

# Ferramenta Administrativa para Ambientes Computacionais

Liandro Rossini  
Antônio Carlos Tavares

# Roteiro

- ✓ Introdução
- ✓ Objetivo
- ✓ Revisão bibliográfica
- ✓ Desenvolvimento da ferramenta
- ✓ Resultados
- ✓ Conclusões

# Introdução

## ✓ Ambiente Computacional

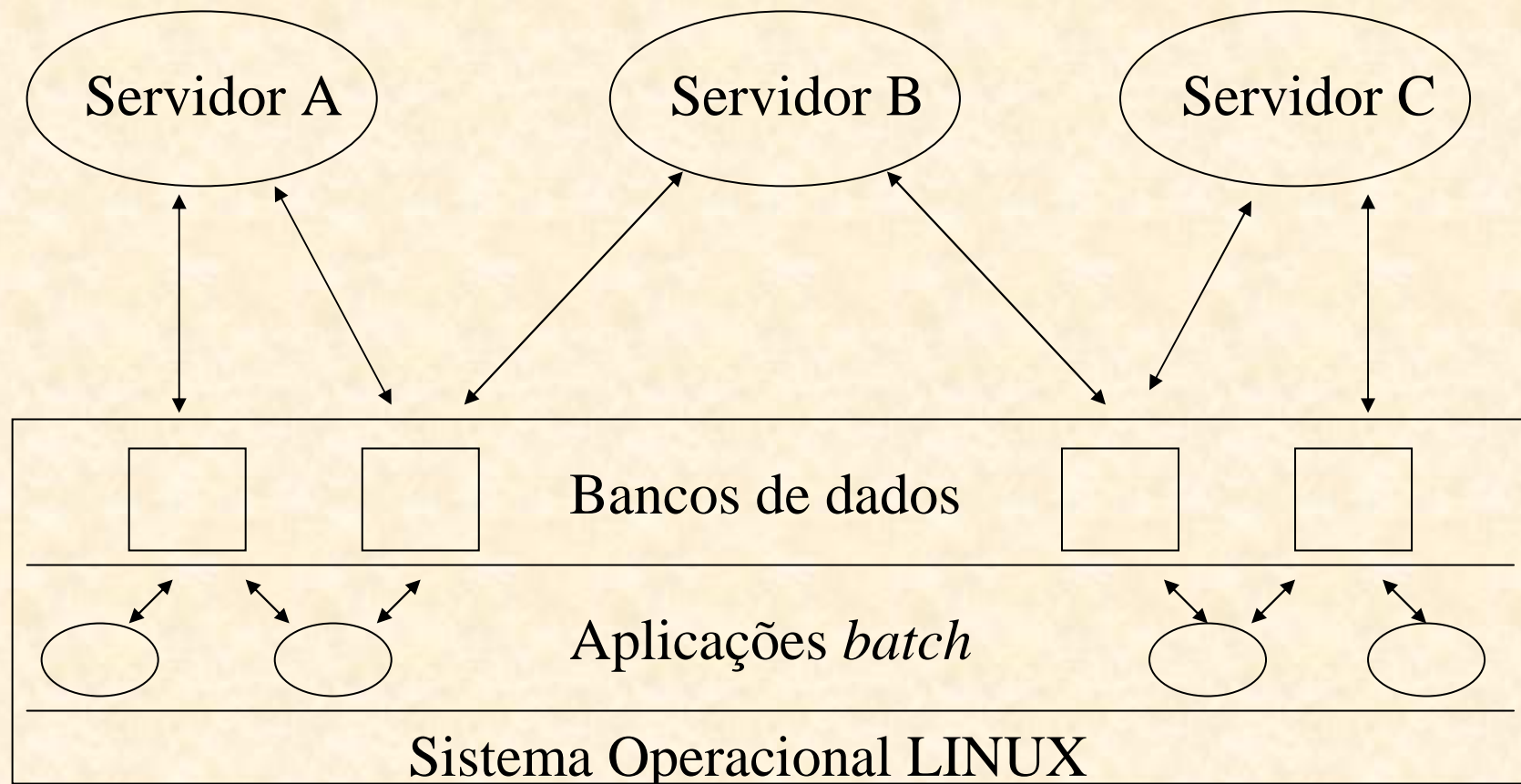
- Ambiente composto por servidores, banco de dados, aplicações...

## ✓ Ferramenta Administrativa

- Deve permitir controle sobre situações conhecidas, através da aplicação de soluções.

# Ambiente Computacional

Servidores de aplicações WEB



# Objetivos do Trabalho

- ✓ gerar um módulo para cadastrar os tipos de problemas mais corriqueiros do ambiente (falta de espaço, *logs* não atualizados no prazo esperado, processos inativos, processos ativos porém inoperantes);
- ✓ disponibilizar um módulo para cadastrar e parametrizar as situações que serão monitoradas no ambiente
- ✓ desenvolver um escalonador que deverá ler as informações cadastradas e atuar sobre cada tipo de problema.

# Sistema Operacional

- ✓ Controla todos os recursos e fornece a base sobre a qual os aplicativos são escritos (compartilhamento de recursos);
- ✓ Execução concorrente dos processos;
- ✓ Disponibiliza uma série de comandos que são utilizados para interagir com o sistema operacional.

# Sistema Operacional

Exemplos:

“ps” – lista os processos em execução;

“df ”– exibe o percentual da utilização das estruturas físicas;

“|” – armazena a saída de um comando que será utilizada como entrada de um próximo comando;

“grep” – busca por uma expressão;

“rm” – remover arquivo;

“kill” – finalizar processo.

# Sistema Operacional

✓ Possui a definição de “processo”, que consiste basicamente em um programa sendo executado.

## **UNIX**

✓ Um dos primeiros sistemas operacionais que implementou suporte a multiusuários para minicomputadores.



# Sistema Operacional

“O UNIX é dominante em estações de trabalho e em outros computadores topo de linha, servidores de rede e máquinas com processadores de alto desempenho” (TANENBAUM, 2000).

- ✓ Por isso, ainda é o mais utilizado para o tipo de arquitetura de ambiente computacional citado.

## **LINUX**

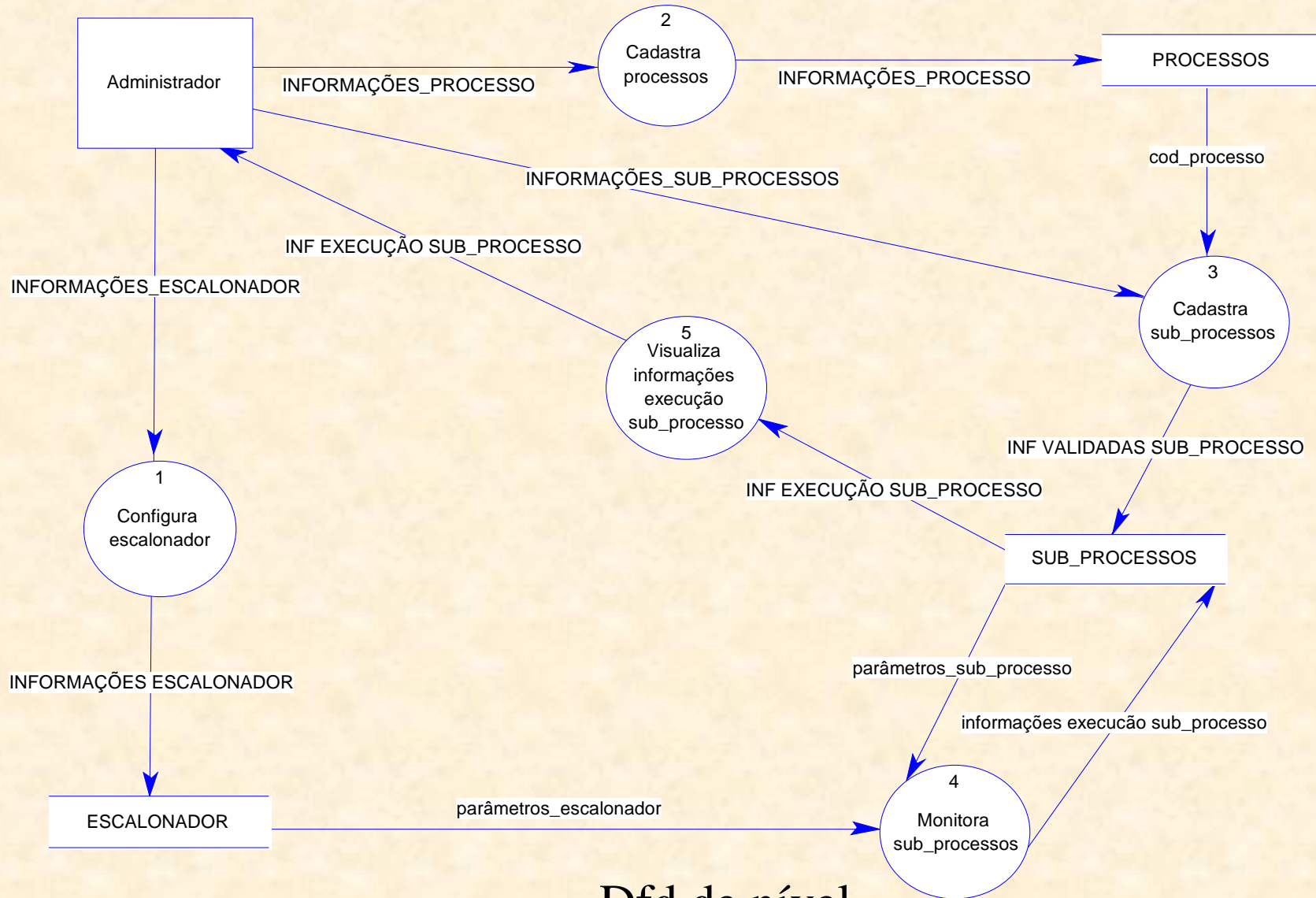
- ✓ Arquitetura UNIX-*like* e foi escolhido por ser *freeware*.

# Desenvolvimento da Ferramenta

## ESPECIFICAÇÃO DE REQUISITOS:

- ✓ manter processos ativos;
- ✓ procurar por mensagens de erro nos logs gerados;
- ✓ identificar aplicações ativas porém, inoperantes;
- ✓ monitorar espaço em estrutura física.

# Especificação



Dfd de nível

# Implementação

Linguagens utilizadas:

Escalonador: C

Base de dados: PostGre

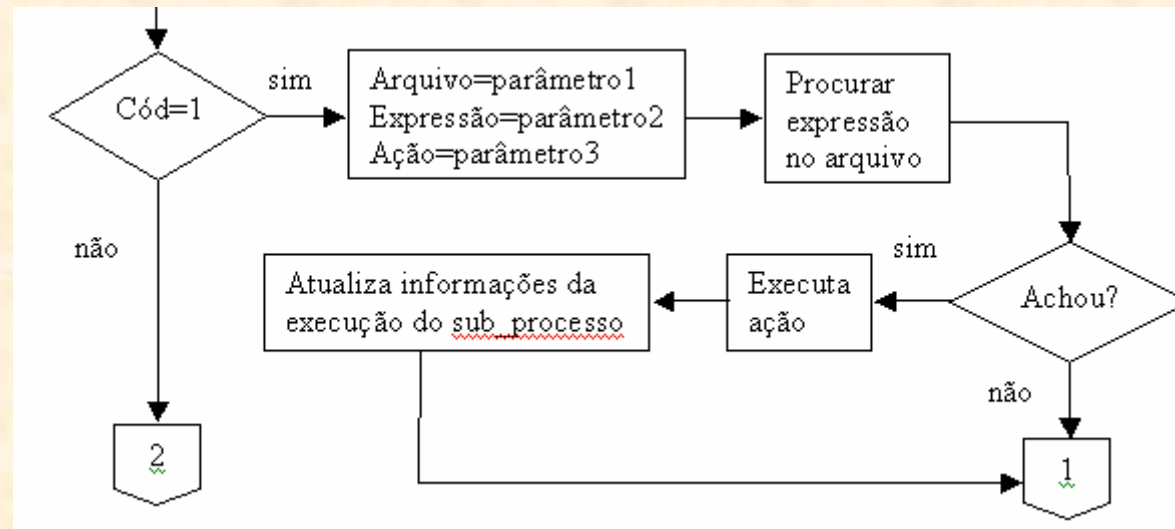
Interface da aplicação: PHP

```

switch(p_cod_processo)
{
  case 1: printf("%d-%s \n",p_cod_processo, p_dsc_processo);
          // wrk_a=arquivo de log
          // wrk_b=string a ser procurada
          // wrk_c=comando a ser executado caso a string seja encontrada.

          strcpy(wrk_comando, "grep -q "); // montar comando para executar no s.o
          strcat(wrk_comando, wrk_b);
          strcat(wrk_comando, " ");
          strcat(wrk_comando, wrk_a);
          execucao = system(wrk_comando); // executar comando e guardar retorno
          if (execucao == 0 ) // se a string foi encontrada, executa
            {

```



Web Admin - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop  Search Print

Home Bookmarks Red Hat Network Support Shop Products Training

[Pagina inicial](#) [Scheduler](#) [Processos](#) [Sub-Processos](#)

# Processos

Codigo:

Nome:

Descricao:

Caminho do log:

Codigo do processo	Nome	Descricao	Caminho do log
1	Buscar palavra	Buscar por uma expressao em um log	/tcc/prog/logs/buscar_palavra.log
2	Verificar data logs	Verificar data de ultima alteracao do log	/tcc/prog/logs/tpo_atualz_logs.log
3	Iniciar Prog.	Iniciar programa se estiver inativo	/tcc/prog/logs/iniciar_progs.log
4	Verif. espaco	Verificar espaco em disco	/tcc/prog/logs/espaco_em_disco.log

Done

Web Admin - Mozilla

Sun Nov 14 6:50 PM



Web Admin - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop  Search Print

Home Bookmarks Red Hat Network Support Shop Products Training

[Pagina inicial](#) [Scheduler](#) [Processos](#) [Sub-Processos](#)

# Sub-processos

Codigo do processo:

Codigo do sub-processo:

Nome:

Caminho do log:

[Parametros:](#)

Buscar

Alterar

Inserir

Apagar

Cod Pro	Cod Sub	Nome	Caminho do log	Mensagem do log	Data ultima exec.	Parametros
1	1	Atualizar base	<a href="#">/tcc/prog/logs/sub_01_01.log</a>	String encontrada!	02/12/2004 16:34:11	/tcc/prog/logs/actual
2	1	Registra acesso	<a href="#">/tcc/prog/logs/sub_02_01.log</a>	Log nao atualizado no tempo esperado!	12/12/2004 15:38:16	/tcc/prog/logs/regis
3	1	IBM - MQSeries	<a href="#">/tcc/prog/logs/sub_03_01.log</a>	Programa nao esta ativo, sera iniciado.	12/12/2004 15:57:43	/tcc/prog/bin/amqcr
4	1	/dev/hda3	<a href="#">/tcc/prog_logs/sub_04_01.log</a>	Utilizacao da estrutura em estado critico !	12/12/2004 15:57:43	/dev/hda3#99#/tcc/

Done

Web Admin - Mozilla root@localhost:/var/www/html

root@localhost:~

Tue Dec 14 1:11 AM

Web Admin - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop  Search Print

Home Bookmarks Red Hat Network Support Shop Products Training

[ler](#) [Processos](#) [Sub-Processos](#)

# rocessos

mensagem do log	Data ultima exec.	Parametros
ing encontrada!	02/12/2004 16:34:11	/tcc/prog/logs/atualizar_base.log#NOT CONNECTED#/tcc/prog/shells/iniciar_listener.sh
g nao atualizado no tempo esperado!	12/12/2004 15:38:16	/tcc/prog/logs/registra_acesso.log#900#/tcc/prog/shells/registra_acesso.sh
ograma nao esta ativo, sera iniciado.	12/12/2004 15:57:43	/tcc/prog/bin/amqcrsta#/tcc/prog/shells/iniciar_mqseries.sh
ilizacao da estrutura em estado critico !	12/12/2004 15:57:43	/dev/hda3#99#/tcc/prog/shells/limpar_hda3.sh

Done

Web Admin - Mozilla root@localhost:/var/www/html

root@localhost:~

Tue Dec 14 1:12 AM



# Testes e Validação

- ✓ Atualização de um arquivo com uma expressão;
- ✓ Verificação da data de atualização de um arquivo;
- ✓ Monitoramento de um processo;
- ✓ Verificação da utilização de uma estrutura.

# Conclusões

- ✓ A ferramenta mostrou-se eficiente;
- ✓ avaliou os problemas sempre da forma correta;
- ✓ não aplicou “soluções indevidas”;
- ✓ tempo de avaliação dos sub-processos é comprometido se existirem muitos sub-processos cadastrados.

# Extensões

- ✓ Criar um módulo para cada tipo de processo (para melhorar a performance);
- ✓ desenvolver um “status” para cada tipo de sub-processo: ativo e inativo, para que possa existir um histórico.
- ✓ trabalhar conceito de prioridade entre os sub-processos.