

CONHECIMENTOS ESPECÍFICOS

Um sistema operacional moderno possui várias funções que o diferenciam de um sistema de *software* comum. Acerca das funções de um sistema operacional moderno, julgue os itens a seguir.

- 46 Um processo em um sistema operacional é considerado um programa em execução que necessita de um endereço de memória para armazenar o programa executável, os dados do programa e a sua pilha.
- 47 Chamadas de sistema — *system calls* — podem ser interpretadas como uma interface entre programas ou aplicações e o sistema operacional. O Assembly é um exemplo de linguagem de programação em que as chamadas de sistema podem ser codificadas.
- 48 Um sistema operacional monolítico permite fazer as chamadas de sistema por meio de chamadas ao *kernel*, permitindo ainda, por meio de parâmetros, o acesso a serviços em locais bem definidos, como por exemplo em registradores.
- 49 O *deadlock* ocorre em decorrência de uma série de situações combinadas. Na condição de direito de posse, um recurso previamente garantido a um processo pode ser retirado forçosamente pelo sistema operacional.
- 50 Relacionado à gerência de processos no sistema operacional, um novo processo pode ser criado por meio da chamada FORK. Os dados de um processo novo criado utilizando-se o FORK são idênticos ao do processo pai e permanecem idênticos durante a existência do processo pai.

Orientação a objetos possui uma série de características que a torna particular para o desenvolvimento de aplicações. Em relação aos princípios de programação orientada a objetos para o desenvolvimento de aplicativos, julgue os itens que se seguem.

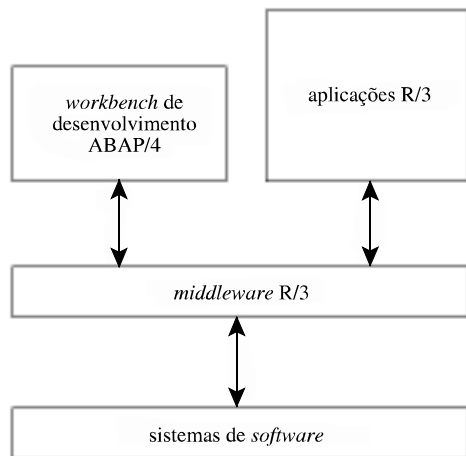
- 51 Em orientação a objetos, cada objeto possui um identificador único dependente dos valores dos atributos, que define o estado do objeto, distinguindo-o dos demais.
- 52 Um método, em orientação a objetos, representa possíveis comportamentos associados a um determinado objeto, representando ainda uma ação que pode ser executada pelo objeto ou sobre ele, de forma que a sua execução geralmente modifica o estado do objeto.
- 53 Em orientação a objetos, caracteriza a sobrecarga o uso de um mesmo nome para denotar operações diferentes, aplicadas a diferentes classes de objetos, em que cada operação pode apresentar, com relação às outras, significados diferentes para diferentes contextos.
- 54 A proteção dos atributos de um determinado objeto à corrupção ou alteração de seus valores por outro objeto é um dos benefícios do polimorfismo.
- 55 O tempo de vida de um objeto agregado depende do agregador, o que implica que, se o agregador for removido, o objeto agregado também o será.

Um sistema gerenciador de banco de dados coordena as requisições de usuários e programas a um banco de dados, de forma que os usuários não precisam se preocupar com a localização e a organização dos dados nem com o fato de outros usuários estarem acessando o banco de dados simultaneamente. Acerca de sistemas gerenciadores de banco de dados, julgue os itens seguintes.

- 56 Um banco de dados relacional pode ser definido como um conjunto de dados organizados em tabelas que possuem um número arbitrário de colunas e um número específico de linhas.
- 57 Um modelo de dados normalizado inserido em um banco de dados deve respeitar a primeira, a segunda e a terceira formas normais (1FN, 2FN e 3FN). A quarta e a quinta formas normais (4FN e 5FN) só são efetivamente utilizadas em banco de dados orientado a objetos, em que a coleção de classes é extremamente complexa.
- 58 Em bancos de dados relacionais, as restrições de integridade são regras relacionadas aos valores que podem ser armazenados em relações e que devem ser satisfeitas sempre que possível.
- 59 Em bancos de dados relacionais, uma linha em uma relação representa uma instância com valores definidos a partir de um conjunto de valores possíveis, que recebe o nome de tupla, que por sua vez é composta de valores designados como atributos.
- 60 Um sistema gerenciador de banco de dados deve permitir a redundância de dados e garantir o controle da integridade do banco de dados. Para que isso seja possível, o sistema deve ter conhecimento dessa redundância e assumir a responsabilidade de propagar as atualizações.

Com relação ao protocolo IP, suas características e funcionalidades em redes de computadores, julgue os seguintes itens.

- 61 O TCP/IP é de fato uma pilha de protocolos TCP/IP, da qual fazem parte os protocolos TCP, UDP, ICMP e *ethernet* de camada 2 do modelo OSI.
- 62 O esquema de endereçamento do TCP/IP, associado com a máscara de rede, permite definir a classe à qual o endereço pertence e a super-rede ou a sub-rede a que pertence o *host*.
- 63 Na camada de transporte do TCP/IP, encontram-se o UDP e o TCP. O TCP, por ser orientado a conexão, possui características mais avançadas, tais como retransmissão, janelamento e confirmação de dados, do que o UDP, que, apesar de ser orientado a conexão, é menos confiável.
- 64 O ARP e o RARP são protocolos responsáveis por fazer a conversão de um nome de computador em um endereço IP e a conversão de um endereço IP em um nome de computador, respectivamente.
- 65 O RIP e o IGP são protocolos de roteamento em redes de computadores e implementam algoritmos responsáveis por manter uma tabela de roteamento consistente. Esses algoritmos podem ser classificados em estado de enlace (*link state*) e vetor distância (*distance vector*).



A figura acima ilustra a arquitetura R/3. Acerca dessa arquitetura, julgue os itens que se seguem.

- 66** A camada de *middleware* propicia que as aplicações sejam dependentes das interfaces do sistema, principalmente aquelas relacionadas à arquitetura para bancos de dados R/3, o que garante um melhor tratamento para as transações de negócios.
- 67** A camada de *workbench* de desenvolvimento ABAP/4 é considerada o ambiente de programação para o desenvolvimento de soluções de negócios embasadas em sistemas cliente/servidor.
- 68** Uma arquitetura R/3 descentralizada não deve ser implementada em sistemas altamente complexos.
- 69** A arquitetura SAP R/3TM é considerada um pacote de *software* altamente configurável utilizado para a construção de aplicações nas áreas de ERP (*enterprise resource planning*).

A base de sistemas R/3 garante a integração de todos os módulos de aplicações, permitindo que vários módulos de programas possam ser distribuídos em diferentes computadores em uma rede, o que caracteriza a arquitetura dos serviços lógicos de sistemas R/3. Com relação à arquitetura dos sistemas R/3, julgue os itens subsequentes.

- 70** A lógica de negócios na arquitetura R/3 é executada na camada de aplicação, que depende do banco de dados residente na camada inferior.
- 71** Vários serviços são disponíveis na camada de aplicação SAP R/3, entre os quais o serviço de mensagem, que é responsável por controlar os serviços de comunicação entre sistemas R/3 e sistemas externos.
- 72** Na arquitetura de sistemas R/3, a camada de apresentação é responsável pela gerência da interface utilizada por usuários para realizar operações de acesso, manipulação e visualização de resultados.

Em ambientes R/3, o servidor de banco de dados pode ser considerado a camada mais importante. O R/3 utiliza o banco de dados como um depósito central dos dados e dos metadados. Acerca do banco de dados R/3, julgue os itens a seguir.

- 73** Em relação ao armazenamento físico das informações, a tecnologia SAP R/3 recomenda que a base de dados resida em ambientes que utilizem espelhamento de discos por meio de RAID 1 ou RAID 5.
- 74** Após a instalação de um banco de dados R/3 no SQL *server* da Microsoft, o banco de *logs* é capaz de registrar todas as alterações realizadas no banco de dados, o que permite a recuperação dos dados em caso de falha. O *tempdb* armazena todos os arquivos temporários e as *stored procedures*.
- 75** Quando utilizado com banco de dados Oracle, a ferramenta SAPDBA, em conjunto com outras ferramentas de *backup*, suporta atividades de administração de banco de dados. A integração com o sistema R/3 é garantida porque o SAPDBA preenche requerimentos específicos exigidos pelo R/3.
- 76** O *database utility* representa a interface entre dicionários ABAP e a base de dados relacional que sustenta o sistema R/3. Esse utilitário permite a criação, a conversão e a deleção de objetos do dicionário ABAP.
- 77** A função de travamento (*lock*) em bancos R/3 define um objeto no dicionário de informações que, quando ativo, gera duas funções: uma permite o travamento do objeto e a outra, a sua liberação.
- 78** Uma tabela de travamento (*lock*) é utilizada para gerenciar travamentos. Para cada pedido de *lock*, o sistema verifica se já existe um pedido na tabela de *lock*. Se existir, ele sobrescreve o *lock* antigo com o novo *lock* para evitar colisões de travamento.
- 79** A ferramenta CCMS (*computing center management system*) permite ao administrador monitorar, controlar e configurar um sistema R/3.

O SAP R/3 provê uma série de tecnologias para comunicação, processamento e segurança. Acerca das tecnologias SAP R/3, julgue os itens seguintes.

- 80** O SAP *communication server* é utilizado nos processos de envio e recebimento de informações. Essa tecnologia utiliza procedimentos do tipo *store and forward*.
- 81** O SAP *router* é um sistema da tecnologia SAP que permite realizar conexão indireta para auxiliar programas que não conseguem se comunicar por problemas de configuração de rede.
- 82** O sistema SAP *office* fornece aplicativos para edição de texto, *open database connectivity* (ODBC) e outros sistemas de auxílio à produção em escritório.
- 83** O sistema SAP R/3 permite a comunicação remota entre dois sistemas SAP independentes. Por outro lado, a comunicação entre cliente e servidor SAP é controlada pelo serviço de comunicação local SAP *local communication*.

- 84** O sistema de transporte em sistemas R/3 provê utilidades para controlar atualizações e fazer a movimentação de objetos de sistema entre sistemas R/3, permitindo ainda o *download* e a implementação de correções.
- 85** O módulo *secure network communications* em sistemas R/3 faz uso de algoritmos de criptografia proprietários da SAP R/3, garantindo uma maior segurança na comunicação entre sistemas R/3.
- 86** A interface de RFC (*remote function call*) permite chamada de funções entre sistemas R/3, sistemas SAP e sistemas externos. Por questões de compatibilidade, essa interface permite ainda chamada de funções entre sistemas R/3 e R/2.

Sistemas SAP R/3 são formados por uma série de módulos, tais como *basic components*, *AC financial* e *logistics*. Com relação a esses módulos e suas funções, julgue os seguintes itens.

- 87** O SAP *Session Manager* é uma ferramenta para *login* em sistemas SAP e para controle da sessão. Esse módulo permite o uso das funcionalidades de sistemas R/3 e gerencia as sessões em um ou mais sistemas em diferentes clientes SAP.
- 88** Sistemas SAP classificam o uso de memórias em dois grandes grupos: R/3 *extended memory* e *private memory*.
- 89** A interface R/3 *Accounting* constitui uma API (*application programming interface*) que coleta dados estatísticos relacionados às contas dos clientes no sistema R/3.
- 90** Para a execução de programas externos ao sistema, o R/3 usa módulos especiais ou permite que a execução seja realizada por meio de *menus* do CMMS (*computing center management system*).
- 91** Uma instância R/3 define um grupo de recursos, geralmente suportado dentro de um único servidor de aplicações ou de um banco de dados no ambiente cliente/servidor do R/3. Esse grupo de recursos pode constituir uma memória, um processo em uso, entre outros.
- 92** O sistema DMS (*document management system*) no R/3 engloba um conjunto de funções para gerenciar documentos que podem ser armazenados tanto internamente quanto externamente ao sistema R/3.
- 93** O RFC (*remote function call*) *interface generator* é uma biblioteca de funções que provê facilidades para gerar e baixar via rede programas para uma estação de trabalho ou um PC.
- 94** Um procedimento comum em sistemas SAP R/3, após a instalação do gerenciador de memória, é observar o consumo de memória paginada (*swap*) no sistema operacional, uma vez que, se a memória *swap* estiver mal configurada, o SAP R/3 pode ficar suscetível a falhas.
- 95** O serviço SAPconnect é implementado em sistemas R/3 para suportar serviços de telecomunicações, tais como *fax*, *paggers*, Internet e X400, e o envio de dados para impressoras e para outros sistemas R/3.

Um dicionário ABAP descreve e gerencia de forma centralizada todas as definições de dados usados em um sistema R/3. Acerca do dicionário do sistema ABAP, julgue os itens que se seguem.

- 96** O dicionário ABAP utiliza tipos de objetos em sua coleção. Entre esses objetos, encontram-se as tabelas e as *views*.
- 97** Apesar de o dicionário estar integrado com as funções de desenvolvimento do *workbench*, nem todos os componentes do *workbench* podem acessar diretamente as definições armazenadas no dicionário ABAP.
- 98** Informações novas ou modificadas são disponibilizadas em processo *batch* no dicionário ABAP para os componentes do sistema, o que garante a integridade da informação armazenada.
- 99** Tabelas são definidas no dicionário ABAP dependentemente da base de dados.
- 100** *Views* são visões lógicas de uma ou mais tabelas em SAP. A estrutura das *views* é definida na base de dados que é utilizada no dicionário ABAP.
- 101** Funções-módulo, que podem ser usadas por programas aplicativos, são geradas a partir das definições de funções de travamento (*lock*) de um objeto no dicionário ABAP.
- 102** Um domínio, parte integrante de um dicionário ABAP, define um *range* de valores válidos da informação para todos os campos que se referem ao domínio.

Acerca do ABAP *development workbench*, que é usado para a criação de aplicações em ambiente gráfico, julgue os itens subseqüentes.

- 103** Cada aplicação ABAP é considerada uma transação ou um relatório.
- 104** Uma transação em ABAP, após encontrar informações dos usuários, executa uma ou mais ações relevantes.
- 105** Relatórios em ABAP requerem alta interação com o usuário. Um relatório é um tipo de programa ABAP que gera uma saída quando executado.
- 106** Em desenvolvimento ABAP, uma classe de desenvolvimento é considerada um tipo de objeto de desenvolvimento. Em ABAP, cada uma dessas classes está associada a apenas um objeto.
- 107** A principal função do *workbench organizer* é a criação de repositório de classes no dicionário ABAP.
- 108** O *repository browser* provê um contexto que facilita o seguimento de relacionamento entre sistemas em produção ABAP *workbench*.
- 109** A função de análise de ambiente é utilizada para encontrar referências internas de um objeto.
- 110** O ABAP *workbench tools* é uma coleção de ferramentas composta, entre outras ferramentas, de um editor ABAP e um criador de funções.

Para encontrar erros em programas ABAP, é utilizado o ABAP *debugger*. Acerca da depuração de erros em programas ABAP, julgue os itens a seguir.

- 111 O ABAP *debugger* só pode ser ativado durante a execução de um programa em fase de depuração de dados.
- 112 O ABAP *debugger* é capaz de mostrar o conteúdo de uma tabela durante a depuração de um programa, porém a edição da tabela nesse modo de execução não é permitida.
- 113 Um ponto de parada (*breakpoint*) no ABAP *debugger* constitui uma instrução ao processador, em tempo de execução, para que este interrompa o programa para que seja iniciado o *debugger*.
- 114 Da mesma forma que os *breakpoints*, os *watchpoints* ativam o *debugger* quando o conteúdo de um campo é alterado.

Acerca dos modos de execução para a depuração de dados por meio do ABAP *debugger*, julgue os itens a seguir.

- 115 O modo *single step* permite navegar pelo programa passo a passo, em que cada passo é constituído por uma única instrução por vez.
- 116 No ABAP *debugger*, para executar um programa linha por linha, utiliza-se a função *jump line*.
- 117 A função *return* no ABAP *debugger* retorna um programa para a posição em que ele recupera o controle da execução.
- 118 A função *continue* no ABAP *debugger* realiza a execução de um programa até que ocorra um *breakpoint* dinâmico ou estático.

Em ABAP, as tabelas internas provêem um meio de retirar a informação de uma estrutura fixa e armazená-la na memória de trabalho. Com relação às tabelas internas no ABAP, julgue os seguintes itens.

- 119 As informações nas tabelas internas são armazenadas linha por linha, sendo que cada linha possui uma estrutura diferente da estrutura utilizada para a linha precedente.
- 120 Tabelas internas possuem a função de matrizes no ABAP.
- 121 Como as tabelas são consideradas objetos estáticos, elas desempenham um papel importante na gerência de execução dos programas.
- 122 O uso de tabelas internas é altamente desaconselhado caso o programador deseje processar um conjunto de dados com uma estrutura fixa dentro de um programa.
- 123 Para armazenar e formatar dados de um banco de dados em um programa, o uso das tabelas internas é considerado fundamental.
- 124 As tabelas internas podem existir tanto como tipos de dados como objetos de dados.
- 125 Para as tabelas internas, existem três tipos de chaves: a chave interna, a chave padrão e a chave estrangeira.

O ABAP possui uma série de módulos utilizados no desenvolvimento de aplicações e sistemas. Acerca dos módulos de função do ABAP, julgue os itens a seguir.

- 126 A função ABAP_DOCU_DOWNLOAD permite realizar o *download* de uma documentação ABAP no formato PDF.
- 127 A função APPL_LOG_INIT verifica a existência de um objeto específico e remove todas as informações associadas a ele da base de dados.
- 128 A função ARFC_GET_TID retorna o endereço IP de um terminal no formato hexadecimal.
- 129 A função APPL_LOG_DISPLAY permite analisar os *logs* no banco de dados.
- 130 A função BAL_CNTL_CREATE cria um controle para que um *log* seja mostrado.
- 131 A função BAL_DB_SEARCH procura por *logs* no banco de dados.
- 132 A função BAL_DSP_LOG_TECHNICAL_DATA gera a saída de todas as informações no cabeçalho do *log*.
- 133 A função BAL_DSP_MSG_LONGTEXT mostra mensagens de texto longo.

Em programas ABAP, os diálogos ABAP são utilizados para a comunicação com usuários. Eles permitem ainda explicar como passar dados entre diálogos de usuários e programas ABAP. Acerca de diálogos ABAP, julgue os itens subsequentes.

- 134 As telas são consideradas o tipo mais comum de diálogo que pode ser utilizado em programas ABAP e são definidas diretamente dentro dos programas ABAP.
- 135 Campos de texto são considerados elementos de diálogo que podem ser alterados pelos usuários ou por um programa ABAP.
- 136 A criação de telas que serão utilizadas exclusivamente para a entrada de dados pode ser realizada por meio das ferramentas normais de diálogo.
- 137 Para a chamada de uma tela, pode ser utilizada a função CALL SELECTION-SCREEN, e deve ter o número 1.000 dentro do sistema SAP R/3.
- 138 Ao se executar a instrução ABAP para a criação de uma lista, é criada a referida lista no servidor de banco de dados, onde ela é armazenada no *buffer*.

Telas de seleção constituem um dos três tipos de telas em sistemas R/3. Elas são utilizadas quando se deseja entrar um valor único para um campo ou um conjunto de campos ou permitir um critério de seleção. Com relação às telas de seleção em sistemas ABAP, julgue os itens seguintes.

- 139 A ABAP permite a criação de quantas telas de seleção-padrão se deseja, mas um número limitado por programa de telas de seleção.
- 140 Uma tela de seleção-padrão para um programa executável só pode ser conectada a uma base de dados lógica.

- 141** Em telas de seleção padrão, as seleções dinâmicas evitam que usuários sejam confundidos durante a conexão a sistemas fora do ambiente SAP R/3.
- 142** Uma base de dados lógica usa suas próprias seleções para permitir um número irrestrito de seleções a serem lidas na base de dados.
- 143** A possibilidade de seleções dinâmicas deve ser codificada no programa da base de dados lógica. A ferramenta ABAP *workbench* define quais campos o usuário pode definir como seleções dinâmicas.
- 144** Durante o processamento de telas de seleção, o ambiente de tempo de execução do ABAP gera eventos especiais para as telas de seleção.
- 145** A forma básica de um evento de uma tela de seleção é o evento AT SELECTION-SCREEN.

Programas ABAP contêm uma grande variedade de funções divididas em diversas categorias, em uma interface de usuário. Acerca das interfaces de usuário para programas ABAP, julgue os itens que se seguem.

- 146** Uma instância de uma interface de usuário consistindo em uma barra de *menu*, uma barra de ferramentas-padrão e uma barra de ferramentas da aplicação é chamada de GUI *status*. Nessa situação, a configuração de uma chave de função faz parte do *application server*.
- 147** Uma GUI *status* é definida como o principal objeto de uma interface de usuário.
- 148** Para uma determinada aplicação, o título para uma GUI pode ser definido pelo usuário. O título em uma GUI auxilia o usuário na manipulação simultânea de várias telas.
- 149** Em um programa ABAP, para criar um título em uma GUI, é correto o uso do seguinte comando: SET TITLEBAR <NOME>.
- 150** Em tempo de execução, o título de uma GUI pode ser modificado usando a seguinte instrução: ABAP SET TITLEBAR <NOME> INCLUDING <VAR1> <VAR2>...<VARN>.

Cabe ao administrador de sistemas a criação de autorizações que são designadas aos usuários individuais, organizados na forma de grupos de usuários. Acerca da autorização de usuários em ABAP, julgue os itens a seguir.

- 151** Uma forma possível para que um administrador de sistemas R/3 mantenha a autorização de usuários é utilizar o *user authorization generator*.
- 152** Uma checagem de autorização pode ser iniciada utilizando-se a função AUTHORITY-CHECK. O nível de autorização que um usuário pode ter é determinado pelo servidor de banco de dados por meio de objetos de autorização nas tabelas de sistema.
- 153** Durante a execução de uma transação, um grande número de objetos de autorização é chamado, porém esse procedimento é realizado uma única vez pelo AUTHORITY-CHECK.

154 Por meio da transação AUTH_SWITCH_OBJECTS, o sistema é capaz de executar todas as checagens de autorização.

155 Sempre que necessário, o administrador pode modificar a autorização em sistemas em produção usando o ABAP *workbench development authorizations*. Esse é um procedimento recomendado em sistemas R/3.

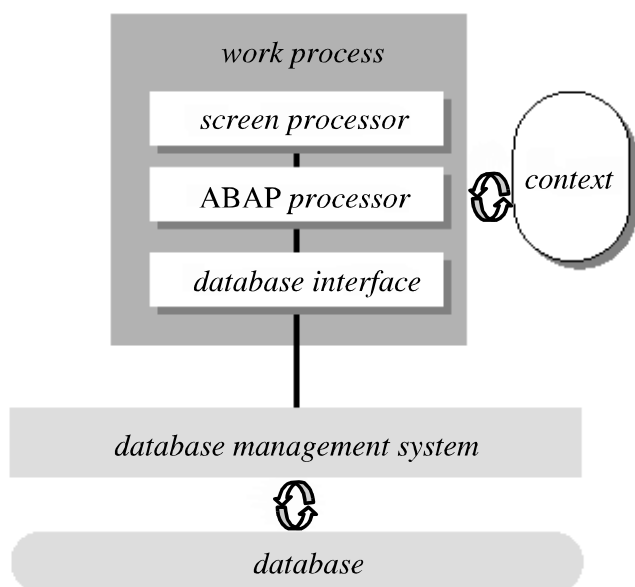
156 Para que uma checagem de autorização possa ser executada em programas ABAP, ela deve estar inserida nas tabelas de autorização no banco de dados.

157 Antes de gerar um perfil de autorização, o sistema SAP R/3 verifica se o usuário tem poder administrativo para executar a referida função.

158 A alteração de senhas de usuários é um procedimento realizado apenas pelo administrador de perfis e leva o tempo necessário para a atualização da base de senhas que é feita em processamento batch.

A transação de uma chamada de função remota (*transactional RFC* — tRFC) é uma transação definida pelo destino alvo, pelo nome da função e pelos valores correntes de um parâmetro. Acerca de tRFC, julgue os itens a seguir.

- 159** Um componente tRFC, quando executado, carrega as funções chamadas pela RFC em conjunto com as informações correspondentes na base de dados R/3, incluindo o identificador da transação (TID).
- 160** Requisições tRFC são transferidas, na forma de *byte-stream*, sem o parâmetro da informação, usando TCP/IP ou X400.
- 161** Quando em chamadas remotas síncronas, os parâmetros de exportação e das tabelas precisam ser especificados para as tRFC.
- 162** Se parâmetros do tipo EXPORT estão especificados nas interfaces de funções, é permitida ao usuário a chamada de funções remotas com o mecanismo transacional.
- 163** Quando um sistema remoto está indisponível, o sistema SAP agenda o relatório RSARFCSE para processamento em memória principal, com o TID (TRANSACTION ID) como valor variável.
- 164** Quando um sistema está indisponível, um relatório a ser processado gera execução de chamadas assíncronas repetidamente até que ele consiga a conexão com o sistema desejado.
- 165** Funções de chamada remota assíncrona (aRFC) são consideradas similares a uma tRFC porque em ambos os casos o usuário não precisa esperar pelo término da execução.
- 166** Existem cinco características distintas que são utilizadas para distinguir uma aRFC de uma tRFC.



A figura acima ilustra os componentes de um processo de trabalho em programação ABAP. Acerca desses componentes, julgue os seguintes itens.

- 167** O *screen processor* executa a lógica de fluxo de uma tela. A base de sistemas R/3 possui uma linguagem especial para a programação de lógicas de fluxo de telas.
- 168** O *ABAP processor* executa a lógica de processamento de um programa aplicativo.

169 Considerando a camada de *database interface*, as funções da DDL (*data definition language*) e da DCL (*data control language*), partes integrantes do padrão SQL (*structured query language*), são executadas pelo ABAP *dictionary* em sistemas R/3.

170 O controle de transações é função da camada de *database interface*.

171 O acesso ao ABAP *dictionary* é função da camada de *database interface*.

172 A administração dos *buffers* de tabelas no servidor de aplicação é função da camada de *database management system*.

173 O acesso ao repositório de objetos em sistemas R/3 é função da camada de *database*.

Acerca da estruturação e da execução de um programa ABAP, julgue os itens a seguir.

174 Programas ABAP são responsáveis pelo processamento de informações dentro de passos individuais de diálogos de um programa aplicativo.

175 A segunda parte de um programa ABAP é a declaração das informações globais.

