```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

using api_pim.Entities;
using System.Net;
using api_pim.Exceptions;

namespace api_pim.Controllers;

[ApiController]
[Route("api/[controller]")]
public class TipoCargoController : ControllerBase
{
    private readonly ApplicationDbContext _context;

    private readonly ILogger<TipoCargoController> _logger;

    public TipoCargoController(ApplicationDbContext context, ILogger<TipoCargoController> logger)
    {
        _context = context;
        _logger = logger;
    }

    [HttpGet]
    [Authorize]
    public IActionResult Get()
    {
        try
        {
            var result = _context.TipoCargo.ToList();
            _logger.LogInformation("TipoCargoController.Get -> [Success]");
            return Ok(result);
        }
        catch (Exception)
        {
            _logger.LogError("TipoCargoController.Get -> [Error]");
           throw new ApiException((int) HttpStatusCode.InternalServerError, $"Erro interno [{ErrorCode.GTC}]");
        }
    }

    [HttpPost]
    [Authorize]
    public IActionResult Create([FromBody] TipoCargo request)
    {
        try
        {
            if (request == null)
            {
                return BadRequest(new { message = "Dados do tipo de cargo inválidos." });
            }

            _context.TipoCargo.Add(request);
            _context.SaveChanges();
            _logger.LogInformation("TipoCargoController.Get -> [Success]");

            return Ok($"Tipo de cargo [{request.Valor}] criado com sucesso");
        }
        catch (Exception)
        {
            _logger.LogError("TipoCargoController.Create -> [Error]");
            throw new ApiException((int) HttpStatusCode.InternalServerError, $"Erro interno [{ErrorCode.CTC}]");
        }
    }

    [HttpDelete]
    [Authorize]
    public IActionResult Delete(int cod) {
        try
        {
            var tipoCargo = _context.TipoCargo.FirstOrDefault(x => x.Cod == cod);
            if (tipoCargo == null) {
                return NotFound(new { message = "Tipo de cargo não encontrado" });
            }

            _context.TipoCargo.Remove(tipoCargo);
            _context.SaveChanges();
            return NoContent();
        }
        catch (Exception)
        {
            _logger.LogError("TipoCargoController.Create -> [Error]");
            throw new ApiException((int) HttpStatusCode.InternalServerError, $"Erro interno [{ErrorCode.DTC}]");
        }
    }
}
```