PROGRAMAÇÃO PARA INTERNET

Print dos Códigos da API

```java
package com.example.projetopi.modelo;

import java.sql.Date;

public class Bem {

    private int id;
    private String tipoBem;
    private Date dataAquisicao;
    private double valorAquisicao;
    private String situacaoBem;
    private double vidaUtil;
    private int turnosTrabalho;
    private Date dataVenda;
    private double valorvenda;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getTipoBem() {
        return tipoBem;
    }
    public void setTipoBem(String tipoBem) {
        this.tipoBem = tipoBem;
    }
    public Date getDataAquisicao() {
        return dataAquisicao;
    }
    public void setDataAquisicao(Date dataAquisecao) {
        this.dataAquisicao = dataAquisecao;
    }
    public double getValorAquisicao() {
        return valorAquisicao;
    }
    public void setValorAquisicao(double valorAquisicao) {
        this.valorAquisicao = valorAquisicao;
    }
    public String getSituacaoBem() {
        return situacaoBem;
    }
    public void setSituacaoBem(String situacaoBem) {
        this.situacaoBem = situacaoBem;
    }
    public double getVidaUtil() {
        return vidaUtil;
    }
    public void setVidaUtil(double vidaUtil) {
        this.vidaUtil = vidaUtil;
    }
    public int getTurnosTrabalho() {
```

```java
package com.example.projetopi.dao;

import java.sql.Connection;



public class BemDao {

    public static boolean inserir(Bem bem) {
        try {
            String sql = "insert into depreciacao (tipo_bem, dataaquisicao, valoraquisicao, situacaobem, vidautil, turnostrabalho, valorvenda,

            Connection conn = ConnectionFactory.getConnection();
            PreparedStatement pstm = conn.prepareStatement(sql);
            pstm.setString(1, bem.getTipoBem());
            pstm.setDate(2, bem.getDataAquisicao());
            pstm.setDouble(3, bem.getValorAquisicao());
            pstm.setString(4, bem.getSituacaoBem());
            pstm.setDouble(5, bem.getVidaUtil());
            pstm.setInt(6, bem.getTurnosTrabalho());
            pstm.setDouble(7, bem.getValorvenda());
            pstm.setDate(8, bem.getDataVenda());
            pstm.executeUpdate();
            return true;

        } catch (Exception e) {
            System.out.print("Erro ao enserir! " + e.getMessage());
            return false;
        }
    }


    public static boolean alterar(Bem bem) {
        try {
            String sql = "update depreciacao set tipo_bem = ?, dataaquisicao = ?, valoraquisicao = ?, situacaobem = ?, vidautil = ?, turnostra
            Connection conn = ConnectionFactory.getConnection();
            PreparedStatement pstm = conn.prepareStatement(sql);
            pstm.setString(1, bem.getTipoBem());
            pstm.setDate(2, bem.getDataAquisicao());
            pstm.setDouble(3, bem.getValorAquisicao());
            pstm.setString(4, bem.getSituacaoBem());
            pstm.setDouble(5, bem.getVidaUtil());
            pstm.setInt(6, bem.getTurnosTrabalho());
            pstm.setDouble(7, bem.getValorvenda());
            pstm.setDate(8, bem.getDataVenda());
            pstm.executeUpdate();
            return true;
        }catch (Exception e) {
            System.out.print("Erro ao Alterar!" + e.getMessage());
            return false;
        }
    }
```

```java
@RestController
@RequestMapping("/bem")
public class BemRecursos {

    @CrossOrigin
    @GetMapping
    public List<Bem> listagem() {
        List<Bem> lista = BemDao.listagem();
        for (Bem bem : lista) {

        double i =0;
        int periododepre = 0;
        double da;
        double vc;
        double gp;
        double vd = 0;

        da = (bem.valorDepre(vd) * bem.calcTaxa(i) * periododepre) / 12;

        vc = bem.getValorAquisicao() - da;

        gp = da - vc;

        }
        return lista;
    }
    @CrossOrigin
    @GetMapping("/{id}")
    public Bem retornarPortId(@PathVariable int id) {
        return BemDao.retornaPorId(id);
    }

    @CrossOrigin
    @PostMapping

    public ResponseEntity<Bem> iserir(@RequestBody Bem bem, HttpServletResponse response) {

        BemDao.inserir(bem);

        bem.setId(BemDao.retornaUltimoId());
        URI uri = ServletUriComponentsBuilder.fromCurrentRequestUri().
                path("/{id}").buildAndExpand(bem.getId()).toUri();
        return ResponseEntity.created(uri).body(bem);
    }

    @CrossOrigin
    @DeleteMapping("/{id}")
    @ResponseStatus(HttpStatus.NO_CONTENT)
    public void delete(@PathVariable int id) {
        BemDao.excluir(id);
    }

    @CrossOrigin
    @PutMapping("/{id}")
    public ResponseEntity<Bem> update(@PathVariable int id, @RequestBody Bem bem){
    Bem bemBanco = BemDao.retornaPorId(id);
    BeanUtils.copyProperties(bem, bemBanco, "id");
    BemDao.alterar(bemBanco);
    return ResponseEntity.ok(bemBanco);
```

Projeto Angula

```
<div class="container">
    <form>
      <div class="card-header">
         <div class="row">
            <div class="col-8">
               Simulador de Depreciação
            </div>
             <div class="col-4">
             <button type="button" class="btn btn-primary"
             (click)= "pag1">Cadastro</button>
             <button type="button" class="btn btn-Warning"
             (click)= "pag2">Depreciação</button>

          </div>

         </div>
      </div>

      <div class="alert alert-primary" role ="alert">
         {{mens}}
      </div>

      <div class="card-body">
         <div class="form-row">


            <div class="form-group col-md-6">
             <label for="inputNome">Data de Aquisição</label>
             <input type="text" class="form-control"
             name="inputDataAquisicao"[(ngModel)]="obj.dataAquisicao">
          </div>

            <div class="form-group col-md-6">
             <label for="inputNome">Valor de Aquisição</label>
             <input type="text" class="form-control"
             name="inputValorAquisicao"[(ngModel)]="obj.valorAquisicao">
          </div>



            <div class="form-group col-md-6">
             <label for="inputNome">Vida Util</label>
             <input type="text" class="form-control"
             name="inputVidaUtil"[(ngModel)]="obj.vidaUtil">
          </div>

            <div class="form-group col-md-6">
             <label for="inputNome">Turnos Trabalhados</label>
             <input type="text" class="form-control"
```

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import {FormsModule} from '@angular/forms';


import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { Pag1Component } from './pag1/pag1.component';
import { Pag2Component } from './pag2/pag2.component';
import { Pag3Component } from './pag3/pag3.component';
import { Pag1Service } from './services/pag1.service';

import { HttpClientModule } from '@angular/common/http';

@NgModule({
  declarations: [
    AppComponent,
    Pag1Component,
    Pag2Component,
    Pag3Component
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    FormsModule
  ],
  providers: [Pag1Service],

  bootstrap: [AppComponent]
})
export class AppModule { }
```

```ts
import { Component, OnInit } from '@angular/core';
import { Bem } from '../model/pag1.model';
import { Pag1Service } from '../services/pag1.service';


@Component({
  selector: 'app-pag1',
  templateUrl: './pag1.component.html',
  styleUrls: ['./pag1.component.css']
})
export class Pag1Component implements OnInit {

  lista: Bem[] = [];
  obj: Bem = {id: 0, tipoBem: '',dataAquisicao:0, valorAquisicao:0, situacaoBem:'', vidaUtil:0, turnosTrabalho:0, datavenda:0, valorvenda:0};
  mens = '';

  constructor(private api: Pag1Service) { }

  ngOnInit() {
    this.consultar();
  }

  consultar(){
    this.api.consultar()
    .toPromise()
    .then
    (res => {
      this.lista = res;
    });
  }

  adicionar(){
    this.api.adicionar(this.obj)
    .toPromise()
    .then(Bem => {
      this.mens = "Cadastro adicionado com sucesso";
      this.consultar();
    });
  }

  excluir(){
    this.api.excluir(this.obj.id)
      .toPromise()
      .then(Bem => {
        this.mens = "Bem excluida com sucesso";
        this.consultar();
      });
  }
```

```html
<div class="container">
    <form>
      <div class="card-header">
        <div class="row">
          <div class="col-8">
            Cadastro
          </div>
          <div class="col-4">
           <button type="button" class="btn btn-primary"
           (click)= "pag1">Cadastro</button>
           <button type="button" class="btn btn-Warning"
           (click)= "pag2"><a href="D:\AngulaPI\ProjetoPI\src\app\pag2\pag2.component.html">Depreciação</a></button>

         </div>
          <div class="col-4">
            <button type="button" class="btn btn-primary"
            (click)= "adicionar()">Adicionar</button>
            <button type="button" class="btn btn-Warning"
            (click)= "alterar()">Alterar</button>
            <button type="button" class="btn btn-danger"
            (click)= "excluir()">Excluir</button>
          </div>
        </div>
      </div>

      <div class="alert alert-primary" role ="alert">
        {{mens}}
      </div>

      <div class="card-body">
        <div class="form-row">
          <div class="form-group col-md-6">
            <label for="inputCodigo">Código</label>
            <input type="number" class="form-control"
            name="inputCodigo" [(ngModel)]= "obj.id">
          </div>
          <div class="form-group col-md-6">
            <label for="inputNome">Nome do Bem</label>
            <input type="text" class="form-control"
            name="inputTipoBem"[(ngModel)]="obj.tipoBem">
          </div>
          <div class="form-group col-md-6">
           <label for="inputNome">Data de Aquisição</label>
           <input type="text" class="form-control"
           name="inputDataAquisicao"[(ngModel)]="obj.dataAquisicao">
         </div>
```

```typescript
import { Component, OnInit } from '@angular/core';
import { Bem } from '../model/pag1.model';
import { Pag1Service } from '../services/pag1.service';


@Component({
  selector: 'app-pag2',
  templateUrl: './pag2.component.html',
  styleUrls: ['./pag2.component.css']
})
export class Pag2Component implements OnInit {

  lista: Bem[] = [];
  obj: Bem = {id: 0, tipoBem: '',dataAquisicao:0, valorAquisicao:0, situacaoBem:'', vidaUtil:0, turnosTrabalho:0, datavenda:0, valorvenda:0};
  mens = '';

  constructor(private api: Pag1Service) { }

  ngOnInit() {
    this.consultar();
  }

  consultar(){
    this.api.consultar()
    .toPromise()
    .then
    (res => {
      this.lista = res;
    });
  }

  adicionar(){
    this.api.adicionar(this.obj)
    .toPromise()
    .then(Bem => {
      this.mens = "Descrição" + Bem + " adicionada com sucesso";
      this.consultar();
    });
  }

  excluir(){
    this.api.excluir(this.obj.id)
      .toPromise()
      .then(Bem => {
        this.mens = "Bem excluida com sucesso";
        this.consultar();
      });
  }
```

Cadastro

Cadastro | Depreciação

Adicionar | Alterar | Excluir

Código: 0

Nome do Bem:

Data de Aquisição: 0

Valor de Aquisição: 0

Situação do Bem:

Vida Util: 0

Turnos Trabalhados: 0

Data da Venda: 0

valor da Venda: 0

Código	Nome do Bem	Data de Aquisição	Valor de Aquisição	Situação do Bem	Vida Util	Turnos Trabalhados	Data da Venda	valor da Venda	...
9	Casa	09/08/2015	300000	Novo	25	0 0		0 0	Carregar
6	Computador	19/08/2017	4000	Novo	5	8		0 0	Carregar
7	Moto	19/08/2017	8000	Novo	5	0 0		0 0	Carregar

Simulador de Depreciação

Cadastro | Depreciação

Data de Aquisição: 2016-08-11

Valor de Aquisição: 5000

Vida Util: 10

Turnos Trabalhados: 1

Data da Venda:

valor da Venda: 0

Período a Depreciar: 40

Valor da Depreciação: 1500

valos Contábil: 3500

Perda ou Ganho: -2000

Código	Nome do Bem	Data de Aquisição	Valor de Aquisição	Situação do Bem	Vida Util	Turnos Trabalhados	Data da Venda	valor da Venda	...
9	Casa	2015-08-09	300000	Novo	25	0		0	Carregar
6	Computador	2017-08-19	4000	Novo	5	8		0	Carregar
10	Equipamentos	2016-08-11	5000	Novo	10	1		0	Carregar
7	Moto	2017-08-19	8000	Novo	5	0		0	Carregar