Para asignar un campo del dataset a una etiqueta

Interfaz de usuario gráfica, Aplicación, Tabla, Word, Excel

Descripción generada automáticamente

Para buscar valores en un dataset en código en modo debug

? ds.Tables.Item(0).Rows(0).Item(1)

Pin 1045

Correo [salvador.ruiz@agilesofr.mx](mailto:salvador.ruiz@agilesofr.mx)

Wal68259

https://github.com/ArturoDevAgile/Reporteador

Cuando el reporte es con vista, el diseñador se abre sobre el primero en la lista

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Para ver la relación de las etiquetas con el dataset es en el segundo, el designer

Una captura de pantalla de una computadora

Descripción generada automáticamente

## Cambiar la configuración de compilación

Para cambiar la configuración de compilación, ya sea:

* En la barra de herramientas, elija **Depurar** o **Liberar** de la lista **Configuraciones de la solución** .

configuración de compilación de barras de herramientas

o

* En el menú **Generar , seleccione Administrador de configuración** , luego seleccione **Depurar** o **Liberar** .

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para ejecutar en modo debug cambiarlo a “Debug”

# ¿Cómo se utiliza Github pages?

[GitHub](https://github.com/) es un sitio "social coding". Te permite subir repositorios de código para almacenarlo en el **sistema de control de versiones** [Git](http://git-scm.com/). Tu puedes colaborar en proyectos de código, y el sistema es código abierto por defecto, lo que significa que cualquiera en el mundo puede encontrar tu código en GitHub, usarlo, aprender de el, y mejorarlo. ¡Tú puedes hacer eso con el código de otras personas tambien! Este artículo provee una guía básica para publicar contenido usando la característica gh-pages de Github.

## [Publicando contenido](https://developer.mozilla.org/es/docs/Learn/Common_questions/Using_Github_pages#publicando_contenido)

Github es una comunidad muy importante y útil para involucrarse, y Git/GitHub es un [sistema de control de versiones](http://git-scm.com/book/en/v2/Getting-Started-About-Version-Control) muy popular — la mayoría de las empresas de tecnología ahora lo utilizan en su flujo de trabajo. GitHub tiene una característica muy útil llamada [GitHub pages](https://pages.github.com/), que te permite publicar el código del sitio en vivo en la Web.

### [Configuración básica de Github](https://developer.mozilla.org/es/docs/Learn/Common_questions/Using_Github_pages#configuraci%C3%B3n_b%C3%A1sica_de_github)

1. Primero que todo, [instala Git](http://git-scm.com/downloads) en tu máquina. Este es el software del sistema de control de versiones subyacente en el que GitHub funciona.
2. Seguido, [Regístrate para una cuenta de GitHub](https://github.com/join). Es simple y fácil.
3. Una vez te hayas registrado, inicia sesión en [github.com](https://github.com/) con tu nombre de usuario y contraseña.

### [Preparando tu código para subirlo](https://developer.mozilla.org/es/docs/Learn/Common_questions/Using_Github_pages#preparando_tu_c%C3%B3digo_para_subirlo)

Tú puedes almacenar cualquier código que tu quieras en un repositorio de Github, pero para usar la característica páginas de Github con pleno efecto, tu código debe estar estructurado como un sitio web típico, por ejemplo que el punto de entrada primario sea un archivo HTML llamado index.html.

La otra cosa que necesitas hacer antes de seguir adelante es inicializar el directorio de código como un repositorio Git. para hacer esto:

1. Apunta la línea de comandos a tu directorio test-site (o como se llame el directorio que contiene tu sitio web). Para esto, usa el comando cd (Es decir "cambio de directorio"). Esto es lo que deberías digitar si has puesto tu sitio web en un directorio llamado test-site en tu escritorio:

cd Desktop/test-site

1. Cuando la línea comandos está apuntando hacia el interior del directorio de tu sitio web, digita el siguiente comando, que le dice a la herramienta git para convertir el directorio en un repositorio git:
2. git init

#### An aside on command line interfaces

La mejor manera de subir tu código a Github es mediante la linea de comandos — esta es una ventana donde tú escribe comandos para hacer cosas como crear archivos y ejecutar programas, en lugar de hacer clic dentro de una interfaz de usuario. Se verá algo como esto:

Una captura de pantalla de una computadora

Descripción generada automáticamente

**Nota**: Tú también podrías considerar una [interfaz gráfica de usuario de Git](http://git-scm.com/downloads/guis) para hacer el mismo trabajo, si te sientes incómodo con la linea de comandos.

Cada sistema operativo viene con una herramienta de línea de comandos:

* **Windows**: **Command Prompt** se puede acceder pulsando la tecla Windows, tipeando Command Prompt, Y elegirlo de la lista que aparece. Nota que Windows tiene sus propias convenciones de comando diferentes de Linux y OS X, así que los comandos abajo pueden variar en su máquina.
* **OS X**: **Terminal** se puede encontrar en Aplicaciones > Utilidades.
* **Linux**: Por lo general, puede extraer una terminal con Ctrl + Alt + T. Si eso no funciona, busca **Terminal** en una barra de aplicaciones o menú.

Esto puede parecer un poco espantoso al principio, pero no te preocupes — que pronto conseguiras la caída de los conceptos básicos. Tú le dices a la computadora que haga algo en la terminal, digitando un comando y oprimiendo la tecla Enter, como se ha visto anteriormente.

### [Creando un repositorio para tu código](https://developer.mozilla.org/es/docs/Learn/Common_questions/Using_Github_pages#creando_un_repositorio_para_tu_c%C3%B3digo)

1. A continuación, tu necesitas crear un nuevo repositorio para colocar tus archivos en el. Has clic en el signo más (+) en la parte superior derecha de la página de inicio de GitHub, luego escoge Nuevo Repositorio.
2. En esta página, en la caja Nombre del Repositorio, digita el nombre para tu repositorio de código, por ejemplo my-repository.
3. También llena una descripción para decir lo que tu repositorio va a contener. Tu pantalla debe mostrar algo como esto:  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente
4. Has Clic en Crear repositorio; Esto debería llevarte a la siguiente página:   
   Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamente

### [Subiendo tus archivos a GitHub](https://developer.mozilla.org/es/docs/Learn/Common_questions/Using_Github_pages#subiendo_tus_archivos_a_github)

1. En la página actual, tú estás interesado en la sección …o empujar un repositorio existente desde la línea de comandos. Tú deberias ver dos lineas de código listado en esta sección. Copia la totalidad de la primera línea, pégala en la línea de comandos, presiona la tecla Enter. El comando debería mostrarte algo como esto:

git remote add origin https://github.com/chrisdavidmills/my-repository.git

1. A continuación, digita los siguientes dos comandos, presionando Enter despues de cada uno. Estos preparán el código para subirlo a GitHub, y pida a GIt que administre estos archivos.
2. git add --all

git commit -m 'adding my files to my repository'

1. Por último, empuja el código hasta GitHub yendo a la página de GitHub en la que estas e ingresando en la terminal el segundo de los dos comandos que vimos …o empuje un repositorio existente desde la sección de línea de comandos:

git push -u origin master

1. Ahora necesitas crear la rama gh-pages de tu repositorio; actualiza la página actual y verá una página del repositorio algo así como la de abajo. Tú necesitas presionar el boton que dice Branch: ***master***, digita gh-pages en el campo de texto, luego presiona el boton azul que dice Create branch: gh-pages. Esto crea una rama de código especial llamada gh-pages que es publicada en una ubicación especial. La URL toma la forma username.github.io/my-repository-name, asi en mi caso de ejemplo, la URL debería ser https://chrisdavidmills.github.io/my-repository. La página mostrada es la página index.html.  
   Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamente
2. Navega tu dirección web de GitHub pages en un nuevo ta del navegador, y tu deberias ver tu sitio en linea! Mandalo por correo electrónico a tus amigos y muestra tu dominio.

**Nota**: Si te atascas, la [página de inicio de GitHub Pages](https://pages.github.com/) también es muy útil.

### [Un mayor conocimiento de GitHub](https://developer.mozilla.org/es/docs/Learn/Common_questions/Using_Github_pages#un_mayor_conocimiento_de_github)

Si deseas realizar más cambios en su sitio de prueba y cargarlos en GitHub, simplemente tendrás que realizar el cambio en tus archivos como antes. A continuación, debes introducir los siguientes comandos (pulsando Intro después de cada uno) para empujar los cambios a GitHub:

git add --all

git commit -m 'another commit'

git push

Puedes reemplazar otro commit con un mensaje más adecuado para describir qué cambio acaba de hacer.