

Aplicaciones Web

Grado en Ingeniería Informática

Examen de Teoría (1ª conv.) – 5 de junio de 2024



universidad
de león

Leer atentamente antes de comenzar el examen

- Dispones de un máximo de 2 horas para completar el examen.
- No es posible abandonar el aula durante al menos la primera media hora tras dar comienzo el examen.
- Debes responder a cada pregunta en el espacio habilitado en las propias hojas de examen a menos que se indique lo contrario en el enunciado.
- Está prohibido escribir en la tabla de calificaciones de más abajo.
- Está prohibido el uso de cualquier tipo de dispositivo electrónico durante la realización del examen.

Nombre: _____

Tabla de calificaciones

Pregunta	Puntos	Calificación
1	9	
2	11	
3	6	
4	3	
5	5	
6	5	
7	5	
8	8	

Pregunta	Puntos	Calificación
9	24	
10	5	
11	4	
12	5	
13	5	
14	5	
Total	100	

DAR LA VUELTA A LA HOJA ÚNICAMENTE CUANDO SE INDIQUE

Fundamentos de Aplicaciones Web

1. Identificar los componentes de las URLs de la forma más específica posible.

(1a) (4 puntos)

`https://pokeapi.co/api/v2/pokemon?limit=20&offset=1`

Solución:

`https: //pokeapi.co /api/v2/pokemon ?limit=20&offset=1`

Esquema

Autoridad

Parte jerárquica

Parámetros de consulta

(1b) (5 puntos)

`mongodb://h4x0r:1234@cluster0.aw.mongodb.net:1337/uber`

Solución:

`mongodb: //h4x0r:1234@ cluster0.aw.mongodb.net :1337 /uber`

Esquema

Información del usuario

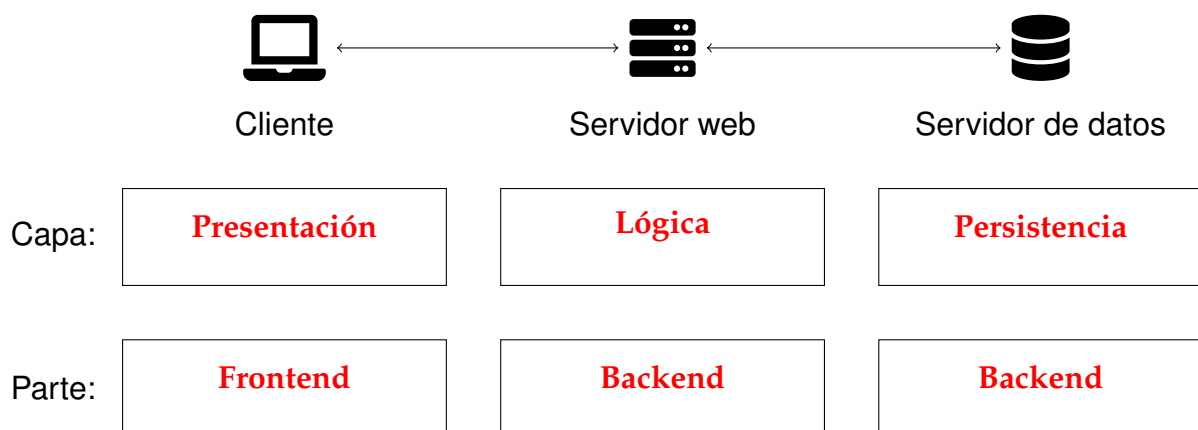
Máquina ó host

Puerto

Path ó ruta

2. Responder a las preguntas relativas al diseño de una aplicación web.

(2a) (6 puntos) El siguiente diagrama muestra una posible arquitectura software de tres capas para una aplicación web. Etiquetar cada una de las capas y señalar cuál/es pertenecen a la parte del *frontend* y cuál/es pertenecen a la parte del *backend*.



(2b) **(3 puntos)** Completar las definiciones relativas al patrón de diseño software MVC con el nombre del componente adecuado.

- Controlador hace de intermediario entre los otros dos componentes.
- Vista gestiona el diseño y la presentación de los datos.
- Modelo gestiona los datos y, en parte, la lógica de negocio.

(2c) **(1 punto)** ¿Cómo se llama el patrón de diseño de aplicaciones web que permite acceder a todo su contenido a través de múltiples documentos web?

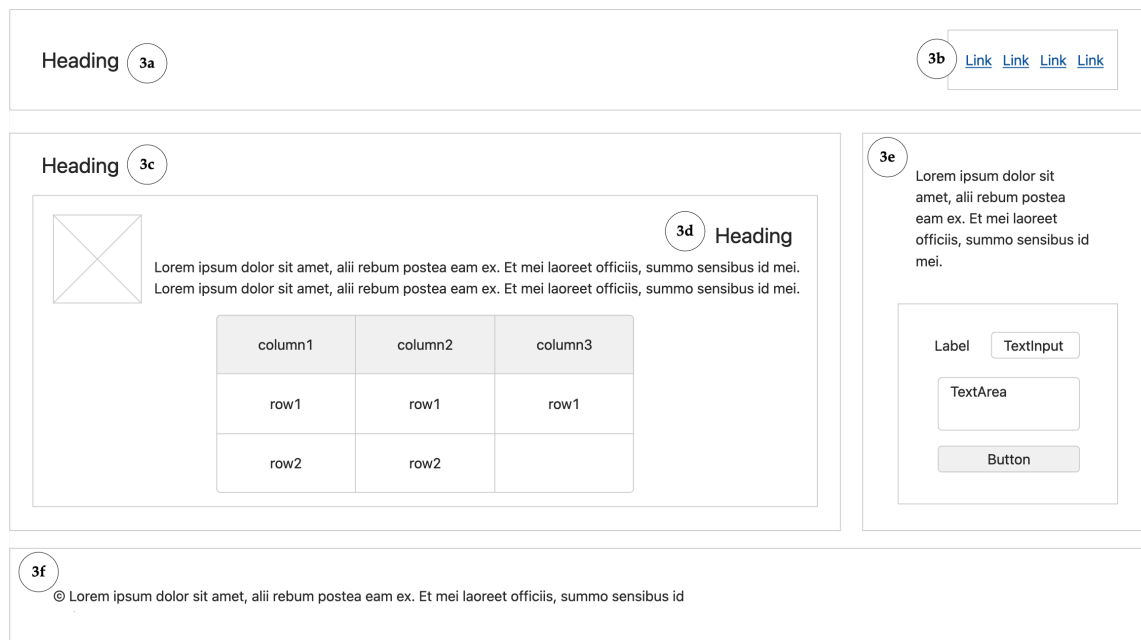
- ☐ PWA ☐ SPA ☒ MPA ☐ MVC

(2d) **(1 punto)** ¿Cómo se llama el patrón de diseño de aplicaciones web que permite acceder a todo su contenido a través de un único documento web?

- ☐ PWA ☒ SPA ☐ MPA ☐ MVC

HTML

3. La siguiente imagen muestra el prototipo de una página web.



Identificar cada uno de los contenedores señalados utilizando exclusivamente elementos HTML semánticos.

(3a) **(1 punto)** Contenido introductorio:

<header>

(3b) **(1 punto)** Navegación:

<nav>

(3c) **(1 punto)** Contenido principal:

<main>

(3d) **(1 punto)** Apartado dentro de una sección:

<article>

(3e) **(1 punto)** Contenido no relacionado:

<aside>

(3f) **(1 punto)** Pie de página:

<footer>

CSS

4. Calcular la especificidad de los selectores utilizando el formato X-X-X-X.

(4a) (1 punto) `body > p + img {}` 0-0-0-3

(4b) (1 punto) `main#container .grid {}` 0-1-1-1

(4c) (1 punto) `button:hover p::first-line {}` 0-0-1-3

5. Considera el siguiente código.

```
1 <body>
2   <div>Uno <span>Dos</span> Tres</div>
3   <div>Cuatro</div>
4 </body>
```

Escribir las reglas CSS necesarias para:

(5a) (2 puntos) Ocultar el elemento `` pero sin eliminarlo de la maquetación.

Solución:

```
span {
  visibility: hidden;
}
```

(5b) (3 puntos) Hacer que el elemento `<body>` utilice una maquetación de tipo caja flexible y que todos los elementos contenidos estén distribuidos horizontalmente.

Solución:

```
body {
  display: flex;
  flex-direction: row;
}
```

JavaScript

6. Responder a las preguntas relativas a la API DOM.

(6a) (1 punto) ¿Qué propiedad de `Document` permite acceder al elemento raíz del documento?

☒ `documentElement` ☐ `head` ☐ `body` ☐ `html`

(6b) (1 punto) ¿Qué propiedad de `Element` permite modificar directamente la estructura de un elemento HTML?

☐ `nodeName` ☐ `textContent` ☒ `innerHTML` ☐ `outerHTML`

(6c) (3 puntos) Completar las definiciones relativas a las fases de la propagación de eventos de JavaScript.

- **Objetivo** ejecuta el manejador asociado al elemento que originó el evento.
- **Burbujeo** comienza en el elemento anidado más interno y va recorriendo hacia arriba el DOM hasta llegar al elemento externo.
- **Captura** comienza en el elemento más externo y va recorriendo hacia abajo el DOM hasta llegar al elemento anidado.

7. (5 puntos) Considera el siguiente código.

```
1 <form id="form">
2   <label>Nombre: <input type="text"></label>
3   <button>Enviar</button>
4 </form>
5 <script>
6   const handler1 = (event) => { event.stopPropagation();
7                                   console.log("form"); };
8   const handler2 = (event) => { event.preventDefault();
9                                   console.log("label"); };
10  const handler3 = (event) => { event.preventDefault();
11                                console.log("input"); };
12  const handler4 = (event) => { event.stopPropagation();
13                                event.preventDefault();
14                                console.log("button"); };
15  document.querySelector("form")
16    .addEventListener("click", handler1, false);
17  document.querySelector("label")
18    .addEventListener("click", handler2, true);
19  document.querySelector("input")
20    .addEventListener("click", handler3, true);
21  document.querySelector("button")
22    .addEventListener("click", handler4, false);
23 </script>
```

¿Qué mensaje se imprime por consola al hacer click sobre el elemento <input>?

Solución:

label
input
form

ESPACIO DEJADO EN BLANCO A PROPÓSITO

Comunicación Cliente-Servidor

8. Considera el siguiente mensaje HTTP.

Método HTTP **Path** **Versión HTTP**

```
POST /objects HTTP/2
User-Agent: curl/8.4.0
Accept: */*
Host: api.restful-api.dev
Content-Type: application/json
Content-Length: 58
```

Cabeceras

Línea en blanco

```
{ "name": "Aplicaciones Web", "data": { "course": 2024 } }
```

Cuerpo/body

Responder a las preguntas relativas a dicho mensaje.

(8a) (6 puntos) Etiquetar sobre el propio mensaje todas sus partes.

(8b) (1 punto) ¿Qué tipo de mensaje es?

Petición



(8c) (1 punto) ¿A qué URL va dirigido?

api.restful-api.dev/objects

9. (24 puntos) Diseñar una API REST que permita representar dos servicios *independientes pero relacionados*: uno relativo a artistas musicales y otro a álbumes de canciones. Para ello deben cumplirse los siguientes requisitos:

- Todos los recursos relativos a artistas son identificados con el sustantivo `artists` y los relativos a álbumes son identificados con el sustantivo `albums`.
- Para identificar un recurso específico utilizar el parámetro `:id` en la URL.
- Todos los endpoints están agrupados dentro de una URL base `/api/`
- Cuando un recurso de cualquier servicio es actualizado, ya sea de tipo singleton o colección, la respuesta de la API REST será devolver el código de estado relativo a recurso movido temporalmente y el cuerpo del mensaje contendrá la URL a la que redireccionar al cliente.

Teniendo en cuenta lo anterior, completar la tabla relativa al diseño de la API REST.

Descripción	Método HTTP	Endpoint	Cód. estado		Contenido del cuerpo	
					Petición	Respuesta
Obtener una colección de los álbumes de un artista específico	GET	/api/artists/:id/albums	200	404	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Crear un álbum específico	POST	/api/albums	201	400	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Actualizar un artista específico	PUT	/api/artists/:id	307	409	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Eliminar un álbum específico	DELETE	/api/albums/:id	204	404	<input type="checkbox"/>	<input type="checkbox"/>

10. (5 puntos) Considera el siguiente código.

```
1  const promise = (a, b) => {  
2    return new Promise((resolve, reject) => {  
3      setTimeout(() => {  
4        if (a < 0 || b < 0) {  
5          return reject("Números negativos");  
6        } else {  
7          resolve(a + b);  
8        }  
9      }, 1000);  
10   });  
11 };  
12 promise(0, 2)  
13 .then((data) => { console.log(data);  
14               return promise(data, 4); })  
15 .then((data) => { console.log(data); })  
16 .catch((error) => { console.log(error); });
```

¿Qué mensaje se imprime por consola?

Solución:

2
6

Servidor Web

11. Responder a las preguntas relativas a Servidores Web y a Node.js.

- (11a) (2 puntos) Un servidor HTTP es un socket abierto que escucha constantemente peticiones en un puerto de red para procesarlas y enviarlas de vuelta.
- (11b) (1 punto) Node.js es utilizable únicamente a través del objeto global `window`.
☐ Verdadero ☒ Falso
- (11c) (1 punto)Cuál de los siguientes parámetros NO pertenece a la función `wrapper` de Node.js.
☐ `exports` ☐ `__dirname` ☒ `req` ☐ `module` ☐ `__filename`

12. Escribir las sentencias relativas a Express.js asumiendo que el módulo `express` ha sido cargado previamente.

- (12a) (1 punto) Instanciar una aplicación de Express.js denominada `http_server`.

Solución:

```
const http_server = express();
```

- (12b) (1 punto) Crear un middleware denominado `controller` cuya única finalidad sea delegar en el siguiente middleware que haya en el *stack*.

Solución:

```
const controller = (req, res, next) => { next(); };
```

- (12c) (1 punto) Instanciar un enrutador denominado `pkmn_api`.

Solución:

```
const pkmn_api = express.Router();
```

- (12d) (1 punto) Definir un middleware de enrutamiento para peticiones de tipo `DELETE` utilizando `pkmn_api` y asociando el endpoint `/pokemon` al controlador `controller`.

Solución:

```
pkmn_api.delete("/pokemon", controller);
```

- (12e) (1 punto) Definir un middleware de aplicación utilizando `http_server` para asociar el endpoint `/api/v2/` al enrutador `pkmn_api`.

Solución:

```
http_server.use("/api/v2/", pkmn_api);
```

Autenticación

13. Responder a las preguntas relativas a la autenticación de usuarios.

- (13a) (1 punto) ¿Qué forma de almacenar una sesión ofrece un acceso rápido pero afecta a la escalabilidad de la aplicación web cuando el servidor tiene una alta carga de trabajo?

☐ Base de datos ☐ Caché de datos ☐ Cookie ☒ **Memoria RAM**

- (13b) (1 punto) Una cookie es información temporal que un servidor almacena para identificar posteriormente a un cliente específico.

☐ Verdadero ☒ **Falso**

- (13c) (3 puntos) Indica en el orden correcto los tres componentes que componen la estructura de un token JWT.

header . payload . signatura

Persistencia

14. Responder a las preguntas relativas a la persistencia de datos.

(14a) **(4 puntos)** ¿A qué términos de una base de datos MongoDB corresponden los siguientes términos de una base de datos relacional?

- | | |
|------------------|----------------------|
| ■ Base de datos: | <u>Base de datos</u> |
| ■ Tabla: | <u>Colección</u> |
| ■ Fila: | <u>Documento</u> |
| ■ Columna: | <u>Campo</u> |

(14b) **(1 punto)** Mongoose es un paquete para Node.js que implementa un mecanismo de:

- ☐ ORM ☒ **ODM** ☐ DAO ☐ API

FIN DEL EXAMEN 😊