

PROYECTO “CONSTRUCTORES S.A.S”

Elaborado por

CÉSAR ANDRÉS LATORRE FORERO

INDICE

1. OBJETIVO.....	3
2. INTRODUCCIÓN.....	3
3. ENVIRONMENT.....	4
4. MODELO DE DATOS	4
5. CASOS EXITOSOS DE LAS SOLICITUDES Y ORDENES DE CONSTRUCCIÓN.....	8
6. CASOS NO EXITOSOS DE LAS SOLICITUDES Y ORDENES DE CONSTRUCCION.....	13
7. PROCESO AUTOMATICO DE CAMBIOS DE ESTADO.....	14
8. GENERACION DEL INFORME.....	16

1. OBJETIVO

La empresa “Constructores S.A.S” requiere desarrollar una aplicación capaz de gestionar sus solicitudes de construcción en el terreno designado para la ciudadela, donde las solicitudes de construcción solo serán montadas por el arquitecto.

2. INTRODUCCIÓN

La empresa “Constructores S.AS” requiere de un aplicativo que permita gestionar sus solicitudes de construcción, según el terreno asignado por la ciudadela. El aplicativo debe permitir lo siguiente:

- En la ciudadela se podrán construir cinco diferentes tipos de construcciones, las cuales requieren una cantidad específica de material para poder construirse y llevan un tiempo específico en ser terminada:
 - 1) Casa (Ce: 100, Gr: 50, Ar: 90, Ma: 20, Ad: 100) => 3d
 - 2) Lago (Ce: 50, Gr: 60, Ar: 80, Ma: 10, Ad: 20) => 2d
 - 3) Cancha de fútbol (Ce: 20, Gr: 20, Ar: 20, Ma: 20, Ad: 20) => 1d
 - 4) Edificio (Ce: 200, Gr: 100, Ar: 180, Ma: 40, Ad: 200) => 6d
 - 5) Gimnasio (Ce: 50, Gr: 25, Ar: 45, Ma: 10, Ad: 50) => 2d
- Se requiere especificar el tipo de construcción y en que coordenada (x,y) se realizará la construcción. Si se pudo realizar la solicitud, posterior a esto se crea la orden de construcción con un estado “pendiente”.
- La ciudadela no tiene una fecha exacta de entrega, por lo anterior cada orden de construcción ampliará el plazo de entrega de la ciudadela.
- Cada construcción se arranca el día siguiente al día solicitado o al día siguiente de terminar la construcción previamente programado y termina en la noche del día de finalización calculado.
- Solo se podrá llevar a cabo construcción a construcción (de manera lineal en el tiempo).
- Cada día se debe validar dos cosas:
 - 1) En las mañanas se debe validar si se debe colocar en el estado “en progreso” alguna orden de construcción y actualizar el estado dado el caso.
 - 2) En las noches se debe validar si se debe colocar en el estado “finalizado” alguna orden de construcción y actualizar el estado dado el caso.
- El sistema debe permitir hacer solicitudes de construcción así no haya terminado alguna construcción previa. El sistema solo debe restringir la solicitud de construcción si no posee los materiales y si en las coordenadas ya existe alguna orden de construcción ya sea programada, en progreso o finalizada.

- El sistema debe notificar si fue posible hacer la solicitud de construcción. La creación de la orden de construcción se hace luego de notificarle al usuario que se pudo realizar la solicitud.
- El sistema debe permitir consultar cual es la fecha en que termina el proyecto. El proyecto debe cambiar la fecha dependiendo de los pedidos programados.
- El sistema debe permitir solicitar un informe de que construcciones están pendientes, cuantas por tipo se han terminado y cuantas en progreso.

3. ENVIRONMENT

- Eclipse IDE for Enterprise Java Developers, Versión 2020-01 (4.17.0).
- Open JDK versión 11.
- SQL y PL/SQL Oracle versión 11g.
- Spring Tools 4, Versión 4.11.0.

4. MODELO DE DATOS

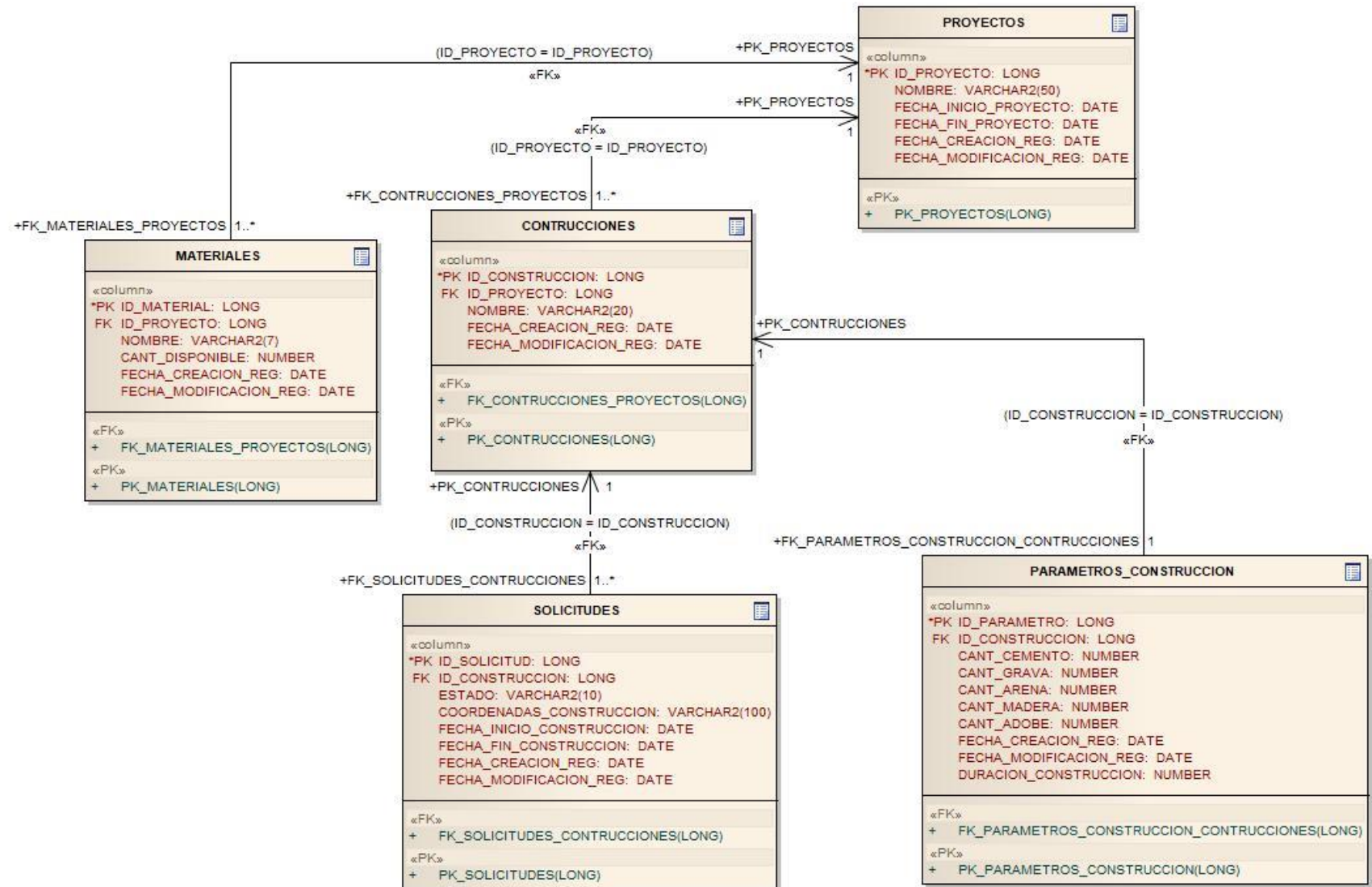
Para el proyecto, se construye las siguientes tablas:

- **PROYECTOS:** Tabla que contiene la información general del proyecto, donde se puede visualizar las fechas de inicio y finalización del mismo. Esta tabla contiene las siguientes columnas:
 - 1) **ID_PROYECTO:** Llave primaria de la tabla.
 - 2) **NOMBRE:** Nombre del proyecto, en este caso es el nombre de la compañía constructora.
 - 3) **FECHA_INICIO_PROYECTO:** Campo que contiene la fecha de inicialización del proyecto. Es la fecha de inicio de la primera orden.
 - 4) **FECHA_FIN_PROYECTO:** Campo que contiene la fecha de finalización del proyecto. Es la fecha fin de la última orden de construcción.
 - 5) **FECHA_CREACION_REG:** Campo que contiene la fecha de creación del registro en la BD.
 - 6) **FECHA_MODIFICACION_REG:** Campo que contiene la fecha de modificación del registro en la BD. Cada vez que se realiza un cambio en el registro (actualización de las fechas de inicio y fin de construcción) este campo se actualiza.
- **CONSTRUCCIONES:** Tabla que contiene la información básica de las construcciones que se van a realizar en la ciudadela.
 - 1) **ID_CONSTRUCCION:** Llave primaria de la tabla.

- 2) **ID_PROYECTO:** Campo que contiene se utiliza como llave foránea con la tabla de PROYECTOS. Así se obtiene de la información de que construcciones están asociadas al proyecto.
 - 3) **NOMBRE:** Nombre de la construcción (Ejemplo, Casa, Lago, Cancha Fútbol...).
 - 4) **FECHA_CREACION_REG:** Campo que contiene la fecha de creación del registro en la BD.
 - 5) **FECHA_MODIFICACION_REG:** Campo que contiene la fecha de modificación del registro en la BD.
- **SOLICITUDES:** Tabla que contiene la información de las solicitudes y estados de las ordenes de construcción.
 - 1) **ID_SOLICITUD:** Llave primaria de la tabla.
 - 2) **ID_CONSTRUCCION:** Campo que se utiliza como llave foránea para acceder a la información de la tabla de CONSTRUCCIONES.
 - 3) **ESTADO:** Estado de las ordenes de construcción (PENDIENTE, EN PROGRESO y FINALIZADO).
 - 4) **COORDENADAS_CONSTRUCCION:** Campo que contiene la información de las coordenadas donde se realizará la construcción.
 - 5) **FECHA_INICIO_CONSTRUCCION:** Campo que contiene la fecha de inicio de la construcción.
 - 6) **FECHA_FIN_CONSTRUCCION:** Campo que contiene la fecha de finalización de la construcción.
 - 7) **FECHA_CREACION_REG:** Campo que contiene la fecha de creación del registro en la BD.
 - 8) **FECHA_MODIFICACION_REG:** Campo que contiene la fecha de modificación del registro en la BD. Cada vez que se realiza un cambio en el registro (actualización de las fechas de inicio y fin de construcción) este campo se actualiza.
 - **PARAMETROS_CONSTRUCCION:** Tabla que contiene la información concerniente a la cantidad de material que se requiere cada construcción.
 - 1) **ID_PARAMETRO:** Llave primaria de la tabla.
 - 2) **ID_CONSTRUCCION:** Campo que es utilizado como llave foránea para acceder a la tabla de CONSTRUCCION.
 - 3) **CANT_CEMENTO:** Cantidad de cemento que se requiere para determinada construcción.
 - 4) **CANT_GRAVA:** Cantidad de grava que se requiere para determinada construcción.
 - 5) **CANT_ARENA:** Cantidad de arena que se requiere para determinada construcción.

- 6) **CANT_MADERA:** Cantidad de madera que se requiere para determinada construcción.
 - 7) **CANT_ADOBE:** Cantidad de adobe que se requiere para determinada construcción
 - 8) **FECHA_CREACION_REG:** Campo que contiene la fecha de creación del registro en la BD.
 - 9) **FECHA_MODIFICACION_REG:** Campo que contiene la fecha de modificación del registro en la BD.
 - 10) **DURACION_CONSTRUCCION:** Cantidad de días en que se debe realizar determinada construcción.
- **MATERIALES:** Tabla que contiene la información de la cantidad de material disponible para cada proyecto.
 - 1) **ID_MATERIAL:** Llave primaria de la tabla.
 - 2) **NOMBRE:** Nombre del material.
 - 3) **CANT_DISPONIBLE:** Cantidad disponible de cada material para el proyecto.
 - 4) **FECHA_MODIFICACION_REG:** Campo que contiene la fecha de modificación del registro en la BD.
 - 5) **DURACION_CONSTRUCCION:** Cantidad de días en que se debe realizar determinada construcción.

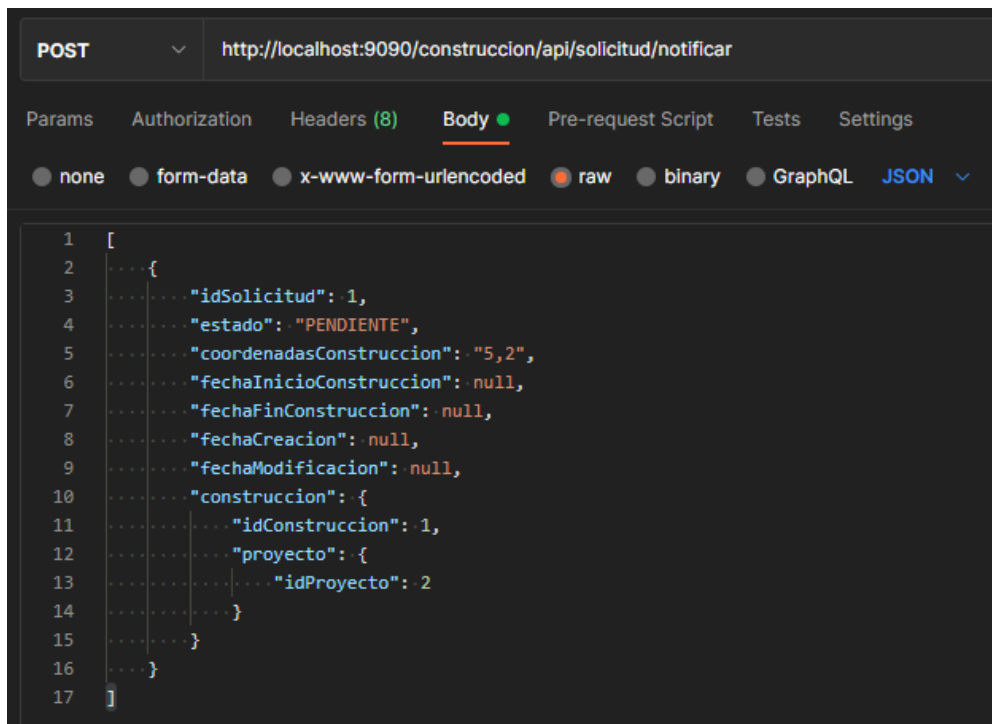
class Esquema1



5. CASOS EXITOSOS DE LAS SOLICITUDES Y ORDENES DE CONSTRUCCIÓN

El proceso de la solicitud y de las ordenes de construcción se realiza de la siguiente forma:

- 1) Se ejecuta el siguiente endpoint para la generación de la solicitud: <http://localhost:9090/construccion/api/solicitud/notificar>.
- 2) En el JSON es importante enviar el estado de la solicitud, el id, las coordenadas, el identificador del proyecto al que pertenece la construcción y el identificador del tipo de construcción (Si va ser una Casa, estado, lago...).



```
POST http://localhost:9090/construccion/api/solicitud/notificar

Body

[
  {
    "idSolicitud": 1,
    "estado": "PENDIENTE",
    "coordenadasConstruccion": "5,2",
    "fechaInicioConstruccion": null,
    "fechaFinConstruccion": null,
    "fechaCreacion": null,
    "fechaModificacion": null,
    "construccion": {
      "idConstruccion": 1,
      "proyecto": {
        "idProyecto": 2
      }
    }
  }
]
```

- 3) Cuando se envía la solicitud, el sistema responde con un JSON que le indica que la solicitud ha sido realizada exitosamente:



```
Body Cookies Headers (5) Test Results

Status: 201 Created Time: 217 ms Size: 262 B Save Response

Pretty Raw Preview Visualize JSON

{
  "estadoNotificacion": "OK",
  "mensajeNotificacion": "Se ha realizado la solicitud exitosamente"
}
```

- 4) Al ingresar a la BD se puede visualizar las fechas de inicio y fin del proyecto, donde se garantiza que la fecha de creación de registro es la fecha en que se generó la solicitud, y la construcción inicia al día siguiente de ser aprobada. También se puede visualizar que la cantidad de días entre el inicio y el fin de la construcción corresponde al tiempo máximo para la construcción de una casa.


```

4 select co.nombre
5       ,sc.estado
6       ,sc.fecha_inicio_construccion
7       ,sc.fecha_fin_construccion
8       ,sc.fecha_creacion_reg
9 from solicitudes sc
10 inner join construcciones co
11      on co.id_construccion = sc.id_construccion;

```

	NOMBRE	ESTADO	FECHA_INICIO_CONSTRUCCION	FECHA_FIN_CONSTRUCCION	FECHA_CREACION_REG
1	CASA	PENDIENTE	21/08/21	25/08/21	20/08/21

5) Se realiza una nueva solicitud para la construcción de un lago:

```

POST http://localhost:9090/construccion/api/solicitud/notificar

Body
[
  {
    "idSolicitud": 2,
    "estado": "PENDIENTE",
    "coordenadasConstruccion": "7,3",
    "fechaInicioConstruccion": null,
    "fechaFinConstruccion": null,
    "fechaCreacion": null,
    "fechaModificacion": null,
    "construccion": {
      "idConstruccion": 2,
      "proyecto": {
        "idProyecto": 2
      }
    }
  }
]

```

6) Aunque aún no se ha terminado la construcción de la casa, el sistema responde OK a la solicitud:

```

Body
{
  "estadoNotificacion": "OK",
  "mensajeNotificacion": "Se ha realizado la solicitud exitosamente"
}

```

7) Al ingresar a la BD se puede visualizar las fechas de inicio y fin del proyecto, donde se garantiza que la construcción del lago solo empezará cuando se termine la construcción de la casa:

```

5 select co.nombre
6     ,sc.estado
7     ,sc.fecha_inicio_construccion
8     ,sc.fecha_fin_construccion
9     ,sc.fecha_creacion_reg
10 from solicitudes sc
11 inner join construcciones co
12     on co.id_construccion = sc.id_construccion;

```

	NOMBRE	ESTADO	FECHA_INICIO_CONSTRUCCION	FECHA_FIN_CONSTRUCCION	FECHA_CREACION_REG
1	CASA	PENDIENTE	21/08/21	25/08/21	20/08/21
2	LAGO	PENDIENTE	26/08/21	29/08/21	20/08/21

- 8) Al ejecutar el endpoint <http://localhost:9090/construccion/api/proyecto/all> se genera un JSON con la información del inicio y fin del proyecto. Se puede observar que el proyecto inició con la fecha de inicio de la casa y el proyecto finaliza cuando se termine la construcción del lago:

Body Cookies Headers (5) Test Results 200 OK 326 ms 360 B Save Response

Pretty Raw Preview Visualize JSON

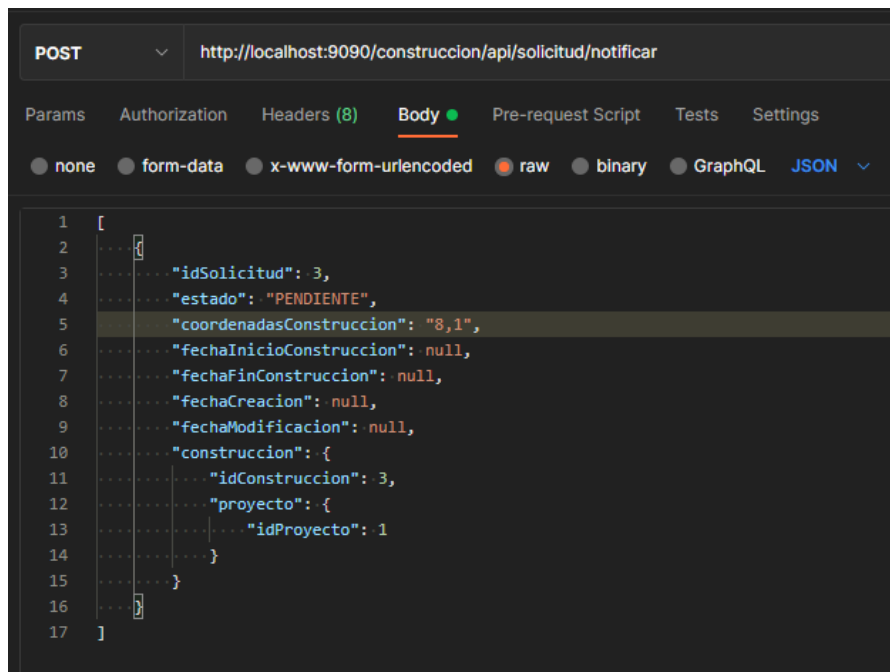
```

1 [
2   {
3     "idProyecto": 1,
4     "nombre": "Constructores S.A.S",
5     "fechaInicioProyecto": "2021-08-21T09:43:43",
6     "fechaFinProyecto": "2021-08-29T09:43:43",
7     "fechaCreacion": "2021-08-17T00:00:00",
8     "fechaModificacion": null
9   }
10 ]

```

- 9) La fecha final del proyecto es dinámica, si se realiza la solicitud de la construcción de otra obra, la fecha cambia:

Solicitud:

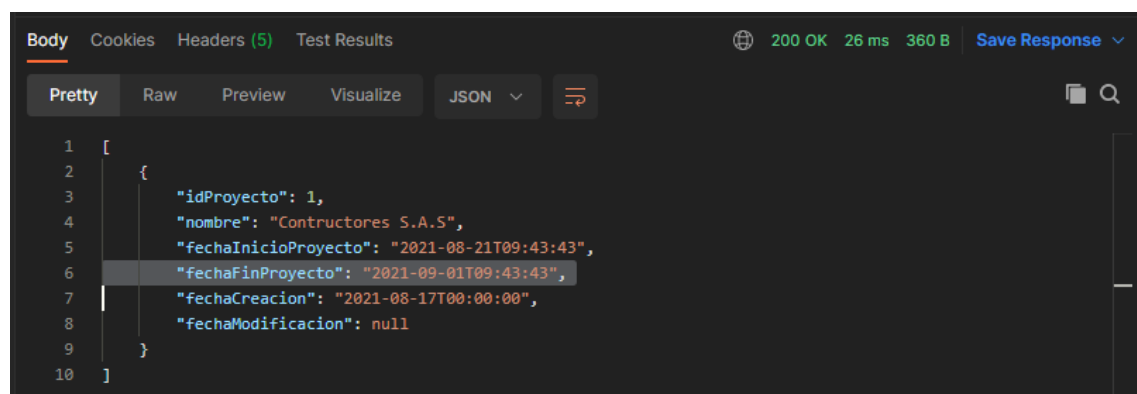


Registros de las solicitudes en la BD:

```
5 select co.nombre
6      ,sc.estado
7      ,sc.fecha_inicio_construccion
8      ,sc.fecha_fin_construccion
9      ,sc.fecha_creacion_reg
10 from solicitudes sc
11 inner join construcciones co
12      on co.id_construccion = sc.id_construccion;
```

	NOMBRE	ESTADO	FECHA_INICIO_CONSTRUCCION	FECHA_FIN_CONSTRUCCION	FECHA_CREACION_REG
1	CASA	PENDIENTE	21/08/21	25/08/21	20/08/21
2	LAGO	PENDIENTE	26/08/21	29/08/21	20/08/21
3	CANCHA DE FÚTBOL	PENDIENTE	30/08/21	01/09/21	20/08/21

Nueva fecha de finalización del proyecto:



- 10) En la creación de cada solicitud se valida la cantidad disponible de material para la construcción. La información del material disponible se encuentra en la tabla MATERIALES:

```
14 select * from materiales;
```

	ID_MATERIAL	NOMBRE	CANT_DISPONIBLE	FECHA_CREACION_REG	FECHA_MODIFICACION_REG
1	1	CEMENTO	7230	17/08/21	20/08/21
2	2	GRAVA	8570	17/08/21	20/08/21
3	3	ARENA	7470	17/08/21	20/08/21
4	4	MADERA	9430	17/08/21	20/08/21
5	5	ADOBE	7260	17/08/21	20/08/21

- 11) La cantidad disponible de material disminuye cada vez que se aprueba una nueva solicitud. Por ejemplo:

Se realiza una nueva solicitud (EDIFICIO):

http://localhost:9090/construccion/api/solicitud/notificar

POST http://localhost:9090/construccion/api/solicitud/notificar

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "idSolicitud": 4,
3   "estado": "PENDIENTE",
4   "coordenadasConstruccion": "8,1",
5   "fechaInicioConstruccion": null,
6   "fechaFinConstruccion": null,
7   "fechaCreacion": null,
8   "fechaModificacion": null,
9   "construccion": {
10    "idConstruccion": 4,
11    "proyecto": {
12      "idProyecto": 1
13    }
14  }
15 }
16 }
17 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 69 ms Size: 262 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "estadoNotificacion": "OK",
3   "mensajeNotificacion": "Se ha realizado la solicitud exitosamente"
4 }
```

Se valida en la BD la creación de la nueva solicitud:

```

5 select co.nombre
6      ,sc.estado
7      ,sc.fecha_inicio_construccion
8      ,sc.fecha_fin_construccion
9      ,sc.fecha_creacion_reg
10 from solicitudes sc
11 inner join construcciones co
12      on co.id_construccion = sc.id_construccion
13 order by fecha_inicio_construccion asc;
14

```

	NOMBRE	ESTADO	FECHA_INICIO_CONSTRUCCION	FECHA_FIN_CONSTRUCCION	FECHA_CREACION_REG
1	CASA	PENDIENTE	21/08/21	25/08/21	20/08/21
2	LAGO	PENDIENTE	26/08/21	29/08/21	20/08/21
3	CANCHA DE FÚTBOL	PENDIENTE	30/08/21	01/09/21	20/08/21
4	EDIFICIO	PENDIENTE	02/09/21	09/09/21	20/08/21

Se valida la cantidad de material en la BD

```

17 select * from materiales;

```

	ID_MATERIAL	NOMBRE	CANT_DISPONIBLE	FECHA_CREACION_REG	FECHA_MODIFICACION_REG
1	1	CEMENTO	7030	17/08/21	20/08/21
2	2	GRAVA	8470	17/08/21	20/08/21
3	3	ARENA	7290	17/08/21	20/08/21
4	4	MADERA	9390	17/08/21	20/08/21
5	5	ADOBE	7060	17/08/21	20/08/21

Se puede visualizar que las cantidades disponibles de cada material ha disminuido por la nueva orden solicitada.

6. CASOS NO EXITOSOS DE LAS SOLICITUDES Y ORDENES DE CONSTRUCCION

No se puede realizar una solicitud cuando se presentan los siguientes dos escenarios:

- Cuando no existe material suficiente para la nueva construcción

Ejemplo:

- Se modifica los valores concernientes a la cantidad de material disponible para el proyecto de tal forma que sea insuficiente para solicitar una nueva construcción:

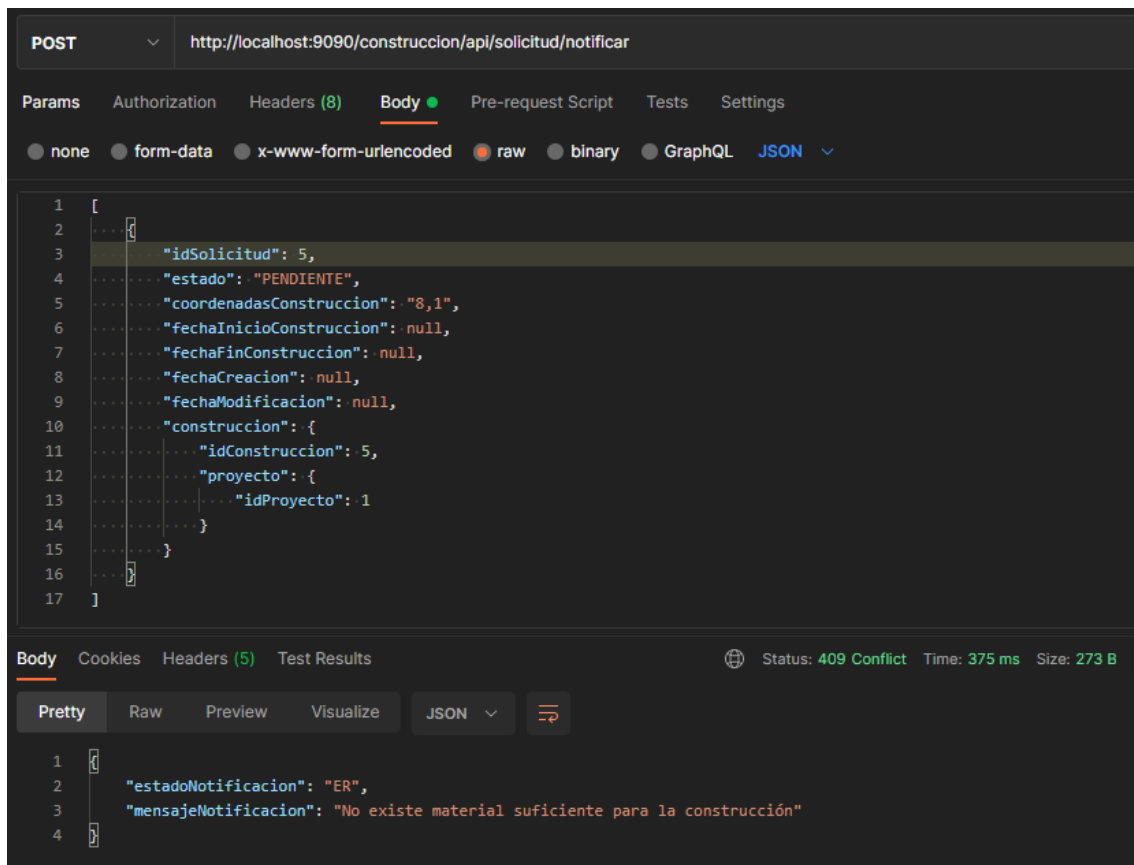
```

17 select * from materiales for update;

```

	ID_MATERIAL	NOMBRE	CANT_DISPONIBLE	FECHA_CREACION_REG	FECHA_MODIFICACION_REG
1	1	CEMENTO	10	17/08/2021	20/08/2021 10:26:27
2	2	GRAVA	5	17/08/2021	20/08/2021 10:26:27
3	3	ARENA	8	17/08/2021	20/08/2021 10:26:27
4	4	MADERA	21	17/08/2021	20/08/2021 10:26:27
5	5	ADOBE	9	17/08/2021	20/08/2021 10:26:27

b) Se genera una nueva solicitud



Se puede visualizar que el sistema respondió con error indicando que no se pudo generar la solicitud por falta de material.

7. PROCESO AUTOMATICO DE CAMBIOS DE ESTADO



















Cuando se realiza una solicitud, esta queda en estado “PENDIENTE” como se puede observar en la siguiente imagen:

SQL

Output

Statistics

```
1 SELECT SC.NOMBRE,
2       SO.ESTADO,
3       SO.FECHA_INICIO_CONSTRUCCION,
4       SO.FECHA_FIN_CONSTRUCCION,
5       SO.FECHA_CREACION_REG
6 FROM SOLICITUDES SO
7 INNER JOIN CONSTRUCCIONES SC
8 ON SC.ID_CONSTRUCCION = SO.ID_CONSTRUCCION;
```

	NOMBRE		ESTADO	FECHA_INICIO_CONSTRUCCION	FECHA_FIN_CONSTRUCCION	FECHA_CREACION_REG	
▶ 1	CASA	...	PENDIENTE	21/08/21	...	25/08/21	...
2	LAGO	...	PENDIENTE	26/08/21	...	29/08/21	...
3	CANCHA DE FÚTBOL	...	PENDIENTE	30/08/21	...	01/09/21	...

Para garantizar el cambio de estados se crean los siguientes JOBS Oracle:

- **JOB_ACT_ORDENES_INICIAL:** Job que permite actualizar el estado PENDIENTE a PROGRESO de las ordenes de construcción.

```
BEGIN
SYS.DBMS_SCHEDULER.CREATE_JOB (
  job_name      => 'JOB_ACT_ORDENES_INICIAL'
,job_type      => 'STORED PROCEDURE'
,start_date    => TO_TIMESTAMP_TZ('19/08/2021 08:00:00.000 AM -05:00','dd/MM/yyyy hh12:mi:ss.ff AM tzz')
,repeat_interval => 'FREQ=DAILY'
,auto_drop     => TRUE
,job_action    => 'PR_ACT_ESTADO_INICIAL'
,comments      => 'Job que permite actualizar el estado PENDIENTE a PROGRESO de las ordenes de construccion.
);
DBMS_SCHEDULER.ENABLE
(
  name      => 'JOB_ACT_ORDENES_INICIAL'
);
END;
```

Como se puede observar en la imagen del JOB_ACT_ORDENES_INICIAL en el campo “start_date” se configura para que el JOB se ejecute a las 8 de la mañana con la zona horaria GTM-5 que le corresponde a Colombia. En el campo “job_action” se le indica que el JOB debe ejecutar el procedimiento PR_ACT_ESTADO_INICIAL que es el que permite cambiar el estado PENDIENTE a PROGRESO y en el campo “repeat-interval” se le indica que debe ejecutarse diariamente.

- **JOB_ACT_ORDENES_FINALIZADO:** Job que permite actualizar el estado PROGRESO a FINALIZADO de las ordenes de construcción.

```
BEGIN
SYS.DBMS_SCHEDULER.CREATE_JOB (
  job_name      => 'JOB_ACT_ORDENES_FINALIZADO'
,job_type      => 'STORED PROCEDURE'
,start_date    => TO_TIMESTAMP_TZ('19/08/2021 08:00:00.000 PM -05:00','dd/MM/yyyy hh12:mi:ss.ff AM tzz')
,repeat_interval => 'FREQ=DAILY'
,auto_drop     => TRUE
,job_action    => 'PR_ACT_ESTADO_FINALIZADO'
,comments      => 'Job que permite actualizar el estado PROGRESO a FINALIZADO de las ordenes de construccion'
);
DBMS_SCHEDULER.ENABLE
(
  name      => 'JOB_ACT_ORDENES_FINALIZADO'
);
END;
```

Como se puede observar en la imagen del JOB_ACT_ORDENES_FINALIZADO en el campo “start_date” se configura para que el JOB se ejecute a las 8 de la noche con la zona horaria GTM-5 que le corresponde a Colombia. En el campo “job_action” se le indica que el JOB debe ejecutar el procedimiento PR_ACT_ESTADO_FINAL que es el que permite cambiar el estado PROGRESO a FINALIZADO y en el campo “repeat-interval” se le indica que debe ejecutarse diariamente.

El seguimiento de las ejecuciones de cada JOB se puede hacer realizando la siguiente consulta en la BD:

```
SELECT JOB.job_name,
       JOB.start_date,
       JOB.REPEAT_INTERVAL,
       JOB.last_start_date,
       JOB.LAST_RUN_DURATION,
       JOB.next_run_date
FROM ALL_SCHEDULER_JOBS JOB
WHERE JOB_NAME IN ('JOB_ACT_ORDENES_INICIAL',
                  'JOB_ACT_ORDENES_FINALIZADO');
```

```

11 SELECT JOB.job_name,
12        JOB.start_date,
13        JOB.REPEAT_INTERVAL,
14        JOB.last_start_date,
15        JOB.LAST_RUN_DURATION,
16        JOB.next_run_date
17 FROM ALL_SCHEDULER_JOBS JOB
18 WHERE JOB_NAME IN ('JOB_ACT_ORDENES_INICIAL',
19                  'JOB_ACT_ORDENES_FINALIZADO');

```

	JOB_NAME	START_DATE	REPEAT_INTERVAL	LAST_START_DATE	LAST_RUN_DURATION	NEXT_RUN_DATE
1	JOB_ACT_ORDENES_INICIAL	19/08/21 08:00:00,000000 -05:00	FREQ=DAILY	20/08/21 08:00:00,385000 -05:00	+000000000 00:00:00,063000	21/08/21 08:00:00,405000 -05:00
2	JOB_ACT_ORDENES_FINALIZADO	19/08/21 20:00:00,000000 -05:00	FREQ=DAILY	20/08/21 07:58:31,122000 -05:00	+000000000 00:00:00,125000	20/08/21 20:00:00,137000 -05:00

8. GENERACION DEL INFORME

El informe se genera al ejecutar el siguiente endpoint:
<http://localhost:9090/construccion/api/informe/generar>

GET <http://localhost:9090/construccion/api/informe/generar> Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 366 ms 250 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "estado": "OK",
3   "mensaje": "Se generó el informe",
4   "ruta": "C:\\Users\\WIN10\\Documents"
5 }

```

El sistema responde con un JSON indicando que se ha generado el informe exitosamente y guardado en una ruta específica. Este se genera con el nombre de “Informe_Estados.txt” y el resultado es el siguiente:

Fecha de generación: 20/08/21

```

=====
                INFORME DE ESTADOS DE LAS CONSTRUCCIONES
=====
PENDIENTE                EN PROCESO                FINALIZADO
-----
3                        0                        0
|

```