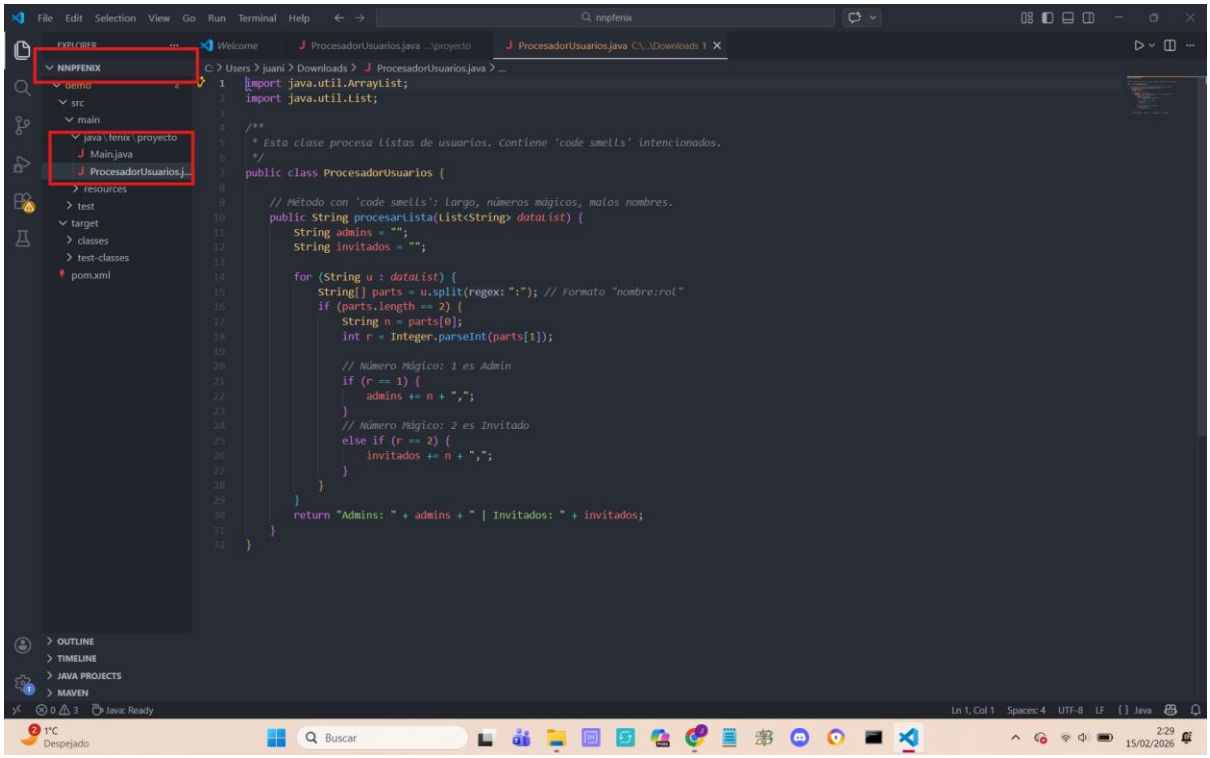
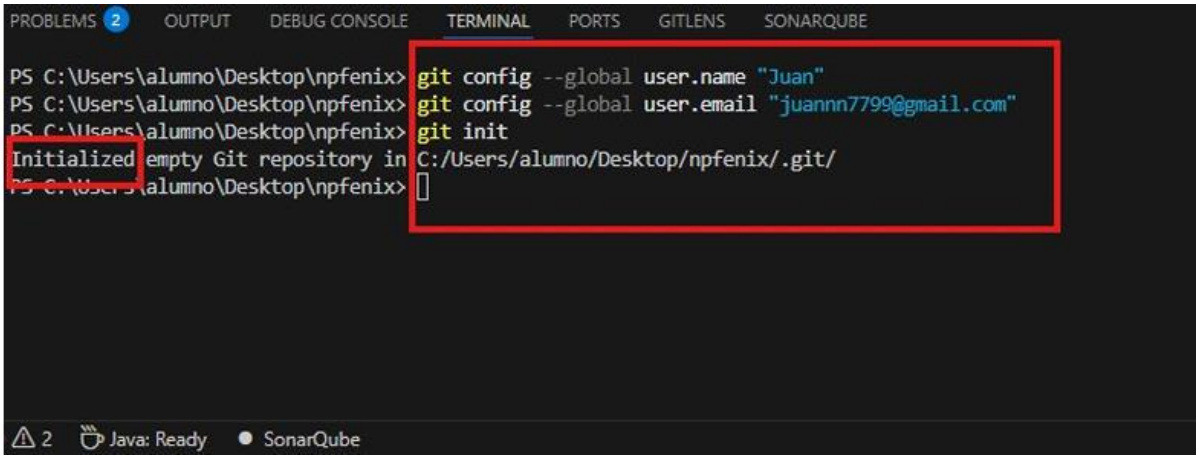
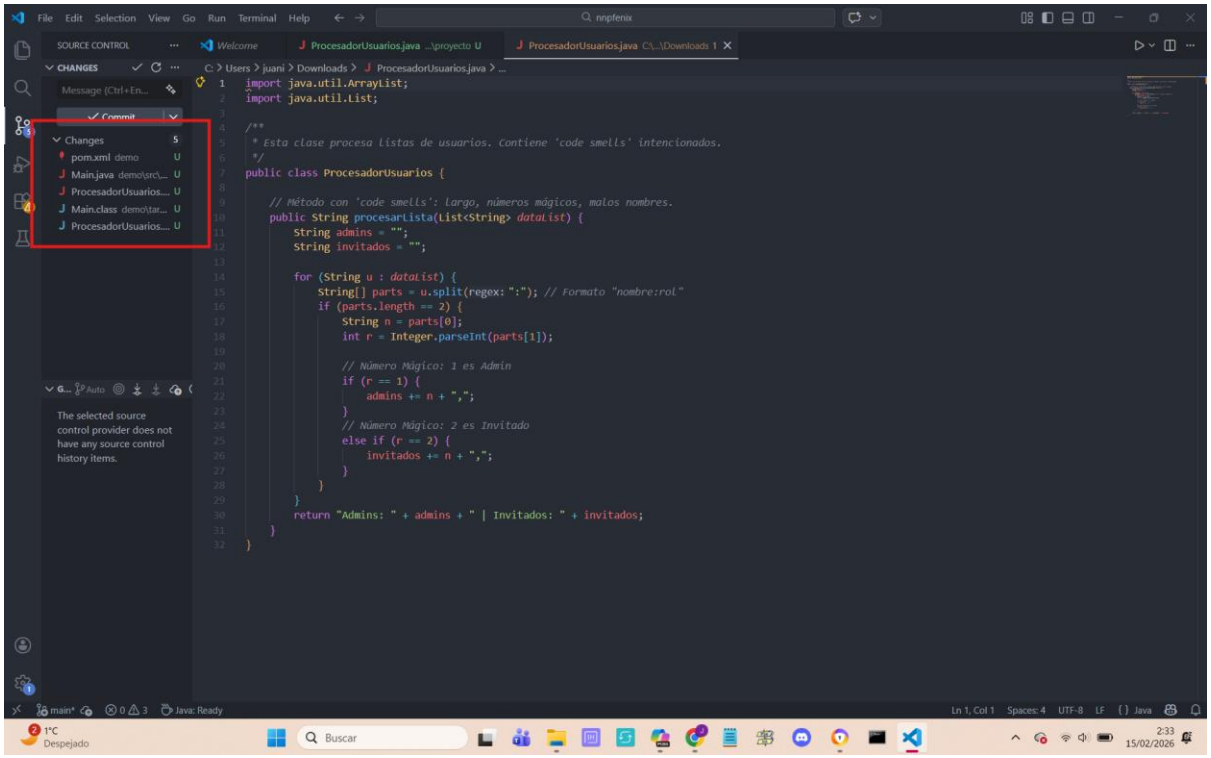
PROYECTO FÉNIX JUAN SILVESTRE DURO 1ºDAM

PASO 1 — PASO 1 — Crear el proyecto en el IDE Captura: Proyecto creado con la clase abierta.

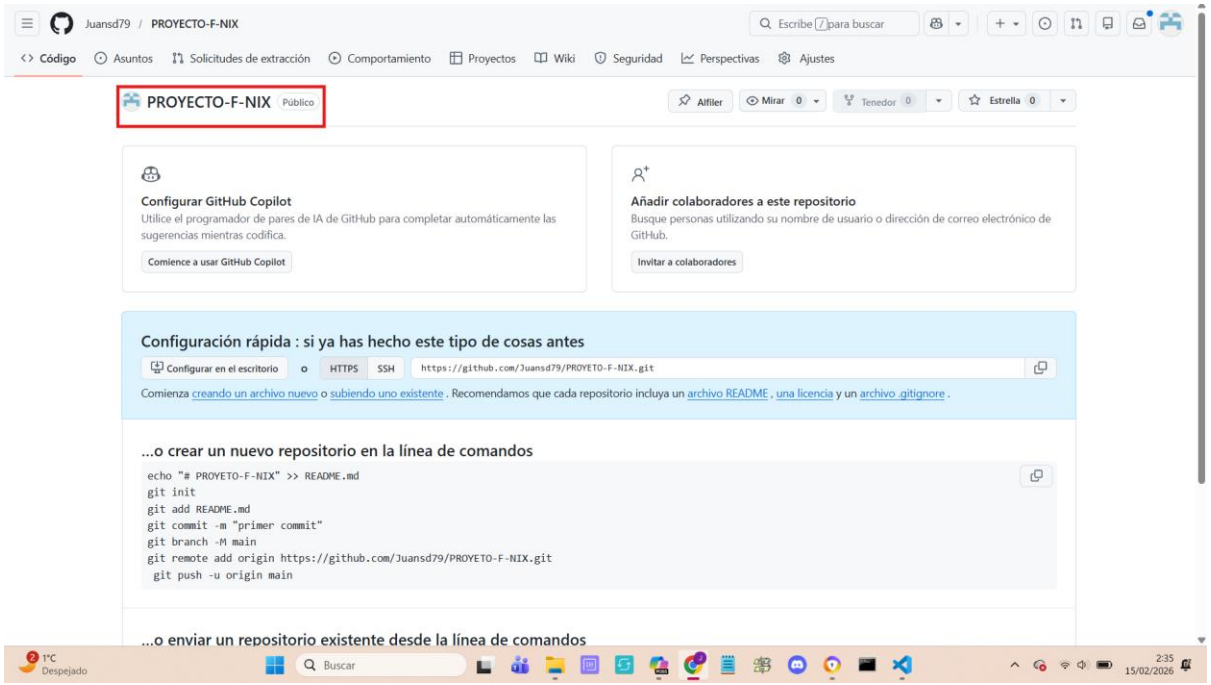


PASO 2 — Inicializar repositorio Git local Captura: Git inicializado.

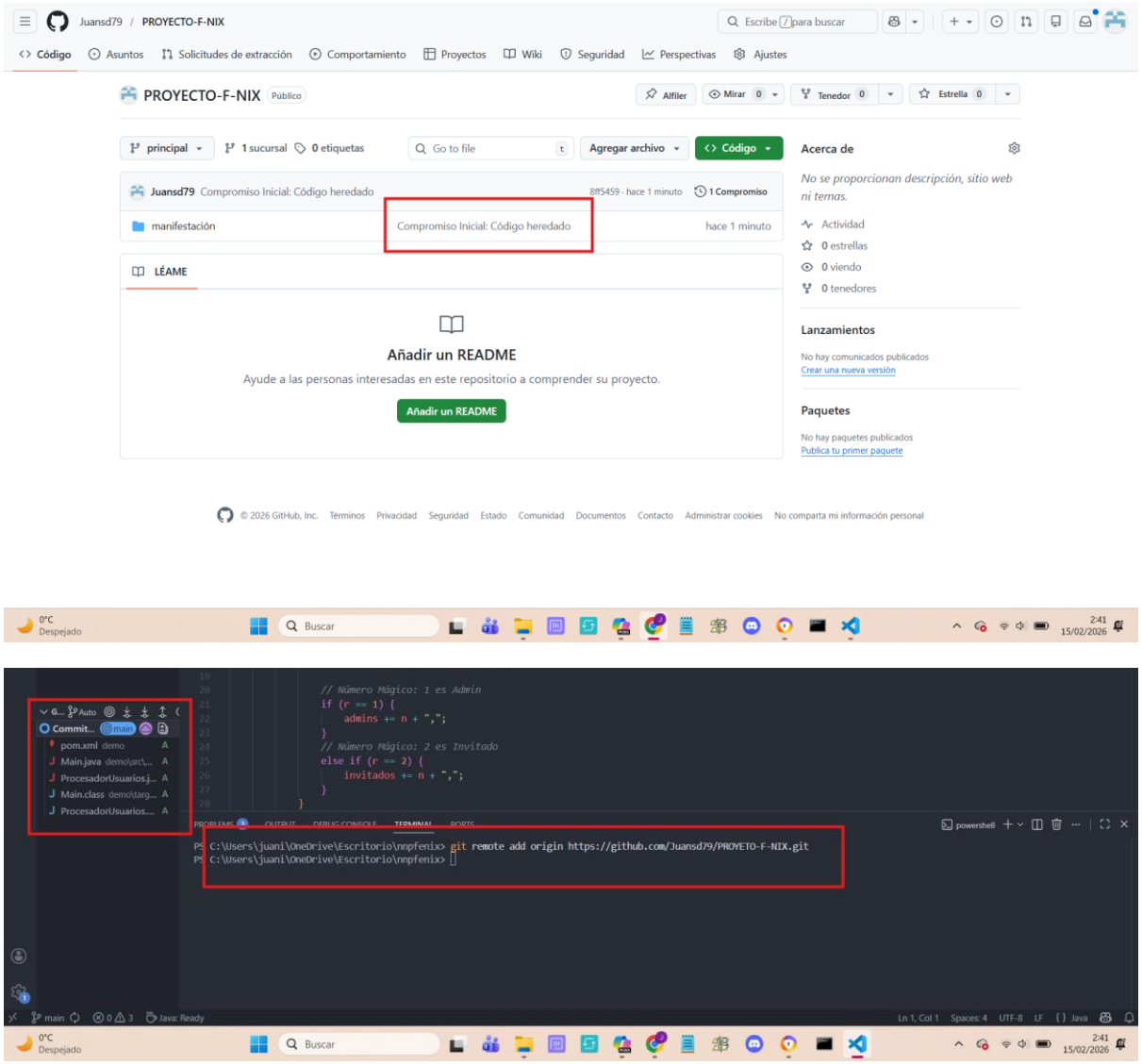




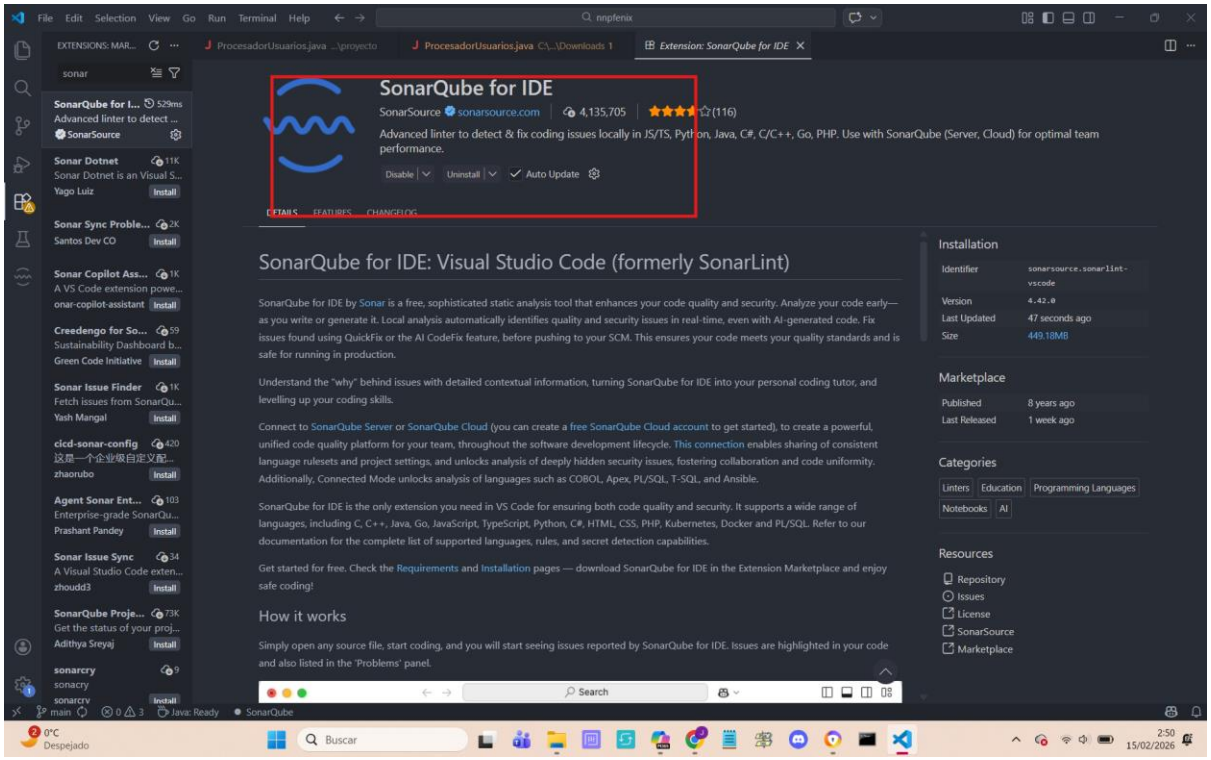
PASO 3 — Crear repositorio en GitHub creamos un nuevo repositorio para subir los push



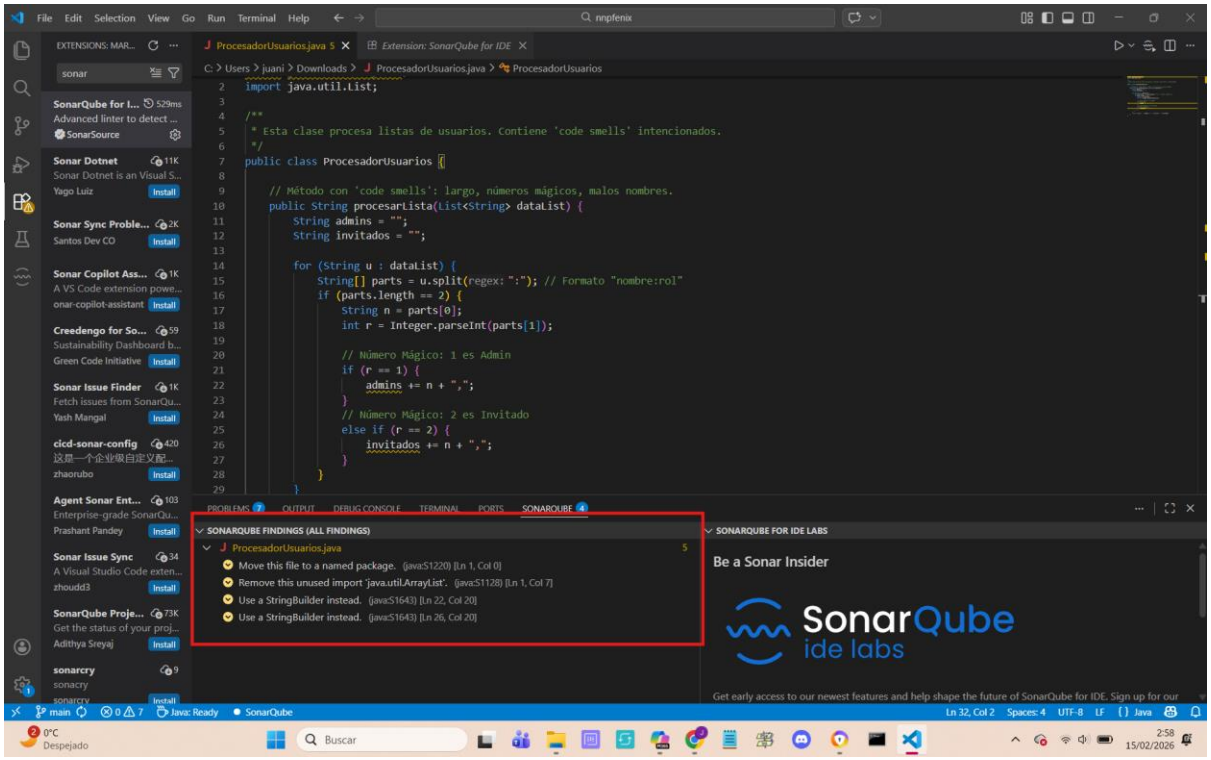
PASO 4 — Conectar repositorio local con remoto se concecta para poder mostrar los cambios



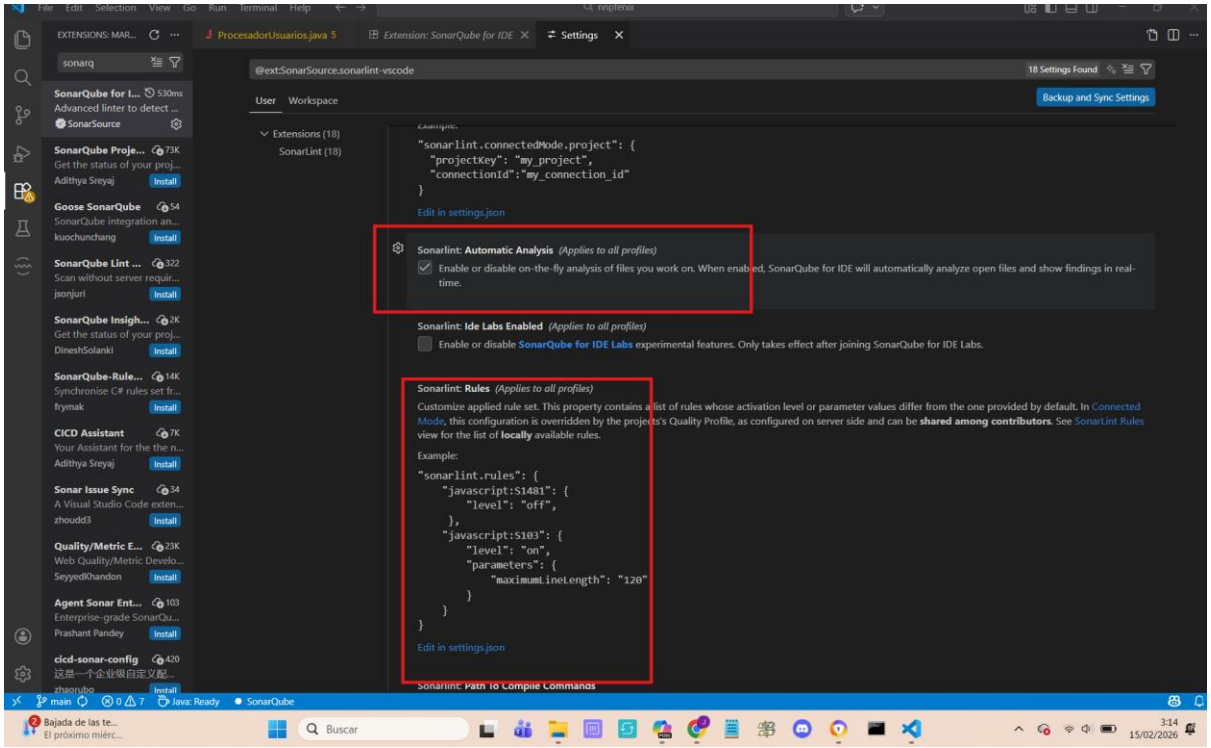
PASO 6 — Instalar SonarQube se instala para luego en setings entrar en sonarlint y acceder a las normas para activarlas

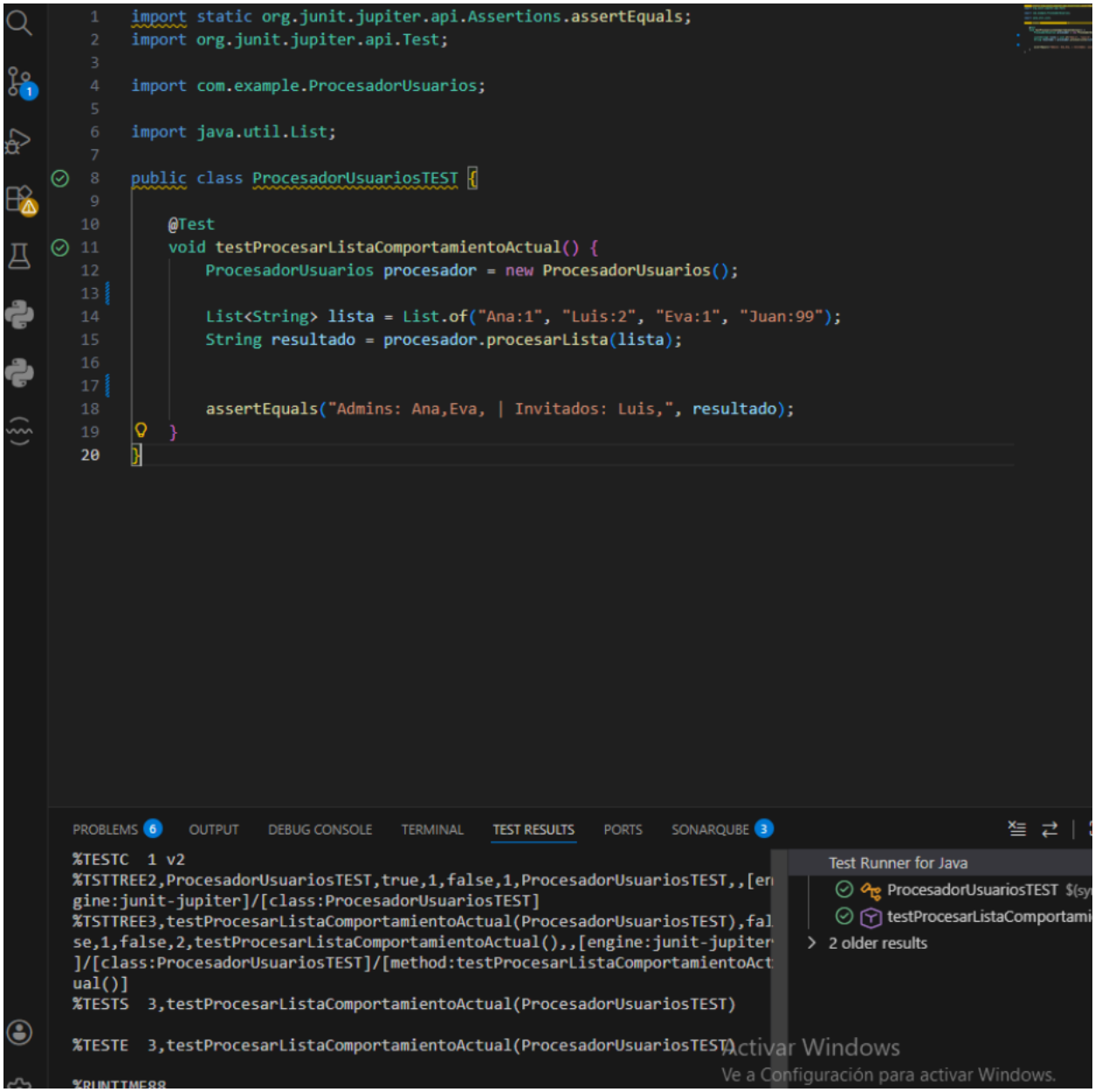


PASO 7 — Ejecutar análisis se ejecuta el analisis para ver los problemas

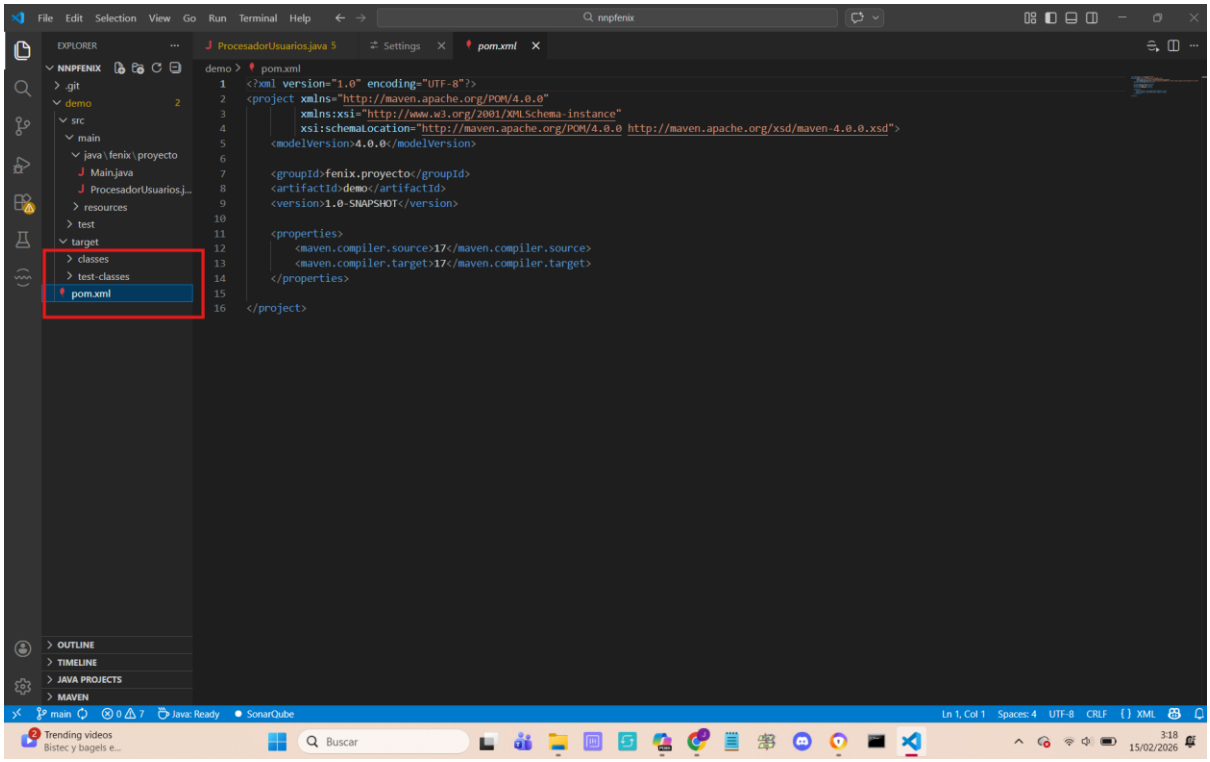


PASO 8 — Revisar reglas para comprobar que todo esta en orden y los tests pasan correctamente

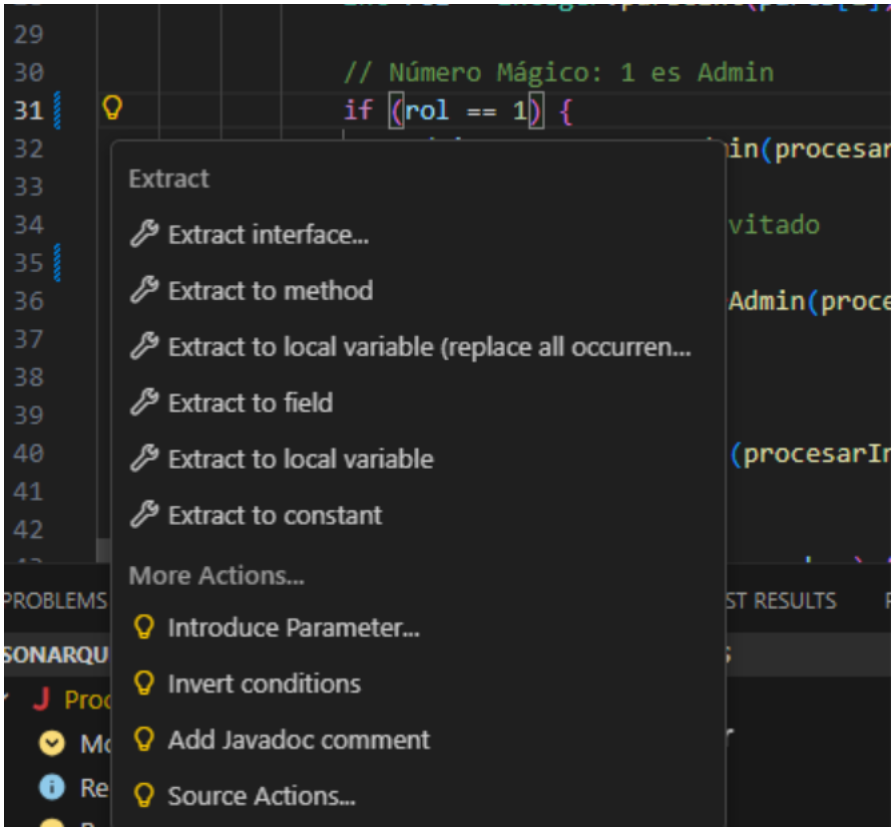


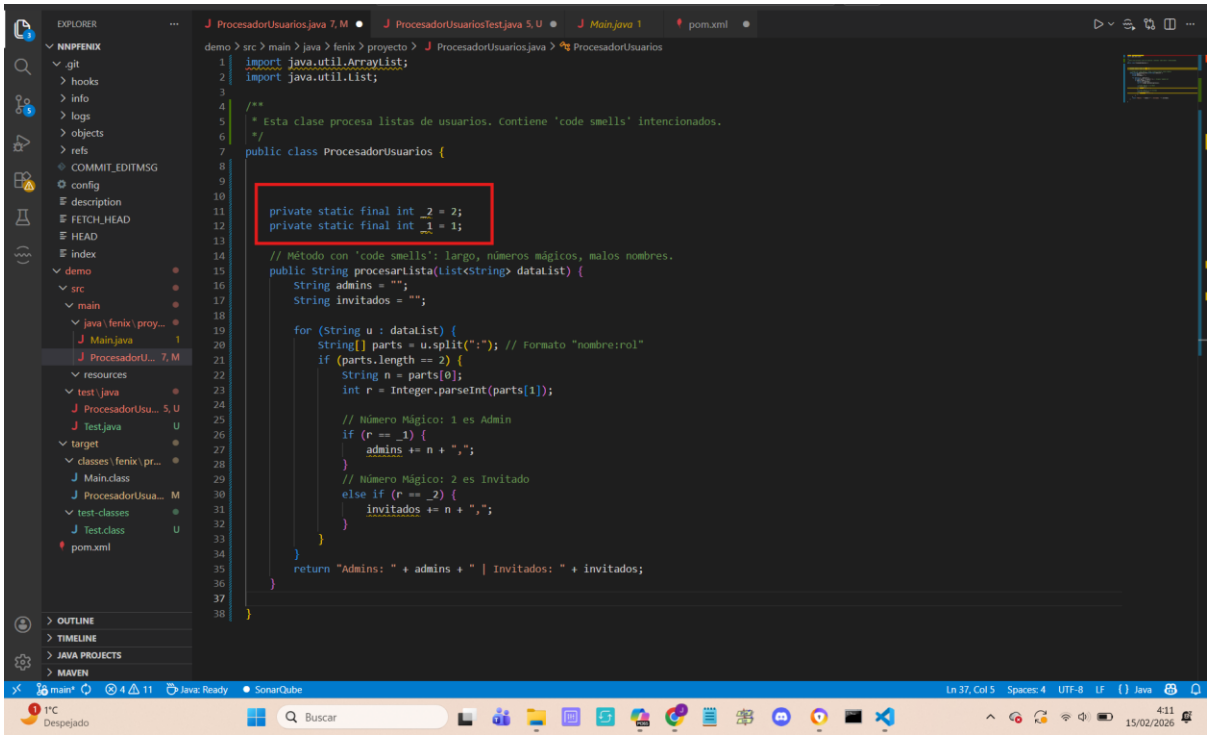


PASO 9 — Añadir JUnit al proyecto se añade junit al proyecto

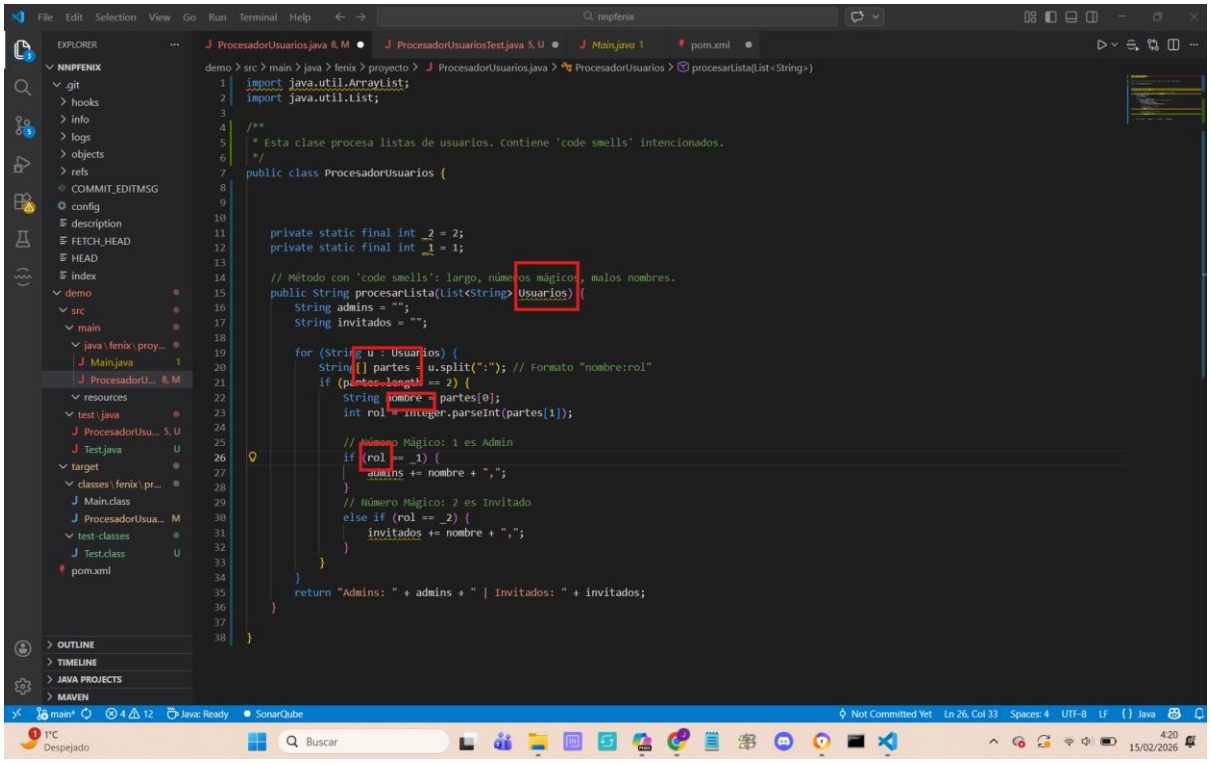


PASO 13 — Extraer constantes escogemos los numeros y entramos en la opcion extract y damos a extract to constant

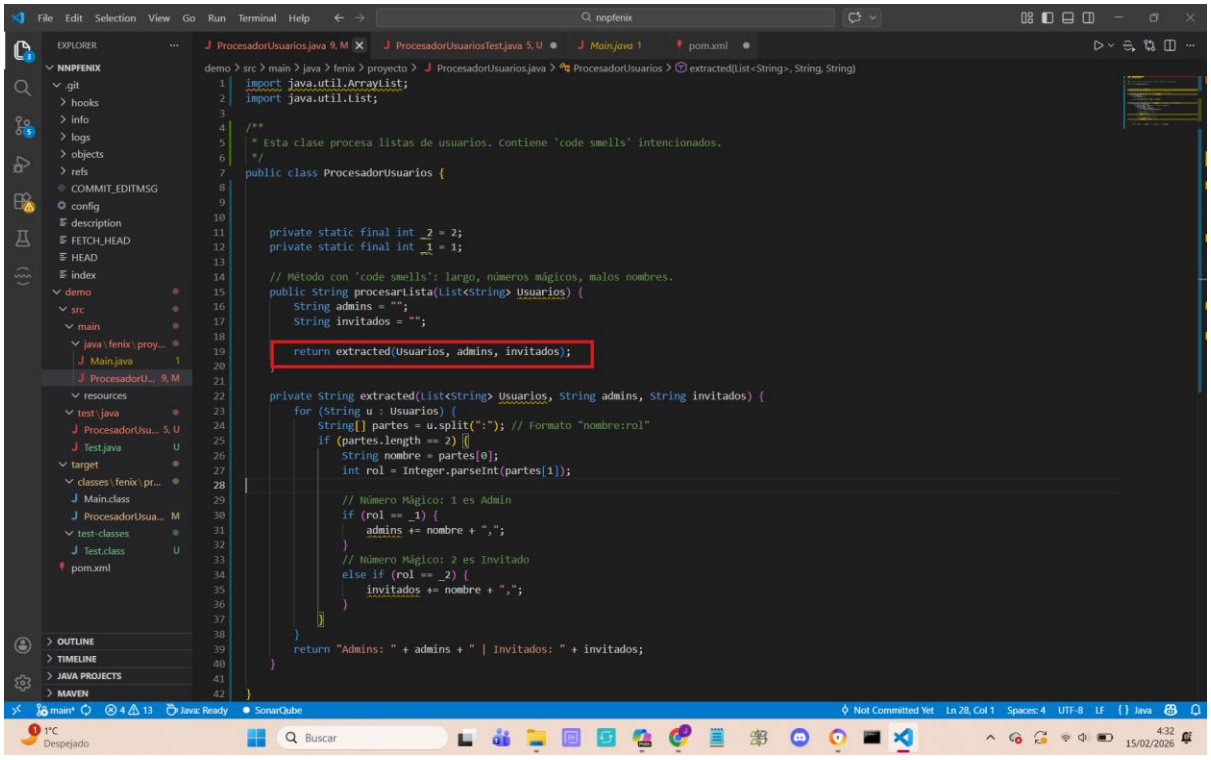




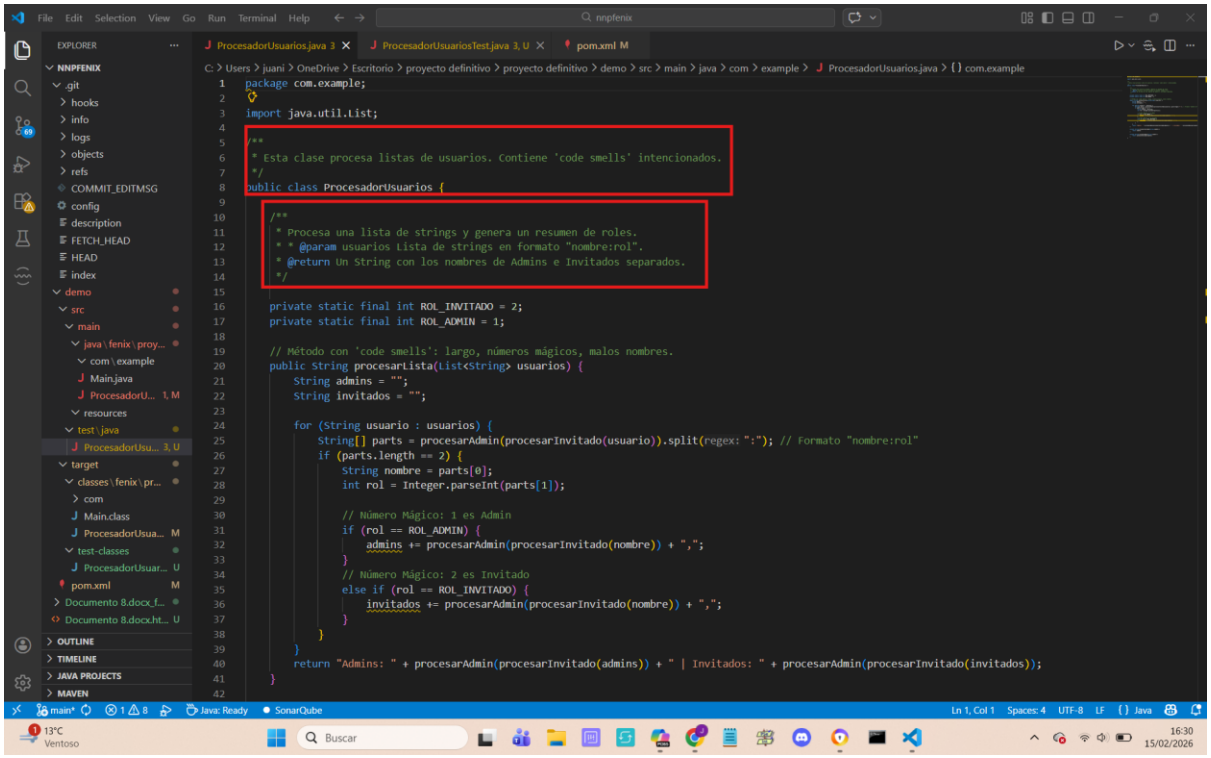
PASO 14 — Renombrar variables utilizamos el comando f2 para cambiar los nombres de las variables rol n part



PASO 15 — Extract Method extraemos el método del bulce if

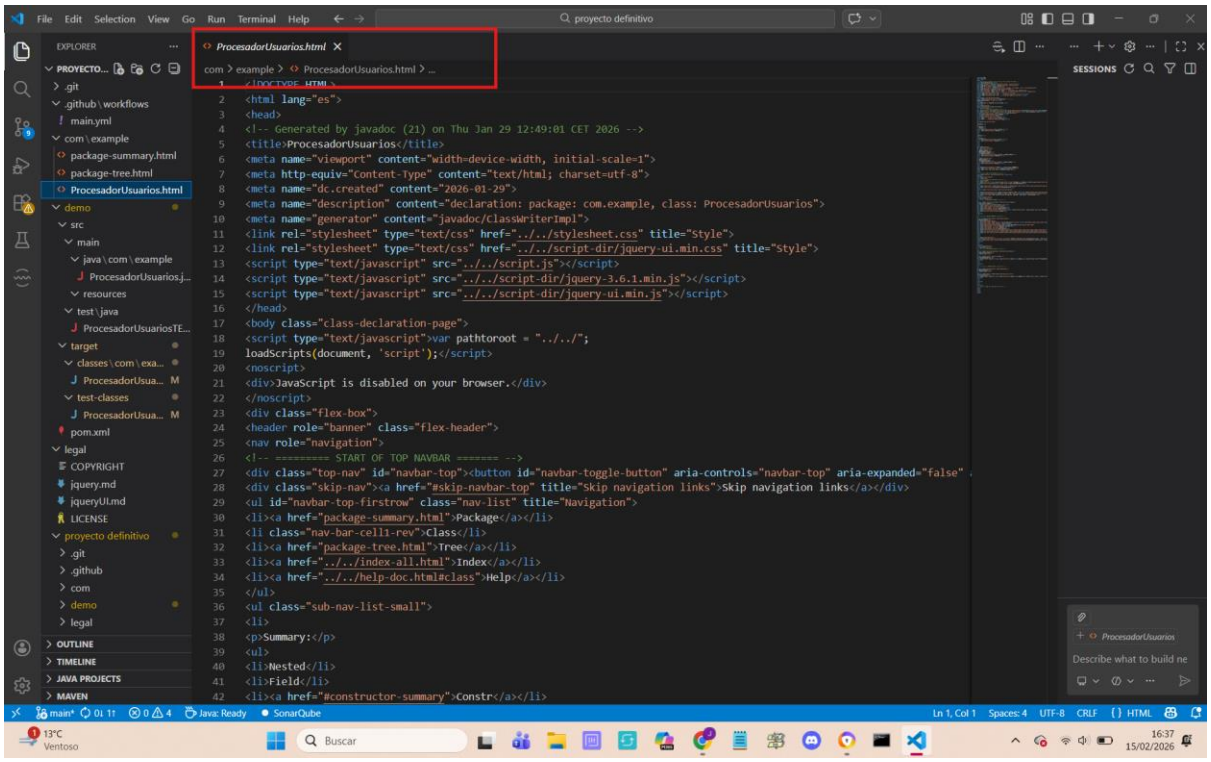


PASO 18 — Añadir Javadoc se añade los comentarios javadoc



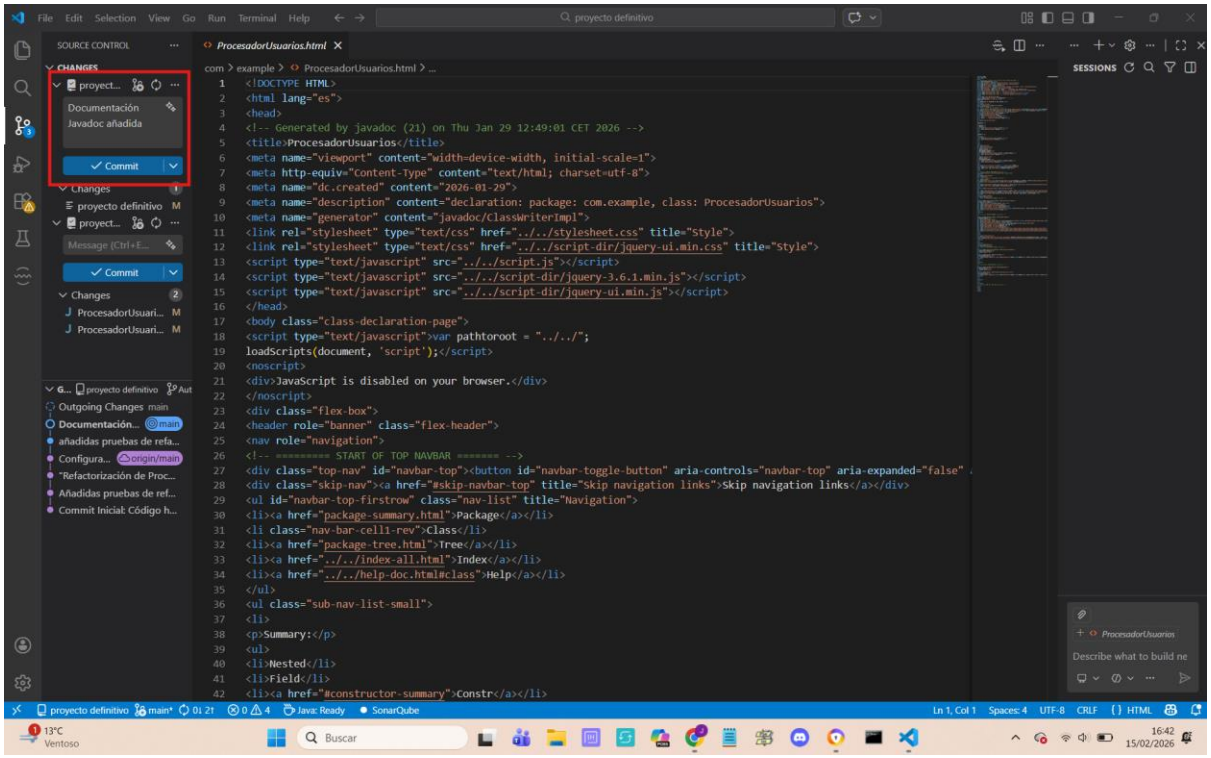
```java
package com.example;

import java.util.List;

/**
 * Esta clase procesa listas de usuarios. Contiene 'code smells' intencionados.
 */
public class ProcesadorUsuarios {

    /**
     * Procesa una lista de strings y genera un resumen de roles.
     * * @param usuarios Lista de strings en formato "nombre:rol".
     * @return Un String con los nombres de Admins e Invitados separados.
     */

    private static final int ROL_INVITADO = 2;
    private static final int ROL_ADMIN = 1;

    // Método con 'code smells': largo, números mágicos, malos nombres.
    public String procesarLista(List<String> usuarios) {
        String admins = "";
        String invitados = "";

        for (String usuario : usuarios) {
            String[] parts = procesarAdmin(procesarInvitado(usuario)).split(regex: ":"); // Formato "nombre:rol"
            if (parts.length == 2) {
                String nombre = parts[0];
                int rol = Integer.parseInt(parts[1]);

                // Número Mágico: 1 es Admin
                if (rol == ROL_ADMIN) {
                    admins += procesarAdmin(procesarInvitado(nombre)) + ",";
                }
                // Número Mágico: 2 es Invitado
                else if (rol == ROL_INVITADO) {
                    invitados += procesarAdmin(procesarInvitado(nombre)) + ",";
                }
            }
        }
        return "Admins: " + procesarAdmin(procesarInvitado(admins)) + " | Invitados: " + procesarAdmin(procesarInvitado(invitados));
    }
```

PASO 19 — Generar documentación se genera la documentacion html

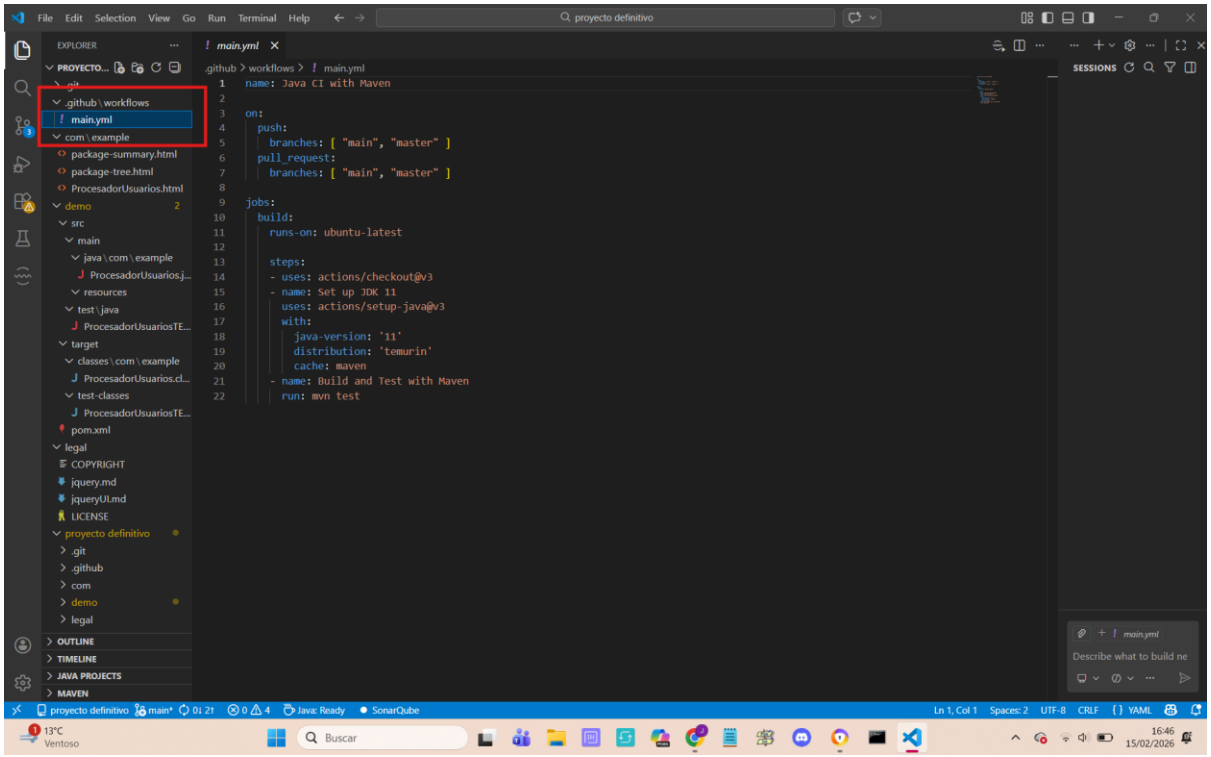


```
<!DOCTYPE HTML>
<html lang="es">
<head>
<!-- Generated by javadoc (21) on Thu Jan 29 12:49:01 CET 2026 -->
<title>ProcesadorUsuarios</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="dc.created" content="2026-01-29">
<meta name="description" content="declaration: package: com.example, class: ProcesadorUsuarios">
<meta name="generator" content="javadoc/ClassWriterImpl">
<link rel="stylesheet" type="text/css" href="../../stylesheet.css" title="Style">
<link rel="stylesheet" type="text/css" href="../../script-dir/jquery-ui.min.css" title="Style">
<script type="text/javascript" src="../../script.js"></script>
<script type="text/javascript" src="../../script-dir/jquery-3.6.1.min.js"></script>
<script type="text/javascript" src="../../script-dir/jquery-ui.min.js"></script>
</head>
<body class="class-declaration-page">
<script type="text/javascript">var pathtoroot = "../../";
loadScripts(document, 'script');</script>
<noscript>
<div>JavaScript is disabled on your browser.</div>
</noscript>
<div class="flex-box">
<header role="banner" class="flex-header">
<nav role="navigation">
<!-- ========= START OF TOP NAVBAR ======= -->
<div class="top-nav" id="navbar-top"><button id="navbar-toggle-button" aria-controls="navbar-top" aria-expanded="false"
<div class="skip-nav"><a href="#skip-navbar-top" title="Skip navigation links">Skip navigation links</a></div>
<ul id="navbar-top-firstrow" class="nav-list" title="Navigation">
<li><a href="package-summary.html">Package</a></li>
<li class="nav-bar-cell1-rev">Class</li>
<li><a href="package-tree.html">Tree</a></li>
<li><a href="../../index-all.html">Index</a></li>
<li><a href="../../help-doc.html#class">Help</a></li>
</ul>
<ul class="sub-nav-list-small">
<li>
<p>Summary:</p>
<ul>
<li>Nested</li>
<li>Field</li>
<li><a href="#constructor-summary">Constr</a></li>
```

PASO 20 — Commit documentación hacemos un commit de la documentacion para que haya constancia



PASO 21 — Activar GitHub Actions entramos en github lo activamos y hacemos en archivo yaml



PASO 22 — Verificar que tenga: se comprueba en el archivo que corra el test con maven