

Tabla resumen de cumplimiento de solicitudes del Desafío Crear Proyecto Contact Manager:

Detalle de lo Solicitado	Cumplimiento
1. Crear proyecto con https://start.spring.io/ maven y Spring Boot 2.1.6, considerando los siguientes datos del proyecto:	
1.1. groupId: cl.desafiolatam.	Sí
1.2. artifactId: contactmanager.	Sí
1.3. packaging: war.	Sí
1.4. Utilizar Java 8.	Sí
2. Crear la estructura del proyecto, clase controller, vista y archivo web.xml de la siguiente manera:	
2.1. Crear Packages Controller - cl.desafiolatam.controller.	Sí
2.2. Crear clase controller - ContactController.java.	Sí
2.3. Crear archivo web.xml, configurando el dispatcher de Spring.	No
2.4. Crear método getContactList y devolver la palabra Hola Mundo!! (Esto es, solo para comprobar que el proyecto fue creado correctamente).	No
2.5. Crear la Vista contactManager.jsp bajo la carpeta WEB-INF/views del proyecto.	Sí
2.6. Configurar la vista en el archivo application.properties.	Sí
2.7. Probar en la url http://localhost:8080/contactManager/ .	En la url http://localhost:9080/contactManager/
3. Crear la interfaz como se muestra en Requerimientos funcionales. Una vez creada la interfaz, programar el método getContactList para que muestre una lista de datos tipo Contacto.	
3.1. El método debe ser de tipo GET.	Sí
3.2. Este método actuará como inicio de la aplicación, por lo que para acceder a él se debe hacer mediante la siguiente URL: http://localhost:8080/contactManager/	Sí, pero en la url http://localhost:9080/contactManager/
3.2. La lista, debe ser una Lista de Contactos, por lo que se debe desarrollar la clase Contacto.java en base a los atributos del contacto. List <Contacto> ` listaContacto. Podemos tener un solo contacto el cual puede ser reemplazado y eliminado.	La clase se llama Contact.java, tiene los atributos del contacto y la List<Contact> es listContact. El contacto puede ser agregado y eliminado.
4. Programar los siguientes métodos restantes en el controlador:	

Detalle de lo Solicitado	Cumplimiento
4.1. addContact: Método encargado de agregar un contacto. Este método debe ser de tipo POST y debe ser accedido mediante la siguiente URL: http://localhost:8082/contactManager/addContact .	Se accede en la url http://localhost:9080/contactManager/addContact
4.2. Utilizar @ModelAttribute.	Sí
4.3. Recibir desde la Vista, el objeto contact. Utilizar < form:form /> de Spring MVC.	Sí
4.4. El método, debe crear un id para cada Ítem de la lista.	Sí
4.5. deleteContact: Método encargado de eliminar un contacto. Este método debe ser de tipo POST y debe ser accedido mediante la siguiente URL: http://localhost:8082/contactManager/deleteContact/?id=	Se accede en la url http://localhost:9080/contactManager/deleteContact/?id= + idSeleccionado