

SI-junio-2017.pdf



Anónimo



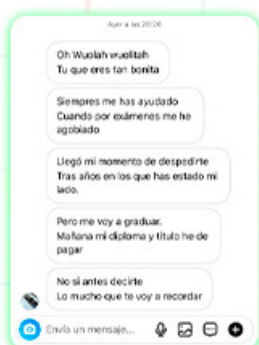
Sistemas Informaticos I



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

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 wuolah
Tu que eres tan bonita

Sistemas Informáticos I

Convocatoria Extraordinaria – 1ª parte

26 junio 2017

Test	1	2	3	4	5	6	Total
------	---	---	---	---	---	---	-------

Apellidos: _____

Nombre: _____

Test (4 puntos): 8 preguntas. Cada respuesta correcta suma 0,5 puntos, cada respuesta incorrecta resta 0,25. Respuesta en blanco no restan.

- ¿Cuál de las siguientes afirmaciones sobre el modelo arquitectónico Cliente/Servidor es verdadera?
 - Recomienda no implementar funcionalidad en la componente cliente (cliente pesado).
 - Fue el modelo propuesto inicialmente para la Web pero cayó en desuso porque no permitía la generación dinámica de páginas.
 - Es uno de los dos modelos utilizados para implementar la World Wide Web (el otro es el modelo Peer2peer)
 - Dota al sistema de escalabilidad, tanto horizontal como vertical.
- Cuál de estas características NO es propia de la arquitectura Cliente/Servidor:
 - Es un sistema simétrico
 - Los procesos clientes solicitan servicios
 - Los procesos clientes pueden ejecutar servicios
 - Los procesos servidores ejecutan servicios
- En los sistemas distribuidos basados en la World Wide Web, qué protocolos o lenguajes (entre otros) son interpretados por el navegador?
 - HTML y JSP
 - HTML y PHP
 - XSL, Javascript y CSS
 - HTML, Javascript y CGI
- ¿Cuál de los siguientes **no** es un método utilizado para mantener sesiones en las aplicaciones WWW?
 - CGIs
 - Reescritura de URLs
 - Cookies
 - Campos ocultos en los formularios
- ¿Cuál de estas características **no** es propia de CSS?
 - Ayuda a la separación entre el contenido de un documento y la forma de presentarlo
 - Ayuda a compartir formato entre múltiples páginas
 - Incentiva a ofrecer siempre la misma presentación para distintos dispositivos
 - Ayuda a dar flexibilidad de presentación
- ¿Cuál de las siguientes afirmaciones es **falsa**?
 - jQuery es una biblioteca que no agrega nada nuevo a JavaScript, en el sentido que lo mismo se puede hacer sin utilizar jQuery, pero tal vez con más trabajo.
 - jQuery surge para reemplazar lo que antes se hacía con Servlets y JSPs.
 - jQuery sólo se puede utilizar en aquellos navegadores que lo incluyen en su propio código.
 - La tecnología JavaScript está orientada a aumentar la funcionalidad del lado del cliente.

WUOLAH

7. ¿Cuál de estas afirmaciones sobre AJAX es **falsa**?
- a) Es un conjunto de técnicas para intercambiar información con el servidor y actualizar partes de la página web sin recargar la página completa.
 - b) Las aplicaciones AJAX dependen del navegador y de la plataforma.
 - c) No es otro lenguaje de programación, sino una forma nueva de usar tecnología existente.
 - d) Google popularizó el uso de AJAX.
8. ¿Qué conseguimos con la línea de código `evt.preventDefault()`?

```
function calcular(evt) {  
    evt.preventDefault();  
}  
document.getElementById('micheckbox').addEventListener('click', calcular, false);
```

- a) Lanza un evento llamado `preventDefault`.
- b) Ejecutar un método que habrá que definir y se llamará `preventDefault`
- c) Desasignar la respuesta por defecto de un evento.
- d) Ninguna de las tres anteriores es correcta.

EJERCICIO

Se pide implementar una aplicación Web que facilite realizar peticiones de citas médicas. En la aplicación se puede utilizar HTML, JavaScript (puede incluir el uso de jQuery), CSS y PHP. Todos los ficheros estarán alojados en el mismo directorio del mismo servidor.

Ejercicio 1 (1 punto): Escribe el código necesario para mostrar el formulario de la Fig.

1. El primer campo se llamará “nombre” y el segundo “numero”. Al pulsar en el botón siguiente se mostrará el formulario de la Fig. 2.

Reserva de cita médica - Inicio

Home About us Contacts

Nombre

Num Seguridad Social

Copyright 2016 ElProfesor.

Fig 1. Formulario inicial de la aplicación.

Ejercicio 2 (1 punto): Escribe el código de un fichero llamado `estilos.css` mediante el cual al pasar el curso sobre uno de los elementos del menú de navegación (elementos dentro de una división llamada “nav”) cambie el color del texto y el fondo de la opción (colores a su elección). Además, para ello modifique lo que sea necesario del ejercicio anterior.

Ejercicio 3 (1 punto): Escribe el código necesario para mostrar el formulario de la Fig.

2. El campo donde se guarda tipo de cita se llamará “tipo”. Al pulsar en el botón

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



WUOLAH

siguiente se mostrará el formulario de la Fig. 3 en todos los casos, excepto si el paciente selecciona "Especialista", en este caso se le mostrará este mensaje "Todavía no podemos ofrecerle reserva de especialistas, disculpe las molestias".

Fig 2. Elección de médico

Ejercicio 4 (1 punto): Escribe el código necesario Javascript para que sólo sean mandados los datos del formulario de la Fig. 1 si en el campo para el Numero de la Seguridad Social contiene sólo dígitos numéricos.

Ejercicio 5 (1 punto): Escribe el código necesario para mostrar el formulario de la Fig. 3. Los campos para guardar la preferencia del paciente se llamarán "dia" y "hora". Al pulsar en el botón siguiente se mostrará el mensaje final de la Fig. 4.

Fig 3. Elección de día y hora

Ejercicio 6 (1 punto): Escribe el código necesario para mostrar el mensaje final de la Fig. 4.

Fig 4. Mensaje final con la reserva realizada

Sistemas Informáticos I

Convocatoria Extraordinaria

Junio de 2017

1	2	3	Tot.
---	---	---	------

Apellidos: _____

Nombre: _____

Ejercicio 1 (2 puntos): 4 preguntas. Cada respuesta correcta suma 0,5 puntos.

1. ¿Cuál de estas afirmaciones es **cierta**?

- a) Todo procedimiento almacenado es un trigger.
- b) El lenguaje PL/pgSQL permite crear funciones que se ejecutan en el gestor de base de datos PostgreSQL.
- c) En un procedimiento almacenado, el resultado de cualquier consulta se puede almacenar en una variable utilizando SELECT INTO.
- d) En un trigger, el uso de BEFORE y AFTER aplicado a INSERT tienen el mismo resultado.

2. ¿Cuál de estas afirmaciones es **falsa** cuando se desea acceder a una base de datos desde PHP?

- a) La conexión a una base de datos se puede realizar con la función `pg_connect()`.
- b) La conexión a una base de datos se puede hacer creando una instancia de la clase PDO.
- c) Todas las instrucciones PDO, sin cambio alguno, pueden ser procesadas por cualquier gestor de bases de datos.
- d) PDO es una extensión de PHP.

3. Se tiene la siguiente definición de tabla:

create table datos (id integer, valor numeric, propiedad varchar(5));

Señale, de entre las siguientes, la instrucción que **no da error**:

- a) `select id, valor, propiedad from datos group by valor, propiedad;`
- b) `select count(id), valor from datos group by valor, propiedad;`
- c) `select count(id), valor, propiedad from datos group by valor;`
- d) Todas las instrucciones anteriores dan error.

4. ¿Cuál de estas afirmaciones es **falsa**?

- a) Las bases de datos actuales son todas bases de datos relacionales.
- b) Una base de datos puede modelarse como: una colección de entidades, relaciones entre entidades.
- c) Una entidad débil es aquella que no tiene clave primaria.
- d) Hablar del esquema de una base de datos no es hablar de sus datos.

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 titulo 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 wuolah
Tu que eres tan bonita

Ejercicio 2 (4 puntos): Dado el siguiente fragmento de código HTML, perteneciente a la página verproductividad.html

```
<FORM ACTION="verproductividad.php" METHOD=POST>
&ltp>Minimo:
&ltINPUT NAME="minimo" TYPE=Text
SIZE="10" MAXLENGTH="8">
&ltp>Máximo:
&ltINPUT NAME="maximo" TYPE=Text
SIZE="10" MAXLENGTH="8">
&ltINPUT TYPE=RESET VALUE="Borrar">
&ltINPUT TYPE=SUBMIT VALUE="Enviar">
</FORM>
```

Implementar en PHP (utilizando el extensión PDO) una función que retorne un fichero HTML que contenga una tabla que muestre para cada sucursal del banco su productividad y su ciudad, siempre que dicha productividad esté comprendida en el rango de valores indicado por los parámetros *minimo* y *maximo*.

La *productividad* de una sucursal se define como la diferencia entre la suma de los saldos de todas las cuentas de dicha sucursal y la suma de todas las cantidades prestadas en los préstamos asociados a dicha sucursal. Cualquier valor de productividad puede ser tanto positivo como negativo.

La salida, si se ha introducido en el formulario *minimo*=-1000 y *maximo*=1000, sería:

Productividad	Sucursal	Ciudad
-550	Round Hill	Horseneck
200	Mianus	Horseneck
280	Central	Rye

Ejercicio 3 (4 puntos): Se va a crear una tabla nueva llamada "TopSolventes". En ella se desea almacenar de las DOS ciudades que tienen los mejores promedios del saldo de sus cuentas asociadas: su nombre al que llamaremos *ciudad_sucursal* y dicho valor al que llamaremos *solventia*. Escribir la definición de dicha tabla (véase ejemplo) y un trigger en PL/pgSQL (la función y la definición del trigger) que mantenga actualizada dicha tabla en el futuro cuando haya cambios en los datos almacenados.

Ejemplo de salida de la nueva tabla:

ciudad_sucursal	solventia
character varying(15)	numeric
Rye	215.0000
Round Hill	700.0000

PROBLEMA
A partir del modelo de datos implementado en la base de datos "Banco":

```
create table prestamo
(nombre_prestamo varchar(15) not null unique,
nombre_sucursal varchar(15) not null references sucursal,
cantidad numeric not null,
primary key(numero_prestamo),
foreign key(nombre_sucursal)
references sucursal);

create table cliente_cuenta
(id_cliente varchar(15) not null,
numero_cuenta varchar(15) not null,
primary key(id_cliente,
numero_cuenta),
foreign key(numero_cuenta)
references sucursal(numero_cuenta),
foreign key(id_cliente)
references cliente(id));

create table cliente_prestamo
(id_cliente varchar(15) not null,
numero_prestamo varchar(15) not null,
primary key(id_cliente,
numero_prestamo),
foreign key(id_cliente)
references cliente(id),
foreign key(numero_prestamo)
references prestamo(numero_prestamo));
```

Ejemplo de datos almacenados en cada tabla:

cliente_sucursal : Tabla			
id	nombre_cliente	calle_cliente	ciudad_cliente
21-12-3123	Jones	Main	Rye
019-28-3746	Smith	Main	Rye
97-69-9011	Hayes	Main	Horseneck
552-56-5555	Cory	North	Rye
244-66-8900	Longway	Park	Pittsford
953-96-3863	Turner	Pulham	Stamford
111-11-1111	Williams	Nassau	Princeton
22-22-2222	Adams	Spring	Pittsford
333-33-3333	Johnson	Ama	Palo Alto
444-44-4444	Giam	Sand Hill	Woodsdale
555-55-5555	Green	Wyaluk	Stamford
777-77-7777	Smith	University	San Jose
888-88-8888	Melrose	Princeton	Rye
999-99-9999	McBride	Safety	Rye

cliente_cuenta : Tabla			
id_cliente	numero_cuenta	saldo	ciudad_cliente
333-33-3333	A-101	500	Downtown
019-28-3746	A-215	700	Mianus
677-69-9011	A-102	400	Perryridge
953-96-3863	A-305	350	Round Hill
333-33-3333	A-201	900	Perryridge
21-12-3123	A-217	700	Redwood
244-66-8900	A-222	750	Englewood
888-88-8888	A-353	850	Central
019-28-3746	A-444	625	North Town

Sistemas Informáticos I
Convocatoria Extraordinaria – 3ª parte
26 de junio 2017

1	2	3	4	5	Tot.
---	---	---	---	---	------

Apellidos: _____

Nombre: _____

Ejercicio 1 (2 puntos): Explique en qué consiste un ataque de SQL-injection, y las formas más habituales de evitarlo.

Ejercicio 2 (2 puntos): Data la siguiente secuencia de ejecución de acciones pertenecientes a 3 transacciones distintas, **explicar el grado máximo de aislamiento** que puede tener cada una de ellas y por qué, asumiendo que todas tienen por lo menos aislamiento grado 0.

Transacción 1	Transacción 2	Transacción 3
Begin		
	Begin	
		Begin
Read y		
	Write y	
Read y		
		Read z
	Write z	
	End	
End		
		Write z
		End

Ejercicio 3 (2 puntos): Dada la siguiente secuencia de operaciones pertenecientes a una transacción

Write y
Read x
Read z
Write y
Read z
Write x

Colocar en cada columna los locks (escritura o lectura) y unlocks que corresponderían si la transacción fuese del grado correspondiente a esa columna.

	Grado 0	Grado 1	Grado 2	Grado 3
Write y				
Read x				
Read z				
Write y				
Read z				
Write x				

Ejercicio 4:

a) (1 punto): Suponga que la siguiente secuencia de operaciones pertenece a una transacción con nivel de **aislamiento 1**, que pretende maximizar el acceso concurrente a los recursos dentro de los límites impuestos por el nivel de aislamiento indicado. Explique si existe algún error y cuál es.

Slock x
Read x
Unlock x
Xlock y
Write y
Xlock x
Write x
Unlock x
Read z
Write y
Unlock y

b) (1 punto): Suponga que la siguiente secuencia de operaciones pertenece a una transacción con nivel de **aislamiento 2**, que pretende maximizar el acceso concurrente a los recursos dentro de los límites impuestos por el nivel de aislamiento indicado. Explique si existe algún error y cuál es.

Slock x
Read x
Unlock x
Xlock y
Write y
Xlock x
Write x
Unlock x
Read z
Write y
Unlock y

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

Ejercicio 5 (2 puntos): Dadas las siguientes tablas de la base de datos *Peliculas*

Data Output	Explain	Message	Data Output	Explain	Data Output	Explain	Messages	History
	actor_id smallint	name character varying(45)		actor_id smallint	film_id smallint		film_id smallint	name character varying(45)
1	1	Harrison Ford	1	1	2	1	1	LOTR: The Fellowship of the Ring
2	2	Viggo Mortensen	2	1	3	2	2	Star Wars: Episode IV
3	3	Robert De Niro	3	2	1	3	3	Indiana Jones and the Last Crusade

Escribir la salida del siguiente fragmento de código PHP:

```
<?php
function printTableActor($dbRef){
    $sql = "SELECT * FROM actor";
    foreach ($dbRef->query($sql) as $row){
        print $row['actor_id'] . ' - ' . $row['name'] . '<br/>';
    }
    print '<p>';
}

print "Ejemplo de transacciones en PHP-PDO.<p>";
function concurrente(){
    try{
        $db2 = new PDO("mysql:host=localhost;port=3306;dbname=film", "root", "");
        $db2->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $db2->beginTransaction();
        print ("concurrente<br>");
        printTableActor($db2);
        $sql = "DELETE FROM actor WHERE actor_id=3";
        $db2->exec($sql);
        printTableActor($db2);
        $db2->commit();
    }
    catch (PDOException $e){
        print "Un error en concurrente. <p>";
        echo $e->getMessage();
        print "<p>";
    }
}

try{
    $db = new PDO("mysql:host=localhost;port=3306;dbname=film", "root", "");
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $db->beginTransaction();
    printTableActor($db);
    $sql = "DELETE FROM actor WHERE actor_id=3";
    $db->exec($sql);
    concurrente();
    $db->rollBack();
    printTableActor($db);
}
catch (Exception $e){
    print "Un error<p>";
    echo $e->getMessage() . "<p>";
}

// Continuar la ejecución
echo 'Fin del script';
?>
```

WUOLAH