

Prueba Técnica Python



Requisitos técnicos:

- Python 3.7
- Framework Flask

Ejercicio 1:

- Crear un servicio REST para la entrega de patentes y/o ID, compuesto por:
 - 1.- un endpoint donde se ingrese una patente y este retorne el ID asociado a la patente.
 - 2.- un endpoint donde se ingrese el ID y este retorne la patente.
- El dominio de las patentes corresponde al siguiente: [AAAA000, AAAA001, ..., ZZZZ999], el ID es secuencial, comienza en 1.
 - Entonces: AAAA000 tiene ID =1, AAAA001 tiene ID = 2 y sucesivamente.
- Se excluye del dominio letra Ñ.

Ejercicio 2:

Basado en una matriz de $R \times C$, entregando coordenadas de X, Y , se debe sumar los valores hacia las coordenadas de origen $(0,0)$.

Restricciones:

- . $R > 0, C > 0$, donde R y C son enteros.
- . $X \geq 0, Y \geq 0$; X e Y son enteros.
- . El llenado de la matriz está dado por la variable Z , de tipo entero donde $0 < Z \leq 1.000.000$
- . Donde sea $R = 1$ completar con Z . Si $R > 1$ incrementar cada columna R_n con $Z + R_n - 1$



Ejemplo:

Matriz R x C = 4, 3

Z = 2

X = 1

Y = 2

Por tanto, se obtiene la siguiente matriz:

	C= 1	C =2	C= 3
R = 1	2	2	2
R = 2	3	3	3
R = 3	4	4	4
R = 4	5	5	5

Dada las coordenadas X e Y la matriz da como resultado 18 :

	x0	x1
y0	2	2
y1	3	3
y2	4	4

•

Esto debe ser entregado en el mismo proyecto REST anterior, en un endpoint distinto.

donde su input son los parámetros R, C, Z, X, Y.

el servicio debe retornar la sumatoria solicitada.

Plazo de entrega 48 hrs desde el momento de recepción las de instrucciones.

*Criterios de evaluación:

- Coherencia de código:
 - El uso de patrón o patrones de diseño, no excluyente.
 - El uso e implantación de SOLID.
 - El orden de la distribución de las carpetas/archivos (servicios/controladores/repositorios/modelos/ util /helpers, etc)
- Estilo de programación, se valora PEP8
- Manejo de excepciones.
- El uso typing
- El uso de loop, manipulación de arreglos, manipulación de diccionarios, objetos, clases, variables, instancias de objetos, decoradores
- Instrucciones de despliegue del componente, se valora Makefile.
- Sobre Makefile , se valora despliegue en docker
- La documentación README
- Test unitarios, Mock, Spy, stubs, los pesos y relevancia de este criterio son por orden de implementación:
 - Test unitarios
 - Test paramétrico
 - Test de integración
 - El coverage >= 80%
 - Indistinto a completar el coverage del 80%, se evaluará la lógica de construcción del o los test.