



**PROYECTO ESTABLECIMIENTO DE UN SISTEMA NACIONAL DE
ESTADÍSTICAS FORESTALES Y COMERCIALIZACIÓN DE MADERA
PD 406/06 Rev. 1 (M)**

Gobierno del Ecuador

Ministerio del Ambiente

Junio 2012

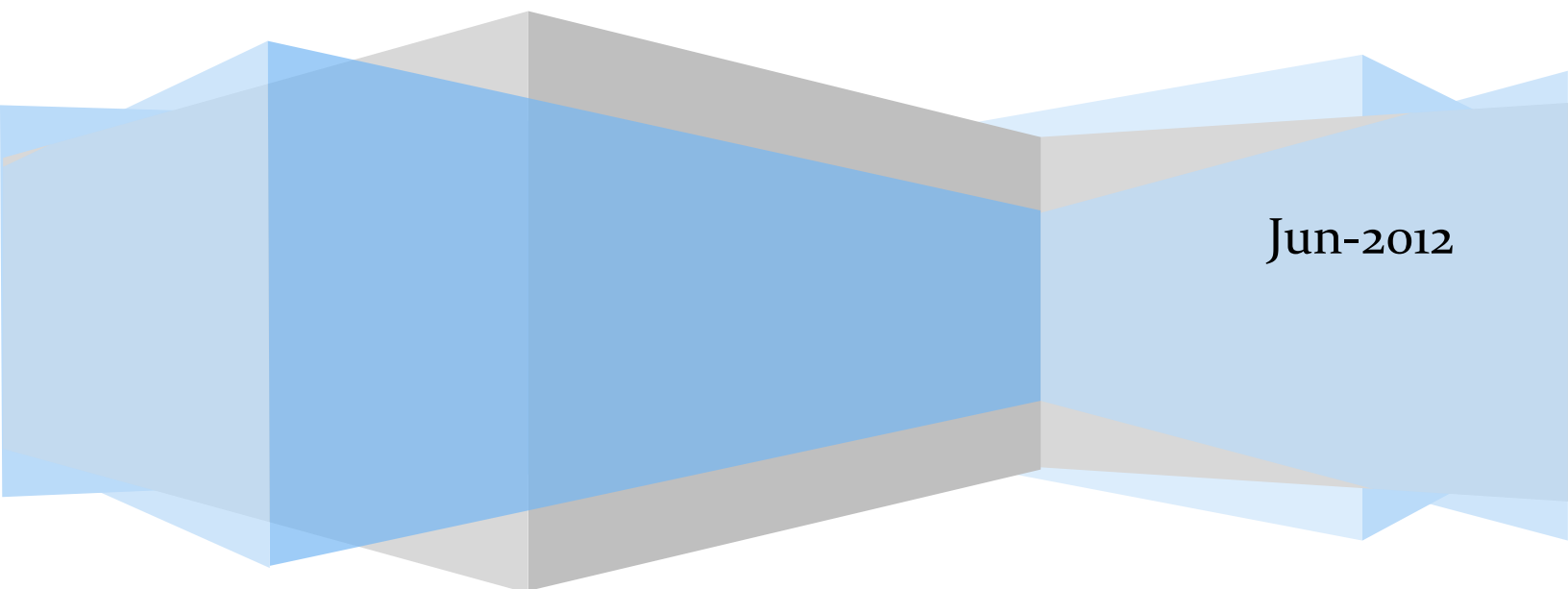
MANUAL DE PROGRAMACIÓN

SISTEMA DE ADMINISTRACIÓN FORESTAL SAF V.2

Diseño y estándares de programación

El presente manual describe el diseño, la estructura de cada componente y base de datos y el estándar de codificación del Sistema de Administración Forestal, SAF. Para realizar el diseño de cada componente se considero los procesos desarrollados en la versión 1.0 y los nuevos requerimientos levantados en el proceso de retroalimentación del sistema a nivel nacional.

Los procesos implementados en cada componente se encuentran normados en el actual Régimen Forestal.



Jun-2012

Contenido

DISEÑO SAF v.2.....	3
Lista de módulos y esquemas del Sistema de Administración Forestal SAF v2.0.....	3
Términos utilizados para el diseño y codificación	4
Estructura de módulos del SAF v.2	5
MÓDULO: Anillo de Seguridades	6
MÓDULO: Registro Nacional de Actividades Forestales.....	9
MÓDULO: Gestión de planes, programas, licencias y guías.....	11
MÓDULO: Importaciones y exportaciones de productos forestales.....	14
MÓDULO: Destino final de la madera	15
MÓDULO: Administración de procesos legales.....	17
Diagrama general de base de datos del SAF	17
Módulo del Sistema de Información Forestal.....	18
MÓDULO: Sistema de gestión de contenido	18
MÓDULO: Componentes De Información Forestal.....	18
Metodología	20
Modelamiento	22
Referencias:	23
ESTÁNDARES DE CODIFICACIÓN	24
Arquitectura MVC.....	24
Codeigniter y Zend Framework.....	24
Formato de archivos PHP	25
Contenido de Directorios	25
Código PHP de modelos vistas y controladoras	26
Controladores.....	27
Modelos	28
Vistas	35
Formularios	35
Referencias:	38

DISEÑO SAF v.2

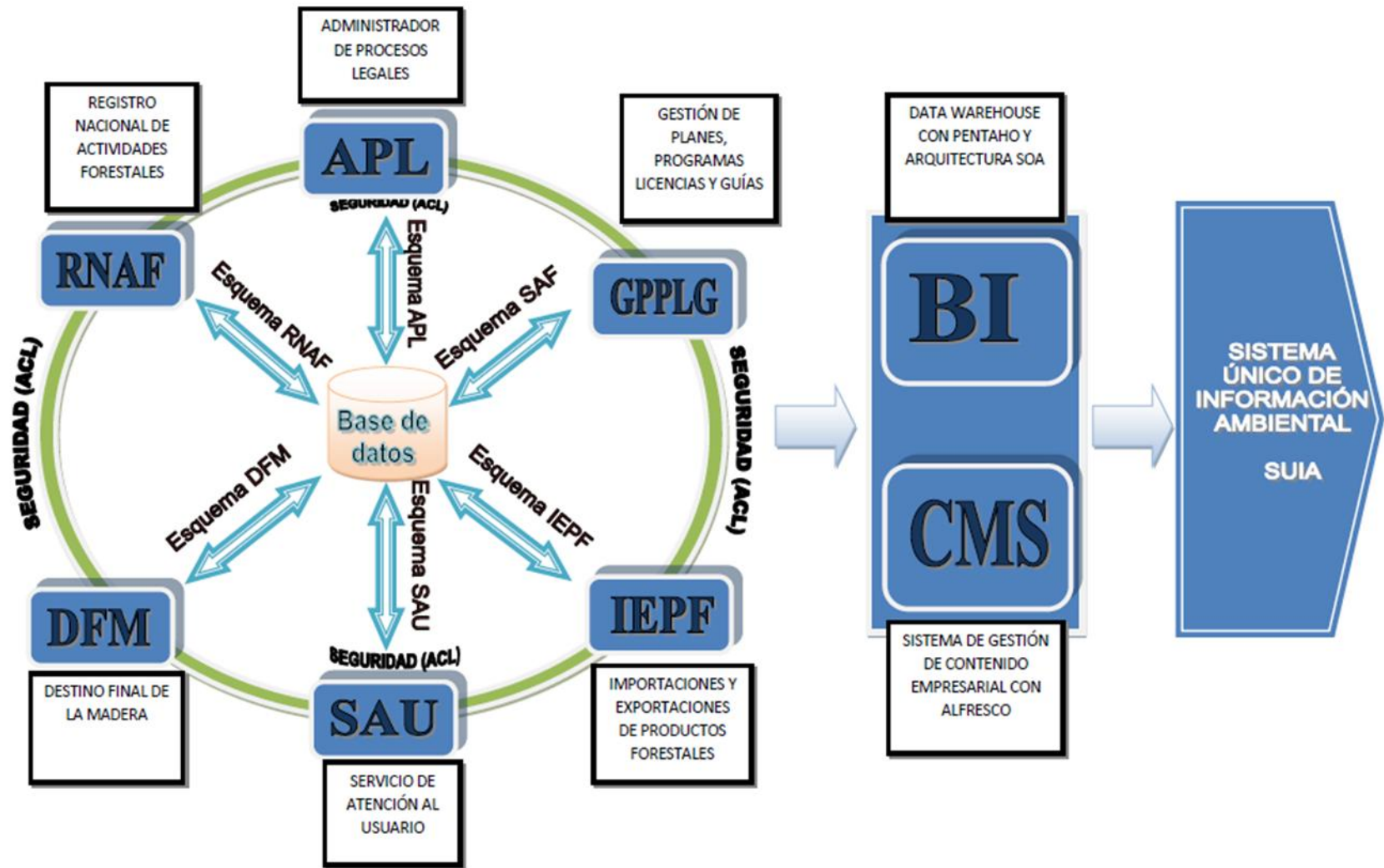
Lista de módulos y esquemas del Sistema de Administración Forestal SAF v2.0

Componente	Siglas	Esquema (BD)
Anillo de Seguridades	AS-ACL	seguridades
Registro Nacional de Actividades Forestales	RNAF	rnaf
Gestión de planes, programas, licencias y guías	GPPLG	saf
Importaciones y exportaciones de productos forestales	IEPF	iepf
Destino final de la madera	DFM	dfm
Administración de procesos legales	APL	apl
Servicios de atención al usuario	SAU	Sau
Módulo de Información Gerencial		
Business Intelligence	BI	Data Warehouse
Sistema de gestión de contenido	CMS	Repositorio de Alfresco

Términos utilizados para el diseño y codificación

Termino	Descripción
SAF	Sistema de Administración Forestal
SUIA	Sistema Único de Información Ambiental
Business Intelligence	Se denomina inteligencia empresarial, inteligencia de negocios o BI (del inglés business intelligence) al conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.
Data warehouse	Es un repositorio de datos de muy fácil acceso, alimentado de numerosas fuentes, transformadas en grupos de información sobre temas específicos de negocios, para permitir nuevas consultas, análisis, reporteador y decisiones.
Sistema de gestión de contenidos	Un sistema de gestión de contenidos (en inglés Content Management System, abreviado CMS) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás roles.
Alfresco	Alfresco es un sistema de administración de contenidos libre, está diseñado para usuarios que requieren un alto grado de modularidad y rendimiento escalable.
ACL	Una lista de control de acceso o ACL (del inglés, access control list) es un concepto de seguridad informática usado para fomentar la separación de privilegios. Es una forma de determinar los permisos de acceso apropiados a un determinado objeto, dependiendo de ciertos aspectos del proceso para los ingresos al sistema.
Programación orientadas a objetos	La programación orientada a objetos o POO (OOP según sus siglas en inglés), es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. La cual es utilizada para el desarrollo de los módulos y aplicaciones dentro del sistema SAF v2.0
Codeigniter y Zend Framework	Son framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. La arquitectura utilizada por estos framework es modelo - vista – controlador (MVC)
MVC	Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

Estructura de módulos del SAF v.2



Este anillo de seguridad está diseñado dentro de una arquitectura de ACL del framework de Zend , Zend_Acl provee la implementación de un sistema simple y flexible de Listas de Control de Acceso (ACL) para la administración de privilegios. En general, una aplicación puede utilizar las ACL para controlar el acceso a ciertos objetos protegidos, que son requeridos por otros objetos.

Para los propósitos de esta documentación:

- Un recurso es un objeto al cual el acceso está controlado.
- Un rol es un objeto que puede solicitar acceso a un recurso.

En términos generales, Los roles solicitan acceso a los recursos. Por ejemplo, si una persona solicita acceso a un automóvil, entonces la persona se convierte en el rol solicitante, y el automóvil en el recurso, puesto que el acceso al automóvil puede no estar disponible a cualquiera.

Los servicios de cada nivel son:

- **Nivel Público (Público en general)**
 - Muestra información de contacto de las oficinas técnicas de acuerdo al lugar donde está ubicada la industria.
 - Despliega un directorio de negocios con un buscador de acuerdo a la ubicación y de la actividad forestal que realiza.
 - Muestra la base legal con que se sustenta el desarrollo del SAF e información de importancia para el sector forestal
 - Ayuda en línea mediante videos y un canal directo a los técnicos para solicitar ayuda o realizar alguna sugerencia o denuncia.

Privilegios otorgados a los usuarios de este nivel público:

Nombre	Permisos Individuales	Hereda permisos de
Invitado	Lectura	Ninguno

- **Nivel Registrado (Usuario registrado externo)**

Para este nivel se crea una cuenta única para cada actor que interviene en la cadena de valor de la madera al cual se lo conoce como **Usuario Externo** el procedimiento para crear esta cuenta se la realiza mediante el componente de Registro Nacional de Actividades Forestales (RNAF) y la información debe ser certificada por un funcionario de la oficina técnica (usuario interno) asignada de acuerdo a la ubicación donde se ejecuta la actividad.

Privilegios otorgados a los usuarios de este nivel registrado:

Nombre	Permisos Individuales	Hereda permisos de
Invitado	Lectura (html, pdf, hoja electrónica)	Ninguno
Usuarios Externos registrados en RNAF	<i>Crear, Actualizar y borrar:</i> Solicitudes de planes, programas de aprovechamiento, imprimir guías, solicitar certificados para exportar.	Invitado

- **Nivel administrativo (Funcionarios del MAE que realizan el seguimiento)**

Los Usuarios Internos son creados por el **Administrador DNF** en Planta Central por petición del Jefe Provincial de la Oficina Técnica mediante memorando generado a través del Sistema de Gestión Documental (Quipux).

Esta aspecto es restringido, solo considera funcionarios del Ministerio del Ambiente.

Privilegios otorgados a los usuarios de este nivel registrado:

Nombre	Permisos Individuales	Hereda permisos de
Invitado	Lectura (html, pdf, hoja electrónica)	Ninguno
Usuarios Externos registrados en el RNAF	<i>Crear, Actualizar y borrar:</i> Solicitudes de planes, programas de aprovechamiento, imprimir guías, solicitar certificados para exportar.	Invitado
Oficina técnica	Certificar información seguimiento y aprobación de solicitudes de planes y programas de aprovechamiento y emisión de certificado de exportación.	Propietario
Supervisores	Seguimiento y Verificación de Planes y programas	Invitado
Administrador - DNF	Crear usuarios, Seguimiento a nivel nacional y generar reportes gerenciales	Oficina técnica
Administrador del Sistema	(Todos los accesos permitidos)	Ninguno

Tablas del MÓDULO:

Tabla	Descripción
seg_recurso	Recursos del sistema
seg_role	Roles o perfiles de usuarios, tanto para usuarios externos o internos
seg_role_recurso	Relaciona los roles con los recursos
seg_usuario	Usuarios internos y externos que pueden ingresar al sistema
seg_usuario_role	Roles asignados a cada usuario

Referencia: <http://framework.zend.com/manual/en/zend.acl.html>

MÓDULO: Registro Nacional de Actividades Forestales RNAF

Este componente se desarrolla con el fin de estandarizar, actualizar y sistematizar el registro nacional de actividades económicas forestales a nivel nacional.

El Registro Forestal está amparado en el art. 102 de la codificación de la Ley Forestal y de Conservación de Áreas Naturales y Vida Silvestre, en el cual señala la obligatoriedad de la inscripción de las personas naturales o jurídicas que realicen actividades económicas forestales.

Por lo antes indicado los actores involucrados en la cadena de valor forestal, tendrán que acceder a la página web del Ministerio del Ambiente para generar en línea su solicitud de inscripción en el Registro Forestal y posteriormente la emisión del respectivo certificado.

Paralelamente el Ministerio del Ambiente con su implementación pretende cerrar el ciclo de control en el aprovechamiento de los recursos forestales, ya que en su aplicación se ha creado un mecanismo de registro y control que regulariza y supervisa el abastecimiento de productos forestales a las industrias forestales (pequeñas, medianas y grandes) y depósitos de madera.

Los actores quienes deben estar registrados y deben tener acceso al sistema SAF

Personas natural o jurídica que se dediquen a:

- Aprovechamiento y comercialización nacional e internacional de la madera.
- Industrias que realizan transformación primaria y secundaria de la madera incluyendo los depósitos que solo acopian madera.
- Delegados.
- Propietarios de predios con bosque nativo y/o plantaciones forestales y otras conexas.
- Viveros.
- Motosierrista
- Transportista de la madera

Tablas del MÓDULO:

Tabla	Descripción
rifcaractviveros	Características de vivero
rifcaratplantaciones	Características de la plantación
rifcartografia	Información de la cartografía del predio
rifcomentarios	Almacena los comentarios de cada solicitud
rifcontacto	Información de ubicación de los contactos de acuerdo al SRI
rifindustria	Información de industria primaria o secundaria.
rifmotosierrista	Almacena las características de las motosierras
rifpredio	Información del predio
rifpropietario	Tabla donde se almacena la información de los propietarios.
rifsolicitudes	Almacena todas las solicitudes con sus diferentes estados
riftransportista	Características del vehículo
rifviveros	Información de los viveros

MÓDULO: Gestión de planes, programas, licencias y guías

GPPLG

Tablas del MÓDULO:

Tabla	Descripción
adm_cri_eval_pl	Almacena los criterios de evaluación del Plan
adm_cri_eval_pr	Almacena los criterios de evaluación del Programa
adm_cri_eval_tpl	Relaciona el Tipo de Plan y los criterios de evaluación del Plan
adm_cri_eval_tpr	Relaciona el Tipo de Programa y los criterios de evaluación del Programa
adm_cri_evaluacion_plan	Relaciona el Plan de Manejo Integral y los criterios de evaluación del Plan
adm_cri_evaluacion_prog	Relaciona el Programa de Aprovechamiento y los criterios de evaluación del Programa
adm_datos_seguros	
adm_distancia	Almacena la información de las distancias entre cantones para obtener tiempos de recorridos
adm_doc_requeridos_plan	Documentos requeridos para Planes
adm_doc_requeridos_prog	Documentos requeridos para Programas
adm_doc_requeridos_tipo	Relaciona los Tipos de documentos requeridos con los Planes
adm_doc_requeridos_tipo_pr	Relaciona los Tipos de documentos requeridos con los Programas
adm_guia_balsa_pigue	Almacena información de Guías de Balsa y Pigüe

adm_guia_canje	Tabla cabecera de Guía de Canje
adm_guia_canje_detalle	Guías de Circulación adjuntas a la Guía de Canje
adm_guia_especies	Información de especies en Guía de Movilización
adm_guia_movilizacion	Almacena datos de Guía de Movilización
adm_guia_no_maderable	Almacena datos de Guía no Maderable
adm_guia_seguridad	Asignación de especies valoradas a programas
adm_historia_bloqueos	Almacena el historial de las personas o procesos bloqueados en el sistema
adm_licencia	Información de Licencias
adm_licencia_especies	Información de especies por licencias
adm_motivo_bloqueo	Almacena información de los motivos de bloqueo que se utilizaran para bloquear propietarios, ejecutores, regentes forestales, programas, etc.
adm_pago_general	Información de pagos
adm_pagos	Almacena el catalogo de los diferentes pagos con sus respectivos valores
adm_pagos_licencia	Información de pagos de licencias
adm_pagos_tipo_pr	
adm_plan_manejo	Almacena los Planes de Manejo Integral
adm_predio	Almacena los predios de Planes y Programas creados
adm_producto_madera	Información de tipo de productos maderables
adm_programa	Almacena los Programas de Aprovechamiento
adm_programa_especies	Almacena la información de las especies en determinado Programa

adm_propietario	Esta tabla guarda los datos del propietario y del ejecutor
adm_regente_forestal	Lista de Regentes Forestales
adm_tipo_destino	Tipos de destino de productos
adm_tipo_persona	Tipo de persona (Natural o Jurídica)
adm_tipo_plan	Tipo de Planes
adm_tipo_prog	Tipo de Programas
adm_zonificacion	Información de tipos de zonificación
adm_zonificacion_plan	Zonificación de Planes
adm_zonificacion_tipo	Relaciona el tipo de zonificación y los planes
crl_acciones_legales	Control de acciones legales
crl_acta_retencion	Control de acta de retención
crl_comprobar_guia	Control para comprobar Guía
crl_denuncia_regente	Control de denuncia de regente
crl_historial_retencion	Control de Historial de Retención
crl_informe_campo	Control de Informe de Campo
crl_informe_regente	Control de Informe de Regente
crl_procesos_legales	Control de Procesos Legales
crl_retencion_especies	Control de Especies Retenidas
crl_verificacion_guia	Control de Guías verificadas
des_guia_balsa_pigüe	Destino de guía de Balsa y Pigüe
det_guia_balsa_pigüe	Guía detenida de Balsa y Pigüe
det_guia_no_maderable	Guía detenida de Producto no Maderable

MÓDULO: Importaciones y exportaciones de productos forestales

IEPF

Con propósito de agilizar los trámites de certificación para exportar e importar productos forestales maderables y no maderables, este módulo está integrado con el sistema de ventanilla única para comercio exterior. Ventanilla única es una plataforma desarrollada por el Ministerio Coordinador de Producción, Empleo y Competitividad para uso de todas las instituciones públicas en sus procedimientos de exportación e importación.

Levanta los procedimientos a efectuarse por las oficinas técnicas del MAE y las enlaza con los requerimientos de la SENAE, Servicio Nacional de Aduanas del Ecuador.

Tablas del módulo:

Tabla	Descripción
cex_administracion	Gestiona las solicitudes de exportaciones e importaciones, es la tabla principal
cex_detalle	Almacena los detalles de productos, especies dimensiones, volumen de cada solicitud
cex_solicitante	Almacena los datos del solicitante de la autorización
cex_importador	Almacena los datos del importador de los productos
admcatalogos	A esta tabla se añadieron registros para que los pueda tomar el procedimiento de exportaciones e importaciones.
rif_contacto, admdivpolitica, admoficinastecnicas, etc	Las tablas antes mencionadas se relacionan con estas tablas ya existentes e

MÓDULO: Destino final de la madera

DFM

Para realizar un control eficaz y garantizar que la madera existente en los establecimientos comerciales provenga del aprovechamiento de bosques y plantaciones autorizados por el Ministerio del Ambiente; el SAF enlaza el aprovechamiento en campo y su respectiva movilización con el destino final (industria y depósitos), para ello en el sistema se ha creado un panel con cuatro formularios de registros de ingreso, origen y guías asignados a los destinos finales.

Para la *adquisición de madera* un destino final deberá aceptar la asignación de madera a través de la generación de una guía de movilización a su favor. En lo que respecta al volumen de madera, el propietario del destino final tiene que aceptar la asignación de madera, conforme el tipo de producto, especie, dimensiones y cantidad al aceptarlo el sistema automáticamente se enlaza con la base de datos del SAF estableciendo una relación directa con el origen de esa madera, por tipo de programa, número de la licencia, y demás información respecto al origen de esa madera.

Nro. Guía	Fecha de ingreso de la guía	Código	Tipo de producto	Especie	Dimensiones			Cantidad	Volumen ingresado (m ³)
					Largo (cm)	Ancho/diámetro (cm)	Espe-sor (cm)		
24CC D11	30-04-2010	1233	Tablones	Canelo	300	25	5	200	7,50

En lo concerniente al registro de *ventas de madera* que realiza la industria o depósito, deberá ingresar el mes de ventas, la especie y reportar el volumen mensual vendido:

Registro del mes de ventas	Especie	Volumen vendido mensual (m ³)
Marzo	Laurel	23
Mayo	Canelo	2

El registro de la madera transformada en stock se realizará mensualmente, donde la industria o depósito ingresa el tipo de producto terminado, cantidad, especie y volumen utilizado en la elaboración del producto, determinado éste en función de las dimensiones y cantidad de las piezas (tablas, tablones, etc.) según sea el caso.

Tipo de producto acabado	Cantidad	Especie	Volumen transformado en stock (m ³)
Puertas	5	Canelo	15

Para conocer el saldo actual de madera, el sistema opera automáticamente y realiza los cálculos según las compras, ventas y madera transformada, el volumen inicial será el que se hizo constar en el kardex de descargo elaborado por el supervisor forestal. Esto beneficia a la industria o depósito, permitiéndole llevar un autocontrol del producto forestal.

Saldo kardex inicial (m ³)	Especie	Volumen comprado (m ³)	Volumen vendido (m ³)	Volumen transformado en stock (m ³)	Saldo actual o a la fecha de evaluación (m ³)
90	Laurel		23		67
200	Canelo	7,5	2	15	220,5

Tablas del MÓDULO:

Este MÓDULO se encuentra en la etapa de socialización del diseño para recoger nuevos requerimientos y modelar la base de datos final.

MÓDULO: Administración de procesos legales

APL

Este componente es necesario para dar seguimiento y control a los procesos legales iniciados por infringir el actual régimen forestal. Permite ubicar rápidamente un caso en función a criterios de búsqueda, como: tipo de proceso, estado procesal, sujeto procesal, origen, medidas cautelares, abogado responsable y otros.

Con este componente el SAF pretende:

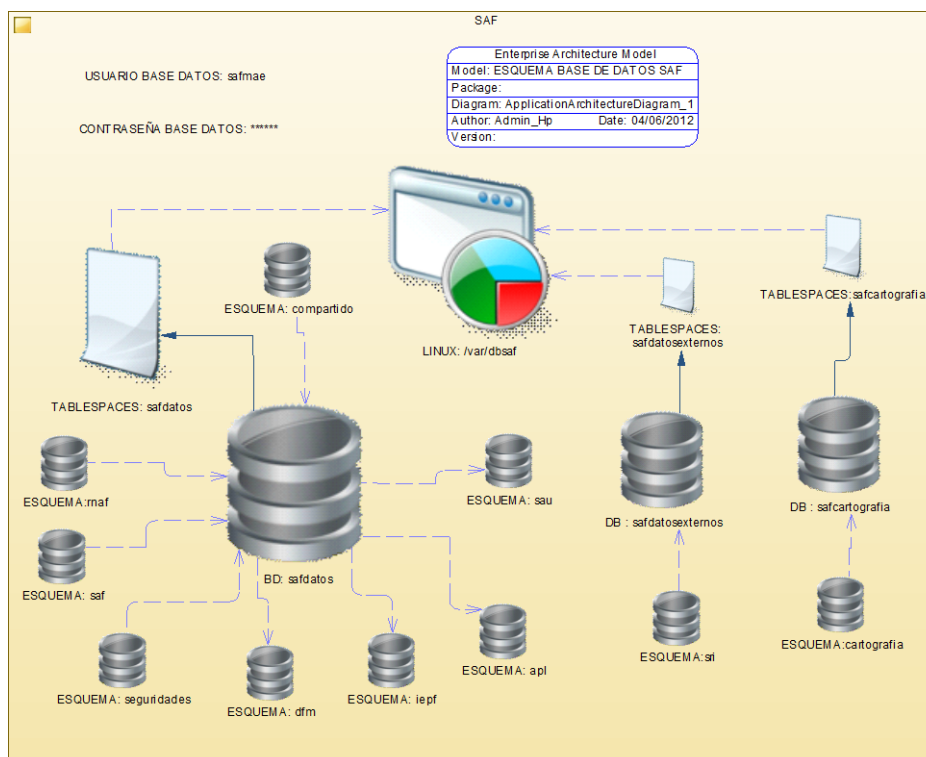
- Mejorar la administración y seguimientos de casos.
- Encontrar información rápida y oportuna.
- Emitir reportes de seguimiento.

Tablas del componente:

Este componente se encuentra en la etapa de análisis para poder integrarse con el proceso similar que maneja el SUIA.

Diagrama general de base de datos del SAF

Todos estos MÓDULOS almacenan información en la base de datos de acuerdo a la siguiente estructura



Módulo del Sistema de Información Forestal

MÓDULO: Sistema de gestión de contenido

CMS

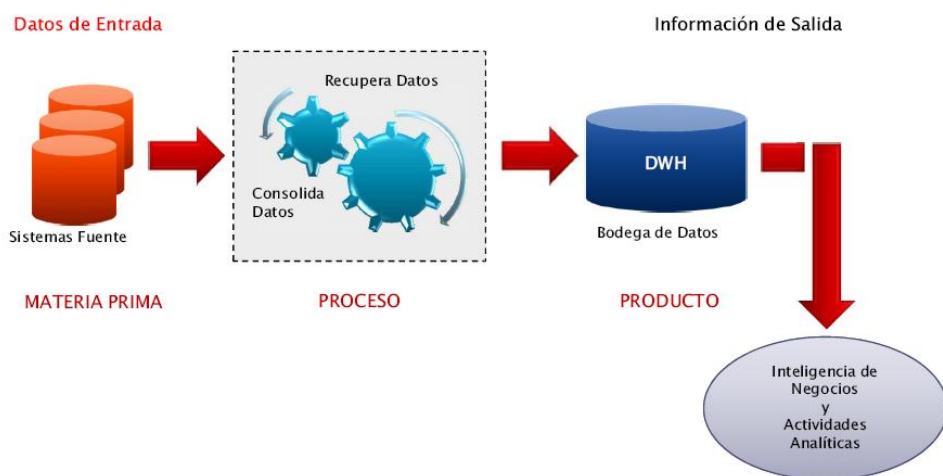
Este componente se encuentra en la etapa de análisis conjuntamente con el MÓDULO del SUIA, para los documentos que subidos al sistema SAF sean administrados desde la misma aplicación de Alfresco.

MÓDULO: Componentes De Información Forestal

CIG

Para el desarrollo de este componente se utiliza la herramienta Pentaho el cual es una plataforma de BI “orientada a la solución” y “centrada en procesos”.

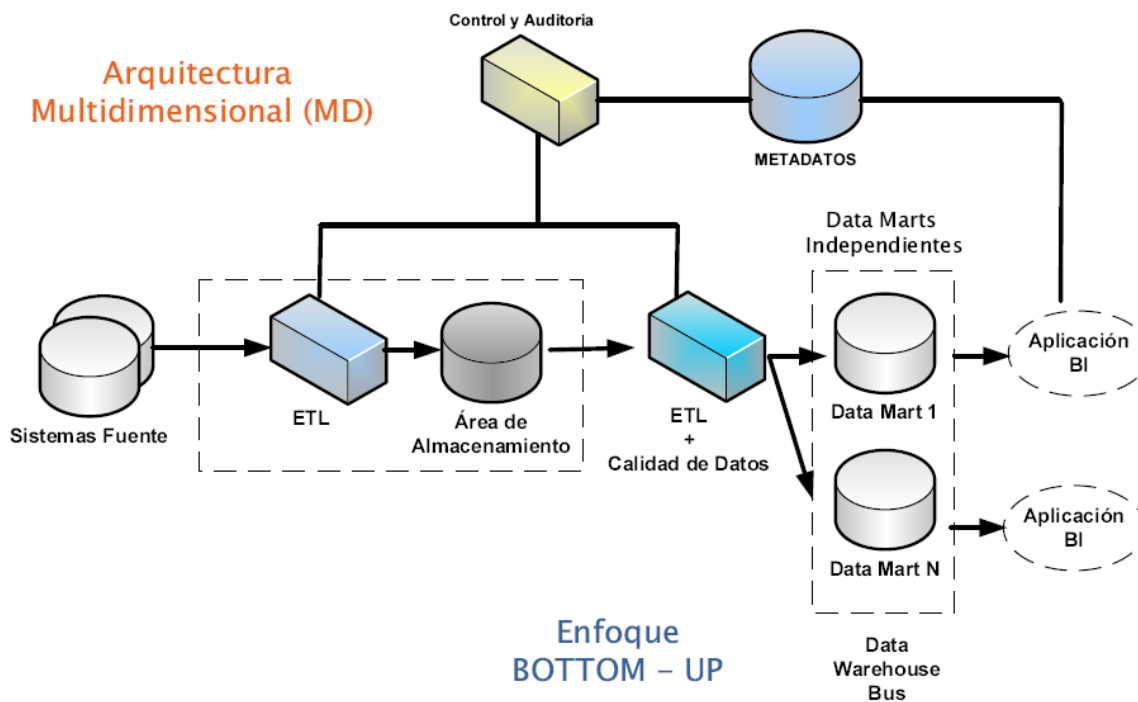
Data Warehousing



En este componente el proceso es recoger información de varias fuentes y diferentes repositorios como se muestra en la figura para luego ser consolidada en una bodega de datos (DWH), para su fácil utilización en analizar y crear reportes complejos.

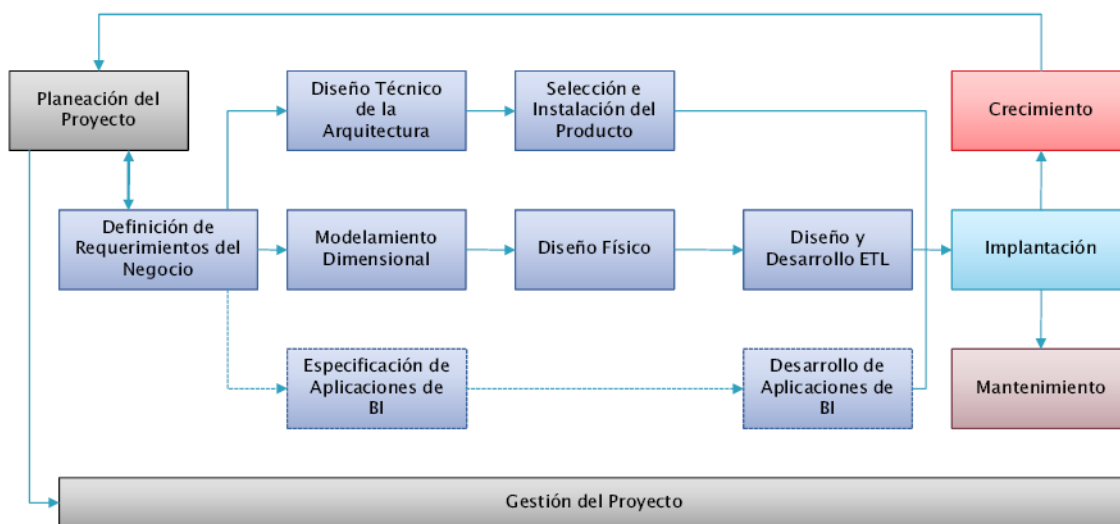
Con esta arquitectura se logra lo siguiente:

- Toda la información está un solo lugar (Una sólo fuente de la verdad)
- Información actualizada
- Acceso rápido
- Contiene información histórica
- Fácil de comprender.
- Contiene definiciones claras y uniformes
- Datos estandarizados



La arquitectura Multidimensional utilizada para el desarrollo de este componente es de Ralph Kimbal como presenta en la figura, que consiste en definir un almacén de datos como: "una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis". También se determina varios Data Mart para componer un datawarehouse .

Metodología



La Planeación del Proyecto busca identificar la definición y el alcance del proyecto del DWH, incluyendo las justificaciones de la implementación y las evaluaciones de factibilidad.

La definición de los requerimientos es un factor determinante en el éxito de un proceso de DWH es la interpretación correcta de los diferentes niveles de requerimientos expresados por los distintos grupos de usuarios.

Modelado Dimensional, determina los datos necesarios para cumplir los requerimientos analíticos de los usuarios.

Diseño físico, es un elemento principal de este proceso es la definición de estándares del entorno de la base de datos.

Diseño y Desarrollo del ETL, Las principales actividades de esta fase del ciclo de vida son: la extracción, la transformación y la carga (ETL process). Se definen como

procesos de extracción aquellos requeridos para obtener los datos que permitirán efectuar la carga del Modelo Físico diseñado.

El Diseño técnico de la Arquitectura, Se deben tener en cuenta tres factores: los requerimientos del negocio, los actuales entornos técnicos y las directrices técnicas y estratégicas futuras planificadas por la organización para poder establecer el diseño de la arquitectura técnica del entorno de DWH.

Selección e instalación del producto, una vez evaluados y seleccionados los componentes determinados, se procede con la instalación y prueba en un ambiente integrado.

Especificación de Aplicaciones de BI, en esta etapa se identifican los roles o perfiles de usuarios para los diferentes tipos de aplicaciones necesarias en base al alcance de los perfiles detectados (autoridades, analistas, usuarios en general)

Desarrollo de aplicaciones de BI, donde acceden al DWH por medio de herramientas de productividad basadas en GUI (Graphical User Interface). Las herramientas pueden incluir software de consultas, generadores de reportes, procesamiento analítico en línea o herramientas de Datamining dependiendo de los tipos de usuarios y sus requerimientos particulares.

Implantación: representa la convergencia de la tecnología, los datos y las aplicaciones de usuarios finales accesibles para el usuario del portal de información forestal.

Mantenimiento y Crecimiento: se remarca siempre, la creación de un DWH es un proceso (de etapas bien definidas, con comienzo y fin, pero de naturaleza espiral) que acompaña a la evolución del sistema durante toda su historia. Se necesita continuar con las actualizaciones de forma constante para poder seguir la evolución de las metas por conseguir.

Definición de Requerimientos

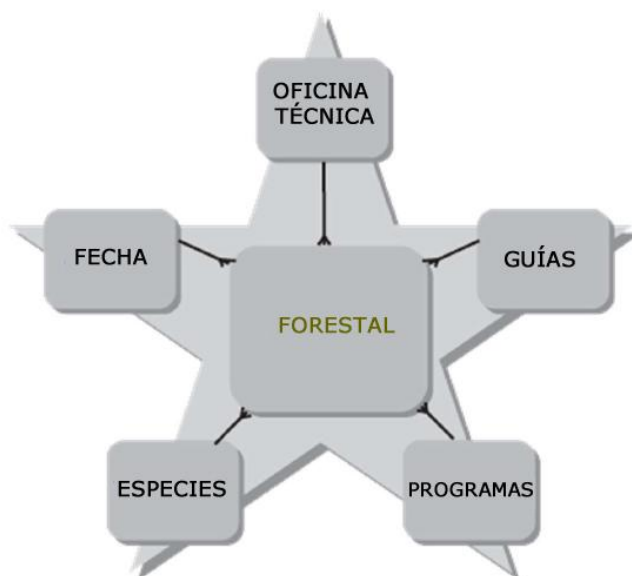
Conocer las fuentes de datos, alcance, preguntas del negocio a responder, formatos de entrega de información, etc.

Por lo menos en ésta fase documentar lo siguiente:

Entrada	Descripción
Tema	Área o proceso de negocio
Audiencia	Para quién es la solución
Propietario	Quién será el propietario de la solución
Necesidad del Usuario	Qué necesitan los usuarios y cómo usarán la solución
Preguntas a responder	Preguntas que serán respondidas por la solución
Beneficios del negocio	Qué ganará el negocio construyendo el DWH
Mecanismo de entrega	Emails, Gráficos, Reportes, Tableros de Mando, Cubos de Datos, etc.
Fuentes de información	A qué personas se puede preguntar sobre el negocio
Fuentes de datos	De qué sistemas y qué bases de datos se obtendrán los datos
Estimación	Estimación de tiempo para desarrollar la solución

Modelamiento

El modelo utilizado es el modelo estrella, consisten en tener una tabla conocida como tabla de HECHOS, y los puntos de la estrella son las tablas de DIMENSIONES, como se muestra en la siguiente figura.



Integración de datos.

En esta etapa se realiza las actividades para llenar la bodega de datos

- La integración de datos se descompone 3 actividades
 - Extracción: Adquirir datos de 1 o más fuentes T
 - Transformación: Cambiar la forma y contenido de los datos
 - Loading (Carga): cargar los datos en los repositorios y el DWH

Herramientas utilizadas para crear esta arquitectura son:

Herramienta	Versión	Descripción
Pentaho BI Server	3.8.0	Servidor Central de Inteligencia de Negocios, Portal Web de Entrega de Información
Pentaho Design Studio (PDS)	4.0.0	Organización de soluciones, creación de xaction (Acciones de Secuencias)
Pentaho Data Integration (PDI)	4.2.1	Creación y ejecución de procesos ETL
Pentaho Metadata Editor (PME)	4.1.0	Creación y publicación de metadatos para reportes
Pentaho Report Designer (PRD)	3.8.2	Creación y publicación de reportes
Pentaho Schema Workbench (PWS)	3.3.0	Creación y publicación de cubos para análisis OLAP

Referencias:

<http://www.alfresco.com/es/>

<http://www.pentaho.com/>

ESTÁNDARES DE CODIFICACIÓN

Estándar	Descripción
Sistema operativo	Linux
Lenguaje	PHP
Base de datos	Postgres
Arquitectura	MVC
Framework	Codeigniter y Zend Framework

Arquitectura MVC

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.



Un diagrama sencillo que muestra la relación entre el modelo, la vista y el controlador

Codeigniter y Zend Framework

Tanto Codeigniter y Zend Framework son framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. La arquitectura utilizada por estos framework es modelo - vista - controlador

Temas incluidos en los estándares de código:

- Dar formato a archivos PHP
- Convenciones de nombrado
- Estilo de código
- Documentación integrada

Los estándares de código resultan importantes en cualquier proyecto de desarrollo, pero son especialmente importantes cuando muchos desarrolladores trabajan en el mismo proyecto. Los estándares de código ayudan a asegurar que el código tenga una alta calidad, menos errores, y pueda ser mantenido fácilmente.

Formato de archivos PHP

Todos los nombres de archivos de soporte y código fuente van en minúsculas

No usar CAMELCASE (ejemplo: NombreFuncion), **usar** guion bajo (ejem: nombre_funcion)

version de php usada: PHP 5.3.5

Contenido de Directorios

assets

Todos los archivos no php van en el directorio assets:

assets \
 \js (Todos los scripts de javascript)
 \css (Todas las hojas de estilos)
 \img (Todas las imagenes)

Todos los nombres de archivos van en minúsculas y sin espacios, el nombre del archivo debe tener relación con la vista donde se lo llama, salvo el caso que sea de una librería externa al programa. Por ejemplo:

```
ema_index_view.php -> assets\js\ema_index_view.js
```

El único archivo que contiene código css personalizado para el saf2 es:

```
assets\css\saf2.css
```

Los demás archivos css son librerías de soporte para la interfaz gráfica del saf2.

Código PHP de modelos vistas y controladoras

Todos el código fuente va en las siguientes carpetas

```
application \
  \modules
    \nombre_MÓDULO1
      \controllers
        \controller1.php
        \controller2.php
      \models
        \nombre_tabla1_model.php
        \nombre_tabla2_model.php
      \views
        \controller1_header_view.php
        \controller1_footer_view.php
        \controller1_index_view.php
        \controller1_listar_view.php
        \controller1_guardar_view.php
```

Para la generación de tablas, enlaces, formularios, usar en la medida de lo posible:

- Para tablas, la table class de codeigniter (Las tablas de las genera en el controlador y se la pasa como parámetro a la vista)
- Para enlaces anchor y en general las funciones de url helper

- Para formulario usar el form helper

Controladores

Los controladores tienen el nombre del submódulo al que corresponden, por ejemplo:

pagos.php

Si está formado por dos o más palabras usar el guión bajo en el nombre:

causales_retencion.php

Las funciones estándar en el controlador son:

```
private $MÓDULO = 'nombre_MÓDULO->';

public function __construct()
{
    parent::__construct();
    // Acl
    $this->my_login->esta_logeado($this->MÓDULO.$this->router->class,
        $this->router->method, '/login/index');

    //validaciones
    $this->form_validation->set_rules('crt_estado','Estado Causal
        Retencion','required');
}

public function index()
{
}

public function guardar()
{
}
```

```
public function modificar($id = NULL)
{

}

public function listar($pagina = NULL)
{

}

public function eliminar($ID_DE_LA_TABLA)
{

}

public function buscar($pagina = NULL)
{

}

// Funciones customizadas del controlador

public function nombre_funcion()
{

}

public function nombre_funcion2()
{

}
```

Los nombres de las funciones deben ser verbos o palabras que documenten su finalidad por ejemplo:

```
public function listar_regentes_bloqueados()

public function evaluacion_tecnica()
```

Modelos

Los nombres de los modelos tienen el nombre de la tabla a la que representan

agregando el sufijo _model por ejemplo:

adm_zonificacion_model.php

usar el guión bajo si son más de dos palabras.

Plantilla de Modelo:

```
<?php
/**
 * adm_regente_forestal_model.php
 *
 * @package   Package Name
 * @subpackage Subpackage
 * @category  Category
 * @author    ronaldpatino
 */
class Adm_regente_forestal_model extends CI_Model{

    // Campos de la tabla

    public $reg_id           = "";
    public $reg_ci_ruc       = "";
    public $reg_apellidos   = "";
    public $reg_nombres     = "";
    public $reg_direccion   = "";
    public $reg_telefono    = "";
    public $reg_celular     = "";
    public $reg_email       = "";
    public $reg_registro_forestal = "";
    public $reg_emision_aval = "";
    public $reg_caduca_aval = "";
    public $reg_ip_crea     = "";
    public $reg_fecha_crea  = "";
    public $reg_usuario_crea = "";
    public $reg_ip_modifica = "";
    public $reg_fecha_modifica = "";
    public $reg_usuario_modifica = "";
    public $reg_estado     = "";

    /**
```

```
* Constructor de la clase
*
* @param void
* @return void
*/
public function __construct()
{
    // Call the Model constructor
    parent::__construct();
}

/**
 * Funcion que guarda los datos pasados como parametro la
 * en la tabla
 *
 * @param array $reg_datos
 * @return void
 */
public function save($reg_datos)
{
    $this->db->insert('saf.adm_regente_forestal', $reg_datos);
}

/**
 * Funcion nos da el siguiente valor de la tabla de
 * secuencia
 *
 * @param string $secuencia
 * @return int $nextval
 */
public function nextval($secuencia = ' saf.seq_id_regente_forestal ')
{
    $nextval = FALSE;

    $sql = "SELECT nextval('".$secuencia."')";
    $query = $this->db->query($sql);

    if ($query->num_rows() > 0)
    {
        $row = $query->row();
        $nextval = $row->nextval;
    }
}
```

```
    }
    return $nextval;
}

/**
 * Funcion que retorna un array de todos los items de la tabla
 * si pasamos start y count nos permite paginar
 *
 * @param int $start
 * @param int $count
 * @return array
 */
function findAll($start = NULL, $count = NULL)
{
    return $this->find(NULL, $start, $count);
}

/**
 * Funcion generica que retorna un array de objetos de los items de la
 * tabla en la base de datos
 *
 * @param array $filters
 * @param int $start
 * @param int $count
 * @return array
 */
function find($filters = NULL, $start = NULL, $count = NULL) {

    $results = array();

    // Load the database library
    $this->load->database();
    //$this->db->select('ume_id','ume_nombre','ume_abreviacion');
    $this->table_record_count = $this->db->count_all( 'saf.adm_regente_forestal' );

    // Filter could be an array or filter values or an SQL string.
    $where_clause = "";
    if ($filters) {
        if ( is_string($filters) ) {
            $where_clause = $filters;
        }
        elseif ( is_array($filters) ) {
```



```
        if ( count($filters) > 0 ) {
            foreach ($filters as $field => $value) {
                $this->db->where($field, $value);
            }
        }
    }

}

if ($start) {
    if ($count) {
        $this->db->limit($start, $count);
    }
    else {
        $this->db->limit($start);
    }
}

$query = $this->db->get( 'saf.adm_regente_forestal' );

if ($query->num_rows() > 0)
{
    return $query->result();
}
else
{
    return FALSE;
}
}

/**
 * Funcion que llama a $this->find pasando como parametros
 * una lista de condiciones a filtrar
 *
 * @param array $filter_rules
 * @param int $start
 * @param int $count
```

```
* @return array
*/
function findByFilter($filter_rules, $start = NULL, $count = NULL) {
    return $this->find($filter_rules, $start, $count);
}

/**
 * Funcion que actualiza la tabla
 * con los valores pasados como parametros
 *
 * @param array $filter_rules
 * @param array $data
 * @return void
 */
function update($keyvalue, $data) {

    // Load the database library
    $this->load->database();

    $this->db->where('reg_id', $keyvalue);

    $this->db->update('saf.adm_regente_forestal', $data);

}

/**
 * Funcion que borra datos de la la tabla
 * con el id pasado como parametros
 *
 * @param array $filter_rules
 * @param array $data
 * @return boolean
 */
function delete_by_pkey($idField) {

    // Load the db library
    $this->load->database();

    $this->db->where('reg_id', $idField);
    $this->db->delete('saf.adm_regente_forestal');
```

```
        return true;
    }

/**
 * Funcion que cuenta los items en la tabla
 *
 * @param array $filter_rules
 * @param array $data
 * @return void
 */
function count_usuarios()
{
    return $this->db->count_all_results('saf.adm_regente_forestal');
}

/* End of file adm_regente_forestal_model.php */
/* Location:
./application/modules/seguimientos/model/adm_regente_forestal_model.php */
```

Todas las consultas sql deben tener el siguiente formato:

Palabras reservadas de SQL van en una sola línea y en mayúsculas, las variables en otra línea y en minúsculas, ejemplo:

```
$sql = "SELECT
    oft_id,
    oft_nombre,
    reg_nombre
FROM
    saf.seg_oficina_tecnica,
    saf.seg_regional
WHERE
    saf.seg_oficina_tecnica.reg_id = saf.seg_regional.reg_id
ORDER BY
    reg_nombre, oft_nombre";
```

Vistas

En el caso de las vistas se crea una carpeta dentro de la carpeta views con el nombre del submódulo al que corresponden:

```
views \  
  \causales_retencion  
    \crt_index_view.php  
    \crt_listar_view.php
```

dentro de esta carpeta creamos los archivos de vistas

Las vistas tienen como prefijo dos o tres letras del nombre del controlador:

causales_retencion.php -> **crt_index_view.php**

Las vistas tienen el sufijo **_view**

crt_index_**view**.php

Las vistas deben tener el nombre de la función que las llama por ejemplo:

function index() - > crt_**index**_view.php

function listar() - > crt_**listas**_view.php

Formularios

Para la creación de formularios hacer uso de la siguiente aplicación para generar código:

<http://sprawsm.com/uni-form/builder/>

Copiar el código generado y colocarlo en la vista personalizada los valores necesarios

Ejemplo de un formulario:

```
<form class="uniForm" >
  <div class="header">
    <h2>UNIDAD DE MEDIDA</h2>
    <p>Crear Unidad de Medida</p>
  </div>

  <fieldset class="inlineLabels">
    <div class="ctrlHolder">
      <label for="">Etiqueta del Control</label>
      <input type="text" class="textInput" maxlength="13" size="13" id="ci_ruc"
value="" name="ci_ruc">
      <p class="formHint">Información del Control</p>
    </div>

    <div class="buttonHolder">

      <a class="secondaryActionLink" title="←Listado de Unidades de Medida"
href="http://localhost/saf2/index.php/catalogos/unidad_medida/listar">&larr
; Listado de Unidades de Medida</a>

      <button class="primaryAction" type="submit"
name="modificar">Grabar Unidad de Medida</button>
    </div>
  </fieldset>

</form>
```

O puede ser una variante en la clase del fieldset, que la dejamos vacía, para ver los resultados podemos usar : <http://sprawsm.com/uni-form/builder/> para ver el resultado si quitamos

```
<form class="uniForm" >
  <div class="header">
    <h2>UNIDAD DE MEDIDA</h2>
    <p>Crear Unidad de Medida</p>
  </div>
```

```
<fieldset class="">
  <div class="ctrlHolder">
    <label for="">Etiqueta del Control</label>
    <input type="text" class="textInput" maxlength="13" size="13"
id="ci_ruc" value="" name="ci_ruc">
    <p class="formHint">Información del Control</p>
  </div>

  <div class="buttonHolder">

    <a class="secondaryActionLink" title="Listado de Unidades de Medida"
href="http://localhost/saf2/index.php/catalogos/unidad_medida/listar">&larr
; Listado de Unidades de Medida</a>

    <button class="primaryAction" type="submit"
name="modificar">Grabar Unidad de Medida</button>
  </div>

</fieldset>

</form>
```

El header que va dentro del form contiene:

```
<h2>NOMBRE DEL MENU PADRE</h2>
<p>Nombre del Item Actual del Menu</p>
```

- Nombre del menú padre va siempre en mayúsculas
- Nombre del Item Actual va siempre en CamelCase

Para enlaces de botones en la parte inferior el contenedor es:

```
<div class="buttonHolder">
</div>
```

Cuando tenemos un enlace que regresa a la pantalla padre usamos

```
<a class="secondaryActionLink" title="Nombre de la Pantalla a la que Regresa"
```

```
href="http://localhost/saf2/index.php/catalogos/unidad_medida/listar">&larr;  
LNombre de la Pantalla a la que Regresa</a>
```

Los botones tienen el siguiente código:

```
<button class="primaryAction" type="submit" name="modificar">Grabar Unidad de  
Medida</button>
```

Siempre usar un verbo en la acción que hace el botón.

Para los nombres de atributos id y name en el fomulario usar el nombre del campo en la base de datos.

Referencias:

http://codeigniter.com/user_guide/

<http://manual.zfdes.com/es/coding-standard.html>

Ministerio del Ambiente

Dirección Nacional Forestal

Dirección: Madrid y Andalucía cuarto piso

Teléfono: 00-593-30987600 ext. 1412

e-mail: dirección.forestal@ambiente.gob.ec

Descargo de responsabilidad:

Coordinadora del proyecto: Estefanía Arias

Elaborado por: Álvaro Sánchez