

---

# **Módulo Laboratorio : Rodam Análisis**

***Versión 1.0***

**Jonatan Ahumada Fernández**

**30 de marzo de 2022**



---

## Contents:

---

<b>1. Documento del Proyecto</b>	<b>1</b>
1.1. Módulo Laboratorio . . . . .	1
1.2. Estimaciones . . . . .	2
1.3. WBS . . . . .	2
<b>2. Documento de Especificación de Requerimientos</b>	<b>3</b>
2.1. Introducción . . . . .	3
2.2. Descripción General . . . . .	3
2.3. Funcionalidades del sistema . . . . .	4
<b>3. Documento de Diseño</b>	<b>9</b>
3.1. Introducción . . . . .	9
3.2. Guia conceptual . . . . .	10
3.3. Diseño de base de datos . . . . .	10
3.4. El control de calidad . . . . .	13
3.5. Emision de certificados . . . . .	13
<b>4. Plan de pruebas</b>	<b>15</b>
4.1. Introducción . . . . .	15
4.2. Necesidades del tipo de pruebas . . . . .	15
4.3. Estimación de riesgos . . . . .	15
4.4. La organización de las pruebas . . . . .	16
4.5. Pruebas unitarias . . . . .	16
4.6. Pruebas de integracion . . . . .	17
4.7. Pruebas manuales . . . . .	17
4.8. Validación . . . . .	17
<b>5. Prueba de aceptación</b>	<b>19</b>
5.1. Resumen . . . . .	19
5.2. Documentos relacionados . . . . .	19
5.3. Objetivo . . . . .	20
5.4. Definiciones . . . . .	20
5.5. Configurando la sesion . . . . .	20
5.6. Creación de análisis . . . . .	27
5.7. Creación de Grupos . . . . .	31
5.8. Ingreso de un producto . . . . .	33
5.9. Especificar un producto . . . . .	35

5.10.	Ingreso de una muestra . . . . .	43
5.11.	Ingreso de lectura . . . . .	43
5.12.	Informacion del cliente . . . . .	49
5.13.	Información de la muestra . . . . .	50
5.14.	Resumen analisis . . . . .	50
5.15.	Equipos utilizados . . . . .	51
<b>6.</b>	<b>Reporte de pruebas</b>	<b>53</b>
<b>7.</b>	<b>Diccionario de datos</b>	<b>59</b>
<b>8.</b>	<b>Reporte de validación</b>	<b>63</b>
8.1.	Introducción y alcance . . . . .	63
8.2.	Descripción del sistema . . . . .	63
8.3.	Manejo de riesgo . . . . .	64
8.4.	Riesgo 2: el artefacto principal, el certificado de emisión, es incompleto . . . . .	64
8.5.	Estrategia de validación . . . . .	66
8.6.	Entregables . . . . .	66
<b>9.</b>	<b>Scripts</b>	<b>67</b>
9.1.	Sincronización entre desarrollo y produccion . . . . .	67

---

## Documento del Proyecto

---

**author** Jonatan Ahumada Fernández

**contact** [jaumaf@hotmail.com](mailto:jaumaf@hotmail.com)

**date** último build el 2022-03-29

El objetivo de este documento es resumir y dar trazabilidad al desarrollo del Sistema de Información de Rodam Análisis.

El sistema documental del laboratorio caracteriza los procesos del establecimiento de la siguiente manera :

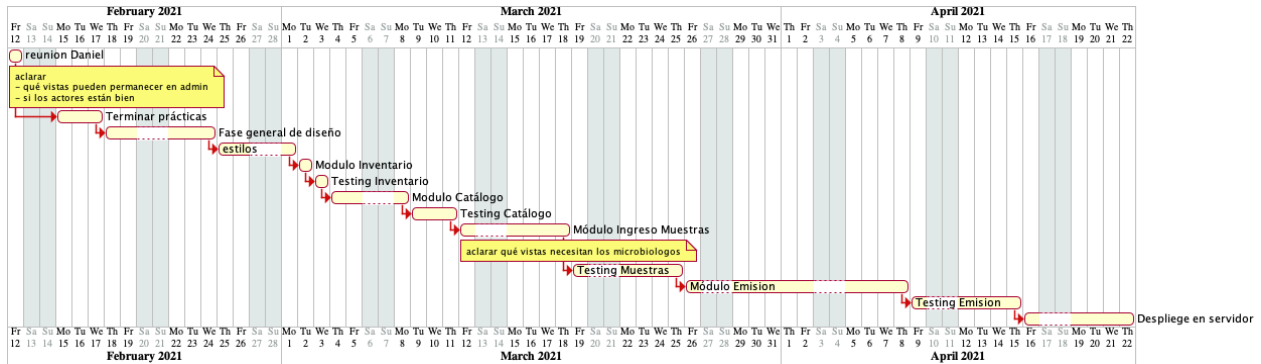
- Proceso de análisis de laboratorio
- Proceso de aseguramiento de la calidad
- Proceso de gestión del cliente
- proceso de gestión del medio ambiente

En vista de que la implementación de todo el sistema de información tiene un alcance considerable. Se optó por segmentar el proyecto en pedazos más pequeños.

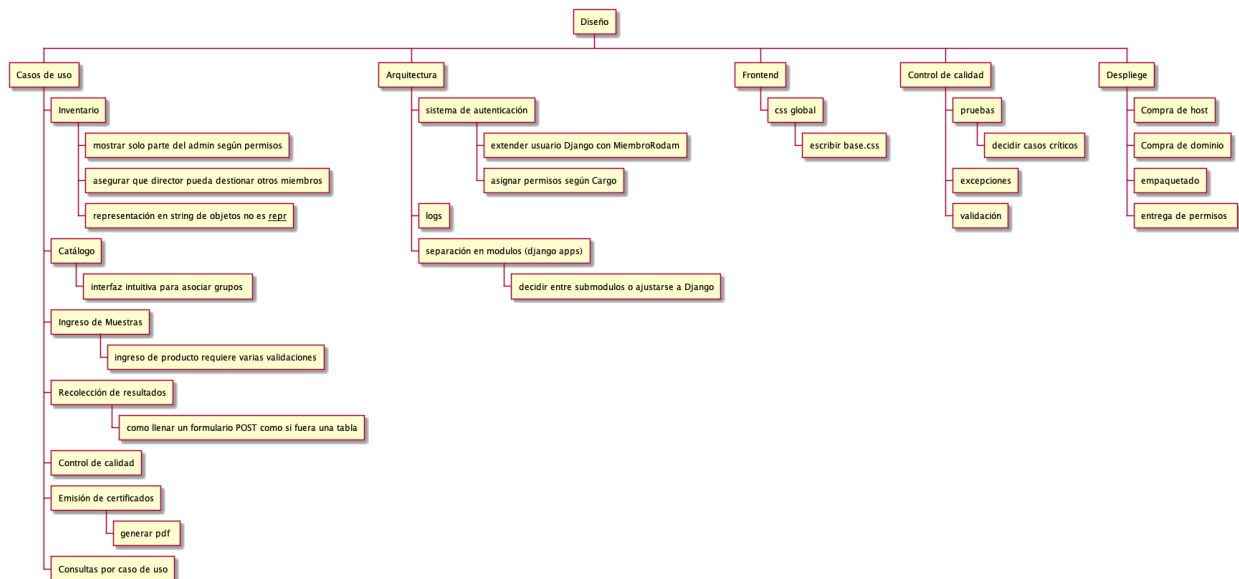
### 1.1 Módulo Laboratorio

Se optó por empezar por el **Módulo Laboratorio**, en vista de que el cliente priorizaba la agilización del proceso de análisis del laboratorio. Para mayor información ver la *Especificación de Requerimientos* para el **Módulo Laboratorio**.

## 1.2 Estimaciones



## 1.3 WBS



---

# Documento de Especificación de Requerimientos

---

**author** Jonatan Ahumada Fernández

**contact** [jaumaf@hotmail.com](mailto:jaumaf@hotmail.com)

**date** último build el 2022-03-29

## 2.1 Introducción

Este documento es el *SRS* para el **Módulo Laboratorio** del **Sistema de Información de Rodám Análisis S.A.** Su propósito es describir los requerimientos que debe cumplir el **Módulo Laboratorio**

Este documento está dirigido principalmente a los miembros del equipo de desarrollo, pero también será de gran utilidad para los *stakeholders* que se pregunten ¿qué hace el **Módulo Laboratorio**?

### 2.1.1 Referencias a otros documentos

Para una visión general del **Sistema de Información de Rodám Análisis S.A** se debe consultar el *Documento de proyecto*.

## 2.2 Descripción General

El **\*Módulo Laboratorio\*** se ocupa del ciclo de vida de los *análisis de laboratorio*. Rodam Análisis caracteriza su proceso de análisis de laboratorio de la siguiente manera:

1. Ingreso
2. Preprocesamiento
3. Siembra
4. Lectura

5. Recolección de resultados
6. Emisión de certificados

El caso de uso principal de la *aplicación de escritorio* será registrar el flujo de un análisis, desde que entra el sistema mediante su ingreso, hasta la producción de su certificado de análisis.

Por su parte, el **Módulo Web** se ocupará de emitir esos certificados a los clientes de Rodam Análisis por medio de una página web.

## 2.3 Funcionalidades del sistema

Nos referimos a las funcionalidades del sistema como *casos de uso*. Es útil considerar a los casos de uso como pertenecientes *submódulo*. A esto lo llamamos *caracterización de los casos de uso*. Esto facilita saber a qué parte del **Módulo Laboratorio** nos referimos.

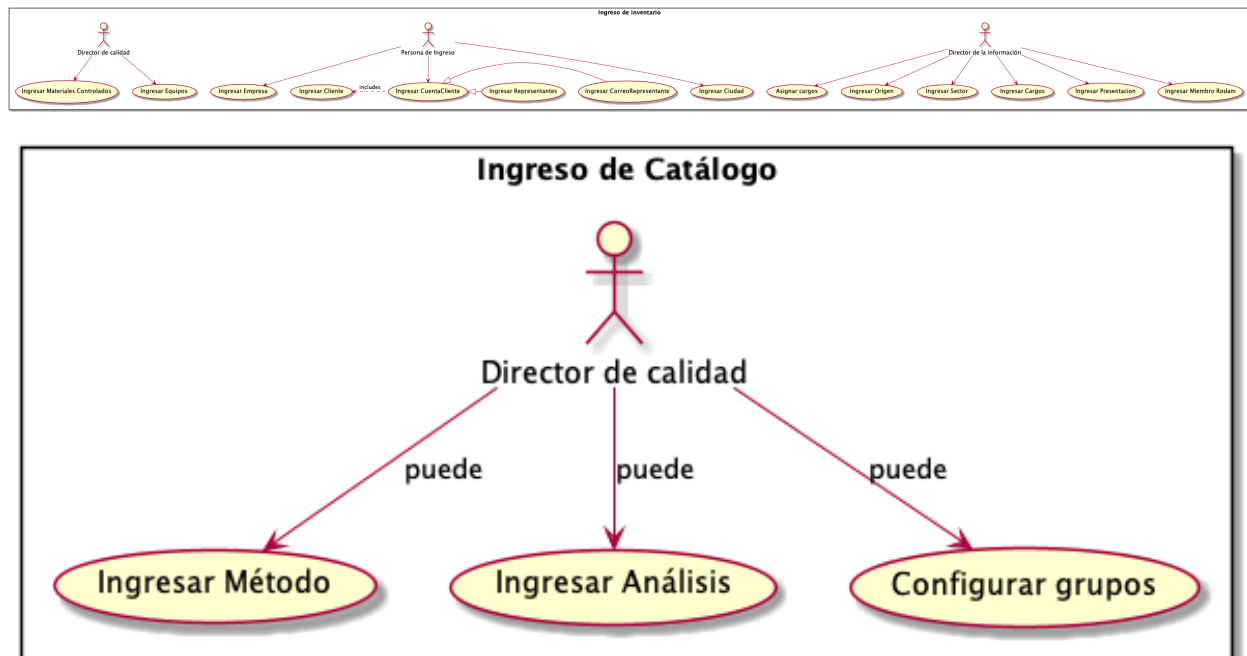
Los submódulos existentes en este momento son:

1. Inventario 3. Catálogo 2. Ingreso de Muestras 3. Recolección de resultados 4. Control de calidad 4. Emisión de certificados

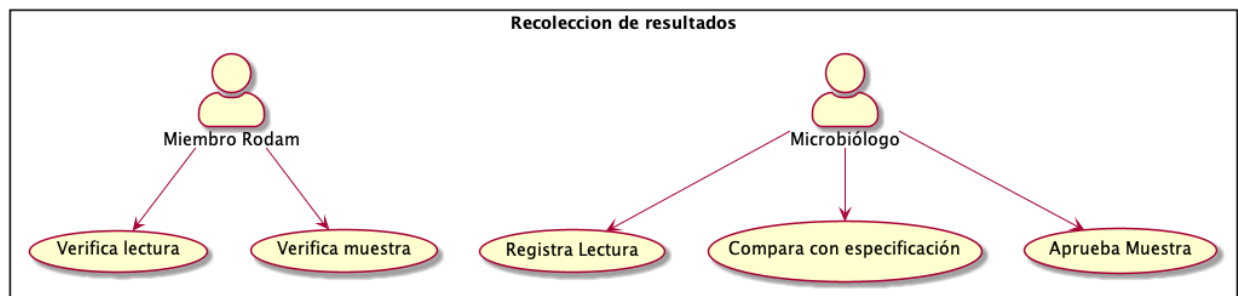
### 2.3.1 Caracterización de los casos de uso

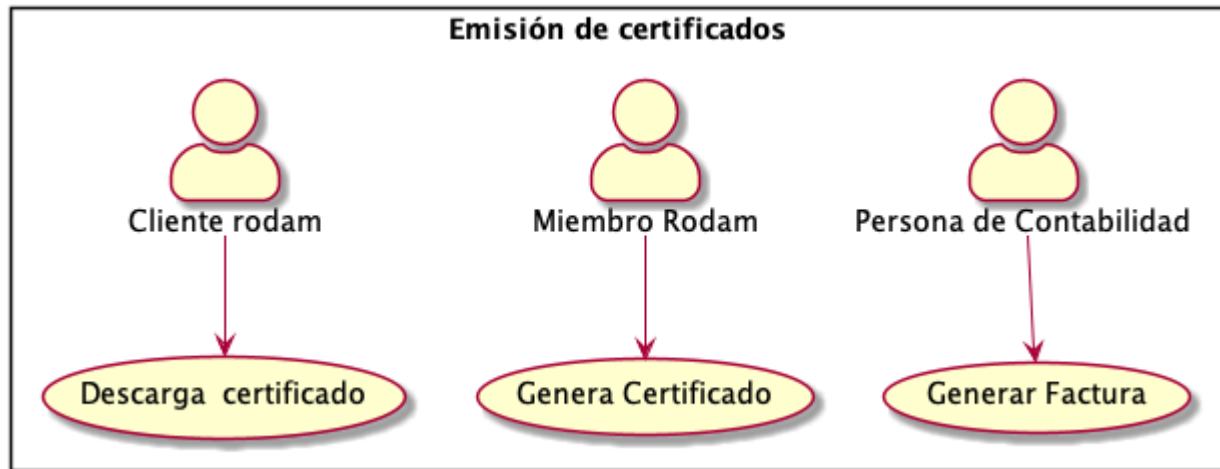
Ahora veremos cómo estos casos de uso se subdividen naturalmente en submódulos. Son agrupaciones de tareas relacionadas dentro del **Modulo Laboratorio**.

A continuación veremos los casos de uso en más detalle.





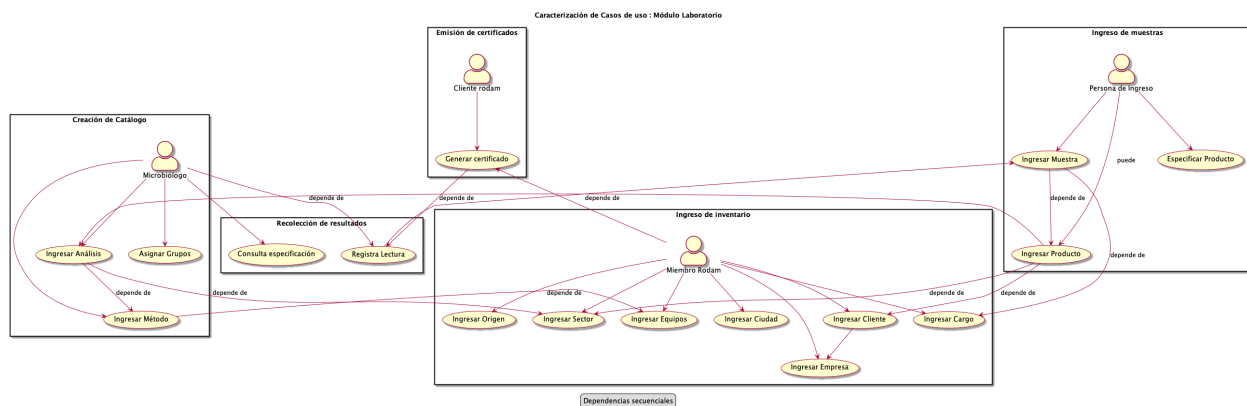




### 2.3.2 Dependencias secuenciales

Las dependencias secuenciales de los casos de uso muestran qué funcionalidades dependen de que otras se hayan ejecutado previamente. Por ejemplo, ingresar un análisis depende de que haya métodos que agregar en ese análisis.

Esta visualización es útil para diseñar el flujo de eventos de una interfáz de usuario.



### 2.3.3 Requisitos no funcionales

- portal web debe ceñirse a libro de marca
- la pagina debe ser accesible por los clientes en internet y no presentar demoras de más de 1 minuto al cargar la página
- asegurar la integridad de los datos (backups, logs) regulares por semana.
- implementar un sistema de roles y permisos (confidencialidad)
- se deben seguir las guías de GAMP

## **2.3.4 Requisitos detallados**

### **1. autenticar usuario**

Al ingresar a la url del sitio web hay una vista para autenticarse. No es necesario registrarse a la página pues el Director de la información se encarga de ingresar nuevos usuarios.

Una vez el usuario se autentica. Se le muestra un dashboard. El dashboard mostrará únicamente las acciones que ese tipo de usuario tiene permitidos. (Ver casos de uso)

### **Flujos alternos**

1. Si las credenciales son invalidas se deberá redirigir a la pagina de login y el formulario debe indicar que se incurrió en un error de validación. Basta con reportar un solo tipo de error (no hay que diferenciar error de usuario o contraseña o permitir registro), ya que la responsabilidad recaera en el director de la información de pasar los datos correctamente de manera interna en el laboratorio.

### **2. ingresar muestra**

La persona de ingreso accede a la url. Se autentica. Luego, en el dashboard selecciona “Ingreso de muestras”.

Se le muestra el formulario muestras que consta de los siguientes campos:

- id\_origen : combo box seleccionable
- presentacion : combo box seleccionable
- lote\_muestra : campo de entrada de texto. Se debe validar siempre un valor numérico
- ingreso\_muestra : La fecha del ingreso de la muestra
- descripcion : un text area
- tamano\_muestra : campo numero
- unidades\_tamano: campo de texto

El usuario diligencia todos y cada uno de los campos.

### **Flujos alternos**

1. Si hay problemas de validación la página redirige a sí misma con los datos del POST y las validaciones pertinentes

### **3. control de los medios**

El director de calidad selecciona “control de los medios” en su dashboard. Una vista le muestra las muestras que aún no cuentan con un control hecho. En la lista de muestras hay una columna con un enlace que redirige al formulario detalle de control de calidad. Cuando se da click, redirige al formulario. Este consta de los siguientes campos:

- lote\_material : textfield. Valida entrada numerica
- id\_material\_de\_control : Combobox seleccionable que muestra no el id, sino el nombre,
- lote\_material\_de\_control : textfield. Valida entrada numerica
- especificacion : textfield
- resultado : textfield

- fecha\_resultado : datepicker
- concepto : textarea

1. Si hay problemas de validación la página redirige a sí misma con los datos del POST y las validaciones pertinentes

Muestras para control		
Muestra	estado	detalle
12321	pendiente	link
43211	iniciado	link
14122	pendiente	link
132123	pendiente	link
12323	iniciado	link
123232	pendiente	link
34341	pendiente	link
343342	iniciado	link
123312	pendiente	link

lote material	
especificacion	
resultado	
lote material de control	
material x	▼
feccha	
concepto	
<input type="button" value="Ingresar"/>	

**author** Jonatan Ahumada Fernández

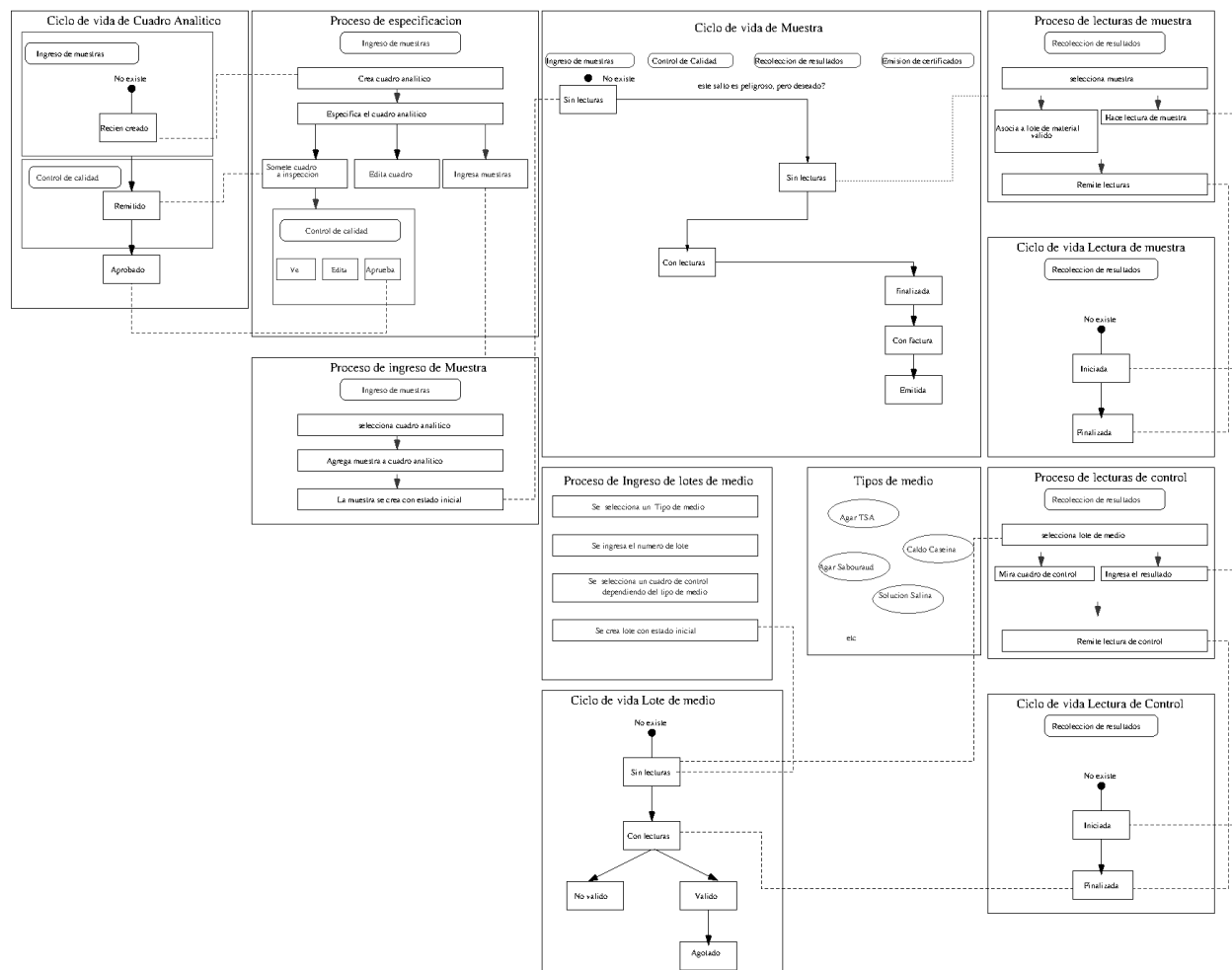
**contact** [jaumaf@hotmail.com](mailto:jaumaf@hotmail.com)

**date** último build el 2022-03-29

### 3.1 Introducción

Aquí se documenta el diseño del módulo *laboratorio*. Para ver qué funcionalidades debe cumplir, se debe revisar la *Especificación de Requerimientos*.

## 3.2 Guia conceptual



## 3.3 Diseño de base de datos

La parte más crítica del sistema se considera el diseño de la base de datos, pues eventualmente reemplazará parte del sistema documental de Rodam Análisis. Esta será la capa sobre la cual se construyan interfaces gráficas. Es, entonces, importante que las relaciones y las tablas sean suficientes.

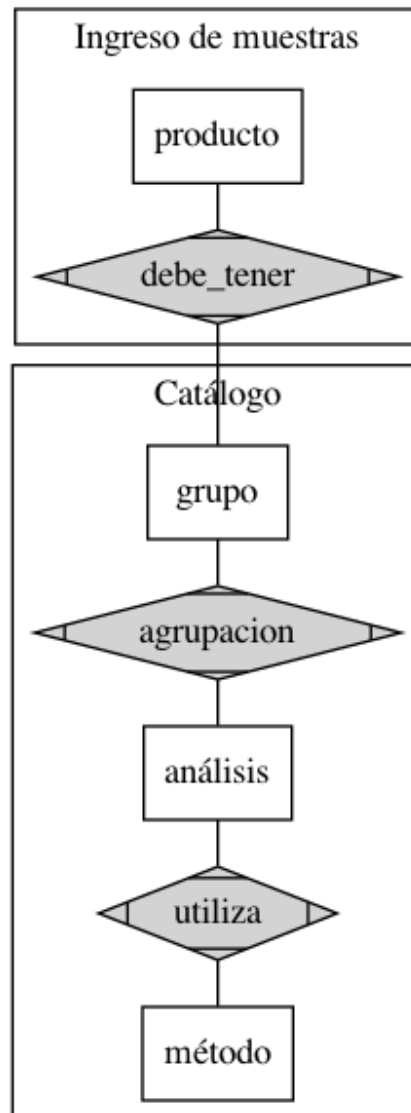
El diseño conceptual y el mapeo de los diagramas ER hacia los modelos relacionales es un proceso costoso en tiempo. Por las necesidades del desarrollo, se han optado por ciclos de desarrollo cortos. Por ende, aquí se expondrán solo los diagramas para las casos de uso más críticos.

### 3.3.1 El ingreso de muestras

El ingreso de muestras es un proceso bastante frecuente en el laboratorio y debe automatizarse al mayor grado posible. En el ingreso se relaciona un producto que proviene de un cliente del laboratorio con uno (o más) servicios que ofrece el laboratorio.

Un producto, casi siempre, es objeto de la misma serie de procedimientos, sujetos a los mismos parámetros. La asociación del producto con los servicios del laboratorio, por lo tanto, se decidió hacer una vez. Cuando el mismo cliente requiera un análisis para ese producto en particular, el laboratorio ya sabe qué procedimientos hacer sobre ese producto. A eso se le llama la *especificación de un producto*. Por otro lado, a la instancia particular del producto que se somete a los procedimientos, se le llama *muestra* del producto.

Aquí se presenta un diagrama entidad relación ajustado al caso de uso particular del ingreso de muestras.



## Relaciones muchos a muchos como asociaciones binarias

Notemos que los rombos con esquinas triangulares denotan relaciones de muchos a muchos. La técnica general de

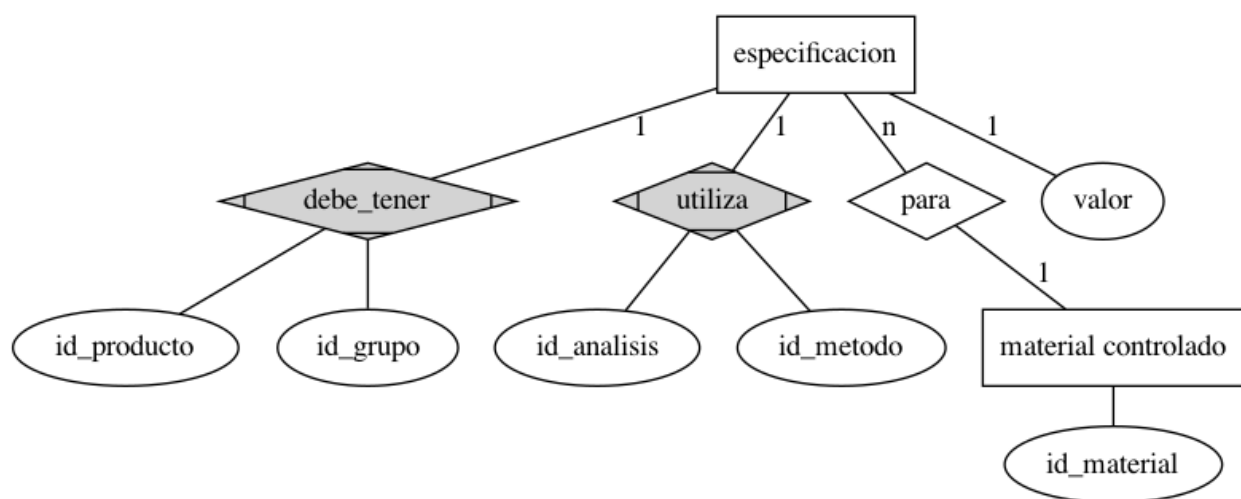
represtar ese tipo de relaciones es usar una tabla de asociacion en el modelo relacional. Por ende, si se revisa el *Dominio.py* se encontraran esas relaciones construidas como tablas. Esas tablas tienen además una llave sustituta (*surrogate key*).

La función de la llave sustita es doble. Por un lado, codifica una 2-tupla a 1-tupla.

Por otro lado, Por las necesidades impuestas en el proceso de ingreso, era deseable restringir las opciones de la persona de ingreso a solo relacionar el producto con ciertos análisis dependiendo de los grupos que tuviera la especificacion. Es decir, la responsabilidad de la persona de ingreso se debe minimizar y, reciprocamente, el sistema debe capturar correctamente los servicios (el catálogo que ofrece el laboratorio). En general, no es posible construir este tipo de relaciones semánticas en relaciones de grado superior. Aún si se pudiese, se considera una «buena práctica» transformar las relaciones de orden superior en relaciones binarias. En general, las tablas presentan mucho menor redundancia y la búsqueda de llave fórneas en la API de una aplicación cliente será menos dispendiosa.

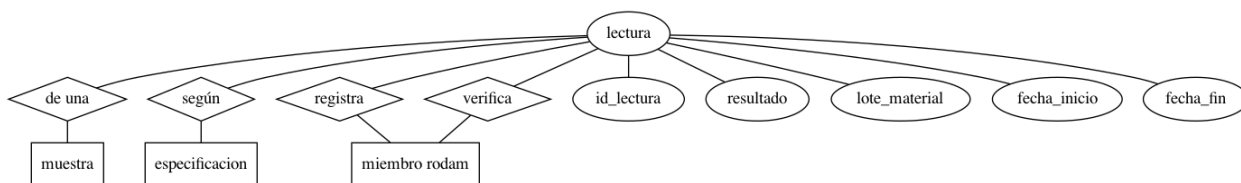
Ciertas herramientas, por ejemplo, Oracle Express o las formas de Microsoft Access solo construyen vistas entre dos tablas y no más.

La especificación, por lo tanto, no depende, de las entidades base (producto, grupo, analisis, metodo), sino de relaciones entre ellos, de la siguiente manera:



Especificación de un producto

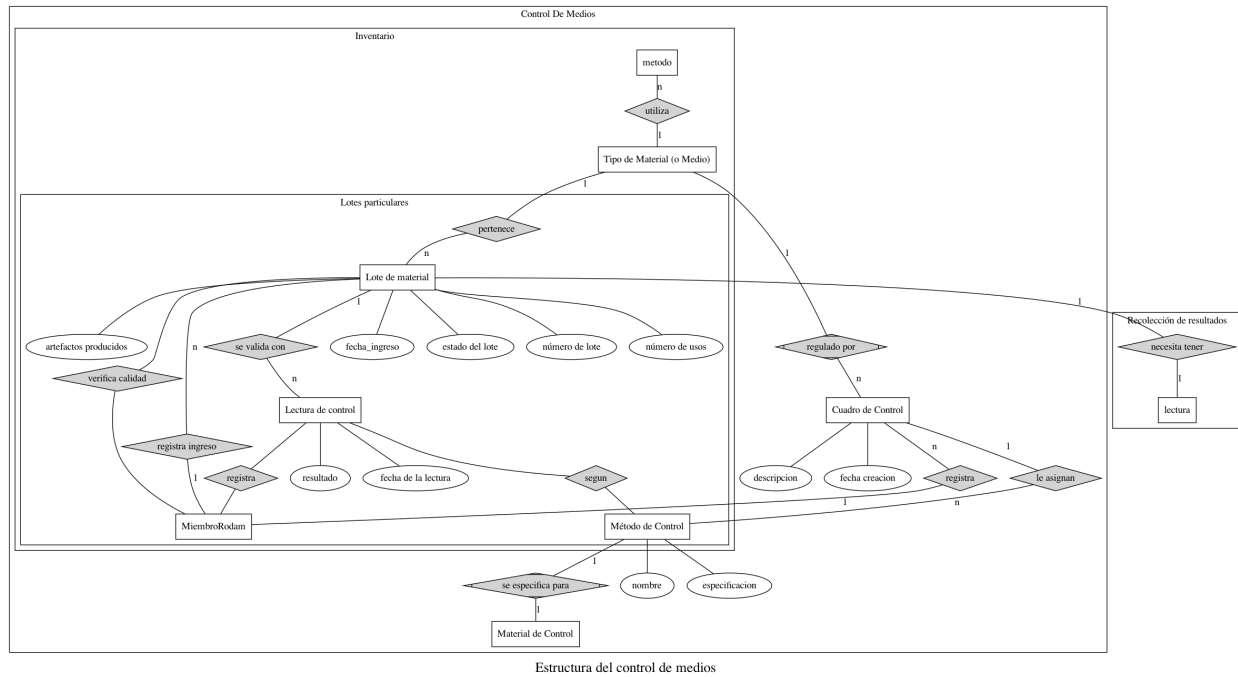
Así, mismo, una lectura se compone por:



Registro de lectura

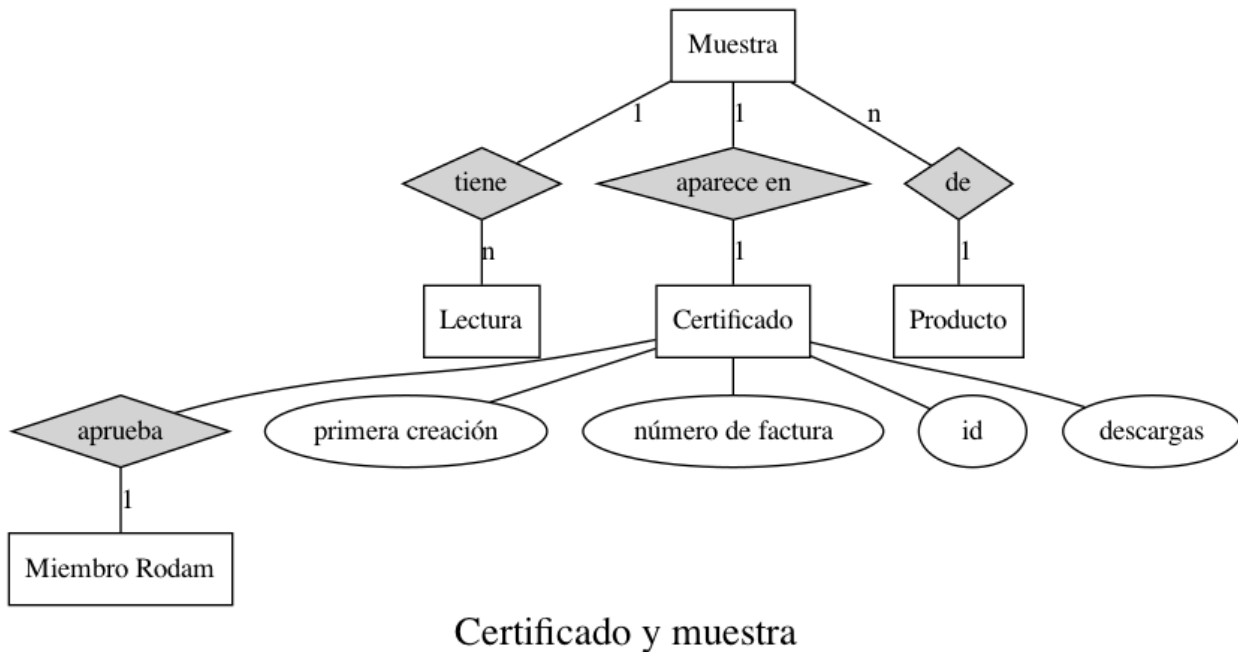


### 3.4 El control de calidad



### 3.5 Emisión de certificados

Aquí se muestra la relación que tiene un certificado con las demás entidades.





**author** Jonatan Ahumada Fernández

**contact** [jaumaf@hotmail.com](mailto:jaumaf@hotmail.com)

**date** último build el 2022-03-30

### 4.1 Introducción

Este documento constituye el plan de pruebas para el sistema de información de Rodam análisis. El propósito de este documento es explicar qué tipo de pruebas se necesitan llevar a cabo y por qué, con base a la estimación de riesgos y las funcionalidades descritas en la *Especificación de Requerimientos*.

### 4.2 Necesidades del tipo de pruebas

Al ser un sistema de categoría GAMP tipo 5 (a la medida), las pruebas realizadas sobre el software son variadas y de distintas naturaleza. Sobre el software se realizarán pruebas a) unitarias b) de integración c) manuales y d) de validación.

### 4.3 Estimación de riesgos

Aquí se hará un resumen sucinto de los riesgos más relevantes del presente sistema.

### 4.3.1 Riesgo alto

- el modelo de datos del sistema es insuficiente y/o genera inconsistencias en los datos
- el artefacto principal, el certificado de emisión, es incompleto
- el artefacto principal, el certificado de emisión, no es flexible a posteriores cambios
- los datos no son accesibles o de difícil interpretación
- los datos en los formularios no son restringidos de forma adecuada y se permiten ingresar datos erróneos o inconsecuentes

### 4.3.2 Riesgo medio

- los requerimientos cambian a medida que se avanza el desarrollo y se obtiene mayor información sobre las necesidades
- cambiar los procesos de negocio que antiguamente se hacían con hojas de cálculo puede dar lugar a confusiones

### 4.3.3 Riesgo bajo

- las componentes utilizados, tanto de software como de hardware, son insuficientes o presentan fallas
- la interfaz gráfica no es inmediatamente comprensible o estéticamente desactualizada

## 4.4 La organización de las pruebas

Como se establece en la *Especificación de Requerimientos*, el software está conceptualmente definido en áreas funcionales o submodulos. Así, la estructura general de las pruebas se organiza según esta estructura y no según el tipo de prueba (unitaria o de integración). Un ejemplo de la organización de las pruebas es:

```
tests
├── artefactos
├── emision
├── formularios
├── procesos
├── reporter
├── services
├── teoremas
├── validaciones
└── vistas
```

## 4.5 Pruebas unitarias

Las pruebas unitarias constituyen la parte más cuantiosa de las pruebas. Su objetivo es asegurar que el comportamiento de una unidad de computación sea el estipulado por los requisitos. Por este motivo, se utilizan pruebas automatizadas. Estas pruebas son en sí mismas piezas de código ejecutable y su salida produce un reporte similar a este:

```
test_se_necesita_tener_firma_y_logo (tests.emision.test_emisorDeCertificado.Test_
↪EmisorDeCertificado) ... ok
```

## 4.6 Pruebas de integracion

El objetivo de las pruebas de integración es asegurar que dos o más módulos computacionales se relacionan entre sí de forma adecuada. Estas pruebas también son automatizadas. Como la codificación de las pruebas de integración es compleja, su número es reducido. Los criterios para realizar una prueba de integración son su efecto en la cadena de valor y el riesgo que conlleva el fallo de los componentes implicados.

Las áreas que más tienen pruebas de integracion son: el área de emisión de certificados, el área de recolección de resultados y el los procesos de validación al ingresar, especificar y realizar lecturas de una muestra.

Por ejemplo, como la emisión de certificado es la culminación del proceso y genera un artefacto importante (el certificado de emisión). Hay pruebas de integración que generan una muestra automáticamente, hacen las lecturas y luego exportan un certificado:

```
test_se_imprime_con_datos_falsos (tests.reporter.test_certificado_de_emision.PDF_
↪Reporter) ... skipped ''
test_se_integra_con_la_muestra (tests.reporter.test_certificado_de_emision.PDF_
↪Reporter) ... ok
```

## 4.7 Pruebas manuales

El objetivo de las pruebas manuales es verificar aspectos del software para los que el desarrollo de pruebas automatizadas consumiría demasiados recursos (tiempo de desarrollo y complejidad del sistema). Por lo general, las pruebas manuales se utilizan para verificar: la navegación por el sitio web, y los opciones desplegables en formularios y la autorización a cierta funcionalidad del sistema dependiendo del usuario.

## 4.8 Validación

El objetivo de las pruebas de validación es verificar que la funcionalidad sea utilizable por parte del usuario final. Cada vez que se finaliza un sprint, la nueva funcionalidad se sube al ambiente de producción. Luego, se notifica de la actualización al dueño del producto. Este recibe un breve resumen de la funcionalidad tomada del sistema de control de cambios. Un ejemplo de esta notificacion es:

```
commit 510a495eb0bf59161765850bcfa116c8ee275e96 (correcciones_daniel_2022_02_11)
Author: Jonathan Ahumada <jaumaf@Jonathans-MacBook-Air.local>
Date: Mon Feb 14 10:42:19 2022 -0500

- RA-219 el cuadro analitico se incluye al momento de exportar el csv de la factura

commit e91134564cf64e9eddf26c5aa79abc013e772ba6
Author: Jonathan Ahumada <jaumaf@Jonathans-MacBook-Air.local>
Date: Mon Feb 14 10:29:00 2022 -0500

- RA-218 el campo de observaciones aparece en el formulario de editar lectura en_
↪sala

commit 740f5deac17c26790ef0081bcc97e84b6e49e294
Author: Jonathan Ahumada <jaumaf@Jonathans-MacBook-Air.local>
Date: Mon Feb 14 09:44:28 2022 -0500

- RA-217 ahora es 'Marcar inicio en laboratorio' en vez de 'Marcar inicio de_
↪lecturas'
```

Luego, el cliente valida (hace una inspección manual de la funcionalidad) sobre el sistema en producción. Si la funcionalidad no cumple con lo esperado, se abre una tarjeta en el sistema de seguimiento de incidencias (Jira).

---

## Prueba de aceptación

---

**author** Jonatan Ahumada Fernández

**contact** [jaumaf@hotmail.com](mailto:jaumaf@hotmail.com)

**date** último build el 2022-03-29

### 5.1 Resumen

Este cuaderno muestra la funcionalidad de la base de datos relacional utilizada para el módulo **Laboratorio** del sistema de información de Rodam Análisis S.A.S. Las funcionalidades se muestran *independientemente* de la interfaz de usuario que luego se implemente sobre el modelo. Idealmente, el modelo deberá cumplir con todos los casos de uso, para luego pasar a la implementación de su interfaz.

Con este cuaderno tanto el *product owner* como el desarrollador podrán evaluar su comprensión mutua sobre el sistema y poner a prueba nuevas ideas.

Este cuaderno es ejecutable en Binder. Lo puedes correr aquí:

### 5.2 Documentos relacionados

- La *Especificación de Requerimientos* detalla los requerimientos y los submódulos del laboratorio.
- El *Documento de Diseño* estudia las relaciones y las entidades del módulo.

## 5.3 Objetivo

Evaluar el alcance y las limitaciones del *backend* del módulo Laboratorio.

Como criterios para evaluar la idoneidad del modelo, propongo estas consideraciones:

- ¿las tablas permiten hacer las consultas que necesito en este momento?
- ¿las tablas aseguran la integridad referencial de mi modelo?
- ¿parece posible extender el sistema con requerimientos que puedan salir a futuro?

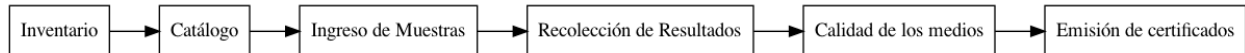
## 5.4 Definiciones

El *backend* está compuesto por dos capas. La primera de ellas es la **capa de la base de datos relacional**. La segunda de ellas es la **capa del ORM**, que mapea el modelo relacional al paradigma de orientación a objetos.

Tanto en el desarrollo como en esta *prueba de aceptación* nos valdremos del **ORM**, por las ventajas que ofrece. Sin embargo, el **modelo relacional** es el cimiento del sistema. Por eso, la exposición vendrá acompañada de consultas directas a una instancia de base de datos `sqlite` llamada `aceptacion.bd`.

### 5.4.1 Repaso por los casos de uso

Para consultar los casos de uso en mayor detalle, consultar la *Especificación de requerimientos* del módulo. El cuaderno seguirá el camino de las **dependencias secuenciales** esbozadas en ese documento. Un breve resumen del flujo de eventos del laboratorio sería:



### 5.4.2 Inicio del flujo

Primero, carguemos el módulo de Python con el dominio:

```
%cd ~/.devs/ # ubicarse un directorio arriba del paquete rodam
from rodam.modelos import Dominio
import rodam.conf as conf
```

```
/Users/jaumaf/.devs
```

## 5.5 Configurando la sesion

```
Dominio.Base.metadata.drop_all(conf.ACEPTACION_ENG)
```

Probemos que no haya datos:

```
!sqlite3 -header -column rodam/aceptacion.db '.schema' # no debe retornar nada
```

```
conf.Base.metadata.create_all(conf.ACEPTACION_ENG)
session = conf.sessionmaker(bind=conf.ACEPTACION_ENG)
```



```
sesion = session()
```

### 5.5.1 Inventario

Los objetos del Inventario son dependencias de los demás. Creemos, entonces, los objetos del inventario:

```
#ciudades

bogota = Dominio.Ciudad(
    id_ciudad = 1,
    nom_ciudad = "Bogotá",
    indicativo = 1
)

medellin = Dominio.Ciudad(
    nom_ciudad = "Medellín",
    indicativo = 4,
    id_ciudad = 2)

# sectores

farmaceutico = Dominio.Sector(
    id_sector = 1,
    nom_sector = "Farmaceutico"
)

cosmetico = Dominio.Sector(
    id_sector = 2,
    nom_sector = "Cosmético")

# origenes

prod_terminado = Dominio.Origen(
    nom_origen = "Producto terminado",
    id_origen = 1
)

muestreo_planta = Dominio.Origen(
    nom_origen = "Muestreo en planta",
    id_origen = 2
)

# empresas

conalcos = Dominio.Empresa(
    id_empresa = 1,
    nit_empresa = 90024,
    nom_empresa = "Compañía Nacional de Cosméticos CONALCOS",
    id_ciudad = bogota.id_ciudad,
    pagina = "conalcos.com.co",
    correo = "contacto@conalco.com.co",
    direccion = "calle 34 Bis 39-33",
)

remo = Dominio.Empresa(
    id_empresa = 2,
    nit_empresa = 90025,
```

(continué en la próxima página)

(proviene de la página anterior)

```
nom_empresa = "Laboratorios REMO S.A.S",
id_ciudad = medellin.id_ciudad,
pagina = "labremo.com.co",
correo = "contacto@labremo.com.co",
direccion = "calle 54 11-22",
)

# clientes
conalcos_cliente = Dominio.Cliente(
id_cliente = 1,
id_empresa = conalcos.id_empresa,
retencion = 0.14,
IVA = 0.15,
ISA = 0.16,
ICA = 0.17,
)

remo_cliente = Dominio.Cliente(
id_cliente = 2,
id_empresa = remo.id_empresa,
retencion = 0.14,
IVA = 0.15,
ISA = 0.16,
ICA = 0.17)

# cargos
jefe_laboratorio = Dominio.Cargo(
    id_cargo = 1,
    nom_cargo= "Jefe de laboratorio",
    observaciones = "No sabría que poner aquí"
)

jefe_calidad = Dominio.Cargo(
    id_cargo = 2,
    nom_cargo= "Jefe de calidad",
    jefe = jefe_laboratorio.id_cargo,
    reemplazo = jefe_laboratorio.id_cargo,
    observaciones = "No sabría que poner aquí"
)

microbiologo = Dominio.Cargo(
id_cargo = 3,
nom_cargo = "Microbiólogo",
jefe = jefe_calidad.id_cargo,
observaciones = "No sabría que poner aquí"
)

# equipos
cabina = Dominio.Equipo(
id_equipo = 1,
nom_equipo = "Cabina de flujo laminar",
ref_documental = "E001"
```

(continué en la próxima página)

(proviene de la página anterior)

```
)

micropipeta = Dominio.Equipo(
id_equipo = 2,
nom_equipo = "Micropipeta",
ref_documental = "E019",
)

balanza = Dominio.Equipo (
id_equipo = 3,
nom_equipo = "Balanza",
ref_documental = "E020",
)

incubadora = Dominio.Equipo(
id_equipo = 4,
    nom_equipo = "Incubadora",
    ref_documental = "E010",
)

caja_petri = Dominio.Equipo(
id_equipo = 5,
    nom_equipo = "Caja de Petri",
    ref_documental = "E011",
)

turbidimetro = Dominio.Equipo(
id_equipo = 6,
    nom_equipo = "Turbidimetro",
    ref_documental = "E012",
)

# Materiales controlados
no_aplica = Dominio.MaterialControlado(
nom_material = "N. A"
)

bacilus = Dominio.MaterialControlado(
nom_material = "Bacillus subtilis spizizenii ATCC 6633",
)

pseudomona = Dominio.MaterialControlado(
nom_material = "Pseudomona aeruginosa ATCC 9027",
)

staph_aureus = Dominio.MaterialControlado(
nom_material = "Staphylococcus Aureus ATCC 6538",
)

candida = Dominio.MaterialControlado(
nom_material = "Candida albicans ATCC 10231",
)

aspergillus = Dominio.MaterialControlado(
nom_material = "Aspergillus bresiliensis ATCC 16404",
```

(continué en la próxima página)

(proviene de la página anterior)

```
)

inventario = [bogota, medellin, cosmetico, farmaceutico,
              prod_terminado, muestreo_planta, conalcos,
              remo, conalcos_cliente, remo_cliente,
              jefe_laboratorio, jefe_calidad,
              cabina, micropipeta, balanza, incubadora,
              caja_petri, turbidimetro, no_aplica, bacilus, pseudomona,
              staph_aureus, candida, aspergillus]
```

Ahora agreguemos los objetos del inventario y veamos un ejemplo de cómo se vería una de sus tablas:

```
sesion.add_all(inventario)
sesion.commit()
```

```
!sqlite3 -header -column rodam/aceptacion.db 'select * from equipo;' # ejemplo de lo
↪que acabamos de ingresar
```

id_equipo	nom_equipo	ref_documental
1	Cabina de flujo laminar	E001
2	Micropipeta	E019
3	Balanza	E020
4	Incubadora	E010
5	Caja de Petri	E011
6	Turbidimetro	E012

## 5.5.2 Creación de Catálogo

Con el inventario listo, ahora realizaremos el proceso de creación del catálogo. El elemento atómico de un análisis de laboratorio es un método. Debemos crear primero los métodos existentes en el laboratorio.

```
# metodos

enriquecimiento = Dominio.Metodo(
    id_metodo = 1,
    nom_metodo = "Enriquecimiento",
    desc_metodo = "5g de producto en 45 de TSB, incubar 24 h a 30-25 grados",
    ref_documental = "IN006",
    observaciones= "",
    material = "Agar TSA" ,
    equipos = [cabina, balanza]
)

siembra_incub = Dominio.Metodo(
    id_metodo = 2,
    nom_metodo = "Siembra e incubación",
    desc_metodo = "Siembra en superficie: 0.1 ml en TSA, incubar 3 dias a 30-35°C",
    ref_documental = "IN007",
    observaciones= "",
    material = "Agar Sabouraud",
```

(continué en la próxima página)

(proviene de la página anterior)

```

    equipos = [micropipeta, incubadora]
)

conteo_petri = Dominio.Metodo(
    id_metodo = 3,
    nom_metodo = "Conteo en caja de Petri",
    desc_metodo = "Recuento en caja de petri",
    ref_documental = "IN009",
    observaciones= "",
    material = "Agar TSB",
    equipos = [balanza, caja_petri]
)

diluc_neutralizacion = Dominio.Metodo(
    id_metodo = 4,
    nom_metodo = "Dilución - neutralización",
    desc_metodo = "8 ml de producto (o solucion del producto) durante 1 minuto",
    ref_documental = "IN010",
    observaciones= "",
    material = "Caldo caseina",
    equipos = [balanza]
)

inoculo = Dominio.Metodo(
    id_metodo= 5,
    nom_metodo = "Inóculo de material controlado",
    ref_documental = "IN011",
    observaciones = "",
    material = "Solución salina",
    equipos = [turbidimetro]
)

enriq_peptona = Dominio.Metodo(
    id_metodo = 8,
    nom_metodo = "Enriquecimiento + Peptona",
    desc_metodo = "5 ml de Buffer Peptona + 45 ml de TSB, incubar 24 h a 30-35 C",
    material = "Agar TSB",
    ref_documental = "IN014"
)

diluc_neut5 = Dominio.Metodo(
    id_metodo = 9,
    nom_metodo = "Dilución - neutralización 5 min",
    desc_metodo = "8 ml de producto (o solucion del producto) durante 5 minutos",
    ref_documental = "IN015",
    observaciones= "",
    material = "Caldo caseina",
    equipos = [balanza]
)

```

(continué en la próxima página)

(proviene de la página anterior)

```
)
diluc_neut15 = Dominio.Metodo(
    id_metodo = 10,
    nom_metodo = "Dilución - neutralización 15 min",
    desc_metodo = "8 ml de producto (o solución del producto) durante 15 minutos",
    ref_documental = "IN016",
    observaciones= "",
    material = "Caldo caseina",
    equipos = [balanza]
)

diluc_neut30 = Dominio.Metodo(
    id_metodo = 11,
    nom_metodo = "Dilución - neutralización 30 min",
    desc_metodo = "8 ml de producto (o solución del producto) durante 30 minutos",
    ref_documental = "IN017",
    observaciones= "",
    material = "Caldo caseina",
    equipos = [balanza]
)

diluc_neut45 = Dominio.Metodo(
    id_metodo = 12,
    nom_metodo = "Dilución - neutralización 45 min",
    desc_metodo = "8 ml de producto (o solución del producto) durante 45 minutos",
    ref_documental = "IN018",
    observaciones= "",
    material = "Caldo caseina",
    equipos = [balanza]
)

metodos = [enriquecimiento, siembra_incub, conteo_petri, diluc_neutralizacion,
    inoculo, enriq_peptona, diluc_neut5, diluc_neut15, diluc_neut30, diluc_
↪neut45]
```

```
sesion.add_all(metodos)
```

```
sesion.commit()
```

```
!sqlite3 -header -markdown rodam/aceptacion.db 'select nom_metodo, material, ref_
↪documental from metodo;' # lo que acabamos de ingresar
```

nom_metodo	material	ref_documental
Enriquecimiento	Agar TSA	IN006
Siembra e incubación	Agar Sabouraud	IN007
Conteo en caja de Petri	Agar TSB	IN009
Dilución - neutralización	Caldo caseina	IN010
Inóculo de material controlado	Solución salina	IN011
Enriquecimiento + Peptona	Agar TSB	IN014
Dilución - neutralización 5 min	Caldo caseina	IN015

(continué en la próxima página)

(proviene de la página anterior)

Dilución - neutralización 15 min   Caldo caseina   IN016
Dilución - neutralización 30 min   Caldo caseina   IN017
Dilución - neutralización 45 min   Caldo caseina   IN018

## 5.6 Creación de análisis

Los analisis tienen muchos métodos. Así que podremos simplemente agregarlos al atributo `metodos` correspondiente:

```
# Analisis

meso_aerob = Dominio.Analisis(
id_analisis = 1,
nom_analisis = "Recuento de Mesofilos Aeorobios",
id_sector = farmaceutico.id_sector,
ref_documental= "PR001",
)

# asociaciones
util1 = Dominio.Utiliza(
id_utiliza = 1,
metodo = enriquecimiento
)

util2 = Dominio.Utiliza(
id_utiliza = 2,
metodo = siembra_incub,
)

util3= Dominio.Utiliza(
id_utiliza = 3,
metodo = conteo_petri)

meso_aerob.metodos= [util1,util2, util3]
```

```
sesion.add(meso_aerob)
```

```
sesion.commit()
```

La asociación entre analisis y metodo queda registrada en la tabla utiliza

```
!sqlite3 -header -markdown rodam/aceptacion.db 'select * from utiliza;' # lo que_
↪acabamos de ingresar
```

id_utiliza	id_analisis	id_metodo	
1	1	1	
2	1	2	
3	1	3	

```
# hongos y levaduras

hongos_lev = Dominio.Analisis(
    id_analisis = 2,
    nom_analisis = "Recuento de Hongos y Levaduras",
    id_sector = farmaceutico.id_sector,
    ref_documental= "PR002",
)

util4 = Dominio.Utiliza(
    id_utiliza = 4,
    metodo = enriquecimiento
)

util5 = Dominio.Utiliza(
    id_utiliza = 5,
    metodo = siembra_incub
)

util6 = Dominio.Utiliza(
    id_utiliza = 6,
    metodo = conteo_petri
)

hongos_lev.metodos = [util4, util5, util6]
```

```
sesion.add(hongos_lev)
sesion.commit()
```

```
# E. Coli

e_coli = Dominio.Analisis(
    id_analisis = 3,
    nom_analisis = "Presencia de E. Choli",
    id_sector = farmaceutico.id_sector,
    ref_documental= "PR003"
)

util7 = Dominio.Utiliza(
    id_utiliza = 7,
    metodo = enriquecimiento
)

util8 = Dominio.Utiliza(
    id_utiliza = 8,
    metodo = siembra_incub
)

util9 = Dominio.Utiliza(
    id_utiliza = 9,
```

(continué en la próxima página)



(proviene de la página anterior)

```
metodo = conteo_petri
)

e_coli.metodos = [util17, util18, util19]
```

```
sesion.add(e_coli)
sesion.commit()
```

```
# Actividad bactericida

a_bactericida = Dominio.Analisis(
nom_analisis = "Actividad Bactericida Básica",
id_sector = cosmetico.id_sector,
ref_documental = "PR004",

)

util10 = Dominio.Utiliza(
id_utiliza = 10,
metodo = enriquecimiento
)

util11 = Dominio.Utiliza(
id_utiliza = 11,
metodo = siembra_incub
)

util12 = Dominio.Utiliza(
id_utiliza = 12,
metodo = inculo
)

util15 = Dominio.Utiliza(
id_utiliza = 15,
metodo = siembra_incub
)

util16 = Dominio.Utiliza(
id_utiliza = 16,
metodo = conteo_petri
)

util17 = Dominio.Utiliza(
id_utiliza = 17,
metodo = enriq_peptona)

util18 = Dominio.Utiliza(
id_utiliza = 18,
metodo = diluc_neut5)

util20 = Dominio.Utiliza(
```

(continué en la próxima página)

(proviene de la página anterior)

```
id_utiliza = 20,
metodo = diluc_neut15)

util20 = Dominio.Utiliza(
id_utiliza = 21,
metodo = diluc_neut30)

util21 = Dominio.Utiliza(
id_utiliza = 22,
metodo = diluc_neut45)

util22 = Dominio.Utiliza(
id_utiliza = 23,
metodo = diluc_neutralizacion)

a_bactericida.metodos = [util110, util111, util112, util115, util116,
                        util117, util118, util20, util21, util22]
```

```
sesion.add(a_bactericida)
```

```
sesion.commit()
```

Revisemos la relación utiliza con la siguiente consulta:

```
select utiliza.id_utiliza, analisis.nom_analisis, analisis.ref_documental, metodo.
nom_método, metodo.ref_documental from utiliza JOIN analisis on utiliza.id_analisis =
analisis.id_analisis JOIN metodo on utiliza.id_método = metodo.id_método;
```

```
!sqlite3 -header -markdown rodam/aceptacion.db 'select utiliza.id_utiliza, analisis.
nom_analisis, analisis.ref_documental, metodo.nom_método, metodo.ref_documental
from utiliza JOIN analisis on utiliza.id_analisis = analisis.id_analisis JOIN
metodo on utiliza.id_método = metodo.id_método;'
```

id_utiliza	nom_analisis	ref_documental	nom_método
1	Recuento de Mesofilos Aeorobios	PR001	Enriquecimiento
2	Recuento de Mesofilos Aeorobios	PR001	Siembra e
3	Recuento de Mesofilos Aeorobios	PR001	Conteo en caja de
4	Recuento de Hongos y Levaduras	PR002	Enriquecimiento
5	Recuento de Hongos y Levaduras	PR002	Siembra e
6	Recuento de Hongos y Levaduras	PR002	Conteo en caja de
7	Presencia de E. Choli	PR003	Enriquecimiento
8	Presencia de E. Choli	PR003	Siembra e
9	Presencia de E. Choli	PR003	Conteo en caja de

(continúe en la próxima página)

(proviene de la página anterior)

10	Actividad Bactericida Básica	PR004	Enriquecimiento	└
↪	IN006			
11	Actividad Bactericida Básica	PR004	Siembra e	└
↪incubación	IN007			
12	Actividad Bactericida Básica	PR004	Inóculo de material	└
↪controlado	IN011			
15	Actividad Bactericida Básica	PR004	Siembra e	└
↪incubación	IN007			
16	Actividad Bactericida Básica	PR004	Conteo en caja de	└
↪Petri	IN009			
17	Actividad Bactericida Básica	PR004	Enriquecimiento +	└
↪Peptona	IN014			
18	Actividad Bactericida Básica	PR004	Dilución -	└
↪neutralización 5 min	IN015			
21	Actividad Bactericida Básica	PR004	Dilución -	└
↪neutralización 30 min	IN017			
22	Actividad Bactericida Básica	PR004	Dilución -	└
↪neutralización 45 min	IN018			
23	Actividad Bactericida Básica	PR004	Dilución -	└
↪neutralización	IN010			

## 5.7 Creación de Grupos

Los análisis pueden pertenecer a varios grupos.

```

grupo_unico = Dominio.Grupo(
nom_grupo = "Grupo único",
desc_grupo = "Para análisis simples que no requieren grupos")

recuento_inoc= Dominio.Grupo(
nom_grupo = "Recuento del inóculo",
desc_grupo = "Recuento del inóculo para actividad bactericida básica")

toxicidad = Dominio.Grupo(
nom_grupo = "Control de toxicidad del neutralizante",
desc_grupo = "Pseudomonas Aeruginosa y staphylococcus Aureus",
)

control_neutralizacion = Dominio.Grupo(
nom_grupo = "Control del método de dilución-neutralización",
)

prueba_dilucion = Dominio.Grupo(
nom_grupo = "Prueba de dilución - neutralización"
)

# estos analisis quedan asociados al grupo único

agrupal = Dominio.Agrupacion(
id_agrupacion = 1,
analisis = meso_aerob
)

```

(continué en la próxima página)

(proviene de la página anterior)

```
agrupa2 = Dominio.Agrupacion(
id_agrupacion = 2,
analisis = hongos_lev
)

agrupa3 = Dominio.Agrupacion(
id_agrupacion = 3,
analisis = e_coli
)

grupo_unico.analisis = [agrupa1, agrupa2, agrupa3]
```

```
sesion.add(grupo_unico)
sesion.commit()
```

```
!sqlite3 -header -markdown rodam/aceptacion.db 'select * from agrupacion'
```

id_agrupacion	id_grupo	id_analisis
1	1	1
2	1	2
3	1	3

```
# actividad bactericida tiene estos grupos == estos grupos tienen actividad_
↔bactericida

agrupa4 = Dominio.Agrupacion(
id_agrupacion = 4,
analisis = a_bactericida
)

agrupa5 = Dominio.Agrupacion(
id_agrupacion = 5,
analisis = a_bactericida
)

agrupa6 = Dominio.Agrupacion(
id_agrupacion = 6,
analisis = a_bactericida)

agrupa7 = Dominio.Agrupacion(
id_agrupacion = 7,
analisis = a_bactericida
)

toxicidad.analisis.append(agrupa4)
recuento_inoc.analisis.append(agrupa5)
control_neutralizacion.analisis.append(agrupa6)
prueba_dilucion.analisis.append(agrupa7)
```

```
sesion.add_all([toxicidad, recuento_inoc,prueba_dilucion])
```

```
sesion.commit()
```

```
!sqlite3 -header -markdown rodam/acceptacion.db 'select grupo.nom_grupo, analisis.nom_
↪ analisis from agrupacion join grupo on agrupacion.id_grupo = grupo.id_grupo join_
↪ analisis on agrupacion.id_analisis = analisis.id_analisis'
```

nom_grupo	nom_analisis
Grupo único	Recuento de Mesofilos Aerobios
Grupo único	Recuento de Hongos y Levaduras
Grupo único	Presencia de E. Choli
Control de toxicidad del neutralizante	Actividad Bactericida Básica
Recuento del inóculo	Actividad Bactericida Básica
Prueba de dilución - neutralización	Actividad Bactericida Básica

### 5.7.1 Ingreso de Muestras

Una vez sabemos qué servicios están en nuestro catálogo, podemos empezar a ingresar productos y a realizar análisis sobre esas muestras.

## 5.8 Ingreso de un producto

En **Rodam Análisis S.A.S**, un producto se ingresa una sola vez. A partir de allí, el producto queda enlazado con una serie de especificaciones que indican qué grupos, análisis y métodos deben correrse sobre una muestra de ese producto.

Luego del ingreso del producto, cada **muestra** de ese producto siempre tendrá las especificaciones estipuladas para el producto.

Primero empecemos con la creación de un producto:

```
# producto 1
xinefax = Dominio.Producto(
id_producto= 1,
nom_producto = "Xinefax" ,          # libre
forma_farmaceutica = "Emulsión",    # libre
id_sector = farmaceutico.id_sector, # dependencia
id_cliente = conalcos_cliente.id_cliente
)
sesion.add(xinefax)
sesion.commit()

# producto 2

seximax = Dominio.Producto(
id_producto = 2,
nom_producto = "Seximax",
forma_farmaceutica = "Loción",
id_sector = cosmetico.id_sector,
id_cliente = remo_cliente.id_cliente
)

sesion.add(seximax)
sesion.commit()
```

Ahora debemos indicar qué grupos tendrá el producto.

```
# El producto debe tener grupos ya existentes en el catálogo

debe1 = Dominio.Debe_tener(
id_debe_tener = 1,
grupo = recuento_inoc
)

debe2 = Dominio.Debe_tener(
id_debe_tener = 2,
grupo = toxicidad
)

debe3 = Dominio.Debe_tener(
id_debe_tener = 3,
grupo = control_neutralizacion
)

debe4 = Dominio.Debe_tener(
id_debe_tener = 4,
grupo = prueba_dilucion
)

xinefax.grupos = [debe1, debe2, debe3, debe4]

debe5 = Dominio.Debe_tener(
id_debe_tener = 5,
grupo = grupo_unico
)

seximax.grupos = [debe5]
```

```
sesion.commit()
```

```
!sqlite3 -header -markdown rodam/acceptacion.db 'select * from debe_tener;'
```

id_debe_tener	id_producto	id_grupo
1	1	3
2	1	2
3	1	5
4	1	4
5	2	1

## 5.9 Especificar un producto

Sabemos que *xinefax debe tener* los grupos estipulados en el resultado anterior. Pero aún no hemos especificado cuáles de los análisis disponibles en esos grupos se tienen que correr.

Sin embargo, podemos saber qué análisis están disponibles dentro de qué grupo (esto puede ser útil, en una interfaz de usuario):

```
!sqlite3 -header -markdown rodam/aceptacion.db "select agrupacion.id_agrupacion,
↳ grupo.nom_grupo, grupo.id_grupo, analisis.nom_analisis, analisis.id_analisis from
↳ agrupacion join grupo on agrupacion.id_grupo = grupo.id_grupo join analisis on
↳ agrupacion.id_analisis = analisis.id_analisis"
```

id_agrupacion	nom_grupo	id_grupo	
↳ nom_analisis	id_analisis		
-----	-----	-----	-----
1	Grupo único	1	Recuento
↳ de Mesofilos Aeorobios	1		
2	Grupo único	1	Recuento
↳ de Hongos y Levaduras	2		
3	Grupo único	1	
↳ Presencia de E. Choli	3		
4	Control de toxicidad <b>del</b> neutralizante	2	
↳ Actividad Bactericida Básica	4		
5	Recuento <b>del</b> inóculo	3	
↳ Actividad Bactericida Básica	4		
7	Prueba de dilución - neutralización	4	
↳ Actividad Bactericida Básica	4		
6	Control <b>del</b> método de dilución-neutralización	5	
↳ Actividad Bactericida Básica	4		

Tambien necesitaremos especificar qué metodos para qué analisis. Entonces debemos buscar:

```
!sqlite3 -header -markdown rodam/aceptacion.db "select id_utiliza, id_analisis,
↳ metodo.nom_metodo, utiliza.id_metodo, metodo.ref_documental from utiliza join
↳ metodo on utiliza.id_metodo = metodo.id_metodo"
```

id_utiliza	id_analisis	nom_metodo	id_metodo	ref_
↳ documental				
-----	-----	-----	-----	-----
1	1	Enriquecimiento	1	IN006
↳				
2	1	Siembra e incubación	2	IN007
↳				
3	1	Conteo en caja de Petri	3	IN009
↳				
4	2	Enriquecimiento	1	IN006
↳				
5	2	Siembra e incubación	2	IN007
↳				
6	2	Conteo en caja de Petri	3	IN009
↳				
7	3	Enriquecimiento	1	IN006
↳				
8	3	Siembra e incubación	2	IN007
↳				

(continué en la próxima página)

(proviene de la página anterior)

9	3	Conteo en caja de Petri	3	IN009	
↩					
10	4	Enriquecimiento	1	IN006	
↩					
11	4	Siembra e incubación	2	IN007	
↩					
12	4	Inóculo de material controlado	5	IN011	
↩					
15	4	Siembra e incubación	2	IN007	
↩					
16	4	Conteo en caja de Petri	3	IN009	
↩					
17	4	Enriquecimiento + Peptona	8	IN014	
↩					
18	4	Dilución - neutralización 5 min	9	IN015	
↩					
21	4	Dilución - neutralización 30 min	11	IN017	
↩					
22	4	Dilución - neutralización 45 min	12	IN018	
↩					
23	4	Dilución - neutralización	4	IN010	
↩					

Para este producto, hemos decidido que

Para el *recuento del inóculo* queremos: - inóculo \**Stafilococcus*,

- inóculo de *Pseudomona*.
- siembra e incubación para cada uno
- conteo en caja de petri para cada uno

Para *Control de toxicidad de nautralizante*:

- queremos nuestro enriquecimiento + peptona
- inóculo de *Stafilococcus*
- Inóculo de *pseudomona*
- siembra e incubación para cada uno
- conteo en caja de petri para cada uno

Para *Control del método de dilución - neutralización* queremos: - nuestro enriquecimiento común

- nuestro inóculos para *Pseudomona* y *Stafilococcus*
- siembra e incubación para cada uno
- recuento en caja de petri para cada uno

Para la *prueba de dilución* queremos correr todos los metodos de dilución :

- IN015- IN018 para cada cepa
- nuestro enriquecimiento TSB
- siembra incubación para cada uno
- Conteo en caja de Petri para cada uno



```

# Creamos las especificaciones

## Recuento

# stafiloccocus e
recuento_inoc1 = Dominio.Especificacion(
id_debe_tener = 1,
id_utiliza = 12,
id_material_controlado = 4 ,
valor = "170<t<230 "
)

# Pseudomona en recuento
recuento_inoc2 = Dominio.Especificacion(
id_debe_tener = 1,
id_utiliza = 12,
id_material_controlado = 3,
valor = "170<t<230 "
)

# Siembra
recuento_inoc3 = Dominio.Especificacion(
id_debe_tener = 1,
id_utiliza = 15,
id_material_controlado = 4,
valor = "<200 ufc"
)

recuento_inoc4 = Dominio.Especificacion(
id_debe_tener = 1,
id_utiliza = 15,
id_material_controlado = 3,
valor = "<200 ufc"
)

# petri

recuento_inoc5 = Dominio.Especificacion(
id_debe_tener = 1,
id_utiliza = 16,
id_material_controlado = 4,
valor = "<200 ufc"
)

recuento_inoc6 = Dominio.Especificacion(
id_debe_tener = 1,
id_utiliza = 16,
id_material_controlado = 3,
valor = "<200 ufc"
)

recuento = [recuento_inoc1, recuento_inoc2, recuento_inoc3, recuento_inoc4, recuento_
↪ inoc5, recuento_inoc6]

## Control de toxicidad

```

(continué en la próxima página)

(proviene de la página anterior)

```
# peptona
control_toxicidad1 = Dominio.Especificacion(
id_debe_tener= 2,
id_utiliza = 17,
valor = "<100 ufc",
id_material_controlado = 4,
)

control_toxicidad1_2 = Dominio.Especificacion(
id_debe_tener= 2,
id_utiliza = 17,
valor = "<100 ufc",
id_material_controlado = 3,
)

# staph
control_toxicidad2 = Dominio.Especificacion(
id_debe_tener= 2,
id_utiliza = 14,
id_material_controlado = 4,
valor = "170<t<230"
)

# pseudo
control_toxicidad3 = Dominio.Especificacion(
id_debe_tener= 2,
id_utiliza = 13,
id_material_controlado = 3,
valor = "170<t<230"
)

# siembra
control_toxicidad4 = Dominio.Especificacion(
id_debe_tener= 2,
id_utiliza = 15,
id_material_controlado = 4,
valor = "<100"
)

control_toxicidad4_2 = Dominio.Especificacion(
id_debe_tener= 2,
id_utiliza = 15,
id_material_controlado = 3,
valor = "<100"
)

# petri
control_toxicidad5 = Dominio.Especificacion(
id_debe_tener= 2,
id_utiliza = 16,
valor = "<130 ufc",
id_material_controlado = 4,
)

control_toxicidad5_2 = Dominio.Especificacion(
id_debe_tener= 2,
id_utiliza = 16,
valor = "<130 ufc",
```

(continué en la próxima página)

(proviene de la página anterior)

```
id_material_controlado = 3,
)

control_toxicidad = [control_toxicidad1,
                     control_toxicidad1_2,
                     control_toxicidad2,
                     control_toxicidad3,
                     control_toxicidad4,
                     control_toxicidad4_2,
                     control_toxicidad5,
                     control_toxicidad5_2]

## Prueba dilucion

prueba_dilucion1 = Dominio.Especificacion(
id_debe_tener = 4,
id_utiliza = 18,
valor = "tiempo de exposicion"
)
prueba_dilucion2 = Dominio.Especificacion(
id_debe_tener = 4,
id_utiliza = 21,
valor = "tiempo de exposicion"
)
prueba_dilucion3 = Dominio.Especificacion(
id_debe_tener = 4,
id_utiliza = 22,
valor = "tiempo de exposicion"
)

prueba_dilucion = [prueba_dilucion1,
                   prueba_dilucion2,
                   prueba_dilucion3]

## Control de dilucion

# enriq nomral
control_dilucion1 = Dominio.Especificacion(
id_debe_tener = 3,
id_utiliza = 10,
id_material_controlado = 4,
valor = "<150 ufc"
)

control_dilucion1_2 = Dominio.Especificacion(
id_debe_tener = 3,
id_utiliza = 10,
id_material_controlado = 3,
valor = "<150 ufc"
)

#staph
```

(continué en la próxima página)

(proviene de la página anterior)

```
control_dilucion2 = Dominio.Especificacion(  
id_debe_tener = 3,  
id_utiliza = 13,  
id_material_controlado = 4,  
valor = "170<t<230")  
  
# pseudo  
control_dilucion3 = Dominio.Especificacion(  
id_debe_tener = 3,  
id_utiliza = 14,  
id_material_controlado = 3,  
valor = "170<t<230"  
)  
  
# siembra  
control_dilucion4 = Dominio.Especificacion(  
id_debe_tener = 3,  
id_utiliza = 15,  
id_material_controlado = 4,  
valor = "< 120 ufc"  
)  
  
control_dilucion4_2 = Dominio.Especificacion(  
id_debe_tener = 3,  
id_utiliza = 15,  
id_material_controlado = 3,  
valor = "< 120 ufc"  
)  
  
# petri  
control_dilucion5 = Dominio.Especificacion(  
id_debe_tener = 3,  
id_utiliza = 16,  
id_material_controlado = 4,  
valor = "<230 ufc"  
)  
  
control_dilucion5_2 = Dominio.Especificacion(  
id_debe_tener = 3,  
id_utiliza = 16,  
id_material_controlado = 3,  
valor = "<230 ufc"  
)  
  
control_dilucion = [control_dilucion1,  
                    control_dilucion1_2,  
                    control_dilucion2,  
                    control_dilucion3,  
                    control_dilucion4,  
                    control_dilucion4_2,  
                    control_dilucion5,  
                    control_dilucion5_2]
```

```
# agregamos cada grupo de especificaciones
```

(continúe en la próxima página)

(proviene de la página anterior)

```
sesion.add_all(recuento)
sesion.add_all(control_toxicidad)
sesion.add_all(prueba_dilucion)
sesion.add_all(control_dilucion)
```

```
sesion.commit()
```

```
espec_2_1 = Dominio.Especificacion(
id_debe_tener = debe5.id_debe_tener,
id_utiliza = 1,      # Aquí no aplica el material controlado porque
valor = "< 140 ufc"  # mesofilos son muchos. Se puede dejar en blanco
)

espec_2_2 = Dominio.Especificacion(
id_debe_tener = debe5.id_debe_tener,
id_utiliza = 2,
valor = "< 145 ufc"
)

espec_2_3 = Dominio.Especificacion(
id_debe_tener = debe5.id_debe_tener,
id_utiliza = 3,
valor = "< 150 ufc"
)
```

```
sesion.add_all([espec_2_1, espec_2_2, espec_2_3])
```

```
sesion.commit()
```

Esta vista resume los datos relevantes para especificaciones:

```
!sqlite3 -header -markdown rodam/aceptacion.db 'select * from especificaciones_con_
↪nombres'
```

nom_producto	nom_grupo	nom_metodo	material	nom_
↪ analisis				
↪ material_controlado		valor		
-----				
↪				
↪				
↪				
↪ Xinefax	Recuento	del inóculo		Actividad
↪ Bactericida Básica		Inóculo de material controlado	Solución salina	
↪ Staphylococcus Aureus ATCC 6538		170<t<230		
↪ Xinefax	Recuento	del inóculo		Actividad
↪ Bactericida Básica		Inóculo de material controlado	Solución salina	
↪ Pseudomona aeruginosa ATCC 9027		170<t<230		
↪ Xinefax	Recuento	del inóculo		Actividad
↪ Bactericida Básica		Siembra e incubación	Agar Sabouraud	
↪ Staphylococcus Aureus ATCC 6538		<200 ufc		
↪ Xinefax	Recuento	del inóculo		Actividad
↪ Bactericida Básica		Siembra e incubación	Agar Sabouraud	
↪ Pseudomona aeruginosa ATCC 9027		<200 ufc		
↪ Xinefax	Recuento	del inóculo		Actividad
↪ Bactericida Básica		Conteo en caja de Petri	Agar TSB	
↪ Staphylococcus Aureus ATCC 6538		<200 ufc		

(continué en la próxima página)

(proviene de la página anterior)

Xinefax	Recuento <del>del</del> inóculo	Actividad	└
↪Bactericida Básica	Conteo en caja de Petri	Agar TSB	└
↪Pseudomona aeruginosa ATCC 9027	<200 ufc		
Xinefax	Control de toxicidad <del>del</del> neutralizante	Actividad	└
↪Bactericida Básica	Enriquecimiento + Peptona	Agar TSB	└
↪Staphylococcus Aureus ATCC 6538	<100 ufc		
Xinefax	Control de toxicidad <del>del</del> neutralizante	Actividad	└
↪Bactericida Básica	Enriquecimiento + Peptona	Agar TSB	└
↪Pseudomona aeruginosa ATCC 9027	<100 ufc		
Xinefax	Control de toxicidad <del>del</del> neutralizante	Actividad	└
↪Bactericida Básica	Siembra e incubación	Agar Sabouraud	└
↪Staphylococcus Aureus ATCC 6538	<100		
Xinefax	Control de toxicidad <del>del</del> neutralizante	Actividad	└
↪Bactericida Básica	Siembra e incubación	Agar Sabouraud	└
↪Pseudomona aeruginosa ATCC 9027	<100		
Xinefax	Control de toxicidad <del>del</del> neutralizante	Actividad	└
↪Bactericida Básica	Conteo en caja de Petri	Agar TSB	└
↪Staphylococcus Aureus ATCC 6538	<130 ufc		
Xinefax	Control de toxicidad <del>del</del> neutralizante	Actividad	└
↪Bactericida Básica	Conteo en caja de Petri	Agar TSB	└
↪Pseudomona aeruginosa ATCC 9027	<130 ufc		
Xinefax	Prueba de dilución - neutralización	Actividad	└
↪Bactericida Básica	Dilución - neutralización 5 min	Caldo caseina	N. A └
↪	tiempo de exposicion		
Xinefax	Prueba de dilución - neutralización	Actividad	└
↪Bactericida Básica	Dilución - neutralización 30 min	Caldo caseina	N. A └
↪	tiempo de exposicion		
Xinefax	Prueba de dilución - neutralización	Actividad	└
↪Bactericida Básica	Dilución - neutralización 45 min	Caldo caseina	N. A └
↪	tiempo de exposicion		
Xinefax	Control <del>del</del> método de dilución-neutralización	Actividad	└
↪Bactericida Básica	Enriquecimiento	Agar TSA	└
↪Staphylococcus Aureus ATCC 6538	<150 ufc		
Xinefax	Control <del>del</del> método de dilución-neutralización	Actividad	└
↪Bactericida Básica	Enriquecimiento	Agar TSA	└
↪Pseudomona aeruginosa ATCC 9027	<150 ufc		
Xinefax	Control <del>del</del> método de dilución-neutralización	Actividad	└
↪Bactericida Básica	Siembra e incubación	Agar Sabouraud	└
↪Staphylococcus Aureus ATCC 6538	< 120 ufc		
Xinefax	Control <del>del</del> método de dilución-neutralización	Actividad	└
↪Bactericida Básica	Siembra e incubación	Agar Sabouraud	└
↪Pseudomona aeruginosa ATCC 9027	< 120 ufc		
Xinefax	Control <del>del</del> método de dilución-neutralización	Actividad	└
↪Bactericida Básica	Conteo en caja de Petri	Agar TSB	└
↪Staphylococcus Aureus ATCC 6538	<230 ufc		
Xinefax	Control <del>del</del> método de dilución-neutralización	Actividad	└
↪Bactericida Básica	Conteo en caja de Petri	Agar TSB	└
↪Pseudomona aeruginosa ATCC 9027	<230 ufc		
Seximax	Grupo único	Recuento de	└
↪Mesofilos Aeorobios	Enriquecimiento	Agar TSA	N. A └
↪	< 140 ufc		
Seximax	Grupo único	Recuento de	└
↪Mesofilos Aeorobios	Siembra e incubación	Agar Sabouraud	N. A └
↪	< 145 ufc		
Seximax	Grupo único	Recuento de	└
↪Mesofilos Aeorobios	Conteo en caja de Petri	Agar TSB	N. A └
↪	< 150 ufc		

## 5.10 Ingreso de una muestra

Una vez ingresado el xinefax, con sus respectivas especificaciones, se puede ahora ingresar una *muestra* de ese producto. Un producto podra tener muchas muestras, pero como las especificaciones están asociadas al producto, no se tendrá que repetir el proceso de especificación.

```
# Creo que en el proceso de laboratorio alguien es responsable del ingreso de la
↪muestra
# Voy a inventar a alguien por el momento que quede asociado al ingreso de muestra.

ernesto_segura = Dominio.MiembroRodam(
id_miembro = 3,
nom_miembro = "Ernesto Segura",
id_cargo = jefe_calidad.id_cargo)

muestra = Dominio.Muestra(
id_muestra = 1,
id_producto = xinefax.id_producto,
id_origen = muestreo_planta.id_origen,
presentacion = 'tabletas',
lote_muestra = "12345",
tamano_muestra= 500,
unidades_tamano = "mg",
registra = ernesto_segura.id_miembro

)
```

```
sesion.add(ernesto_segura)
sesion.add(muestra)
sesion.commit()
```

```
!sqlite3 -header -column rodam/aceptacion.db 'select id_muestra as id, id_producto as
↪prod, id_origen as orig, presentacion as pres, lote_muestra as lote, tamano_muestra
↪as tamaño, ingreso_muestra as ingreso, registra from muestra;'
```

id	prod	orig	pres	lote	tamaño	ingreso	registra
1	1	2	tabletas	12345	500	2021-02-02 16:42:05.599979	3

## 5.11 Ingreso de lectura

Viendo las especificaciones del xinefax, el microbiólogo ha corrido cuidadosamente cada uno de los métodos de laboratorio asociados a ese producto. Es hora, entonces, de crear una lectura. Para ingresar una lectura, necesitamos ser un MiembroRodam. Primero creemoslo.

```
yuli_largo = Dominio.MiembroRodam(
id_miembro = 1,
nom_miembro = "Yuli Largo",
id_cargo = microbiologo.id_cargo
)

daniel_arias = Dominio.MiembroRodam(
```

(continué en la próxima página)

(proviene de la página anterior)

```
id_miembro= 2,  
nom_miembro = "Daniel Arias",  
id_cargo = jefe_laboratorio.id_cargo )
```

```
sesion.add_all([yuli_largo, daniel_arias])  
sesion.commit()
```

```
!sqlite3 -header -column rodam/aceptacion.db 'select * from miembro_rodam;'
```

id_miembro	nom_miembro	id_cargo	ingreso	salida
1	Yuli Largo	3		
2	Daniel Arias	1		
3	Ernesto Segura	2		

```
# cuántas especificaciones hay?
```

```
!sqlite3 rodam/aceptacion.db 'select count(*) from especificacion'
```

```
28
```

Vemos que la remisión del producto es bastante extensa. Para este ejemplo, solo recolectaremos la lectura de *algunas* de las especificaciones.

```
lectura1 = Dominio.Lectura(  
id_muestra = muestra.id_muestra,  
id_especificacion = 1,  
valor_lectura = "112 ufc",  
lote_medio = "M314-20",  
registra = yuli_largo.id_miembro,  
verifica = daniel_arias.id_miembro,  
concepto = False  
)  
  
lectura2 = Dominio.Lectura(  
id_muestra = muestra.id_muestra,  
id_especificacion = 2,  
valor_lectura = "211 ufc",  
registra = yuli_largo.id_miembro,  
verifica = daniel_arias.id_miembro,  
lote_medio = "M315-20",  
concepto = False  
)  
  
lectura3 = Dominio.Lectura(  
id_muestra = muestra.id_muestra,  
id_especificacion = 3,  
valor_lectura = "211 ufc",  
registra = yuli_largo.id_miembro,  
verifica = daniel_arias.id_miembro,  
lote_medio = "M316-20",  
concepto = False  
)  
  
lectura4 = Dominio.Lectura(  

```

(continué en la próxima página)



(proviene de la página anterior)

```

id_muestra = muestra.id_muestra,
id_especificacion = 4,
valor_lectura = "211 ufc",
registra = yuli_largo.id_miembro,
verifica = daniel_arias.id_miembro,
lote_medio = "M316-20",
concepto = False
)
lectura5 = Dominio.Lectura(
id_muestra = muestra.id_muestra,
id_especificacion = 5,
valor_lectura = "211 ufc",
registra = yuli_largo.id_miembro,
verifica = daniel_arias.id_miembro,
lote_medio = "M316-20",
concepto = False)

lectura6 = Dominio.Lectura(
id_muestra = muestra.id_muestra,
id_especificacion = 6,
valor_lectura = "211 ufc",
registra = yuli_largo.id_miembro,
verifica = daniel_arias.id_miembro,
lote_medio = "M316-20",
concepto = False)

lectura7 = Dominio.Lectura(
id_muestra = muestra.id_muestra,
id_especificacion = 7 ,
valor_lectura = "211 ufc",
registra = yuli_largo.id_miembro,
verifica = daniel_arias.id_miembro,
lote_medio = "M316-20",
concepto = False)

# Para control de los medios
lectura18 = Dominio.Lectura(
id_muestra = muestra.id_muestra,
id_especificacion = 18,
valor_lectura = "211 ufc",
lote_medio = "M341-22",
registra = yuli_largo.id_miembro,
verifica = daniel_arias.id_miembro,
concepto = True
)

lecturas = [lectura1, lectura2, lectura3, lectura4, lectura5, lectura6, lectura7, ↵
↵lectura18]

```

```
sesion.add_all(lecturas)
```

```
sesion.commit()
```

```
!sqlite3 -header -column rodam/aceptacion.db 'select * from lectura;'
```

id_lectura	id_especificacion	id_muestra	valor_lectura	fecha_lectura	
↳ lote_medio	registra	verifica	concepto	observaciones	
↳ -----	-----	-----	-----	-----	↳
1	1	1	112 ufc	2021-02-02 16:42:06.181797	↳
↳ M314-20	1	2	0		
2	2	1	211 ufc	2021-02-02 16:42:06.183285	↳
↳ M315-20	1	2	0		
3	3	1	211 ufc	2021-02-02 16:42:06.183608	↳
↳ M316-20	1	2	0		
4	4	1	211 ufc	2021-02-02 16:42:06.183816	↳
↳ M316-20	1	2	0		
5	5	1	211 ufc	2021-02-02 16:42:06.184012	↳
↳ M316-20	1	2	0		
6	6	1	211 ufc	2021-02-02 16:42:06.184231	↳
↳ M316-20	1	2	0		
7	7	1	211 ufc	2021-02-02 16:42:06.184489	↳
↳ M316-20	1	2	0		
8	18	1	211 ufc	2021-02-02 16:42:06.184865	↳
↳ M341-22	1	2	1		

Unamos las lecturas con las especificaciones para obtener un resumen del analisis

```
!sqlite3 -header -markdown rodam/aceptacion.db "select * from especificacion,
↳ lectura where lectura.id_especificacion = especificacion.id_especificacion"
```

id_especificacion	id_debe_tener	id_utiliza	id_material_controlado	valor	
↳ id_lectura	id_especificacion	id_muestra	valor_lectura	fecha_	
↳ lectura	lote_medio	registra	verifica	concepto	observaciones
↳ -----	-----	-----	-----	-----	↳
↳ -----	-----	-----	-----	-----	↳
1	1	12	4	170<t<230	↳
↳ 1	1	1	112 ufc	2021-02-02	↳
↳ 16:42:06.181797	M314-20	1	2	0	
2	1	12	3	170<t<230	↳
↳ 2	2	1	211 ufc	2021-02-02	↳
↳ 16:42:06.183285	M315-20	1	2	0	
3	1	15	4	<200 ufc	↳
↳ 3	3	1	211 ufc	2021-02-02	↳
↳ 16:42:06.183608	M316-20	1	2	0	
4	1	15	3	<200 ufc	↳
↳ 4	4	1	211 ufc	2021-02-02	↳
↳ 16:42:06.183816	M316-20	1	2	0	
5	1	16	4	<200 ufc	↳
↳ 5	5	1	211 ufc	2021-02-02	↳
↳ 16:42:06.184012	M316-20	1	2	0	
6	1	16	3	<200 ufc	↳
↳ 6	6	1	211 ufc	2021-02-02	↳
↳ 16:42:06.184231	M316-20	1	2	0	
7	2	17	4	<100 ufc	↳
↳ 7	7	1	211 ufc	2021-02-02	↳
↳ 16:42:06.184489	M316-20	1	2	0	
18	3	10	4	<150 ufc	↳
↳ 8	18	1	211 ufc	2021-02-02	↳
↳ 16:42:06.184865	M341-22	1	2	1	

Podemos revisar los datos más relevantes de la lectura con la vista:

```
!sqlite3 -header -markdown rodam/aceptacion.db "select * from lectura_con_nombres;"
```

nom_producto	id_muestra	nom_grupo	nom_analisis	nom_metodo	nom_material
			especificacion	resultado	material
			verifica	concepto	
----- ----- ----- ----- ----- -----					
----- ----- ----- ----- ----- -----					
----- ----- ----- ----- ----- -----					
----- ----- ----- ----- ----- -----					
Xinefax	1		Recuento del	inóculo	
Actividad Bactericida Básica			Inóculo de material controlado		Staphylococcus
Aureus ATCC 6538	170<t<230	112 ufc	Solución salina	M314-20	1
2	0				
Xinefax	1		Recuento del	inóculo	
Actividad Bactericida Básica			Inóculo de material controlado		Pseudomona
aeruginosa ATCC 9027	170<t<230	211 ufc	Solución salina	M315-20	
1	2	0			
Xinefax	1		Recuento del	inóculo	
Actividad Bactericida Básica			Siembra e incubación		Staphylococcus
Aureus ATCC 6538	<200 ufc	211 ufc	Agar Sabouraud	M316-20	1
2	0				
Xinefax	1		Recuento del	inóculo	
Actividad Bactericida Básica			Siembra e incubación		Pseudomona
aeruginosa ATCC 9027	<200 ufc	211 ufc	Agar Sabouraud	M316-20	
1	2	0			
Xinefax	1		Recuento del	inóculo	
Actividad Bactericida Básica			Conteo en caja de Petri		Staphylococcus
Aureus ATCC 6538	<200 ufc	211 ufc	Agar TSB	M316-20	1
2	0				
Xinefax	1		Recuento del	inóculo	
Actividad Bactericida Básica			Conteo en caja de Petri		Pseudomona
aeruginosa ATCC 9027	<200 ufc	211 ufc	Agar TSB	M316-20	
1	2	0			
Xinefax	1		Control de toxicidad del	neutralizante	
Actividad Bactericida Básica			Enriquecimiento + Peptona		Staphylococcus
Aureus ATCC 6538	<100 ufc	211 ufc	Agar TSB	M316-20	1
2	0				
Xinefax	1		Control del	método de dilución-neutralización	
Actividad Bactericida Básica			Enriquecimiento		Staphylococcus
Aureus ATCC 6538	<150 ufc	211 ufc	Agar TSA	M341-22	1
2	1				

### 5.11.1 Control de calidad de los medios

El control de calidad se debe asociar a una muestra. En esa muestra, tendríamos que ver qué medios son susceptibles al control de calidad, y realizar varios controles por cada medio. En este caso, observamos que el Agar TSA del necesita un control de calidad del medio. Haremos el ejemplo con el lote M341-22, teniendo en mente que cada uno de los lotes implicados en una muestra tendrían que pasar por control del medio:

```
# lectura 15 (Enriquecimiento normal - medio TSA)

control_medio1 = Dominio.Control_Material(
lote_material = "M341-22",
id_material_controlado = bacilus.id_material,
```

(continué en la próxima página)

(proviene de la página anterior)

```
lote_material_controlado = "486-862-5-a10-4",
especificacion = "Entre 50 y 150 ufc",
resultado = "98 ufc",
concepto = True
)

control_medio2 = Dominio.Control_Material(
lote_material = "M341-22",
id_material_controlado = pseudomona.id_material,
lote_material_controlado = "484-862-5-a10-4",
especificacion = "Entre 50 y 150 ufc",
resultado = "98 ufc",
concepto = True
)

control_medio3 = Dominio.Control_Material(
lote_material = "M341-22",
id_material_controlado = staph_aureus.id_material,
lote_material_controlado = "483-862-5-a10-4",
especificacion = "Entre 50 y 150 ufc",
resultado = "98 ufc",
concepto = True
)

control_medio4 = Dominio.Control_Material(
lote_material = "M341-22",
especificacion = "Entre 50 y 150 ufc",
lote_material_controlado = "286-862-5-a10-4",
resultado = "98 ufc",
concepto = True
)

control_medio5 = Dominio.Control_Material(
lote_material = "M341-22",
id_material_controlado = aspergillus.id_material,
especificacion = "Entre 50 y 150 ufc",
lote_material_controlado = "386-862-5-a10-4",
resultado = "98 ufc",
concepto = True
)

control_del_medio = [control_medio1, control_medio2, control_medio3, control_medio4,
↪control_medio5]
```

```
sesion.add_all(control_del_medio)
sesion.commit()
```

```
!sqlite3 -header -markdown rodam/aceptacion.db 'select lote_medio, nom_microorganismo,
↪ ATC, lote_cepa, especificacion, resultado, fecha_resultado from control_medio,
↪microorganismo where microorganismo.id_microorganismo = control_medio.id_
↪microorganismo '
```

lote_medio	nom_microorganismo	ATC	lote_cepa	
↪especificacion	↪resultado	↪fecha_resultado		

(continué en la próxima página)

(proviene de la página anterior)

----- ----- ----- ----- -----
↪----- ----- ----- -----
M341-22   Bacillus subtilis spizizenii   6633   486-862-5-a10-4   Entre 50 y 150 ufc   98 ufc   2021-01-29 12:48:14.117903
↪150 ufc   98 ufc   2021-01-29 12:48:14.120013
M341-22   Pseudomona aeruginosa   9027   484-862-5-a10-4   Entre 50 y 150 ufc   98 ufc   2021-01-29 12:48:14.120613
↪150 ufc   98 ufc   2021-01-29 12:48:14.121559
M341-22   Staphylococcus Aureus   6538   483-862-5-a10-4   Entre 50 y 150 ufc   98 ufc   2021-01-29 12:48:14.121559
↪150 ufc   98 ufc   2021-01-29 12:48:14.121559
M341-22   Aspergillus bresilensis   16404   386-862-5-a10-4   Entre 50 y 150 ufc   98 ufc   2021-01-29 12:48:14.121559
↪150 ufc   98 ufc   2021-01-29 12:48:14.121559

l ote _me dio	nom_ microorga- nismo	A T C	lot e_cepa	especi fica- cion	res ult ado	fe cha_resultado
M 341 -22	Bacillus subtilis spizizenii	6 6 3 3	486 -862-5 - a10-4	Entre 50 y 150 ufc	98 ufc	2021-01-29 12:48:14.117903
M 341 -22	Pseudomona aeruginosa	9 0 2 7	484 -862-5 - a10-4	Entre 50 y 150 ufc	98 ufc	2021-01-29 12:48:14.120013
M 341 -22	Staphylococcus Aureus	6 5 3 8	483 -862-5 - a10-4	Entre 50 y 150 ufc	98 ufc	2021-01-29 12:48:14.120613
M 341 -22	Aspergillus bresilensis	1 6 4 0 4	386 -862-5 - a10-4	Entre 50 y 150 ufc	98 ufc	2021-01-29 12:48:14.121559

### 5.11.2 Emisión de certificados

En esta sección veremos cómo podemos construir un certificado de análisis a partir de las tablas que tenemos. Con consultas más elaboradas, podemos reconstruir los aspectos principales del certificado de análisis actual:

## 5.12 Informacion del cliente

Con una consulta como:

```
select nom_empresa as empresa, nit_empresa as nit, direccion, pagina, correo from empresa, ciudad where empresa.id_ciudad = ciudad.id_ciudad and empresa.id_empresa = 1;
```

Podemos generar:

```
!sqlite3 -header -line rodam/aceptacion.db 'select nom_empresa as empresa, nit_empresa as nit, direccion, pagina, correo from empresa, ciudad where empresa.id_ciudad = ciudad.id_ciudad and empresa.id_empresa = 1;'
```

```
empresa = Compañía Nacional de Cosméticos CONALCOS
nit = 90024
direccion = calle 34 Bis 39-33
pagina = conalcos.com.co
correo = contacto@conalco.com.co
```

## 5.13 Información de la muestra

Con una consulta como :

```
select nom_producto, presentacion,tamano_muestra as tamano, unidades_tamano as
↪unidades, ingreso_muestra, nom_miembro as registra from producto, muestra, miembro_
↪rodam m where producto.id_producto = muestra.id_muestra and muestra.registra = m.id_
↪miembro
```

produce:

```
!sqlite3 -header -line rodam/aceptacion.db 'select nom_producto, presentacion,tamano_
↪muestra as tamano, unidades_tamano as unidades, ingreso_muestra, nom_miembro as
↪registra from producto, muestra, miembro_rodam m where producto.id_producto =
↪muestra.id_muestra and muestra.registra = m.id_miembro'
```

```
nom_producto = Xinefax
presentacion = tabletas
tamano = 500
unidades = mg
ingreso_muestra = 2021-02-02 16:42:05.599979
registra = Ernesto Segura
```

## 5.14 Resumen analisis

```
!sqlite3 -header -markdown rodam/aceptacion.db "select * from lectura_con_nombres"
```

nom_producto	id_muestra	nom_grupo	nom_metodo	nom_material
especificacion	resultado	material	lote_medio	registra
verifica	concepto			
Xinefax	1	Recuento del inóculo		
Actividad Bactericida Básica	Inóculo de material controlado	Staphylococcus		
Aureus ATCC 6538	170<t<230	112 ufc	Solución salina	M314-20
2	0			
Xinefax	1	Recuento del inóculo		
Actividad Bactericida Básica	Inóculo de material controlado	Pseudomona		
aeruginosa ATCC 9027	170<t<230	211 ufc	Solución salina	M315-20
1	2	0		
Xinefax	1	Recuento del inóculo		
Actividad Bactericida Básica	Siembra e incubación	Staphylococcus		
Aureus ATCC 6538	<200 ufc	211 ufc	Agar Sabouraud	M316-20
2	0			
Xinefax	1	Recuento del inóculo		
Actividad Bactericida Básica	Siembra e incubación	Pseudomona		
aeruginosa ATCC 9027	<200 ufc	211 ufc	Agar Sabouraud	M316-20
1	2	0		
Xinefax	1	Recuento del inóculo		
Actividad Bactericida Básica	Conteo en caja de Petri	Staphylococcus		
Aureus ATCC 6538	<200 ufc	211 ufc	Agar TSB	M316-20
2	0			

(continué en la próxima página)

(proviene de la página anterior)

Xinefax	1	Recuento <b>del</b> inóculo		
↪Actividad Bactericida Básica	Conteo en caja de Petri	Pseudomona		
↪aeruginosa ATCC 9027	<200 ufc	211 ufc	Agar TSB	M316-20
↪1	2	0		
Xinefax	1	Control de toxicidad <b>del</b> neutralizante		
↪Actividad Bactericida Básica	Enriquecimiento + Peptona	Staphylococcus		
↪Aureus ATCC 6538	<100 ufc	211 ufc	Agar TSB	M316-20
↪	2	0		1
Xinefax	1	Control <b>del</b> método de dilución-neutralización		
↪Actividad Bactericida Básica	Enriquecimiento	Staphylococcus		
↪Aureus ATCC 6538	<150 ufc	211 ufc	Agar TSA	M341-22
↪	2	1		1

## 5.15 Equipos utilizados

nom_equipo	ref_documental
Cabina de flujo laminar	E001
Balanza	E020
Balanza	E020
Micropipeta	E019
Incubadora	E010
Balanza	E020
Caja de Petri	E011





---

Reporte de pruebas

---

**author** Jonatan Ahumada Fernández

**contact** [jaumaf@hotmail.com](mailto:jaumaf@hotmail.com)

**date** último build el 2022-03-29

Aquí se presenta el reporte de cada prueba unitaria o de integración automatizada. Si se desea revisar con más detalle cada prueba se debe buscar en la carpeta *tests* del proyecto de Django cada prueba individual.

```
Creating test database for alias 'default' ('file:memorydb_default?mode=memory&
↪cache=shared')...
test_crea_un_certificado_y_reconstruye_objetos (tests.emision.test_
↪emisorDeCertificado.Test_EmisorDeCertificado) ... ok
test_datos_muestra_generan_hash (tests.emision.test_emisorDeCertificado.Test_
↪EmisorDeCertificado) ... ok
test_hash_cambia_si_alteran_datos (tests.emision.test_emisorDeCertificado.Test_
↪EmisorDeCertificado) ... ok
test_se_necesita_tener_firma_y_logo (tests.emision.test_emisorDeCertificado.Test_
↪EmisorDeCertificado) ... ok
test_transforma_datos_a_json (tests.emision.test_emisorDeCertificado.Test_
↪EmisorDeCertificado) ... ok
test_dropdowns_de_grupo_son_restringidos (tests.formularios.test_especificar_un_
↪ analisis.TestFormularioParaEspecificarUnAnalisis) ... ok
test_dropdowns_de_metodo_son_restringidos (tests.formularios.test_especificar_un_
↪ analisis.TestFormularioParaEspecificarUnAnalisis) ... ok
test_el_compositor_se_instancia (tests.formularios.test_especificar_un_analisis.
↪ TestFormularioParaEspecificarUnAnalisis) ... ok
test_genera_data_inicial (tests.formularios.test_especificar_un_analisis.
↪ TestFormularioParaEspecificarUnAnalisis) ... ok
test_genera_el_formset_al_init (tests.formularios.test_especificar_un_analisis.
↪ TestFormularioParaEspecificarUnAnalisis) ... ok
test_enlaza_datos_del_post (tests.formularios.test_lecturaDeMuestra.
↪ TestFormularioDeLecturasDeMuestraCompuesto)
Me toca hacer un mock del client y pasar un formulario gigante de post ... ok
test_produce_numero_de_formularios_correcto (tests.formularios.test_lecturaDeMuestra.
↪ TestFormularioDeLecturasDeMuestraCompuesto) ... ok
```

(continué en la próxima página)

(proviene de la página anterior)

```

test_se_instancia_correctamente (tests.formularios.test_lecturaDeMuestra.
↪TestFormularioDeLecturasDeMuestraCompuesto)
atributos se calculan según lo esperado ... ok
test_formulario_recibe_muestra_y_quien_registra_automaticamente (tests.formularios.
↪test_lectura_individual.TestFormularioDeLecturaDeMuestraIndividual) ... ok
test_retorna_formulario_a_partir_de_especificacion (tests.formularios.test_lectura_
↪individual.TestFormularioDeLecturaDeMuestraIndividual) ... ok
test_cambio_el_certificado_vigente_por_uno_nuevo (tests.procesos.base.
↪configuracionCambiaEnTiempoReal.ConfiguracionCambieEnTiempoDeEjecucion) ... ok
test_remisor_marca_el_uso (tests.procesos.calidad.medios.test_cuadro_control_marca_
↪uso.TestFlujoRegularDeCalidad) ... ok
test_flujo (tests.procesos.calidad.test_flujo_regular.TestFlujoRegularDeCalidad) ...
↪ok
test_metodos_asociados_al_analisis (tests.procesos.catalogo.test_datos_asociados_al_
↪analisis.TestMetodosAsociadosAlAnalisis)
Director de la informacion ingresa analisis y le ... ok
test_datos_asociados_al_metodo (tests.procesos.catalogo.test_datos_asociados_al_
↪metodo.test_datos_asociados_al_metodo)
Los metodos quedan con equipos ... ok
test_el_grupo_tiene_analisis (tests.procesos.catalogo.test_el_grupo_tiene_analisis.
↪GrupoTieneAnalisis)
los grupos quedan asociados con el analisis ... ok
test_el_cuadro_queda_aprobado_por_calidad (tests.procesos.ingreso.test_cuadro_queda_
↪aprobado_por_calidad.TestCuadroQuedaAprobadoPorCalidad)
Luego de que el cuadro se aprueba por el servicio, ... ok
test_el_cuadro_queda_con_analisis_asignados (tests.procesos.ingreso.test_cuadro_queda_
↪con_analisis_asignados.TestCuadroQuedaConAnalisisAsignados)
los cuadros quedan con analisis ... ok
test_datos_asociados_al_cuadro (tests.procesos.ingreso.test_datos_asociados_al_cuadro.
↪TestDatosAsociadosAlCuadro)
Los cuadros quedan con responsable, ... ok
test_especificaciones_se_crean (tests.procesos.ingreso.test_especificaciones_se_crean.
↪TestDatosAsociadosAlCuadro)
las especificaciones quedan registradas en la ... ok
test_datos_asociados_al_cuadro (tests.procesos.inventario.test_desglose.
↪TestFlujoRegular)
Los cuadros quedan con responsable, un producto y un estado ... ok
test_datos_asociados_al_metodo (tests.procesos.inventario.test_desglose.
↪TestFlujoRegular)
Los metodos quedan con equipos ... ok
test_datos_asociados_al_producto (tests.procesos.inventario.test_desglose.
↪TestFlujoRegular)
El producto queda con el cliente y el sector ... ok
test_el_cuadro_queda_aprobado_por_calidad (tests.procesos.inventario.test_desglose.
↪TestFlujoRegular) ... skipped ''
test_el_cuadro_queda_con_analisis_asignados (tests.procesos.inventario.test_desglose.
↪TestFlujoRegular)
los cuadros quedan con analisis ... ok
test_el_grupo_tiene_analisis (tests.procesos.inventario.test_desglose.
↪TestFlujoRegular)
los grupos quedan asociados con el analisis ... ok
test_especificaciones_se_crean (tests.procesos.inventario.test_desglose.
↪TestFlujoRegular)
las especificaciones quedan registradas en la ... ok
test_metodos_asociados_al_analisis (tests.procesos.inventario.test_desglose.
↪TestFlujoRegular)
Director de la informacion ingresa analisis y le ... ok

```

(continué en la próxima página)

(proviene de la página anterior)

```
test_metodos_pertenecen_al_analisis (tests.procesos.inventario.test_desglose.
↳ TestFlujoRegular)
Reviso si el servicio de especificacion obtiene los metodos utilizados en el
↳ analisis ... skipped ''
test_flujo (tests.procesos.inventario.test_flujo.Flujo) ... ok
test_carga_configuracion (tests.procesos.login.test_dashboard_carga_configuracion.
↳ TestConfiguracionDeDashboard)
Se carga un diccionario de configuración que en settings ... ok
test_miembro_rodam_genera_django_user (tests.procesos.login.test_miembro_rodam_genera_
↳ django_user.TestMiembroRodamGeneraUsuarioDjango)
Director de información crea un miembroRodam y automáticamente se crea un django_user
↳ ... ok
test_se_imprime_con_datos_falsos (tests.reporter.test_certificado_de_emision.PDF_
↳ Reporter) ... skipped ''
test_se_integra_con_la_muestra (tests.reporter.test_certificado_de_emision.PDF_
↳ Reporter) ... ok
test_el_archivo_tiene_numero_de_filas_correctas (tests.reporter.test_lote_de_medio_
↳ reporter.TestMediosReporter) ... ok
test_el_reporter_escribe_a_un_file_object (tests.reporter.test_lote_de_medio_reporter.
↳ TestMediosReporter) ... ok
test_la_tabla_extrae_correctamente_las_lecturas_de_los_lotr (tests.reporter.test_
↳ lote_de_medio_reporter.TestMediosReporter) ... ok
test_se_crea_un_csv_en_los_artefactos (tests.reporter.test_muestra_reporter.
↳ TestMuestraReporter) ... ok
test_no_hay_errores_en_init (tests.services.test_
↳ EspecificacionDeAnalisisEnCuadroAdaptor.
↳ TestEspecificacionDeAnalisisEnCuadroAdaptor) ... ok
test_puede_cargar_objetos_de_los_managers (tests.services.test_
↳ EspecificacionDeAnalisisEnCuadroAdaptor.
↳ TestEspecificacionDeAnalisisEnCuadroAdaptor) ... ok
test_retorna_los_grupos_relevantes (tests.services.test_
↳ EspecificacionDeAnalisisEnCuadroAdaptor.
↳ TestEspecificacionDeAnalisisEnCuadroAdaptor) ... ok
test_retorna_los_utilizas (tests.services.test_
↳ EspecificacionDeAnalisisEnCuadroAdaptor.
↳ TestEspecificacionDeAnalisisEnCuadroAdaptor) ... ok
test_cada_fila_en_especificaciones_solo_tiene_str (tests.services.test_
↳ LoteDeMedioAdaptor.TestLoteDeMedioAdaptor) ... ok
test_carga_cuadro_al_inicializar (tests.services.test_LoteDeMedioAdaptor.
↳ TestLoteDeMedioAdaptor) ... ok
test_carga_lote_de_medio (tests.services.test_LoteDeMedioAdaptor.
↳ TestLoteDeMedioAdaptor)
el adaptador carga el objeto ... ok
test_carga_metodos_al_inicializar (tests.services.test_LoteDeMedioAdaptor.
↳ TestLoteDeMedioAdaptor) ... ok
test_el_diccionario_tiene_los_metodos_del_objeto (tests.services.test_
↳ LoteDeMedioAdaptor.TestLoteDeMedioAdaptor) ... ok
test_especificaciones_se_muestran_como_lista_de_listas (tests.services.test_
↳ LoteDeMedioAdaptor.TestLoteDeMedioAdaptor) ... ok
test_identifica_muestras_que_han_utilizado_el_lote (tests.services.test_
↳ LoteDeMedioAdaptor.TestLoteDeMedioAdaptor) ... ok
test_retorna_un_diccionario (tests.services.test_LoteDeMedioAdaptor.
↳ TestLoteDeMedioAdaptor) ... ok
test_carga_la_muestra (tests.services.test_MuestraAdaptor.TestMuestraAdaptor) ... ok
test_columnas_para_normalizacion (tests.services.test_MuestraAdaptor.
↳ TestMuestraAdaptor) ... ok
test_distintos_analisis_aparecen_en_diccionario (tests.services.test_MuestraAdaptor.
↳ TestMuestraAdaptor)
```

(continué en la próxima página)

(proviene de la página anterior)

```
Cada analisis tiene su propia llave ... ok
test_get_all_especificaciones (tests.services.test_MuestraAdaptor.TestMuestraAdaptor) ...
↳... ok
test_get_analisis_asignados (tests.services.test_MuestraAdaptor.TestMuestraAdaptor) ..
↳. ok
test_llaves_de_diccionario_son_strings (tests.services.test_MuestraAdaptor.
↳TestMuestraAdaptor) ... ok
test_se_extraen_los_equipos_utilizados (tests.services.test_MuestraAdaptor.
↳TestMuestraAdaptor) ... ok
test_tabla_de_emision_filtra_por_marcas_de_emision (tests.services.test_
↳MuestraAdaptor.TestMuestraAdaptor) ... ok
test_tabla_de_lecturas_es_una_lista_de_listas (tests.services.test_MuestraAdaptor.
↳TestMuestraAdaptor) ... ok
test_tabla_especificaciones_es_lista_de_listas (tests.services.test_MuestraAdaptor.
↳TestMuestraAdaptor) ... ok
test_se_identifican_los_lotes_de_medio_utilizados (tests.services.test_
↳MuestraAdaptorGestionaLotes.TestMuestraAdaptorGestionaLotes) ... ok
test_se_pueden_extraer_las_lecturas_de_control (tests.services.test_
↳MuestraAdaptorGestionaLotes.TestMuestraAdaptorGestionaLotes) ... ok
test_cambia_el_adaptor_de_lote (tests.services.test_PdfAdaptor.
↳ComportamientoPDFAdaptor) ... ok
test_el_formato_es_inmutable (tests.services.test_PdfAdaptor.
↳ComportamientoPDFAdaptor) ... ok
test_exporta_data_de_certificado (tests.services.test_PdfAdaptor.
↳ComportamientoPDFAdaptor) ... ok
test_puede_formatear_un_item (tests.services.test_PdfAdaptor.
↳ComportamientoPDFAdaptor) ... ok
test_se_puede_construir_una_filas (tests.services.test_PdfAdaptor.
↳ComportamientoPDFAdaptor) ... ok
test_seccion_cliente_es_una_filas_con_3_columnas (tests.services.test_PdfAdaptor.
↳ComportamientoPDFAdaptor) ... ok
test_seccion_muestra_es_una_tabla_de_5_por_4 (tests.services.test_PdfAdaptor.
↳ComportamientoPDFAdaptor) ... ok
test_cuenta_correctamente_las_especificaciones_completas (tests.services.test_
↳SalaDeControlAdaptor.TestSalaDeControlAdaptor) ... ok
test_cuenta_correctamente_las_especificaciones_pendientes (tests.services.test_
↳SalaDeControlAdaptor.TestSalaDeControlAdaptor) ... ok
test_genera_resumen (tests.services.test_SalaDeControlAdaptor.
↳TestSalaDeControlAdaptor) ... ok
test_se_generan_mensajes_para_errores_de_validacion (tests.services.test_
↳SalaDeControlAdaptor.TestSalaDeControlAdaptor) ... ok
test_valida_si_pueden_aprobarse_las_lecturas (tests.services.test_
↳SalaDeControlAdaptor.TestSalaDeControlAdaptor) ... ok
test_encabezados_de_lectura_cambian (tests.services.test_SalaDeLecturasParaMuestra.
↳TestSalaDeLecturasAdaptor) ... ok
test_hace_el_init (tests.services.test_SalaDeLecturasParaMuestra.
↳TestSalaDeLecturasAdaptor) ... ok
test_muestra_mensaje_de_validacion (tests.services.test_SalaDeLecturasParaMuestra.
↳TestSalaDeLecturasAdaptor) ... ok
test_no_valida_si_hay_especificaciones_pendientes (tests.services.test_
↳SalaDeLecturasParaMuestra.TestSalaDeLecturasAdaptor) ... ok
test_produce_resumen (tests.services.test_SalaDeLecturasParaMuestra.
↳TestSalaDeLecturasAdaptor) ... ok
test_reporta_lecturas_completas (tests.services.test_SalaDeLecturasParaMuestra.
↳TestSalaDeLecturasAdaptor) ... ok
test_reporta_lecturas_pendientes (tests.services.test_SalaDeLecturasParaMuestra.
↳TestSalaDeLecturasAdaptor) ... ok
```

(continúe en la próxima página)

(proviene de la página anterior)

```
test_tabla_de_lecturas_es_reducida (tests.services.test_SalaDeLecturasParaMuestra.
↳TestSalaDeLecturasAdaptor) ... ok
test_valida_cuando_las_especificaciones_estan_completas_y_hay_fecha_de_inicio (tests.
↳services.test_SalaDeLecturasParaMuestra.TestSalaDeLecturasAdaptor) ... ok
test_armar_dashboard_para_rol (tests.services.test_dashboard_dinamico.TestDashboard) .
↳... ok
test_booleano_cambia_a_string (tests.services.test_
↳loteDeMedioParaCertificadoDeEmision.TestLoteDeMedioParaCertificadoDeEmision)
Esto ya no aplica. Hay que quitarlo ... skipped ''
test_cambios_quedan_en_log_entry (tests.teoremas.test_LogEntry.TestLogEntry) ... ok
test_email_usando_smtp_externo (tests.teoremas.test_email.TestEmail)
Envia mail usando un servicio SMTP real. ... ok
test_email_usando_smtp_local (tests.teoremas.test_email.TestEmail)
Envía correo con adjunto usando un servidor SMTP Local ... ok
test_por_defecto_en_memoria (tests.teoremas.test_email.TestEmail)
Muestra comportamiento básico del backend en tests. ... ok
test_como_hacer_una_muestra (tests.teoremas.test_factories.TestFactories) ... ok
test_no_puedo_inicializar_atributos_en_un_stub (tests.teoremas.test_factories.
↳TestFactories) ... ok
test_es_hashable (tests.teoremas.test_named_tuple_es_hashable.
↳TestNamedTupleEsHashable) ... ok
test_comparaciones_basicas (tests.teoremas.test_organizar_fechas.TestFactories) ... ok
test_comparaciones_con_datetimes_localizadas (tests.teoremas.test_organizar_fechas.
↳TestFactories) ... ok
test_localizacion (tests.teoremas.test_organizar_fechas.TestFactories) ... ok
test_se_ordenan_en_una_lista (tests.teoremas.test_organizar_fechas.TestFactories) ...
↳ok
test_reglas_no_permiten_aprobar_un_lote (tests.validaciones.lote_de_medio.test_
↳validar_aprobacion.TestValidacionesLoteDeMedio)
El lote se intenta aprobar. Pero como *no* cumple las reglas de remisión el ... ok
test_reglas_permiten_aprobar_un_lote (tests.validaciones.lote_de_medio.test_validar_
↳aprobacion.TestValidacionesLoteDeMedio)
Como el lote cumple con las reglas, se puede aprobar sin problemas ... ok
test_remisor_remite (tests.validaciones.lote_de_medio.test_validar_aprobacion.
↳TestValidacionesLoteDeMedio)
Remitir es confirmar que los datos del lote están ... ok
test_especificaciones_completas_pero_no_cumplen_no_puede_solicitar (tests.
↳validaciones.lote_de_medio.test_validar_solicitar_aprobacion.
↳TestValidacionesLoteDeMedio) ... ok
test_especificaciones_completas_y_cumplen_puede_solicitar (tests.validaciones.lote_de_
↳medio.test_validar_solicitar_aprobacion.TestValidacionesLoteDeMedio) ... ok
test_especificaciones_incompletas_no_puede_solicitar_aprobacion (tests.validaciones.
↳lote_de_medio.test_validar_solicitar_aprobacion.TestValidacionesLoteDeMedio) ... ok
test_CuadrosPorAprobar (tests.vistas.calidad.test_views_retornan_ok.ViewsRetornanOk)
Aprobar cuadros analíticos ... skipped ''
test_aprobarCuadroAnalitico (tests.vistas.calidad.test_views_retornan_ok.
↳ViewsRetornanOk)
Vista aprobar cuadro analítico ... skipped ''
test_ingresar_analisis (tests.vistas.catalogo.test_views_retornan_ok.ViewsRetornanOk)
Formulario de análisis ... skipped ''
test_ingresar_grupo (tests.vistas.catalogo.test_views_retornan_ok.ViewsRetornanOk)
Formulario de grupo ... skipped ''
test_ingresar_metodo (tests.vistas.catalogo.test_views_retornan_ok.ViewsRetornanOk)
Formulario de método ... skipped ''
test_CambiaConceptoDeLecturaPorId (tests.vistas.emision.test_views_retornan_ok.
↳ViewsRetornanOk)
Modificar lectura ... skipped ''
```

(continué en la próxima página)

(proviene de la página anterior)

```
test_MuestrasPendientesPorAprobacionFinal (tests.vistas.emision.test_views_retornan_
↳ok.ViewsRetornanOk)
Muestras por aprobacion final ... skipped ''
test_CuadroAnalitico_detalle (tests.vistas.ingreso.test_views_retornan_ok.
↳ViewsRetornanOk) ... skipped ''
test_CuadroAnalitico_ingresar (tests.vistas.ingreso.test_views_retornan_ok.
↳ViewsRetornanOk) ... skipped ''
test_cuadroAnalitico_remitir (tests.vistas.ingreso.test_views_retornan_ok.
↳ViewsRetornanOk) ... skipped ''
test_especificacion_ingresar (tests.vistas.ingreso.test_views_retornan_ok.
↳ViewsRetornanOk) ... skipped ''
test_muestra_ingresar (tests.vistas.ingreso.test_views_retornan_ok.ViewsRetornanOk) ..
↳. skipped ''
test_muestra_lista (tests.vistas.ingreso.test_views_retornan_ok.ViewsRetornanOk) ...↳
↳skipped ''
test_el_pipe_funciona (tests.emision.test_baseBuilder.PDFBuilder) ... ok
test_envolver_en_parrafos_ignora_items_que_no_son_strings (tests.emision.test_
↳baseBuilder.PDFBuilder) ... ok
test_envuelve_en_parrafos_los_items_de_una_tabla (tests.emision.test_baseBuilder.
↳PDFBuilder) ... ok
test_multidimensiona_un_string (tests.emision.test_baseBuilder.PDFBuilder) ... ok
test_se_construye_la_historia (tests.emision.test_baseBuilder.PDFBuilder) ... ok

-----
Ran 125 tests in 105.408s

OK (skipped=17)
Destroying test database for alias 'default' ('file:memorydb_default?mode=memory&
↳cache=shared')...
```

## CAPÍTULO 7

### Diccionario de datos

**author** Jonatan Ahumada Fernández

**contact** [jaumaf@hotmail.com](mailto:jaumaf@hotmail.com)

**date** último build el 2022-03-29

Tabla 1: Diccionario

dato	atributos	tipo	longitud	formato	rango	explicacion	observaciones
ciudad	id_ciudad	INTEGER					
	nom_ciudad	VARCHAR					
	indicativo	VARCHAR					
presentacion	id_presentacion	INTEGER					
	nom_presentacion	VARCHAR					
	descripcion_presentacion	VARCHAR					
cargo	id_cargo	INTEGER					
	nom_cargo	VARCHAR					
	jefe	INTEGER					
	reemplazo	INTEGER					
origen	observaciones	VARCHAR					
	id_origen	INTEGER					
	nom_origen	VARCHAR					
	desc_origen	VARCHAR					
sector	id_sector	INTEGER					
	nom_sector	VARCHAR					
equipo	id_equipo	INTEGER					
	nom_equipo	VARCHAR					
	ref_documental	VARCHAR					
metodo	id_metodo	INTEGER					
	nom_metodo	VARCHAR					
	desc_metodo	VARCHAR					
	ref_documental	VARCHAR					

continué en

Tabla 1 – proviene de la página anterior

	observaciones	VARCHAR					
	material	VARCHAR					
grupo	id_grupo	INTEGER					
	nom_grupo	VARCHAR					
	desc_grupo	VARCHAR					
control_material	id_control_material	INTEGER					
	lote_material	VARCHAR					
	id_material_de_control	VARCHAR					
	lote_material_de_control	VARCHAR					
	especificacion	VARCHAR					
	resultado	VARCHAR					
	fecha_resultado	DATETIME					
	concepto	BOOLEAN					
metodo_equipo	id_metodo	INTEGER					
	id_equipo	INTEGER					
analisis	id_analisis	INTEGER					
	nom_analisis	VARCHAR					
	id_sector	INTEGER					
	descripcion	VARCHAR					
	ref_documental	VARCHAR					
empresa	id_empresa	INTEGER					
	id_ciudad	INTEGER					
	nit_empresa	INTEGER					
	cod_verificacion	INTEGER					
	nom_empresa	VARCHAR					
	direccion	VARCHAR					
	pagina	VARCHAR					
	correo	VARCHAR					
	descripcion	VARCHAR					
	observaciones	VARCHAR					
utiliza	id_utiliza	INTEGER					
	id_analisis	INTEGER					
	id_metodo	INTEGER					
agrupacion	id_agrupacion	INTEGER					
	id_grupo	INTEGER					
	id_analisis	INTEGER					
cliente	id_cliente	INTEGER					
	id_empresa	INTEGER					
	retencion	FLOAT					
	IVA	FLOAT					
	ICA	FLOAT					
	fecha_inicio	DATETIME					
	fecha_final	DATETIME					
	descripcion	VARCHAR					
producto	id_producto	INTEGER					
	id_cliente	INTEGER					
	id_sector	INTEGER					
	forma_farmaceutica	VARCHAR					
	nom_producto	VARCHAR					
muestra	id_muestra	INTEGER					

continué en



Tabla 1 – proviene de la página anterior

	id_producto	INTEGER					
	id_origen	INTEGER					
	presentacion	VARCHAR					
	lote_muestra	VARCHAR					
	ingreso_muestra	DATETIME					
	descripcion	VARCHAR					
	tamano_muestra	INTEGER					
	unidades_tamano	VARCHAR					¿por qué separar?
	solicitado	VARCHAR					¿cuál es su uso?
	aceptada	BOOLEAN					
	registra	INTEGER					
	verifica	INTEGER					el registro?
	terminada	BOOLEAN					
	concepto	BOOLEAN					
	aprueba	VARCHAR					aprueba qué?
	fecha_final	DATETIME					
	factura	INTEGER					
	observaciones_muestra_inicio	VARCHAR					
	observaciones_muestra_fin	VARCHAR					
debe tener	id_debe_tener	INTEGER					
	id_producto	INTEGER					
	id_grupo	INTEGER					
especificacion	id_especificacion	INTEGER					
	id_debe_tener	INTEGER					
	id_utiliza	INTEGER					
	id_material_controlado	INTEGER					
	valor	VARCHAR					
lectura	id_lectura	INTEGER					
	id_especificacion	INTEGER					
	id_muestra	INTEGER					
	valor_lectura	VARCHAR					
	fecha_lectura	DATETIME					
	lote_medio	VARCHAR					
	registra	INTEGER					
	verifica	INTEGER					
	concepto	BOOLEAN					
	observaciones	VARCHAR					
material_de_control	id_material	INTEGER					
	nom_material	VARCHAR					



---

## Reporte de validación

---

**author** Jonatan Ahumada Fernández

**contact** [jaumaf@hotmail.com](mailto:jaumaf@hotmail.com)

**date** último build el 2022-03-29

### 8.1 Introducción y alcance

Este documento constituye el reporte de validación para el sistema de información de Rodam análisis. El propósito de este documento es explicar cómo se mitigaron los riesgos implicados en el proceso, identificar los criterios de logro y documentar los hitos más relevantes durante el proceso.

La estimación de los riesgos más relevantes puede leerse en el *Plan de Pruebas*. La documentación de los requisitos puede verse en la *Especificación de Requerimientos*

### 8.2 Descripción del sistema

El sistema de información es una aplicación web para el ingreso de muestras, la recolección de las lecturas de las muestras y la emisión de certificados microbiológicos del laboratorio Rodam Análisis. El sistema es accesible mediante internet con usuario y contraseña y es desplegado en un entorno de servidor virtual, del cual Rodam Análisis es dueño. Un a vista con mayor profundidad puede verse en el *Documento de Diseño*.

## 8.3 Manejo de riesgo

En esta sección se describirán las acciones de mitigación de riesgos empleadas para los riesgos identificados.

### 8.3.1 Riesgo 1: modelo de datos del sistema es insuficiente y/o genera inconsistencias en los datos

En un sistema de información, más que en un sistema de uso comercial, la integridad de los datos es crítica. Por tal motivo, desde el inicio del proyecto se llevaron a cabo las siguientes acciones:

- elaboración de una prueba de aceptación que mostrara el flujo de los datos a través de la futura aplicación (veáse Prueba de Aceptación)
- normalización de la base de datos hasta tercera forma normal. Esto evitará inconsistencias de lectura, escritura y actualización
- pruebas unitarias que muestran el flujo de los datos de la aplicación

## 8.4 Riesgo 2: el artefacto principal, el certificado de emisión, es incompleto

El objetivo principal del sistema es automatizar el proceso de emisión del certificado, que requería mucha labor manual. Para asegurar el cumplimiento de este requerimiento se llevaron a cabo las siguientes acciones:

- se hizo una réplica 1 a 1 del certificado de análisis microbiológico anterior
- se hizo una suite de pruebas automatizadas para revisar el funcionamiento de la emisión en cualquier momento

### 8.4.1 Riesgo 3: el artefacto principal, el certificado de emisión, no es flexible a posteriores cambios

- se diseñó una arquitectura que permite la trazabilidad del certificado y la inclusión de nuevos campos al formulario

### 8.4.2 Riesgo 4: los datos no son accesibles o de difícil interpretación

Otro de los objetivos del sistema era facilitar la obtención de datos de laboratorio de forma estructurada. Para esto se llevaron a cabo las siguientes acciones:

- diseño de base de datos separa las tablas por áreas funcionales
- se proveyó funcionalidad de exportar datos a formato csv. En algunos casos, los registros son seleccionables de una lista y exportables en bache para posterior procesamiento.
- Se conectó la aplicación PGAdmin al servidor virtual, lo que permite en todo momento hacer consultas a la base de datos

#### **8.4.3 Riesgo 5: los datos en los formularios no son restringidos de forma adecuada y se permiten ingresar datos erróneos o inconsecuentes**

- El código de los formularios está separado en módulos. Esto permite inspeccionar su funcionamiento de forma aislada
- existen pruebas unitarias que comprueban que ciertos datos se excluyen o incluyen dependiendo de diversas condiciones (estado del lote, por ejemplo)
- El framework utilizado (Django), tiene soporte incluido para validación de formularios

#### **8.4.4 Riesgo 6: los requerimientos cambian a medida que se avanza el desarrollo y se obtiene mayor información sobre las necesidades**

- en fases tempranas del ciclo de vida hubo una etapa de análisis, donde se propuso el modelo de datos.
- con el avance del proyecto, se empezó a trabajar en un marco ágil donde se registraban las tareas en un /issue tracker/ (Jira)
- las reuniones de seguimiento fueron semanales y constante
- cuando se hizo el primer despliegue se adoptó un marco de CD (continuous delivery), lo que permitía ver actualizaciones de forma muy rápida

#### **8.4.5 Riesgo 7: cambiar los procesos de negocio que antiguamente se hacían con hojas de cálculo puede dar lugar a confusiones**

- El proceso de migración fue gradual
- se proveyeron scripts de SQL para exportar csv que emulaban el sistema documental anterior

#### **8.4.6 Riesgo 8: las componentes utilizados, tanto de software como de hardware, son insuficientes o presentan fallas**

- la selección del framework de programación fue lenta y deliberada. Se hicieron prototipos antes de iniciar el desarrollo en el framework final.
- El framework seleccionado (Django) tiene una extensa base de usuario, ampliamente documentado y probado. Es reconocido por su filosofía de diseño y confiabilidad.
- Se hizo una prueba inicial del proveedor del Servidor Virtual (IaaS) desde el 6 de Junio de 2021, antes de iniciar la primera operación el 9 de septiembre de 2021

#### **8.4.7 Riesgo 9: la interfaz gráfica no es inmediatamente comprensible o estéticamente desactualizada**

- la arquitectura del software permite cambiar la hoja de estilos
- el lenguaje de plantillas de Django permitirá alterar estructura visible de la aplicación con relativa facilidad en un futuro

## 8.5 Estrategia de validación

La estrategia de validación consiste en 2 criterios. Si la funcionalidad se enmarca dentro de algún el componente con un nivel de riesgo alto, debe tener una prueba unitaria asociada. El segundo criterio de validación es que reciba aprobación del experto de dominio en el ambiente de producción. Así, la estrategia de validación sigue metodología SCRUM y hace uso de tarjetas o incidencias en un tablero. Si hay problemas con alguna funcionalidad, se reporta y se incluye dentro de un sprint.

Un ejemplo de un tablero de incidencias es el siguiente:

The screenshot shows the Jira interface for the 'rodam analisis' project. The 'Issues' section is active, displaying a list of bugs. The filters are set to 'Project: rodam analisis' and 'Type: Bug'. The table below represents the data shown in the screenshot.

Type	Key	Summary	Assignee	Reporter
Bug	RA-212	Las microbiólogas pueden remitir un lote de medio, pero no pueden solicitar aprobación	Jonatan Ahumada	Daniel Arias Mutis
Bug	RA-211	En la vista "ingresar producto" organizar cliente por orden alfabético	Jonatan Ahumada	Daniel Arias Mutis
Bug	RA-208	Eliminar la vista 'muestras remitidas' porque aparentemente no hace nada	Jonatan Ahumada	Daniel Arias Mutis
Bug	RA-194	Poner ciudad en la dirección del certificado de emisión	Unassigned	Daniel Arias Mutis
Bug	RA-185	En 'Muestras pendientes' se están mostrando muestras con cuadros aprobados. La gracia de la vista es mostrar muestras que no están en proceso de lectura porque su cuadro no ha sido aprobado	Unassigned	Jonatan Ahumada
Bug	RA-171	quitar botón de csv individual	Unassigned	Jonatan Ahumada
Bug	RA-169	Limitar todos el filtro de "Muestras Aprobadas" a los elementos que hay en la vista	Unassigned	Jonatan Ahumada
Bug	RA-168	en general limitar todos los filtros a los elementos que hay en la vista	Unassigned	Daniel Arias Mutis

## 8.6 Entregables

- Acceso al repositorio de código y documentación
- aplicación desplegada en servidor virtual
- adecuación del servidor virtual para backups y restauración

Este documento registra algunas operaciones comunes.

## 9.1 Sincronización entre desarrollo y produccion

Los ambientes de desarrollo y producción son distintos. Esto algunas veces puede conllevar a que las ramas de desarrollo y producción se bifurquen. Algunos cambios, denominados «en caliente», se hicieron directamente sobre la aplicación en el servidor de producción. Alguno de estos cambios luego se replicaron en desarrollo, pero algunos otros no.

Más aun, lo que se replicaron fueron hechos en diferente orden, lo que puede o no plantear un problema para las migraciones de Django. Recordemos que las migraciones de django guardan una referencia a su antecesor, por lo que el orden de las migraciones es un factor relevante a la hora de correr comandos como *manage.py migrate*

La única forma de garantizar consistencia entre las dos ramas bifurcadas es revisar el schema de la base de datos como tal. Algunos comandos útiles son:

```
pg_dump -W -p 3333 -h localhost -U postgres -d rodam -s > schema_prod_rodam_`date +  
↪ %m%d%y_%H%M`.sql
```

copia el schema de la bd de producción

similarmente, para crear un tunnel ssh

```
ssh -L 3333:127.0.0.1:5432 -M -S /tmp/db-backup-socket -fNT dev@$RODAM_IP
```