

# Proyecto 2

Carlos Gerardo Acosta Hernández  
Andrea Itzel González Vargas  
Luis Pablo Mayo Vega

Redes de Computadoras  
Facultad de Ciencias, UNAM

## Índice

<b>1. Especificación de requerimientos</b>	<b>2</b>	3.1. Especificación del ambiente de desarrollo . . . . .	3
1.1. Enunciado del problema . . . . .	2	3.2. Diagrama de clases (Maybe no, pero al menos mención de la estructura programática del proyecto, lo que sea más fácil jijijijijijij)	3
1.2. Objetivo de la aplicación . . . . .	2		
<b>2. Diseño del protocolo</b>	<b>2</b>	<b>4. Uso y pruebas del protocolo</b>	<b>3</b>
2.1. Máquina de Estados Finita . . . . .	2	4.1. Manual de uso . . . . .	3
2.2. Descripción de los estados . . . . .	2	4.2. Demostración del funcionamiento (por caso de uso ijijisjiajij) . . . . .	3
2.3. Descripción de los mensajes en la comunicación cliente-servidor . . . . .	3		
2.4. Diseño de la base de datos . . . . .	3		
<b>3. Implementación del protocolo</b>	<b>3</b>		

# 1. Especificación de requerimientos

## 1.1. Enunciado del problema

De acuerdo con el modelo TCP/IP, la *Capa de Aplicación* es la de mayor importancia y en la que se sustenta todo el desarrollo de redes de computadoras pues está compuesta por los protocolos y demás servicios encargados de manejar, intercambiar o decodificar los datos que los usuarios se envían a través de distintos hosts para comunicarse. En otras palabras, se busca que dos o más procesos en distintas computadoras puedan ser capaces de intercambiar información y de esta manera otorgar una mayor capacidad de procesamiento y mejor rendimiento del que se tendría con un único host.

Sin embargo, la *Capa de Aplicación* no puede trabajar sola, siendo mediante un proceso de encapsulación en cada una de las capas, que los datos(PDU) se envían a las capas inferiores, añadiendo cada una de las capas información que le concierne para que sus protocolos puedan manejarlos.

Como sabemos, protocolos como HTTP, DNS, FTP, SMTP o DHCP tienen cada uno una estructura distinta ya bien definida y estandarizada en los artículos de *Request for Comments* publicados, pero eso no significa que sean los únicos protocolos disponibles en la *Capa de Aplicación*. La gran ventaja de esta capa es su adaptabilidad para que se desarrollen protocolos de acuerdo a las necesidades de la aplicación sobre la que se usarán.

## 1.2. Objetivo de la aplicación

Esta aplicación fue creada con el objetivo de implementar un protocolo de la capa de aplicación en el que el usuario, conectado del lado del cliente, solicite un pokémon a capturar. El servidor elegirá aleatoriamente alguno de los pokémon que estén en su base de datos, se lo ofrecerá al usuario y, si éste acepta capturarlo, también aleatoriamente se indicará si logró capturarlo o no.

El usuario también podrá ser capaz de consultar su Pokédex, con los nombres de los pokémon que ha capturado y la imagen de cada pokémon que seleccione.

El protocolo que hemos diseñado se enfocará en las acciones del usuario y el servidor que involucren una comunicación entre ambos, cada una con un tipo de mensaje específico. Es decir, si un usuario quisiera capturar un pokémon, solo tendría que enviar un mensaje con el código que el servidor entienda como "quiero capturar un pokémon" sin considerar otros aspectos como el nombre del pokémon o el tamaño de la imagen que contiene a tal pokémon.

En ese sentido, diseñar nuestro propio protocolo nos permite tener mayor control sobre la tasa de transferencia de los datos dentro de la aplicación, disminuyendo los costos de la comunicación y mejorando el desempeño del programa.

### 1.2.1. Casos de uso

Actor	Caso de uso	Descripción
Pokentrenador	Iniciar sesión	El usuario se conecta con el servidor y se le muestra el menú principal de la aplicación
Pokentrenador	Solicitar captura de pokémon	El usuario solicita al servidor que le muestre un pokémon para capturar
Pokentrenador	Consultar pokédex	El usuario hace una consulta para buscar un pokémon en su pokédex
Pokentrenador	Lanzar pokebola	El usuario intenta capturar el pokémon que le mostró el servidor, con un número de intentos limitado
Pokentrenador	Cerrar sesión	Cierra la conexión con el servidor

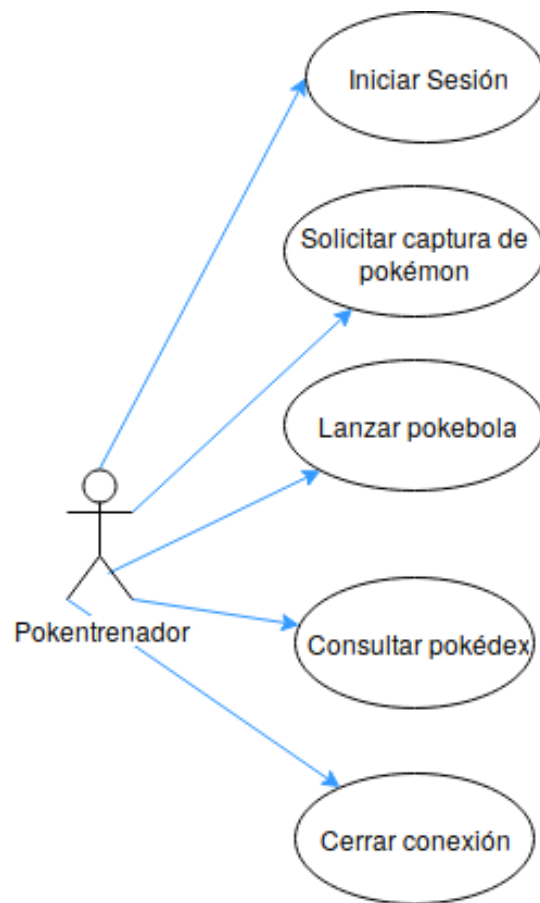
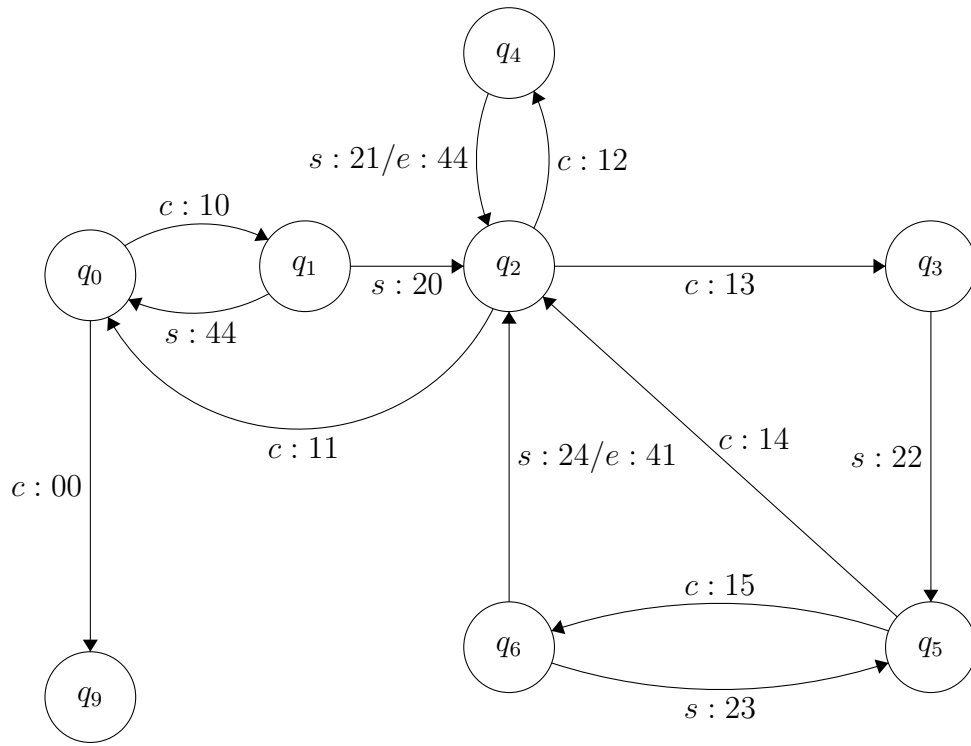


Figura 1: Diagrama de casos de uso de la aplicación

## 2. Diseño del protocolo

### 2.1. Máquina de Estados Finita

Máquina de Estados Finita para el protocolo de la capa de aplicación.



## 2.2. Descripción de los estados

Estado	Descripción
$q_0$	Conexión establecida, inicio de aplicación.
$q_1$	Inicio de sesión.
$q_2$	Menú de juego.
$q_3$	Solicitud de captura de <i>póke</i> mon.
$q_4$	Búsqueda de un <i>póke</i> mon en la <i>póke</i> dex.
$q_5$	Aparición de un <i>Póke</i> mon salvaje.
$q_6$	Intento de captura de <i>póke</i> mon.
$q_9$	Cierre de conexión.

## 2.3. Descripción de los mensajes en la comunicación cliente-servidor

Código	Segmento	Descripción
00	code num_con	Termina la conexión identificada con num_con.
10	code num_con name	Solicitud de inicio de sesión del cliente. El parámetro <b>name</b> se refiere al nombre de usuario de la persona conectándose a través del cliente y su tamaño está en el rango de 1 a 32 bytes.
11	code num_con	Solicitud de cierre de sesión del cliente.
12	code num_con name	Consulta del usuario a su <i>Pokédex</i> .
13	code num_con	Menú de captura.
14	code num_con	El usuario rechaza el pokémon ofrecido aleatoriamente.
15	code num_con	Intento de captura del pokémon ofrecido aleatoriamente.
20	code	inicio de sesión exitoso.
21	code long nombre longitud imagen	Pokédex query.
22	code name num_intentos_restantes	funcionamiento aleatorio.
23	code	Pokémon no capturado.
24	code  "2i"	Pokémon capturado.
41	code	Maximum Number of Attempts Reached.
44	code	Not Found Error.
46	code	No existe conexión.
47	code	Sin sesión.

## 2.4. Diseño de la base de datos

# 3. Implementación del protocolo

## 3.1. Especificación del ambiente de desarrollo

Lenguaje de programación:	Java	versión:	1.8
Herramienta de construcción de software:	Apache Ant	versión:	1.9.9
Sistema manejador de base de datos:	sqlite	versión:	2.8.17

- 3.2. Diagrama de clases (Maybe no, pero al menos mención de la estructura programática del proyecto, lo que sea más fácil jijijijijijij)
- 4. Uso y pruebas del protocolo
  - 4.1. Manual de uso
  - 4.2. Demostración del funcionamiento (por caso de uso ijijisjiajij)