Unyleya
EDUCACIONAL

Microservices

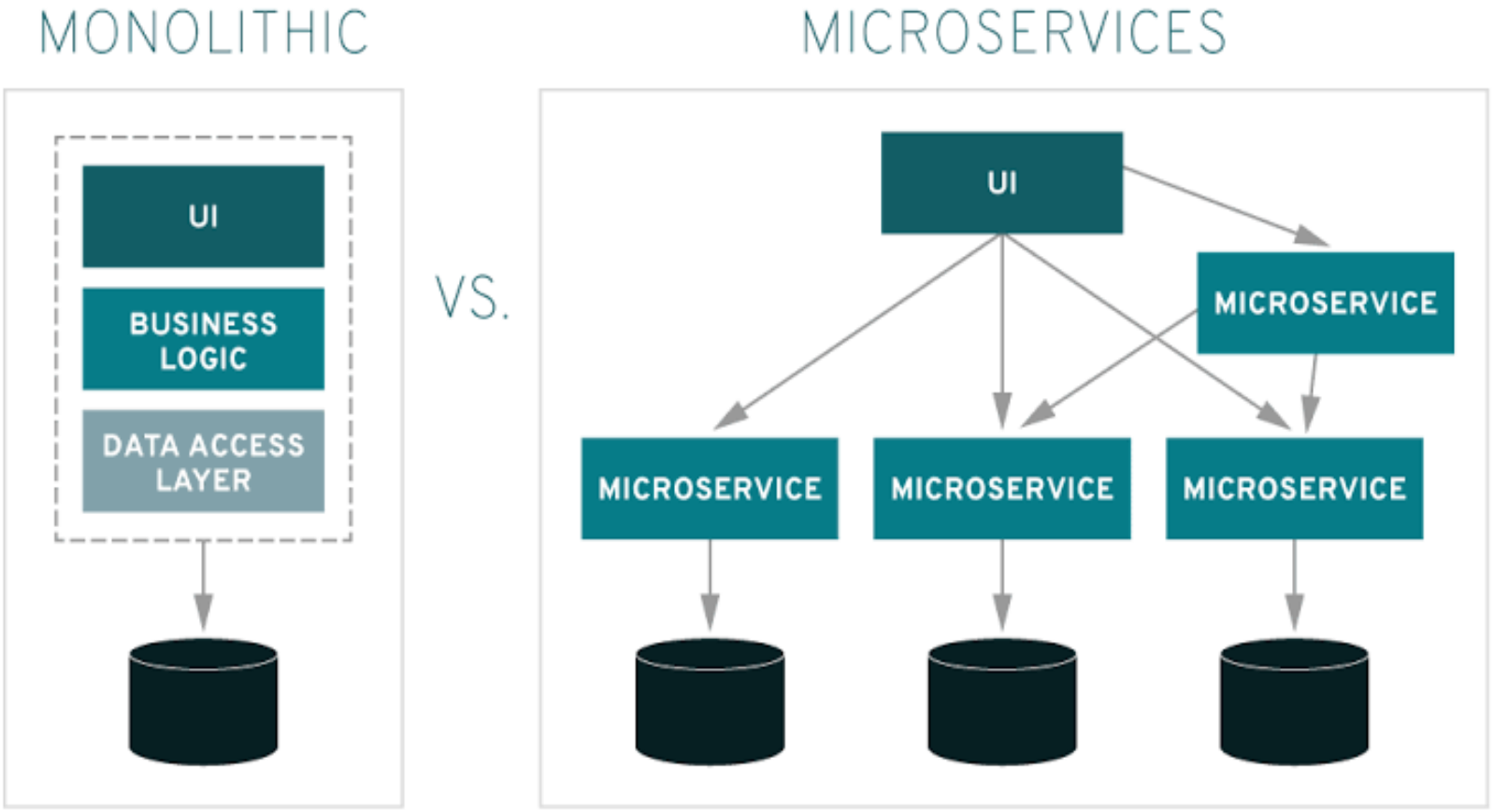
O conceito de microservices

Microservices ou microserviços é uma arquitetura utilizada para a construção de softwares, oferecendo uma opção a arquitetura monolítica.

A arquitetura monolítica trabalha com uma estrutura única para a prestação de todos os serviços relativos ao software.

Já com a arquitetura de microserviços, as principais funcionalidades de uma aplicação podem ser segmentadas em serviços que se complementam, mas são isolados.

Microserviços x Monolítica



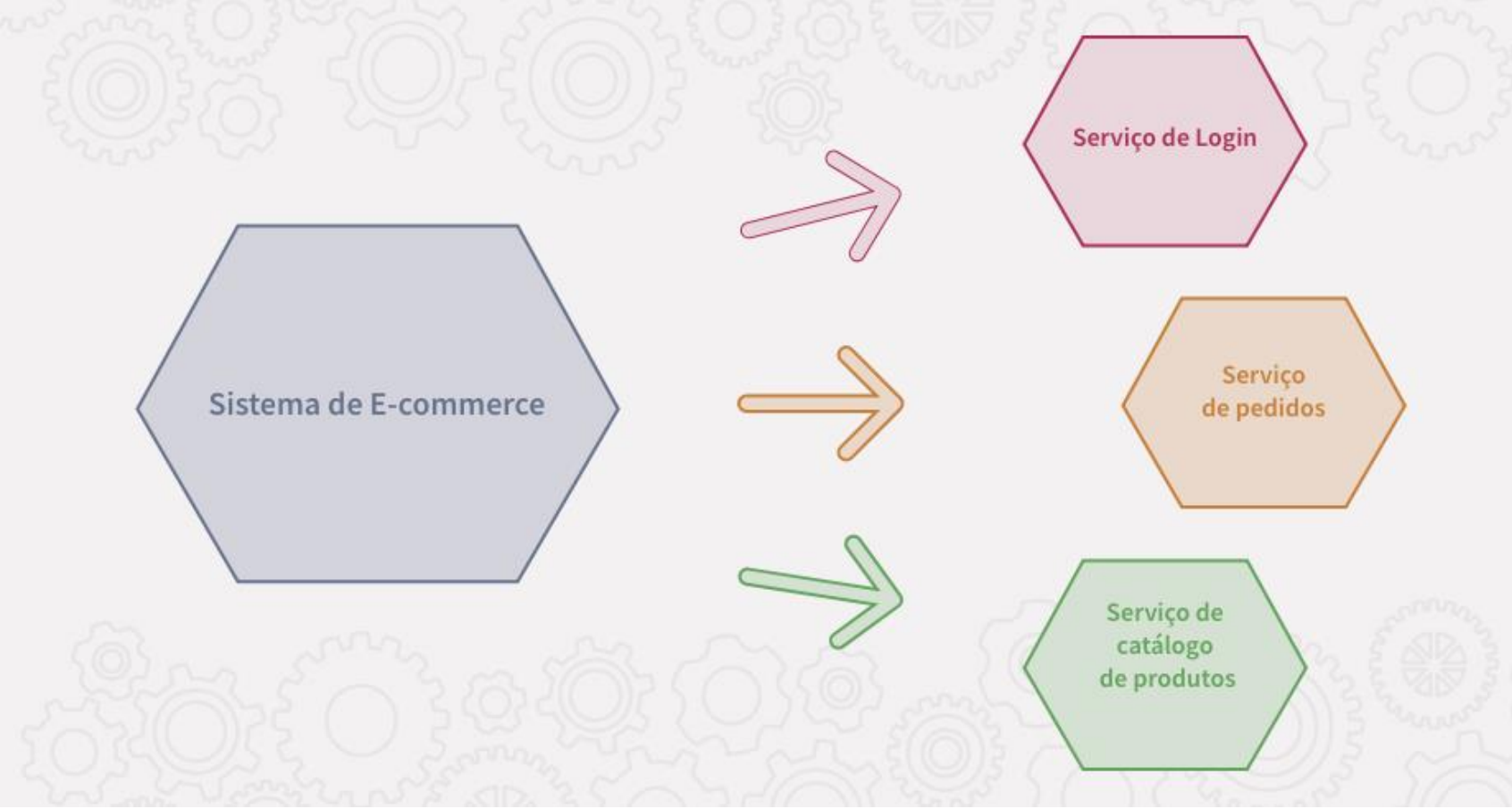
O conceito de microservices

É possível afirmar que um software desenvolvido na arquitetura de microserviços na verdade é um conjunto de pequenos softwares.

Para cada um dos pequenos softwares é criado um processo específico e se comunicam através de recursos muito leves.

Com isso é possível dar manutenção e realizar o deploy de partes específicas do software. Isso facilita muito a prestação de um serviço de qualidade para os usuários finais, mais rápido e mais acessível.

Exemplo microservices



A arquitetura monolítica

A arquitetura monolítica coloca todos os serviços dentro de uma única estrutura, que é executada através de processo único.

No aplicativo monolítico a escalabilidade só pode ser realizada de forma horizontal, através do balanceamento de carga.

Referências

https://www.redhat.com/pt-br

Obrigada!

Ana Laurentino

Unyleya
EDUCACIONAL

Quando utilizar os microservices

Por que utilizar a arquitetura de microservices?

Os microserviços devem ser utilizados quando se estiver trabalhando com a construção de um software com domínios bastante específicos.

Por exemplo, se tiver um software que é dividido para uso interno e acesso externo. É interessante que se separe as duas perspectivas em microserviços, fazendo com que as regras de negócio de acesso externo sejam diferentes do acesso interno.

Por que utilizar a arquitetura de microservices?

Ao se dividir uma aplicação em pequenas partes, a escalabilidade ganha novas possibilidades, porque passa a ser possível escalar cada um dos microserviços de forma independente. Com isso podem ser escalados somente os serviços mais críticos.

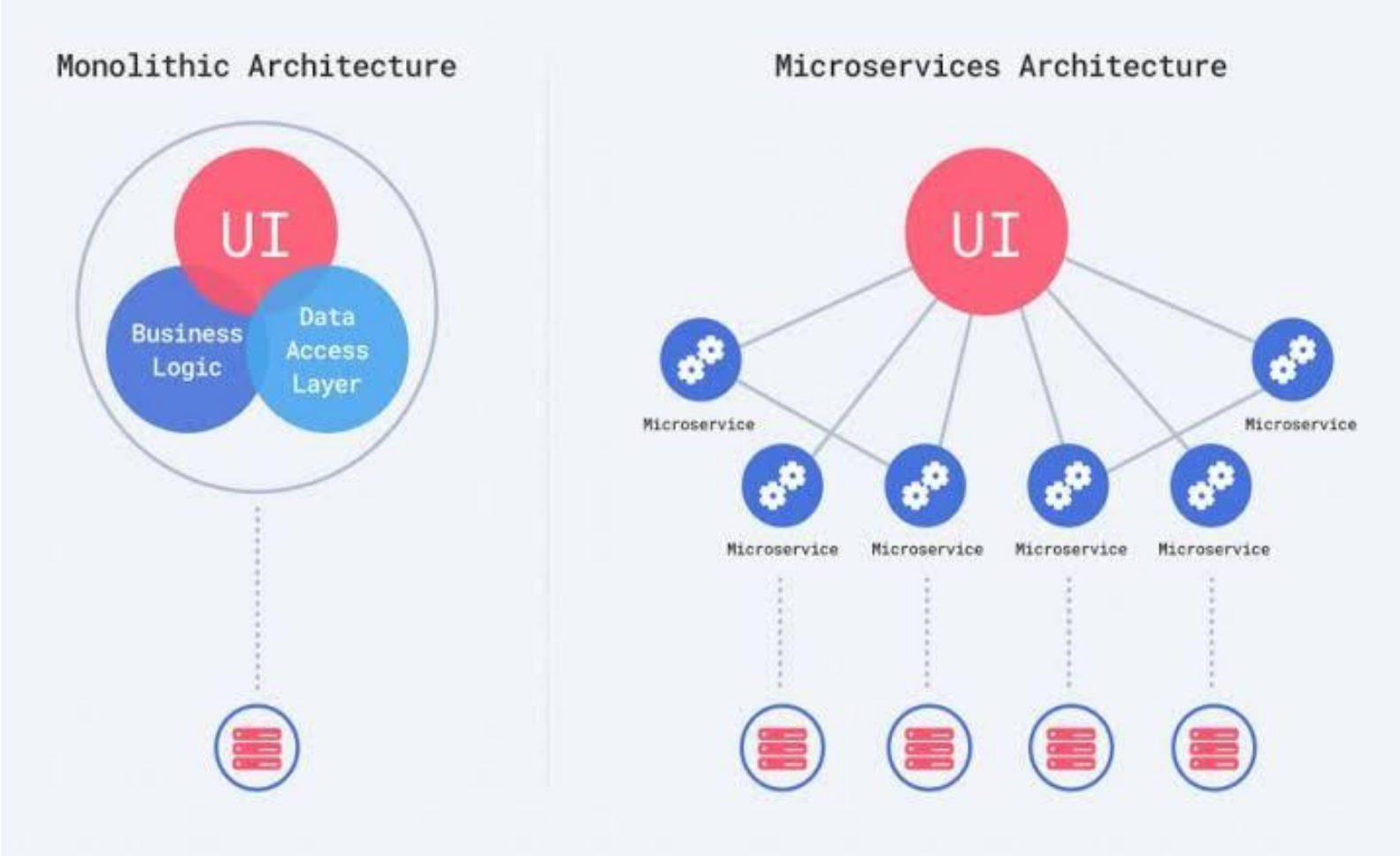
Supondo que um microserviço de compras seja o mais utilizado dentro de um e-commerce, é possível destinar os recursos de processamento mais para ele do que para os demais.

Por que utilizar a arquitetura de microservices?

Um outro grande destaque da utilização dos microserviços é a possibilidade de se isolar tecnologias utilizadas.

Com isso podem ser utilizadas tecnologias diferentes em cada um dos microserviços, fazendo com que seja possível escolher sempre as melhores para cada caso, não tendo que se preocupar tanto com a compatibilidade entre elas.

Por que utilizar a arquitetura de microservices?



Fonte: Felix (2019)

Por que utilizar a arquitetura monolítica?

Quando não se tem domínios específicos para serem separados em diferentes microserviços é mais fácil utilizar o desenvolvimento na arquitetura monolítica.

A debugação do código também é mais simples do que na arquitetura de microserviços, que precisa de ferramentas específicas.

Por que utilizar a arquitetura monolítica?

O deploy também pode ser executado de forma única facilitando assim a atualização da aplicação.
A preocupação de afetar outros serviços não existe, porque ele é único.

A quantidade de dados que são trafegados na rede também é menor por conta do processo único.

Referências

FELIX, W.; Como escolher entre arquitetura de microsserviços e monolítica. 2019. Disponível em: <https://waldyrfelix.com.br/os-crit%C3%A9rios-de-decis%C3%A3o-para-escolher-entre-os-estilos-de-arquitetura-monol%C3%ADtica-e-microsservi%C3%A7o-f2534be6a575>. Acesso em 20 Jan. 2020.

Obrigada!

Ana Laurentino

Unyleya
EDUCACIONAL

O protocolo REST

O protocolo REST

O **RE**presentational **S**tate **T**ransfer, popularmente conhecido de REST, é um protocolo utilizado para o desenvolvimento de aplicações distribuídas.

O protocolo foi proposto por um dos criadores do tão conhecido protocolo HTTP.

Os desenvolvedores perceberam uma outra utilidade para o REST, no desenvolvimento de softwares na arquitetura de microserviços.

O protocolo REST

O conjunto de princípios do padrão REST pode ser utilizado para facilitar alguns aspectos do desenvolvimento de aplicações web.

Todas aplicações lidam com a gerência de informações, que para o protocolo REST é chamada de recursos.

Os recursos no REST são tipificados, como se fosse uma categorização das informações, pois é importante que cada recurso tenha um identificador único.

O protocolo REST

A identificação única para cada recurso é importante para que a aplicação possa sempre identificar sobre qual recurso deve operar.

O processo de identificação de qual recurso o usuário está solicitando ação é realizado através do chamado Uniform Resource Identifier - URI.

Princípios REST

São princípios do funcionamento do REST:

- Dê a todas as coisas um Identificador;
- Vincule as coisas;
- Utilize métodos padronizados;
- Recursos com múltiplas representações;
- Comunique sem estado.

Criando um identificador

Exemplos de identificadores URI:

- http://example.com/customers/1234
- http://example.com/orders/2007/10/776654
- http://example.com/products/4554
- http://example.com/processes/salary-increase-234

Vinculando as coisas

```
<order self="http://example.com/customers/1234">
   <amount>23</amount>
   <product ref="http://example.com/products/4554"></product>
   <customer ref="http://example.com/customers/1234"></customer>
 </order>
```

Utilizando métodos padrão

```
class Resource {

      Resource(URI u);

      Response get();

      Response post(Request r);

      Response put(Request r);

      Response delete();

}
```

Referências

FELIX, W.; Como escolher entre arquitetura de microsserviços e monolítica. 2019. Disponível em: <https://waldyrfelix.com.br/os-crit%C3%A9rios-de-decis%C3%A3o-para-escolher-entre-os-estilos-de-arquitetura-monol%C3%ADtica-e-microsservi%C3%A7o-f2534be6a575>. Acesso em 20 Jan. 2020.

Obrigada!

Ana Laurentino

Unyleya
EDUCACIONAL

URI e URL

URI E URL

Os termos URI e URL podem causar confusão em muitas pessoas por ser difícil identificar automaticamente a diferença entre eles.

Todos esses termos estão relacionados ao funcionamento do protocolo HTTP.

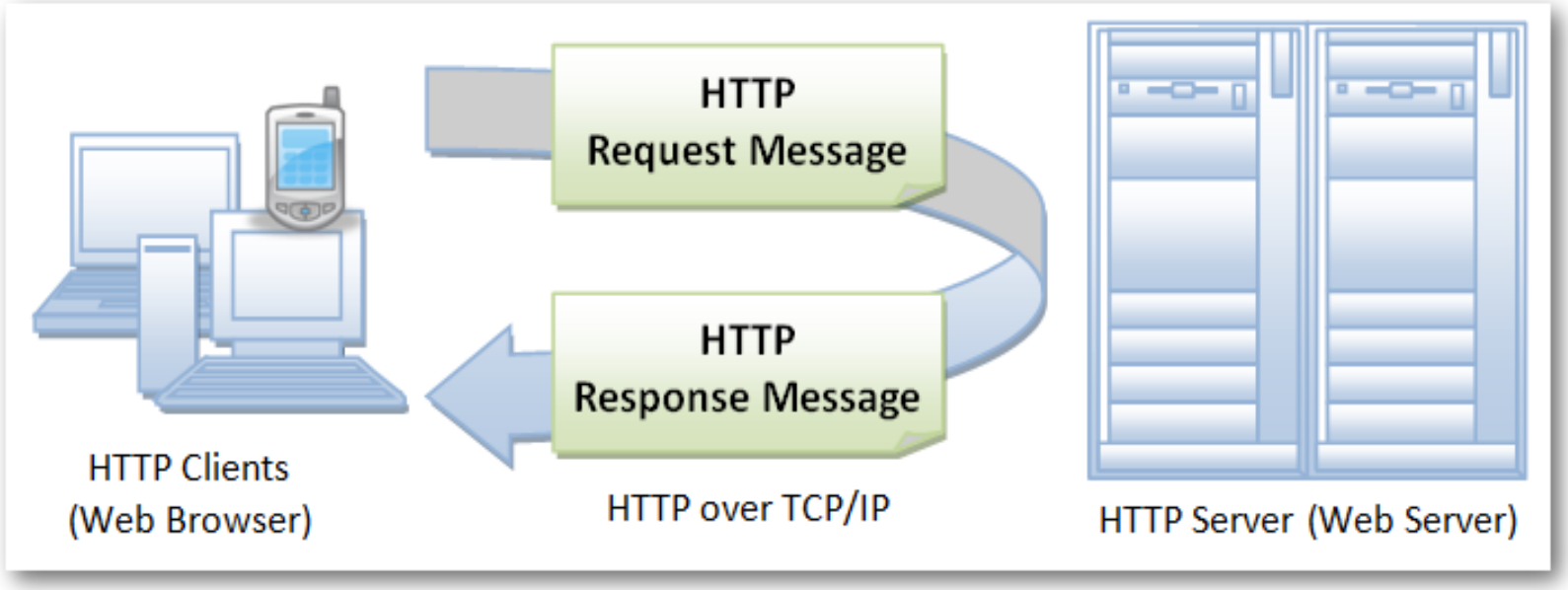
O protocolo HTTP

O protocolo HTTP propõe um conjunto de regras que devem ser seguidas para que diferentes máquinas possam estabelecer comunicação entre si, através de uma linguagem de comunicação intermediária.

O protocolo HTTP trabalha com o sistema de requisições para a comunicação entre o cliente e o servidor.

Neste protocolo, sempre que o usuário deseja solicitar uma ação ao servidor, ele encaminha um pacote de dados contendo os parâmetros necessários a uma URL ou URI.

O protocolo HTTP



URL

URL - Uniform Resource Locator, no português “Localizador de recursos universais”.

A URL contém em si informações sobre o HOST e o recurso que se deseja acessar para que a máquina receptora receba essa informações e saiba como se portar.

Cada URL faz uma associação entre um endereço com um determinado recurso disponível dentro da internet.

URL

Exemplos de URL:

- www.google.com.br
- www.globo.com.br
- www.uol.com.br

A URL pode ser entendida como uma parte da URI.

URN

URN - Uniform Resource Name, chamado de nome de recursos universal, junto com a URL irá fazer parte da URI.

São exemplos de URN:

- contato.html
- servicos.html

URI

URI - Uniform Resource Identifier, no português “Identificador de Recurso Universal”, como o nome já indica serve para a identificação de um recurso.

Esse identificador é utilizado, por exemplo, para identificar imagens, páginas, arquivos, etc. A identificação é importante porque tudo dentro da internet precisa ser único para que nunca aconteça de se confundir o retorno de uma requisição.

Uma URI é formada pela união entre a URL e URN:

- https://noticias.uol.com.br/erratas/

Referências

INFORMÁTICA PARA TODOS. Protocolo HTTP. 2017. Disponível em: <http://aebenficaonline.blogspot.com/2017/01/protocolo-http_16.html. Acesso em: 20 Jan. 2020.

Obrigada!

Ana Laurentino

Unyleya
EDUCACIONAL

Designer de URI

O designer de caminhos URI

Os segmentos de caminho URI são separados por uma barra e cada objeto que precisar ser representado abre uma possibilidade de trabalho para o designer.

Cada segmento deve ser estabelecido com nome representativo para indicar a funcionalidade que ele contém.

O designer de caminhos URI

O ideal é que os segmentos de URI sejam sempre definidos com o texto no singular para a identificação de um documento.

www.url.com.br/documentos/certificadoPaulo.pdf

Quando a URI estiver fazendo referência ao conjunto de objetos como coleções de fotografias, arquivos, textos, etc, deve-se utilizar o identificador no plural.

www.url.com.br/fornecedores/contatos

Controller

Os Controllers são responsáveis por executar determinadas ações, por isso devem ser nomeados com verbos para que fique bastante sugestivo.

Por exemplo, se o controller tive a ação de cadastro do usuário, deve-se utilizar o verbo cadastrar.

www.url.com.br/usuario/cadastrar

Segmentos variáveis

Os segmentos podem ser estáticos ou podem variar dinamicamente, recebendo valores atribuídos.

Neste momento entra o protocolo REST para fazer a substituição do valor dentro da URL.

www.url.com.br/usuario/u-19

CRUD

É muito importante para a segurança da aplicação que os nomes dos CRUDs não sejam inseridos dentro de URIs.

Os CRUDs são as operações básicas para manipulação de bases de dados.

Por exemplo, se estivermos fazendo uma operação que se aplica a tabela de usuários, não deve ser utilizado o termo.

www.url.com.br/atualizar/u-19

Obrigada!

Ana Laurentino

Unyleya
EDUCACIONAL

Arquétipos de recursos

Arquétipos de recursos

Para o manuseio de recursos é preciso compreender e utilizar os tipos possíveis:

- document;
- collection;
- store;
- controller.

Document

O documento é relativo a uma instância ou objeto de banco de dados, deve sempre ser tratado no singular.

Essa é a representação adequada para o estado de um documento.

Pode incluir campos com valores e links para recursos externos.

www.url.com.br/fornecedor/material-escolar/infantil

Collection

É um recurso que agrupa em si várias instâncias de algum objeto.

O cliente pode solicitar a inserção de um novo objeto junto a instância.

www.url.com.br/fabricante-bolsas/ecobag

Store

Se trata do repositório que é gerenciado pelo cliente.

Neste repositório é oferecido o serviço HTTP que podem ser utilizados pelo cliente.

www.url.com.br/loja-roupas/vestidos/345

Controller

Os controllers guardam modelos de ações ou procedimentos que podem ser executados.

Nesses procedimentos podem ser necessários a passagem de parâmetros, entrada e saída, além de valores para retorno.

Obrigada!

Ana Laurentino