

### **Organización del Grupo:**

- Para comenzar, en cuanto a la comunicación entre Lex y Yacc, se decidió transmitir toda la información mediante tokens. A partir de allí, Boselli realizó una implementación inicial del Analizador Léxico mientras que González y Etcharren realizaban la correspondiente al Analizador Sintáctico.
  - Al finalizar las implementaciones "alfa", se solucionaron en conjunto los conflictos shift/reduce y se refinaron ambos analizadores. Luego, para los casos de testing: comentarios, "end" y "begin" balanceados, a cargo de Boselli, if anidados, funciones, while, a cargo de González y Etcharren a cargo de la implementación para descartar correctamente los comentarios largos. Se realizó una última reunión donde se revisaron en conjunto los puntos a mejorar para dar por finalizada esta etapa.
- 

### **Decisiones de Diseño:**

- Comunicación entre Lex y Yacc mediante tokens. Esto será útil en un futuro para el analizador semántico.
  - Crear una librería "types.c" donde se definieron todos los tipos a utilizar. Para el análisis léxico y sintáctico se crearon dos de ellos: *TokenStr*, utilizado para enviar el nombre de una variable declarada con su numero de linea y *TokenVal* utilizado para literales enviando su valor y número de línea.
  - Para el resto de los tokens se enviará solo el número de línea, ya que podrán identificarse directamente por su nombre.
- 

### **Detalles de implementación interesantes:**

- Implementación de una función error que permitirá especificar correctamente el error en casos de:

- Un carácter extraño.
- Un entero mayor a 32 bits.
- Un comentario largo no cerrado.

El resto de los errores serán notificados mediante *yyerror()*.

- Implementación correcta de comentarios largos mediante condiciones de entrada de Flex.