

# Tabla de Productos y Precios

## Módulo TyHM

Santiago Diez Juan Rivero Agustín Rodríguez Tomás Boismene Lucas Patiri  
Tomás Cerutti

Año 2024

### Resumen

Este documento presenta una tabla de productos y precios.

## 1. Tutorial básico de R

### 1.1. Asignación de variables

Puedes asignar valores a variables utilizando el operador `<-` o `=`

```
x <- 5  
y = 10
```

## 2. Operaciones aritméticas

```
suma <- 3 + 4  
resta <- 7 - 2  
multiplicacion <- 5 * 2  
division <- 10 / 2  
potenciacion <- 2^3
```

## 3. Vectores

### 3.1. Los vectores son una colección ordenada de elementos del mismo tipo.

```
vector_numerico <- c(1, 2, 3, 4, 5)  
vector_caracter <- c("a", "b", "c", "d", "e")
```

## 4. Gráficos

### 4.1. R es conocido por su capacidad para generar gráficos de alta calidad.

Puedes crear gráficos utilizando funciones como `plot()`, `hist()`, `barplot()`, entre otras.

## 5. Funciones

### 5.1. R tiene muchas funciones integradas que puedes usar para realizar tareas específicas.

Por ejemplo, la función `mean()` calcula la media de un conjunto de números.

```
numeros <- c(1, 2, 3, 4, 5)
media <- mean(numeros)
```

## 6. Data frames

### 6.1. Los data frames son estructuras de datos bidimensionales que se usan para almacenar conjuntos de datos.

Puedes crear data frames utilizando la función `data.frame()`.

```
edades <- c(25, 30, 35)
nombres <- c("Juan", "María", "Pedro")
df <- data.frame(Edad = edades, Nombre = nombres)
```

## 7. Indexación de vectores y data frames

### 7.1. Puedes acceder a elementos individuales de un vector o data frame utilizando corchetes `[]`.

```
primer_elemento <- vector_numerico[1]
nombre_primera_fila <- df[1, "Nombre"]
```

## 8. Imprimir la tabla

```
print(productos)
## Producto Precio
## 1 PC1 1000000
## 2 PC2 800000
## 3 PC3 1200000
## 4 PC4 950000
```

## 9. ¿Como pudimos implementar esta plantilla?

Utilizamos los archivos `arxiv.sty`, `references.bib` y `tecnicas.tex` para poder implementarlo. Dichos archivos se deben encontrar en la misma carpeta que este documento

Basicamente en TUTORIALR hicimos una gráfica y una tabla para nuestra empresa que nos permita comparar distintos productos que necesitamos. Ahora publicamos este archivo en Github y nos preguntamos...

## 10. ¿Que es Github?

GitHub es una plataforma en línea basada en la tecnología de control de versiones Git, que permite a los desarrolladores y equipos colaborar en proyectos de software desde cualquier lugar del mundo. Esencialmente, GitHub actúa como un repositorio en la nube para código fuente y otros tipos de archivos relacionados con el desarrollo de proyectos.

## 11. ¿Para qué sirve GitHub?

GitHub utiliza Git, un sistema de control de versiones distribuido, para ayudar a los equipos a manejar cambios en el código fuente a lo largo del tiempo. Esto permite que varios desarrolladores trabajen en el mismo proyecto sin interferir en el trabajo de los demás. Además de almacenar código, GitHub facilita el seguimiento de problemas, la colaboración, y la documentación de proyectos.

### 11.1. Funciones principales de GitHub

1. Control de versiones: Permite a los usuarios rastrear y comparar los cambios realizados en el código, recuperar versiones anteriores y gestionar distintas ramas de desarrollo dentro de un proyecto.
2. Colaboración: Los usuarios pueden trabajar en equipos, comentar sobre los cambios en el código, hacer sugerencias de mejoras y revisar el código de otros antes de que se integre al proyecto principal (master).
3. Gestión de proyectos: A través de funciones como 'issues' (problemas), 'milestones' (hitos) y 'boards' (tableros tipo Kanban), los equipos pueden organizar tareas, priorizar trabajo y seguir el progreso del proyecto.
4. Integración Continua/Despliegue Continuo (CI/CD): GitHub se integra con varias herramientas que permiten automatizar pruebas y despliegue de software, ayudando a los equipos a detectar errores temprano y gestionar despliegues de manera eficiente.

5. Documentación: Con GitHub Pages, los usuarios pueden crear sitios web para alojar la documentación de sus proyectos directamente desde los repositorios de GitHub.
6. GitHub Actions: Permite automatizar flujos de trabajo basados en eventos dentro del propio GitHub, como enviar notificaciones cuando alguien hace un 'push' de código o fusiona un 'pull request'.

## 11.2. ¿Por qué conviene usar GitHub?

1. Mejora la colaboración: Al estar basado en la nube, GitHub permite que los equipos colaboren en proyectos desde diferentes ubicaciones. Las herramientas de revisión de código y los 'pull requests' facilitan el trabajo colaborativo y la mejora continua del código.
2. Escalabilidad: Desde proyectos pequeños hasta grandes, GitHub puede manejar eficientemente la carga de trabajo, permitiendo a los desarrolladores centrarse en mejorar el software y no en los problemas de infraestructura.
3. Comunidad y visibilidad: Al ser la plataforma de hosting de código más popular, GitHub proporciona una gran visibilidad a los proyectos. Los desarrolladores pueden contribuir a proyectos de código abierto y ganar reconocimiento en la comunidad.
4. Recursos educativos: GitHub es también una herramienta educativa poderosa; muchas instituciones y cursos en línea lo utilizan para enseñar programación y desarrollo de software.
5. Integraciones y herramientas: GitHub se integra con una multitud de herramientas de terceros que amplían sus funcionalidades, desde sistemas de prueba automatizada hasta integraciones con servicios de nube y herramientas de desarrollo.

En resumen, GitHub no solo es útil para desarrolladores individuales, sino también esencial para la colaboración en equipo en proyectos de cualquier tamaño. Facilita el control de versiones, la gestión de proyectos y la colaboración continua, lo que lo convierte en una herramienta indispensable en el mundo del desarrollo de software.

<https://github.com/tomasboismene/modulo1>

<https://github.com/AIRodriguezz/modulo-2>