FACULDADE PITÁGORAS

WILLIAN PINTO DA SILVA ARAUJO

Projeto Integrado IV.

SÃO JOSÉ DE RIBAMAR

2022

WILLIAN PINTO DA SILVA ARAUJO

Projeto Integrado IV.

Atividade avaliativa apresentada à FACULDADE PITÁGORAS, para a disciplina Projeto Integrado IV, do 4º período, do curso de Análise e desenvolvimento de Sistemas.

SÃO JOSÉ DE RIBAMAR

2022

Sumário

[INTRODUÇÃO 5](#_Toc103444782)

[DESENVOLVIMENTO 6](#_Toc103444783)

[CONSIDERAÇÕES FINAIS 8](#_Toc103444784)

[REFERÊNCIAS 10](#_Toc103444785)

# INTRODUÇÃO

Este Trabalho visa responder ao questionário dos Projetos Integrados IV do curso de análise e desenvolvimento do sistema, voltado a área de tecnologia.

# DESENVOLVIMENTO

**Tarefa 1:**

Vamos supor que você vai iniciar uma jornada empreendedora. Junto com alguns sócios, decidiu criar um negócio a partir da ideia de uma aplicação inovadora. Esta aplicação precisa de um ambiente de TI com recursos computacionais de alto desempenho e ambiente de nuvem devido à redução de cursos e abstração da complexidade da infraestrutura. O próximo passo é determinar qual dos modelos de serviço é mais apropriado para este caso: SaaS, PaaS ou IaaS. Descreve analisando caso a caso.

**R:**

A nomenclatura IaaS, PaaS e SaaS refere-se ao modelo de fornecimento de soluções em nuvem.

Os termos significam:

* PaaS - Plataforma como Serviço.
* IaaS - Infraestrutura como Serviço.
* SaaS - Software como Serviço.

Para tal cenário teórico, para recursos computacionais de alto desempenho e ambiente de nuvem faz-se necessário contratar um serviço de Infraestrutura como Serviço (IaaS). Com isso, é possível terceirizar toda infraestrutura da aplicação e se preocupar apenas com o desenvolvimento da aplicação.

**Tarefa 2:**

Com o desenvolvimento dos sistemas, surgiu a necessidade de, dentro de um mesmo processo, possuir trechos que executassem concorrentemente, e aí temos o conceito de thread.

1. Definição de thread;
2. Pesquise os estados das threads;
3. Os benefícios da programação multithread;
4. Operações de thread.

**R:**

1. Ordem de execução, com instruções encadeadas que são desempenhadas uma por vez.

* NEW - ela foi criada e está pronta para iniciar (start())
* RUNNABLE - ela está rodando (não existe o estado RUNNING)
* BLOCKED - ela está travada, em geral por Lock ou alguma operação de IOWAITING - ela está aguardando outra thread para rodar
* TIMED\_WAITING - a mesma coisa, mas há um limite de tempo que ela esperará
* TERMINATED - ela terminou a execução, mas ainda existe (não existe o estado DEAD)

1. Capacidade de resposta - o multithreading - permite que um programa continue em execução mesmo se uma parte dele estiver bloqueada ou executando uma operação demorada, aumentando assim a capacidade de resposta ao usuário. Compartilhamento de recursos, permite que um aplicativo tenha vários threads de atividade no mesmo espaço de endereço. Economia alocar memória e recursos para a criação de processos, encadeamentos compartilham memória com o processo ao qual pertence, é mais econômico criar e alternar encadeamentos de contexto.
2. criar, terminar, thread join e thread

**Tarefa 3:**

Descreva detalhadamente todos os métodos HTTP existentes e cite um exemplo de utilização para cada um deles.

**R:**

* GET: é o mais utilizado, tem a função de solicitar ou requisitar algum recurso ao servidor, É geralmente usado para o retorno d;e informações sobre uma entidade especificamente ou uma lista delas.

Ex: GET /voutestarissoaqui/ HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)

Host: http://testealgumacoisa.com

Connection: Keep-Alive

* POST: usado para enviar informações ao servidor, é utilizado de forma mais popular em cadastros de dados retirados de um formulário, e também quando enviamos um arquivo para o servidor

Ex: POST /cadastro.php HTTP/1.1

Host: www.teste.com.br

nome=fulano&sobrenome=etc

* PUT: é responsável por editar arquivos e informações já existentes. Se o recurso já existir, ele deve ser atualizado. Se não existir, pode ser criado.

Ex: PUT /new.html HTTP/1.1

Host: example.com

Content-type: text/html

Content-length: 16

* DELETE: deleta arquivos ou informações presentes no banco de dados.

Ex: DELETE /arquivo.html HTTP/1.1

* HEAD: retorna apenas o cabeçalho da resposta

Ex: HEAD /index.html

* OPTIONS: É responsável por retornar as informações referentes ao servidor.

Ex: OPTIONS /index.html HTTP/1.1

* TRACE: reenvia a última requisição para o servidor e verifica se houve alguma interferência de servidores de terceiros.

Ex: TRACE /index.html

* CONNECT: é responsável por conectar a API com algum servidor proxy. É mais utilizado para tentar a conexão com sites que possuem segurança e que não permitem conexão direta

Ex: CONNECT www.exemplo.com:443 HTTP/1.1

**Tarefa 4:**

Em aplicações Android podemos trabalhar com a persistência de dados locais por meio da criação de bancos de dados utilizando a biblioteca SQLite. Descreva as principais características do SQLite para aplicações Android, bem como as principais vantagens e desvantagens de sua utilização.

**R:**

# É formado por uma biblioteca em linguagem C, essa biblioteca está disponível para diversas plataformas, sendo a mais utilizada no sistema Android, o SQLite é um dos bancos de dados relacional mais conhecidos, por ser capaz de criar uma estrutura com um banco de dados embutido. Funciona como um servidor próprio e independente, pode ser executado na mesma instância – eliminando assim as consultas e processos separados. A biblioteca SQLite é gerada e armazenada diretamente no arquivo do banco de dados.

# O SQLite no Android:

# As principais características do SQLite para aplicações Android são:

# Possui um banco de dados SQL embutido.

# É uma biblioteca em linguagem C.

# Possui diversas funções para manipulação de dados.

# É gratuito e de código aberto.

# Vantagens:

# É estável, multiplataforma e compatível com versões anteriores;

# Seu código é de domínio público e gratuito, ou seja, todos podem utilizar;

# Não necessita de instalação ou configuração;

# É gratuito e de código aberto;

# Guarda o banco de dados em um único arquivo.

# Desvantagens:

# É mais simples e não suporta um volume grande de informações;

# É mais restrito em relação aos formatos dos arquivos aceitos.

# A API do SQLite não é tão robusta quanto às outras bibliotecas de banco de dados

# CONSIDERAÇÕES FINAIS

Através desta atividade pude expandir mais meus conhecimentos sobre modelos de serviço SaaS, PaaS ou IaaS. Aprender mais sobre o conceito de thread, suas formas de operação e benefícios do multithread. Conhecer de forma mais detalhada os métodos http e suas aplicações e as vantagens e desvantagens de utilizar o SQLite em aplicações Android.

# REFERÊNCIAS

* Anderson, Tim (2007-06-21). "Size isn't everything for the modest creator of SQLite". The Guardian. Archived from the original on 2021-10-24. Retrieved 2018-01-31.
* Allen, Grant; Owens, Mike (2011). The Definitive Guide to SQLite. Apress. ISBN 9781430232261.
* Kenny, Kevin (16 December 2008). "TCL CORE TEAM ANNOUNCES: Harrison, Hopp, Ingham, Welch leave Tcl Core Team". tcl-core (Mailing list). Archived from the original on 2021-10-24.
* "O'Reilly Open-Source Convention 2004 - Speaker". O'Reilly. 2004. Archived from the original on 2021-10-24.
* "#201: SQLite with Richard Hipp - The Changelog". The Changelog. 2016-04-29. Archived from the original on 2021-10-24. Retrieved 2016-05-03.