

Sesión 4

Análisis de prestaciones

Alejandro Mengual

Carlos Peris

Raul Pina

David Sanchis

- 1. Proyecto**
 - 1.1. Máximas prestaciones**
 - 1.1.1. LAN**
 - 1.1.2. WAN**
- 2. Estadísticas**
 - 2.1. Gráficas**
 - 2.1.1. Escalabilidad de clientes**
 - 2.1.2. Incremento de mensajes por aumento de vecinos**
 - 2.1.3. Detección de cuellos de botella**
 - 2.1.4. Análisis del throughput**
- 3. Equipo**
 - 3.1. Planificación inicial en contraposición con la final**
 - 3.2. Lecciones**

Proyecto

Máximas prestaciones

Las máximas prestaciones son diferentes según qué conexión usemos, si es en local, para los equipos más potentes (como la torre de David o el servidor de Alex) la simulación les corre sin errores al máximo de capacidad de su red (esta no admite más conexiones por el buffer de recepción).

Sin embargo, equipos menos potentes con diferencia soportan menos (como la torre de Carlos, la cual soporta 8000).

LAN

En el caso de la conexión en LAN, llega a 3200 (a lo cual puede añadirse luego clientes en local hasta el límite del buffer de recepción).

Pruebas realizadas con:

Portátil Carlos: 2 procesos de 250 clientes

Portátil Raul: 2 procesos de 400 clientes

Portátil Alex: 2 procesos de 550 clientes

Portátil David: 2 procesos de 400 clientes

Torre David: Servidor y lanza 2 procesos con 8000 clientes en local

La conexión fue mediante cables ethernet. Como al router de David no se le podían conectar más de 4 cables, uno de ellos se conectó a un router que trajo Alex, al cual le conectamos el resto de cables ethernet.

También probamos a meterle móviles por WIFI. Pero debido a que el WIFI es más lento y que además solo puede atender a un solo dispositivo a la vez (a lo cual se le suma que como tienen procesadores más pequeños, en los móviles las iteraciones van más lentas) el uso de este solo terminaba lastrando la simulación, limitando el número máximo de otros equipos que podrían dar más pero que tenían que bajar para no colapsar la red, así que decidimos eliminarlo.

WAN

Las pruebas en WAN son realmente poco fiables, pero queríamos ver cómo iba con conexiones muy lejanas. Terminó funcionando con 1200 clientes (a falta de añadir los del portátil de Raúl, pero no nos pudimos coordinar para estar todos. De todas formas, dudamos que fuese a subir mucho más).

Pruebas realizadas con:

Torre Carlos: 2 procesos de 50 clientes

Portátil Carlos: 2 procesos de 25 clientes

Torre Alex: 2 procesos de 150

Portátil Alex: 2 procesos de 75 clientes

Portátil David: 2 procesos de 100 clientes

Torre David: 2 procesos de 200

Servidor Alex: Servidor (no le metimos en local porque realmente no eran unas pruebas tan definitivas)

En este caso, todos los portátiles se conectaban por WIFI.

Estadísticas

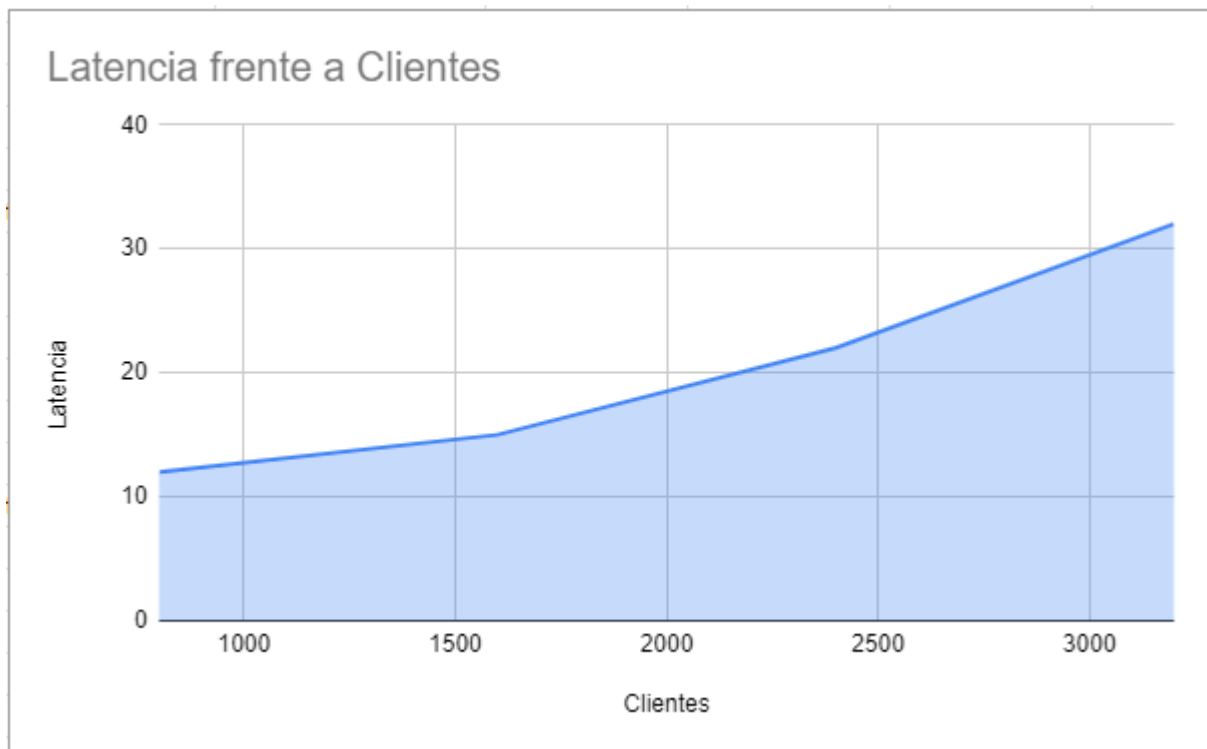
A continuación, expondremos detalladamente los datos estadísticos que hemos recopilado simulando el proyecto. Cabe destacar que los datos que pueden ofrecernos conclusiones reales son los realizados en LAN, ya que en WAN hay demasiada interferencia debido a la cantidad de dispositivos intermediarios y las distancias entre el servidor y los clientes. Sin embargo, incluiremos las gráficas de las pruebas realizadas en WAN para que sea visible el aumento entre el máximo y mínimo obtenidos en simulaciones.

Gráficas

Las siguientes gráficas muestran los valores máximos y mínimos para cada uno de los parámetros. Ello quiere decir que las gráficas muestran también cómo puede variar los resultados según las exigencias que piden cada conjunto de parámetros.

Escalabilidad de clientes

Podemos observar cómo las latencias aumentan en función del número de clientes. Podemos explicarlo por una sencilla razón: Aunque los mensajes dentro de los grupos sean en esencia los mismos, ya que el número de vecinos no aumenta, sí lo hace el reparto en CPU de los hilos que actúan como clientes, aumentando así el tiempo que se tarda en responder a cada iteración.



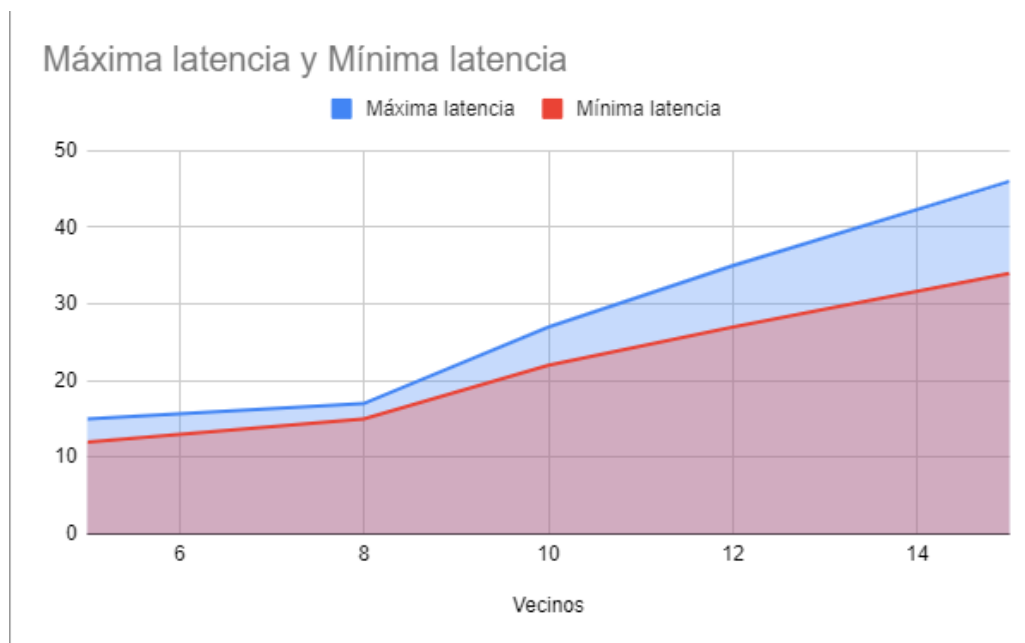
Podemos observar cómo la gráfica se puede asemejar a la función:
 $\text{Latencia}(\text{NumClientes}) ::= 10 + 3 \cdot 2^{\text{NumClientes}}$

Esta función, al menos para los primeros parámetros, nos ofrece una alta capacidad predictiva, ya que el error no sería muy elevado si calculamos la diferencia entre lo que nos devuelve la función y los datos reales.

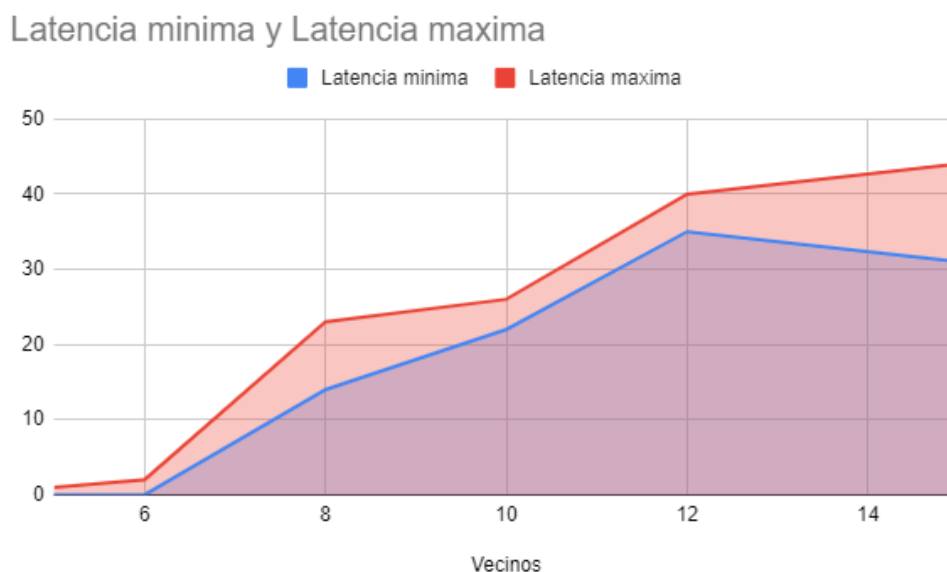
Incremento de mensajes por aumento de vecinos

Mediante esta gráfica nos es posible valorar el grado de aumento de mensajes necesarios para comunicar un grupo con más vecinos, ya que al tener que esperar más mensajes de confirmación, la latencia aumentará.

WAN:



LAN:



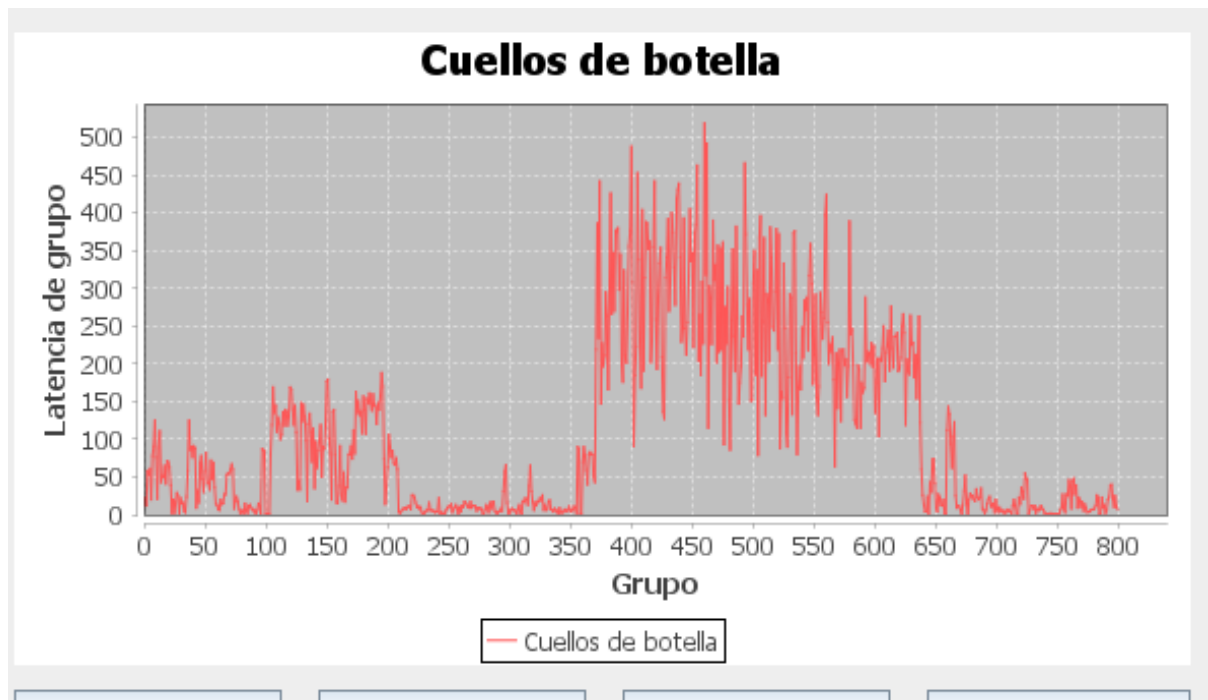
Podemos apreciar que evidentemente la latencia aumenta conforme más vecinos hay. Pero además, si quisiéramos saber qué nos es más rentable (suponiendo que queremos sacar el máximo número de vecinos pero que también nos importa la latencia), nos saldría una gráfica tal como esta (vecinos/latencia):



Resulta curioso que (obviando el caso de 5 y 6 vecinos), en el punto de 10 vecinos nos compensaría en mayor medida. Por lo tanto, tras sacar esta gráfica decidimos que todas las pruebas serían con 10 vecinos por grupo, puesto que es un número significativo de mensajes transmitidos y a su vez es lo más óptimo a nivel de latencia.

Detección de cuellos de botella

Si observamos cada grupo individualmente qué latencia tiene, la gráfica tiene un aspecto tal como este:

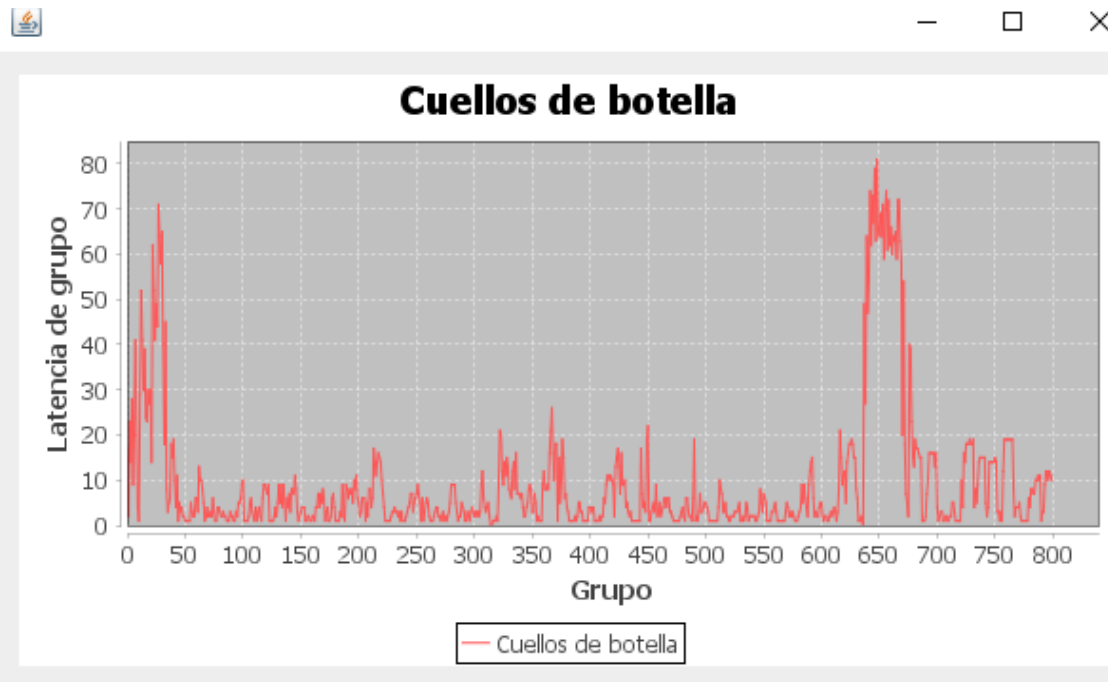


Se puede observar como hay oscilaciones y en ciertos puntos se producen picos. Bien, eso es debido a que la CPU se llena de hilos que gestionar y todos los clientes tardan más en acceder a ella para contestar sus respectivos mensajes. Por lo tanto, para apaciguar de alguna manera ese colapso de CPU, decidimos establecer una función que permitiese a la CPU ir más desahogada, siendo la salida de la función el tiempo que se debe esperar entre el lanzamiento de un grupo y otro (cuestión de milisegundos).

La función escogida, debido al carácter oscilatorio de la función, fue la siguiente:
 $\text{Espera}(\text{NumGrupo}) ::= x \cdot \sin(\text{NumGrupo}) + x$

Siendo x un valor que le otorgamos nosotros, según queramos espaciar el lanzamiento. No es recomendable que el número sea muy alto, pues no se conseguiría el efecto de "simultaneidad" deseado. Y por supuesto, el tiempo de espera es entre lanzamiento de grupos, pero una vez se comienza a lanzar uno, se lanza por completo, todos los clientes a la vez.

Una vez implementada dicha función y con una $x = 5$, el gráfico con los mismos parámetros nos salió de la siguiente manera:

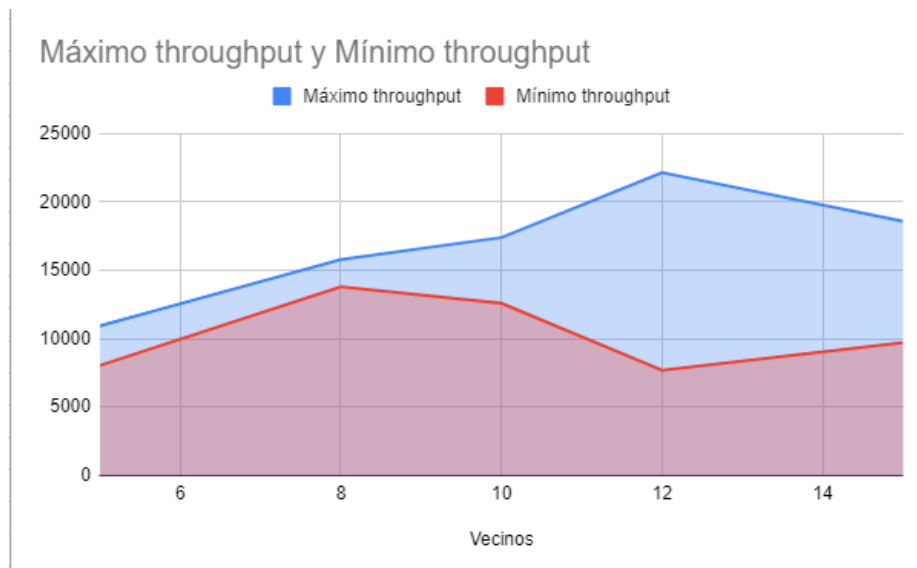


Se puede apreciar que incluso cuando llega a los picos, la latencia del grupo en cuestión se trata casi de los mínimos de la de arriba.

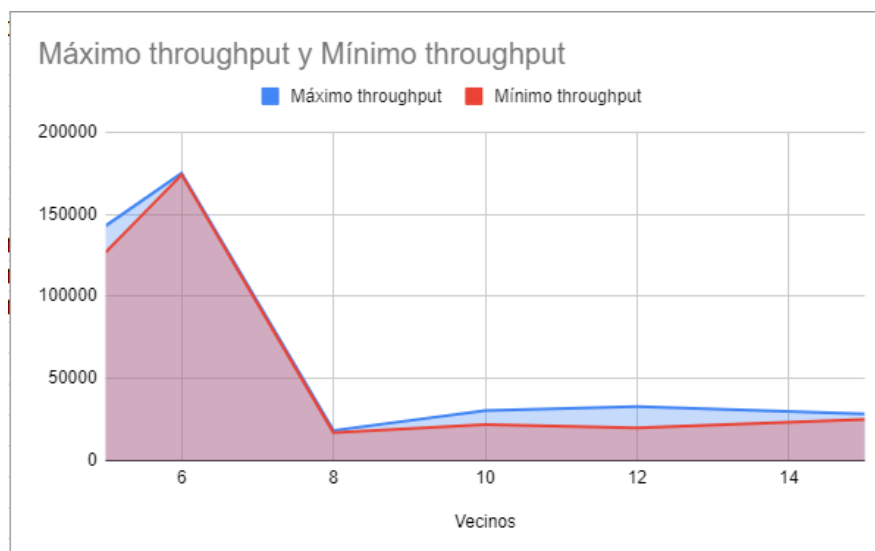
Análisis del throughput

Por último, la gráfica del throughput puede que sea la más importante de todas, pues al final lo que nos interesa es que haya el mayor flujo posible de datos por unidad de tiempo en el sistema, y a ser posible de manera constante.

WAN:



LAN:



Visiblemente es muy notable la diferencia entre simular la aplicación en LAN y WAN. En WAN los cambios en el throughput no tienen ninguna correlación, es casi un caos puesto que influye mucho las interferencias en la red.

Sin embargo, en LAN podemos sacar conclusiones como las siguientes:

El throughput mejorará cuanto menor sea la latencia resultante (puesto que la misma información en menor tiempo es un mayor flujo de la información) y también mejorará en relación con el ancho de banda de la red, puesto que permitirá la simultaneidad de más mensajes al mismo tiempo.

Sin embargo, empeorará si usamos WIFI en lugar de red cableada (los mensajes tardan más) o si aumenta la distancia entre hosts. El hecho de que los mensajes tarden más en llegar implica que los hilos esperarán durante más tiempo. Por lo tanto enviarán menos mensajes por unidad de tiempo.

No tiene por qué estar directamente relacionado con el aumento de vecinos, puesto que a menor número de vecinos, disminuye la latencia pero también la cantidad de mensajes mandados, y por lo tanto los bytes totales.

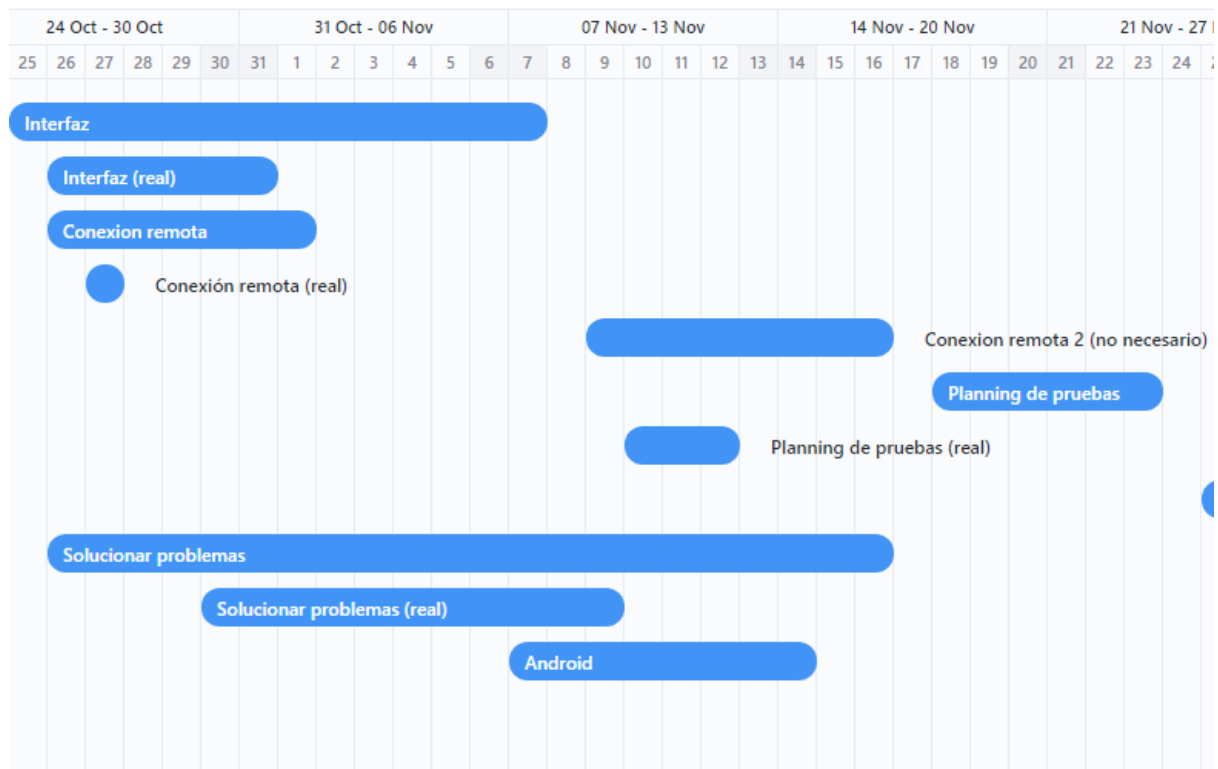
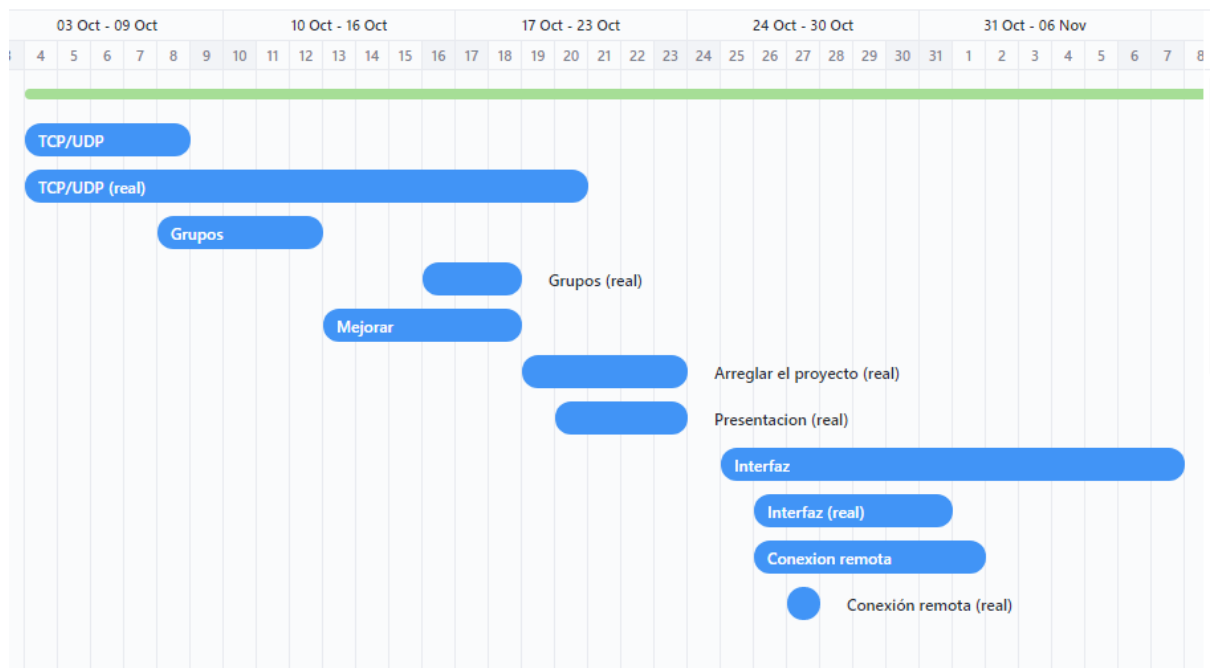
Equipo

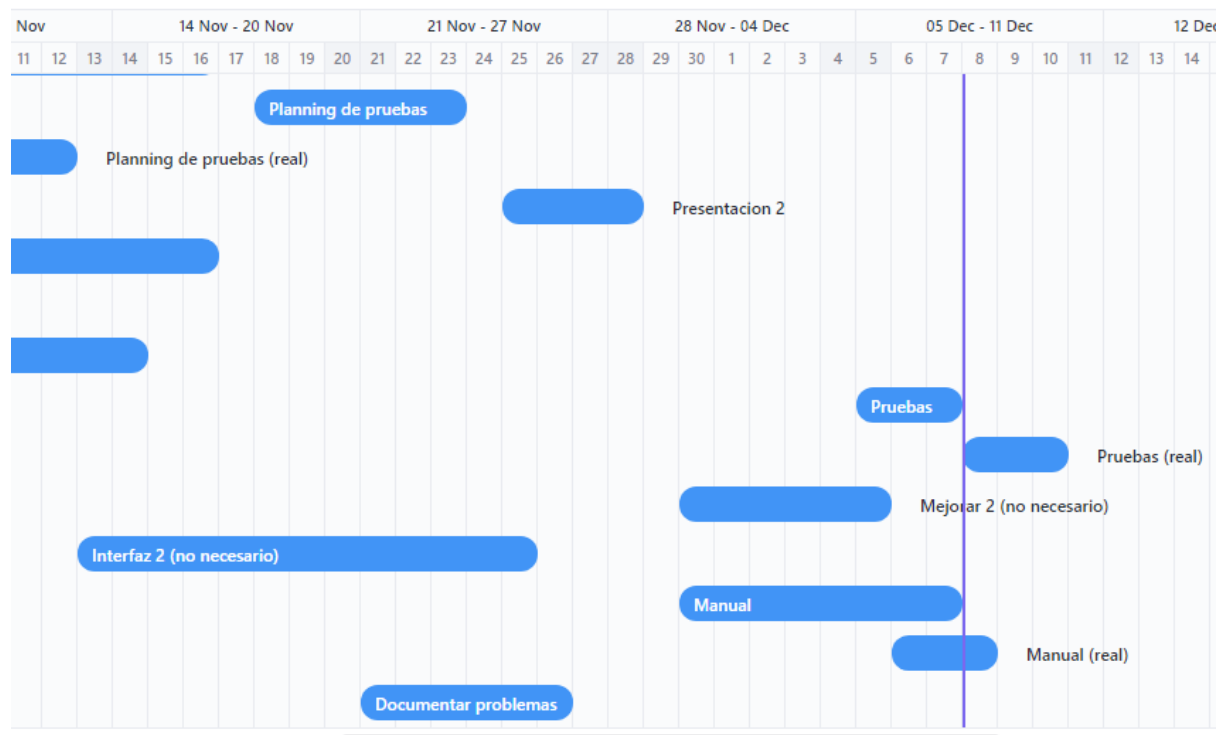
En este último punto explicaremos cómo ha sido el trabajo desde un punto de vista interno del equipo, puntos donde el proyecto se ha complicado y otros puntos donde sobreestimamos el coste que llevaría realizar ciertas actividades.

Planificación inicial en contraposición con la final

Durante el desarrollo del proyecto hemos anotado cada actividad cuánto nos costó en horas, así como las fechas de inicio y la que se podría considerar que la funcionalidad está implementada.

Se puede observar todos los cambios en el siguiente [Diagrama de Gantt](#).





Los puntos que han cambiado más significativamente el proceso de organización son los siguientes:

- Falta de coordinación por parte del equipo a la hora de hacer actividades que requieren de actividad conjunta (incompatibilidades de horarios, etc). Este punto ha hecho que el trabajo individual tome mucha más fuerza, dejando el trabajo en grupo para días concretos en los que se probaban muchas cosas (pero producía retrasos). Esta incompatibilidad ha sido asumible ya que tenemos varios dispositivos por persona y hemos podido usar la conexión remota de forma individual (entre varios dispositivos).
- Los retrasos principales fueron producidos porque el traslado a TCP/UDP no era del todo correcto y por insistir demasiado terminamos robando tiempo a muchas funcionalidades más importantes.

Lecciones

Durante el desarrollo del proyecto, y gracias a la experiencia adquirida a lo largo del mismo, nos hemos dado cuenta de que si a día de hoy hubiésemos tenido que comenzar desde cero (incluso sin tener ninguna información, tan solo con la experiencia adquirida de investigación) habríamos tomado una serie de decisiones diferentes.

- Primero de todo, pondríamos un orden de prioridades estricto, dejando las adiciones y mejoras para cuando lo primordial fuese completamente funcional, y no dejándolo al mismo nivel todo. Es decir, asegurarnos de tener un buen esqueleto y funcionalidades básicas.
- En segundo lugar, tomaríamos las elecciones importantes (como entorno de desarrollo, SO) con más confianza ya que esta era la primera vez que nos enfrentábamos a un proyecto desde cero y sin demasiadas pistas. Pero ahora que hemos adquirido la experiencia de investigar y tomar nuestras decisiones como equipo, tomaríamos decisiones que, aunque más arriesgadas por lo desconocido del entorno, serían mejores en cuanto a las prestaciones que nos otorgaría.
- Por último, ha habido un elemento que ha lastrado en gran medida la eficiencia del equipo, y ha sido la falta de coordinación. Es decir, la incompatibilidad de horarios, porque aunque todos hayamos invertido muchas horas en que todo saliese bien y hemos tocado un poco de todas las partes (no ha habido tareas aisladas para cada persona, sino que todos hemos participado y comprendido cada parte del proyecto) costaba mucho quedar para organizar tutorías o para hacer las pruebas y simulaciones. Es por ello que una lección que nos llevamos es la necesidad de hacer una tabla al principio del proyecto donde queden registrados los horarios de cada uno, indicando en qué momentos está disponible para reuniones, y a ser posible intentar reservar esos momentos para tenerlas.