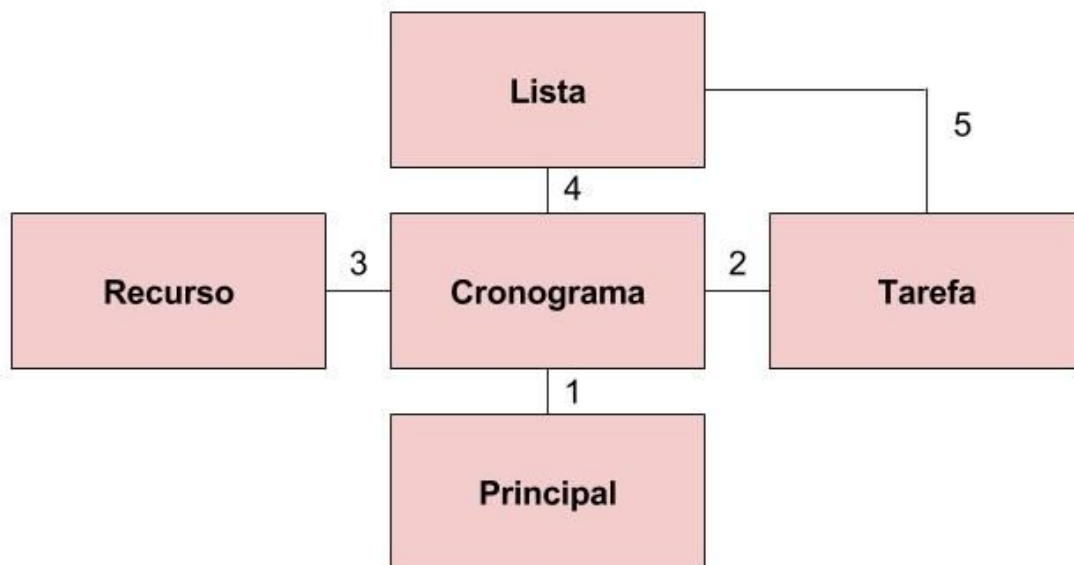


# Diagrama de Arquitetura da Aplicação



1 - Cronograma.h

- Servidor: Cronograma
- Cliente: Principal

```
CRO_tpCondRet CRO_CriarCronograma ( tcCronograma ** ctCronograma );
```

```
CRO_tpCondRet CRO_DestruirCronograma( tcCronograma ** ctCronograma );
```

```
CRO_tpCondRet CRO_InserirTarefa( tcCronograma * ctCronograma,
                                char * novoNome, char * novaDescricao,
                                int duracao );
```

```
CRO_tpCondRet CRO_InserirRecurso( tcCronograma * ctCronograma,
                                char * novoNome);
```

```
CRO_tpCondRet CRO_RemoveTarefa( tcCronograma * cCronograma,
                                int idParaRemover );
```

```
CRO_tpCondRet CRO_RemoveRecurso( tcCronograma * cCronograma,
                                int idParaRemover );
```

```
CRO_tpCondRet CRO_ConectarTarefas( tcCronograma * cCronograma,
                                int idTarefaSucessora, int idTarefaPredecessora );
```

```
CRO_tpCondRet CRO_ImprimeListaTarefa( tcCronograma * cCronograma );

CRO_tpCondRet CRO_ImprimeListaRecurso( tcCronograma * cCronograma );

CRO_tpCondRet CRO_CaminhoCritico( tcCronograma * cCronograma );

CRO_tpCondRet CRO_ImprimeCaminhoCritico( tcCronograma * cCronograma );

CRO_tpCondRet CRO_AlterarTarefa( tcCronograma * cCronograma,
                                int idTarefa, char * novoNome,
                                char * novaDescricao, int novaDuracao );

CRO_tpCondRet CRO_AlterarRecurso( tcCronograma * cCronograma,
                                  int idRecurso, char * novoNome );

CRO_tpCondRet CRO_ImprimeCronograma( tcCronograma * cCronograma );
```

## 2 - Tarefa.h

- Servidor: Tarefa
- Cliente: Cronograma

```
TRF_tpCondRet TRF_CriarTarefa( tcTarefa ** ctTarefa, char * novoNome, char *  
novaDescricao, int duracao );
```

```
void TRF_DestruirTarefa( tcTarefa ** ctTarefa );
```

```
TRF_tpCondRet TRF_ConectarTarefas( tcTarefa ** ctTarefaSucessora,  
                                   tcTarefa ** ctTarefaPredecessora  
                                   );
```

```
TRF_tpCondRet TRF_AlterarTarefa( tcTarefa ** ctTarefa, char * novoNome,  
                                char * novaDescricao  
                                int duracao );
```

```
TRF_tpCondRet TRF_ConsultarIdTarefa( tcTarefa ** ctTarefa, int * idConsultado );
```

```
TRF_tpCondRet TRF_ConsultarNomeTarefa( tcTarefa ** ctTarefa, char ** nomeConsultado  
);
```

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

```
TRF_tpCondRet TRF_ConsultarDuracaoTarefa( tcTarefa ** ctTarefa,  
                                           int * duracaoConsultada );
```

```
TRF_tpCondRet TRF_CadastrarIdRecurso( tcTarefa ** ctTarefa, int idRecurso );
```

```
TRF_tpCondRet TRF_ConsultarIdRecurso( tcTarefa ** ctTarefa, int * idRecurso );
```

```
TRF_tpCondRet TRF_ImprimeTarefa( tcTarefa * ctTarefa );
```

```
TRF_tpCondRet TRF_TemPredecessores( tcTarefa * ctTarefa, int * temPredecessoras );
```

```
TRF_tpCondRet TRF_TemSucessores( tcTarefa * ctTarefa, int * temSucessoras );
```

```
TRF_tpCondRet TRF_CalculaCaminhoCritico( tcTarefa * ctTarefa );
```

```
TRF_tpCondRet TRF_ImprimeCaminhoCritico( tcTarefa * ctTarefa );
```

```
TRF_tpCondRet TRF_ImprimeBasicoTarefa( tcTarefa * ctTarefa );
```

### 3 - 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);

REC_tpCondRet REC_ImprimeRecurso(tcRecurso * ptRecurso);
```

#### 4 - Lista.h

- Servidor: Lista
- Cliente: Cronograma

```
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 );
```

```
LIS_tpCondRet LIS_InserirElementoFinal( LIS_tppLista pLista, void * pValor);
```

```
void * LIS_PopPrimeiro( LIS_tppLista pLista );
```

## 5 - 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 );
```

```
LIS_tpCondRet LIS_InserirElementoFinal( LIS_tppLista pLista, void * pValor);
```

```
void * LIS_PopPrimeiro( LIS_tppLista pLista );
```