**Evaluación Desarrollador Backend**

Ud. debe resolver la mayor cantidad de preguntas propuestas en 4 horas.

**EVALUACIÓN EN BASE DE DATOS**

**PREGUNTA 1**

Independientemente del motor que se use ¿Cómo implementaría un flujo de respaldo de su base de datos? Justifique su respuesta

En MongoDb Atlas tenemos 3 nodos por si uno no funciona, inmediatamente se puede usar uno de los nodos restantes y la información es la misma en todas. No tendríamos problemas de ese lado. El mismo AWS RDS cuenta con esto también.

Las mismas copias de seguridad por si hay problemas. Mejor si es que son automatizadas.

Independientemente de las bases de datos, estas se pueden versionar y pueden haber ramas. Por ejemplo con el paquete flask-migrations hay ramas. Y en cierto modo versiones de la estructura de la base de datos.

Ver en los logs de la base de datos si algo no se ejecuto.

Sobre el flujo, si contara con los servicios cloud de AWS, ATLAS o azure, revisaria si ese nodo de la instancia está bien. No tendría muchos problemas. Ya que la informacion es la misma en todos los nodos. Ejecutar lo que falto en los logs. Luego las copias de seguridad.

**PREGUNTA 2**

¿Por qué en bases de datos, tratar con ***null*** es un caso especial?

Por ejemplo,

¿Por qué en `***SELECT \* FROM "Table" WHERE "field" = null;***` no devuelve ningún registro que tenga valor `***NULL***` en el campo `***field***`?

Porque en realidad null representa un valor desconocido, lo práctico es verlo como sin valor. A la hora de guardar una instancia no se dio datos en ese campo.

Porque no se debe usar = sino “IS”, es decir la consulta debe ser asi:

`***SELECT \* FROM "Table" WHERE "field" IS NULL;***`

**EVALUACIÓN EN PROGRAMACIÓN PROCEDURAL**

Debe utilizar python o php para resolver estos problemas:

**PROBLEMA 1**

**Nombre del script:** random\_accumulator.(php|.py|.js|.rb|.java|.cpp)

Generar aleatoriamente números enteros entre 1 y 100 hasta que el valor acumulado de la suma de los números generados sea mayor que 10000. Escribir una función que devuelva el valor acumulado de la suma hasta el valor más cercano que no exceda el valor 10000. Invoque a esta función para imprimir dicho valor como única salida del programa.

**PROBLEMA 2**

**Nombre del script:** client\_add (php|.py|.js|.rb|.java|.cpp)

Escribir un programa script que realice la inserción de un registro de un cliente. Para lograr cumplir el reto propuesto, debe poder insertarse (a sí mismo) como cliente con sus nombres, apellidos y demás datos solicitados en el registro. Se debe usar el API público del aplicativo WHMCS, que se encuentra ubicado en el siguiente enlace:

<https://empresas.yachay.lat>/includes/api.php

Como referencia, se le anexa un script que le permite listar los clientes:

<https://empresas.yachay.lat/testJT/client_list.php>

Y la salida del siguiente comando “head”, conocerá los accesos y el token del API:

*# head client\_list.php*

*<?php*

*$whmcsUrl = "https://empresas.yachay.lat/";*

*// Learn more at http://docs.whmcs.com/API\_Authentication\_Credentials*

*$api\_identifier = "gTxMO3VojXzm6g8vTk6rWt1wjAt5BMdE";*

*$api\_secret = "dUA5kAV2aJC2xkHDKTrnrNLHECrjxhGG";*

*$accesskey = 'AccesoTemporal$2022$RCP';*

*$ch = curl\_init();*

*curl\_setopt($ch, CURLOPT\_URL, $whmcsUrl . 'includes/api.php');*

Documentación en línea de la API:

<https://developers.whmcs.com/api/authentication/>

<https://developers.whmcs.com/api/access-control/>

<https://developers.whmcs.com/api-reference/addclient/>

Se anexan los campos personalizados (custom) de la entidad cliente:

1. Tipodocumento|Tipo de documento
2. Numerodedocumento|Número de documento

