

Desarrollo/Interview

From TecnoWiki
< Desarrollo

Contents

- 1 Entrevistas Área Tecnología
 - 1.1 Ejercicios Propuestos
 - 1.1.1 Ejercicio 1
 - 1.1.2 Ejercicio 2
 - 1.1.3 Ejercicio 3
 - 1.1.4 Ejercicio 4
 - 1.1.5 Ejercicio 5
 - 1.1.6 Ejercicio 6

Entrevistas Área Tecnología

El objetivo de la entrevista técnica es evaluar tanto los conocimientos técnicos de la persona entrevistada en algunas de las diferentes tecnologías que utilizamos, como también evaluar su capacidad de análisis y resolución de problemas.

Para evaluar sobre los criterios mencionados anteriormente, se presentaran una serie de ejercicios los cuales deben ser resueltos en lenguaje **perl** o **python** (a elección de la persona entrevistada).

En una **primer etapa** de la entrevista se le presentan los ejercicios a resolver al entrevistado y se le responden sus dudas sobre el planteamiento y alcance de los mismos. También se acordará un periodo de tiempo **de no mas de 2 semanas** durante el cual el entrevistado intentará resolver los ejercicios propuestos, para luego coordinar una **segunda etapa** de la entrevista en donde el entrevistado presentará los ejercicios y se le harán preguntas relacionadas a estos y a las tecnologías empleadas para resolverlos.

Los ejercicios deben subirse a un repositorio **git** (<https://github.com/>) de la persona entrevistada, no hace falta que sea público pero debe poder ser accedido por el/los entrevistadores durante la entrevista.

Ejercicios Propuestos

El entrevistado debe subir sus ejercicios resueltos a su repositorio **git** (<https://github.com/>) y agregar:

- Código fuente de cada ejercicio
- Un archivo **README.md** (<https://www.makeareadme.com/>) que indique:
 - Indicaciones de cómo instalar las dependencias necesarias según las tecnologías utilizadas (MySQL, librerías, etc...)
 - Indicaciones de cómo ejecutar los scripts de cada ejercicio
- De ser necesario, un Dump de la base de datos

Ejercicio 1

Realizar un script a ejecutar por consola, en lenguaje **perl** o **python**, que realice las siguientes tareas:

- Defina un string llamado **cadena** el cual contiene la siguiente información:

```
# Perl
my $cadena =
'Jacinta_Flores$Juan_Carlos_Feletti$Pedro_Lugones$Ana_Maria_Galindez$Juana_Bermudez$Rafael_Ernesto_Brahms$Beatriz_Vale
nte$Ulma_Fabiana_Goya$Martina_Nicolesi$Betania_Miraflores$Fermin_Olivetti$Ana_Luz_Narosky$Graciana_Arruabarrena$Joel_P
erez$Valentina_Feller$Hector_Tadeo_Siemens$Natalia_Martinevsky$Ernesto_Nicolini$Pia_Paez$Fermin_Obdulio_Camilo_Galinde
z$Delfina_Beirut$Walter_Mantinoli$Celina_Celia_Samid$Ulises_Malo$Juana_Varela$Melquiades_Jose_Li$Radamel_Servini$Filem
on_Salsatti$Celeste_Faim$Valerio_Martin_Rosseti$Jeremias_Farabutti$Veronica_Nefertiti$Ana_Delia_Pereyra$Hermenilda_Car
la_Rutini$Valerio_Tunuyan$Silvia_Solano$Beatriz_Bevacqua$Manuel_Martinez$Berto_Carlos_Kigali$Juan_Manuel_Miraflores$Ni
colas_Kligorsky$Maria_Laura_Berotti';
```

```
# Python
cadena = "Jacinta_Flores$Juan_Carlos_Feletti$Pedro_Lugones$Ana_María_Galíndez$
Juana_Bermudez$Rafael_Ernesto_Brahms$Beatriz_Valente$Ulma_Fabiana_Goya$Martina_Nicolesi$
Betania_Miraflores$Fermin_Olivetti$Ana_Luz_Narosky$Graciana_Arruabarrena$Joel_Pérez$
Valentina_Feller$Hector_Tadeo_Siemens$Natalia_Martinevsky$Ernesto_Nicolini$Pia_Paez$
Fermin_Obdulio_Camilo_Galíndez$Delfina_Beirut$Walter_Mantinoli$Celina_Celia_Samid$
Ulises_Malo$Juana_Varela$Melquiades_José_Li$Radamel_Servini$Filemón_Salsatti$Celeste_Faim$
Valerio_Martín_Rosseti$Jeremías_Farabutti$Verónica_Nefertiti$Ana_Delia_Pereyra$
Hermenilda_Carla_Rutini$Valerio_Tunuyán$Silvia_Solano$Beatriz_Bevacqua$Manuel_Martínez$
Berto_Carlos_Kigali$Juan_Manuel_Miraflores$Nicolás_Kligorsky$María_Laura_Berotti"
```

Y luego realizar lo siguiente:

- Aplicar un método o una expresión regular para reemplazar los ` ` por espacios.
- Separar cada nombre completo (basarse en el carácter \$ que es el separador de cada persona)
- Agregar los nombres a una lista (array) llamada **alumnos**.
- Imprimir por pantalla la lista de nombres, uno por linea.

Ejercicio 2

Realizar un script a ejecutar por consola, en lenguaje **perl** o **python**, que realice las siguientes tareas:

- Definir una función llamada **ordenar** que reciba una lista (array) de nombres y apellidos (nombre/s seguido/s de un apellido) y los procese de la siguiente manera:
- Ubicar primero el apellido seguido por una coma (asumir que cada nombre de la lista recibida tiene un sólo apellido)
- Luego, ordenar la lista por orden alfabético, según el apellido (pista: pueden utilizar el método **sorted()** (<https://docs.python.org/3.5/howto/sorting.html>) en python, o **sort()** (<https://perldoc.perl.org/functions/sort>) en perl)
- Por último, iterar sobre los elementos de la lista (array) y asignarlos como valores a un diccionario (hash) cuyas claves sean enteros ascendentes e imprimir por pantalla el diccionario (hash) generado.

Ejemplo:

```
# Perl
my @lista_nombres = ("Jacinta Flores", "Juan Carlos Feletti", "Pedro Lugones", "Ana Maria Galindez");

# Resultado
{1 => "Feletti, Juan Carlos", 2 => "Flores, Jacinta", 3 => "Galíndez, Ana María", 4 => "Lugones, Pedro"}
```

```
# Python
lista_nombres = ["Jacinta Flores", "Juan Carlos Feletti", "Pedro Lugones", "Ana María Galíndez"]

# Resultado
{1: "Feletti, Juan Carlos", 2: "Flores, Jacinta", 3: "Galíndez, Ana María", 4: "Lugones, Pedro"}
```

Ejercicio 3

Realizar un script a ejecutar por consola, en lenguaje **perl** o **python**, que realice las siguientes tareas:

1. Obtener datos de usuarios del endpoint <http://packages.educativa.com/samples/usuarios.json>
2. Obtener datos de cursos del endpoint <http://packages.educativa.com/samples/cursos.json>
3. Parsear los datos de cada JSON e imprimirlos en pantalla

- Hints para Perl:

- Se recomienda usar la lib **LWP::UserAgent** (<https://metacpan.org/pod/LWP::UserAgent#SYNOPSIS>) para hacer la petición y luego la lib **JSON** (<https://metacpan.org/pod/JSON>) para parsear los datos JSON a Perl.
- Puede ser necesario instalar la lib **JSON** en Perl: `sudo cpan JSON`

- Hints para Python:

- Se recomienda usar la lib **requests** (<https://requests.readthedocs.io/en/latest/>) para hacer la petición y luego la lib **json** (<https://docs.python.org/3.5/library/json.html>) para parsear los datos JSON a Python.
- Puede ser necesario instalar la lib **requests** en Python: `pip install requests`

Ejercicio 4

Crear una base de datos en **MySQL** llamada **edu_challenge_db** de la siguiente manera:

```
sudo mysql -u root    #puede ser necesario según distro de Linux

CREATE DATABASE edu_challenge_db CHARACTER SET utf8;
CREATE USER 'edu_challenge_user'@'localhost' IDENTIFIED BY '3duc4';
GRANT ALL PRIVILEGES ON edu_challenge_db.* TO 'edu_challenge_user'@'localhost';
```

Realizar un script a ejecutar por consola, en lenguaje **perl** o **python**, que realice las siguientes tareas:

- Se debe poder conectar a la DB mencionada anteriormente con los datos indicados.
- Luego sobre dicha DB, debe crear las siguientes 3 tablas:

```
Tabla: usuarios
Columnas:
  id_usuario INT UNSIGNED NOT NULL
  nombre VARCHAR(255) NOT NULL
  apellido VARCHAR(255) NOT NULL
```

```
Tabla: cursos
Columnas:
  id_curso INT UNSIGNED NOT NULL
  nombre VARCHAR(255) NOT NULL
  cupo INT UNSIGNED NOT NULL
  id_docente INT UNSIGNED NOT NULL
```

```
Tabla: usuarios_cursos
Columnas:
  id_curso INT UNSIGNED NOT NULL
  id_alumno INT UNSIGNED NOT NULL
```

- Indicar a criterio propio cuales columnas serán claves primarias (PK) y cuales serán claves foráneas (FK)

- Hints para Python:

- Se recomienda usar la lib **mysql-connector-python** (<https://pypi.org/project/mysql-connector-python/>) para hacer la conexión con la DB. Para instalarla: `pip install mysql-connector-python`
- Tutorial y ejemplos: <https://www.geeksforgeeks.org/python-mysql/>
- Hints para Perl:
 - Se recomienda usar la lib **DBI** (<https://metacpan.org/pod/DBI>) para hacer la conexión con la DB. Para instalarla: `sudo cpan DBI && sudo cpan DBD:mysql`
 - Tutorial y ejemplos: <https://www.perltutorial.org/perl-dbi/>

Ejercicio 5

Realizar un script a ejecutar por consola, en lenguaje **perl** o **python**, que realice las siguientes tareas:

- Utilizando los datos obtenidos en el Ejercicio 3, popular las tablas creadas en el Ejercicio 4.
- **Mostrar el contenido de las tablas en pantalla.** Debe presentarse como una sola tabla conteniendo todos los datos.

id_usuario	id_curso	nombre y apellido usuario	nombre curso
1	3	Akira Kurosawa	Filosofia
1	4	Akira Kurosawa	Biologia
2	3	Woody Allen	Filosofia
3	1	Groucho Marx	Matematica

Ejercicio 6

Realizar un script a ejecutar por consola, en lenguaje **perl** o **python**, que realice las siguientes tareas a partir de los datos guardados en base de datos del Ejercicio 5:

- **Permita inscribir un usuario a un curso**

Consideraciones:

- Si el curso ya tiene el cupo de inscripciones completo (hay tantos usuarios como valor de cupo del curso), no se inscribe el usuario y muestra un mensaje indicando la situación.
- Si el usuario ya está inscripto en el curso, indicar un mensaje indicando la situación.
- Si el usuario es el docente del curso, no puede inscribirse y debe indicar un mensaje indicando la situación.

- **Permita eliminar la inscripción de un usuario en un curso**

- Al ejecutar el script sin parámetros, se deben mostrar las opciones. Ejemplo:

```
perl inscribir_usuario.pl (o python3 inscribir_usuario.py)
```

Debe indicar una de las siguientes opciones:

- 1) mostrar lista de inscripciones
- 2) inscribir usuario
- 3) eliminar inscripción

Por ejemplo, si queremos listar las inscripciones:

```
perl inscribir_usuario.pl 1 (o python3 inscribir_usuario.py 1)
```

id_usuario	id_curso	nombre_usuario	nombre_curso	...
1	3	Akira	Filosofia	...

1	4	Akira	Biologia	...
2	3	Woody	Filosofia	...
3	1	Groucho	Matematica	...

Por ejemplo, si queremos inscribir al usuario 4 en el curso 2:

```
perl inscribir_usuario.pl 2 4 2 (o python3 inscribir_usuario.py 2 4 2)
```

Se inscribió el usuario 4 al curso 2

Retrieved from "<http://tecnowiki.e-ducativa.x/index.php?title=Desarrollo/Interview&oldid=109283>"

This page was last edited on 5 September 2022, at 11:33.