Informatica y logistica

MANUAL TECNICO MODULO OPERACION

SISTEMA DE ATENCION AL CONTRIBUYETE (SAC)

Contenido

[1. Breve descripción del Módulo de Operación. 2](#_Toc433197625)

[2. Diagrama Entidad – Relación 3](#_Toc433197626)

[3. Estructura de Componentes (Capa vista, controlador, servicio DAO y estructura de reportes) 4](#_Toc433197627)

[Vistas o páginas HTML 4](#_Toc433197628)

[Controladores 4](#_Toc433197629)

[Descripción de controladores 4](#_Toc433197630)

[4. Repositorios. 8](#_Toc433197631)

[Ubicación de repositorios. 8](#_Toc433197632)

[Descripción de Repositorios 8](#_Toc433197633)

[5. Componentes. 10](#_Toc433197634)

[Listado de Componentes 10](#_Toc433197635)

[6. Configuración de proyectos (@Configuration y @Beans en general, esto es todo el paquete de sv.gob.mh.dgii.colas.config) 11](#_Toc433197636)

[7. Documentos Relacionados a la Base de Datos 14](#_Toc433197637)

[Tablas 14](#_Toc433197638)

[8. Seguridad 19](#_Toc433197639)

[Seguridad y Roles para ambiente de desarrollo y test 19](#_Toc433197640)

[Anexos 20](#_Toc433197641)

[Anexo 1 20](#_Toc433197642)

# Breve descripción del Módulo de Operación.

El módulo de Operación tiene por objetivo el llevar un registro detallado de las actividades realizadas por los técnicos de atención durante su jornada laboral y evaluar el nivel de satisfacción de los contribuyentes con respecto a los tiempos de espera, tiempo de atención de los servicios que se presta en los centros de servicio de la DGII.

En este sub módulo podrá: Gestionar a contribuyentes en cola, Reasignación y Escalamiento de contribuyentes en trámite, Tomar Pausa o Receso y Realizar Evaluación.

# Diagrama Entidad – Relación

A continuación se listan las diferentes tablas utilizadas en el Módulo deOperación seguidos de los nombres de los esquemas a los cuales pertenecen.

| Nombre del esquema | Nombres de tablas | Anexo |
| --- | --- | --- |
| GESTIÓN COLAS | GC\_CONF\_LLAMADO  GC\_CONF\_TRAMITE  GC\_PREGUNTAS  GC\_PREGUNTAS\_RESPUESTAS  GC\_RESPUESTAS  GC\_TIQUETE  GC\_TRAMITE  GC\_USER\_LOG  GC\_USUARIO  GC\_JERARQUIA  GC\_JERARQUIA\_SECCION | **VER ANEXO 1** |
| CATALOGOS | TB\_LISTAS\_VALOR | **VER ANEXO 2** |

TABLA 2.1 Nombre de Esquemas y Tablas respectivas.

# Estructura de Componentes (Capa vista, controlador, Repositorios)

## Vistas o páginas HTML

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

|  |  |  |
| --- | --- | --- |
| Nombre de opción | Ubicación | Nombre |
| Operación | /Other Sources/src/main/resources/templates.operacion | operacion.html |

TABLA 3.1 Nombre de las paginas HTML y su ubicación por opción.

## Controladores

En la siguiente tabla se muestran todas las clases controladores que se usan en el Modulo de Operación y la ubicación del paquete en el cual se encuentran.

|  |  |  |
| --- | --- | --- |
| Nombre de opción | Ubicación | Nombre |
| Operación | sv.gob.mh.dgii.colas.controllers.operacion | OperacionController |
|  |  |

TABLA 3.2 Listado de los controladores por opción y su ubicación.

## Descripción de controladores

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

| **PANTALLA OPERACIÓN.** | |
| --- | --- |
| **Controlador:** | **OperacionController.java** |
| **Descripción de Métodos de Negocio** | |
| **@RequestMapping("/operacion/home")**  **public String operacion(ModelMap map):**  Método encargado de iniciar el mapeo de los objetos.  **@RequestMapping(value = "/siguiente", method = RequestMethod.POST, headers = "Accept=application/json", produces = MediaType.APPLICATION\_JSON\_VALUE, consumes = MediaType.APPLICATION\_JSON\_VALUE)**  **public @ResponseBody Map<String, Object> tra(@RequestBody Map<String, Long> map):**  Método encargado de llamar al número siguiente de la cola, para lo cual primero compara, que gcTiqueteId sea diferente de cero para obtener un mapeo de las variables, seteando el estado con setMEstado invocado por gcTiqueteNoAtendido, seteando a setcUsuarioAtendio usando a gcTiqueteNoAtendido para invocarlo y luego haciendo un save con gcTiqueteRepository.  gMetodo que Genera el mapeo de las variables y setea el valor a siguiente con gcTiqueteEnAtencion, si gcTiqueteEnAtencion es igual a nulo de despliega el mensaje: "No hay contribuyentes en la cola". Si tiene valores gcTiqueteEnAtencion entonces se declara el String turno para invocar después de desplegar el mensaje "Llamando al número: " el correlativo con getSCorrelativo. Coloca las variables a siguiente y obtiene el Id del tiquete con getNTiqueteId y correlativo usando getSCorrelativo, para llamar a siguiente con el método llamar y colocar tiquete usando el correlativo y escritorio con getNEscritorioId llamando a getCIdentificador.  **private GcTiquete siguiente():**  Método que se encarga de generar un tiquete validando el escalamiento para el usuario y establece los parámetros como fecha del llamado y hora  **private String getUnidad():**  Método encargado de devolver unidad. Crea una variable de tipo string llamada unidad la cual esta compuesta por getUnidadRecep invocada mediante getPrincipal, valida que si unidad es nula entonces unidad es getUbicacionFisica invocado por getPrincipal.  **@RequestMapping(value = "/iniciar\_tramite", method = RequestMethod.POST, headers = "Accept=application/json", produces = MediaType.APPLICATION\_JSON\_VALUE, consumes = MediaType.APPLICATION\_JSON\_VALUE)**  **public @ResponseBody**  **TiqueteDao initTram(@RequestBody Map<String, Long> map):**  Método encargado de devolver miTiquete el cual lleva seteado los valores de setFhiProceso, setsCorrelativo, y setnTramiteId. Este método se encarga de encontrar y mapear el ID del tiquete que continua, la búsqueda es realizada mediante findOne, setea el estado de 3 para un llamado y luego setea de los datos en gcTiquete para luego con gcTiquete invocar y setar los datos en miTiquete. Los datos incluyen el formato h:mm:ss.  **@RequestMapping(value = "/finalizar\_tramite", method = RequestMethod.POST, headers = "Accept=application/json", produces = MediaType.APPLICATION\_JSON\_VALUE, consumes = MediaType.APPLICATION\_JSON\_VALUE)**  **public @ResponseBody**  **TiqueteDao finalizarTram(@RequestBody Map<String, Long> map):**  Este método se encarga de devolver dao. Usa GcTiquete para encontrar y mapear el ID del tiquete que continua, setee en setMEstado el valor de 4, setea los tiempos de procesos con los nuevos tiempos usando getServerDateTime y hace un save. TiqueteDao declara dao y este toma el valor de un nuevo TiqueteDao.  **@RequestMapping(value = "/llamarDeNuevo", method = RequestMethod.POST, headers = "Accept=application/json", produces = MediaType.APPLICATION\_JSON\_VALUE, consumes = MediaType.APPLICATION\_JSON\_VALUE)**  **public @ResponseBody Integer enableReservacion(@RequestBody Map<String, String> map):**  Metodo que valida gcTiquete y se asegura que no vaya nulo, y este setea el estado, la fecha y obtiene la hora y luego guarda los parámetros seteados y retorna a 0  **@RequestMapping(value = "/tramites", method = RequestMethod.GET, headers = "Accept=application/json", produces = "application/json")**  **public @ResponseBody**  **List<GcTramite> tramites(ModelMap map):**  Metodo encargado de genera una lista con los tramites y seteando en ella setGcUsuarioList, setGcReservaCitaList, getNServiciosId, setGcConfTramiteList, setGcTiqueteList siempre que listTramites sea igual a nulo. Devuelve listTramites.  **@RequestMapping(value = "/pausa", method = RequestMethod.POST, headers = "Accept=application/json", produces = MediaType.APPLICATION\_JSON\_VALUE, consumes = MediaType.APPLICATION\_JSON\_VALUE)**  **public @ResponseBody Map<String, String> pausa(@RequestBody Map<String, Long> request, ModelMap map):**  Método encargado de poner en pausa la atención de tiquetes por cualquiera de los motivos siguientes: TOMAR RECESO, TOMAR PAUSA, TOMAR TIEMPO DE ALMUERZO. Usando GcUserLog declara gcUserLog y con gcUserRepository encuentra el usuario y obtiene desde la Base de Datos, a gcUserLog se le setea setCUsuario, setFhiEventom, setNEventoId, setSDetalles y hace un save. Devuelve mapa.  **@RequestMapping(value = "/finPausa", method = RequestMethod.POST, headers = "Accept=application/json", produces = MediaType.APPLICATION\_JSON\_VALUE, consumes = MediaType.APPLICATION\_JSON\_VALUE)**  **public @ResponseBody Map<String, String> finPausa(@RequestBody Map<String, Long> request, ModelMap map):**  Metodo que finaliza la pausa que ha sido creada por cualquiera de los motivos siguientes: TOMAR RECESO, TOMAR PAUSA, TOMAR TIEMPO DE ALMUERZO. Para lo cual obtiene la pausa con GcUserLog, declara usando GcUserLog para declarar gcUserLogRepository y llamar a enPausa del usuario con getUsuario. Si la pausa existe entonces setea la fecha de fin con setFhfEvento y hace un save. Devuelve mapa.  **@RequestMapping(value="/getPreguntas" , method=RequestMethod.GET, headers="Accept=application/json", produces="application/json")**  **public @ResponseBody List<Map<String,Object>> getPreguntas(ModelMap map):**  Método encargado de setear las preguntas en la opción de Realizar mantenimiento cuando se inicia el trámite del tiquete, para ello crea una lista y se llena mediante el for y en gcPreguntas coloca la pregunta dPregunta, que obtiene de getDPregunta y el id de la pregunta usando para guardarlo nPreguntaId invocado por getNPreguntaId dejando el espacio de respuesta en blanco. Genera una lista con los objetos mapeados en un arrayList. Devuelve preguntayRespuestasList.  **@RequestMapping(value = "/setRespuestas", method = RequestMethod.POST, headers = "Accept=application/json", produces = MediaType.APPLICATION\_JSON\_VALUE, consumes = MediaType.APPLICATION\_JSON\_VALUE)**  **public @ResponseBody List<Map<String,Object>> setRespuestas(@RequestBody Map<String, List> respuestas, ModelMap map):**  Método encargado de setear las respuestas en la opción de Realizar mantenimiento cuando se inicia el trámite del tiquete, en gcRespuestas coloca las posibles opciones de respuestas mapeando una lista en cada pregunta desplegada. Devuelve preguntayRespuestasList.  **@RequestMapping(value = "/llamado/getData", method = RequestMethod.POST, headers = "Accept=application/json", produces = MediaType.APPLICATION\_JSON\_VALUE, consumes = MediaType.APPLICATION\_JSON\_VALUE)**  **public @ResponseBody Map<String, String> llamadoGetData(@RequestBody Map<String, String> centro)**  Metodo encargado del proceso de llamado para el siguiente tiquete según el correlativo que tenga  **@RequestMapping(value="/operacion/getTramites" , method=RequestMethod.GET, headers="Accept=application/json", produces="application/json")**  **public @ResponseBody List<GcConfTramite> servicios(ModelMap map):**  Metodo encargado de declarar el listTramites y a lUnidadRecep del tipo List y los inicializa a nulo para tratar de declarar una unidad de Recepcion con el string, crear un tipo List con los catalogos utilizando TbListasValor y con un for llegan la lista de unidadRecep agregando con getClista que es invocado mediante lUnidadRecep.  Inicializa la listTramites con gcConfTramiteRepository que invoca a listaTramitesYEscalamientoByCS y mediante un for hace el llenado de la tabla con tramite invoca a getNTramite y este invoca a setGcUsuarioList, setGcReservaCitaList, setGcTiqueteList, getNServiciosId y setGcConfTramiteList mientras que getCUnidadRecep invoca a setRcTramites, setEdDeclaracioneses y setEdNotaAbonos. Devuelve listTramites.  **@RequestMapping(value="/operacion/getEscalamiento/{idTiquete}" , method=RequestMethod.*GET*, headers="Accept=application/json", produces="application/json")**  **public @ResponseBody List<Map<String,String>> escalamiento(@PathVariable(value = "idTiquete") long idTiquete, ModelMap map)**  Metodo que declara el jerarquiByTiqueteList y los inicializa comparándolo con gcJerarquiaRepository y mediante un for de GcJerarquia gcJerarquia : jeraruiaByTiqueteIdList hace el llenado de la tabla, mapeando nTramiteId y sNombre  **@RequestMapping(value = "/operacion/realizarEscalamiento", method = RequestMethod.*POST*, headers = "Accept=application/json", produces = MediaType.*APPLICATION\_JSON\_VALUE*, consumes = MediaType.*APPLICATION\_JSON\_VALUE*) public @ResponseBody Map<String,String> realizarEscalamiento(@RequestBody Map<String, Long> escalamiento)**  Según el ID del tiquete en atención se determina cuáles son los jefes o supervisores disponibles para que los contribuyentes realicen el escalamiento, después este se retorna la lista para ser mostrada en pantalla. | |

TABLA 3.3 OperacionController.java

## JavaScript

En la siguiente tabla se muestran todos los archivos javascript que se usan en el Modulo de Operación y la ubicación del paquete en el cual se encuentran.

|  |  |  |
| --- | --- | --- |
| Nombre de opción | Ubicación | Nombre |
| Operación | Src/main/resources/static/js | Operación.js |
|  |  |

TABLA 3.4 Listado de los archivos js y su ubicación.

## Descripción de JavaScript

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

| **PANTALLA OPERACIÓN.** | |
| --- | --- |
| **JavaScript:** | **Operacion.js** |
| **Descripción de Métodos de Negocio** | |
| **function siguiente():**  Método encargado de ir al controlador y traer el siguiente tiquete se utiliza el método ajax con la siguiente url /colas/llamarDeNuevo y se le envía la variable “gcTiqueteId”.  **function llamardeNuevo():**  Método encargado de llamar de nuevo al tiquete después de esperar el tiempo configurado en la aplicación si la atención no ha comenzado.  **function startTime():**  Método encargado de contar el tiempo de atención.  **$("#iniciar\_tramite").click(function():**  Función que se utiliza para iniciar y finalizar un trámite, primero se coloca 0 a la variable llamando para indicar que ya estamos en atención, se verifica si se va a finalizar o a iniciar la atención por medio de la variable “atendiendo” si la variable tiene valor de 0 indica que vamos a iniciar la atención y si no indica que la vamos a terminar la atención. Si el caso es iniciar la atención bloqueamos los botones de llamar siguiente, tomar pausa o receso y salir, luego usa el método del controlador para iniciar la atención, con la url “/colas/iniciar\_tramite” muestra el cuadro de información de contribuyente en atención y se inicializa el método para el conteo de tiempo que durara la atención, si el caso es finalizar el tramite, se desbloquean los botones de llamar al siguiente, tomar pausa o receso, salir, y se bloquean los botones de iniciar la atención, escalmiento y reasignación y se procede a consumir el método que finaliza la atención utilizando la url siguiente “/colas/finalizar\_tramite” y le eviamos el id del tiquete que queremos finalizar la atención.  **$("#reasignar").click(function ():**  Función que se encarga de realizar la reasignación de un tiquete usando l siguiente url “/colas/me/insertData” y se le envía el siguiente json = { "tramite" : nTramiteId ,"prioridad" : 0,"holgura" : 0, "opt" : 0, "idTiquete" : varGcTiqueteIdReasig } donde nTramiteId es el tramite al cual se va hacer la reasignación, idTiquete es el id del tiquete al cual se le realiza la reasignación.  **$("#escalar").click(function() :**  Función que se encarga de realizar el escalamiento mediante la url “colas/operacion/realizarEscalamiento” para la cual se le envían los parámetros de jerarquiaId, idTiquete para realizar el escalamiento. | |

# Repositorios.

## Ubicación de repositorios.

|  |  |  |
| --- | --- | --- |
| Nombre de la opción | Ubicación | Nombre |
| OPERACION | sv.gob.mh.dgii.colas.repositories | **GcConfLlamadoRepository** |
| **GcConfTramiteRepository** |
| **GcEventosRepository** |
| **GcPreguntasRepository** |
| **GcPreguntasRespuestasRepository** |
| **GcRespuestasRepository** |
| **GcTiqueteRepository** |
| **GcTramiteRepository** |
| **GcUnidadRecepRepository** |
| **GcUserLogRepository** |
| **GcUsuarioRepository** |
| **RcRucRepository** |
| **TbListasValorRepository** |
| **GcJerarquiaRepository** |
| **GcJerarquiaSeccionRepository** |

TABLA 4.1 Listado de repositorios

## Descripción de Repositorios

En la siguiente tabla se describen cada uno de los repositorios usados en el módulo Operación.

|  |  |
| --- | --- |
| OPCION: OPERACION | |
| Repositorio | **Descripción de métodos** |
| GcConfLlamadoRepository | El repositorio utiliza los métodos siguientes:  **public List<GcConfLlamado> getConfsByCS(String cunidadRecep):** Método que utiliza el Query "SELECT g FROM GcConfLlamado g WHERE g.bActiva=1 and c\_unidad\_recep = ?1" |
| GcConfTramiteRepository | El repositorio utiliza los métodos siguientes:  **public List<GcConfTramite> listaTramitesYEscalamientoByCS(List<String> unidadRecep):** Método que utiliza el Query "SELECT c FROM GcConfTramite c where C\_UNIDAD\_RECEP in( ?1 ) and (c.nAtencionProm > 0 or c.nTiempoEspera is not null or c.nTiempoHolgura is not null) and c.nTramiteId.bActiva = 1 ORDER BY nTramiteId.nServiciosId.nServiciosId ASC "  **public List<GcConfTramite> listaTramitesByCS2(String unidadRecep):** Metodo que utiliza el Query "SELECT c FROM GcConfTramite c where C\_UNIDAD\_RECEP = ?1 and (c.nAtencionProm > 0 or c.nTiempoEspera is not null or c.nTiempoHolgura is not null) and c.nTramiteId.bActiva = 1 and c.nTramiteId.bEscalamiento = 0 ORDER BY nTramiteId.nServiciosId.nServiciosId ASC " |
| GcEventosRepository | **METODO GENERICO FINDONE.** |
| GcPreguntasRepository | El repositorio utiliza los métodos siguientes:  **public List<GcPreguntas> preguntaYrespuestas(String cUnidadRecep):** Este método utiliza el query "SELECT c FROM GcPreguntas c where C\_UNIDAD\_RECEP = ?1 " |
| GcPreguntasRespuestasRepository | **METODO GENERICO SAVE.** |
| GcRespuestasRepository | El repositorio utiliza los métodos siguientes:  **public List<GcRespuestas> respuestasByPregutna(Long nPreguntaId):** Método que ejecuta el query "SELECT c FROM GcRespuestas c where N\_PREGUNTA\_ID = ?1 " |
| GcTiqueteRepository | El repositorio utiliza los métodos siguientes:  **Public List<GcTiquete> getNext(@Param("cUnidadRecep")String cUnidadRecep,@Param("nTramiteId")List<Long> nTramiteId):** Metodo para ejecutar el query "SELECT t FROM GcTiquete t INNER JOIN t.nTramiteId s INNER JOIN t.nPrioridadId p WHERE t.cUnidadRecep.cunidadRecep = :cUnidadRecep AND t.mEstado = 1 AND trunc(FH\_LLEGADA) =trunc(sysdate) AND t.nTramiteId.nTramiteId in(:nTramiteId)"  **public Timestamp getServerDateTime()**: Metodo encargado de ejecutar el query "SELECT SYSDATE FROM DUAL".  **public List<GcTiquete> getEnEsperaSeccionList(@Param("cUnidadRecep")String cUnidadRecep);**    @Query("SELECT t FROM GcTiquete t INNER JOIN t.nTramiteId s INNER JOIN t.nPrioridadId p WHERE t.cUnidadRecep.cunidadRecep = :cUnidadRecep AND t.mEstado = 1 AND trunc(FH\_LLEGADA) = trunc(sysdate) ORDER BY FH\_LLEGADA DESC") |
| GcTramiteRepository | El repositorio utiliza los métodos siguientes:  **public List<GcTramite> getTramitesByUser(String user):** Este método es el encargado de ejecutar el query "SELECT t FROM GcTramite t INNER JOIN t.gcUsuarioList u WHERE u.cUsuario = ?1 ".  **public List<GcTramite> getTodosTramitesByServiceId(Long servicioId);** Este método es el encargado de ejecutar el query  @Query("SELECT t FROM GcTramite t WHERE t.nServiciosId.nServiciosId = ?1 ORDER BY nTramiteId ASC ") |
| GcUnidadRecepRepository | **public List<TbUnidadRecep> getUnidadesEnServicio();**  @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')") |
| GcUserLogRepository | El repositorio utiliza los métodos siguientes:  **public List<GcUserLog> enPausa(@Param("cUsuario")String cUsuario):** "SELECT t FROM GcUserLog t where t.cUsuario.cUsuario = :cUsuario AND fhfEvento IS NULL AND N\_EVENTO\_ID IN( '1','2','3')" |
| GcUsuarioRepository | **METODO GENERICO FINONE.** |
| RcRucRepository |  |
| GcJerarquiaSeccionRepository | El repositorio utiliza los métodos siguientes  **public List<GcJerarquiaSeccion> findAllByUnidad(String cUnidadRecep);**  @Query("SELECT t FROM GcJerarquiaSeccion t where t.cUnidadRecep = ?1")    **public String getJerarquiaSecByNameId(String unidadRecep,Long nServiciosId,Long nJerarquiaId,Long nJrqSecId);**  @Query("SELECT t.sUsuario FROM GcJerarquiaSeccion t where C\_UNIDAD\_RECEP = ?1 and N\_SERVICIOS\_ID = ?2 and N\_JERARQUIA\_ID=?3 and N\_JRQ\_SEC\_ID != ?4")    **public Long getMaxJerarquiaSecId();**  @Query("SELECT max(t.nJrqSecId)+1 FROM GcJerarquiaSeccion t ")    **public List<GcJerarquiaSeccion> getJerarquiaBySeccion(String cUnidadRecep, Long nServiciosId);**  @Query("SELECT t FROM GcJerarquiaSeccion t where t.cUnidadRecep = ?1 and t.nServiciosId.nServiciosId = ?2 ")  **public Long getRelacionGerarquia(Long nJerarquiaId);**  @Query("SELECT count(t) FROM GcJerarquiaSeccion t where nJerarquiaId.nJerarquiaId= ?1") |
| GcJerarquiaRepository | **public String getJerarquiaByName(String sJerarquia);**  Método que ejecuta la siguiente sentencia:  @Query("SELECT t.sJerarquia FROM GcJerarquia t where REPLACE(upper(t.sJerarquia),' ','') = REPLACE(upper(?1) ,' ','')")    public String getJerarquiaByNameId(String sJerarquia,Long nJerarquiaId);  Método que ejecuta la siguiente sentencia:  @Query("SELECT t.sJerarquia FROM GcJerarquia t where REPLACE(upper(t.sJerarquia),' ','') = REPLACE(upper(?1) ,' ','') and t.nJerarquiaId != ?2 ")  **public Long getMaxJerarquiaId();**  Método que ejecuta la siguiente sentencia:  @Query("SELECT max(t.nJerarquiaId)+1 FROM GcJerarquia t ") |
| TbListasValorRepository | El repositorio utiliza los métodos siguientes:  **public List<TbListasValor> getUnidadesReceptoras(String unidadRecep):** Metodo encargado de ejecutar el query "SELECT det FROM TbListasValorDet det WHERE det.id.clista = (SELECT lt.id.clista FROM TbListasValor lt WHERE lt.id.clista = ( SELECT vdet.cvalor FROM TbListasValorDet vdet WHERE vdet.id.clistaDet = ?1 AND vdet.id.cmodulo = 'GC')) " |

Tabla 4.2 Nombre y descripción de repositorios

# Componentes.

## Listado de Componentes

|  |  |
| --- | --- |
| Nombre del Componente | Ubicación |
| AppInfoInterceptor.java | sv.gob.mh.dgii.colas.components |

TABLA 5.1 Listado y Ubicación de componentes

**Descripción de Componentes.**

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

TABLA 5.2 Nombre y descripción de componentes

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

| Nombre del archivo de configuración | Ubicación  sv.gob.mh.dgii.colas.config | Descripción |
| --- | --- | --- |
| HibernateConfig | **public LocalSessionFactoryBean alertsSessionFactory();**  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".  **public HibernateTransactionManager transactionManager();**  Método que ajusta la instancia que debe gestionar las transacciones y obtiene el objeto del método alertsSessionFactory().  **public HibernateExceptionTranslator exceptionTranslation();**  Método que crea una nueva instancia de **HibernateExceptionTranslator.**  **final Properties hibernateProperties();**  Método que establece las propiedades de la clase Properties   |  | | --- | |  | | | |
| RepositoryConfig | **@EnableJpaRepositories(basePackages = { "sv.gob.mh.dgii" },**  **includeFilters = @ComponentScan.Filter(pattern = ".\*.repositories.\*", type = FilterType.REGEX))**  Esta anotación me dice que paquete inyectara como repositorios.  **@ComponentScan(basePackages = "sv.gob.mh.dgii", useDefaultFilters = false, includeFilters = @Filter(pattern = ".\*.components.\*", type = FilterType.REGEX));**  Configura directivas de escaneo de componentes y me dice que paquete inyectara como repositorios. | | |
| SecurityConfig | **protected void configure(HttpSecurity http);**  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.  **public DefaultLdapAuthoritiesPopulator ldapAuthoritiesPopulator();**  Constructor de escenarios de búsqueda de grupo y  suministra los contextos utilizados para buscar roles de usuario.  **public DgiiFilterInvocationSecurityMetadataSource dgiiSecurityMetadataSource();**  Método que invoca el paquete "sv.gob.mh.dgii.colas.security.PropertyFileSecurityBuilder".  **public AuthenticationManager authenticationManagerBean();**  Método de anulación authenticationManagerBean en WebSecurityConfigurerAdapter para exponer el AuthenticationManager construido usando configure(AuthenticationManagerBuilder)  **public AffirmativeBased accessDecisionManager();**  Método que concreta de **[AccessDecisionManager](http://docs.spring.io/autorepo/docs/spring-security/3.2.3.RELEASE/apidocs/org/springframework/security/access/AccessDecisionManager.html" \o "interfaz en org.springframework.security.access)**que otorga acceso si cualquier AccessDecisionVoter devuelve una respuesta afirmativa.  **public RoleVoter roleVoter();**  Método que especifica un prefijo de rol al usuario.  **public AuthenticatedVoter authenticatedVoter();**  Método para la autenticación de usuario dependiendo si es anónima o si desea que se recuerde.  **public FilterSecurityInterceptor dgiiFilterSecurityInterceptor();**  Método que realiza el  manejo de la seguridad de los recursos HTTP a través de un filtro de aplicación.  **BaseLdapPathContextSource contextSource();**  Interfaz para ser implementado por ContextSources que son capaces de proporcionar la ruta LDAP base.  **public void configureAuthentification(AuthenticationManagerBuilder auth)**  Método que obtiene la configuración requerirá que cualquier URL que se solicita será necesario un usuario con el rol de "ROLE\_USER". | | |
| SecurityWebApplicationInitializer | Clase que extiende de AbstractSecurityWebApplicationInitializer la cual sobre-escribe los siguientes Métodos:  **public class SecurityWebApplicationInitializer extends AbstractSecurityWebApplicationInitializer.** | | |
| WebConfig | Clase que extiende de WebMvcConfigurerAdapter la cual sobre-escribe los siguientes Métodos:  **public void configureMessageConverters(List<HttpMessageConverter<?>> converters)**  Para personalizar la configuración importado, implementar la interfaz [WebMvcConfigurer](https://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/web/servlet/config/annotation/WebMvcConfigurer.html" \o "interfaz en org.springframework.web.servlet.config.annotation) o más probablemente extender el método vacío clase base [WebMvcConfigurerAdapter](https://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/web/servlet/config/annotation/WebMvcConfigurerAdapter.html" \o "clases en org.springframework.web.servlet.config.annotation) y anular métodos individuales  **public String appName();**  Método que obtiene el nombre completo de la app.  **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.    **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.  **public ViewResolver viewResolver() ;**  Método para ver si el estado no cambia durante el funcionamiento de la aplicación.  **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. | | |
| WebSocketConfig | **public void configureMessageBroker(MessageBrokerRegistry config):**  Método encargado de configurar enableSimpleBroker y setApplicationDestinationPrefixes.  **public void registerStompEndpoints(StompEndpointRegistry registry):**  Método encargado de configurar el sock en addEndpoint withSockJS. | | |

TABLA 6.1 Configuraciones del Módulo de Colas

# 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\_CONF\_LLAMADO  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\_CONFLLAMADO\_ID | NUMBER | No | Llave primaria de la tabla |

TABLA 7.1 Diccionario de Datos de la tabla GC\_CONF\_LLAMADO

| GC\_CONF\_TRAMITE  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 reasignación | |
| 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 7.2 Diccionario de Datos de la tabla GC\_CONF\_TRAMITE

| GC\_PREGUNTAS  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 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 7.3 Diccionario de Datos de la tabla GC\_PREGUNTAS

| GC\_PREGUNTAS\_RESPUESTAS  SEQ\_GC\_PREGUNTAS\_RESPUESTAS | | | |
| --- | --- | --- | --- |
| Nombre del atributo | Tipo de dato | Null | Descripción |
| N\_PREGUNTA\_RESPUESTA\_ID | NUMBER | No | Referencia a la pregunta que se ha respondido |
| N\_RESPUESTA\_ID | NUMBER | No | Referencia a la respuesta que ha sido contestada |
| FH\_RESPONDIO | DATE | Yes | Fecha y hora en que la pregunta fue respondida |
| C\_USUARIO | VARCHAR2(256 BYTE) | No | Código del usuario |

TABLA 7.4 Diccionario de Datos de la tabla GC\_PREGUNTAS\_RESPUESTAS

| GC\_TIQUETE  SECUENCIA SEQ\_GC\_TIQUETE | | | |
| --- | --- | --- | --- |
| Nombre del atributo | Tipo de dato | Null | Descripción |
| N\_TIQUETE\_ID | NUMBER | No | Llave primaria de la tabla |
| C\_UNIDAD\_RECEP | VARCHAR2(5 BYTE) | No | Código del centro de servicio |
| N\_RESERVA\_CITA\_ID | NUMBER | Yes | Referencia a la reserva de cita que dió origen al tiquete |
| N\_PRIORIDAD\_ID | NUMBER | No | Prioridad del tiquete |
| N\_TRAMITE\_ID | NUMBER | No | Referencia al tramite que esta atendiendo el tiquete |
| N\_TIQUETE\_REA | NUMBER | Yes | Tiquete de donde fue reasignado al tiquete actual |
| S\_CORRELATIVO | VARCHAR2(25 BYTE) | No | Número del tiquete |
| NIT | VARCHAR2(14 BYTE) | Yes | Nit del contribuyente |
| M\_ESTADO | VARCHAR2(1 BYTE) | No | Estado en que se encuentra el tiquete |
| FH\_LLEGADA | DATE | Yes | Fecha y hora en que el contribuyente llegó a la cola |
| FH\_LLAMADO | DATE | Yes | Fecha y hora en que el contribuyente fue llamado |
| FHI\_PROCESO | DATE | Yes | Fecha y hora en que la atención inició |
| FHF\_PROCESO | DATE | Yes | Fecha y hora en que la atención finalizó |
| C\_USUARIO\_ATENDIO | VARCHAR2(256 BYTE) | Yes | Usuario que atendió el trámite |
| C\_USUARIO\_CREA | VARCHAR2(256 BYTE) | Yes | Usuario que crea el registro |
| N\_TIEMPO\_HOLGURA | NUMBER | Yes | Tiempo de holgura del trámite seleccionado |

TABLA 7.5 Diccionario de Datos de la tabla GC\_TIQUETE

| GC\_TRAMITE  SECUENCIA SEQ\_GC\_TRAMITE | | | | |
| --- | --- | --- | --- | --- |
| Nombre del atributo | Tipo de dato | Null | Descripción | Nombre de secuencia |
| N\_TRAMITE\_ID | NUMBER | No | Llave primaria de la tabla | N\_TRAMITE\_ID |
| N\_SERVICIOS\_ID | NUMBER | No | Referencia al servicio a que pertenece el trámite | N\_SERVICIOS\_ID |
| N\_PESO | NUMBER | No | Cuantificación de la prioridad que debe tener el tramite luego de aplicarle todos los criterios | N\_PESO |
| B\_NIT\_REQUERIDO | NUMBER(1,0) | No | Bandera que indica si es obligatorio proporionar el nit | B\_NIT\_REQUERIDO |
| B\_ESCALAMIENTO | NUMBER(1,0) | No | Bandera que indica si el tramite será escalado al supervisor | B\_ESCALAMIENTO |
| D\_TRAMITE | VARCHAR2(256 BYTE) | Yes | Descripción del registro | D\_TRAMITE |
| C\_USUARIO\_CREA | VARCHAR2(100 BYTE) | No | Código del usuarioque crea el registro | C\_USUARIO\_CREA |
| C\_USUARIO\_MODI | VARCHAR2(100 BYTE) | No | Código del usuarioque modifica el registro | C\_USUARIO\_MODI |
| FI\_VIGENCIA | DATE | No | Fecha en que el registro es creado | FI\_VIGENCIA |
| FF\_VIGENCIA | DATE | Yes | Fecha en que el registro deja de tener vigencia | FF\_VIGENCIA |
| F\_MODIFICA | DATE | Yes | Fecha en que el registro fue modificado | F\_MODIFICA |
| B\_ACTIVA | NUMBER(1,0) | No | Bandera que indica si el registro esta activo o no | B\_ACTIVA |
| S\_NOMBRE | VARCHAR2(256 BYTE) | No | Nombre del tramite | S\_NOMBRE |
| N\_ORDEN | NUMBER | Yes | Orden en que se presentaran los tramites en pantalla | N\_ORDEN |

TABLA 7.6 Diccionario de Datos de la tabla GC\_TRAMITE

| GC\_RESPUESTAS  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 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 7.7 Diccionario de Datos de la tabla GC\_RESPUESTAS

| GC\_USER\_LOG  SECUENCIA SEQ\_GC\_USER\_LOG | | | |
| --- | --- | --- | --- |
| Nombre del atributo | Tipo de dato | Null | Descripción |
| N\_EVENTO\_ID | NUMBER | Yes | Referencia al evento que llevo a cabo el usuario |
| N\_USR\_LOG\_ID | NUMBER | No | Llave primaria de la tabla |
| FHI\_EVENTO | DATE | No | Fecha y hora en que se llevó a cabo el evento |
| S\_DETALLES | VARCHAR2(512 BYTE) | Yes | Detalles del evento del usuario |
| C\_USUARIO | VARCHAR2(256 BYTE) | No | Código del usuario |
| FHF\_EVENTO | DATE | Yes |  |

TABLA 7.8 Diccionario de Datos de la tabla GC\_USER\_LOG

| GC\_USUARIO | | | |
| --- | --- | --- | --- |
| 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 7.9 Diccionario de Datos de la tabla GC\_USUARIO

| GC\_CONF\_LLAMADO  SECUENCIA SEQ\_GC\_CONF\_LLAMADO | | | |
| --- | --- | --- | --- |
| 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 7.10 Diccionario de Datos de la tabla GC\_CONF\_LLAMADO

# Seguridad

Estos son los roles de seguridad usados en el Módulo de operación.

## Seguridad y Roles para ambiente de desarrollo y test

|  |  |  |
| --- | --- | --- |
| Nombre de la opción | Url | Roles |
| OPERACION | /operación/home | ROLE\_GC\_R4, ROLE\_GC\_R6 |

TABLA 10.1 Roles por opción.

# C:\Users\Fernando Melgar\Desktop\ER_OPERACION01_001.pngAnexos

## Anexo 1





