

Documentación para gestor documental Aurora

Christian Fingerhuth

30-04-2021

Abstract

Información sobre sistema gestor documental Aurora. Este documento no constituye información oficial.

Contents

1	Modelo de datos	3
1.1	Proceso	3
1.1.1	SGDP_PROCESOS	3
1.1.2	SGDP_TAREAS	3
1.1.3	SGDP_TAREAS_ROLES	4
1.1.4	Instancia	4
1.1.5	SGDP_INSTANCIAS_DE_PROCESOS	4
1.1.6	SGDP_INSTANCIAS_DE_TAREAS	5
1.1.7	SGDP_USUARIOS_ASIGNADOS_A_TAREAS	5
1.1.8	SGDP_SEGUIMIENTO_INTANCIA_PROCESOS	5
1.1.9	“SGDP_RESPONSABILIDAD”	5
1.1.10	“SGDP_REFERENCIAS_DE_TAREAS”	5
1.1.11	SGDP_DOCUMENTOS_DE_SALIDA_DE_TAREAS	5
1.1.12	SGDP_RESPONSABILIDAD_TAREA	5
1.1.13	SGDP_TIPOS_DE_DOCUMENTOS	5
1.1.14	SGDP_USUARIOS_ASIGNADOS_A_TAREAS	5
1.1.15	Relación inversa entre Instancia de Proceso e Instancia de Tarea	6
1.1.16	Estados	6
1.1.17	InstanciaProcesoMetadata	6
1.2	Definiciones	6
1.2.1	Responsabilidades vs Roles	6
1.2.2	Responsabilidades	7
1.2.3	Permisos	7
1.2.4	Funciones SQL	8
1.2.5	Columnas faltantes	8
1.2.6	Nombre parámetro	8
1.2.7	Migrations	8

1.2.8	Configuraciones	8
1.3	Propuestas cambios mayores	9
1.3.1	Campos Json	9
1.3.2	Lectura XML	9
1.3.3	Instancias	9
1.3.4	Resumen cambios mayores	10
1.4	Otros	10
1.4.1	Error de Tipo de Documento	10
1.4.2	Sistema php	10
1.4.3	Base dato: estándares	10
1.5	Manejo Orgánica	11
1.6	Datos para servidor de prueba de Archivo Nacional	11
1.7	Log	11
2	Instalación	12
2.1	Repositorios	12
2.2	Librerías	13
2.3	Crear bases de datos	14
2.3.1	Carga inicial	15
2.4	Instalación de Alfresco	15
2.4.1	Configurar Alfresco probando	16
2.5	Instalación LDAP	18
2.6	Instalación SGDP - Wildfly	19
2.6.1	Wildfly	19
2.6.2	Instalación SGDP	20
2.7	Integración Client Api	21
2.8	Compilar SGDP	21
3	Otros	21
3.1	Instalación carga subproceso (php) Camunda Version CF	22
3.2	Instalación carga subproceso (php) Camunda	22
4	Sistema local (personal de CFFM)	23
4.1	sgdp	23
5	Documentación	23

1 Modelo de datos

1.1 Proceso

Cada proceso se guarda en la base de datos SGDP_PROCESOS, y cada uno de las tareas (cuadros) se guardan en SGDP_TAREAS.

1.1.1 SGDP_PROCESOS

- nombre : Nombre del proceso. **Requerido**
- descripcion : Descripción
- macroproceso : ID del macroproceso **Requerido**
- vigente : *Revisar si conviene establecer default true*
- dias_habiles_max_duracion : **Requerido**
- unidad : ID de la unidad. *Revisar si tiene utilidad este campo*
- confidencial : *Revisar si tiene utilidad este campo*
- x_bpmn : *Desconocido*
- codigo_proceso : *Desconocido*
- fecha_creacion : Fecha de creación del proceso

1.1.2 SGDP_TAREAS

- nombre : Nombre de la Tarea **Requerido**
- descripcion : Descripción
- proceso : ID del proceso **Requerido**
- dias_habiles_max_duracion : Días hábiles de máxima duración **Requerido** *Revisar si tienen utilidad este campo*
- orden : **Requerido** *Revisar si tienen utilidad este campo*
- vigente : *Desconocido*
- solo_informar : *Desconocido: tiendo a pensar que indica que no tiene acciones posteriores*
- etapa : ID a etapa. Esta etapa se define en el BPMN
- obligatoria : *Desconocido el efecto*
- es_ultima_tarea : **Revisar uso. ¿Puede haber más de una última tarea?*
- tipo_de_bifurcacion : *Desconocido el efecto*
- puede_visar_documentos : *Desconocido el efecto*
- puede_aplicar_fea : *Desconocido el efecto*
- url_control : *Desconocido el efecto*
- id_diagrama : *Desconocido el efecto*
- asigna_num_doc : *Desconocido el efecto*
- esperar_resp : *Desconocido el efecto*
- conforma_expediente : *Desconocido el efecto*
- dias_reseteo : *Desconocido el efecto*
- tipo_reseteo : *Desconocido el efecto*
- url_ws : *Desconocido el efecto, sin embargo se filtra por este boolean en la Bandeja de Entradas*
- distribuye : *Desconocido el efecto*

- numeracion_auto : *Desconocido el efecto*

1.1.3 SGDP_TAREAS_ROLES

Esta tabla no se ocupa

1.1.4 Instancia

las tablas *Instancia* contienen la información de cada expediente, una con la información espejo del proceso (SGDP_INSTANCIAS_DE_PROCESOS) y otra con las tareas (SGDP_INSTANCIAS_DE_TAREAS).

Al momento de crear una Instancia, se generan todas las Instancias de Tarea, y en la medida que se avanza, van cambiando de estado. Esto (parece que) facilita el proceso de anulación y retroceder en el proceso.

1.1.5 SGDP_INSTANCIAS_DE_PROCESOS

- proceso : ID del proceso **Requerido**
- fecha_inicio : Fecha Inicio
- d_fecha_fin : Fecha Fin
- a_nombre_expediente : Nombre del expediente. Definir como se construye este campo, suponiendo que cada Institución tendrá otras definiciones.
- d_fecha_vencimiento_usuario : Fecha en que se acaba el tiempo. Es útil tener este campo calculado para evitar cálculos.
- estado_de_proceso : *Revisar como se fija cada uno de estos estados NUEVO, ASIGNADO, FINALIZADO, ANULADO*
- id_expediente : Definir como se construye este campo, suponiendo que cada Institución tendrá otras definiciones.
- id_instancia_de_proceso_padre : *Revisar utilidad de este campo*
- id_usuario_inicia : *En teoría este campo se puede obtener de SGDP_INSTANCIAS_DE_TAREAS. Que exista el campo reduce la cantidad de queries. A lo mejor pasar a JSON*
- id_usuario_termina : *En teoría este campo se puede obtener de SGDP_INSTANCIAS_DE_TAREAS. Que exista el campo reduce la cantidad de queries. A lo mejor pasar a JSON*
- b_tiene_documentos_en_cms : *Revisar utilidad de este campo*
- d_fecha_vencimiento : *¿Cuál es la diferecni con fecha_vencimiento_usuario*
- a_emisor : *Desconocido*
- a_asunto : *Desconocido*
- id_unidad : *Desconocido*
- id_acceso : *Desconocido*
- id_instancia_proceso_metadata : ID a MetaData de este proceso.
- id_tipo : ID a Tipo (Documento o Expediente)
- d_fecha_expiracion : *¿Cuál es la diferecni con fecha_vencimiento_usuario*

1.1.6 SGDP_INSTANCIAS_DE_TAREAS

- instancia_de_proceso : ID de la instancia del proceso **Requerido**
- tarea : ID de la tarea **Requerido**
- fecha_asignacion : *¿Diferencia entre fecha de asignación y fecha de inicio?*
- fecha_inicio : *¿Diferencia entre fecha de asignación y fecha de inicio?*
- fecha_finalizacion : Fecha en que se terminó/finalizó
- fecha_anulacion : Fecha en que se anuló.
- razon_anulacion : Texto con razón de anulación
- fecha_vencimiento : Fecha en que se vence el plazo según el tiempo definido en Tarea *Verificar*
- estado_de_tarea : NUEVO, ASIGNADO, FINALIZADO, ANULADO
- fecha_vencimiento_usuario : *Desconocido*
- usuario_que_asigna : *Verificar: supongo que es quien asignó, que sería lo mismo que quien finalizó anterior*

1.1.7 SGDP_USUARIOS_ASIGNADOS_A_TAREAS

- instancia_de_tarea : ID de la instancia de la tarea
- id_usuario: Usuario a quien tiene asignado.

1.1.8 SGDP_SEGUIMIENTO_INTANCIA_PROCESOS

1.1.9 “SGDP_RESPONSABILIDAD”

1.1.10 “SGDP_REFERENCIAS_DE_TAREAS”

dado que no hay paralelos, en teoría también podría ir en Tareas

1.1.11 SGDP_DOCUMENTOS_DE_SALIDA_DE_TAREAS

también debiese poder ir en Tareas

1.1.12 SGDP_RESPONSABILIDAD_TAREA

es el par [tarea, respobsabilidad (lane)], pero pq no va como atributo en Tarea?

1.1.13 SGDP_TIPOS_DE_DOCUMENTOS

Son los tipos de documentos que existen. No son únicos, es decir, pueden haber más de uno con el mismo nombre, ya que dependen de cada tarea. Se asocian a InstanciaTarea a través de “SGDP_ARCHIVOS_INST_DE_TAREA”

1.1.14 SGDP_USUARIOS_ASIGNADOS_A_TAREAS

Esta es una relación siempre OneToOne, por lo que yo creo que sería mas simple colocar una columna ID_USUARIO directamente en la tabla SGDP_INSTANCIAS_DE_TAREAS . Para facilitar parte de la transición se puede reemplazar la tabla con una vista “SELECT

ID_INSTANCIA_DE_TAREA, ID_USUARIO FROM SGDP_INSTANCIAS_DE_TAREAS".
Otra opción es mantener integridad a través del *save* de SGDP_USUARIOS_ASIGNADOS_A_TAREAS.

1.1.15 Relación inversa entre Instancia de Proceso e Instancia de Tarea

Podría ser útil (para reducir cantidad de queries) tener un campo *ID_INSTANCIA_TAREA* en la tabla *SGDP_INSTANCIAS_DE_PROCESOS* con el ID de la tarea activa.

1.1.16 Estados

SGDP_ESTADOS_DE_PROCESOS y SGDP_ESTADOS_DE_TAREAS.
Juntar ambas tablas en una sola, ya que contienen los mismos campos y valores.

1.1.17 InstanciaProcesoMetadata

Parece una buena opción, de tener cierta información del proceso mismo. Sin embargo, sugiero que esto a lo mejor se asocie a un JSON en la misma tabla de Instancia del Proceso.

1.2 Definiciones

1.2.1 Responsabilidades vs Roles

En algún momento se pensó la opción de igualar los Roles a las Responsabilidades. Esto no tiene mucho sentido, ya que son funciones distintas. Los Roles son las acciones que un usuario puede hacer, y acciones más de tipo administración del proceso (Crear, reasignar, etc), y loas Responsabilidades están asociadas a quien puede hacer cierta tarea.

Sin embargo, si debiese ser necesario redefinir algunos roles que generaron dificultades. Por ejemplo, un proceso lo debiese poder iniciar cualquier persona que tenga definida esa responsabilidad; y no depende der un rol. Eso fue un caso que “enredó” el entendimiento de ambos conceptos.

Los Roles tienen asociados permisos, siendo estos los siguientes:

- ADJUNTAR_DOC_EN_TODA_ETAPA
- AUTO_ASIGNA_PRIMERA_TAREA
- CREAR_EXPEDIENTE
- ENVIO_ARCHIVO_NACIONAL
- INGRESA_DATOS_ADICIONALES_AL_SUBIR_ARCHIVO
- INICIAR_TODOS_LOS_PROCESOS
- MODIFICA_ARCHIVOS
- NO_FILTRA_POR_CONFIDENCIALIDAD
- PUEDE_BUSCAR
- PUEDE_CERRAR_EXPEDIENTE
- PUEDE_DES_VINCULAR_EXPEDIENTES

- PUEDE_FIRMAR_CON_APPLET
- PUEDE_FIRMAR_CON_FEA
- PUEDE_MANTENER_AUTORES
- PUEDE_MANTENER_DATOS
- PUEDE_MANTENER_LISTA_DISTRIBUCION
- PUEDE_MANTENER_NOTIFICACIONES_PREDETERMINADAS
- PUEDE_MANTENER_PARAMETROS
- PUEDE_MANTENER_PROCESOS_SOL_CREAC_EXP
- PUEDE_REABRIR_EXPEDIENTE_Y_SATAR_TAREA
- PUEDE_VER_DASHBOARD
- PUEDE_VER_INDICADORES
- PUEDE_VER_MANTENEDORES
- PUEDE_VER_TAREAS_EN_EJECUCION
- PUEDE_VINCULAR_EXPEDIENTES
- PUEDE_VISAR_DOCUMENTO
- REASIGNA_TAREA
- SUBIR_CARTA

De los cuales hay que diferenciar dos tipos, los asociados a tareas administrativas generales (PUEDE_VER_DASHBOARD), y aquellos asociados al flujo del procesos (CREAR_EXPEDIENTE). No es una separación dicotómica, ya que REASIGNA_TAREA podría estar en ambos grupos. Creo que es importante ordenar este concepto, especialmente si se quiere considerar la orgánica institucional, en donde los permisos también tengan que diferenciarse por zona.

1.2.2 Responsabilidades

Se ha conversado la opción que no se tenga que, para cada proceso, definir los usuarios asociados a cada responsabilidad (swimmlane). La alternativa sería para cada usuario definirle las Responsabilidades que tiene por defecto. Luego habrían dos posibilidades:

- En el SGDP, al momento de seguir con el siguiente paso, además de los usuarios definidos manualmente en el subsistema “carga de subproceso”, también busque todas las personas asociadas a las responsabilidad correspondiente .
- En el subsistema “carga de subproceso” exista un botón de “precargar” y ahí se carguen con la información de usuario-responsabilidad. Este último es un cambio más simple, ya que no hay que intervenir el SGDP, pero tendría que hacerse periódicamente la acción de “precargar”.

1.2.3 Permisos

Es común (y recomendado para sistemas grandes) que exista el concepto de Rol para agrupar distintos permisos a un usuario, tal como existe en el SGDP. Sin embargo, también debiese existir la opción, de asignar permisos individuales a cada usuario.

1.2.4 Funciones SQL

En el script de carga inicial de sql se observa que hay varias funciones sql. Hasta donde he podido ver, estas no se ocupan; pero es necesario confirmar eso. Si es que se están ocupando, creo que es importante darlas de baja lo antes posible, ya que toda la lógica del sistema debiese estar en el código y no en la base de datos.

1.2.5 Columnas faltantes

- B_TIENE_PARAM_POR_TAREA: En los script de instalación falta una columna B_TIENE_PARAM_POR_TAREA en la tabla SGBP_PROCESOS `ALTER TABLE sgdg."SGDP_PROCESOS" ADD "B_TIENE_PARAM_POR_TAREA" bool NULL DEFAULT false;`
- B_ES_SNC : `ALTER TABLE sgdg."SGDP_PARAMETRO_DE_TAREA" ADD "B_ES_SNC" bool NULL DEFAULT true;`
- B_VIGENTE : `ALTER TABLE sgdg."SGDP_PARAMETRO_DE_TAREA" ADD "B_VIGENTE" bool NULL DEFAULT true;`
- A_VALOR_PARAM_NO_SETEADO: `ALTER TABLE sgdg."SGDP_TIPO_PARAMETRO_DE_TAREA" ADD "A_VALOR_PARAM_NO_SETEADO" varchar(1000) NULL ;`
- A_VALOR_PARAM_SETEADO: `ALTER TABLE sgdg."SGDP_TIPO_PARAMETRO_DE_TAREA" ADD "A_VALOR_PARAM_SETEADO" varchar(1000) NULL ;`

1.2.6 Nombre parámetro

Esto ya fue respondido por SCJ: cambiar esta documentación.

En `CrearExpedienteCMSServiceImpl` de SGBP se define el nombre del parámetro como *nombExp*, sin embargo el *crearExpediente.post.js* de Alfresco utiliza el nombre *nombreExp*. Debe ser un arrestre de versiones distintas. ¿Cuál utilizar? ¿Habrá que cambiar en otras partes?

1.2.7 Migrations

Continuando con el campo anterior, sería bueno contar con algún sistema de registro de cambios en los modelos de datos. Ideal sería asociado a algún framework. Sin embargo, también servirían archivos ordenados por fecha, indicando el cambio hecho y su código sql correspondiente.

1.2.8 Configuraciones

Las configuraciones del sistema debiesen concentrarse en un archivo. No sé como se hace en Spring, pero supongo que esta información debiese dar una idea <https://docs.spring.io/spring-boot/docs/1.5.6.RELEASE/reference/html/boot-features-external-config.html>.

Esto también permitiría dejar de utilizar los parámetros de configuración que están en una tabla de la base de datos.

1.3 Propuestas cambios mayores

1.3.1 Campos Json

Creo que sería muy útil generar para las tablas principales campos Json para guardar información. Con esto no me refiero pasar a un modelo de base de datos NoSql, sino que simplemente para facilitar guardar dos tipos de campos:

- Parámetros no esenciales y que están en forma no estructurada, por ejemplo los parámetros adicionales de cada proceso
- Poder ocupar este json para etapas de desarrollo. Es más simple guardar información, sin tener que necesariamente, definir como se guardará la información. Esto al menos, durante la etapa de desarrollo de nuevas funcionalidades, y tener que evitar crear tantas nuevas columnas, que posiblemente no terminarán ocupándose.

1.3.2 Lectura XML

Creo que el sistema no debiese capturar el XML y pasarlo a información relacional en las distintas tablas SGDP_TAREAS, SGDP_PROCESOS, SGDP_TIPOS_DE_DOCUMENTOS, SGDP_PARAMETROS, etc. Toda la información ya está contenida en el XML. Es cierto que la lectura de XML es un poco engorrosa, pero es posible, y al final son un par de líneas de código, para cada tarea determinar cuales son sus siguientes posibles tareas. En cierto sentido esto ya se hace al momento de cargar el subproceso.

Hice una Clase en python que lee el XML y trae toda la información que uno necesite: primera tarea, tarea siguiente y anterior según tarea actual, etc.

Mi opinión es ir siempre al XML para obtener información del proceso: el XML puede estar como archivo en el sistema, o como un texto en una BD. Esto traería, al menos, dos beneficios:

- Varias de las tablas ya no serían necesarias
- Mayor flexibilidad para ir agregando cosas al BPMN y no tener que cambiar la BD para capturar esas definiciones adicionales: cancelar, procesos paralelos, etc.

A lo mejor un primer desarrollo, sería pasar la Clase Python que hice a una API que reciba un XML (y pueda quedar cacheado) y entregue información requerida (en json)

1.3.3 Instancias

Asociado a lo anterior, tampoco veo el beneficio de crear los SGDP_INSTANCIAS_DE_TAREAS al crear el expediente. Creo mejor ir creando un registro *RegistroTarea* en la medida que se va avanzando (y si se necesita algún dato del proceso completo, siempre se puede consultar el XML). Extendiendo esto, si es que se retrocede, se crea un *RegistroTarea* nuevo. De esa forma, también se lleva en ese *RegistroTarea* el log de todo el proceso.

1.3.4 Resumen cambios mayores

Esto implicaría un cambio sustancialmente mayor. Pero asociado a también ocupar campos JSON, en vez de columnas para cada dato que se quiera guardar, daría mucho mayor flexibilidad. Llevándolo a un extremo, lo único que se necesitarían son

- Tabla *Proceso* con nombre del proceso y el XML (y seguramente algún tema de versionamiento)
- Tabla *Expediente* con el detalle del expediente (y relacionada a *Proceso*)
- Tabla *Tarea* con la información de cada una de las tareas activas e históricas (relacionada a *Expediente*)
- Tabla *Usuario* con todo lo del usuario (y un campo json con sus responsabilidades, zonas, roles)

Esto es *llevándolo a un extremo*, ya que obviamente se necesitan muchas tablas auxiliares.

1.4 Otros

1.4.1 Error de Tipo de Documento

ACTUALIZAR: fue respondido por SCJ

En el archivo TipoDeDocumentoServiceImpl está la línea

```
... = tipoDeDocumentoDao.getTipoDeDocumentoPorIdTipoDeDocumento(idTipoDeDocumento);
```

y por lo tanto, según entiendo debiese haber en el archivo TipoDeDocumento.java una NamedQuery con ese mismo nombre *getTipoDeDocumentoPorIdTipoDeDocumento*. Pero no está.

1.4.2 Sistema php

Creo que es el sistema más débil de todo el proyecto (no ocupa capa de “datos” tipo DAO o ORM).

1.4.3 Base dato: estándares

No tienen ninguna relevancia para el funcionamiento propiamente tal, pero desde el punto de vista de simplificar al momento de desarrollar, mi sugerencia es:

- No utilizar “schemas” en las base de datos, sino que utilizar el por defecto “public”. Esto simplifica la cantidad de texto al momento de escribir queries.
- Utilizar nombres de tablas y columnas en minúscula, que es el por defecto en postgres; y simplifica la escritura de queries.
- Nombres de columnas sin necesidad de poner el tipo de columna (“B_”, “A_”) y sin apellidos: id_param_tarea -> id y a_nombre_param_tarea -> nombre

Estos puntos anteriores son convenciones que se toman, y son totalmente debatibles. Hacer el cambio implicaría un esfuerzo grande, especialmente para la SCJ en modificar su sistema actual. Al menos, los frameworks actuales, a través de ORM, DAO, permiten abstraerse de este problema, permitiendo llamar la columna en el formato “simple”. haciendo el ORM la conversión del nombre al momento de crear la query.

1.5 Manejo Orgánica

Este texto fue escrito en febrero 2021. Desde esa fecha este concepto se ha estado desarrollando en otro documento.

Aduana: Nacional / Regional / Comunal

- Personas se asignan a mas de una zona; y la dependencia
- Es necesario registrar la “dependencia” de zonas, pero solo con fines “mostrar información”
- Al crear un expediente se debe revisar que las zona del suauiro correspondoa una de las zonas del proceso
- Y se debe asignar una zona
- En el proceso mismo, se muestran los usuario de la zona en la cual se encuatra el expeidnet en ese omento, y se puede modificar
- Cada “caja” se debe definir si en ese paso se puede cambiar, y otras “reglas” aue vayan apareciendo
- Procesos internos de revisión , pero el proceso se mantiene denro de la caja.
- Cada proceso tiene asignado una zona, la cual puede ir cambiando. Según esa zona se muestran “primero” los usuarios de esa zona
- Hay un administrador por zona?
- Quién puede ver esos procesos?

1.6 Datos para servidor de prueba de Archivo Nacional

dante / Dante-2021 daniel / Daniel-2021 user4 / User4-2021

http://sgdoc.tecnova.cl/sgdoc/proceso/bpm/asig_user.php

1.7 Log

Es muy útil entender los distintos Log de sistema.

- Alfresco: tienen un archivo de log dentro de la carpeta de alfresco llamado *alfresco.log*, el cual es muy útil para debuggear

2 Instalación

Las siguientes instrucciones corresponden a una instalación en servidor Ubuntu 20.04 (en servidores EC2 de AWS). Está pensada en lograr tener un script de instalación automático y con el menor input posible por parte del usuario de instalación: esto era muy útil en los primeros pasos de aprendizaje, para poder fácilmente instalar distintos servidores y hacer pruebas.

Este proceso de instalación, sin duda, puede mejorarse en las siguientes iteraciones:

- Se utiliza wildfly (SGDP) y tomcat (Alfresco). Hasta donde tengo entendido se podría utilizar tomcat también para ejecutar el SGDP.
- Para todos los subsistemas se utiliza la misma clave 'gest1469'
- Esta instalación no es segura
- Instalación en servidores independientes para cada sistema (Alfresco, SGDO, Base de datos)

En grandes rasgos, la instalación corresponde a:

- Descargar los repositorios GIT
- Instalar librerías de uso general
- Crear bases de datos
- Instalación de Alfresco con Tomcat
- Instalación de LDAP
- Instalación de Wildfly y SGDP
- Instalación de sistema carga de procesos (php)

Para esta instalación se requieren los puertos abiertos 80, 5432, 8080, 8180, 22, 8983, 8090, 10000

Se requiere una máquina con al menos 4GB (medium)

2.1 Repositorios

Se instala y utiliza CLI de github para descargar los distintos repositorios que se instalarán.

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-key C99B11DEB97541F0
sudo apt-add-repository https://cli.github.com/packages
sudo apt update
sudo apt install gh
gh auth login
```

Seguir los pasos para validar la cuenta (es necesario abrir una url) (c@a.cl, SHub5).

A continuación crear una carpeta para guardar los repositorios, y en algunos casos cambiar de rama. No todos estos repositorios se están ocupando en esta

instalación. Están todos los repositorios de la SCJ y un repositorio personal con archivos de ayuda para la instalación.

```
mkdir ~/repositorios
cd ~/repositorios

gh repo clone cfingerh/aurora
cd aurora
git checkout instalacion
cd ..

cd ~/repositorios

gh repo clone SuperintendenciaDeCasinosCL/custom-authentication-repo
gh repo clone SuperintendenciaDeCasinosCL/FirmaAvanzada
gh repo clone SuperintendenciaDeCasinosCL/integracion-client-api
gh repo clone SuperintendenciaDeCasinosCL/num-doc-tipo-ws-rest
gh repo clone SuperintendenciaDeCasinosCL/numeracion-documentos-ws
gh repo clone SuperintendenciaDeCasinosCL/pdf-converter-net
gh repo clone SuperintendenciaDeCasinosCL/SGDP
gh repo clone SuperintendenciaDeCasinosCL/sgdp-carga-subProcesos

gh repo clone cfingerh/aurora-cargaphp

gh repo clone SuperintendenciaDeCasinosCL/SGDP-DOCUMENTACION
gh repo clone SuperintendenciaDeCasinosCL/sgdp-mantenedor-autores
gh repo clone SuperintendenciaDeCasinosCL/Web-Scripts-Alfresco-SGDP
```

2.2 Librerías

Existen una serie de librerías que se utilizan durante el proceso de instalación; y algunas que no son requeridas, pero son de uso general para facilitar el trabajo en el servidor. Durante la instalación de esas librerías, es posible que sea necesario confirmar unos términos de uso de una de las librerías (ttf-mscorefonts-installer).

```
sudo apt update -y
sudo apt upgrade -y
sudo apt install postgresql postgresql-contrib -y
sudo apt-get install openjdk-8-jdk -y
sudo apt install tmux -y
sudo apt install vim -y
sudo apt install git -y
sudo apt install unzip -y
sudo apt install mlocate -y
sudo apt install maven -y
sudo apt install wget -y
sudo apt install ttf-mscorefonts-installer fonts-noto fontconfig libcups2 libfontconfig1 lib
```

```
sudo apt install libreoffice -y
sudo updatedb
```

El siguiente comando captura la IP del servidor y la guarda en la variable de entorno, y se utilizará en otras partes de la instalación. REVISAR. ALGUNAS VECES NO FUNCIONA LA PAGINA ifconfig.me

```
URL=$(curl ifconfig.me)
```

2.3 Crear bases de datos

Ambos sistemas Alfresco y SGDP utilizan postgres como su motor de datos. Tengo entendido que Alfresco también podría funcionar sobre otros motores, pero para esta instalación (y en general también es la recomendación), se utilizará postgres.

Postgres se instaló en la sección anterior, y por defecto debe haberse instalado la versión 12 (psql -V psql). Sin embargo, otras versiones “cercanas” debiesen funcionar correctamente.

Se crean las bases de datos y usuarios para acceder a estas

```
sudo -u postgres psql --command "CREATE USER alfresco with password 'alfresco';"
sudo -u postgres psql --command "CREATE USER sgdp with password 'gest1469';"
sudo -u postgres createdb alfresco
sudo -u postgres createdb sgdp
sudo -u postgres psql -d sgdp --command "CREATE SCHEMA sgdp AUTHORIZATION sgdp;"
sudo -u postgres psql --command "
ALTER ROLE alfresco SUPERUSER NOCREATEDB CREATEROLE INHERIT LOGIN;
ALTER ROLE sgdp SUPERUSER NOCREATEDB CREATEROLE INHERIT LOGIN;"
```

El siguiente paso es opcional, y permite configurar postgres para acceso remoto, pero debiese revisarse para sistema en producción.

```
sudo sed -i 's/ident/md5/g' /etc/postgresql/12/main/pg_hba.conf
sudo sed -i 's/peer/md5/g' /etc/postgresql/12/main/pg_hba.conf
sudo sed -i "s/#listen_addresses = 'localhost'/listen_addresses = '*' /g" /etc/postgresql/12/main/pg_hba.conf
sudo sed -i 's/127.0.0.1/32/0.0.0.0/0/g' /etc/postgresql/12/main/pg_hba.conf
sudo service postgresql restart
sudo service postgresql reload
```

A continuación, para evitar poner siempre la clave de postgres, se crea el archivo pgpass.

```
cd ~
echo "
localhost:5432:alfresco:alfresco:alfresco
localhost:5432:sgdp:sgdp:gest1469
">>.pgpass
chmod 600 .pgpass
```

2.3.1 Carga inicial

Existe un SQL con carga la estructura inicial de las base de datos, y con datos básicos. Sin embargo, es necesario modificar en esos datos iniciales el valor de la IP de Alfresco. El script siguiente modifica el sql con la IP del sistema (considerando que en este caso, Alfresco está en el mismo servidor) y carga los datos

```
cd ~
sed -i "s/192.168.1.92:8080/$URL:8080/g" ~/repositorios/aurora/Instalacion/sgdp_datos_inicial.sql
psql -U sgdp -h localhost -d sgdp -f ~/repositorios/aurora/Instalacion/sgdp_datos_inicial.sql
```

2.4 Instalación de Alfresco

Para la instalación de Alfresco existe un proceso guiado para definir la configuración de instalación. En este caso, creamos un archivo de configuración (install_opts) que se utiliza de input al instalar y así evitar el proceso guiado.

Se descarga e instala la versión 5.2. Es necesario revisar que esta versión sea segura (revisar correo de José Riffo del 20 de marzo 2021)

```
rm install_opts
touch install_opts
echo "
mode=unattended
enable-components=javaalfresco,alfrescosolr4,alfrescogoogledocs, libreofficecomponent
disable-components=postgres

jdbc_url=jdbc:postgresql://localhost/alfresco
jdbc_driver=org.postgresql.Driver
jdbc_database=alfresco
jdbc_username=alfresco
jdbc_password=alfresco

prefix=/home/ubuntu/alfresco/

alfresco_admin_password=gest1469
" >> install_opts

wget https://download.alfresco.com/release/community/201707-build-00028/alfresco-community-installer-201707-linux-x64.bin
chmod a+x alfresco-community-installer-201707-linux-x64.bin
./alfresco-community-installer-201707-linux-x64.bin --optionfile install_opts
rm alfresco-community-installer-201707-linux-x64.bin
rm install_opts
HOME_ALFRESCO=/home/ubuntu/alfresco
$HOME_ALFRESCO/alfresco.sh start
echo "http://$URL:8080/share/ "
```

Debiese funcionar la página `http://$URL:8080/share/` y poder ingresar con usuario 'admin' y clave 'gest1469' (puede demorar varios segundos)

2.4.1 Configurar Alfresco probando

```

"bash echo " authentication.chain=alfinst:alfrescoNtlm,custauth1:customauthenticator,ad1:ldap-
ad ntlm.authentication.sso.enabled=false " >> /home/ubuntu/alfresco/tomcat/shared/classes/alfresco-
global.properties

mkdir /home/ubuntu/alfresco/tomcat/shared/classes/alfresco/extension/subsystems/
mkdir /home/ubuntu/alfresco/tomcat/shared/classes/alfresco/extension/subsystems/Authentication/
mkdir /home/ubuntu/alfresco/tomcat/shared/classes/alfresco/extension/subsystems/Authentication/ldap-
ad/ mkdir /home/ubuntu/alfresco/tomcat/shared/classes/alfresco/extension/subsystems/Authentication/ldap-
ad/ad1/

echo " ldap.authentication.allowGuestLogin=false ldap.authentication.userNameFormat=%s@localhost
ldap.authentication.java.naming.provider.url=ldap://localhost:389 ldap.synchronization.java.naming.security.pri
ldap.synchronization.java.naming.security.credentials=gest1469 ldap.synchronization.groupSearchBase=dc=app,
ldap.synchronization.userSearchBase=dc=app,dc=local" >> /home/ubuntu/alfresco/tomcat/shared/classes/alfre
ad/ad1/ldap-ad-authentication.properties

cp /home/ubuntu/repositorios/custom-authentication-repo/custom-authentication-
repo-1.0-SNAPSHOT.jar $HOME_alfresco/tomcat/webapps/alfresco/WEB-
INF/lib $HOME_alfresco/alfresco.sh stop $HOME_alfresco/alfresco.sh
start

$HOME_alfresco/alfresco.sh stop rm -f $HOME_alfresco/alf_data/solr4/index/workspace/SpacesStore/
rm -f $HOME_alfresco/alf_data/solr4/index/archive/SpacesStore/index/
rm -f $HOME_alfresco/alf_data/solr4/model/ rm -f $HOME_alfresco/alf_data/solr4/content/

cp /home/ubuntu/repositorios/SGDP-DOCUMENTACION/all-in-one-repo-
amp-1.0-SNAPSHOT.amp $HOME_alfresco/amps cp /home/ubuntu/repositorios/SGDP-
DOCUMENTACION/all-in-one-share-amp-1.0-SNAPSHOT.amp $HOME_alfresco/amps
$HOME_alfresco/alfresco.sh start

Aplica los AMP (y se generan los jar para los amp)

cd $HOME_alfresco/bin
yes y | ./apply_amps.sh -force -nobackup
$HOME_alfresco/alfresco.sh stop
$HOME_alfresco/alfresco.sh start

REVISAR: acá parece que deja de funcionar el login normal

cp ~/repositorios/Web-Scripts-Alfresco-SGDP/*.* $HOME_alfresco/tomcat/webapps/alfresco/WEB-INF/lib
$HOME_alfresco/alfresco.sh stop
$HOME_alfresco/alfresco.sh start

rm -f $HOME_alfresco/alf_data/solr4/index/workspace/SpacesStore/index/*

```



```

rm -f $HOME_ALFRESCO/alf_data/solr4/index/archive/SpacesStore/index/*
$HOME_ALFRESCO/alfresco.sh start
echo http://$URL:8080/solr4/admin/cores?action=FIX

$HOME_ALFRESCO/alfresco.sh stop
cd $HOME_ALFRESCO/tomcat/webapps/alfresco/WEB-INF/classes/alfresco/extension/templates/webs
rm -f subirCartas.post.*
cd ~

cp /home/ubuntu/repositorios/SGDP-DOCUMENTACION/sgdp-carpetas.xml /home/ubuntu/alfresco/tomcat/webapps/alfresco/WEB-INF/classes/alfresco/extension/templates/webs
cp /home/ubuntu/repositorios/SGDP-DOCUMENTACION/sgdp-documentos.xml /home/ubuntu/alfresco/tomcat/webapps/alfresco/WEB-INF/classes/alfresco/extension/templates/webs

$HOME_ALFRESCO/alfresco.sh start

sed -i '44d' $HOME_ALFRESCO/tomcat/shared/classes/alfresco/web-extension/share-config-custom.xml
sed -i '40d' $HOME_ALFRESCO/tomcat/shared/classes/alfresco/web-extension/share-config-custom.xml

cp /home/ubuntu/repositorios/custom-authentication-repo/custom-authentication-repo-1.0-SNAPSHOT.jar $HOME_ALFRESCO/tomcat/webapps/alfresco/WEB-INF/lib
cp ~/repositorios/Web-Scripts-Alfresco-SGDP/*.* $HOME_ALFRESCO/tomcat/webapps/alfresco/WEB-INF/classes/alfresco/extension/templates/webs
$HOME_ALFRESCO/alfresco.sh stop
$HOME_ALFRESCO/alfresco.sh start

# parece que nada de esto es necesario
# http://54.207.252.22:8080/share/
# echo http://$URL:8080/alfresco/s/index
# (pero primero hay que hacer login) hacer un script con python?
# http://$URL:8080/alfresco/s/index con post
# {reset: on, submit: Refresh Web Scripts}

echo "Pausa para cargar alfresco"
sleep 2m

export URL
cd ~
python3 repositorios/aurora/Instalacion/crear_carpetas.py

ACA FALTA CAMBIAR LOS PERMISSION (TAMBIEN EL INHERIT PERMISSION)

Debiese funcionar http://54.94.29.51:8080/alfresco/service/api/login.json?u=prueba&pw=a1c54595d
aunque no entiendo pq, ya que el usuari prueba todavía no existe
y http://54.94.29.51:8080/share/page/user/admin/dashboard

```

2.5 Instalación LDAP

En esta etapa se instalará LDAP en una versión local.

```
export DEBIAN_FRONTEND=noninteractive
sudo apt-get install -y slapd ldap-utils
```

Solicitará ingresar un password (gest1469)

Algunas veces he tenido problema con la instalación de arriba. REVISAR. Si es necesario borrar todo

```
sudo service slapd stop
sudo apt-get -y remove --purge slapd
rm -rf /var/lib/ldap
sudo rm -rf /etc/ldap/
sudo apt-get -y remove --purge ldap-utils
```

Luego se debe configurar

```
sudo dpkg-reconfigure slapd
```

con la opciones:

- no
- app.local
- app.local
- gest1469
- gest1469
- no
- yes

Se creó una un archivo de carga inicial de objetos. REVISAR, ya que creo que no son necesarios varios de esos. y da error

```
cd ~/repositorios/aurora/Instalacion
ldapadd -x -D cn=admin,dc=app,dc=local -f ldap_data.ldif -w gest1469
```

Instalar un gestor de ldap en php.

```
sudo apt-get install ldap-account-manager -y
echo http://$URL/lam/templates/login.php
```

Ingresar a [http://\\$URL/lam/templates/login.php](http://$URL/lam/templates/login.php)

- Lam Configuration -> Edit Server Profiles -> Ingresar con lam / lam
- En la parte abajo modificar password a gest1469
- En TreeSufix modificar -> dc=app,dc=local
- SecuritySettings -> list of valid users -> cn=admin,dc=app,dc=local
- Account Type -> LDAP suffix -> dc=app,dc=local (en ambos)
- Save
- Entrar con admin / gest1469
- Primer Crear un Group con name:

- Crear un usuario para probar: last_name “prueba” y oassword gest1469

Y el siguiente comando debiese devolver “success”:

```
ldapwhoami -vvv -h localhost -p 389 -D 'cn=prueba,dc=app,dc=local' -x -w gest1469
ldapwhoami -vvv -h localhost -p 389 -D 'cn=admin,dc=app,dc=local' -x -w gest1469
```

2.6 Instalación SGDP - Wildfly

SGDP corre sobre wildfly (no me queda claro si también podría correr sobre tomcat)

2.6.1 Wildfly

Se descargar wildfly, se instala y se deja como servicio. Temporalmente también detenemos tomcat/alfresco

```
$HOME_ALFRESCO/alfresco.sh stop
cd ~
sudo groupadd -r wildfly
sudo useradd -r -g wildfly -d /opt/wildfly -s /sbin/nologin wildfly
wget https://download.jboss.org/wildfly/19.1.0.Final/wildfly-19.1.0.Final.zip
unzip wildfly-19.1.0.Final.zip
rm wildfly-19.1.0.Final.zip
sudo mv wildfly-19.1.0.Final /opt/
cd /opt
sudo mv wildfly-19.1.0.Final wildfly
WILDFLY=/opt/wildfly
sudo chown -RH wildfly: /opt/wildfly

sudo mkdir -p /etc/wildfly
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh /opt/wildfly/bin/
sudo sh -c 'chmod +x /opt/wildfly/bin/*.sh'
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/
sudo chown -RH wildfly: /opt/wildfly
sudo systemctl daemon-reload
sudo systemctl stop wildfly
sudo systemctl start wildfly
sudo systemctl enable wildfly
```

Se crea un usuario administrador para wildfly. Además por defecto wildfly corre en el puerto 8080, pero ya que en ese está tomcat, es necesario modificar eso. Después de ese cambio, se puede volver a iniciar tomcat/alfresco.

```
cd $WILDFLY/bin
sudo sh ./add-user.sh -u administrador -p gest1469
```

```
sudo sed -i 's/port-offset:0/port-offset:1/g' /opt/wildfly/standalone/configuration/standalone.xml
```

```
sudo systemctl stop wildfly
sudo systemctl start wildfly
sudo systemctl enable wildfly
echo http://$URL:8081
```

```
$HOME_ALFRESCO/alfresco.sh start
```

Otras configuraciones

```
sudo sed -i '$a WILDFLY_CONSOLE_BIND=0.0.0.0' /etc/wildfly/wildfly.conf
sudo sed -i 's/-b $3/-b $3 -bmanagement $4 /g' /opt/wildfly/bin/launch.sh
sudo sed -i 's/WILDFLY_BIND/WILDFLY_BIND $WILDFLY_CONSOLE_BIND /g' /etc/systemd/system/wildfly.service
sudo mkdir /var/run/wildfly/
sudo chown wildfly: /var/run/wildfly/
sudo systemctl daemon-reload
sudo systemctl restart wildfly
```

2.6.2 Instalación SGDP

Se crea también un usuario inicial en el sistema SGDP y cambiis en la base

```
psql -d sgdp -U sgdp -h localhost
```

```
INSERT INTO "SGDP_USUARIOS_ROLES" ("ID_ROL", "ID_USUARIO", "ID_UNIDAD") VALUES (2,'prueba',3)
update "SGDP_USUARIOS_ROLES" set "B_ACTIVO" = true, "A_NOMBRE_COMPLETO" ='Andres Prueba Pr
```

```
alter table sgdp."SGDP_HISTORICO_DE_INST_DE_TAREAS"
add column "N_DIAS_OCUPADOS" int2 ;
```

```
alter table sgdp."SGDP_HISTORICO_DE_INST_DE_TAREAS"
add column "N_HORAS_OCUPADAS" int2 ;
```

```
alter table sgdp."SGDP_HISTORICO_DE_INST_DE_TAREAS"
add column "N_MINUTOS_OCUPADOS" int2 ;
```

```
ALTER TABLE sgdp."SGDP_PROCESOS" ADD "B_TIENE_PARAM_POR_TAREA" bool NULL DEFAULT false;
```

Modificar el archivo

```
vim /home/ubuntu/repositorios/SGDP/src/cl/gob/scj/sgdp/config/security-context.xml
```

y modificar la clase *org.springframework.security ldap.DefaultSpringSecurityContextSource*

```
<constructor-arg value="ldap://localhost/dc=app,dc=local"></constructor-arg>
<property name="userDn" value="cn=admin,dc=app,dc=local"></property>
<property name="password" value="gest1469"></property>
```

Wildfly debiese estar disponible en [http://\\$URL:9991/console/index.html](http://$URL:9991/console/index.html) .
(usuario administrador clave gest 1469)

- Se debe cargar el JDBC de postgres.

- Se debe crear un DataSource en (Configuration -> Subsystems -> Data-Source & Drivers -> Datasources)

Se puede hacer por comando:

```
cd ~
wget https://jdbc.postgresql.org/download/postgresql-42.2.18.jar
sudo bash /opt/wildfly/bin/jboss-cli.sh --connect --controller=127.0.0.1:9991 --command="dep
sudo bash /opt/wildfly/bin/jboss-cli.sh --connect --controller=127.0.0.1:9991 --command="dat
rm postgresql-42.2.18.jar

sudo systemctl daemon-reload
sudo systemctl restart wildfly
```

2.7 Integración Client Api

La instalación de SGDP requiere otros previos. Esta compilación ha fallado porque no estaba actualizado.

```
cd /home/ubuntu/repositorios/integracion-client-api
mvn install -Dmaven.test.skip=true
```

La alternativa es ocupar el JAR compilado

```
mkdir ~/.m2/
mkdir ~/.m2/repository
mkdir ~/.m2/repository/cl/
mkdir ~/.m2/repository/cl/gob/
mkdir ~/.m2/repository/cl/gob/scj/
mkdir ~/.m2/repository/cl/gob/scj/integracion-client-api/
mkdir ~/.m2/repository/cl/gob/scj/integracion-client-api/0.0.2/
cp /home/ubuntu/repositorios/integracion-client-api/integracion-client-api-0.0.2.jar ~/.m2/1
```

2.8 Compilar SGDP

Compilar SGDP y cargar a wildfly

```
cd /home/ubuntu/repositorios/SGDP
mvn clean install -Dmaven.test.skip=true
sudo bash /opt/wildfly/bin/jboss-cli.sh --connect --controller=127.0.0.1:9991 --command="dep
sudo systemctl restart wildfly
echo http://$URL:8081/sgdp
```

3 Otros

- revisar nuevas columnas en BD
- permisos Alfresco de carpetas

3.1 Instalación carga subprocesso (php) Camunda Version CF

```
cd ~
sudo apt install php-pgsql -y
sudo apt-get install php libapache2-mod-php -y

sudo cp -R ~/repositorios/aurora-cargaphp/sgdoc /var/www/html/
o
sudo ln -s /home/ubuntu/repositorios/aurora-cargaphp/sgdoc /var/www/html/sgdoc2
```

```
sudo systemctl restart apache2
echo http://$URL/sgdoc/proceso/bpm/asig_user.php
```

Es necesario cambiar la configuración en

```
cd ~/repositorios/aurora-cargaphp/sgdoc/
cp configuracion_example.php configuracion.php
vim configuracion.php
```

```
sudo systemctl restart apache2
```

Este módulo ocupa una función `ereg_replace()` que no está en las versiones más nuevas de php (>7).

3.2 Instalación carga subprocesso (php) Camunda

```
cd ~
sudo apt install php-pgsql -y
sudo apt-get install php libapache2-mod-php -y

sudo cp -R repositorios/sgdp-carga-subProcesos/sgdoc /var/www/html/
```

```
sudo systemctl restart apache2
echo http://$URL/sgdoc/proceso/bpm/asig_user.php
```

Es necesario cambiar la configuración en varios archivos (REVISAR: FALTAN OTROS)

```
sudo sed -i 's/172.16.10.61/localhost/g' /var/www/html/sgdoc/proceso/bpm/connect.php
sudo sed -i 's/S4cc84cJ/alfresco/g' /var/www/html/sgdoc/proceso/bpm/connect.php
sudo sed -i 's/Error al /Error 1 al/g' /var/www/html/sgdoc/proceso/bpm/connect.php
```

```
sudo sed -i 's/172.16.10.61/localhost/g' /var/www/html/sgdoc/proceso/bpm/logica/connect.php
sudo sed -i 's/S4cc84cJ/alfresco/g' /var/www/html/sgdoc/proceso/bpm/logica/connect.php
```

```
sudo sed -i 's/Error al /Error 2 al/g' /var/www/html/sgdoc/proceso/bpm/logica/connect.php

sudo sed -i 's/scjedb.supercasino.cl/localhost/g' /var/www/html/sgdoc/logica/connect.php
sudo sed -i 's/postgresSCJ/gest1469/g' /var/www/html/sgdoc/logica/connect.php
sudo sed -i 's/usuario_postgres/sgdp/g' /var/www/html/sgdoc/logica/connect.php
sudo sed -i 's/5444/5432/g' /var/www/html/sgdoc/logica/connect.php
sudo sed -i 's/sgdoc/sgdp/g' /var/www/html/sgdoc/logica/connect.php
sudo sed -i 's/Error al /Error 3 al/g' /var/www/html/sgdoc/logica/connect.php

sudo systemctl restart apache2
```

Este módulo ocupa una función `ereg_replace()` que no está en las versiones más nuevas de php (>7).

4 Sistema local (personal de CFFM)

el SgdocPhp corre en apache puerto 80 `http://127.0.0.1/sgdoc/`

4.1 sgdp

En la “properties”->Targeted Runtimes hay que seleccionar el wildfly 19.

en Eclipse se instaló wildfly y sobre eso se corrió el sgdp. Hay que detener el tomcat si es que existe, ya que ocupa el mismo puerto

```
sudo systemctl stop tomcat9
```

```
wget https://jdbc.postgresql.org/download/postgresql-42.2.18.jar
```

```
sudo sh ~/wildfly19/wildfly-19.1.0.Final/bin/add-user.sh -u administrador -p gest1469
sudo bash ~/wildfly19/wildfly-19.1.0.Final/bin/jboss-cli.sh --connect --controller=127.0.0.1
sudo bash ~/wildfly19/wildfly-19.1.0.Final/bin/jboss-cli.sh --connect --controller=127.0.0.1
sudo bash ~/wildfly19/wildfly-19.1.0.Final/bin/jboss-cli.sh --connect --controller=127.0.0.1
```

a la consola de wildfly se entra: `http://127.0.0.1:9990/console/index.html`
con administrador y gest1469 y el sgdp en `http://127.0.0.1:8080/sgdp`

5 Documentación

Esta documentación se genera desde el archivo `.md` `pandoc Documentacion.md -o Documentacion.pdf -N --toc`