



Taller 1
Trabajo de Laboratorio
Curso nocturno 2017/18
Licenciatura en Informática



Tutor:	Gómez, Federico	
Integrantes:	Pías, Richard	- 1.924.591-2
	Segovia, Joaquín	- 4.739.544-4
	Torres, Mathias	- 4.223.291-4



PROBLEMA



SOLUCIÓN



CalcuSimple



UDE

Fallar 1 - Curso 2017 - 2018

Trabajo de Laboratorio

Nombre del problema

análisis

Análisis del aspecto estructural del código fuente en el cual se trabaja

PROGRAMA código

Inicio del programa

VARIABLES x num resu

Instancia de declaración de variables

INSTRUCCIONES

Inicio de las instrucciones, o ejecución del lenguaje

LEER x

num = x

MOSTRAR

resu = RES 10 num

MOSTRAR resu

diseño



estructuras

```
typedef enum {FALSE, TRUE} boolean;
```


estructuras

```
typedef enum {FALSE, TRUE} boolean;
```

```
const int MAX = 80;
```

```
typedef char * StringDyn;
```

estructuras

```
typedef enum {FALSE, TRUE} boolean;
```

```
const int MAX = 80;
```

```
typedef char * StringDyn;
```

```
typedef struct nodoListaStringAlias {  
    StringDyn info;  
    nodoListaStringAlias * sig;  
} nodoListaString;
```

estructuras

```
typedef enum {FALSE, TRUE} boolean;
```

```
const int MAX = 80;
```

```
typedef char * StringDyn;
```

```
typedef struct nodoListaStringAlias {  
    StringDyn info;  
    nodoListaStringAlias * sig;  
} nodoListaString;
```

```
typedef enum {LEER, MOSTRAR, enum_AS1, enum_AS2,  
enum_AS3, enum_AS4, enum_AS5, enum_AS6} enumOpsBasicas;
```

estructuras

```
typedef enum {FALSE, TRUE} boolean;
```

```
const int MAX = 80;
```

```
typedef char * StringDyn;
```

```
typedef struct nodoListaStringAlias {  
    StringDyn info;  
    nodoListaStringAlias * sig;  
} nodoListaString;
```

```
typedef enum {LEER, MOSTRAR, enum_AS1, enum_AS2,  
enum_AS3, enum_AS4, enum_AS5, enum_AS6} enumOpsBasicas;
```

```
typedef enum {SUM, RES, MUL, DIV} enumOpsArits;
```

estructuras

```
typedef struct{  
    StringDyn nombre;  
    int valor;  
} variable;
```

estructuras

```
typedef struct{  
    StringDyn nombre;  
    int valor;  
} variable;
```

```
typedef struct nodoVariableAlias{  
    variable info;  
    nodoVariableAlias *Hizq;  
    nodoVariableAlias *Hder;  
} nodoVariable;  
typedef nodoVariable *arbolVariables;
```

printf(“ tipos de instrucciones ”);

	1er	2do	3ero	4to	5to
1	LEER	var			
2	MOSTRAR	var			
3 (AS1)	var	=	int		
4 (AS2)	var	=	var		
5 (AS3)	var	=	FUNC	int	int
6 (AS4)	var	=	FUNC	int	var
7 (AS5)	var	=	FUNC	var	int
8 (AS6)	var	=	FUNC	var	var

***FUNC = {SUM, RES, MUL, DIV}**

estructuras

```
typedef struct{  
    enumOpsArits func;  
    int valor1;  
    int valor2;  
} AS3;
```


estructuras

```
typedef struct{  
    enumOpsArits func;  
    int valor1;  
    int valor2;  
} AS3;
```

```
typedef struct{  
    enumOpsArits func;  
    int valor1;  
    StringDyn nomVariable;  
} AS4;
```

estructuras

```
typedef struct{  
    enumOpsArits func;  
    int valor1;  
    int valor2;  
} AS3;
```

```
typedef struct{  
    enumOpsArits func;  
    int valor1;  
    StringDyn nomVariable;  
} AS4;
```

```
typedef struct{  
    enumOpsArits func;  
    StringDyn nomVariable;  
    int valor1;  
} AS5;
```

estructuras

```
typedef struct{  
    enumOpsArits func;  
    int valor1;  
    int valor2;  
} AS3;
```

```
typedef struct{  
    enumOpsArits func;  
    int valor1;  
    StringDyn nomVariable;  
} AS4;
```

```
typedef struct{  
    enumOpsArits func;  
    StringDyn nomVariable;  
    int valor1;  
} AS5;
```

```
typedef struct{  
    enumOpsArits func;  
    StringDyn nomVariable1;  
    StringDyn nomVariable2;  
} AS6;
```

estructuras

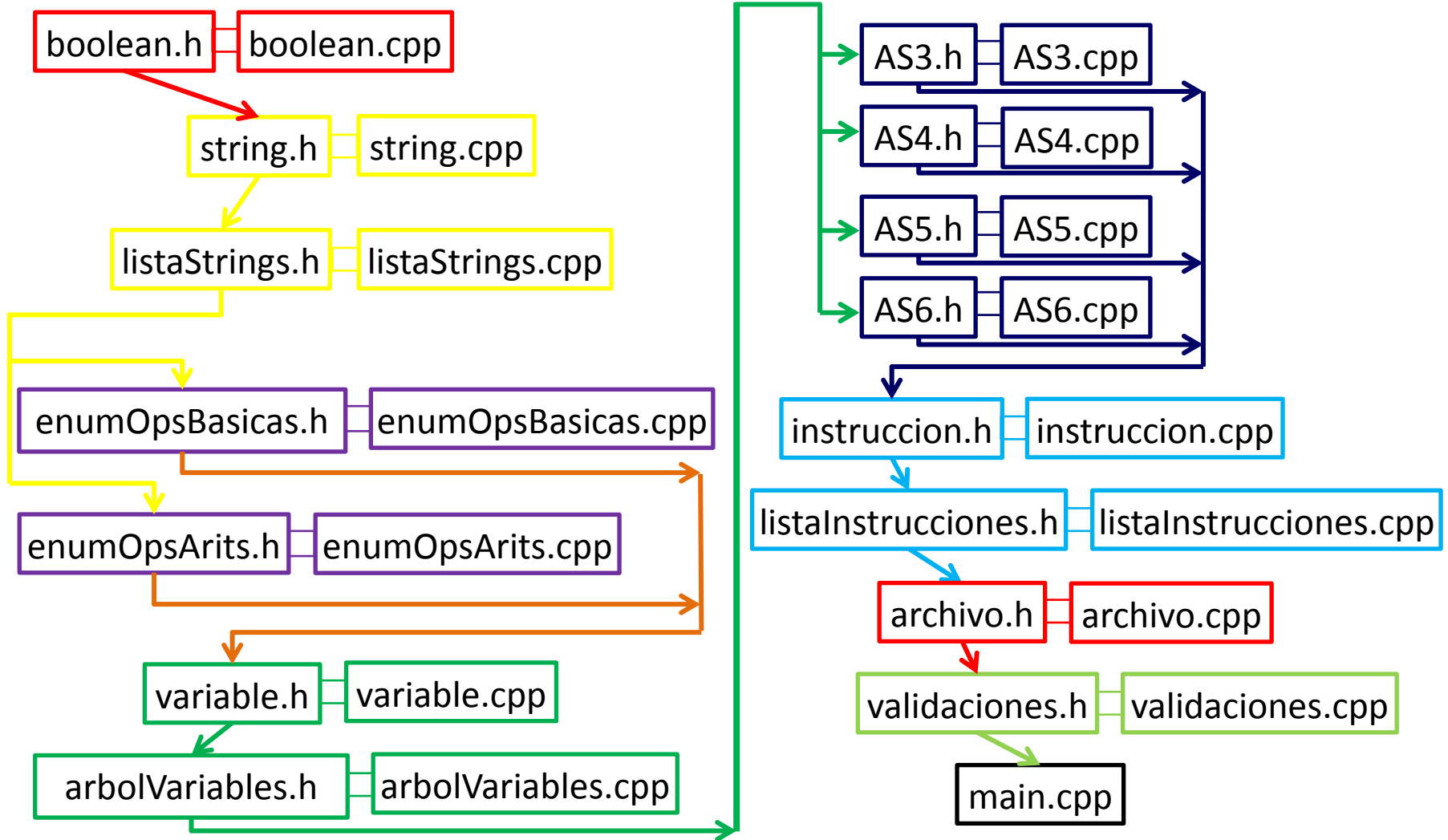
```
typedef struct {  
    StringDyn nombreVar;  
    enumOpsBasicas tipoInstruccion;  
    union {  
        int numeroEntero;  
        StringDyn nomVar1;  
        AS3 AS3_funcIntInt;  
        AS4 AS4_funcIntVar;  
        AS5 AS5_FuncVarInt;  
        AS6 AS6_FuncVarVar;  
    } discInstruccion;  
} instruccion;
```

estructuras

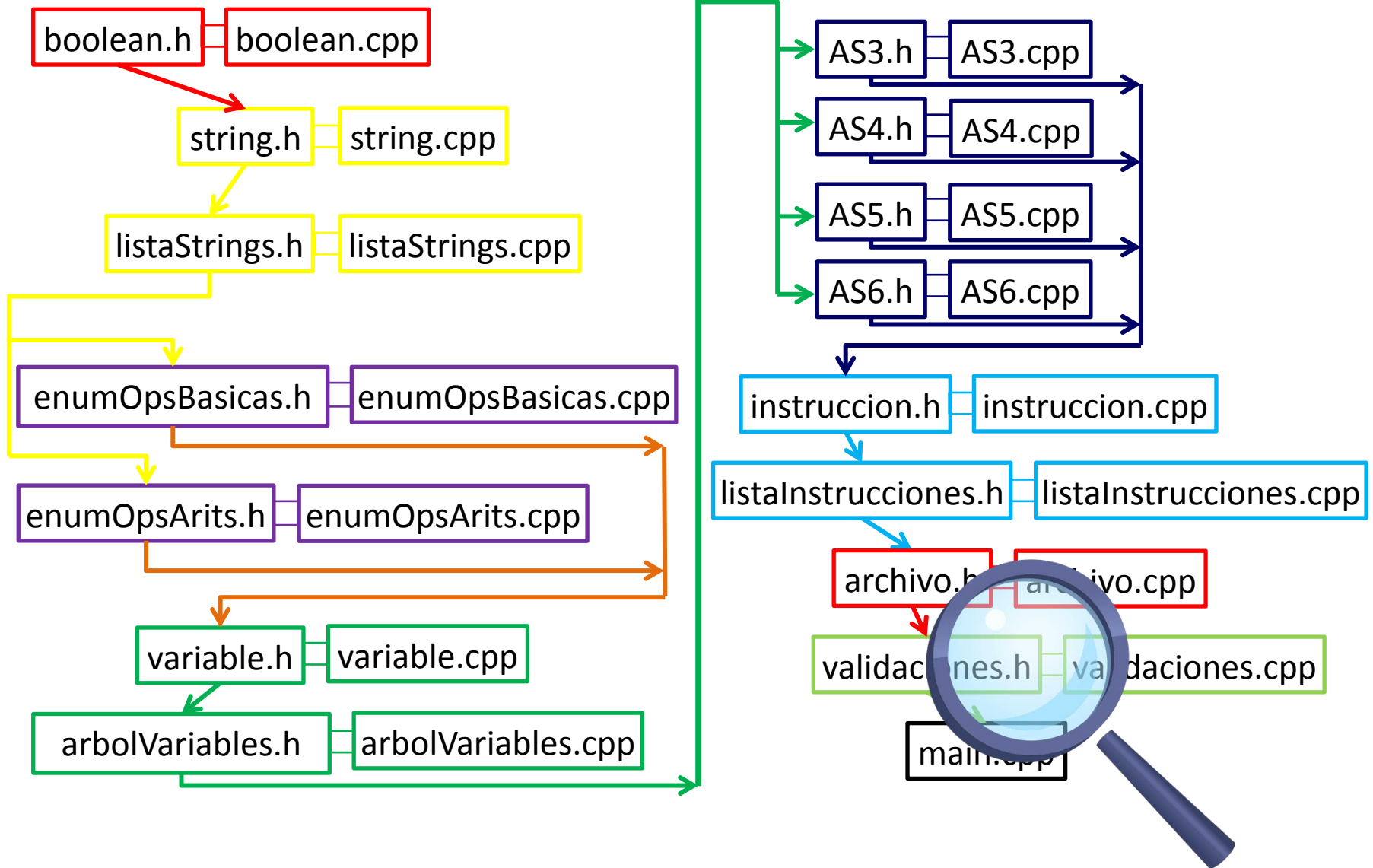
```
typedef struct {  
    StringDyn nombreVar;  
    enumOpsBasicas tipoInstruccion;  
    union {  
        int numeroEntero;  
        StringDyn nomVar1;  
        AS3 AS3_funcIntInt;  
        AS4 AS4_funcIntVar;  
        AS5 AS5_FuncVarInt;  
        AS6 AS6_FuncVarVar;  
    } discInstruccion;  
} instruccion;
```

```
typedef struct nodoInstruccionAlias{  
    instruccion info;  
    nodoInstruccionAlias * sig;  
} nodoInstruccion;  
typedef nodoInstruccion * listaInstrucciones;
```

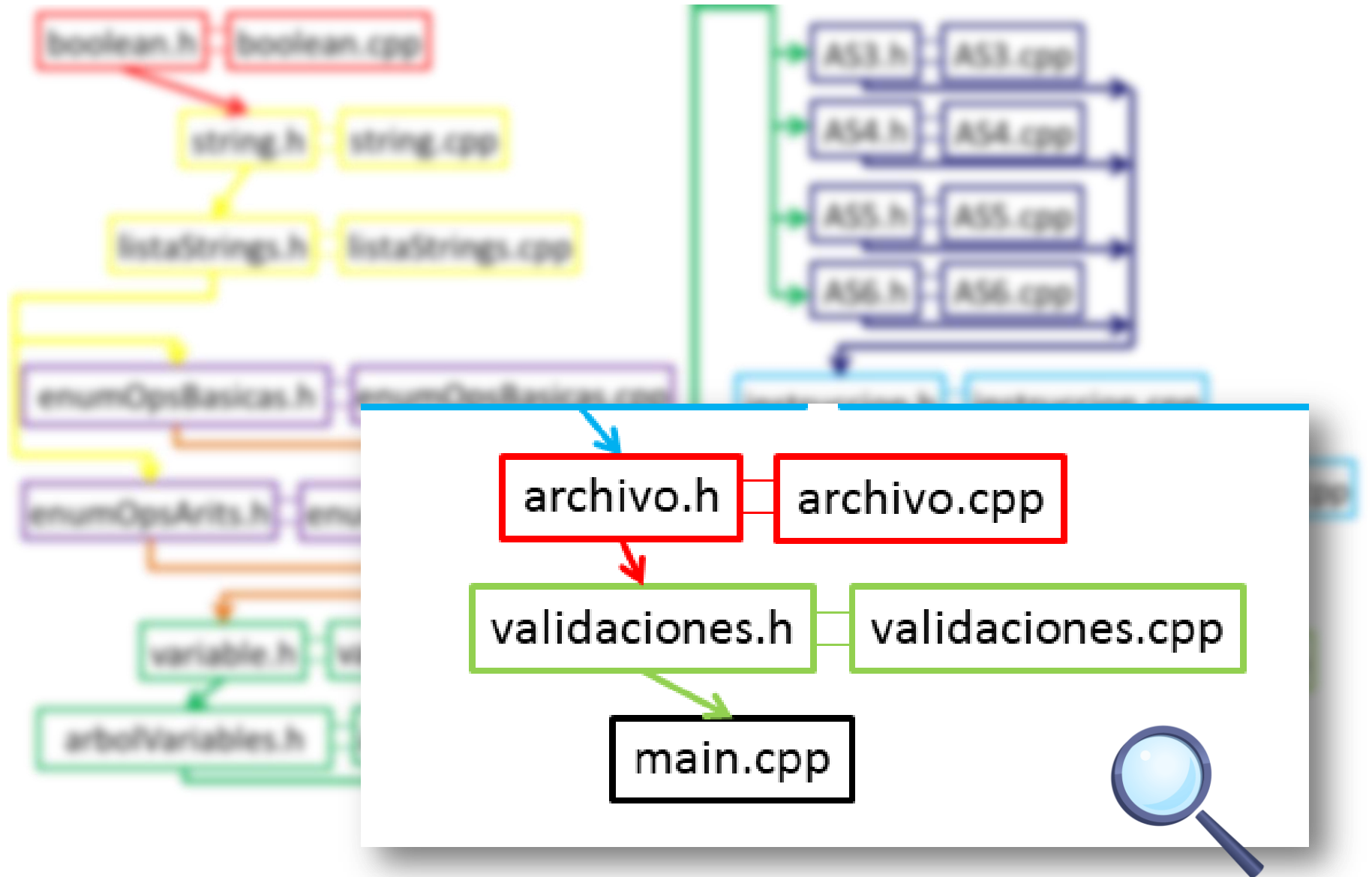
printf(" diagrama de jerarquías ");



printf(" diagrama de jerarquías ");



printf(" diagrama de jerarquías ");



printf(" pseudocódigo ");

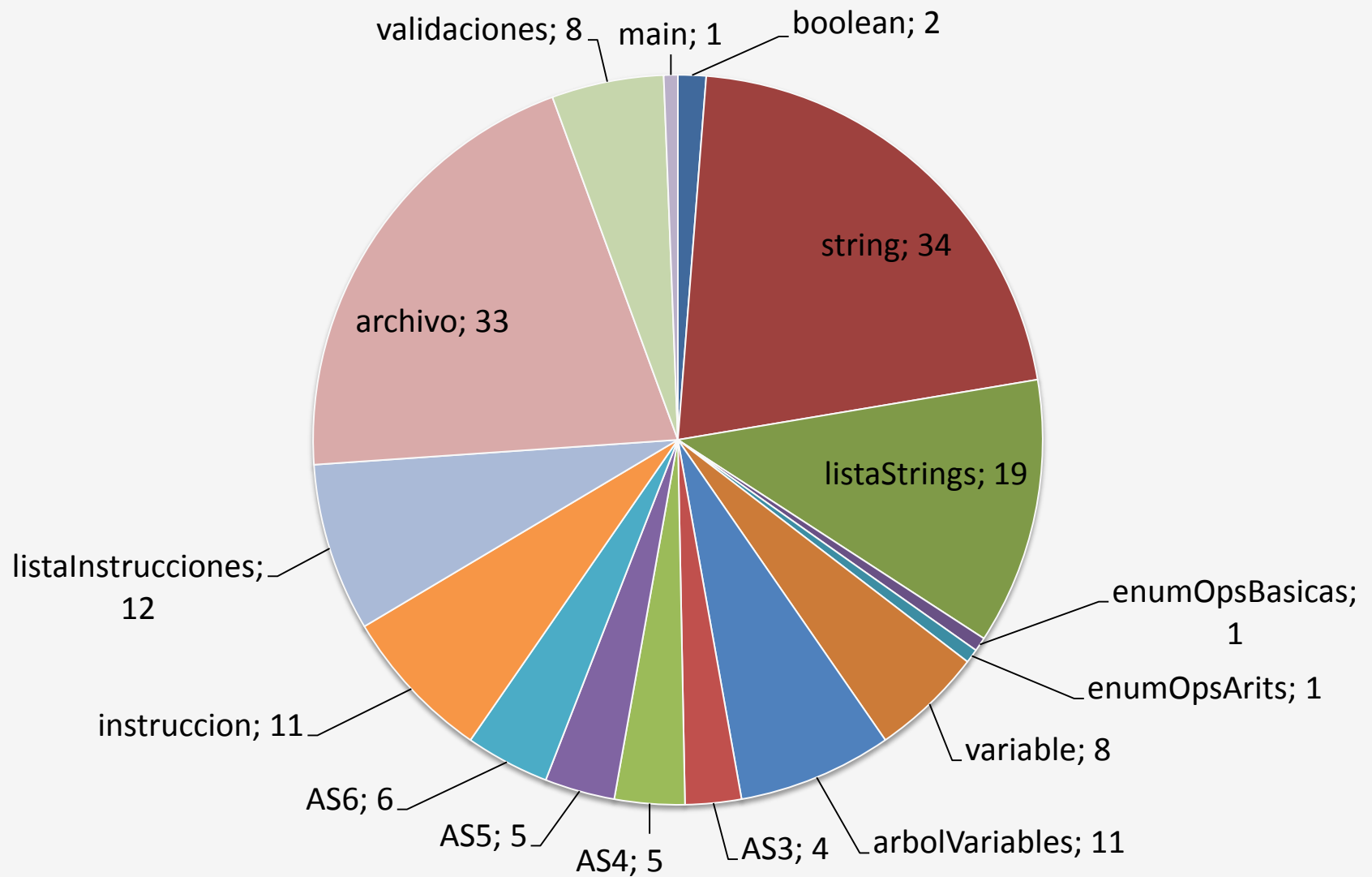
Taller1-1eraEntrega_Pias_Segovia_Torres - PS.txt

```
1  ///UDE, Lic. En Informática, Curso 2017/18 - Nocturno - Taller 1 - Tutor: Gómez, Federico
2  ///Integrantes Grupo F62: Pías, Richard - 1.924.591-2 ; Segovia, Joaquín - 4.739.544-4 ; Torres, Mathias - 4.223.291-4
3
4
5  Levantar la línea escrita en un String
6
7  Partir la línea en subString.
8
9  Si (el primer subString contiene MAYÚSCULAS)
10
11     Mensaje de error ("El comando ingresado solo puede contener letras minúsculas");
12     volver al Prompt del programa;
13
14  Sino Si(el primer subString contiene NÚMEROS)
15
16     Mensaje de error ("El comando ingresado no puede contener números");
17     volver al Prompt del programa;
18
19  Sino Si(la cantidad de subString = 1)
20
21     Si (el primer substring NO es "salir")
22
23         Mensaje de error ("El comando ingresado no es valido, el único comando con un solo argumento debe ser 'salir'");
24         volver al Prompt del programa;
25
26     Sino
27
28         Salgo del Programa
29
30     FinSi
31
32  Sino Si(la cantidad de subString = 2)
33
34     Si (el primer subString contiene MAYÚSCULAS)
35
36         Mensaje de error ("El comando ingresado solo puede contener letras minúsculas");
37         volver al Prompt del programa;
38
39     Sino Si(el primer subString contiene NÚMEROS)
40
41         Mensaje de error ("El comando ingresado no puede contener números");
42         volver al Prompt del programa;
```

Spaces: 4

Plain Text

● ● ● #Cabezales sintácticos = 160



Cabezales +relevantes: String.h

```
boolean StringDyn_contiene_MAYUSCULAS(StringDyn s1);
boolean StringDyn_contiene_minusculas(StringDyn s1);
boolean StringDyn_contiene_numeros(StringDyn s1);
int StringDyn_cantidad_de_SubString(StringDyn s1);
boolean StringDyn_equalSignoIgual(StringDyn s1);
boolean StringDyn_equalAnyFUNCarits(StringDyn s1);
boolean StringDyn_equalNumeroEntero(StringDyn s1);
void StringDynPROC_separarPrimerPalabra(StringDyn linea, StringDyn &palabra,
StringDyn &restoLinea);
void StringDyn_copiarDadoLargo(StringDyn &s1,StringDyn s2, int largo, int inicio);
int StringDynFUNC_stringToNumeric(StringDyn s1);
```

Cabezas + relevantes: listaStrings.h

```
void listaStringsPROC_instertBack(listaStrings &ls, StringDyn info);
```

```
StringDyn listaStringsFUNC_darSIGUIENTEstr(listaStrings ls, int numero);
```

```
void PROC_partirStrEnSubStrs(StringDyn linea, listaStrings &lsSubsString);
```

variable.h

```
boolean variableFUNC_MenorQue(variable v1, variable v2);
```

```
void variablePROC_modificarValor(variable &v, int newValor);
```

Cabezas + relevantes: arbolVariables.h

```
void arbolVariablesPROC_agregarVariableAlArbol(arbolVariables  
&av, variable v);
```

```
void arbolVariablesPROC_modificarVariableEnABB(arbolVariables  
&abb, StringDyn nombreVar, int nuevoValor);
```

```
boolean arbolVariablesFUNC_verificarExistenciaString(arbolVariables  
av, StringDyn str);
```

```
void arbolVariablesPROC_PrintNombreValor(arbolVariables abb,  
StringDyn nombreVar);
```

```
int arbolVariablesFUNC_valorDeVariable(arbolVariables abb,  
StringDyn nombreVar);
```

cabezales +relevantes: instruccion.h

```
void instruccionFUNC_cargar(instruccion &i, StringDyn nombreVar, enumOpsBasicas  
tipoInstruccion, int numeroEntero, StringDyn nomVar1, AS3 AS3_funcIntInt, AS4  
AS4_funcIntVar, AS5 AS5_FuncVarInt, AS6 AS6_FuncVarVar);
```

listaInstrucciones.h

```
void listaInstruccionesPROC_instertBack(listaInstrucciones &li, instruccion info);  
instruccion listaInstruccionesFUNC_darInstruccion(listaInstrucciones li, int numero);  
void listaInstruccionesPROC_agregarInstruccionLEER(StringDyn str, listaInstrucciones &li);  
void listaInstruccionesPROC_agregarInstruccionMOSTRAR(StringDyn str, listaInstrucciones &li);  
void listaInstruccionesPROC_agregarInstruccionAScompuesta(arbolVariables abb, listaInstrucciones  
&li, listaStrings ls, int nroLinea, boolean &huboError, int &nroLineaWarning, boolean  
&compiloConCero);  
void listaInstruccionesPROC_agregarInstruccionASsimppl(arbolVariables abb, listaInstrucciones  
&li, listaStrings ls, int nroLinea, boolean &huboError);
```

Cabezales +relevantes: archivos.h

```
boolean archivoFUNC_existeArchivo(StringDyn strNombreArchivo_con_extension);  
void archivoPROC_crearArchivo(FILE * &archivo, StringDyn nombreArchivo);  
void archivoPROC_abrirArchivo(FILE * &archivo, StringDyn nombreArchivo, StringDyn  
como);  
StringDyn archivoFUNC_cargarLaSiguienteLinea(FILE * archivo);  
void archivoPROC_bajarABB_de_Variables_al_Archivo(arbolVariables abb, FILE *  
archivo);  
void archivoPROC_levantoABB_de_Variables_a_Memoria(arbolVariables &abb, FILE *  
archivo);  
void archivoPROC_listaInstrucciones_BajarLista_al_Archivo(listaInstrucciones li, FILE *  
archivo);  
void archivoPROC_listaInstrucciones_LevantarLista_a_Memoria(listaInstrucciones &li,  
FILE * archivo);
```

Cabezales +relevantes: validaciones.h

```
void validacionesPROC_guardarVariablesEnABB(listaStrings ls, arbolVariables  
&abb, boolean &huboError);
```

```
void validacionesPROC_bucleInstruccionesEjecucion(arbolVariables abb,  
listaInstrucciones li, int &nroLinea, boolean &huboError);
```

```
void PROC_compilar(StringDyn nombrePrograma);
```

```
void PROC_ejecutar(StringDyn nombrePrograma);
```

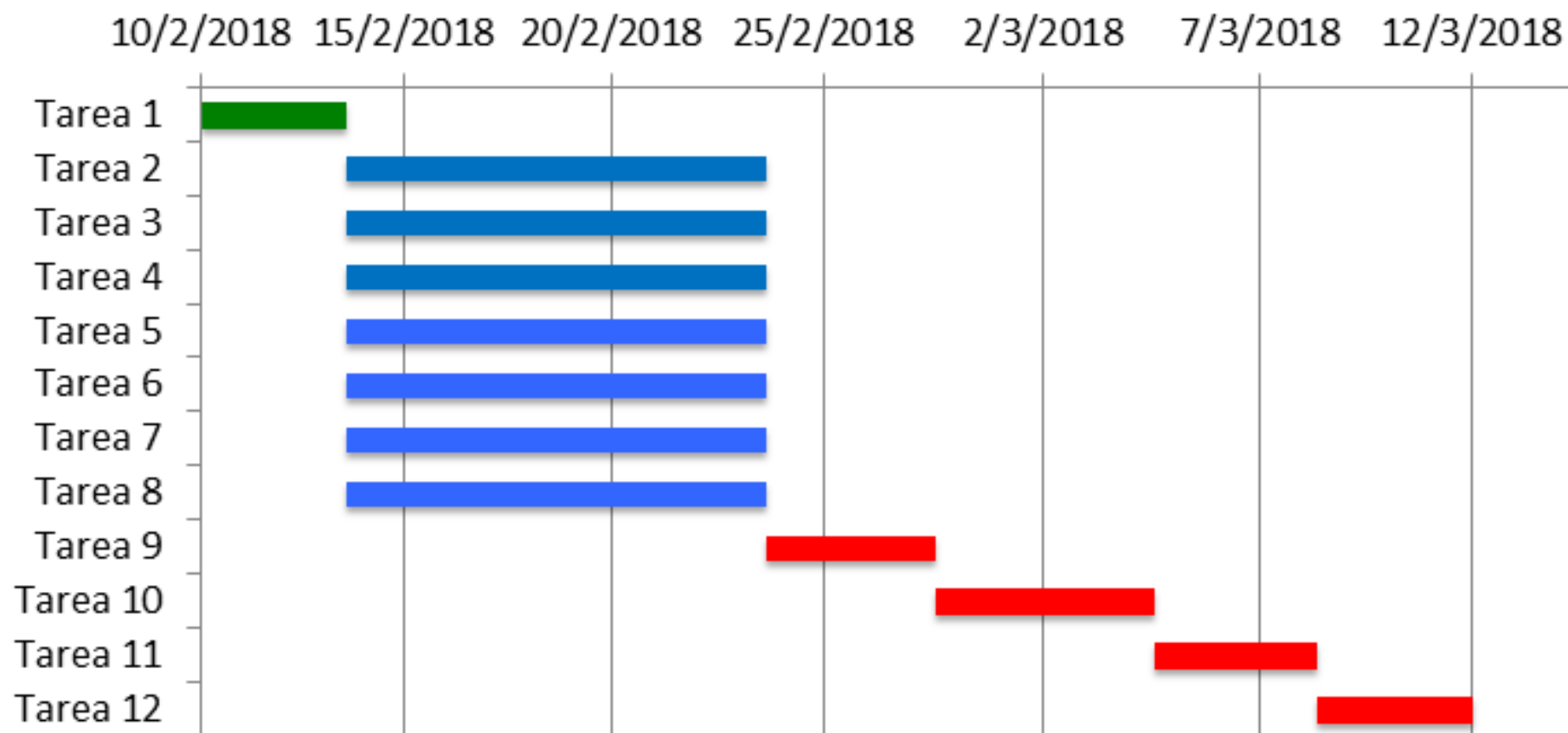
main.cpp

```
int main();
```


planificación

Nombre de la tarea	Fecha de inicio	Fecha final	Duración (días)	Descripción
Tarea 1	10/2/2018	14/2/2018	4	Desarrollo del main.cpp, contemplando los tres comandos y los cabecales
Tarea 2	14/2/2018	24/2/2018	10	Desarrollo del módulo string y listaStrings, incluyendo el testing unitario
Tarea 3	14/2/2018	24/2/2018	10	Desarrollo del módulo enumOpsBasicas y enumOpsArits, incluyendo el testing unitario
Tarea 4	14/2/2018	24/2/2018	10	Desarrollo del módulo variable y arbolVariables, incluyendo el testing unitario
Tarea 5	14/2/2018	24/2/2018	10	Desarrollo del módulo AS3, AS4, AS5 y AS6, incluyendo el testing unitario
Tarea 6	14/2/2018	24/2/2018	10	Desarrollo del módulo instruccion y listaInstrucciones, incluyendo el testing unitario
Tarea 7	14/2/2018	24/2/2018	10	Desarrollo del módulo validaciones, incluyendo el testing unitario
Tarea 8	14/2/2018	24/2/2018	10	Desarrollo del módulo archivo, incluyendo el testing unitario
Tarea 9	24/2/2018	28/2/2018	4	Testing regresivo y testing completo
Tarea 10	28/2/2018	5/3/2018	5	Ajustes y correcciones de issues/bugs reportados en el testing
Tarea 11	5/3/2018	8/3/2018	3	Ajustes de documentación y armado de zip entregable
Tarea 12	8/3/2018	12/3/2018	4	Armado de presentación (ppt), speeches y DEMO para la defensa

Taller 1 - Planilla de diagrama de Gantt





desarrollo



Taller 1: Distribucion de Tareas - Avance



Archivo Editar Ver Insertar Formato Datos Herramientas Complementos Ayuda Todos los cambios se han guardado en Drive

100% - € % .0 .00 123 - Arial - 10 - B I S A - [Color] - [Background Color] - [Text Color] - [List] - [Align] - [Indent] - [Link] - [Filter] - [Sum] -

fx Richard

	A	B	C	D	E	F	G	H
1	Módulos	Nombre	Estado	Punto	Cabecal	NOTAS	falta	modificando
245		Mathias	resuelto	2	void booleanPROC_mostrar(boolean booleanoAcargar);			
246								
247	fecha	Joaquin	resuelto	1	boolean FechaCompletaFUNC_IsValid(FechaCompleta fecha);			
248		Mathias	resuelto	2	void FechaCompletaPROC_Carga(FechaCompleta &fecha);			
249		Mathias	preguntar	3	void FechaCompletaPROC_Desplegado(FechaCompleta fecha);			
250		Joaquin	testear	4	int FechaComparaFechas(FechaCompleta fecha1, FechaCompleta fecha2);			
251		Richard	testear	5	int FechaCompletaFUNC_SelectoraDia(FechaCompleta fecha);			
252		Mathias	resuelto	6	int FechaCompletaFUNC_SelectoraMes(FechaCompleta fecha);			
253		Mathias	resuelto	7	int FechaCompletaFUNC_SelectoraAnio(FechaCompleta fecha);			
254								
255	string	Richard	falta	1	void StringDyn_crear(StringDyn &s);			
256		Mathias	resuelto	2	void StringDyn_crearDadoLargo(StringDyn &s, int largo);			
257		Mathias	preguntar	3	void StringDyn_destruir(StringDyn &s);			
258		Joaquin	falta	4	int StringDyn_largo(StringDyn s);			
259		Richard	modificando	5	void StringDyn_copiar(StringDyn &s1, StringDyn s2);			
260		Mathias	testear	6	void StringDyn_scan(StringDyn &s);			
261		Joaquin	resuelto	7	void StringDyn_concatenar(StringDyn &s1, StringDyn s2);			
262		Richard	testeando/modif	8	void StringDyn_swap(StringDyn &s1, StringDyn &s2);			
263		Richard	resuelto	9	void StringDyn_print(StringDyn s);			
264		Mathias	modificando	10	boolean StringDyn_menor(StringDyn s1, StringDyn s2);			
265		Mathias	preguntar	3	void StringDyn_destruir(StringDyn &s);			
266		Joaquin	falta	4	int StringDyn_largo(StringDyn s);			
267		Richard	modificando	5	void StringDyn_copiar(StringDyn &s1, StringDyn s2);			
268		Mathias	testear	6	void StringDyn_scan(StringDyn &s);			
269		Joaquin	resuelto	7	void StringDyn_concatenar(StringDyn &s1, StringDyn s2);			
270		Richard	testeando/modif	8	void StringDyn_swap(StringDyn &s1, StringDyn &s2);			
271		Richard	resuelto	9	void StringDyn_print(StringDyn s);			
272		Mathias	modificando	10	boolean StringDyn_menor(StringDyn s1, StringDyn s2);			
273		Mathias	resuelto	11	boolean StringDyn_equal(StringDyn s1, StringDyn s2);			
274		Joaquin	testear	12	boolean StringDyn_contiene_MAYUSCULAS(StringDyn s1);			
275		Richard	falta	13	boolean StringDyn_contiene_minusculas(StringDyn s1);			
276		Mathias	modificando	14	boolean StringDyn_contiene_numeros(StringDyn s1);			
277		Mathias	resuelto	15	int StringDyn_cantidad_de_SubString(StringDyn s1);			
278		Joaquin	modificando	16	boolean StringDyn_equalSignalqual(StringDyn s1);			
279		Mathias	resuelto	17	boolean StringDyn_equalAnyFUNCarits(StringDyn s1);			
280		Richard	testeando/modif	18	boolean StringDyn_equalNumeroEntero(StringDyn s1);			
281		Joaquin	resuelto	19	void StringDynPROC_iniStrDynVarsCompilar(StringDyn &PROGRAMA, StringDyn &VARIABLES, StringDyn &INSTRUCCIONES, StringDyn &LEER, StringDyn &MOSTRAR);			
282		Richard	modificando	20	void StringDynPROC_iniStrDynVarsMain(StringDyn &salir, StringDyn &compilar, StringDyn &ejecutar);			
283		Joaquin	testear	21	void StringDynPROC_iniStrDynVarsOpsArits(StringDyn &SUM, StringDyn &RES, StringDyn &MUL, StringDyn &DIV);			
284		Richard	resuelto	22	StringDynPROC_iniStrDynExtensionesArchivos(StringDyn &ext_csim, StringDyn &ext_vars, StringDyn &ext_inst)			



Implementacion

Avance Implementation

Testing

Avance Testing

Recuento de Estado - Total = 154

modificando

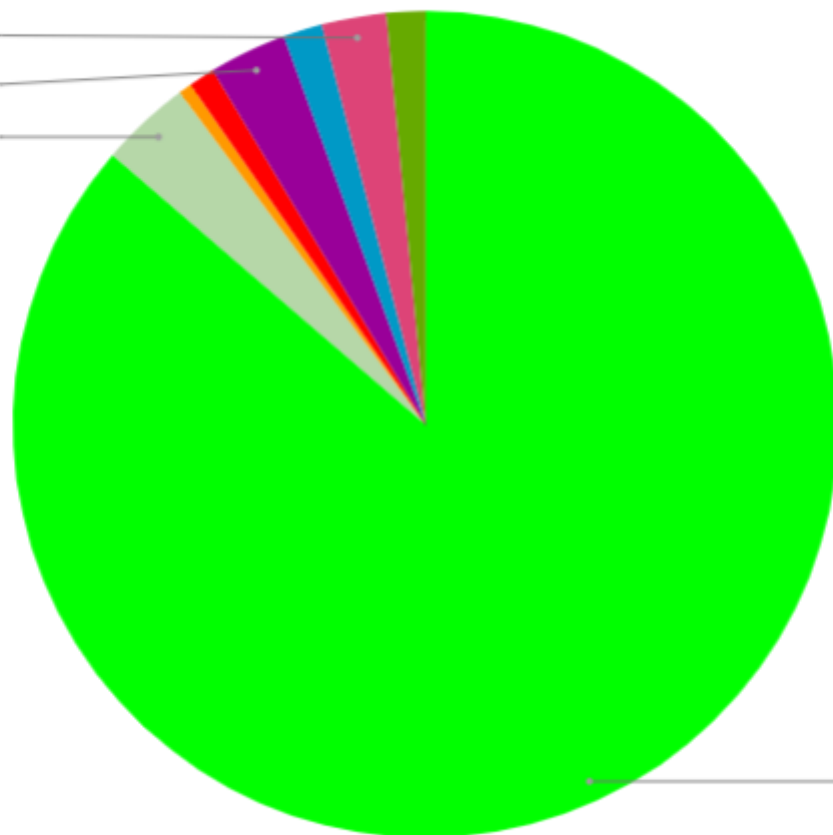
2,5%

testear

3,0%

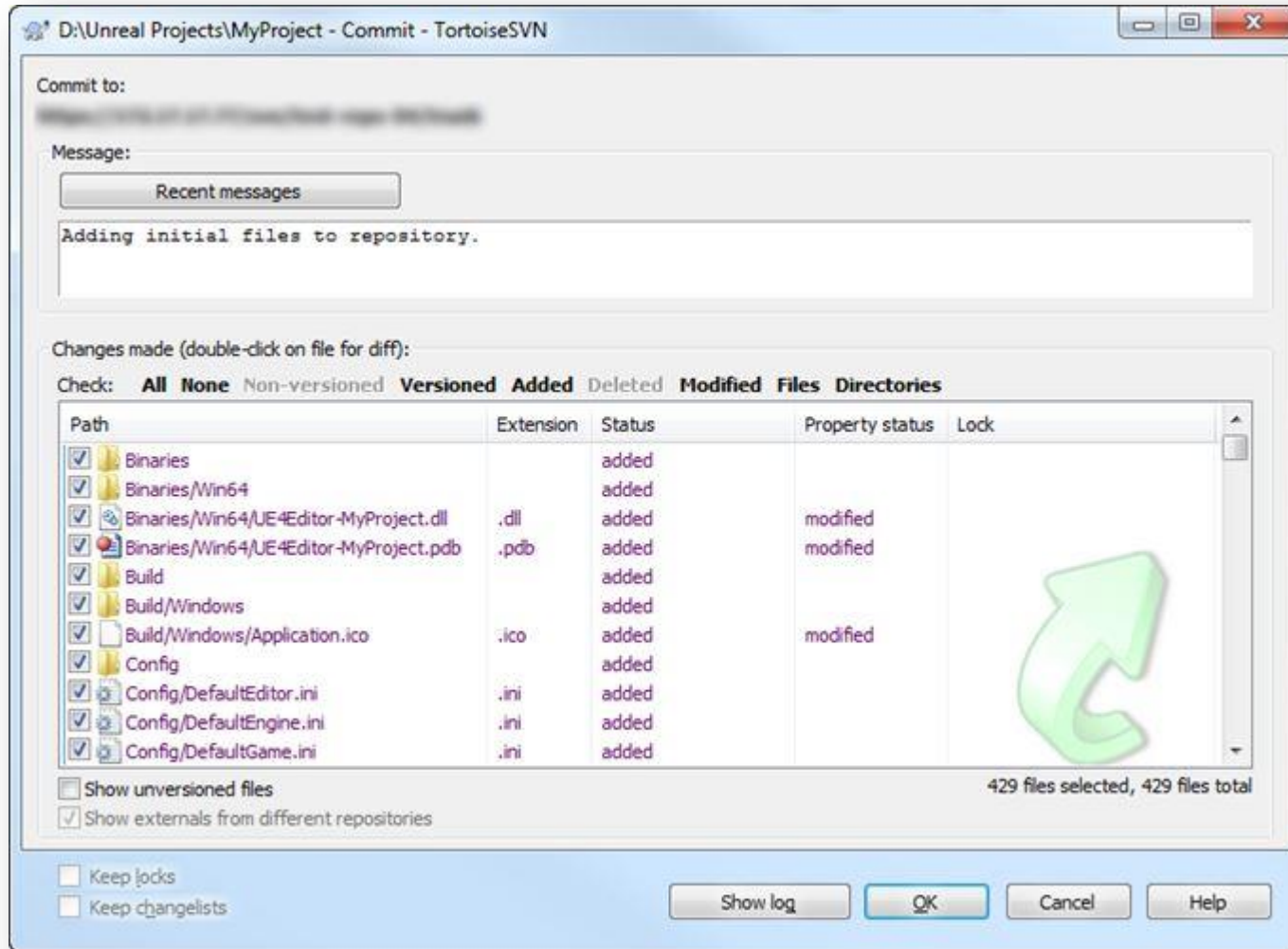
testeando

3,6%

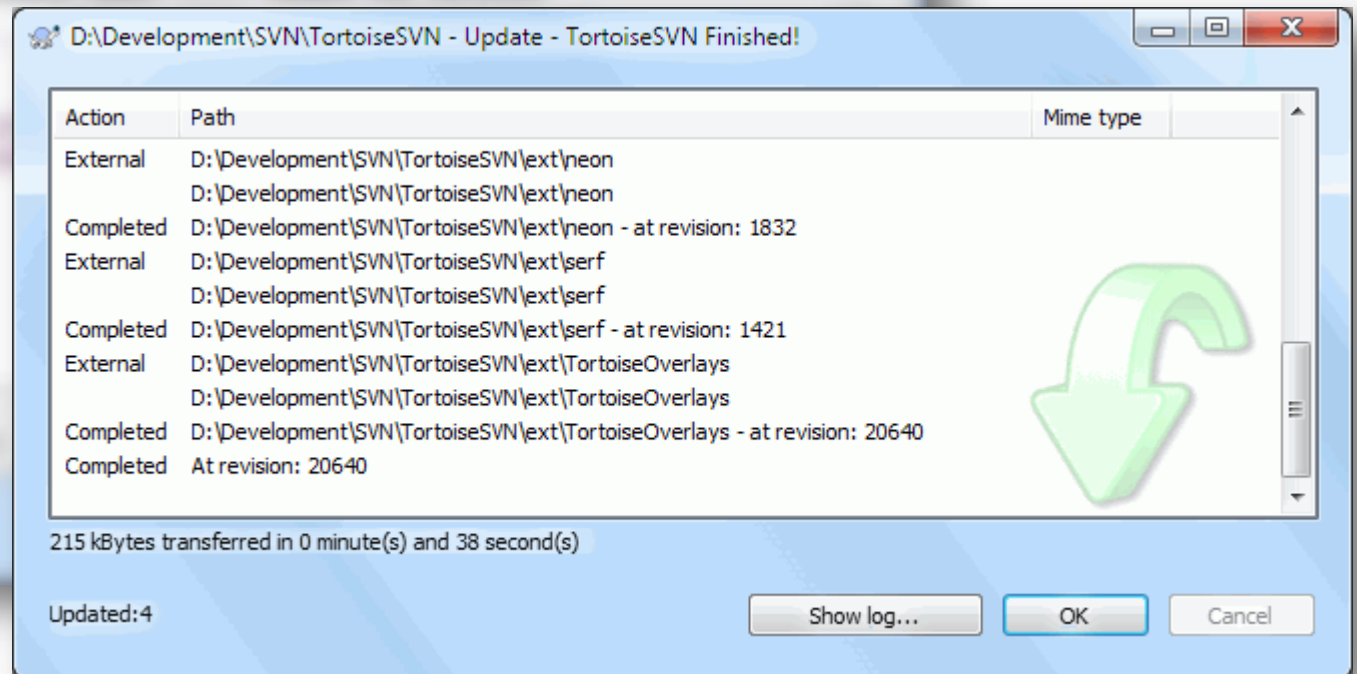


resuelto
86,3%

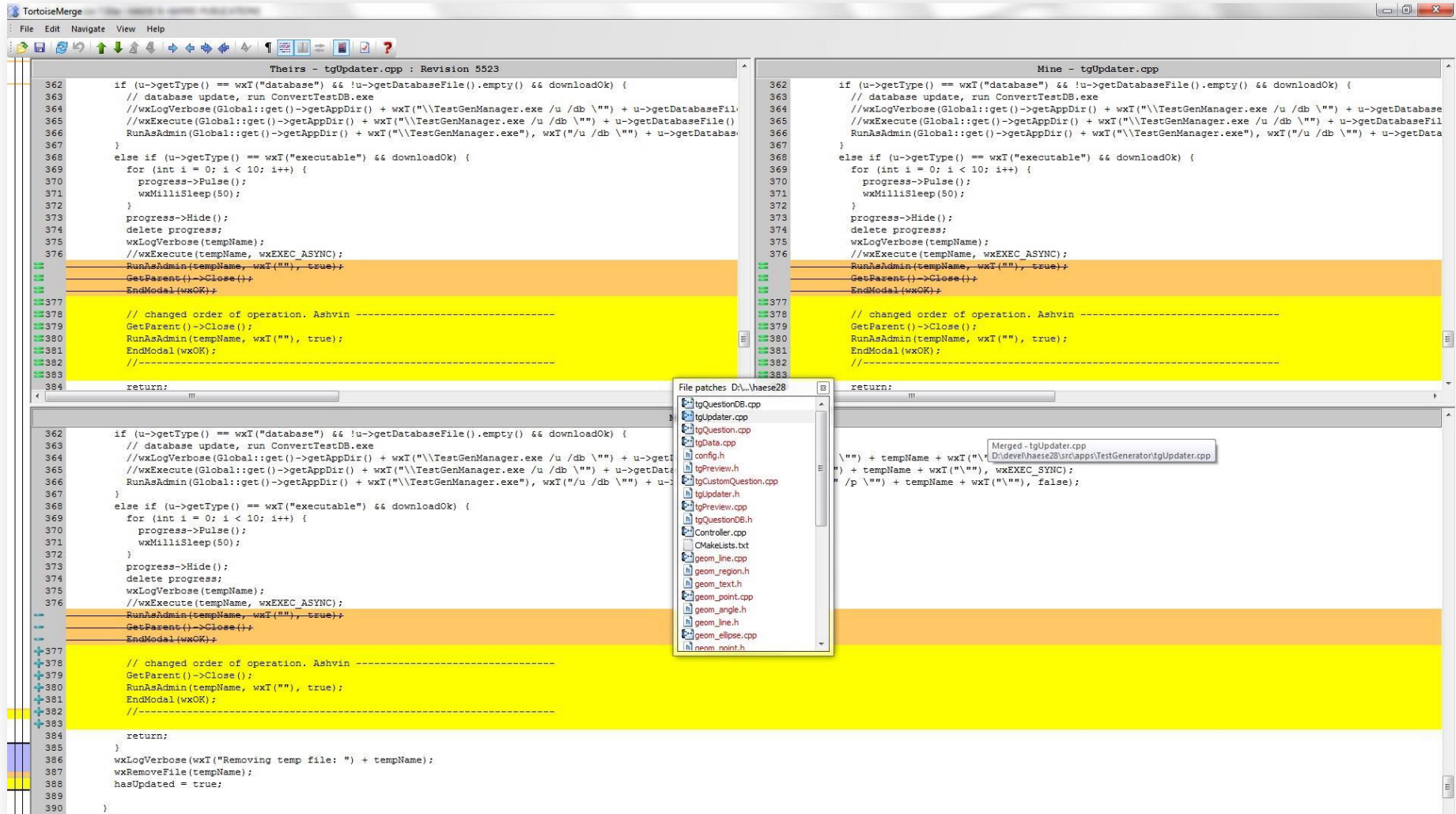
printf(" SVN ");



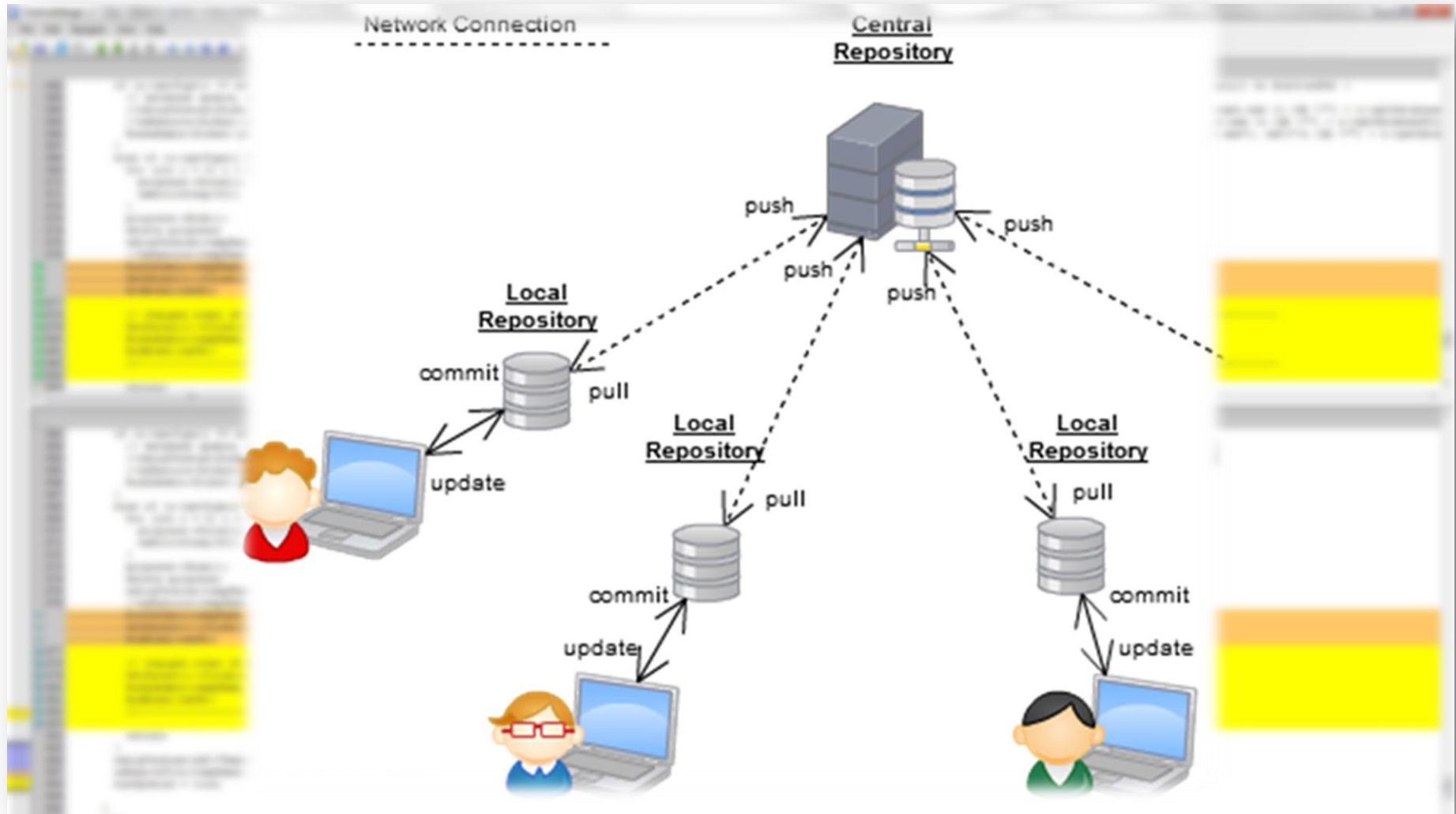
printf(" SVN ");



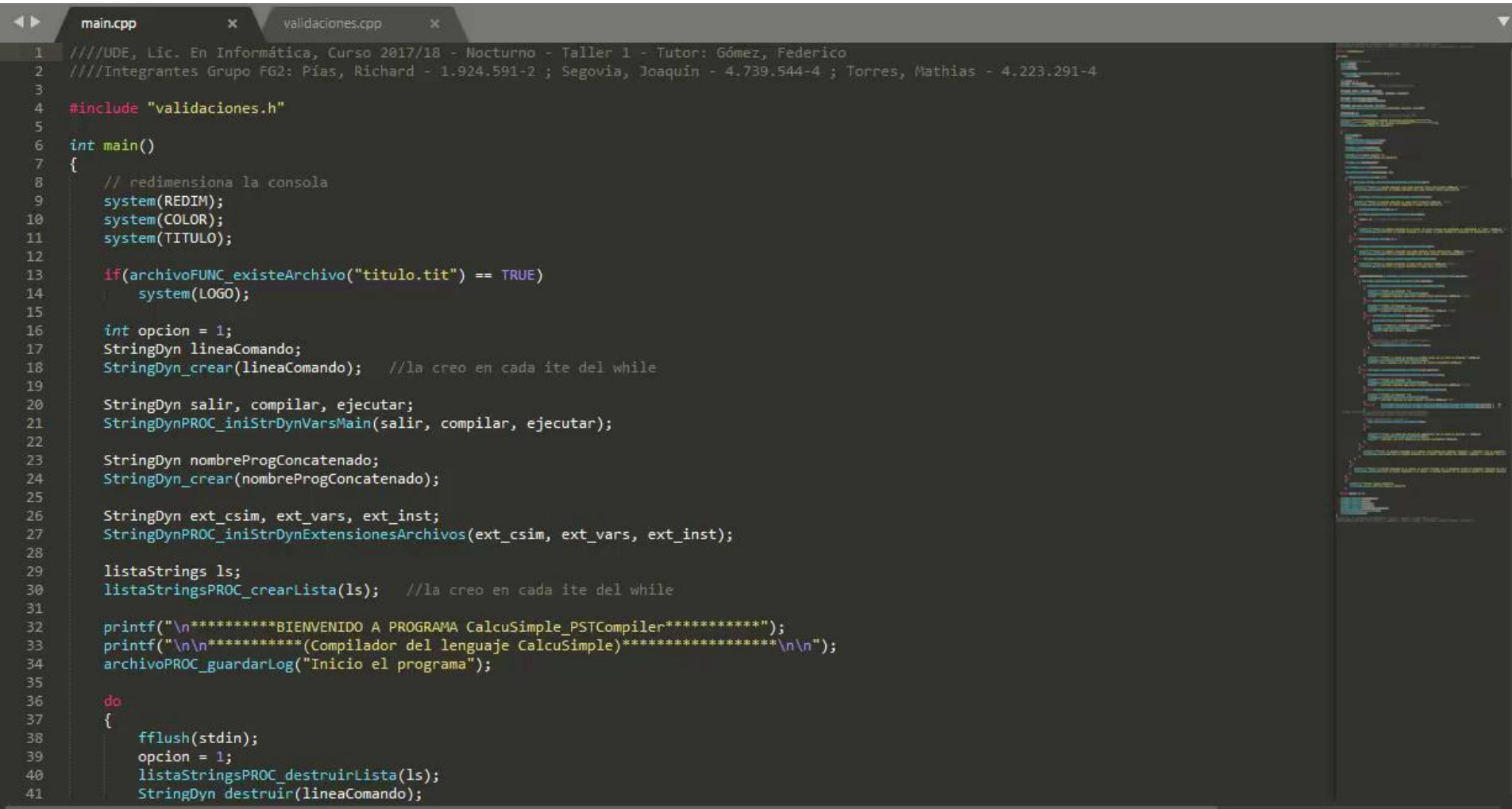
printf("SVN");



printf(" **SVN** ");



printf(" "core" ");



```
1  ///UDE, Lic. En Informática, Curso 2017/18 - Nocturno - Taller 1 - Tutor: Gómez, Federico
2  ///Integrantes Grupo FG2: Pías, Richard - 1.924.591-2 ; Segovia, Joaquín - 4.739.544-4 ; Torres, Mathias - 4.223.291-4.
3
4  #include "validaciones.h"
5
6  int main()
7  {
8      // redimensiona la consola
9      system("REDIM");
10     system("COLOR");
11     system("TITULO");
12
13     if(archivoFUNC_existeArchivo("titulo.tit") == TRUE)
14         system("LOGO");
15
16     int opcion = 1;
17     StringDyn lineaComando;
18     StringDyn_crear(lineaComando); //la creo en cada ite del while
19
20     StringDyn salir, compilar, ejecutar;
21     StringDynPROC_iniStrDynVarsMain(salir, compilar, ejecutar);
22
23     StringDyn nombreProgConcatenado;
24     StringDyn_crear(nombreProgConcatenado);
25
26     StringDyn ext_csim, ext_vars, ext_inst;
27     StringDynPROC_iniStrDynExtensionesArchivos(ext_csim, ext_vars, ext_inst);
28
29     listaStrings ls;
30     listaStringsPROC_crearLista(ls); //la creo en cada ite del while
31
32     printf("\n*****BIENVENIDO A PROGRAMA CalcuSimple PSTCompiler*****");
33     printf("\n\n***** (Compilador del lenguaje CalcuSimple) *****\n\n");
34     archivoPROC_guardarLog("Inicio el programa");
35
36     do
37     {
38         fflush(stdin);
39         opcion = 1;
40         listaStringsPROC_destruirLista(ls);
41         StringDyn destruir(lineaComando);
```

● ● ● Anécdotas - Complicaciones

- Empezamos por el main
- Modificaciones del diagrama entre archivo y validaciones
- Conversión strToInt
- Números negativos
- Caso Delete()
- Archivo: bajar en lugar de subir en ejecución
- Codeblocks 16 para debug con todas las variables



testing



Taller 1: Distribucion de Tareas - Avance



Archivo Editar Ver Insertar Formato Datos Herramientas Com



100% -

€

%

.0

.00

123 -

Arial

-

10

fx

Nombre Archivo

	A	B	
1	Nombre Archivo	Estado	Descripcion
2		resuelto	
3	a	falta	pPROGRAMA
4	aa	falta	PROGRAMa
5	ab	falta	PROGrAMA
6	ac	falta	PROGRRAMA
7	ad	falta	PROGRAMA2
8	ae	falta	2PROGRAMA
9	af	falta	PROG2RAMA
10	ag	falta	
11	ah	falta	falta completar todos los casos...
12	ai	falta	a partir de la fecha 4 del 3
13	aj	falta	
14	ak	falta	
15	al	falta	



Taller 1: Distribucion de Tareas - Avance



Archivo

Editar

Ver

Insertar

Formato

Datos

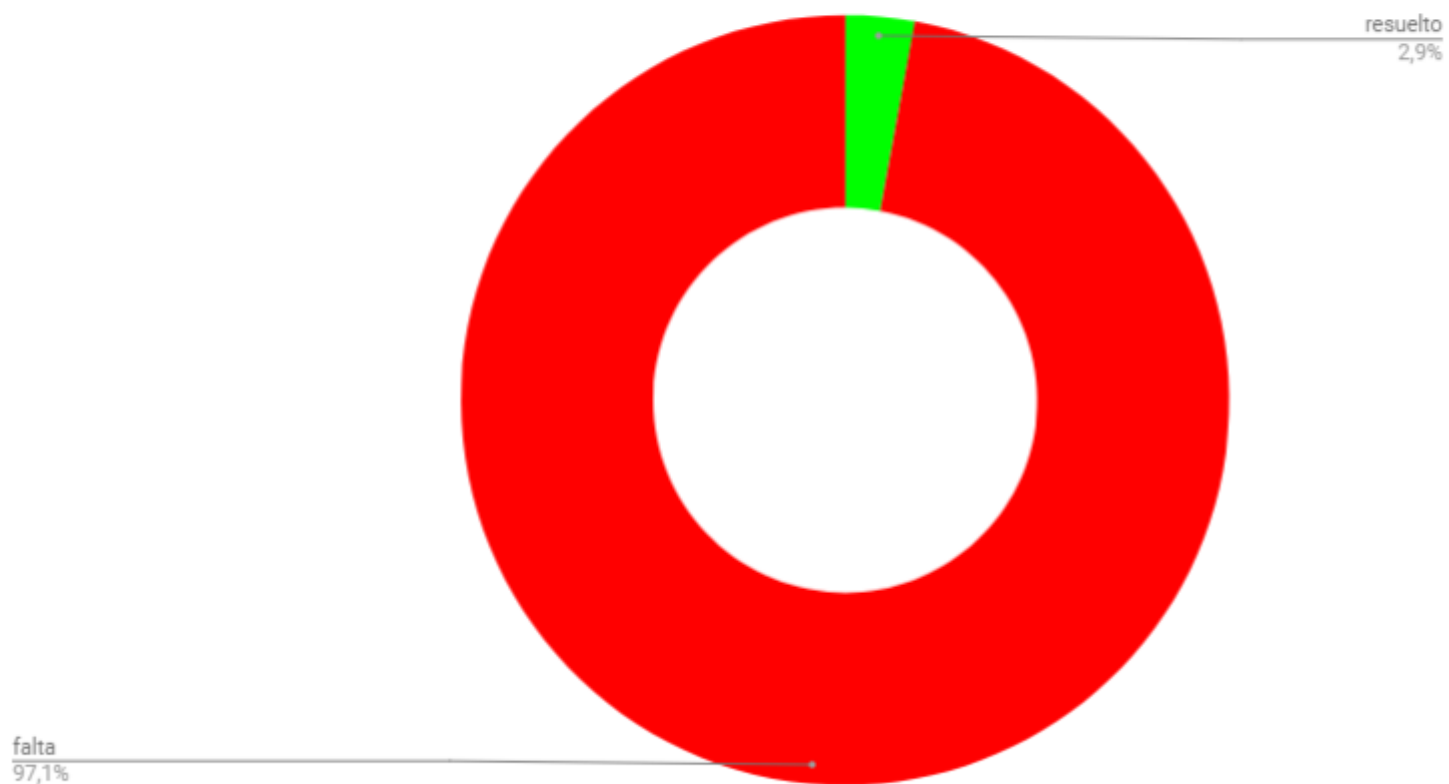
Herramientas

Complementos

Ayuda

Todos los cambios se han guardado

Recuento de Estado



Implementacion ▾

Av Avance Testing ▾



Explorar

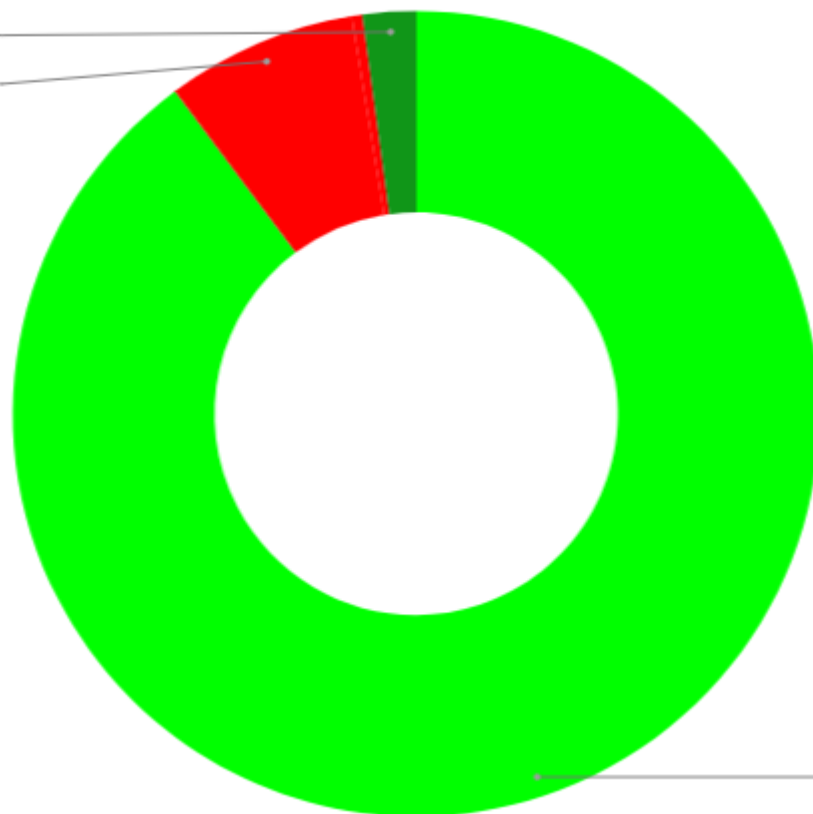
Recuento de Estado

testear fuerte por tercero

2,1%

falta

7,7%



resuelto
89,8%



consolidación

conclusiones

valor agregado

- Tildes como interfaz mas amigable.
- Errores mas precisos con número de línea del código fuente.
- Alerta en compilación para DIV entre 0.
- Alerta en compilación para variables sin uso.
- Log de acciones.
- Líneas que comienzan con `'//'` o `'/*'`, son tomadas como comentarios y no se guardan en `listaInstruccion`



DEMO

```
printf(" preguntas? ");
```

|

```
printf("\n gracias ;");
```

|



Taller 1
Trabajo de Laboratorio
Curso nocturno 2017/18
Licenciatura en Informática



Tutor:	Gómez, Federico.	
Integrantes:	Pías, Richard	- 1.924.591-2
	Segovia, Joaquín	- 4.739.544-4
	Torres, Mathias	- 4.223.291-4