



USAC

TRICENTENARIA

Universidad de San Carlos de Guatemala

INGENIERIA

CUNOC

Laboratorio de Estructura de Datos

Proyecto 2: Manual de Usuario

Sección: A

Nombre:

Registro académico:

Mariano Francisco Camposeco Camposeco

202030987

Quetzaltenango, 09 de mayo de 2022.

INTRODUCCIÓN

El uso de las API ha aumentado por las necesidades de las poblaciones, por lo tanto es de gran importancia su uso, por lo tanto esta API, permite aprender de una manera divertida, mediante el juego solitario pirámide, donde podemos agregar cartas, eliminar, obtener el orden graficar, esto accediendo desde métodos post, get, delete.

OBJETIVOS

Objetivo general:

- Experimentar de buena manera el uso de una API de manera divertida.

Objetivos específicos:

1. Aprender a usar los distintos métodos de solicitud.
2. Generar arboles AVL.
3. Experimentar el uso de la API.

Requerimientos

→Windows

En casos de tener sistema operativo Windows necesita lo siguiente:

Windows 10 (8u51 y superiores)

Windows 8.x (escritorio)

Windows 7 SP1

Windows Vista SP2

Windows Server 2008 R2 SP1 (64 bits)

Windows Server 2012 y 2012 R2 (64 bits)

RAM: 128 MB

Espacio en disco: 124 MB para JRE; 2 MB para Java Update

Procesador: Mínimo Pentium 2 a 266 MHz

Exploradores: Internet Explorer 9 y superior, Firefox

→Mac OS X

En casos de tener sistema operativo Mac OS X necesita lo siguiente:

Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+

Privilegios de administrador para la instalación

Explorador de 64 bits

Se requiere un explorador de 64 bits (Safari, por ejemplo) para ejecutar Oracle Java en Mac.

→Linux

En casos de tener sistema operativo Linux necesita lo siguiente:

Oracle Linux 5.5+1

Oracle Linux 6.x (32 bits), 6.x (64 bits)²

Oracle Linux 7.x (64 bits)² (8u20 y superiores)

Red Hat Enterprise Linux 5.5+1 6.x (32 bits), 6.x (64 bits)²

Red Hat Enterprise Linux 7.x (64 bits)² (8u20 y superiores)

Suse Linux Enterprise Server 10 SP2+, 11.x

Suse Linux Enterprise Server 12.x (64 bits)² (8u31 y superiores)

Ubuntu Linux 12.04 LTS, 13.x

Ubuntu Linux 14.x (8u25 y superiores)

Ubuntu Linux 15.04 (8u45 y superiores)

Ubuntu Linux 15.10 (8u65 y superiores)

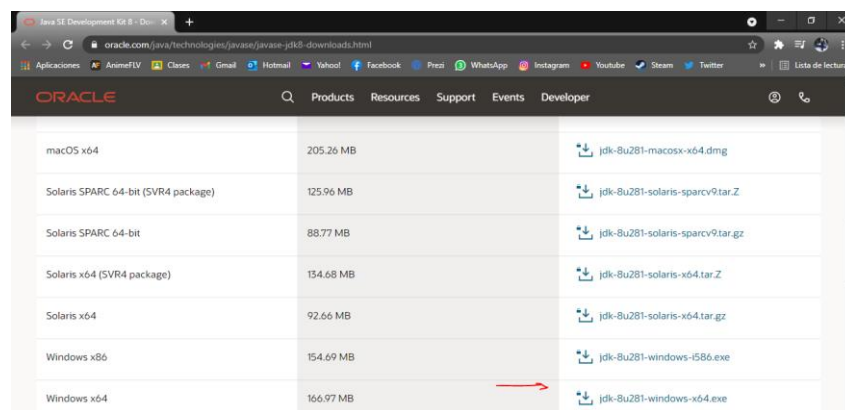
Exploradores: Firefox

Guía de uso

Antes de ejecutar el sistema

Primero necesitas tener instalado un jdk de java para poder ejecutar el juego, para ello accede al siguiente enlace, seleccionas tu sistema operativo y lo descargas :

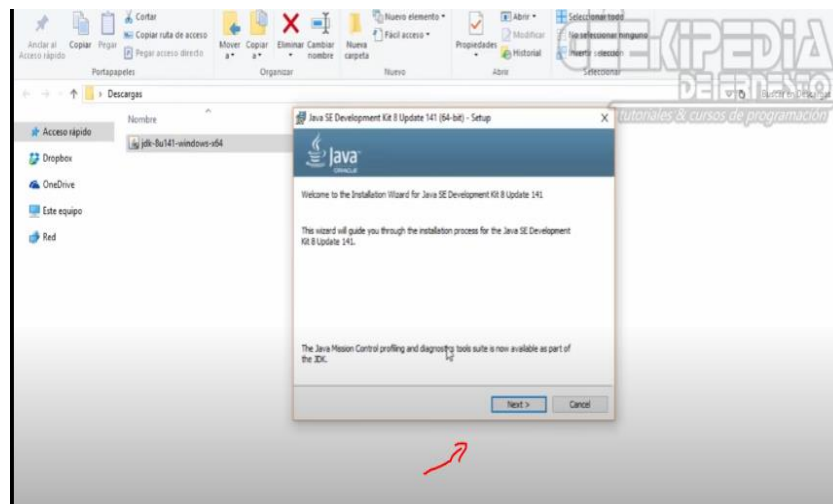
<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>



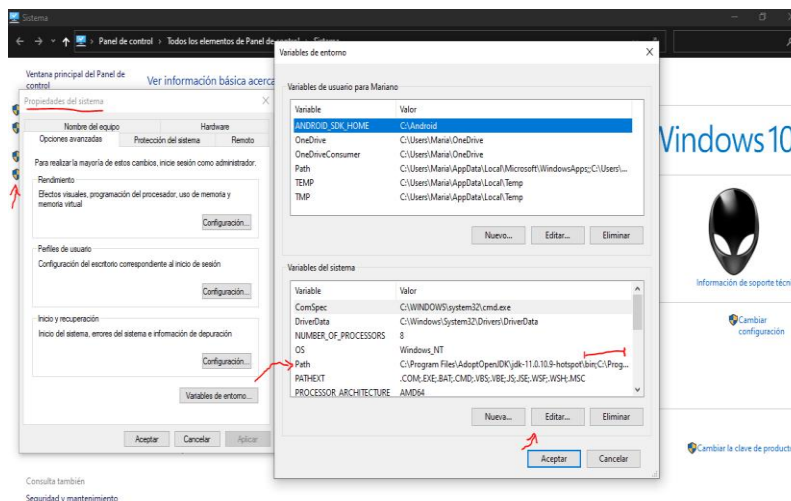
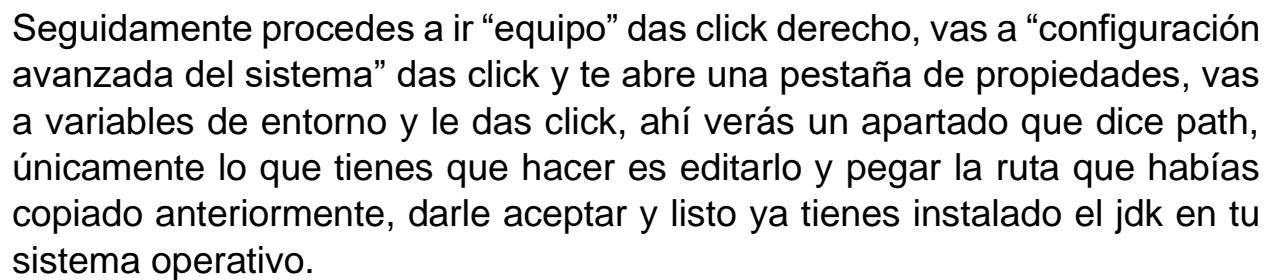
macOS x64	205.26 MB	jdk-8u281-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	125.96 MB	jdk-8u281-solaris-sparcv9tar.Z
Solaris SPARC 64-bit	88.77 MB	jdk-8u281-solaris-sparcv9tar.gz
Solaris x64 (SVR4 package)	134.68 MB	jdk-8u281-solaris-x64tar.Z
Solaris x64	92.66 MB	jdk-8u281-solaris-x64tar.gz
Windows x86	154.69 MB	jdk-8u281-windows-i586.exe
Windows x64	166.97 MB	jdk-8u281-windows-x64.exe

Al instalar el jdk

Al ejecutar el archivo descargado lo abres y prosigues a dar la técnica siguiente, siguiente no te preocupes, es seguro.

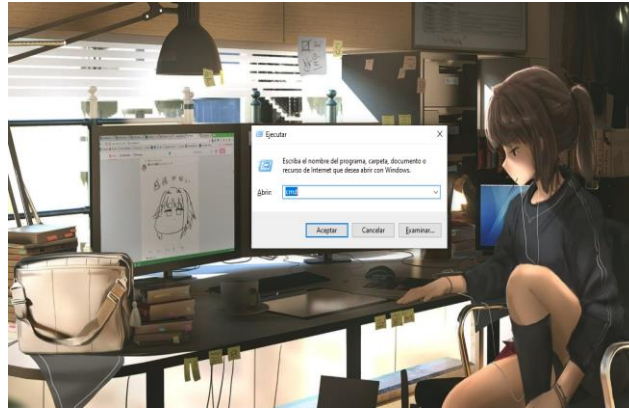


Para configurarlo entras a la carpeta bin y copias la ruta que se señala a continuación, la copias.

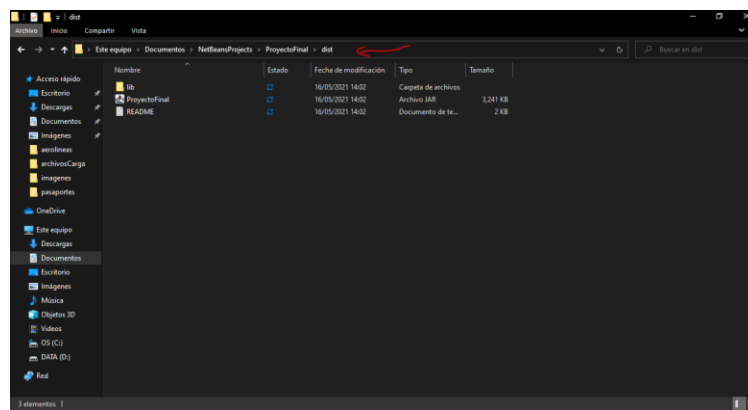


Para ejecutar el programa:

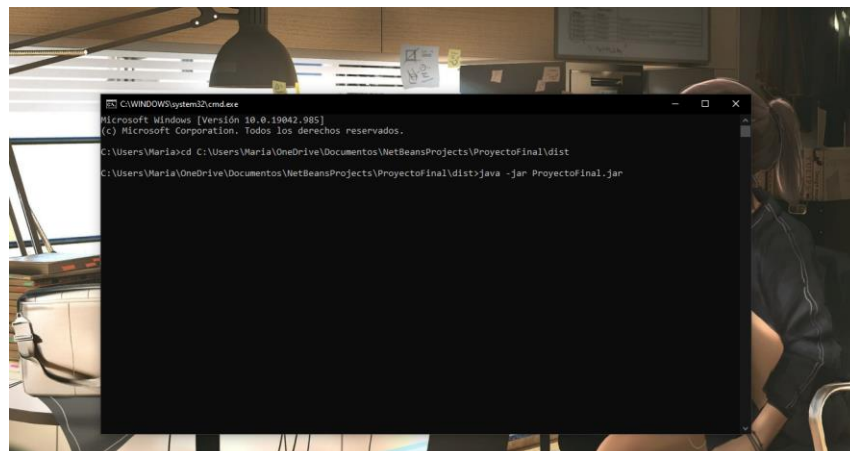
Ahora para ejecutar primero tienes que abrir CMD, presionando Tecla Windows+r y con la pestaña abierta escribes “cmd” y le das aceptar.



Seguidamente copias la dirección donde se encuentra el jdk.



Escribes `cd "ruta de enlace copiada"` y le das enter, entonces únicamente tendrías que poner `java -jar SegundoProyecto.jar` das enter y te el programa



Programas adicionales:

Instalar y configurar gnrok:

https://www.youtube.com/watch?v=YA_xMQOIZo0&t=267s

<https://ngrok.com/download>

Instalar y configurar graphviz:

https://www.youtube.com/watch?v=JYAHwQ_tMG0&t=116s

<https://graphviz.org/download/>

Empezamos

Usando una solicitud POST a la ruta "Game/start": Para iniciar con las cartas que se metan, poniendo un numero y la carta tomando en cuenta el valor:

Carta	Valor	Carta	Valor
As	1	7	7
2	2	8	8
3	3	9	9
4	4	10	10
5	5	J	11
6	6	Q	12
		K	13

Overview | POST http://localhost:8080/Game/start | GET http://localhost:8080/Game/start | GET http://localhost:8080/Game/start | GET http://localhost:8080/Game/start | GET http://localhost:8080/Game/start | POST http://localhost:8080/Game/start | GET http://localhost:8080/Game/start | GET http://localhost:8080/Game/start | DEL http://localhost:8080/Game/start | + | No Environment

Save | Send

POST http://localhost:8080/Game/start

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL | JSON

```
1 [
2   "0": "9♥",
3   "1": "10♠",
4   "2": "A♥",
5   "3": "4♠",
6   "4": "K♠",
7   "5": "9♥",
8   "6": "8♥"
9 ]
```

Body | Cookies | Headers (5) | Test Results

Status: 200 OK Time: 78 ms Size: 163 B Save Response

Pretty | Raw | Preview | Visualize | Text

1

Find and Replace | Console | Cookies | Capture requests | Bootcamp | Runner | Trash

Usando una solicitud POST a la ruta "Game/add": Para agregar una carta a nuestro árbol de manera eficaz, si ya se encuentra tirará error:

Overview | POST http://localhost:8080/Game/add | GET http://localhost:8080/Game/add | GET http://localhost:8080/Game/add | GET http://localhost:8080/Game/add | GET http://localhost:8080/Game/add | POST http://localhost:8080/Game/add | GET http://localhost:8080/Game/add | GET http://localhost:8080/Game/add | DEL http://localhost:8080/Game/add | + | No Environment

Save | Send

POST http://localhost:8080/Game/add

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL | JSON

```
1 [
2   "insert": "4♥"
3 ]
```

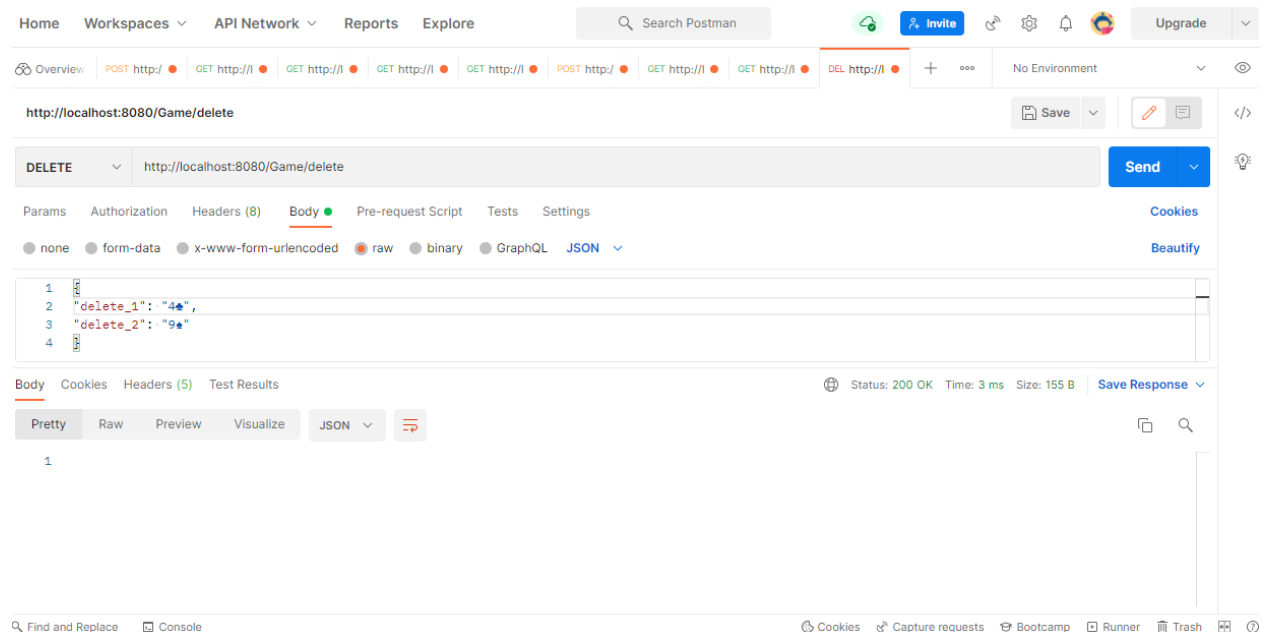
Body | Cookies | Headers (5) | Test Results

Status: 200 OK Time: 4 ms Size: 163 B Save Response

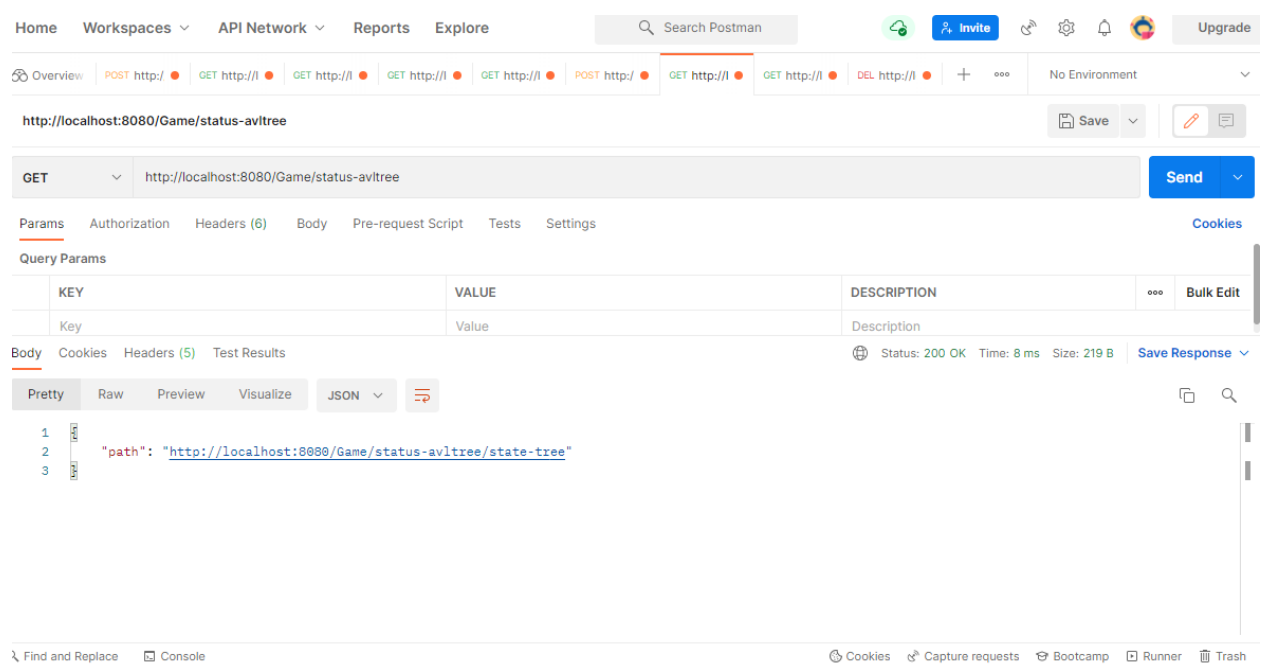
Pretty | Raw | Preview | Visualize | Text

1

Usando una solicitud DELETE a la ruta "Game/delete": Para eliminar carta, puede enviarse tanto 1 como dos solicitudes, para esto tiene que sumar 13, o ser 13 en sí, si tira error puede, que tenga hijos o no sea la suma o no se encuentre, (verificar código de errores al final del documento).



Usando una solicitud GET a la ruta "Game/status-avltree": Genera y permite observar la ruta donde se encuentra nuestro árbol avl y al ingresar a ese enlace podemos ver el árbol.



Home Workspaces API Network Reports Explore Search Postman

Overview POST http:// GET http:// GET http:// GET http:// GET http:// POST http:// GET http:// GET http:// DEL http:// + No Environment

http://localhost:8080/Game/status-avltree/state-tree Save

GET http://localhost:8080/Game/status-avltree/state-tree Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 6 ms Size: 16.14 KB Save Response

Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

Usando una solicitud GET a la ruta "Game/get-level?level={no.level}":
Permite observar lo que contiene un nodo en determinado nivel, tomar
encuentra que empieza del nivel 1 en adelante.

Overview POST http:// GET http:// GET http:// GET http:// GET http:// POST http:// GET http:// GET http:// DEL http:// + No Environment

http://localhost:8080/Game/get-level?level=2 Save

GET http://localhost:8080/Game/get-level?level=2 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> level	2			
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 11 ms Size: 180 B Save Response

Pretty Raw Preview Visualize JSON

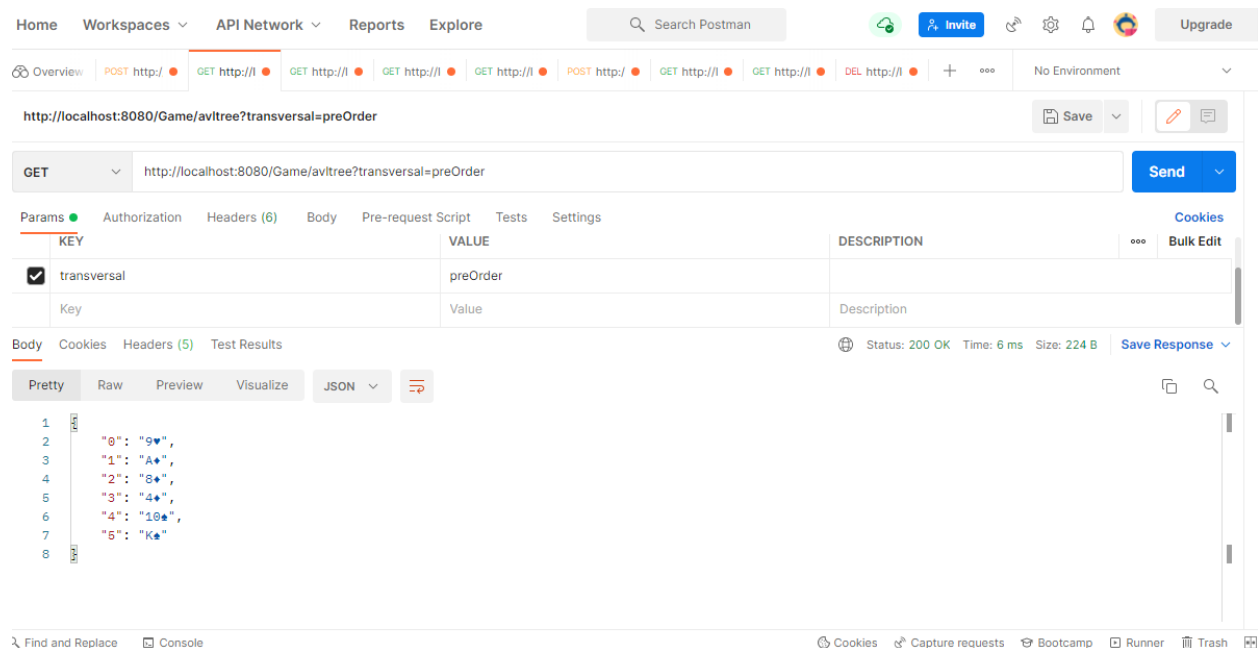
```

1  {
2    "0": "A♦",
3    "1": "10♠"
4  }

```

Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

Usando una solicitud GET a la ruta "Game/avltree?transversal={preOrder/inOrder/postOrder}":Permite observar los recorridos de nuestro árbol, tanto en preOrden, inOrden y postOrden.



Interpretación de códigos de error:

Códigos de error

Error	Código de Respuesta
La carta no se encuentra en el árbol avl (eliminar)	Status Code 404
Los valores de las cartas no suman 13	Status Code 406
La carta no se puede eliminar ya que cuenta con hijos	Status Code 409
La carta a insertar esta duplicada	Status Code 406
Cualquier otro error	Status Code 400