RESOLUCION EXAMEN TECNICO - DARWIN CARDENAS NAVARRO

Instrucciones

Las soluciones de los problemas deberán ser entregadas en un repositorio git y darnos accesos para la revisión

Se está subiendo el proyecto en la ruta:  
https://github.com/DHCARDENASN/test-qsystem

Se crea un microservicio en arquitectura Hexagonal, se utiliza para que el microservicio sea fácil de mantener, escalable y que cumplan con los principios de diseño sólido.

El proyecto se realizó en STS versión 4.20 con java 17, en los ejercicios resueltos se solicita que se muestren objetos, estos no me mostraran en consola sino a través de JSON así que se necesita POSTMAN para la validación, estoy adjuntando en la raíz del repositorio el archivo “collection-test.json” para realizar las peticiones asociadas al examen técnico.

La información que se mostrara será guardada en tiempo real para facilitar el desarrollo utilice Firebase, cree un base de datos en el Cloud FireStore y el archivo de conexión esta embebida en el proyecto en el archivo “serviceAccountKey.json”, para un uso practico esa BD estará disponible hasta el 13/10/2023, luego procederé a eliminarlo.

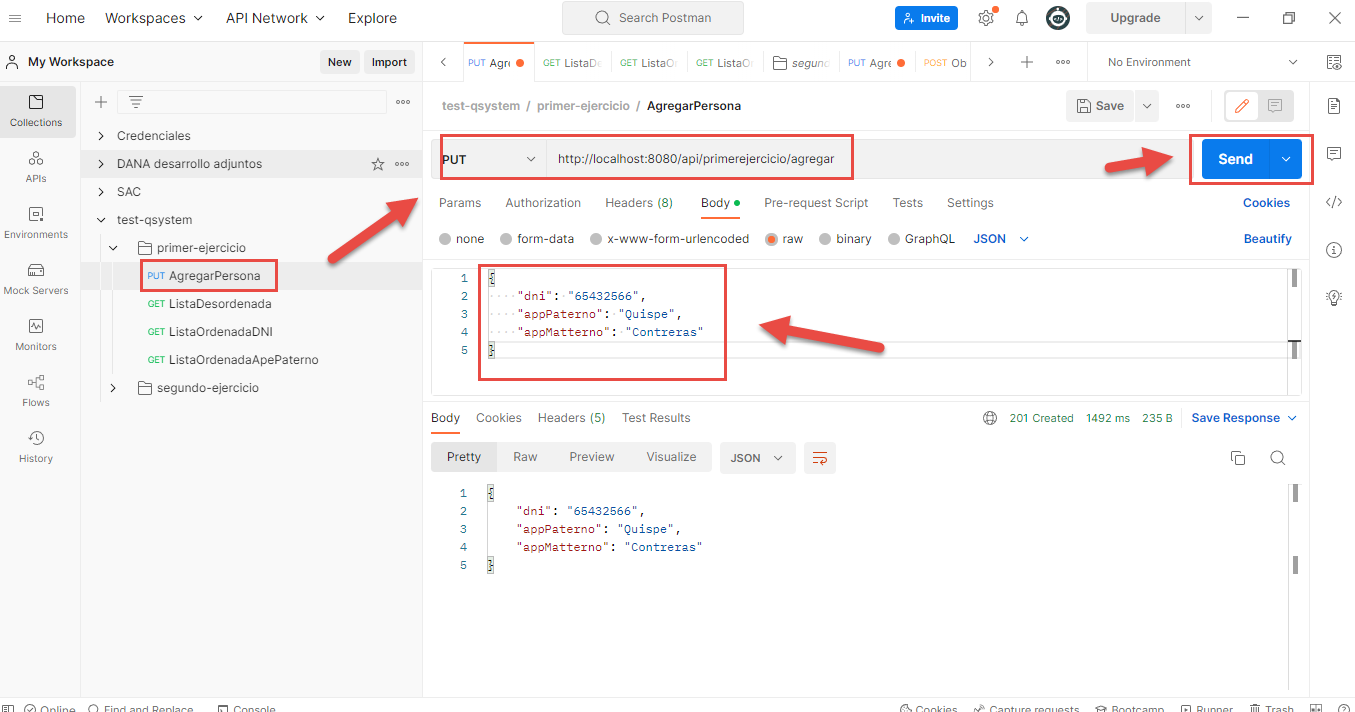
Antes de empezar se tiene que descargar el proyecto y ejecutar el springboot(microservicio) de manera local, tener en cuenta que se usa la versión de java 17.

SOLUCION:

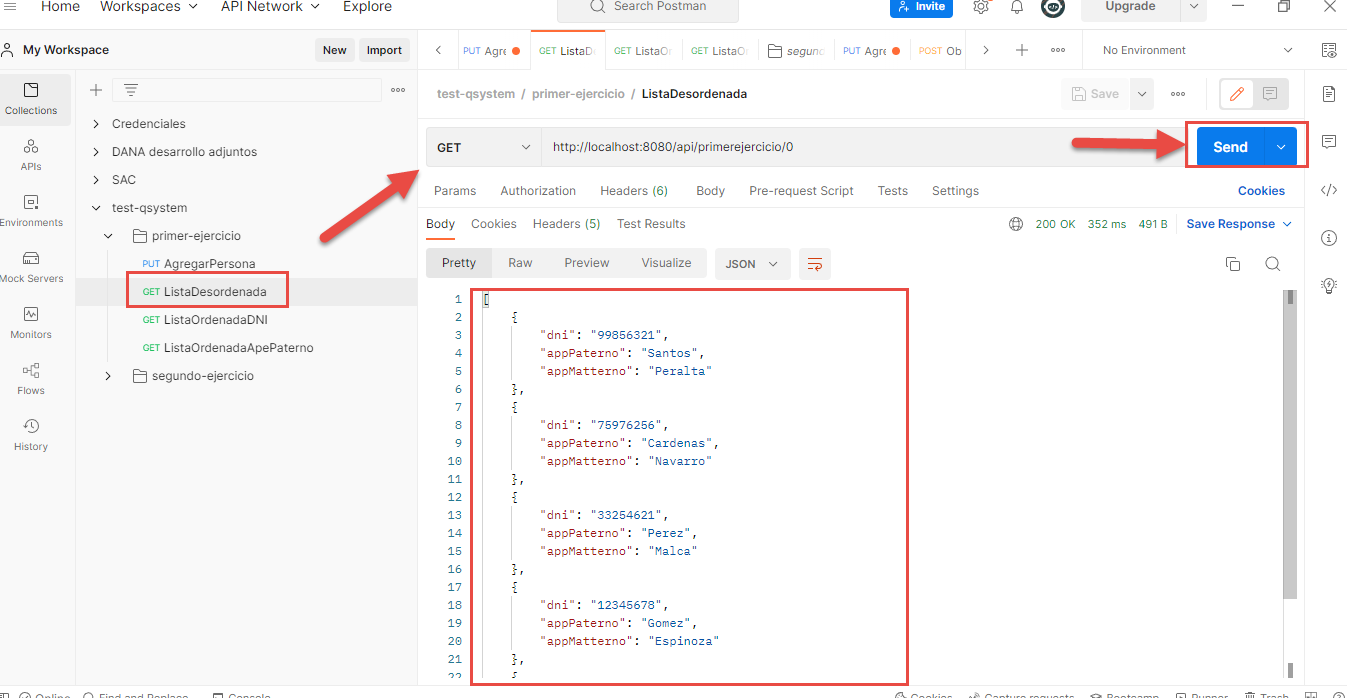
Ejercicio # 1

Crear una lista del Objeto Persona (dni, appPaterno, appMatterno) y deberás ordenar la lista por DNI y de ahí por apellido paterno, pintando la lista desordenada y de ahí ordenada.

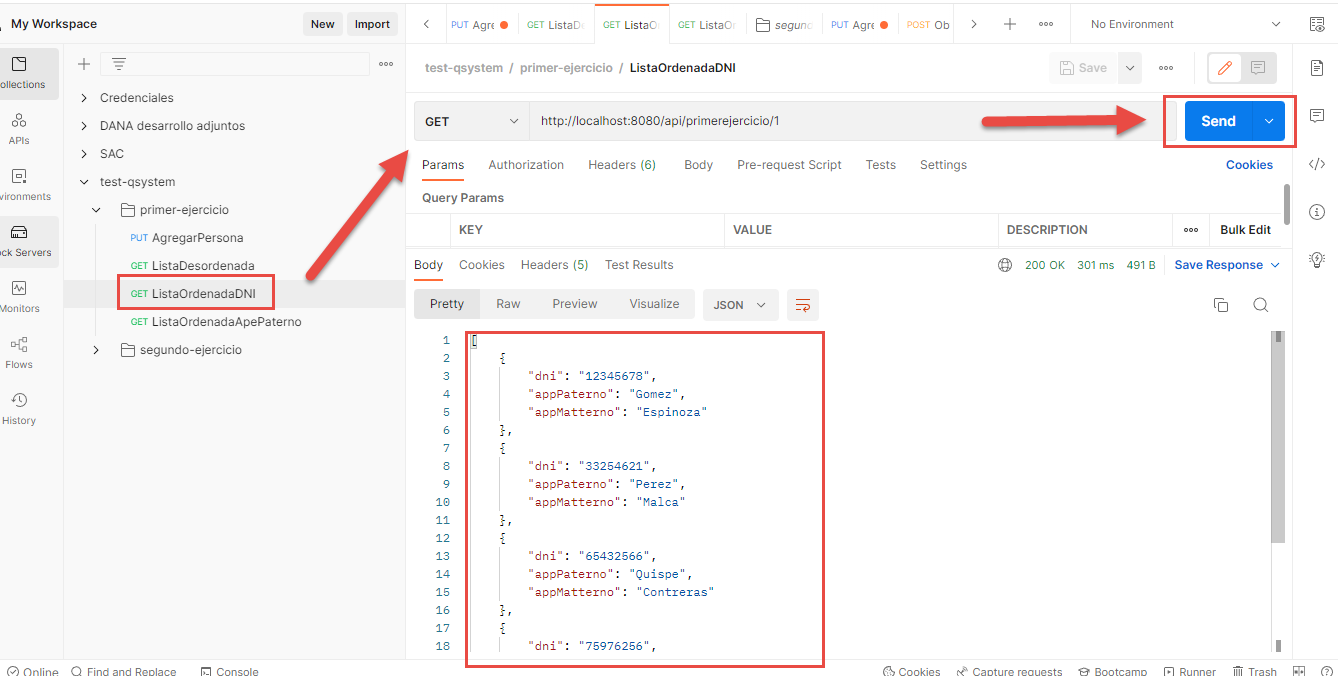
* 1. Se tiene en la collection un servicio para poder agregar mas personas, se agregaron unas cuantas para la resolución.

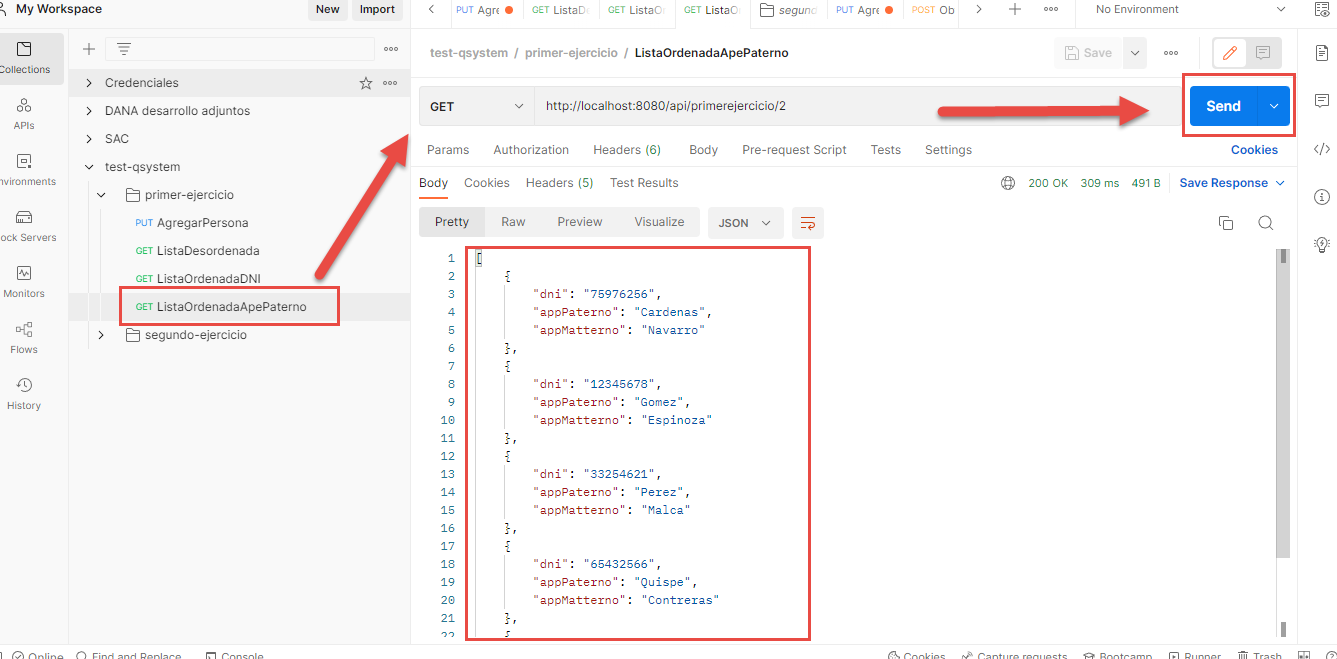


* 1. Una vez que se hayan agregado personas se procede a validar la pregunta que listar de manera desordenada, se ejecuta el request “ListaDesordenada”



* 1. Se solicita listar las personas por DNI, se ejecuta el request “ListaOrdenadaDNI”



* 1. Se solicita listar las personas por apellido paterno, se ejecuta el request “ListaOrdenadaApePaterno”

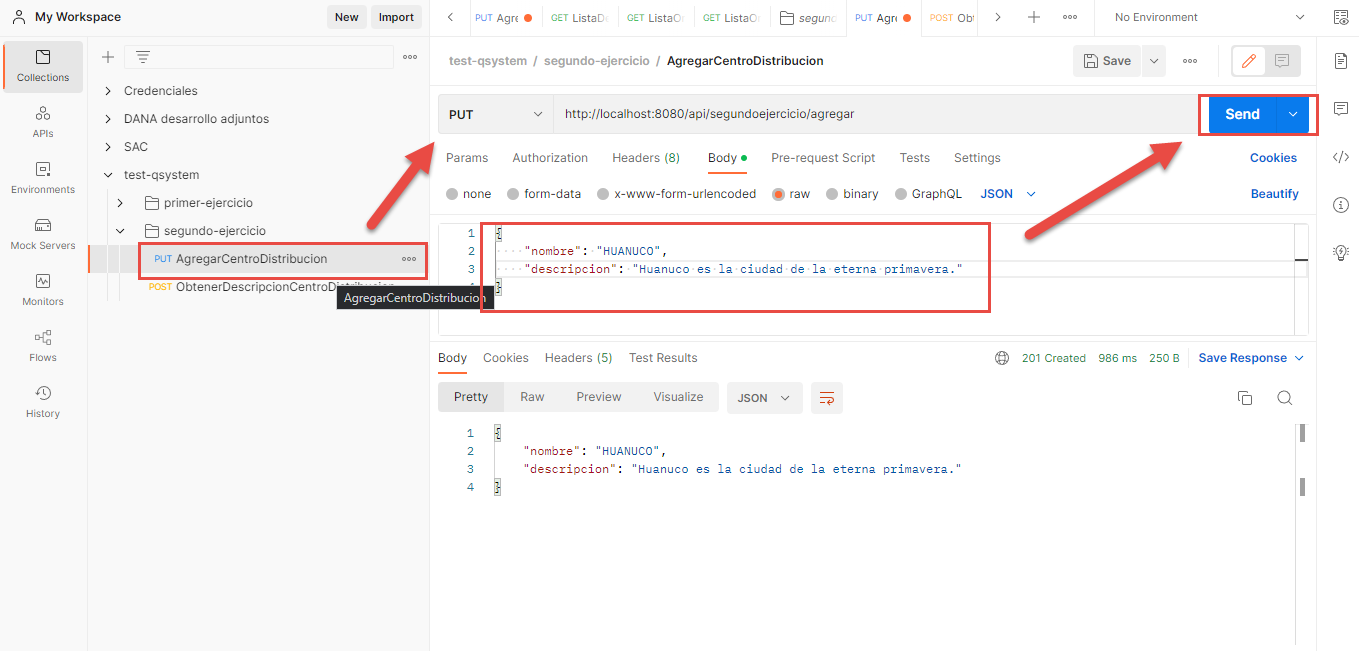
Ejercicio # 2

Se necesita un método que reciba el nombre del centro de distribución y que de acuerdo con el nombre del centro de distribución realice un comportamiento especifico al centro de distribución (puedes pintar un simple texto).

Lista de nombres de los centros de distribución:

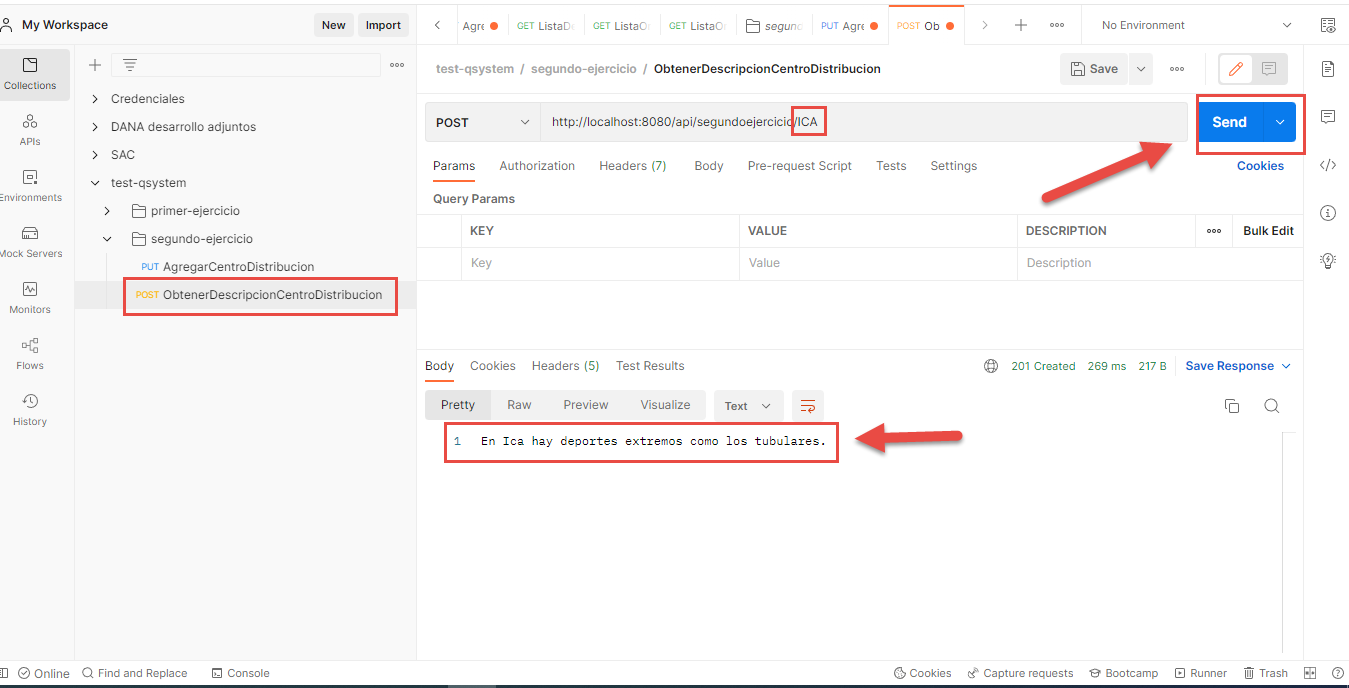
LURIN, ICA, TRUJILLO, HUANUCO

2.1) Se crea un request para poder agregar Centros de distribución adicionales, ya se tiene cargados en Firebase los centros de LURIN, ICA, TRUJILLO, HUANUCO

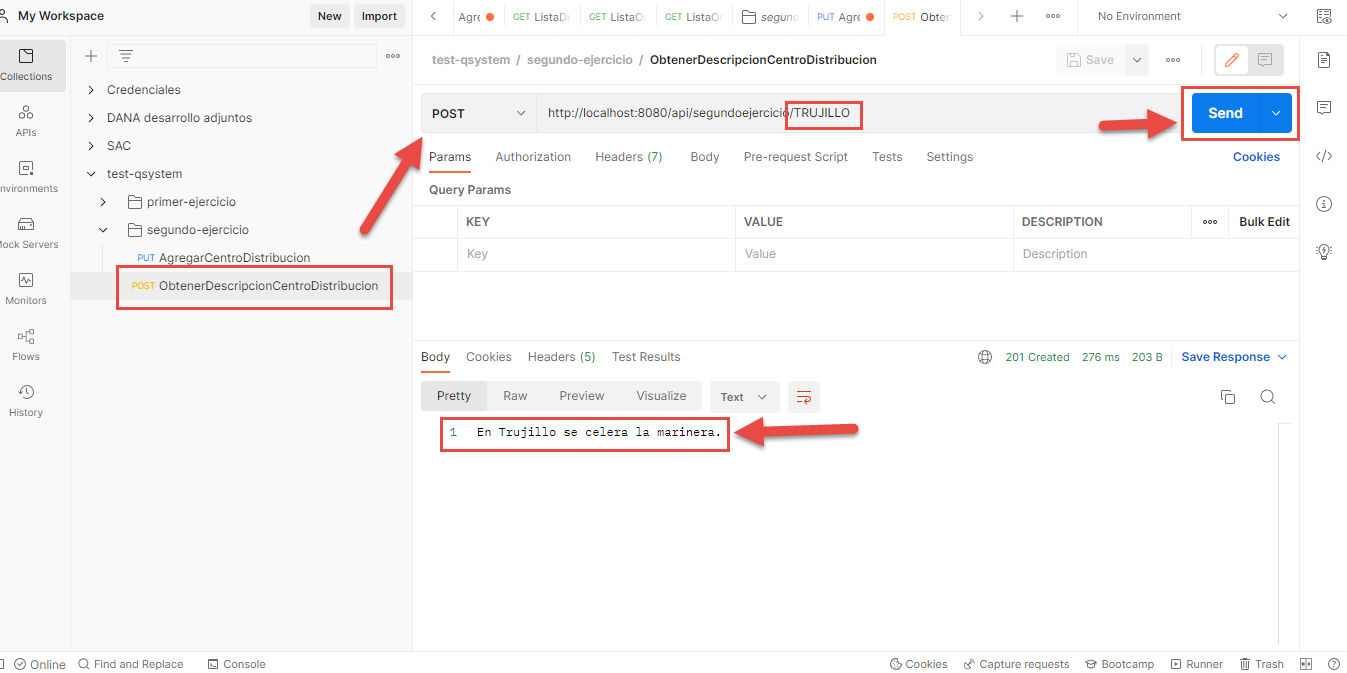


2.2) Según lo solicitado se ingresa el nombre de un centro de distribución y sale un simple texto, el centro de distribución se envia como un @PathVariable en la petición.

Ejemplo1



Ejemplo2



Ejercicio #3

Se necesita que diferentes comportamientos bajo una misma firma(declaración de método) estén accesibles a una aplicación y **puedan ser llamados de manera rápida y clara**.

Esta pregunta se da solución en la estructura del proyecto al ser de Arquitectura Hexagonal ya que se utilizan puertos como interfaces y estas definen las interacciones entre las distintas capas de la aplicación.  
 y **puedan ser llamados de manera rápida y clara** por los adaptadores que son componentes que implementan las interfaces.

