

Yinkana

David.Villa@uclm.es

Ana.Rubio@uclm.es

(versión 2021.03.08)

La yinkana es un ejercicio de programación de aplicaciones cliente/servidor con sockets. El alumno debe escribir un programa en lenguaje Python (versión ≥ 3.5) que completará de forma autónoma (sin intervención alguna del usuario) una serie de retos de programación de redes, con dificultad y valor crecientes. Se puede utilizar cualquier módulo de la librería estándar¹, excepto los indicados en las instrucciones de cada reto. El uso de módulos ajenos a la librería estándar debe ser solicitado a través del foro de la asignatura y autorizado explícitamente por los profesores.

El programa escrito por el alumno para superar la yinkana debe estar escrito en un solo fichero `.py` y ejecutarse desde la máquina llamada *the-hub*.

1. Acceso

En breve recibirás un correo electrónico en tu cuenta `@alu.uclm.es` indicando tu usuario, tu clave privada SSH (ej: `juan_nadie`) y su certificado correspondiente (ej: `juan_nadie-cert.pub`).

Para acceder a *the-hub* debes conectar con el servidor SSH disponible en `arco.esi.uclm.es:3333`. Requiere la conexión a través de la VPN de la UCLM. Puedes conectar con el siguiente comando:

```
$ ssh usuario@arco.esi.uclm.es -p 3333
```

Como *usuario* debes utilizar el nombre de usuario que se te ha facilitado respetando las minúsculas.

1.1. Configuración SSH

Como se ha visto en clase y se indica en el manual de laboratorio, puedes simplificar el comando anterior creando un fichero de configuración llamado `~/.ssh/config` en tu computador con el siguiente contenido:

```
Host the-hub
  Hostname arco.esi.uclm.es
  Port 3333
  User juan_nadie
```

Este fichero debe tener permisos de lectura/escritura solo para el usuario o será ignorado. Una vez tengas esta configuración puedes conectar simplemente con:

```
juan@laptop:~$ ssh the-hub
juan_nadie@arco.esi.uclm.es's password:
```

¹<https://docs.python.org/3/library/>

1.2. Acceso con certificado

Para evitar tener que escribir la clave cada vez que conectes a *the-hub*, puedes utilizar la clave privada y el certificado que has recibido. Copia estos ficheros en tu directorio `~/ .ssh` y dale los permisos adecuados, al menos a la clave privada (0600).

```
juan@laptop:~$ ls ~/.ssh
-rw----- 1 juan juan 2590 mar  8 15:47 juan_nadie
-rw-r--r-- 1 juan juan 2026 mar  8 15:47 juan_nadie-cert.pub
[...]
```

Edita el fichero de configuración para añadir la ubicación de la clave privada:

```
Host the-hub
  Hostname arco.esi.uclm.es
  Port 3333
  User juan_nadie
  IdentityFile ~/.ssh/juan_nadie
```

Una vez hecho esto, podrás conectar sin tener que escribir la contraseña cada vez.

1.3. Editar y ejecutar tu programa

Lo más sencillo es editar el programa en tu portátil y copiarlo a *the-hub* cada vez que quieras probar una modificación. Suponiendo que tu programa se llama `solucion-yinkana.py`, lo puedes hacer con el siguiente comando —siempre que hayas definido la configuración de la sección anterior.

```
juan@laptop:~$ scp solucion-yinkana.py the-hub:
```

IMPORTANTE Ten en cuenta que la máquina *the-hub* puede ser destruida y regenerada periódicamente. Eso significa que el fichero que subas podría desaparecer sin previo aviso. Por tanto, mantén una copia de tu programa bajo tu control.

2. Primer reto

Una vez hayas iniciado sesión en *the-hub* puedes acceder al reto 0 conectando con el servidor TCP en:

rick:2000

Al conectar a dicho servidor, obtendrás las instrucciones para identificarte y pasar al siguiente reto, y una vez superado ese reto, las del siguiente, etc. La yinkana consta de 8 retos en total incluyendo el anterior.

3. Evaluación

Para lograr la máxima puntuación, el alumno deberá presentar una versión preliminar que resuelva al menos los cuatro primeros retos [0 – 3] antes del 16 de abril de 2021. La entrega final será el **17 de mayo de 2021, 23:59h**.

Adicionalmente el alumno deberá realizar una defensa presencial de su programa tras la entrega. Se publicará un horario para que cada alumno defienda su programa ante su profesor de laboratorio.

3.1. Puntuación

Cada reto (excluyendo el 0 y el 7) se valora en dos aspectos: funcionalidad y calidad. La suma total de la puntuación de los retos se barema sobre un máximo de 16 puntos. Además la defensa se puntúa con un máximo de 4 puntos adicionales, resultado un total de 20 puntos que corresponden a la actividad «Realización de prácticas en laboratorio» referida en la guía docente.

La **funcionalidad** valora que se resuelva el reto conforme a sus especificaciones y sea consistente, es decir, funcione adecuadamente en todas y cada una de las sucesivas ejecuciones. Si solo funciona a veces, se considera que no funciona.

En cuanto a la **calidad** del código, se valoran los siguientes aspectos:

- Estructura y organización del código, mediante clases, funciones o métodos.
- Elección de nombres significativos para funciones, variables, etc.
- Evitar uso de variables globales innecesarias.
- Evitar anidación injustificada de estructuras de control (más de 2 niveles).
- Evitar uso incorrecto de tipos de datos, código redundante, código muerto, etc.
- Evitar bloques (funciones/métodos) demasiado largos (más de 10-12 líneas).

La puntuación de cada reto se especifica a continuación:

Reto	1	2	3	4	5	6
Funcionalidad	2	4	5	7	8	10
Calidad	1	2	3	4	6	8

Penalizaciones

Se contemplan 4 motivos por los que la puntuación total de la actividad puede sufrir penalizaciones:

- No realizar la entrega preliminar: -3 puntos.
- No realizar la defensa oral: -3 puntos.
- Usar una versión de Python anterior a la 3.5: -3 puntos.
- Errores de sintáxis que impiden ejecutar el programa: -3 puntos.

En este último caso se pedirá al alumno presentar una versión corregida.

4. Condiciones

El alumno debe tener especial cuidado con la incorporación de código ajeno en su programa. Está PERMITIDO siempre que se cumplan las siguientes condiciones:

- Los fragmentos copiados y su autor están claramente identificados.
- El alumno comprende y es capaz de explicar con TODO DETALLE la funcionalidad del código copiado.
- El autor del código original permite la copia mediante la licencia correspondiente y esta aparece claramente junto al código original.

- El autor del original mantiene su código públicamente accesible (y el alumno proporciona el enlace en forma de comentario).
- El autor del original no es ni ha sido un alumno de la UCLM.

Infringir **cualquiera** de estas normas al utilizar código ajeno se considerará plagio. Cometer plagio implica una calificación de 0 en la actividad «Realización de prácticas de laboratorio» según el artículo 9 de la «Guía de Evaluación del Estudiante»:

Art. 9. Realización fraudulenta de pruebas de evaluación.

1. La constatación de la realización fraudulenta de una prueba de evaluación o el incumplimiento de las instrucciones fijadas para la realización de la prueba dará lugar a la calificación de suspenso (con calificación numérica de 0) en dicha prueba. En el caso particular de las pruebas finales, el suspenso se extenderá a la convocatoria correspondiente.

5. Entrega

Tanto la entrega final como la preliminar deberán realizarse mediante una tarea *moodle* en la página de la asignatura. No se aceptarán entregas fuera de plazo, por correo electrónico ni cualquier otro medio.