BANCO DO ESTADO DO RIO GRANDE DO SUL S.A. (BANRISUL)
CONCURSO PÚBLICO PARA O PROVIMENTO DE VAGAS PARA O CARGO DE TÉCNICO EM TECNOLOGIA DA INFORMAÇÃO II

EDITAL Nº 7 – BANRISUL, DE 18 DE JUNHO DE 2025

O Presidente do Banco do Estado do Rio Grande do Sul (Banrisul) torna públicos os gabaritos oficiais preliminares do **modelo padrão** das **provas objetivas** e os **padrões preliminares de resposta** da **prova discursiva**, referentes ao concurso público para o provimento de vagas no cargo de Técnico em Tecnologia da Informação II.

Por questão de segurança, foram aplicadas provas diferenciadas quanto à ordem da numeração das questões. Ainda, por questão de segurança, não houve explicitação do tipo de prova designado para cada candidato. Assim, para a verificação preliminar das respostas de cada questão, o candidato poderá utilizar o **modelo padrão** das provas e os gabaritos padrões.

1 DOS GABARITOS OFICIAIS PRELIMINARES

1.1 Gabaritos oficiais preliminares do **modelo padrão** das provas objetivas disponível no endereço eletrônico http://www.cebraspe.org.br/concursos/banrisul_25.

1.1.1 CONHECIMENTOS BÁSICOS PARA TODAS AS ÁREAS

Questão	Gabarito
1	D
2	E
3	C
4	A
5	C
6	C
7	E
8	A
9	A
10	C
11	C
12	D
13	E
14	B
15	E
16	C
17	D
18	E
19	C
20	A
21	C
22	D
23	B
24	B
25	C
26	D

Questão	Gabarito
27	D
28	B
29	A
30	A

1.1.2 ÁREA 1: ADMINISTRAÇÃO DE BANCO DE DADOS

Questão	Gabarito
31	E
32	D
33	C
34	B
35	C
36	E
37	D
38	B
39	C
40	D
41	E
42	D
43	D
44	B
45	B
46	E
47	A
48	C
49	D
50	E
51	C
52	B
53	A
54	B
55	D
56	C
57	D
58	E
59	B
60	A
61	B
62	C
63	D
64	C
65	E
66	A
67	D

Questão	Gabarito
68	C
69	D
70	D
71	B
72	A
73	E
74	D
75	E
76	C
77	E
78	A
79	C
80	C

1.1.3 ÁREA 2: DESENVOLVIMENTO DE *SOFTWARE*

Questão	Gabarito
31	A
32	E
33	B
34	C
35	E
36	D
37	A
38	A
39	D
40	A
41	D
42	D
43	A
44	B
45	B
46	E
47	A
48	C
49	E
50	C
51	A
52	C
53	B
54	C
55	D
56	C
57	E
58	B

Questão	Gabarito
59	A
60	C
61	D
62	C
63	E
64	B
65	A
66	D
67	C
68	A
69	D
70	E
71	B
72	C
73	E
74	E
75	C
76	C
77	C
78	C
79	E
80	B

1.1.4 ÁREA 3: TECNOLOGIA: SEGURANÇA, INFRA E OPERAÇÃO

Questão	Gabarito
31	A
32	E
33	E
34	D
35	C
36	A
37	A
38	C
39	D
40	A
41	E
42	E
43	E
44	A
45	E
46	D
47	A
48	C
49	B

Questão	Gabarito
50	A
51	D
52	C
53	A
54	B
55	A
56	A
57	C
58	C
59	E
60	A
61	B
62	D
63	A
64	A
65	D
66	D
67	C
68	C
69	D
70	D
71	B
72	E
73	A
74	C
75	D
76	B
77	A
78	E
79	C
80	C

1.1.5 ÁREA 4: TRANSFORMAÇÃO DIGITAL E GESTÃO DE TI

Questão	Gabarito
31	B
32	D
33	B
34	B
35	B
36	D
37	A
38	B
39	B
40	C

Questão	Gabarito
41	E
42	D
43	C
44	D
45	E
46	D
47	D
48	C
49	C
50	B
51	E
52	A
53	B
54	C
55	A
56	E
57	D
58	C
59	E
60	A
61	A
62	B
63	C
64	B
65	B
66	C
67	D
68	A
69	D
70	B
71	D
72	B
73	C
74	E
75	A
76	C
77	C
78	D
79	A
80	B

2 DOS PADRÕES DE RESPOSTA DA PROVA DISCURSIVA

2.1 ÁREA 1: ADMINISTRAÇÃO DE BANCO DE DADOS

PADRÃO DE RESPOSTA

1 O(A) candidato(a) deverá definir DDL (*data definition language*), que é a linguagem de definição de dados composta pelos comandos responsáveis pela definição e manutenção de objetos, como tabelas, no banco de dados. Dos comandos de DDL a seguir, o(a) candidato(a) deverá citar dois: CREATE, DROP, ALTER, TRUNCATE e COMMENT.

2 O(A) candidato(a) deverá explicar que um *trigger* é um tipo de ação iniciada a partir de determinado evento, como uma condição, com o objetivo de monitorar o banco de dados. Uma situação na qual um *trigger* pode ser utilizado no setor bancário é a seguinte: o disparo de um *trigge*r quando um depósito acima de determinado valor definido for efetuado. Outra situação possível: o disparo de um *trigger* quando uma transferência de valor elevado for efetuada no horário noturno. Outras situações podem ser citadas, mas devem estar relacionadas ao setor bancário.

3 Um *deadlock* ocorre quando duas transações bloqueiam recursos necessários para outra transação, o que resulta em uma situação em que nenhuma transação pode continuar a ser executada. Uma *view* é uma tabela derivada de outras tabelas, que podem ser tabelas básicas ou visões previamente definidas. Uma *view* não necessariamente existe em forma física, sendo considerada uma tabela virtual, ao contrário das tabelas básicas, cujas tuplas sempre estão armazenadas fisicamente no banco de dados.

2.2 ÁREA 2: DESENVOLVIMENTO DE SOFTWARE

PADRÃO DE RESPOSTA

O(A) candidato(a) deverá apresentar corretamente os cinco princípios que compõem o acrônimo SOLID: **S** [***single responsibility principle* (SRP)**]: cada classe deve ter apenas uma responsabilidade, ou seja, uma única razão para mudar. Isso evita que uma classe acumule muitas funções distintas, tornando-a mais simples e coesa. **O** [***open/closed principle* (OCP)**]: as entidades de *software* (classes, módulos, funções) devem estar abertas para extensão, mas fechadas para modificação. Isso permite que novos comportamentos sejam adicionados sem alterar o código existente, o que reduz o risco de introdução de erros. **L** [***liskov substitution principle* (LSP)**]: os objetos de uma subclasse devem ser capazes de substituir os objetos da superclasse sem alterar o comportamento correto do sistema. Isso garante integridade e previsibilidade no uso de herança. **I** [***interface segregation principle* (ISP)**]: é preferível ter várias interfaces específicas e pequenas a ter uma interface única e grande. Isso evita que classes sejam obrigadas a implementar métodos que não fazem sentido para elas. **D** [***dependency inversion principle* (DIP)**]: os módulos de alto nível não devem depender de módulos de baixo nível; ambos devem depender de abstrações. Isso promove um sistema desacoplado, o que facilita manutenção, testes e evolução do código.

Dos exemplos a seguir, o(a) candidato(a) deverá apresentar dois exemplos de aplicação ou de violação, como os seguintes: i) violação do SRP – uma classe Relatorio, que possui métodos para gerar relatórios, persistir dados em banco e enviar *e-mails*. Isso fere o SRP, pois mistura responsabilidades de geração de dados, persistência e comunicação. A correção seria dividir essa classe em três, cada uma focada em uma única responsabilidade; ii) aplicação do OCP – um sistema de cálculo de imposto que possui uma interface ICalculoImposto e classes, como CalculoImpostoBrasil, CalculoImpostoEUA etc. Se surge um novo país, cria-se uma nova classe que implementa essa interface, sem alterar o código existente. Isso demonstra o princípio de estar fechado para modificação, mas aberto para extensão; iii) violação do LSP – uma classe Ave tem um método voar(). Uma subclasse Pinguim estende Ave, mas não pode voar. Isso viola LSP, pois não é possível substituir Ave por Pinguim sem quebrar o funcionamento.

A solução seria repensar a hierarquia, criando uma interface IVoador separada; iv) aplicação do ISP – em vez de ter uma interface IMultifuncional com métodos imprimir(), digitalizar(), fax(), o sistema define interfaces específicas: IImpressora, IDigitalizadora, IFax. Assim, uma classe ImpressoraBasica implementa apenas IImpressora; v) aplicação do DIP – uma classe ProcessadorDePagamento não depende diretamente de uma classe concreta, como ServicoPayPal, mas de uma interface IGatewayPagamento. Assim, é possível usar PayPal, Stripe ou outro serviço sem alterar a classe principal.

O(A) candidato(a) deverá demonstrar compreensão dos benefícios técnicos trazidos pelos princípios e deverá apresentar dois possíveis impactos, como os seguintes: i) alta manutenibilidade – o código fica mais organizado, com classes menores, mais simples e focadas, o que facilita entender e modificar; ii) melhor testabilidade – com baixo acoplamento e alto uso de abstrações, as classes podem ser facilmente testadas de forma isolada; iii) alta escalabilidade – é mais fácil adicionar novos recursos e comportamentos, pois o sistema é projetado para extensão sem alterar código existente; iv) redução de *bugs* – menos dependências ocultas e menos efeitos colaterais ao modificar uma parte do sistema; v) facilidade na refatoração e evolução – as mudanças são localizadas, sem impacto indesejado em outras partes do sistema.

2.3 ÁREA 3: TECNOLOGIA: SEGURANÇA, INFRA E OPERAÇÃO

PADRÃO DE RESPOSTA

A rede deve ser constituída de: pelo menos um equipamento *firewall* com três interfaces de rede; um *switch core*, com capacidade de roteamento (camada 3) e criação de listas de acesso (ACL) e pelo menos seis interfaces de rede; pelo menos cinco *switches* de acesso, para conexão dos computadores clientes, cada um com pelo menos 24 portas. Também pode ter um outro *switch* de acesso para conexão da rede de servidores, mas opcionalmente os serviços de rede DNS, DHCP e diretório (p. ex. LDAP) poderão rodar no equipamento *firewall*. A descrição do cabeamento e dos tipos de interfaces de rede, das velocidades e das portas de *switches* são opcionais, podendo ser: cabeamento estruturado categoria 6 ou 7; conexão entre *switches* e *firewall* por meio de fibra óptica; velocidades de conexão de 10 Gbps entre ativos de rede e servidores *firewall*; e portas de acesso dos *switches* de 1 Gbps.

O endereço de rede 192.168.23.0/24 deve ser subdividido em oito subredes, utilizando-se a máscara 255.255.255.224 (ou /27, na nomenclatura abreviada). Uma subrede (p. ex. 192.168.23.0/27) será usada para a rede de servidores "DMZ Interna" (se houver). A segunda (192.168.23.32/27) será para conexão entre *firewall* e *switch core*. Outras cinco subredes serão designadas uma para cada departamento: 192.168.23.64/27, 192.168.23.96/27, 192.168.23.128/27, 192.168.23.160/27, 192.168.23.192/27 (restando uma subrede possível: 192.168.23.224/27). Em cada uma dessas subredes, será possível alocar até 30 endereços IP, sendo um para a interface do *switch core* (*gateway* da subrede) e as demais para os equipamentos (p. ex. 192.168.23.1 até 192.168.23.30 na primeira subrede com endereço de rede 192.168.23.0 e endereço *broadcast* 192.168.23.31).

1 No *firewall* serão criados filtros na camada IP (camada 3) de forma que permita somente a entrada e a saída de tráfego autorizado: entrada de consultas DNS; saída de conexões HTTP/HTTPS para a Internet; tráfego de dados entre servidores e clientes; bloqueio de comunicação entre as subredes de clientes etc. No *switch core*, serão criadas listas de acesso (ACL), de forma que também proíba qualquer tráfego entre as subredes dos departamentos, mas que permita a comunicação dos clientes com os servidores e com o *firewall* e a Internet. Os

switches de acesso, bem como *firewall* e *switch core* estarão instalados em ambientes fisicamente seguros, com controle rígido de acesso. Neles não serão permitidas conexões de equipamentos *wireless* (roteadores Wi-Fi). As portas dos *switches* de acesso serão configuradas para aceitar apenas um endereço MAC, evitando-se, assim, que se conectem roteadores ou outros derivadores nos pontos de rede dos computadores clientes.

2.4 ÁREA 4: TRANSFORMAÇÃO DIGITAL E GESTÃO DE TI

PADRÃO DE RESPOSTA

a) O levantamento de requisitos é a base do desenvolvimento de *software*, especialmente no cenário da transformação digital e gestão de TI. Ele assegura que as soluções tecnológicas sejam perfeitamente alinhadas às necessidades estratégicas da organização. Sem requisitos claros, projetos podem desviar do objetivo, o que resulta em *softwares* ineficazes, custos elevados e prazos estourados. Sua importância reside em alinhamento estratégico, o que garante que a tecnologia contribua diretamente para os objetivos institucionais, como a otimização de serviços públicos, a redução de riscos e custos, o que minimiza retrabalhos e funcionalidades desnecessárias, e impacta positivamente a qualidade e o orçamento dos projetos; o aumento da qualidade e satisfação, pois um *software* bem especificado é mais útil, intuitivo e aceito pelos usuários, o que otimiza a eficiência e a inovação.

b) O(A) candidato(a) deverá descrever três técnicas de levantamento de requisitos no desenvolvimento de *software* com suas principais características e **c)** indicar uma vantagem e uma limitação de cada uma delas. A seguir, estão apresentados exemplos de possíveis técnicas.

- Técnica 1 (entrevistas): refere-se a conversas diretas com *stakeholders* para coletar necessidades e expectativas; permitem o levantamento de requisitos com bastante profundidade e flexibilidade na exploração de detalhes, porém, podem ser demoradas e podem resultar em requisitos muito subjetivos ou até conflitantes entre si.
- Técnica 2 (*workshops*): refere-se a reuniões colaborativas, facilitadas, com múltiplos *stakeholders* para definição e priorização; promovem consenso rápido, eficiência e resolução precoce de conflitos, porém a logística relacionada à realização desses *workshops* pode ser complexa e há risco de um participante dominar a discussão.
- Técnica 3 (análise de documentos e sistemas existentes): refere-se à revisão de documentos (manuais, relatórios) e inspeção de sistemas legados; fornece informações objetivas e uma base sólida do "estado atual", porém essa documentação pode estar desatualizada ou incompleta, e pode haver falta de contexto.

d) Contextos apropriados para cada técnica:

1. Entrevistas

Contexto apropriado: projetos com poucos *stakeholders*-chave (ex.: entender as necessidades de um diretor para um novo sistema de gestão) ou para aprofundar requisitos específicos em sistemas complexos.

Exemplo: entrevistar líderes de departamento para definir as funcionalidades essenciais de um novo sistema de controle de ponto digital.

2. *Workshops*

Contexto apropriado: projetos com muitos *stakeholders* e necessidade de consenso (ex.: alinhar diversas áreas para um portal de serviços unificado) ou em fases de redesenho de processos.

Exemplo: realizar um *workshop* para definir requisitos para a digitalização de processos de licitação, envolvendo equipes de compras, jurídica e financeira.

3. Análise de documentos e sistemas existentes

Contexto apropriado: modernização de sistemas legados (ex.: compreender regras de negócio antigas em um sistema financeiro) ou para ganhar conhecimento inicial em um novo domínio antes de interagir com usuários.

Exemplo: analisar manuais e relatórios atuais de um sistema de atendimento ao cidadão antes de propor uma plataforma.

FERNANDO GUERREIRO DE LEMOS
Presidente do Banco do Estado do Rio Grande do Sul (Banrisul)