Desenbahia	DESENBAHIA - AGÊNCIA DE FOMENTO DO ESTADO DA BAHIA S/A
	EDITAL Nº 01/2014
	NÍVEL SUPERIOR
	ANALISTA DE SISTEMAS

NOME DO CANDIDATO	ASSINATURA DO CANDIDATO	RG DO CANDIDATO

LEIA COM ATENÇÃO AS INSTRUÇÕES ABAIXO

INSTRUÇÕES GERAIS

I. Nesta prova, você encontrará 05 (cinco) páginas numeradas sequencialmente, contendo 40 (quarenta) questões correspondentes às seguintes disciplinas: Língua Portuguesa (10 questões), Conhecimentos Gerais (10 questões), Conhecimentos Específicos (20 questões) e 02 (duas) questões dissertativas.

II. Verifique se seu nome e número de inscrição estão corretos no cartão de respostas. Se houver erro, notifique o fiscal.

III. Assine e preencha o cartão de respostas nos locais indicados, com caneta azul ou preta.

IV. Verifique se a impressão, a paginação e a numeração das questões estão corretas. Caso observe qualquer erro, notifique o fiscal.

V. A resposta para as questões dissertativas deverá ter no máximo 15 linhas.

VI. Você dispõe de 4 (quatro) horas para fazer esta prova. Reserve os 20 (vinte) minutos finais para marcar o cartão de respostas.

VII. O candidato só poderá retirar-se do setor de prova 2 (duas) horas após seu início.

VIII. O candidato não poderá levar o caderno de questões. O caderno de questões será publicado no site da espp, no prazo recursal contra gabarito.

IX. Marque o cartão de respostas cobrindo fortemente o espaço correspondente à letra a ser assinalada, conforme o exemplo no próprio cartão de respostas.

X. A leitora óptica não registrará as respostas em que houver falta de nitidez e/ou marcação de mais de uma alternativa.

XI. O cartão de respostas não pode ser dobrado, amassado, rasurado ou manchado. Exceto sua assinatura, nada deve ser escrito ou registrado fora dos locais destinados às respostas.

XII. Ao terminar a prova, entregue ao fiscal o cartão de respostas e este caderno. As observações ou marcações registradas no caderno não serão levadas em consideração.

XIII. É terminantemente proibido o uso de telefone celular, pager ou similares.

Boa Prova!

ESPP

DESTAQUE AQUI

Desenbahia

GABARITO DO CANDIDATO - RASCUNHO

ESPP

Nome:	Assinatura do Candidato:	Inscrição:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

RASCUNHO

QUESTÕES DISSERTATIVAS

Transcreva a resposta da questão dissertativa em até 15 linhas, com caneta azul ou preta, para o cartão de respostas.

1) A Engenharia de Requisitos é um dos tópicos importantes na Engenharia de Software. Descreva suas principais características.

A engenharia de requisitos (no contexto da engenharia de software) é um processo que engloba todas as atividades que contribuem para a produção de um documento de requisitos e sua manutenção ao longo do tempo. O processo de engenharia de requisitos é composto por quatro atividades de alto nível: identificação; análise e negociação; especificação e documentação; validação. Este processo deve ser precedido de estudos de viabilidade que, a partir das restrições do projeto, determinam se este é ou não viável e se deve prosseguir para a identificação dos requisitos. Uma outra atividade que se pode considerar que faz também parte deste processo, se incluirmos a fase posterior à produção do documento (isto é, a sua "manutenção"), é a gestão dos requisitos (change management, sendo que as alterações podem ser causadas pelos mais diversos fatores desde inovações tecnológicas a mudanças na natureza do negócio (e consequentemente nos requisitos), entre outras.

2) Quanto a Programação Estruturada *versus* Programação Orientada a Objetos faça um comparativo entre esses dois tipos de programação.

A Análise Estruturada foi popularizada por Tom DeMarco, surgindo então, um importante paradigma de desenvolvimento de sistemas. Passamos a conviver com Diagramas de Fluxos de Dados (DFD), Diagramas de Entidades e Relacionamentos (DER), entre outros. Muito se conseguiu melhorar no processo de desenvolvimento com esta metodologia, fazendo com que o desenvolvimento de sistemas fosse enxergado com mais esmero, respeitando sua complexidade. Tudo isso levou consecutivamente a produtos de melhor qualidade. Todavia, existem problemas nessa metodologia. Há uma dificuldade em garantir compatibilidade entre as fases de análise e projeto e, posteriormente, do projeto para a implementação.

Não podemos esquecer que a comunicação entre desenvolvedores e usuários é difícil, em virtude dos diagramas não serem muito expressivos fora da compreensão da equipe de desenvolvimento. Validações dos usuários até ocorrem, mas no geral não refletem o universo dos requisitos do sistema. Até os desenvolvedores mais resistentes à mudança de paradigma reconhecem que não é fácil conseguir uma validação precisa de um usuário, a partir das ferramentas usadas na análise estruturada. Mas, atualmente, a complexidade, a urgência e a adaptabilidade dos novos aplicativos têm levado a se repensar os prós e contras dessa abordagem.

A partir de meados da década de 70, métodos orientados a objetos começaram a surgir, vislumbrando a mudança de paradigma. O principal ganho com esse novo paradigma é o fato de que se uniformizaram os modelos usados para análise, projeto e implementação. Os principais diagramas são usados em todas as fases, mudando-se apenas sua visão. Partindo do conceito de orientação de objetos, há a unificação da perspectiva funcional e de dados. Não podemos esquecer o melhor dos mundos: a comunicação entre desenvolvedores e usuários tornou-se mais facilitada.

As linguagens tradicionais possuem o foco nos programas, responsáveis pela parte ativa da aplicação. Os dados são manuseados de forma passiva, perdendo, em alguns casos, sua importância de contexto. Em contrapartida, nas linguagens orientadas a objeto, os dados estão protegidos por uma cápsula, na qual residem procedimentos que intermediam o acesso a eles. A Orientação a Objetos veio propiciar à comunidade de desenvolvedores: aumento de produtividade, diminuição do custo de desenvolvimento e principalmente de manutenção, maior portabilidade e reutilização de código.