

Instrumento de Instalación de la Extensión Tech Debt Metrics

Descripción:

Esta herramienta automatizada es una herramienta que permite calcular la deuda técnica de un programa, lo cual es útil para la medición y desarrollo de la presente prueba estática, además, es una extensión que se le añade al editor de código Visual Studio Code, la cual despliega información relacionada con la deuda técnica de cada método de una sección de código.

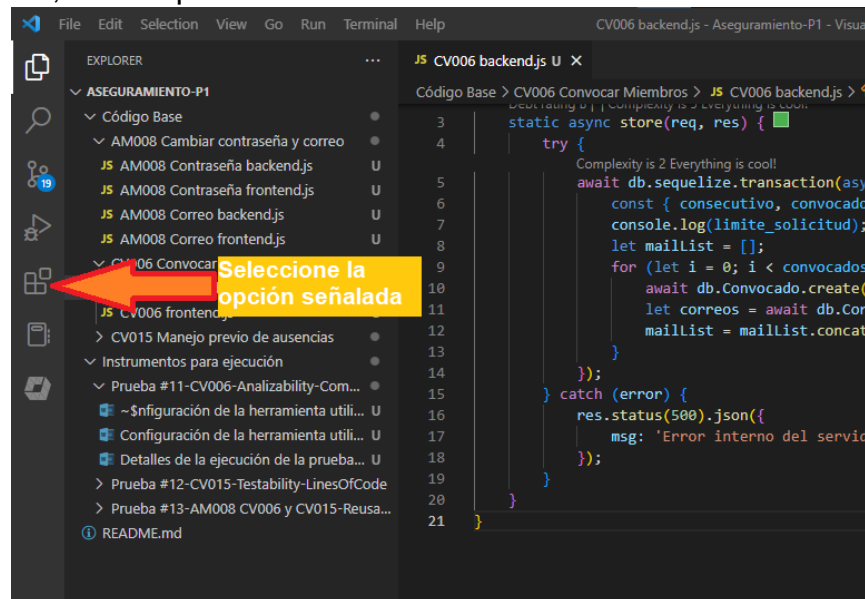
Enlace de referencia:

<https://marketplace.visualstudio.com/items?itemName=Stepsize.tech-debt-tracker>

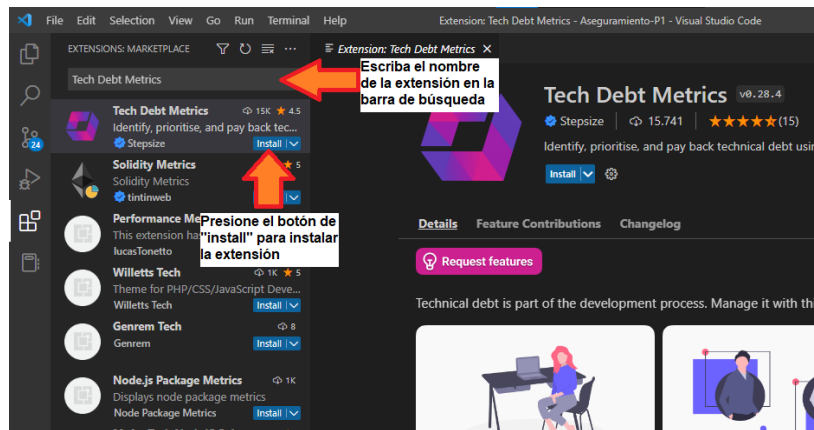
Instalación y configuración:

A continuación, se detallan los pasos para instalar la extensión VS Code Counter en Visual Studio Code partiendo del hecho que ya se cuenta instalado con este editor de código en su sistema:

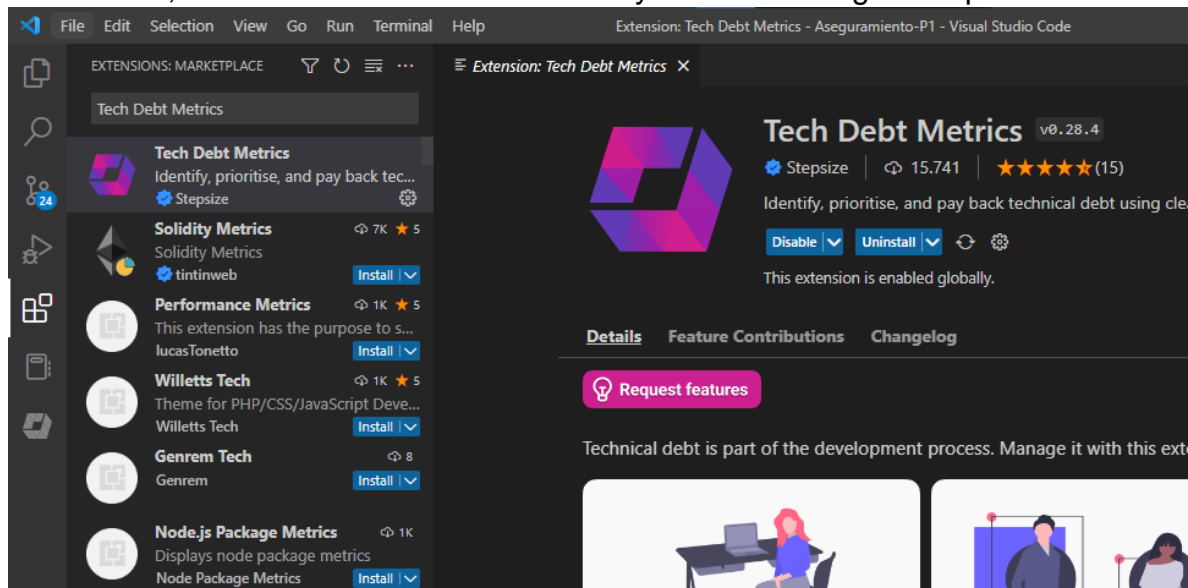
- 1) Primeramente, abra la pestaña de extensiones del editor:



- 2) Seguidamente, ingrese el nombre de la extensión y presione instalar, como se muestra aquí:



- 3) Finalmente, la extensión se habrá instalado y mostrará la siguiente pantalla:



- 4) Con esto termina el proceso de instalación, y además sería suficiente para poder obtener los datos de la deuda técnica de los métodos del código, la cual se podrá observar directamente desde el código, como en el siguiente ejemplo:

Primer Proyecto del curso de Aseguramiento de la Calidad del Software
Tecnológico de Costa Rica, II semestre 2022
Subgrupo de trabajo 06

```
Complexity is 4 Everything is cool! | Debt rating A |
static async changePassword(req, res) {
  const { cedula, clave } = req.body;
  try {
    Complexity is 1 Everything is cool!
    await db.sequelize.transaction(async t => {
      const encryptedPass = await this.encryptPassword(clave);
      await db.Usuario.update({ clave: encryptedPass },
        { where: { cedula: cedula } });
      res.json({
        success: true
      });
    });
  } catch (error) {
    res.status(500).json({
      msg: 'Error interno del servidor.'
    });
  }
}

Complexity is 2 Everything is cool! | Debt rating A |
static async encryptPassword(password) {
  const salt = await bcrypt.genSalt(10);
  const hash = await bcrypt.hash(password, salt);
  return hash;
}
```

En donde, como se observa se le atribuye un rating de deuda a cada función, tanto a changePassword como a encryptPassword, lo cual se usará para la prueba ya descrita.

Aquí finaliza este instrumento de configuración para la extensión VS Code Counter.