

VICERRECTORADO DOCENTE

CONSEJO ACADÉMICO

 $\textbf{C\'odigo:} \ \text{GUIA-PRL-001}$

Aprobación: 2016/04/06

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

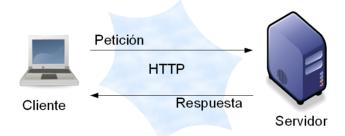


PRÁCTICA DE LABORATORIO

PRÁCTICA: 1 TÍTULO PRÁCTICA: Consumo de APIs web (plataformas en la nube) PBJETIVOS Conocer las arquitecturas y patrones arquitectónicos web para el diseño de aplicaciones web Interactuar con servicios web de plataformas en la nube Desarrollar una aplicación web usando una de las API gratuitas de API List Fun. Tener en cuenta que se deben aplicar buenas prácticas para el desarrollo de la interfaz gráfica de usuario, para la cuál se permite utilizar plantilla de Bootstrap. NSTRUCCIONES Requisitos: La aplicación Web debe permitir buscar la información a través de un nombre. ACTIVIDADES POR DESARROLLAR 1. Identificar gráficamente la arquitectura web de la aplicación a desarrollar. Cienerar una llave para consumir los servicios web de la API (opcional, depende de la API seleccionada). Crear un repositorio en GitHub con el nombre "Practica01 – Consumo de APIs en la nube" 4. Desarrollar una aplicación con HTML + CSS + Javascript + Web Services para buscar información y visualizar toda la información disponible a través de la API. S. Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. G. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica S. Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones apropiadas y la firma de cada estudiante	CARRERA: ING. DE SISTEMAS			ASIGNATURA: APLICACIONES DISRTIBUIDA	
Conocer las arquitecturas y patrones arquitectónicos web para el diseño de aplicaciones web Interactuar con servicios web de plataformas en la nube Desarrollar una aplicación web usando una de las API gratuitas de API List Fun. Tener en cuenta que se deben aplicar buenas prácticas para el desarrollo de la interfaz gráfica de usuario, para la cuál se permite utilizar plantilla de Bootstrap. NSTRUCCIONES Requisitos: La aplicación Web debe permitir buscar la información a través de un nombre. Además, se deberá visualizar toda la información disponible de la base de datos. ACTIVIDADES POR DESARROLLAR I. Identificar gráficamente la arquitectura web de la aPI (opcional, depende de la API seleccionada). Crear un repositorio en GitHub con el nombre "Practica01 – Consumo de APIs en la nube" Desarrollar una aplicación con HTML + CSS + Javascript + Web Services para buscar información y visualizar toda la información disponible a través de la API. Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones	NRO. PRÁCTICA:	1	de APIs web (plataformas en la nube)		
API List Fun. Tener en cuenta que se deben aplicar buenas prácticas para el desarrollo de la interfaz gráfica de usuario, para la cuál se permite utilizar plantilla de Bootstrap. NSTRUCCIONES Requisitos: La aplicación Web debe permitir buscar la información a través de un nombre. Además, se deberá visualizar toda la información disponible de la base de datos. ACTIVIDADES POR DESARROLLAR I Identificar gráficamente la arquitectura web de la aplicación a desarrollar. Generar una llave para consumir los servicios web de la API (opcional, depende de la API seleccionada). Crear un repositorio en GitHub con el nombre "Practica01 – Consumo de APIs en la nube" Desarrollar una aplicación con HTML + CSS + Javascript + Web Services para buscar información y visualizar toda la información disponible a través de la API. Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones	• Conocer las	-			
ACTIVIDADES POR DESARROLLAR 1. Identificar gráficamente la arquitectura web de la aplicación a desarrollar. 2. Generar una llave para consumir los servicios web de la API (opcional, depende de la API seleccionada). 3. Crear un repositorio en GitHub con el nombre "Practica01 – Consumo de APIs en la nube" 4. Desarrollar una aplicación con HTML + CSS + Javascript + Web Services para buscar información y visualizar toda la información disponible a través de la API. 5. Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. 6. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. 7. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica 8. Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones	INSTRUCCIONES		API List Fun. Tener e para el desarrollo de l permite utilizar planti Requisitos: La aplicac través de u	en cuenta que se deben aplicar buenas prácticas a interfaz gráfica de usuario, para la cuál se lla de Bootstrap. ión Web debe permitir buscar la información a un nombre.	
 Identificar gráficamente la arquitectura web de la aplicación a desarrollar. Generar una llave para consumir los servicios web de la API (opcional, depende de la API seleccionada). Crear un repositorio en GitHub con el nombre "Practica01 – Consumo de APIs en la nube" Desarrollar una aplicación con HTML + CSS + Javascript + Web Services para buscar información y visualizar toda la información disponible a través de la API. Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones 			de la base	de datos.	
 Generar una llave para consumir los servicios web de la API (opcional, depende de la API seleccionada). Crear un repositorio en GitHub con el nombre "Practica01 – Consumo de APIs en la nube" Desarrollar una aplicación con HTML + CSS + Javascript + Web Services para buscar información y visualizar toda la información disponible a través de la API. Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones 					
 Crear un repositorio en GitHub con el nombre "Practica01 – Consumo de APIs en la nube" Desarrollar una aplicación con HTML + CSS + Javascript + Web Services para buscar información y visualizar toda la información disponible a través de la API. Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones 					
 Desarrollar una aplicación con HTML + CSS + Javascript + Web Services para buscar información y visualizar toda la información disponible a través de la API. Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones 				<u> </u>	
visualizar toda la información disponible a través de la API. 5. Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. 6. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. 7. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica 8. Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones					
 Realizar varios commits en la herramienta GitHub que demuestren el desarrollo de la aplicación. Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones 					
 Generar el informe de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente. Implementar el README del repositorio del proyecto con la misma información del informe de la práctica Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones 		<u> </u>			
 Implementar el README del repositorio del proyecto con la misma información del informe de la práctica Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones 				<u>-</u>	
8. Subir al AVAC el informe del proyecto en formato *.pdf. El informe debe contar con conclusiones					
	8. Subir al AV	AC el i	nforme del proyecto en formato *.		

DESARROLLO

1) Identificar gráficamente la arquitectura web de la aplicación a desarrollar.



2) consumir los servicios web

cdnjs.cloudflare.com

cdnjs.cloudflare.com es una red de entrega de contenido ultrarrápida, confiable y disponible globalmente para bibliotecas de código abierto.

Para cargar una biblioteca distribuida, copie y pegue el fragmento HTML de esa biblioteca en su página web. Por ejemplo, para cargar jQuery, inserte el fragmento en su página web:

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js" integrity="sha512-bLT0Qm9VnAYZDflyKcBaQ2gg0hSYNQrJ8RilYldYQ1FxQYoCLtUjuuRuZo+fjqhx/qtq/1itJ0C2ejDxltZVFg=="crossorigin="anonymous"></script>

cdnjs proporciona una API simple para permitir que cualquiera pueda consultar rápidamente las bibliotecas que tenemos en la CDN.

Proporcionamos dos puntos finales principales, el librariespunto final que permite a cualquier persona buscar en el conjunto completo de bibliotecas que ofrecemos y el libraries/:librarypunto final que permite obtener datos detallados para una biblioteca específica.

3) Crear un repositorio en GitHub con el nombre "Practica01 – Consumo de APIs en la nube"

https://github.com/zhimi99/APLICACIONES DISTRIBUIDAS

4) Desarrollar una aplicación con HTML + CSS + Javascript + Web Services para buscar información

```
k rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <title>FETCH</title>
</head>
<body>
  <div class="container">
    <div class="row mt-3">
      <div class="col">
        <h2>Listado de datos</h2>
        <div class="my-3">
          <thead>
             ID
              Nombre
              Email
              Compañía
             </thead>
            </div>
      </div>
    </div>
 </div>
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-</p>
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-</pre>
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-</p>
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  <script src="js/fetch.js"></script>
</body>
</html>
                             JAVASCRIPT
const tabla = document.querySelector('#lista-usuarios tbody');
function cargarUsuarios(){
  fetch('usuarios.json')
```

```
.then(respuesta => respuesta.json()) //Indicamos el formato en que se desea obtener la
información
    .then(usuarios => {
      usuarios.forEach(usuario => {
        const row = document.createElement('tr');
        row.innerHTML += `
           ${usuario.id}
           ${usuario.name}
           ${usuario.email}
           ${usuario.company.name}
        tabla.appendChild(row);
      });
    }) // Aquí mostramos dicha información
    .catch(error => console.log('Hubo un error : ' + error.message))
}
cargarUsuarios();
```

CONCLUSIONES:

• Podemos concluir que gracias a los servicos web, nos facilita el uso de herramientas necesarias para poder desarollar nuestras aplicaciones, Los usuarios pueden escalar los servicios para ajustarlos a sus necesidades, personalizar las aplicaciones y acceder a los servicios de cloud desde cualquier sitio con una conexión a internet todo esto e sposible gracias a los servicios en la nube. Así como también, podrán consumir APIs y manipular objetos JSON.