

Containerização

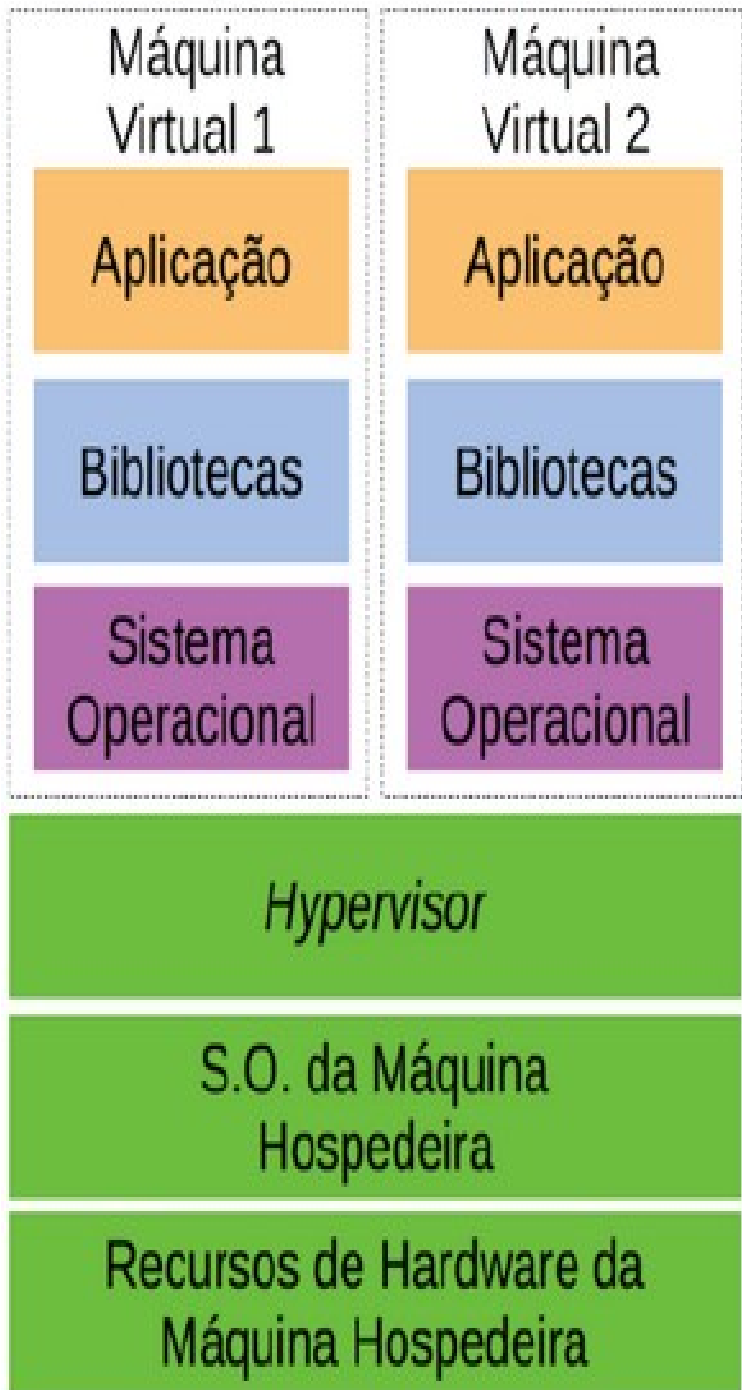
Sistemas Distribuídos

Eduardo Furlan Miranda

2025-09-28

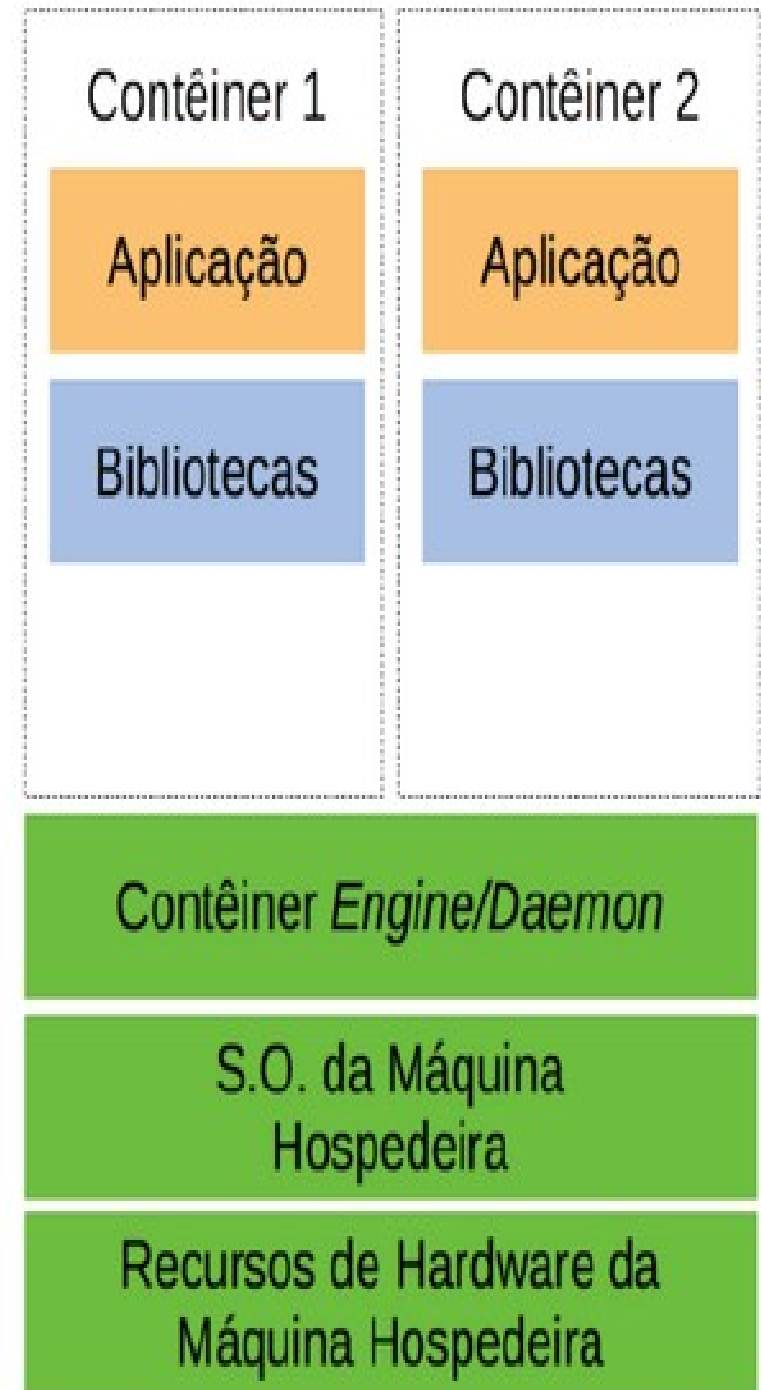
Adaptado de: PEREIRA, C. S. Sistemas Distribuídos.
Londrina: EDE SA, 2019. ISBN 978-85-522-1443-4.

Máquinas Virtuais



Contêineres

4

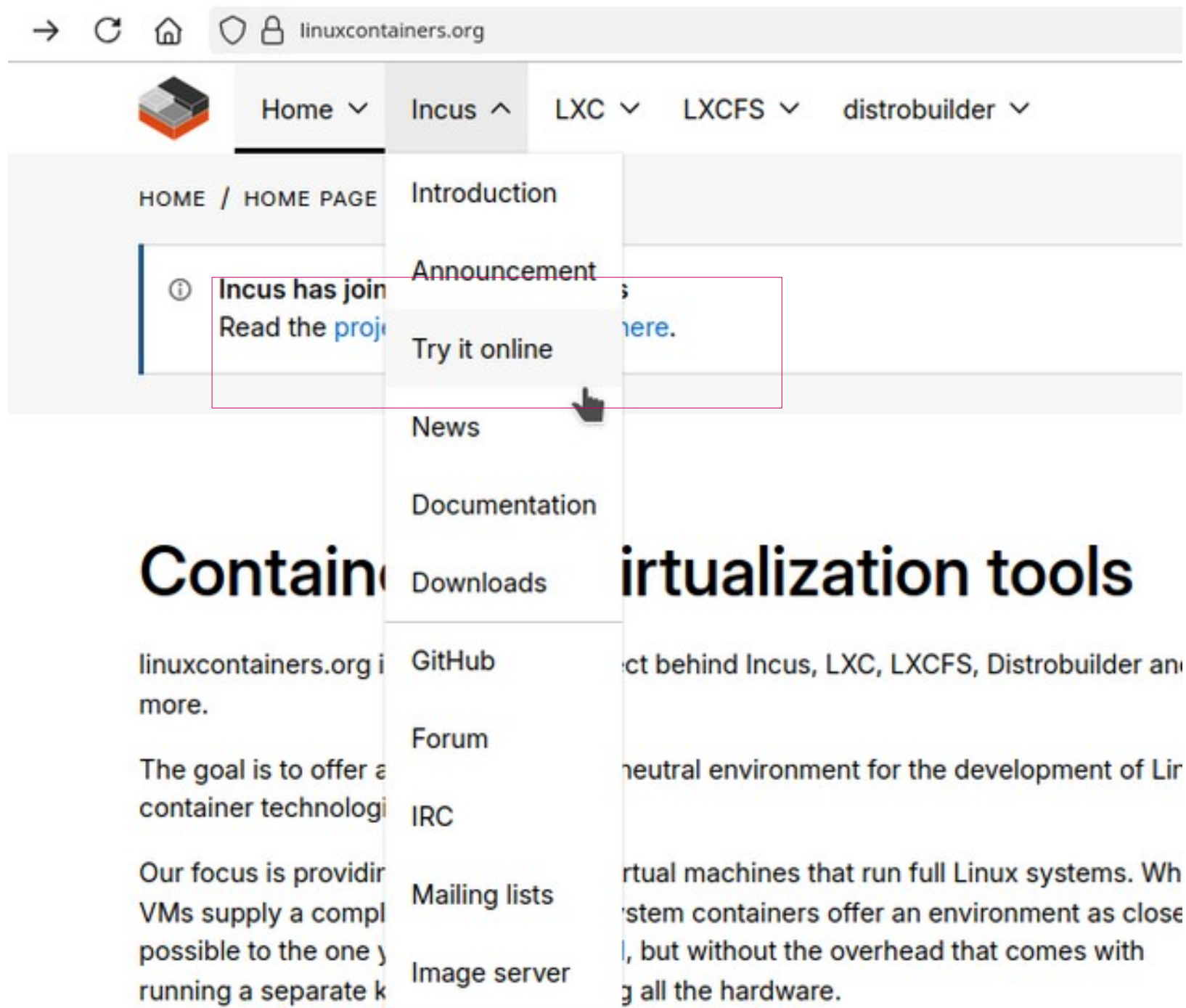


- O contêiner isola os processos de uma aplicação do resto do sistema
 - A aplicação dentro do contêiner roda nativamente no kernel do sistema operacional do hospedeiro
 - Não há uma camada de tradução ou simulação
- Um contêiner não é uma "emulação" da aplicação
 - Emulação é quando um sistema (software ou hardware) finge ser outro sistema diferente
 - P. ex. um emulador de Super Nintendo no seu computador
 - O seu PC não entende o código do Super Nintendo diretamente
 - O emulador precisa traduzir as instruções e o hardware em tempo real

- A ideia do container é colocar dentro tudo que precisa para rodar um determinado software
 - Na máquina que vai rodar o container não precisa instalar tudo de novo
- Permite a criação de ambientes virtuais isolados, e um não interfere no outro
 - Instala uma versão de um software em um container, e outra versão do software em outro container, e um não interfere no outro
- Container é mais leve que máquina virtual pois não precisa instalar o sistema operacional completo no container
 - Compartilha o Kernel da máquina hospedeira

- Quando temos várias versões de softwares em uma máquina, é possível usar gerenciadores de ambiente, como “conda”
- Container: quando precisamos rodar em máquinas e ambientes diferentes e não sabemos se tem todos os softwares instalados,
 - ou então se vamos utilizar Python, PostgreSQL, e Nginx cada um com seu ambiente, ou
 - se precisamos garantir que o ambiente de execução seja 100% reproduzível por qualquer pessoa, em qualquer lugar
- O container contém, além das versões de softwares, o sistema de arquivos, as dependências do sistema operacional, e até mesmo o próprio sistema operacional
 - Ex.: container rodando Fedora em um hospedeiro Ubuntu

- Container aproveita o Kernel do hospedeiro
 - No Windows aproveita o WSL2(Linux) e NT Kernel
 - No Linux um container windows não funciona
- Container por ser mais leve, permite usar vários na mesma máquina, criando uma arquitetura modular para aplicações
- Fácil de migrar de máquina, usar em cloud
- Container engines: Docker, LXD, Windows Server Containers, ...
- Supercomputador: Apptainer (não precisa de root)
- O Kernel do Linux possui recursos nativos para containers
 - A implementação de referência é o LXD
 - LXC: parte de baixo nível do LXD; LXCFS: sistema de arquivos



The screenshot shows the website linuxcontainers.org. The navigation bar includes links for Home, Incus, LXC, LXCFS, and distrobuilder. The Incus dropdown menu is open, showing options: Introduction, Announcement, Try it online (highlighted by a mouse cursor), News, Documentation, Downloads, GitHub, Forum, IRC, Mailing lists, and Image server. A sidebar on the left contains a notification: 'Incus has joined the Linux Foundation' with a link to 'Read the project page here.' The main content area features the heading 'Container virtualization tools' and introductory text about the project's goals and focus.

linuxcontainers.org

Home ▾ Incus ▲ LXC ▾ LXCFS ▾ distrobuilder ▾

HOME / HOME PAGE

Incus has joined the Linux Foundation
Read the [project page](#) here.

Try it online

Introduction
Announcement
News
Documentation
Downloads
GitHub
Forum
IRC
Mailing lists
Image server

Container virtualization tools

linuxcontainers.org is the project behind Incus, LXC, LXCFS, Distrobuilder and more.

The goal is to offer a neutral environment for the development of Linux container technology.

Our focus is providing virtual machines that run full Linux systems. When system containers offer an environment as close as possible to the one you are running a separate kernel, but without the overhead that comes with all the hardware.



Menu

INCUS / TRY IT ONLINE

Step by step instructions ^

1

2

3

4

5

6

7

8

9

1: Introduction

You are now inside an Incus virtual machine running on a remote Incus cluster.

Terminal (29 minutes, 10 seconds remaining) (Try a [Web UI](#))

```
demo@incus-os:~$
```



```
demo@incus-os:~$ incus launch images:ubuntu/22.04
MeuPrimeiroContainer
Launching MeuPrimeiroContainer
```

```
demo@incus-os:~$ incus list
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
MeuPrimeiroContainer	RUNNING	10.21.232.102 (eth0)	fd42:71f7:53e5:ddb5:1266:6aff:fe7c:5847 (eth0)	CONTAINER	0

NAME	STATE	IPV4	IPV6...	TYPE	SNAPSHOTS
MeuPrimeiroContainer	RUNNING	10.21.232.102 (eth0)	fd42....	CONTAINER	0

```
demo@incus-os:~$ incus exec MeuPrimeiroContainer -- lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.5 LTS
Release:        22.04
Codename:       jammy
```

```
demo@incus-os:~$ incus info MeuPrimeiroContainer
Name: MeuPrimeiroContainer
Description:
Status: RUNNING
Type: container
Architecture: x86_64
PID: 1530
Created: 2025/09/30 16:36 UTC
Last Used: 2025/09/30 16:37 UTC
Started: 2025/09/30 16:37 UTC
```

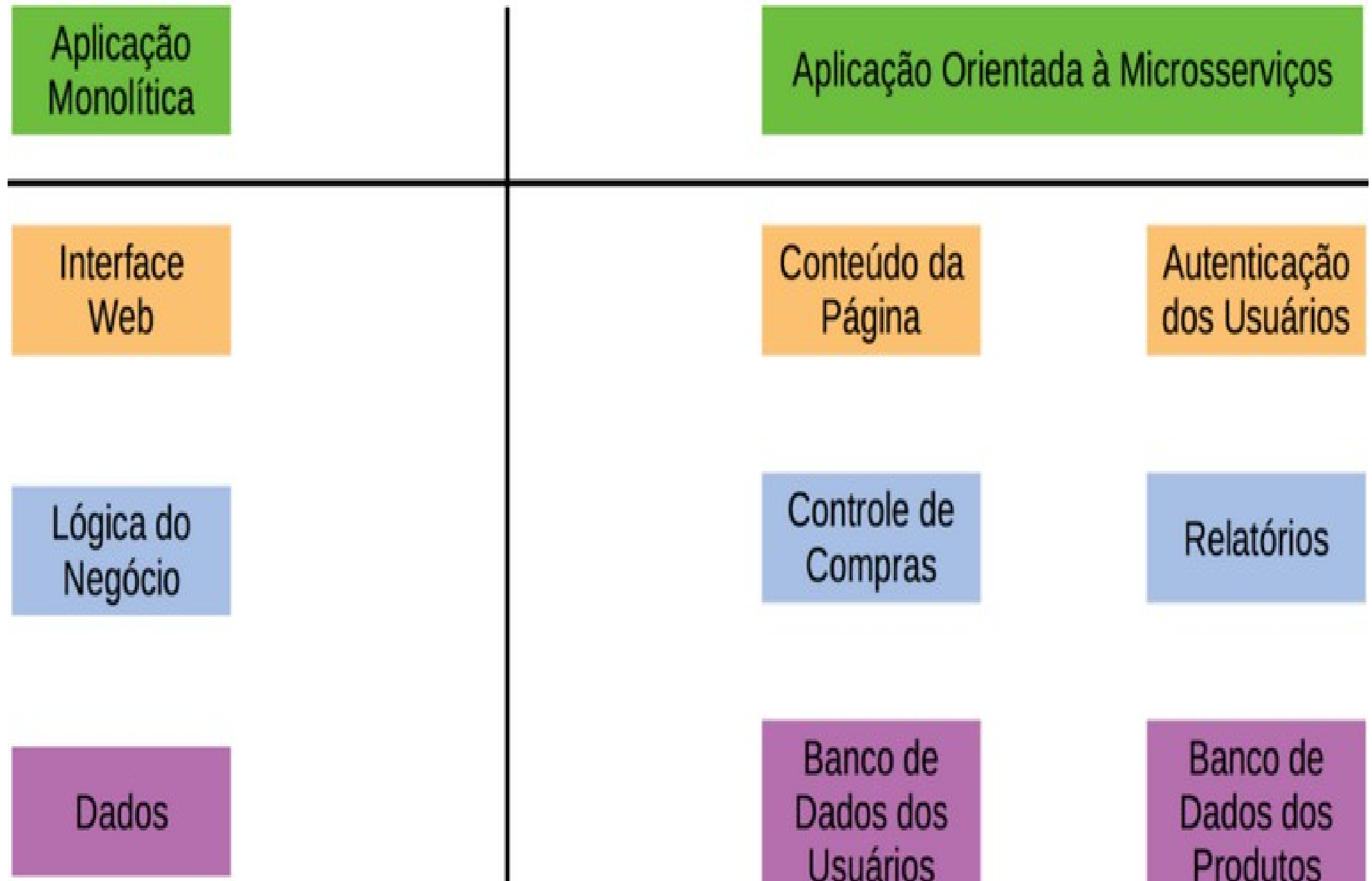
Resources:

```
Processes: 13
Disk usage:
  root: 3.99MiB
CPU usage:
  CPU usage (in seconds): 3
Memory usage:
  Memory (current): 58.40MiB
Network usage:
(...)
  Packets received: 0
  Packets sent: 0
  IP addresses:
    inet: 127.0.0.1/8 (local)
    inet6: ::1/128 (local)
```

```
demo@incus-os:~$ incus stop  
MeuPrimeiroContainer  
demo@incus-os:~$ incus delete  
MeuPrimeiroContainer
```

Container e sistemas distribuídos

- Sistemas distribuídos fazem uso extensivo dos contêineres no contexto de microsserviços
- Coloca-se banco de dados em um container, framework para front-end em outro, back-end em outro, etc
- Cria-se fraca dependência entre os ambientes, tornando mais modular, isolado, um não interfere no outro
- Mais fácil de substituir um componente, fazer manutenção, de escalar, de ter o mesmo ambiente entre PCs
- Desenvolvedores trabalham independentemente nos módulos
- Sem container precisa instalar todas as dependências no hospedeiro, e podem surgir conflitos



- Ferramentas de orquestração de contêineres são usadas para gerenciar uma quantidade grande, ex.: Kubernetes/Google, e Swarm/Docker

