

**OPCIÓN A**

DNI:	
------	--

ORDE do 31 de xaneiro de 2024 (DOG do 14 de Febreiro de 2024). SERVIZO SECUNDARIA  
CORPO: 590 - Profesores de ensino secundario  
ESPECIALIDADE: 107 – Informática

**INSTRUCCIÓNS**

- A duración total desta parte é de 3 horas.
- Proporcionanse 5 cadernos. Todos deben entregarse ao rematar a proba:
  - Opción A.
  - Opción B.
  - Caderno de Respostas Opción A.
  - Caderno de Respostas Opción B.
  - Follas en branco (non se correxirán).
- A persoa aspirante escollerá e realizará unha das dúas opcións de exame propostas e responderá no caderno de respostas correspondente a dita opción.
- A persoa aspirante indicará o seu DNI no recadro da opción escollida e no caderno de respostas da opción escollida.
- 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 corraxirá nada a lapis ou con tinta que se poida borrar.
- Non se admiten preguntas. Calquera aclaración ou interpretación do enunciado que a persoa aspirante considere oportuno para a súa valoración, deberá indicala no caderno de respostas.
- Todos os dispositivos móbiles deberán estar apagados ao igual que reloxos dixitais ou dispositivos análogos. Deberán estar metidos no sobre que o tribunal proporciona para tal fin.



## Exercicio 1 (2 puntos)

Quérese desenvolver un sistema de xestión da oferta formativa de Formación Profesional (FP) en Galicia utilizando a linguaxe de programación **Python**. O programa contén as seguintes clases:

- **CentroEducativo**: representa un centro educativo cos atributos: nome do centro, provincia, concello, ciclo formativo que se imparte e número de prazas dese ciclo formativo.
- **CicloFormativo**: representa un ciclo formativo cos atributos: nome do ciclo, familia profesional e grao (básico, medio, superior ou curso especialización).
- **XestionFP**: xestiona a oferta formativa, permitindo engadir novos centros educativos e ciclos, e realizar algunha análise estatística sobre a oferta formativa de FP.

Dentro destas clases **completa o código das funcións incompletas no correspondente apartado do caderno de respostas** para realizar o seguinte:

### Apartado 1.1 (0.5 puntos)

- (función `listar_centros(self)`): mostrar por pantalla todos os centros educativos de Galicia que imparten FP. A función debe imprimir en cada liña toda a información de cada centro (nome do centro educativo, provincia, concello, ciclo formativo que imparte, familia profesional, grao e número de prazas ofertadas). É preciso completar as funcións `__str__` das dúas clases **CicloFormativo** e **CentroEducativo** para dito obxectivo.

A continuación, móstrase un exemplo de resultado por pantalla de dita función supoñendo que se engadiron previamente os tres centros A, B e C, cunha oferta de 3 ciclos de FP entre todos os centros, 1 mesmo ciclo impartido en dous centros diferentes:

Centro A - A Coruña - Ribeira - Desenvolvemento de Aplicacións Multiplataforma (Informática, Grao superior) - 30 prazas

Centro B - A Coruña - Santiago de Compostela - Xestión Administrativa (Administración, Grao medio) - 60 prazas

Centro A - A Coruña - Ribeira - Informática e comunicacións (Informática, Grao básico) - 30 prazas

Centro C - Ourense - Ourense - Desenvolvemento de Aplicacións Multiplataforma (Informática, Grao superior) - 60 prazas

### Apartado 1.2 (0,75 puntos)

- (función `buscar_por_concello(self, concello)`): buscar os centros educativos con formación profesional nun concello específico. O parámetro de entrada é unha cadea de texto co concello e o parámetro de saída debe ser unha lista con toda a información dos centros educativos con FP dese concello. Dita información na lista debe incluír nome do centro, provincia, concello, nome do ciclo, familia, grao e prazas.

### Apartado 1.3 (0,75 puntos)

- (función `prazas_por_familia(self)`): debe devolver un dicionario en Python no que cada entrada sexa o nome da familia profesional (clave) e o número de prazas ofertadas nesa familia profesional (valor).

Na función `__main__` da clase `XestionFP` engádense todos os ciclos de FP, os centros educativos e as chamadas ás funcións correspondentes. Indícanse en comentarios as posibles tarefas a realizar en dita función, pero non é preciso completar nada nesa función neste exame.

```
class CicloFormativo:
    def __init__(self, nome_ciclo, familia_profesional, grao):
        """ Inicializa unha nova instancia de CicloFormativo.
        :param nome_ciclo: Nome do ciclo formativo.
        :param familia: Familia profesional do ciclo formativo.
        :param grao: Grao do ciclo (básico, medio, superior ou curso
especialización)."""
        self.nome_ciclo = nome_ciclo
        self.familia_profesional = familia_profesional
        self.grao = grao

    def __str__(self):
        """ Devolve unha representación en cadea do ciclo formativo. """
        #A COMPLETAR

class CentroEducativo:
    def __init__(self, nome_centro, provincia, concello, ciclo_formativo, prazas):
        """ Inicializa unha nova instancia de CentroEducativo.
        :param nome: Nome do centro educativo.
        :param provincia: Provincia na que se atopa o centro.
        :param concello: Concello no que se atopa o centro.
        :param ciclo_formativo: Instancia de CicloFormativo ofrecida polo centro.
        :param prazas: Número de prazas ofertadas. """
        self.nome_centro = nome_centro
        self.provincia = provincia
        self.concello = concello
        self.ciclo_formativo = ciclo_formativo
        self.prazas = prazas

    def __str__(self):
        """ Devolve unha representación en cadea do centro educativo. """
        #A COMPLETAR

class XestionFP:
    def __init__(self):
        """ Inicializa unha nova instancia de XestionFP. """
        self.centros = []

    def engadir_centro(self, centro):
        """ Engade un centro educativo á lista de centros.
        :param centro: Instancia de CentroEducativo a engadir. """
        self.centros.append(centro)
        print(f"Centro engadido: {centro}")

    def listar_centros(self):
        """ Imprime por pantalla todos os centros educativos """
        #A COMPLETAR

    def buscar_por_concello(self, concello):
        """ Busca centros educativos nun concello específico.
        :param concello: Nome do concello.
        :return: Lista de centros educativos no concello especificado """
        #A COMPLETAR
```

```

def prazas_por_familia(self):
    """ Calcula o número total de prazas ofertadas por cada familia
profesional.
    :return: Dicionario coa familia profesional como clave e o número de prazas
como valor."""
    #A COMPLETAR

if __name__ == '__main__':
    # Código Principal
    xestion = XestionFP()
    # Engadir ciclos formativos de exemplo
    # Engadir centros educativos de exemplo
    # Listar todos os centros educativos
    # Solicitar un concello por teclado e buscar centros por concello
    # Buscar prazas ofertadas por familia profesional

```

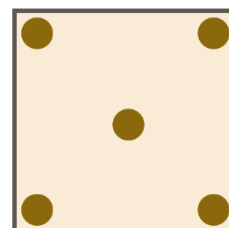
## Exercicio 2 (2 puntos)

Partindo do documento HTML dado a continuación, e sen modificalo, crear **un documento CSS (no correspondente espazo do caderno de respostas)** que faga que ao abrir este HTML nun navegador se vexa a imaxe achegada (similar á cara do 5 dun dado). É obrigatorio usar **CSS-Grid** para conseguilo. O tamaño dos elementos non é relevante pero usaranse sempre unidades de medida relativas. Escoller as cores de fondo e bordes que se considere.

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>DADO</title>
    <link rel="stylesheet" href="estilo.css">
</head>
<body>
    <div class="caixa">
        <div class="circulo circulo1"></div>
        <div class="circulo circulo2"></div>
        <div class="circulo circulo3"></div>
        <div class="circulo circulo4"></div>
        <div class="circulo circulo5"></div>
    </div>
</body>
</html>

```



## Exercicio 3 (2 puntos)

Dada unha aplicación web en **PHP** que simule unha tenda de galletas, debe permitir aos usuarios ver os produtos dispoñibles, engadilos ou eliminalos a un carrito de compras e calcular o total da compra. Os produtos e os seus prezos almacenaranse nunha base de datos MySQL.

O contido da clase *Producto* ten a seguinte estrutura. Para simplificar a lectura, supoñemos que os métodos *get* e *set* están implementados para todos os atributos co formato *getId()/setId()* correspondente (dos cales poderás facer uso sen necesidade de

implementalos). Tamén supón que o obxecto *\$pdo* ten referencia a unha instancia PDO que manexa a base de datos da aplicación:

```
<?php
class Produto {
    private $id;
    private $nome;
    private $prezo;
    private $cantidade;

    public function __construct($id, $nome, $prezo, $cantidade) {
        $this->id = $id;
        $this->nome = $nome;
        $this->prezo = $prezo;
        $this->cantidade = $cantidade;
    }

    public static function obterProdutos() {
        global $pdo;
        $stmt = $pdo->query("SELECT * FROM produtos");
        $produtos = [];
        while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
            $produtos[] = new Produto($row['id'], $row['nome'], $row['prezo'],
$row['cantidade']);
        }
        return $produtos;
    }

    public static function obterProdutoPorId($id) {
        global $pdo;
        $stmt = $pdo->prepare("SELECT * FROM produtos WHERE id = :id");
        $stmt->execute(['id' => $id]);
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        if ($row) {
            return new Produto($row['id'], $row['nome'], $row['prezo'],
$row['cantidade']);
        }
        return null;
    }
}
?>
```

Pídesse que fagas algunhas das funcións da clase *CarroDeCompras* (**no correspondente espazo do caderno de respostas**), de xeito que permita realizar as accións que se indican a continuación. Crea os métodos segundo esta cabeceira:

### Apartado 3.1 (1 punto)

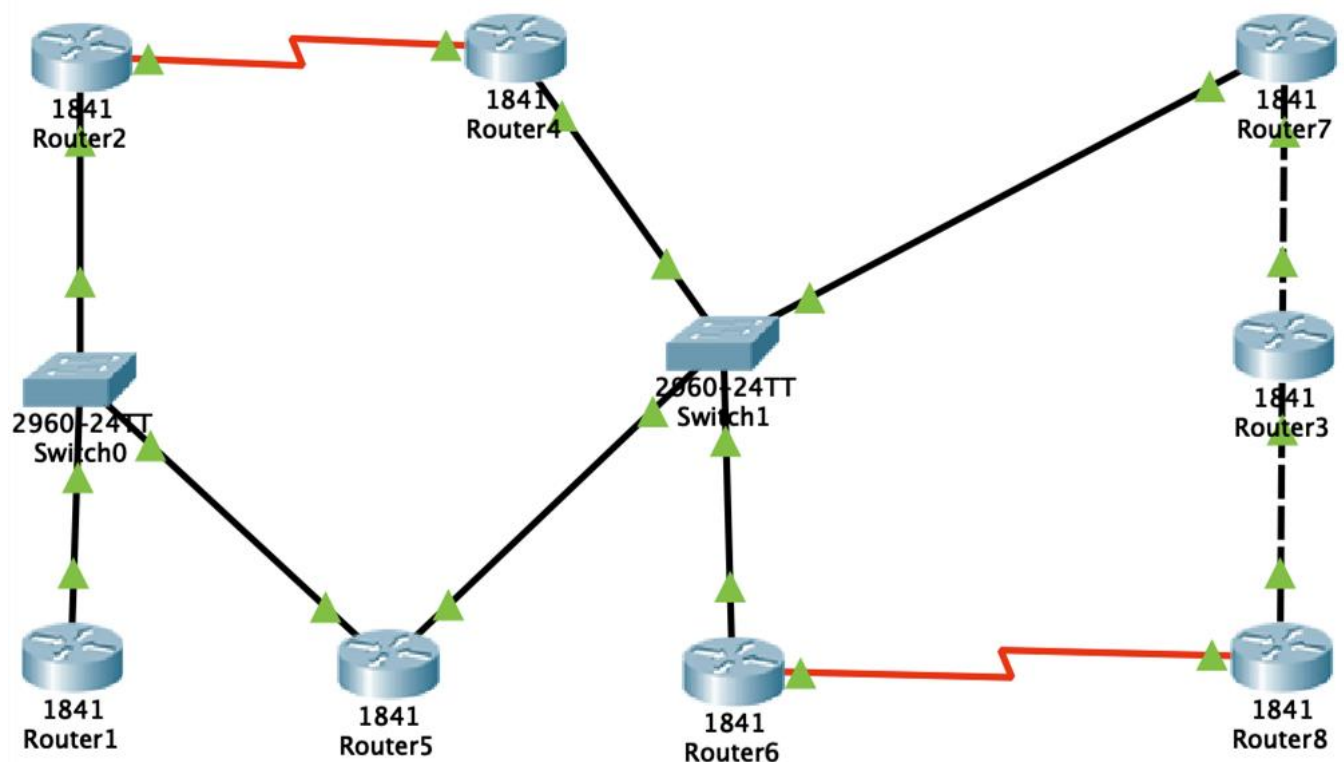
- *public function eliminarProduto(\$nomeProduto)*: elimina unha unidade, no caso de chegar a cero o número de elementos, elimínalo do carrito.

### Apartado 3.2 (1 punto)

- *public function obterTotal()*: devolve o prezo total dos produtos engadidos ao carrito.

### Exercicio 4 (2 puntos)

Na seguinte **topoloxía de rede** todos os routers empregan **OSPFv2** como protocolo de enrutamento dinámico. Dada a configuración de interfaces dos routers indicada, e o número ID de cada un dos routers que foi configurado manualmente:



<b>Router1</b> ID: 1.1.1.1 Interface E0: 10.55.100.1/24	<b>Router2</b> ID: 2.2.2.2 Interface E0: 10.55.100.2/24 Interface S0: 206.155.101.2/27
<b>Router3</b> ID: 3.3.3.3 Interface E0: 10.55.120.3/24 Interface E1: 10.55.130.3/24	<b>Router4</b> ID: 4.4.4.4 Interface E0: 10.55.110.4/24 Interface S0: 206.155.101.4/27
<b>Router5</b> ID: 5.5.5.5 Interface E0: 10.55.100.5/24 Interface E1: 10.55.110.5/24	<b>Router6</b> ID: 6.6.6.6 Interface E0: 10.55.110.6/24 Interface S0: 206.155.201.6/27
<b>Router7</b> ID: 7.7.7.7 Interface E0: 10.55.110.7/24 Interface E1: 10.55.120.7/24	<b>Router8</b> ID: 8.8.8.8 Interface E0: 10.55.130.8/24 Interface S0: 206.155.201.8/27

**Apartado 4.1 (1 punto)**

- Determinar o DR (designated router) para cada REDE tendo en conta que as prioridades das interfaces dos routers son iguais.

**Apartado 4.2 (1 punto)**

- Determinar o BDR (backup designated router) para cada REDE tendo en conta que as prioridades das interfaces dos routers son iguais.

**Ejercicio 5 (2 puntos)**

Transformar o seguinte esquema E/R nun **esquema relacional**, indicando, en cada relación, as claves primarias con subliñado no/os campo/s correspondentes:

nomTabla (Pk, campo1, campo2, ...)

Por claridade, recoméndase non nomear a clave allea igual que o campo referenciado e utilizar o formato:

FOREING KEY \_\_\_\_\_ REFERENCES \_\_\_\_\_ ON \_\_\_\_\_

