**1. ¿Qué diferencia existe entre una petición HTTP generada por un agente de usuario de forma asincrónica respecto a una sincrónica? ¿Cómo puede distinguir una aplicación web entre ambas?**

a) En la petición web sincrónica, el agente usuario, habiendo enviado solicitud al servidor por medio de Ajax, se queda esperando la respuesta y hasta tanto no la reciba no permite al usuario interactuar con ciertas partes de una página, una vez recibidos los datos, se puede seguir interactuando con la misma. Asi sucede, en el llenado de un formulario, como por ejemplo, los campos

País: =>

Ciudad: =>

Acá no tiene sentido, que el usuario busque entre todas las ciudad de todo el mundo. De esta forma, solo se puede llenar el campo Ciudad una vez que el campo país es llenado. Se envía una petición en segundo plano al servidor donde se consulta las ciudades pertenecientes a al país introducido y se envía las respuestas al agente usuario.

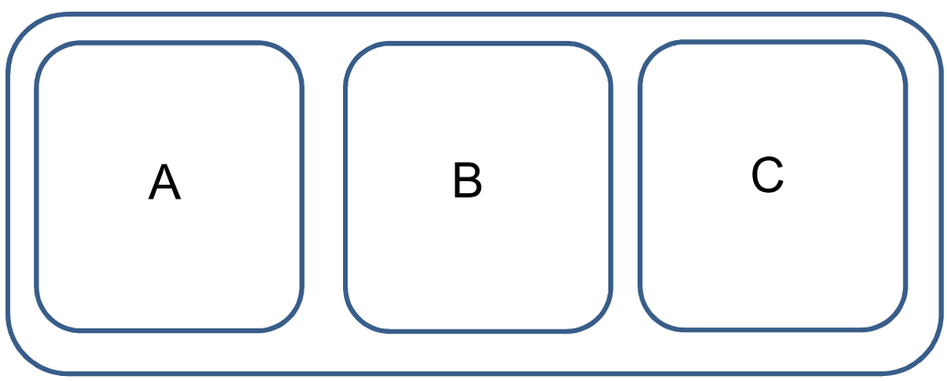
b) En la petición web asincrónica, independientemente de las comunicaciones que el agente usuario realice al servidor u otro servicio dado, el agente usuario puede seguir trabajando, ya sea si recibió la información de respuesta o no.

**2. Que diferencias existen entre el diseño responsivo y el universal? ¿En qué conceptos hay que hacer hincapié al momento de definir las media querys en cada caso?**

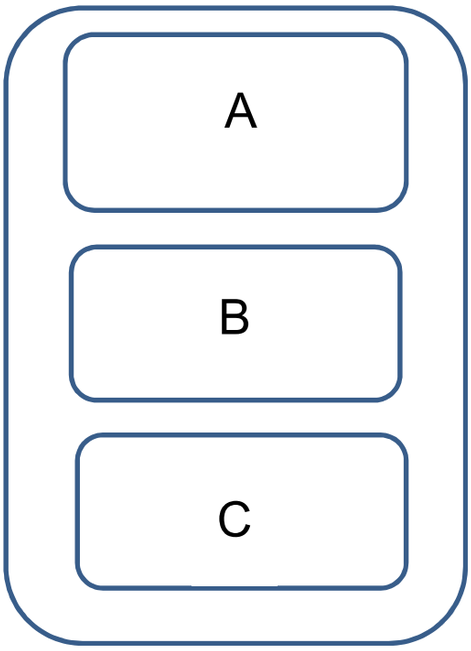
Un diseño responsivo se adapta a los diferentes dispositivos en los que el servicio web es servido, en el diseño universal, no cuenta con esta ventaja y genera dificultades en el manejo e interacción con la página. Dentro de las media query las características a tener en cuenta son:

- el ancho máximo de la página: es decir el max-width en el cual el diseño va a cambiar.
- Los contenedores deben ser flexibles, deben contar con la característica de moverse dependiendo del tamaño de la pantalla. Como asi también, otras características que lo complementan, como el *justify-content* logrando que los contenedores permanezcan a cierta distancia de los demás elementos, y sus vecinos.

  Si la pantalla tiene este ancho **A**, y luego pasa a ser más angosto




Los contenedores en su interior, se moverían según la característica de *wrap* que tengan. (wrap, no-wrap, wrap-reverse)




**3. ¿Por qué decimos que no son directamente comparables REST y SOAP en el contexto de los Web Services?**

Dado el contexto de Servicios Web, ambos son tecnologías orientadas a distintas características, es decir, REST está orientado a recursos, pedís un recurso de un API REST y lo tenés, en cambio SOAP está orientado al servicio, a la actividad, a la comunicación.

**4. Explique brevemente tres principios de desarrollo seguro y de un ejemplo para cada uno.**

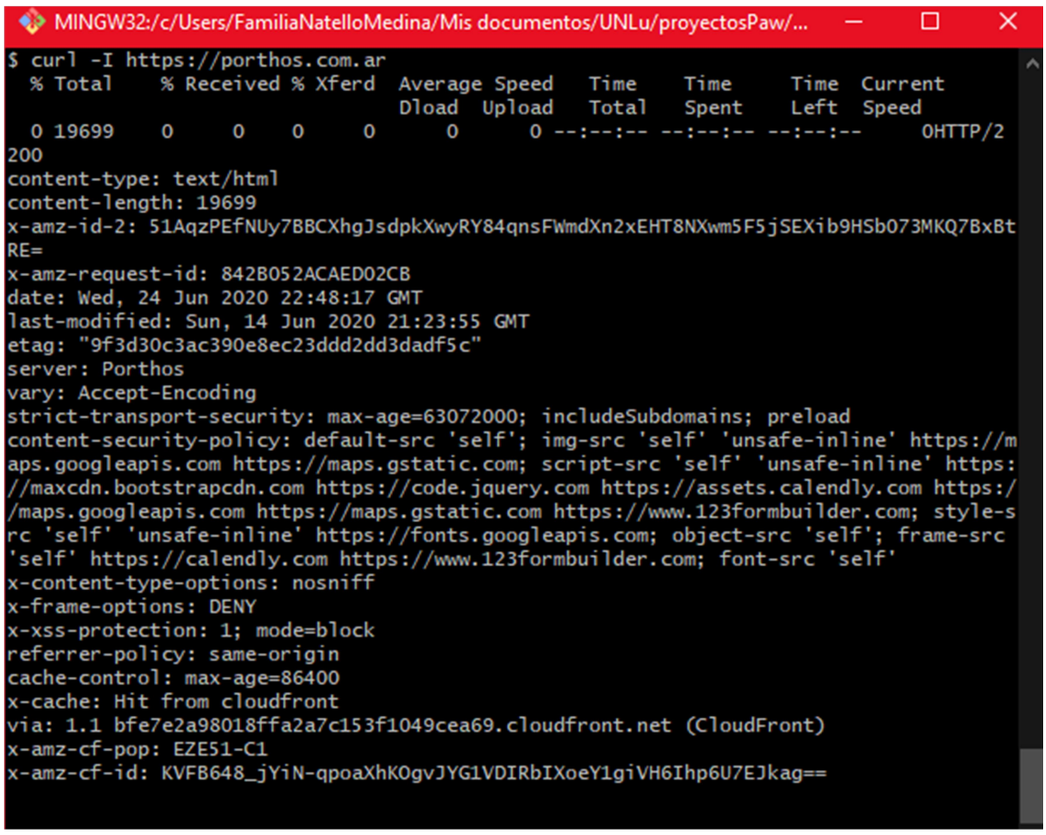
En cuanto a principios básicos de desarrollo seguro, podemos mencionar:

1- **A la Disponibilidad:** la cual busca que un sistema esté disponible *"24/7".* Es decir, si acaba de salir una noticia hace menos de una hora, yo quisiera poder consultarla a cualquier hora, y que la pagina esté disponible.
2- **A la Integridad**: se busca que los datos no puedan ser alterada, o sea que no sea inútil o inconsistente.
3- **A la Confidencialidad**: Trata de proteger el riesgo de revelación de los datos, por ejemplo el número de tarjeta de crédito en el proceso de pago de un servicio. Si vamos a pagar una suscripción de un diario, que en el proceso de pago no se terminan revelando los números de tarjeta y seguro.

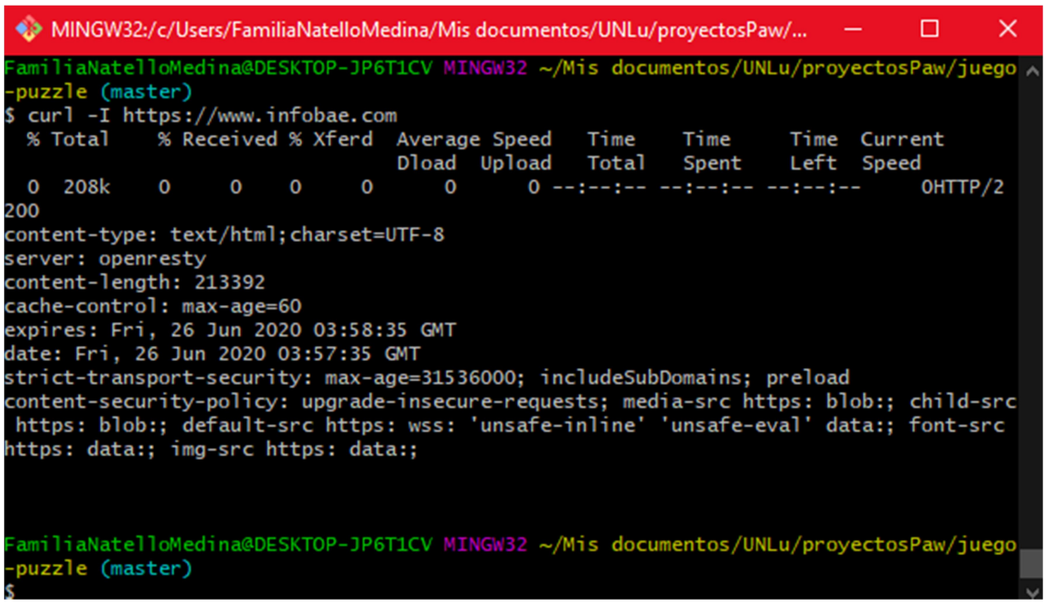
**5. ¿Cómo se relaciona el header HTTP Content-Security-Policy con la seguridad de un sistema web y por qué es fundamental su uso hoy en día? ¿Se puede implementar esto mismo de otra forma que no sea vía header HTTP (a nivel del server web)?**

En este punto estamos hablando de la política de seguridad en cuanto al contenido que sirve la página, es decir, le dice al navegador de donde sacar lo que necesita, de que fuentes son aceptables o no. Puedo hacer que *checke* el lugar de donde trae los contenidos.

A nivel de HTML se puede hacer mediante META`s y a nivel servidor se puede hacer mediante headers.




Por ejemplo, haciendo una consulta a los headers de una *página conocida*, se puede observar, que el Content-security-policy, nos está diciendo de donde sacar los script, de que lugares son aceptables o no.




En este caso, la página de ***https://www.infobae.com***, lo que está diciendo que quiere asegurarse que todos los recursos se carguen únicamente a través de canales seguros.

**6. ¿Por qué es útil un buen análisis de riesgos a la hora de priorizar las mejoras de seguridad que podamos aplicar a nuestro sistema web?**

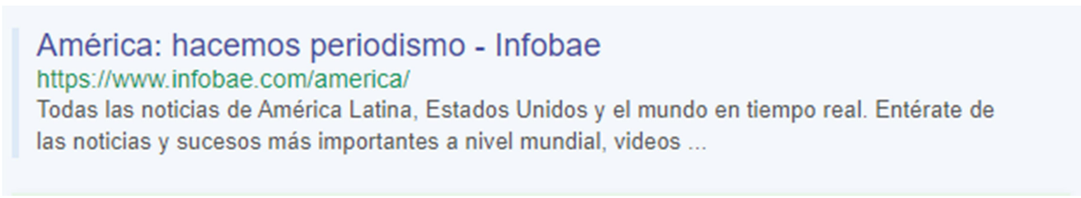
Realizando un buen análisis de riesgos, nos podemos evitar costos innecesarios en el servicio que brinda la página.

Entonces, por ejemplo: dada una página web de venta de motos, que solo usa el servicio web, para promocionar el producto pero que las ventas, las realiza offline, no sirve aumentar los costos a base de aumentar la disponibilidad de la página. Si en todo caso, la llegasen a hackear, se resetea todo el servicio y tratándose de una página estática, en cuestión de poco tiempo vuelve a tener la página andando.

**7. Describa cómo generar una buena estrategia de SEO a partir del uso de herramientas semánticas.**

Dandole un buen uso a las herramientas semánticas, podemos por ejemplo, al momento de usar el TAG de imagen, también usar el TAG ALT= “imagen-selva” el cual, si uno no tiene la imagen o la misma tarda en cargar, ponga un texto alternativo, lo cual es interesante porque ese texto se alcanza a ver para conexiones lentas y para browsers de texto.

Tengo que usar un TITLE  descriptivo del contenido de la página.




Por eso es importante, que este meta tag este incoporado en los METADESCRIPCION

El TAG en META description = es muy importante porque es lo que va a aparecer en los resultados de las busquedas

En el title se pueden agregar palabras clave que estén contenidas dentro de la etiqueta.

El uso de las etiquetas H1 y H2, son también bien punteadas, y serán agregadas a la página.

**8. ¿Cuáles son las ventajas y desventajas del modelo serverless en el cloud respecto al modelo tradicional basado en infraestructura (servers físicos / VMs).**

| Serverless en el cloud | Servers físicos/ VirtualMachines |
|---|---|
| ***Disponibilidad***: El servidor se puede clonar, si se cae, lo puedo tener una copia en otro sitio, geográficamente distinto, que de caerse el principal, garantiza el estar siempre disponible. El famoso ***cero down time*** | ***Disponibilidad***: Se llega a caer, el servidor local, y se cae todo el sistema. Lo cual puede suceder con una falta de suministro eléctrico, etc. |
| Logra abstraer al desarrollador de la infraestructura. | Debo hacer un mantenimiento a la infraestructura y considerar el costo del mantenimiento “*de los fierro”*. |
| Si yo quiero que 1 millón de personas, ejecuten mi código python y el servicio no se caiga, uso una plataforma como servicio y se que no se va a caer nunca, porque de eso se encarga el proveedor. | Debo adaptar mi servidor para lograr una disponibilidad en peticiones del servicio. |
| Es rápidamente escalable | No es rápidamente escalable. |
| Puedo pagar por un servicio que me provea un servicio de base de datos del estilo Oracle, sin tener que pagar licencia por el mismo. De eso se encargar el proveedor y yo solo le pago por lo que me brinda, y puedo configurarlo como quiero. | Tengo que pagar licencia, si deseo que mi servicio web corra sobre una base de datos propietaria. |
| Uso el software como servicio, es decir el SaaS (Software-as-a-Service). | Tengo que instalar el servidor de correo (por ejemplo) y mantenerlo. |

**9. Imagine tiene que implementar un sistema de firma digital: dado un pdf de entrada debe devolverlo firmado digitalmente. Para ello, y dado que debe integrarse a sistemas web existentes, debe diseñar una arquitectura que facilite dicha integración. Comente sobre los componentes de la misma y qué cuestiones contempla, dificultades, etc.**

Puedo crear una API REST a la cual cada vez que se pide el recurso, se loguea el usuario solicitante, *“cuando se conecta” (timestamp)*, *“como se conecto”*, “*que recurso pidió”, “desde donde se conectó”*. Así mismo puedo guardar el LOG en otro servidor a parte (entiéndase en un server cloud), que se dedique exclusivamente al LOG, para evitar guardar los LOG en el servidor local, evitando hackeos en la base de datos del servidor, me borren los LOG´s, Entonces, en el servidor cloud la API REST dedicada al LOG solo se use para agregar y no modificar, ni borrar los LOGs.

Como voy a trabajar con información sensible, voy a buscar que mi servicio trabaje con dos tipos de datos, encriptados en tránsito y encriptados en reposo. De esta forma me aseguro contemplar dos aspectos de la integridad evitando la alteración de los datos.

**10. Suponga que está desarrollando una API que puede ser consumida utilizando diferentes formatos de intercambio de datos ¿De qué forma puede determinar el backend el formato a utilizar para atender un cliente determinado? ¿Cómo debería comportarse el mismo en caso de no conocer el formato solicitado?**

Dependiendo del tipo de datos a intercambiar, me va a determinar el tipo de formato a usar, en el caso de tratarse de un archivo del tipo MIME, elegiré enviar los datos mediante forma XML, mientras que si trabajo con datos puros, me inclinare a trabajar con formatos del tipo JSON. Así mismo, si el cliente es un dispositivo móvil, por una cuestión de agilidad y uso de datos, me inclinare a usar formatos de intercambio JSON.

De no saber el tipo de formato que se solicita, puedo desde el backend, tener una lista de los tipos de móviles, que tendré que mantener actualizada, preguntar sobre los HEADER: $_SERVER[‘HTTP_USER_AGENT], $_SERVER[‘’HTTP_X_WAP_PROFILE] y $_SERVER[‘HTTP_PROFILE’] para ir determinando con qué tipo de usuario estoy lidiando y en base a eso con tipo de formato de intercambio de datos trabajar.

Aquí un código, que me pareció apropiado adjuntar

```php
<?php
$tablet_browser = 0;
$mobile_browser = 0;

if (preg_match('/(tablet|ipad|playbook)|(android(?!.*(mobi|opera mini)))/i', strtolower($_SERVER['HTTP_USER_AGENT']))) {
    $tablet_browser++;
}

if (preg_match('/(up.browser|up.link|mmp|symbian|smartphone|midp|wap|phone|android|iemobile)/i', strtolower($_SERVER['HTTP_USER_AGENT']))) {
    $mobile_browser++;
}

if ((strpos(strtolower($_SERVER['HTTP_ACCEPT']),'application/vnd.wap.xhtml+xml') > 0) or ((isset($_SERVER['HTTP_X_WAP_PROFILE']) or isset($_SERVER['HTTP_PROFILE'])))) {
    $mobile_browser++;
}

$mobile_ua = strtolower(substr($_SERVER['HTTP_USER_AGENT'], 0, 4));
$mobile_agents = array(
    'w3c ','acs-','alav','alca','amoi','audi','avan','benq','bird','blac',
    'blaz','brew','cell','cldc','cmd-','dang','doco','eric','hipt','inno',
    'ipaq','java','jigs','kddi','keji','leno','lg-c','lg-d','lg-g','lge-',
    'maui','maxo','midp','mits','mmef','mobi','mot-','moto','mwbp','nec-',
    'newt','noki','palm','pana','pant','phil','play','port','prox',
    'qwap','sage','sams','sany','sch-','sec-','send','seri','sgh-','shar',
    'sie-','siem','smal','smar','sony','sph-','symb','t-mo','teli','tim-',
    'tosh','tsm-','upg1','upsi','vk-v','voda','wap-','wapa','wapi','wapp',
    'wapr','webc','winw','winw','xda ','xda-');

if (in_array($mobile_ua,$mobile_agents)) {
    $mobile_browser++;
}

if (strpos(strtolower($_SERVER['HTTP_USER_AGENT']),'opera mini') > 0) {
    $mobile_browser++;
    //Check for tablets on opera mini alternative headers
    $stock_ua = strtolower(isset($_SERVER['HTTP_X_OPERAMINI_PHONE_UA'])?$_SERVER['HTTP_X_OPERAMINI_PHONE_UA']:(isset($_SERVER['HTTP_DEVICE_STOCK_UA'])?$_SERVER['HTTP_DEVICE_STOCK_UA']:''));
    if (preg_match('/(tablet|ipad|playbook)|(android(?!.*mobile))/i', $stock_ua)) {
      $tablet_browser++;
    }
}

if ($tablet_browser > 0) {
   // do something for tablet devices
   print 'is tablet';
}
else if ($mobile_browser > 0) {
   // do something for mobile devices
   print 'is mobile';
}
else {
   // do something for everything else
   print 'is desktop';
}

?>
```