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

import br.com.padon.application.repositorys.FuncionarioRepository;
import jakarta.servlet.FilterChain;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.stereotype.Component;
import org.springframework.web.filter.OncePerRequestFilter;

import java.io.IOException;

@Component
public class SecurityFilter extends OncePerRequestFilter {
    @Autowired
    TokenService tokenService;
    @Autowired
    FuncionarioRepository funcionario;

    @Override
    protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response, FilterChain filterChain) throws ServletException, IOException {
        var token = this.recoverToken(request);
        if (token != null) {
            var login = tokenService.validateToken(token);
            UserDetails user = funcionario.findByUsuario(login);
            var authentication = new UsernamePasswordAuthenticationToken(user, null, user.getAuthorities());
            SecurityContextHolder.getContext().setAuthentication(authentication);
        }
        filterChain.doFilter(request, response);
    }

    private String recoverToken(HttpServletRequest request) {
        var authHeader = request.getHeader("token");
        if (authHeader == null || authHeader.isBlank() || authHeader.equals("null")) return null;
        return authHeader.replace("Bearer ", "");
    }
}

```