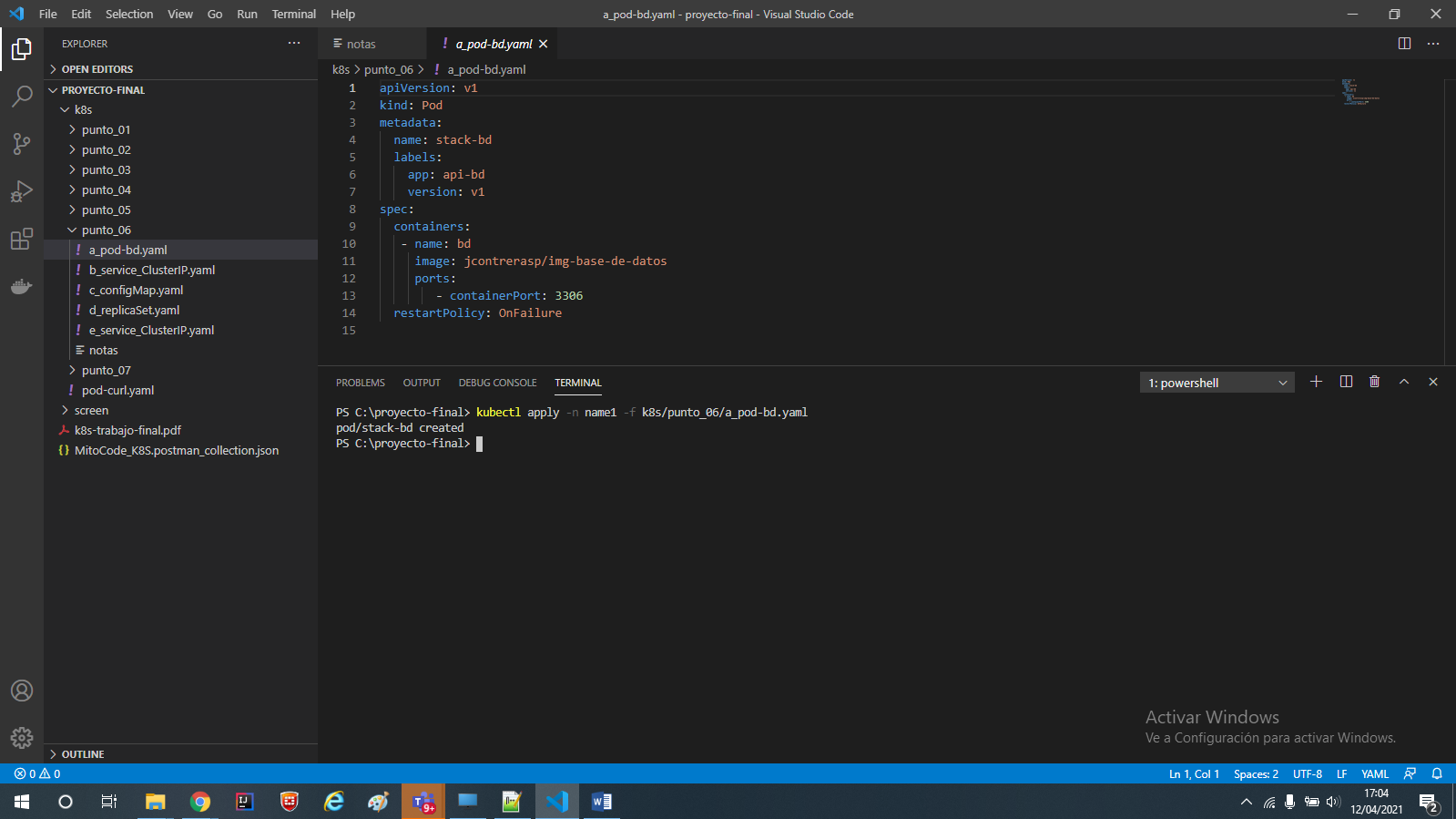
**Punto\_06: En el Namespaces name1 realizar la siguiente implementación:**

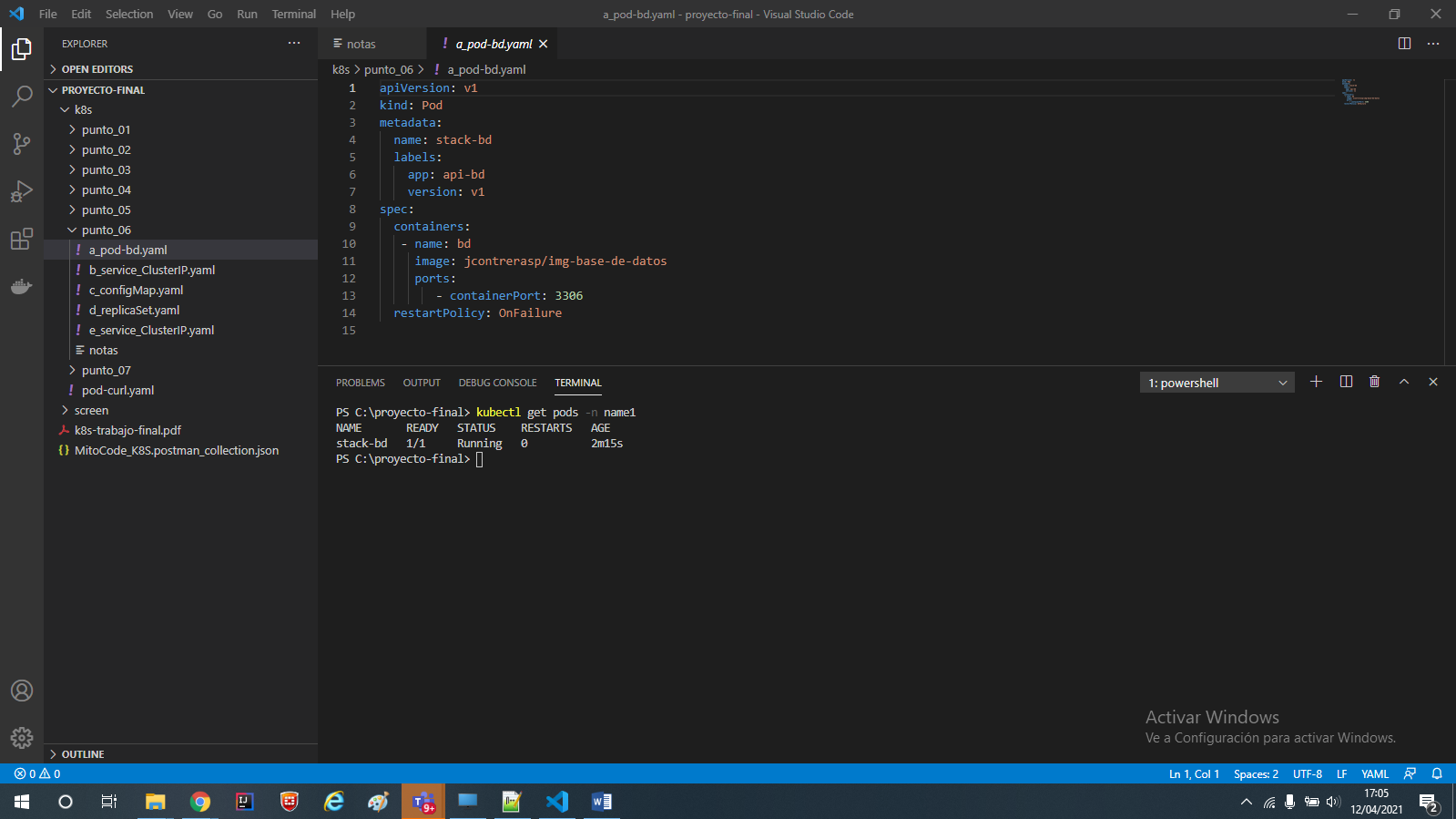
1. **Crear la base de datos en un Pod**
2. **Ejecutamos el manifiesto “a\_pod-bd.yaml” de la carpeta k8s/punto\_06**

**kubectl apply -n name1 -f k8s/punto\_06/a\_pod-bd.yaml**



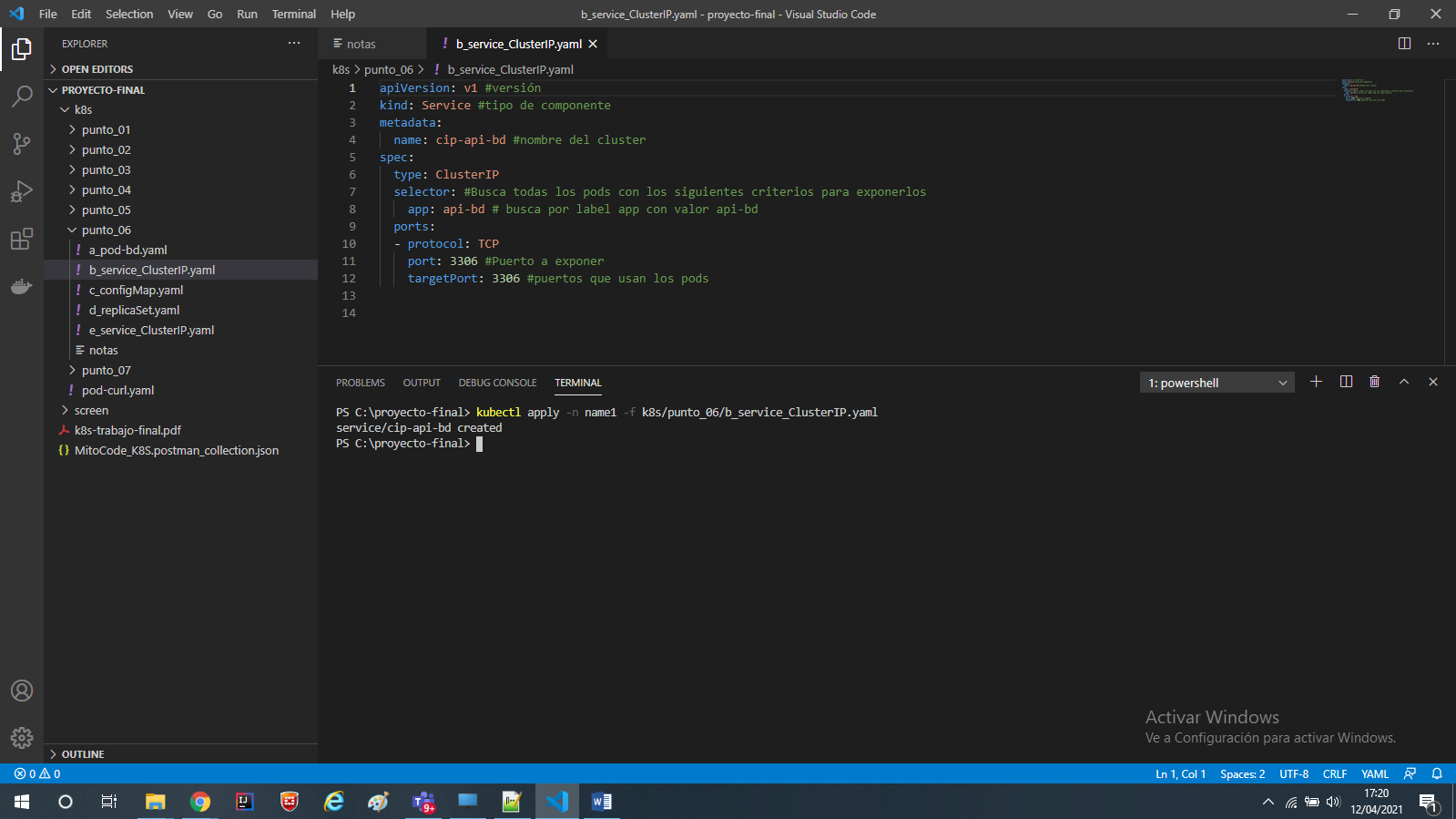
1. **Validamos que el pod se creó en el namespace name1 correctamente**

**kubectl get pods -n name1**



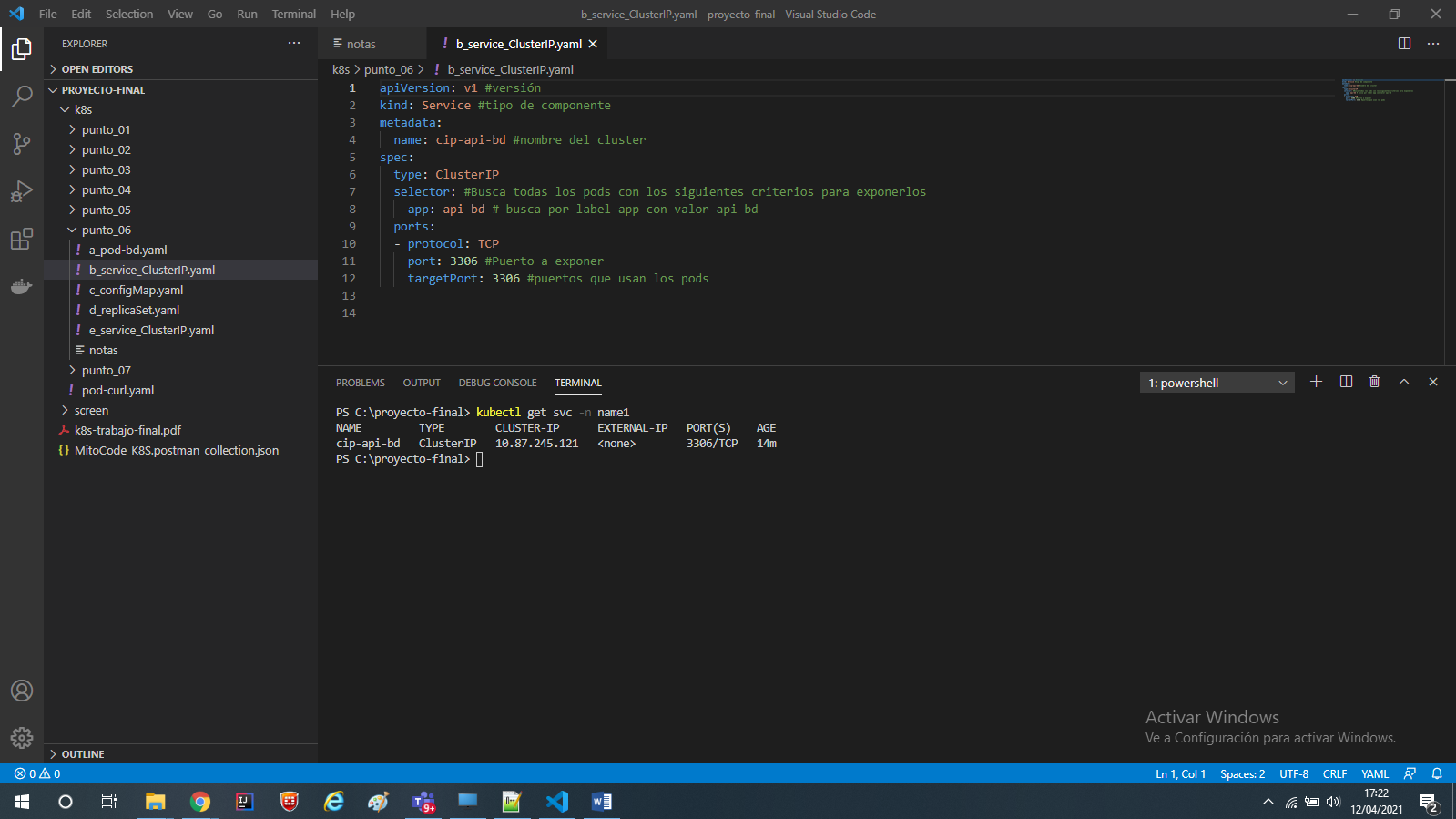
1. **Crear un Service de tipo ClusterIP que permita exponer la base de datos**
2. **Ejecutamos el manifiesto “b\_service\_ClusterIP.yaml” de la carpeta k8s/punto\_06**

**kubectl apply -n name1 -f k8s/punto\_06/b\_service\_ClusterIP.yaml**



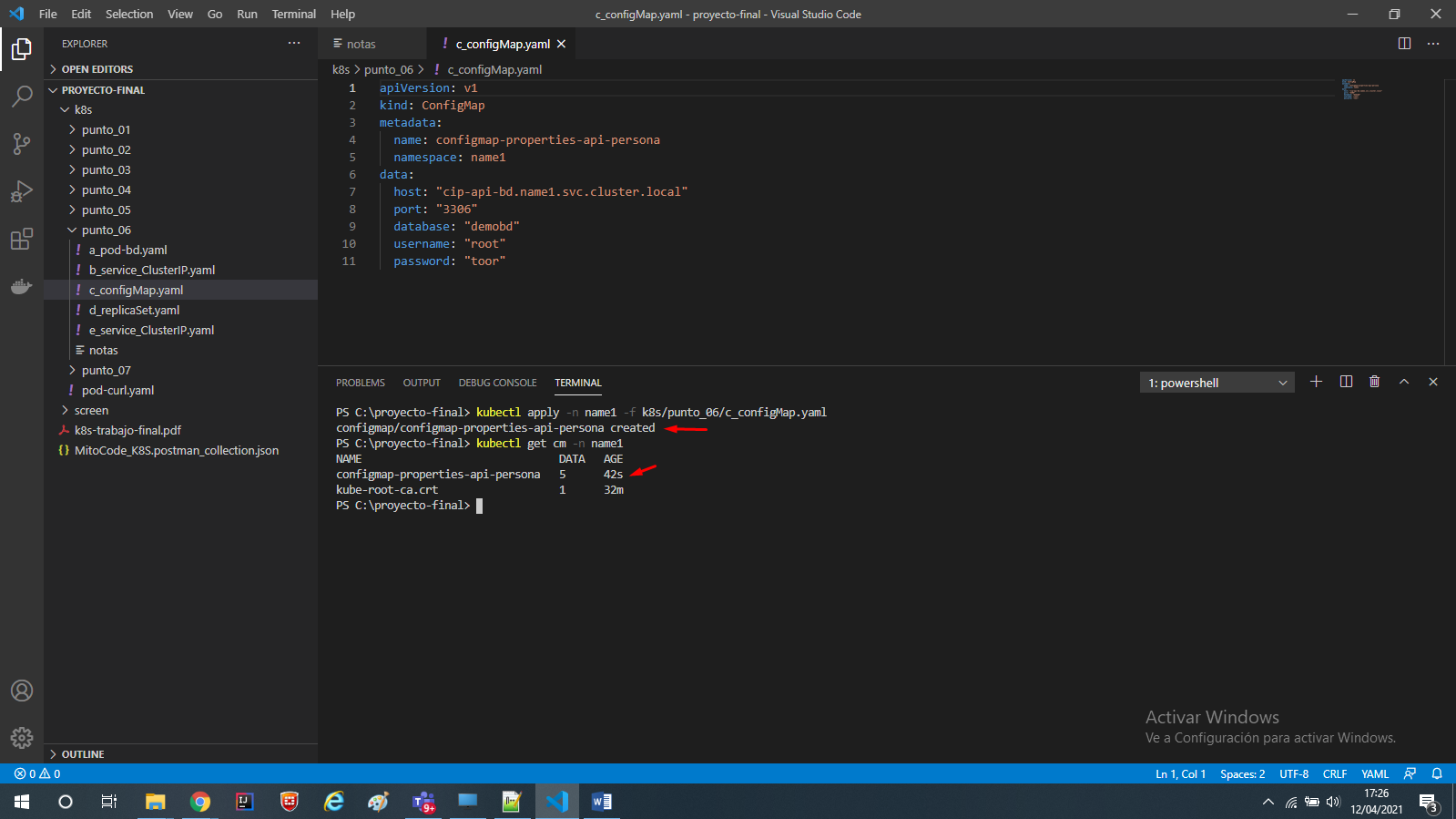
1. **Validamos que se haya creado el clusterIP**

**kubectl get svc -n name1**



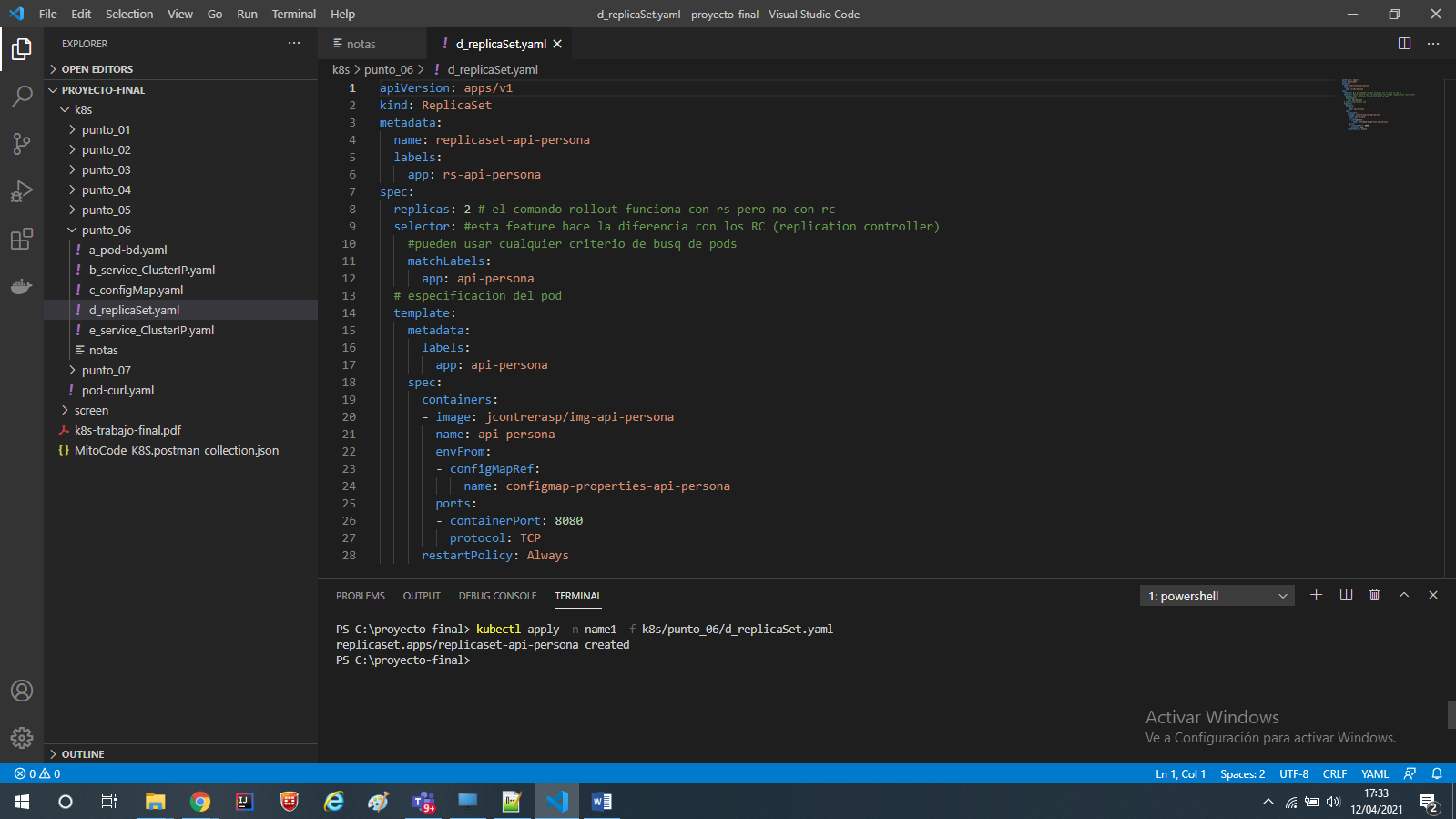
1. **Crear un ConfigMap que permita desacoplar la configuración del aplicativo**
2. **Ejecutamos el manifiesto “c\_configMap.yaml” de la carpeta k8s/punto\_06 y validamos que se haya creado correctamente**

**kubectl apply -n name1 -f k8s/punto\_06/c\_configMap.yaml  
kubectl get cm -n name1**



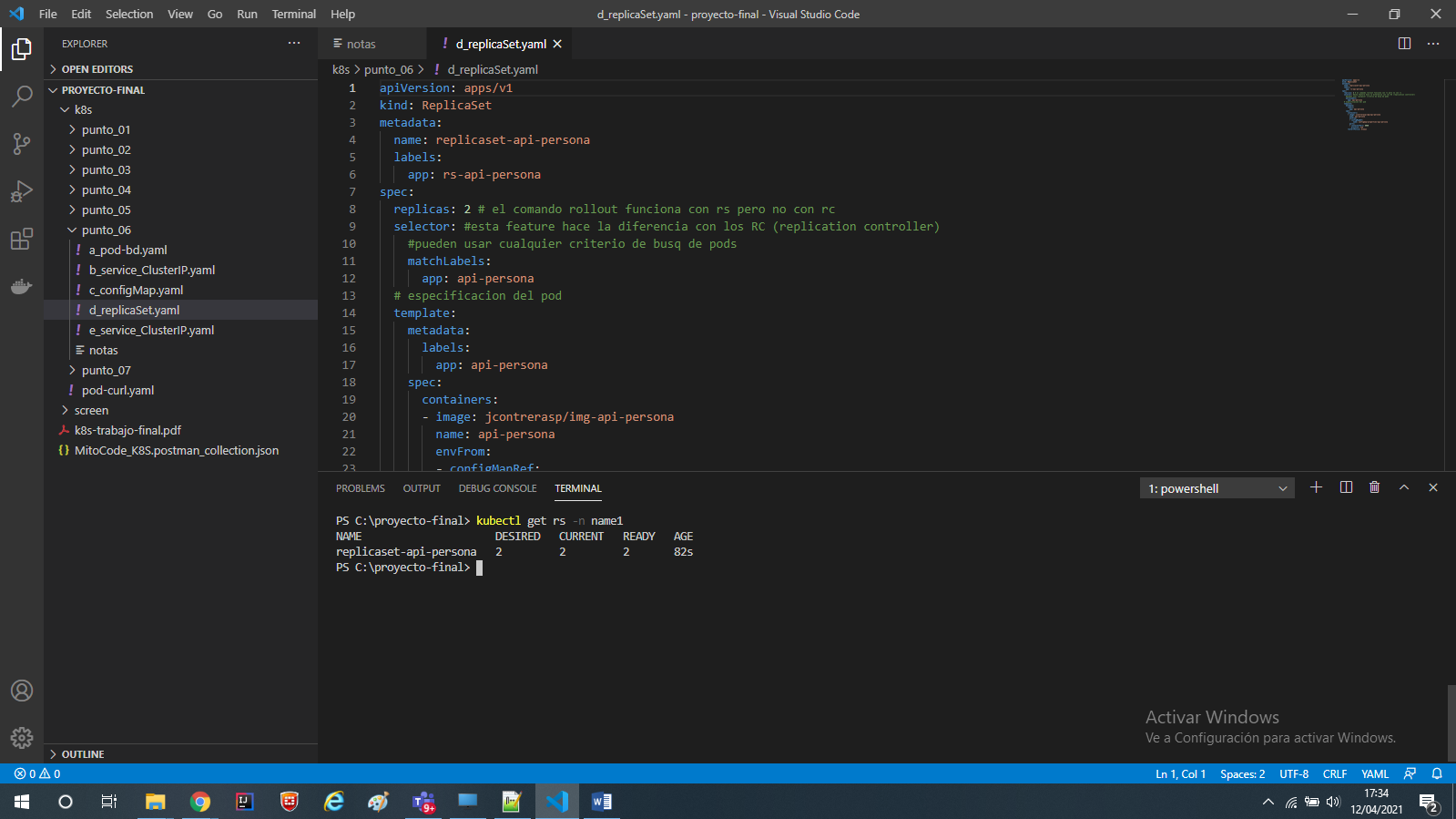
1. **Crear la aplicación en un ReplicaSet con 2 réplicas que use el ConfigMap y se conecte al Service de la base de datos**
2. **Ejecutamos el manifiesto “d\_replicaSet.yaml” de la carpeta k8s/punto\_06**

**kubectl apply -n name1 -f k8s/punto\_06/d\_replicaSet.yaml**



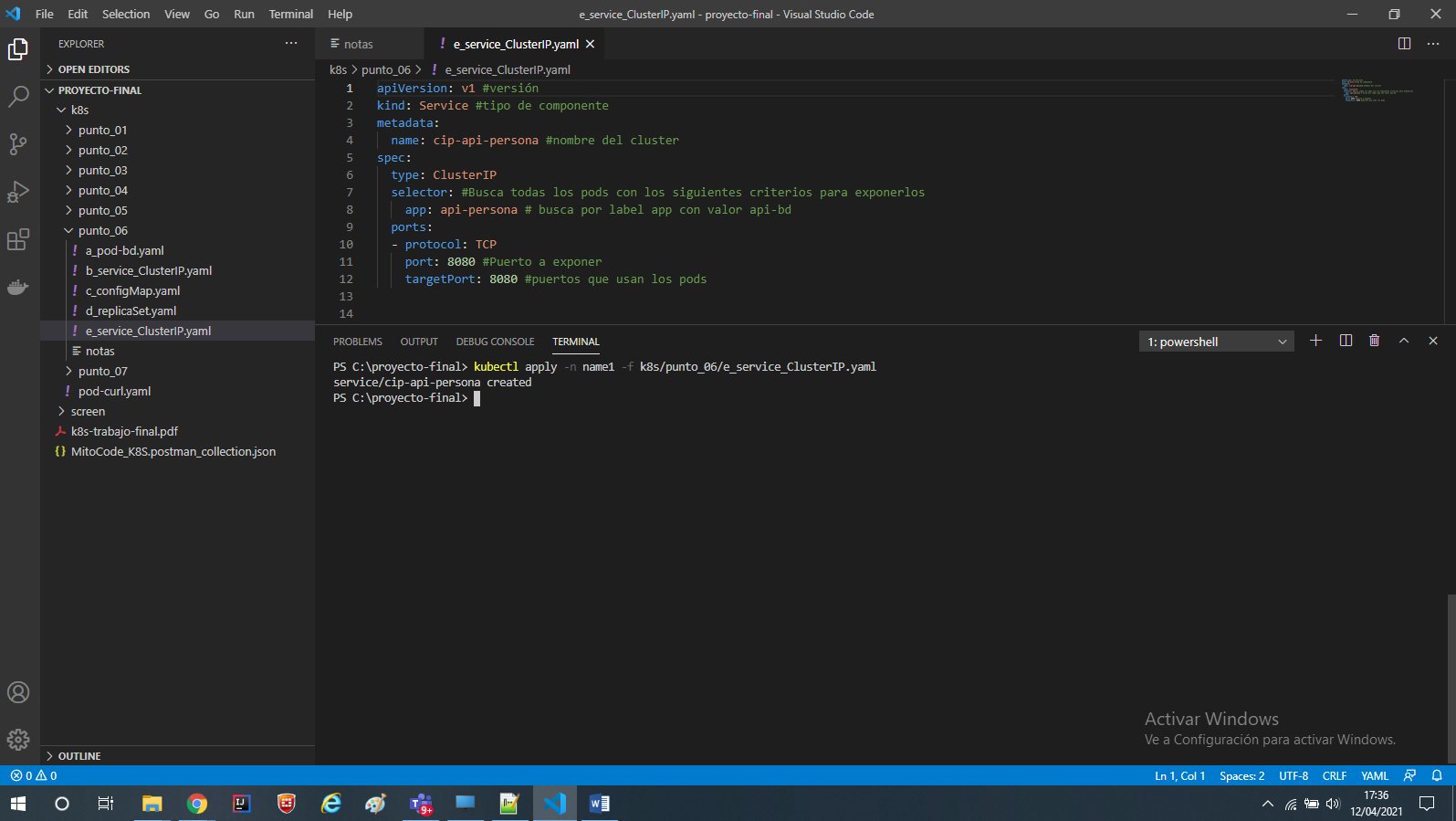
1. **Validamos que se haya creado correctamente**

**kubectl get rs -n name1**



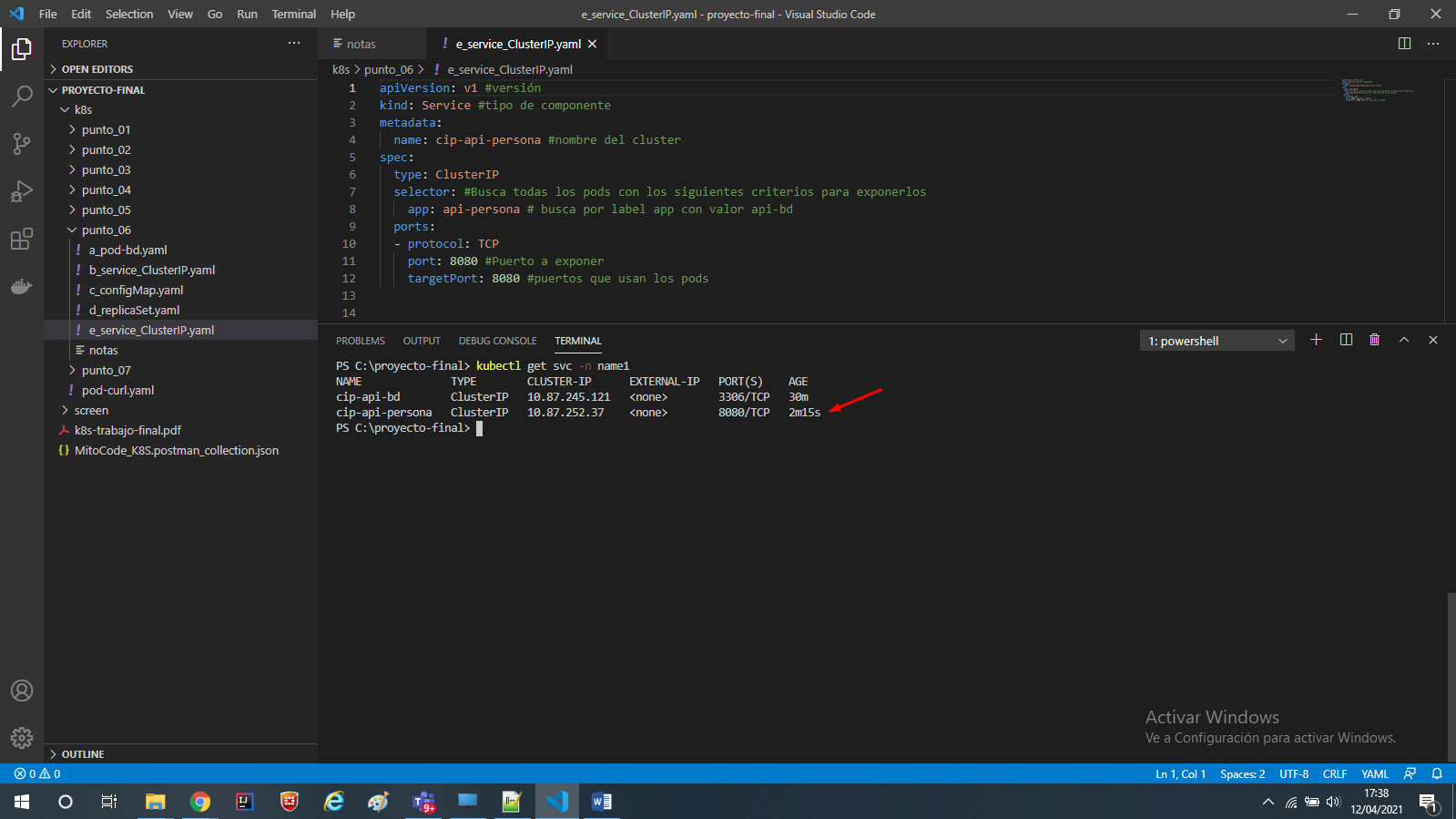
1. **Crear un Service de tipo ClusterIP que permita exponer el aplicativo**
2. **Ejecutamos el manifiesto “e\_service\_ClusterIP.yaml” que se encuentra en la carpeta k8s/punto\_06**

**kubectl apply -n name1 -f k8s/punto\_06/e\_service\_ClusterIP.yaml**



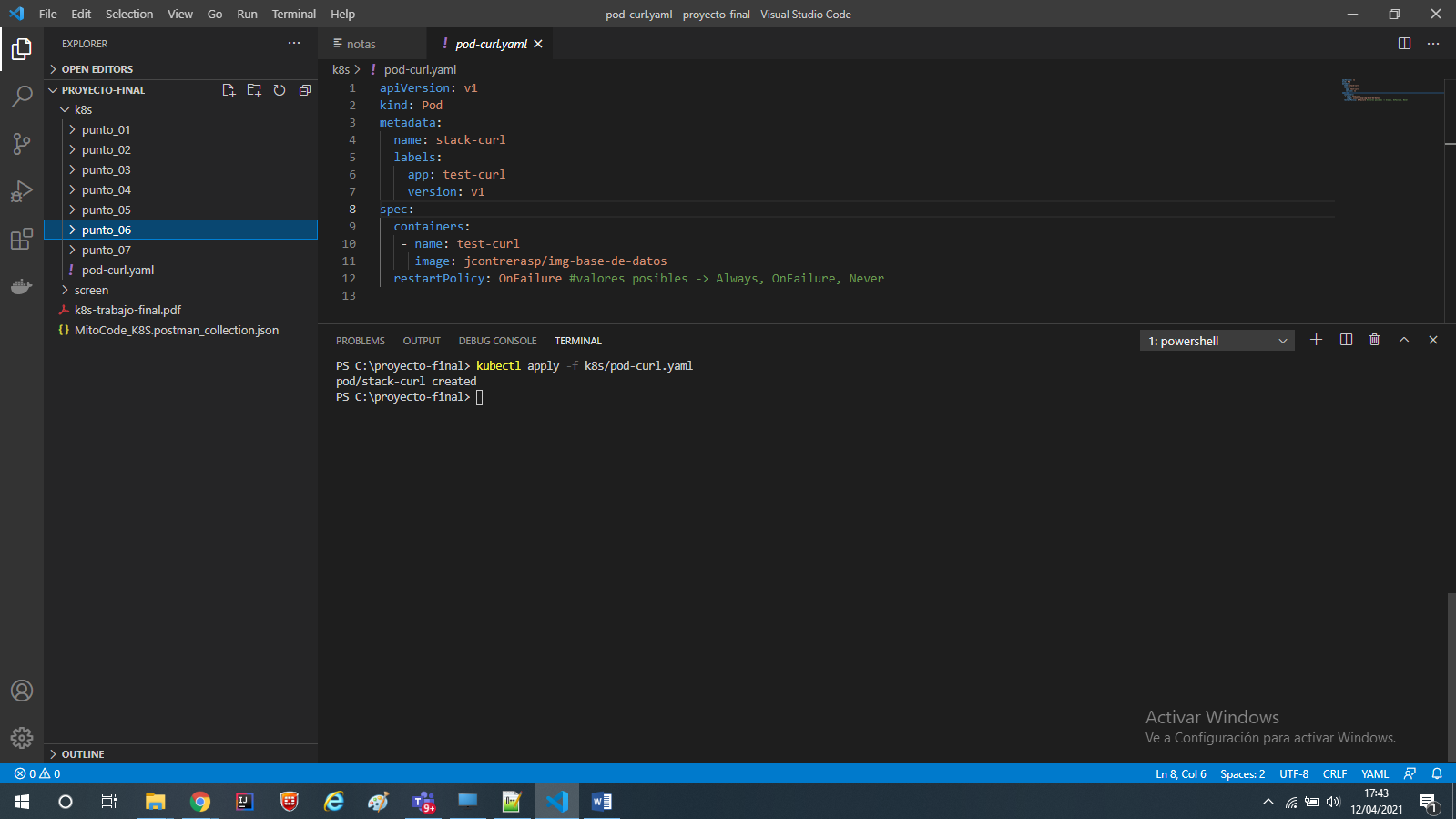
1. **Validamos que se haya creado correctamente**

**kubectl get svc -n name1**



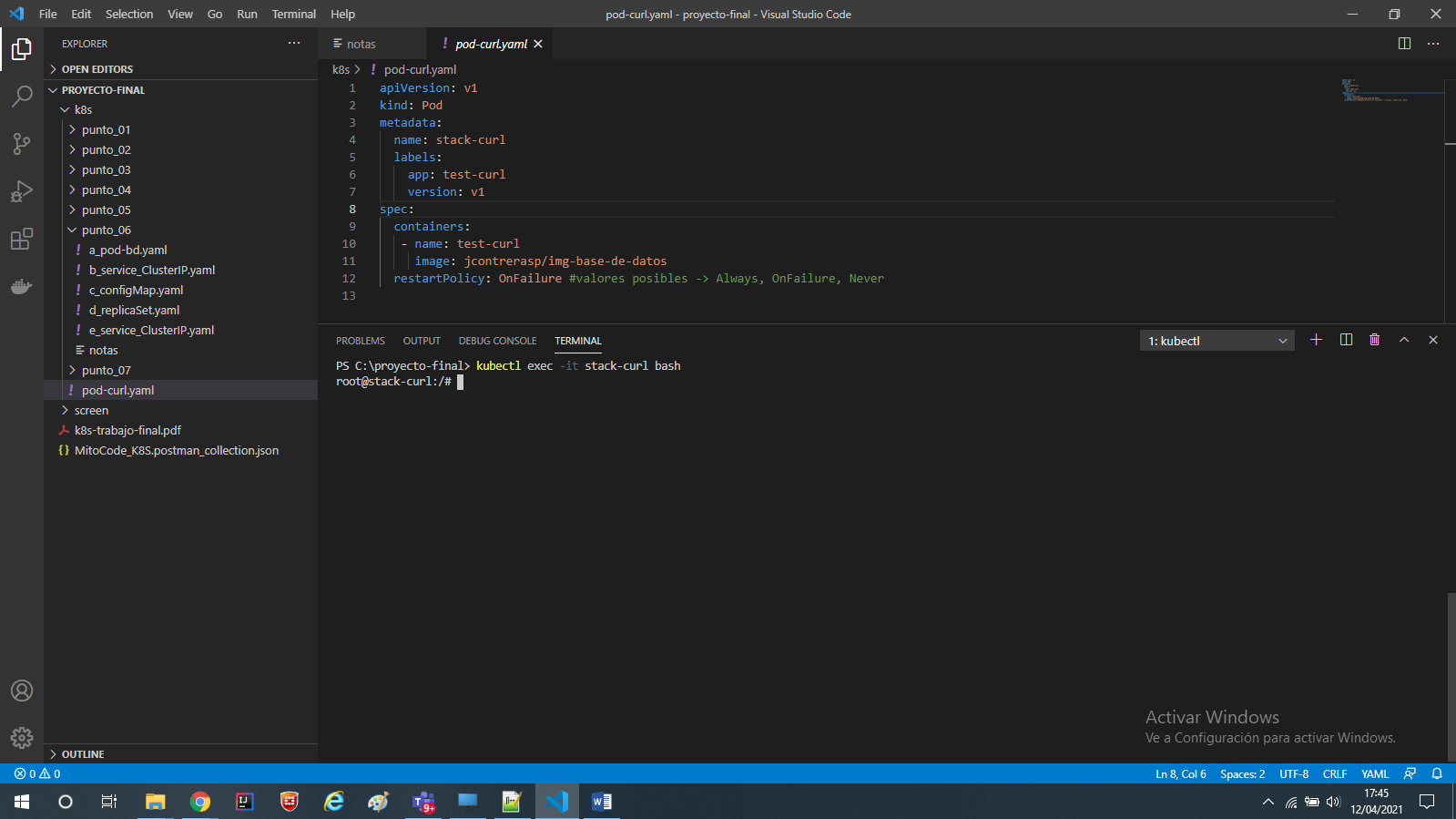
1. **Realizar pruebas de funcionamiento mediante un Pod que contenga un contenedor que permita hacer un curl a los métodos del aplicativo**
2. **Creamos un pod cualquiera para hacer las consultas ejecutando el manifiesto “pod-curl.yaml” que se encuentra dentro de la carpeta k8s**

**kubectl apply -f k8s/pod-curl.yaml**



1. **Ingresamos al bash del container del pod creado**

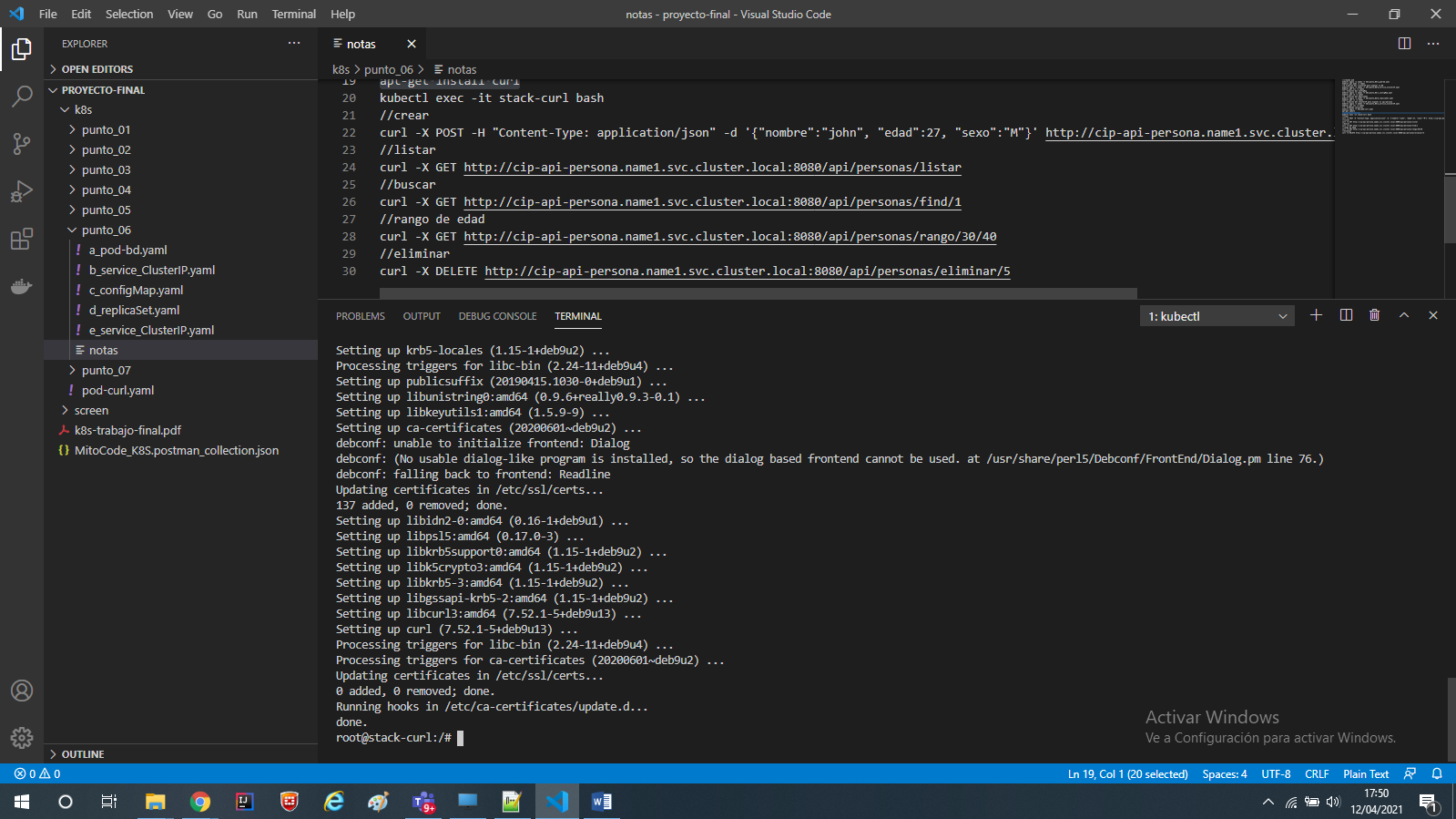
**kubectl exec -it stack-curl bash**



1. **Instalamos el curl**

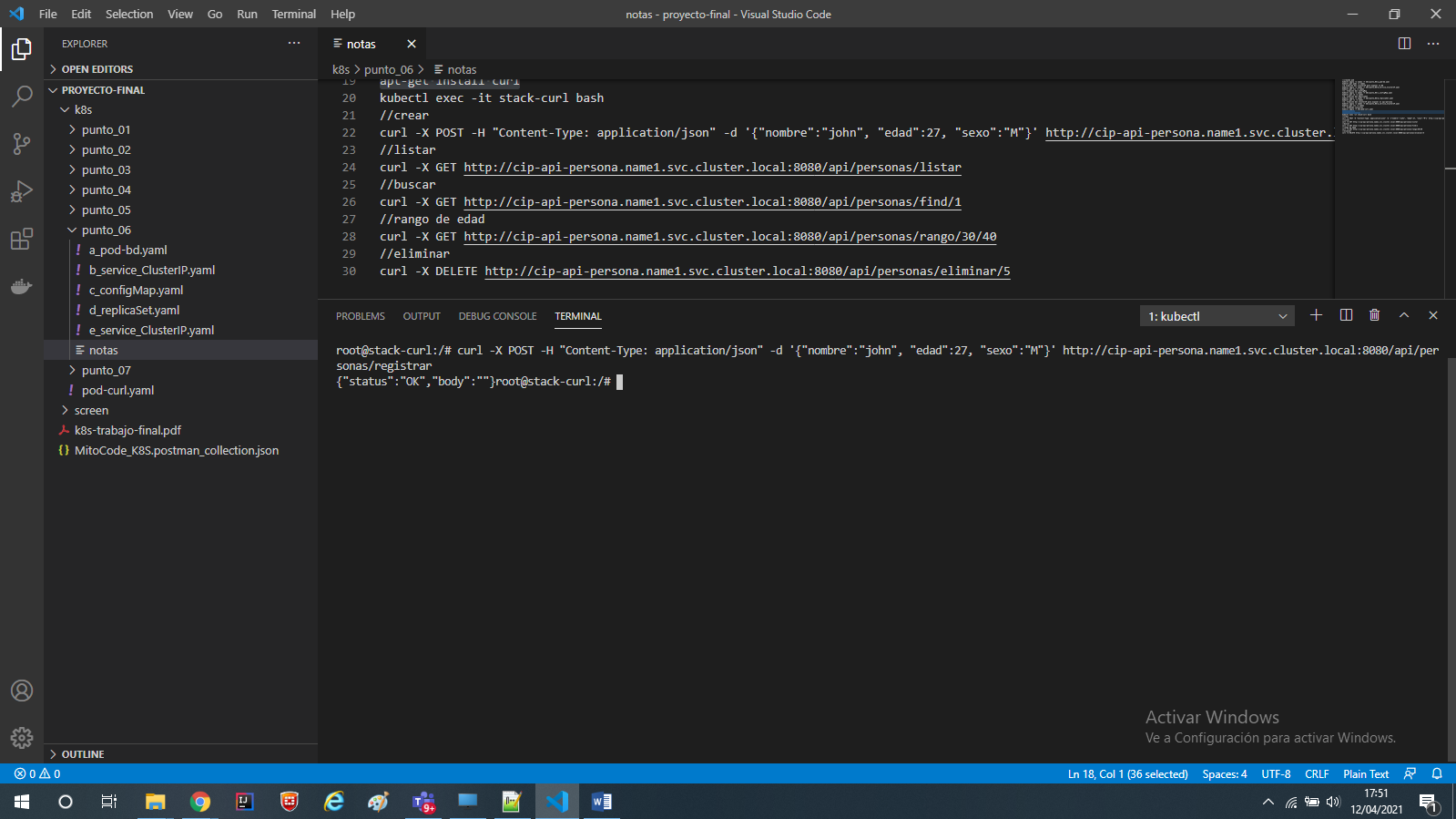
**apt-get update**

**apt-get install curl**



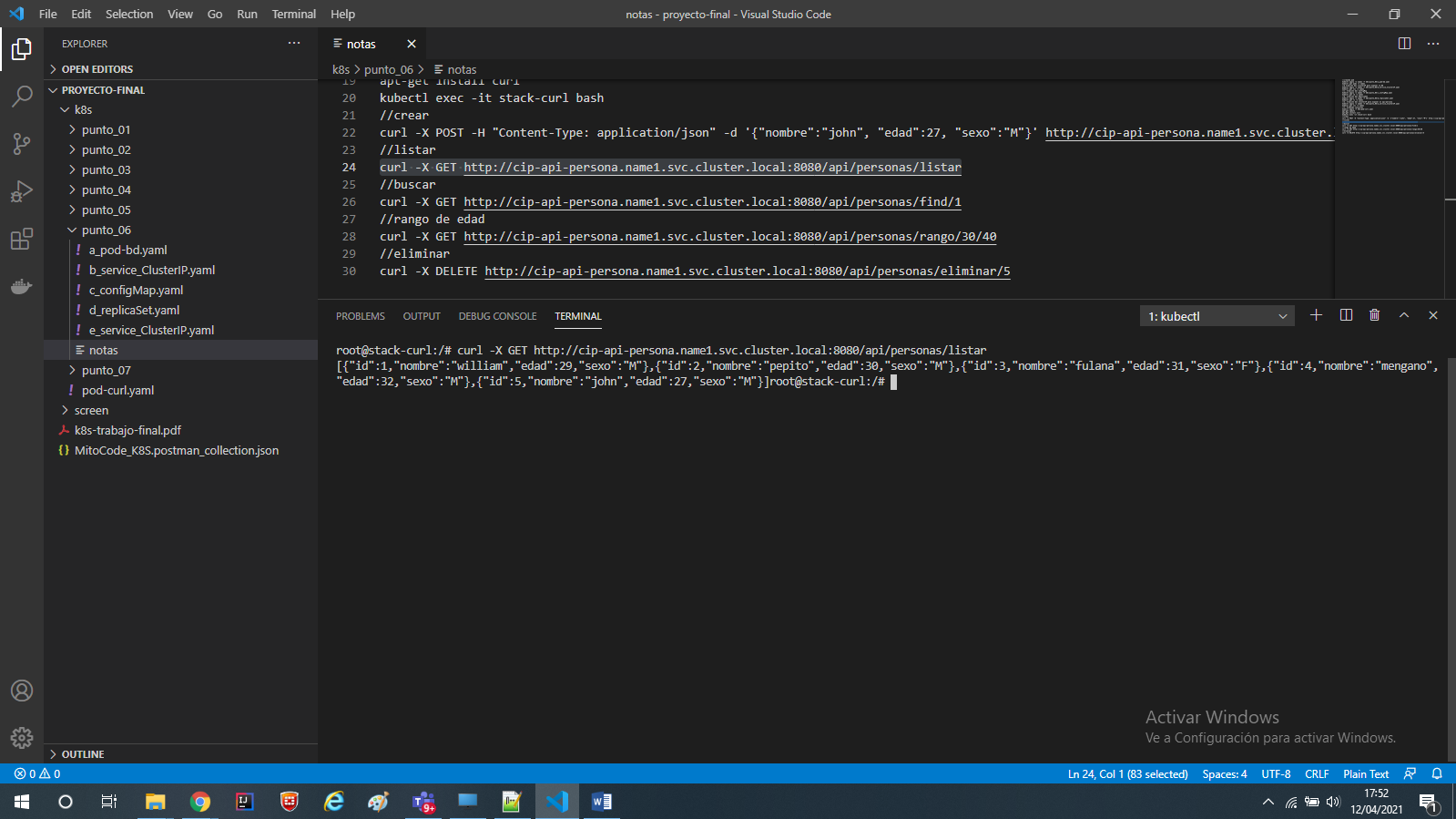
1. **Ejecutamos las pruebas de creación**

**curl -X POST -H "Content-Type: application/json" -d '{"nombre":"john", "edad":27, "sexo":"M"}'** [**http://cip-api-persona.name1.svc.cluster.local:8080/api/personas/registrar**](http://cip-api-persona.name1.svc.cluster.local:8080/api/personas/registrar)



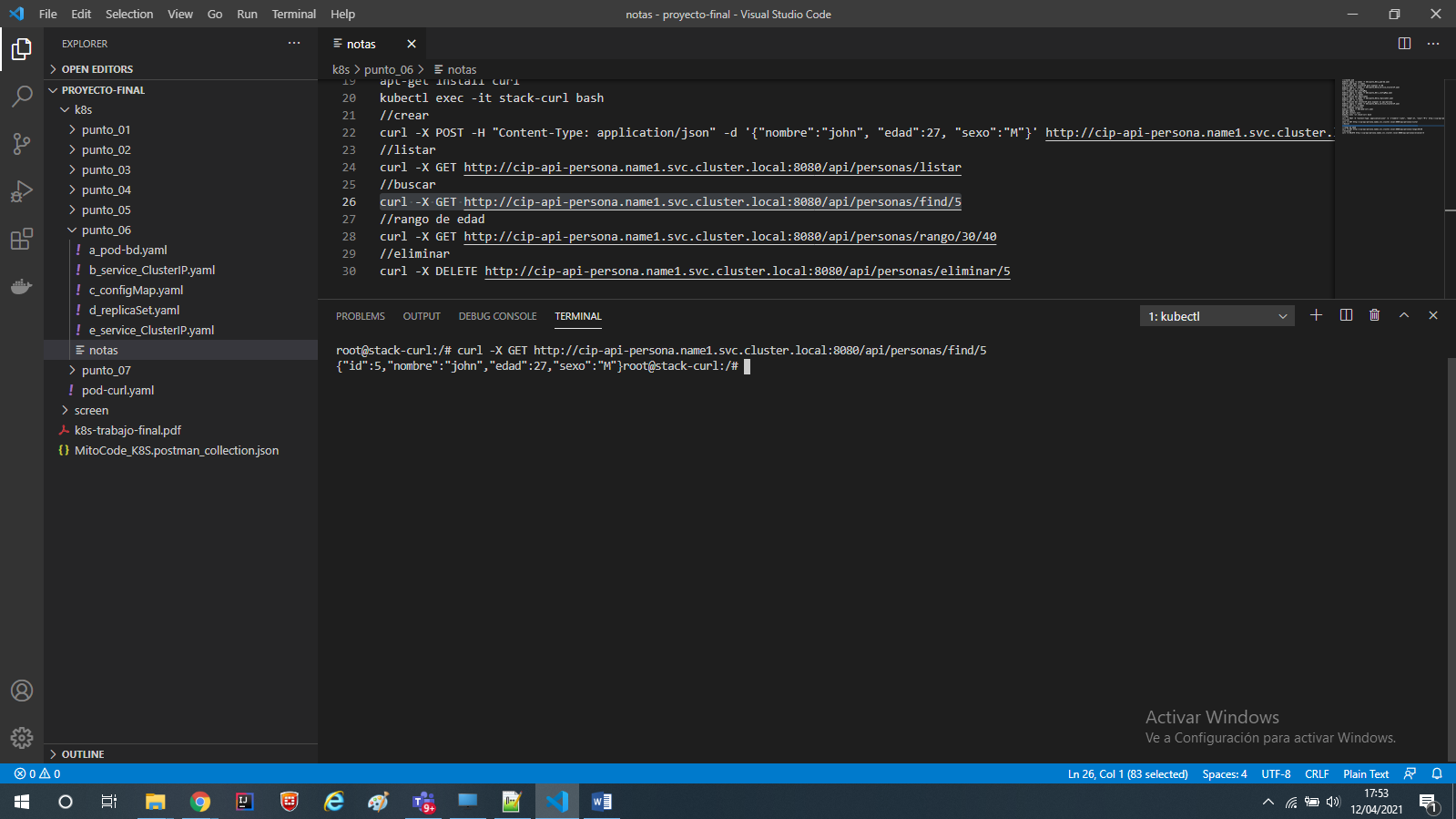
1. **Ejecutamos pruebas de Listar**

**curl -X GET http://cip-api-persona.name1.svc.cluster.local:8080/api/personas/listar**



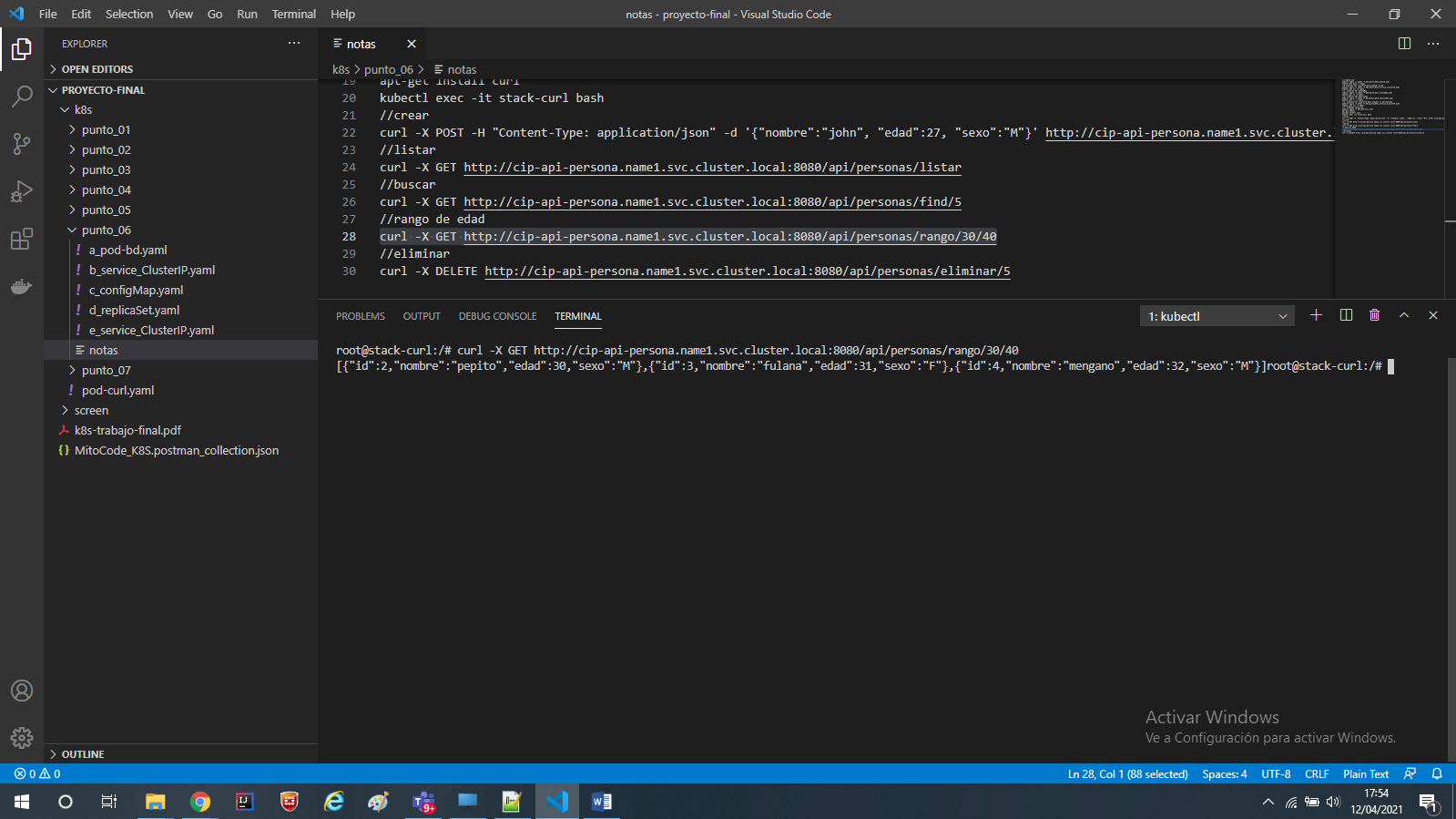
1. **Ejecutamos pruebas de buscar**

**curl -X GET** [**http://cip-api-persona.name1.svc.cluster.local:8080/api/personas/find/5**](http://cip-api-persona.name1.svc.cluster.local:8080/api/personas/find/5)



1. **Ejecutamos pruebas de rango de edad**

**curl -X GET** [**http://cip-api-persona.name1.svc.cluster.local:8080/api/personas/rango/30/40**](http://cip-api-persona.name1.svc.cluster.local:8080/api/personas/rango/30/40)



1. **Ejecutamos pruebas de eliminar**

**curl -X DELETE** [**http://cip-api-persona.name1.svc.cluster.local:8080/api/personas/eliminar/5**](http://cip-api-persona.name1.svc.cluster.local:8080/api/personas/eliminar/5)

