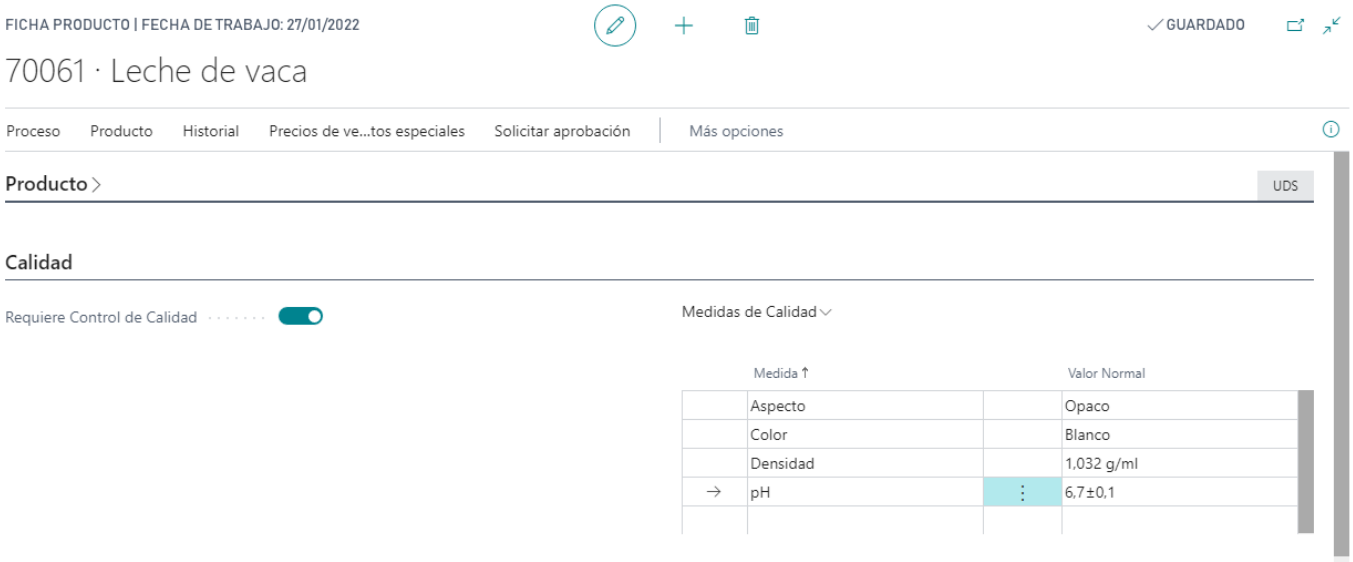
Ejercicio 3. Datos maestros

Empezaremos haciendo el desarrollo relacionado con los datos maestros para obtener el siguiente resultado en la Ficha del Producto:



Para aislar nuestro desarrollo del desarrollo que otros compañeros puedan estar haciendo sobre el mismo proyecto, crearemos una rama de Git en la que trabajar. Cuando terminemos nuestra parte del desarrollo, llevaremos los cambios de nuestra rama particular a la rama principal del proyecto y eliminaremos nuestra rama.

En este ejercicio aprenderemos:

- Que son las ramas de Git
- A crear una rama de Git
- A hacer un merge de dos ramas distintas de Git

Indicaciones

Para completar el ejercicio realiza las siguientes acciones:

1. Crea una nueva rama de Git
2. Crea y muestra en la ficha del producto el campo "Requiere Control de Calidad"
3. Crea y muestra en la ficha del producto una tabla de "Medidas de Control de Calidad"
4. Haz un merge de tu rama de desarrollo con la rama principal
5. Elimina la rama del desarrollo

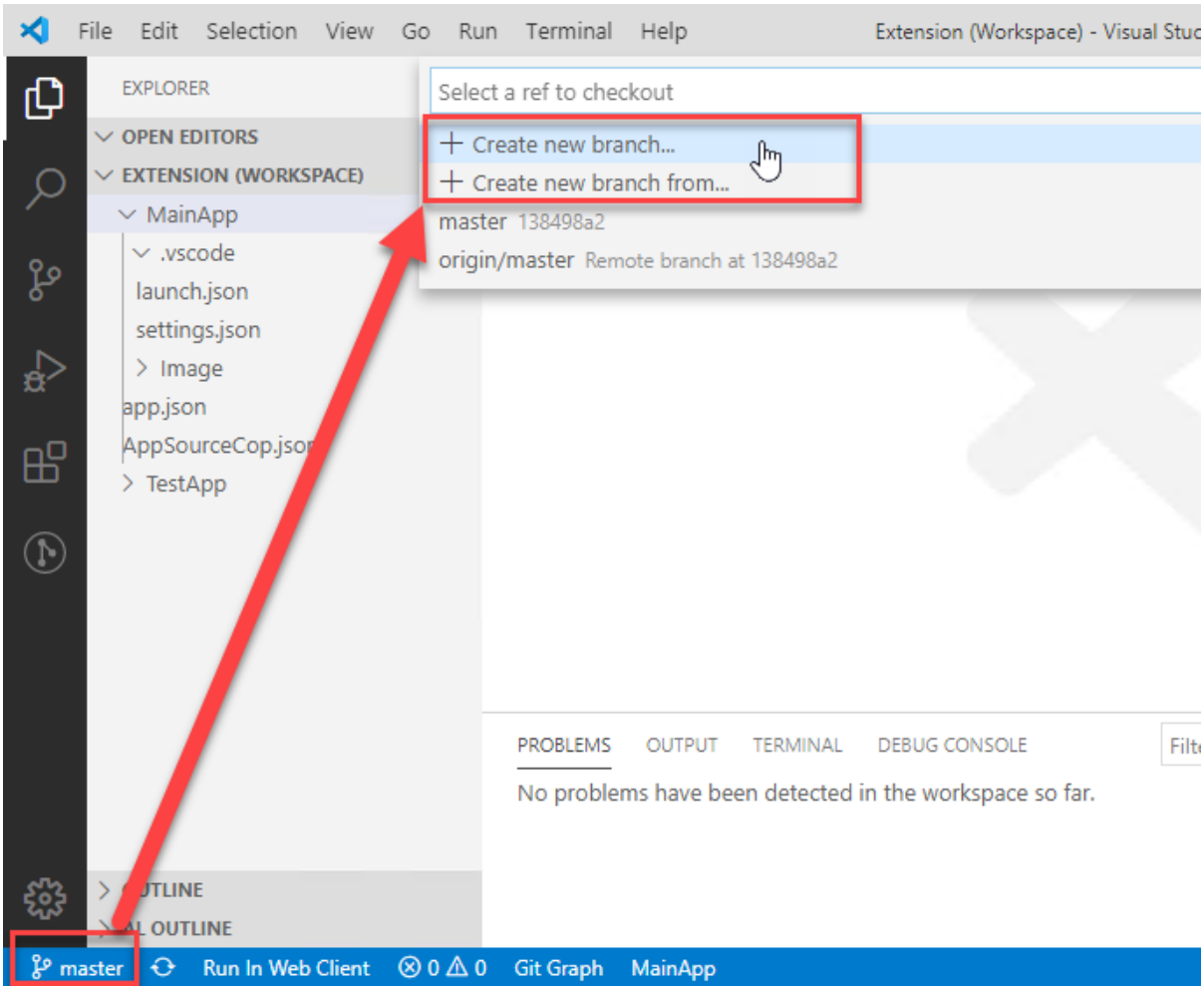
Indicaciones paso a paso

Para completar el ejercicio sigue los siguientes pasos en Visual Studio Code:

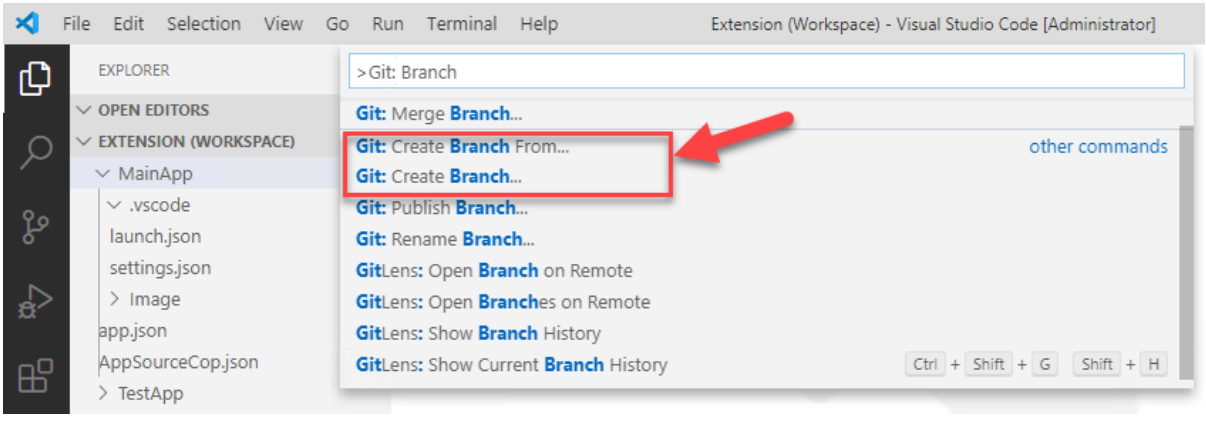
1. Creación de una nueva rama de Git llamada *Feature_MasterData_**TusIniciales***

   La creación de una nueva rama de Git puede hacerse de distintas formas:

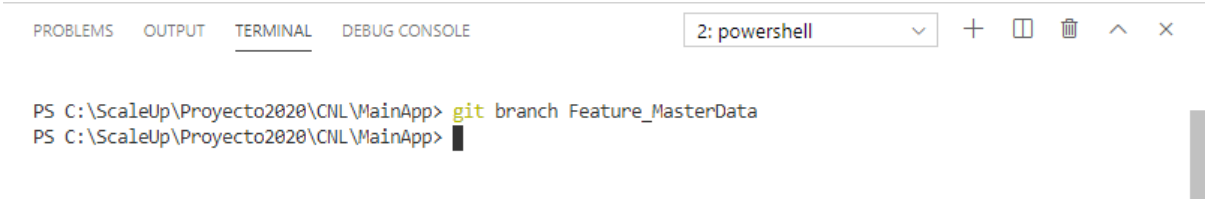
- Haciendo click en el selector de ramas que hay en el pie y seleccionando la opción **+ Create new branch** o la opción **+ Create new branch from**



- En la paleta de comandos de Visual Studio Code, ejecutando el comando **Git: Create Branch...** o el comando **Git: Create Branch From...**



- En el terminal, ejecutando el comando **git branch NombreDeLaRama**



2. Campo "Requiere Control de Calidad" en la ficha del producto

1. Crea una tableextension de la tabla Item con la siguiente información:

```
tableextension 50100 "Clip Item" extends Item
{
    fields
    {
        field(50100; "Clip Requires Quality Control"; Boolean)
        {
            Caption = 'Requires Quality Control', comment =
'ESP="Requiere Control de Calidad"';
            DataClassification = CustomerContent;
        }
    }
}
```

2. Crea una pageextension de la página Item Card con la siguiente información

```
pageextension 50100 "Clip Item Card" extends "Item Card"
{
    layout
    {
        addafter(Item)
        {
            group(ClipQualityControl)
            {
                Caption = 'Quality', comment = 'ESP="Calidad"';
                field("Clip Requires Quality Control"; "Clip Requires
Quality Control")
                {
                    ApplicationArea = Basic, Suite;
                    ToolTip = 'Specifies whether the item must go
through a quality control process when it is being purchased', comment
= 'ESP="Especifica si el producto debe someterse a un proceso de
control de calidad cuando es comprado"';
                }
            }
        }
    }
}
```

3. Publica el desarrollo y comprueba que sea correcto
4. Haz un commit con los cambios y sincronízalo con el repositorio remoto

3. "Medidas de Calidad" en la ficha del producto

1. Crea una tabla con la siguiente información:

```
table 50100 "Clip Quality Measures"
{
    Caption = 'Quality Measures', comment = 'ESP="Medidas de Calidad"';
    fields
    {
        field(1; "Item No."; Code[20])
        {
            Caption = 'Item No.', comment = 'ESP="Nº Producto"';
            TableRelation = Item;
            DataClassification = CustomerContent;
        }
        field(2; Measure; Text[20])
        {
            Caption = 'Measure', comment = 'ESP="Medida"';
            DataClassification = CustomerContent;
        }
        field(3; "Normal Value"; Text[50])
        {
            Caption = 'Normal Value', comment = 'ESP="Valor Normal"';
            DataClassification = CustomerContent;
        }
    }

    keys
    {
        key(PK; "Item No.", Measure)
        {
            Clustered = true;
        }
    }
}
```

2. Crea una página con la siguiente información:

```
page 50100 "Clip Quality Measures"
{
    Caption = 'Quality Measures', comment = 'ESP="Medidas de Calidad"';
    PageType = ListPart;
    UsageCategory = None;
    SourceTable = "Clip Quality Measures";

    layout
    {
        area(Content)
        {
            repeater(Group)
            {
                field(Measure; Measure)
                {
                    ApplicationArea = Basic, Suite;
                    ToolTip = 'Type of measure to do on the item',
```

```
comment = 'ESP="Tipo de medida a realizar en el producto"';
                }
                field("Normal Value"; "Normal Value")
                {
                    ApplicationArea = Basic, Suite;
                    ToolTip = 'Normal value for the measure', comment =
'ESP="Valor normal para la medida"';
                }
            }
        }
    }
}
```

3. Añade el siguiente control a la Ficha del Producto:

```
part("Clip Quality Measures"; "Clip Quality Measures")
{
    ApplicationArea = Basic, Suite;
    SubPageLink = "Item No." = field("No.");
}
```

4. Haz un commit con los cambios y sincronízalo con el repositorio remoto

4. Merge de la rama de desarrollo con la rama principal.
   Ejecuta los siguientes comando fe git en el terminal.

   1. Sincronización de master

   ```
   git checkout master
   git pull
   ```

   La primera instrucción hará que nos movamos de la rama en la que nos encontrábamos a la rama master.
   La segunda instrucción descargara los cambios que hubiera en el remoto, para asegurar que estamos sincronizados.

   2. Incorporación de los cambios que vienen de Master (FI - Forward Integration)

   ```
   git checkout BranchName
   git merge --squash master
   ```

   La primera instrucción hará que nos movamos a la rama *BranchName*.
   La segunda instrucción hará un merge con el contenido de la master.
   La opción --squash hará que el contenido de la master se traslade a la rama de forma conjunta.

   3. Commit del merge de los cambios que vienen de Master

```
git commit -m "Incorporate changes coming from master"
git push
```

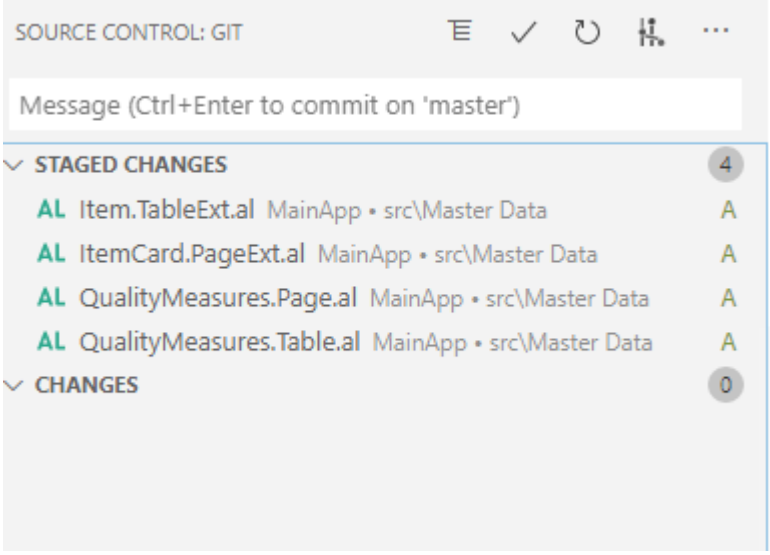
La primera instrucción hará el commit.
La segunda instrucción subirá el commit al remoto.

4. Incorporación de los cambios que vienen de Rama (RI - Reverse Integration)

```
git checkout master
git merge --squash BranchName
```

La primera instrucción hará que nos movamos a la rama master.
La segunda instrucción hará un merge con el contenido de la Rama

El resultado será un único set de cambios que aparecerán en estado Staged



5. Commit del merge de los cambios que vienen de Rama

```
git commit -m "Incorporate BranchName"
git push
```

La primera instrucción registra el commit.
La segunda instrucción sube el commit al remoto.

6. Eliminación de la Rama

```
git branch -d BranchName
git push origin --delete BranchName
```

La primera instrucción elimina la rama en local.
La segunda instrucción elimina la rama en el remoto.

7. Limpieza de las ramas eliminadas por otros.

```
git fetch --prune
git branch -vv
```

La primera instrucción se conecta al remoto para actualizar la lista de ramas existentes.
La segunda instrucción nos devuelve una lista de ramas locales que ya no existen en el remoto (contienen el texto *:gone]*)

Eliminaremos todas las ramas locales que no existan en el remoto con la siguiente instrucción:

```
git branch -d BranchName
```