



UNIVERSIDAD EMPRESARIAL SIGLO XXI

*30 AÑOS INNOVANDO LA EDUCACIÓN*

LICENCIATURA EN INFORMÁTICA

SEMINARIO DE PRÁCTICA DE INFORMÁTICA - | INF275-11265 |

---

# SISTEMA DE GESTIÓN Y ALERTAS DE VENCIMIENTOS PARA ESTUDIO CONTABLE

---

TRABAJO PRÁCTICO N° 4 - MODULO N° 4

PRESENTADO POR:

ESTUDIANTE	<u>NOMBRE Y APELLIDO DEL ALUMNO:</u> NICOLAS CARMELO SKIBICKI
	<u>ID DEL ALUMNO:</u> VINF015102
TITULAR EXPERTO	ANA CAROLINA FERREYRA
TITULAR DISCIPLINAR	PABLO ALEJANDRO VIRGOLINI

---

FECHA DE ENTREGA: 16/11/2025

PERÍODO DE CURSADA: TUTORIAS CUATRIMESTRALES 1/25 (AGOSTO-2025 a DICIEMBRE-2025)

MODALIDAD: INDIVIDUAL

LINK A REPOSITORIOS:

[https://drive.google.com/drive/folders/1yNLhEo6m2ae-Wii\\_ztgfTqM13cfegti?usp=sharing](https://drive.google.com/drive/folders/1yNLhEo6m2ae-Wii_ztgfTqM13cfegti?usp=sharing)

<https://github.com/NicoSki/Seminario-de-practica>

# Índice

## TP N°1

Título del proyecto.....	3
Introducción.....	3
Justificación.....	3
Definiciones del proyecto y del sistema.....	4
Elicitación.....	4
Conocimiento del negocio.....	5
Propuesta de solución.....	6
Captura de requisitos.....	6
Inicio del análisis: casos de uso. ....	8

## TP N°2

Devolución TP N°1 - 19 de sep en 15:01.....	17
Etapa de análisis.....	17
Etapa de diseño.....	19
Etapa de implementación.....	20
Etapa de pruebas.....	21
Definición de base de datos para el sistema.....	24
Diagrama entidad-relación de la base de datos.....	24
Creación de las tablas MySQL.....	25
Inserción, consulta y borrado de registros.....	28
Presentación de las consultas SQL.....	32

## TP N°3

Devolución TP N°2 - 07 de oct en 10:34.....	32
Explicación del desarrollo en Java.....	32
Presentación del desarrollo en Java.....	33
El programa compila y se ejecuta correctamente.....	34

**TP N°4**

Selección del patrón de diseño y justificación.....	36
Explicación del desarrollo, usando un patrón MVC.....	37
Presentación del desarrollo en Java.....	40
Presentación del proyecto en un video de 3 minutos.....	41
Bibliografía.....	42

## Título del proyecto:

*Sistema de gestión y alertas de vencimientos para estudio contable.*

## Introducción:

El presente proyecto de sistema se desarrolla frente a un contexto de un estudio contable que requiere optimizar de forma efectiva la gestión de vencimientos fiscales. Dado que actualmente, el estudio contable maneja de forma manual las fechas críticas de ciertos impuestos, entre ellos, impuesto al valor agregado (IVA), ingresos brutos (IIBB), seguridad e higiene (SeH). Generando riesgos de mora y estrés operativo.

### Antecedentes:

- La obtención de los vencimientos de los diferentes impuestos es por medio de un proceso manual en donde se ingresan a los organismos oficiales: ARCA, ARBA, Sitios web de municipios
- La falta de un proceso automatizado ha generado ocasionales olvidos y multas. Impactando directamente en la relación del estudio contable con sus clientes
- El estudio contable no cuenta con una base de datos unificada que centralice aquellas obligaciones fiscales de los clientes según el tipo de régimen del contribuyente: Autónomo (Responsable Inscripto) o Monotributista.

El proyecto pretende desarrollar un sistema web que agilice/automaticice el recordatorio de los vencimientos de los diferentes impuestos, mejore la precisión y permita al estudio contable reemplazar los tiempos de búsquedas por actividades de mayor valor.

## Justificación:

### *Razones para el desarrollo del sistema:*

- Necesidades cubiertas:
  - Suprimir el error humano en la verificación de los vencimientos
  - Centralizar las diferentes obligaciones tributarias de la cartera de clientes
  - Minimizar el riesgo de sanción por mora
- Contribución con tiempos modernos:
  - Reemplazar un proceso manual por una automatización
  - Posibilidad de integrar/vincular APIs de los organismos tributarios con el sistema
- Puntos clave para el estudio contable:
  - Ahorro de costos: Evitar multas sobre los impuestos vencidos
  - Ser más eficientes: Reducir tiempos muertos de búsquedas
  - Competitividad: Aumenta la calidad y confiabilidad del servicio frente a la cartera de clientes y clientes futuros

## **Definiciones del proyecto y del sistema:**

### *Objetivo del proyecto:*

Desarrollar un sistema para gestionar y notificar de forma proactiva los diferentes vencimientos de los impuestos (IVA - IBB - SeH) para un estudio contable

### *Objetivo general del sistema:*

Automatizar el registro, seguimiento y alerta de los vencimientos fiscales de los impuestos, en donde se garantice la puntualidad en las presentaciones

### **Alcances del sistema:**

- Incluye:
  - Registro de clientes y los impuestos asociados
  - Calendarización de vencimientos de los impuestos
  - Alertas informativas vía email
  - Sistema de reportes
- No incluye:
  - Presentaciones automáticas de los impuestos
  - Integración directa con los organismos gubernamentales
  - Gestión de pagos de los impuestos

## **Elicitación:**

### *Técnicas a aplicar:*

- Entrevistas con el contador y empleados del estudio: La finalidad de este tipo de elic平ación es comprender la dinámica y las prácticas diarias del personal del estudio para comprender más a fondo los puntos de mejora
- Observaciones de procesos actuales: Tener conocimientos del paso a paso de los diferentes procesos para la liquidación de los diferentes impuestos, entre ellos, el análisis de las planillas utilizadas (Excel puede ser un ejemplo), conocer los recordatorios actuales con los que cuenta el estudio (Marcas de fechas de vencimiento en calendarios físicos)
- Historial de multas por mora: Esta información será de utilidad para realizar comparaciones entre el estudio contable actual vs el estudio contable luego de la implementación del sistema a desarrollar
- Cuestionarios: Este medio de elic平ación será útil para dimensionar los diferentes comportamientos de los empleados del estudio. Preguntas útiles para estos casos serían:
  - Cuantas veces entras a verificar los vencimientos de los impuestos?
  - Cuanto tiempo te toma esta acción?
  - Si un cliente te solicita un vencimiento, de donde obtienes los datos?

Cabe destacar que en este tipo de prácticas optaremos por preguntas cerradas y abiertas

*Hallazgos posteriores a la elicitación:*

- El 100% de la liquidación de los impuestos se gestionan de forma manual
- Se identifico que el 20% de los impuestos liquidados por el estudio contable son emitidos posteriormente a sus fechas de vencimientos
- Necesidad de alertas previas a fechas de vencimientos de impuestos (4 días antes del vencimiento se debe emitir una alerta vía email)
- Requerimiento indispensable:
  - Diferenciar los vencimientos por jurisdicción, terminación de CUIT del cliente ya que cada impuesto se rige por normas diferentes
- Planillas de datos locales:
  - Cada empleado cuenta con un porcentaje de la cartera de clientes del estudio contable, por lo que la información almacenada se encuentra únicamente en el ordenador de cada empleado

**Conocimiento del negocio:**

*Proceso actual de liquidación de impuestos:*

1. Solicitud de facturación mensual al cliente
2. División y reorganización de compras/ventas del cliente
3. Carga manual en planilla de Excel
4. Revisión semanal de planilla y sitios webs de los organismos para verificar vencimientos próximos
5. Carga manual de vencimientos según orden de aparición en los sitios webs de los organismos
6. Liquidación de impuesto según urgencia
7. Carga manual optativa respecto a la fecha de emisión/liquidación del correspondiente impuesto
8. Envío de impuesto al cliente según el tipo de pago deseado por el mismo (Pago manual o pago electrónico)

*Aspectos a considerar:*

- Una vez obtenidos los vencimientos de los impuestos, no existe ningún aviso cercano a su fecha de vencimiento
- Error humano en la carga de fechas de vencimientos
- No se consideran los feriados nacionales a la hora de la carga de datos
- Control nulo de presentaciones & envío de impuestos a clientes
- Comunicación no efectiva entre colegas (Como consecuencia de los almacenamientos locales de las planillas)

## **Propuesta de solución:**

### *Propuesta funcional:*

- El sistema debe contar con un login con credenciales especiales, cada usuario tendrá acceso a diferentes secciones del sistema:
  - Rol para el empleado: este rol permitirá realizar acciones relacionadas al CRUD de clientes e impuestos, también tendrá la posibilidad consultar diferentes reportes con filtros particulares
  - Rol para el contador: este rol permitirá al contador visualizar reportes relacionados con los diferentes impuestos de su cartera de clientes
- Carga automática de vencimientos para los diferentes impuestos por medio de web scraping
- Alertas automáticas por medio de email tanto para el empleado como para el contador 4 días antes de los correspondientes vencimientos de los impuestos

### *Propuesta técnica:*

- ✓ Frontend: HTML, CSS
- ✓ Backend: JAVA para la lógica del negocio
- ✓ Base de datos: MySQL
- ✓ Alertas: Librería para envíos de correos electrónicos compatibles con JAVA

La conexión de red para los ordenadores de los empleados, los cuales llamaremos “administradores” en el siguiente apartado, serán por medio de un cable Ethernet. Lo cual brinda confiabilidad y eficiencia para tanto para las gestiones de los clientes como los correspondientes envíos de alertas por vía mail

## **Captura de requisitos:**

### Requerimientos funcionales:

Nº de requerimiento	Descripción
RF-01	El sistema deberá permitir al contador registrar un nuevo usuario, al cual se le asignaran credenciales de acceso
RF-02	El sistema deberá permitir al contador remover las credenciales de acceso al sistema
RF-03	El sistema deberá permitir al actor ingresar al portal del sistema mediante una autenticación usuario/contraseña
RF-04	El sistema deberá permitir al administrador registrar un nuevo cliente
RF-05	El sistema deberá permitir al administrador modificar un cliente
RF-06	El sistema deberá permitir al administrador eliminar un cliente
RF-07	El sistema deberá permitir al administrador consultar un cliente
RF-08	El sistema deberá permitir al administrador listar los clientes
RF-09	El sistema deberá permitir al administrador registrar una nueva obligación tributaria
RF-10	El sistema deberá permitir al administrador modificar una obligación tributaria

RF-11	El sistema deberá permitir al administrador eliminar una obligación tributaria
RF-12	El sistema deberá permitir al administrador consultar una obligación tributaria
RF-13	El sistema deberá permitir al administrador listar las obligaciones tributarias
RF-14	El sistema deberá permitir al administrador asociar un cliente con una o más obligaciones tributarias
RF-15	El sistema deberá permitir al administrador eliminar la asociación de un cliente con una obligación tributaria
RF-16	El sistema deberá permitir al administrador listar los vencimientos de las obligaciones tributarias
RF-17	El sistema deberá permitir al administrador listar a los clientes por tipo de obligación tributaria
RF-18	El sistema deberá permitir al administrador acceder a un calendario con los feriados nacionales de forma anual
RF-19	El sistema deberá enviar un mail al administrador 4 días antes del vencimiento de una obligación tributaria
RF-20	El sistema deberá permitir al administrador imprimir un reporte

Requerimientos no funcionales:

Nº de requerimiento	Descripción
RNF-01	El sistema deberá correr en un sistema operativo Windows 10 o superior
RNF-02	El sistema deberá contar con el lenguaje de programación “JAVA” para su implementación
RNF-03	El sistema deberá operar en un equipo que tenga como mínimo 1 terabyte de memoria
RNF-04	El sistema deberá estar operativo en su 99% en los horarios laborales
RNF-05	El sistema deberá contar con firewall para proteger los datos sensibles de los clientes
RNF-06	El sistema deberá contar con el almacenamiento de la información en una base de datos relacional (MySQL)
RNF-07	El sistema debe ser instalado en todos los ordenadores del estudio contable
RNF-08	El sistema debe contar con el patrón de diseño MVC (Modelo – vista - controlador) para la persistencia y consulta de los datos en la base de datos seleccionada

Requerimientos candidatos:

Nº de requerimiento	Descripción
RC-01	Desarrollo de aplicación móvil para gestión de alertas. Recopilar las funcionalidades que se encuentran en el sistema para la PC y adaptarlo a un dispositivo móvil

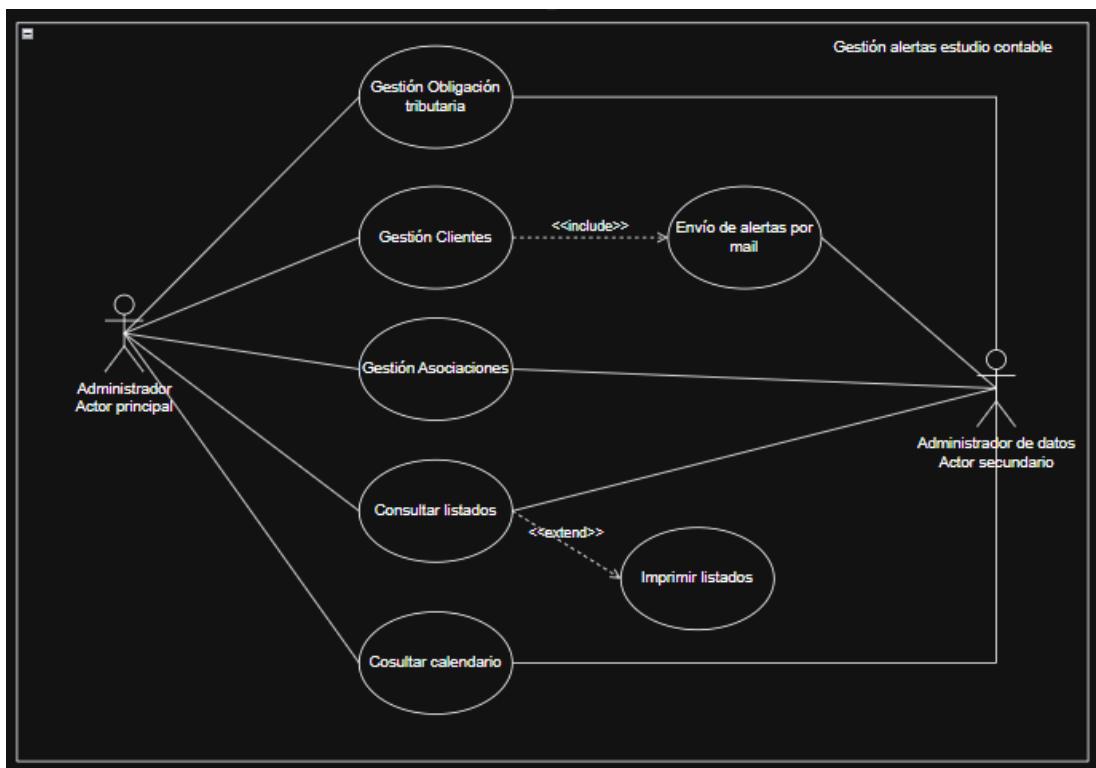
## Inicio del análisis: Casos de uso

*Aclaración:* Por cuestiones de prolijidad y visualización más clara del diagrama de casos de uso, opte por dividir los diagramas en dos partes. Por un lado, la parte del administrador y por otro la parte del contador. Cabe destacar que, a la hora de una presentación profesional, se los debe mostrar en el mismo diagrama:

### Identificación de autores:

- **Administrador:** Es uno de los usuarios principales del sistema, actor humano, es el responsable de la gestión integral de la información tributaria.
- **Contador:** Usuario con privilegios administrativos, actor humano, enfocado en la gestión de accesos al sistema
- **Administrador de datos:** Actor no humano que se encarga de ejecutar los procesos basados en las reglas predefinidas

### Administrador:



**Contador:****Casos de uso:**

Requerimiento	Caso de uso	Actor principal	Descripción
RF-01	CU001	Contador	Registrar un nuevo usuario
RF-02	CU002	Contador	Eliminar un usuario
RF-04	CU003	Administrador	Registrar un nuevo cliente
RF-05	CU004	Administrador	Modificar un cliente
RF-06	CU005	Administrador	Eliminar un cliente
RF-07	CU006	Administrador	Consultar un cliente
RF-08	CU007	Administrador	Listar clientes
RF-09	CU008	Administrador	Registrar obligación tributaria
RF-10	CU009	Administrador	Modificar obligación tributaria
RF-11	CU010	Administrador	Eliminar obligación tributaria
RF-12	CU011	Administrador	Consultar obligación tributaria
RF-13	CU012	Administrador	Listar obligaciones tributarias
RF-14	CU013	Administrador	Asociar una obligación tributaria a un cliente
RF-15	CU014	Administrador	Eliminar una obligación tributaria de un cliente
RF-20	CU015	Administrador	Imprimir reporte
RF-16	CU016	Administrador	Consultar calendario con vencimientos

**Descripción de casos de usos:**

<b>Caso de uso</b>	CU001
<b>Actores</b>	Contador, Administrador de datos
<b>Referencias</b>	RF-001
<b>Descripción</b>	Registrar un nuevo usuario en el sistema, asignándole nuevas credenciales
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El contador inicio la sesión en el sistema</li> <li>2. El contador cuenta con los privilegios administrativos del sistema</li> </ol>
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El contador ingresa al apartado “Registrar empleado”</li> <li>2. El sistema proyecta un nuevo formulario de registro</li> <li>3. El contador completa el formulario con: <ul style="list-style-type: none"> <li>• Nombre del empleado</li> <li>• Apellido del empleado</li> <li>• Email del empleado</li> <li>• Nombre de usuario del empleado</li> </ul> </li> <li>4. El contador selecciona el tipo de rol para el empleado</li> <li>5. El contador confirma el registro de los datos/privilegios del empleado en el sistema</li> <li>6. El sistema almacena el registro en la base de datos y le otorga los privilegios al empleado</li> <li>7. El sistema confirma el nuevo registro por medio de un mensaje en pantalla</li> </ol>
<b>Postcondición</b>	Se ha registrado un nuevo usuario para poder utilizar las funcionalidades del sistema
<b>Flujo alternativo</b>	<p>Validación fallida:</p> <ol style="list-style-type: none"> <li>1. El sistema informa al contador la falta de datos previo al registro</li> <li>2. El sistema vuelve al paso N° 3 y remarca las casillas de información faltante</li> </ol>
<b>Excepciones</b>	<p>E1: Se interrumpió la conexión con la base de datos</p> <p>E2: El email del nuevo usuario ya se encuentra registrado</p>

<b>Caso de uso</b>	CU002
<b>Actores</b>	Contador, Administrador de datos
<b>Referencias</b>	RF-02
<b>Descripción</b>	Remover un usuario del sistema, quitarle las credenciales de acceso
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El contador inicio la sesión en el sistema</li> <li>2. El contador cuenta con los privilegios administrativos del sistema</li> </ol>
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El contador ingresa al apartado “Empleados”</li> <li>2. El sistema proyecta un listado con todos los empleados registrados</li> <li>3. El contador selecciona el botón “Eliminar”</li> <li>4. El contador confirma la eliminación del empleado</li> </ol>
<b>Postcondición</b>	Se han removido los privilegios del empleado para utilizar las funcionalidades del sistema
<b>Flujo alternativo</b>	<p>Validación fallida:</p> <ol style="list-style-type: none"> <li>1. El sistema informa al contador que no se logró eliminar los privilegios del empleado</li> <li>2. El sistema vuelve al paso N° 2</li> </ol>
<b>Excepciones</b>	No aplica

<b>Caso de uso</b>	CU003
<b>Actores</b>	Administrador, Administrador de datos
<b>Referencias</b>	RF-04
<b>Descripción</b>	Registrar un nuevo cliente en el sistema
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El administrador inicia la sesión en el sistema</li> <li>2. El administrador debe contar con los privilegios administrativos del sistema</li> </ol>
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El administrador ingresa al apartado “Nuevo cliente”</li> <li>2. El administrador completa el formulario con: <ul style="list-style-type: none"> <li>• Razón social del cliente</li> <li>• CUIT del cliente</li> <li>• Email del cliente</li> <li>• Dirección del cliente</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>• Tipo de contribuyente del cliente</li> <li>• Impuestos del cliente</li> <li>• Inicio de actividad del cliente</li> <li>• Actividad del cliente</li> </ul> <ol style="list-style-type: none"> <li>3. El administrador confirma el registro de los datos cliente en el sistema</li> <li>4. El sistema almacena el registro en la base de datos</li> <li>5. El sistema confirma el nuevo registro por medio de un mensaje en pantalla</li> </ol>
<b>Postcondición</b>	Se ha registrado un nuevo cliente en el sistema
<b>Flujo alternativo</b>	<p>Validación fallida:</p> <ol style="list-style-type: none"> <li>1. El sistema informa al administrador la falta de datos previo al registro</li> <li>2. El sistema vuelve al paso N° 2 y remarca las casillas de información faltante</li> </ol>
<b>Excepciones</b>	<p>E1: Se interrumpió la conexión con la base de datos</p> <p>E3: El CUIT del cliente ya se encuentra registrado</p>

<b>Caso de uso</b>	CU004
<b>Actores</b>	Administrador, Administrador de datos
<b>Referencias</b>	RF-05
<b>Descripción</b>	Modificar un cliente en el sistema
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El administrador inicia la sesión en el sistema</li> <li>2. El administrador debe contar con los privilegios administrativos del sistema</li> </ol>
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El administrador ingresa al apartado “Clientes”</li> <li>2. El sistema proyecta un listado con todos los clientes registrados</li> <li>3. El administrador selecciona el botón “Editar” en la fila del cliente deseado</li> <li>4. El administrador realiza las modificaciones correspondientes</li> <li>5. El administrador confirma las modificaciones</li> </ol>
<b>Postcondición</b>	Se han modificado los datos del cliente
<b>Flujo alternativo</b>	Validación fallida:

	<ol style="list-style-type: none"> <li>1. El sistema informa al administrador la falta de datos previo al registro</li> <li>2. El sistema vuelve al paso N° 2</li> </ol>
<b>Excepciones</b>	<p>E1: Se interrumpió la conexión con la base de datos</p> <p>E3: El CUIT del cliente ya se encuentra registrado</p>

<b>Caso de uso</b>	CU005
<b>Actores</b>	Administrador, Administrador de datos
<b>Referencias</b>	RF-006
<b>Descripción</b>	Eliminar un cliente del sistema
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El administrador inicia la sesión en el sistema</li> <li>2. El administrador debe contar con los privilegios administrativos del sistema</li> </ol>
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El administrador ingresa al apartado “Clientes”</li> <li>2. El sistema proyecta un listado con todos los clientes registrados</li> <li>3. El administrador selecciona el botón “Eliminar” en la fila del cliente deseado</li> <li>4. El administrador confirma la eliminación del cliente</li> </ol>
<b>Postcondición</b>	Se ha eliminado al cliente del sistema
<b>Flujo alternativo</b>	<p>Validación fallida:</p> <ol style="list-style-type: none"> <li>1. El sistema informa al administrador que no se logró eliminar al cliente</li> <li>2. El sistema vuelve al paso N° 2</li> </ol>
<b>Excepciones</b>	E1: Se interrumpió la conexión con la base de datos

<b>Caso de uso</b>	CU006
<b>Actores</b>	Administrador, Administrador de datos
<b>Referencias</b>	RF-07
<b>Descripción</b>	Consultar un cliente del sistema
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El administrador inicia la sesión en el sistema</li> <li>2. El administrador debe contar con los privilegios administrativos del sistema</li> </ol>

<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El administrador ingresa al apartado “Clientes”</li> <li>2. El sistema proyecta un listado con todos los clientes registrados</li> <li>3. El administrador selecciona el botón “Detalle” en la fila del cliente deseado</li> <li>4. El administrador accede a un overview de la información del cliente</li> </ol>
<b>Postcondición</b>	Se han verificado los datos del cliente en el sistema
<b>Flujo alternativo</b>	Validación fallida: <ol style="list-style-type: none"> <li>1. El sistema informa al administrador que no se logró acceder a la información del cliente</li> <li>2. El sistema vuelve al paso N° 2</li> </ol>
<b>Excepciones</b>	E1: Se interrumpió la conexión con la base de datos

<b>Caso de uso</b>	CU007
<b>Actores</b>	Administrador, Administrador de datos
<b>Referencias</b>	RF-08
<b>Descripción</b>	Listar clientes del sistema
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El administrador inicia la sesión en el sistema</li> <li>2. El administrador debe contar con los privilegios administrativos del sistema</li> </ol>
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El administrador ingresa al apartado “Clientes”</li> <li>2. El sistema proyecta un listado con todos los clientes registrados</li> </ol>
<b>Postcondición</b>	Se han listado todos los clientes en el sistema
<b>Flujo alternativo</b>	Validación fallida: <ol style="list-style-type: none"> <li>1. El sistema informa al administrador que no se logró acceder al listado completo de los clientes</li> </ol>
<b>Excepciones</b>	E1: Se interrumpió la conexión con la base de datos

**Aclaración:** Cabe destacar que, para los siguientes casos de uso, se aplicaría la misma lógica:

- CU008
- CU009
- CU010
- CU011
- CU012

<b>Caso de uso</b>	CU015
<b>Actores</b>	Administrador, Administrador de datos
<b>Referencias</b>	RF-20
<b>Descripción</b>	Imprimir un reporte
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El administrador inicia la sesión en el sistema</li> <li>2. El administrador debe contar con los privilegios administrativos del sistema</li> <li>3. El administrador debe tener acceso a una impresora láser o de tinta</li> <li>4. La conexión de la impresora al ordenador del administrador debe ser por medio de un cable USB o por vía bluetooth</li> </ol>
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El administrador ingresa al apartado “Reportes”</li> <li>2. El administrador debe seleccionar el tipo de reporte que desee imprimir</li> <li>3. El administrador puede optar por aplicar filtros al reporte</li> <li>4. El administrador debe confirmar la orden de impresión para ser ejecutada</li> </ol>
<b>Postcondición</b>	Se ha impreso el reporte seleccionado por el administrador
<b>Flujo alternativo</b>	<p>Validación fallida:</p> <ol style="list-style-type: none"> <li>1. El sistema informa al administrador que no se logró imprimir el listado seleccionado</li> <li>2. El sistema vuelve al paso N°2</li> </ol>
<b>Excepciones</b>	E1: Se interrumpió la conexión con la base de datos

<b>Caso de uso</b>	CU016
<b>Actores</b>	Administrador, Administrador de datos
<b>Referencias</b>	RF-16
<b>Descripción</b>	Consultar calendario de vencimientos
<b>Precondición</b>	1. El administrador inicia la sesión en el sistema 2. El administrador debe contar con los privilegios administrativos del sistema
<b>Flujo principal</b>	1. El administrador ingresa al apartado “Calendario” 2. El administrador selecciona el tipo de obligación tributaria que desea consultar 3. El administrador confirma la consulta
<b>Postcondición</b>	Se han consultado los vencimientos de las obligaciones tributarias
<b>Flujo alternativo</b>	Validación fallida: 1. El sistema informa al administrador que no se lograron mostrar los vencimientos 2. El sistema vuelve al paso N°2
<b>Excepciones</b>	E1: Se interrumpió la conexión con la base de datos

## TP N°2:

### Devolución TP N°1:

Comienzo la resolución del correspondiente TP haciendo énfasis en el feedback mencionado en la fecha **19 de sep en 15:01, luego de la corrección del TP N°1:**

1. LOS TITULARES SOMOS AL REVÉS → Los titulares fueron invertidos en la portada
2. INCLUIR TÍTULO EN PORTADA → Titulo añadido en la portada
3. Propuesta de solución:

Voy a ampliar dicha sección partiendo desde el contenido detallado en “Propuesta de solución” en la página N° 6:

- Carga automática de vencimientos: El sistema debe obtener de forma dinámica y automatizada los diferentes vencimientos de las obligaciones tributarias que liquida el

estudio contable. Este proceso se realizará por medio de un método conocido como “Web Scraping” o también conocido como “Raspar la web” para capturar las diferentes etiquetas del esqueleto del sitio web (HTML) y obtener los datos que se relacionan con los vencimientos del impuesto en cuestión

- Alertas automáticas: Por medio de una librería compatible con JAVA, se enviarán notificaciones vía email con 4 días de anticipación previas a cada vencimiento para que los administradores del sistema puedan presentar las correspondientes obligaciones tributarias sin sufrir afectar su fecha de caducidad.
- Conocimientos del negocio: Cabe destacar que todas las liquidaciones de las obligaciones tributarias de los diferentes clientes son efectuadas en la modalidad “a mes vencido”. Esto quiere decir que se liquidan con un mes de diferencia, ejemplo práctico:
  - Un administrador debe liquidar la obligación tributaria de los IIBB (Ingresos Brutos) de un cliente del periodo de Octubre/25, pues la liquidación de la correspondiente obligación se va a efectuar en el mes de Noviembre/25.

### **Etapa de análisis:**

Aclaración: Tal como se aclaró con el titular experto, me voy a enfocar en el desarrollo de la parte funcional del sistema, es decir, en el desarrollo de las notificaciones del sistema.

A su vez, cabe destacar que se toman como práctica básica el desarrollo de cada uno de los casos del CRUD para los diferentes casos de usos, entre ellos, las obligaciones tributarias, los clientes y los usuarios. Tal como se mencionó anteriormente, el foco no estará en el desarrollo de los mismo.

El desarrollo se centrará en siguiente RF:

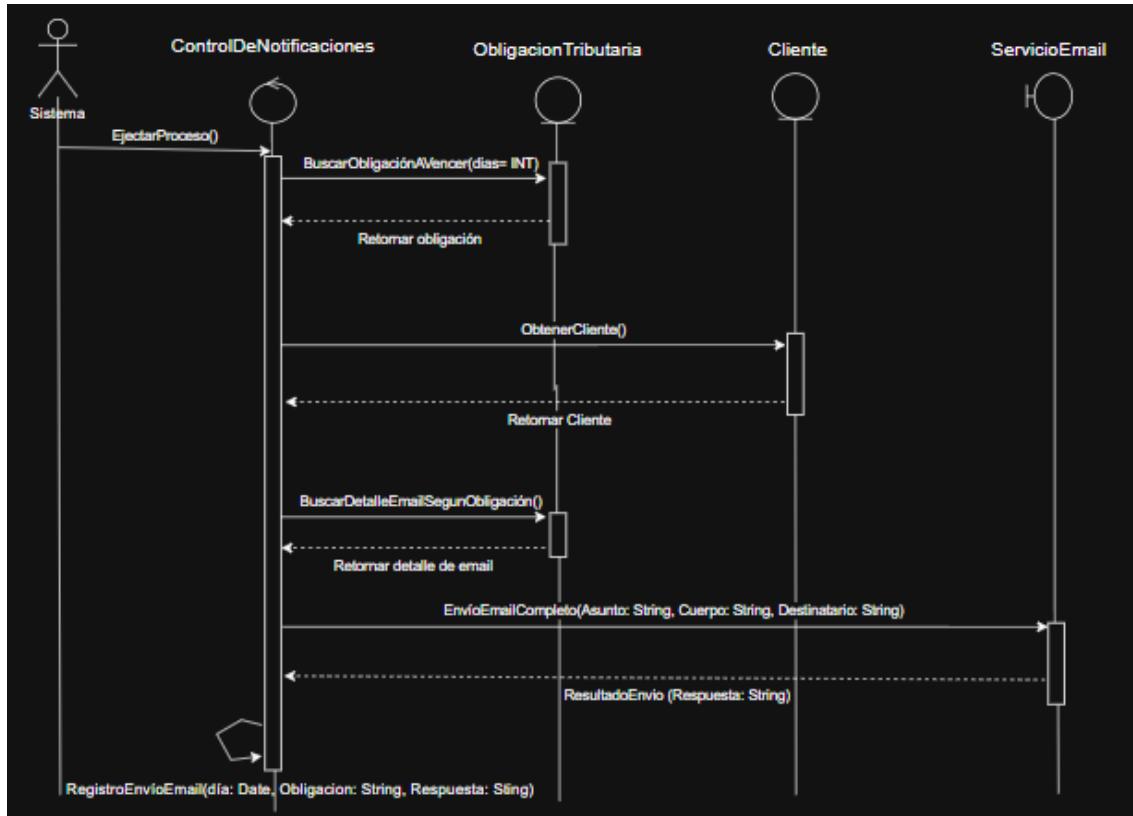
Requerimiento Funcional	Descripción
RF-19	El sistema deberá enviar un mail al administrador 4 días antes del vencimiento de una obligación tributaria

Detalles previos al desarrollo del diagrama de secuencia:

### **Clases:**

- Control:
  - ControlDeNotificaciones: Esta clase será la encargada de dirigir el completo proceso de la notificación
- Entidad:
  - ObligacionTributaria: Tipo de impuesto al que se le desea consultar su vencimiento

- Cliente: Se refiere al cliente del estudio contable
- Interfaz:
  - ServicioEmail: Abstracción del sistema para el envío de los emails



Explicación del diagrama de secuencia:

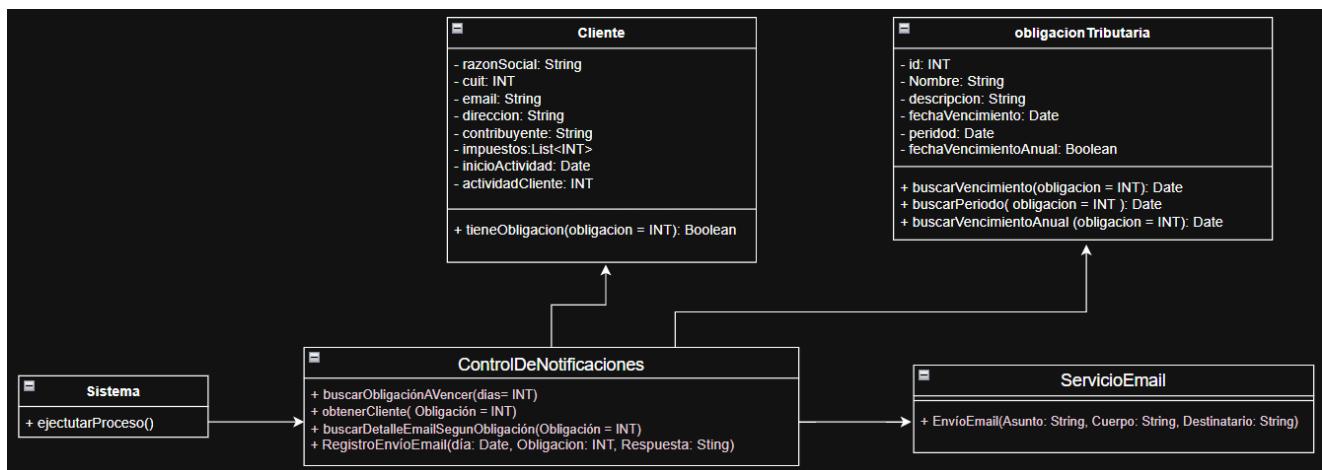
- 1) EjecutarProceso(): Es aquella función que inicia el proceso
- 2) ControlDeNotificaciones – ObligacionTributaria: Se ejecuta una consulta por medio de la función BuscarObligaciónAVencer(días= INT). Esta función cuenta con un parámetro contante, cuyo valor será el N° 4, el cual hace referencia a la cantidad de días previos antes del vencimiento de la obligación, esta función devolverá aquella obligación tributaria que se encuentre en el plazo mas cercano a su vencimiento
- 3) ControlDeNotificaciones – Cliente: Una vez obtenido el tipo de obligación, se procede a consultar quienes son los clientes que se les debe realizar la correspondiente liquidación de la obligación tributaria. Esta consulta se realiza por medio de la función, ObtenerCliente(), la cual nos retornan todos los clientes involucrados en dicha liquidación
- 4) ControlDeNotificaciones – ObligacionTributaria: En esta etapa se extrae el cuerpo del email para luego ser enviado en la notificación. Este proceso se lleva a cabo por la función BuscarDetalleEmailSegunObligación()
- 5) ControlDeNotificaciones – ServicioEmail: Una vez obtenido todos los datos necesarios, se procede al envío de la notificación para todos los administradores del sistema (En este caso serían los empleados del estudio contable) a través de la función EnvíoEmailCompleto(Asunto: String, Cuerpo: String, Destinatario: String, Vencimiento:

Date). Este proceso devuelve una respuesta de éxitos o falla según la condición del envío de la notificación

- 6) ControlDeNotificaciones – ControlDeNotificaciones: Por último, se almacena un registro con la respuesta obtenida a la hora del envío de la notificación. En este caso se utiliza la función: RegistroEnvíoEmail(día: Date, Obligacion: String, Respuesta: String)

Cabe destacar que con el correspondiente desarrollo del diagrama de secuencia estamos contribuyendo con alta cohesión, bajo acoplamiento, mantenibilidad y escalabilidad del sistema.

### Etapa de diseño:



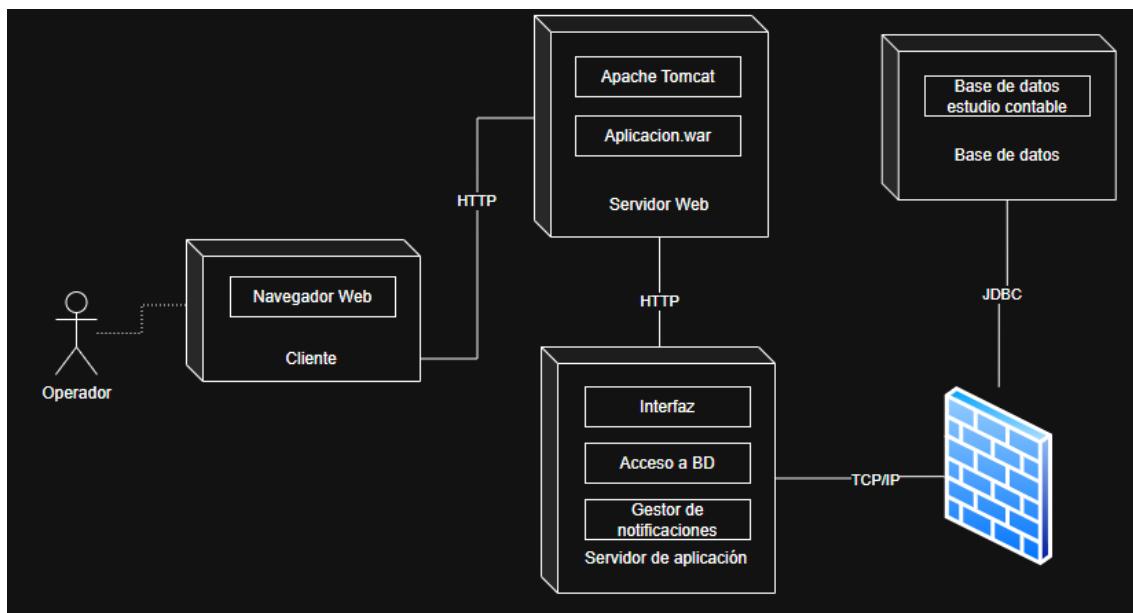
### Etapa de implementación:

Continuo el desarrollo de la etapa de implementación remarcando los requerimientos del sistema:

Nº de requerimiento	Descripción
RNF-01	El sistema deberá correr en un sistema operativo Windows 10 o superior
RNF-02	El sistema deberá contar con el lenguaje de programación “JAVA” para su implementación
RNF-03	El sistema deberá operar en un equipo que tenga como mínimo 1 terabyte de memoria
RNF-04	El sistema deberá estar operativo en su 99% en los horarios laborales
RNF-05	El sistema deberá contar con firewall para proteger los datos sensibles de los clientes
RNF-06	El sistema deberá contar con el almacenamiento de la información en una base de datos relacional (MySQL)
RNF-07	El sistema debe ser instalado en todos los ordenadores del estudio contable

RNF-08	El sistema debe contar con el patrón de diseño MVC (Modelo – vista - controlador) para la persistencia y consulta de los datos en la base de datos seleccionada
--------	---

### Diagrama de despliegue:



Añado algunos conceptos que serán de utilidad para las rutas de comunicación:

- HTTP: Protocolo estándar para que el usuario pueda acceder al sistema por medio de un navegador web
- TCP/IP: Base de comunicación para los dispositivos entre sí en una red privada o en internet. Esencial para la seguridad y conectividad
- JDBC: Estándar para conectar aplicaciones JAVA con base de datos relacionales

### Etapa de pruebas:

Para el desarrollo de etapa correspondiente, voy a presentar un plan de pruebas básico para el requerimiento que se detalla a continuación:

Requerimiento Funcional	Descripción
RF-19	El sistema deberá enviar un mail al administrador 4 días antes del vencimiento de una obligación tributaria

- Plan de pruebas:

Requerimiento Funcional	Código de prueba	Tipo de prueba	Técnica propuesta	Observaciones
RF-19	CPC01	Componente	Cobertura	Aplicación de caja blanca al método <code>bascarObligacionesAVencer()</code>
RF-19	CPC02	Componente	Partición de equivalencia-frontera	Aplicación de caja negra al método <code>bascarObligacionesAVencer()</code>
RF-19	CPI03	Integración	Pruebas de flujo completo	Verificación de integración con BD y servicio de email
RF-19	CPS04	Sistema	Pruebas funcionales	Verificación de los requerimientos funcionales

- Caso de pruebas:

Basaremos nuestro caso de prueba al código de prueba CPC02 en donde evaluaremos:

Requerimientos relacionados:

**Aclaración:** Cabe destacar que aparecen nuevos requerimientos para el desarrollo de este apartado, los cuales son continuación de los RF definidos previamente en la pagina N°6

Requerimiento	Descripción
RF-21	El sistema no debe enviar la notificación vía email si el plazo previo a cada obligación es mayor o menor a 4 días
RF-22	El sistema no debe enviar la notificación vía email si el vencimiento de la obligación ya pasó
RF-23	El sistema debe registrar la respuesta del envío de la notificación (Exitoso/Fallido)

- Partición de equivalencias:

**Equivalencia válida:**

Clase	Rango	Comportamiento
CEV1	Días = 4	Envío éxitos de notificación vía email

**Equivalencias inválidas:**

Clase	Rango	Comportamiento
CEI1	Días < 4	No se envía la notificación, mucha anticipación al vencimiento
CEI2	Días > 4	No se envía la notificación

- Análisis de valores frontera:

Clase de equivalencias	Límite de frontera	Límite inferior	Límite superior
CEV1	4	3	5
CEI1	0	-1	1
CEI2	6	5	7

- Comportamiento del sistema:

Dato de entrada	Comportamiento	Mensaje del sistema
-1	No se envía la notificación	No se puede enviar un numero en negativo
0	No se envía la notificación	La obligación vence hoy
1	No se envía la notificación	Falta 1 día para el vencimiento de la obligación
2	No se envía la notificación	Faltan 2 días para el vencimiento de la obligación
3	No se envía la notificación	Faltan 3 días para el vencimiento de la obligación
4	Se envía la notificación	Se envía la notificación con una anticipación de 4 días previos al vencimiento de la obligación
5	No se envía la notificación	Faltan 5 días para el vencimiento de la obligación
6	No se envía la notificación	Faltan 6 días para el vencimiento de la obligación

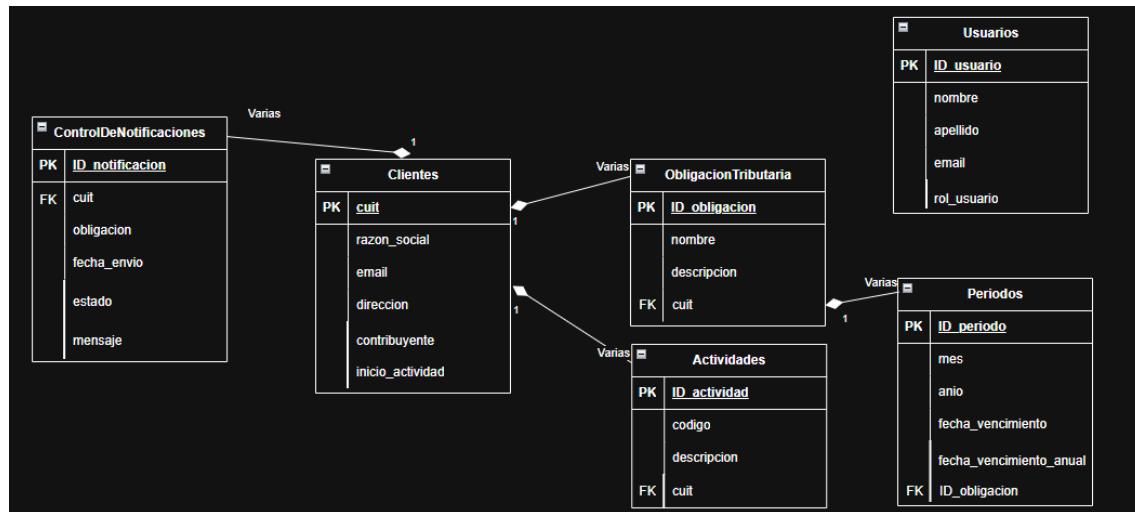
**Definición de base de datos para el sistema:**

Para llevar a cabo la funcionalidad principal sobre el envío de notificaciones del sistema, se debe contar con una base de datos relacional como lo puede ser MySQL, a su vez, debe ser la misma que acompaña a los requerimientos no funcionales definidos anteriormente. Estos tipos de prácticas se emplean para contar con una mayor flexibilidad, rendimiento y escalabilidad.

El uso correspondiente de la BD es fundamental para la persistencia de los diferentes datos, como los pueden ser los clientes, usuarios del sistema, las obligaciones tributarias, ya que estos datos serán de utilidad para compararlos y obtener resultados que le darán la vida funcional al sistema.

Contar con el almacenamiento de los datos permitirá organizar y tener mas claridad no solo en la cantidad de clientes del estudio contable, sino que también contar con información relevante de los mismo, tener conocimiento y anticiparse ante los diferentes tipos de obligaciones tributarias que se les deben liquidar. Sumado a ello, contar con la posibilidad de realizar diferentes reportes para consultas específicas y casos opcionales de impresión por si se desea contar con la información en un formato físico.

### Diagrama entidad-relación de la base de datos:



### Creación de las tablas MySQL:

Luego de definir el diagrama entidad-relación del apartado anterior, continuo el desarrollo de la creación de las respectivas tablas en la BD seleccionada:

**Tabla CLIENTES:**

```
CREATE TABLE CLIENTES (
    cuit INT PRIMARY KEY NOT NULL,
    razon_social VARCHAR(255) NOT NULL,
    email VARCHAR(255),
    direccion VARCHAR(255),
    contribuyente VARCHAR(50) NOT NULL,
    inicio_actividad DATE NOT NULL
);
```

**Tabla OBLIGACIONTRIBUTARIA:**

```
CREATE TABLE OBLIGACIONTRIBUTARIA (
    id_obligacion INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(255) NOT NULL,
    descripcion VARCHAR(255),
    cuit INT
);
```

**Tabla USUARIOS:**

```
CREATE TABLE USUARIOS (
    id_usuario INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    email VARCHAR(255) NOT NULL,
    rol VARCHAR(50) NOT NULL
);
```

**Tabla ACTIVIDADES:**

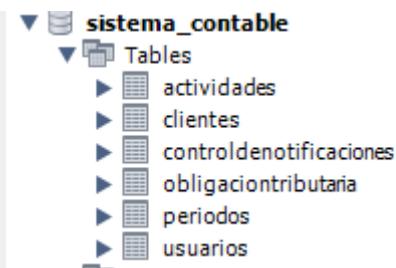
```
CREATE TABLE ACTIVIDADES (
    id_actividad INT PRIMARY KEY AUTO_INCREMENT,
    codigo int NOT NULL,
    descripcion VARCHAR(255),
    cuit INT
);
```

**Tabla PERIODOS:**

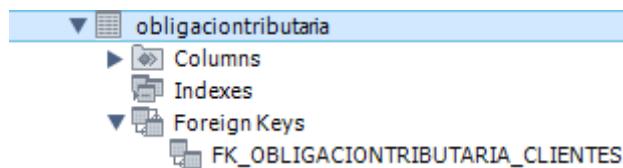
```
CREATE TABLE PERIODOS (
    id_periodo INT PRIMARY KEY AUTO_INCREMENT,
    #Verifico que los periodos sean de enero a diciembre
    mes INT NOT NULL CHECK (mes BETWEEN 1 AND 12),
    anio INT NOT NULL,
    fecha_vencimiento DATE NOT NULL,
    fecha_vencimiento_anual BOOLEAN DEFAULT TRUE,
    ID_obligacion INT
);
```

**Tabla CONTROLDENOTIFICACIONES:**

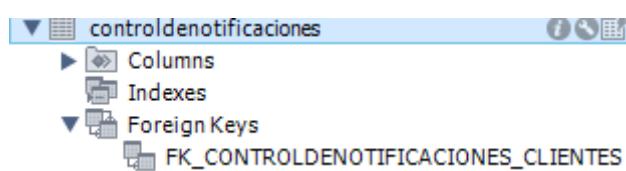
```
CREATE TABLE CONTROLDENOTIFICACIONES (
    id_notificacion INT PRIMARY KEY AUTO_INCREMENT,
    cuit INT NOT NULL,
    obligacion VARCHAR(50),
    fecha_envio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    estado VARCHAR(255),
    mensaje VARCHAR(255)
);
```

**Claves Foráneas:****Tabla OBLIGACIONTRIBUTARIA:**

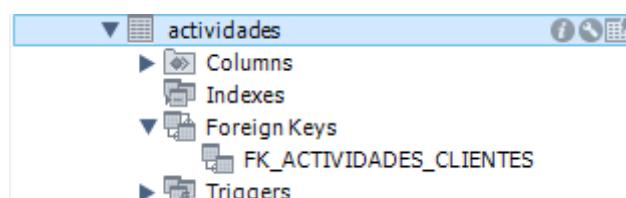
- ALTER TABLE OBLIGACIONTRIBUTARIA ADD CONSTRAINT  
FK\_OBLIGACIONTRIBUTARIA\_CLIENTES FOREIGN KEY (cuit) REFERENCES  
CLIENTES(cuit);

**Tabla CONTROLDENOTIFICACIONES:**

- ALTER TABLE CONTROLDENOTIFICACIONES ADD CONSTRAINT  
FK\_CONTROLDENOTIFICACIONES\_CLIENTES FOREIGN KEY (cuit) REFERENCES  
CLIENTES(cuit);

**Tabla ACTIVIDADES:**

- ALTER TABLE ACTIVIDADES ADD CONSTRAINT FK\_ACTIVIDADES\_CLIENTES  
FOREIGN KEY (cuit) REFERENCES CLIENTES (cuit);



**Tabla PERIODOS:**

- ALTER TABLE PERIODOS ADD CONSTRAINT FK\_PERIODOS\_OBLIGACIONTRIBUTARIA FOREIGN KEY (id\_obligacion) REFERENCES OBLIGACIONTRIBUTARIA(id\_obligacion);

▼ periodos

- Columns
- Indexes
- ▼ Foreign Keys
  - FK\_PERIODOS\_OBLIGACIONTRIBUTARIA
- Triggers

**Inserción, consulta y borrado de registros:**

Con la finalidad de realizar un test respecto a las tablas creadas en el apartado anterior, voy a agregar diferentes registros para las entidades: CLIENTES, ACTIVIDADES, OBLIGACIONTRIBUTARIA, PERIODOS:

- **Inserciones**

**CLIENTES:**

```
INSERT INTO CLIENTES (cuit, razon_social, email, direccion, contribuyente, inicio_actividad) VALUES
```

```
(1,'Juan Carlos SRL', 'JC@gmail.com', 'Las flores', 'Responsable Inscripto','2001-10-05'),
(2,'Ramon Dominguez SA', 'RD@gmail.com', '9 de Julio', 'Responsable Inscripto','1990-01-05'),
(3,'Marcos Sanchez SRL', 'MS@gmail.com', '3 de Febrero', 'Monotributo','1980-08-25');
```

	cuit	razon_social	email	direccion	contribuyente	inicio_actividad
►	1	Juan Carlos SRL	JC@gmail.com	Las flores	Responsable Inscripto	2001-10-05
	2	Ramon Dominguez SA	RD@gmail.com	9 de Julio	Responsable Inscripto	1990-01-05
*	3	Marcos Sanchez SRL	MS@gmail.com	3 de Febrero	Monotributo	1980-08-25
	NULL	NULL	NULL	NULL	NULL	NULL

Pues, si intento ingresar nuevamente un cliente con el mismo cuit, obtengo el siguiente error:

```
✖ 247 15:12:48 INSERT INTO CLIENTES (cuit, razon_social, email, direccion, contribuyente, i... Error Code: 1062. Duplicate entry '1' for key 'PRIMARY'
```

Con estos detalles podemos afirmar que funciona correctamente.

**ACTIVIDADES:**

```
INSERT INTO ACTIVIDADES (codigo, descripcion, cuit) VALUES
(464112, 'Venta al por mayor de artículos de mercería', 1),
(464121, 'Venta al por mayor de prendas de vestir de cuero',2),
(464211,'Venta al por mayor de libros y publicaciones',3);
```

	<b>id_actividad</b>	<b>codigo</b>	<b>descripcion</b>	<b>cuit</b>
►	6	464112	Venta al por mayor de artículos de mercería	1
	7	464121	Venta al por mayor de prendas de vestir de cuero	2
*	8	464211	Venta al por mayor de libros y publicaciones	3
*	NULL	NULL	NULL	NULL

**OBLIGACIONTRIBUTARIA:**

```
INSERT INTO OBLIGACIONTRIBUTARIA (nombre, descripcion, cuit) VALUES
('IIBB', 'Ingresos Brutos', 1),
('IVA', 'Impuesto al Valor Agregado',2),
('SEH', 'Seguridad e Higiene',3);
```

	<b>id_obligacion</b>	<b>nombre</b>	<b>descripcion</b>	<b>cuit</b>
►	1	IIBB	Ingresos Brutos	1
	2	IVA	Impuesto al Valor Agregado	2
*	3	SEH	Seguridad e Higiene	3
*	NULL	NULL	NULL	NULL

**PERIODOS:**

```
INSERT INTO PERIODOS (mes, anio, fecha_vencimiento, ID_obligacion) VALUES
(1,2025, '2025-02-17',1),
(2,2025, '2025-03-17',1),
(1,2025, '2025-04-17',2),
(2,2025, '2025-02-20',2),
(1,2025, '2025-03-20',3),
(2,2025, '2025-04-20',3);
```

	id_periodo	mes	anio	fecha_vencimiento	fecha_vencimiento_anual	ID_obligacion
▶	1	1	2025	2025-02-17	1	1
	2	2	2025	2025-03-17	1	1
	3	1	2025	2025-04-17	1	2
	4	2	2025	2025-02-20	1	2
	5	1	2025	2025-03-20	1	3
*	6	2	2025	2025-04-20	1	3
	NULL	NULL	NULL	NULL	NULL	NULL

- **Consultas:**

Es muy genérico el campo de las consultas, ya que estas vienen acompañadas dependiendo del tipo de necesidad del usuario. Para estos casos de prueba, voy a optar por los siguientes 3 contextos:

1. Se desean visualizar la cuit, razon social y obligaciones de todos los clientes:

```
SELECT
    c.cuit,
    c.razon_social,
    ob.descripcion
FROM CLIENTES AS c INNER JOIN OBLIGACIONTRIBUTARIA AS ob;
```

	cuit	razon_social	descripcion
▶	1	Juan Carlos SRL	Ingresos Brutos
	2	Ramon Dominguez SA	Ingresos Brutos
	3	Marcos Sanchez SRL	Ingresos Brutos
	1	Juan Carlos SRL	Impuesto al Valor Agregado
	2	Ramon Dominguez SA	Impuesto al Valor Agregado
	3	Marcos Sanchez SRL	Impuesto al Valor Agregado
	1	Juan Carlos SRL	Seguridad e Higiene

2. Se desea visualizar los vencimientos de los periodos de la obligación tributaria "IIBB"

```
SELECT
    ob.nombre,
    p.mes,
    p.anio,
    p.fecha_vencimiento
```

FROM OBLIGACIONTRIBUTARIA AS ob INNER JOIN PERIODOS AS p WHERE ob.id\_obligacion=p.id\_obligacion AND ob.nombre='IIBB';

	nombre	mes	anio	fecha_vencimiento
▶	IIBB	1	2025	2025-02-17
	IIBB	2	2025	2025-03-17

3. Se desean visualizar todos los vencimientos futuros del cliente 'Juan Carlos SRL'

SELECT

```
c.razon_social,  
ob.nombre,  
p.mes AS 'Periodo',  
p.fecha_vencimiento
```

FROM CLIENTES AS c, OBLIGACIONTRIBUTARIA AS ob, PERIODOS AS p  
WHERE c.razon\_social='Juan Carlos SRL';

	razon_social	nombre	Perido	fecha_vencimiento
▶	Juan Carlos SRL	IIBB	1	2025-02-17
	Juan Carlos SRL	IVA	1	2025-02-17
	Juan Carlos SRL	SEH	1	2025-02-17
	Juan Carlos SRL	IIBB	2	2025-03-17
	Juan Carlos SRL	IVA	2	2025-03-17
	Juan Carlos SRL	SEH	2	2025-03-17
	Juan Carlos SRL	TTRR	1	2025-04-17

- **Borrado:**

DELETE FROM CONTROLDENOTIFICACIONES;

DELETE FROM OBLIGACIONTRIBUTARIA;

DELETE FROM PERIODOS;

DELETE FROM CLIENTES;

DELETE FROM ACTIVIDADES;

### **Presentación de las consultas SQL:**

El desarrollo de las prácticas aplicadas en las secciones anteriores se encuentran desarrolladas paso a paso en el archivo adjunto “comandosSQL.sql”

---

### **TP N°3:**

#### **Devolución TP N°2:**

Comienzo la resolución del correspondiente TP haciendo énfasis en el feedback mencionado en la fecha **07 de oct en 10:34**.

- REORDENAR EL DOCUMENTO. BIBLIOGRAFÍA VA AL FINAL → las bibliografías fueron centralizadas al final de la entrega de la sección del TP N°2

Comienzo con el desarrollo del TP N°3:

#### **Explicación del desarrollo en Java**

Para el correspondiente desarrollo de la consigna, voy a continuar haciendo referencia a los requerimientos no funcionales detallados en entregas anteriores, por otro lado, también voy a hacer énfasis en las características y propiedades del PUD. Sumado a ello, basare el desarrollo de la codificación en JAVA aplicando tanto la teoría como las buenas prácticas mencionadas en las materias: “Taller de Algoritmos y Estructura de Datos I” & “Programación Orientada a Objetos”. Dentro del correspondiente desarrollo vamos a encontrar diferentes tipos de buenas prácticas relacionadas a la POO como lo pueden ser el encapsulamiento, herencia, polimorfismo, abstracción, entre otras. Cuya finalidad es desarrollar un software modular, flexible y mantenable. Por otro lado, se utilizará el patrón de diseño MVC (Modelo, vista, controlador).

**Aclaración:** Para el desarrollo del software, se utilizará el IDE “Apache NetBeans”, cabe destacar que en esta entrega no se verán reflejada una interfaz gráfica de usuario (IGU), la misma será desarrollada en la entrega final junto con la anidación de la correspondiente base de datos del sistema. Por último, pero no menos importante, los datos no serán consumidos desde la base de datos, los mismos se crearán de forma estática para verificar la funcionalidad del software. A su vez, las funcionalidades del sistema se van a basar en el envío de la notificación, por lo que las validaciones del CRUD no serán foco de la correspondiente presentación (Estas funcionalidades serán entregadas en el último TP)

## Presentación del desarrollo en Java

### MVC:

Comienzo el desarrollo en JAVA creando un nuevo proyecto denominado “sistemaDeGestionYAlertas” y a su vez, creo tres diferentes paquetes, los cuales serán “modelo”, “vista”, “controlador”. Dentro de los correspondientes paquetes, voy a crear cada una de las clases que van a componer dichos paquetes:

### Modelo:

Comienzo creando las clases “Cliente”, “ObligacionTributaria”, “Periodo”, “ControlDeNotificacion” (Cabe destacar que las clases siguen la lógica definida en TPs anteriores). Las cuales me serán de utilidad para poder llegar a la funcionalidad de envío de notificaciones que se estableció para el correspondiente software desde un principio:

- **ObligacionTributaria**
- **Cliente**
- **Periodo**

Los modelos definidos van a almacenar los atributos y métodos correspondientes a sus respectivas clases, por otro lado, se definieron los getters y setters que serán de utilidad para el controlador.

La declaración de las respectivas clases se encontrará en la sección de “modelo” del archivo “sistemaDeGestionYAlertas.java”

### Controlador:

Con la finalidad de centrar el desarrollo en la función principal del correspondiente sistema, voy a definir de forma estática la creación de los Clientes, Periodos, Obligaciones Tributarias. Por lo que sus controladores no solicitarán ningún tipo de ingreso de datos, los mismos deberán ser seleccionados a la hora de interactuar con la Vista. A su vez, dentro de los controladores definidos, podemos encontrar dos métodos principales, por un lado, se añadieron datos de prueba dentro de los constructores y por otro lado se creó una función para listar la información, la cual será interactiva para el usuario en la vista

- **Cientes**
- **Periodo**
- **Obligaciones Tributarias**

Por otro lado, a la hora de desarrollar la solución que controlaba el envío de las notificaciones, me tope con una problemática. **¿Como podía hacer para relacionar las clases Cliente, Obligación Tributaria y Periodo para poder filtrar aquellos clientes cuyos impuestos sean del periodo anterior y tengan una consideración de 4 días de anticipación previo a su vencimiento?**

Pues, como solución a ello, opte por crear una nueva clase controlador llamado “ControlNotificacion”. En este controlador podemos identificar la unión entre las tres clases mencionadas anteriormente, por otro lado, realizar las verificaciones correspondientes a la hora de notificar a los colaboradores. El detalle del funcionamiento de dicho controlador se encuentra descripto en el archivo bajo el nombre “ControlNotificacion”

### **Vista:**

Por último, creo la vista, en donde el usuario tenga la capacidad de interactuar con el sistema y seleccionar la opción respecto a su necesidad. Cabe destacar que la vista fue desarrollada con el fin de capturar cualquier tipo de excepción del usuario. Una de las excepciones mas comunes es el ingreso de un numero por fuera del rango o el ingreso de un String o Char en un campo que solo acepta enteros. El desarrollo de la vista permite capturar estas excepciones y mantener la aplicación viva y sin interrupciones

#### ➤ **VistaPrincipal**

La misma se encuentra dentro del paquete “vista” con su respectiva explicación.

Para finalizar con la ejecución final del sistema, se creó por defecto el paquete “com.mycompany.sistemadegestionyalertas” y dentro del mismo el archivo “SistemaDeGestionYAlertas.java”, en donde se llama a la vista para poder inicializar el sistema dado que este archivo tiene la clase Main. Cabe destacar que esta ejecución puede ser modificada y ajustada a cualquier otra clase.

### **El programa compila y se ejecuta correctamente:**

#### ➤ **Inicio del programa:**

```
*Bienvenido al SISTEMA DE GESTION Y ALERTAS*
=====MENU DE OPCIONES=====
1-Listar a los clientes
2-Listar Obligaciones
3-Listar Periodos
4-Envio notificacion
5-Salir del sistema
Ingrrese una opcion:
```

#### ➤ **Ingreso exitoso del dato dentro del rango del menú de opciones y devolución**

```
Ingrese una opcion:
1
-----LISTA CLIENTES-----
CUIT: 12123456 | Razon Social: Juan Perez | Email: JP@gmail.com | Direccion: Las flores 1778 | Contribuyente: Responsable Inscripto | Inicio Actividad: 2001-12-20 |
CUIT: 21356854 | Razon Social: Carla Dominguez | Email: CD@gmail.com | Direccion: Santa Fe 600 | Contribuyente: Monotributo | Inicio Actividad: 1987-05-01 |
CUIT: 34586987 | Razon Social: Marcos Lopez | Email: ML@gmail.com | Direccion: Belgrano 9875 | Contribuyente: Responsable Inscripto | Inicio Actividad: 2007-11-15 |
CUIT: 23569752 | Razon Social: Martina Azul | Email: MA@gmail.com | Direccion: Calchaqui 100 | Contribuyente: Monotributo | Inicio Actividad: 1995-02-06 |
*Bienvenido al SISTEMA DE GESTION Y ALERTAS*
====MENU DE OPCIONES====
1-Listar a los clientes
2-Listar Obligaciones
3-Listar Periodos
4-Envio notificacion
5-Salir del sistema
Ingrese una opcion:
```

➤ **Ingreso fallido del dato por fuera del rango del menú y devolución**

```
Ingrese una opcion:
500
-----
Ingrese un numero entero del 1 al 5
-----
*Bienvenido al SISTEMA DE GESTION Y ALERTAS*
====MENU DE OPCIONES====
1-Listar a los clientes
2-Listar Obligaciones
3-Listar Periodos
4-Envio notificacion
5-Salir del sistema
Ingrese una opcion:
```

➤ **Test de la funcionalidad principal del programa (Opción 4)**

```
Ingrese una opcion:
4
=====
Asunto: Recordatorio de vencimiento de la obligacion SEH
Cuerpo Notificacion
: Estimado colaborador

Esta notificacion tiene como finalidad reportar que la obligacion SEH vence el dia 2025-10-31
Tenga en consideracion que los clientes que deben presentar la correspondiente declaracion son:
--> Juan Perez
--> Marcos Lopez

Por favor realizar la presentacion lo antes posible

Atte: Sistema de Gestión y Alertas
=====
*Bienvenido al SISTEMA DE GESTION Y ALERTAS*
====MENU DE OPCIONES====
1-Listar a los clientes
2-Listar Obligaciones
3-Listar Periodos
4-Envio notificacion
5-Salir del sistema
Ingrese una opcion:
```

➤ **Fin de la ejecución del programa (Opción 5)**

```
Ingrese una opcion:  
5  
Gracias por utilizar el sistema contable  
-----  
BUILD SUCCESS  
-----  
Total time: 03:57 min  
Finished at: 2025-10-26T22:46:32-03:00  
-----
```

---

## TP N°4:

### **Selección del patrón de diseño y justificación**

El patrón de diseño seleccionado para el desarrollo de la entrega final del proyecto es el patrón de diseño MVC (Modelo, Vista, Controlador), parte de su selección se debe a que promueve la separación de responsabilidades, generando un sistema más modular, mantenible y escalable.

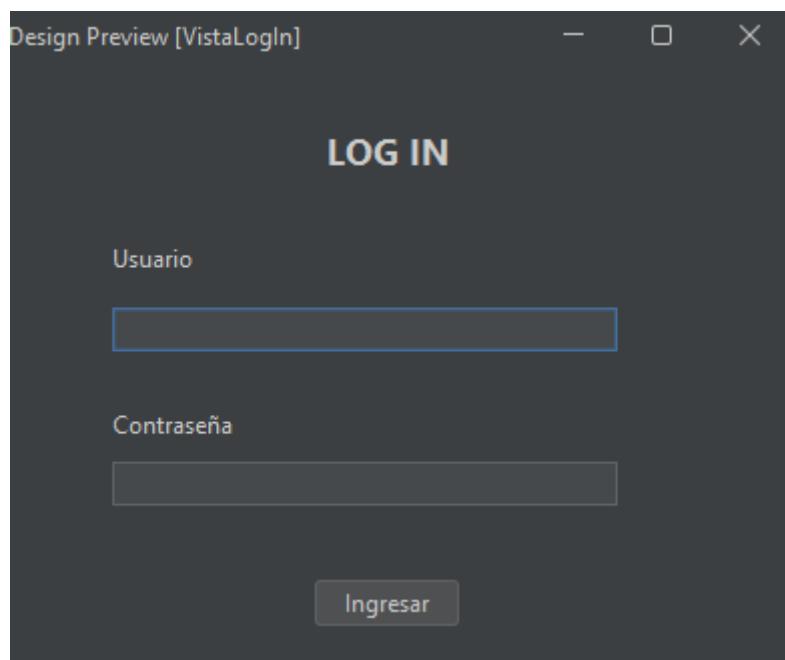
Dentro de este patrón y en desarrollo del correspondiente sistema, podemos encontrar algunos beneficios:

- **Mantenibilidad y organización:** Al aislar la lógica de negocio de la interfaz del usuario, los cambios entre las diferentes partes del sistema tienen menos probabilidades de afectar a las otras, simplificando el mantenimiento del código a largo plazo y abriendo la posibilidad de implementar mejoras a futuro
- **Escalabilidad:** La estructura modular facilita la adición de nuevas funcionalidades sin tener que rediseñar toda la arquitectura que ya existe
- **Facilita el desarrollo para múltiples desarrolladores:** Puede ocurrir que otros desarrolladores quieran sumar nuevas funcionalidades al sistema o deseen trabajar en forma simultánea en el modelo, por medio de este patrón, agilizamos dicho desarrollo
- **Flexibilidad en la interfaz:** La separación de la lógica de presentación permite realizar modificaciones sin tener que reescribir la lógica de negocio

## **Explicación del desarrollo, usando un patrón MVC**

El desarrollo del sistema utilizando el patrón MVC comienza por medio de la creación de las interfaces gráficas en la sección de “**VISTAS**”, dentro de las mismas encontraremos:

- **VistaLogin**



En esta vista encontraremos un formulario de ingreso, el mismo se encargará de solicitar las credenciales para el acceso al sistema, cabe destacar que este es un medio de validación para que solo aquellos colaboradores con privilegios puedan acceder al sistema.

Esta vista esta acompañada de mensajes de verificación en caso de hallar algún error en las credenciales, las mismas serán reflejadas en el video de presentación.

- **VistaPrincipal**



Por otro lado, el sistema cuenta con la vista principal, en la misma se encuentran todas las interfaces del sistema. Cada una de las interfaces es utilizada para reflejar diferentes funcionalidades, las cuales se detallan a continuación:

- **Clientes:** En esta sección se permite realizar un CRUD para los diferentes clientes, en la parte inferior de la sección se encuentran las cuatro funcionalidades que componen al CRUD
- **Obligaciones:** Esta sección se encarga de mostrar las obligaciones que fueron asignadas a cada cliente. Solo se pueden visualizar el listado de las mismas
- **Periodos:** Esta sección se encarga de mostrar los vencimientos, meses y años a los cuales hacen referencia las obligaciones tributarias creadas, las mismas fueron “IIBB”, “SEH” e “IVA”. Solo se pueden visualizar el listado de las mismas
- **Control:** Esta sección se encarga de almacenar el registro del envío de notificaciones, tanto para verificar el estado del envío, como para tener un control de la cantidad de tiempo de anticipación al anticipo del vencimiento de la obligación (Recordar que son 4 días antes del vencimiento). Solo se pueden visualizar el listado de las mismas
- **Notificación:** Esta sección simula el proceso que se ejecutaría de forma automática en el sistema para no depender de una acción manual, a fines prácticos, se creó esta instancia manual para poder realizar la prueba de presentación final del proyecto con el correspondiente video.

### Aclaración:

La finalidad del correspondiente proyecto es reflejar la automatización de las notificaciones para recordar y anticipar a los colaboradores las fechas próximas de los vencimientos de las obligaciones. El foco no estuvo en el desarrollo de los diferentes CRUD, solamente se realizó uno para los clientes, cabe destacar que se aplicarían las mismas prácticas para las otras secciones en caso de que se hubieran desarrollado.

- **Modelos**

Para el desarrollo de los modelos, opte por crear dos clases diferentes, por un lado, el modelo de la clase y por otro el modelo DATA ACCESS OBJECT de la clase. La motivación detrás de estas prácticas se basaron en contribuir con el principio de responsabilidad única y aislar la lógica de negocio de la lógica de la persistencia de los datos.

Continuando con el foco en el sistema de notificaciones, los modelos creados fueron los siguientes:

- **ModeloControl**

La definición de este modelo se centra principalmente en los datos, que representan la estructura para el registro de las notificaciones. También fueron definidos sus getters y setters.

- **ModeloControlDAO**

Por otro lado, esta clase se encarga de encapsular la lógica para interactuar con la base de datos respecto de la entidad mencionada anteriormente. Su función principal es abstraer y gestionar las operaciones de persistencia, entre ellos podemos encontrar:

- Gestionar la conexión con la base de datos
- Registrar una nueva notificación
- Listar las notificaciones

El comando utilizado para realizar la inserción en la base de datos es:

**INSERT INTO controldenotificaciones (cuit, obligacion, estado) VALUES (?,?,?)**

- **ModeloNotificacionDAO**

Por último, esta clase se encarga de acceder a los datos, su función es orquestar un proceso de notificación completo. Entre sus funciones principales podemos encontrar:

- Consultar vencimientos
- Generar contenido para el email
- Enviar la notificación
- Registrar el control

El comando utilizado para realizar la inserción en la base de datos es:

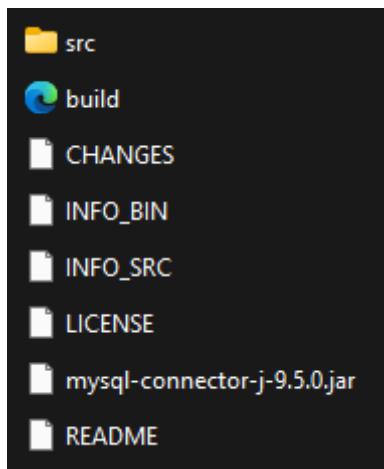
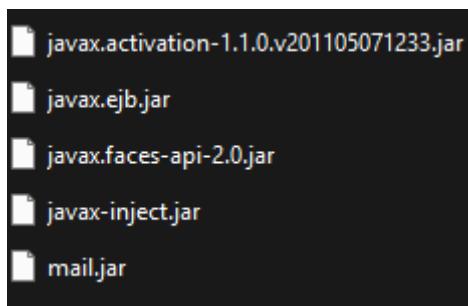
**SELECT c.razon\_social, o.nombre, p.fecha\_vencimiento FROM clientes c INNER JOIN obligaciontributaria o ON c.cuit = o.cuit INNER JOIN periodos p ON o.nombre =**

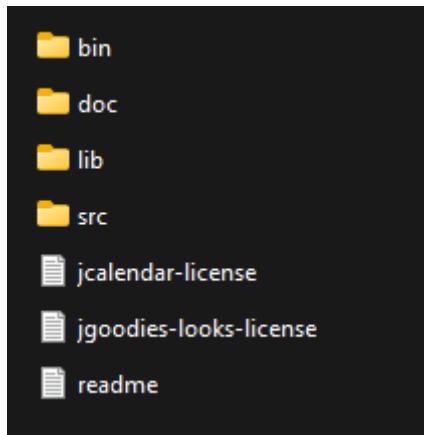
```
p.nombre_obligacion WHERE o.nombre = ( SELECT nombre_obligacion FROM periodos  
WHERE fecha_vencimiento >= CURDATE() AND fecha_vencimiento = DATE_ADD(?, INTERVAL  
4 DAY) ORDER BY fecha_vencimiento ASC LIMIT 1) GROUP BY c.cuit, c.razon_social,  
o.nombre, p.fecha_vencimiento
```

*Cabe destacar que fue de mucha utilidad consultar los materiales prácticos de la materia “BASE DE DATOS I” ya que los mismos fueron de vital utilidad para realizar todas las funcionalidades en un solo request*

### Presentación del desarrollo en Java

Algunas librerías que fueron necesarias para la conexión con la base de datos y la conexión con la casilla de email para el envío de las notificaciones. Tambien para poder tener la sección de calendarios en la paleta de diseño de NetBeans





Por otro lado, se utilizo XAMPP para crear un servidor en web local en mi PC y poder utilizar las funcionalidades de MySQL



Datos útiles para utilizar el sistema:

**Usuario:** admin

**Contraseña:** 123

### **Presentación del proyecto en un video de 3 minutos**

El mismo se encuentra adjunto en los archivos de entrega del correspondiente TP

## Bibliografía:

- **EcuRed. (s.f.).** Flujo de Trabajo Análisis y Diseño
- **Sección N°4:** Diagramas del modelo de diseño. Análisis y Diseño de Software
- **Guevara, J. (s.f.).** Análisis y diseño detallado de aplicaciones informáticas de gestión
- **Piattini M. G., Calvo-Manzano J. A., Cervera, J. Fernández, L. (2004).** Análisis y diseño detallado de aplicaciones informáticas de gestión
- **Booch, G., Jacobson, I. y Rumbaugh, J. (1999).** El lenguaje unificado de modelado
- **Arias Chaves, M. (2005).** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software
- **Braude, E. (2003).** Ingeniería de software. Una perspectiva orientada a objetos
- **Contardo, M. V. y Dejón, G. V. (s.f.).** Introducción al Diseño y Desarrollo de Software con UML y Java
- **Torossi, G. (s.f.).** El Proceso Unificado de Desarrollo de Software.
- **Booch, G., Rumbaugh, J. y Jacobson, I. (1999).** El lenguaje unificado de Modelado Madrid
- **CASANDRASOFT. (s.f.).** Análisis y diseño de productos de software.
- **Sección N°3-M3:** Procesos y objetos del modelo de diseño. *Análisis y Diseño de Software*
- **Sección N°1-M4:** Requerimientos de la implementación. *Análisis y Diseño de Software*
- **Sección N°2-M4:** Diagrama de despliegue y diagrama de componentes. *Análisis y Diseño de Software*
- **Booch, G., Rumbaugh, J. y Jacobson, I. (1999).** El lenguaje unificado de Modelado Madrid
- Link de investigación: <https://diagramasuml.com/despliegue/>
- Link de investigación: <https://www.uml-diagrams.org/deployment-diagrams.html>
- **Jacobson, I.; Booch, G. y Rumbaugh, J. (2000).** El Proceso Unificado de Desarrollo de Software
- **Sección N°3-M4:** Modelo de prueba. *Análisis y Diseño de Software*
- M4 L3 MODELOS DE PRUEBAS PARA PRUEBAS DEL SISTEMA, PDF con ejercicio práctico
- **Date, C. (1998).** Introducción a los sistemas de base de datos
- **Sección N°2 M2:** Normalización de tablas de un modelo relacional. *Base de datos I*
- **Elmasri, R., & Navathe, S. (2002).** Fundamentos de Sistemas de Bases de datos. Madrid: Pearson Educación.
- **Sección N°2 M1:** Componentes del modelo relacional en MySQL. *Base de datos I*
- **MySQL. (2021).** MySQL 8.0 Reference manual
- **Sección N°2 M3:** Sentencias DML Insert, Update y Delete. *Base de datos I*

- **Kendall, K., & Kendall, J. (2011).** Análisis y diseño de sistemas. Pearson Education