

UNIVERSIDAD DE BUENOS AIRES

TALLER DE PROGRAMACIÓN (75.42)

**DeBondis.com**

*Cristian González*

*94719*

# Índice

<b>1. Clases</b>	<b>2</b>
1.1. Server . . . . .	2
1.2. Client . . . . .	2
1.3. Date . . . . .	2
1.4. WaitCharacter . . . . .	2
1.5. TimeStop . . . . .	2
1.6. WaitClient . . . . .	2
1.7. Thread . . . . .	2
1.8. Colectivo . . . . .	2
1.9. Colectivo Recorrido . . . . .	2
1.10. Attend Client . . . . .	2
<b>2. Links</b>	<b>3</b>

## **1. Clases**

### **1.1. Server**

Modela la responsabilidad del servidor, se encarga de subir toda la información a partir de los archivos, cómo también hacer que todos los hilos puedan acceder a toda la información sobre los recorridos y los colectivos en tránsito.

### **1.2. Client**

Se encarga de poder interpretar los comandos y enviar la información pertinente para su consulta, una vez que tenga una respuesta del server (ya sea un error o no) imprimir el resultado por pantalla al cliente.

### **1.3. Date**

Diseña esta clase para encapsula la fecha y hora, a modo de encapsular un poco el comportamiento y poder usarlo con mayor comodidad, por ejemplo haciendo sobrecarga del operator -. También tiene una responsabilidad de dado un timeLinux poder setear todas las variables, hora, día, etc.

### **1.4. WaitCharacter**

Clase que hereda de Thread que encarga de esperar por stdin el caracter "qz en caso de que suceda eso, avisar el server que debe .pagarse".

### **1.5. TimeStop**

Esta se encarga de encapsula el tiempo que existe entre 2 Paradas, de tal forma poder consultarle el tiempo que existe entre las paradas, si cierta parada es el first o el end.

### **1.6. WaitClient**

Se encarga de esperar una nueva conexión, una vez que un nuevo cliente se conecta, este se encarga de lanzar otro hilo para atenderlo, cómo también se encarga de dada un vez conexión chequear cómo se encuentran los otros clientes (si el cliente de cierto hilo tiene alguna nueva consulta). Para la decisión de chequear cada cierto tiempo fue a modo de que supongo que el lapso entre que se conecta un cliente y otro no es mucho.

### **1.7. Thread**

Se encarga de de modelar el comportamiento de un thread, para crearlo se tiene que proveer los datos que va a necesitar ese thread como la función/método a ejecutar.

### **1.8. Colectivo**

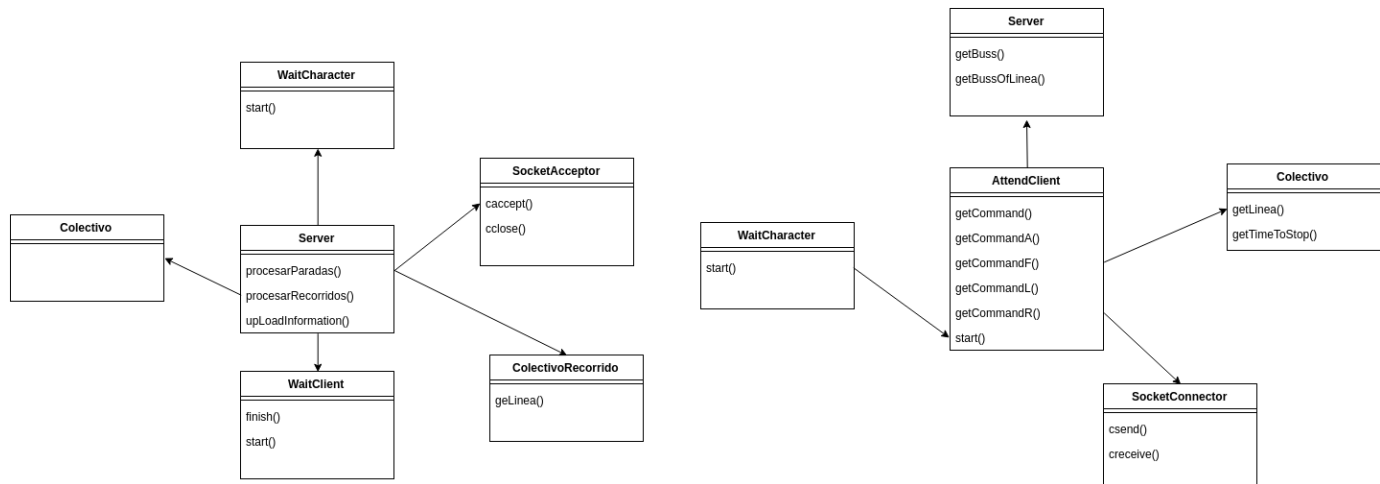
Se encarga de ser contener la hora y fecha en la cual salió y qué recorrido tiene.

### **1.9. Colectivo Recorrido**

Este encapsula todo el recorrido de un colectivo, es la información que se sube de paradas.txt y bondis.txt de tal forma que en esta clase se puede obtener toda la información respecto a tiempo para llegar de una parada a otra o sí pasa por cierta parada.

### **1.10. Attend Client**

Esta es la clase más compleja, se encarga de resolver las consultas de un cliente, poder pedirle al server toda la información necesaria para resolver dicha consulta y enviar al cliente la respuesta de la query.



## 2. Links

Link del repositorio: <https://github.com/Cristian3629/DeBondis.com>