

Algoritmos 1

Licenciatura en Ciencia de Datos -

UNSAM Examen parcial

Forma de Entrega: Archivos de código fuente (*.java) y entrega del proyecto java a través del assignment de GitHub.

Criterios mínimos de aprobación:

- No deben existir errores de compilación.
- Implementar la estructura básica de la solución contemplando el POO.
- Completar correctamente la funcionalidad requerida según el escenario propuesto.

Criterios de evaluación:

- Dominio en el diseño de soluciones utilizando conceptos del paradigma orientado a objetos.
- Análisis y comprensión de problemas y su resolución mediante el uso de estructuras de datos adecuadas.
- Conocimiento y aplicación de buenas prácticas de desarrollo vistas en la materia.
- Capacidad para construir diseños de software extensible, mantenible y reutilizable.

Una empresa de seguros presta un servicio de emergencias mecánicas. Los clientes están conformes con el servicio pero hay algunos reclamos por la tardanza. La Gerencia de Calidad en el asunto y nos pidió que construyamos un sistema para registrar prestaciones, el tiempo entre que el cliente solicitó y que fue atendido y su nivel de satisfacción. El sistema debe permitir registrar asegurados, prestador, tipos de servicio que brinda el prestador, solicitudes de servicio y devoluciones de los asociados.

El asegurado se identifica con su DNI, su nombre y apellido, un email y un teléfono celular para contacto. Además se registra la fecha del último pago. Si la compañía no registra pagos en los últimos tres meses el asegurado es considerado moroso y al pedir un servicio se le informa que debe regularizar su situación. A su vez el asegurado tiene distintos planes: Tranquilo, Comodo y Lujo.

El prestador se identifica por la CUIT, el nombre y apellido, un email y un teléfono de contacto. Un prestador puede ofrecer distintos tipos de servicios: acarreo, recarga de batería, cambiar neumático y mecánica ligera. Se estima que cada servicio tiene distintas duraciones: acarreo 40 min, recarga de batería 15 min, cambiar neumático 30 min y mecánica ligera 1 hora.

Una solicitud de servicio tiene asociado un cliente, un prestador asignado, un tipo de tarea y una hora estimada de atención. La hora estimada se calcula con la estimación por los trabajos que hayan en cola. Al registrarse un pedido se busca el prestador que menos pedidos tenga por delante.

Al crearse se le asigna un número de identificación y la hora. La solicitud pasa por distintos estados: asignada, en proceso, realizada, rechazada. Al asignarse en la cola de un prestador se la marca como asignada.

Al llegar al sitio de atención el prestador indica que el nro de prestación está en proceso y se registra la hora. Al completarse el cliente informa que está realizada y se registra la hora en que se completó. En cualquier momento el usuario puede considerarla rechazada. Registrando la hora.

Antes de aceptar una solicitud hay ciertas reglas para los asociados. Los del plan Tranquilo pueden hacer hasta 2 solicitudes por mes. Los usuarios del plan Cómodo pueden pedir todos los que quieran. Los socios Lujo, además de pedir todos los servicios que quieran son colocados como primeros en la cola de atención del proveedor con menos pedidos.

Cuando un asociado marca como realizada una atención se registra la hora en que fue completo y se le pide un número del 1 al 5 para indicar su satisfacción. (1 peor, 5 mejor). Cuando se rechaza no se pide solicitud y se pide que elija la causa entre: demora, no se resolvió el problema técnico, otras causas.

La Gerencia de Calidad nos pide un reporte de cada prestador con su valoración. Por otra parte nos pide un informe de asociados con la cantidad de atenciones por mes y el promedio de valoraciones y la cantidad de rechazos.

Se solicita Implementar las siguientes funcionalidades:

- a) Implementar todas las clases necesarias que se describen en el enunciado, incluyendo los métodos necesarios para resolver las funcionalidades solicitadas.
- b) Implementar las siguientes operaciones (ver anexo 1):
 - 1. Crear una empresa de seguros que se llame “La Pancha”.
 - 2. Registrar 2 prestadores.
 - 3. Registrar 4 asegurados, al menos uno de cada tipo, uno moroso.
 - 4. Solicitar 8 prestaciones de distintos servicios.
 - 5. Solicitar 1 prestación y rechazarla por morosidad.
 - 6. Solicitar 1 prestación y rechazarla por plan.
 - 7. Llevar 4 solicitudes hasta estado aceptado. Rechazar 1.
 - 8. Mostrar todos las solicitudes del sistema.
 - 9. Mostrar todas las solicitudes en estado asignada y el tiempo que lleva de espera.
 - 10. Mostrar todos los asegurados con al menos 3 solicitudes.

Anexo 1: Ejemplo de Main

Se puede utilizar el siguiente método main para probar el ejercicio, donde ya hay un escenario armado para verificar la funcionalidad.

```
Aseguradora aseguradora = new Aseguradora("La Pancha");
// Prestadores
Prestador rene = aseguradora.registrarPrestador("René", 30376571064, "rene@mail.com", "1123147898",
List.of("acarreo", "batería"));
Prestador coco = aseguradora.registrarPrestador("Coco", 30415336566, "coco@mail.com", "1132143123",
List.of("acarreo", "batería", "neumatico", "mecanica_ligera"));
// Asegurados
Asegurado laura = aseguradora.registrarAsegurado("Laura Gómez", 45234111, "laura@mail.com", "1132939391",
"tranquilo", LocalDateTime.now());
Asegurado juan = aseguradora.registrarAsegurado ("Juan Rodríguez", 50234111, "juan@mail.com", "1132939391",
"comodo", LocalDateTime.now());
Asegurado pepe = aseguradora.registrarAsegurado ("José Pérez", 50432341, "jose@mail.com", "1145879391", "lujo",
LocalDateTime.now());
Asegurado toto = aseguradora.registrarAsegurado ("Toto Sánchez", 47894657, "laura@mail.com", "1156119874",
"lujo", LocalDateTime.now().minusDays(100)); // moroso
// Solicitudes
Solicitud solicitud1 = aseguradora.solicitar(laura, "bateria");
Solicitud solicitud2 = aseguradora.solicitar(pepe, "mecanica_ligera");
Solicitud solicitud3 = aseguradora.solicitar(juan, "acarreo");
Solicitud solicitud4 = aseguradora.solicitar(pepe, "acarreo");
Solicitud solicitud5 = aseguradora.solicitar(laura, "acarreo");
Solicitud solicitud6 = aseguradora.solicitar(juan, "bateria");
Solicitud solicitud7 = aseguradora.solicitar(pepe, "neumatico");
Solicitud solicitud8 = aseguradora.solicitar(juan, "acarreo");
try {
    asegurador.solicitar(laura, "neumatico")
} catch (RuntimeException rex) {
    System.out.println(rex.getMessage()); // se excedió el límite de solicitudes de este mes.
}
try {
    asegurador.solicitar(toto, "acarreo");
} catch (RuntimeException rex) {
    System.out.println(rex.getMessage()); // Debe regularizar su cuenta con vencimiento:<fecha>
}
// Llevar solicitudes a realizado
aseguradora.llegoAlSitioDeAtencion(solicitud1); aseguradora.atencionRealizada(solicitud1, 5);
aseguradora.llegoAlSitioDeAtencion(solicitud2); aseguradora.atencionRealizada(solicitud2, 3);
aseguradora.llegoAlSitioDeAtencion(solicitud3); aseguradora.atencionRealizada(solicitud3, 4);
aseguradora.llegoAlSitioDeAtencion(solicitud4); aseguradora.atencionRealizada(solicitud4, 1);
// Rechazar
aseguradora.solicitudRechazada(solicitud8, "demora");
// Mostrar todas las solicitudes
System.out.println("Solicitudes registradas:");
aseguradora.listarSolicitudes();
// Mostrar todas las solicitudes en estado asignada y el tiempo que lleva de espera.
System.out.println("Solicitudes en estado asignada y tiempo que lleva de espera:");
aseguradora.listarSolicitudesEnEspera();
// Mostrar los asegurados con al menos 3 solicitudes realizadas.
System.out.println("\nAseguradores con al menos 3 solicitudes realizadas:");
aseguradora.mostrarAseguradosConAlMenosSolicitudes(3);
```