

parcial1-resp.pdf



VELARIS



Sistemas Informaticos I



3º Grado en Ingeniería Informática



**Escuela Politécnica Superior
Universidad Autónoma de Madrid**



**Que no te escriban poemas de amor
cuando terminen la carrera**



*(a nosotros por
suerte nos pasa)*

WUOLAH

WUOLAH

Oh Wuolah wuolithah
Tu que eres tan bonita

Nombre:

Python	<input type="checkbox"/> CLIENTE	<input checked="" type="checkbox"/> SERVIDOR	<input type="checkbox"/> AMBOS
SQL	<input type="checkbox"/> CLIENTE	<input checked="" type="checkbox"/> SERVIDOR	<input type="checkbox"/> AMBOS
XML	<input type="checkbox"/> CLIENTE	<input type="checkbox"/> SERVIDOR	<input checked="" type="checkbox"/> AMBOS

(1 punto) Selecciona las opciones correctas:

1. XML no hace nada, simplemente es una herramienta que permite almacenar e intercambiar datos de forma independientemente del hardware y el software que se utilice.[\[CORRECTO\]](#)
2. JavaScript es una variante simplificada de Java para ejecutar código funcional en un navegador.[\[INCORRECTO, Java y JavaScript son dos lenguajes con propósitos diferentes\]](#)
3. DHTML (*Dynamic HTML*) es la tecnología que surgió para permitir implementar páginas dinámicas en el lado servidor, haciendo que el contenido de los documentos HTML suministrado por los servidores web dejara de ser estático.[\[INCORRECTO, se utiliza en el lado cliente\]](#)
4. Antes que se inventara AJAX no se podía actualizar dinámicamente un documento HTML cargado en un navegador con contenidos obtenidos del servidor, solo se podía navegar a un nuevo documento.[\[CORRECTO\]](#)
5. Hay dos posibles mecanismos para enviar un formulario web del cliente al servidor, GET y POST. La diferencia entre ambos es que el método POST es seguro, mientras que el método GET no lo es al pasar los parámetros en la URL (*query string*).[\[INCORRECTO, además de GET y POST hay DELETE, PATCH y PUT\]](#)
6. El desarrollo de bibliotecas y frameworks AJAX está haciendo que se esté extendiendo el desarrollo de las llamadas aplicaciones SPA (*Single-Page Applications*, aplicaciones de una única página).[\[CORRECTO\]](#)

(0.75 puntos) ¿Qué significa que un fichero XHTML es un fichero HTML "bien formado"?

Cuando un documento XHTML se dice que está "bien formado", significa que sigue las reglas de sintaxis de XML y cumple con los estándares de XHTML. Esto incluye:

- Utilizando etiquetas de apertura y cierre correctas para todos los elementos.
- Utilizando etiquetas de cierre para todos los elementos vacíos.
- Utilizando etiquetas en minúsculas.
- Utilizando atributos en minúsculas.
- Utilizando comillas en torno a los valores de los atributos.
- Utilizando un DTD (Document Type Definition) válido para declarar el tipo de documento.

Un documento XHTML bien formado se mostrará correctamente en un navegador y también será fácil de procesar por aplicaciones y programas automatizados.

(0.75 puntos) Dada la siguiente definición de objeto JavaScript:

```
var persona = {nombre: "Juan", apellido: "Pérez"};
```

definir e incluir en el objeto una función *toString* que devuelva la concatenación del nombre y el apellido, de forma que la salida de *persona.toString()* sea "Juan Pérez"

```
var persona = {
  nombre: "Juan",
  apellido: "Pérez",
  toString: function () {
    return this.nombre + " " + this.apellido;
  },
};
```

**Que no te escriban poemas de amor
cuando terminen la carrera ▶▶▶▶▶▶▶▶**
(a nosotros por suerte nos pasa) 😊



WUOLAH

Nombre:

(0.5 puntos) ¿A nivel de arquitectura software, cuáles son las diferencias fundamentales entre una aplicación web diseñada para ejecutarse en internet y una diseñada para ejecutarse en una intranet?

- **Diseño escalable:** Una aplicación web diseñada para ejecutarse en Internet debe ser escalable para manejar un gran número de usuarios y tráfico, lo que puede requerir una arquitectura distribuida y un uso eficiente de recursos en el servidor. Por otro lado, una aplicación web diseñada para ejecutarse en una intranet puede tener requisitos de escalabilidad menos exigentes.
- **Seguridad:** Una aplicación web diseñada para ejecutarse en Internet debe tener medidas de seguridad adicionales para protegerse contra posibles amenazas externas, como ataques de hackers. Esto puede incluir la implementación de protocolos de seguridad como HTTPS, autenticación de usuarios y autorización. Por otro lado, una aplicación web diseñada para ejecutarse en una intranet puede depender en gran medida de las medidas de seguridad de la red privada para protegerse.
- **Integración con servicios externos:** Una aplicación web diseñada para ejecutarse en Internet puede requerir integrarse con una variedad de servicios y aplicaciones externos, como servicios de terceros de pago o de autenticación. Por otro lado, una aplicación web diseñada para ejecutarse en una intranet puede ser integrada directamente con las aplicaciones y servicios existentes en la red privada.
- **Despliegue:** Una aplicación web diseñada para ejecutarse en Internet debe ser desplegada en un entorno de producción seguro y escalable, como un proveedor de infraestructura en la nube o un servidor dedicado. Por otro lado, una aplicación web diseñada para ejecutarse en una intranet puede ser desplegada en un entorno de red privada, como un servidor local.
- **Monitoreo y mantenimiento:** Una aplicación web diseñada para ejecutarse en Internet requiere un monitoreo y mantenimiento constantes para garantizar su disponibilidad y rendimiento, mientras que una aplicación web diseñada para ejecutarse en una intranet puede requerir un monitoreo y mantenimiento menos intensivo debido a la menor cantidad de usuarios y tráfico.

(0.75 puntos) Dado el siguiente documento HTML:

```
<html>
<head>
  <script>
    function enviar() {
      var xmlhttp = new XMLHttpRequest();
      xmlhttp.open("GET", "http://servidor/info_externa.html", true);
      xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState === 4 && xmlhttp.status === 200) {
          var div = document.getElementById("contenedor");
          div.innerHTML = xmlhttp.responseText;
        }
      };
      xmlhttp.send();
    }
  </script>
</head>

<body>
  <div id="contenedor">Contenido temporal...</div>
  <input type="button" onclick="enviar()" value="Pedir AJAX" />
</body>
</html>
```



WUOLAH

No si antes decirte
Lo mucho que te voy a recordar

Pero me voy a graduar.
Mañana mi diploma y título he de pagar

Llegó mi momento de despedirte
Tras años en los que has estado mi
lado.

Siempre me has ayudado
Cuando por exámenes me he
agobiado

Oh Wuolah wuoliah
Tu que eres tan bonita

Implementar el cuerpo de la función *enviar()* que hace una petición POST AJAX asíncrona al servidor para actualizar el contenido del div con id "contenedor" con el documento *info_externa.html*.

Nombre:

Dado el siguiente fragmento de código HTML y asumiendo que se ha cargado la biblioteca *jQuery*:

```
<div id="contenedor">
  <div id="contenedor" class="impar">
    <div name="t1" id="t1_1">Texto 1.1</div>
    <div name="t1" id="t1_2">Texto 1.2</div>
  </div>
  <div class="par" name="t2">Texto 2</div>
  <div class="impar" name="t3">Texto 3</div>
  <div class="par" name="t4">Texto 4</div>
  <div class="impar" name="t5">Texto 5</div>
  <div class="par" name="t6">Texto 6</div>
  <div class="impar" name="t7">Texto 7</div>
  <div class="par" id="contenedor" name="t8">
    Texto 8
    <div id="contenedor">
      <div name="t8" id="t8_1">Texto 8.1</div>
      <div name="t8" id="t8_2">Texto 8.2</div>
    </div>
  </div>
</div>
```

(0.5 puntos) Implementar la función JavaScript *rojo()* que **utilizando jQuery** haga que el texto de todos los divs cuyo id sea "contenedor" que se encuentren dentro de otro div con id igual a "contenedor" pase a ser de color rojo.

```
function rojo() {
  $("div").parent("#contenedor").css("color", "red");
}
```

(0.5 puntos) Implementar la función JavaScript *impar()* que **utilizando jQuery** añada un div con el texto "Impar" de color azul al final de todos los divs de la clase impar.

```
function impar() {
  $(".impar").after($("<div>").css("color", "blue").html("impar"));
}
```

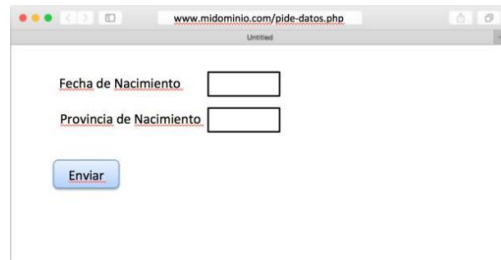
(0.75 puntos) Implementar la función JavaScript *par()* que **sin utilizar jQuery** añada un div con el texto "Par" de color azul al final de todos los divs de la clase par.

```
function par() {
  var pares = document.getElementsByClassName("par");
  for (var i = 0; i < pares.length; i++) {
    var d = document.createElement("div");
    d.innerHTML = "par";
    d.style = "color:blue";
    pares[i].appendChild(d);
  }
}
```


(3 puntos) Suponga que ya existe un código HTML que genera el siguiente formulario:



Escribir el código Python (utilizando la biblioteca *Flask*) que realice el procesamiento al pinchar el botón "Aceptar": se debe procesar la información del nombre y generar un nuevo formulario HTML (el formato es aproximado):



Finalmente se deben procesar los datos y mostrar un mensaje final donde se hayan calculado si es el cumpleaños del usuario y en ese caso saludarlo "Feliz cumpleaños nombre-usuario", donde *nombre-usuario* corresponde al nombre que el usuario introdujo en el primer formulario. Si no es su cumpleaños, se deben indicar cuántos días faltan para su próximo cumpleaños. Para ello, asuma que el campo "Fecha de Nacimiento" le devuelve una cadena de la forma "AAAA-MM-DD". Para saber la fecha actual puede utilizar la siguiente función:

`time.strftime(format[, t])`

Convert a tuple or `struct_time` representing a time as returned by `gmtime()` or `localtime()` to a string as specified by the *format* argument. If *t* is not provided, the current time as returned by `localtime()` is used. *format* must be a string. `ValueError` is raised if any field in *t* is outside of the allowed range. `strftime()` returns a locale dependent byte string; the result may be converted to unicode by doing `strftime(<myformat>).decode(locale.getlocale()[1])`.

Que, por ejemplo, se puede usar de la siguiente forma:

```
>>>import time
>>>print (time.strftime("%d/%m/%y"))
22/05/14
```

Este ejercicio tiene truco, porque está utilizando dos lenguajes de programación en servidor: PHP y Python. Esto no es común en la misma página. De hecho el formulario generado en Flask se llama con extensión `.php`

`index.html`

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Index</title>
</head>
<body>
  <form method="post" action="{{url_for('pidedatos')}}">
```


Que no te escriban poemas de amor
cuando terminen la carrera ▶▶▶▶▶▶▶▶



WUOLAH

(a nosotros por suerte nos pasa)

No si antes decirte
Lo mucho que te voy a recordar

Pero me voy a graduar.
Mañana mi diploma y título he de
pagar

Llegó mi momento de despedirte
Tras años en los que has estado mi
lado.

Siempre me has ayudado
Cuando por exámenes me he
agobiado

Oh Wuolah wuolilah
Tu que eres tan bonita

```
Nombre: <input name="nombre" type="text"><br/>
<input type="submit" value="Aceptar">
</form>
</body>
</html>
```

pidedatos.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pide datos</title>
</head>
<body>
  <form method="post" action="{{url_for('saludar')}}">
    <input name="nombre" type="hidden" value="{{nombre}}">
    Fecha de nacimiento: <input name="fecnac" type="text"><br/>
    Provincia de nacimiento: <input name="pronac" type="text"><br/>
    <input type="submit" value="Aceptar">
  </form>
</body>
</html>
```

saludar.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Saludar</title>
</head>
<body>
  {% if dias == 0 %}
    <p>Felicitades {{nombre}} que hoy es tu cumpleaños!</p>
  {% else %}
    <p>Hola {{nombre}}, faltan {{dias}} dias para tu cumpleaños!</p>
  {% endif %}
</body>
</html>
```

routes.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from app import app
from flask import render_template, request, url_for
import time, os, sys
from datetime import date

@app.route('/')
@app.route('/index', methods=['GET', 'POST'])
def index():
    return render_template('index.html')
```

```

@app.route('/pidedatos', methods=['GET','POST'])
def pidedatos():
    nomb=""
    if "nombre" in request.form:
        nomb = request.form['nombre']
    return render_template('pidedatos.html', nombre = nomb)

@app.route('/saludar', methods=['GET','POST'])
def saludar():
    nomb=""
    fnac=""
    pnac=""
    if "nombre" in request.form:
        nomb = request.form['nombre']
        fnac = request.form['fecnac']
        pnac = request.form['pronac']
        d = cumple(fnac)
    return render_template('saludar.html', nombre = nomb, dias = d)

# HELPERS
def separar_fecha(fecha):
    f = fecha.split("-")
    return int(f[0]),int(f[1]),int(f[2])

def cumple(fecha):
    hoy = time.strftime("%Y-%m-%d")
    fna, fnm, fnd = separar_fecha(fecha)
    ha, hm, hd = separar_fecha(hoy)
    # Usaremos el año actual para las fechas
    if date(ha,fnm,fnd) == date(ha,hm,hd):
        # Si coinciden el mes y el dia (hoy == cumple)
        # es el cumple hoy
        return 0
    elif date(ha,hm,hd) < date(ha,fnm,fnd):
        # Si faltan dias para el cumple (hoy < cumple)
        return (date(ha,fnm,fnd) - date(ha,hm,hd)).days
    else:
        # si ya pasamos el cumple,toca el año que viene
        # (hoy > cumple), sumamos un año al año actual
        return (date(ha+1,fnm,fnd) - date(ha,hm,hd)).days

```