

Yinkana

(Convocatoria ordinaria)

David.Villa@uclm.es

Joseluis.Segura@uclm.es

Profesor.Mhervas@uclm.es

Ángel Suárez-Bárcena Velázquez

(versión 2023.03.15)

La yinkana es un ejercicio de programación de aplicaciones cliente/servidor con sockets. El alumno debe escribir un programa en lenguaje Python (versión $\geq 3.9.2$) 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. Tampoco está permitido ejecutar comandos u otras aplicaciones desde el programa del alumno.

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

1. Acceso

En el correo electrónico que recibiste en tu cuenta @alu.uclm.es para la convocatoria ordinaria se indica tu usuario, tu clave privada SSH (ej: juan_nadie) y su certificado correspondiente (ej: juan_nadie-cert.pub).

Para poder acceder a la máquina *the-hub* es necesario conectar con el servidor SSH disponible en 161.22.47.12:2222. No se requiere conexión a través de la VPN de la UCLM. Puedes conectar con el siguiente comando:

```
$ ssh juan_nadie@161.22.47.12 -p 2222
```

En lugar de *juan_nadie* debes escribir el nombre de usuario que se te ha asignado 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 161.22.47.12
  Port 2222
  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:

¹<https://docs.python.org/3/library/>

```
juan@laptop:~$ ssh the-hub
juan_nadie@161.22.47.12's password:
```

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 161.22.47.12
  User juan_nadie
  Port 2222
  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:

yinkana:2000

Al conectar a este 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

La entrega final será el **1 de junio de 2023, 23:59h**. Para esta entrega no habrá defensa aunque eventualmente el profesor se pondrá en contacto con alumnos concretos en caso de existir dudas.

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 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. Si existen casos no contemplados o faltan mecanismos de gestión de errores, la puntuación se verá afectada negativamente aunque el programa nunca falle.

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, duplicado, muerto, etc.
- Evitar bloques (funciones/métodos) demasiado largos (más de 10-12 líneas).

Funcionalidad y calidad se evalúan con una única nota por reto ponderada al 50 %. La puntuación de cada reto se especifica a continuación:

Reto	1	2	3	4	5	6
Nota	3	6	8	11	14	18

Penalizaciones

Se contemplan 3 motivos por los que la puntuación total de la actividad puede sufrir penalizaciones:

- Usar una versión de Python anterior a la 3.9.2: -3 puntos sobre 20.
- Errores de sintaxis que impiden ejecutar el programa: -3 puntos sobre 20.
- Incumplir otras condiciones del enunciado: hasta -10 puntos sobre 20.

En el caso de errores de sintaxis, si supone cambios menores, se pedirá al alumno presentar una versión corregida.

Además, el alumno debe estar inscrito a alguno de los grupos de laboratorio, ya que esto determina que profesor será responsable de su evaluación. Si el alumno incumple esta condición tendrá una penalización adicional de -5 puntos sobre 20.

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 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

La entrega final deberá 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.