**41** [**Q1880410**](https://www.qconcursos.com/questoes-de-concursos/questoes/2fc2af8f-a3) [Arquitetura de Software](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/arquitetura-de-software) **Prova:**[CESPE / CEBRASPE - 2022 - DPE-RO - Analista da Defensoria Pública - Redes e Comunicação de Dados](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2022-dpe-ro-analista-da-defensoria-publica-redes-e-comunicacao-de-dados)

Entre os critérios de classificação de arquiteturas para aplicações *web*, aquele que determina a disponibilidade da solução é a

**A** confiabilidade.

**B** funcionalidade.

**C** usabilidade.

**D** *performance.*

**E** suportabilidade.

A capacidade do produto de software de manter um nível de desempenho especificado, quando usado em condições

especificadas.

Suas subcaracterísticas são:

* **Maturidade:** Capacidade do produto de software de evitar falhas decorrentes de defeitos no software.
* **Tolerância a Falhas**: Capacidade do produto de software de manter um nível de desempenho especificado em casos de defeitos no software ou de violação de sua interface especificada.
* **Recuperabilidade:** Capacidade do produto de software de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.
* **Conformidade**: Capacidade do produto de software de estar de acordo com normas, convenções ou regulamentações relacionadas à confiabilidade.

fonte: <https://pt.wikipedia.org/wiki/ISO/IEC_9126#Confiabilidade>

**42** [**Q1880407**](https://www.qconcursos.com/questoes-de-concursos/questoes/2fb975cd-a3) [SOAP (Simple Object Access Protocol)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/soap-simple-object-access-protocol) **Prova:**[CESPE / CEBRASPE - 2022 - DPE-RO - Analista da Defensoria Pública - Redes e Comunicação de Dados](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2022-dpe-ro-analista-da-defensoria-publica-redes-e-comunicacao-de-dados)

O protocolo utilizado para efetuar chamadas de procedimentos remotos e que é independente de plataforma é o

**A** REST (*Representational State Transfer*).

**B** SOAP (*Simple Object Access Protocol*).

**C** JSON (*JavaScript Object Notation*).

**D** WSDL (*Web Services Description Language*).

**E** XML (*eXtensible Markup Language*).

**43** [**Q1880404**](https://www.qconcursos.com/questoes-de-concursos/questoes/2fafd144-a3) [WebServices](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/webservices) **Prova:**[CESPE / CEBRASPE - 2022 - DPE-RO - Analista da Defensoria Pública - Redes e Comunicação de Dados](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2022-dpe-ro-analista-da-defensoria-publica-redes-e-comunicacao-de-dados)

A arquitetura para *web service*s na qual os componentes são definidos em tempo de construção é denominada estrutura

**A** de concorrência.

**B** funcional.

**C** de implementação.

**D** de desenvolvimento.

**E** física.

**Gabarito: D**

Mas para mim, a resposta correta seria "arquitetura baseada em contrato". A arquitetura para web services na qual os componentes são definidos em tempo de **construção**é denominada arquitetura baseada em contrato (ou "contract-first" em inglês).

**44** [**Q1875900**](https://www.qconcursos.com/questoes-de-concursos/questoes/fc1fcaed-98) [Arquitetura de Software](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/arquitetura-de-software) **Prova:**[CESPE / CEBRASPE - 2021 - TJ-RJ - Analista Judiciário - Analista de Gestão de TIC](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-tj-rj-analista-judiciario-analista-de-gestao-de-tic)

Construído(a) em arquitetura distribuída em grande escala, com capacidade de armazenar e processar conjuntos de dados não estruturados, a fim de agrega-los sobre clientes de diferentes fontes, enriquecê-los, limpá-los e analisa-los para entender melhor às jornadas dos clientes caracteriza um(a)

**A** virtualização de dados.

**B** *storage*de objetos.

**C** *data lake.*

**D** *desktop* como serviço (DaaS).

**E** *software* como serviço (SaaS).

**45** [**Q1874541**](https://www.qconcursos.com/questoes-de-concursos/questoes/27b85719-94) [WebServices ,](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/webservices)[SOAP (Simple Object Access Protocol)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/soap-simple-object-access-protocol) **Prova:**[CESPE / CEBRASPE - 2021 - TJ-RJ - Analista Judiciário - Analista de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-tj-rj-analista-judiciario-analista-de-sistemas)

No*web service* SOAP, versão 1.2, o elemento responsável por impor as regras que gerenciam a troca de mensagens SOAP e acessar os serviços fornecidos pelos protocolos subjacentes por meio de uma ou mais ligações SOAP é conhecido como

**A** módulo SOAP.

**B** nó SOAP.

**C** função SOAP.

**D** aplicativo SOAP.

**E** recurso SOAP.

**46** [**Q1874529**](https://www.qconcursos.com/questoes-de-concursos/questoes/279443be-94) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - TJ-RJ - Analista Judiciário - Analista de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-tj-rj-analista-judiciario-analista-de-sistemas)

A coleção GoF (*Gang of Four*) é formada por padrões orientados a objetos, separados em categorias. A categoria padrões estruturais é responsável por

**A** estabelecer um *design*de objetos reutilizáveis.

**B** fornecer maneiras eficientes para criar objetos.

**C** descrever como os objetos são colocados juntos.

**D** descrever como os objetos interagem.

**E** distribuir responsabilidades entre os objetos.

A - estabelecer um *design*de objetos reutilizáveis. (O que se busca ao utilizar **PADRÕES DE PROJETO**)

B - fornecer maneiras eficientes para criar objetos. (PADRÕES **CRIACIONAIS**)

C - descrever como os objetos são colocados juntos. (PADRÕES **ESTRUTURAIS)**

D - descrever como os objetos interagem (PADRÕES **COMPORTAMENTAIS**)

E - distribuir responsabilidades entre os objetos.(**INFORMATION EXPERT** dos Padrões **GRASP**)

descrever como os objetos são colocados juntos

designs estruturais visam facilitar a relação entre entidades. os padrões são:

**Adapter** pattern

**Composite** pattern

**Decorator** pattern

**Extensibility** pattern

**Facade** pattern

**Flyweight** pattern

**Marker** pattern

**Proxy** pattern

**47** [**Q1831257**](https://www.qconcursos.com/questoes-de-concursos/questoes/2edf8bae-2f) [SOAP (Simple Object Access Protocol)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/soap-simple-object-access-protocol) **Prova:**[CESPE / CEBRASPE - 2021 - SEFAZ-CE - Auditor Fiscal de Tecnologia da Informação da Receita Estadual](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-sefaz-ce-auditor-fiscal-de-tecnologia-da-informacao-da-receita-estadual)

Julgue o próximo item, relativo a arquitetura de *software* e à ITIL 4.

Suponha que alguém deseje criar um serviço na Web no formato cliente-servidor com base na arquitetura orientada a serviços, utilizando transferência de arquivos em formato XLT a serem entregues por meio de HTTP. Nesse caso, não será possível utilizar o RESTful para essa finalidade, porque ele é um protocolo desenvolvido a partir de SOAP e, portanto, não é compatível com XLT nem com a arquitetura cliente-servidor.

**✅Gabarito(Errado)**

**XLT**

Quando um cliente faz uma solicitação usando uma API RESTful, essa API transfere uma representação do estado do recurso ao solicitante ou endpoint. Essa informação (ou representação) é entregue via HTTP utilizando um dos vários formatos possíveis: Javascript Object Notation (JSON), HTML, XLT, Python, PHP ou texto sem formatação.

**Cliente/Servidor**

Tem uma arquitetura cliente/servidor formada por clientes, servidores e recursos, com solicitações gerenciadas por HTTP.

Fonte: <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>

**GABARITO: ERRADO**

Glossário para entender a questão:

➳ A arquitetura cliente servidor é uma arquitetura de aplicação distribuída, ou seja, na rede existem os fornecedores de recursos ou serviços a rede, que são chamados de servidores, e existem os requerentes dos recursos ou serviços, denominados clientes.

➳ Arquitetura orientada a serviços (SOA) é um tipo de design de software que torna os componentes reutilizáveis usando interfaces de serviços com uma linguagem de comunicação comum em uma rede.

➳ Um arquivo XLT é um modelo criado pelo Microsoft Excel, um aplicativo de planilhas incluído no conjunto do Microsoft Office. Ele contém formatação e dados padrão para uma planilha e é usado como base para criar novos arquivos .XLS . Os arquivos XLT são salvos no formato de arquivo binário do Excel, que era o formato principal até que o formato Office Open XML ( .XLTX file extension) o substituísse no Excel 2007 para Windows e 2008 para Mac.

➳ HTTP é um protocolo de transferência que possibilita que as pessoas que inserem a URL do seu site na Web possam ver os conteúdos e dados que nele existem. A sigla vem do inglês Hypertext Transfer Protocol. Esse sistema é a base da comunicação que existe em toda a Internet em que os sites e conteúdos que tragam hiperlinks possam ser encontrados mais facilmente pelo público por meio de um clique do mouse ou um toque na tela.

➳ RESTful: capacidade de determinado sistema aplicar os princípios de REST.

➳ REST significa Representational State Transfer. Em português, Transferência de Estado Representacional. Trata-se de uma abstração da arquitetura da Web. Resumidamente, o REST consiste em princípios/regras/constraints que, quando seguidas, permitem a criação de um projeto com interfaces bem definidas. Desta forma, permitindo, por exemplo, que aplicações se comuniquem.

➳ SOAP é um formato de mensagem XML usado nas interações de serviços da web. As mensagens SOAP são tipicamente enviadas através de HTTP ou JMS, mas outros protocolos de transporte podem ser utilizados. O uso de SOAP em um serviço da web específico é descrito por uma definição WSDL.

Fontes:

↬ <https://www.gta.ufrj.br/ensino/eel878/redes1-2016-1/16_1/p2p/modelo.html>

↬ <https://www.redhat.com/pt-br/topics/cloud-native-apps/what-is-service-oriented-architecture>

↬ <https://filememo.info/extension/xlt>

↬ <https://rockcontent.com/br/blog/http/>

↬ <https://becode.com.br/o-que-e-api-rest-e-restful/>

↬ <https://www.ibm.com/docs/pt-br/integration-bus/10.0?topic=services-what-is-soap>

**48** [**Q1831204**](https://www.qconcursos.com/questoes-de-concursos/questoes/2e2b761f-2f) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - SEFAZ-CE - Auditor Fiscal de Tecnologia da Informação da Receita Estadual](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-sefaz-ce-auditor-fiscal-de-tecnologia-da-informacao-da-receita-estadual)

Com relação à arquitetura de desenvolvimento de *software*, julgue o item a seguir.

Aplicando-se o padrão de projetos especialista da informação da abordagem GRASP no desenvolvimento de*software* orientado a objetos, ficará claramente definida de quem é a responsabilidade pela criação de nova instância de uma classe.

No GRASP, a responsabilidade pela criação de nova instância de uma classe é definida pelo creator (criador), não pelo especialista da informação.

"...ficará claramente definida de **QUEM** é a responsabilidade..."

Creio que o erro da questão está na palavra **QUEM**a qual induz o candidato a pensar em responsabilidade atribuída a **uma pessoa.**

O GRASP consiste em diretrizes para atribuir responsabilidade a CLASSES e OBJETOS **(não a pessoas),** em projeto orientado a objetos.

Portanto, questão errada.

Os padrões GRASP fornecem uma abordagem sistemática para a atribuição de responsabilidades às classes do projeto.

Para se definirem as responsabilidades de um objeto, leva-se em conta o que este elemento irá "fazer" e/ou "saber".

<https://www.devmedia.com.br/desenvolvimento-com-qualidade-com-grasp/28704>

Se a pessoa quiser saber **quem**é responsável, pode-se fazer isso pela **matriz RACI:**

* **R:** Responsible ou Responsável;
* **A:**Accountable ou Aprovador/Autoridade;
* **C:**Consulted ou Consultado;
* **I:** Informed ou Informado.

Caso se queira saber as **classes** e **objetos** responsáveis, como já muito bem explicado pelo colega *João Neto*, usa-se **GRASP**.

**GABARITO ERRADO**

**49** [**Q1814493**](https://www.qconcursos.com/questoes-de-concursos/questoes/6138d24e-16) [WebServices](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/webservices) **Prova:**[CESPE / CEBRASPE - 2021 - PG-DF - Analista Jurídico - Analista de Sistema - Desenvolvimento de Sistema](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-pg-df-analista-juridico-analista-de-sistema-desenvolvimento-de-sistema)

A respeito dos padrões XML, SOAP e WSDL, julgue o item que se segue.

WSDL é uma linguagem baseada em XML e utilizada para descrever *web services*.

A tecnologia**WEB Services**é fundamentada na linguagem XML, utilizada para a representação dos dados e como base para outros padrões. Os padrões fundamentais da tecnologia **WEB Services**são:

**1 - WSDL (WEB Services Definition Language)** → linguagem baseada em XML para definição formal de interfaces de serviços. Padrão para entender/documentar as operações de serviço.

Fonte: meus resumos

JUSTIFICATIVA: CERTO. Os formatos de mensagens usadas em web services são definidos mediante schema XML. Um web service é descrito pelo WSDL, que mapeia mensagens trocadas para operações agrupadas e descreve como essas operações podem ser chamadas por meio de protocolo particular de transporte.

**50** [**Q1814491**](https://www.qconcursos.com/questoes-de-concursos/questoes/613009c9-16) [SOAP (Simple Object Access Protocol)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/soap-simple-object-access-protocol) **Prova:**[CESPE / CEBRASPE - 2021 - PG-DF - Analista Jurídico - Analista de Sistema - Desenvolvimento de Sistema](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-pg-df-analista-juridico-analista-de-sistema-desenvolvimento-de-sistema)

A respeito dos padrões XML, SOAP e WSDL, julgue o item que se segue.

O protocolo SOAP tem como desvantagem pouca capacidade de expansão e de adequação ao ambiente de rede.

Tendo em vista que **SOA (Arquitetura Orientada a Serviços)**→ É um **paradigma**para organizar e usar competências distribuídas que podem estar sob controle de domínios diferentes. E o WEB Service é uma tecnologia que pode ser usada para implementar a SOA, que busca uma **INTEROPERABILIDADE**entre sistemas. Não faz sentido dizer que ele tem "pouca capacidade de expansão e de adequação ao ambiente de rede".

"pouca capacidade de expansão e adequação ao ambiente de rede" - Acertei pois levei em consideração que o SOAP pode utilizar tanto HTTP, FTP e outros protocolos de rede. Questões de T.I. da Cespe são muito chatas.

JUSTIFICATIVA: ERRADO. O protocolo SOAP se expande para baixo e para cima, tornando-se aderente aos padrões de protocolo de comunicação. A expansão para baixo está concretizada no fato de que esse protocolo é, por concepção, independente do protocolo de transporte. Podem-se enviar mensagens SOAP usando-se HTTP ou SMTP/POP, por exemplo.

**51** [**Q1814490**](https://www.qconcursos.com/questoes-de-concursos/questoes/612c01a1-16) [SOAP (Simple Object Access Protocol)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/soap-simple-object-access-protocol) **Prova:**[CESPE / CEBRASPE - 2021 - PG-DF - Analista Jurídico - Analista de Sistema - Desenvolvimento de Sistema](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-pg-df-analista-juridico-analista-de-sistema-desenvolvimento-de-sistema)

Acerca da arquitetura orientada a serviço e de*web services*, julgue o item a seguir.

O uso do protocolo SOAP em conjunto do *parsing* do XML, reduz custos na aquisição de memória e processadores para servidores, já que o formato deste protocolo prioriza desempenho da aplicação e pouco uso de processador.

O SOAP é mais pesado que o REST, por exemplo.

E só o fato de fazer o parsing do XML já requer algum tipo de processamento

protocolo SOAP emprega XML em vez do formato binário, o que torna o tamanho dos dados cerca de 400% maior que os dados binários. O parsing das informações XML contidas no envelope SOAP também requer esforço de processamento e consumo de tempo.

**Fonte: Tec Concursos.**

JUSTIFICATIVA: ERRADO. O protocolo SOAP emprega XML em vez do formato binário, o que torna o tamanho dos dados cerca de 400% maior que os dados binários. O parsing das informações XML contidas no envelope SOAP também requer esforço de processamento e consumo de tempo

**52** [**Q1814487**](https://www.qconcursos.com/questoes-de-concursos/questoes/611ead9c-16) [WebServices](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/webservices) **Prova:**[CESPE / CEBRASPE - 2021 - PG-DF - Analista Jurídico - Analista de Sistema - Desenvolvimento de Sistema](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-pg-df-analista-juridico-analista-de-sistema-desenvolvimento-de-sistema)

Acerca da **arquitetura orientada a serviço e de *web services***, julgue o item a seguir.

A principal característica de um *web service*é integrar sistemas heterogêneos.

Utilizando a tecnologia **Web Service**, uma aplicação pode invocar outra para efetuar tarefas simples ou complexas mesmo que as duas aplicações estejam em **diferentes sistemas e escritas em linguagens diferentes.** Por outras palavras, os Web Services fazem com que os seus recursos estejam **disponíveis**para que qualquer aplicação cliente possa operar e extrair os recursos fornecidos pelo WebService.

JUSTIFICATIVA: CERTO. Com o avanço da Internet e dos protocolos de comunicação baseados em XML, surgiram os web services, com a característica de integrar sistemas heterogêneos.

**53** [**Q1756818**](https://www.qconcursos.com/questoes-de-concursos/questoes/42e5fa41-d1) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - SERPRO - Analista - Especialização: Desenvolvimento de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-serpro-analista-especializacao-desenvolvimento-de-sistemas)

Considerando o padrão GRASP, julgue o item a seguir.

Atribuir responsabilidades para abstrações, e não para objetos, faz parte do padrão Expert.

ERRADO - Atribuir responsabilidades para abstrações, e não para objetos, faz parte do padrão Expert.

O CERTO é: **Information Expert** (Especialista na Informação) — Determina quando devemos delegar a responsabilidade para um outro objeto que seja especialista naquele domínio.

**Polymorphism**(Polimorfismo) — As responsabilidades devem ser atribuídas a **abstrações e não a objetos concretos**, permitindo que eles possam variar conforme a necessidade.

Fonte: <https://medium.com/@leandrovboas/padrões-grasp-padrões-de-atribuir-responsabilidades-1ae4351eb204>

**Especialista na Informação** (ou **Especialista da Informação**, ou apenas **Especialista**, ou ainda seus equivalentes em inglês ***Information Expert*** e ***Expert***) é um , parte do conjunto de princípios conhecido como  (*General Responsibility Assignment Software Patterns*). É uma abordagem genérica que visa atribuir a responsabilidade de fazer ou conhecer algo ao "especialista na informação" — a  que possui a informação necessária para cumprir tal responsabilidade.

**Fonte: Wikipedia.**

**54** [**Q1756817**](https://www.qconcursos.com/questoes-de-concursos/questoes/42e2e8d6-d1) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - SERPRO - Analista - Especialização: Desenvolvimento de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-serpro-analista-especializacao-desenvolvimento-de-sistemas)

Considerando o padrão GRASP, julgue o item a seguir.

Observa-se a utilização do padrão Controller quando uma classe recebe a responsabilidade de lidar com eventos do sistema.

**Controller**é um padrão do catálogo GRASP.

O padrão **controlador** atribui a responsabilidade de manipular eventos do sistema para uma classe que não seja de interface do usuário (UI) que representa o cenário global ou cenário de . Um objeto controlador é um objeto de interface não-usuário, responsável por receber ou manipular um evento do sistema.

Fonte: https://pt.wikipedia.org/wiki/GRASP\_(padr%C3%A3o\_orientado\_a\_objetos)

O padrão **controlador** atribui a responsabilidade de manipular eventos do sistema para uma classe que não seja de interface do usuário (UI) que representa o cenário global ou cenário de . Um objeto controlador é um objeto de interface não-usuário, responsável por receber ou manipular um evento do sistema.

Um caso de uso controlador deve ser usado para lidar com *todos* os eventos de casos de uso e pode ser usado para mais de um caso de uso (por exemplo, para casos de uso como *Criar usuário* e *Excluir usuário*, pode ter um único *UserController*, em vez de dois casos de uso *controllers* separados).

É definido como o primeiro objeto além da camada UI que recebe e coordena ("controla") operações do sistema. O controlador deve delegar o trabalho que precisa ser feito para outros objetos; ele coordena ou controla a atividade. Ele não deve fazer muito trabalho por si próprio. O *Controller* GRASP pode ser considerado uma parte da camada de aplicação/serviço  (assumindo que a aplicação tenha feito uma distinção explícita entre a camada de aplicativo/serviço e a camada de domínio em um sistema orientado a objetos com camadas comuns em uma arquitetura lógica do sistema de informações).

**55** [**Q1756816**](https://www.qconcursos.com/questoes-de-concursos/questoes/42df5123-d1) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - SERPRO - Analista - Especialização: Desenvolvimento de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-serpro-analista-especializacao-desenvolvimento-de-sistemas)

Com relação a padrões comportamentais, julgue o item seguinte.

O padrão Command permite suportar operações reversíveis e diferentes requisições parametrizadas pelo cliente.

Na programação orientada a objeto, o command é um padrão no qual um objeto é usado para encapsular toda informação necessária para executar uma ação ou acionar um evento em um momento posterior.

O Padrão Command tem como definição encapsular uma solicitação como um objeto, o que lhe permite parametrizar outros objetos com diferentes solicitações, enfileirar ou registrar solicitações e implementar recursos de cancelamento de operações. Isso inclui informações como o nome do método, o objeto que o método pertence e os valores dos parâmetros do método.

Fonte: Wikipedia.

O **Padrão Command** tem como intenção "Encapsular uma **requisição como um objeto**, permitindo que clientes parametrizem diferentes requisições, filas ou requisições de log, **e suportar operações reversíveis**".

Fonte: <https://diogomoreira.gitbook.io/padroes-de-projeto/padroes-gof-comportamentais/padrao-command>

Ao todo são nove os padrões GRASP, os quais serão discutidos em maiores detalhes nas próximas seções:

**Information Expert**(Especialista na Informação) — Determina quando devemos delegar a responsabilidade para um outro objeto que seja especialista naquele domínio.

**Creator** (Criador) — Determina qual classe deve ser responsável pela criação certos objetos.

**Controller**(Controlador) — Atribui a responsabilidade de lidar com os eventos do sistema para uma classe que representa a um cenário de caso de uso do sistema global.

**Low Coupling** (Baixo Acoplamento)— Determina que as classes não devem depender de objetos concretos e sim de abstrações, para permitir que haja mudanças sem impacto.

**High Cohesion**(Alta Coesão)— este princípio determina que as classes devem se focar apenas na sua responsabilidade.

**Polymorphism**(Polimorfismo) — As responsabilidades devem ser atribuídas a abstrações e não a objetos concretos, permitindo que eles possam variar conforme a necessidade.

**Pure Fabrication**(Pura Fabricação) — é uma classe que não representa nenhum conceito no domínio do problema, ela apenas funciona como uma classe prestadora de serviços, e é projetada para que possamos ter um baixo acoplamento e alta coesão no sistema.

**Indirection**(indireção)— este princípio ajuda a manter o baixo acoplamento, através de delegação de responsabilidades através de uma classe mediadora.

**Protected Variations** (Proteção contra Variações) **—**Protege o sistema com a variação de componentes, encapsulando o comportamento que realmente importa.

**56** [**Q1756815**](https://www.qconcursos.com/questoes-de-concursos/questoes/42dba736-d1) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - SERPRO - Analista - Especialização: Desenvolvimento de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-serpro-analista-especializacao-desenvolvimento-de-sistemas)

Com relação a padrões comportamentais, julgue o item seguinte.

O padrão Iterator oferece uma forma flexível de uso de herança para estender uma funcionalidade.

**Decorators:**oferecem uma alternativa flexível ao uso de herança para estender uma funcionalidade, com isso adiciona-se uma responsabilidade ao objeto e não à classe

**Iterator:** permite percorrer elementos de uma coleção sem expor sua representação (lista, pilha, árvore, etc.)

**✅Gabarito(Errado)**

Questão sobre Iterator da FCC

**Iterator** é um padrão comportamental que fornece uma maneira de acessar sequencialmente uma coleção de objetos sem expor a sua implementação.

Fonte: Q1316543

e-

**Iterator** - Fornecer uma maneira de acessar, sequencialmente, os elementos de uma agregação de objetos sem expor a sua representação subjacente.

O padrão Iterator oferece uma forma flexível de uso de herança para estender uma funcionalidade. errado

Decorators

O padrão **Iterator** permite o acesso sequencial aos elementos de um conjunto sem expor sua implementação subjacente. O padrão **Iterator** também é responsável por toda a tarefa de iteração, eliminando assim a responsabilidade adicional, simplificando, assim a sua aplicação e deixando a responsabilidade onde deveria estar.

**57** [**Q1756814**](https://www.qconcursos.com/questoes-de-concursos/questoes/42d886b1-d1) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - SERPRO - Analista - Especialização: Desenvolvimento de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-serpro-analista-especializacao-desenvolvimento-de-sistemas)

Acerca de padrões estruturais, julgue o item subsequente.

Um cliente que precisa tratar, de maneira uniforme, objetos individuais e suas composições deve utilizar, para essa finalidade, o padrão Facade.

"objetos individuais e suas composições" -> **composite**

✅Gabarito(Errado)

Segue uma questão da Fepese

**Facade**fornece uma interface unificada para um conjunto de objetos que constituem um subsistema. Define uma interface de mais alto nível que torna o subsistema mais fácil de usar.

Fonte: Q119997

2015

Para tratar de **maneira uniforme objetos individuais** em **estruturas de árvores**

que representem hierarquias partes-todo, o padrão composite é mais adequado que o decorator.

certa

Um cliente que precisa tratar, de maneira uniforme, objetos individuais e suas composições deve utilizar, para essa finalidade, o padrão Facade. errado

 (composite)

**58** [**Q1756813**](https://www.qconcursos.com/questoes-de-concursos/questoes/42d55cef-d1) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - SERPRO - Analista - Especialização: Desenvolvimento de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-serpro-analista-especializacao-desenvolvimento-de-sistemas)

Acerca de padrões estruturais, julgue o item subsequente.

O propósito do padrão Adapter é separar uma abstração de sua implementação, para que as duas possam variar e ser independentes.

**Bridge:** desacopla abstração de sua implementação de forma que ambas possam variar

**Adapter:** converte uma interface em outra

2011

**Desacoplar uma abstração de sua implementação** para que ambas possam variar independentemente é o padrão BRIDGE

certa

✅**Gabarito(Errado)**

Adapter converte a interface de uma classe em outra interface esperada pelos clientes permitindo que certas classes trabalhem em conjunto, pois de outra forma, seria impossível por causa de suas interfaces incompatíveis.

Fonte: Q502279

2017

Considere que é necessário fazer um programa que vá funcionar em várias plataformas, como Windows, Linux, Mac OS etc. O programa fará uso de diversas abstrações de janelas gráficas, por exemplo, janela de diálogo, janela de aviso, janela de erro etc. Um Analista sugeriu o uso de um padrão de projeto GoF estrutural que fornece um nível de abstração maior que o Adapter, pois são separadas as implementações e as abstrações, permitindo que cada uma varie independentemente.

O padrão sugerido pelo Analista é o

***a) Bridge e as implementações seriam as classes de janela das plataformas.***

b) Prototype, pois especifica os tipos de plataformas a serem criadas usando uma instância protótipo e cria novas interfaces pela cópia desse protótipo.

c) MVC, de forma que a camada Model fique responsável por acionar as janelas, a camada Controller exiba os dados do Model ao usuário e a camada View armazene as classes que fazem a comunicação com o Banco de Dados.

d) Visitor e as interfaces seriam executadas nos elementos de cada plataforma.

e) State, que permite a uma plataforma alterar o comportamento de sua interface.

O propósito do padrão **Adapter** é separar uma abstração de sua implementação, para que as duas possam variar e ser independentes. errado

**59** [**Q1756812**](https://www.qconcursos.com/questoes-de-concursos/questoes/42d264a5-d1) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - SERPRO - Analista - Especialização: Desenvolvimento de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-serpro-analista-especializacao-desenvolvimento-de-sistemas)

A respeito de padrões de criação, julgue o item subsecutivo.

O padrão Prototype cria novos objetos a partir da cópia de quaisquer objetos, sem aproveitar o estado do objeto copiado.

O Prototype copia todo o estado do objeto. Um exemplo prático é o método "clone()" do Java

**60** [**Q1756811**](https://www.qconcursos.com/questoes-de-concursos/questoes/42cf5a4b-d1) [Padrões de projeto (Design Patterns)](https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-arquitetura-de-software/padroes-de-projeto-design-patterns) **Prova:**[CESPE / CEBRASPE - 2021 - SERPRO - Analista - Especialização: Desenvolvimento de Sistemas](https://www.qconcursos.com/questoes-de-concursos/provas/cespe-cebraspe-2021-serpro-analista-especializacao-desenvolvimento-de-sistemas)

A respeito de padrões de criação, julgue o item subsecutivo.

Conforme o padrão Singleton, um sistema que necessite de um único objeto de uma classe, após o programa instanciar o objeto, não deve ter permissão de criar objetos adicionais dessa classe.

O construtor de uma classe Singleton é privado. Isso impossibilita que outras classes a instancie

**✅Gabarito(Certo)**

Outra questão da FCC

Padrão de criação, que busca garantir que um objeto terá apenas uma única instância, ou seja, uma classe irá gerar apenas um objeto e que este estará disponível de forma única para todo o escopo de uma aplicação.

Fonte: Q486731