

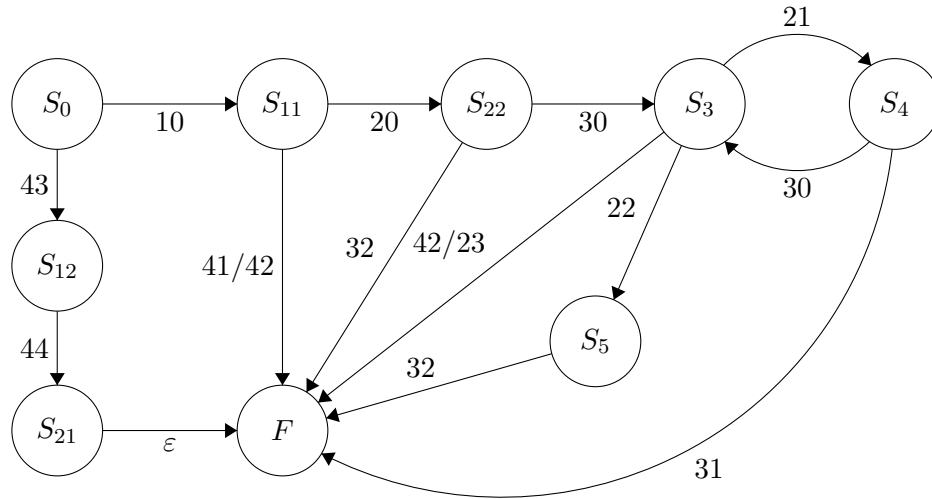
Facultad de Ciencias, UNAM
Creación e implementación de un protocolo de la capa de Aplicación
Redes de Computadoras

Marisol Amézcua Lopez

16 de diciembre de 2018

Objetivo: Implementar una aplicación que dada una base de datos con items y usuarios sea posible tener un registro de los items de cada usuario así como poder capturar nuevos items y registrarlos en el historial del usuario.

El autómata de la aplicación es el siguiente:



Codigo	Descripción
10	Se solicita al servidor capturar un pokemon
20	Desea capturar al pokemon x?
21	Desea volver a capturar al pokemon x? Le quedan k intentos
22	Felicidades, capturaste al pokemon x. Despliega imagen de pokemon x.
23	Número de intentos agotados.
30	Sí
32	Terminando sesión/No
41	Error con el id que ingreso el usuario.
42	Codigo de error interno en algun punto
43	Se solicita al servidor desplegar la pokedex
44	Pokedex obtenida exitosamente. Se despliegan los nombres de los pokemones capturados.

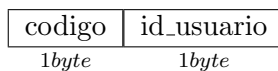
Nota: En el código 32 se englobo el 31 y 32 debido a que no aportaba ningún cambio el 31.

La estructura de los mensajes es el siguiente:

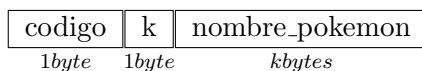
- Los mensajes de los códigos 23,30,32,41,42 y 44 constan de un solo byte.



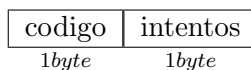
- El mensaje para el codigo 10 y 43 consta de un byte para el numero de codigo y un byte para el id del usuario:



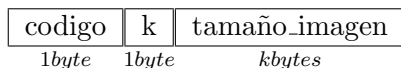
- El mensaje para el codigo 20 consta de un byte seguido de un byte para la cantidad de bytes del nombre del pokemon y al final esta el nombre del mismo:



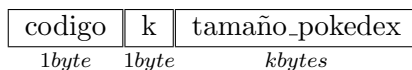
- El mensaje 21 consta de 1 byte para el codigo y otro para el numero de intentos restantes:



- El mensaje 22 contiene 1 byte para el codigo, 4 bytes para el tamaño de la imagen y k bytes para el tamaño de la imagen.



- El codigo 44 contiene 1 byte para el numero de código, 1 byte para el tamaño del string que representa la pokedex y k bytes que son la pokedex:



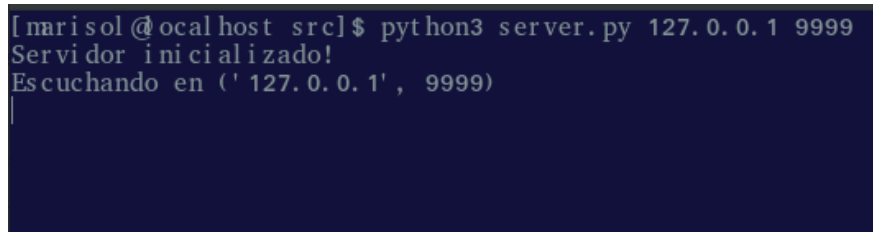
Nota: A algunos codigos se les cambiaron el número de bytes para facilitar la implementación.

1. Uso de la aplicación

En ambos casos lo primero es correr el servidor, para esto es necesario ejecutar el comando

```
python3 server.py <IP> <Puerto>
```

Si todo sale bien veremos un mensaje de que el servidor se encuentra escuchando el puerto, en otro caso veremos un mensaje de error.

A terminal window with a dark blue background. The prompt is [marisol@ocalhost src]\$. The command entered is python3 server.py 127.0.0.1 9999. The output shows 'Servidor inicializado!' followed by 'Escuchando en (' 127.0.0.1', 9999)' and a cursor on the next line.

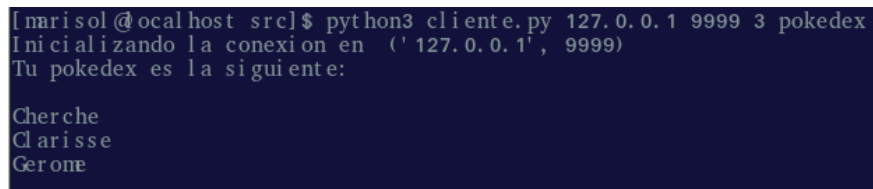
```
[marisol@ocalhost src]$ python3 server.py 127.0.0.1 9999
Servidor inicializado!
Escuchando en (' 127.0.0.1', 9999)
|
```

Figura 1: Imagen de como se observa en la terminal el iniciar el servidor

- Caso de Pokedex

Si queremos desplegar la pokedex de un usuario ejecutamos lo siguiente:

```
python3 server.py <IP> <Puerto> <Id_usuario> pokedex
```

A terminal window with a dark blue background. The prompt is [marisol@ocalhost src]\$. The command entered is python3 cliente.py 127.0.0.1 9999 3 pokedex. The output shows 'Inicializando la conexion en (' 127.0.0.1', 9999)', 'Tu pokedex es la siguiente:', and a list of names: Cherche, Clarisse, and Gerone.

```
[marisol@ocalhost src]$ python3 cliente.py 127.0.0.1 9999 3 pokedex
Inicializando la conexion en (' 127.0.0.1', 9999)
Tu pokedex es la siguiente:
Cherche
Clarisse
Gerone
```

Figura 2: Captura de como se ve la pokedex

- Caso de atrapar pokemones

Para este caso lo necesario es ejecutar lo siguiente:

```
python3 server.py <IP> <Puerto> <Id_usuario>
```

Si queremos atrapar a un pokemon el servidor nos dara un nombre de dicho pokemon y si queremos atraparlo. Al darle si correremos nuestro primer intento, en caso de no atraparlo nos dira los intentos restantes, y en caso de atraparlo nos mandara un mensaje indicando esto y desplegara la imagen de dicho pokemon.

En todos los casos el servidor nos dira que cliente entro y desde que puerto ademas de avisarnos cuando haya terminado de atender a un cliente.

```

[narisol@localhost src]$ python3 cliente.py 127.0.0.1 9999 3
Iniciando la conexion en ('127.0.0.1', 9999)
Gusta capturar al pokemon Felicia?
[si/No]si
¿Intentar captura de nuevo? Quedan 9 intentos.
[si/No]si
Felicidades, has atrapado a Felicia
Terminando sesion

```

Figura 3: Captura de la interaccion con el servidor al intentar obtener un pokemon.

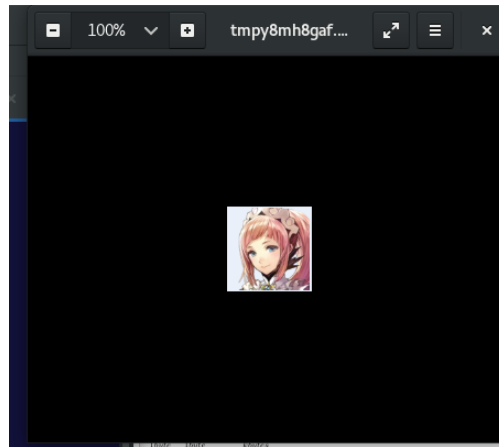


Figura 4: Captura de como se observa la imagen al capturar al pokemon.

```

[narisol@localhost src]$ python3 server.py 127.0.0.1 9999
Servidor inicializado!
Escuchando en ('127.0.0.1', 9999)
Nuevo cliente! Su direccion es: ('127.0.0.1', 55212)
Cerrando sesion del cliente: ('127.0.0.1', 55212)
Nuevo cliente! Su direccion es: ('127.0.0.1', 55214)
Cerrando sesion del cliente: ('127.0.0.1', 55214)

```

Figura 5: Captura de lo obtenido del servidor con la interaccion.

2. Requerimientos para la instalacion del proyecto

Dado que esta desarrollado en Python es necesario tener Python 3 instalado y pip, para la base de datos de utilizo SQLite, por lo que sera necesario para las consultas de la base. Una vez con eso para instalar los modulos basta con ejecutar:

```
pip install -r paquetes.txt
```

Esto instalara los paquetes necesarios, al final solo es necesario ejecutar los modulos como se explico anteriormente.