

Trabajo Práctico Especial 2019/1  
Revisión 0

Resumen

Este documento describe el Trabajo Especial de la materia Protocolos de Comunicación para la cursada del primer cuatrimestre del año 2019.

Terminología

Las palabras clave "DEBE", "NO DEBE", "OBLIGATORIO", "DEBERÁ", "NO DEBERÁ", "DEBERÍA", "NO DEBERÍA", "RECOMENDADO", "PUEDE" y "OPCIONAL" en este documento serán interpretadas como se describe en el RFC 2119 [RFC2119].

Tabla de Contenidos

1. Requerimientos Funcionales . . . . .	2
1.1. Objetivo . . . . .	2
1.2. Formas de uso . . . . .	2
1.3. Concurrencia y Disponibilidad . . . . .	2
1.4. Fallos . . . . .	2
1.5. Métodos a implementar . . . . .	2
1.6. Registros de acceso . . . . .	3
1.7. Transformaciones de respuestas . . . . .	3
1.8. Métricas . . . . .	3
1.9. Configuración dinámica y monitoreo . . . . .	3
2. Requerimientos No Funcionales . . . . .	4
2.1. API de Sockets a utilizar . . . . .	4
2.2. Performance . . . . .	4
2.3. Argumentos de las aplicaciones . . . . .	4
3. Trabajo en Grupo . . . . .	4
3.1. Alumnos Recursantes . . . . .	5
3.2. Herramientas de colaboración . . . . .	5
4. Presentación Oral . . . . .	5
5. Entrega . . . . .	6
5.1. Material a entregar . . . . .	6
5.2. Forma de Entrega . . . . .	7
5.3. Fecha de Entrega . . . . .	7
5.4. Calificación . . . . .	7
5.5. Entrega tardía . . . . .	8
5.6. Código de Honor . . . . .	8
6. Normative References . . . . .	9

## 1. Requerimientos Funcionales

### 1.1. Objetivo

El objetivo del trabajo es implementar un servidor proxy para el protocolo HTTP versión 1.1 (Hypertext Transfer Protocol) [RFC7230] que pueda ser usado por un User Agents tal como curl, Mozilla Firefox, Internet Explorer y Google Chrome para navegar por Internet.

Los alumnos DEBEN demostrar habilidad para la programación de aplicaciones cliente/servidor con sockets, la comprensión de estándares de la industria, y la capacidad de diseñar protocolos de aplicación.

### 1.2. Formas de uso

El servidor proxy DEBE poder ser usado desde los User Agents con configuración previa de los mismos y como proxy transparente [RFC1919]. No se desea implementar un servidor cache (sección 13 del [RFC7230]).

### 1.3. Concurrencia y Disponibilidad

El servidor proxy DEBE soportar múltiples clientes de forma concurrente y simultánea. Se DEBE tener en cuenta en la implementación aquellos factores que afecten la disponibilidad del servicio.

### 1.4. Fallos

El servidor proxy DEBE reportar los fallos a los User-Agents usando toda la potencia del protocolo HTTP.

Por ejemplo, si el servidor no puede establecer la conexión al servidor porque no recibe respuesta de éste luego de cierto tiempo máximo de espera, podría reportar el problema al cliente con un código 504.

### 1.5. Métodos a implementar

Se DEBE soportar por lo menos los métodos GET, HEAD, PUT, DELETE y POST. Si no se soporta el resto de los métodos se DEBE comunicar al cliente con el status code apropiado.

### 1.6. Registros de acceso

El servidor proxy DEBE dejar registros de los accesos en la consola y/o en un archivo que permitan entender que peticiones están pasando por el proxy y su resultado.

Consejo: es conveniente observar como registran los accesos los servidores HTTP tradicionales. Imprimir todos los bytes que pasa por el proxy no es algo que se desea.

### 1.7. Transformaciones de respuestas

El servidor proxy DEBE implementar mecanismos que permita transformar el cuerpo de ciertas respuestas HTTP utilizando aplicaciones externas. De estar activada la funcionalidad se basará por lo menos en un predicado basado en media-types (por ejemplo text/\*). Las aplicaciones externas recibirán por entrada estándar el cuerpo de la respuesta y dejará en su salida estándar el cuerpo procesado.

### 1.8. Métricas

El sistema DEBE implementar mecanismos que recolecten métricas para entender el funcionamiento del sistema

- o conexiones concurrentes
- o cantidad de accesos históricos
- o bytes transferidos
- o cualquier otra métrica que considere oportuno para el entendimiento del funcionamiento dinámico del sistema

Las estadísticas PUEDEN ser volátiles (si se reinicia el servidor las estadísticas pueden perderse).

### 1.9. Configuración dinámica y monitoreo

Las transformaciones DEBEN poder modificarse en tiempo de ejecución (sin reiniciar el servidor). Las diferentes implementaciones PUEDEN decidir disponibilizar otros cambios de ejecución en tiempo de ejecución de otras configuraciones (memoria utilizada en I/O, timeouts, etc).

Las métricas (Sección 1.8) recolectadas también deben poder obtenerse en tiempo de ejecución.

Para cumplir con esto deberá diseñar e implementar uno o varios protocolos. Estos protocolos deberán funcionar sobre SCTP. Los protocolos DEBERÁN estar debidamente documentados.

DEBERÁ implementar una aplicación cliente que use dicho protocolo.

## 2. Requerimientos No Funcionales

Además de los requerimientos listados en el documento Lineamiento de Programación (Material Didáctico > Trabajo Práctico Especial), se suman los siguientes requerimientos no funcionales:

### 2.1. API de Sockets a utilizar

La implementación del proxy DEBE utilizar sockets en modo no bloqueante multiplexada.

### 2.2. Performance

Se DEBE tener en cuenta todos los aspectos que hagan a la buena performance, escalabilidad y disponibilidad del servidor. Se espera que se maneje de forma eficiente los flujos de información (por ejemplo no cargar en memoria mensajes muy grandes, ser eficaz y eficiente en el intérprete de mensajes). El informe DEBE contener información sobre las pruebas de stress. Algunas preguntas interesantes a responder son:

- o ¿Cual es la máxima cantidad de conexiones simultáneas que soporta?
- o ¿Cómo se degrada el throughput?

### 2.3. Argumentos de las aplicaciones

Tanto la aplicación servidor, como la aplicación cliente de configuración/monitoreo DEBERÁN manejar los argumentos de línea de comandos de cierta forma uniforme (por ejemplo -c <puerto> podría especificar el puerto utilizado para el protocolo de configuración/monitoreo). Los detalles de qué parámetros se deben manejar será publicado en otro documento.

## 3. Trabajo en Grupo

Para la realización del Trabajo Práctico los alumnos DEBERÁN conformar grupos y trabajar en equipo. Se espera la participación equitativa de todos los integrantes de un grupo durante la realización y exposición del trabajo. Todos los alumnos DEBEN demostrar que han participado de la realización del trabajo y que han

desarrollaron las competencias alineadas a los objetivos del trabajo (Sección 1.1).

### 3.1. Alumnos Recursantes

Los alumnos que se encuentren recursando la materia y que hayan aprobado en su última cursada un Trabajo Práctico Especial de similares características al que se describe en este documento pueden optar entre las siguientes alternativas:

1. Realizar el Trabajo Práctico Especial descrito en este documento.

El mismo deberá ser aprobado. Se tomará la calificación que surja de la misma. En caso de desaprobación no se tendrá en cuenta ejecuciones pasadas.

2. No realizar el Trabajo Práctico Especial de este cuatrimestre.

El alumno mantendrá la nota del último Trabajo Práctico Especial de similar características aprobado. El alumno DEBE exteriorizar dicha decisión mediante el canal adecuado (tablero de discusión) antes de la fecha límite de inscripción de grupos.

### 3.2. Herramientas de colaboración

Los grupos tendrán a su disposición las herramientas provistas por el módulo Grupos de Campus ITBA y un repositorio GIT Bitbucket para el código (y otros archivos). Se espera transparencia en el desarrollo del trabajo. Como ejemplo a un comportamiento poco transparente se puede describir la utilización de otro repositorio de código.

Es posible que durante la ejecución del Trabajo Práctico se publiquen pequeñas asignaciones a resolver mediante Campus ITBA. La intención de dichas preguntas será evaluar el avance del trabajo y disparar discusiones que pueden enriquecer el trabajo.

### 4. Presentación Oral

El Jueves 30 de Mayo durante la clase teórica cada grupo dispondrá de no más de 15 minutos donde DEBERÁ exponer un resumen del trabajo, detallando como han organizado técnicamente el trabajo, describiendo los problemas encontrados y las técnicas utilizadas para solucionarlos.

## 5. Entrega

### 5.1. Material a entregar

Cada grupo DEBE entregar todo el material necesario para reproducir el Trabajo Práctico. Como mínimo DEBE contener:

- a. Un informe en formato PDF [RFC3778] o text/plain (con codificación UTF-8) que contenga las siguientes secciones (respetando el orden):
  1. Índice (por lo tanto las hojas deberán estar numeradas).
  2. Descripción detallada de los protocolos y aplicaciones desarrolladas.
  3. Problemas encontrados durante el diseño y la implementación.
  4. Limitaciones de la aplicación.
  5. Posibles extensiones.
  6. Conclusiones.
  7. Ejemplos de prueba.
  8. Guía de instalación detallada y precisa. No es necesario desarrollar un programa instalador.
  9. Instrucciones para la configuración.
  10. Ejemplos de configuración y monitoreo.
  11. Documento de diseño del proyecto (que ayuden a entender la arquitectura de la aplicación).
- b. Documento utilizados como soporte durante la presentación. Deben estar en formato PDF [RFC3778].
- c. Códigos fuente y archivos de construcción
- d. Un archivo README en la raíz que describa al menos:
  - \* la ubicación de todos los materiales previamente enumerados
  - \* el procedimiento necesario para generar una versión ejecutable de las aplicaciones

- \* la ubicación de los diferentes artefactos generados
- \* cómo se debe ejecutar las diferentes artefactos generados (y sus opciones)

#### 5.2. Forma de Entrega

La entrega se realizará por Campus ITBA en la asignación creada para ello.

Se DEBE entregar un tarball que sea el producto de clonar el repositorio GIT (por lo tanto el repositorio GIT DEBE contener todos los materiales de entrega (Sección 5.1)), y su historia.

#### 5.3. Fecha de Entrega

Las entregas se podrán realizar hasta el Martes 18 de Junio a las 16.05hs (horario Buenos Aires). Una vez realizada la entrega los grupos DEBERÁN mostrar el correcto funcionamiento del sistema con casos de prueba provisto por los equipos y provistos ese día por la Cátedra.

#### 5.4. Calificación

Para aprobar el Trabajo Práctico se DEBE cumplir TODAS las siguientes condiciones:

- o El material entregado DEBE estar completo (por ejemplo no se puede corregir si falta el informe o alguna clase)
- o Se utilizan únicamente las librería permitidas para los usos definidos.
- o DEBE ser correcta las cuestiones de entradas/salida no bloqueante. Por ejemplo las lecturas, escrituras y el establecimiento de nuevas conexiones DEBEN ser mediante suscripciones y no bloquearse.
- o DEBE ser correcta las cuestiones relacionadas a la lectura/ escrituras parciales.
- o Sumar CUATRO puntos de calificación.

La calificación ocurre dentro de una escala de DIEZ puntos destinando...

- o CINCO puntos a la evaluación de la funcionalidad básica. El guión de prueba será publicado unos días antes de la entrega. Cada prueba tendrá un puntaje asignado.
- o TRES puntos a la correctitud de implementación, el diseño del código (selección adecuada de algoritmos, estructura de programa limpia, etc;) y al estilo (¿está debidamente comentado?, ¿es legible? ...)
- o DOS puntos a la participación de las asignaciones intermedias, la presentación oral, y el informe.

Sumará UN punto adicional a la calificación final aquel grupo que habiendo aprobado el Trabajo Práctico logre el mejor throughput entre todos los grupos en un escenario provisto por los docentes, descrito utilizando JMeter (o similar).

La calificación es de carácter grupal; pero si hay evidencias de que un alumno de un grupo no participó en la elaboración, o éste no puede defender o demostrar su participación, entonces el alumno no podrá aprobar el Trabajo Práctico.

#### 5.5. Entrega tardía

Las entregas que ocurran entre 0 horas (inclusive) y 24 horas (exclusivo) luego de la fecha de entrega especificada perderán 25% de la calificación.

Las entregas que ocurran entre 24 horas (inclusive) y 48 horas (exclusivo) luego de la fecha de entrega especificada perderán 50% de la calificación.

Las entregas que ocurran entre 48 horas (inclusive) y 72 horas (exclusivo) luego de la fecha de entrega especificada perderán 75% de la calificación.

Luego de 72 horas el trabajo será considerado NO APROBADO, con una calificación de CERO puntos.

#### 5.6. Código de Honor

Las entregas deben ser creaciones originales de los alumnos de cada grupo. Estas creaciones pueden ser resultado de discusiones en general o en particular en las clase (clase teórica, discusiones prácticas, exposición de grupos), pero las implementaciones DEBEN ser originales. Está permitido utilizar parcialmente o totalmente el código fuente presentando por la cátedra en clase.



## 6. Normative References

- [RFC1919] Chatel, M., "Classical versus Transparent IP Proxies", RFC 1919, DOI 10.17487/RFC1919, March 1996, <<http://www.rfc-editor.org/info/rfc1919>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3778] Taft, E., Pravetz, J., Zilles, S., and L. Masinter, "The application/pdf Media Type", RFC 3778, DOI 10.17487/RFC3778, May 2004, <<http://www.rfc-editor.org/info/rfc3778>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.