

Aviso

En este documento se define el procedimiento de verificación del funcionamiento del API para interconectarse a la PDN.

El conjunto de pruebas se divide en tres:

- Pruebas de seguridad
- Pruebas funcionales
- Pruebas de estrés

Cada conjunto de pruebas se compone de diversos casos de prueba, en cada caso de prueba se contempla información propia como: identificador, nombre y descripción del caso de prueba, fecha y ambiente de ejecución (Desarrollo o Producción), pasos para ejecutar la prueba, datos de entrada (estos pueden ser diversos y cambiar en cada ejecución), el resultado esperado y el resultado obtenido de la observación de este último dependerá de la evaluación del caso de prueba como exitoso o fallido, pudiendo anotar en cualquier caso las observaciones pertinentes.

Cabe señalar que para la aprobación de la interconexión con la Plataforma Digital Nacional, es necesario contar con todos los casos de prueba evaluados exitosamente. En caso contrario, será necesario realizar los ajustes para cumplir con todos los casos de prueba y solicitar nuevamente la validación por parte del equipo de la Plataforma Digital Nacional.

Datos de la ejecución

Datos de la ejecución			
Fecha de solicitud		Fecha de revisión	
Institución / Dependencia			
Ámbito de gobierno			
URL' s			

Resumen de la ejecución

Resumen de la ejecución		
Pruebas realizadas	Pruebas aprobadas	Pruebas no aprobadas
Estatus de la solicitud de conexión		
Aprobada / No aprobada		

Resultados de la ejecución

Pruebas de seguridad

Caso de prueba			
Número caso	CPDEC-01	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Generación de token		
Descripción	Se valida que se genere correctamente el token		
Pasos de ejecución	1.- Realizar en una herramienta de pruebas una nueva petición post 2.- Se configura en el Body, con un x-www-form-urlencoded los datos proporcionados para la generación del token 3.- Ejecutar		
Datos de entrada	URL de autenticación= grant_type = password username (GT password) = password (GT password) = client_id = client_secret = scope (opcional) =		
Resultado esperado	<pre>{ "token_type": "Bearer", "expires_in": NNNN, "access_token": "access_token", "refresh_token": "refresh_token" }</pre>		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-02	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Verificación de expiración del token		
Descripción	Se valida que el token expire en el tiempo especificado en la respuesta de su generación.		
Pasos de ejecución	1.- Generar un token y anotar el parámetro devuelto “expires_in” 2.- Dividir el periodo de tiempo en 3 (periodo) 3.- Ejecutar una consulta sin parámetros cada periodo 4.- Ejecutar una consulta sin parámetros una vez concluido el tiempo especificado en “expires_in”		
Datos de entrada	Para solicitar el token URL de autenticación= grant_type = username (GT password) = password (GT password) = client_id = client_secret = scope (opcional) = Para los pasos 3 y 4 <pre>{ }</pre>		
Resultado esperado	De la generación del token <pre>{ "token_type": "Bearer", "expires_in": NNNN, "access_token": "access_token", "refresh_token": "refresh_token" }</pre> Del paso 3 https://app.swaggerhub.com/apis/pdn-mx/s1-Declaraciones-Inicial/2.0#/res/Declaraciones Del paso 4 (Indicando acceso no autorizado) https://app.swaggerhub.com/apis/pdn-mx/s1-Declaraciones-Inicial/2.0#/res/Error		
Resultado obtenido			
Observaciones			

Pruebas funcionales

Caso de prueba			
Número caso	CPDEC-03	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Respuesta sin envío de parámetros		
Descripción	Se valida que el API responda cumpliendo con el esquema de declaración de tipo "INICIAL" .		
Pasos de ejecución	1.- Realizar en una herramienta de pruebas una nueva petición post 2.- Se configura en el Body, con raw tipo JSON un objeto vacío 3.- Ejecutar 4.- Se buscan los registros que tengan el valor de tipo "INICIAL" y se validan contra su esquema correspondiente		
Datos de entrada	<pre>{ }</pre>		
Resultado esperado	Esquema en https://app.swaggerhub.com/apis/pdn-mx/s1-Declaraciones-Inicial/2.0#/res/Declaraciones		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-04	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Respuesta sin envío de parámetros		
Descripción	Se valida que el API responda cumpliendo con el esquema de declaración de tipo “MODIFICACIÓN” .		
Pasos de ejecución	1.- Realizar en una herramienta de pruebas una nueva petición post 2.- Se configura en el Body, con raw tipo JSON un objeto vacío 3.- Ejecutar 4.- Se buscan los registros que tengan el valor de tipo “MODIFICACIÓN” y se validan contra su esquema correspondiente		
Datos de entrada	<pre>{ }</pre>		
Resultado esperado	Esquema en https://app.swaggerhub.com/apis/pdn-mx/s1-Declaraciones-Modificacion/2.0#/resDeclaraciones		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-05	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Respuesta sin envío de parámetros		
Descripción	Se valida que el API responda cumpliendo con el esquema de declaración de tipo “CONCLUSIÓN” .		
Pasos de ejecución	1.- Realizar en una herramienta de pruebas una nueva petición post 2.- Se configura en el Body, con raw tipo JSON un objeto vacío 3.- Ejecutar 4.- Se buscan los registros que tengan el valor de tipo “CONCLUSIÓN” y se validan contra su esquema correspondiente		
Datos de entrada	<pre>{ }</pre>		
Resultado esperado	Esquema en https://app.swaggerhub.com/apis/pdn-mx/s1-Declaraciones-Conclusion/2.0#/resDeclaraciones		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-06	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de paginación		
Descripción	Se valida que al enviar una página superior al totalRows/pageSize se obtenga el resultado correcto.		
Pasos de ejecución	1.- Consultar sin parámetros y anotar el totalRows 2.- Calcular pageCalculado = techo(totalRows/10) + 1 3.- Realizar una nueva consulta enviando en el parámetro page el valor calculado		
Datos de entrada	{ "page" : pageCalculado}		
Resultado esperado	<pre> { "pagination": { "hasNextPage": false, "pageSize": 10, "page": pageCalculado, "totalRows": N }, "results": [] } </pre>		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-07	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de pageSize		
Descripción	Se valida que al enviar una pageSize distinta del defecto de la consulta devuelva la cantidad de resultados especificados.		
Pasos de ejecución	1.- Enviar en la petición el atributo pageSize con valor = 1 2.- Enviar en la petición el atributo pageSize con valor = 15 3.- Enviar en la petición el atributo pageSize con valor = 30		
Datos de entrada	{ "pageSize" : N}		
Resultado esperado	<pre> { "pagination": { "hasNextPage": boolean, "pageSize": <pageSize>, "page": 1, "totalRows": N }, "results": [] } </pre>		
Resultado obtenido	En cada una de las ejecuciones, validar que el número de resultados devueltos sea igual al número especificado en pageSize		
Observaciones			

Caso de prueba			
Número caso	CPDEC-08	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por id		
Descripción	Se valida que filtre por id		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo id , extraído de un registro previamente obtenido 3.- Ejecutar		
Datos de entrada	<pre>{ "query": { "id": "idPrueba" } }</pre>		
Resultado esperado	<pre>"pagination": { "hasNextPage": false, "pageSize": 10, "page": 1, "totalRows": 1 }, "results": [{...}]</pre>		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-09	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por id inexistente		
Descripción	Se valida que el filtro por id funcione correctamente al enviar un id inexistente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo id con un valor aleatorio no contenido en los datos 3.- Ejecutar		
Datos de entrada	<pre>{ "query": { "id": "idPrueba" } }</pre>		
Resultado esperado	<pre>"pagination": { "hasNextPage": false, "pageSize": 10, "page": 1, "totalRows": 0 }, "results": []</pre>		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-10	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por nombres		
Descripción	Se valida que filtre por nombres		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo nombres 3.- Ejecutar		
Datos de entrada	<pre>{ "query":{ "nombres":"<Nombre>" } }</pre>		
Resultado esperado	<p>En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada.</p> <p>En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío.</p> <p>De igual forma, que el resultado en pagination sea congruente con la información devuelta.</p>		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-11	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por nombres incompleto		
Descripción	Se valida que filtre por nombres cómo like		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo nombres con un valor previamente extraído y anotar el totalRows 3.- Enviar en query el atributo nombres eliminando las últimas 2 o 3 letras del valor previamente utilizado y anotar el totalRows		
Datos de entrada	<pre>{ "query":{ "nombres": "<Nombre>"/ "<Letras>" } }</pre>		
Resultado esperado	El totalRows de la ejecución del paso 3, debe ser igual o mayor al totalRows de la ejecución del paso 2.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-12	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por nombres sin case sensitive.		
Descripción	Se valida que la consulta por nombres no sea case sensitive.		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo nombres con un valor previamente extraído y anotar el totalRows . 3.-Enviar en query el atributo nombres con el mismo valor del paso 2 mezclando minúsculas y mayúsculas ejecutar y anotar el totalRows		
Datos de entrada	<pre>{ "query":{ "nombres": "NOMbre" } }</pre>		
Resultado esperado	Todas las ejecuciones deben devolver el mismo valor en totalRows		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-13	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por nombres con y sin acento.		
Descripción	Se valida la consulta por nombres con y sin acento.		
Pasos de ejecución	1.-Enviar en query el atributo nombres con acento. 2.-Ejecutar y anotar el totalRows 3.-Enviar en query el atributo nombres sin acento 4.-Ejecutar y anotar el totalRows .		
Datos de entrada	<pre>{ "query":{ "nombres": "<nombreConAcento>" / "<nombreSinAcento>" } }</pre>		
Resultado esperado	En ambas ejecuciones debe ser el mismo valor de totalRows		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-14	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por primerApellido		
Descripción	Se valida el funcionamiento del filtro primerApellido		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo primerApellido 3.- Ejecutar		
Datos de entrada	<pre>{ "query" : { "primerApellido" : "<Apellido>" } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-15	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por primerApellido incompleto		
Descripción	Se valida el funcionamiento del filtro primerApellido como like		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo primerApellido con un valor previamente extraído y anotar el totalRows 3.- Enviar en query el atributo primerApellido eliminando las últimas 2 o 3 letras del valor previamente utilizado y anotar el totalRows		
Datos de entrada	<pre>{ "query":{ "primerApellido": "<primerApellido>/ "<Letras>" } }</pre>		
Resultado esperado	El totalRows de la ejecución del paso 3, debe ser igual o mayor al totalRows de la ejecución del paso 2.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-16	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por primerApellido sin case sensitive.		
Descripción	Se valida que la consulta por primerApellido no sea case sensitive.		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo primerApellido con un valor previamente extraído y anotar el totalRows. 3.-Enviar en query el atributo nombres con el mismo valor del paso 2 mezclando minúsculas y mayúsculas ejecutar y anotar el totalRows		
Datos de entrada	<pre>{ "query":{ "primerApellido": "<PRImerAPEllIDO>" } }</pre>		
Resultado esperado	Todas las ejecuciones deben devolver el mismo valor en totalRows		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-17	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por primerApellido con y sin acento.		
Descripción	Se valida la consulta por primerApellido con y sin acento.		
Pasos de ejecución	1.-Enviar en query el atributo primerApellido con acento. 2.-Ejecutar y anotar el totalRows 3.-Enviar en query el atributo primerApellido sin acento 4.-Ejecutar y anotar el totalRows.		
Datos de entrada	<pre>{ "query":{ "primerApellido": "<primerApellidoConAcento>" / "<primerApellidoSinAcento>" } }</pre>		
Resultado esperado	En ambas ejecuciones debe ser el mismo valor de totalRows		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-18	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por segundoApellido		
Descripción	Se valida el funcionamiento del filtro segundoApellido		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo segundoApellido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query" : { "segundoApellido" : "<segundoApellido>" } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-19	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por segundoApellido incompleto		
Descripción	Se valida el funcionamiento del filtro segundoApellido cómo like		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo segundoApellido con un valor previamente extraído y anotar el totalRows 3.- Enviar en query el atributo segundoApellido eliminando las últimas 2 o 3 letras del valor previamente utilizado y anotar el totalRows		
Datos de entrada	<pre>{ "query":{ "segundoApellido": "<segundoApellido>"/ "<Letras>" } }</pre>		
Resultado esperado	El totalRows de la ejecución del paso 3, debe ser igual o mayor al totalRows de la ejecución del paso 2.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-20	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por segundoApellido sin case sensitive.		
Descripción	Se valida que la consulta por segundoApellido no sea case sensitive.		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo segundoApellido con un valor previamente extraído y anotar el totalRows. 3.-Enviar en query el atributo segundoApellido con el mismo valor del paso 2 mezclando minúsculas y mayúsculas ejecutar y anotar el totalRows		
Datos de entrada	<pre>{ "query":{ "segundoApellido": "<SEgunDOapeLLIdo>" } }</pre>		
Resultado esperado	Todas las ejecuciones deben devolver el mismo valor en totalRows		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-21	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por segundoApellido con y sin acento.		
Descripción	Se valida la consulta por segundoApellido con y sin acento.		
Pasos de ejecución	1.-Enviar en query el atributo segundoApellido con acento. 2.-Ejecutar y anotar el totalRows 3.-Enviar en query el atributo segundoApellido sin acento 4.-Ejecutar y anotar el totalRows.		
Datos de entrada	<pre>{ "query":{ "segundoApellido": "<segundoApellidoConAcento>" / "<segundoApellidoSinAcento>" } }</pre>		
Resultado esperado	En ambas ejecuciones debe ser el mismo valor de totalRows		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPSS-22	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por nombres y primerApellido.		
Descripción	Se valida que filtre por nombres y primerApellido.		
Pasos de ejecución	1.- Se crea una petición post. 2.- Enviar en el query el atributo nombres y primerApellido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query":{ "nombres":"<Nombre>", "primerApellido":"<primerApellido>" } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPSS-23	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por nombres y segundoApellido.		
Descripción	Se valida que filtre por nombres y segundoApellido.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo nombres y segundoApellido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query":{ "nombres":"<Nombre>", "segundoApellido":"<segundoApellido>" } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPSS-24	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por primerApellido y segundoApellido.		
Descripción	Se valida que filtre por primerApellido y segundoApellido.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo primerApellido y segundoApellido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query":{ "primerApellido":"<primerApellido>", "segundoApellido":"<segundoApellido>" } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPSS-25	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por nombre, primerApellido y segundoApellido.		
Descripción	Se valida que filtre por nombre, primerApellido y segundoApellido.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo nombre, primerApellido y segundoApellido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query":{ "nombres":"<Nombre>", "primerApellido":"<primerApellido>", "segundoApellido":"<segundoApellido>" } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPSS-26	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por escolaridadNivel.		
Descripción	Se valida el funcionamiento del filtrado por escolaridadNivel.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo escolaridadNivel, con algunos de los valores posibles que se encuentran dentro de la sección datosCurricularesDeclarante->escolaridad->nivel 3.- Ejecutar.		
Datos de entrada	<pre>{ "query":{ "escolaridadNivel": "<EscolaridadNivel>" } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-27	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por escolaridadNivel inexistente.		
Descripción	Se valida que el filtro por escolaridadNivel funcione correctamente al enviar una clave inexistente.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo escolaridadNivel con un valor aleatorio no válido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "escolaridadNivel": "escolaridadNivelPrueba" } }</pre>		
Resultado esperado	<pre>"pagination": { "hasNextPage": false, "pageSize": 10, "page": 1, "totalRows": 0 }, "results": []</pre>		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-28	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por nombreEntePublico.		
Descripción	Se valida que la consulta por nombreEntePublico funcione correctamente.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo nombreEntePublico 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "nombreEntePublico": "<nombreEntePublico>" } } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-29	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por nombreEntePublico incompleto.		
Descripción	Se valida la consulta por nombreEntePublico cómo like.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo nombreEntePublico con un valor previamente extraído y anotar el totalRows. 3.- Enviar en query el atributo nombreEntePublico eliminando las últimas 2 o 3 letras del valor previamente utilizado y anotar el totalRows.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "nombreEntePublico": "<nombreEntePublico>/ "<Letras>" } } }</pre>		
Resultado esperado	El totalRows de la ejecución del paso 3, debe ser igual o mayor al totalRows de la ejecución del paso 2.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-30	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por nombreEntePublico con y sin acento.		
Descripción	Se valida la consulta por nombreEntePublico con y sin acento.		
Pasos de ejecución	1.-Enviar en query el atributo nombreEntePublico con acento. 2.-Ejecutar y anotar el totalRows 3.-Enviar en query el atributo nombreEntePublico sin acento 4.-Ejecutar y anotar el totalRows.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "nombreEntePublico": "<nombreEntePublicoSinAcento>" / "<nombreEntePublicoConAcento>" } } }</pre>		
Resultado esperado	En ambas ejecuciones debe ser el mismo valor de totalRows		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-31	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por nombreEntePublico sin case sensitive.		
Descripción	Se valida que la consulta por nombreEntePublico no sea case sensitive.		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo nombreEntePublico con un valor previamente extraído y anotar el totalRows. 3.-Enviar en query el atributo nombreEntepublico con el mismo valor del paso 2 mezclando minúsculas y mayúsculas ejecutar y anotar el totalRows		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "nombreEntePublico": "<NObreENtePUBliCO>" } } }</pre>		
Resultado esperado	Todas las ejecuciones deben devolver el mismo valor de totalRows.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-32	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por entidadFederativa.		
Descripción	Se valida que el filtro por entidadFederativa funcione correctamente.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo entidadFederativa, extraído de un registro previamente obtenido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "entidadFederativa": "<entidadFederativa>" } } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-33	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por entidadFederativa inexistente.		
Descripción	Se valida que el filtro por entidadFederativa funcione correctamente al enviar un valor inexistente.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo entidadFederativa con un valor aleatorio no válido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "entidadFederativa": "entidadFederativaPrueba" } } }</pre>		
Resultado esperado	<pre>"pagination": { "hasNextPage": false, "pageSize": 10, "page": 1, "totalRows": 0 }, "results": []</pre>		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-34	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por municipioAlcaldia.		
Descripción	Se valida que el filtro por municipioAlcaldia funcione correctamente.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo municipioAlcaldia, extraído de un registro previamente obtenido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "municipioAlcaldia": "<municipioAlcaldia>" } } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-35	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por municipioAlcaldia inexistente.		
Descripción	Se valida que el filtro por municipioAlcaldia funcione correctamente al enviar un valor inexistente.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo municipioAlcaldia con un valor aleatorio no válido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "municipioAlcaldia": "municipioAlcaldiaPrueba" } } }</pre>		
Resultado esperado	<pre>"pagination": { "hasNextPage": false, "pageSize": 10, "page": 1, "totalRows": 0 }, "results": []</pre>		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-36	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por EmpleoCargoComision.		
Descripción	Se valida que la consulta por EmpleoCargoComision funcione correctamente.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo EmpleoCargoComision 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "empleoCargoComision": "<EmpleoCargoComision>" } } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-37	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por EmpleoCargoComision incompleto.		
Descripción	Se valida la consulta por EmpleoCargoComision cómo like.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo EmpleoCargoComision con un valor previamente extraído y anotar el totalRows. 3.- Enviar en query el atributo EmpleoCargoComision eliminando las últimas 2 o 3 letras del valor previamente utilizado y anotar el totalRows.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "empleoCargoComision": "<EmpleoCargoComision>/ "<Letras>" } } }</pre>		
Resultado esperado	El totalRows de la ejecución del paso 3, debe ser igual o mayor al totalRows de la ejecución del paso 2.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-38	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por EmpleoCargoComision con y sin acento.		
Descripción	Se valida la consulta por EmpleoCargoComision con y sin acento.		
Pasos de ejecución	1.-Enviar en query el atributo EmpleoCargoComision con acento. 2.-Ejecutar y anotar el totalRows 3.-Enviar en query el atributo EmpleoCargoComision sin acento 4.-Ejecutar y anotar el totalRows.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "empleoCargoComision": "<EmpleoCargoComisionConAcento>" / "<EmpleoCargoComisionSinAcento>" } } }</pre>		
Resultado esperado	En ambas ejecuciones debe ser el mismo valor de totalRows		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-39	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por EmpleoCargoComision sin case sensitive.		
Descripción	Se valida que la consulta por EmpleoCargoComision no sea case sensitive.		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo EmpleoCargoComision con un valor previamente extraído y anotar el totalRows. 3.-Enviar en query el atributo EmpleoCargoComision con el mismo valor del paso 2 mezclando minúsculas y mayúsculas, ejecutar y anotar el totalRows		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "empleoCargoComision": "<EMpleoCARgoCOmISION>" } } }</pre>		
Resultado esperado	Ambas ejecuciones deben devolver el mismo valor de totalRows.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-40	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por nivelOrdenGobierno.		
Descripción	Se valida que filtre por nivelOrdenGobierno.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo nivelOrdenGobierno. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "nivelOrdenGobierno": "nivelOrdenGobierno" } } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-41	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por nivelEmpleoCargoComision.		
Descripción	Se valida que la consulta por nivelEmpleoCargoComision funcione correctamente.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo nivelEmpleoCargoComision 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "nivelEmpleoCargoComision": "<nivelEmpleoCargoComision>" } } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-42	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por nivelEmpleoCargoComision incompleto.		
Descripción	Se valida la consulta por nivelEmpleoCargoComision cómo like.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo nivelEmpleoCargoComision un valor previamente extraído y anotar el totalRows. 3.- Enviar en query el atributo nivelEmpleoCargoComision las primeras 2 o 3 letras del valor previamente utilizado y anotar el totalRows.		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "nivelEmpleoCargoComision": "<nivelEmpleoCargoComision>"/ "<Letras>" } } }</pre>		
Resultado esperado	El totalRows de la ejecución del paso 3, debe ser igual o mayor al totalRows de la ejecución del paso 2.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-43	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por nivelEmpleoCargoComision sin case sensitive.		
Descripción	Se valida que la consulta por nivelEmpleoCargoComision no sea case sensitive.		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo nivelEmpleoCargoComision con un valor previamente extraído y anotar el totalRows. 3.-Enviar en query el atributo nivelEmpleoCargoComision con el mismo valor del paso 2 mezclando minúsculas y mayúsculas ejecutar y anotar el totalRows		
Datos de entrada	<pre>{ "query": { "datosEmpleoCargoComision": { "nivelEmpleoCargoComision": "<nivelEmpleoCargoComision>" } } }</pre>		
Resultado esperado	Ambas ejecuciones deben devolver el mismo valor de totalRows.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-44	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por superficieConstruccion		
Descripción	Se valida que la consulta por superficieConstruccion funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo superficieConstruccion con un valor mínimo, previamente extraído de un registro obtenido 3.- Ejecutar 4.- Enviar en query el atributo superficieConstruccion con un valor máximo, previamente extraído de un registro obtenido 5.- Ejecutar 6.-Enviar en query el atributo superficieConstruccion con un valor mínimo y máximo, previamente extraído de un registro obtenido 7.-Ejecutar		
Datos de entrada	<pre>{ "query": { "bienesInmuebles": { "superficieConstruccion":{"min":valor}/{ "max":valor} / { "min":valo,"max":valor} } } }</pre>		
Resultado esperado	Al comparar los resultados de las ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-45	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por superficieTerreno		
Descripción	Se valida que la consulta por superficieTerreno funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo superficieTerreno con un valor mínimo, previamente extraído de un registro obtenido 3.- Ejecutar 4.- Enviar en query el atributo superficieTerreno con un valor máximo, previamente extraído de un registro obtenido 5.- Ejecutar 6.-Enviar en query el atributo superficieTerreno con un valor mínimo y máximo, previamente extraído de un registro obtenido 7.-Ejecutar		
Datos de entrada	<pre>{ "query": { "bienesInmuebles": { "superficieTerreno":{"min":valor}/{ "max":valor} / { "min":valo,"max":valor} } } }</pre>		
Resultado esperado	Al comparar los resultados de las ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-46	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por valorAdquisicion		
Descripción	Se valida que la consulta por valorAdquisicion funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo valorAdquisicion con un valor mínimo, previamente extraído de un registro obtenido 3.- Ejecutar 4.- Enviar en query el atributo valorAdquisicion con un valor máximo, previamente extraído de un registro obtenido 5.- Ejecutar 6.-Enviar en query el atributo valorAdquisicion con un valor mínimo y máximo, previamente extraído de un registro obtenido 7.-Ejecutar		
Datos de entrada	<pre>{ "query": { "bienesInmuebles": { "valorAdquisicion":{"min":valor}/{ "max":valor} / { "min":valo,"max":valor} } } }</pre>		
Resultado esperado	Al comparar los resultados de las ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-47	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de consulta por totalIngresosNetos		
Descripción	Se valida que la consulta totalIngresosNetos funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en query el atributo totalIngresosNetos con un valor mínimo, previamente extraído de un registro obtenido 3.- Ejecutar 4.- Enviar en query el atributo totalIngresosNetos con un valor máximo, previamente extraído de un registro obtenido 5.- Ejecutar 6.-Enviar en query el atributo totalIngresosNetos con un valor mínimo y máximo, previamente extraído de un registro obtenido 7.-Ejecutar		
Datos de entrada	<pre>{ "query":{ "totalIngresosNetos":{"min":valor}/{ "max":valor} / {"min":valo,"max":valor} } }</pre>		
Resultado esperado	Al comparar los resultados de las ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-48	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por formaAdquisicion.		
Descripción	Se valida el funcionamiento del filtrado por formaAdquisicion.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo formaAdquisicion, con algunos de los valores posibles que se encuentran dentro de la sección formaAdquisicion->clave 3.- Ejecutar.		
Datos de entrada	<pre>{ "query": { "bienesInmuebles": { "formaAdquisicion": "<formaAdquisicion>" } } }</pre>		
Resultado esperado	En caso de haber registros, se valida que la respuesta cumpla el estándar, y que cada ítem cumpla con la condición solicitada. En caso de no haber registros que cumplan la condición, se espera que devuelva un arreglo vacío. De igual forma, que el resultado en pagination sea congruente con la información devuelta.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-49	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Filtrado por formaAdquisicion inexistente.		
Descripción	Se valida que el filtro de formaAdquisicion funcione correctamente al enviar una clave inexistente.		
Pasos de ejecución	1.- Crear una petición post. 2.- Enviar en query el atributo formaAdquisicion con un valor aleatorio no válido. 3.- Ejecutar.		
Datos de entrada	<pre>{ "query":{ "bienesInmuebles":{ "formaAdquisicion": "Letras" } } }</pre>		
Resultado esperado	<pre>"pagination": { "hasNextPage": false, "pageSize": 10, "page": 1, "totalRows": 0 }, "results": []</pre>		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-50	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por nombres		
Descripción	Se valida que el ordenamiento por nombres funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo nombres de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo nombres de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort":{ "nombres":"asc"/"desc" } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-51	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por primerApellido		
Descripción	Se valida que el ordenamiento por primerApellido funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo primerApellido de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo primerApellido de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort":{ "primerApellido":"asc"/"desc" } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-52	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por segundoApellido		
Descripción	Se valida que el ordenamiento por segundoApellido funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo segundoApellido de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo segundoApellido de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort":{ "segundoApellido":"asc"/"desc" } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-53	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por escolaridadNivel		
Descripción	Se valida que el ordenamiento por escolaridadNivel funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo escolaridadNivel de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo escolaridadNivel de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort":{ "escolaridadNivel":"asc"/"desc" } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-54	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por nombreEntePublico		
Descripción	Se valida que el ordenamiento por nombreEntePublico funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo nombreEntePublico de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo nombreEntePublico de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "datosEmpleoCargoComision": { "nombreEntePublico": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-55	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por entidadFederativa		
Descripción	Se valida que el ordenamiento por entidadFederativa funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo entidadFederativa de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo entidadFederativa de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "datosEmpleoCargoComision": { "entidadFederativa": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-56	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por municipioAlcaldia		
Descripción	Se valida que el ordenamiento por municipioAlcaldia funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo municipioAlcaldia de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo municipioAlcaldia de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "datosEmpleoCargoComision": { "municipioAlcaldia": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-57	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por empleoCargoComision		
Descripción	Se valida que el ordenamiento por empleoCargoComision funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo empleoCargoComision de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo empleoCargoComision de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "datosEmpleoCargoComision": { "empleoCargoComision": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-58	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por nivelEmpleoCargoComision		
Descripción	Se valida que el ordenamiento por nivelEmpleoCargoComision funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo nivelEmpleoCargoComision de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo nivelEmpleoCargoComision de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "datosEmpleoCargoComision": { "nivelEmpleoCargoComision": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-59	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por nivelOrdenGobierno		
Descripción	Se valida que el ordenamiento por nivelOrdenGobierno funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo nivelOrdenGobierno de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo nivelOrdenGobierno de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "datosEmpleoCargoComision": { "nivelOrdenGobierno": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-60	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por totalIngresosNetos		
Descripción	Se valida que el ordenamiento totalIngresosNetos funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo totalIngresosNetos de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo totalIngresosNetos de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort":{ "totalIngresosNetos":"asc"/"desc" } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-61	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por superficieConstruccion		
Descripción	Se valida que el ordenamiento por superficieConstruccion funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo superficieConstruccion de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo superficieConstruccion de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "bienesInmuebles": { "superficieConstruccion": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-62	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por superficieTerreno		
Descripción	Se valida que el ordenamiento por superficieTerreno funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo superficieTerreno de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo superficieTerreno de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "bienesInmuebles": { "superficieTerreno": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-63	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por formaAdquisicion		
Descripción	Se valida que el ordenamiento por formaAdquisicion funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo formaAdquisicion de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo formaAdquisicion de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "bienesInmuebles": { "formaAdquisicion": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Caso de prueba			
Número caso	CPDEC-64	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de ordenamiento por valorAdquisicion		
Descripción	Se valida que el ordenamiento por valorAdquisicion funcione correctamente		
Pasos de ejecución	1.- Crear una petición post 2.- Enviar en sort el atributo valorAdquisicion de manera ascendente 3.- Ejecutar 4.- Enviar en sort el atributo valorAdquisicion de manera descendente 5.- Ejecutar		
Datos de entrada	<pre>{ "sort": { "bienesInmuebles": { "valorAdquisicion": "asc"/"desc" } } }</pre>		
Resultado esperado	Al comparar los resultados de ambas ejecuciones, se debe verificar que hay cambios en el conjunto de resultados devueltos, además de respetar el orden solicitado.		
Resultado obtenido			
Observaciones			

Pruebas de estrés

Caso de prueba			
Número caso	CPDEC-65	Ambiente:	Desarrollo / Producción
Evaluación	Exitoso/ Fallido	Fecha de ejecución	
Nombre	Validación de carga		
Descripción	Se valida enviando un volumen de peticiones concurrentes soportadas en un lapso de tiempo determinado.		
Pasos de ejecución	1.- Configurar la prueba con los datos de entrada y ejecutar.		
Datos de entrada	Peticiones = 2000, concurrencia = 100, tiempo = 10 segundos, body { }		
Resultado esperado	Cada petición enviada se debe resolver de manera exitosa dentro del tiempo especificado, soportando además el nivel de concurrencia especificado.		
Resultado obtenido			
Observaciones			