Web services basados en JavaScript y Google Apps Script

Objetivo

Lograr que el estudiante no solo comprenda el concepto de web service sino que lo haga en el contexto de IoT, comprobando el valor que tienen en las soluciones IoT o IIoT.

Resultados de aprendizaje

- El estudiante tendrá los conocimientos para crear un web service propio. Por ahora lo podrá hacer solo con herramientas de Google Apps Script/JavaScript, pero será consciente que en el fondo no importa el lenguaje de programación u plataforma usada.
- El estudiante sabrá que la creación de un web service requiere una condiciones especiales, no es algo que se pueda hacer en un computador local, es una solución web.
- El estudiante conocerá la utilidad que los web services pueden brindarle a las soluciones IoT.

Conocimientos previos

Comunicación:

La comunicación entre cliente y servidor se basa en protocolos estándar, como es el caso del protocolo HTTP para aplicaciones web, o protocolos personalizados en otros casos.
El cliente envía una solicitud al servidor, que incluye información sobre la acción que se debe realizar y, a menudo, datos adicionales.
El servidor procesa la solicitud y envía una respuesta al cliente, con los resultados de la acción solicitada o los datos solicitados.
La comunicación cliente-servidor se basa en un modelo de "solicitud y respuesta", donde el cliente inicia la interacción enviando una solicitud y el servidor responde con una respuesta correspondiente.

Web Service

Es una solución a nivel de servidor. Es un concepto similar al de una API, en el sentido de tener entradas, salidas, realizar tareas en la nube, tener una URL con comandos (endpoint). Pero entre las diferencias pueden estar las siguientes:

- Un Web Services puede ser hecho para algo tan sencillo como sumar “a” y “b” o para realizar una tarea más compleja,como la de procesar datos. De manera que no requiere todo ese soporte técnico que tiene una API, por eso le llaman también microservicios. Puede ser visto como uno de los componentes que un desarrollador crea o usa cuando programa soluciones distribuidas en la nube.
- Un Web Service a menudo sigue estándares web específicos, como SOAP (Simple Object Access Protocol) o REST (Representational State Transfer), que definen cómo se deben realizar las solicitudes y respuestas. Las API pueden usar diversos protocolos y no siempre siguen estándares específicos, lo que les da más flexibilidad en su diseño.

Protocolo HTTP

Es parte de la comunicación. En los web services se usa protocolo HTTP

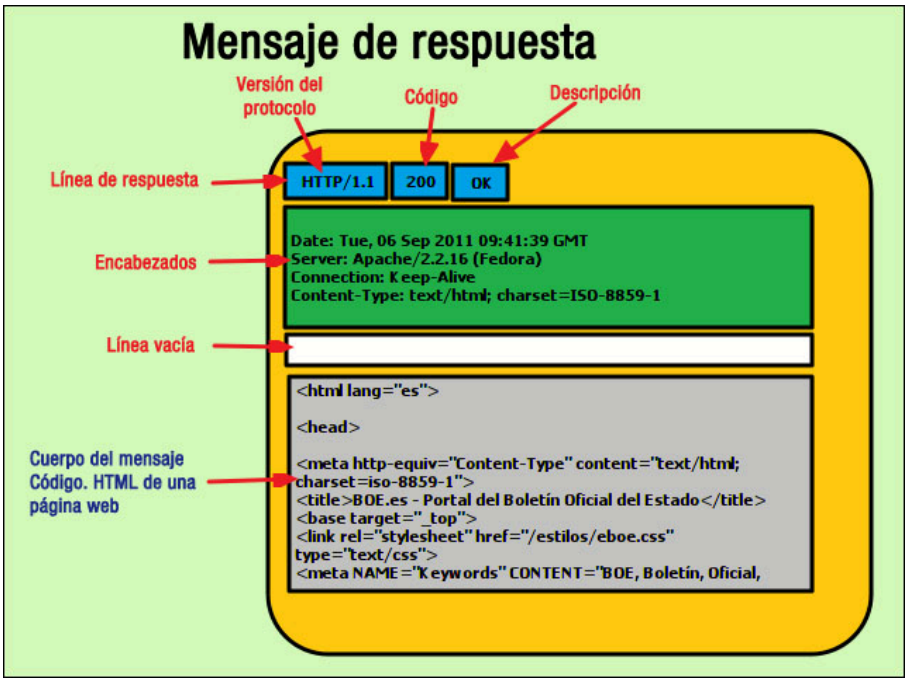


Fig. 1

Un mensaje HTTP puede ser visto como una especie de tabla, como se muestra en la Fig.1. La primera fila es la línea de respuesta y allí; la primera celda es la versión del protocolo, por ejemplo HTTP/1.1; la segunda es el “Código” de respuesta es un valor numérico que representa como ha sido recibida y procesada la petición a la que se está respondiendo; la tercera celda - la “Descripción” es una frase corta que describe lo que se está indicando con el código de respuesta (a cada código corresponde una descripción).
En la segunda fila está el “Encabezado” de una respuesta que indica opciones relativas a la respuesta. La tercera línea está vacía y la cuarta es el cuerpo del mensaje.

REST

Roy Fielding en su tesis doctoral titulada "Architectural Styles and the Design of Network-based Software Architectures", se hizo quizá la siguiente pregunta: ¿será posible lograr con los recursos en la nube algo parecido a lo que se logra en la programación

orientada a objetos, es decir que en la nube existe un recurso que al instanciarlo pueda ser visto como algo que sigue en la nube pero que puede ser parte de cualquier aplicación ? ¿qué hay que hacer para lograrlo?. Haciendo la analogía con una página web, para que esta tenga una URL debe ser ubicada en un servidor hecho justo para páginas web, lo mismo pasa aquí, se requiere un servidor que facilite la creación de esos objetos que podemos llamar web services. También se requiere que todos los web services sigan reglas comunes para compartir información. Pues bien, el conjunto de esas reglas y recursos necesarios es lo que se llamó REST (Representational State Transfer) y representan una arquitectura de diseño de web services, es decir, un estilo arquitectónico para diseñar sistemas distribuidos, no incluye formatos y por eso no se considera como un protocolo en el sentido tradicional de los protocolos de comunicación TCP-IP. En realidad REST se apoya en el protocolo HTTP, lo complementa con reglas para facilitar las soluciones distribuidas en la nube. La gran diferencia con la programación orientada a objetos es que los objetos están en tu computador, los web services están en la nube, disponibles para que miles de aplicaciones los instancien. Gracias a REST las soluciones resultan distribuidas en la nube de manera compatible. En conclusión gracias a REST existen reglas claras de compatibilidad para crear y consumir web services sin importar el lenguaje de programación de la aplicación que consume ni del lenguaje de programación en el que el web service ha sido hecho.

REST es una arquitectura de diseño que se utiliza para crear servicios web que son eficientes, escalables y fáciles de entender. La arquitectura se basa en la idea de que los recursos (como datos o funcionalidades) se representan a través de URLs, y las operaciones se realizan utilizando los métodos HTTP estándar (GET, POST, PUT, DELETE, etc.). Justamente los Web Services se crean bajo la arquitectura REST.

API RESTFul

Si una API sigue de manera estricta los principios de la arquitectura REST se considera que es una API RESTFul

SOAP

SOAP (Simple Object Access Protocol) es un protocolo de comunicación utilizado para intercambiar mensajes entre sistemas distribuidos en una red, como la web. A diferencia de REST, que se basa en la representación de recursos a través de URLs y el uso de métodos HTTP estándar, SOAP utiliza un enfoque más estructurado y formal para la comunicación entre sistemas. Al igual que REST, SOAP también surgió como una solución para la interoperabilidad entre sistemas distribuidos. En lugar de utilizar métodos HTTP estándar, SOAP define su propio formato de mensaje XML para representar la información que se intercambia entre los sistemas. Esto proporciona una mayor flexibilidad y permite representar datos más complejos y estructurados que con REST.
SOAP se basa en el concepto de operaciones remotas, donde un sistema puede invocar métodos o funciones en otro sistema a través de la red. Para facilitar esta comunicación, SOAP define un conjunto de reglas y estándares para la estructura y el formato de los

mensajes, incluyendo la especificación de encabezados, cuerpo del mensaje y detalles de la operación solicitada.

Una de las características clave de SOAP es su capacidad para garantizar la seguridad y la confiabilidad de la comunicación entre sistemas, a través del uso de mecanismos como la autenticación, la autorización y la integridad de los mensajes.

Desambiguación entre REST y SOAP

"Imaginemos que las unidades militares tienen sistemas de comunicación altamente estructurados y formalizados, donde cada mensaje sigue un protocolo rígido y específico, similar a SOAP. Estos protocolos definen palabras clave y formatos de mensaje precisos que son conocidos y utilizados exclusivamente por las unidades militares, lo que garantiza una comunicación segura y confiable, pero a menudo a expensas de la flexibilidad y la eficiencia. REST es más bien comparable con la normatividad que las empresas de mensajería deben cumplir para que los paquetes enviados desde un lugar de la tierra lleguen a otro, usando los medios de transporte disponibles. Estas normativas establecen reglas comunes y estándares compartidos, pero permiten cierta flexibilidad en la forma en que se entregan los paquetes. De manera similar, REST establece principios y reglas para el diseño de servicios web, pero permite una mayor flexibilidad y adaptabilidad en comparación con SOAP.

Consideracions sobre las solicitudes GET

- Las solicitudes GET se usan cuando el volumen de información que fluye desde el servidor al cliente es notablemente mayor. Por ejemplo, cuando una persona consulta una página web, envía poca información para ubicar esa página, pero de retorno recibe todo la necesario para desplegar esa página en pantalla
- Lo que hacen los navegadores cuando se le entrega una URL son siempre solicitudes GET.
- No se deben usar solicitudes GET para enviar archivos a un servidor, porque no soporta mucho volumen de información de subida, solo de bajada.
- Cuando se usa GET, el cliente solo pasa unas variables en forma de una URL junto con una query string. Los datos van en la query string:
  ¿Qué pasa después? que el navegador o la función que invoca el web service, arma un JSON para llevar esos datos desde el cliente hasta el servidor. Ese JSON tiene una estructura específica, con varias claves/valor. Esta es la estructura:

  Ejemplo de un web service que tiene la URL:
  https://script.google.com/a/macros/e3t.uis.edu.co/s/AKfycbx6trDXZpyXlW8D5kMsUC7o75n9YieSyH_VD95h0TiRXAkyummZ6gVJ4wqMlLIOyr8S/exec

  Pero que necesita enviar datos con la query string ?a=2&b=1:

https://script.google.com/a/macros/e3t.uis.edu.co/s/AKfycbx6trDXZpyXlW8D5kMsUC7o75n9YieSyH_VD95h0TiRXAkyummZ6gVJ4wqMlLIOyr8S/exec?a=2&b=1

Estructura que adquiere el JSON que viaja desde el cliente hasta el servidor cuando se envía la URL con la query string:

```
{ contextPath: '',
  queryString: 'a=2&b=1',
  parameters: { a: [ '2' ], b: [ '1' ] },
  parameter: { b: '1', a: '2' },
  contentLength: -1
}
```

Codigo del web service para el ejemplo anterior:

```
function doGet(e) {
  var x = parseFloat(e.parameter.a);
  var y = parseFloat(e.parameter.b);
  var z = x + y;
  var respuesta = {
    sum: z
  };

  return
ContentService.createTextOutput(JSON.stringify(respuesta)).setMimeType(ContentService
.MimeType.JSON);
}
```

Estructura del JSON que viaja desde el servidor hasta el cliente:

```
{
  "sum": 3
}
```

- Por lo tanto, al recibir una solicitud tipo GET, el servidor debe buscar la carga útil en la clave “parameter”. Para el ejemplo anterior, la carga útil es: { b: '1', a: '2' }
- La estructura del JSON que viaja desde el servidor hasta el cliente es la que esté definida en código del web service.

Consideracions sobre las solicitudes POST

- En una solicitud tipo POST, igual que en una tipo GET, se envía información a un servidor y recibe igualmente información, pero hay una gran diferencia - que la información que se envía tiene un volumen notablemente mayor que la que se recibe. Por ejemplo, cuando se quiere enviar un archivo a un servidor.
- Cuando se crea un web service basado en POST puede ser usado por el código html, pero este web service no se puede invocar desde un navegador usando la url con query string como se haría con un webservice basado en GET.
- En otras palabras, las query string no aplican en las llamadas POST y por lo tanto no aplica usar una URL con query string en un navegador. La única manera es usar aplicaciones hechas en python, php, html o cualquier otro lenguaje de programación

- Otra diferencia importante respecto a GET, es que en las llamadas POST, la estructura del JSON que viaja desde el cliente es diferente, como se muestra en este ejemplo

Este ejemplo está pensado solo en recibir información de un cliente y luego reenviarsela de regreso, como una confirmación de lo que hemos recibido. El ejemplo está pensado más que todo para comprender lo que ocurre en el viaje de la información

```
function doPost(cargaBruta) {

  // cargaBruta llega en forma de objeto de datos aunque el viaje haya sido un JSON
  // Para nuestra prueba necesitamos que cargaBruta se convierta a JSON
  var cargaBrutaJSON = JSON.stringify(cargaBruta);

  var mensajeDeControl = "He recibido la Carga bruta: "+ cargaBrutaJSON;

  console.log(mensajeDeControl);

// Enviamos al cliente remoto una respuesta informando qué nos ha llegado la carga
  return ContentService.createTextOutput("Gracias. Hemos atendido su POST. Carga bruta recibida: "+cargaBrutaJSON);
}
```

Ejemplo del cliente usando Python

```
import requests
url= 'la url de tu web service aquí'

# Ojo: En python lo que se prepara es un diccionario. Ya muy internamente, el sistema de envío lo
# convertirá en texto, es decir en JSON pero eso no lo veremos por comando.
cargaUtil={'nombre':'pepito','edad':'25'}
respuesta = requests.post(url, data=cargaUtil)
print(respuesta.text)
```

Del código anterior, vemos que en el cliente se preparó el siguiente diccionario como carga util a ser enviada:

```
{
  'nombre':'pepito',
  'edad':'25'
}
```

Pero de la confirmación recibida, vemos que lo que viajó desde el cliente hasta el servidor fue el siguiente JSON:

```
{
   "queryString":"",
   "contextPath":"",
   "parameter":{},
   "contentLength":34,
   "parameters":{},
   "postData":{
      "contents":"{
         'nombre': 'pepito',
```

```
        'edad': '25'
      }",
      "length":34,
      "name":"postData",
      "type":"application/json"}}
```

En el servidor, la función doPost(cargaBruta) recibe lo mismo pero en forma de un objeto de datos

```
cargaBruta=
{
   queryString: "",
   contextPath: "",
   parameter: {},
   contentLength: 34,
   parameters: {},
   postData: {
      contents: '{
         nombre: "pepito",
         edad: "25"
      }',
      length: 34,
      name: "postData",
      type: "application/json"
   }
}
```

- Está claro que la información desde el cliente hasta el servidor es un JSON, pero ojo, que a doPost no le llega ese JSON tal y como viaja, hay algo que forma parte de la plataforma GAS que en realidad le entrega al doPost la misma información pero en forma de un objeto de datos. Es decir, en el ejemplo anterior "cargaBruta" salió desde el cliente como un JSON, pero llegó al servidor como un objeto de datos.
- En conclusión, en el servidor, la información útil debe ser buscada asi, si se requiere como un JSON:
  cargaUtil_Json=cargaBruta.postData.contents
  OJO: en realidad lo que llega es un texto, solo que es muy usual que ese texto represente a un JSON, pero no es obligatoriamente un JSON.

  o así, si se requiere como un objeto de datos, a partir de un JSON:
  cargaUtil=JSON.parse(cargaBruta.postData.contents);

Ejemplo en el cual se extrae la carga útil en el servidor

```
function doPost(cargaBruta) {
  var cargaUtil=JSON.parse(cargaBruta.postData.contents);
  Logger.log(cargaUtil);
// Enviamos al cliente remoto una respuesta informando qué nos ha llegado la carga
  return ContentService.createTextOutput("Gracias. Hemos atendido su POST.");
}
```

Función de pruebas que simula una llamada remota

```
// la siguiente función es para simular datos que pudiesen ser enviados desde
invocaciones
// remotas cuando el web service se ponga en funcionamiento
function simuladorJSONViajante() {
  var datosJson =
'{"contentLength":34,"queryString":"","postData":{"contents":"{\\"nombre\\":
\\"pepito\\", \\"edad\\":
\\"25\\"}","length":34,"name":"postData","type":"application/json"},"contextPath":""
,"parameters":{},"parameter":{}}';
  return datosJson;
}


// La siguiente función simula lo que ocurre cuando alguna aplicación remota invoca
a
// nuestro web service
function prueba() {
  var jsonViajante = simuladorJSONViajante();
  var datosO = JSON.parse(jsonViajante);
  doPost(datosO);
}
```

Reto general para esta práctica

Buscaremos implementar el web service más básico posible para conquistar ese concepto. Asociando esto a IoT y con la intencionalidad de crear un negocio para surtir neveras que solicitan surtido. El avance con esta práctica consiste en lograr demostrar que un cliente creado con Python en un computador o creado en cualquier otro lenguaje y en cualquier otro medio, puede enviarle datos a una especie de agente en la nube bien sea para procesar los datos en la nube, para visualizarlos, para almacenarlos o para cualquier otra cosa que requiera la solución IoT más adelante. En el fondo se va a implementar una aplicación cliente servidor si consideramos que el código en Python corresponde al cliente y el web service al servidor.

Tarea 1. Conocer el protocolo http

Conocemos que el protocolo HTTP tiene las partes que aparecen en la Fig 1. Todos los días usas ese protocolo cuando llamas páginas web en un navegador. Ahora harás lo mismo con código python con el fin de acceder a mayores detalles del protocolo.

Use el siguiente código en Python para explorar cada una de las partes del protocolo. Caso en que se usa usa el método GET. Es lo normal para invocar páginas web.

```
import requests
url = "aqui la url de una página web"
#  Hacemos una solicitud tipo get a la URL, como lo haría un navegador
respuesta = requests.get(url)

# Obtenemos valores de los diferentes campos de la respuesta recibida del servidor
Codigo=respuesta.status_code
Descripcion=respuesta.ok
EncabezadoRespuesta=respuesta.headers
CuerpoRespuesta=respuesta.text
Metodo=respuesta.request.method
EncabezadoSolicitud= response.request.headers
CuerpoSolicitud=respuesta.request.body

print("El Código de la respuesta es: ", Codigo)
print("La descripción de la respuesta tiene ok? ", Descripcion)
print("Los encabezados de la respuesta: ", EncabezadoRespuesta)
print("Cuerpo de la respuesta es: ", CuerpoRespuesta)
print("Método de la solicitud: ",Metodo)
print("los encabezados de la solicitud: ",EncabezadoSolicitud)
print("Cuerpo de la solicitud: ",CuerpoSolicitud)
```

Llene los espacios de abajo con los valores obtenidos, conforme al protocolo que se muestra en la Fig.1:

Línea de respuesta	Versión del protocolo	Código	Descripción
	Según GPT el más común es de HTTPS/1.1	200	True
Encabezado	{'Content-Type': 'text/html; charset=utf-8', 'X-Frame-Options': 'DENY', 'Vary': 'Sec-Fetch-Dest, Sec-Fetch-Mode, Sec-Fetch-Site', 'Cache-Control': 'no-cache, no-store, max-age=0, must-revalidate', 'Pragma': 'no-cache', 'Expires': 'Mon, 01 Jan 1990 00:00:00 GMT', 'Date': 'Mon, 26 Aug 2024 04:16:26 GMT', 'P3P': 'CP="This is not a P3P policy! See g.co/p3phelp for more info."', 'Content-Security-Policy': "base-uri 'self';object-src 'none';report-uri /_/view/cspreport;script-src 'nonce-OX3mXknW0CRjHNn0EJMNjQ' 'unsafe-inline' 'unsafe-eval';worker-src 'self';frame-ancestors https://google-admin.corp.google.com/", 'Cross-Origin-Opener-Policy': 'unsafe-none', 'Cross-Origin-Resource-Policy': 'same-site', 'reporting-endpoints': 'default="/web-reports?jobset=prod&bl=editors.sites-viewer-frontend_20240813.02_p0&clss=1&context=eJwNyH1M1HUcB_DPvvf58GSlkMsFygYrXeTBgTUPBE74cehRrbxa399whAvOPMgTDjjF_gi9mA2iucyNMOJ4CgJvEpumW1v2sBVtbllzWGvDHtcCs1NOj8q73n-8_nml_Z08lKIpvErTM2s1PQefrNNk79O0E9KPa8qBeL-m-97RxCOa1oJjVpP1rKYy2PulpgD8NKcpAp5Lmjrgq280XQW9oOkl-HxJ03fw7rKm0zC0oukMeOOaOuGFB03ywb9ZJq1ab9LyBpM426SOh0zqhganSW1wEI7BA3tM2gibWk3aC16_SZ1wqsOkaSh5dJ7ikLR5nrLh_rqIygL3voiqBcfPN9UO2FQQVQXQbYuqPrB1RZUdbhyJqjvQ2XxHHYG50pi6DB-XxdQXcNsTU3GY6ImpM_Di1yuqCfrdcRUCHyVUAGLrE4o2JNTyoYT6D9yHE6oWdvcpSwM8bbFYNLh_s1hq4dCNEu6Cgug23gqqv5TT4MLpUv4UMraUcyYcbSznHsgVB-eBq9jBu6BkzsGVkDK9nddApKaCVyB4vYJ74WR_JQ_C-elKvghl4Up2wskWgwehIWBwM1zqMvgKFAYNrgA1Y3AafDBr8IdQfA4Pk1cMnoHuRYP7IPW6wengjBj8FPhyqjgAeU1VXATjF6o4DMGLVdwLq5WT14G9ycnb4eFWJ1vhtd-d_AbY66v5aG8198Dbb-7gMXjrvZ08APsyXNwCv8AivHrYxcfgr9ddfBt-TK7hX2FjWg3nw4mZITkFVz8akmtg9YTkcZCpkNwLhT-EpBgu3wrJ9_BPLCRqJSQ3R4clHh-WpMSwuPJHZBeklo9IOtjqRsQOyf5RWQ2JW6OyP2tMOmDq5TGZhdbjY9IJ17LH5Q9IHRiXdNj25PtigFqYkCFjUibgiZ5JeRYyz05KLrj_nJJaeAWCkHt3SvJgvnFaFqC8OSzVkHM-LI-AZzks-yFYtyi9YDqWZA_s_mxJGiDjnpSB4My5pDWhu4PfWjL5ea_nYO5mT6O3_YC_Lb_N2-5pswZwHr91r_-Ar93ja6wvshU9ZrMXbsm3FdW32P4HU1KGtw&build-label=editors.sites-viewer-frontend_20240813.02_p0&imp-sid=CLb_-5vmkYgDFTXxcwQdIm0heA&is-cached-offline=false"', 'Referrer-Policy': 'origin', 'Content-Encoding': 'gzip',		

	'Server': 'ESF', 'X-XSS-Protection': '0', 'X-Content-Type-Options': 'nosniff', 'Set-Cookie': 'NID=517=nbx-HSaIyb1xN6AyRjhod3gRvL3S9WdL6qjpnySmcliQMJWi9w6SK7n4 -A9dQlEIDqanu0FVKOW5DAAqim3ZO3y-5h9EAK0rFPlSsX0ciVrfmr3gaLSTJHMmr aQxIkoWHTIaq5U1XArIpsQt4CmmCXzChzTUfklMCgFPG7Rjp5Q; expires=Tue, 25-Feb-2025 04:16:26 GMT; path=/; domain=.google.com; HttpOnly', 'Transfer-Encoding': 'chunked'}
Línea vacía	
Cuerpo	<!DOCTYPE html><html lang="en-US" itemscope itemtype="http://schema.org/WebPage"><head><meta charset="utf-8"><script nonce="OX3mXknW0CRjHNn0EJMNjQ">var DOCS_timing={}; DOCS_timing['sl']=new Date().getTime();</script><script nonce="OX3mXknW0CRjHNn0EJMNjQ">function _DumpException(e) {throw e;}</script><script data-id="_gd" nonce="OX3mXknW0CRjHNn0EJMNjQ">window.WIZ_global_data = {"nQyAE":{}};</script><script nonce="OX3mXknW0CRjHNn0EJMNjQ">_docs_flag_initialData={"atari-emtpr":false,"atar i-ebidm":true,"atari-ebids":true,"atari-eibrm":false,"docs-text-elei":false,"doc s-text-usc":true,"atari-bae":false,"docs-text-emtps":true,"docs-text-etsrdpn":fa lse,"docs-text-etsrds":false,"docs-text-endes":false,"docs-text-escpv":true,"doc s-text-ecfs":false,"docs-text-ecis":false,"docs-text-eectfs":false,"docs-text-ed ctzs":true,"docs-text-eetxpc":false,"docs-text-eetxp":false,"docs-text-ertkmcp": true,"docs-text-ettctvs":false,"docs-text-ettts":false,"docs-text-escoubs":false ,"docs-text-escivs":false,"docs-text-escitrbs":false,"docs-text-ecgvd":false,"do cs-text-esbbcis":true,"docs-text-esbbcts":false,"docs-text-etccdts":false,"docs- text-etcchrs":false,"docs-text-etctrs":false,"docs-text-eltbbs":false,"docs-text -ecltts":false,"docs-text-etb":false,"docs-text-esbefr":false,"docs-text-ipi":fa lse,"docs-etshc":false,"docs-text-tbcb":2.0E7,"docs-efsmsdl":false,"docs-text-et of":false,"docs-text-ehlb":false,"docs-text-epa":true,"docs-text-dwit":false,"do cs-text-elawp":false,"docs-eec":false,"docs-ecot":"","docs-text-enbcr":false,"do cs-sup":"","umss":false,"docs-eldi":false,"docs-dli":false,"docs-liap":"/logImpr essions","ilcm":{"eui":"AHKXmL3h9AsqWpI67PpVtU23mGMV76ukYzLWlaM0XYIiRbSYALbbtNxJ nczZN6saKL2bk8eadXwH","je":1,"sstu":1724645786976181,"si":"CLb_-5vmkYgDFTXxcwQdI m0heA","gsc":null,"ei":[5703839,5704621,5706832,5706836,5707711,5737784,5737800, 5738513,5738529,5740798,5740814,5743108,5743124,5747263,5748013,5748029,5752678, 5752694,5753313,5753329,5754213,5754229,5755080,5755096,5758807,5758823,5762243, 5762259,5764252,5764268,5765535,5765551,5766761,5766777,5773662,5773678,5774331, 5774347,5774836,5774852,5776501,5776517,5784931,5784947,5784951,5784967,5791766, 5791782,5796135,5796151,5796457,5796473,5797275,5797291,14101306,14101502,141015 10,14101534,49372435,49372443,49375314,49375322,49472063,49472071,49622823,49622 831,49623173,49623181,49643568,49643576,49644015,49644023,49769337,49769345,4982 2921,49822929,49823164,49823172,49833462,49833470,49842855,49842863,49924706,499 24714,50221720,50221728,50266222,50266230,50273528,50273536,50297076,50297084,50 297426,50297434,50498907,50498915,50529103,50529111,50586962,50586970,70971256,7 0971264,71038255,71038263,71079938,71079946,71085241,71085249,71185170,71185178, 71197826,71197834,71238946,71238954,71252297,71252305,71289146,71289154,71346952 ,71346960,71387889,71387897,71429507,71429515,71473301,71473309,71478200,7147820 8,71478589,71478597,71528597,71528605,71530083,71530091,71532749,71532757,715333 61,71533377,71544834,71544842,71545513,71545521,71546425,71546433,71554472,71554 480,71560069,71560077,71560457,71560465,71561541,71561549,71573870,71573878,7159 2490,71592498,71612837,71612845,71614595,71614603,71631375,71631383,71658040,716 58048,71659813,71659821,71689860,71689868,71720760,71732098,71732106,71798420,71 798436,71868050,71868058,71897704,71897712,71897827,71897835,71924351,71924359,7 1928046,71928054,71960540,71960548,71961126,71961134,94351507,94351515,94353368, 94353376,94397741,94397749,94415365,94415373,94427697,94427705,94434257,94434265 ,94435578,94435586,94479346,94502654,94502662,94514761,94514769,94518793,9451880 1,94526768,94526776,94597639,94597647,94630911,94641005,94641013,94661802,946618 10,94670961,94670969,94707424,94707432,94738953,94738961,94784571,94784579,94887 682,94913439,94913447,94931531,94931539,94942490,94942498,95087186,95087194,9508 7227,95087235,95092002,95092010,95118551,95118559,95251262,95251270,95270945,952 70953,95286373,95286381,99310979,99310987,99368792,99368800,99402331,99402339]," crc":0,"cvi":[]},"docs-ccdil":false,"docs-eil":true,"info_params":{},"buildLabel ":"editors.sites-viewer-frontend_20240813.02_p0","docs-show_debug_info":false,"a tari-jefp":"/_/view/jserror","docs-jern":"view","atari-rhpp":"/_/view","docs-ecu ach":false,"docs-cclt":2033,"docs-ecci":true,"docs-esi":false,"docs-efypr":true, "docs-eyprp":true}; _docs_flag_cek= null ; if (window['DOCS_timing']) {DOCS_timing['ifdld']=new Date().getTime();}</script><meta name="viewport" content="width=device-width, initial-scale=1"><meta http-equiv="X-UA-Compatible" content="IE=edge"><meta name="referrer" content="origin"><link rel="icon" href="https://ssl.gstatic.com/atari/images/public/favicon.ico"><meta property="og:title" content="firstPage"><meta property="og:type" content="website"><meta property="og:url" content="https://sites.google.com/view/dagd-page/p%C3%A1gina-principal"><meta property="og:description" content="Acá esta la calculadora creada en las clases de IoT del 20/08/24 y del 22/08/24. "><meta itemprop="name" content="firstPage"><meta itemprop="description" content="Acá esta la calculadora creada en las clases de IoT del 20/08/24 y del 22/08/24. "><meta itemprop="url" content="https://sites.google.com/view/dagd-page/p%C3%A1gina-principal"><link href="https://fonts.googleapis.com/css?family=Roboto%20Slab%3A300%2C400%2C700%7C Droid%20Sans%3A400%2C700&display=swap" rel="stylesheet" nonce="aJUBU2O-V6QxpstE4dGZLw"><link

	href="https://fonts.googleapis.com/css?family=Google+Sans:400,500|Roboto:300,400,500,700|Source+Code+Pro:400,700&display=swap" rel="stylesheet" nonce="aJUBU2O-V6QxpstE4dGZLw"><link href="https://fonts.googleapis.com/css?family=Lato%3Ai%2Cbi%2C700%2C400&display=swap" rel="stylesheet" nonce="aJUBU2O-V6QxpstE4dGZLw"><style nonce="aJUBU2O-V6QxpstE4dGZLw">@media only screen and (max-width: 479px){.jgG6ef{font-size: 17.0pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.jgG6ef{font-size: 17.0pt;}}@media only screen and (min-width: 768px) and (max-width: 1279px){.jgG6ef{font-size: 18.0pt;}}@media only screen and (min-width: 1280px){.jgG6ef{font-size: 18.0pt;}}</style><link rel="stylesheet" href="https://www.gstatic.com/_/atari/_/ss/k=atari.vw.fz7XVYswIj4.L.W.O/am=WMEABA/d=1/rs=AGEqA5kpvfW56z_Au0NcvT7zsLW4VSc1EA" data-id="_cl" nonce="aJUBU2O-V6QxpstE4dGZLw"><script nonce="OX3mXknW0CRjHNn0EJMNjQ"></script><title>firstPage</title><style jsname="ptDGoc" nonce="aJUBU2O-V6QxpstE4dGZLw">.ImnMyf{background-color: rgba(255,255,255,1); color: rgba(33,33,33,1);}.Vs12Bd{color: rgba(33,33,33,1);}.S5d9Rd{background-color: rgba(255,213,79,1); color: rgba(33,33,33,1);}.O13XJf{height: 340px;}.O13XJf .IFuOkc{background-image: url(https://ssl.gstatic.com/atari/images/level-header.png); background-color: rgba(255,213,79,1); background-position-y: center;}.O13XJf .IFuOkc:before{background-color: rgba(33,33,33,1); opacity: 0.5; display: block;}.O13XJf .zfr3Q{color: rgba(255,255,255,1);}.O13XJf .qnVSj{color: rgba(255,255,255,1);}.O13XJf .Glwbz{color: rgba(255,255,255,1);}.O13XJf .qLrapd{color: rgba(255,255,255,1);}.O13XJf .aHM7ed{color: rgba(255,255,255,1);}.O13XJf .NHD4Gf{color: rgba(255,255,255,1);}.O13XJf .QmpIrf{background-color: rgba(255,255,255,1); border-color: rgba(255,255,255,1); color: rgba(0,0,0,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}@media only screen and (max-width: 479px){.O13XJf .QmpIrf{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.O13XJf .QmpIrf{font-size: 13pt;}}@media only screen and (max-width: 479px){.O13XJf{height: 250px;}}.SBrW1{height: 430px; padding-bottom: 120px; padding-top: 120px;}.Wew9ke{fill: rgba(255,255,255,1);}.fOU46b .YSH9J{color: rgba(255,255,255,1);}.fOU46b .KJll8d{background-color: rgba(255,255,255,1);}.fOU46b .Mz8gvb{color: rgba(255,255,255,1);}.fOU46b .G8QRnc .Mz8gvb{color: rgba(33,33,33,1);}.fOU46b .G8QRnc .YSH9J{color: rgba(33,33,33,1);}.fOU46b .G8QRnc .iWs3gf.chg4Jd:focus{background-color: rgba(33,33,33,0.1199999973);}.fOU46b .G8QRnc .KJll8d{background-color: rgba(33,33,33,1);}.fOU46b .usN8rf .YSH9J{color: rgba(33,33,33,1);}.fOU46b .usN8rf .KJll8d{background-color: rgba(33,33,33,1);}.fOU46b .usN8rf .Mz8gvb{color: rgba(33,33,33,1);}.fOU46b .aCIEDd .YSH9J{color: rgba(33,33,33,1);}.fOU46b .aCIEDd .KJll8d{background-color: rgba(33,33,33,1);}.fOU46b .aCIEDd .Mz8gvb{color: rgba(33,33,33,1);}.fOU46b .a3ETed .YSH9J{color: rgba(33,33,33,1);}.fOU46b .a3ETed .KJll8d{background-color: rgba(33,33,33,1);}.fOU46b .a3ETed .M9vuGd{border-bottom-color: rgba(0,0,0,0.150000006);}.fOU46b .a3ETed .Mz8gvb{color: rgba(33,33,33,1);}.fOU46b .zDUgLc{opacity: 0;}.fOU46b .LBrwzc .zDUgLc{opacity: 1; border-bottom-style: none;}.fOU46b .GBy4H .zDUgLc{opacity: 1;}@media only screen and (min-width: 1280px){.XeSM4.b2Iqye.fOU46b .LBrwzc .tCHXDc{color: rgba(255,255,255,1);}}.XeSM4.b2Iqye.fOU46b .LBrwzc .iWs3gf.chg4Jd:focus{background-color: rgba(255,255,255,0.1199999973);}@media only screen and (min-width: 1280px){.KuNac.b2Iqye.fOU46b .GBy4H .tCHXDc{color: rgba(33,33,33,1);}}.KuNac.b2Iqye.fOU46b .GBy4H .iWs3gf.chg4Jd:focus{background-color: rgba(33,33,33,0.1199999973);}@media only screen and (min-width: 1280px){.KuNac.G9Qloe.fOU46b .GBy4H .tCHXDc{color: rgba(255,255,255,1);}}.KuNac.G9Qloe.fOU46b .GBy4H .iWs3gf.chg4Jd:focus{background-color: rgba(255,255,255,0.1199999973);}@media only screen and (min-width: 1280px){.XeSM4.G9Qloe.fOU46b .LBrwzc .tCHXDc{color: rgba(33,33,33,1);}}.XeSM4.G9Qloe.fOU46b .LBrwzc .iWs3gf.chg4Jd:focus{background-color: rgba(33,33,33,0.1199999973);}.LBrwzc .YSH9J{color: rgba(33,33,33,1);}.LBrwzc .oNsfjf{color: rgba(33,33,33,1);}.LBrwzc .KJll8d{background-color: rgba(33,33,33,1);}.LBrwzc .YTv4We.chg4Jd:focus:before{border-color: rgba(33,33,33,1); display: block;}.LBrwzc .Mz8gvb{color: rgba(33,33,33,1);}.LBrwzc .wgxiMe{background-color: rgba(255,255,255,1);}.LBrwzc .zDUgLc{border-bottom-color: rgba(204,204,204,1); border-bottom-width: 1px; border-bottom-style: solid;}.GBy4H .wgxiMe{background-color: rgba(0,0,0,1);}.JzO0Vc{background-color: rgba(79,71,78,1);}.M63kCb{background-color: rgba(245,242,240,1);}.zfr3Q{font-family: 'Droid Sans'; color: rgba(33,33,33,1); font-size: 13pt; line-height: 1.55; margin-top: 14px;}.qnVSj{color: rgba(33,33,33,1);}.Glwbz{color: rgba(33,33,33,1);}.qLrapd{color: rgba(33,33,33,1);}.aHM7ed{color: rgba(33,33,33,1);}.NHD4Gf{color: rgba(33,33,33,1);}.zfr3Q .OUGEr{color: rgba(33,33,33,1);}.dhtgD{text-decoration: none; border-bottom-style: solid; border-bottom-width: 1px; font-weight: 700;}.dhtgD:hover{border-bottom-color: rgba(255,213,79,1); border-bottom-style: solid; border-bottom-width: 1px;}.lQAHbd .dhtgD:visited{border-bottom-color: rgba(245,242,240,1);}.lQAHbd .dhtgD:hover{border-bottom-color: rgba(33,33,33,1);}.dhtgD:active{border-bottom-color: rgba(255,213,79,1); border-bottom-style: solid; border-bottom-width: 1px;}.dhtgD:visited{border-bottom-color: rgba(255,213,79,1); border-bottom-style: solid; border-bottom-width: 1px;}.duRjpb{font-family: 'Roboto Slab'; font-size: 37pt; line-height: 1.25; font-weight: 300; margin-top: 20px;}.Ap4VC{margin-bottom: -20px;}.JYVBee{font-family: 'Roboto Slab'; font-size: 28pt; line-height: 1.21; font-weight: 300; margin-top: 20px;}.CobnVe{margin-bottom: -20px;}.OmQG5e{font-family: 'Droid Sans';

font-size: 16pt; line-height: 1.27; margin-top: 18px; font-weight: 400;}.GV3q8e{margin-bottom: -18px;}.TMjjoe{font-family: 'Roboto Slab'; font-size: 9pt; line-height: 1.2; margin-top: 0px;}.Zjiec{font-family: 'Roboto Slab'; font-weight: 300; font-size: 15pt; line-height: 1.4; margin-top: 48px; margin-left: 48px; margin-bottom: 62px; margin-right: 32px;}.XMyrgf{margin-top: 48px; margin-left: 48px; margin-bottom: 0px; margin-right: 32px;}.PsKE7e{font-family: 'Droid Sans'; font-weight: 700; font-size: 12pt; padding-left: 10px; padding-right: 10px;}.lhZOrc{font-weight: 700;}.lhZOrc .hDrhEe{border-bottom-width: 7px; border-bottom-color: rgba(255,213,79,1); border-bottom-style: solid;}.TlfmSc{font-family: 'Roboto Slab'; font-weight: 300; font-size: 15pt; line-height: 1.4;}.zDUgLc{background-color: rgba(79,71,78,1); opacity: 1;}.baZpAe{background-color: rgba(255,255,255,1);}.N0neUc{background-color: rgba(255,255,255,0);}.ySLm4c{background-color: rgba(255,255,255,0); font-family: 'Droid Sans';}.KUB40b{background-color: transparent;}.OWlOyc{background-color: rgba(255,255,255,0);}.iwQgFb{background-color: rgba(0,0,0,0.150000006); height: 3px; margin-top: 8px;}.CbiMKe{background-color: rgba(255,213,79,1);}.QmpIrf{background-color: rgba(255,213,79,1); border-color: rgba(255,213,79,1); color: rgba(33,33,33,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.LRAOtb{background-color: rgba(255,255,255,1);}.qeLZfd .zfr3Q{color: rgba(33,33,33,1);}.qeLZfd .zfr3Q .OUGEr{color: rgba(33,33,33,1);}.qeLZfd .qnVSj{color: rgba(33,33,33,1);}.qeLZfd .Glwbz{color: rgba(33,33,33,1);}.qeLZfd .qLrapd{color: rgba(33,33,33,1);}.qeLZfd .aHM7ed{color: rgba(33,33,33,1);}.qeLZfd .NHD4Gf{color: rgba(33,33,33,1);}.qeLZfd .LRAOtb{background-color: transparent;}.qeLZfd .baZpAe{background-color: transparent;}.qeLZfd .iwQgFb{background-color: rgba(0,0,0,0.150000006);}.lQAHbd:before{background-color: rgba(255,213,79,1); display: block;}.lQAHbd .zfr3Q{color: rgba(33,33,33,1);}.lQAHbd .zfr3Q .OUGEr{color: rgba(33,33,33,1);}.lQAHbd .qnVSj{color: rgba(33,33,33,1);}.lQAHbd .Glwbz{color: rgba(33,33,33,1);}.lQAHbd .qLrapd{color: rgba(33,33,33,1);}.lQAHbd .aHM7ed{color: rgba(33,33,33,1);}.lQAHbd .NHD4Gf{color: rgba(33,33,33,1);}.lQAHbd .LRAOtb{background-color: transparent;}.lQAHbd .baZpAe{background-color: transparent;}.lQAHbd .iwQgFb{background-color: rgba(0,0,0,0.150000006);}.lQAHbd .QmpIrf{background-color: rgba(255,255,255,1); border-color: rgba(255,255,255,1); color: rgba(0,0,0,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.lQAHbd .CbiMKe{background-color: rgba(255,255,255,1);}@media only screen and (max-width: 479px){.lQAHbd .QmpIrf{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.lQAHbd .QmpIrf{font-size: 13pt;}}.LB7kq .IFuOkc{border-bottom-color: rgba(255,213,79,1); border-bottom-width: 10px; border-bottom-style: solid;}.LB7kq .duRjpb{font-size: 63pt; line-height: 1.25; margin-bottom: 19px;}@media only screen and (max-width: 479px){.LB7kq .duRjpb{font-size: 41pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.LB7kq .duRjpb{font-size: 53pt;}}.gk8rDe .duRjpb{font-size: 48pt;}.LB7kq.gk8rDe .zfr3Q{color: rgba(0,0,0,1);}@media only screen and (max-width: 479px){.gk8rDe .duRjpb{font-size: 32pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.gk8rDe .duRjpb{font-size: 41pt;}}.LB7kq .baZpAe{background-color: transparent;}.LB7kq .JYVBee{font-size: 28pt; line-height: 1.18;}@media only screen and (max-width: 479px){.LB7kq .JYVBee{font-size: 21pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.LB7kq .JYVBee{font-size: 25pt;}}.cJgDec .baZpAe{background-color: transparent;}.cJgDec .baZpAe .OUGEr{color: rgba(255,255,255,1);}.cJgDec .baZpAe .LRAOtb{background-color: transparent;}.cJgDec .zfr3Q{color: rgba(255,255,255,1);}.cJgDec .qnVSj{color: rgba(255,255,255,1);}.cJgDec .Glwbz{color: rgba(255,255,255,1);}.cJgDec .qLrapd{color: rgba(255,255,255,1);}.cJgDec .aHM7ed{color: rgba(255,255,255,1);}.cJgDec .NHD4Gf{color: rgba(255,255,255,1);}.cJgDec .IFuOkc:before{background-color: rgba(33,33,33,1); opacity: 0.5; display: block;}.cJgDec .QmpIrf{background-color: rgba(255,255,255,1); border-color: rgba(255,255,255,1); color: rgba(0,0,0,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}@media only screen and (max-width: 479px){.cJgDec .QmpIrf{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.cJgDec .QmpIrf{font-size: 13pt;}}.tpmmCb .baZpAe{background-color: transparent;}.tpmmCb .baZpAe .OUGEr{color: rgba(0,0,0,1);}.tpmmCb .baZpAe .LRAOtb{background-color: transparent;}.tpmmCb .zfr3Q{color: rgba(0,0,0,1);}.tpmmCb .qnVSj{color: rgba(0,0,0,1);}.tpmmCb .Glwbz{color: rgba(0,0,0,1);}.tpmmCb .qLrapd{color: rgba(0,0,0,1);}.tpmmCb .aHM7ed{color: rgba(0,0,0,1);}.tpmmCb .NHD4Gf{color: rgba(0,0,0,1);}.tpmmCb .IFuOkc:before{background-color: rgba(255,255,255,1); display: block;}.tpmmCb .Wew9ke{fill: rgba(0,0,0,1);}.tpmmCb .QmpIrf{background-color: rgba(255,255,255,1); border-color: rgba(255,255,255,1); color: rgba(0,0,0,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}@media only screen and (max-width: 479px){.tpmmCb .QmpIrf{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.tpmmCb .QmpIrf{font-size: 13pt;}}.gk8rDe{padding-top: 60px;}.gk8rDe .QmpIrf{background-color: rgba(255,213,79,1); border-color: rgba(255,213,79,1); color: rgba(33,33,33,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}@media only screen and (max-width: 479px){.gk8rDe .QmpIrf{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.gk8rDe .QmpIrf{font-size: 13pt;}}@media only screen and (max-width: 479px){.gk8rDe{padding-top: 32px; padding-bottom: 32px;}}.YSH9J{color: rgba(255,255,255,1);}.iWs3gf.chg4Jd:focus{background-color: rgba(255,255,255,0.1199999973);}.oNsfjf{color: rgba(255,255,255,1);}.wgxiMe{background-color: rgba(79,71,78,1);}.qV4dIc{border-bottom-color: rgba(255,255,255,0); border-bottom-style: solid; border-bottom-width: 8px; padding-top: 14px;

padding-bottom: 6px; padding-left: 2px; padding-right: 2px; margin-left: 10px; margin-right: 10px;}.jgXgSe{line-height: 28px;}.u5fiyc{line-height: 28px;}.M9vuGd{border-bottom-color: rgba(255,213,79,1);}.eWDljc{background-color: rgba(79,71,78,1); padding-bottom: 28px;}.IKA38e{padding-left: 36px; margin-top: 20px;}.hDrhEe{font-family: 'Droid Sans'; margin-left: 12px; margin-right: 12px; padding-left: 0px; display: inline-block; max-width: 90%;}.baH5ib .hDrhEe{margin-left: 4px; padding-left: 0px;}.PsKE7e:hover{opacity: 0.6;}.BFDQOb:hover{opacity: 0.6;}.QcmuFb{padding-left: 20px;}.vDPrib{padding-left: 40px;}.TBDXjd{padding-left: 60px;}.bYeK8e{padding-left: 80px;}.CuqSDe{padding-left: 100px;}.Havqpe{padding-left: 120px;}.JvDrRe{padding-left: 140px;}.o5lrIf{padding-left: 160px;}.yOJW7c{padding-left: 180px;}.rB8cye{padding-left: 200px;}.RuayVd{padding-right: 20px;}.YzcKX{padding-right: 40px;}.reTV0b{padding-right: 60px;}.vSYeUc{padding-right: 80px;}.PxtZIe{padding-right: 100px;}.ahQMed{padding-right: 120px;}.rzhcXb{padding-right: 140px;}.PBhj0b{padding-right: 160px;}.TlN46c{padding-right: 180px;}.GEdNnc{padding-right: 200px;}.xkUom{border-color: rgba(33,33,33,1); color: rgba(33,33,33,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.xkUom:hover{background-color: rgba(33,33,33,0.1000000015); font-size: 13pt; line-height: 22px;}.KjwKmc{color: rgba(33,33,33,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.KjwKmc:hover{background-color: rgba(33,33,33,0.1000000015); font-size: 13pt; line-height: 22px;}.lQAHbd .xkUom{border-color: rgba(33,33,33,1); color: rgba(33,33,33,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.lQAHbd .xkUom:hover{background-color: rgba(255,255,255,0.1000000015);}.lQAHbd .KjwKmc{color: rgba(33,33,33,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.lQAHbd .KjwKmc:hover{background-color: rgba(255,255,255,0.1000000015);}@media only screen and (max-width: 479px){.lQAHbd .xkUom{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.lQAHbd .xkUom{font-size: 13pt;}}@media only screen and (max-width: 479px){.lQAHbd .KjwKmc{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.lQAHbd .KjwKmc{font-size: 13pt;}}.lQAHbd .Mt0nFe{border-color: rgba(0,0,0,0.200000003);}.cJgDec .xkUom{border-color: rgba(255,255,255,1); color: rgba(255,255,255,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.cJgDec .xkUom:hover{background-color: rgba(255,255,255,0.1000000015);}.cJgDec .KjwKmc{color: rgba(255,255,255,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.cJgDec .KjwKmc:hover{background-color: rgba(255,255,255,0.1000000015);}@media only screen and (max-width: 479px){.cJgDec .xkUom{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.cJgDec .xkUom{font-size: 13pt;}}@media only screen and (max-width: 479px){.cJgDec .KjwKmc{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.cJgDec .KjwKmc{font-size: 13pt;}}.tpmmCb .xkUom{border-color: rgba(0,0,0,1); color: rgba(0,0,0,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.tpmmCb .xkUom:hover{background-color: rgba(33,33,33,0.1000000015);}.tpmmCb .KjwKmc{color: rgba(0,0,0,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.tpmmCb .KjwKmc:hover{background-color: rgba(33,33,33,0.1000000015);}@media only screen and (max-width: 479px){.tpmmCb .xkUom{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.tpmmCb .xkUom{font-size: 13pt;}}@media only screen and (max-width: 479px){.tpmmCb .KjwKmc{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.tpmmCb .KjwKmc{font-size: 13pt;}}.gk8rDe .xkUom{border-color: rgba(33,33,33,1); color: rgba(33,33,33,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.gk8rDe .xkUom:hover{background-color: rgba(33,33,33,0.1000000015);}.gk8rDe .KjwKmc{color: rgba(33,33,33,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.gk8rDe .KjwKmc:hover{background-color: rgba(33,33,33,0.1000000015);}@media only screen and (max-width: 479px){.gk8rDe .xkUom{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.gk8rDe .xkUom{font-size: 13pt;}}@media only screen and (max-width: 479px){.gk8rDe .KjwKmc{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.gk8rDe .KjwKmc{font-size: 13pt;}}.O13XJf .xkUom{border-color: rgba(255,255,255,1); color: rgba(255,255,255,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.O13XJf .xkUom:hover{background-color: rgba(255,255,255,0.1000000015);}.O13XJf .KjwKmc{color: rgba(255,255,255,1); font-family: 'Droid Sans'; font-size: 13pt; line-height: 22px;}.O13XJf .KjwKmc:hover{background-color: rgba(255,255,255,0.1000000015);}@media only screen and (max-width: 479px){.O13XJf .xkUom{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.O13XJf .xkUom{font-size: 13pt;}}@media only screen and (max-width: 479px){.O13XJf .KjwKmc{font-size: 13pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.O13XJf .KjwKmc{font-size: 13pt;}}.Y4CpGd{font-family: 'Droid Sans'; font-size: 13pt;}.CMArNe{background-color: rgba(245,242,240,1);}@media only screen and (max-width: 479px){.duRjpb{font-size: 26pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.duRjpb{font-size: 32pt;}}@media only screen and (max-width: 479px){.JYVBee{font-size: 21pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.JYVBee{font-size: 25pt;}}@media only screen and (max-width: 479px){.OmQG5e{font-size: 15pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.OmQG5e{font-size: 15pt;}}@media only screen and (max-width: 479px){.TMjjoe{font-size: 9pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.TMjjoe{font-size: 9pt;}}@media only screen and (max-width: 479px){.Zjiec{font-size: 14pt;}}@media only screen and (min-width: 480px) and (max-width: 767px){.Zjiec{font-size: 15pt;}}@media only screen and

(max-width: 479px){.PsKE7e{font-size: 12pt;}}@media only screen and (min-width:
480px) and (max-width: 767px){.PsKE7e{font-size: 12pt;}}@media only screen and
(max-width: 479px){.TlfmSc{font-size: 14pt;}}@media only screen and (min-width:
480px) and (max-width: 767px){.TlfmSc{font-size: 15pt;}}@media only screen and
(max-width: 479px){.QmpIrf{font-size: 13pt;}}@media only screen and (min-width:
480px) and (max-width: 767px){.QmpIrf{font-size: 13pt;}}@media only screen and
(max-width: 479px){.xkUom{font-size: 13pt;}}@media only screen and (min-width:
480px) and (max-width: 767px){.xkUom{font-size: 13pt;}}@media only screen and
(max-width: 479px){.xkUom:hover{font-size: 13pt;}}@media only screen and
(min-width: 480px) and (max-width: 767px){.xkUom:hover{font-size: 13pt;}}@media
only screen and (max-width: 479px){.KjwKmc{font-size: 13pt;}}@media only screen
and (min-width: 480px) and (max-width: 767px){.KjwKmc{font-size: 13pt;}}@media
only screen and (max-width: 479px){.KjwKmc:hover{font-size: 13pt;}}@media only
screen and (min-width: 480px) and (max-width: 767px){.KjwKmc:hover{font-size:
13pt;}}section[id="h.455cc887f3e563eb_3"] .IFuOkc:before{opacity:
0.7;}</style><script nonce="OX3mXknW0CRjHNn0EJMNjQ">_at_config =
[null,"AIzaSyChg3MFqzdi1P5J-YvEyakkSA1yU7HRcDI","897606708560-a63d8ia0t9dhtpdt4i
3djab2m42see7o.apps.googleusercontent.com",null,null,null,null,null,null,null,nu
ll,null,null,null,"SITES_%s",null,null,null,null,null,null,null,null,null,["AHKX
mL3h9AsqWpI67PpVtU23mGMV76ukYzLWlaM0XYIiRbSYALbbtNxJnczZN6saKL2bk8eadXwH",1,"CLb
_-5vmkYgDFTXxcwQdIm0heA",1724645786976181,[5703839,5704621,5706832,5706836,57077
11,5737784,5737800,5738513,5738529,5740798,5740814,5743108,5743124,5747263,57480
13,5748029,5752678,5752694,5753313,5753329,5754213,5754229,5755080,5755096,57588
07,5758823,5762243,5762259,5764252,5764268,5765535,5765551,5766761,5766777,57736
62,5773678,5774331,5774347,5774836,5774852,5776501,5776517,5784931,5784947,57849
51,5784967,5791766,5791782,5796135,5796151,5796457,5796473,5797275,5797291,14101
306,14101502,14101510,14101534,49372435,49372443,49375314,49375322,49472063,4947
2071,49622823,49622831,49623173,49623181,49643568,49643576,49644015,49644023,497
69337,49769345,49822921,49822929,49823164,49823172,49833462,49833470,49842855,49
842863,49924706,49924714,50221720,50221728,50266222,50266230,50273528,50273536,5
0297076,50297084,50297426,50297434,50498907,50498915,50529103,50529111,50586962,
50586970,70971256,70971264,71038255,71038263,71079938,71079946,71085241,71085249
,71185170,71185178,71197826,71197834,71238946,71238954,71252297,71252305,7128914
6,71289154,71346952,71346960,71387889,71387897,71429507,71429515,71473301,714733
09,71478200,71478208,71478589,71478597,71528597,71528605,71530083,71530091,71532
749,71532757,71533361,71533377,71544834,71544842,71545513,71545521,71546425,7154
6433,71554472,71554480,71560069,71560077,71560457,71560465,71561541,71561549,715
73870,71573878,71592490,71592498,71612837,71612845,71614595,71614603,71631375,71
631383,71658040,71658048,71659813,71659821,71689860,71689868,71720760,71732098,7
1732106,71798420,71798436,71868050,71868058,71897704,71897712,71897827,71897835,
71924351,71924359,71928046,71928054,71960540,71960548,71961126,71961134,94351507
,94351515,94353368,94353376,94397741,94397749,94415365,94415373,94427697,9442770
5,94434257,94434265,94435578,94435586,94479346,94502654,94502662,94514761,945147
69,94518793,94518801,94526768,94526776,94597639,94597647,94630911,94641005,94641
013,94661802,94661810,94670961,94670969,94707424,94707432,94738953,94738961,9478
4571,94784579,94887682,94913439,94913447,94931531,94931539,94942490,94942498,950
87186,95087194,95087227,95087235,95092002,95092010,95118551,95118559,95251262,95
251270,95270945,95270953,95286373,95286381,99310979,99310987,99368792,99368800,9
9402331,99402339]],null,null,null,null,0,null,null,null,null,null,null,null,null
,null,"https://drive.google.com",null,null,null,null,null,null,null,null,null,0,
1,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,nul
l,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,nul
l,null,null,null,null,null,null,null,null,null,null,null,null,null,null,"v2inter
nal","https://docs.google.com",null,null,null,null,null,null,"https://sites.goog
le.com/new/",null,null,null,null,null,0,null,null,null,null,null,null,null,null,
null,null,null,null,null,null,null,null,null,null,null,null,null,1,"",null,null,
null,null,null,null,null,null,null,null,null,6,null,null,"https://accounts.
google.com/o/oauth2/auth","https://accounts.google.com/o/oauth2/postmessageRelay
",null,null,null,null,78,"https://sites.google.com/new/?usp\u003dviewer_footer",
null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,
null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,
null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,
null,null,null,null,null,"https://www.gstatic.com/atari/embeds/83a60601c213b72fb
19c1855fb0c5f26/intermediate-frame-minified.html",0,null,"v2beta",null,null,null
,null,null,null,4,"https://accounts.google.com/o/oauth2/iframe",null,null,null,n
ull,null,null,"https://310827503-atari-embeds.googleusercontent.com/embeds/16cb2
04cf3a9d4d223a0a3fd8b0eec5d/inner-frame-minified.html",null,null,null,null,null,
null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,
null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,
null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,
null,null,null,null,null,null,null,null,null,null,null,null,0,null,null,nul
l,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,nul
l,"https://sites.google.com/view/dagd-page/p%C3%A1gina-principal",null,null,null
,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null
,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,0,nu
ll,null,null,null,null,null,0,null,"",null,null,null,null,null,null,null,null,nu
ll,null,null,null,null,null,null,null,null,null,null,null,null,null,0,null,null,
null,null,null,null,null,null,null,null,null,null,null,null,null,1,null,null,[17
24645786976,"editors.sites-viewer-frontend_20240813.02_p0","662445483",null,1,1,
""],null,null,null,null,0,null,null,0,null,null,null,null,null,null,null,null,20
,500,"https://domains.google.com",null,0,null,null,null,null,null,null,null,null
,null,null,null,0,null,null,null,null,null,null,null,null,null,null,1,0,1,0,0,0,
0,null,null,null,null,null,"https://www.google.com/calendar/embed",null,null,nul
l,null,0,null,null,null,null,null,null,null,null,null,null,0,null,null,null,null
,null,1,1,null,null,null,null,null,null,"PROD",0,null,0,null,0]; window.globals
=

{"enableAnalytics":true,"webPropertyId":"","showDebug":false,"hashedSiteId":"051dda2d4751bd8a39432a33842747aca33386578b715e5090bab349423ff965","normalizedPath":"view/dagd-page/página-principal","pageTitle":"Página principal"}; function gapiLoaded() {if (globals.gapiLoaded == undefined) {globals.gapiLoaded = true;} else {globals.gapiLoaded();}}window.messages = []; window.addEventListener && window.addEventListener('message', function(e) {if (window.messages && e.data && e.data.magic == 'SHIC') {window.messages.push(e);}});</script><script src="https://apis.google.com/js/client.js?onload=gapiLoaded" nonce="OX3mXknW0CRjHNn0EJMNjQ"></script><script nonce="OX3mXknW0CRjHNn0EJMNjQ">(function(){}).call(this);
</script><script nonce="OX3mXknW0CRjHNn0EJMNjQ">const imageUrl = null ;
      function bgImgLoaded() {
        if (!globals.headerBgImgLoaded) {
          globals.headerBgImgLoaded = new Date().getTime();
        } else {
          globals.headerBgImgLoaded();
        }
      }
      if (imageUrl) {
        const img = new Image();
        img.src = imageUrl;
        img.onload = bgImgLoaded;
        globals.headerBgImgExists = true;
      } else {
        globals.headerBgImgExists = false;
      }
      </script></head><body dir="ltr" itemscope itemtype="http://schema.org/WebPage" id="yDmH0d" css="yDmH0d"> <div jscontroller="o1L5Wb" data-sitename="dagd-page" data-search-scope="1" data-universe="1" jsmodel="fNFZH" jsaction="Pe9H6d:cZFEp;WMZaJ:VsGN3;hJluRd:UADL7b;zuqEgd:HI9w0;tr6QDd:Y8aXB;MxH79b:xDkBfb;JIbuQc:SPXMTb(uxAMZ),LjG1Ed(a6mxbb);" jsname="G0jgYd"> <div role="button" class="U26fgb mUbCce fKz7Od h3nfre M9Bg4d" jscontroller="VXdfxd" jsaction="click:cOuCgd; mousedown:UX7yZ; mouseup:lbsD7e; mouseenter:tfO1Yc; mouseleave:JywGue; focus:AHmuwe; blur:O22p3e; contextmenu:mg9Pef;touchstart:p6p2H; touchmove:FwuNnf; touchend:yfqBxc(preventDefault=true); touchcancel:JMtRjd;" jsshadow jsname="GeGHKb" aria-label="Back to site" aria-disabled="false" tabindex="0" data-tooltip="Back to site" data-tooltip-vertical-offset="-12" data-tooltip-horizontal-offset="0"> <span jsslot class="xjKiLb"><svg class="V4YR2c" viewBox="0 0 24 24" focusable="false"><path d="M0 0h24v24H0z" fill="none"/><path d="M20 11H7.83l5.59-5.59L12 4l-8 8 8 8 1.41-1.41L7.83 13H20v-2z"/></svg> <div class="rFrNMe b7AJhc zKHdkd" jscontroller="pxq3x" jsaction="clickonly:KjsqPd; focus:Jt1EX; blur:fpfTEe; input:Lg5SV" jsshadow jsname="OH0EC" aria-expanded="true"> <span jsslot class="A37UZe qgcB3c iHd5yb"><div role="button" class="U26fgb mUbCce fKz7Od i3PoXe M9Bg4d" jscontroller="VXdfxd" jsaction="click:cOuCgd; mousedown:UX7yZ; mouseup:lbsD7e; mouseenter:tfO1Yc; mouseleave:JywGue; focus:AHmuwe; blur:O22p3e; contextmenu:mg9Pef;touchstart:p6p2H; touchmove:FwuNnf; touchend:yfqBxc(preventDefault=true); touchcancel:JMtRjd;" jsshadow jsname="lfEfFf" aria-label="Search" aria-disabled="false" tabindex="0" data-tooltip="Search" data-tooltip-vertical-offset="-12" data-tooltip-horizontal-offset="0"> <span jsslot class="xjKiLb"><svg class="vu8Pwe" viewBox="0 0 24 24" focusable="false"><path d="M15.5 14h-.79l-.28-.27C15.41 12.59 16 11.11 16 9.5 16 5.91 13.09 3 9.5 3S3 5.91 3 9.5 5.91 16 9.5 16c1.61 0 3.09-.59 4.23-1.57l.27.28v.79l5 4.99L20.49 19l-4.99-5zm-6 0C7.01 14 5 11.99 5 9.5S7.01 5 9.5 5 14 7.01 14 9.5 11.99 14 9.5 14z"/><path d="M0 0h24v24H0z" fill="none"/></svg> <div class="xq3j6

ERcjC"> <div role="button" class="U26fgb mUbCce fKz7Od JyJRXe M9Bg4d" jscontroller="VXdfxd" jsaction="click:cOuCgd; mousedown:UX7yZ; mouseup:lbsD7e; mouseenter:tfO1Yc; mouseleave:JywGue; focus:AHmuwe; blur:O22p3e; contextmenu:mg9Pef;touchstart:p6p2H; touchmove:FwuNnf; touchend:yfqBxc(preventDefault=true); touchcancel:JMtRjd;" jsshadow jsname="m1xNUe" aria-label="Back to site" aria-disabled="false" tabindex="0" data-tooltip="Back to site" data-tooltip-vertical-offset="-12" data-tooltip-horizontal-offset="0"> <span jsslot class="xjKiLb"><svg class="V4YR2c" viewBox="0 0 24 24" focusable="false"><path d="M0 0h24v24H0z" fill="none"/><path d="M20 11H7.83l5.59-5.59L12 4l-8 8 8 8 1.41-1.41L7.83 13H20v-2z"/></svg> <input type="search" class="whsOnd zHQkBf" jsname="YPqjbf" autocomplete="off" tabindex="0" aria-label="Search this site" value="" aria-disabled="false" autofocus role="combobox" data-initial-value=""/> Search this site <span jsslot class="A37UZe sxyYjd MQL3Ob"><div role="button" class="U26fgb mUbCce fKz7Od Kk06A M9Bg4d" jscontroller="VXdfxd" jsaction="click:cOuCgd; mousedown:UX7yZ; mouseup:lbsD7e; mouseenter:tfO1Yc; mouseleave:JywGue; focus:AHmuwe; blur:O22p3e; contextmenu:mg9Pef;touchstart:p6p2H; touchmove:FwuNnf; touchend:yfqBxc(preventDefault=true); touchcancel:JMtRjd;" jsshadow jsname="pZn8Oc" aria-label="Clear search" aria-disabled="false" tabindex="0" data-tooltip="Clear search" data-tooltip-vertical-offset="-12" data-tooltip-horizontal-offset="0"> <span jsslot class="xjKiLb"><svg class="fAUEUd" viewBox="0 0 24 24" focusable="false"><path d="M19 6.41L17.59 5 12 10.59 6.41 5 5 6.41 10.59 12 5 17.59 6.41 19 12 13.41 17.59 19 19 17.59 13.41 12z"></path><path d="M0 0h24v24H0z" fill="none"></path></svg> <style nonce="aJUBU2O-V6QxpstE4dGZLw">.rrJNTc{opacity: 0;}.bKy5e{pointer-events: none; position: absolute; top: 0;}</style> <button class="VfPpkd-LgbsSe VfPpkd-LgbsSe-OWXEXe-dgl2Hf LjDxcd XhPA0b LQeN7 WsSUlf jz7fPb" jscontroller="soHxf" jsaction="click:cOuCgd; mousedown:UX7yZ; mouseup:lbsD7e; mouseenter:tfO1Yc; mouseleave:JywGue; touchstart:p6p2H; touchmove:FwuNnf; touchend:yfqBxc; touchcancel:JMtRjd; focus:AHmuwe; blur:O22p3e; contextmenu:mg9Pef;mlnRJb:fLiPzd;" data-idom-class="LjDxcd XhPA0b LQeN7 WsSUlf jz7fPb" jsname="z2EeY" tabindex="0"> Skip to main content</button> <button class="VfPpkd-LgbsSe VfPpkd-LgbsSe-OWXEXe-dgl2Hf LjDxcd XhPA0b LQeN7 WsSUlf br90J" jscontroller="soHxf" jsaction="click:cOuCgd; mousedown:UX7yZ; mouseup:lbsD7e; mouseenter:tfO1Yc; mouseleave:JywGue; touchstart:p6p2H; touchmove:FwuNnf; touchend:yfqBxc; touchcancel:JMtRjd; focus:AHmuwe; blur:O22p3e; contextmenu:mg9Pef;mlnRJb:fLiPzd;" data-idom-class="LjDxcd XhPA0b LQeN7 WsSUlf br90J" jsname="ilzYPe" tabindex="0"> Skip to navigation</button> <div class="fktJzd AKpWA fOU46b yMcSQd Ly6Unf G9Qloe XeSM4 XxIgdb" jsname="UzWXSb" data-uses-custom-theme="false" data-legacy-theme-name="Level" data-legacy-theme-font-kit="Slab" data-legacy-theme-color-kit="LightYellow" jscontroller="Md9ENb" jsaction="gsiSmd:Ffcznf;yj5fUd:cpPetb;HNXL3:q0Vyke;e2SXKd:IPDu5e;BdXpgd:nhk7K;rcuQ6b:WYd;"><header id="atIdViewHeader"> firstPage </header> <section id="h.INITIAL_GRID.jedn7z1ufdvm" class="yaqOZd LB7kq nyKByd O13XJf"> <h1

	id="h.1jqlwm7tllfa" dir="ltr" class="zfr3Q duRjpb CDt4Ke " style="text-align: center;">Calculadora dagd</h1> </section><section id="h.455cc887f3e563eb_11" class="yaqOZd"> Acá esta la calculadora creada en las clases de IoT del 20/08/24 y del 22/08/24. </section><section id="h.455cc887f3e563eb_3" class="yaqOZd cJgDec nyKByd" style=""><div class="IFuOkc" style="background-position: center center; background-image: url(https://lh3.googleusercontent.com/W6qRAX0Yhit-OkwKnDb4EXPb0iFilRR6WNsvjNgVNb8dVNnHWA92hWyCfvNJB9OAH3OcAlFCWOCFpNl3wmKdm-g=w16383); background-size: cover;"> <iframe jsname="L5Fo6c" class="YMEQtf" sandbox="allow-scripts allow-popups allow-forms allow-same-origin allow-popups-to-escape-sandbox allow-downloads allow-modals allow-storage-access-by-user-activation" frameborder="0" aria-label="Apps Script" src="https://script.google.com/macros/s/AKfycbzvIuqwYaAsGzTY2J5j-7iLPeU-EXeUGZHDmmpBZZkrmDwbwiMtqHpt_HRwhIQMVCRb/exec" allowfullscreen></iframe> </section> <footer jsname="yePe5c"></footer> <div role="button" class="U26fgb JRtysb WzwrXb I12f0b K2mXPb zXBiaf ynRLnc" jscontroller="iSvg6e" jsaction="click:cOuCgd; mousedown:UX7yZ; mouseup:lbsD7e; mouseenter:tfO1Yc; mouseleave:JywGue; focus:AHmuwe; blur:O22p3e; contextmenu:mg9Pef;touchstart:p6p2H; touchmove:FwuNnf; touchend:yfqBxc(preventDefault=true); touchcancel:JMtRjd;;keydown:I481le;" jsshadow jsname="Bg3gkf" aria-label="Site actions" aria-disabled="false" tabindex="0" aria-haspopup="true" aria-expanded="false" data-menu-corner="bottom-start" data-anchor-corner="top-start"> <span jsslot class="MhXXcc oJeWuf"><svg width="24" height="24" viewBox="0 0 24 24" focusable="false" class=" NMm5M"><path d="M11 17h2v-6h-2v6zm1-15C6.48 2 2 6.48 2 12s4.48 10 10 10 10-4.48 10-10S17.52 2 12 2zm0 18c-4.41 0-8-3.59-8-8s3.59-8 8-8 8 3.59 8 8-3.59 8-8 8zM11 9h2V7h-2v2z"/></svg> <div class="JPdR6b hVNH5c" jscontroller="uY3Nvd" jsaction="IpSVtb:TvD9Pc;fEN2Ze:xzS4ub;frq95c:LNeFm;cFpp9e:J9oOtd; click:H8nU8b; mouseup:H8nU8b; keydown:I481le; keypress:Kr2w4b; blur:O22p3e; focus:H8nU8b" role="menu" tabindex="0" style="position:fixed"> <span jsslot class="z80M1 FeRvI" jsaction="click:o6ZaF(preventDefault=true); mousedown:lAhnzb; mouseup:Osgxgf; mouseenter:SKyDAe; mouseleave:xq3APb;touchstart:jJiBRc; touchmove:kZeBdd; touchend:VfAz8" jsname="j7LFlb" aria-label="Google Sites" role="menuitem" tabindex="-1"> Google Sites <span jsslot class="z80M1 FeRvI" jsaction="click:o6ZaF(preventDefault=true); mousedown:lAhnzb; mouseup:Osgxgf; mouseenter:SKyDAe; mouseleave:xq3APb;touchstart:jJiBRc; touchmove:kZeBdd; touchend:VfAz8" jsname="j7LFlb" data-disabled-tooltip="Report abuse is not available in preview mode" aria-label="Report abuse" role="menuitem" tabindex="-1"> <div class="uyYuVb oJeWuf" jscontroller="HYv29e" jsaction="JIbuQc:dQ6O0c;"

jsname="lV5oke"
data-abuse-proto="%.@.null,null,"https://sites.google.com/view/dagd-page/p%C3%A1gina-principal"]"
data-abuse-reporting-widget-proto="%.@.null,"https://sites.google.com/view/dagd-page/p%C3%A1gina-principal"]"> Report abuse <span jsslot class="z80M1 FeRvI"
jsaction="click:o6ZaF(preventDefault=true); mousedown:lAhnzb; mouseup:Osgxgf; mouseenter:SKyDAe; mouseleave:xq3APb;touchstart:jJiBRc; touchmove:kZeBdd; touchend:VfAz8" jsname="j7LFlb" aria-label="Page details" role="menuitem" tabindex="-1"> Page details <div jscontroller="j1RDQb" jsaction="focusin:gBxDVb(srlkmf); focusout:zvXhGb(srlkmf); click:ro2KTd(psdQ5e);JIbuQc:DSypkd(Bg3gkf);MxH79b:JdcaS;rcuQ6b:rcuQ6b;" class="LqzjUe ynRLnc" data-last-updated-at-time="1724643057203" data-is-preview="false"> <div role="button" class="U26fgb mUbCce fKz7Od kpPxtd QMuaBc M9Bg4d" jscontroller="VXdfxd" jsaction="click:cOuCgd; mousedown:UX7yZ; mouseup:lbsD7e; mouseenter:tfO1Yc; mouseleave:JywGue; focus:AHmuwe; blur:O22p3e; contextmenu:mg9Pef;touchstart:p6p2H; touchmove:FwuNnf; touchend:yfqBxc(preventDefault=true); touchcancel:JMtRjd;" jsshadow jsname="Bg3gkf" aria-label="Site actions" aria-disabled="false" tabindex="-1" aria-hidden="true"> <span jsslot class="xjKiLb"><svg width="24" height="24" viewBox="0 0 24 24" focusable="false" class=" NMm5M"><path d="M11 17h2v-6h-2v6zm1-15C6.48 2 2 6.48 2 12s4.48 10 10 10 10-4.48 10-10S17.52 2 12 2zm0 18c-4.41 0-8-3.59-8-8s3.59-8 8-8 8 3.59 8 8-3.59 8-8 8zM11 9h2V7h-2v2z"/></svg> Page updated <div role="button" class="U26fgb kpPxtd J7BuEb" jsshadow jsname="V2zOu" aria-disabled="false" tabindex="0">Google Sites <div class="YkaBSd" jscontroller="HYv29e" jsaction="click:dQ6O0c;" data-abuse-proto="%.@.null,null,"https://sites.google.com/view/dagd-page/p%C3%A1gina-principal"]"
data-abuse-reporting-widget-proto="%.@.null,"https://sites.google.com/view/dagd-page/p%C3%A1gina-principal"]"><div role="button" class="U26fgb kpPxtd J7BuEb" jsshadow aria-label="Report abuse" aria-disabled="false" tabindex="0">Report abuse <script nonce="OX3mXknW0CRjHNn0EJMNjQ">DOCS_timing['cov']=new Date().getTime();</script><script src="https://www.gstatic.com/_/atari/_/js/k=atari.vw.en_US.9AVl5f74Px0.O/am=WMEABA/d=1/rs=AGEqA5mtYF2mB47pP6J7FYDdX4SHsG-0MA/m=view" id="base-js" nonce="OX3mXknW0CRjHNn0EJMNjQ"></script> </body></html>

Supongamos que te han dado la tarea de escribir el código en python para un navegador, que ya tienes el código de la interfaz gráfica y se llama mediante la función interfazgrafica(parametro), cuál de los parámetros calculados deberías entregarle para que esa función no reciba información que no necesita?

Siguiendo con el ejemplo de la calculadora:

Si la función `interfazgrafica(parametro)` sólo necesita recibir el resultado de la operación para mostrarlo en la interfaz gráfica del navegador, se debería entregar **únicamente el parámetro `resultado`**. De esta manera, la función no recibe información adicional que no es necesaria para su propósito.

- **`resultado`**: Este es el valor calculado por la función `calculadora` y es lo que la interfaz gráfica probablemente necesita mostrar.
- **`mensajeSecreto`**: Es un dato adicional que no parece ser relevante para la interfaz gráfica, por lo tanto, no es necesario enviarlo.

```
import requests
# URL del Web Service
url =
"https://script.google.com/macros/s/AKfycby4LZg2tNzxqCQs0VfRL6SLLHS3fGpBRTLEAVD-fY9k4SiwpWkV69dplItPxCJD6g/exec"
# Parámetros de la query string
params = {
    "operacion": "suma",
```

```
    "a": 4,
    "b": 2
}
# Realizar la solicitud GET al Web Service
response = requests.get(url, params=params)
# Verificar si la solicitud fue exitosa
if response.status_code == 200:
    # Obtener el resultado del cálculo
    resultado = response.json().get("resultado")
    # Llamar a la función de la interfaz gráfica con el resultado
    interfazgrafica(resultado)
else:
    print(f"Error en la solicitud: {response.status_code}")
```

En este ejemplo, solo se pasa el `resultado` a la función `interfazgrafica()`, asegurando que no se envíen datos innecesarios como el `mensajeSecreto`.

Cuando tu visitas una nueva página escribiendo una URL en un navegador, ¿qué método de llamado web usas (GET, POST, PUT, DELETE)?

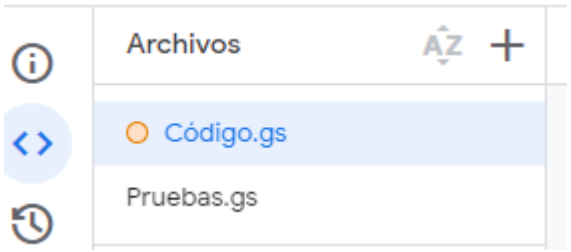
Se usa el método GET.

Tarea 2. Crear un web service básico para comprender las solicitudes POST

Paso 1. Crear un proyecto desde cero para el emprendimiento

En GAS crea un nuevo proyecto, dale un nombre y agrega los siguientes archivos

- Código.gs: ya está por defecto, no debes agregarlo, lo usaremos solo para código limpio, es decir, evitaremos escribir allí código de prueba
- Pruebas: allí escribiremos funciones temporales, que pueden cambiar para estar invocando pruebas al código. En otras palabras, desde aquí le haremos pruebas al código del futuro web service antes de implementarlo. Lo que, una vez finalicen las pruebas, podrá ser considerado como basura para el código limpio



Paso 2. Creamos el script para un web service que sirva para comprender el viaje de los datos.

El problema a resolver con este sencillo web service consiste en lo siguiente:

Cuando se usa Python o cualquier otro lenguaje o aplicación para enviar una carga útil a un webservice, mediante una solicitud POST, lo que llega al punto final es una carga bruta compuesta por la carga útil y datos extra. Necesitamos comprender la carga bruta con el fin de poder identificar allí a la carga útil.

Dicho en otras palabras: necesitamos crear nuestro propio sitio web servicio, que pueda ser llamado mediante el método POST. Una vez puesto en funcionamiento, podrá ser llamado desde lugares remotos que envían carga útil, pero gracias a las funciones que incluiremos en el servicio web podremos comparar lo que llega (la carga bruta) con respecto a la carga útil

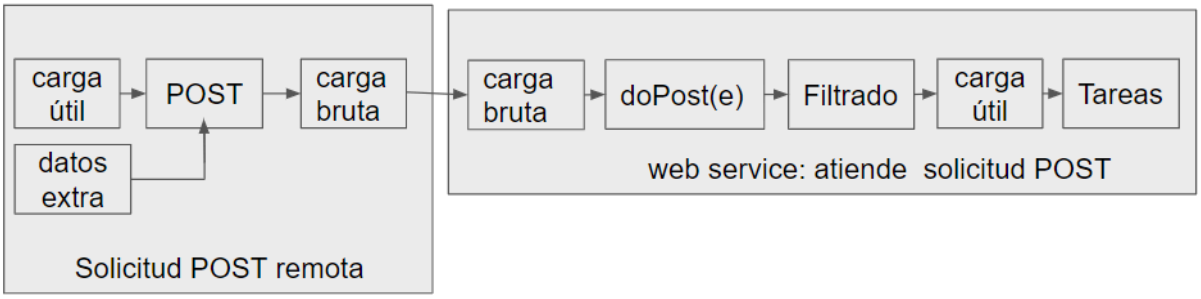


Fig. 2

En el Archivo Código debes crear la función doPost(e). Hazlo usando el siguiente ejemplo:
Nota: observa en la Fig 2 donde se ubica la función doPost(e) en este web service

```
function doPost(cargaBruta) {

  // Preparación de un mensaje de texto para la realización de pruebas
  // La cargaBruta viaja por la red como texto,en formato JSON, pero el método doPost()
  // en javascript la interpreta como un objeto de datos. Por eso, para imprimirla, la
  // convertimos a JSON con JSON.stringify()
  var cargaBrutaJSON = JSON.stringify(cargaBruta);
  var mensajeDeControl = "He recibido la Carga bruta: "+ cargaBrutaJSON;

  // Imprimimos mensajeDeControl ( cuando esto sea un web service el mensaje aparecerá
  // en la consola del cliente)
  console.log(mensajeDeControl);


// Enviamos al cliente remoto una respuesta informando qué nos ha llegado la carga
  return ContentService.createTextOutput("Gracias. Hemos atendido su POST. Carga bruta
recibida: "+cargaBrutaJSON);
}
```

En el Archivo Pruebas.gs creamos una función de prueba() para comprobar si hay errores en el funcionamiento del código. Eso lo hacemos porque será mucho más difícil o imposible encontrar errores cuando el web service entre en funcionamiento.

```
// la siguiente función es para simular datos que pudiesen ser enviados desde invocaciones
// remotas cuando el web service se ponga en funcionamiento
function simuladorDatosNevera() {
  var datos = {
    nevera: "neveraChachones",
    producto: "huevos",
    cantidad: 10,
    pesos: 30000
  };
  return datos;
}

// La siguiente función simula lo que ocurre cuando alguna aplicación remota invoca a
// nuestro web service
function prueba(){
```

```
  doPost(simuladorDatosNevera());
}
```

Corremos la función prueba(). Comprobamos que no hayan errores y todo funciona desde el script

Modifica los datos del simulador para que sean más propios para tu grupo de trabajo. Escribe abajo como es tu simulador y el resultado de correr la función prueba()

Como es tu simulador de datos:

```
function simuladorDatosNevera() {
  var datos = {
    nevera: "neveraDAGD",
    producto: "huevos",
    cantidad: 10,
    pesos: 30000,
    producto2: "queso",
    cantidad2: 3,
    pesos2: 20000,
    producto3: "carne",
    cantidad3: 1,
    pesos3: 12000
  };

  return datos;
}
```

Cual es el resultado de la función prueba():

Paso 3. El reto ahora es convertir el código anterior en un web service y obtener su URL

El proceso a seguir es: Menú superior horizontal > Implementar > Nueva Implementación > Seleccionar tipo > Aplicación web > Descripción > le das un nombre > Ejecutar como yo > Quien tiene acceso > Cualquier usuario > Implementar > copia la URL de la aplicación web

Escribe aquí la URL de tu microservicio (web service):

https://script.google.com/macros/s/AKfycbwvERuvMjgv1yZrQN_fmp9TGPOe0llb5pkq2Qu5216gwg2wyZnFWTcB7x-qzgOpa4so/exec

Paso 4. Probar el uso del web service. Eso implica crear un cliente que consuma el web service de manera remota. El cliente puede ser un código en python que envía una solicitud POST. Para comprobar que las cosas funcionan, el cliente recibirá de vuelta el contenido en

una variable `respuestaAlCliente`, lo cual no es otra cosa que la carga bruta. Entonces, podremos comprender la diferencia entre la carga útil y la carga bruta

Código en Python para invocar el web service con una solicitud tipo POST

```
import requests
url= 'la url de tu web service aquí'

# Ojo: En python lo que se prepara es un diccionario. Ya muy internamente, el sistema de envío lo
# convertirá en texto, es decir en JSON pero eso no lo veremos por comando.
cargaUtil={'nombre':'pepito','edad':'25'}
respuesta = requests.post(url, data=cargaUtil)
print(respuesta.text)
```

Cambia la cargaUtil para que sea diferente al ejemplo. Escribe aquí la respuesta obtenida

```
# Creacion del cliente para el apartado del Web Service:

import requests
url=
"https://script.google.com/macros/s/AKfycbysbVZK3KHtGvoM4a1H9UJ1FtldYeMQzC7rBETi57WFhyWfZdUUWZGvzxxvEolLqfIG/exec"

# Ojo: En python lo que se prepara es un diccionario. Ya muy internamente, el sistema de envío lo
# convertirá en texto, es decir en JSON pero eso no lo veremos por comando.
cargaUtil={'nevera':'neveraDAGD','producto':'carne','cantidad':'2','pesos':'20000'}
respuesta = requests.post(url, data=cargaUtil)
print(respuesta.text)
print("\n")
print("La URL usada en el POST es: ", respuesta.request.url)
```

Agrega el siguiente comando para conocer cuál es la URL que Python realmente ha enviado para invocar el web service:

```
print("La URL usada en el POST es: ", respuesta.request.url)
```

Escribe el resultado aquí:

La URL usada en el POST es:
https://script.googleusercontent.com/macros/echo?user_content_key=Q8nMEi9cuwwEkcBQmBTzE02a-gzi-73ONAcoO42e0ND69jBdXtPBQc5phKDnzjLOtq9f_S8HEWS5FfAd0OTOTDQLFzhzq-Qem5_BxDlH2jW0nuo2oDemN9CCS2h10ox_1xSncGQajx_ryfhECjZEnOOHFQBB7j2yaiYJMuCuPGPZi_-0m7q1Ajf9uWPBG0qXNjaPV6IuQCqfTB-CjcGQxLW9B27l7s8lRrP28O2J4McitqJ_28Cxuw&lib=MBDAM1cZAzCsDn6Q6OD41B0ulueVC4bK2

Pregúntale a Chat GPT a qué se puede deber que la URL usada no es la misma que le dimos con el método POST:

No es la misma URL debido a que la URL pública de Google Apps Script es la **puerta de entrada** pública a la aplicación web. Los usuarios o sistemas externos pueden hacer

solicitudes a esta URL para interactuar con el script. Se pueden manejar solicitudes GET, POST, etc., dependiendo de cómo se haya configurado el script.
Cuando se envía una solicitud a la URL pública, lo que pasa internamente es que Google redirige esa solicitud a una URL interna en sus servidores, esta URL suele estar en el dominio 'script.googleusercontent.com'. Este dominio pertenece a Google y es utilizado para manejar de manera segura la ejecución de los scripts.

Tarea 3. Mejorar el web service para que pueda identificar la carga útil

Llegó la hora de pensar en procesamiento en la nube para filtrar la carga bruta y obtener la carga útil. En realidad esto equivale a mejorar el código del web service (servidor).

Paso 1. Deducción del filtro a introducir en el web service para detectar la carga útil

Análisis desde el punto de vista de Python. Corre nuevamente el código Python de la tarea anterior para responder las siguientes preguntas.

¿Cuál es la carga útil? La enviada por el código en Python:
Para mi ejemplo es:
{'nombre':'pepito','edad':'25'}
¿Cual es para tu caso?

{'nevera':'neveraDAGD','producto':'carne','cantidad':'2','pesos':'20000'}

¿Cuál es la carga bruta?. Se refiere a la carga que llega al web service, pero nuestra aplicación en python la conoce ya que hemos programado al web service para que la devuelva. Como cuando una persona le envía un paquete a su amigo y luego le pregunta - hey dime qué es lo que te llegó, pero el amigo le contesta - pues me llegó tu regalito pero también una publicidad de la empresa de correos. Es decir ambas parte conocen que llegó al punto final:
para mi ejemplo lo que llega es:
{"contextPath":"","parameter":{"nombre":"pepito","edad":"25"},"parameters":{"edad":["25"],"nombre":["pepito"]},"postData":{"contents":"nombre=pepito&edad=25","length":21,"name":"postData","type":"application/x-www-form-urlencoded"},"contentLength":21,"queryString":""}
¿Cual es para tu caso?


```
{
  "contextPath": "",
  "parameter": {"nevera": "neveraDAGD", "producto": "carne", "cantidad":
"2", "pesos": "20000" },
  "parameters": {"nevera": ["neveraDAGD"], "producto": ["carne"],
"cantidad": ["2"], "pesos": ["20000"]},
  "postData": { "contents":
"nevera=neveraDAGD&producto=carne&cantidad=2&pesos=20000", "length": 49,
    "name": "postData", "type": "application/x-www-form-urlencoded" },
  "contentLength": 49,
  "queryString": ""
```

```
}
```

Responda a las preguntas: ¿es la carga un diccionario o un JSON?. ¿Es posible ver que la carga bruta contiene a la útil?.

1. La carga útil (`cargaUtil`) es un **diccionario** en Python. En Python, los diccionarios son estructuras de datos que almacenan pares clave-valor, y no son lo mismo que JSON, aunque se pueden convertir fácilmente. Cuando se utiliza `requests.post(url, data=cargaUtil)`, el diccionario se convierte automáticamente en una cadena en formato **URL-encoded** (por ejemplo: `nevera=neveraDAGD&producto=carne&cantidad=2&pesos=20000`). Este es el formato utilizado para enviar datos en solicitudes HTTP tipo POST cuando se utiliza el parámetro `data`. Si se quiere enviar la carga como **JSON**, tendrías que usar el parámetro `json` en lugar de `data`, como: `requests.post(url, json=cargaUtil).` Esto enviaría la carga útil como un objeto JSON en el cuerpo de la solicitud, con el tipo de contenido application/json.
2. **Con `data=cargaUtil`**: La carga útil se convierte en una cadena de texto en formato **URL-encoded**. Para ver esta carga bruta, puedes inspeccionar `respuesta.request.body`, que mostrará una cadena con los datos codificados de la siguiente manera:
   ```
   nevera=neveraDAGD&producto=carne&cantidad=2&pesos=20000
   ```

Con `json=cargaUtil`: La carga útil se convertiría en una cadena JSON. Al inspeccionar `respuesta.request.body`, verías la carga útil en formato JSON:

```
{"nevera":"neveraDAGD","producto":"carne","cantidad":2,"pesos":20000}
```

Análisis desde el punto de vista de JavaScript en Google Apps Script.
Recordemos que un diccionario en Python equivale a un objeto de datos en JavaScript y que JSON es la versión en texto de ese objeto o diccionario. JSON no depende de un lenguaje de programación. En JavaScript la carga bruta y la carga útil son objetos de datos

¿Cuál es la carga útil? El equivalente en JavaScript al diccionario Python para la carga útil:

Nota 1: javascript tiene dos formas de escribir un objeto de datos (diccionario): la primera es exactamente como se hace con un diccionario en python, donde la clave va entre “”. El otro método es que la clave no lleva “”
Nota 2: para leer un valor de una clave en javascript hay igualmente dos métodos: el primero es exactamente como se hace en python con los diccionarios. El segundo es usando nombreObjeto.clave.

Por ejemplo, para mi caso la carga útil es:

```
{
nombre:'pepito',
edad:'25'}
}
```

¿Cual es para tu caso?

```
{
```

```
nevera:'neveraDAGD',
producto:'carne',
cantidad:2,
pesos:20000
}
```

¿Cuál es la carga bruta? El equivalente en JavaScript al diccionario Python para la carga bruta:
Para mi caso es:

```
    {
    contextPath: "",
    parameter: { nombre: "pepito", edad: "25" },
    parameters: { edad: ["25"], nombre: ["pepito"] },
    postData: { contents: "nombre=pepito&edad=25", length: 21,
                name: "postData", type: "application/x-www-form-urlencoded" },
    contentLength: 21,
    queryString: ""
    }
```

¿Cual es para tu caso?

```
{
  contextPath: "",
  parameter: { nevera: "neveraDAGD", producto: "carne", cantidad:
"2", pesos: "20000" },
  parameters: {
    nevera: ["neveraDAGD"],
    producto: ["carne"],
    cantidad: ["2"],
    pesos: ["20000"]
  },
  postData: {
    contents:
"nevera=neveraDAGD&producto=carne&cantidad=2&pesos=20000",
    length: 49,
    name: "postData",
    type: "application/x-www-form-urlencoded"
  },
  contentLength: 49,
  queryString: ""
}
```

Escriba el código en JavaScript que permite, en el script del web service, hacer el filtrado para obtener la carga útil a partir de la carga Bruta.
Solución: En la respuesta anterior, vemos claramente que la carga útil está en la clave “parameter”. Gracias a la magia de los objetos de datos que provienen de un JSON, el filtro es tan sencillo como el siguiente comando:

```
var cargaUtil = cargaBruta.parameter
```

Escribe el código completo para tu función doPost() considerando la carga útil. El siguiente es un ejemplo:

```
/**
 * Recibe una carga bruta proveniente de un cliente y le devuelve un mensaje para notificar
 * qué es lo que ha recibido y qué carga útil ha deducido.
```

```
 */
function doPost(cargaBruta) {

  // El filtrado para deducir la carga útil
  var cargaUtil = cargaBruta.parameter;

  // Preparación de un mensaje de texto para la realización de pruebas
  var cargaBrutaJSON = JSON.stringify(cargaBruta);
  var cargaUtilJSON = JSON.stringify(cargaUtil);
  var mensajeDeControl = "He recibido la Carga bruta: "
    + cargaBrutaJSON + ".\n\nHe deducido que la carga util es: " + cargaUtilJSON

  // Envío de mensaje a la consola y al cliente
  console.log(mensajeDeControl);
  return ContentService.createTextOutput(mensajeDeControl);
}
```

Realizar las pruebas a nivel del script. Nota: convertir este script a web service y comprobar si funciona al invocarlo desde un cliente parece ser lo más obvio. Pero si algo falla, si hemos escrito algún error, este método va a ser engorroso. Lo mejor es asegurar que no hay fallas en el código antes de convertirlos a web service. Vamos al archivo Pruebas y simulemos allí la carga bruta, luego la tomamos en en cuenta en la función pruebas() como se muestra aquí:

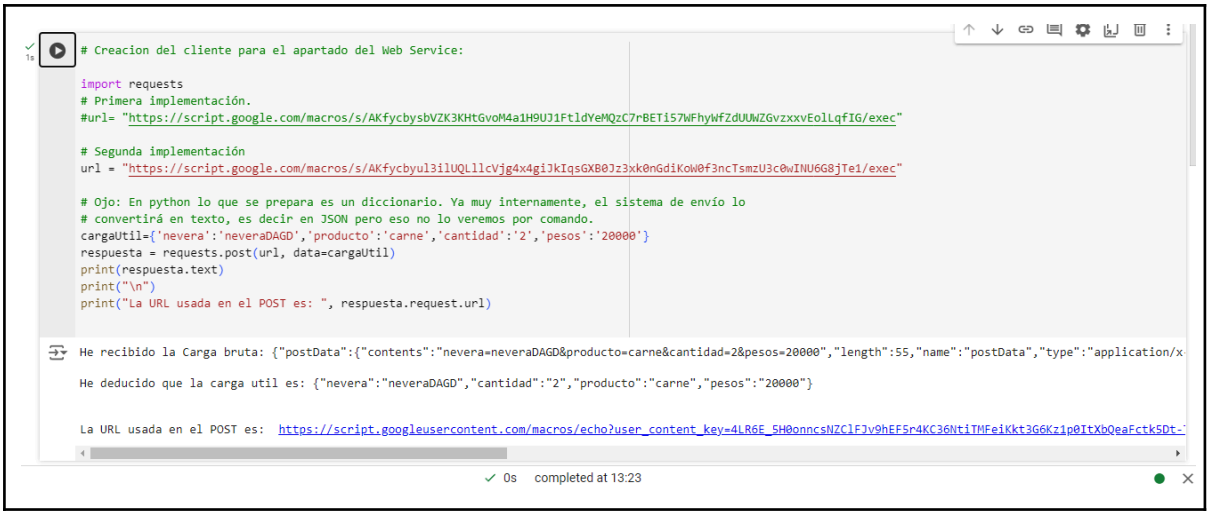
```
function simuladorCargaBruta(){
    return  {
    contextPath: "",
    parameter: { nombre: "pepito", edad: "25" },
    parameters: { edad: ["25"], nombre: ["pepito"] },
    postData: { contents: "nombre=pepito&edad=25", length: 21,
      name: "postData", type: "application/x-www-form-urlencoded" },
    contentLength: 21,
    queryString: ""
    }
}
function prueba(){
  doPost(simuladorCargaBruta());
}
```

Corra la función prueba() y analiza si todo funciona correctamente. Escribe el resultado aquí:

Paso 2. Concluir el web service y realizar pruebas finales.

En el paso anterior logramos identificar no solo la carga bruta, sino también la carga útil, pero necesitamos que eso se vea reflejado como web service.

Ahora, concluir el web service, es tan sencillo como volver a implementarlo. Para ello ten en cuenta que: ● En GAS una nueva implementación significa la creación de un nuevo web service sin que el anteriormente creado deje de existir ● Por lo anterior, obtendrás una nueva URL para tu web service ● Puedes ver las diferentes implementaciones que tengas para tu web service así: En el IDE de GAS > Menú horizontal > Implementar > Gestionar Implementaciones.
Escribe a continuación la URL para el nuevo web service:
https://script.google.com/macros/s/AKfycbyul3ilUQLllcVjg4x4qiJkIqsGXB0Jz3xk0nGdiKoW0f3ncTsmzU3c0wINU6G8jTe1/exec
Corrige el código de tu cliente en Python con la nueva URL. Córrelo para comprobar que todo funciona a las mil maravillas. Escribe abajo el resultado



```python
# Creacion del cliente para el apartado del Web Service:

import requests
# Primera implementación.
#url= "https://script.google.com/macros/s/AKfycbysbVZK3KHtGvoM4a1H9UJ1FtldYeMQzC7rBETi57WFhyWfZdUUWZGvzxxvEolLqfIG/exec"

# Segunda implementación
url = "https://script.google.com/macros/s/AKfycbyul3ilUQLllcVjg4x4giJkIqsGXB0Jz3xk0nGdiKoW0f3ncTsmzU3c0wINU6G8jTe1/exec"

# Ojo: En python lo que se prepara es un diccionario. Ya muy internamente, el sistema de envío lo
# convertirá en texto, es decir en JSON pero eso no lo veremos por comando.
cargaUtil={'nevera':'neveraDAGD','producto':'carne','cantidad':'2','pesos':'20000'}
respuesta = requests.post(url, data=cargaUtil)
print(respuesta.text)
print("\n")
print("La URL usada en el POST es: ", respuesta.request.url)
```

```
He recibido la Carga bruta: {"postData":{"contents":"nevera=neveraDAGD&producto=carne&cantidad=2&pesos=20000","length":55,"name":"postData","type":"application/x

He deducido que la carga util es: {"nevera":"neveraDAGD","cantidad":"2","producto":"carne","pesos":"20000"}


La URL usada en el POST es:  https://script.googleusercontent.com/macros/echo?user_content_key=4LR6E_5H0onncsNZClFJv9hEF5r4KC36NtiTMFeiKkt3G6Kz1p0ItXbQeaFctk5Dt-
```

✓ 0s completed at 13:23