XUNTA DE GALICIA | CONSELLERÍA DE CULTURA, EDUCACIÓN, FORMACIÓN PROFESIONAL E UNIVERSIDADES

Nome e apelidos	DNI/NIE	OPCIÓN EXAME

ORDE DO 28 de Xaneiro de 2022 (DOG DO 31 de Xaneiro de 2022). SERVIZO SECUNDARIA
CORPO: 590 - Profesores de ensino secundario
ESPECIALIDADE: 107 - Informática

PRIMEIRA PROBA – PARTE A

INSTRUCIÓNS

- A duración total desta parte é de 3 horas.
- Entrégase un caderno con dúas opcións de exame.
- O/a aspirante escollerá e realizará unha das dúas opcións de exame propostas, que deberá indicar no recadro correspondente na cabeceira desta páxina e na do caderno de respostas, xunto co seu nome, apelidos e o seu DNI/NIE.
- As follas grampadas non se poden separar.
- De precisar algún folio adicional, este será solicitado a un membro do tribunal, quen o grampará ao resto dos folios.
- As respostas deben realizarse con bolígrafo de tinta negra ou azul, que non se poida borrar. Non se permite o uso de cintas ou líquidos correctores. Non se corrixirá nada a lapis ou con tinta que se poida borrar.
- Non se admiten preguntas. Calquera aclaración ou interpretación do enunciado que o/a aspirante considere oportuno para a súa valoración, deberá indicala no caderno de respostas.
- Non se pode utilizar ningún material auxiliar, e están prohibidos os móbiles, reloxos, mans libres ou calquera outro dispositivo electrónico, que deberán estar **completamente apagados** e gardados nos bolsos debaixo do propio asento.

OPCIÓN A

EXERCICIO 1 (2 puntos)

EXERCICIO 1.1

A federación galega de Taekwondo necesita para o seu sitio *web* unha aplicación en ***PHP*** (versión 7) para as inscricións externas, inscricións en competicións de clubs doutras comunidades ou do estranxeiro. Permitirá o rexistro do adestrador en aberto, logo este fará *login* e o seu usuario quedará gardado nunha variable de sesión que lle permitirá acceder ao formulario de alta e á listaxe dos seus deportistas.

Dando por suposta a seguinte conexión a base de datos, pídese:

```
<?php
try {
    $bd = new PDO('mysql:host=localhost;dbname=taekwondo, 'root', '');

} catch (PDOException $e) {
    print "¡Error!: " . $e->getMessage() . "<br/>";
    die();
}
?>
```

a) **Alta adestrador**: Dada a táboa *adestrador(usuario, contrasenha, nome, apelidos, nomeclub).*

   Escriba un fragmento de código que, empregando a conexión *$db*, insira na táboa *adestrador* todos os datos, supoñendo que chegan por POST cos mesmos nomes que os campos da base de datos.

   - A inserción realizarase protexendo contra unha posible inxección ***SQL***. Razoar a resposta.
   - Comprobarase se hai erro na inserción.
   - Para o contrasinal gardarase un resumo ***SHA1***, non se garda en claro.

b) Escriba un fragmento de código que amose os deportistas relacionados co adestrador identificado pola variable de sesión *usuario*. Mostrar unicamente o campo *nome* usando algunha función nativa de ***PHP*** para protexerse de ataques ***Cross-Site Scripting***. Unicamente se poderá acceder se a variable de sesión *usuario* o permite.

EXERCICIO 1.2

Partindo dunha base de datos denominada *EXAMEN* cunha táboa denominada *USUARIO*, cos seguintes campos:

- *Id_user*: enteiro sen signo. (chave primaria).
- *login*: cadea de 50 caracteres máximo. Obrigatorio.
- *password*: cadea de 8 caracteres. Obrigatorio.

E empregando a conexión:

```
$conexion = mysqli_connect($host.":".$puerto, $user, $pass)
```

Onde:

```
//IPserverMySQL ($host="localhost";) puerto servidor MySQL ($puerto ="3306";)
//Nombre usuario ($user="root";)Password ($pass="abc123.";)
```

A partir dun formulario simple, que se presenta a continuación, cree un *script* ***PHP*** que comprobe se o *login* e a *password* introducidas no formulario se corresponden con algún usuario dos que existen na táboa

USUARIO da base de datos *EXAMEN*. Se non é así, imprima a mensaxe "login ou password INCORRECTA" e lle dea a opción de volvelo tentar (só unha vez mais).

VerificarCredenciales.html

```
<!DOCTYPE html>
<html lang="es">
	<head>
		<meta charset="utf-8">
		<title>Verificar credenciales usuario</title>
	</head>
	<body>
		<h4> Verificar credenciales usuario</h4>
		<form action="VerificarCredenciales.php" method="POST">
			<fieldset>
				<legend>Usuarios</legend>
				<h5>Usuario</h5>
				<input type="text" name="user" placeholder="user">
				<br>
				<h5>Password</h5>
				<input type="password" name="password" placeholder="password">
				<br>
			</fieldset>
			<br>
			<input type="submit" value="Verificar">
		</form>
	</body>
</html>
```

NOTA: Para a realización do exercicio, empregue a extensión ***MySQLi***, que ofrece acceso a todas as características do motor de BD ***MySQL*** e reduce os tempos de espera na execución das sentenzas, e faga as comprobacións que sexan necesarias.

EXERCICIO 1.3

Cree un *script* ***PHP*** que devolva a seguinte saída empregando un *array* asociativo a partires dos datos proporcionados:

Indice: **nombre** Contenido: **Josefa**

Indice: **apellido1** Contenido: **Pérez**

Indice: **apellido2** Contenido: **Rubio**

Indice: **edad** Contenido: **60**

EXERCICIO 2 (2 puntos)

EXERCICIO 2.1

Realice a implementación do método ***cifra***.

O cifrado César é un tipo de cifrado de substitución no que unha letra do texto orixinal é substituída por outra letra que é un número fixo de posicións posteriores no alfabeto.

Pídese cifrar o nome da cadea de cada contacto usando o cifrado César cun desprazamento de 5 posicións. Por exemplo, a 'a' cifrarase coa 'f'.

Supoñer que todos os nomes están en minúsculas e están compostos só polas seguintes letras: *a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z*.

EXERCICIO 2.2

Tras a análise da asignación de *Módulos* e *Profesores* nun centro de formación de ciclos de Informática, atopamos que a información que precisamos é a seguinte:

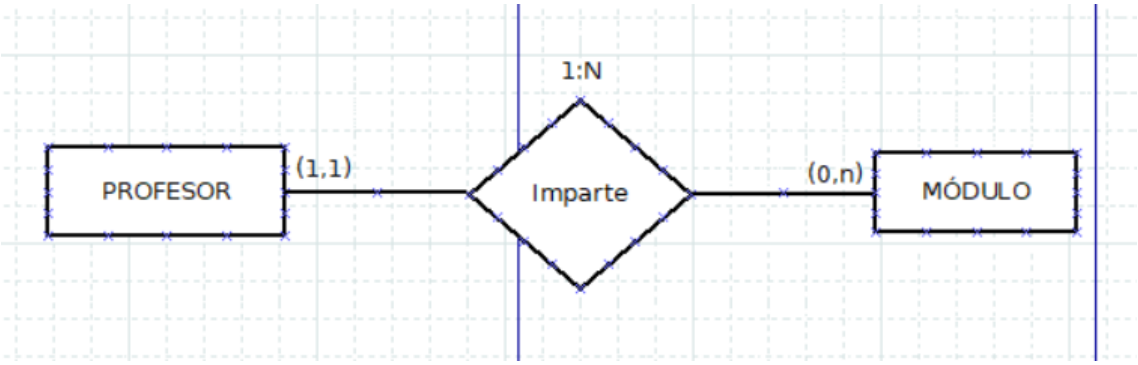
MÓDULOS:

Cod_Módulo	**Título**	**Num_Horas**	**Num_Sesiones**	**Grupo**	**Tutoría**
MP0372	Xestión de Bases de Datos	187	7	1º DAW	No
MP0373-1	Linguaxes de Marcas e Sistemas de Xestión	133	5	1º DAW	Sí
MP0373-2	Linguaxes de Marcas e Sistemas de Xestión	133	5	1º ASIR	Sí
…					

PROFESORES:

Cod_Profesor	**Nombre**	**Apellidos**	**Fecha_Incorporación**
11111111A	Manuel	Rodríguez Pérez	12-09-2010
22222222B	Paula	Martos Méijome	17-09-2016
33333333C	Rebeca	Smith	16-09-2020
…			

E tendo en conta que a relación entre ambas táboas é a seguinte:



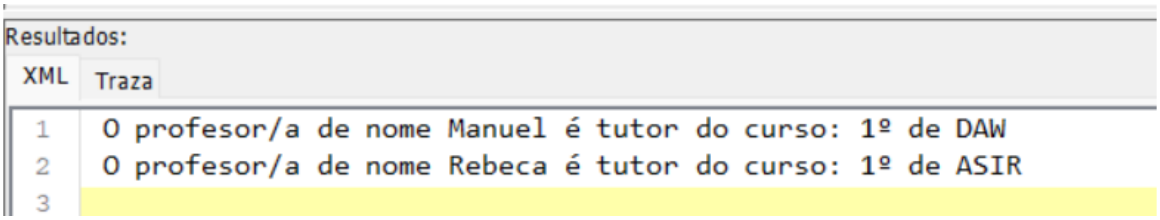
Imaxe 1

a) Defina o esquema ***XML Schema*** dun documento chamado *asignacións*, tendo en conta que :

- O documento chamado *asignacións* deberá ter a capacidade de almacenar toda a información contida nas táboas, polo que será imprescindible empregar os valores das cabeceiras para a definición das etiquetas. O documento deberá explorar as capacidades de aniñamento que posúe ***XML*** para establecer as relacións.
- Os códigos funcionarán como chaves e terán a categoría de atributo.
- Debe seleccionar os tipos o mais axustados posible á información contida nas táboas.
- Os profesores poderán non ter asignado ningún módulo.
- Os profesores poderán ter un ou dos apelidos.
- O número de sesións semanais é como mínimo 2 e como máximo 9.
- Debe definir o *tipocodigo* tendo en conta que o formato do *cod_profesor* sempre se corresponde con 8 díxitos e unha letra maiúscula.
- Debe definir o *tipogrupo* que permitirá manter por separado o curso, 1º o 2º, e o ciclo, DAW o ASIR.

b) Indique a expresión **XPath** que lle permita obter o número de sesións asignadas a un determinado profesor, coñecendo o seu *cod_profesor*.

c) Indique a expresión **XQuery** que lle permita obter un listado dos titores e o seu curso de titoría. A solución deberá ter o formato da seguinte imaxe:



Imaxe 2

EXERCICIO 3 (2 puntos)

Un administrador de sistemas informáticos debe saber explotar os Sistemas Xestores de Bases de Datos. Deberá levar a cabo unha serie de tarefas de administración sobre o sistema de base de datos ***MySQL*** que teñen implementado nunha determinada empresa (sabendo que é unha versión 5.7 ou superior e cun sistema ***Windows***). Conteste ás seguintes cuestións:

Supoña que iniciou sesión como *root* e polo tanto, está traballando co cliente *mysql* a través da liña de comandos.

a) O servidor mantén diferentes variables de estado que proporcionan información sobre as operacións levadas a cabo. Que variable permite coñecer o número de intentos de conexión erróneos ao servidor ***MySQL***?

Supoña que de aquí en adiante estará situado no *prompt* do sistema.

b) O sitio *web* que xestionamos aumentou considerablemente o número de visitas por parte dos usuarios, os cales fan uso das conexións á nosa base de datos ***MySQL*** e debido a iso xéranse tamén erros. Unha posible solución sería aumentar o número de conexións abertas ao mesmo tempo. Indique o comando a executar dende a consola de Windows para establecer o número de conexións en 200 para o noso servidor ***MySQL*** e que ese cambio sexa permanente.

c) Cal é a variable do sistema que permite a un usuario de ***MySQL*** conectarse ao servidor dende calquera dirección IP que non sexa a dirección local? Indique cal é o proceso a seguir para que permita conectarse dende outras direccións IP ao servidor ***MySQL***.

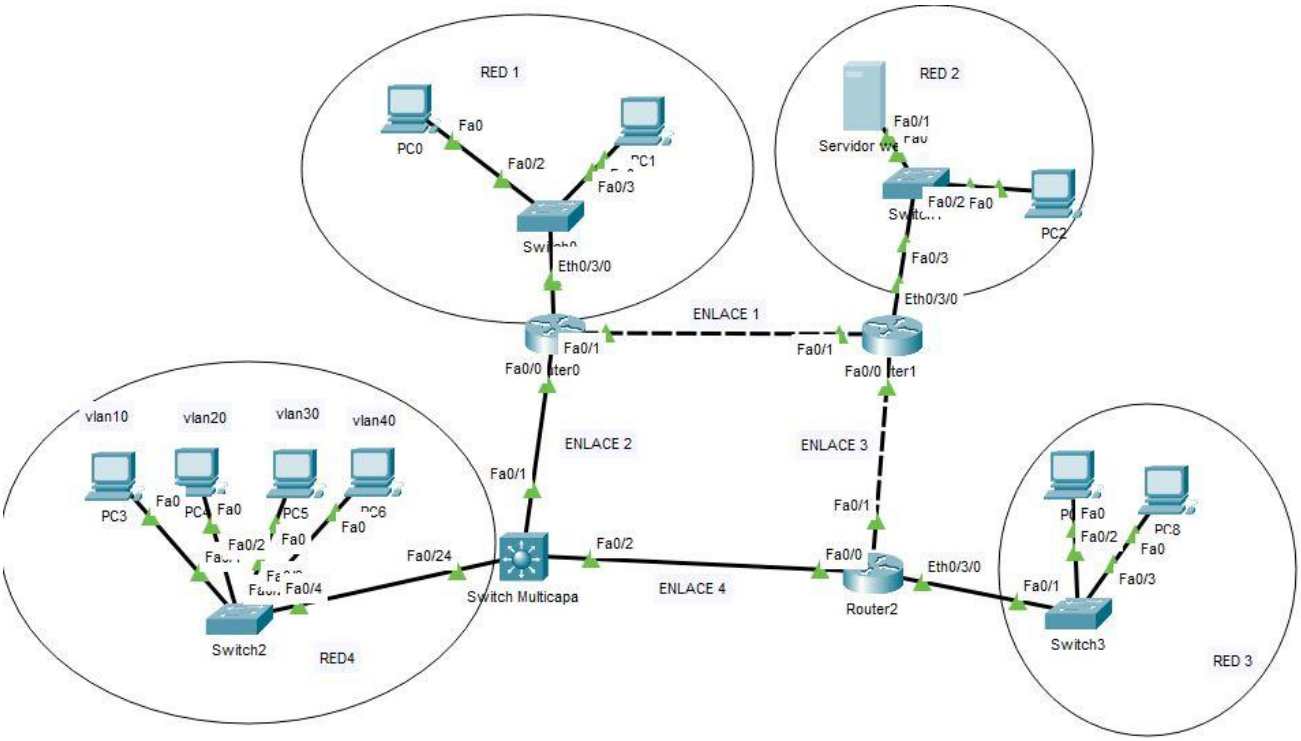
d) Hai que levar a cabo a realización dunha copia de seguridade das táboas *Cliente* e *Proveedor* que pertencen á base de datos *Suministros* e que se gardarán nun arquivo chamado "*Copia_003. sql*". Escriba a sintaxe correcta para realizar dita copia.

e) Cree un novo usuario "*invitado*", de uso temporal, para acceder ao servidor. Dito usuario só terá acceso dende o equipo con IP 192.168.100.100, e chave "*invitado01*" que debe cambiar cada 15 días. Ademais, habilite o seguimento dos inicios de sesión erróneos que tivese, de tal xeito que se teñen lugar 3 intentos de acceso incorrectos bloquearase temporalmente a conta durante 2 días.

f) O usuario "*invitado"* terá permisos para seleccionar datos e inserir rexistros na táboa *Cliente*. Na táboa *Proveedor* só poderá modificar os campos *telefono* e *direccion*, e ver o *nome*. Ambas táboas pertencen á base de datos *Suministros*.

g) Cree unha vista chamada "*v1*" que permita seleccionar todos os rexistros da táboa *Cliente* onde a *deuda* sexa inferior a 150€, de xeito que as inserción que se fagan posteriormente a través desta vista, deberán ser verificadas previamente.

EXERCICIO 4 (2 puntos)



Imaxe 3

Na topoloxía da *Imaxe 3* existen 4 redes locais que se desexa conectar entre si mediante *routers* e un *switch multicapa,* distribuídos segundo o esquema, partindo das seguintes consideracións:

- Para asignar direccións a todo o esquema, vaise empregar a rede que contén á IP 10.129.0.0/12.
- A rede local conectada ao *Router0*, *RED1*, poderá albergar ata 1020 dispositivos.
- A rede local conectada ao *Router1, RED2,* deberá ter unha capacidade de ata 2046 dispositivos.
- *RED3*, conectada ao *Router2* deberá ter capacidade para 508 dispositivos.
- *RED4*, conectada ao ***switch multicapa*** configúrase inicialmente para que poida abarcar 200 dispositivos.
- Na *RED4*, decídese crear 4 ***VLAN*** de igual tamaño que repartan por completo e a partes iguais dita rede.

a) Pídese identificar as direccións *IPs* e máscaras de todos os *routers* e *switch multicapa* (interfaces virtuais asociados á interface física das *VLANs*) tendo en conta que serán as últimas *IPs* dispoñibles das subredes calculadas. Para iso, realice a división en subredes para o direccionamento de todas as redes tendo en conta os seguintes criterios:

- O rango de direccións que queden libres despois de asignar os direccionamentos debe ser o maior posible.
- Débese axustar o tamaño da rede ao mais pequeno posible para o número de equipos empregados.
- Teña en conta que as portas de enlace precisan dunha *IP*.

- Os enlaces entre os *routers* unicamente deben permitir dúas *IPs* e deben empregar o rango de direccións seguinte ao da derradeira subrede calculada das redes *RED1, RED2, RED3*, e *RED4*. Débense elixir nesta orde, o *enlace1*, *enlace2*, *enlace3* e *enlace4.*
- Débese cubrir unha táboa como a seguinte, para todas as interfaces:

Interface	Dirección IP	Máscara
Router0 Eth0/3/0		
[…]		

Debe solucionar as seguintes tarefas cos comandos oportunos ***CLI***, que se deberán escribir completos (sen abreviar).

b) Supoñendo o resto do esquema configurado correctamente, configure o *switch multicapa* para que as 4 ***VLAN*** poidan saír ao exterior. Para este apartado supoña as seguintes redes.

- *VLAN10*: 10.0.0.0/26
- *VLAN20*: 10.0.0.64/26
- *VLAN30*: 10.0.0.128/26
- *VLAN40*: 10.0.0.192/26.

Empregue a derradeira *IP* das ***VLAN*** como porta de enlace. Supoña que o *switch multicapa* xa ten configurados os enlaces que o unen co *Router0* e *Router2* e o encamiñamento axeitado.

c) Configure NAT dinámico na rede *RED1* conectada ao *Router0*, de tal xeito que os equipos se conecten ao resto das redes cunha dirección entre 10.0.0.90-10.0.0.95/26, independentemente da interface pola que saian os datos. Supoña o resto da configuración xa realizada correctamente e compatible co direccionamento solicitado no exercicio. Supoña que os equipos da *RED1* están na rede 10.0.0.0/26

d) Deséxase mellorar a seguridade da rede mediante unha ***ACL*** que non permita facer *ping* dende a *RED1* a ningún equipo da *RED2* pero si o permite da *RED2* á *RED1*.

Os equipos da *RED1* unicamente poderán conectarse ao servidor *web* (10.0.128.2) vía ***HTTPS*** e a ningún outro porto dese equipo nin ningún equipo doutra rede.

Os equipos da *RED1* poderán conectarse ao resto de equipos da *RED2* empregando portos ***TCP*** menores de 1024 e ningún mais.

O resto do tráfico debe estar bloqueado.

Escriba a ***ACL*** e asígnea á interface e *router* axeitado. Supoña o resto da configuración xa realizada e para este exercicio tome como *RED1* a rede 10.0.0.0/19 e para a *RED2* a rede 10.0.128.0/19.

EXERCICIO 5

EXERCICIO 5.1

Contamos na nosa rede local con varios servizos *web* aos que se pode acceder mediante o protocolo ***HTTP***, entre os que se atopa o seguinte:

Servizo	IP	Porto de escoita do servizo
www.ies.gal	192.168.0.33	8080

Deséxase instalar e configurar un ***proxy* inverso** con ***Nginx*** nun servidor con dirección IP 192.168.0.33/24. Este *proxy* inverso proporcionará seguridade aos nosos servizos ao permitir acceder a estes mediante o protocolo ***HTTPS***.

O servidor ***DNS*** empregado na rede resolve os nomes de todos os servizos *web* coa dirección *IP* 192.168.0.33/24, servidor onde está instalado o *proxy* inverso.

Para poñer en funcionamento o protocolo ***HTTPS***, contamos ademais cos seguintes ficheiros no servidor:

O certificado ***SSL****: /etc/nginx/cert.crt*

A chave privada do certificado ***SSL***: */etc/nginx/cert.key*

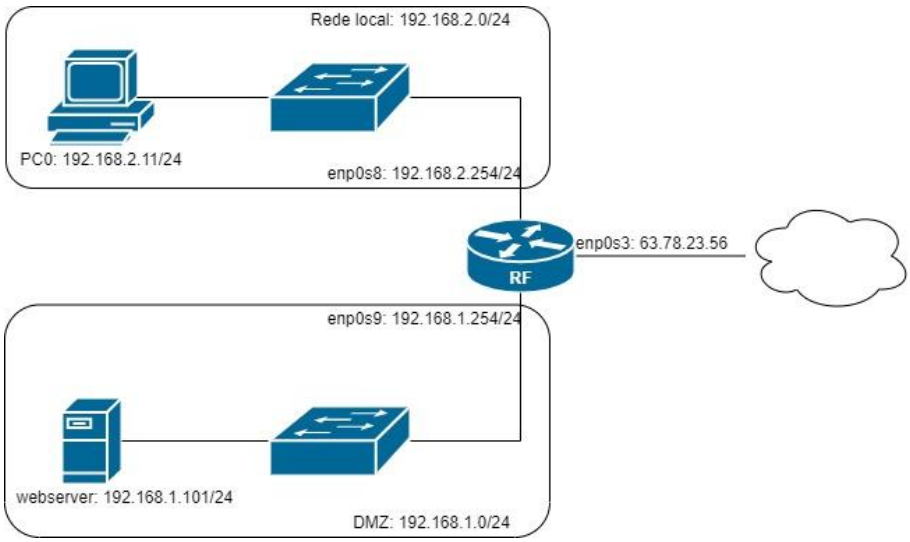
Engada ao ficheiro */etc/nginx/conf.d/default.conf* as liñas necesarias para configurar o *proxy* inverso para o servizo **www.ies.gal**.

NOTA: Código auxiliar para utilizar na configuración do *proxy* inverso.

```
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

EXERCICIO 5.2

Temos o seguinte diagrama de rede:



Imaxe 4

O *router* de fronteira **RF** é unha máquina con ***Debian 11***. Este debe proporcionar saída a Internet aos equipos das redes 192.168.1.0/24 (conectada a través da interface *enp0s9*) e 192.168.2.0/24 (conectada a través da interface *enp0s8*) a través da súa interface de rede *enp0s3* conectada á rede pública.

Configure o *router* mediante ***IPTables***. Este debe proporcionar as seguintes funcionalidades:

- Por defecto denegarase todo o tráfico agás as excepcións descritas nos restantes puntos.
- Permitir a saída á rede pública aos equipos das redes conectadas polas interfaces *enp0s8* e *enp0s9* mediante ***NAT***.
- Permitir dende a rede 192.168.2.0/24 conectar ao *router* de fronteira mediante ***SSH***.
- As redes 192.168.1.0/24 e 192.168.2.0/24 tan só poden acceder a Internet utilizando os protocolos ***HTTP***, ***HTTPS*** e ***DNS***.
- Permitir dende a rede 192.168.2.0/24 o acceso ao servidor *web* (só a este equipo, non a toda a rede) unicamente polo protocolo ***HTTP***.
- As peticións feitas dende Internet ao porto 443 do *router* deben redirixirse ao servidor *web* cara a ese mesmo porto (débese engadir unha entrada que permita ese tráfico).

O *router* de fronteira en ***Debian*** xa ten correctamente configuradas as súas interfaces de rede e ten activado o encamiñamento de paquetes.

Completa o seguinte *script* coa configuración de ***IPtables***:

NOTA: non é necesario utilizar estruturas condicionais e iterativas. Tan só engadir os comandos necesarios secuencialmente para realizar o exercicio.

```
#!/bin/bash

############ Borramos todas as regras existentes
iptables -t filter -F
iptables -t nat -F

########### Politicas por defecto
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Completar...

# Gardamos as regras ante un posible reinicio
iptables-save > /etc/iptables/rules.v4
```

OPCIÓN B

EXERCICIO 1 (2 puntos)

A partires do formulario de busca ***HTML*** proporcionado (*Imaxe 1*) e **SEN MODIFICALO**, engade o código Javascript puro (sen librarías externas) necesario para implementar os seguintes apartados:

a) Modificar a aparencia dos campos de texto, de tal xeito que a súa cor de fondo se volva gris claro unicamente mentres o foco se atope en cada un deles.

b) Evitar que nos campos de texto se introduzan caracteres que non sexan alfanuméricos.

c) Unha vez se cubran os datos no formulario, deberanse enviar mediante petición AJAX, empregando ***XMLHttpRequest*** ao servicio ***REST*** "*https://searchusers.me/api/*". A resposta obtida terá a estrutura da *Imaxe 2*, en formato JSON. Ningún dos parámetros de busca será obrigatorio. A petición ***AJAX*** realizarase nunha función *pedirDatos()* que reciba como parámetros tres cadeas de texto que se correspondan cos campos de busca do formulario.

d) Ao recuperar os datos da chamada ***AJAX***, deberanse presentar en formato táboa (na táboa "*resultados*") empregando funcións de árbore ***DOM***, onde existirá unha fila de 3 columnas para cada usuario: unha columna para o seu nome e apelido, e outra co seu *email*. A táboa resultante debería quedar como a da *Imaxe 3*.

***Nota**: Non é necesario traducir o valor dos campos*

```html
<html>
    <head>
        <style>
            body {
                font-family: arial, helvetica;
                }
            table, tr, td {
              table-layout: fixed;
              width: 50%;
              border-collapse: collapse;
              border: 3px solid purple;
            }
        </style>
        <script src="https://code.jquery.com/jquery-3.3.1.js"></script>
        <script>
            [... COMPLETAR CON JAVASCRIPT...]
        </script>
    </head>
    <body>
        <form id="formulario">
            <p>Nombre: <input type="text" id="nombre" size="40"></p>
            <p>Apellido: <input type="text" id="apellido" size="50"></p>
            <p>Sexo:
                <input type="radio" name="hm" value="h"> Hombre
                <input type="radio" name="hm" value="m" checked> Mujer
            </p>
            <p>
                <input type="button" id="btn_submit" value="Enviar" onclick="enviarForm();">
                <input type="reset" value="Borrar">
            </p>
        </form>
        <table id="resultados">
        </table>
    </body>
</html>
```

Imaxe 1: Formulario HTML

```
{
    "results": [
        {
            "gender": "female",
            "name": {
                "title": "Miss",
                "first": "Catalina",
                "last": "Herrera"
            },
            "location": {
                "city": "Hospitalet de Llobregat",
                "state": "Región de Murcia",
                "country": "Spain"
            },
            "email": "catalina.herrera@example.com",
            "phone": "937-052-026",
            "id": {
                "name": "DNI",
                "value": "01958256-O"
            }
        }
    ]
}
```


Imaxe 2: Resposta servicio REST en formato JSON

Nombre:

Apellido:

Sexo: Hombre Mujer

Enviar Borrar

Josefine Mortensen	josefine.mortensen@example.com
Alicia Dumas	alicia.dumas@example.com
Nanna Nielsen	nanna.nielsen@example.com

Imaxe 3: Táboa de resultados

EXERCICIO 2 (2 puntos)

EXERCICIO 2.1

Implemente en ***Python*** (versións 3.7 ata 3.10) a clase **Tarxeta**. Non pode usarse ningunha libraría non estándar.

A clase **Tarxeta** terá os atributos *numero* de tipo *str* e *valida_ata* representa unha data *(so dia, mes e ano)*. A clase debe ser **inmutable**, de tal xeito que unha vez construída non sexa posible cambiar o valor dos seus atributos nin engadir novos.

Para construír unha instancia da clase **Tarxeta** esta debe validar:

- Que o número proporcionado (en forma de cadea de texto) ten 16 díxitos numéricos e cumpre coa fórmula de *Lunh* (ver extracto da Wikipedia adxunto na *Imaxe 4*).
- Que o atributo *valida_ata* sexa posterior á data do mesmo momento en que se está a crear a instancia.
- Se non se cumpre calquera destes requisitos a creación da instancia debe producir unha excepción, do tipo que se crea conveniente se é xerada por vostede.

La fórmula verifica un número contra su dígito de chequeo incluido, el cual es usualmente agregado a un número de cuenta parcial para generar el número de cuenta completo. Este número de cuenta debe pasar la siguiente prueba:

1. A partir del dígito de chequeo incluido, el cual está a la derecha de todo el número, ir de derecha a izquierda duplicando cada segundo dígito.
2. Sumar los dígitos del resultado: (ejemplo: 10 = 1 + 0 = 1, 14 = 1 + 4 = 5) juntos con los dígitos sin duplicar del número original.
3. Si el total de módulo 10 es igual a 0 (si el total termina en cero), entonces el número es válido de acuerdo con la fórmula Luhn, de lo contrario no es válido.

Supongamos un ejemplo de un número de cuenta "7992739871", que contará con un dígito de control adicional, por lo que es de la forma 7992739871x:

Dígitos del número de cuenta	7	9	9	2	7	3	9	8	7	1	x
Duplicar dígitos pares	7	18	9	4	7	6	9	16	7	2	x
Sumar los dígitos	7	1+8	9	4	7	6	9	1+6	7	2	=67

Imaxe 4: Algoritmo de Lunh. Fonte Wikipedia. https://es.wikipedia.org/wiki/Algoritmo_de_Luhn

Un exemplo do código que faría uso da clase pedida neste exercicio debería producir todos os resultados coa cadea *Ben*.

```
if __name__ == "__main__":
    # Obxecto cun número válido e unha data no futuro se crea correctamente
    t = Tarxeta("1173551919182361", datetime.date(2023, 12, 1))
    print("Creacion de tarxeta correcta: Ben")
    try:
        # Obxecto Tarxeta cun número inválido
       t = Tarxeta("4036587647758561", datetime.date(2023, 12, 1))
        print("Creación numero inválido: Mal")
    except Exception:
        print Creación numero inválido a: Ben")
```

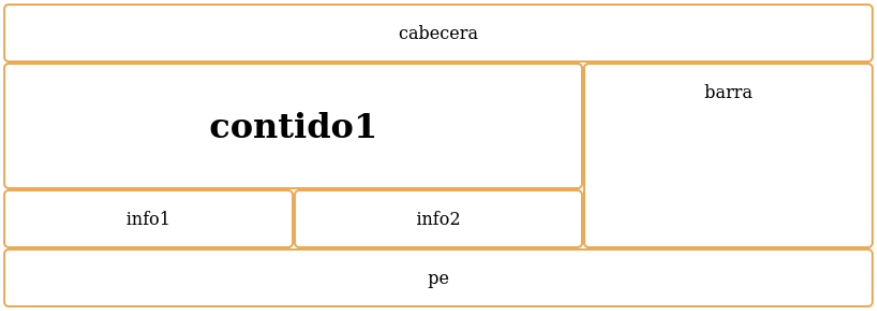
```
try:
    # Obxecto cunha data inválida
    t = Tarxeta("4036587647758561", datetime.date(2021, 12, 1))
    print("Creación con data non válida: Mal")
except Exception:
    print("Creación con data non válida: Ben")

try:
    # Modificar valor do atributo: exception
    t.numero = "4056589647758561"
    print("Modificación atributos: Mal")
except Exception:
    print("Modificación atributos: Ben")
```

EXERCICIO 2.2

Tomando como punto de referencia e partida o seguinte arquivo ***HTML***:

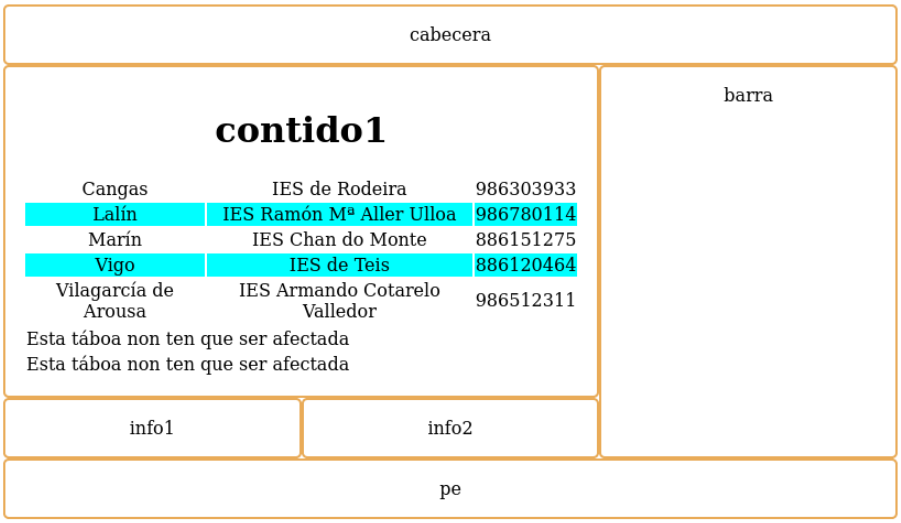
```
<!doctype html>
<html lang=es><head><meta charset=utf-8><title>Oposicion 2022</title>
    <link rel="stylesheet" href="styles.css"></head>
<body>
    <div class="wrapper">
      <div class="contido1">
        <h1>contido1</h1>
        <table> <tbody>
          <tr><td>Cangas</td><td>IES de Rodeira</td><td>986303933</td></tr>
          <tr><td>Lalín</td><td>IES Ramón Mª Aller Ulloa</td><td>986780114</td></tr>
          <tr><td>Marín</td><td>IES Chan do Monte</td><td>886151275</td></tr>
          <tr><td>Vigo</td><td>IES de Teis</td><td>886120464</td></tr>
          <tr><td>Vilagarcía </td><td>IES Armando Cotarelo
Valledor</td><td>986512311</td></tr>
        </tbody> </table>
        <table> <tbody>
            <tr><td>Esta táboa non ten que ser afectada</td></tr>
            <tr><td>Esta táboa non ten que ser afectada</td></tr>
          </tbody> </table>
      </div>
      <div class="info1">info1</div>
      <div class="info2">info2</div>
      <div class="cabecera">cabecera</div>
      <div class="barra">barra</div>
      <div class="pe">pe</div>
    </div>
```

a) Engada as regras ***CSS*** necesarias ao ficheiro *styles.css* para que a disposición dos seguintes 6 *div* (*cabecera, contido1, barra, info1, info2, p2*) quede coma na *Imaxe 5*. Teña en conta que os bloques *info1, info2* e *barra* teñen o mesmo ancho. O alto de cada un dos bloques variará dependendo do seu contido. Os bordes dos bloques son para que sirvan de orientación, non hai que introducilos.



Imaxe 5

b) Engada as regras ***CSS*** necesarias ao ficheiro *styles.css* para que as filas pares da primeira das táboas que se atopan dentro do *div* de clase *contido1* teñan de color de fondo *cian*, tal como se amosa na *Imaxe 6*.



Imaxe 6

EXERCICIO 3 (2 puntos)

EXERCICIO 3.1

Dado o esquema da base de datos **empregados,** sobre os empregados e os departamentos dunha determinada empresa.

emp(emp_num, emp_nome, emp_posto, emp_xefe, emp_dataing, emp_salario, emp_comm, emp_depnum)

dep(dep_num, dep_nome, dep_loc)

a) Realice as consultas ***SQL*** necesarias para levar a cabo as seguintes accións. Deberanse realizar nunha única consulta:
   - Obter os nomes, salarios e comisións dos empregados que perciben un salario situado entre a metade do máximo salario e a media dos salarios.
   - Obter os empregados que non teñen ningún outro ao seu cargo.
   - Obter o nome dos empregados e o nome do departamento no que traballan. Deberanse obter tamén os nomes dos empregados que non teñen asignado departamento, neste caso debe saír a mensaxe: ´Empregado sen departamento´, e tamén os departamentos que non teñen asignados traballadores, neste caso debe saír a mensaxe: ´Departamento sen traballadores´.

b) Cree un ***trigger*** que se dispare nas insercióǹs na táboa *emp* e non permita engadir un empregado con soldo superior que o do seu xefe.

EXERCICIO 3.2

Dado o seguinte fragmento do ficheiro ***JSON*** no que se atopa definida a colección *libraría*:

```
{
    "_id": "5e84af3a20ecc186bd78f815",
    "index": 0,
    "titulo": "O lapis do carpinteiro",
    "activo": true,
    "coste": 18,
    "autor": {
      "nombre": "Manuel",
      "apellido": "Rivas"
    },
    "editorial": "Xerais",
    "publicacion" :1998
  }, {
    "_id": "5e84af3ae7c2629840994b01",
    "index": 1,
    "titulo": "La comunidad del anillo",
    "activo": true,
    "coste": 25,
    "autor": {
      "nombre": "J.R.",
      "apellido": "Tolkien"
    },
    "editorial": "XUMONK",
    "publicacion" :1937
  }
```

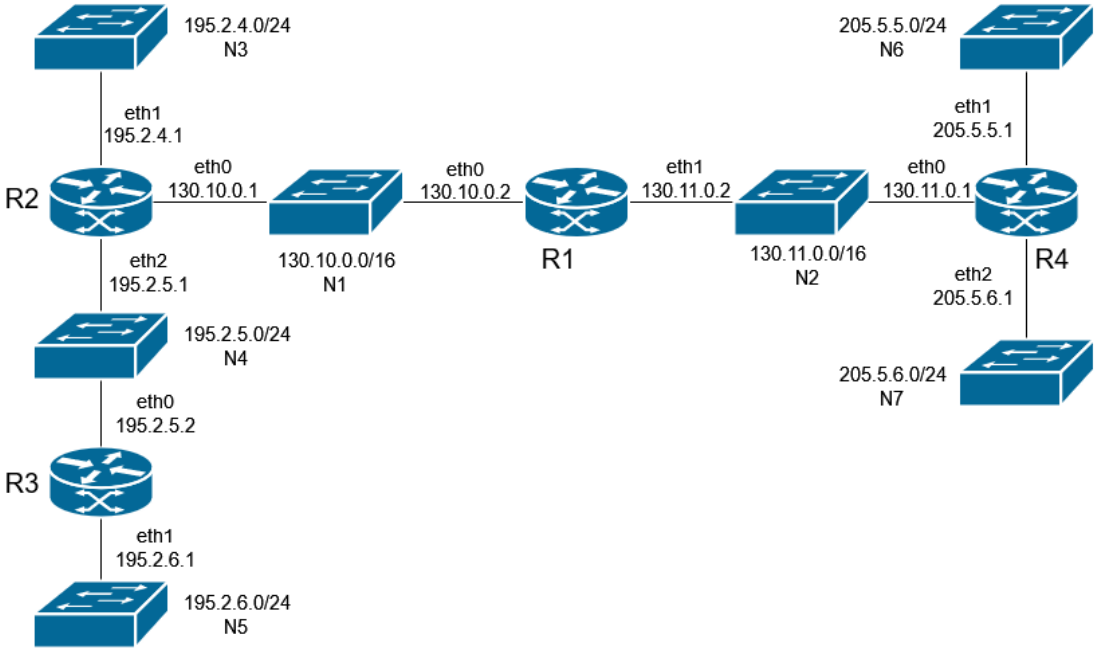
Indique a continuación as sentenzas en ***MongoDB*** para:

a) Obter os nomes dos autores con apelido *Tolkien*.
b) Obter os libros publicados en *1998* ou en *2006*.
c) Obter o libro que ocupa a segunda posición.

EXERCICIO 4 (2 puntos)

EXERCICIO 4.1

Dada a seguinte imaxe (*Imaxe 7*)



Imaxe 7

a) Indique cal sería a táboa de encamiñamento do *router* **R1** empregando a seguinte táboa:

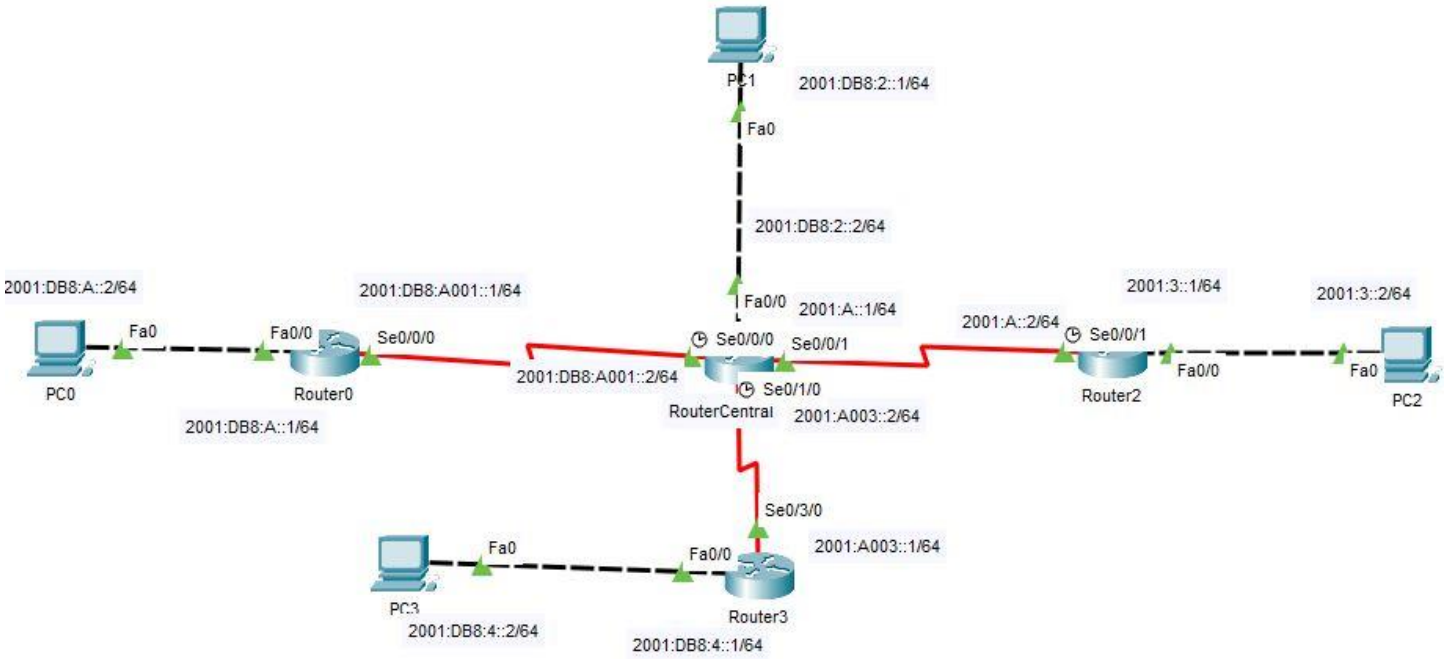
Dirección rede	Máscara	Seguinte salto	Interfaz	Nº saltos

b) A continuación, supoñendo o ficheiro de configuración de ***network/interfaces*** dunha distribución ***Ubuntu 18.02***, indique cal sería o contido do mesmo para o *router* **R2,** tendo en conta que se deberá especificar tanto a configuración das tarxetas de rede, como a configuración do encamiñamento.

Para os nomes das interfaces empregue os asignados no diagrama.

EXERCICIO 4.2

Dado o seguinte esquema (*Imaxe 8*), responda ás seguintes cuestións indicando tódolos comandos ***CLI*** completos (sen abreviar).



Imaxe 8

a) Supoña que na topoloxía da *Imaxe 8* se está empregando RIPng e que o direccionamento ***IPv6*** está configurado correctamente en todos os *routers*. Supoña tamén que ***RIPng*** está configurado correctamente nos *Router0, Router2 e Router 3*. Configure ***RIPng*** no *RouterCentral* para que todos os equipos do esquema teñan conectividade entre eles.

b) Supoña que o administrador de rede decide cambiar o encamiñamento dinámico por encamiñamento estático. Asumindo todo o direccionamento e encamiñamento estático ***IPv6*** correctamente configurado en *Router0, RouterCentral* e *Router2*. Pídese configurar o ***Router3*** para que todos os equipos do esquema teñan conectividade entre eles.

c) Asumindo o encamiñamento estático do apartado anterior correctamente configurado, decídese engadir a rede 2001:DB8:A004::/64 que unirá *Router3* con *Router0*. Engada os comandos ***CLI*** necesarios para que ese enlace sexa unha ruta que permita a comunicación unicamente en caso de caída do enlace do *Router3* co *RouterCentral*. Asuma que a nova interfaz do *Router3* é *Serial 0/3/1.*

EXERCICIO 5 (2 puntos)

EXERCICIO 5.1

Engádense 3 discos a un equipo o cal ten instalado previamente un sistema operativo ***Ubuntu*** ou ***Debian***, mediante o seu administrador de discos.

a) Supoñendo que a máquina dispón orixinalmente dun único disco, */dev/sda*, os novos discos engadidos pasarán a ser identificados polo sistema como */dev/sdb, /dev/sdc* e */dev/sdd*. Co software de xestión de volumes ***RAID*** **mdadm,** configure o seguinte volume ***RAID*** (supoñendo que o nome da máquina é *mv1*).

b) Unha vez creado o ***RAID***, cál é o comando para eliminar algún dos discos do ***RAID***?

Tipo RAID	Dispositivo RAID	Número de discos	Nome de discos	Punto montaxe
RAID5	*/dev/md/mv1:raid5*	3	*/dev/sdb, /dev/sdc, /dev/sdd*	*/mnt/miraid5*

EXERCICIO 5.2

Indique que comandos terá que executar cada un dos usuarios que interveñen na comunicación (*empregado1, empregado2* e *empregado3*) no suposto caso de que o ***empregado3*** queira enviar de maneira **confidencial** un arquivo chamado "***datos-privados-3.pdf***" aos outros dous usuarios *empregado1* e *empregado2*. Ningún dos tres usuarios dispón de chaves asimétricas, polo que os usuarios que as requiran, e só aqueles que as requiran, deberán xeralas. Para a distribución das chaves farase uso dun servidor de chaves públicas ***PGP*** "*pgp.rediris.es*".

Supoña que os identificadores das chaves asimétricas xeradas polos empregados *empregado1* y *empregado2* tras xerar as chaves son *1111111A* e *2222222B*, respectivamente.

Especifique que comandos executa cada usuario na súa sesión.

Emisor	Receptores	Distribución de chaves públicas	Documento a enviar cifrado	Tipo Criptografía
empregado3	*empregado1* *empregado2*	*pgp.rediris.es*	*datos-privados-3.pdf*	Asimétrica - GPG

EXERCICIO 5.3

Deséxase despregar o sitio *web* seguro **bolechas.com** utilizando contedores ***Docker***. Contamos cos seguintes ficheiros necesarios para a configuración deste:

- Contido do sitio *web*: *index.html*
- Certificado ***SSL***: *bolechas.pem*
- Chave privada do certificado ***SSL***: *bolechas.key*

O contedor ***Docker*** a crear debe de respectar os seguintes requisitos:

- Partir da imaxe ***Debian***.
- Utilizar como servidor *web* ***Apache 2***.
- Realizar redirección do protocolo inseguro ***HTTP*** cara o protocolo seguro ***HTTPS***.

a) Cree o ficheiro de definición do *Virtualhost* para o sitio **bolechas.com**. Este utilizarase na configuración de ***Apache*** do contedor de ***Docker***. O nome do ficheiro debe ser *bolechas.conf*.

   NOTA: proporciónase esta directiva que pode ser de utilidade na definición do V*irtualhost*:

   ```
   RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=302]
   ```

b) Complete o ficheiro ***Dockerfile*** para a creación da imaxe do sitio *web* **bolechas.com**.

   NOTA: Considere que os ficheiros especificados no enunciado e o de definición do *Virtualhost* atópanse no mesmo directorio que o ***Dockerfile***.

   ```
   FROM debian

   # Completar ...

   ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
   ```

Indique o comando para crear un contedor a partir dunha imaxe denominada **bolechas** creada a partir do ***Dockerfile*** do apartado anterior. O nome do contedor debe ser **bolechas_web**