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

import br.com.padon.application.models.Funcionario;
import com.auth0.jwt.JWT;
import com.auth0.jwt.algorithms.Algorithm;
import com.auth0.jwt.exceptions.JWTCreationException;
import com.auth0.jwt.exceptions.JWTVerificationException;
import org.springframework.stereotype.Service;

import java.time.Instant;
import java.time.LocalDateTime;
import java.time.ZoneOffset;

@Service
public class TokenService {
    private final String secret = "Padon";

    public String generateToken(Funcionario user) {
        try {
            Algorithm algorithm = Algorithm.HMAC256(secret);
            return JWT.create()
                    .withIssuer("auth-api")
                    .withSubject(user.getUsername())
                    .withExpiresAt(genExpirationDate())
                    .sign(algorithm);
        } catch (JWTCreationException exception) {
            throw new RuntimeException("Error while generating token", exception);
        }
    }

    public String validateToken(String token) {
        try {
            Algorithm algorithm = Algorithm.HMAC256(secret);
            return JWT.require(algorithm)
                    .withIssuer("auth-api")
                    .build()
                    .verify(token)
                    .getSubject();
        } catch (JWTVerificationException exception) {
            return "";
        }
    }

    private Instant genExpirationDate() {
        return LocalDateTime.now().plusHours(12).toInstant(ZoneOffset.of("-03:00"));
    }
}
```