaho\ metadata-editor	Módulo de Data Modeling de Pentaho
C:\Pentaho\ report-designer	Módulo Reporteador
C:\Pentaho\schema-workbench	También conocido como Mondrian, aquí se generan los esquemas o definiciones del cubo.

Los módulos de administration-console y bi-server deben ser iniciados para poder comenzar a usar la plataforma, cada programa tiene dos archivos batch que levantan y detienen los servicios, cada servicio ocupa un número de puerto diferente a continuación se detalla esta información.

Tabla 4.22 Descripción de los módulos de administration-console y bi-server

ARCHIVO	MÓDULO	ACCIÓN	PUERTO A USAR
C:\Pentaho\administration-console\start-pac.bat	Administration-console	Inicia el servicio de administración.	8099
C:\Pentaho\administration-console\stop-pac.bat	Administration-console	Detiene el servicio de administración.	
C:\Pentaho\biserver-ce\start-pentaho.bat	Biserver-ce	Inicia el servicio de la consola de usuario.	8080
C:\Pentaho\biserver-ce\stop-pentaho.bat	Biserver-ce	Detiene el servicio de la consola de usuario.	

Se abre el módulo administration-console y se incluye una nueva conexión de a la base de datos “cubo_central”:

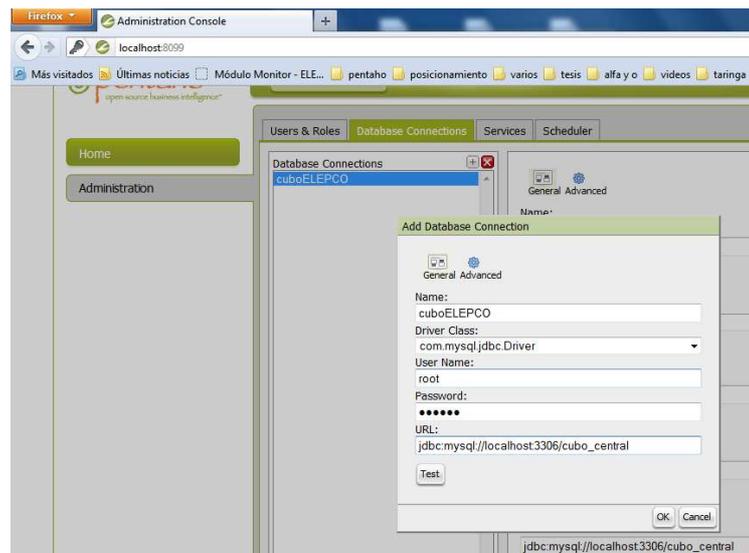


Imagen 4.36 Nueva conexión de a la base de datos “cubo_central”.

En cuanto a las seguridades, se crean los usuarios que ha notificado el dueño del producto, entre ellos están:

- Gerente de generación.
- Jefe de generación.
- Gerente de comercialización.

Pentaho Bi Server tiene un archivo de configuración de la clave maestra para publicaciones, esta clave servirá para que los otros módulos puedan publicar sus productos (esquema de cubos, archivo de metadatos, reportes, etc.) al servidor, este archivo se encuentra en la siguiente dirección:

C:\pentaho\biserver-ce\pentaho-solutions\system\publisher_config.xml

La clave por defecto es “password”, pero se la cambia a una contraseña más segura, para esto se debe editar el archivo y remplazar la contraseña, como lo muestra el siguiente ejemplo:

```
<publisher-config>  
    <publisher-password>micontraseña</publisher-password>  
</publisher-config>
```

Pentaho maneja soluciones que son espacios de trabajo donde se deben colocar los reportes predefinidos, las estructuras de los cubos y los metadatos, la ruta que Pentaho tiene destinado para estos espacios de trabajo está en la siguiente ruta:

C:\pentaho\biserver-ce\pentaho-solutions

Dentro de esta ruta se crea un nuevo directorio con el nombre ELEPCO y será aquí donde se almacenen todos los archivos necesarios para la

implementación, una vez realizadas estas configuraciones la consola de usuario se verá de la siguiente forma:



Imagen 4.37 Consola de usuario.

Creación de un proceso de ETL para la carga de datos.

Como se habló anteriormente un proceso ETL es un proceso de lectura, transformación y carga, en Pentaho la herramienta Data Integration es la encargada de proveer esta funcionalidad, aquí se creará una transformación por cada tabla del cubo, y un trabajo que se encargara de llamar a todas las transformaciones.

Esta tarea se divide en varias subtareas que se listan a continuación:

- Crear Transformación para la tabla dim_central.
- Crear Transformación para la tabla dim_generador.
- Crear Transformación para la tabla dim_geografica.

- Crear Transformación para la tabla dim_variable.
- Crear Transformación para la tabla dim_tiempo.
- Crear Transformación para la tabla fact_generacion.
- Crear el trabajo para ejecutar todas las transformaciones.

Crear transformación para la tabla dim central.

El objetivo de esta transformación será copiar la información de las centrales hidroeléctricas en la tabla dim_central.

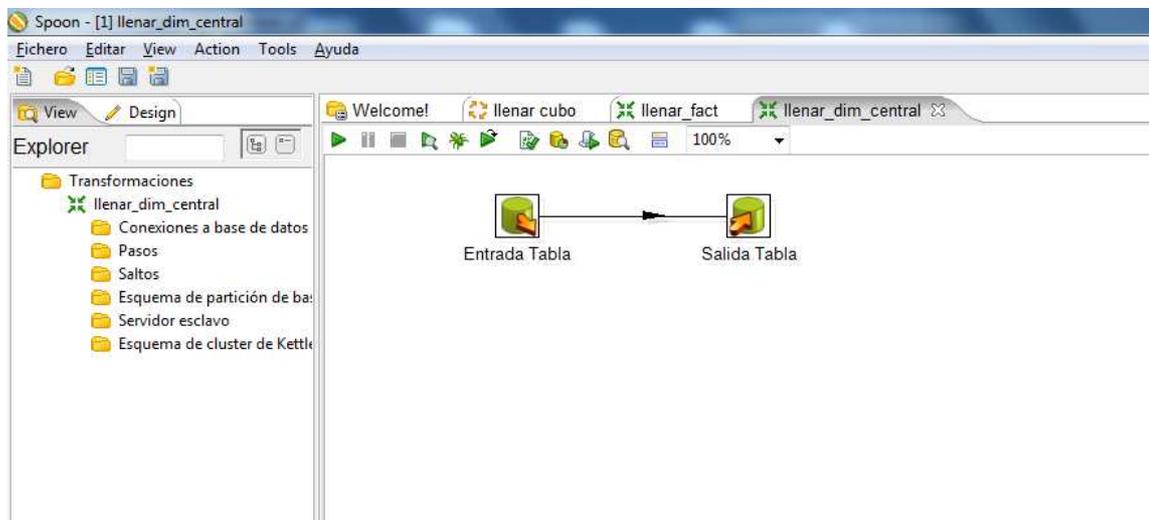


Imagen 4.38 Programación desarrollada para realizar transformación.

Como se puede observar es una transformación donde se cumplen dos pasos:

- Entrada Tabla: Realiza la lectura de datos de la tabla “Central” de la base de datos “Central”.
- Salida Tabla: Inserta la información obtenida por el paso anterior en la tabla “dim_central” de la base de datos “cubo_central”.

Crear Transformación para la tabla dim generador

Esta transformación necesitara dos pasos al igual que la transformación para la tabla dim_central. A continuación se presenta la programación de esta transformación.

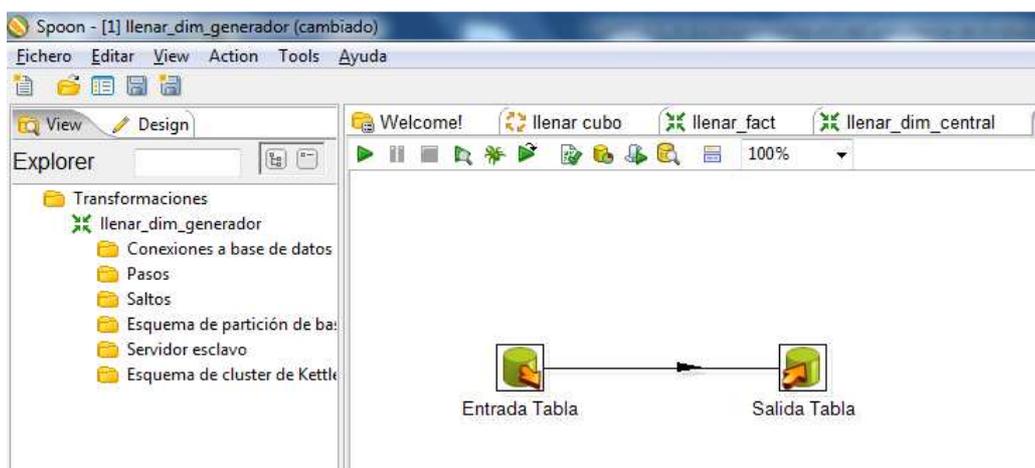


Imagen 4.39 Programación tabla dim_central.

Los pasos que ejecutará son:

- Entrada Tabla: Realiza la lectura de datos de la tabla “Generador” de la base de datos “Central”.
- Salida Tabla: Inserta la información obtenida por el paso anterior en la tabla “dim_generador” de la base de datos “cubo_central”.

Crear Transformación para la tabla dim geográfica.

Esta transformación necesita unir cuatro tablas que contienen la información de las ubicaciones geográficas en la base de datos “central”, la transformación se ejecutará en dos pasos como se ve en la siguiente captura:

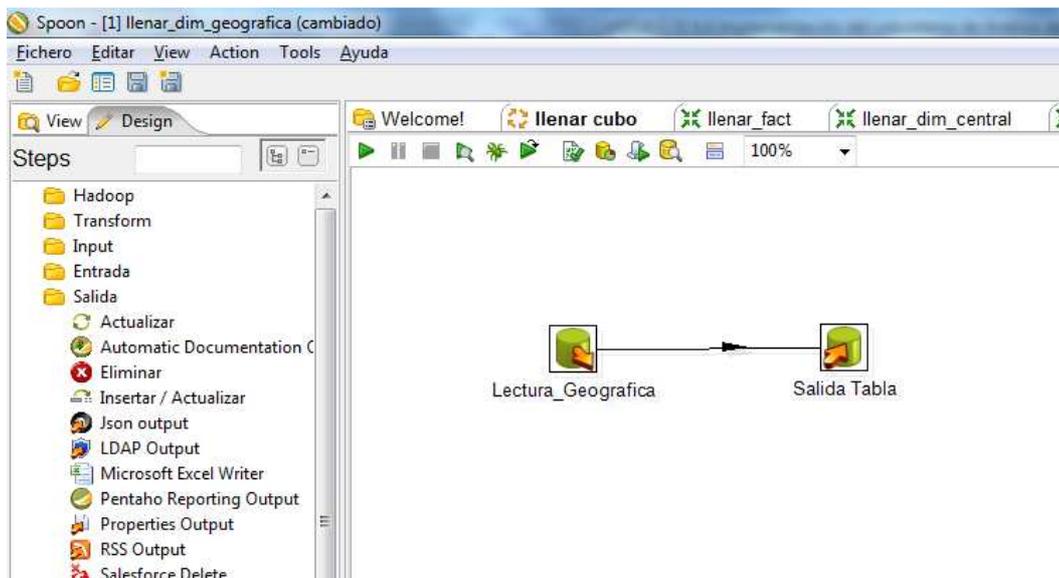


Imagen 4.40 Programación tabla dim_geográfica.

Los pasos que ejecutará son:

- Lectura Geográfica: Ejecuta una consulta SQL que toma los códigos de la región, provincia y ciudad de la tabla “central” y realiza una unión con las tablas “región”, “provincia” y “ciudad” para obtener las descripciones de cada elemento a continuación se muestra la consulta ejecutada por este paso.

```

SELECT distinct r.id, r.nombre, p.id, p.nombre, ci.id, ci.nombre
FROM central c
inner join region r on c.region_id = r.id
inner join provincia p on c.provincia_id = p.id
inner join ciudad ci on c.ciudad_id = ci.id
where ci.id not in (select id_ciudad from cubo_central.dim_geografica);

```

- Salida Tabla: Inserta la información obtenida por el paso anterior en la tabla “dim_geografica” de la base de datos “cubo_central”.

Crear Transformación para la tabla dim variable.

Esta transformación necesitara dos pasos. A continuación se presenta la programación de esta transformación.



Imagen 4.41 Programación tabla dim_variable.

Los pasos que ejecutará son:

- Entrada Tabla: Realiza la lectura de datos de la tabla “configuración_memoria” de la base de datos “Central”.
- Salida Tabla: Inserta la información obtenida por el paso anterior en la tabla “dim_variable” de la base de datos “cubo_central”.

Crear Transformación para la tabla dim tiempo.

La dimensión de tiempo está relacionada con las fechas de lectura de los GPU, se realiza un lectura de cada GPU cada cinco minutos, pero si se representara todas las lecturas de los GPU en el cubo crecería rápidamente y le

quitaría eficiencia al servidor y esta dimensión sería muy complicada de usar para el análisis, por esta razón se va a tomar una lectura de cada hora, es decir; que se requiere generar en la tabla “dim_tiempo” con una serie de fechas ascendentes a partir de una fecha dada con una hora de diferencia entre ellas, la estructura de la tabla dim_tiempo contiene los siguientes campos:

- Id_tiempo
- Anio
- Mes
- Dia
- Hora
- Mes_nombre

Todos estos campos se deben generar en la transformación, a continuación se muestra unos registros de ejemplo para esta tabla:

Tabla 4.23 Registros de Ejemplo

id_tiempo	Anio	Mes	mes_nombre	día	hora
2012-06-26 01:00:00	2012	6	Junio	26	1
2012-06-26 02:00:00	2012	6	Junio	26	2
2012-06-26 03:00:00	2012	6	Junio	26	3
2012-06-26 04:00:00	2012	6	Junio	26	4
2012-06-26 05:00:00	2012	6	Junio	26	5
2012-06-26 06:00:00	2012	6	Junio	26	6
2012-06-26 07:00:00	2012	6	Junio	26	7
2012-06-26 08:00:00	2012	6	Junio	26	8
2012-06-26 09:00:00	2012	6	Junio	26	9
2012-06-26 10:00:00	2012	6	Junio	26	10
2012-06-26 11:00:00	2012	6	Junio	26	11
2012-06-26 12:00:00	2012	6	Junio	26	12

El siguiente gráfico corresponde a la programación de la transformación para llenar la tabla dim_tiempo

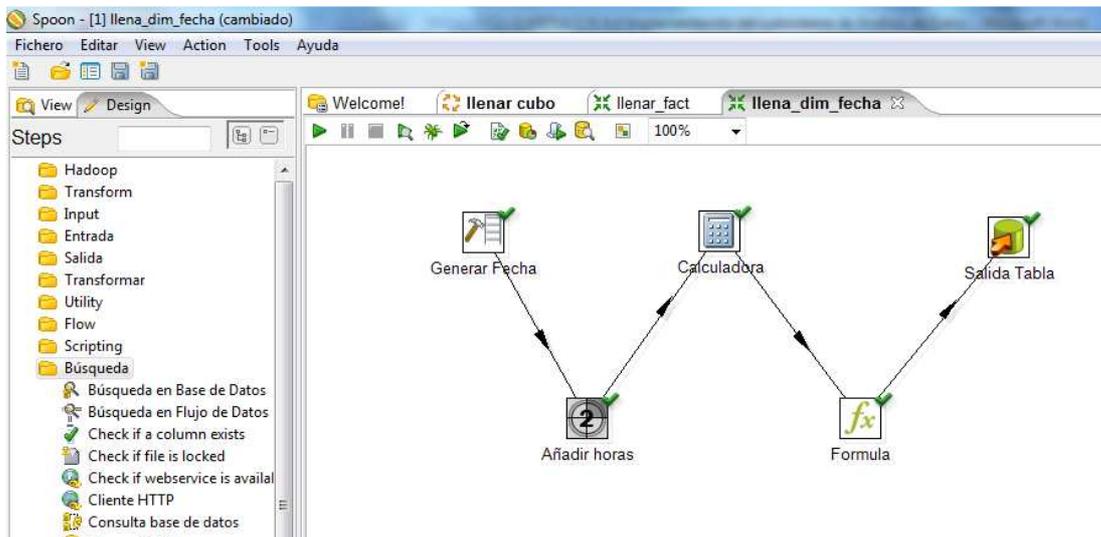


Imagen 4.42 Programación tabla dim_tiempo.

Los pasos que ejecutará son:

- Generar Fecha: Toma una fecha inicial y genera n copias de esta.
- Añadir horas: Crea una matriz de dos columnas donde la primera columna es el conjunto de fechas devueltas por el paso anterior y adjunta otra columna con un valor secuencial.
- Calculadora: Toma la matriz que entrega el paso anterior y suma el valor secuencial a la fecha en horas, de esta manera se ha generado una colección de fechas secuenciales con una hora de separación.
- Fórmula: Aquí se separa la fecha generada en años, meses, días, horas y se anexa el nombre del mes.
- Salida Tabla: Inserta la información obtenida por el paso anterior en la tabla “dim_tiempo” de la base de datos “cubo_central”.

Crear Transformación para la tabla fact generacion.

La tabla Fact_generacion debe contener la información de generación eléctrica (corrientes, voltajes, potencias) por cada hora para evitar sobrecargar la tabla con demasiada información. Para esto se debe unir la tabla de “lectura_gpu” de la base de datos “central” con la información de las dimensiones, a continuación se muestra la programación de la transformación.

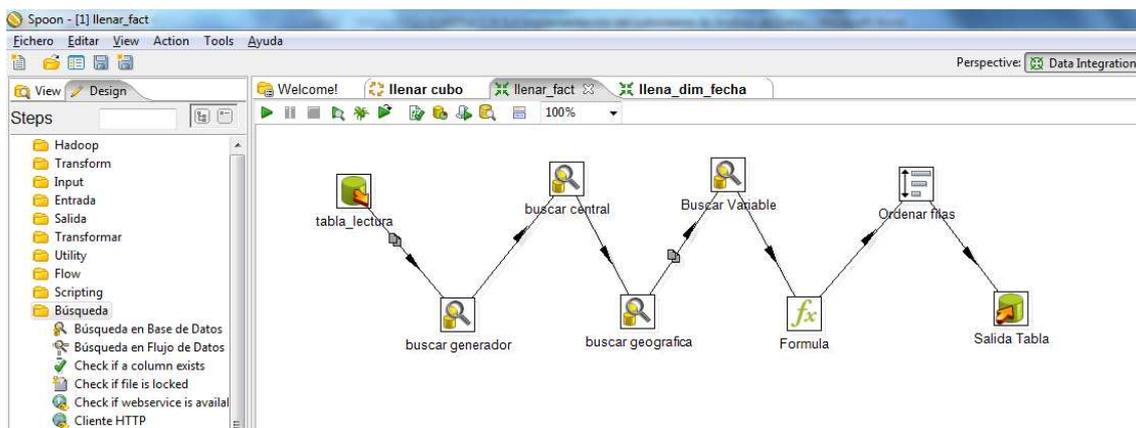


Imagen 4.43 Programación Fact_generacion.

Los pasos que ejecutará son:

- Tabla_lectura: Realiza la lectura de datos de la tabla “lectura_gpu” de la base de datos “Central”.
- Buscar generador: Este paso busca el código del generador de la tabla dim_generador.
- Buscar central: Este paso busca el código de la central de generación de la tabla dim_central.
- Buscar geográfica: Este paso busca el código de la ubicación geográfica de la tabla dim_geografica.

- Buscar variable: Este paso busca el código de la variable de la tabla dim_variable.
- Fórmula: quita los minutos y los segundos de la fecha obtenida de la tabla lectura_gpu para que se puedan relacionar con la dimensión fecha.
- Ordenar filas: ordena todos los registros obtenidos por la fecha de lectura.
- Salida Tabla: Inserta la información obtenida por el paso anterior en la tabla “fact_generacion” de la base de datos “cubo_central”.

Crear el trabajo para ejecutar todas las transformaciones.

Una vez que se tienen las transformaciones para cada tabla del cubo, se procede a desarrollar un trabajo que las integrará, el gráfico a continuación muestra la programación de este trabajo:

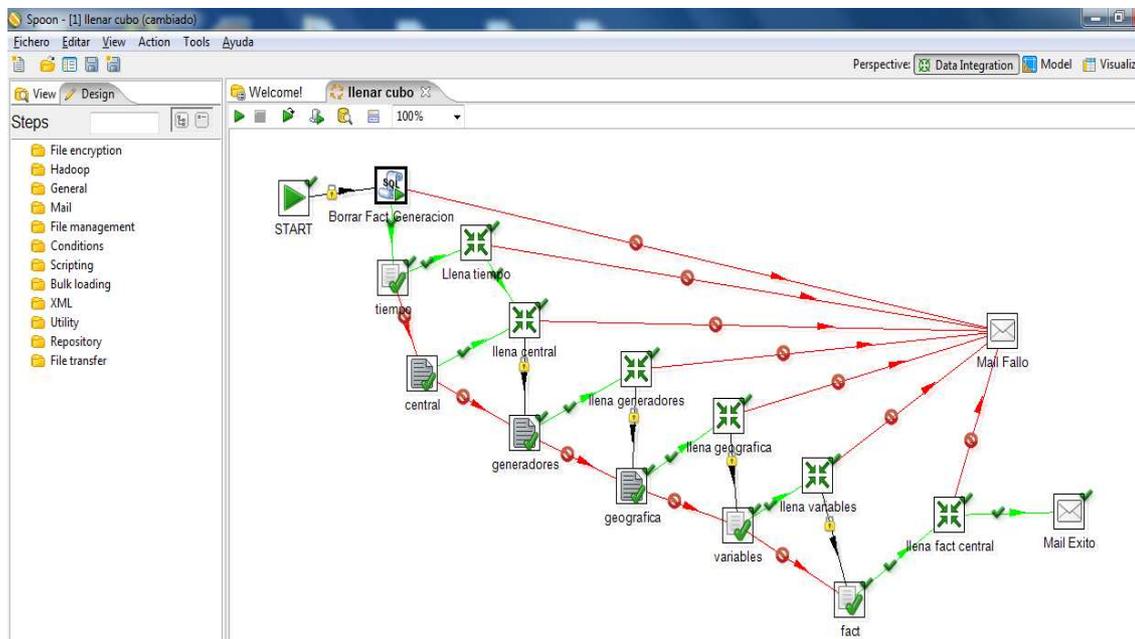


Imagen 4.44 Trabajo llenar cubo.

Como se ve en el diagrama, el primer paso que se ejecuta es borrar la tabla Fact_generacion para que a continuación se puedan modificar las dimensiones, cada transformación vacía las tablas de las dimensiones y las vuelve a cargar por eso se debe vaciar la tabla fact_generación para que no exista error de integridad con las dimensiones que se actualizarán. Luego de borrar la tabla fact_generacion, se procede a revisar si existen los archivos de las transformaciones, si existen se ejecutan las transformaciones caso contrario se continua con la siguiente transformación, si alguna transformación no se ejecutó con éxito se envía un correo electrónico indicando el fallo del mismo, caso contrario si se ejecutaron todas las transformaciones sin problemas se envía un correo electrónico al dueño del producto indicando el éxito del proceso.

Por último se generará una tarea programada en Windows para que ejecute automáticamente esta transformación, para esto Data Integration tiene un módulo llamado Kitchen que mediante una línea de comando desde la consola de Windows se puede ejecutar una Tarea, el comando que ejecutará la tarea programada de Windows es:

```
C:\pentaho\data-integration>Kitchen.bat /file:"C:\Pentaho\data-integration  
\Repositorio\llenar cubo.kjb"
```

Carta de Burndown

Una vez finalizado el desarrollo del primer sprint, se completa la pila del sprint actualizando el día de finalización y los puntos restantes y se genera el gráfico del Burndown

Tabla 4.24: Información a utilizarse para elaborar la carta de Burndown

No.	Historia de usuario	Categoría	Estimación inicial en puntos.	Finalización	Puntos Restantes
1	Desarrollo de la base de datos para el cubo.	Análisis de datos	6	Día 4	96
2	Instalación y configuración de la Suite Pentaho.	Análisis de datos	24	Día 8	72
3	Creación de un proceso de ETL para la carga de datos.	Análisis de datos	24	Día 15	48

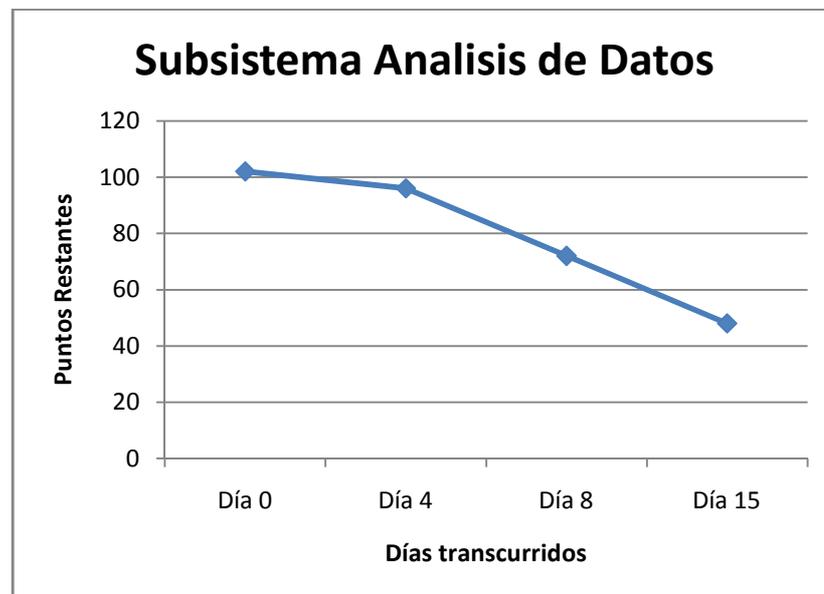


Gráfico 4.26 Carta de Burndown final del primer sprint.

Revisión.

Una vez terminado el desarrollo de las actividades planificadas en este sprint se presenta el avance al dueño del producto. El objetivo es presentar la estructura desarrollada y los usuarios creados para el cubo.

El dueño del producto expresa conformidad con lo desarrollado hasta el momento y se continúa con el siguiente sprint.

4.4.2.5.- Segundo Sprint

Planificación

El objetivo de este sprint es diseñar las estructuras de datos creando la estructura del cubo y de los metadatos que van a ser leídos desde Pentaho y van usados para crear los reportes y por último desarrollar los reportes prediseñados requeridos por ELEPCO S.A.

Tabla 4.25 Pila del segundo sprint, subsistema de Análisis de Datos

No.	Historia de usuario	Categoría	Importancia	Estimación inicial en puntos.
4	Creación de Metadatos.	Análisis de datos	5	12
5	Creación del cubo.	Análisis de datos	5	12
6	Desarrollo de reportes.	Análisis de datos	10	24

Creación de Metadatos.

Como se vio en el Capítulo 3 Pentaho maneja una herramienta de Modelamiento de Datos que lo que hace es crear un archivo de metadatos que viene a representar una abstracción de la base de datos en función de las reglas del negocio, para crear este archivo se usa la herramienta Metadata Editor, este archivo luego se exportará a un repositorio de Pentaho y servirá tanto para realizar reportes Ad-Hoc en la plataforma de Pentaho como para crear reportes personalizados con la herramienta Report Designer.

Para crear este archivo se partirá del diseño de la base de datos “cubo_central”.

Metadata Editor representa una vista de datos como Dominio y dentro de este dominio se encuentran las tablas de negocio, sus relaciones con los tipos de uniones entre las tablas (inner join, full outer, left join, right join), estas tablas son representaciones de las tablas de la base de datos, se van a importar todas las tablas de la base de datos “cubo_central” al proyecto y una vez incluidas al programa se editarán los nombres de las tablas y de las columnas para que contengan nombres más fáciles de entender para el usuario, se ocultarán las columnas que no son relevantes como los campos Id de cada tabla, y también se definen las agregaciones por defecto para las dos variables que contiene la tabla de hechos:

- Valor: Es el valor de la lectura de cada variable (corriente, voltaje, potencia), las variables de corriente y voltaje no pueden ser sumadas, por esta razón los totalizadores de estas variables deben ser en promedios.
- Valor potencia: este valor al contrario de las variables de corriente y voltaje si se debe sumar, puesto que representa el valor total de la potencia generada en un lapso de tiempo.
- Una vez que se han realizado las tareas anteriores se procede a declarar a cada tabla como una vista de negocio (Business View) y se da por concluido el modelamiento de datos, la imagen 4.45 muestra el modelamiento obtenido por Pentaho Metadata Editor.

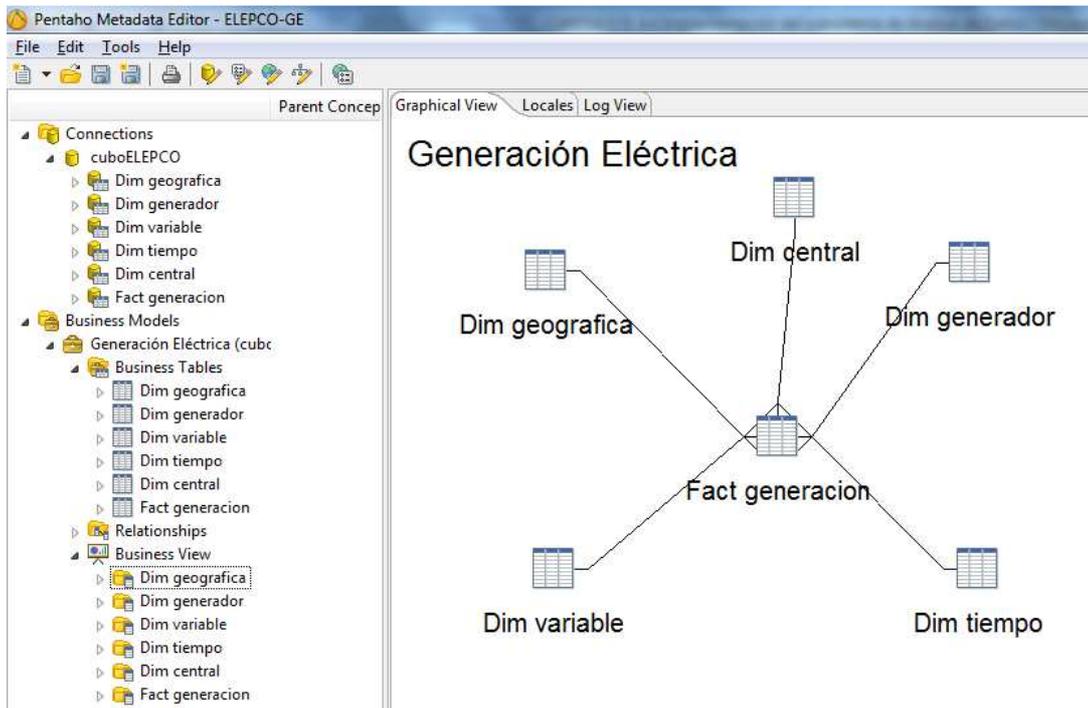


Imagen 4.45 Modelamiento obtenido por Pentaho Metadata Editor.

Una vez que esté listo el diseño de los metadatos de la base de datos “cubo_central” se lo publica en el repositorio de Pentaho.

The screenshot shows the 'Publish To Server' dialog box. It contains the following fields and controls:

- Publish Information:** A section with the instruction: "Enter your server and solution information here to publish your domain to a solution location of your choice on the Pentaho BI Server."
- Filename:** A dropdown menu with 'metadata.xmi' selected.
- Publish Location:** A dropdown menu with 'ELEPCO' selected.
- Web Publish URL:** A dropdown menu with 'http://localhost:8080/pentaho/RepositoryFilePublisher' selected.
- Publish Password:** A text input field containing seven dots (password masked).
- Server Userid:** A text input field containing 'joe'.
- Server Password:** A text input field containing seven dots (password masked).
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

Imagen 4.46 Configuración usada para la publicación.

Y luego de haber publicado la estructura de metadatos en el servidor se puede observar que Pentaho ya muestra esta estructura para poder realizar reportes Ad-hoc como se muestra en la siguiente captura de pantalla:

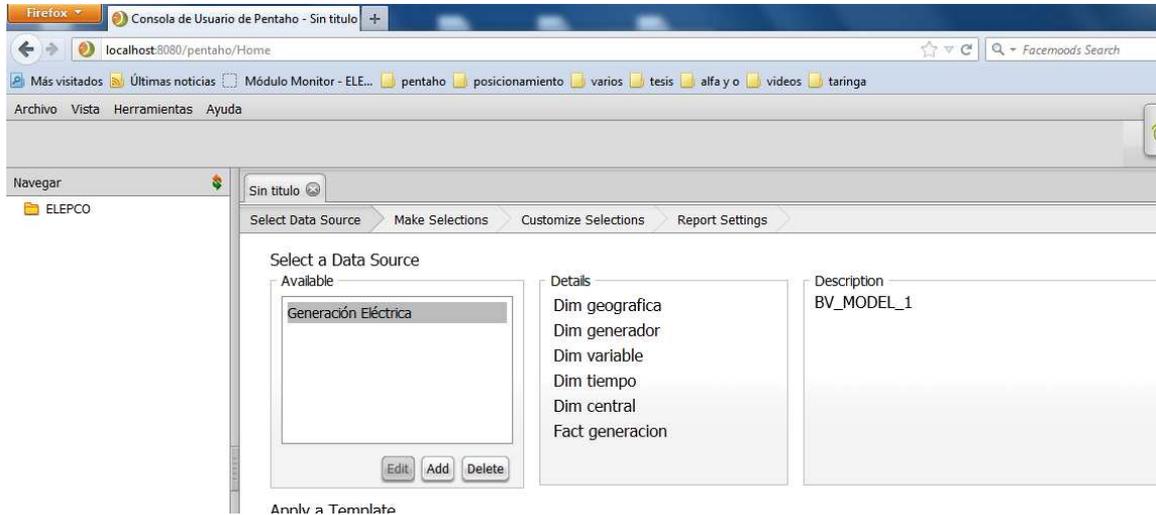


Imagen 4.47 Estructura para realizar reportes.

Creación del cubo.

El objetivo de esta sección es diseñar el cubo de información a partir de la base de datos “cubo_central” desarrollado en el apartado **“Desarrollo de la base de datos para el cubo”**, para este punto se ocupará la herramienta Pentaho Schema Workbench o también conocida como Mondrian; como se vio anteriormente este programa se encarga de definir una estructura que permitirá a Pentaho representarla como una vista multidimensional de la base de datos y proveer el servicio OLAP y el generador de tablas y gráficos dinámicos.

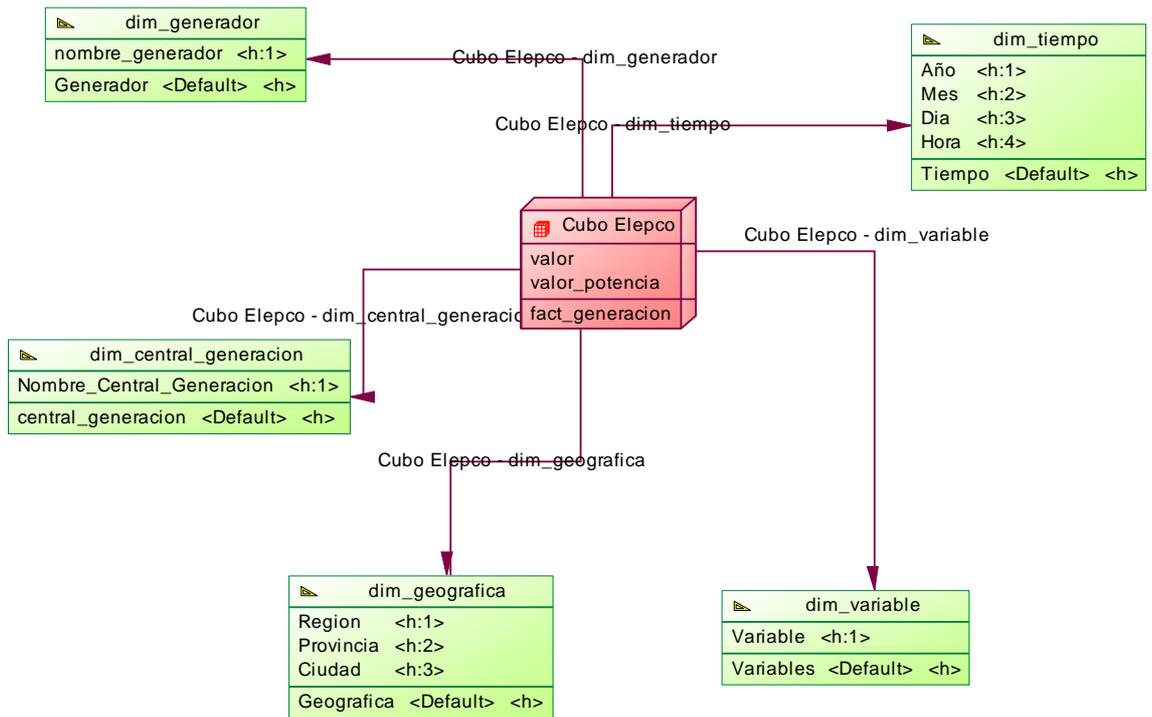


Gráfico 4.27 Diagrama multidimensional de la base de datos “cubo_central”

En el Schema Workbench se comienza definiendo la conexión con la base de datos “cubo_central”, y luego se crea un nuevo esquema para comenzar con la creación de la definición del cubo.

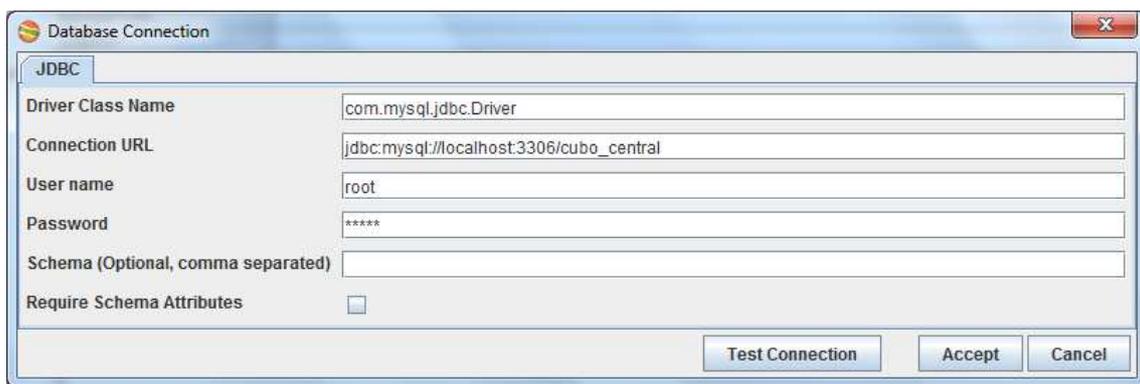


Imagen 4.48 Ventana de conexión a la base de datos.

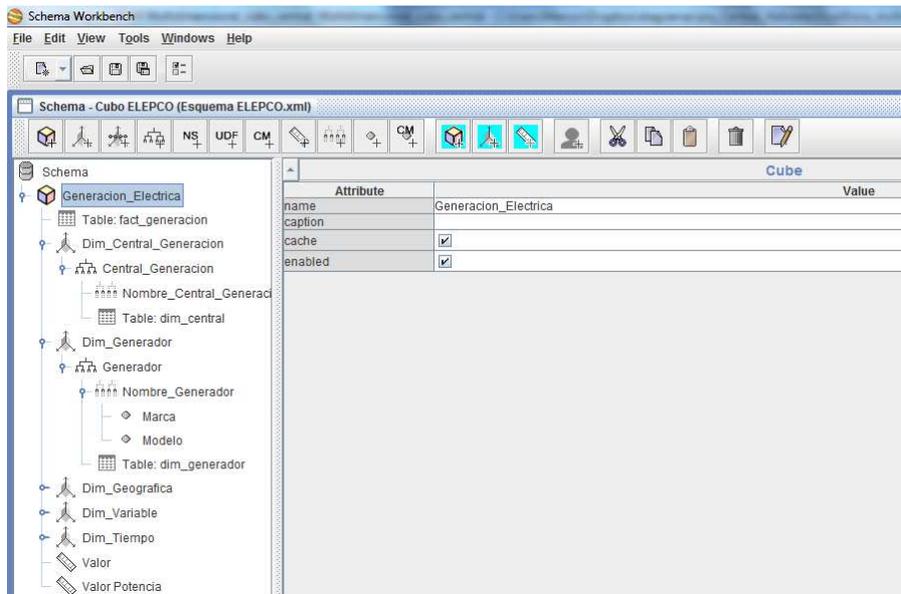


Imagen 4.49 Definición del cubo.

Tabla 4.26 Significado de cada icono de la estructura.

ÍCONO	DESCRIPCIÓN
	Cubo de información
	Dimensión
	Jerarquía
	Nivel
	Atributo
	Tabla
	Métrica

Al igual que Metadata Editor, Schema Workbench permite publicar la estructura del cubo creada en el repositorio de Pentaho, para ello se debe escoger la opción “File->Publish...” y se muestra en la captura de pantalla a continuación:



Imagen 4.50 Conexión al repositorio de Pentaho desde Schema Workbench.

En la Imagen 4.50 se muestra la forma de conectarse al repositorio de Pentaho, luego de acceder con las claves necesarias aparece otra pantalla (Imagen 4.51) preguntando en qué solución del repositorio se debe almacenar este cubo, a continuación la captura de esta pantalla:

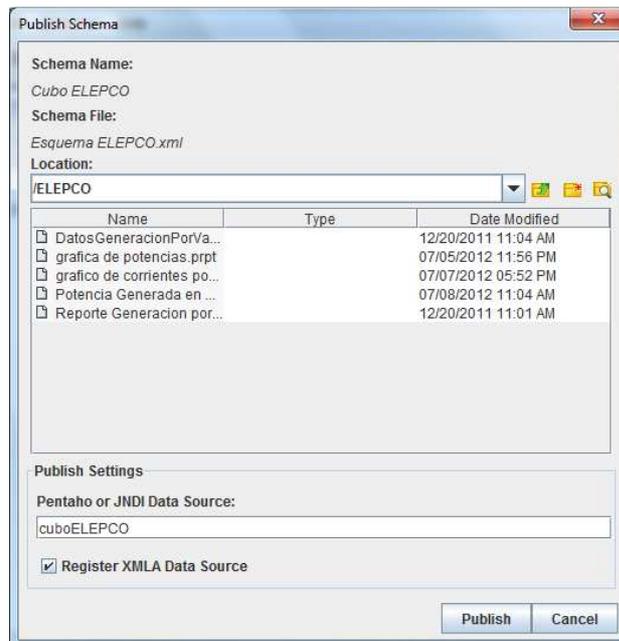


Imagen 4.51 Solución del repositorio en donde se debe almacenar.

Una vez que este registrado en el repositorio de Pentaho el esquema del cubo ya se lo puede revisar en la consola del usuario como un nuevo esquema para los reportes Ad-Hoc de vistas de análisis como se ve en la siguiente captura de pantalla:

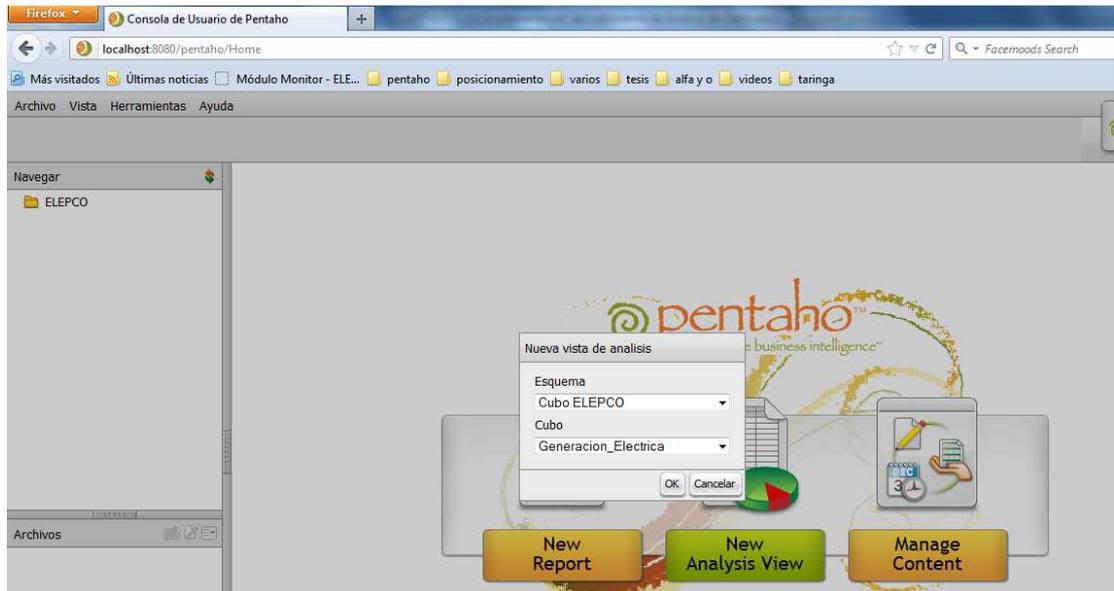


Imagen 4.52 Consola del Usuario, esquema del Cubo.

Al escoger el esquema Cubo ELEPCO a continuación muestra ya la tabla cruzada (pívo table) creada a partir de la definición del cubo:

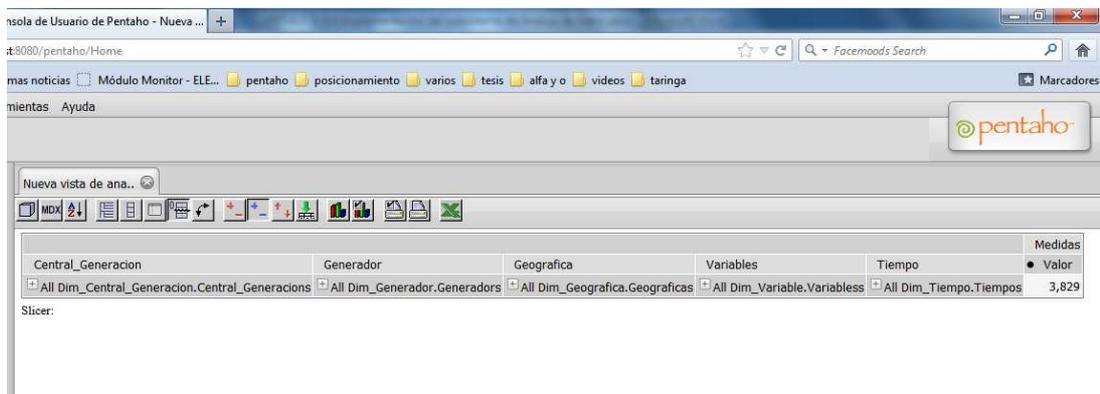


Imagen 4.53 Pivot Table.

Desarrollo de reportes.

El objetivo de esta historia de usuario es desarrollar los reportes solicitados por ELEPCO y que estarán prediseñados en la plataforma, esta tarea se dividirá en las siguientes tareas:

- Desarrollar reporte de curvas de potencia generada hora a hora por cada día de cada generador (cuatro en total) en kw por cada fase.
- Desarrollar reporte de curvas de potencia generada hora a hora por cada día de cada generador (cuatro en total) en kw por todas las fases.
- Desarrollar reporte de curvas de la potencia total de todos los generadores en kw/h.
- Desarrollar reporte de curvas de variación de voltaje de barras por cada grupo y fase.
- Desarrollar reporte de Curvas de variación de corriente en función del tiempo de cada grupo y fase.

La herramienta que se usa para crear estos reportes será el Report Designer que viene con la suite de Pentaho.

Desarrollar reporte de curvas de potencia generada hora a hora por cada día de cada Generador (cuatro en total) en kw por cada fase.

Para desarrollar este reporte se va a partir como fuente de datos las siguientes tablas de la base de datos "cubo_central":

- Fact_generacion
- Dim_generador
- Dim_tiempo
- Dim_variable

Una vez incluidas estas tablas al reporte se va a añadir un gráfico de líneas donde se mostrará en el eje de las X la fecha y hora de generación, y en el eje de las Y las fases, en el área de valores se va a mostrar el valor de la potencia.

Por último se añadirán dos parámetros para que el usuario pueda elegir qué grupo de generación desea ver y las horas a mostrar.

Report Designer permite publicar directamente los reportes a la solución de Pentaho que se creó anteriormente, esto permitirá que el usuario tenga acceso al reporte desde la consola de usuario.

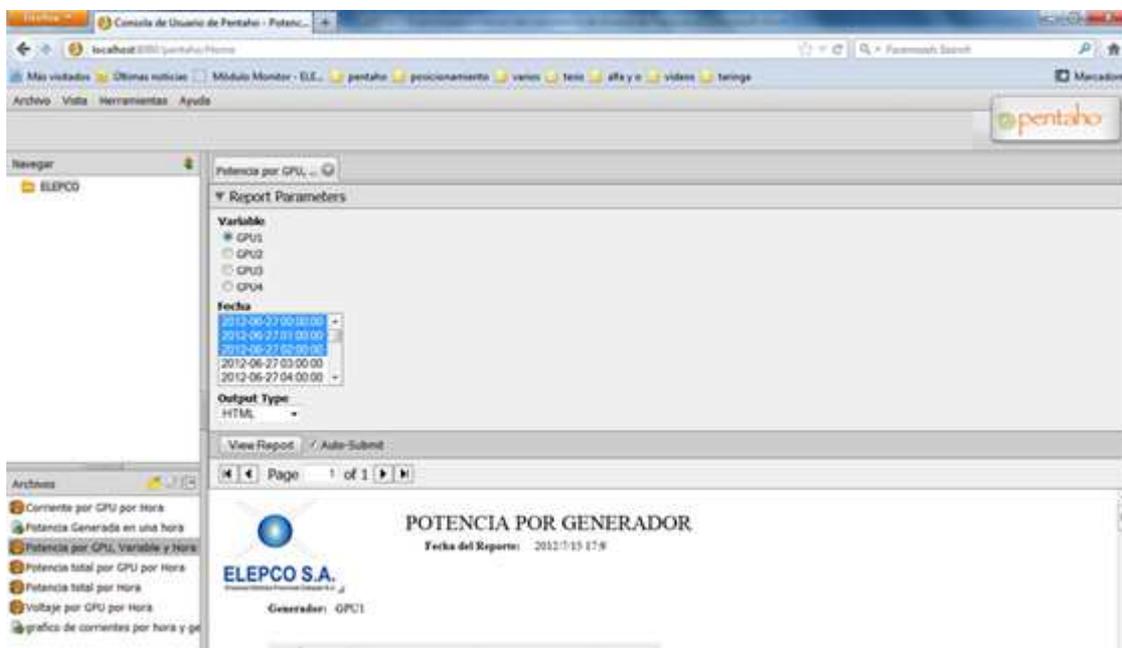


Imagen 4.54 Área de parámetros del reporte.

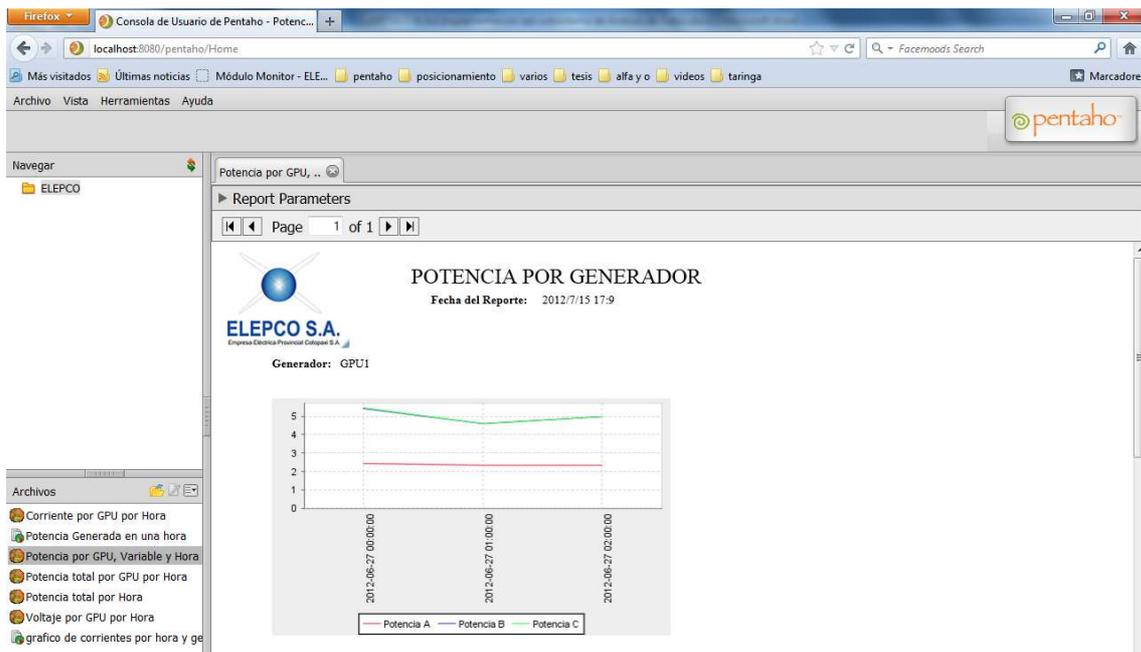


Imagen 4.55 Área del reporte.

Desarrollar reporte de curvas de potencia generada hora a hora por cada día de cada generador (cuatro en total) en kw por todas las fases.

En este reporte se va a necesitar las siguientes tablas de la base de datos “cubo_central”:

- Fact_generacion
- Dim_generador
- Dim_tiempo

Se necesita totalizar los valores de las tres potencias de cada generador por lo que en la consulta SQL se aumentan las agregaciones por generador y fecha de lectura.

Al igual que el reporte anterior se añadirán dos parámetros para que el usuario pueda elegir qué grupo de generación desea ver y las horas a mostrar.

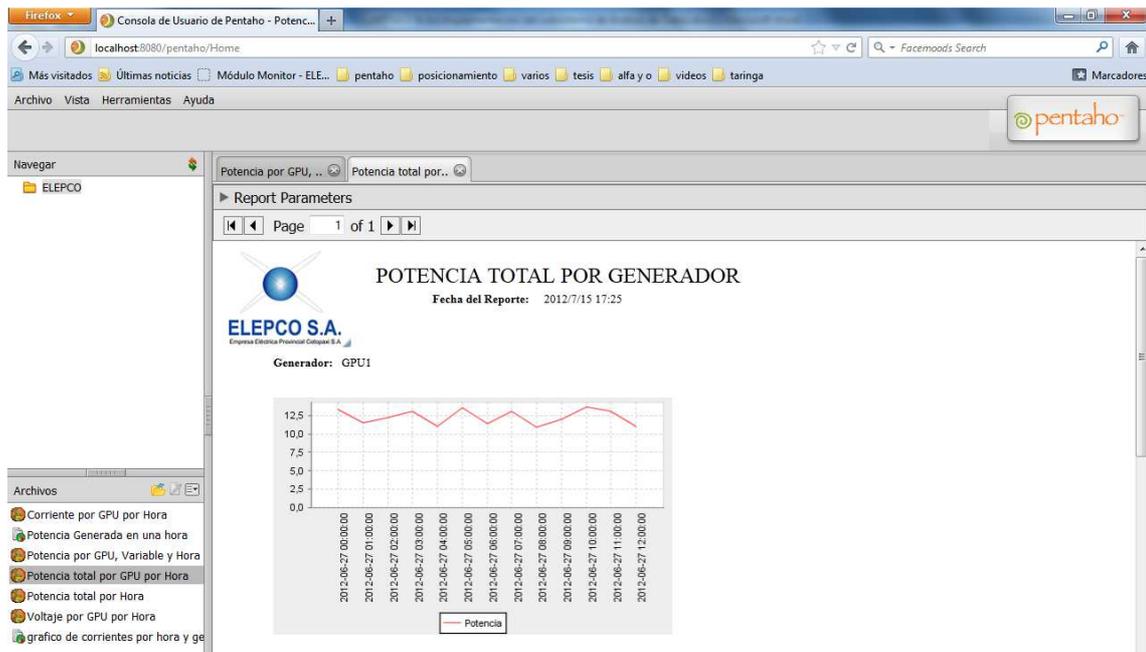


Imagen 4.56 Área del reporte.

Desarrollar reporte de curvas de la potencia total de todos los generadores en kw/h.

La potencia total de todos los generadores es igual a la suma de todas las potencias de todos los generadores, las tablas necesarias para este reporte serán:

- Fact_generacion
- Dim_tiempo

En este caso se totalizará todas potencias agrupadas por el factor tiempo.

Se añadirá un parámetro para que el usuario elija las horas a mostrar.

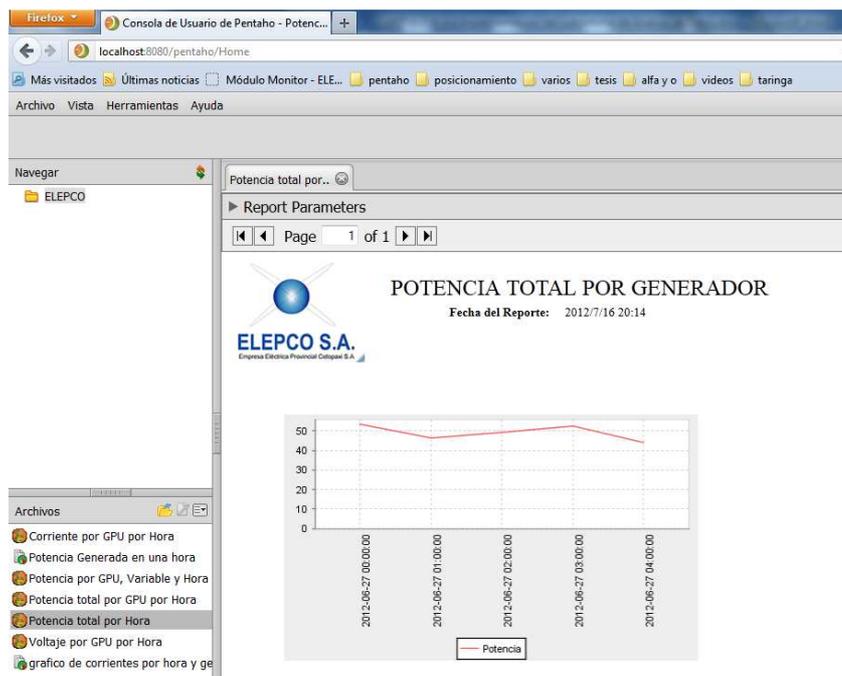


Imagen 4.57 Área del reporte.

Desarrollar reporte de curvas de variación de voltaje de barras por cada grupo y fase.

Este reporte es similar al reporte de potencias por grupo y fase por lo que se necesitan las siguientes tablas:

- Fact_generacion
- Dim_generador
- Dim_tiempo
- Dim_variable

En este caso se filtrarán todas las lecturas que contengan el tipo de variable "V" (Voltaje).

Los parámetros que se colocarán en este reporte son:

- Grupo de Generación
- Tiempo

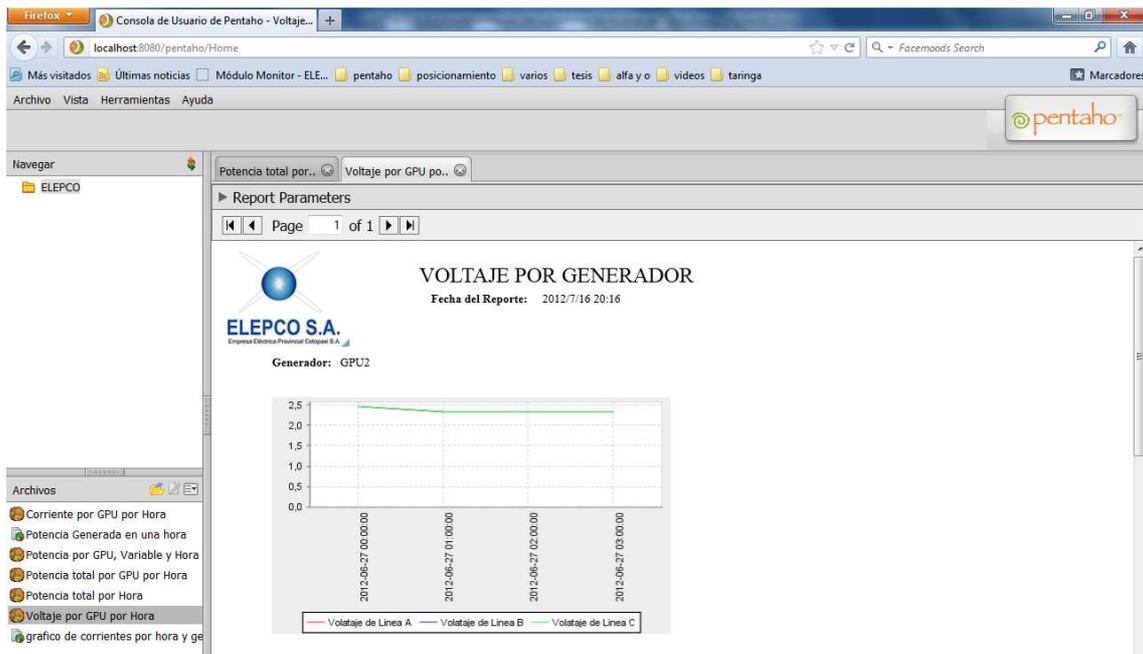


Imagen 4.58 Reporte voltaje por generador.

Desarrollar reporte de Curvas de variación de corriente en función del tiempo de cada grupo y fase.

Para este reporte las tablas necesarias son:

- Fact_generacion
- Dim_generador
- Dim_tiempo
- Dim_variable

Los filtros que se crearán son:

- Generador
- Tiempo.

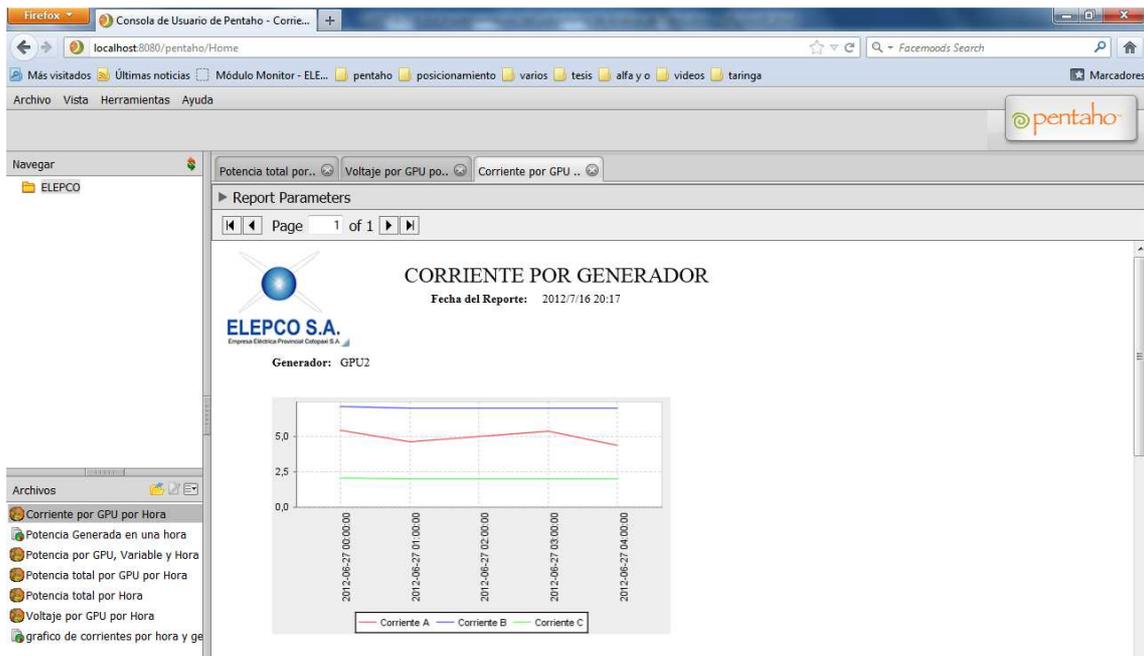


Imagen 4.59 Reporte corriente por generador

Carta de Burndown

Una vez finalizado el desarrollo, se completa la pila del sprint actualizando el día de finalización y se genera el gráfico del Burndown.

Tabla 4.27: Información a utilizarse para elaborar la carta de Burndown

No	Historia de usuario	Categoría	Estimación inicial en puntos.	Finalización	Puntos Restantes
1	Desarrollo de la base de datos para el cubo.	Análisis de datos	6	Día 4	96
2	Instalación y configuración de la Suite Pentaho.	Análisis de datos	24	Día 8	72
3	Creación de un proceso de ETL para la carga de datos.	Análisis de datos	24	Día 15	48
4	Creación de Metadatos.	Análisis de datos	12	Día 18	36
5	Creación del cubo.	Análisis de datos	12	Día 20	24
6	Desarrollo de reportes.	Análisis de datos	24	28	0

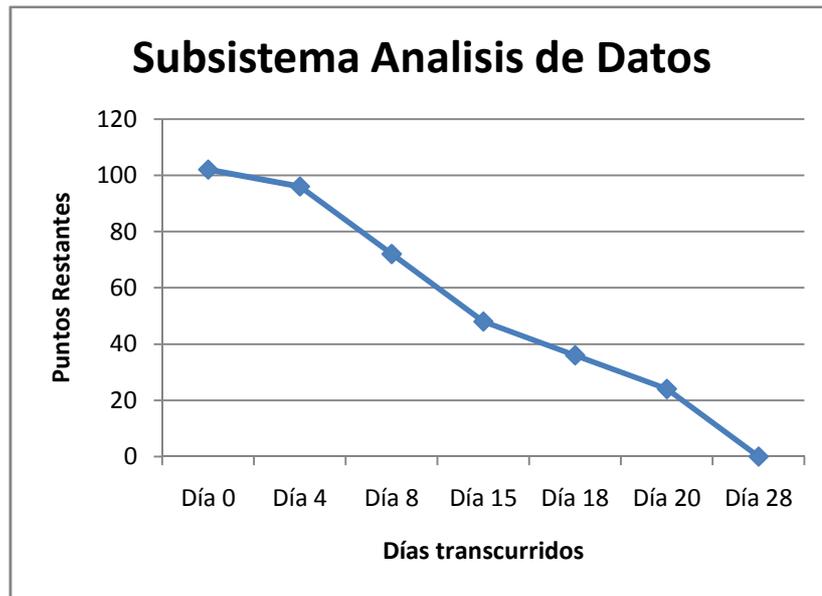


Gráfico 4.28 Carta de Burndown final del subsistema Análisis de Datos.

Revisión.

Este es el último Sprint del sub sistema Análisis, el dueño del producto acepta el resultado final de los reportes predefinidos y de la estructura generada para generar reportes ad-hoc desde el subsistema. El subsistema fue implementado en el tiempo previsto.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES.

5.1.- Conclusiones.

- El sistema fue desarrollado e implementado en la Empresa Eléctrica Provincial de Cotopaxi, recolecta información de generación eléctrica de forma automática, permite las funciones de monitoreo y control de los equipos de campo, ayuda con la planeación de mantenimiento de los equipos y emite reportes de la generación eléctrica ayudando al usuario con su tarea de análisis de información.
- Con el sistema se ha logrado automatizar todo el proceso de monitoreo por lo que ya no se necesita la intervención de los operadores en el ingreso de datos y acondicionamiento y por esta razón se ha logrado que la autenticidad de los reportes sea exacta con la realidad y por ende se agiliza el proceso análisis en un 90%, tomando en cuenta que anteriormente para calcular el total generado de un equipo se demoraba 3 horas ahora con el sistema desarrollado lo pueden obtener en menos de 5 minutos.
- La metodología de desarrollo software Scrum utilizada para la implementación del presente proyecto, permite llevar a cabo un proceso de desarrollo de forma ordenada, flexible a los cambios y ágil en comparación a las metodologías tradicionales como RUP.
- La reutilización de código agiliza el proceso de desarrollo. En el proyecto se desarrollaron componentes reutilizables como por ejemplo en el subsistema de monitoreo y control las gráficas de los equipos de campo, los estilos de las tablas.

- Se utilizó el lenguaje de programación PHP para el desarrollo del subsistema de Monitoreo y Control obteniendo un buen resultado en la arquitectura diseñada permitiendo tener un buen resultado en las seguridades del sistema y en la estabilidad.
- Con el sub sistema de análisis de datos se logra que el usuario pueda disminuir tiempo en analizar la información obtenida, además se obtiene reportes gerenciales fáciles de entender. El uso de un cubo de información resume y consolida la información de forma rápida según los niveles de agrupación que se crearon.
- Para la documentación de los requerimientos se utilizó el formato de la IEEE 830 el mismo que permitió tener un documento formal en cual basarse para desarrollar el proyecto.

5.2.- Recomendaciones.

- La información de la base de datos “central” crece muy rápido y en un año puede demorar las consultas al mismo, por lo que se recomienda tener la información del último año en esta base de datos y mantener toda la información histórica en la base de datos “cubo_central” donde ya se encuentra resumida.
- Al poder únicamente acceder al sistema a través de la red de la empresa, se recomienda utilizar internet como alternativa de conexión remota a los subsistemas del proyecto desarrollado.

- Mantener respaldos de toda la aplicación, y respaldar las bases de datos periódicamente para evitar pérdida de información debido a eventos inesperados.
- Monitorear el enlace entre los equipos de campo y el computador de lectura periódicamente para evitar pérdida de información a causa de la ruptura de los enlaces.

BIBLIOGRAFÍA.

- http://es.wikipedia.org/wiki/Instrumentaci%C3%B3n_electr%C3%B3nica
- http://books.google.com.ec/books?id=7Poo_SxtYKEC&pg=PA11&lpg=PA11&dq=caracteristicas+de++hp+vee&source=bl&ots=Wkslxc4MmA&sig=_y3BkvwysbHomnhJBcRGsGkfc&hl=es&ei=Df1UTNzyPML38Aa2yLjhCA&sa=X&oi=book_result&ct=result&resnum=4&ved=0CDQQ6AEwAw#v=onepage&q&f=false
- <http://jeffreytravis.com/lost/labsql.html>
- <http://www.maestrosdelweb.com/editorial/herramientas-adecuadas-para-el-diseno-y-desarrollo-de-un-sitio-web/>
- <http://davidasorey.net/static/perl-php-bbdd-tutorial/>
- <http://www.enterate.unam.mx/Articulos/2006/abril/perl.htm>
- <http://www.php-es.com/faq.languages.html>
- <http://es.wikipedia.org/wiki/PHP>
- <http://www.superhosting.cl/manuales/introduccion-a-la-programacion-en-php.html>
- <http://www.maestrosdelweb.com/editorial/phpintro/>
- <http://www.adrformacion.com/cursos/php/leccion1/tutorial1.html>
- http://www.webtaller.com/manual-php/introduccion_php.php
- http://www.programacion.com/articulo/por_que_elegir_php_143
- <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>
- <http://www.umanizales.edu.co/programs/ingenieria/Ventana/ventana12/articulo25.pdf>

- http://es.wikipedia.org/wiki/MySQL_Cluster
- <http://es.wikipedia.org/wiki/MySQL>
- http://netbeans.org/index_es.html
- <http://playa-soluciones.cl/blog/7-investigacion-y-desarrollo/31-razones-por-las-cuales-usar-netbeans-para-proyectos-php>
- <http://es.wikipedia.org/wiki/NetBeans>
- http://es.wikipedia.org/wiki/Mapeo_objeto-relacional
- <http://www.propelorm.org/>
- <http://es.wikipedia.org/wiki/JavaScript>
- <http://es.wikipedia.org/wiki/AJAX>
- <http://es.wikipedia.org/wiki/JQuery>
- <http://es.wikipedia.org/wiki/SCADA>
- [http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_softwa
re](http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_softwa_re)
- <http://www.slideshare.net/fmmeson/metodologias-de-desarrollo-de-software-tradicionales-vs-agiles>
- <http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>
- <http://www.scrum.org/scrumguideenglish/>
- www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf
- <http://sites.google.com/site/oeguzman/losrolesdescrum>
- www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf
- <http://www.gestiopolis.com/canales2/gerencia/1/busint.htm>
- <http://www.monografias.com/trabajos14/bi/bi.shtml>

- <http://www.monografias.com/trabajos75/inteligencia-negocios/inteligencia-negocios.shtml>
- http://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos
- <http://www.monografias.com/trabajos75/inteligencia-negocios/inteligencia-negocios2.shtml>
- http://www.sinnexus.com/business_intelligence/datawarehouse.aspx
- <http://www.monografias.com/trabajos75/inteligencia-negocios/inteligencia-negocios2.shtml>
- <http://es.wikipedia.org/wiki/OLAP>
- <http://www.monografias.com/trabajos75/inteligencia-negocios/inteligencia-negocios.shtml>
- http://www.mextisa.com.mx/site/index.php?option=com_content&view=article&id=7&Itemid=7&limitstart=1
- <http://www.trevenque.es/index.php/BI-en-las-diferentes-areas-empresariales/471/0/>
- <http://es.wikipedia.org/wiki/Echelon>
- <http://ignaciobustillo.com/blog/blog/63-ique-es-pentaho>
- <http://pentaho.almacen-datos.com/mondrian.html>
- <http://mistock.lcompras.biz/paradigmas23/1812-ireport>
- <http://community.pentaho.com/>
- <http://jasperforge.org/projects/>
- <http://www.birt-exchange.com/be/downloads/>
- <http://www.jedox.com/en/jedox-downloads/>
- <http://jquery.com/demo/thickbox/>

ANEXO A

Especificación de requisitos del sistema

1. Introducción

1.1. Propósito

El presente documento tiene como objetivo indicar de una manera clara y sencilla el alcance del sistema solicitado por la Empresa Eléctrica Provincial de Cotopaxi (ELEPCO), en forma detallada se revisará cada aspecto importante para el proyecto siguiendo las normas para la documentación de los requisitos. Este documento será el punto de inicio del proyecto y estará vigente hasta la culminación del proyecto, será utilizado por el gestor del proyecto y los desarrolladores quienes se guiarán con las funciones detalladas aquí.

1.2. Ámbito del sistema

El sistema que se desarrollará se lo nombrará Sistema de Monitoreo ELEPCO, estará dividido en tres subsistemas que serán:

- **Obtención de datos.**

Este será la base de todo el sistema, porque será el encargado de estar continuamente en comunicación con los equipos de campo a ser monitoreados, su función será hacer una lectura periódica de los GPU's y los tanques de agua, los mismos que serán provistos por un sistema ya existente, es decir; se debe realizar una extensión a este sistema para que todos los datos de lectura se realice un proceso de acondicionamiento o transformación a valores reales, también debe calcular la potencia generada y almacenar estos datos en una base de datos. No se realizarán tareas de control para modificación de procesos

(encendido, apagado, cambios de configuración) porque esta tarea está restringida por los GPU's para uso desde sus diplay's.

Esta extensión del sistema deberá ser realizado en la misma herramienta con la que está desarrollado el sistema actual la misma que es LabView 8.5, herramienta de instrumentación virtual para control y automatización de equipos electrónicos.

Se debe realizar un sistema independiente autónomo que luego se integrará al sistema existente.

- **Monitor.**

Será una interface web donde que su función principal es mostrar los valores actuales de los generadores y de los tanques para que el operador pueda realizar las funciones de monitoreo. Este módulo tendrá todas las funcionalidades que conlleva el monitoreo, excepto las de control de procesos, entre las alarmas que se monitorearán están: encendido de equipos, apagado de equipos, otras alarmas se encuentran restringidas por los GPU's.

- **Analizador Datos Históricos**

Será un sistema reporteador, para esto se requiere el uso de un sistema de inteligencia de negocios para aprovechar las herramientas que tienen estos sistemas para la explotación de datos, comparación y análisis.

Se debe realizar un estudio de análisis para determinar la suite de Business Intelligence más apropiada para el proyecto, se necesita una Suite de software libre, que contenga las siguientes herramientas:

- Gestor OLAP.
- Herramienta ETL.

- Dashboarding o tablero de instrumentos.
- Herramienta para modelado de datos (Data Modeling).
- Herramienta para definir reglas de negocio.
- Herramienta para definir lógica de negocio.

1.3. Definiciones, Acrónimos y Abreviaturas

- OLAP: Procesamiento analítico en línea (On line analytical processing)
- ETL: Extraer, transformar y cargar (Extract, transform and load)
- GPU: Unidad de protección del generador (Generator protection unit)

1.4. Visión General del Documento

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

2. Descripción General

2.1. Perspectiva del producto

La perspectiva del sistema será la recopilación de datos de los GPU's y sensores de nivel de agua de la central Hidroeléctrica de ELEPCO S.A., el monitoreo de las variables que estarán en continua lectura y el análisis de los datos históricos que se irán guardando con el funcionamiento del sistema.

Funciones del producto

El sistema comienza a partir del subsistema de captura de datos este recibirá los datos de un programa dedicado a la lectura de los equipos de campo, instalado en una máquina conectada a los sensores, se recibirá un arreglo de datos con el valor de cada componente, los valores recibidos no estarán acondicionados sino que vendrán con los valores almacenados en la memoria los mismos que están entre 0 y 4095, aplicando unas fórmulas que serán provistas por ELEPCO S.A. se obtendrá el valor acondicionado o valor real de la variable, luego de esto se calculará la potencia generada y se almacenará en la base de datos, con la fecha y hora de la lectura.

En el subsistema Monitor se presentará en forma gráfica el estado tanto de los generadores de electricidad como de los tanques de agua y se emitirá una señal de aviso de los eventos importantes como encendido de equipos, o bajos niveles de agua entre otros, además se podrá consultar la información histórica de los datos almacenados por equipo en un rango de fechas. El sistema será de arquitectura abierta; es decir, que permitirá un fácil crecimiento del módulo con gestión de menús y opciones, se administrarán los equipos con una distribución que permita clasificarlos por ubicación y por central de generación puesto que luego se desea que el sistema pueda administrar otras centrales de generación eléctrica que posee la empresa.

Dentro las funciones de visualización de datos están:

- Generar gráficos del estado actual de los equipos, el mismo que tendrá una actualización automática y mostrara las últimas 3 horas de generación (valor configurable).
- Generar un gráfico de resumen del valor de las potencias generadas en total por todos los equipos de la central hidroeléctrica.
- Generar gráficos históricos de generación eléctrica.
- Generar gráficos del estado actual de los niveles de agua, actualizable automáticamente y mostrará el último valor obtenido de la lectura del sensor.

Este módulo también tendrá funciones de administración como:

- Administración de usuarios y grupos de usuarios.
- Administración de centrales de generación.
- Administración de GPU's.
- Administración de tanques.
- Asignación de permisos de grupos de usuarios a menús y opciones.
- Administración de menús y opciones.
- Administración de estructura geográfica (Región, Provincia, Ciudad) de los equipos.
- Administración de configuraciones.
- Configuración del comportamiento del módulo monitor.
- Administración de la planificación de mantenimiento de los equipos.

Contará también un con un planificador de mantenimiento tanto de equipos como de los tanques de agua y otros equipos que están involucrados en el proceso de generación eléctrica y que por causa de su mantenimiento se requiera

alguna pausa en el proceso con el objetivo de documentar las pausas que se han realizado y las futuras pausas que podrán hacerse al proceso por motivos de mantenimiento de equipos.

El módulo de Análisis de Datos Históricos presentará unas opciones de reportes predefinidos y de desarrollo de reportes, entre los tipos de reportes que se podrán obtener están:

- Tablas cruzadas.
- Reportes planos.
- Gráficas de pasteles.
- Gráficos de líneas.
- Gráficos de Barras.

2.2. Características de los usuarios.

El subsistema Capturador de datos será un programa autónomo, no manejará funciones por usuarios porque tendrá una sola función repetitiva y debe ser lo más ágil posible, y su uso estará a cargo de un técnico de ELEPCO.

El subsistema Monitor será multiusuario con un súper usuario que es el "Admin" perteneciente al grupo de usuarios "Administradores", el será quien tiene acceso a todas las opciones del sistema con todos los permisos de lectura, escritura, eliminación, el usuario "Admin" podrá otorgar permisos a cada grupo de usuario a cada menú y opción del sistema. A cada opción se podrá atribuir derechos de escritura, lectura y eliminación.

Los usuarios para el subsistema de Analisis de Datos Históricos estarán gestionados por la herramienta de Inteligencia de Negocios seleccionada, se requiere que sea multiusuario con niveles de acceso.

2.3. Restricciones

La restricción más importante es la lectura de datos que por ser equipos electrónicos se debe ocupar software orientado a esa especialidad, la captura de datos se realizará desde un programa desarrollado en Labview el mismo que se comunicará con otro sistema desarrollado por ELEPCO S.A. para la captura de datos, este sistema debe funcionar bajo el sistema operativo Windows.

El subsistema de monitoreo será desarrollado en el lenguaje PHP con Ajax, JavaScript, JQuery y otras librerías de diseño y contenido. El IDE de desarrollo será NetBeans 6.8 Early Access for PHP y se usará el gestor de bases de datos MySql 5.1.41.

Para el subsistema de Análisis de Datos Históricos se debe investigar una plataforma de Inteligencia de Negocios que mejor se adapte para las necesidades del proyecto, entre los requerimientos realizados por ELEPCO para la selección de la herramienta están:

- Licencia de Software Libre
- Posea servidor OLAP
- Posea Data Modeling
- Posea editor de reportes
- Posea una herramienta ETL (Extracción, transformación y carga)

Además que sea de fácil uso y contenga una base de conocimientos online y una comunidad activa de usuarios.

2.4. Suposiciones y Dependencias

El sistema está desarrollado y probado bajo un ambiente Windows XP y Vista, pero puede funcionar en versiones superiores como Windows 7 y versiones de servidor desde Windows 2000 server en adelante, los módulos de monitoreo y análisis de datos históricos pueden funcionar en otras plataformas como Linux pero no se realizarán pruebas de funcionamiento en estos sistemas operativos.

2.5. Requisitos Futuros

- Control remoto de los equipos.
- Aumento de equipos de campo.
- Nuevas alarmas.

3. Requisitos Específicos.

3.1. Interfaces Externas

El sistema contará con 2 tipos de interfaces externas, para el módulo de Captura de Datos será una interface de escritorio sencilla, tendrá una sola pantalla y funciones específicas para la lectura y almacenamiento en la base de datos, con configuraciones para el período de lectura, tiempo de timeout.

Los módulos de Monitoreo y análisis de datos históricos tendrán una interface web.

3.2. Funciones

3.2.1. Subsistema Capturador de Datos

- Lectura y acondicionamiento de los valores recibidos del sistema principal de ELEPCO.

- Cálculo de la potencia generada.
- Almacenamiento de los datos recolectados en una base de datos.

3.2.2. Subsistema Monitor

- Presentar los datos de generación eléctrica y niveles de agua en forma gráfica.
- En la pantalla principal siempre presentar el estado (encendido o apagado) de cada equipo.
- Ingreso al sistema mediante usuario y contraseña.

Aparte de las funciones detalladas anteriormente tendrá las siguientes funciones u opciones agrupadas en menús que serán los siguientes

- Estado de Generadores
 - o Visualización de datos actuales de generación general y por GPU.
 - o Creación de gráficos históricos de generación eléctrica: Posibilidad de obtener gráficas de líneas de un rango de fechas por medida (Potencia, Voltaje, Amperaje, Nivel de Agua) y por unidad de campo (GPU).
- Menú de Mantenimiento de datos.
 - o Gestión de usuarios: administración de la información de las personas quienes tendrán acceso al sistema.
 - o Gestión de grupos de usuarios: administración de la información de grupos de usuarios o perfiles de acceso al sistema.
 - o Gestión de Menús y Opciones: administración de las funciones del sistema, es decir; se ingresan las opciones agrupadas en diferentes

menús que tendrá el sistema, esto nos dará mayor facilidad para agregar o quitar funciones.

- Gestión de Plantas Generadoras: Se requiere un identificador de la planta, un nombre, y una estructura geográfica que estará diseñada por Región, Provincia, Ciudad y Dirección.
- Gestión de la estructura geográfica: Mantenimiento de cada nivel (Región Provincia, Ciudad).
- Administración de Unidades de Campo (GPU y Tanque): Especificación de cada uno de los elementos a ser monitoreados por el sistema, se asignará cada uno de ellos a la central de generación a la que corresponde.
- Configuraciones
 - Gestión de permisos: asignación de permisos de Lectura, escritura y eliminación para uso de los menús y opciones a cada grupo de usuario.
 - Configuración de las lecturas de los equipos de campo: Cada equipo de campo tiene un controlador (PLC), en este controlador tiene un almacén de datos con posiciones donde se almacenara una lectura diferente del equipo (voltaje, potencia, corriente)
- Mantenimiento de equipos
 - Mantenimiento de Generadores: Ingreso, actualización, eliminación de los planes de mantenimiento de cada generador, se especificará si el mantenimiento es preventivo, correctivo o predictivo.

- Mantenimiento de Tanques: Ingreso, actualización, eliminación de los planes de mantenimiento de cada tanque de agua, se especificará si el mantenimiento es preventivo, correctivo o predictivo.
- Estado de tanques
 - Visualización de datos actuales de generación general y por GPU.
 - Creación de gráficos históricos de generación eléctrica: Posibilidad de obtener gráficas de líneas de un rango de fechas por medida (Potencia, Voltaje, Amperaje, Nivel de Agua) y por unidad de campo (GPU).

3.2.3. Subsistema de Análisis de Datos Históricos.

Dentro de los reportes que se generarán aquí serán:

- Curva de potencia generada hora a hora por cada día de cada grupo (cuatro en total) en kw.
- Curva de la energía generada en kw/h.
- Curvas de variación de voltaje de barras.
- Curvas de variación de corriente en función del tiempo de cada grupo.

3.3. Requisitos de Rendimiento

El sistema será multiusuario y debe tener capacidad de soportar hasta 10 clientes concurrentes.

3.4. Restricciones de Diseño

No aplica.

3.5. Atributos del sistema

- Los módulos de Monitor y de análisis de información histórica serán multiusuario.
- Los usuarios que tengan derechos de administradores serán quienes puedan configurar los parámetros del sistema.
- El módulo de captura de datos estará bajo las seguridades del sistema ya existente en ELEPCO.
- El módulo de captura será configurado a partir del sistema ya existente en ELEPCO al cual se integrará.

APÉNDICE A

BIOGRAFÍA AUTOR

Marcos Patricio Armas Villamarín, nació en Latacunga, el 25 de Abril de 1985 sus padres Marcos Armas y María Villamarín, su educación primaria lo realizó en la escuela Isidro Ayora y la secundaria en el Instituto Tecnológico Superior Vicente León en la ciudad de Latacunga, posteriormente ingresó a la Escuela Politécnica del Ejército en la carrera de Ingeniería en Sistemas e Informática.

Actualmente se encuentra trabajando en Soft Service Pluss S.C.C. empresa dedicada al desarrollo y soporte del sistema de administración de recursos humanos DataLife ®.

APÉNDICE B

BIOGRAFÍA AUTOR

Santiago David Lucio Cruz, nació en Quito, el 07 de Mayo de 1983 sus padres Miguel Lucio y Nelly Cruz, su educación primaria lo realizó en la escuela Isidro Ayora y la secundaria en el Instituto Superior Tecnológico “Ramón Barba Naranjo” en la ciudad de Latacunga, posteriormente ingresó a la Escuela Politécnica del Ejército en la carrera de Ingeniería en Sistemas e Informática.

Actualmente se encuentra trabajando en el Banco Procredit empresa financiera.

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADA(O) POR

Marcos Patricio Armas Villamarín

Santiago David Lucio Cruz

DIRECTOR DE LA CARRERA

Ing. Mauricio Campaña O.

Lugar y fecha: Sangolquí, noviembre del 2012