

E-Planning Developer Test

La siguiente evaluación tiene por objetivo conocer tu perfil y estilos para desarrollar, y determinar si es coincidente con la búsqueda que realizamos.

Ejercicios de programación

Premisas:

- Debes resolver los puntos en lenguaje Perl. Sabemos que nunca usaste Perl. En estos links podés encontrar algunos tutoriales simples (No hace falta tener conocimientos muy profundos del mismo para esta evaluación):
<https://learnxinyminutes.com/docs/perl/>
<http://qntm.org/files/perl/perl.html>
- Vamos a evaluar la legibilidad y organización del código.
- No hace falta que nos entregues test del código implementado, pero sí que al ejecutarlo cumpla con su tarea.
- A menos que se especifique lo contrario en el punto, no podés utilizar librerías externas que resuelvan el problema.

Ejercicio 1:

Escribir una función que tome dos cadenas de caracteres y retorne verdadero en caso de que estas sean anagramas.

Ejercicio 2:

Escribir una función que tome como parámetros:

1. Un número entero entre 2 y 25.
2. Una cadena representando un número entero positivo en la base del punto 1.
3. Un número entero entre 2 y 25.

Dicha función debe convertir y retornar el número representado por la cadena del argumento 2 (en la base del argumento 1) a una cadena representando el mismo número pero en la base del argumento 3.

Ejemplos:

```
convertir(2,"101", 10) => 5,  
convertir(16,"a2",2) => 10100010,  
convertir(10,"10",16) => "a",  
convertir(11,"13", 4) => 32,
```

Ejercicio 3:

Implementar un script en Perl que tome como entrada una ciudad e imprima por pantalla el estado del clima en estos momentos.

- Obtener la información de <http://openweathermap.org/api>
- La ciudad debe poder indicarse por nombre o código indistintamente.
- En este punto se pueden utilizar librerías externas, salvo **Weather::OpenWeatherMap**.

Ejemplo:

```
$> clima.pl Tokyo  
$> Rain  
$> 39°C 75% RH
```

Ejercicio 4:

Implementar un script en Perl que tome como parámetro un archivo de texto y un segundo parámetro que indique si debe cifrar o descifrar.

El script debe realizar el cifrado/descifrado del mismo aplicando el método ROT13 (reemplazando el archivo original).

Ejercicio 5:

Implementar un script en Perl que tome como entrada una expresión matemática escrita en notación polaca inversa (https://en.wikipedia.org/wiki/Reverse_Polish_notation) e imprima por pantalla el resultado de la misma.

- Deberá soportar los operadores + , - , * y / .
- Los operandos y operadores se separaran por espacio.

Ejemplo:

```
$> rpnl.pl "31 2 2 + 5 * +"  
$> 51
```

Ejercicio 6:

¿Qué deberías testear si tuvieras que escribir los test de unidad del **Ejercicio 2**? (No hace falta escribir el test, solo describir los casos a testear)

Ejercicios de Base de Datos:

Premisa:

- Se cuenta con una empresa que se dedica tanto a vender productos como a proveer servicios, estos productos y servicios tienen precios asociados.
- La empresa tiene una cartera de clientes, quienes pueden comprar sus productos o contratar sus servicios.
- Para registrar estas transacciones, la empresa emite las facturas correspondientes, las cuales pueden incluir en su detalle todos los productos y servicios que se deseen.
- En función de las facturas emitidas por la empresa, cada uno de los clientes va realizando los pagos correspondientes, teniendo la posibilidad de pagar en un mismo momento más de una factura (siempre se paga la totalidad de las mismas).

Ejercicio 7:

Definir un modelo de datos normalizado que soporte el negocio previamente definido.

Ejercicio 8:

Realizar una consulta SQL que permita determinar el saldo de un cliente específico a la fecha de hoy.

Ejercicio 9:

Realizar una consulta SQL que retorne todos los clientes deudores a la fecha de hoy.