

AVALIAÇÃO DE CANDIDATO À VAGA DE PROGRAMADOR EM DELPHI

Nome:	GIOVANNI BONGIORNI LANER
E-Mail:	giovanni.laner@mv.com.br
Fone de Contato:	51 998099296

PERFIL DO CANDIDATO - REPOSTAS PROVA

LINK GITHUB: <https://github.com/GiovanniLaner/provaHDDDev.git>

1. Quais as linguagens de programação que você trabalha/Estuda? Qual seu grau de intimidade com as mesmas?

Java – Intermediário; estudei durante o curso técnico, onde criei uma aplicação web com java no backend, vue.js no front e banco de dados MySQL.

Python – Iniciante; estudei na cadeira de fundamentos da programação, onde tive como projeto final um sistema de compra de passagens aéreas, somente backend.

JavaScript – Intermediário; estudei durante o curso técnico, onde desenvolvi algumas telas web para exercitar.

2. Quais bancos de dados você já utilizou/estudou? Já trabalhei com Oracle PL/SQL durante estágio na Sulwork. Utilizei MySQL em atividades do curso técnico. E atualmente utilizo PostgreSQL.

3. Tem experiência em criação de triggers e functions no banco de dados? Não possuo uma vasta experiência, somente experiência de treino durante o curso de PostgreSQL que estou cursando pela Udemy.

4. Você já trabalhou como desenvolvedor? Quais tarefas você desempenhou? Ainda não trabalhei como desenvolvedor.

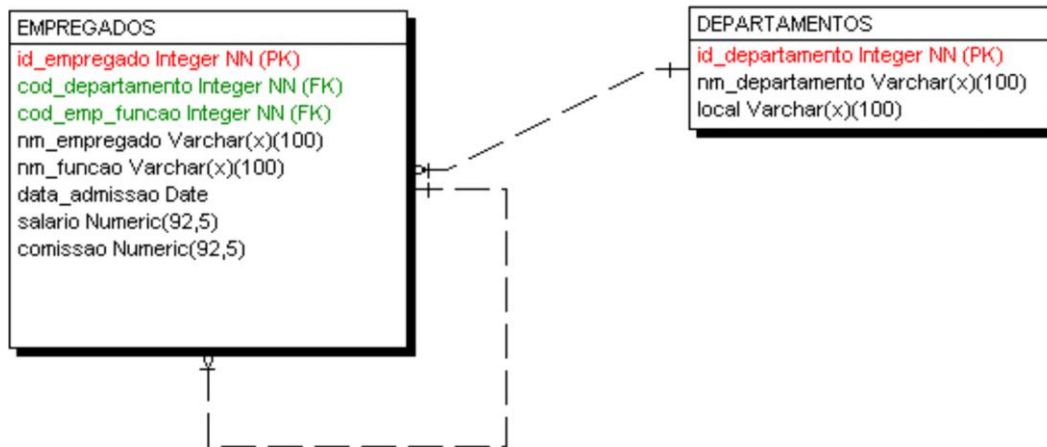
5. Tem experiência com algum componente de terceiro no Delphi? Por exemplo, DevExpress, ACBR? Não possuo essa experiência.

6. Quais geradores de relatórios já utilizou e qual o tempo de experiência com cada um? Atualmente utilizo GeradorJira há 1 ano e 4 meses.

7. Possui algum conhecimento em desenvolvimento/consumo de webservices? Se sim, em quais linguagens? Posso conhecimento prévio em consumo de API RESTful e API REST, XML e XML SOAP.

8. Utilizou alguma ferramenta para versionamento de fontes? Quais elas?
Utilizei somente git para versionamento de código. Utilizei para estudos.

SQL/Delphi



8. Para a realização desta questão, será utilizado compilador Delphi e o Banco de Dados PostgreSQL.

- Crie no banco de dados, as tabelas de Empregados e Departamentos (ref. Imagem acima);

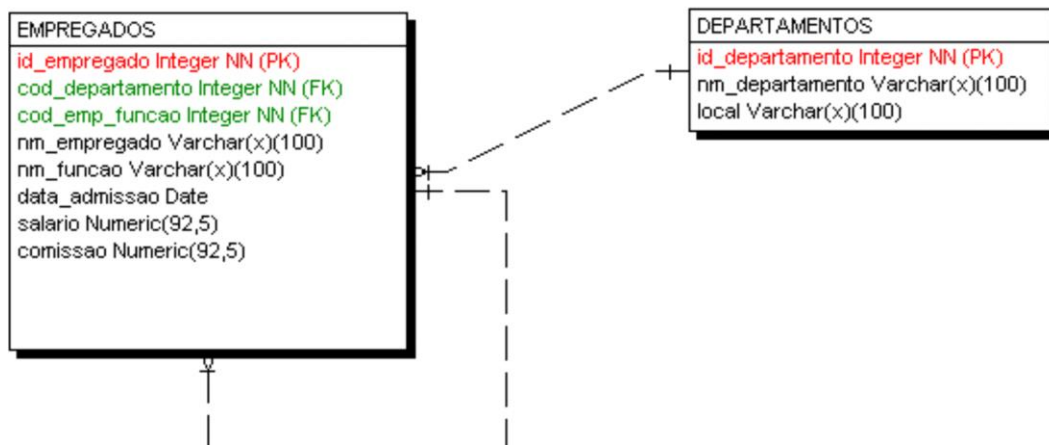
```
CREATE TABLE empregados (  
    id_empregado SERIAL NOT NULL,  
    cod_departamento int NOT NULL,  
    cod_emp_funcao int NOT NULL,  
    nm_empregado varchar(100),  
    nm_funcao varchar(100),  
    data_admissao date,  
    salario decimal,  
    comissao decimal,  
    CONSTRAINT pk_empregado PRIMARY KEY (id_empregado),  
    CONSTRAINT fk_cod_departamento FOREIGN KEY (cod_departamento)  
REFERENCES departamentos (id_departamento),
```

```

        CONSTRAINT fk_cod_emp_funcao FOREIGN KEY (cod_emp_funcao)
REFERENCES empregado_funcao (id_funcao)
);
CREATE TABLE departamentos (
        id_departamento SERIAL NOT NULL,
        nm_departamento varchar(100),
        local varchar(100),
        CONSTRAINT id_departamento PRIMARY KEY
(id_departamento)
);

```

- Popule as tabelas, respeitando o relacionamento, com no mínimo 5 registros;
INSERT INTO empregados (cod_departamento, cod_emp_funcao, nm_empregado, nm_funcao, data_admissao, salario, comissao) VALUES ('1','2','GIOVANNI', 'vendedor' , '2022/01/01', 2500, 50);
INSERT INTO empregados (cod_departamento, cod_emp_funcao,nm_empregado, nm_funcao, data_admissao, salario, comissao) VALUES ('1','1','NICOLAS', 'atendente' , '2022/01/01', 1500, 0);
INSERT INTO empregados (cod_departamento, cod_emp_funcao,nm_empregado, nm_funcao, data_admissao, salario, comissao) VALUES ('1','2','ANDRE', 'vendedor' , '2022/01/01', 2500, 100);
INSERT INTO empregados (cod_departamento, cod_emp_funcao,nm_empregado, nm_funcao, data_admissao, salario, comissao) VALUES ('1','2','MATHEUS', 'vendedor' , '2022/01/01', 2500, 150);
INSERT INTO empregados (cod_departamento, cod_emp_funcao,nm_empregado, nm_funcao, data_admissao, salario, comissao) VALUES ('1','1','STEPHANIE', 'atendente' , '2022/01/01', 1500, 0);
- Crie um Projeto, no Delphi, contendo as telas de cadastro das tabelas criadas;
Projeto criado com o nome de ProjetoTelaCadastro.
- Crie um Relatório para mostrar os dados das referidas tabelas (Listagem de Empregados com Departamento);
- Guarde todos os scripts utilizados para a criação e para popular as tabelas;
- Para acesso aos dados utilize a suite de componentes ZEOS, caso não possua, pode utilizar Firedac;
Tentei utilizar o ZEOS, porém tive alguns contratempos com a configuração dele, optei pelo FireDac devido ao curto prazo que foi dado.
- Para manipulação de dados, pode se utilizar dos componentes da paleta “Data Controls”;
- O layout das telas, padrões de nomenclatura, organização do código serão avaliados;
Foi mantido padrão nas nomenclaturas, identificação do código e no layout da tentei manter o mais agradável possível para a visão do usuário, cuidando a poluição visual.



SQL

Serão utilizadas as tabelas acima para responder as demais questões.

9. Escreva uma consulta para exibir o nome do empregado, a função e a data de admissão. Ordene pelo maior salário.

```
SELECT nm_empregado,  
       nm_funcao,  
       data_admissao  
FROM   empregados  
ORDER BY  
       salario DESC
```

10. Escreva uma consulta para exibir o nome do empregado e a data de admissão para todos os empregados que estão no mesmo departamento do empregado “Marcelo”, excluindo-o do resultado.

```
SELECT nm_empregado,
       data_admissao
FROM   empregados
WHERE  cod_departamento = (SELECT cod_departamento
                           FROM   empregados
                           WHERE  nm_empregado ILIKE '%Marcelo%')
AND NOT
      id_empregado = (SELECT id_empregado
                     FROM   empregados
                     WHERE  nm_empregado ILIKE
'%Marcelo%')
```

11. Escreva uma consulta que mostre o número do empregado e o nome para todos os empregados que trabalham em um departamento com qualquer empregado cujo nome contenha uma letra T.

```
SELECT id_empregado AS numero_empregado,
       nm_empregado
FROM   empregados
WHERE  nm_empregado ILIKE '%T%'
```

12. Escreva uma consulta que mostre o nome do empregado, o departamento, a localização e a função de todos os empregados admitidos há mais de 5 anos, cujo salário ultrapasse o menor salário entre os empregados de função “Gerente”.

```
SELECT e.nm_empregado,
       d.nm_departamento,
       d.local,
       e.nm_funcao
FROM   empregados AS e
INNER JOIN
       departamentos AS d ON e.cod_departamento = id_departamento
WHERE  (DATE_PART('Year', CURRENT_DATE: : date) - DATE_PART('Year', e.data_admissao: :
date)) >= 5
AND    e.nm_funcao ILIKE '%GERENTE%'
AND    e.salario > (SELECT min(salario)
                   FROM   empregados
                   WHERE  nm_funcao ILIKE '%GERENTE%')
```

13. Escreva uma consulta que mostre a soma dos salários, o departamento e o cargo de todos os empregados de admitidos há mais de 1 ano, localizados em “Porto Alegre”.

```
SELECT SUM(e.salarios) AS soma_salarios,
       d.nm_departamento,
       e.nm_funcao
FROM   empregados as e
INNER JOIN
       departamentos as d on e.cod_departamento = id_departamento
WHERE  d.local ILIKE '%Porto Alegre%'
AND    (DATE_PART('Year', CURRENT_DATE: : date) - DATE_PART('Year', e.data_admissao: :
date)) >= 1
```

14. Qual o modo correto utilizado para relacionar a tabela de empregados com a tabela de departamentos?

Modo correto para relacionar tabelas é utilizar as cláusulas JOIN: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN e CROSS JOIN dependendo de qual for o intuito do relacionamento. Para o caso da prova foi INNER JOIN, onde realiza a junção das informações interessadas que existem nas tabelas utilizadas, fazendo uso das fk e pk que forem necessárias para o relacionamento.