

Captura de pantalla de proyecto generado por Spring Initializr	5	1.Spring_Initializr.PNG
Dockerfile desplegando en el contenedor	10	Se agregó archivo DockerFile al proyecto
Archivo application.properties con el formato correcto	5	application.properties
Archivo desplegable usando Gradle o Maven	5	Se crea proyecto desplegable usando Maven
Estructura del proyecto correcta	5	La estructura de las carpetas es la misma a las clases de Spring
Consumo de API pública usando RestTemplate	5	El consumo del API publica se puede ver en la clase ApiPublicaDAOImpl
Conversión del contenido consumido usando JAXB o Jackson	5	El uso de Jackson se puede ver en la clase ApiPublicaDAOImpl
API generada por Spring MVC con un endpoint de cada tipo	10	MVC paquetes -> controller, dao, dto, entity service, utils
Colección de Postman para consumir la API generada	5	Carpeta -> collection_postman
Creación de enlace con MongoDB usando Spring Data JPA	10	application.properties
Uso de MongoTemplate	5	Se implementó en UsuarioDAOImpl
Lectura de MongoDB usando MongoTemplate	10	Se implementó en UsuarioDAOImpl READ
Escritura en MongoDB usando MongoTemplate	10	Se implementó en UsuarioDAOImpl CREATE
Actualización en MongoDB usando MongoTemplate	10	Se implementó en UsuarioDAOImpl UPDATE
Borrado en MongoDB usando MongoTemplate	10	Se implementó en UsuarioDAOImpl DELETE
Prueba unitaria de cada endpoint de la API	20	Test carpeta de controlador
Prueba unitaria de cada operación CRUD	20	Test carpeta de servicio
Uso de Mockito en cada prueba	10	cada Test
Generación de logs por prueba	10	cada Test
Pruebas para Happy Path	10	cada Test
Pruebas para cada Edge Case	10	cada Test

Implementación de JaCoCo o SonarCloud (mínimo 70% de cobertura)	10	
Total	200	
Mínimo aprobatorio	160	