```java
package br.com.padon.application.controllers;

import br.com.padon.application.models.Fornece;
import br.com.padon.application.models.Fornecedor;
import br.com.padon.application.repositorys.ForneceRepository;
import br.com.padon.application.repositorys.FornecedorRepository;
import com.fasterxml.jackson.databind.JsonNode;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Optional;

@RestController
@CrossOrigin
@RequestMapping("/fornecedor")
public class FornecedorController {

    @Autowired
    private FornecedorRepository fornecedor;
    @Autowired
    private ForneceRepository fornece;

    @PostMapping("/create")
    public Fornecedor createFornecedor(@RequestBody JsonNode node) {
        return fornecedor.save(new Fornecedor(
                node.get("endereco").asText(),
                node.get("contato").asText(),
                node.get("telefone").asText(),
                node.get("nome").asText()
        ));
    }

    @GetMapping("/get")
    public List<Fornecedor> getAllFornecedores() {
        return fornecedor.findAll();
    }

    @PostMapping("/save")
    public Fornecedor saveFornecedor(@RequestBody JsonNode
 node) {
        Fornecedor fornecedorById = fornecedor.findById(node.get("id").asInt()).orElseThrow();
        fornecedorById.setEndereco(node.get("endereco").asText());
        fornecedorById.setContato(node.get("contato").asText());
        fornecedorById.setTelefone(node.get("telefone").asText());
        fornecedorById.setNome(node.get("nome").asText());
        return fornecedor.save(fornecedorById);
    }

    @PostMapping("/byid")
    public Optional<Fornecedor> getFornecedor(@RequestBody JsonNode node) {
        return fornecedor.findById(node.get("id").asInt());
    }

    @PostMapping("/delete")
    public boolean deleteFornecedor(@RequestBody JsonNode node) {
        fornecedor.deleteById(node.get("id").asInt());
        return true;
    }

    @PostMapping("/add")
    public Fornece addProduto(@RequestBody JsonNode node) {
        return fornece.save(new Fornece(node.get("produtoId").asInt(), node.get("fornecedorId").asInt(), node.get("preco").asDouble()));
    }

    @PostMapping("/remove")
    public boolean removeProduto(@RequestBody JsonNode node) {
        Fornece result = fornece.getFornece(node.get("produtoId").asInt(), node.get("fornecedorId").asInt());
        if (result == null) {
            return false;
        }

        fornece.delete(result);
        return true;
    }
}

```