MF0492\_3: PROGRAMACIÓN WEB EN EL ENTORNO SERVIDOR

|  |  |
| --- | --- |
| **Denominación del Certificado de Profesionalidad** | (IFCD0210) DESARROLLO DE APLICACIONES CON TECNOLOGÍAS WEB |
| **Nº de curso/Expediente** | 2051/FOD/24/2020 |

|  |  |
| --- | --- |
| **Nombre y apellidos:** |  |
| **Nº de DNI.:** |  |
| **Teléfono de contacto:** |  |
| **E-mail:** |  |
| **Fecha:** |  |

|  |  |
| --- | --- |
| **Duración de la prueba** | 3 horas |
| **Estructura de la prueba** | Test teórico con preguntas tipo test variadas Práctica de codificación de una página web |
| **Criterios de calificación de la prueba** | Cada ejercicio tiene su valoración descrita en el enunciado. Cada UF tiene su valoración independiente de la prueba teórica y práctica con su propia nota en base 10. Se necesita un 4.0 tanto en la parte teórica como en la práctica para hacer media.  Aprobando con una media conjunta de 5.0. Es necesario aprobar todas las UF para superar el examen. Haciendo la media final entre las notas de las 3 UF en escala de 10.  El valor del examen es de un 70% del total de la nota del MF, el 30% restante se obtiene de la evaluación continua. |
| **Materiales que puede utilizar el alumno** | Ordenador, apuntes materiales, repositorio de código. |
| **Respuestas a las preguntas** | Contestar según lo indicado en la pregunta.  Código informático entregado en una carpeta. |

|  |  |
| --- | --- |
| **Firma del docente:** |  |
| **Firma del/a alumno/a:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| NOTA | *UF1844* | *UF1845* | *UF1846* |
|  |  |  |

Examen Teórico

UF1844: Desarrollo de aplicaciones web en el entorno servidor

1 - Contesta verdadero o falso las siguientes preguntas sobre Node – **0.25 por respuesta correcta – 2 Puntos en total**

* Se utiliza en el lado del cliente V F
* Es una versión del lenguaje PHP V F
* Usa el motor V8 de Chrome V F
* No es extensible mediante módulos V F
* Solo podemos programar mediante CLI V F
* Se pueden usar Patrones (MVC, MVP, MVVM) V F
* Es asíncrono V F
* Es multiplataforma (Windows, Linux, Mac…) V F

2 - ¿Cómo ejecutas la aplicación server.js en Node.js? - **– 1 punto**

1. run server.js
2. node server.js
3. exec server.js
4. cmd server.js

3 - ¿Cuáles de los siguientes es un lenguaje válido para Node.js? **– 1 punto**

1. PHP
2. Java
3. JavaScript
4. C#

4 - Selecciona la respuesta correcta a las siguientes preguntas – **0.5 puntos por pregunta correcta – 6 puntos en total**

**¿Qué es una promesa?**

1. Una función dentro de JavaScript
2. Una estructura de datos que guarda valores aleatorios
3. Un objeto que representa el valor de una operación que será resuelta en el futuro
4. La prueba de amor existente entre el cliente y el servidor

**¿Qué herramienta nos ayuda a gestionar los módulos de Node?**

1. Docker
2. NPM
3. Modulizer
4. Apt-get

**¿Método para importar módulos en NodeJS?**

1. require()
2. import()
3. include()
4. package()

**¿Qué archivo es utilizado por NPM para especificar los módulos que se van a utilizar en la aplicación?**

1. package.xml
2. package.json
3. install.json
4. required.json

**¿Cómo podemos comprobar la versión actual de NPM?**

1. npm --ver
2. npm help
3. npm --version
4. npm status

**Express maneja una infraestructura de:**

1. Middleware
2. Betweenware
3. Upperware
4. Midgardware

**La manera en que los módulos de Node.js pueden estar disponibles externamente es con:**

1. module.spread
2. module.exports
3. module.expose
4. module.share

**¿Cómo se termina un proceso en Node.js en Windows?**

1. Ctrl + B
2. Ctrl + K
3. Ctrl + T
4. Ctrl + C

**¿Cuál de los siguientes módulos es necesario para crear un servidor web?**

1. net module
2. http module
3. net module
4. url module

**¿Cuál de los siguientes motores de plantillas pueden utilizarse con Node.js?**

1. EJS
2. Pug
3. Mustache
4. Handbrakes
5. Todos los anteriores

**¿Cuáles son las características básicas del framework Express?**

1. Express permite configurar el middleware para responder a las peticiones HTTP.
2. Express permite definir una tabla de enrutamiento que puede funcionar según el método HTTP y la URL.
3. Express permite renderizar una página HTML dinámica.
4. Todas las anteriores
5. Ninguna de las anteriores

**¿Una aplicación desarrollada en NodeJs puede acceder a qué tipo de bases de datos?**

1. Base de Datos Relacionales
2. Bases de Datos No-SQL
3. Todas las anteriores
4. Ninguna de las anteriores

UF1845: Acceso a datos en aplicaciones web del entorno servidor

1 – Contesta las siguientes preguntas – **0.5 puntos por pregunta acertada – 8 Puntos en Total**

**¿Qué significa el término SQL?**

1. Structured Query Language
2. Simple Question Language
3. Strong Query Library
4. Standard Quality Library

**¿Qué sentencia SQL se utiliza para obtener datos de una base de datos?**

1. SELECT
2. QUERY
3. GET
4. READ

**¿Qué sentencia SQL se utiliza para actualizar datos en una base de datos?**

1. UPDATE
2. UPGRADE
3. MODIFY
4. SET

**¿Qué sentencia SQL se utiliza para insertar nuevos datos en una base de datos?**

1. CREATE NEW
2. INSERT INTO
3. NEW ROW
4. ADD DATA

**¿Cuántas normas formales (NF) existen?**

1. 5
2. 6
3. 3
4. 4

**La palabra clave NULL en SQL se utiliza para:**

1. representar el vacío abisal.
2. representar un valor 0.
3. representar un valor ausente o desconocido. NULL en SQL no representa nada.
4. representar el infinito positivo.

**¿Qué es una clave primaria?**

1. La clave primaria es una columna o combinación de columnas cuyos valores identifican de forma exclusiva cada fila de la tabla.
2. La clave primaria es una columna que puede tener valores NULL.
3. La columna de clave primaria es una columna o combinación de columnas cuyos valores pueden ser no únicos.
4. Es la columna que sobresale por encima de las demás al estar encriptada

**¿Qué es un índice?**

1. Un índice es lo mismo que un alias.
2. Un índice es un atributo de la tabla de la base de datos que acelera la búsqueda de datos dentro de una tabla.
3. Un índice es una forma especial de unir dos o más tablas.
4. Un índice es un puntero a una selección de registros de una tabla

**Las filas de la tabla también se conocen como...**

1. Atributos
2. Registros
3. Campos
4. Objetos

**La normalización es ...**

1. el proceso de añadir una clave primaria a una tabla
2. el proceso de organizar la información almacenada en una base de datos de forma que se elimine la redundancia y la ambigüedad.
3. una forma especial de seleccionar los datos
4. eliminar extravagancias de los datos como valores extremos o nulos

**Mongo es una base de datos NOSQL basada principalmente en:**

1. Clave valor
2. Documentos
3. Columnar
4. Grafos

**RDBMS significa...**

1. Relational Database Management System
2. Real Database Management System
3. Read Database Master System
4. Relationship Database Manager System

**¿Qué es una vista?**

1. Una vista es un diagrama de base de datos.
2. Una vista es un procedimiento almacenado especial que se ejecuta cuando se produce un determinado evento.
3. Una vista es una tabla virtual que resulta de la ejecución de una consulta precompilada. Una vista no forma parte del esquema físico de la base de datos, mientras que las tablas normales sí.
4. Una vista es una selección especifica de registros que están visibles y no ocultos.

**¿Qué palabra clave de SQL se utiliza para recuperar sólo valores únicos?**

1. DISTINCTIVE
2. DISTINCT
3. UNIQUE
4. DIFFERENT

**¿Qué hace la siguiente sentencia SQL? SELECT Cliente, COUNT(Pedidos) FROM Ventas GROUP BY Cliente HAVING COUNT(Pedidos) >5**

1. Selecciona todos los clientes de la tabla Ventas que han hecho más de 5 pedidos.
2. Selecciona todos los clientes de la tabla Ventas
3. Selecciona el número total de pedidos de la tabla Ventas, si este número es superior a 5
4. Selecciona los primeros 5 primeros clientes que tengan pedidos

**Nos referimos a un join como un self-join cuando...**

1. utilizamos la unión a la izquierda y a la derecha a la vez
2. se une una tabla a sí misma
3. se unen más de dos tablas
4. se unen todas las tablas de la base de datos

Une cada una de las fases de diseño de una base de datos con el lenguaje adecuado – **0.25 Por respuesta correcta – 1 Punto Total**

1. Análisis Requisitos SQL
2. Diseño Conceptual UML
3. Diseño Lógico Entidad / Relación
4. Diseño Físico Modelo Relacional

Une cada comando SQL con el tipo de lenguaje al que pertenecen. **0.25 Por respuesta correcta – 1 Punto Total**

1. INSERT DQL (Data Query Language)
2. REVOKE Data Definition Language (DDL)
3. ALTER Data Manipulation Language (DML)
4. SELECT Data Control Language (DCL)

UF1846: Desarrollo de aplicaciones web distribuidas

Selecciona la respuesta correcta entre las diferentes opciones – **1 Puntos por respuesta correcta**

**En la arquitectura cliente-servidor el cliente es el que…**

1. Genera páginas web
2. Hace peticiones y recibe respuestas
3. Recibe peticiones y envía respuestas
4. Busca en la base de datos

**En la arquitectura cliente-servidor el servidor es el que…**

1. Hace peticiones y recibe respuestas
2. Recibe peticiones y envía respuestas
3. Renderiza contenido
4. Interactúa a través de un navegador

**Representa la petición HTTP**

1. Response
2. Route
3. Request
4. Requiere

**Representa la respuesta HTTP:**

1. Response
2. Requiere
3. Request
4. Requiere

**Es el código que indica que la solicitud ha tenido éxito:**

1. 404
2. 500
3. 200
4. 101

**Cuales son tipos de infraestructura orientada a servicios (SOA)**

1. AJAX, SOAP, REST
2. AJAX, PSG, BAYERN
3. EXT, GEL, RELAX
4. XML, JSON, SOCKET

**Los datos de un POST vienen dentro de:**

1. Req.params
2. Req.body
3. Req.query
4. Req.post

**¿Cuál de las siguientes afirmaciones es cierta sobre los servicios web RESTful?**

1. Los servicios web basados en la arquitectura REST se conocen como servicios web RESTful
2. Los servicios web utilizan métodos HTTP para implementar el concepto de arquitectura REST
3. Ambas cosas
4. Ninguna de las anteriores

**¿Cuál de los siguientes no es un método HTTP válido?**

1. GET
2. POST
3. PUT
4. HEADER

**Que método es más adecuado actualmente para una web empresarial**

1. SOAP
2. SOA
3. REST
4. AJAX

**Examen Práctico - MF0491\_3: Programación web en el entorno cliente**

UF1844: Desarrollo de aplicaciones web en el entorno servidor

**Ejercicio NodeJS 1 – Configuración - Total 0.5 puntos**

* Crear una carpeta denominada examen\_*Nombre*.
* En dicha carpeta se guardarán todos los archivos del examen.
* Abre la carpeta con VS Code y trabaja con dicho IDE.
* Esta carpeta posteriormente se subirá al repositorio.
* En este paso puedes clonar el repositorio con la estructura del examen.

**Ejercicio NodeJS 2 – Servidor Web NodeJS - Total 1.5 puntos**

* **0.25 -** Crea un fichero http.js en la raíz del proyecto.
* **0.75 -** Crea un servidor web usando solo NodeJS usando el módulo http
* **0.25 -** Muestra el mensaje “Examen <tu nombre>” cuando se acceda al servidor
* **0.25 -** El puerto del servidor debe ser el 8888. Utiliza una constante para ello.

**Ejercicio NodeJS 3 – Crea una aplicación Web usando Express, y EJS**

* **0.25 -** Crea un archivo **server.js** en la raíz del proyecto
* **0.25 -** El resto de los ficheros y directorios estará contenido en el directorio /app. Puedes usar la plantilla proporcionada
* **0.25 -** Importa el módulo Express.
* **0.25 –** Utiliza el módulo Express – Recuerda que hay que instalar los módulos importados.
* **0.50 -** Lanza el servidor Express en el puerto 4000
* **1.00 -** Configura la ruta **“/” –** Que muestre un objeto JSON con un mensaje: “Examen MF0491”, alumno: <Tu Nombre>, Academia: Focyl.
* **0.50 -** Configura Express para usar el motor de plantillas EJS
* **0.25 -** Configura el directorio /views/ para almacenar las plantillas
* **0.25 -** Configura un directorio de ficheros estáticos /public
* **1.50 –** Crea una ruta alumno que muestre una plantilla al que se le pasen tu nombre y apellidos de datos. Y que muestre una imagen desde /public.
* **1.00 –** Maneja el error 404 desde Express.
* **1.00 –** Crea una plantilla usando EJS con una imagen que esté situada en /public para el error 404
* **0.50 -** Utiliza un patrón de diseño Modelo Vista Controlador
* **0.50 -** El proyecto utiliza una estructura de carpeta organizada jerárquicamente

UF1845: Acceso a datos en aplicaciones web del entorno servidor

**1 - Ejercicio SQL 1 – Gestión de la base de Datos - Total 1 Puntos**

* **1.00 –** Conéctate a la base de datos, mediante interfaz web o un cliente de base de datos (Heidi SQL)

Crea un usuario con tu nombre y contraseña exam3n, dale los permisos mínimos para que pueda hacer operaciones CRUD. Guarda una captura de pantalla similar a la mostrada en /public

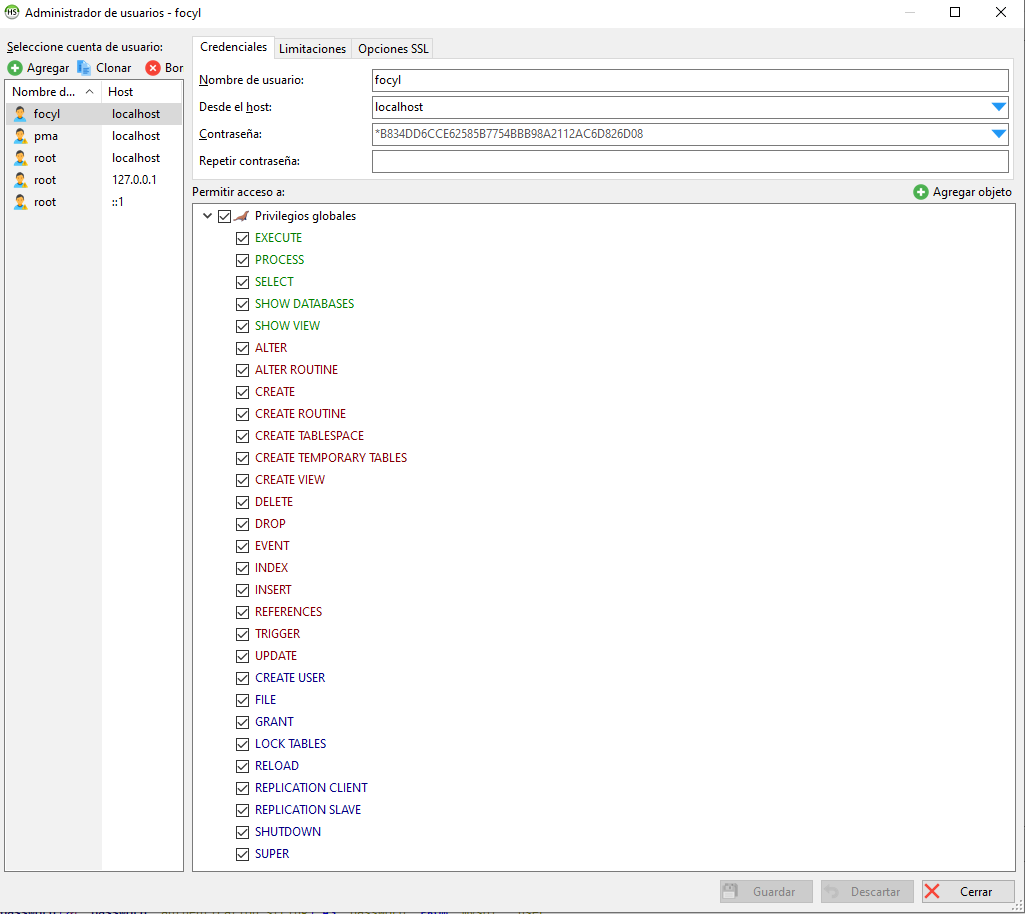


Ilustración - Ejemplo de captura de pantalla de creación de usuario

**2 - Ejercicio SQL 2 – Creación de una base de datos – 4 Puntos**

Crea la base de datos **cursoweb** descrita en el diseño lógico que acompaña a este enunciado. Puedes utilizar cualquier medio o herramienta vista en clase.

Crea un fichero **respuestas.sql** Para cada tabla adjunta el código SQL para su creación.

Inserta al menos 3 registros para cada tabla.

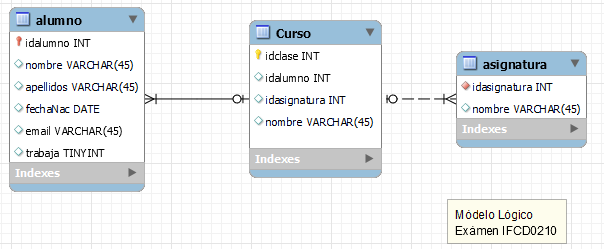


Ilustración - Modelo Lógico de la tabla cursoweb a diseñar

**2 - Ejercicio SQL 3 – Consultas – 5 Puntos**

Realiza las siguientes consultas para la tabla tiendaonline, con la que hemos trabajado durante el curso. Guarda la sentencia SQL para cada consulta en el fichero **respuestas.sql**

**0.5 – Consulta 1** - Obtén toda la información de la tabla cámaras

**0.5 – Consulta 2 -** Lista los nombres, marca y pvp como precio de los artículos

**0.5 – Consulta 3 -** Lista los email y nombre y apellidos de los usuarios ordenados ascendentemente por apellidos y descendentemente por nombre

**0.5 – Consulta 4 -** Lista los televisores de 21 pulgadas o menos de pantalla, eliminando duplicados.

**0.5 – Consulta 5 -** Código y nombre de las provincias que no son Valladolid, Burgos, Madrid, Cuenca

**0.5 – Consulta 6 -** Código de la provincia León – Cuidado con acentos, y errores por la codificación del texto.

**1.0 – Consulta 7 - Número** de pedido e identificador, apellidos y nombre del usuario que realiza el pedido, y nombre de la localidad del usuario (usando join).

**1.0 – Consulta 8 -** DNI, nombre, apellidos y teléfono de los usuarios que han realizado más de un pedido.

UF1846: Desarrollo de aplicaciones web distribuidas

**Ejercicio SOA - 1** – Realiza una conexión a tu servidor local MySQL, con el usuario creado durante en el examen, y a la base de datos del anterior módulo. – **1 Punto**

**Ejercicio SOA – 2** – Crear una API REST para la tabla alumno con los diferentes métodos CRUD – **Total 8 puntos**

Configurar Express para interpretar objetos JSON en las peticiones. – **1 Punto**

Implementar Ruta /alumno con Método GET con consulta SELECT para todos los registros– **2 puntos**

Implementar Ruta /alumno con Método GET con parámetro ID con consulta SELECT WHERE – **2 Puntos**

Desarrollar una de siguientes Rutas **– 3 - Puntos**

* Ruta /alumno con Método POST
* Ruta /alumno con Método PUT
* Ruta /alumno con Método DELETE

**Ejercicio SOA - 3** – Sincroniza la carpeta de tú examen con la carpeta MF0492\_3 dentro de tu repositorio IFCD0210 – **1 Punto.**