Este Desafío lo trabajé en forma diferente a como se solicitaba, pero el programa hace más de lo requerido, que era agregar y borrar contactos desde la capa de servicios. Agrega, lista y borra contactos, incluye mecanismos de control de ingreso de información y los contactos persisten en una base de datos en MySQL.

N°	Requerimiento	Lo que hace el programa
1	En el proyecto Contact Manager, crear la capa de servicio:	Creé el package: com.desafiolatam.services Utilicé anotaciones como Service y Autowired, para conectar con el Repository y agrupar sus funcionalidades o inyectar dependencias. La finalidad es hacer uso de las facilidades que trae el Framework Spring, evitarme todas estas clases e interfaces. Implementé los métodos Save (guardar), DeleteById (borrar por id) y List<Contact>findAll()
1.1.	Crear packages cl.desafiolatam.service	
1.2.	Crear packages cl.desafiolatam.service.impl	
1.3.	Crear la Interfaz ContactService.java	
1.4.	Crear la clase implementación ContactServiceImpl.java	
1.5.	Crear los siguientes métodos getContactList y addContact	
2	Crear la lógica para cada uno de los métodos de la clase service:	ContactRepository se inyecta en ContactService a través de la anotación @Autowired. ContactRepository es una interface que extiende de JpaRepository y que recibe Contact como parámetro. Este es el encargado de encapsular, almacenar, buscar y recuperar los Contact desde la base de datos creada. En el modelo Contact (creado como @Entity) el id, tipo Long, se crea automáticamente, gracias a la anotación @GeneratedValue, evitando así que se pudiese generar un objeto con id cero o que estos se repitan. El método deleteById, es capaz de eliminar un contacto de la lista de contactos.
2.1.	La clase Contact, se debe inyectar a la clase Service	
2.2.	Declarar de forma privada, la lista de contactos	
2.3.	Crear el constructor de la clase Service, en donde se inicialice o se cree listContact	
2.4.	Método getContactList: Debe agregar a listContact el objeto contact, si y sólo si este último tiene un id mayor a cero.	
2.5.	Método addContact: Debe crear un id al contacto con listContact.size() + 1.	
2.6.	Luego, debe agregar a la lista de contacto, el objeto contact.	
2.7.	Método deleteContact: Debe ser capaz de eliminar un contacto o item de la lista de contacto. private List <Contact> listContact;	
3.	Crear el método deleteContact(), en la capa de Servicio para borrar un registro de la lista. Luego Implementar el resto de la lógica, según corresponda, en el controlador y en la vista:	El método deleteContact() es capaz de borrar un registro seleccionado desde la tabla de detalle de contactos en la base de datos creada por el programa. La url es http://localhost:9080/contactManager/deleteContact/?id=2, siendo id el parámetro que recibirá el método del controlador. Utilizo @PathVariable para capturar el “id”
3.1.	deleteContact: Método encargado de eliminar un contacto. Este método debe ser de tipo GET y debe recibir por parámetro el id correspondiente al item a eliminar de la lista.	
3.2.	Para acceder a este método, la url debe ser como sigue: http://localhost:8082/contactManager/deleteContact/?id=2, siendo id el parámetro que recibirá el método del controlador.	
3.3.	Utilizar @RequestParam	
4.	Empaquetar la aplicación	Realizado