

# Redes y Comunicaciones

## Trabajo Práctico

2do Cuatrimestre de 2020

Documento	Nombre	Evaluación Individual
Evaluación Trabajo		

## Objetivo

Desarrollar en lenguaje C o C++ un sistema de reserva de pasajes en ómnibus basado en dos aplicaciones que se ejecutarán en modo consola, una con el rol de servidor y otra con el rol de cliente, comunicadas por sockets.

El sistema será utilizado por una empresa de transporte de pasajeros que realiza viajes de ida y vuelta entre los siguientes destinos:

- Buenos Aires
- Mar del Plata

Esta empresa cuenta con unidades como para proveer servicios en 3 turnos: mañana, tarde o noche.

## Requerimientos funcionales

### Servidor

- Inicia y espera la conexión del cliente. Para esto hará las siguientes validaciones:
  - a) Cuando un cliente intente conectarse, el servidor deberá verificar que no haya otro cliente conectado. En caso de haber un cliente conectado deberá seguir uno de los siguientes flujos:
    1. Si la sesión del cliente conectado expiró (**2 minutos sin actividad**), lo desconecta y acepta la conexión del nuevo.
    2. Si la sesión del cliente conectado está activa, niega la conexión del nuevo cliente enviándole el mensaje *"Solo puede haber un cliente conectado a la vez, por favor inténtelo más tarde"*.
  - b) Al aceptar la conexión de un cliente, el servidor deberá pedir un usuario y contraseña, y deberá seguir uno de los siguientes flujos:
    1. Si el usuario y contraseña son correctos, se permite el ingreso al sistema.
    2. A los 3 intentos fallidos, el servidor rechaza la conexión informando al cliente *"Se superó la cantidad máxima de intentos de ingreso"*.
  - c) Cuando un cliente sale, el servidor se queda esperando a otro.
- Toda la actividad relacionada con la administración de conexiones y sesiones de usuarios realizada por el servidor debe quedar registrada en un archivo de texto llamado **sever.log**
  - A modo de ejemplo:

**Archivo:** server.log

**Contenido:**

```
2020-09-20_11:50: =====
2020-09-20_11:50:          Inicia Servidor
2020-09-20_11:50: =====
2020-09-20_11:50: Socket creado. Puerto de escucha: 5000
....
```

- Las credenciales de los usuarios se consultan desde un archivo de texto con duplas, por ejemplo:

*Usuario\_1;password\_1*

*Usuario\_2;password\_2*

*Usuario\_n;password\_n*

- Los nombres de usuario pueden ser cualquier cadena de hasta 12 caracteres sin espacios (por ejemplo: Pedro, Juan\_2020, etc)
- Toda la actividad realizada por cada usuario debe quedar registrada en archivos de texto independientes.
  - Debe haber un archivo por usuario, y cada vez que éste se conecta al servidor y realiza diferentes acciones, estas se van agregando al mismo.
  - Se recomienda identificar claramente el inicio de cada sesión.
  - A modo de ejemplo:

**Archivo:** usuario\_1.log

**Contenido:**

```
2020-09-20_12:30: =====
2020-09-20_12:30:          Inicia sesión
2020-09-20_12:30: =====
2020-09-20_12:34: IdServicio - Reserva_A1
2020-09-20_12:36: IdServicio - Libera_C10
2020-09-20_12:40: Cierra sesión

2020-09-20_17:30: =====
2020-09-20_17:30:          Inicia sesión
2020-09-20_17:30: =====
...
```

- La información de los servicios se almacena en el servidor utilizando un archivo binario.
- Se debe poder visualizar el contenido total de este archivo binario, convirtiéndolo a un formato legible (puede ser a través de una tercera aplicación, también desarrollada en lenguaje C o C++ ).

## Cliente

- Para conectarse al servidor, primero deberá especificar la dirección IP y el puerto, una vez que el servidor acepte la conexión, deberá introducir un usuario y contraseña.  
En caso que el servidor niegue la conexión, se deberá mostrar el mensaje recibido y mostrar las opciones para conectarse nuevamente.
- Ya conectado al sistema, el cliente dispone de las siguientes opciones:
  - Alta servicio
  - Gestionar pasajes
  - Ver registro de actividades
  - Cerrar sesión
- **Alta servicio:** esta opción permite crear un servicio. Para ello el usuario debe ingresar el Origen, la fecha y el turno.  
Con estos datos se consulta al servidor, que deberá verificar si ya está creado este servicio y si no lo está, deberá crearlo (asumiendo el ómnibus vacío). En ambos casos el servidor debe devolver el resultado de la operación solicitada, y el cliente debe notificar adecuadamente al usuario.
- **Reservar pasajes:** esta opción permite reservar pasajes. Para ello el usuario puede realizar consultas por Origen, por fecha, por turno, o por combinación de estas. El servidor devolverá un listado de los servicios que cumplan con dicho criterio.

El usuario puede entonces elegir entre alguno de los servicios disponibles, para realizar la reserva de pasajes, o crear un nuevo servicio (**Alta servicio**).

Si el usuario desea gestionar pasajes en alguno de los servicios disponibles, debe indicar cual, y se presentará por pantalla el esquema del ómnibus según el estado de ocupación, a modo de ejemplo:

[illegible]

Donde cada letra representa:

O = Asiento libre

X = Asiento ocupado

Los micros son todos de un piso, con 20 asientos por fila, dispuestos en un sector de asientos dobles de un lado, el pasillo y una fila de asientos individuales.

Teniendo la disposición de los asientos en pantalla, se presenta un menú con las acciones correspondientes:

- Reservar un asiento
- Liberar un asiento
- Elegir otro servicio
- Volver al menú anterior

- **Ver registro de actividades:** mediante esta función se solicita al servidor que transfiera al cliente el archivo de registro de actividades del usuario activo. Una vez recibido, se muestra por pantalla el contenido del mismo.

- **Cerrar sesión:** el cliente se desconecta del servidor dejándolo libre para recibir nuevas conexiones.
  - Se recomienda utilizar menús para simplificar la selección de las diferentes opciones.
  - La presentación de los datos debe ser clara (borrando la pantalla cuando corresponda, presentando páginas si las opciones son demasiadas, etc.).

## Requerimientos no funcionales

Lenguaje de programación C/C++.

Entorno de desarrollo a elección del grupo.

Sistema operativo Linux o Windows, a elección del grupo.

## Presentación

19/10/2020: **Consultas.**

01/11/2020 09:59hs: **Entrega del TP vía Campus Virtual**

02/11/2020: **Defensa del TP.**

15/11/2020 09:59hs: **Entrega del recuperatorio del TP vía Campus Virtual**

16/11/2020: **Recuperatorio del TP.**

## Normas de entrega.

El trabajo entregado deberá contener un documento incluyendo:

- El enunciado del trabajo práctico (TODO este documento, incluyendo el anexo).
- La estrategia de resolución del trabajo práctico. Es un texto descriptivo de cómo se estructuró la aplicación, cómo se separaron las capas, relaciones entre las entidades, es decir, todo aquello que consideren significativo para explicar la resolución del trabajo.
- El código fuente del proyecto anexado en una carpeta aparte o subido a un repositorio público (Github, GDrive, etc).
- Todas las hojas deben estar numeradas.
- Video (o enlace a video) en formato mp4 mostrando las pruebas del sistema. La especificación de las pruebas a realizar será indicada oportunamente.

El incumplimiento de cualquiera de las normas de entrega implicará la desaprobación del trabajo práctico.

## Evaluación:

La Evaluación de los trabajos prácticos contará con una etapa grupal y una individual.

- Grupal: Se realizará un conjunto de pruebas sobre el trabajo presentado por los alumnos en presencia de los mismos. Se deberá aprobar la totalidad de las pruebas.
- Individual: Se realizará una evaluación individual oral o escrita para cada alumno. Los temas a evaluar podrán ser, por ejemplo: preguntas sobre el trabajo práctico, codificación de alguna primitiva o modificación del trabajo práctico, etc.

La nota final del trabajo se calculará en función de las notas obtenidas en forma grupal e individual. La nota grupal será el promedio entre la primera presentación y el recuperatorio (en caso de necesitarlo). Por este motivo, SOLO deberán presentarse aquellos grupos que hayan concluido TODO el trabajo práctico ya que no se harán evaluaciones parciales.

## Cuestiones de autoría:

Todo el código debe ser desarrollado íntegramente por cada grupo.

No se permite la reutilización de código de cuatrimestres anteriores o de otras materias.

Ante cualquier duda se deberá consultar con los docentes.

La reutilización de código sin consulta previa será condición suficiente para la desaprobación de la materia.

## Anexo – Correcciones

[illegible]