

Prueba técnica Symfony

En [REDACTED], el sistema de gestión de usuarios almacena la información en un archivo CSV dinámico [users.csv](#) actualizado cada 24 horas. La información está estructurada en las siguientes columnas:

- Identificador de la estancia (columna 0)
- Nombre del cliente (columna 1)
- Apellido del cliente (columna 2)
- Email (columna 3)
- Código ISO2 del país de residencia (columna 4)
- Fecha de creación del usuario (columna 5)
- Fecha de activación del usuario (columna 6)
- Código del cargador (columna 7)

Se necesita un servicio web que consulte el CSV remoto y retorne el listado de todos aquellos usuarios (ordenados por defecto por nombre y apellidos del cliente).

Opcionalmente se podrán realizar peticiones con los siguientes filtros por separado o juntos:

- Usuario cuya diferencia entre el período de registro y activación sea superior o igual a X días que se indicará por el parámetro **activation_length**
- Clientes que sean de N países que se indicarán por un listado de **countries**.

Ejemplo de salida:

```
{
  "items": [
    {
      "id": 740,
      "name": "Nancy",
      "surname": "Frazier",
      "email": "nfrazierkj @uol.com.br",
      "country": "CN",
      "createAt": "20151201",
      "activateAt": "20151203",
      "chargerId": 740
    },
    {
      "id": 169,
      "name": "Stephen",
      "surname": "Morgan",
      "email": "smorgan4o @examiner.com",
      "country": "CN",
      "createAt": "20151001",
      "activateAt": "20151103",
      "chargerId": 169
    }
  ]
}
```

- El servicio creado debe ser RESTful y tener como formato de salida JSON.
- Elige el nombre y verbo para el endpoint, propiedades, etc que te parezcan más adecuados y fáciles de tratar para la aplicación.
- Para simplificar el código, se supone que el fichero CSV siempre tiene un formato correcto y todas las filas tienen el mismo número de columnas.

Criterios de evaluación:

- Que los servicios funcionen y devuelvan lo que se espera.
- Uso de buenas prácticas de Symfony 3.X ó 4.X. PHP 7.X.
- Programación orientada a objetos.
- Se podrá estructurar el código como se quiera, no hay límite en el número de clases a usar, con la prueba se quiere ver la capacidad de estructuración del código del candidato.
- Se valorará positivamente la aplicación de testing unitario, principios SOLID y DDD.