

```

public class CNPJValidator {

    private static final int TAMANHO_CNPJ_SEM_DV = 12;
    private static final String REGEX_CARACTERES_FORMATACAO = "[./-]";
    private static final String REGEX_FORMACAO_BASE_CNPJ = "[A-Z\\d]{12}";
    private static final String REGEX_FORMACAO_DV = "[\\d]{2}";
    private static final String REGEX_VALOR_ZERADO = "^0+$";
    private static final int VALOR_BASE = (int) '0';
    private static final int[] PESOS_DV = { 6, 5, 4, 3, 2, 9, 8, 7, 6, 5, 4, 3, 2 };

    public static boolean isValid(String cnpj) {
        if (cnpj != null) {
            cnpj = removeCaracteresFormatacao(cnpj);
            if (isCnpjFormacaoValidaComDV(cnpj)) {
                String dvInformado = cnpj.substring(TAMANHO_CNPJ_SEM_DV);
                String dvCalculado = calculaDV(cnpj.substring(0, TAMANHO_CNPJ_SEM_DV));
                return dvCalculado.equals(dvInformado);
            }
        }
        return false;
    }

    public static String calculaDV(String baseCnpj) {
        if (baseCnpj != null) {
            baseCnpj = removeCaracteresFormatacao(baseCnpj);
            if (isCnpjFormacaoValidaSemDV(baseCnpj)) {
                String dv1 = String.format("%d", calculaDigito(baseCnpj));
                String dv2 = String.format("%d", calculaDigito(baseCnpj.concat(dv1)));
                return dv1.concat(dv2);
            }
            throw new IllegalArgumentException(String.format("Cnpj %s não é válido para o cálculo do DV", baseCnpj));
        }
    }

    private static int calculaDigito(String cnpj) {
        int soma = 0;
        for (int indice = cnpj.length() - 1; indice >= 0; indice--) {
            int valorCaracter = (int) cnpj.charAt(indice) - VALOR_BASE;
            soma += valorCaracter * PESOS_DV[PESOS_DV.length - cnpj.length() + indice];
        }
        return soma % 11 < 2 ? 0 : 11 - (soma % 11);
    }

    private static String removeCaracteresFormatacao(String cnpj) {
        return cnpj.trim().replaceAll(REGEX_CARACTERES_FORMATACAO, "");
    }

    private static boolean isCnpjFormacaoValidaSemDV(String cnpj) {
        return cnpj.matches(REGEX_FORMACAO_BASE_CNPJ) &&
            !cnpj.matches(REGEX_VALOR_ZERADO);
    }

    private static boolean isCnpjFormacaoValidaComDV(String cnpj) {
        return cnpj.matches(REGEX_FORMACAO_BASE_CNPJ.concat(REGEX_FORMACAO_DV)) &&
            !cnpj.matches(REGEX_VALOR_ZERADO);
    }

}

```