



30-10-2015

MANUAL TECNICO ADMINISTRACION

**Sistema de atención al
contribuyente**

Diana Umanzor
INFOLOGIC



MANUAL TECNICO MODULO DE ADMINISTRACION

Contenido

1. Breve descripción del Módulo de Administracion	2
2. Diagrama Entidad – Relación	3
3. Estructura de Componentes (Capa vista, controlador, servicio DAO y estructura de reportes) 4	
3.1 Controladores.....	7
3.2 Descripción de controladores	8
4. Descripción de POJOS de sesión	39
4.1 Descripción de Clases POJO	41
5. Listado de Repositorios.	49
5.1 Descripción de Repositorios.....	51
6. Listado de Componentes.	78
Descripción de Componentes	78
7. Configuración de proyectos (@Configuration y @Beans en general, esto es todo el paquete de sv.gob.mh.dgii.det.config)	79
8. Documentos Relacionados a la Base de Datos	83
8. Seguridad.....	103
1. Anexos	105



1. Breve descripción del Módulo de Administracion

Prefijo: /admin/

Modulo gestiona el funcionamiento de la administración (Mantenimiento) de los diferentes elementos que componen la parametrización de las Zonas (Áreas), alertas, configuración de los Trámites, los equipos, Escritorios, evaluación, Llamado, multimedia, prioridades, Reasignación, reproducción multimedia, Servicios, Trámites, Usuarios, también administra el formato que tendrán los tickets.



MANUAL TECNICO MODULO DE ADMINISTRACION

2. Diagrama Entidad – Relación

A continuación se listan las diferentes tablas utilizadas en el Módulo de Administración, junto con los nombres de los esquemas a donde pertenecen.

Nombre del esquema	Nombres de tablas	Anexo
Gestion.Cola	GC_ESCRITORIO GC_TRAMITE GC_USUARIO GC_ALERTA GC_ZONA GC_CONF_LLAMADO GC_CONF_TIQUETE GC_CONF_TRAMITE GC_EQUIPOS GC_MULTIMEDIA GC_PREGUNTAS GC_RESPUESTAS GC_PRIORIDAD GC_SERVICIOS GC_TIPO_EQUIPO GC_TRAMITE GC_CONF_TRAMITE	Anexo 1
CATALOGOS	TB_UNIDAD_RECEP TB_LISTAS_VALOR	Anexo 2

Tabla 2.1 tablas con sus esquemas respectivos



MANUAL TECNICO MODULO DE ADMINISTRACION

3. Estructura de Componentes (Capa vista, controlador, servicio DAO y estructura de reportes)

Vistas o páginas .jsp

En la siguiente tabla se muestran todas las páginas .html usadas en el Módulo de Administracion y el lugar en donde se pueden encontrar.

Nombre de opción	Ubicación	Nombre
Configuracion de Usuarios	src/main/resources/templates.admin	confiUsuarios.html form.html index.html verUsuario.html
Mantenimiento de Alertas	src/main/resources/templates.alerta	form.html index.html show.html
Mantenimiento de Configuración de Llamado	src/main/resources/templates.cllamado	cllamados.html newCllamado.html verCllamado.html
Configuracion de Tiquete	src/main/resources/templates.confqtiquete	confitiq.html
Mantenimiento de Configuración de Tramite	src/main/resources/templates.ctramite	ctramites.html newCtramite.html verCtramite.html
Mantenimiento de Equipos	src/main/resources/templates.equipo	equipos.html newEquipo.html verEquipo.html
Mantenimiento de Escritorios	src/main/resources/templates.escriptorio	editEscritorio.html



MANUAL TECNICO MODULO DE ADMINISTRACION

Nombre de opción	Ubicación	Nombre
		escritorios.html newEscritorio.html verEscritorio.html
Mantenimiento de Lista de Reproducción	src/main/resources/ templates.confrepro	Confrepro.html
Mantenimiento de Multimedia	src/main/resources/ templates.multimedia	editMultimedia.html multimedia.html newMultimedia.html verMultimedia.html
Mantenimiento de Configuración - Evaluación	src/main/resources/ templates.adminpregunt as	newPregunta.html preguntas.html verPregunta.html
Mantenimiento de Prioridades - Catálogo	src/main/resources/ templates.prioridad	newPrioridad.html prioridades.html verPrioridad.html
Mantenimiento de Secciones - Catálogo	src/main/resources/ templates.servicio	newServicio.html servicios.html verServicio.html
Mantenimiento de Tipos de Equipos - Catálogo	src/main/resources/ templates.tequipo	newTequipo.html tequipo.html verTequipo.html
Mantenimiento de Tramites - Catálogo	src/main/resources/ templates.tramite	newTramite.html tramites.html verTramite.html



MANUAL TECNICO MODULO DE ADMINISTRACION

Nombre de opción	Ubicación	Nombre
Mantenimiento de Zonas - Catálogo	src/main/resources/ templates.zona	zonas.html
		editZona.html
		newZona.html
		verZona.html
Configuración de Reasignación	src/main/resources/ templates.creassignacion	creassignaciones.html
		newCreassignacion.html
		verCreassignacion.html

Tabla 3.1 nombres y ubicación de las páginas .jsp



MANUAL TECNICO MODULO DE ADMINISTRACION

3.1 Controladores

En la siguiente tabla se muestran todas las clases controladores que se usan en el modulo de Administracion y la ubicación de los controladores que se encuentran en el paquete **sv.gob.mh.dgii.colas.controllers.admin** en donde se encuentran.

Nombre de Opción	Nombre
Configuracion de Usuarios	AdminController.java
Mantenimiento de Alertas	AlertaController.java
Mantenimiento de Configuración de Llamado	ConfLlamadoController.java
Configuracion de Tiquete	ConfTiqueteController.java
Mantenimiento de Configuración de Tramite	ConfTramiteController.java
Mantenimiento de Equipos	EquipoController.java
Mantenimiento de Escritorios	EscritorioController.java
Mantenimiento de Lista de Reproducción	MultimedServicioController.java
Mantenimiento de Multimedia	MultimediaController.java
Mantenimiento de Configuración - Evaluación	PreguntaEvaluacionController.java
Mantenimiento de Prioridades - Catálogo	PrioridadController.java
Mantenimiento de Secciones - Catálogo	ServicioController.java
Mantenimiento de Tipos de Equipos - Catálogo	TipoEqueipoController.java
Mantenimiento de Tramites - Catálogo	TramiteController.java
Mantenimiento de Zonas - Catálogo	ZonaController.java
Configuración de Reasignación	ReasignacionController.JAVA

Tabla 3.2 nombres y ubicación de los controladores



MANUAL TECNICO MODULO DE ADMINISTRACION

3.2 Descripción de controladores

En las siguientes tablas se describen brevemente cada uno de los métodos de los diferentes controladores usados en el Módulo de Administración.

PANTALLA CONFIGURACION DE USUARIOS	
Controlador:	AdminController.java
Descripción de Métodos de Negocio	
public UsuarioPojo usuariopojo(); Método que crea una instancia a la clase UsuarioPojo e invoca los métodos prepareList() y getUnidad(). @RequestMapping("") public String iniciar_usuarios(HttpServletRequest request); Método que carga la url redirect:/admin/ @RequestMapping(value = "/", method = RequestMethod.GET) public String index(HttpServletRequest request, @ModelAttribute UsuarioPojo usuariopojo, final ModelMap model); Método de tipo String que obtiene una lista de todos usuarios de la entidad GcUsuario con el repositorio GcAdminRepositoryImpl ubicado en el paquete sv.gob.mh.dgii.colas.repositories y el método getAllUser() que obtiene un listado de usuarios con sus tramites . @RequestMapping(value = "/agregar", method = RequestMethod.GET) public String agregar(HttpServletRequest request, @ModelAttribute UsuarioPojo usuariopojo, final ModelMap model); Método de tipo String que se encarga de agregar un nuevo usuario creando una instancia a la clase UsuarioPojo, invocando los métodos preparelist() y getUnidad(); seteando los parametros cUsuario, dUsuario, Escritorio y cargando la url admin/form @RequestMapping(value = "/", method = RequestMethod.POST) public String guardar(HttpServletRequest request, @ModelAttribute UsuarioPojo usuariopojo, final ModelMap model); Método de tipo String que guarda el tramite asignado por el usuario inicializando la clase UsuarioPojo, seteando el parámetro AddTramite, utilizando el método save() del repositorio GcAdminRepositoryImpl ubicado en el paquete sv.gob.mh.dgii.colas.repositories y cargando la url redirect:/admin/	



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA CONFIGURACION DE USUARIOS	
Controlador:	AdminController.java
Descripción de Métodos de Negocio	
@RequestMapping(value = "{user}/editar", method = RequestMethod.GET) public String editar(HttpServletRequest request, @ModelAttribute UsuarioPojo usuariopojo, final ModelMap model, @PathVariable(value = "user") String user); Método de tipo String que llena el pojo a modificar con el usuario, escritorio y tramites y lo muestra en la vista cargando la url admin/form	
@RequestMapping(value = "{user}/show", method = RequestMethod.GET) public String mostrar(HttpServletRequest request, @ModelAttribute UsuarioPojo usuariopojo, final ModelMap model, @PathVariable(value = "user") String user); Método de tipo String que obtiene los tramites por usuario llena el pojo con los parámetros seteados usuario, escritorio y los tramites lo muestra en la vista cargando la url admin/verUsuario.	
@RequestMapping(value = "{user}/delete", method = RequestMethod.POST) public String destroy(HttpServletRequest request, @ModelAttribute UsuarioPojo usuariopojo, final ModelMap model, @PathVariable(value = "user") String user); Método de tipo String que obtiene los tramites por usuario y elimina el usuario cargando la url redirect:/admin/	
@RequestMapping(value = "/getuseridap/{usuarioConsulta}", method = RequestMethod.GET) public @ResponseBody String getContribuyenteNombre(HttpServletRequest request, final ModelMap model, @PathVariable(value = "usuarioConsulta") String usuario); Método de tipo String que obtiene nombre del contribuyente buscándolo por la llave primaria en el método findByPrimaryKey(usuario); por medio de personaLdap obtiene el primer nombre, segundo y apellido, obtiene la unidad receptora valida que si la unidad receptora esta nula obtiene la ubicación física.	
@RequestMapping(value = "/addTramite", method = RequestMethod.GET) public String addTramite(HttpServletRequest request, @ModelAttribute UsuarioPojo usuariopojo, final ModelMap model, @RequestParam(value = "lista", required = false) String[] lista); Método de tipo String que obtiene un listado de tramites invocando el método addList() y la muestra en la vista cargando la url admin/form	



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA CONFIGURACION DE USUARIOS	
Controlador:	AdminController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value = "/delTramite", method = RequestMethod.GET)</p> <p>public String delTramite(HttpServletRequest request, @ModelAttribute UsuarioPojo usuariopojo, final ModelMap model, @RequestParam(value = "lista", required = false) String[] lista);</p> <p>Método de tipo String que elimina los tramites por el usuario invocando el método delList();</p> <p>public void addList(UsuarioPojo user, String[] lista);</p> <p>Método que obtiene un listado de tramites modificados de la entidad GcTramite invoca el iterador next();</p> <p>public void delList(UsuarioPojo user, String[] lista);</p> <p>Método que obtiene un listado de tramites modificados de la entidad GcTramite invoca el iterador next();</p> <p>public void removList(UsuarioPojo user, List<GcTramite> listado);</p> <p>Método que elimina el listado de tramites por el id de tramite invoca el método remove();</p> <p>public void prepareList(UsuarioPojo user, String unidad);</p> <p>Método que setea un listado de tramites y de escritorios de las entidades GcTramite y GcEscritorio</p> <p>@RequestMapping(value = "/getEscritorios", method = RequestMethod.GET)</p> <p>public String getEscritorios(HttpServletRequest request, @ModelAttribute UsuarioPojo usuariopojo, final ModelMap model, @RequestParam(value = "cunidadRecepld", required = false) String unidad) ;</p> <p>Método de tipo String que obtiene un listado de escritorios vacios invocando el método getEscritoriosVacios que se encuentra en el repositorio escritorioRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories cargando la url admin/form</p> <p>public String getUnidad();</p> <p>Método de tipo String que obtiene la unidad receptora y la ubicación física.</p>	

Tabla 3.3 controlador **AdminController**



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE ALERTAS	
Controlador:	AlertaController.java
Descripción de Métodos de Negocio	
<p>public GcAlerta gcAlertaList();</p> <p>Método que crea una nueva instancia de la clase GcAlerta</p> <p>@RequestMapping("")</p> <p>public String iniciar(HttpServletRequest request);</p> <p>Método de tipo String que carga la url redirect:/alerta/</p> <p>@RequestMapping(value = "/", method = RequestMethod.GET)</p> <p>public String index(HttpServletRequest request, @ModelAttribute GcAlerta gcalerta, final ModelMap model);</p> <p>Método de tipo String que obtiene todas las alertas activas y agrega el atributo listadoAlerta carga la url alerta/index</p> <p>@RequestMapping("/newAlerta")</p> <p>public String newAlerta(HttpServletRequest request, @ModelAttribute GcAlerta gcalerta, final ModelMap model);</p> <p>Método String que prepara el formulario para agregar una nueva alerta invoca el método prepareList() ; cargando el formulario alerta/form</p> <p>@RequestMapping(value = "{alerta}/editAlerta", method = RequestMethod.GET)</p> <p>public String editAlerta(HttpServletRequest request, @ModelAttribute GcAlerta gcalerta, final ModelMap model, @PathVariable(value = "alerta") Long alerta);</p> <p>Método de tipo String que prepara el formulario para editar la alerta, obtiene el correo de notificación carga la url alerta/form</p> <p>@RequestMapping(value = "/getzonas", method = RequestMethod.GET)</p> <p>public String getZonasXunidad(HttpServletRequest request, @ModelAttribute GcAlerta gcalerta, final ModelMap model, @RequestParam(value = "cunidadRecepld", required = false) String unidad);</p> <p>Método de tipo String que obtiene las zonas por unidad cargando la url alerta/form</p>	



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE ALERTAS	
Controlador:	AlertaController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value = "{alerta}/delete", method = RequestMethod.POST)</p> <p>public String eliminar(HttpServletRequest request, @ModelAttribute GcAlerta gcalerta, final ModelMap model, @PathVariable(value = "alerta") Long alerta);</p> <p>Método de tipo String que elimina la alerta y guarda utilizando el método save();del repositorio gcAlertaRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories carga la url alerta/index</p>	
<p>@RequestMapping(value = "{alerta}/verAlerta", method = RequestMethod.GET)</p> <p>public String verAlerta(HttpServletRequest request, @ModelAttribute GcAlerta gcalerta, final ModelMap model, @PathVariable(value = "alerta") Long alerta);</p> <p>Método que muestra las alertas cargando la url alerta/show</p>	
<p>@RequestMapping(value = "/", method = RequestMethod.POST)</p> <p>public String guardar(HttpServletRequest request, @ModelAttribute GcAlerta gcalerta, final ModelMap model);</p> <p>Método de tipo String que setea los parámetros CUsuarioCrea, FModifica, BActiva, CUsuarioModi valida que se envíe la alerta al correo del usuario invoca el método save() con el repositorio gcAlertaRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories carga la url alerta/</p>	
<p>@RequestMapping(value = "/getusuario", method = RequestMethod.GET, headers = "Accept=application/json", produces = "application/json")</p> <p>@ResponseBody</p> <p>public TipoUsuarioPojo getUsuario(HttpServletRequest request);</p> <p>Método que inicializa el pojo TipoUsuarioPojo obtiene en el parámetro enviado el usuario y la unidad, asigna un tipo de técnico y obtiene los roles por el tipo de usuario.</p>	
<p>public void prepareList(AlertaPojo alertapojo);</p> <p>Método que setea un listado de zonas y obtiene todas las zonas activas.</p>	
<p>@InitBinder</p> <p>public void initBinder(WebDataBinder binder);</p> <p>Inicializa la instancia binder para dar formato a los campos de fecha.</p>	



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE ALERTAS	
Controlador:	AlertaController.java
Descripción de Métodos de Negocio	
public void enviarCorreos(GcAlerta alerta); Método tipo string que inserta registro de la plantilla en tb_emial y envía la alerta al correo.	
public List<TbUnidadRecep> getUnidadesList(); Método que devuelve una lista de unidades receptoras	
public String getUnidad(); Método que obtiene la ubicación física.	

Tabla 3.4 controlador AlertaController



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE CONFIGURACIÓN DE LLAMADO	
Controlador:	ConfLlamadoController.java
Descripción de Métodos de Negocio	
public CllamadoPojo inicial(); Método que crea una instancia a la clase CllamadoPojo. @RequestMapping(value="/", method = RequestMethod.GET) public String indexConfLlamado(HttpServletRequest request, @ModelAttribute CllamadoPojo cllamadoPojo, final ModelMap model); Método que valida que el rol asignado sea igual al rol de administración de usuario en sesión también obtiene todas las configuraciones del repositorio gcConfLlamadoRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories muestra un listado de llamados cargando la url cllamado/cllamados @RequestMapping("/newCllamado") public String newCllamado(HttpServletRequest request, @ModelAttribute CllamadoPojo cllamadoPojo, final ModelMap model); Método que crea un nuevo llamado obtiene una lista de unidades carga la url cllamado/newCllamado @RequestMapping(value="{cllamdo}/editCllamado", method = RequestMethod.GET) public String editCllamado(HttpServletRequest request, @ModelAttribute CllamadoPojo cllamadoPojo, final ModelMap model, @PathVariable(value = "cllamdo") Long cllamdold) ; Método que edita el llamado utilizando el repositorio gcConfLlamadoRepository para editarlo por el id del llamado llena el pojo con los parámetros setados obtenidos en el formulario carga la url carga la url cllamado/newCllamado RequestMapping(value="{cllamdo}/verCllamado", method = RequestMethod.GET) public String verCllamado(HttpServletRequest request, @ModelAttribute CllamadoPojo cllamadoPojo, final ModelMap model, @PathVariable(value = "cllamdo") Long cllamdold); Método de tipo String que muestra los parámetros seteados en el pojo cargando la url cllamado/verCllamado	



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE CONFIGURACIÓN DE LLAMADO	
Controlador:	ConfLlamadoController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value = "{cllamaod}/delete", method = RequestMethod.POST)</p> <p>public String eliminar(HttpServletRequest request, @ModelAttribute CllamadoPojo cllamadoPojo, final ModelMap model, @PathVariable(value = "cllamaod") Long cllamadoid);</p> <p>Método de tipo String que elimina por el id del llamado utilizando el repositorio gcConfLlamadoRepository ubicado en el paquete org.springframework.data.repository si se elimina el registro carga el mensaje "SE ELIMINO EL REGISTRO CORRECTAMENTE!" si no "ERROR: "+e.getMessage() carga la url cllamado/cllamados</p> <p>@RequestMapping(value = "/", method = RequestMethod.POST)</p> <p>public String guardar(HttpServletRequest request, @ModelAttribute CllamadoPojo cllamadoPojo, final ModelMap model);</p> <p>Método de tipo String que guarda los parámetros seteados obtenidos del formularios utilizando los métodos findOne() para guardar por medio del ID del llamado y save() para guardar los parámetros enviados a la variable cLlamado estos métodos se encuentran en el repositorio gcConfLlamadoRepository carga la url cllamado/cllamados.</p> <p>public List<TbUnidadRecep> getUnidadesList();</p> <p>Método que obtiene una lista de unidades receptoras en servicios con el repositorio gcUnidadRecepRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories también obtiene un listado de combinaciones de unidades, captura el nombre de las unidades receptoras setea el listado de las unidades obtenidas.</p> <p>public String getUnidad();</p> <p>Método de tipo String que obtiene la unidad receptora y si viene vacio obtiene la ubicación física.</p>	

Tabla 3.5 controlador ConfLlamadoController



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE CONFIGURACIÓN DE TIQUETE	
Controlador:	ConfTiqueteController.java
Descripción de Métodos de Negocio	
public GcConfTiquete gcctramite(); Método que crea una instancia a la clase GcConfTiquete @RequestMapping(value="/", method = RequestMethod.GET) public String indexConfTiquete(HttpServletRequest request, @ModelAttribute GcConfTiquete gcConfTiquete, final ModelMap model); Método de tipo String que carga la url conftiquete/confTIQ @SuppressWarnings("unchecked") @RequestMapping(value="/getTq", method=RequestMethod.POST, headers="Accept=application/json", produces=MediaType.APPLICATION_JSON_VALUE, consumes=MediaType.APPLICATION_JSON_VALUE) public @ResponseBody String getTiquete(@RequestBody Map<String, Integer> map); Método de tipo String que valida si la unidad receptora viene vacia entonces obtiene la ubicación física y captura n_unidad_tlv con el repositorio gcConfTiqueteRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories obtiene el nombre de la unidad receptora, obtiene una lista de configuraciones del tiquete y obtiene el máximo valor de la versión, también recibe una lista de plantillas para la configuración del tiquete. @RequestMapping(value="/saveTq", method=RequestMethod.POST, headers="Accept=application/json", produces=MediaType.APPLICATION_JSON_VALUE, consumes=MediaType.APPLICATION_JSON_VALUE) public @ResponseBody String saveTiquete(@RequestBody Map<String, String> map); Método que realiza la funcion de guardar el tiquete , aplica un formato a la fecha, captura la ultima versión del tieuqt con el repositorio gcConfTiqueteRepository y el método maximaVersion() ubicado en el repositorio sv.gob.mh.dgii.colas.repositories , guarda el estado del riquete si es activo o inactivo @RequestMapping(value="/chgTq", method=RequestMethod.POST, headers="Accept=application/json", produces=MediaType.APPLICATION_JSON_VALUE, consumes=MediaType.APPLICATION_JSON_VALUE) public @ResponseBody String chgTiquete(@RequestBody Map<String, String> map); Método de tipo String que asigna una plantilla por el numero de versión asignado por el usuario	



MANUAL TECNICO
MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE CONFIGURACIÓN DE TIQUETE	
Controlador:	ConfTiqueteController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value="/reaPla", method=RequestMethod.POST, headers="Accept=application/json", produces=MediaType.APPLICATION_JSON_VALUE,consumes=MediaType.APPLICATION_JSON_VALUE)</p> <p>public @ResponseBody String readPlantilla(@RequestBody Map<String, String> map);</p> <p>Método de tipo String que lee la plantilla invocando el método lineas_plantilla() del repositorio gcConfTiqueteRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories</p> <p>@RequestMapping(value="/mirarTq",method=RequestMethod.POST,headers="Accept=application/json", produces=MediaType.APPLICATION_JSON_VALUE,consumes=MediaType.APPLICATION_JSON_VALUE)</p> <p>public @ResponseBody String seeTiquete(@RequestBody Map<String, String> map);</p> <p>Método de tipo String que muestra el tiquete con los métodos visualizar() y getimg() del repositorio gcConfTiqueteRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories</p>	

Tabla 3.6 controlador ConfLlamadoController



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA DE MANTENIMIENTO DE CONFIGURACIÓN DE TRAMITE	
Controlador:	ConfTramiteController.java
Descripción de Métodos de Negocio	
public GcConfTramite gcctramite(); Método que crea una instancia a la clase GcConfTramite @RequestMapping(value="/", method = RequestMethod.GET) public String indexConfTramite(HttpServletRequest request, @ModelAttribute GcConfTramite gcConfTramite, final ModelMap model); Método que valida que el rol asignado sea igual al rol del usuario administrador si es así captura una lista de configuraciones de tramite general si no captura una lista de tramites locales invocando en los tramites locales invoca el método getUnidad() carga la url ctramite/ctramites @RequestMapping("/newCtramite") public String newCtramite(HttpServletRequest request, @ModelAttribute GcConfTramite gcConfTramite2, final ModelMap model); Método de tipo String que prepara el formulario para agrega uun nuevo tramite carga la url ctramite/newCtramite RequestMapping(value="{ctramite}/editCtramite", method = RequestMethod.GET) public String editEquipo(HttpServletRequest request, @ModelAttribute GcConfTramite gcConfTramite, final ModelMap model, @PathVariable(value = "ctramite") Long ctramiteld); Método de tipo String que realiza la función de editar los tramites por medio de su ID , utiliza los métodos findOne() y getAllTramites() del repositorio gcTramiteRepository carga la url ctramite/newCtramite @RequestMapping(value = "/guardar", method = RequestMethod.POST) public String guardarConfPOST(HttpServletRequest request, @ModelAttribute GcConfTramite gcConfTramite, final ModelMap model); Método de tipo String que guarda por el ID del tramite los parámetros seteados obtenidos en el formulario utiliza el método save() con el repositorio gcConfTramiteRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories y valida que los datos guardados no estén duplicados carga la url ctramite/ctramites	



MANUAL TECNICO
MODULO DE ADMINISTRACION

PANTALLA DE MANTENIMIENTO DE CONFIGURACIÓN DE TRAMITE	
Controlador:	ConfTramiteController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value = "{ctramite}/delete", method = RequestMethod.POST)</p> <p>public String eliminar(HttpServletRequest request, @ModelAttribute GcConfTramite gcConfTramite, final ModelMap model, @PathVariable(value = "ctramite") Long ctramiteld);</p> <p>Método de tipo String que elimina los tramites segun el ID utiliza el Método delete() por medio del repositorio gcConfTramiteRepository</p> <p>@RequestMapping(value="{ctramite}/verCtramite", method = RequestMethod.GET)</p> <p>public String verEquipo(HttpServletRequest request, @ModelAttribute GcConfTramite gcConfTramite, final ModelMap model, @PathVariable(value = "ctramite") Long ctramiteld);</p> <p>Método de tipo String que muestra los tramites por su id carga la url ctramite/verCtramite</p> <p>public List<TbUnidadRecep> getUnidadesList();</p> <p>Método de tipo String que obtiene una lista de unidades receptoras.</p> <p>public String getUnidad();</p> <p>Método de tipo String que obtiene la unidad receptora y si viene vacio obtiene la ubicación física.</p>	

Tabla 3.7 controlador ConfTramiteController



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA DE MANTENIMIENTO DE EQUIPOS	
Controlador:	EquipoController.java
Descripción de Métodos de Negocio	
<p>public GcEquipos gcEquiposList();</p> <p>Método que crea una instancia a la clase GcEquipos</p> <p>@RequestMapping(value = "/", method = RequestMethod.GET)</p> <p>public String indexEquipo(HttpServletRequest request, @ModelAttribute GcEquipos gcEquipos, final ModelMap model);</p> <p>Método de tipo String que obtiene un listado de equipo por unidad y activos con los métodos getEquiposActivos() y getEquiposByCS() del repositorio gcEquipoRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories carga la url equipo/equipos</p> <p>@RequestMapping("/newEquipo")</p> <p>public String newEquipo(HttpServletRequest request, @ModelAttribute GcEquipos gcEquipos, final ModelMap model);</p> <p>Método de tipo String que obtiene un listado de todos los equipos con el método findAll() y el repositorio gcTipoEquipoRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories obtiene getUnidadesList() carga la url equipo/newEquipo</p> <p>@RequestMapping(value = "{equipo}/editEquipo", method = RequestMethod.GET)</p> <p>public String editEquipo(HttpServletRequest request,@ModelAttribute GcEquipos gcEquipos, final ModelMap model,@PathVariable(value = "equipo")</p> <p>Método de tipo String que edita los equipos por medio del ID carga la url equipo/newEquipo</p> <p>@RequestMapping(value = "/guardar", method = RequestMethod.POST)</p> <p>public String guardarEquipo(HttpServletRequest request, @ModelAttribute GcEquipos gcEquipos, final ModelMap model);</p> <p>Método de tipo String que realiza la función de preparar el formulario con los parámetros seteados obtenidos en el formulario que se van a guardar por medio de su ID carga la url equipo/equipos</p>	



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA DE MANTENIMIENTO DE EQUIPOS	
Controlador:	EquipoController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value = "/guardar", method = RequestMethod.GET)</p> <p>public String guardarEquipoGET(HttpServletRequest request, @ModelAttribute GcEquipos gcEquipos, final ModelMap model);</p> <p>Método de tipo String que guarda por el id de equipo los parámetros seteados obtenidos en el formulario utilizando el método save(); con el repositorio gcEquipoRepository carga la url equipo/equipos</p> <p>@RequestMapping(value = "{equipo}/delete", method = RequestMethod.POST)</p> <p>public String eliminar(HttpServletRequest request, @ModelAttribute GcEquipos gcEquipos, final ModelMap model, @PathVariable(value = "equipo") Long equipold) ;</p> <p>Método de tipo String que elimina el registro por el id del equipo utiliza el método delete(); con el repositorio gcEquipoRepository carga la url equipo/equipos</p> <p>@RequestMapping(value = "{equipo}/verEquipo", method = RequestMethod.GET)</p> <p>public String verEquipo(HttpServletRequest request, @ModelAttribute GcEquipos gcEquipos, final ModelMap model, @PathVariable(value = "equipo") Long equipold);</p> <p>Método de tipo String que muestra el equipo por su id carga la url equipo/verEquipo</p> <p>public List<TbUnidadRecep> getUnidadesList();</p> <p>Método de tipo String que obtiene una lista de unidades</p> <p>public String getUnidad();</p> <p>Método de tipo String que obtiene la unidad receptora y si viene vacío obtiene la ubicación física.</p>	

Tabla 3.8 controlador EquipoController



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA DE MANTENIMIENTO DE ESCRITORIOS	
Controlador:	EscritorioController.java
Descripción de Métodos de Negocio	
<p>public EscritorioPojo escritotioPojo();</p> <p>Método que obtiene un listado de zonas si el rol asignado ROLE_GC_R1es igual al rol que tiene el usuario administrador si no obtiene las unidades retorna una lista de unidades y de zonas.</p> <p>@RequestMapping(value="/", method = RequestMethod.GET)</p> <p>public String indexEscritorio(HttpServletRequest request, @ModelAttribute EscritorioPojo escritorioP, final ModelMap model);</p> <p>Método de tipo String que obtiene todos los escritorios si el rol asignado ROLE_GC_R1es igual al rol que tiene el usuario administrador con método allEscritorios()si no invoca el método getEscritoriosByUnidad () del repositorio gcEscritorioRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories; carga la url escritorio/escritorios</p> <p>@RequestMapping("/newEscritorio")</p> <p>public String irAdministracionNewZona(HttpServletRequest request, @ModelAttribute EscritorioPojo escritorioP, final ModelMap model);</p> <p>Método de tipo String que setea un listado de zonas con el repositorio zonaRepository y el método findAll() también seta una lista de unidades invocando el método getUnidadesList() carga la url escritorio/newEscritorio</p> <p>@RequestMapping(value = "/", method = RequestMethod.POST)</p> <p>public String index(HttpServletRequest request, @ModelAttribute EscritorioPojo escritorioP, final ModelMap model);</p> <p>Método de tipo String que obtiene la zona por ID y las unidades receptoras por ID,obtiene la información de los escritorios por el nombre inicializa la clase GcEscritorio y setea los parametros a obtener en el formulario luego utiliza el método save() con el repositorio gcEscritorioRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories para guardar los parametros seteados carga la url escritorio/escritorios.</p>	



MANUAL TECNICO
MODULO DE ADMINISTRACION

PANTALLA DE MANTENIMIENTO DE ESCRITORIOS	
Controlador:	EscritorioController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value = "/getzonas", method = RequestMethod.GET)</p> <p>public String getZonasXunidad(HttpServletRequest request, @ModelAttribute EscritorioPojo escritorioPojo, final ModelMap model, @RequestParam(value = "cunidadRecepld", required = false) String unidad);</p> <p>Método de tipo String que obtiene un listado de zonas por unidad y setea el listado de zonas cargando la url escritorio/newEscritorio</p> <p>@RequestMapping(value="{escritorio}/editEscritorio", method = RequestMethod.GET)</p> <p>public String editEscritorio(HttpServletRequest request, @ModelAttribute EscritorioPojo escritorioP, final ModelMap model, @PathVariable(value = "escritorio") Long escritorioId);</p> <p>Método de tipo String edita la información del escritorio de los parametros seteados carga la url escritorio/editEscritorio</p> <p>@RequestMapping(value = "{nEscritorioId}/delete", method = RequestMethod.POST)</p> <p>public String eliminar(HttpServletRequest request, @ModelAttribute EscritorioPojo escritorioPojo, final ModelMap model, @PathVariable(value = "nEscritorioId") Long nEscritorioId);</p> <p>Método de tipo String que elimina la información del escritorio por el id si no es asignado por un usuario carga la url escritorio/escritorios</p> <p>@RequestMapping(value="{escritorio}/verEscritorio", method = RequestMethod.GET)</p> <p>public String verEscritorio(HttpServletRequest request, @ModelAttribute EscritorioPojo escritorioP, final ModelMap model, @PathVariable(value = "escritorio") Long escritorioId);</p> <p>Método de tipo String que muestra un listado de escritorios con la información de los parametros seteados carga la url escritorio/verEscritorio</p> <p>public List< TbUnidadRecep > getUnidadesList();</p> <p>Método de tipo String retorna una lista de unidades receptoras</p> <p>public String getUnidad();</p> <p>Método de tipo String que obtiene la unidad receptora y si viene vacio obtiene la ubicación física.</p>	

Tabla 3.9 controlador **EscritorioController**



MANUAL TECNICO
MODULO DE ADMINISTRACION

PANTALLA DE MANTENIMIENTO DE LISTA DE REPRODUCCIÓN	
Controlador:	MultimedServicioController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value="/", method = RequestMethod.GET)</p> <p>public String indexConfTiquete(HttpServletRequest request, @ModelAttribute GcConfTiquete gcConfTiquete, final ModelMap model);</p> <p>Método de tipo String que carga la url confrepro/confrepro</p> <p>@RequestMapping(value="/cargar", method = RequestMethod.POST)</p> <p>public @ResponseBody String cargar(@RequestBody Map<String, Integer> map);</p> <p>Método de tipo String que obtiene las unidades en servicio con el Método getUnidadesEnServicio() del repositorio unidadRepository</p> <p>@RequestMapping(value="/leerLista", method = RequestMethod.POST)</p> <p>public @ResponseBody String LeerLista(@RequestBody Map<String, String> map);</p> <p>Método de tipo String que obtiene una lista de unidades obtenida con el método getList() del repositorio gcConfreproRepository del paquete sv.gob.mh.dgii.colas.repositories verifica que si la unidad obtenida esta vacia carga el método getListVacia() del repositorio gcConfreproRepository</p> <p>public List< TbUnidadRecep> getUnidadesList();</p> <p>Método de tipo String retorna una lista de unidades receptoras</p> <p>public String getUnidad();</p> <p>Método de tipo String que obtiene la unidad receptora y si viene vacio obtiene la ubicación física.</p>	

Tabla 3.10 controlador MultimedServicioController



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE MULTIMEDIA	
Controlador:	MultimediaController.java
Descripción de Métodos de Negocio	
public MultimediaPojo serviciomulti(); Método que invoca el método findA() del repositorio multimediaRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories; setea los parametros de la clase GcMultimedia y retorna un listado de servicios multimedia. @RequestMapping(value="/", method = RequestMethod.GET) public String indexServicio(HttpServletRequest request, @ModelAttribute MultimediaPojo servicioP, final ModelMap model) Método de tipo String que invoca el Método findA() del repositorio multimediaRepository setea los parametros obtenidos de GcMultimedia obtiene un listado de los parametros setados y carga la url multimedia/multimedia @RequestMapping(value = "/", method = RequestMethod.POST) public String guardar(HttpServletRequest request, @ModelAttribute GcMultimedia gcMultimedia, final ModelMap model, @RequestParam("contenido") MultipartFile file) Método de tipo String que guarda los parametros seteados obtenidos en el formulario con el Método save() con el repositorio multimediaRepository cargando la url multimedia/multimedia @RequestMapping("/newMultimedia") public String newServicio(HttpServletRequest request, @ModelAttribute GcMultimedia gcMultimedia, final ModelMap model); Método de tipo String que inicializa la clase GcMultimedia carga la url multimedia/newMultimedia @RequestMapping(value="{servicio}/editMultimedia", method = RequestMethod.GET) public String editMultimedia(HttpServletRequest request, @ModelAttribute GcMultimedia gcServicios_2, final ModelMap model, @PathVariable(value = "servicio") Long servicioid); Método de tipo String que edita por medio del ID los parametros seteados del pojo MultimediaPojo obtenidos del formulario y carga la url multimedia/editMultimedia	

PANTALLA MANTENIMIENTO DE MULTIMEDIA	
Controlador:	MultimediaController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value = "/saveMulti", method = RequestMethod.POST)</p> <pre>public @ResponseBody String guardar_js(HttpServletRequest request,@RequestBody Map<String, String> map,final ModelMap model);</pre> <p>Método de tipo String que le aplica un formato a la fecha y guarda los parametros obtenidos de la clase gcMultimedia</p>	
<p>@RequestMapping(value="{multimedia}/verMultimedia", method = RequestMethod.GET)</p> <pre>public String verMultimedia(HttpServletRequest request, @ModelAttribute GcMultimedia gcMultimedia, final ModelMap model, @PathVariable(value = "multimedia") Long multimediaId);</pre> <p>Método de tipo String que busca por medio del id multimedia carga la url multimedia/verMultimedia</p>	
<p>@RequestMapping(value="{multimedia}/eliminar", method = RequestMethod.POST)</p> <pre>public String eliminar(HttpServletRequest request, ModelAttribute GcMultimedia gcMultimedia, final ModelMap model, @PathVariable(value = "multimedia") Long multimediaId);</pre> <p>Método de tipo String que invoca el método borrarC() del repositorio multimediaRepository para eliminar el registro por el ID multimedia carga la url multimedia/multimedia</p>	
<p>@RequestMapping(value="/getBlob", method = RequestMethod.POST)</p> <pre>public @ResponseBody String getBlobs (HttpServletRequest request, @RequestBody Map<String, String> map);</pre> <p>Método que captura el vid y lo almacena al disco duro para que se pueda visualizar</p>	

Tabla 3.11 controlador MultimediaController



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE CONFIGURACIÓN - EVALUACIÓN	
Controlador:	PreguntaEvaluacionController.java
Descripción de Métodos de Negocio	
<p>@ModelAttribute("preguntapojo")</p> <p>public PreguntaEvaluacionPojo preguntapojo();</p> <p>Método de tipo String que crea una nueva instancia de la clase pojo PreguntaEvaluacionPojo</p> <p>@RequestMapping(value = "/", method = RequestMethod.GET)</p> <p>public String index(HttpServletRequest request, @ModelAttribute PreguntaEvaluacionPojo preguntaEvaluacionPojo, final ModelMap model);</p> <p>Método de tipo String que obtiene un listado de preguntas y ordena las respuestas por el id de pregunta.</p> <p>Carga la url adminpreguntas/preguntas</p> <p>@RequestMapping("/newPregunta")</p> <p>public String newPregunta(HttpServletRequest request, @ModelAttribute PreguntaEvaluacionPojo preguntaEvaluacionPojo, final ModelMap model);</p> <p>Método de tipo String que crea una instancia a la clase pojo preguntaEvaluacionPojo, invoca al método getUnidadesList() y carga la url adminpreguntas/newPregunta</p> <p>@RequestMapping(value="{pregunta}/editPregunta", method = RequestMethod.GET)</p> <p>public String editPregunta(HttpServletRequest request, @ModelAttribute PreguntaEvaluacionPojo preguntaEvaluacionPojo, final ModelMap model, PathVariable(value = "pregunta") Long preguntald);</p> <p>Método de tipo String que edita la pregunta por su ID enviándole los parametros seteados obtenidos en el formulario cargando la url adminpreguntas/newPregunta</p> <p>@RequestMapping(value="{pregunta}/verPregunta", method = RequestMethod.GET)</p> <p>public String verPregunta(HttpServletRequest request, @ModelAttribute PreguntaEvaluacionPojo preguntaEvaluacionPojo, final ModelMap model, @PathVariable(value = "pregunta") Long preguntald);</p> <p>Método de tipo String que muestra por el id de pregunta los parametros seteados cargando la url adminpreguntas/verPregunta</p>	



MANUAL TECNICO MODULO DE ADMINISTRACION

PANTALLA MANTENIMIENTO DE CONFIGURACIÓN - EVALUACIÓN	
Controlador:	PreguntaEvaluacionController.java
Descripción de Métodos de Negocio	
@RequestMapping(value="{pregunta}/delete", method = RequestMethod.POST) public String delete(HttpServletRequest request, @ModelAttribute PreguntaEvaluacionPojo preguntaEvaluacionPojo, final ModelMap model,@PathVariable(value = "pregunta") Long preguntald) Método de tipo String que elimina el registro seleccionado con el método delete() del repositorio gcRespuestasRepository ubicado en repositorio sv.gob.mh.dgii.colas.repositories carga la url adminpreguntas/preguntas	
@RequestMapping(value = "/", method = RequestMethod.POST) public String guardar(HttpServletRequest request, @ModelAttribute PreguntaEvaluacionPojo preguntaEvaluacionPojo, final ModelMap model); Método de tipo String que guarda los cambios realizados carga la url adminpreguntas/preguntas	
public List< TbUnidadRecep> getUnidadesList(); Método de tipo String retorna una lista de unidades receptoras	
public String getUnidad(); Método de tipo String que obtiene la unidad receptora y si viene vacio obtiene la ubicación física.	

Tabla 3.12 controlador PreguntaEvaluacionController



MANUAL TECNICO
MODULO DE ADMINISTRACION

MANTENIMIENTO DE PRIORIDADES - CATÁLOGO	
Controlador:	PrioridadController.java
Descripción de Métodos de Negocio	
<pre>public GcPrioridad gcPrioridadList();</pre> <p>Método de tipo String que crea una nueva instancia de la clase pojo GcPrioridad</p> <pre>@RequestMapping(value="/", method = RequestMethod.GET)</pre> <pre>public String indexPrioridad(HttpServletRequest request, @ModelAttribute GcPrioridad gcPrioridad, final ModelMap model);</pre> <p>Método de tipo String que carga un listado de prioridades cargando la url prioridad/prioridades</p> <pre>@RequestMapping(value="{prioridad}/editPrioridad", method = RequestMethod.GET)</pre> <pre>public String editPrioridad(HttpServletRequest request,@ModelAttribute GcPrioridad gcPrioridad, final ModelMap model, @PathVariable(value = "prioridad") Long prioridadId) ;</pre> <p>Método de tipo String que prepara el formulario para editarlo por medio del ID cargando la url prioridad/newPrioridad</p> <pre>@RequestMapping(value = "/", method = RequestMethod.POST)</pre> <pre>public String guardarPrioridad(HttpServletRequest request, @ModelAttribute GcPrioridad gcPrioridad, final ModelMap model);</pre> <p>Método de tipo String que guarda los parámetros seteados obtenidos en el formulario con el método save() con el repositorio gcPrioridadRepository cargando la url prioridad/prioridades</p> <pre>@RequestMapping(value = "{prioridad}/delete", method = RequestMethod.POST)</pre> <pre>public String eliminar(HttpServletRequest request, @ModelAttribute GcPrioridad gcPrioridad, final ModelMap model, @PathVariable(value = "prioridad") Long prioridadId);</pre> <p>Método de tipo String que elimina por el id de la prioridad seleccionada cargando la url prioridad/prioridades</p> <pre>@RequestMapping(value="{prioridad}/verPrioridad", method = RequestMethod.GET)</pre> <pre>public String verPrioridad(HttpServletRequest request, ModelAttribute GcPrioridad gcPrioridad, final ModelMap model, @PathVariable(value = "prioridad") Long prioridadId);</pre> <p>Método de tipo String que muestra la prioridad seleccionada cargando la url prioridad/verPrioridad</p>	

Tabla 3.13 controlador **PrioridadController**



MANUAL TECNICO MODULO DE ADMINISTRACION

MANTENIMIENTO DE PRIORIDADES - CATÁLOGO	
Controlador:	ServicioController.java
Descripción de Métodos de Negocio	
<p>public ServicioPojo serviciopojo();</p> <p>Método de tipo String que inicializa la clase pojo ServicioPojo y retorna una lista de servicios.</p> <p>@RequestMapping(value="/", method = RequestMethod.GET)</p> <p>public String indexServicio(HttpServletRequest request, @ModelAttribute ServicioPojo servicioP, final ModelMap model);</p> <p>Método de tipo String que setea una lista de servicios y carga la url servicio/servicios</p> <p>RequestMapping("/newServicio")</p> <p>public String newServicio(HttpServletRequest request, @ModelAttribute GcServicios gcServicios, final ModelMap model);</p> <p>Método de tipo String que prepara el formulario para agregar un nuevo servicio cargando la url servicio/newServicio</p> <p>@RequestMapping(value="{servicio}/editServicio", method = RequestMethod.GET)</p> <p>public String editZona(HttpServletRequest request, @ModelAttribute GcServicios gcServicios, final ModelMap model, @PathVariable(value = "servicio") Long servicioid);</p> <p>Método de tipo String que prepara el formulario para editar el servicio por su id cargando la url servicio/newServicio</p> <p>@RequestMapping(value = "/", method = RequestMethod.POST)</p> <p>public String guardar(HttpServletRequest request, @ModelAttribute GcServicios gcServicios, final ModelMap model);</p> <p>Método de tipo String que se encarga de guardar los parametros seteados obtenidos del formulario con el método save() y el repositorio servicioRepository ubicado en el paquete GcServicioRepository cargando la url servicio/servicios</p> <p>@RequestMapping(value="{servicio}/verServicio", method = RequestMethod.GET)</p> <p>public String verZona(HttpServletRequest request, @ModelAttribute GcServicios gcServicios, final ModelMap model, @PathVariable(value = "servicio") Long servicioid);</p> <p>Método de tipo String que muestra el servicio seleccionado cargando la url servicio/verServicio</p> <p>@RequestMapping(value = "{servicio}/delete", method = RequestMethod.POST)</p>	



MANUAL TECNICO MODULO DE ADMINISTRACION

MANTENIMIENTO DE PRIORIDADES - CATÁLOGO	
Controlador:	ServicioController.java
Descripción de Métodos de Negocio	
<pre>public String eliminar(HttpServletRequest request, @ModelAttribute ZonaPojo zonap, final ModelMap model, @PathVariable(value = "servicio") Long servicioid);</pre> <p>Método de tipo String que se encarga de eliminar el servicio por medio del id carga la url servicio/servicios</p>	

Tabla 3.14 controlador **ServicioController**



MANUAL TECNICO
MODULO DE ADMINISTRACION

MANTENIMIENTO DE TIPOS DE EQUIPOS - CATÁLOGO	
Controlador:	TipoEqueipoController.java
Descripción de Métodos de Negocio	
<pre>public GcPrioridad gcPrioridadList();</pre> <p>Método que crea una nueva instancia de la clase GcPrioridad</p> <pre>@RequestMapping(value="/", method = RequestMethod.GET)</pre> <pre>public String indexPrioridad(HttpServletRequest request, @ModelAttribute GcTipoEquipo gcTipoEquipo, final ModelMap model)</pre> <p>Método de tipo String que muestra un listado de equipo cargando la url tequipo/tequipos</p> <pre>@RequestMapping("/newTequipo")</pre> <pre>public String newPrioridad(HttpServletRequest request, @ModelAttribute GcTipoEquipo gcTipoEquipo, final ModelMap model);</pre> <p>Método de tipo String prepara el nuevo formulario para agregar nuevo equipo cargando la url tequipo/newTequipo</p> <pre>@RequestMapping(value="{tequipo}/editTequipo", method = RequestMethod.GET)</pre> <pre>public String editPrioridad(HttpServletRequest request, @ModelAttribute GcTipoEquipo gcTipoEquipo, final ModelMap model, @PathVariable(value = "tequipo") Long tequipold);</pre> <p>Método de tipo String que prepara el formulario para editarlo cargando la url tequipo/newTequipo</p> <pre>@RequestMapping(value = "/", method = RequestMethod.POST)</pre> <pre>public String guardarPrioridad(HttpServletRequest request, @ModelAttribute GcTipoEquipo gcTipoEquipo, final ModelMap model);</pre> <p>Método de tipo String que guarda el equipo verificando que no exista carga la url tequipo/tequipos</p> <pre>@RequestMapping(value="{tequipo}/verTequipo", method = RequestMethod.GET)</pre> <pre>public String verPrioridad(HttpServletRequest request, @ModelAttribute GcTipoEquipo gcTipoEquipo, final ModelMap model, @PathVariable(value = "tequipo") Long tequipold) ;</pre> <p>Método de tipo String que muestra el equipo por su id cargando la url tequipo/verTequipo</p>	

Tabla 3.15 controlador TipoEqueipoController



MANUAL TECNICO MODULO DE ADMINISTRACION

MANTENIMIENTO DE TIPOS DE TRAMITES- CATÁLOGO	
Controlador:	TramiteController.java
Descripción de Métodos de Negocio	
<pre>public GcTramite gcTramiteList();</pre> <p>Método que crea una nueva instancia de la clase GcTramite</p> <pre>@RequestMapping(value="/", method = RequestMethod.GET)</pre> <pre>public String indexTramite(HttpServletRequest request, @ModelAttribute GcTramite gcTramite, final ModelMap model);</pre> <p>Método de tipo String que carga un listado de tramites cargando la url tramite/tramites</p> <pre>@RequestMapping("/newTramite")</pre> <pre>public String newTramite(HttpServletRequest request, @ModelAttribute GcTramite gcTramite, final ModelMap model);</pre> <p>Método de tipo String que carga una lista de servicios y tramites cargando la url tramite/newTramite</p> <pre>@RequestMapping(value="{tramite}/editTramite", method = RequestMethod.GET)</pre> <pre>public String editTramite(HttpServletRequest request, @ModelAttribute GcTramite gcTramite, final ModelMap model, @PathVariable(value = "tramite") Long tramiteId);</pre> <p>Método de tipo String que edita los parametros seteados del tramite por su id cargando la url tramite/newTramite</p> <pre>@RequestMapping(value = "/", method = RequestMethod.POST)</pre> <pre>public String guardar(HttpServletRequest request, @ModelAttribute GcTramite gcTramite, final ModelMap model);</pre> <p>Método de tipo String que guarda los parametros seteado obtenidos en el formulario cargando la url tramite/tramites</p> <pre>@RequestMapping(value = "{tramite}/delete", method = RequestMethod.POST)</pre> <pre>public String eliminar(HttpServletRequest request, @ModelAttribute GcTramite gcTramite, final ModelMap model, @PathVariable(value = "tramite") Long tramiteId);</pre> <p>Método de tipo String que obtiene los tramites relacionados por medio de su id carga la url tramite/tramites</p> <pre>@RequestMapping(value="{tramite}/verTramite", method = RequestMethod.GET)</pre>	



MANUAL TECNICO MODULO DE ADMINISTRACION

MANTENIMIENTO DE TIPOS DE TRAMITES- CATÁLOGO	
Controlador:	TramiteController.java
Descripción de Métodos de Negocio	
<pre>public String verTramite(HttpServletRequest request, @ModelAttribute GcTramite gcTramite, final ModelMap model,@PathVariable(value = "tramite") Long tramiteId);</pre> <p>Método de tipo String que muestra el tramite por su id cargando la url tramite/verTramite</p>	

Tabla 3.16 controlador **TramiteController**

MANTENIMIENTO DE ZONAS - CATÁLOGO	
Controlador:	ZonaController.java
Descripción de Métodos de Negocio	
<pre>public GcTramite gcTramiteList();</pre> <p>Método que crea una nueva instancia de la clase ZonaPojo seteando una lista de unidades obtenida del método getUnidadesList();</p> <pre>@RequestMapping(value = "/", method = RequestMethod.GET)</pre> <pre>public String indexZona(HttpServletRequest request, @ModelAttribute ZonaPojo zonapojo, final ModelMap model);</pre> <p>Método de tipo String que recorre una lista de la entidad GcZona y obtiene todas las zonas cargando la url zona/zonas</p> <pre>@RequestMapping("/newZona")</pre> <pre>public String irAdministracionNewZona(HttpServletRequest request, @ModelAttribute ZonaPojo zonapojo, final ModelMap model);</pre> <p>Método de tipo String que setea una lista de unidades obtenidas del pojo ZonaPojo y obtiene el nombre de la máxima zona con el método getMaxZona() del repositorio zonaRepository cargando la url zona/newZona</p>	



MANUAL TECNICO MODULO DE ADMINISTRACION

MANTENIMIENTO DE ZONAS - CATÁLOGO	
Controlador:	ZonaController.java
Descripción de Métodos de Negocio	
public String getUnidad(); Método de tipo String que obtiene la unidad receptora y si viene vacío obtiene la ubicación física.	
@RequestMapping(value = "/", method = RequestMethod.POST) public String index(HttpServletRequest request, @ModelAttribute ZonaPojo zonapojo, final ModelMap model); Método de tipo String que setea los parámetros de la entidad GcZona obtenidos en el formulario y los guarda verificando que los datos no se dupliquen cargando la url zona/zonas	
@RequestMapping(value = "{zona}/editZona", method = RequestMethod.GET) public String editZona(HttpServletRequest request, @ModelAttribute ZonaPojo zonapojo, final ModelMap model, @PathVariable(value = "zona") Long zonaId); Método de tipo String que edita la zona por su id cargando la url zona/editZona	
@RequestMapping(value = "/guardar", method = RequestMethod.POST) public String guardar(HttpServletRequest request, @ModelAttribute ZonaPojo zonapojo, final ModelMap model); Método de tipo String que guarda los parámetros seteados obtenidos en el formulario con el método save(); y el repositorio zonaRepository sv.gob.mh.dgii.colas.repositories cargando la url zona/zonas	
@RequestMapping(value = "{zona}/verZona", method = RequestMethod.GET) public String mostrar(HttpServletRequest request, @ModelAttribute ZonaPojo zonaP, final ModelMap model, @PathVariable(value = "zona") Long zonaId); Método de tipo String que muestra los parámetros seteados cargando la url zona/verZona	
@RequestMapping(value = "{zona}/delete", method = RequestMethod.POST)	



MANUAL TECNICO
MODULO DE ADMINISTRACION

MANTENIMIENTO DE ZONAS - CATÁLOGO	
Controlador:	ZonaController.java
Descripción de Métodos de Negocio	
public String eliminar(HttpServletRequest request, ModelAttribute ZonaPojo zonap, final ModelMap model, @PathVariable(value = "zona") Long zonald); Método de tipo String que elimina los parametros seteados por su id cargando la url zona/zonas	
public List<TbUnidadRecep> getUnidadesList(); Método de tipo String que retorna una lista de unidades receptoras.	
public String getUnidad(); Método de tipo String que obtiene la unidad receptora y si viene vacio obtiene la ubicación física.	
public String getNextString(String letra); Método de tipo String que retorna un conjunto de letras de la A-Z con su posicion.	
@RequestMapping(value = "/getzona", method = RequestMethod.GET) public String getZonasXunidad(HttpServletRequest request, @ModelAttribute ZonaPojo zonaPojo, final ModelMap model, @RequestParam(value = "cunidadRecepId", required = false) String unidad) Método de tipo String que obtiene las zonas por unidad cargando la url zona/newZona	

Tabla 3.17 controlador **ZonaController**



MANUAL TECNICO MODULO DE ADMINISTRACION

CONFIGURACIÓN DE REASIGNACIÓN	
Controlador:	ReasignacionController.java
Descripción de Métodos de Negocio	
<p>public ReasignacionPojo gcreasignacion();</p> <p>Método de tipo String que retorna una lista de tramites y de unidades del pojo ReasignacionPojo.</p> <p>@RequestMapping(value = "/", method = RequestMethod.GET)</p> <p>public String indexConfReasignacion(HttpServletRequest request, @ModelAttribute ReasignacionPojo reasignacionPojo, final ModelMap model);</p> <p>Método de tipo String que obtiene una lista de configuraciones generales con el método listaConfiguracionesGeneral() del repositorio GcConfTramiteRepository ubicado en el paquete sv.gob.mh.dgii.colas.repositories</p> <p>@RequestMapping("/newCreasignacion")</p> <p>public String newCreasignacion(HttpServletRequest request, @ModelAttribute ReasignacionPojo reasignacionPojo, final ModelMap model);</p> <p>Método de tipo String que prepara el formulario para agregar una nueva asignacion seteando una lista de tramites y configuraciones cargando la url creasignacion/newCreasignacion</p> <p>@RequestMapping(value = "{creasignacion}/editCreasignacion", method = RequestMethod.GET)</p> <p>public String editreasignacion(HttpServletRequest request, @ModelAttribute ReasignacionPojo reasignacionPojo, final ModelMap model, @PathVariable(value = "creasignacion") Long creasignacionId);</p> <p>Método de tipo String que prepara el formulario con los parametros seteados para editarlo carga el formulario creasignacion/newCreasignacion</p> <p>@RequestMapping(value = "/", method = RequestMethod.POST)</p> <p>public String guardarReasignacion(HttpServletRequest request, @ModelAttribute ReasignacionPojo reasignacionPojo, final ModelMap model);</p> <p>Método de tipo String que prepara el formulario que guarda los parametros seteados obtenidos en el formulario con el repositorio gcConfTramiteRepository y el método save ubicado en el paquete sv.gob.mh.dgii.colas.repositories carga la url creasignacion/creasignaciones</p>	



MANUAL TECNICO MODULO DE ADMINISTRACION

CONFIGURACIÓN DE REASIGNACIÓN	
Controlador:	ReasignacionController.java
Descripción de Métodos de Negocio	
<p>@RequestMapping(value = "{creasignacion}/delete", method = RequestMethod.POST)</p> <p>public String eliminar(HttpServletRequest request, @ModelAttribute ReasignacionPojo reasignacionPojo, final ModelMap model, @PathVariable(value = "creasignacion") Long creasignacionId);</p> <p>Método de tipo String que elimina los parametros seteados de la configuración del tramite con el repositorio gcConfTramiteRepository y el método save() ubicado en el paquete sv.gob.mh.dgii.colas.repositories</p> <p>@RequestMapping(value = "{creasignacion}/verCreasignacion", method = RequestMethod.GET)</p> <p>public String verCreasignacionList(HttpServletRequest request, @ModelAttribute ReasignacionPojo reasignacionPojo, final ModelMap model, @PathVariable(value = "creasignacion") Long creasignacionId);</p> <p>Método de tipo String que retorna la url creasignacion/verCreasignacion</p> <p>public List<TbUnidadRecep> getUnidadesList();</p> <p>Método de tipo String que retorna una lista de unidades receptoras.</p> <p>public String getUnidad();</p> <p>Método de tipo String que obtiene la unidad receptora y si viene vacio obtiene la ubicación física.</p>	

Tabla 3.18 controlador **ReasignacionController**



MANUAL TECNICO MODULO DE ADMINISTRACION

4. Descripción de POJOS de sesión

Clases POJO

En la siguiente tabla se listan las diferentes clases POJO que se usan en el Modulo de Administracion y los diferentes paquetes en donde se pueden encontrar.

Nombre de Opción	Ubicación	Nombre
Configuracion de Usuarios	sv.gob.mh.dgii.colas.pojos.entities	UsuarioPojo.java
	sv.gob.mh.dgii.colas.pojos.security	PersonW.java
Mantenimiento de Alertas	sv.gob.mh.dgii.colas.pojos.entities	AlertaPojo.java
		TipoUsuarioPojo
Mantenimiento de Configuración de Llamado	sv.gob.mh.dgii.colas.pojos.entities	ClamadoPojo.java
Configuracion de Tiquete	--	--
Mantenimiento de Configuración de Tramite	--	--
Mantenimiento de Equipos	--	--
Mantenimiento de Escritorios	sv.gob.mh.dgii.colas.pojos.entities	EscritorioPojo.java
Mantenimiento de Lista de Reproducción	--	--
Mantenimiento de Multimedia	sv.gob.mh.dgii.colas.pojos.entities	MultimediaPojo.java
Mantenimiento de Configuración - Evaluación	sv.gob.mh.dgii.colas.pojos.entities	PreguntaEvaluacionPojo.java
Mantenimiento de Prioridades - Catálogo	--	--
Mantenimiento de Secciones - Catálogo	sv.gob.mh.dgii.colas.pojos.entities	ServicioPojo.java
		ZonaPojo.java
Mantenimiento de Tipos de Equipos - Catálogo	--	--
Mantenimiento de Tramites - Catálogo	--	--



MANUAL TECNICO MODULO DE ADMINISTRACION

Nombre de Opción	Ubicación	Nombre
Mantenimiento de Zonas - Catálogo	sv.gob.mh.dgii.colas.pojos.entities	ZonaPojo.java
Configuración de Reasignación	sv.gob.mh.dgii.colas.pojos.entities	ReasignacionPojo.java

Tabla 4.1 nombre y ubicación de las clases pojo



MANUAL TECNICO MODULO DE ADMINISTRACION

4.1 Descripción de Clases POJO

En la siguiente tabla se describen las diferentes clases POJO que se usan en el Módulo de Administracion.

Pantalla: Configuración de Usuarios	
Pojo	Descripción
UsuarioPojo.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes</p> <pre>private String cUsuario; private String dUsuario; private GcEscritorio escritorio; private List<GcEscritorio> listadoEscritorio = new AutoPopulatingList<GcEscritorio>(GcEscritorio.class); private List<GcTramite> listadoTramite = new AutoPopulatingList<GcTramite>(GcTramite.class); private List<GcTramite> addTramite = new AutoPopulatingList<GcTramite>(GcTramite.class); private List<GcUsuario> usuarioList = new AutoPopulatingList<GcUsuario>(GcUsuario.class);</pre>
PersonW.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes</p> <pre>private static final long serialVersionUID = - 4454216593897L; private Boolean authorities; private String nit; private String displayName; private String cadmTrib; private String rol; //uniqueMember</pre>



MANUAL TECNICO MODULO DE ADMINISTRACION

	<pre>private String ssello; private Long itecnico; private String cargo; private String tipoUsuario; private String ubicacionFisica; private String unidadRecep; private String rawPassword; private String detFtpHost; private String detFtpUsr; private String detFtpPass; private String detFtpInitialDir; private String nomInstitucion; private String nitRepresentante; private String dui; private Boolean ctrlchgpwd; private String email; private String emailPi; private String emailServ; private String banderas; private Object clave; private String givenName; private String middleName; private String sn; private String requerimiento; private String tipoInstitucion; private String preferredlanguage; private String fh_ingreso_serv; private String uidNumber;</pre>
--	---



MANUAL TECNICO MODULO DE ADMINISTRACION

	<pre>private String unidad; private String gidNumber; private String username; private String orclisenabled; private String pasaporteUsr; private String pasaporteRep; private String ipUsuario;</pre>
--	--

Tabla 4.2 nombre y ubicación de las clases pojo

Pantalla: Mantenimiento de Alertas	
Pojo	Descripción
AlertaPojo.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes</p> <pre>private List<TbUnidadRecep> listadoUnidades = new AutoPopulatingList<TbUnidadRecep>(TbUnidadRecep.class); private List<GcZona> listadoZonas = new AutoPopulatingList<GcZona>(GcZona.class);</pre>
TipoUsuarioPojo.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes</p> <pre>private String usuario; private String tipo;</pre>

Tabla 4.3 nombre y ubicación de las clases pojo



MANUAL TECNICO MODULO DE ADMINISTRACION

Pantalla: Mantenimiento de Configuración de Llamado	
Pojo	Descripción
CllamadoPojo.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes</p> <pre>private static final long serialVersionUID = 1L; private TbUnidadRecep cUnidadRecep; private String unidadRecep; private BigInteger nNumLlamadas; private BigInteger nIntervaloLlamada; private short bActiva; private String sMensaje; private String sDescripcion; private Long nConfllamadold;</pre>

Tabla 4.4 nombre y ubicación de las clases pojo



MANUAL TECNICO MODULO DE ADMINISTRACION

Pantalla: Mantenimiento de Escritorios	
Pojo	Descripción
EscritorioPojo.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes:</p> <pre>private Integer cEscritorio; private String dEscritorio private Long nEscritoriold; private String cunidadRecepId; private Long cZonald; private TbUnidadRecep cUnidadRecep; private String dZona; private String cUsuarioCrea; private String cUsuarioModi; private Date fiVigencia; private BigInteger nNumEscritorio; private GcZona nZonald; private String clIdentificador; private short bActiva; private List<TbUnidadRecep> unidadesList = new AutoPopulatingList<TbUnidadRecep>(TbUnidadRecep.class); private List<GcZona> zonaList = new AutoPopulatingList<GcZona>(GcZona.class); private List<GcEscritorio> escritorioList=new AutoPopulatingList<GcEscritorio>(GcEscritorio.class); private String dunidadRecep; private int nNumEscritoriold;</pre>

Tabla 4.5 nombre y ubicación de las clases pojo



MANUAL TECNICO MODULO DE ADMINISTRACION

Pantalla: Mantenimiento de Multimedia	
Pojo	Descripción
MultimediaPojo.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes:</p> <pre>public Long nMultimediaId; public String sTipo; private Serializable xArchivo; public Long nDuracion; public String fiMultimedia; public String ffMultimedia; public String dMultimedia; public String cUsuarioCrea; public String cUsuarioModi; public String fiVigencia; public String ffVigencia; public String fModifica; public short bActiva; public String sRuta; private String sMultimedia;</pre>

Tabla 4.6 nombre y ubicación de las clases pojo



MANUAL TECNICO MODULO DE ADMINISTRACION

Pantalla: Mantenimiento de Configuración - Evaluación	
Pojo	Descripción
PreguntaEvaluacionPojo.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes:</p> <pre>private String pregunta; private String correcto; private String respuestas; private String respuest; private String unidad; private String descripcion; private Long preguntald;</pre>

Tabla 4.7 nombre y ubicación de las clases pojo

Pantalla: Mantenimiento de Configuración - Evaluación	
Pojo	Descripción
PreguntaEvaluacionPojo.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes:</p> <pre>private String pregunta; private String correcto; private String respuestas; private String respuest; private String unidad; private String descripcion; private Long preguntald;</pre>

Tabla 4.8 nombre y ubicación de las clases pojo



MANUAL TECNICO MODULO DE ADMINISTRACION

Mantenimiento de Secciones - Catálogo	
Pojo	Descripción
PreguntaEvaluacionPojo.java	<p>Las variables que se encuentran dentro de esta clase son las siguientes:</p> <pre>private String pregunta; private String correcto; private String respuestas; private String respuest; private String unidad; private String descripcion; private Long preguntald;</pre>

Tabla 4.9 nombre y ubicación de las clases pojo



MANUAL TECNICO MODULO DE ADMINISTRACION

5. Listado de Repositorios.

Nombre de la opcion	Ubicación	Nombre
Configuracion de Usuarios	sv.gob.mh.dgii.colas.repositories	GcAdminRepositoryImpl.java GcEscritorioRepository.java GcTramiteRepository.java GcUsuarioRepository.java PersonWRepository.java
Mantenimiento de Alertas	sv.gob.mh.dgii.colas.repositories	DeNotifEmailRepository.java GcAlertaRepository.java GcUnidadRecepRepository.java GcZonaRepository.java
Mantenimiento de Configuración de Llamado	sv.gob.mh.dgii.colas.repositories	GcConfLlamadoRepository.java GcUnidadRecepRepository.java
Configuracion de Tiquete	sv.gob.mh.dgii.colas.repositories	GcConfTiqueteRepository.java
Mantenimiento de Configuración de Tramite	sv.gob.mh.dgii.colas.repositories	GcTramiteRepository.java GcUnidadRecepRepository.java
Mantenimiento de Equipos	sv.gob.mh.dgii.colas.repositories	GcEquipoRepository.java GcTipoEquipoRepository.java GcUnidadRecepRepository.java
Mantenimiento de Escritorios	sv.gob.mh.dgii.colas.repositories	GcEscritorioRepository.java GcUnidadRecepRepository.java GcUsuarioRepository.java GcZonaRepository.java
Mantenimiento de Lista de Reproducción	sv.gob.mh.dgii.colas.repositories	GcMultiServicioRepository.java GcUnidadRecepRepository.java
Mantenimiento de Multimedia	sv.gob.mh.dgii.colas.repositories	GcMultimediaRepository.java GcMultimediaRepositoryImpl.java



MANUAL TECNICO MODULO DE ADMINISTRACION

Mantenimiento de Configuración - Evaluación	sv.gob.mh.dgii.colas.repositories	GcPreguntasRepository.java GcRespuestasRepository.java GcUnidadRecepRepository.java
Mantenimiento de Prioridades - Catálogo	sv.gob.mh.dgii.colas.repositories	GcPrioridadRepository.java
Mantenimiento de Secciones - Catálogo	sv.gob.mh.dgii.colas.repositories	GcServicioRepository.java
Mantenimiento de Tipos de Equipos - Catálogo	sv.gob.mh.dgii.colas.repositories	GcTipoEquipoRepository.java
Mantenimiento de Tramites - Catálogo	sv.gob.mh.dgii.colas.repositories	GcServicioRepository.java GcTramiteRepository.java
Mantenimiento de Zonas - Catálogo	sv.gob.mh.dgii.colas.repositories	GcEscritorioRepository.java GcUnidadRecepRepository.java GcZonaRepository.java
Configuración de Reasignación	sv.gob.mh.dgii.colas.repositories	GcConfTramiteRepository.java GcTramiteRepository.java GcUnidadRecepRepository.java

Tabla 5.1 Listado de repositorios



5.1 Descripción de Repositorios

En la siguiente tabla se describen cada una de los repositorios usados en el módulo Declaración Sugerida de Renta (Declaración en Línea)

OPCION: Configuracion de Usuarios	
Repositorio	Descripción de métodos
PersonWRepository.java	public PersonW findByPrimaryKey(String name); Método que busca a la persona por su id
GcTramiteRepository.java	public List<GcTramite> getTramitesByUser(String user); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcTramite t INNER JOIN t.gcUsuarioList u WHERE u.cUsuario = ?1 ")
GcAdminRepository.java	public List<GcUsuario> getAllUser(); Método de tipo String que obtiene todos los usuarios de la entidad GcUsuario
GcEscritorioRepository.java	public List<GcEscritorio> getEscritoriosVacios(String unidadRecep); Método que ejecuta la siguiente sentencia: @Query("SELECT e FROM GcEscritorio e where e.nEscritorioId not in (SELECT u.nEscritorioId FROM GcUsuario u) and c_unidad_recep= ?1 and e.ffVigencia is null")

Tabla 5.2 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Alertas	
Repositorio	Descripción de métodos
GcAlertaRepository.java	public Iterable<GcAlerta> findAllActivas(Date fecha); Método que ejecuta la siguiente sentencia: <code>@Query("SELECT alerta FROM GcAlerta alerta WHERE alerta.bActiva = 1 AND ?1 BETWEEN alerta.fiVigencia AND alerta.ffVigencia ORDER BY alerta.nAlertald ASC")</code>
GcUnidadRecepRepository.java	public List<TbUnidadRecep> getUnidadesEnServicio(); Método que ejecuta la siguiente sentencia: <code>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep NOT IN(SELECT d.id.clistaDet FROM TbListasValorDet d where c_modulo='GC')")</code> public String getNombreUnidadCombinacion(String unidadRecep); Método que ejecuta la siguiente sentencia: <code>@Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)</code> public String getNombreUnidadCombinacion(String unidadRecep); Método que ejecuta la siguiente sentencia: <code>@Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)</code>
GcZonaRepository.java	public List<GcZona> getZonasByUnidad(String cunidadRecepId); Método que ejecuta la siguiente sentencia: <code>@Query("SELECT t FROM GcZona t where C_UNIDAD_RECEP = ?1 and t.ffVigencia is null ")</code>



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

OPCION: Mantenimiento de Alertas	
Repositorio	Descripción de métodos
	public List<GcZona> getAllZonasActivas(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcZona t WHERE t.bActiva = 1 and t.ffVigencia is null")
DeNotifEmailRepository.java	public Long getLast(); Método que ejecuta la siguiente sentencia: @Query("Select max(o.iindice) from DeNotifEmail o ")
CrudRepository.java	public <S extends T> S save(S s); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository

Tabla 5.3 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Configuración de Llamado

Repositorio	Descripción de métodos
GcUnidadRecepRepository.java	public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository
GcUnidadRecepRepository.java	public List< TbUnidadRecep> getUnidadesEnServicio(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep NOT IN(SELECT d.id.clistaDet FROM TbListasValorDet d where c_modulo='GC')") public List< TbUnidadRecep> getUnidadesCombinaciones(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep IN(SELECT MIN(d.id.clistaDet) FROM TbListasValorDet d where c_modulo='GC' group by c_lista)") public String getNombreUnidadCombinacion(String unidadRecep); Método que ejecuta la siguiente sentencia: @Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)
GcConfLlamadoRepository.java	public List<GcConfLlamado> getAllConf(); Método que ejecuta la siguiente sentencia: @Query("SELECT g FROM GcConfLlamado g WHERE g.bActiva=1 ") public List<GcConfLlamado> getConfsByCS(String cunidadRecep); Método que ejecuta la siguiente sentencia: @Query("SELECT g FROM GcConfLlamado g WHERE g.bActiva=1 and c_unidad_recep = ?1")



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Configuración de Llamado	
Repositorio	Descripción de métodos
CrudRepository.java	public void delete(T t); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public <S extends T> S save(S s); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository
GcConfLlamadoRepository.java	public int getCounByCS(String cunidadRecep); Método que ejecuta la siguiente sentencia: @Query("SELECT count(g) FROM GcConfLlamado g WHERE g.bActiva=1 and c_unidad_recep = ?1")

Tabla 5.4 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Configuración de Tiquete

Repositorio	Descripción de métodos
GcConfTiqueteRepository.java	<p>public TbListasValor n_unidad_tlv(String cUnidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("Select a from TbListasValor a where a.id.clista=(select b.id.clista from TbListasValorDet b where b.id.clistaDet= ?1 and b.id.cmodulo='GC')")</pre> <p>public TbUnidadRecep n_unidad_tur(String cUnidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("Select c from TbUnidadRecep c where c.unidadRecep=?1")</pre> <p>public List<GcConfTiquete> listaConfiguraciones(String cUnidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT c FROM GcConfTiquete c where C_UNIDAD_RECEP = ?1 order by nVersion desc")</pre> <p>public List<String> lista_plantilla();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="select n_plantilla_tiq ';' s_nombre from gc_plantilla_tiq",nativeQuery = true)</pre> <p>public GcConfTiquete maxVer(String nv,String unidad);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT c FROM GcConfTiquete c where N_VERSION = ?1 and C_UNIDAD_RECEP=?2 order by nVersion desc")</pre> <p>public String lineas_tiquete(String cUnidadRecep,String maxver);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("select trim(replace(replace(regexp_replace(regexp_replace(X_CONTENIDO,</pre>

OPCION: Configuracion de Tiquete

Repositorio	Descripción de métodos
	<p>'</[^>]*>','%%'),'<[^>]*>','&&'),'&& ',''),'%% ',"')) from GcConfTiquete a where C_UNIDAD_RECEP=?1 and N_VERSION=?2")</p> <p>public int maximaVersion(String unidad);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT nvl(max(nVersion),0) FROM GcConfTiquete c where C_UNIDAD_RECEP=?1")</p> <p>public GcConfTiquete currTq(String id);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT c FROM GcConfTiquete c where N_CONF_TIQ_ID = ?1 order by nVersion desc")</p> <p>public void noActivo(String cUnidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("update GcConfTiquete set B_ACTIVA=0 where C_UNIDAD_RECEP=?1")</p> <p>public void setActivo(String cUnidadRecep, String ver);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("update GcConfTiquete set B_ACTIVA=1 where C_UNIDAD_RECEP=?1 and N_VERSION=?2")</p> <p>public List<GcConfTiquete> listaConfiguraciones(String cUnidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT c FROM GcConfTiquete c where C_UNIDAD_RECEP = ?1 order by nVersion desc")</p>



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Configuración de Tiquete

Repositorio	Descripción de métodos
CrudRepository.java	public <S extends T> S save(S s); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository

Tabla 5.5 Nombre y descripción de repositorios

OPCION: Mantenimiento de Configuración de Tramite

Repositorio	Descripción de métodos
GcUnidadRecepRepository.java	public String getNombreUnidadCombinacion(String unidadRecep); Método que ejecuta la siguiente sentencia: @Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true) public List< TbUnidadRecep> getUnidadesCombinaciones(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep IN(SELECT MIN(d.id.clistaDet) FROM TbListasValorDet d where c_modulo='GC' group by c_lista)") public String getNombreUnidadCombinacion(String unidadRecep); Método que ejecuta la siguiente sentencia: @Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)
GcTramiteRepository.java	public List<GcTramite> getAllTramites(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcTramite t where t.bActiva=1")

Tabla 5.6 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Equipos	
Repositorio	Descripción de métodos
GcEquipoRepository.java	public List<GcEquipos> getEquiposActivos(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcEquipos t where t.bActiva = 1 ") public List<GcEquipos> getEquiposByCS(String unidadRecep); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcEquipos t where c_unidad_recep = ?1 and t.bActiva = 1 ")
CrudRepository.java	public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public <S extends T> S save(S s); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public void delete(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public Iterable<T> findAll(); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository
GcUnidadRecepRepository.java	public List< TbUnidadRecep> getUnidadesEnServicio(); Método que ejecuta la siguiente sentencia:



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Equipos	
Repositorio	Descripción de métodos
	<p>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep NOT IN(SELECT d.id.clistaDet FROM TbListasValorDet d where c_modulo='GC')")</p> <p>public List<TbUnidadRecep> getUnidadesCombinaciones();</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep IN(SELECT MIN(d.id.clistaDet) FROM TbListasValorDet d where c_modulo='GC' group by c_lista)")</p> <p>public String getNombreUnidadCombinacion(String unidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)</p> <p>public List<TbUnidadRecep> getUnidadesEnServicio(String cunidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT t FROM TbUnidadRecep t where t.mservicio IN('C','M') and C_UNIDAD_RECEP = SIIT.PKG_COLAS_UTILS.CENTRO_SERVICIO(nvl(?1,C_UNIDAD_RECEP))")</p>

Tabla 5.6 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Escritorios

Repositorio	Descripción de métodos
GcZonaRepository.java	public List<GcZona> getAllZonas(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcZona t WHERE t.ffVigencia is null ") public List<GcZona> getZonasByUnidad(String cunidadRecepId); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcZona t where C_UNIDAD_RECEP = ?1 and t.ffVigencia is null ")
GcUnidadRecepRepository.java	public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public List<TbUnidadRecep> getUnidadesEnServicio(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep NOT IN(SELECT d.id.clistaDet FROM TbListasValorDet d where c_modulo='GC')") public List<TbUnidadRecep> getUnidadesCombinaciones(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep IN(SELECT MIN(d.id.clistaDet) FROM TbListasValorDet d where c_modulo='GC' group by c_lista)") public String getNombreUnidadCombinacion(String unidadRecep); Método que ejecuta la siguiente sentencia: @Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Escritorios

Repositorio	Descripción de métodos
GcEscritorioRepository.java	<p>public List<GcEscritorio> allEscritorios();</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT e FROM GcEscritorio e where e.ffVigencia is null")</p> <p>public List<GcEscritorio> getEscritoriosByUnidad(String Unidad);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT e FROM GcEscritorio e where C_UNIDAD_RECEP = ?1 and e.ffVigencia is null")</p> <p>public GcEscritorio getEscritorioByName(String cIdentificador, Long Escritoriold, String cUnidadRecep, Long nZonald);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT e FROM GcEscritorio e where c_identificador = ?1 and n_num_escritorio = ?2 and c_unidad_recep = ?3 and n_zona_id= ?4 and e.ffVigencia is null ")</p> <p>public GcEscritorio getEscritorioByNameEdit(String cIdentificador, BigInteger nNumEscritorio, TbUnidadRecep cUnidadRecep, GcZona nZonald, Long nEscritoriold);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT e FROM GcEscritorio e where e.cIdentificador = ?1 and e.nNumEscritorio = ?2 and e.cUnidadRecep = ?3 and e.nZonald = ?4 and e.ffVigencia is null and e.nEscritoriold != ?5")</p>
CrudRepository.java	<p>public <S extends T> S save(S s);</p> <p>Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository</p> <p>public T findOne(ID id);</p> <p>Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository</p>



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Escritorios

Repositorio	Descripción de métodos
GcUsuarioRepository.java	public int getUsersEscritorio(Long escritorio); Método que ejecuta la siguiente sentencia: @Query("SELECT count(u) FROM GcUsuario u where n_escritorio_id=?1")

Tabla 5.7 Nombre y descripción de repositorios

OPCION: Mantenimiento de Lista de Reproducción

Repositorio	Descripción de métodos
GcMultiServicioRepository.java	public List<String> getLista(String cunidadRecep); Método que ejecuta la siguiente sentencia: @Query(value="select str from(" + "select n_multimedia_id ';' d_multimedia ';' n_orden str,n_orden orden " + "from gc_multimedia a inner join gc_multimed_servicio b using (n_multimedia_id) " + "where b_activa=1 and " + "sysdate between a.FI_MULTIMEDIA and a.FF_MULTIMEDIA " + "and (c_unidad_recep=?1) " + "union " + "select n_multimedia_id ';' d_multimedia ';' 10000 str,10000 orden " + "from gc_multimedia a " + "where b_activa=1 and " + "sysdate between a.FI_MULTIMEDIA and a.FF_MULTIMEDIA " + "and n_multimedia_id not in (select n_multimedia_id from gc_multimed_servicio where c_unidad_recep=?1) " + ") order by orden",nativeQuery = true)



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Lista de Reproducción

Repositorio	Descripción de métodos
	<p>public List<String> getListaVacía();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="select n_multimedia_id ';' d_multimedia ';' 10000 from gc_multimedia a " + "where b_activa=1 and " + "sysdate between a.FI_MULTIMEDIA and a.FF_MULTIMEDIA " + "order by n_multimedia_id",nativeQuery = true)</pre> <p>public void borrar(String cUnidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="delete from GC_MULTIMED_SERVICIO where C_UNIDAD_RECEP=?1",nativeQuery = true)</pre> <p>public void insertar(String cUnidadRecep,int nmulti,int norden);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="insert into GC_MULTIMED_SERVICIO(C_UNIDAD_RECEP,N_MULTIMEDIA_ID,N_ORDEN) values(?1,?2,?3)",nativeQuery = true)</pre>
GcUnidadRecepRepository.java	<p>public List<TbUnidadRecep> getUnidadesEnServicio();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep NOT IN(SELECT d.id.clistaDet FROM TbListasValorDet d where c_modulo='GC')")</pre> <p>public List<TbUnidadRecep> getUnidadesCombinaciones();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep IN(SELECT MIN(d.id.clistaDet) FROM TbListasValorDet d where c_modulo='GC' group by c_lista)")</pre>



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Lista de Reproducción	
Repositorio	Descripción de métodos
	<p>public String getNombreUnidadCombinacion(String unidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)</p> <p>public List<TbUnidadRecep> getUnidadesEnServicio(String cunidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT t FROM TbUnidadRecep t where t.mservicio IN('C','M') and C_UNIDAD_RECEP = SIIT.PKG_COLAS_UTILS.CENTRO_SERVICIO(nvl(?1,C_UNIDAD_RECEP))")</p> <p>public String getNombreUnidadCombinacion(String unidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)</p>

Tabla 5.8 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Multimedia

Repositorio	Descripción de métodos
GcMultimediaRepository.java	<p>public List<String> findA();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="SELECT n_Multimedia_Id ';' s_Tipo ';' " + " n_Duracion ';' TO_CHAR(fi_Multimedia,'DD/MM/YYYY') ';' TO_CHAR(ff_Multimedia,'DD/MM/YYYY') ';' " + " d_Multimedia ';' c_Usuario_Crea ';' c_Usuario_Modi ';' " + " to_char(fi_Vigencia,'DD/MM/YYYY') ';' to_char(f_Modifica,'DD/MM/YYYY') ';' " + " b_Activa ';' s_Ruta FROM gc_multimedia order by n_Multimedia_Id ", nativeQuery = true)</pre> <p>public void actualizar(long i,long ndur,String cumod,String dmult,Date fmodi,int bacti,String stipo,Date ffmul,Date fimul,String sruta);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="UPDATE GC_MULTIMEDIA " + " SET N_DURACION = ?2 , " + " C_USUARIO_MODI = ?3 , " + " D_MULTIMEDIA = ?4 , " + " F_MODIFICA = ?5 , " + " B_ACTIVA = ?6 , " + " S_TIPO = ?7 , " + " FF_MULTIMEDIA = ?8 , " + " FI_MULTIMEDIA = ?9 , " + " S_RUTA = ?10 " + " WHERE N_MULTIMEDIA_ID = ?1 ",nativeQuery = true)</pre>



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Multimedia	
Repositorio	Descripción de métodos
	<p>public String findO(Long i);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="SELECT n_Multimedia_Id ';' s_Tipo ';' " + " n_Duracion ';' TO_CHAR(fi_Multimedia,'DD/MM/YYYY') ';' TO_CHAR(ff_ Multimedia,'DD/MM/YYYY') ';' " + " d_Multimedia ';' c_Usuario_Crea ';' c_Usuario_Modi ';' " + " to_char(fi_Vigencia,'DD/MM/YYYY') ';' to_char(f_Modifica,'DD/MM/YYYY') ';' " + " b_Activa ';' s_Ruta ';' s_multimedia ';' TO_CHAR(fi_vigencia,'DD/MM/YY YY') FROM gc_multimedia where n_Multimedia_Id=?1 ", nativeQuery = true)</pre> <p>public void borrarC(long i);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="delete from gc_multimedia where n_multimedia_id=?1",nativeQuery = true)</pre>

Tabla 5.9 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Configuración - Evaluación

Repositorio	Descripción de métodos
GcPreguntasRepository.java	public List<GcPreguntas> preguntaYrespuestas(String cUnidadRecep); Método que ejecuta la siguiente sentencia: @Query("SELECT c FROM GcPreguntas c where C_UNIDAD_RECEP = ?1 ")
CrudRepository.java	public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public void delete(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository
GcRespuestasRepository.java	public List<GcRespuestas> respuestasByPregutna(Long nPreguntald); Método que ejecuta la siguiente sentencia: @Query("SELECT c FROM GcRespuestas c where N_PREGUNTA_ID = ?1 ") public List<GcRespuestas> respuestasByPregutna(Long nPreguntald); Método que ejecuta la siguiente sentencia: @Query("SELECT c FROM GcRespuestas c where N_PREGUNTA_ID = ?1 ")
GcUnidadRecepRepository.java	public List<TbUnidadRecep> getUnidadesEnServicio(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep NOT IN(SELECT d.id.clistaDet FROM TbListasValorDet d where c_modulo='GC')")



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Configuración - Evaluación

Repositorio	Descripción de métodos
	<p>public List<TbUnidadRecep> getUnidadesCombinaciones();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep IN(SELECT MIN(d.id.clistaDet) FROM TbListasValorDet d where c_modulo='GC' group by c_lista)")</pre> <p>public String getNombreUnidadCombinacion(String unidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)</pre>

Tabla 5.10 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Prioridades - Catálogo	
Repositorio	Descripción de métodos
GcPrioridadRepository.java	public Long getPrioridadDuplicada(String nombre,Long id); Método que ejecuta la siguiente sentencia: @Query("SELECT count(p) FROM GcPrioridad p WHERE p.bActiva in(1,0) AND REPLACE(upper(p.sNombre),' ','') = REPLACE(upper(?1) , ' ','') and p.nPrioridadId != ?2")
CrudRepository.java	public Iterable<T> findAll(); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public <S extends T> S save(S s); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository

Tabla 5.11 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Secciones - Catálogo	
Repositorio	Descripción de métodos
GcServicioRepository.java	<p>public List<GcServicios> getAllSecciones();</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT s FROM GcServicios s WHERE s.bActiva in(0,1) ")</p> <p>public Integer getServicioDuplicado(String servicio,Long serviciold);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT count(s) FROM GcServicios s where REPLACE(upper(s.sNombre),' ','') = REPLACE(upper(?1) , ' ','') and s.nServiciosId != nvl(?2,0) and s.bActiva in(0,1) ")</p> <p>public BigInteger getServicioMaxOrden();</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query("SELECT max(s.nOrden)+1 FROM GcServicios s WHERE s.bActiva in(0,1) ")</p> <p>public Integer getServicioRelacionado(Long serviciold);</p> <p>Método que ejecuta la siguiente sentencia:</p> <p>@Query(value="SELECT count(*) FROM gc_servicios s where EXISTS (select * from gc_tramite tra where tra.n_servicios_id= s.n_servicios_id) and s.n_servicios_id=?1 s.b_Activa in(0,1)", nativeQuery = true)</p>
CrudRepository.java	<p>public T findOne(ID id);Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository</p> <p>public <S extends T> S save(S s);</p> <p>Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository</p>

Tabla 5.12 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Tipos de Equipos - Catálogo	
Repositorio	Descripción de métodos
GcTipoEquipoRepository.java	public String getTipoEquipoByName(String sNombre); Método que ejecuta la siguiente sentencia: @Query("SELECT t.sNombre FROM GcTipoEquipo t where t.sNombre = ?1 ")
CrudRepository.java	public Iterable<T> findAll(); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public <S extends T> S save(S s); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public void delete(T t); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository

Tabla 5.13 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Tramites - Catálogo	
Repositorio	Descripción de métodos
GcTramiteRepository.java	public int getTramiteDuplicado(String tramite,Long tramiteId); Método que ejecuta la siguiente sentencia: @Query("SELECT count(s) FROM GcTramite s where REPLACE(upper(s.sNombre),' ','') = REPLACE(upper(?1) , ' ','') and s.nTramiteId != nvl(?2,0)") public int getTramiteRelacionados(Long tramiteId); Método que ejecuta la siguiente sentencia: @Query(value="select count(*) from gc_tramite tra where((EXISTS (select * from gc_conf_tramite ct where ct.n_tramite_id=tra.n_tramite_id)) or (exists (select * from gc_usr_tra ut where ut.n_tramite_id=tra.n_tramite_id))) and tra.n_tramite_id=?1 ", nativeQuery = true)
CrudRepository.java	public Iterable<T> findAll(); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public <S extends T> S save(S s); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public void delete(T t); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository

Tabla 5.14 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Mantenimiento de Zonas - Catálogo

Repositorio	Descripción de métodos
CrudRepository.java	public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public <S extends T> S save(S s); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository
GcZonaRepository.java	public List<GcZona> getAllZonas(); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcZona t WHERE t.ffVigencia is null ") public List<GcZona> getZonasByUnidad(String cunidadRecepld); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcZona t where C_UNIDAD_RECEP = ?1 and t.ffVigencia is null ") public String getMaxZona(String unidadRecep); Método que ejecuta la siguiente sentencia: @Query(value = "SELECT max(s_nombre) FROM gc_zona where C_UNIDAD_RECEP = ?1 and ff_Vigencia is null", nativeQuery = true) public String getZonasByUnidadName(String cunidadRecepld,String sNombre); Método que ejecuta la siguiente sentencia: @Query("SELECT t.sNombre FROM GcZona t where C_UNIDAD_RECEP = ?1 and REPLACE(upper(t.sNombre),' ','')=REPLACE(upper(??2),' ','') and t.ffVigencia is null") public GcZona getZonasByUnidadNameId(String cunidadRecepld,String sNombre, Long nZonald); Método que ejecuta la siguiente sentencia: @Query("SELECT t FROM GcZona t where C_UNIDAD_RECEP = ?1 and t.sNombre = ?2 and t.nZonald != ?3 and rownum=1 and t.ffVigencia is null ")



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

OPCION: Mantenimiento de Zonas - Catálogo	
Repositorio	Descripción de métodos
GcUnidadRecepRepository.java	<p>public TbUnidadRecep getUnidadEnServicioE(String cunidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('C','M') and t.cunidadRecep = ?1 ")</pre> <p>public List<TbUnidadRecep> getUnidadesCombinaciones();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep IN(SELECT MIN(d.id.clistaDet) FROM TbListasValorDet d where c_modulo='GC' group by c_lista)")</pre> <p>public String getNombreUnidadCombinacion(String unidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)</pre> <p>public List<TbUnidadRecep> getUnidadesEnServicio(String cunidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT t FROM TbUnidadRecep t where t.mservicio IN('C','M') and C_UNIDAD_RECEP = SIIT.PKG_COLAS_UTILS.CENTRO_SERVICIO(nvl(?1,C_UNIDAD_RECEP))")</pre>
GcEscritorioRepository.java	<p>public int getEscritoriosByZona(Long zona);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT count(e) FROM GcEscritorio e where n_zona_id = ?1 and e.ffVigencia is null")</pre>

Tabla 5.15 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Configuración de Reasignación

Repositorio	Descripción de métodos
GcConfTramiteRepository.java	<p>public List<GcConfTramite> listaConfiguracionesGeneral();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT c FROM GcConfTramite c where c.nComportamiento is not null")</pre> <p>public List<GcConfTramite> listaConfiguracionesLocal(String cUnidadRecep);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT c FROM GcConfTramite c where C_UNIDAD_RECEP = ?1 and c.nComportamiento is not null")</pre> <p>public GcConfTramite existe(String cUnidadRecep, Long tramiteId);</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT c FROM GcConfTramite c where C_UNIDAD_RECEP = ?1 and N_TRAMITE_ID = ?2 ")</pre> <p>public List<TbUnidadRecep> getUnidadesEnServicio();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep NOT IN(SELECT d.id.clistaDet FROM TbListasValorDet d where c_modulo='GC')")</pre> <p>public List<TbUnidadRecep> getUnidadesCombinaciones();</p> <p>Método que ejecuta la siguiente sentencia:</p> <pre>@Query("SELECT t FROM TbUnidadRecep t WHERE t.mservicio IN('M','C') AND c_unidad_Recep IN(SELECT MIN(d.id.clistaDet) FROM TbListasValorDet d where c_modulo='GC' group by c_lista)")</pre>



MANUAL TECNICO MODULO DE ADMINISTRACION

OPCION: Configuración de Reasignación	
Repositorio	Descripción de métodos
	public String getNombreUnidadCombinacion(String unidadRecep); Método que ejecuta la siguiente sentencia: @Query(value="select D_LISTA from TB_LISTAS_VALOR where c_lista =(select c_lista from TB_LISTAS_VALOR_DET where c_modulo='GC' and c_lista_det=?1 and rownum = 1)",nativeQuery=true)
CrudRepository.java	public T findOne(ID id); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public <S extends T> S save(S s); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository public Iterable<T> findAll(); Método generado de la interfaz CrudRepository ubicada en el repositorio org.springframework.data.repository

Tabla 5.16 Nombre y descripción de repositorios



MANUAL TECNICO MODULO DE ADMINISTRACION

6. Listado de Componentes.

Nombre del Componente	Ubicación
AppInfoInterceptor.java	sv.gob.mh.dgii.det.components

Tabla 6.1 Listado y Ubicación de componentes

Descripción de Componentes.

Nombre del Componente	Descripcion
AppInfoInterceptor.java	<p>Clase que extiende de HandlerInterceptorAdapter la cual sobre-escribe los siguientes Métodos:</p> <pre>public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler);</pre> <p>Esta aplicación siempre devuelve cierto.</p> <pre>public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView);</pre> <p>Esta aplicación está vacía.</p>

Tabla 6.2 Nombre y descripción de componentes



7. Configuración de proyectos (@Configuration y @Beans en general, esto es todo el paquete de sv.gob.mh.dgii.det.config)

Nombre del archivo de configuración	Ubicación sv.gob.mh.dgii.colas.config	Descripción
HibernateConfig	<pre>public LocalSessionFactoryBean alertsSessionFactory();</pre> <p>Método que establece el origen de datos para ser utilizados por la SessionFactory, Especifica el paquete "sv.gob.mh.dgii.model" para buscar la autodetección de sus clases de entidad en la ruta de clases, establece las propiedades de hibernate, establecer la ubicación de un único archivo de configuración de Hibernate XML, por ejemplo, como recurso de ruta de clases "ruta de clases: hibernate.cfg.xml".</p> <pre>public HibernateTransactionManager transactionManager();</pre> <p>Método que ajusta la instancia que debe gestionar las transacciones y obtiene el objeto del método alertsSessionFactory().</p> <pre>public HibernateExceptionHandler exceptionTranslation();</pre> <p>Método que crea una nueva instancia de HibernateExceptionHandler.</p> <pre>final Properties hibernateProperties();</pre> <p>Método que establece las propiedades de la clase Properties</p>	



MANUAL TECNICO
MODULO DE ADMINISTRACION

Nombre del archivo de configuración	Ubicación sv.gob.mh.dgii.colas.config	Descripción
RepositoryConfig	<p>@EnableJpaRepositories(basePackages = { "sv.gob.mh.dgii" }, includeFilters = @ComponentScan.Filter(pattern = ".*repositories.*", type = FilterType.REGEX))</p> <p>Esta anotación me dice que paquete inyectara como repositorios.</p> <p>@ComponentScan(basePackages = "sv.gob.mh.dgii", useDefaultFilters = false, includeFilters = @Filter(pattern = ".*components.*", type = FilterType.REGEX));</p> <p>Configura directivas de escaneo de componentes y me dice que paquete inyectara como repositorios.</p>	
SecurityConfig	<p>protected void configure(HttpSecurity http);</p> <p>Método que contiene información sobre cómo autenticar a los usuarios, Asegura que cualquier petición a nuestra aplicación requiere que el usuario sea autenticado,permite que los usuarios se autentican con formulario basado entrada,permite que los usuarios se autentican con autenticación básica HTTP.</p> <p>public DefaultLdapAuthoritiesPopulator ldapAuthoritiesPopulator();</p> <p>Constructor de escenarios de búsqueda de grupo y suministra los contextos utilizados para buscar roles de usuario.</p> <p>public DgiiFilterInvocationSecurityMetadataSource dgiiSecurityMetadataSource();</p> <p>Método que invoca el paquete "sv.gob.mh.dgii.colas.security.PropertyFileSecurityBuilder".</p> <p>public AuthenticationManager authenticationManagerBean();</p>	



MANUAL TECNICO
MODULO DE ADMINISTRACION

Nombre del archivo de configuración	Ubicación sv.gob.mh.dgii.colas.config	Descripción
	<p>Método de anulación authenticationManagerBean en WebSecurityConfigurerAdapter para exponer el AuthenticationManager construido usando configure(AuthenticationManagerBuilder)</p> <p>public RoleVoter roleVoter();</p> <p>Método que especifica un prefijo de rol al usuario.</p> <p>public AuthenticatedVoter authenticatedVoter();</p> <p>Método para la autenticación de usuario dependiendo si es anónima o si desea que se recuerde.</p> <p>public FilterSecurityInterceptor dgiiFilterSecurityInterceptor();</p> <p>Método que realiza el manejo de la seguridad de los recursos HTTP a través de un filtro de aplicación.</p> <p>BaseLdapPathContextSource contextSource();</p> <p>Interfaz para ser implementado por ContextSources que son capaces de proporcionar la ruta LDAP base.</p> <p>public void configureAuthentication(AuthenticationManagerBuilder auth)</p> <p>Método que obtiene la configuración requerirá que cualquier URL que se solicita será necesario un usuario con el rol de "ROLE_USER".</p>	
SecurityWebApplicationInitializer	<p>Clase que extiende de AbstractSecurityWebApplicationInitializer la cual sobre-escribe los siguientes Métodos:</p> <p>public class SecurityWebApplicationInitializer extends AbstractSecurityWebApplicationInitializer.</p>	



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

Nombre del archivo de configuración	Ubicación sv.gob.mh.dgii.colas.config	Descripción
WebConfig	<p>Clase que extiende de WebMvcConfigurerAdapter la cual sobre-escribe los siguientes Métodos:</p> <p>public String appName(); Método que obtiene el nombre completo de la app.</p> <p>public void addResourceHandlers(ResourceHandlerRegistry registry) ; Agregar controladores para servir recursos estáticos como imágenes, js y css, archivos desde ubicaciones específicas bajo raíz de la aplicación web, la ruta de clase, y otros.</p> <p>public LocaleResolver localeResolver(); Interfaz de estrategias de resolución de localización basadas en web que permite tanto la resolución de la configuración regional a través de la solicitud y la modificación de la configuración regional a través de la solicitud y la respuesta.</p> <p>public ViewResolver viewResolver() ; Método para ver si el estado no cambia durante el funcionamiento de la aplicación.</p> <p>public CommonsMultipartResolver multipartResolver(): Método encargado de setear por defecto la configuración utf-8 y de configurar los tamaños máximos de upload y el uso de la memoria máxima.</p>	
WebSocketConfig	<p>public void configureMessageBroker(MessageBrokerRegistry config): Método encargado de configurar enableSimpleBroker y setApplicationDestinationPrefixes.</p> <p>public void registerStompEndpoints(StompEndpointRegistry registry): Método encargado de configurar el sock en addEndpoint withSockJS.</p>	

TABLA 7.1 Configuraciones del Módulo de Colas



8. Documentos Relacionados a la Base de Datos

Tablas

En las siguientes páginas se describen cada uno de los campos de las tablas de la base de datos. Cuando la **descripción no tiene nada**, quiere decir que dicha tabla no **tiene comentarios en la base de datos**.

GC_ESCRITORIO			
Nombre de secuencia: SEQ_GC_ESCRITORIO			
Nombre del atributo	Tipo de dato	Null	Descripción
N_ESCRITORIO_ID	NUMBER	No	Llave primaria de la tabla
N_ZONA_ID	NUMBER	No	Referencia a la zona a que pertenece el escritorio
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	No	Código del centro de servicio
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no
D_ESCRITORIO	VARCHAR2(256 BYTE)	Yes	Descripción del registro
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
C_IDENTIFICADOR	VARCHAR2(1 BYTE)	Yes	Letra que identifica a un escritorio al momento del llamado



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_ESCRITORIO			
Nombre de secuencia: SEQ_GC_ESCRITORIO			
Nombre del atributo	Tipo de dato	Null	Descripción
N_NUM_ESCRITORIO	NUMBER	No	Número con el cual se identifica al escritorio al momento del llamado
B_FILA_ESP	NUMBER(1,0)	No	Si el escritorio será Fila Especial

Tabla 8.1 GC_ESCRITORIO



MANUAL TECNICO
MODULO DE ADMINISTRACION

GC_TRAMITE			
Nombre de secuencia: SEQ_GC_TRAMITE			
Nombre del atributo	Tipo de dato	Null	Descripción
N_TRAMITE_ID	NUMBER	No	Llave primaria de la tabla
N_SERVICIOS_ID	NUMBER	No	Referencia al servicio a que pertenece el trámite
N_PESO	NUMBER	No	Cuantificación de la prioridad que debe tener el trámite luego de aplicarle todos los criterios
B_NIT_REQUERIDO	NUMBER(1,0)	No	Bandera que indica si es obligatorio proporcionar el nit
B_ESCALAMIENTO	NUMBER(1,0)	No	Bandera que indica si el trámite será escalado al supervisor
D_TRAMITE	VARCHAR2(256 BYTE)	Yes	Descripción del registro
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuario que crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuario que modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no
S_NOMBRE	VARCHAR2(256 BYTE)	No	Nombre del trámite
N_ORDEN	NUMBER	Yes	Orden en que se presentaran los tramites en pantalla

Tabla 8.2 GC_TRAMITE



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_USUARIO			
Nombre de secuencia: --			
Nombre del atributo	Tipo de dato	Null	Descripción
N_ESCRITORIO_ID	NUMBER(38,0)	No	Escritorio asociado al usuario
C_USUARIO	VARCHAR2(256 BYTE)	No	Código del usuario

Tabla 8.3 GC_USUARIO



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_ALERTA			
Nombre de secuencia: SEQ_GC_ALERTA			
Nombre del atributo	Tipo de dato	Null	Descripción
N_ALERTA_ID	NUMBER(38,0)	No	Llave primaria de la tabla
N_SERVICIOS_ID	NUMBER	No	Referencia al servicio al que esta asociado la alerta
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	Yes	Código del centro de servicio
S_MENSAJE	VARCHAR2(1024 BYTE)	No	Mensaje que desplegará o enviará la alerta
S_TIPO	VARCHAR2(15 BYTE)	No	Tipo de alerta
FH_NOTIFICACION	DATE	Yes	Fecha y hora en que se ejecutará la alerta programada
B_CORREO	NUMBER(1,0)	No	Bandera que indica si la Alerta enviará correo o no
B_PANTALLA	NUMBER(1,0)	No	Bandera que indica si se mostrará o no en la pantalla
C_USUARIO_NOTIFICA	VARCHAR2(1024 BYTE)	Yes	Listado de Usuarios a notificar
S_CORREO_NOTIFICA	VARCHAR2(1024 BYTE)	Yes	Listado de correos a los que la alerta notificará
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no

Tabla 8.4 GC_ALERTA



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_ZONA			
Nombre de secuencia: SEQ_GC_ZONA			
Nombre del atributo	Tipo de dato	Null	Descripción
N_ZONA_ID	NUMBER	No	Llave primaria de la tabla
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	No	Código del centro de servicio
N_SILLAS_ESPERA	NUMBER	Yes	Número de sillas en la zona de atención
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no
D_ZONA	VARCHAR2(256 BYTE)	Yes	Descripción del registro
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
S_NOMBRE	VARCHAR2(256 BYTE)	Yes	Nombre de la zona

Tabla 8.5 GC_ZONA



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_CONF_LLAMADO			
Nombre de secuencia: SEQ_GC_CONF_LLAMADO			
Nombre del atributo	Tipo de dato	Null	Descripción
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	No	Código del centro de servicio
N_NUM_LLAMADAS	NUMBER	No	Número de llamadas que se harán al convocar a un nuevo contribuyente
N_INTERVALO_LLAMADA	NUMBER	No	Segundos que transcurirán entre una llamada y la siguiente
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no
S_MENSAJE	VARCHAR2(1024 BYTE)	No	Mensaje que desplegará en pantalla al hacer el llamado
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
S_DESCRIPCION	VARCHAR2(256 BYTE)	Yes	Descripcion del registro
N_CONFLAMADO_ID	NUMBER	No	Llave primaria de la tabla

Tabla 8.6 GC_CONF_LLAMADO



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_CONF_TIQUETE			
Nombre de secuencia: SEQ_GC_CONF_TIQUETE			
Nombre del atributo	Tipo de dato	Null	Descripción
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	No	Código del centro de servicio
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no
X_CONTENIDO	CLOB	No	Formato del tickete que se entregaran en los centros de servicio
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
N_CONF_TIQ_ID	NUMBER	No	Llave primaria de la tabla
N_VERSION	NUMBER	Yes	Versión del tickete en relación alcentro de servicio
X_IMAGEN	BLOB	Yes	Imagen que va en el tickete.

Tabla 8.7 GC_CONF_TIQUETE



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_CONF_TRAMITE			
Nombre de secuencia: SEQ_GC_CONF_TRAMITE			
Nombre del atributo	Tipo de dato	Null	Descripción
N_CONF_TRA_ID	NUMBER	No	Llave primaria de la tabla
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	No	Código del centro de servicio
N_TRAMITE_ID	NUMBER	No	Referencia al tramite a que se refiere la configuración
N_ATENCION_PROM	NUMBER	No	Tiempo promedio de atención
N_TIEMPO_ESPERA	NUMBER	Yes	Tiempo, en minutos, de espera máximo que debe tardarse en la cola
N_TIEMPO_HOLGURA	NUMBER	Yes	Tiempo, en minutos, que debe esperarse antes de incluirlo en la cola
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
N_COMPORTAMIENTO	NUMBER(6,0)	Yes	Comportamiento: 0-al final, 1-al inicio, 2-en medio, 3 por peso
N_PESO	NUMBER(6,0)	Yes	peso de la reasignacion
N_PROM_ESPERA	NUMBER	Yes	Promedio real (en segundos) de espera en cola
N_PROM_ATENCION	NUMBER	Yes	Promedio real (en segundos) de atención en escritorio

Tabla 8.8 GC_CONF_TRAMITE



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_CONF_TRAMITE			
Nombre de secuencia: SEQ_GC_EQUIPOS			
Nombre del atributo	Tipo de dato	Null	Descripción
N_CONF_TRA_ID	NUMBER	No	Llave primaria de la tabla
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	No	Código del centro de servicio
N_TRAMITE_ID	NUMBER	No	Referencia al tramite a que se refiere la configuración
N_ATENCION_PROM	NUMBER	No	Tiempo promedio de atención
N_TIEMPO_ESPERA	NUMBER	Yes	Tiempo, en minutos, de espera máximo que debe tardarse en la cola
N_TIEMPO_HOLGURA	NUMBER	Yes	Tiempo, en minutos, que debe esperarse antes de incluirlo en la cola
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
N_COMPORTAMIENTO	NUMBER(6,0)	Yes	Comportamiento: 0-al final, 1-al inicio, 2-en medio, 3 por peso
N_PESO	NUMBER(6,0)	Yes	peso de la reasignacion
N_PROM_ESPERA	NUMBER	Yes	Promedio real (en segundos) de espera en cola
N_PROM_ATENCION	NUMBER	Yes	Promedio real (en segundos) de atención en escritorio

Tabla 8.9 GC_CONF_TRAMITE



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_MULTIMEDIA			
Nombre de secuencia: SEQ_GC_MULTIMEDIA			
Nombre del atributo	Tipo de dato	Null	Descripción
N_MULTIMEDIA_ID	NUMBER(38,0)	No	Llave primaria de la tabla
S_TIPO	VARCHAR2(256 BYTE)	No	Tipo de contenido multimedia
N_DURACION	NUMBER	No	Tiempo en minutos en que el contenido multimedia se reproducirá
FI_MULTIMEDIA	DATE	No	Fecha de inicio de emisión del contenido
FF_MULTIMEDIA	DATE	No	Fecha en que finaliza la emisión del contenido
D_MULTIMEDIA	VARCHAR2(256 BYTE)	Yes	Descripción del registro
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuario que crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuario que modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no
S_RUTA	VARCHAR2(1024 BYTE)	Yes	Directorio donde se encuentra el contenido multimedia a mostrar
X_ARCHIVO	BLOB	Yes	Archivo con información multimedia.
S_MULTIMEDIA	VARCHAR2(256 BYTE)	No	Nombre del archivo del contenido multimedia

Tabla 8.10 GC_MULTIMEDIA



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_PREGUNTAS			
Nombre de secuencia: SEQ_GC_PREGUNTAS			
Nombre del atributo	Tipo de dato	Null	Descripción
N_PREGUNTA_ID	NUMBER	No	Llave primaria de la tabla
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	No	Código del centro de servicio
S_PREGUNTA	VARCHAR2(256 BYTE)	No	Texto de la pregunta
N_PONDERACION	NUMBER	No	Cuantificación de valor de la pregunta con respecto a la nota global de la evaluación
D_PREGUNTA	VARCHAR2(512 BYTE)	Yes	Descripción del registro
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuario que crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuario que modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado

Tabla 8.11 GC_PREGUNTAS



MANUAL TECNICO
MODULO DE ADMINISTRACION

GC_PREGUNTAS			
Nombre de secuencia: SEQ_GC_PREGUNTAS			
Nombre del atributo	Tipo de dato	Null	Descripción
N_PREGUNTA_ID	NUMBER	No	Llave primaria de la tabla
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	No	Código del centro de servicio
S_PREGUNTA	VARCHAR2(256 BYTE)	No	Texto de la pregunta
N_PONDERACION	NUMBER	No	Cuantificación de valor de la pregunta con respecto a la nota global de la evaluación
D_PREGUNTA	VARCHAR2(512 BYTE)	Yes	Descripción del registro
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuario que crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuario que modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado

Tabla 8.12 GC_PREGUNTAS



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_RESPUESTAS			
Nombre de secuencia: SEQ_GC_RESPUESTAS			
Nombre del atributo	Tipo de dato	Null	Descripción
N_RESPUESTA_ID	NUMBER	No	Llave primaria de la tabla
N_PREGUNTA_ID	NUMBER	No	Referencia a la pregunta a la que pertenece la respuesta
S_RESPUESTA	VARCHAR2(128 BYTE)	No	Texto de la respuesta
N_ESCALA	NUMBER	No	Ponderación numérica que tiene la pregunta
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado

Tabla 8.13 GC_RESPUESTAS



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_PRIORIDAD			
Nombre de secuencia: SEQ_GC_PRIORIDAD			
Nombre del atributo	Tipo de dato	Null	Descripción
N_PRIORIDAD_ID	NUMBER	No	Llave primaria de la tabla
N_PESO	NUMBER	No	Cuantificación de la prioridad
D_PRIORIDAD	VARCHAR2(256 BYTE)	Yes	Descripción del registro
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no
B_FILA_ESP	NUMBER(1,0)	No	Si será asignado a la fila especial
S_NOMBRE	VARCHAR2(256 BYTE)	Yes	Nombre de la prioridad

Tabla 8.14 GC_PRIORIDAD



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_SERVICIOS			
Nombre de secuencia: SEQ_GC_PRIORIDAD			
Nombre del atributo	Tipo de dato	Null	Descripción
N_SERVICIOS_ID	NUMBER(38,0)	No	Llave primaria de la tabla
D_SERVICIOS	VARCHAR2(256 BYTE)	Yes	Descripción del registro
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
N_ORDEN	NUMBER	Yes	Orden en que se presentaran los servicios en pantalla
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no
S_NOMBRE	VARCHAR2(256 BYTE)	Yes	Nombre del centro de servicio

Tabla 8.15



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_TIPO_EQUIPO			
Nombre de secuencia: SEQ_GC_TIPO_EQUIPO			
Nombre del atributo	Tipo de dato	Null	Descripción
N_TIPO_ID	NUMBER	No	Llave primaria de la tabla
D_TIPO_EQUIPO	VARCHAR2(512 BYTE)	Yes	Descripción del registro
S_NOMBRE	VARCHAR2(256 BYTE)	Yes	Nombre del tipo de equipo

Tabla 8.16 GC_TIPO_EQUIPO



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_TRAMITE			
Nombre de secuencia: SEQ_GC_TRAMITE			
Nombre del atributo	Tipo de dato	Null	Descripción
N_TRAMITE_ID	NUMBER	No	Llave primaria de la tabla
N_SERVICIOS_ID	NUMBER	No	Referencia al servicio a que pertenece el trámite
N_PESO	NUMBER	No	Cuantificación de la prioridad que debe tener el tramite luego de aplicarle todos los criterios
B_NIT_REQUERIDO	NUMBER(1,0)	No	Bandera que indica si es obligatorio proporcionar el nit
B_ESCALAMIENTO	NUMBER(1,0)	No	Bandera que indica si el tramite será escalado al supervisor
D_TRAMITE	VARCHAR2(256 BYTE)	Yes	Descripción del registro
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
B_ACTIVA	NUMBER(1,0)	No	Bandera que indica si el registro esta activo o no
S_NOMBRE	VARCHAR2(256 BYTE)	No	Nombre del tramite
N_ORDEN	NUMBER	Yes	Orden en que se presentaran los tramites en pantalla

Tabla 8.17 GC_TRAMITE



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

GC_CONF_TRAMITE			
Nombre de secuencia: SEQ_GC_CONF_TRAMITE			
Nombre del atributo	Tipo de dato	Null	Descripción
N_CONF_TRA_ID	NUMBER	No	Llave primaria de la tabla
C_UNIDAD_RECEP	VARCHAR2(5 BYTE)	No	Código del centro de servicio
N_TRAMITE_ID	NUMBER	No	Referencia al tramite a que se refiere la configuración
N_ATENCION_PROM	NUMBER	No	Tiempo promedio de atención
N_TIEMPO_ESPERA	NUMBER	Yes	Tiempo, en minutos, de espera máximo que debe tardarse en la cola
N_TIEMPO_HOLGURA	NUMBER	Yes	Tiempo, en minutos, que debe esperarse antes de incluirlo en la cola
C_USUARIO_CREA	VARCHAR2(100 BYTE)	No	Código del usuarioque crea el registro
C_USUARIO_MODI	VARCHAR2(100 BYTE)	No	Código del usuarioque modifica el registro
FI_VIGENCIA	DATE	No	Fecha en que el registro es creado
FF_VIGENCIA	DATE	Yes	Fecha en que el registro deja de tener vigencia
F_MODIFICA	DATE	Yes	Fecha en que el registro fue modificado
N_COMPORTAMIENTO	NUMBER(6,0)	Yes	Comportamiento: 0-al final, 1-al inicio, 2-en medio, 3 por peso
N_PESO	NUMBER(6,0)	Yes	peso de la reasignacion
N_PROM_ESPERA	NUMBER	Yes	Promedio real (en segundos) de espera en cola

Tabla 8.17 GC_CONF_TRAMITE



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

TB_LISTAS_VALOR			
Nombre de secuencia: --			
Nombre del atributo	Tipo de dato	Null	Descripción
C_LISTA	VARCHAR2(50 BYTE)	No	Codigo de Parametro o Lista
C_MODULO	VARCHAR2(50 BYTE)	No	Codigo de Modulo
D_LISTA	VARCHAR2(255 BYTE)	Yes	Descripcion de la lista.
B_STATUS	NUMBER(1,0)	Yes	Estado, Activo o Inactivo , 1 o 0
C_USUARIO	VARCHAR2(30 BYTE)	Yes	Usuario
F_INGRESO	DATE	Yes	Fecha de ingreso
SYSTEM_VALUE	NUMBER(1,0)	Yes	Es valor de sistema para 1/0, el usuario puede modificar el parametro cuando es 0

Tabla 8.18 TB_LISTAS_VALOR



MANUAL TECNICO MODULO DE ADMINISTRACION

8. Seguridad

Estos son los roles de seguridad usados en el Módulo de Declaracion Sugerida de Renta:

Seguridad y Roles para ambiente de desarrollo y test

Nombre de la opción	Url	Roles
Configuracion de Usuarios	/admin/	ROLE_GC_R3
Mantenimiento de Alertas	/alerta/	ROLE_GC_R3
Mantenimiento de Configuración de Llamado	/cllamado/	ROLE_GC_R3
Configuracion de Tiquete	/conftiquete/	ROLE_GC_R3
Mantenimiento de Configuración de Tramite	/ctramite/	ROLE_GC_R3
Mantenimiento de Equipos	/tequipo/	ROLE_GC_R3
Mantenimiento de Escritorios	/escritorio/	ROLE_GC_R3
Mantenimiento de Lista de Reproducción	/cllamado/	ROLE_GC_R3
Mantenimiento de Multimedia	/cllamado/	ROLE_GC_R3
Mantenimiento de Configuración - Evaluación	/adminpreguntas/	ROLE_GC_R3
Mantenimiento de Prioridades - Catálogo	/prioridad/	ROLE_GC_R3
Mantenimiento de Secciones - Catálogo	/servicio/	ROLE_GC_R3
Mantenimiento de Tipos de Equipos - Catálogo	/tequipo/	ROLE_GC_R3



**MANUAL TECNICO
MODULO DE ADMINISTRACION**

Nombre de la opción	Url	Roles
Mantenimiento de Trámites - Catálogo	/ctramite/	ROLE_GC_R3
Mantenimiento de Zonas - Catálogo	/zona/	ROLE_GC_R3
Configuración de Reasignación	/creasignacion/	ROLE_GC_R3

Tabla 10.1 Seguridad y Roles



MANUAL TECNICO MODULO DE ADMINISTRACION

1. Anexos Anexo 1

GESTION.COLAS.GC_ALERTA	
P * N_ALERTA_ID	NUMBER (*,0)
* N_SERVICIOS_ID	NUMBER
* C_UNIDAD_RECEP	VARCHAR2 (5 BYTE)
* S_MENSAJE	VARCHAR2 (1024 BYTE)
* S_TIPO	VARCHAR2 (15 BYTE)
FF_NOTIFICACION	DATE
* B_CORREO	NUMBER (1)
* B_PANTALLA	NUMBER (1)
C_USUARIO_NOTIFICA	VARCHAR2 (1024 BYTE)
S_CORREO_NOTIFICA	VARCHAR2 (1024 BYTE)
* C_USUARIO_CREA	VARCHAR2 (100 BYTE)
* C_USUARIO_MODI	VARCHAR2 (100 BYTE)
* FI_VIGENCIA	DATE
FF_VIGENCIA	DATE
F_MODIFICA	DATE
* B_ACTIVA	NUMBER (1)
PK_GC_ALERTA (N_ALERTA_ID)	
PK_GC_ALERTA (N_ALERTA_ID)	
IDX_GC_ALERTA_UNIDAD (C_UNIDAD_RECEP)	

GESTION.COLAS.GC_PRIORIDAD	
P * N_PRIORIDAD_ID	NUMBER
* N_PESO	NUMBER
D_PRIORIDAD	VARCHAR2 (256 BYTE)
* C_USUARIO_CREA	VARCHAR2 (100 BYTE)
* C_USUARIO_MODI	VARCHAR2 (100 BYTE)
* FI_VIGENCIA	DATE
FF_VIGENCIA	DATE
F_MODIFICA	DATE
* B_ACTIVA	NUMBER (1)
* B_FILA_ESP	NUMBER (1)
S_NOMBRE	VARCHAR2 (256 BYTE)
PK_GC_PRIORIDAD (N_PRIORIDAD_ID)	
PK_GC_PRIORIDAD (N_PRIORIDAD_ID)	

GESTION.COLAS.GC_CONF_LLAMADO	
* C_UNIDAD_RECEP	VARCHAR2 (5 BYTE)
* N_NUM_LLAMADAS	NUMBER
* N_INTERVALO_LLAMADA	NUMBER
* B_ACTIVA	NUMBER (1)
* S_MENSAJE	VARCHAR2 (1024 BYTE)
* C_USUARIO_CREA	VARCHAR2 (100 BYTE)
* C_USUARIO_MODI	VARCHAR2 (100 BYTE)
* FI_VIGENCIA	DATE
FF_VIGENCIA	DATE
F_MODIFICA	DATE
S_DESCRIPCION	VARCHAR2 (256 BYTE)
P * N_CONFLAMADO_ID	NUMBER
GC_CONF_LLAMADO_PK (N_CONFLAMADO_ID)	
GC_CONF_LLAMADO_PK (N_CONFLAMADO_ID)	
IDX_GC_CONFLAMADO_UNID (C_UNIDAD_RECEP)	

GESTION.COLAS.GC_MULTIMEDIA	
P * N_MULTIMEDIA_ID	NUMBER (*,0)
* S_TIPO	VARCHAR2 (256 BYTE)
* N_DURACION	NUMBER
* FI_MULTIMEDIA	DATE
FF_MULTIMEDIA	DATE
D_MULTIMEDIA	VARCHAR2 (256 BYTE)
* C_USUARIO_CREA	VARCHAR2 (100 BYTE)
* C_USUARIO_MODI	VARCHAR2 (100 BYTE)
* FI_VIGENCIA	DATE
FF_VIGENCIA	DATE
F_MODIFICA	DATE
* B_ACTIVA	NUMBER (1)
S_RUTA	VARCHAR2 (1024 BYTE)
X_ARCHIVO	BLOB
* S_MULTIMEDIA	VARCHAR2 (256 BYTE)
PK_GC_MULTIMEDIA (N_MULTIMEDIA_ID)	
PK_GC_MULTIMEDIA (N_MULTIMEDIA_ID)	

GESTION.COLAS.GC_CONF_TIQUETE	
* C_UNIDAD_RECEP	VARCHAR2 (5 BYTE)
* B_ACTIVA	NUMBER (1)
* X_CONTENIDO	CLOB
* C_USUARIO_CREA	VARCHAR2 (100 BYTE)
* C_USUARIO_MODI	VARCHAR2 (100 BYTE)
FF_VIGENCIA	DATE
* FI_VIGENCIA	DATE
F_MODIFICA	DATE
P * N_CONF_TIQ_ID	NUMBER
N_VERSION	NUMBER
X_IMAGEN	BLOB
GC_CONF_TIQUETE_PK (N_CONF_TIQ_ID)	
GC_CONF_TIQUETE_PK (N_CONF_TIQ_ID)	
IDX_GCCONFTIQUETE_UNID (C_UNIDAD_RECEP)	

GESTION.COLAS.GC_RESPUESTAS		
P *	N_RESPUESTA_ID	NUMBER
F *	N_PREGUNTA_ID	NUMBER
*	S_RESPUESTA	VARCHAR2 (128 BYTE)
*	N_ESCALA	NUMBER
*	C_USUARIO_CREA	VARCHAR2 (100 BYTE)
*	C_USUARIO_MODI	VARCHAR2 (100 BYTE)
*	FI_VIGENCIA	DATE
	FF_VIGENCIA	DATE
	F_MODIFICA	DATE
PK_GC_RESPUESTAS (N_RESPUESTA_ID)		
FK_GCRESP_GCPREG (N_PREGUNTA_ID)		
PK_GC_RESPUESTAS (N_RESPUESTA_ID)		
IDX_GCRESP_NPREGID (N_PREGUNTA_ID)		



GESTION.COLAS.GC_PREGUNTAS		
P *	N_PREGUNTA_ID	NUMBER
*	C_UNIDAD_RECEP	VARCHAR2 (5 BYTE)
*	S_PREGUNTA	VARCHAR2 (256 BYTE)
*	N_PONDERACION	NUMBER
*	D_PREGUNTA	VARCHAR2 (512 BYTE)
*	C_USUARIO_CREA	VARCHAR2 (100 BYTE)
*	C_USUARIO_MODI	VARCHAR2 (100 BYTE)
*	FI_VIGENCIA	DATE
	FF_VIGENCIA	DATE
	F_MODIFICA	DATE
PK_GC_PREGUNTAS (N_PREGUNTA_ID)		
PK_GC_PREGUNTAS (N_PREGUNTA_ID)		
IDX_GCPREGUN_UNID (C_UNIDAD_RECEP)		

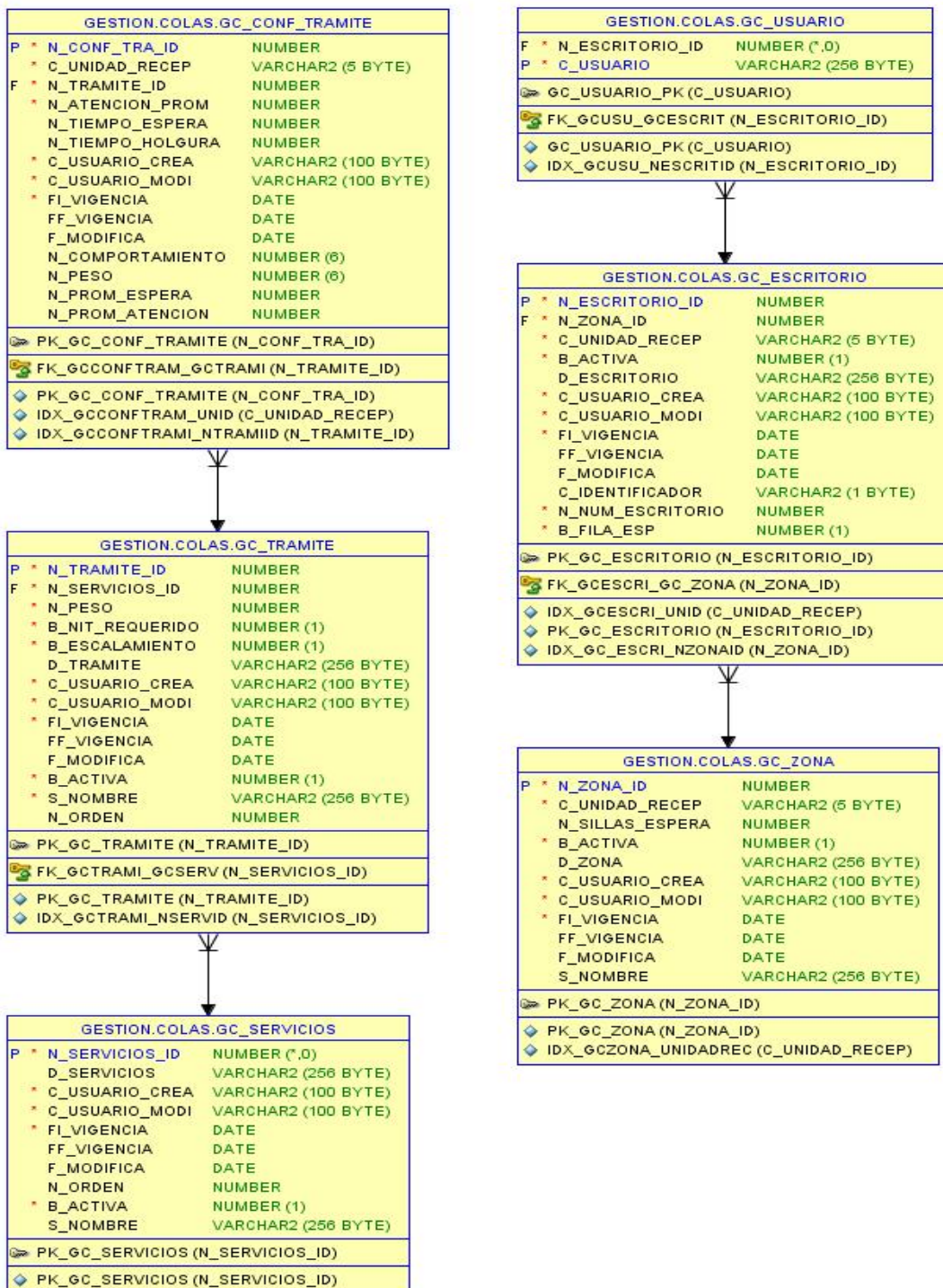
GESTION.COLAS.GC_EQUIPOS		
P *	N_EQUIPO_ID	NUMBER (*,0)
*	C_UNIDAD_RECEP	VARCHAR2 (5 BYTE)
F *	N_TIPO_ID	NUMBER
*	S_MARCA	VARCHAR2 (100 BYTE)
*	S_MODELO	VARCHAR2 (50 BYTE)
*	S_SERIE	VARCHAR2 (50 BYTE)
*	S_NUMERO_INVENTARIO	VARCHAR2 (50 BYTE)
	D_EQUIPO	VARCHAR2 (256 BYTE)
	FF_VIGENCIA	DATE
*	FI_VIGENCIA	DATE
	F_MODIFICA	DATE
*	C_USUARIO_CREA	VARCHAR2 (100 BYTE)
*	C_USUARIO_MODI	VARCHAR2 (100 BYTE)
*	B_ACTIVA	NUMBER (1)
PK_GC_EQUIPOS (N_EQUIPO_ID)		
FK_GCEQUI_GCTIPOEQUI (N_TIPO_ID)		
PK_GC_EQUIPOS (N_EQUIPO_ID)		
IDX_GCEQUI_UNID (C_UNIDAD_RECEP)		
IDX_GCEQUI_NTIPOID (N_TIPO_ID)		



GESTION.COLAS.GC_TIPO_EQUIPO		
P *	N_TIPO_ID	NUMBER
	D_TIPO_EQUIPO	VARCHAR2 (512 BYTE)
	S_NOMBRE	VARCHAR2 (256 BYTE)
PK_GC_TIPO_EQUIPO (N_TIPO_ID)		
PK_GC_TIPO_EQUIPO (N_TIPO_ID)		



MANUAL TECNICO MODULO DE ADMINISTRACION





MANUAL TECNICO MODULO DE ADMINISTRACION

Anexo 2

url:

CATALOGOS.TB_LISTAS_VALOR		
P *	C_LISTA	VARCHAR2 (50 BYTE)
P *	C_MODULO	VARCHAR2 (50 BYTE)
	D_LISTA	VARCHAR2 (255 BYTE)
	B_STATUS	NUMBER (1)
	C_USUARIO	VARCHAR2 (30 BYTE)
	F_INGRESO	DATE
	SYSTEM_VALUE	NUMBER (1)
PK_LISTAS_VALOR (C_LISTA, C_MODULO)		
PK_LISTAS_VALOR (C_LISTA, C_MODULO)		

CATALOGOS.TB_UNIDAD_RECEP		
P *	C_UNIDAD_RECEP	VARCHAR2 (5 BYTE)
*	C_USUARIO	VARCHAR2 (30 BYTE)
*	D_UNIDAD_RECEP	VARCHAR2 (40 BYTE)
	FH_INGRESO	DATE
*	B_STATUS	NUMBER (1)
*	M_TIPO_UNIDAD	VARCHAR2 (1 BYTE)
*	B_DESPLEGABLE	NUMBER (1)
	C_UNIDAD	VARCHAR2 (1 BYTE)
	C_UNIDAD_DGT	VARCHAR2 (25 BYTE)
	C_DEP_MUN	VARCHAR2 (4 BYTE)
	S_UBIC_GEOGRAF	VARCHAR2 (60 BYTE)
	C_UNIDAD_RECEP_SUP	VARCHAR2 (5 BYTE)
	M_SERVICIO	VARCHAR2 (1 BYTE)
PK_TB_UNIDAD_RECEP (C_UNIDAD_RECEP)		
IDX_TB_UNIDAD_RECEP_UNIDAD_STA (B_STATUS, M_TIPO_UNIDAD)		
PK_TB_UNIDAD_RECEP (C_UNIDAD_RECEP)		
IX_TBUNRE_UNIDGT (C_UNIDAD_DGT)		
IDX_TB_UNIDAD_RECEP_C_USUARIO (C_USUARIO)		