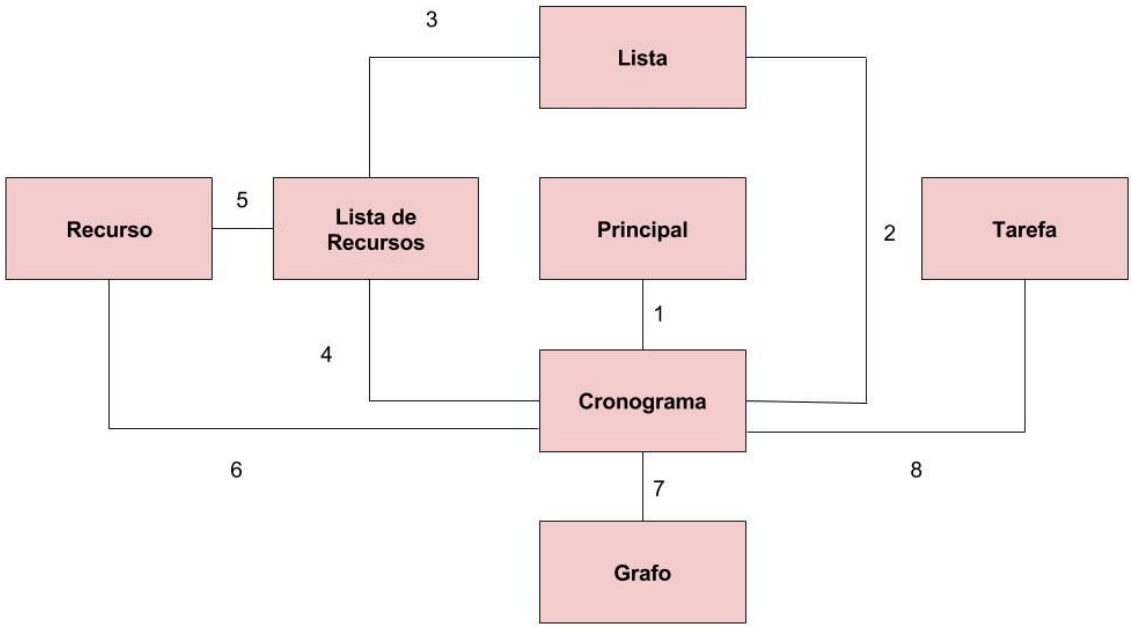
Diagrama de Arquitetura da Aplicação



1 - Cronograma.h

- Servidor: Cronograma
- Cliente: Principal

CRO_CondRet CRO_CriarCronograma (tcCronograma ** ptCronograma);

CRO_CondRet CRO_ListarTarefas(tcCronograma * ptCronograma);

CRO_CondRet CRO_ListarRecursos (tcCronograma * ptCronograma);

CRO_CondRet CRO_ImprimirCaminhoCritico (tcCronograma * ptCronograma);

CRO_CondRet CRO_ImprimirCronograma(tcCronograma * ptCronograma);

2 - Lista.h

- Servidor: Lista
- Cliente: Tarefa

```
LIS_tppLista LIS_CriarLista( void   ( * ExcluirValor ) ( void * pDado ) ) ;

void LIS_DestruirLista( LIS_tppLista pLista ) ;

void LIS_EsvaziarLista( LIS_tppLista pLista ) ;

LIS_tpCondRet LIS_InserirElementoAntes( LIS_tppLista pLista , void * pValor ) ;

LIS_tpCondRet LIS_InserirElementoApos( LIS_tppLista pLista , void * pValor) ;

LIS_tpCondRet LIS_ExcluirElemento( LIS_tppLista pLista ) ;

void * LIS_ObterValor( LIS_tppLista pLista ) ;

void IrInicioLista( LIS_tppLista pLista ) ;

void IrFinalLista( LIS_tppLista pLista ) ;

LIS_tpCondRet LIS_AvancarElementoCorrente( LIS_tppLista pLista , int numElem ) ;

LIS_tpCondRet LIS_ProcurarValor( LIS_tppLista pLista , void * pValor) ;

LIS_tpCondRet LIS_VerificarVazia( LIS_tppLista pLista ) ;
```

3 - Lista.h

- Servidor: Lista
- Cliente: Lista de Recursos

```
LIS_tppLista LIS_CriarLista( void   ( * ExcluirValor ) ( void * pDado ) ) ;

void LIS_DestruirLista( LIS_tppLista pLista ) ;

void LIS_EsvaziarLista( LIS_tppLista pLista ) ;

LIS_tpCondRet LIS_InserirElementoAntes( LIS_tppLista pLista , void * pValor ) ;

LIS_tpCondRet LIS_InserirElementoApos( LIS_tppLista pLista , void * pValor) ;

LIS_tpCondRet LIS_ExcluirElemento( LIS_tppLista pLista ) ;

void * LIS_ObterValor( LIS_tppLista pLista ) ;

void IrInicioLista( LIS_tppLista pLista ) ;

void IrFinalLista( LIS_tppLista pLista ) ;

LIS_tpCondRet LIS_AvancarElementoCorrente( LIS_tppLista pLista , int numElem ) ;

LIS_tpCondRet LIS_ProcurarValor( LIS_tppLista pLista , void * pValor) ;

LIS_tpCondRet LIS_VerificarVazia( LIS_tppLista pLista ) ;
```

4 - ListaRecurso.h

- Servidor: Lista de Recursos
- Cliente: Cronograma

```
LSR_CondRet LSR_CriarLista( tcListaRecurso ** listaRecurso ) ;

void LSR_DestruirLista( tcListaRecurso ** listaRecurso ) ;

void LSR_EsvaziarLista( tcListaRecurso ** listaRecurso ) ;

LSR_CondRet LSR_InserirElemento( tcListaRecurso * listaRecurso , tcRecurso  *
pRecurso)

LSR_CondRet LIS_ExcluirElemento( tcListaRecurso * listaRecurso ) ;

void * LIS_ObterValor( tcListaRecurso * listaRecurso ) ;

LSR_CondRet LSR_PegarIdRecursoDisponivel( tcListaRecurso * listaRecurso ,
                                                                    int * idRecurso ) ;

LSR_CondRet LSR_ProcurarRecursoPorId( tcListaRecurso * listaRecurso , int id) ;

LSR_CondRet LSR_VerificarVazia( tcListaRecurso * listaRecurso ) ;

LSR_CondRet LSR_ListarRecursos( tcListaRecurso * listaRecurso ) ;
```

5 - Recurso.h

- Servidor: Recurso
- Cliente: Lista de Recursos

```
REC_tpCondRet REC_CriarRecurso(tcRecurso ** ptRecurso, char * novoNome) ;

void REC_DestruirRecurso( tcRecurso ** ptRecurso ) ;

REC_tpCondRet REC_AlterarNome(tcRecurso * ptRecurso, char * novoNome);

REC_tpCondRet REC_MarcarComoOcupada(tcRecurso * ptRecurso);

REC_tpCondRet REC_MarcarComoDisponivel(tcRecurso * ptRecurso);

REC_tpCondRet REC_ConsultarId(tcRecurso * ptRecurso, int * id);

REC_tpCondRet REC_ConsultarNome(tcRecurso * ptRecurso, char ** pNome);

REC_tpCondRet REC_ConsultarDisponibilidade(tcRecurso * ptRecurso,
```

int * estaDisponivel);

6 - Recurso.h

- Servidor: Recurso
- Cliente: Cronograma

```
REC_tpCondRet REC_CriarRecurso(tcRecurso ** ptRecurso, char * novoNome) ;

void REC_DestruirRecurso( tcRecurso ** ptRecurso ) ;

REC_tpCondRet REC_AlterarNome(tcRecurso * ptRecurso, char * novoNome);

REC_tpCondRet REC_MarcarComoOcupada(tcRecurso * ptRecurso);

REC_tpCondRet REC_MarcarComoDisponivel(tcRecurso * ptRecurso);

REC_tpCondRet REC_ConsultarId(tcRecurso * ptRecurso, int * id);

REC_tpCondRet REC_ConsultarNome(tcRecurso * ptRecurso, char ** pNome);

REC_tpCondRet REC_ConsultarDisponibilidade(tcRecurso * ptRecurso,
                                           int * estaDisponivel);
```

7 - Grafo.h

- Servidor: Grafo
- Cliente: Cronograma

```
GRA_tpCondRet GRA_CriarGrafo( tcGrafo ** ptGrafo);

GRA_tpCondRet GRA_CriarNo ( tcGrafo * ptGrafo, void * infoNo, int * id);

GRA_tpCondRet GRA_ConectaNos (tcGrafo * ptGrafo, int id1, int id2);

GRA_tpCondRet GRA_DestroiGrafo (tcGrafo ** ptGrafo);

GRA_tpCondRet GRA_PrintaGrafo (tcGrafo ** ptGrafo);
```

8 - Tarefa.h

- Servidor: Tarefa
- Cliente: Lista de Tarefas

```
TRF_tpCondRet TRF_CriarTarefa( void ** ctTarefa, char * novoNome, char *
novaDescricao);

void TRF_DestruirTarefa( void ** ctTarefa ) ;

TRF_tpCondRet TRF_ConectarTarefas( void ** ctTarefaSucessora,
                                                    void ** ctTarefaPredecessora ) ;

TRF_tpCondRet  TRF_AlterarTarefa( void ** ctTarefa, char * novoNome,
                                                    char * novaDescricao  ) ;

TRF_tpCondRet  TRF_ConsultarNomeTarefa( void ** ctTarefa, char ** nomeConsultado ) ;

TRF_tpCondRet  TRF_ConsultarDescricaoTarefa( void ** ctTarefa,
                                                    char ** descricaoConsultada ) ;
```