

PDE

Plataforma Digital Estatal

Plan de pruebas
de las APIs de las dependencias



Contenido

Pruebas de las APIs de la dependencia	1
Objetivo	1
Introducción	1
Resultados de la ejecución	2
Pruebas de seguridad	2
Generación de token	2
Validar la expiración del token	3
Pruebas funcionales	4
Path de acceso /psancionados	4
El método Post	4
El método Get	4
El método Put	5
Validar que el dato de entrada "fechaCaptura" contenga solo números, guion medio y que tenga el formato de AAAA-MM-DD	5
Validar que el dato de entrada "fechaCaptura" no contenga caracteres especiales a excepción del guion medio	6
Validar que el dato de entrada "limite_registros" contenga solo números	6
Validar que el dato de entrada "limite_registros" no contenga letras	7
Validar que el dato de entrada "limite_registros" que cuando este vacío muestre 10 registros ..	7
Validar que el dato de entrada "limite_registros" que cuando coloque 0 muestre 10 registros...	7
Validar que el dato de entrada "limite_registros" no contenga caracteres especiales	8
Validar que el dato de entrada "pagina" contenga solo números	8
Validar que el dato de entrada "pagina" no contenga letras	9
Validar que el dato de entrada "pagina" no contenga caracteres especiales	9
Validar que el dato de entrada "pagina" que cuando coloque 0 muestre la página 1	10
Validar que el dato de entrada "pagina" que cuando este vacío muestre la página 1	10
Validar que el dato de entrada "pagina" que cuando coloque el último número de la página ya no muestre otra página con registros	10
Validar que "fechaCaptura", "limite_registros", "pagina", "hasNextPage", "totalRows", "pageSize", "page" sean correctos	11

Mostrar que el atributo “page” sea el mismo que el dato de entrada “pagina”	13
Validar que el servicio responda correctamente en formato Json	13
Validar que el tamaño de la página sea correcto.....	14
Validar que muestre la página solicitada	14
Validar que el total de registros sean correctos	14
Validar que la página actual cambie a la siguiente página	15
Validar que la operación no sea exitosa y muestre el esquema “ResError”	16
Validar que ingrese a la página a través de https	16
Validar que no ingrese a la página a través de http.....	17
Validar el certificado SSL	17
Validar el API con el esquema de Psancionados	17

Pruebas de las APIs de la dependencia

Sistema de particulares sancionados. Versión 1.

Objetivo

En este documento se define el procedimiento de verificación del funcionamiento del API de la dependencia.

Introducción

El conjunto de pruebas se divide en dos:

1. Pruebas de seguridad
2. Pruebas funcionales

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 las pruebas de la API, 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.

DATOS DE LA EJECUCIÓN	
Fecha de solicitud	Periodo de revisión
Institución / Dependencia	
Ámbito de gobierno	
URLs	
Revisión no.	
Ambiente	Desarrollo Producción

RESUMEN DE LA EJECUCIÓN		
Pruebas realizadas	Pruebas aprobadas	Pruebas no aprobadas

Realizó:

Resultados de la ejecución

Pruebas de seguridad

Generación de token	
Número caso	PDE- PS3-01 Evaluación Exitoso Fallido
Objetivo de la funcionalidad	Generar correctamente el token
Precondiciones	
Resultado esperado	Generar el token en formato Json Responde con los atributos: "access_token": "expires_in": "token_type": "scope": "refresh_token":
Instrucciones	1. Realizar en una herramienta de pruebas una nueva petición post

	<ol style="list-style-type: none"> 2. Ingresar la URL de token 3. Se configura en el Body, con el formato x-www-form-urlencoded los datos proporcionados para la generación del token 4. Hacer clic en el botón "Send" 5. Mostrará un token
Datos de entrada	URL de autenticación= grant_type = username = password = client_id= client_secret= scope =
Resultado obtenido	
Observaciones	

Validar la expiración del token

Número caso	PDE- PS3-02	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Validar que el token expire en 5 minutos en la respuesta de su generación		
Precondiciones	Generar y aplicar token		
Resultado esperado	Se valida que se genere el tiempo del token, con un tiempo de 5 min		
Instrucciones	<ol style="list-style-type: none"> 1. Verificar que el atributo "expires_in" tenga el valor de 300 2. Ingresar a la URL de datos 3. Seleccionar el método Post 4. Configurar el token <ol style="list-style-type: none"> 4.1. Seleccionar en la URL de datos la pestaña de Authorization 4.2. En el apartado Type desplegará un menú en el cual tendrá que seleccionar la opción OAuth 2.0 4.3. Pegar el token de acceso en el apartado de access token 5. Hacer clic en el botón "Send" 6. Esperar un tiempo de 5 minutos. 7. Hacer clic en el botón "Send" 8. Visualizará que el token ha expirado. 		
Datos de entrada			
Resultado obtenido			
Observaciones			

Pruebas funcionales

Path de acceso /psancionados		
Número caso	PDE- PS3-03	Evaluación Exitoso Fallido
Objetivo de la funcionalidad	Verificar el acceso al servicio web para el sistema 3 a través del path /psancionados	
Precondiciones	Generar y aplicar token	
Resultado esperado	Acceso a los datos	
Instrucciones	1. Ingresar a la URL de datos (/psancionados) 2. Hacer clic en el botón "Send" 3. Acceso a los datos	
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 1	
Resultado obtenido		
Observaciones		

El método Post		
Número caso	PDE- PS3-04	Evaluación Exitoso Fallido
Objetivo de la funcionalidad	Verificar el acceso a los datos a través del método Post	
Precondiciones	Generar y aplicar token	
Resultado esperado	Acceso a los datos	
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 1	
Resultado obtenido		
Observaciones		

El método Get		
Número caso	PDE- PS3-05	Evaluación Exitoso Fallido

Objetivo de la funcionalidad	Verificar que no acceda a los datos a través del método Get
Precondiciones	Generar y aplicar token
Resultado esperado	No acceder a los datos
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 1
Resultado obtenido	
Observaciones	

El método Put

Número caso	PDE- PS3-06	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que no acceda a los datos a través del método Put		
Precondiciones	Generar y aplicar token		
Resultado esperado	No acceder a los datos		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 1		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "fechaCaptura" contenga solo números, guion medio y que tenga el formato de AAAA-MM-DD

Número caso	PDE- PS3-07	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "fechaCaptura" contenga solo números, guion medio y que tenga el formato de AAAA-MM-DD		
Precondiciones	Generar y aplicar token		
Resultado esperado	No acceder a los datos		
Datos de entrada	"fechaCaptura" = 2020-01-fu "limite_registros" = 10		

	"pagina" = 1
Resultado obtenido	
Observaciones	

Validar que el dato de entrada "fechaCaptura" no contenga caracteres especiales a excepción del guion medio

Número caso	PDE- PS3-08	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "fechaCaptura" no contenga caracteres especiales a excepción del guion medio		
Precondiciones	Generar y aplicar token		
Resultado esperado	No acceder a los datos		
Datos de entrada	"fechaCaptura" = 2019-&&-01 "limite_registros" = 10 "pagina" = 1		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "limite_registros" contenga solo números

Número caso	PDE- PS3-09	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "limite_registros" contenga solo números		
Precondiciones	Generar y aplicar token		
Resultado esperado	Acceso a los datos		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 1 "pagina" = 10		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "limite_registros" no contenga letras

Número caso	PDE- PS3-10	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "limite_registros" no contenga letras		
Precondiciones	Generar y aplicar token		
Resultado esperado	No acceder a los datos		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = A "pagina" = 1		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "limite_registros" que cuando este vacío muestre 10 registros

Número caso	PDE- PS3-11	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "limite_registros" que cuando este vacío muestre 10 registros		
Precondiciones	Generar y aplicar token		
Resultado esperado	Muestre 10 registros		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = "pagina" = 1		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "limite_registros" que cuando coloque 0 muestre 10 registros

Número caso	PDE- PS3-12	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "limite_registros" que cuando coloque 0 muestre 10 registros		

Precondiciones	Generar y aplicar token
Resultado esperado	Muestre 10 registros
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 0 "pagina" = 1
Resultado obtenido	
Observaciones	

Validar que el dato de entrada "limite_registros" no contenga caracteres especiales

Número caso	PDE- PS3-13	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "limite_registros" no contenga caracteres especiales		
Precondiciones	Generar y aplicar token		
Resultado esperado	No acceder a los datos		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = && "pagina" = 1		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "pagina" contenga solo números

Número caso	PDE- PS3-14	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "pagina" contenga solo números		
Precondiciones	Generar y aplicar token		
Resultado esperado	Acceso a los datos		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 2		

Resultado obtenido	
Observaciones	

Validar que el dato de entrada "pagina" no contenga letras

Número caso	PDE- PS3-15	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "pagina" no contenga letras		
Precondiciones	Generar y aplicar token		
Resultado esperado	No acceder a los datos		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = A		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "pagina" no contenga caracteres especiales

Número caso	PDE- PS3-16	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "pagina" no contenga caracteres especiales		
Precondiciones	Generar y aplicar token		
Resultado esperado	No acceder a los datos		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = &&		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "pagina" que cuando coloque 0 muestre la página 1

Número caso	PDE- PS3-17	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "pagina" que cuando coloque 0 muestre la página 1		
Precondiciones	Generar y aplicar token		
Resultado esperado	Muestre la página 1		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 0		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "pagina" que cuando este vacío muestre la página 1

Número caso	PDE- PS3-18	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el dato de entrada "pagina" que cuando este vacío muestre la página 1		
Precondiciones	Generar y aplicar token		
Resultado esperado	Muestre la página 1		
Datos de entrada	"fechaCaptura" = 2021-05-01 "limite_registros" = 5 "pagina" =		
Resultado obtenido			
Observaciones			

Validar que el dato de entrada "pagina" que cuando coloque el último número de la página ya no muestre otra página con registros

Número caso	PDE- PS3-19	Evaluación	Exitoso Fallido
--------------------	-------------	-------------------	--------------------------

Objetivo de la funcionalidad	Verificar que el dato de entrada "pagina" que cuando coloque el último número de la página ya no muestre otra página con registros
Precondiciones	Generar y aplicar token
Resultado esperado	Si "totalRows">("pagina"*"limite_registros") True si mostrará otra página False no mostrará otra página
Instrucciones	<ol style="list-style-type: none"> 1. Verificar cuantos registros contiene en el atributo "totalRows" 2. Colocar en el dato de entrada "pagina" el total de registros que muestra en el atributo "totalRows" y dejar los mismos datos en "fechaCaptura" y "limite_registros" 3. Hacer clic en el botón "Send" 4. Colocar en el dato de entrada "pagina" el número que tiene sumarle uno y dejar los mismos datos en "fechaCaptura" y "limite_registros" 5. Hacer clic en el botón "Send" 6. Verificar que ya no muestre otra página con registros
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 1 "pagina" = 1
Resultado obtenido	
Observaciones	

Validar que "fechaCaptura", "limite_registros", "pagina", "hasNextPage", "totalRows", "pageSize", "page" sean correctos

Número caso	PDE- PS3-20	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que "fechaCaptura", "limite_registros", "pagina", "hasNextPage", "totalRows", "pageSize", "page" sean correctos		
Precondiciones	Generar y aplicar token		
Resultado esperado	Mostrará los registros que sean mayor a la "fechaCaptura" que se colocó Mostrará que el atributo "pageSize" sea igual que el dato de entrada "limite_registros" Mostrará el total de los registros Mostrará el atributo de "hasNextPage"= true o false dependiendo el caso Mostrará que el atributo "page" sea igual que el dato de entrada "pagina"		
Instrucciones	<ol style="list-style-type: none"> 1. Colocar el dato de entrada "fechaCaptura" la fecha 2000-01-01 1.1. Colocar el dato de entrada "limite_registros" el valor de 10 		

- 1.2. Colocar el dato de entrada "pagina" el valor de 1
2. Hacer clic en el botón "Send"
3. Verificar cuantos registros contiene en el atributo "totalRows"
- 3.1. Colocar en el dato de entrada "limite_registros" el total de registros que muestra en el atributo "totalRows" y dejar los mismos datos en "fechaCaptura" y "pagina"
- 3.2. Hacer clic en el botón "Send"
- 3.3. Para verificar el total de registros verificar que el atributo "pageSize" sea igual que el dato de entrada "limite_registros"
4. Calcula el "limite_registros" = "totalRows" / número de páginas (5) el resultado se tiene que redondear a un número entero mayor al resultado.
5. Colocar en el dato de entrada "limite_registros" el valor del resultado del paso anterior y dejar los mismos datos en "fechaCaptura" y "pagina"
6. Hacer clic en el botón "Send"
7. Verificar que el valor del atributo "page" sea 1 y que el atributo "hasNextPage" muestre "true" lo que significa que todavía hay más páginas
8. Colocar en el dato de entrada "pagina" un valor intermedio del número de páginas que se colocó (3) y dejar los mismos datos en "fechaCaptura" y "limite_registros"
9. Hacer clic en el botón "Send"
10. Verificar el valor del atributo "page" sea (3) dependiendo del valor que se colocó en el paso 12 y que el atributo "hasNextPage" muestre "true" lo que significa que todavía hay más páginas
11. Colocar en el dato de entrada "pagina" el último número de páginas que se colocó en el paso 8 que es (5) y dejar los mismos datos en "fechaCaptura" y "limite_registros"
12. Hacer clic en el botón "Send"
13. Verificar el valor del atributo "page" sea (5) dependiendo del valor que se colocó en el paso 15 y que el atributo "hasNextPage" muestre "false" lo que significa que ya no hay más páginas
- 13.1. Para verificar que ya no muestre más paginas agregue un número más del que tiene en el dato de entrada "pagina" (6)
- 13.2 Hacer clic en el botón "Send"
- 13.3. Verificar que ya no muestre registros lo que significa que ya no hay más páginas.
14. Colocar en el dato de entrada "fechaCaptura" la fecha 2020-01-01, en "limite_registros" = 3 y en "pagina" = 2
15. Hacer clic en el botón "Send"
16. Verificar que los registros muestren una fecha mayor o igual a la "fechaCaptura" que se ingresó a el dato de entrada "fechaCaptura", verificar también que el atributo "pageSize" corresponda con el "limite_registros" que se colocó en el dato de entrada, verificar

	también que el atributo "page" corresponda con la "pagina" que se colocó en el dato de entrada
Datos de entrada	"fechaCaptura" "limite_registros" "pagina"
Resultado obtenido	
Observaciones	

Mostrar que el atributo "page" sea el mismo que el dato de entrada "pagina"

Número caso	PDE- PS3-21	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el atributo "page" sea igual que el dato de entrada "pagina"		
Precondiciones	Generar y aplicar token		
Resultado esperado	Mostrar que el atributo "page" sea igual que el dato de entrada "pagina"		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 4		
Resultado obtenido			
Observaciones			

Validar que el servicio responda correctamente en formato Json

Número caso	PDE- PS3-22	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el servicio responda correctamente en formato Json		
Precondiciones	Generar y aplicar token		
Resultado esperado	Acceso a los datos en formato Json		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 1		
Resultado obtenido			
Observaciones			

Validar que el tamaño de la página sea correcto

Número caso	PDE- PS3-23	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el atributo "pageSize" muestre el valor correcto		
Precondiciones	Generar y aplicar token		
Resultado esperado	Mostrar que el valor de "pageSize" sea el mismo que valor de entrada "limite_registros"		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 1		
Resultado obtenido			
Observaciones			

Validar que muestre la página solicitada

Número caso	PDE- PS3-24	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que muestre la página solicitada		
Precondiciones	Generar y aplicar token		
Resultado esperado	Mostrar que el atributo "page" sea igual al dato de entrada "pagina"		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 2		
Resultado obtenido			
Observaciones			

Validar que el total de registros sean correctos

Número caso	PDE- PS3-25	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que el atributo "totalRows" sea igual que el atributo "pageSize" y también igual que el dato de entrada "limite_registros"		
Precondiciones	Generar y aplicar token		
Resultado esperado	El atributo "totalRows" tiene que ser igual al atributo "pageSize"		

Instrucciones	<ol style="list-style-type: none"> 1. Hacer clic en el botón "Send" 2. Verificar cuantos registros contiene en el atributo "totalRows" <ol style="list-style-type: none"> 2.1. Colocar en el dato de entrada "limite_registros" el total de registros que muestra en el atributo "totalRows" 2.2. Hacer clic en el botón "Send" 2.3. Para verificar el total de registros verificar que el atributo "pageSize" sea igual que el dato de entrada "limite_registros"
Datos de entrada	<pre>"fechaCaptura" = 2000-01-01 "limite_registros" = "pagina"= 1</pre>
Resultado obtenido	
Observaciones	

Validar que la página actual cambie a la siguiente página

Número caso	PDE- PS3-26	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que la página actual cambie a la siguiente página		
Precondiciones	Generar y aplicar token		
Resultado esperado	El atributo "hasNextPage" = "totalRows" > ("pageSize"*"page") false		
Instrucciones	<ol style="list-style-type: none"> 1. Hacer clic en el botón "Send" 2. Verificar cuantos registros contiene el atributo "totalRows" (5) <ol style="list-style-type: none"> 2.1. Existen dos casos <ol style="list-style-type: none"> a) Donde muestre el resultado de "true" <ol style="list-style-type: none"> 2.1a. El atributo "page" es igual al dato de entrada "pagina" donde colocará (1) 2.2a. El atributo "pageSize" es igual al dato de entrada "limite_registros" donde colocará el valor de 1 y en "fechaCaptura" se queda igual 2.3a. Hacer clic en el botón "Send" 2.4a. Validar que el atributo "hasNextPage" sea igual a "true" 2.5a. Verificar si "totalRows" > ("pageSize"*"page") entonces "hasNextPage"= true lo que significa que hay más páginas b) Donde muestre el resultado de false <ol style="list-style-type: none"> 2.6b. Colocar en el dato de entrada "limite_registros" el valor que tiene "totalRows" 2.7b. Colocar en el dato de entrada "pagina" el valor de 1 y en "fechaCaptura" se queda igual 2.8b. Hacer clic en el botón "Send" 2.9b. Validar que el atributo "hasNextPage" sea igual a false 		

	2.10b. Verificar si "totalRows" >("pageSize"*"page") entonces "hasNextPage"= false lo que significa que ya no hay más páginas
Datos de entrada	"fechaCaptura": 2000-01-01 "limite_registros": 10 "pagina": 1
Resultado obtenido	
Observaciones	

Validar que la operación no sea exitosa y muestre el esquema "ResError"

Número caso	PDE- PS3-27	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que la operación sea fallida y muestre el esquema "ResError"		
Precondiciones	Generar y aplicar token		
Resultado esperado	Muestre la estructura { "code": "xxx", "message": "xxx" }		
Instrucciones	1. Hacer clic en el botón "Send" 2. Visualizar un código y un mensaje de error { "code": "xxx", "message": "xxx" }		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = A "pagina" = &		
Resultado obtenido			
Observaciones			

Validar que ingrese a la página a través de https

Número caso	PDE- PS3-28	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Validar que ingrese a la "URL" (https)		
Precondiciones	Generar y aplicar token		
Resultado esperado	Acceda a los datos		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 1		

Resultado obtenido	
Observaciones	

Validar que no ingrese a la página a través de http

Número caso	PDE- PS3-29	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Validar que no ingrese a la "URL" (http)		
Precondiciones	Generar y aplicar token		
Resultado esperado	No acceder a los datos		
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 1		
Resultado obtenido			
Observaciones			

Validar el certificado SSL

Número caso	PDE- PS3-30	Evaluación	Exitoso Fallido
Objetivo de la funcionalidad	Verificar que este actualizado el certificado SSL		
Precondiciones	Generar y aplicar token		
Resultado esperado	Muestre actualizado el certificado SSL		
Datos de entrada	"fechaCaptura" = 2021-05-01 "limite_registros" = 2 "pagina" = 2		
Resultado obtenido			
Observaciones			

Validar el API con el esquema de Psancionados

Número caso	PDE- PS3-31	Evaluación	Exitoso Fallido
--------------------	-------------	-------------------	--------------------------

Objetivo de la funcionalidad	Validar que el API responda cumpliendo con el esquema de Particulares Sancionados
Precondiciones	Generar y aplicar token
Resultado esperado	Valide el API de Psacionados
Instrucciones	1. Hacer clic en el botón "Send" 2.- Se valida contra su esquema correspondiente en https://gitlab.com/plataformadigitalestataledomex/api_docs_pde/-/blob/master/S3/standar_psancionados.json
Datos de entrada	"fechaCaptura" = 2000-01-01 "limite_registros" = 10 "pagina" = 1
Resultado obtenido	
Observaciones	