

Trabajo Práctico Integrador

Sistema de Reporte de Incidentes

Contexto general

Una importante empresa de soporte operativo solicita el diseño y desarrollo de un sistema que le permita la generación y seguimiento de los incidentes que se presentan.

La empresa en cuestión se dedica a brindar soporte operativo sobre distintas aplicaciones (SAP, Tango, etc.) y sistemas operativos (Windows, MacOS, Linux Ubuntu).

El **área de RRHH** se encarga de realizar las altas, bajas y modificaciones de los **técnicos** que se encargan de resolver los **incidentes reportados**.

Cada técnico tiene una o varias **especialidades** y solo se le pueden asignar incidentes que coincidan con las mismas.

El **área comercial** es responsable de incorporar nuevos **clientes** a la empresa. Administra las altas, bajas y modificaciones de los datos de cada uno de ellos.

Finalmente, **la mesa de ayuda** es responsable de atender las llamadas e ingresar al sistema los **incidentes** reportados.

Ciclo de vida de un incidente

Cuando un cliente llama, **la mesa de ayuda** le solicita los datos para identificarlo (**razón social**, **CUIT**) y los ingresa en el sistema para que el mismo le muestre los servicios que el cliente tiene contratados.

El operador (de la mesa de ayuda) solicita que le informen por cuál de esos servicios desea **reportar un incidente**, junto con una **descripción del problema** y el tipo del problema.

Al ingresar el incidente, el sistema devuelve un listado de **técnicos disponibles** para resolver el problema. El operador selecciona uno de los técnicos disponibles y el sistema le informa el **tiempo estimado de resolución**. Luego, informa al cliente que el incidente ha sido ingresado y **la fecha posible de resolución**.

Al confirmarse el incidente, el sistema debe **enviar una notificación al técnico informándole** que tiene un nuevo incidente para resolver.

Cuando el técnico atiende y resuelve el incidente, lo debe marcar como “resuelto”, indicando las consideraciones que crea necesarias. Cuando esto ocurra, el sistema debe enviar un email al cliente informándole que su incidente ya está solucionado.

Otros requerimientos

- El sistema debe permitir al área de RRHH emitir diariamente reportes con los incidentes asignados a cada técnico y el estado de los mismos.
- El sistema debe permitir que el operador agregue “un colchón” de horas estimadas para la resolución del problema, si el mismo es considerado “complejo”.
- El sistema debe permitir el alta de incidentes que contengan un conjunto de problemas de un mismo servicio. Dichos problemas deben estar relacionados.
- El sistema debe dar la posibilidad de informar:
 - Quién fue el técnico con más incidentes resueltos en los últimos N días
 - Quién fue el técnico con más incidentes resueltos de una determinada especialidad en los últimos N días
 - Quién fue el técnico que más rápido resolvió los incidentes

Consideraciones

- Cada tipo de problema particular puede ser solucionado por una o varias especialidades.
- Cada operador puede definir, optativamente, su tiempo estimado de resolución por defecto por tipo de problema; el cual tendrá que ser menor al tiempo máximo de resolución definido para el tipo de problema.
- Cada técnico puede definir su medio preferido de notificación, los cuales pueden ser: Email o WhatsApp. No están definidas las bibliotecas que se utilizarán para realizar estas notificaciones.

Metodología

Se propone una metodología de trabajo **iterativa** e **incremental**. Para esto, el TP se divide en tres entregas, las cuales se realizarán a través del envío del link al repositorio de trabajo por medio del Campus Virtual en la fecha estipulada por el Docente del curso.

En la última entrega (coincidente con la tercera) se deberá exponer el TP frente al curso, mostrando la solución generada y justificando las decisiones tomadas.

Entregas

Entrega 1

En esta primera iteración nos encargaremos de modelar, a nivel datos y objetos, una solución al dominio presentado. Además, comenzaremos con el proceso de codificación de la solución.

En particular, en esta entrega se solicita:

1. Modelo de datos (DER físico) que brinde solución al dominio.
2. Código con modelado de clases. El código debe estar subido a un repositorio de GitHub.

Es necesario que el proyecto Java sea creado como un proyecto “Maven” para poder añadir algunas dependencias. Se recomienda la utilización de la dependencia de [Lombok](#) para facilitar la tarea repetitiva de generación de Setters y Getters de las clases.

Entrega 2

En esta segunda iteración nos encargaremos de mapear, mediante anotaciones JPA, nuestras clases implementadas para poder persistir el modelo en una Base de Datos Relacional. Además, comenzaremos con la generación de los repositorios/servicios para que brinden solución a algunos requerimientos planteados.

En particular, en esta entrega se solicita:

1. Modelado de clases con Mapeo (anotaciones JPA) de entidades para que las mismas sean persistidas mediante el ORM Hibernate.
2. Repositorios/Servicios que den solución a los requerimientos planteados:
 - a. Quién fue el técnico con más incidentes resueltos en los últimos N días
 - b. Quién fue el técnico con más incidentes resueltos de una determinada especialidad en los últimos N días
 - c. Quién fue el técnico que más rápido resolvió los incidentes