

Potenciar Argentina  
FullCoders: curso introductorio  
Comisión 3-21603  
Agosto-Septiembre 2023

---

# TRABAJO FINAL

---

Proyecto: "Suspiros de mujer"

Alumna/o: D'Angelo, Yanela - DNI 39.476.494

Tutor/a: Walter Randazzo

*"La clave para el éxito está en descomponer grandes tareas en pequeñas acciones y avanzar con determinación, recordando siempre que el progreso constante es el camino hacia la realización."*

*Anónimo*

---

El presente trabajo pretende simular la necesidad de organizar el stock de un local de indumentaria y zapatos pequeño, indicando cuales son los productos que se encuentran disponibles como aquellos que son vendidos, ordenados en base al modelo.



(Nombre y logo simulado con <https://www.canva.com/>)

En este proyecto introductorio, se presenta un diagrama de flujo general, un pseudocódigo que simula la carga de productos junto a sus características, su orden y disponibilidad. Además, se teoriza sobre el proceso de gestión del proyecto con las metodologías ágiles trabajadas en el curso, Kanban y Scrum.

Toda la documentación se encuentra en el siguiente repositorio GitHub:  
<https://github.com/yanelad/Organizaci-n-y-control-de-stock>

---

### 1) Descripción del proyecto y requisitos funcionales:

Un pequeño local de indumentaria y zapatos "suspiros de mujer" necesita un sistema de organización y control de stock, en donde conste cada talla, color y precio, además de si se encuentra disponible y en qué cantidad. Voy a necesitar una aplicación para lograr organizarlo.

Título: Sistema de Organización de stock para local / Suspiros de mujer.

El sistema debe proporcionar los siguientes requisitos funcionales:

#### Organización de Stock:

Permitir a los propietarios visualizar en forma inmediata la disponibilidad de productos en stock para brindar la más eficiente atención a sus clientes.  
Visualizar en el listado talla, color y precio.

#### Lista de Precios de Productos:

Mantener una lista de precios actualizada de los productos ofrecidos.  
Permitir la modificación y actualización de los mismos en forma conjunta, según sea necesario en proporción al índice inflacionario.

#### Salida de los productos del stock:

Permitir a los gestores del sistema retirar del stock aquellos productos que, por alguna razón, como podría ser su venta, ya no se encuentren disponibles.

#### Interfaz de Usuario Amigable:

Proporcionar una interfaz de usuario intuitiva y fácil de usar para que tanto los propietarios como nuevos vendedores puedan navegar y utilizar todas las funciones del sistema.

```
*** Ejecución Iniciada. ***
Ingrese el nombre del producto1
> Remera Barbie
Ingrese el talle del producto1
> M
Ingrese el color del producto1
> Rosa
Ingrese la cantidad Disponible en Stock del producto1
> 2
Ingrese el precio del producto1
> $12000
Ingrese el nombre del producto2
> Remera Minnie
Ingrese el talle del producto2
> S
Ingrese el color del producto2
> Rojo
Ingrese la cantidad Disponible en Stock del producto2
> 1
Ingrese el precio del producto2
> $15000
Ingrese el nombre del producto3
> Buzo Stitch
Ingrese el talle del producto3
> L
Ingrese el color del producto3
> Azul
Ingrese la cantidad Disponible en Stock del producto
> 4
Ingrese el precio del producto3
> $22000
Datos del Producto: 1
Codigo: codigo001
Nombre: Remera Barbie
Talle: M
Color: Rosa
Stock: 2
Precio: $12000
*****
```

```

Datos del Producto: 2
Codigo: codigo002
Nombre: Remera Minnie
Talle: S
Color: Rojo
Stock: 1
Precio: $15000
*****

Datos del Producto: 3
Codigo: codigo003
Nombre: Buzo Stitch
Talle: L
Color: Azul
Stock: 4
Precio: $22000
*****

Stock organizado!
¿Código de producto?
> codigo001
Stock: 2
*** Ejecución Finalizada. ***

```

## 2) Diseñar interfaz de usuario mediante la utilización de pseudocódigos en Pseint y diagramas de flujo.

### Algoritmo OrganizacionStock

```

// Defino 3 como cantidad de productos
Dimensionar nombreProducto(3)
Dimensionar talleProducto(3)
Dimensionar colorProducto(3)
Dimensionar stockInicialProducto(3)
Dimensionar precioProducto(3)
Dimensionar codigoProducto(3)
// PRIMERA PARTE: Ingreso de los productos (3)
Para nroProducto<-1 Hasta 3 Hacer
    Escribir 'Ingrese el nombre del producto', nroProducto
    Leer nombre
    nombreProducto[nroProducto] <- nombre
    Escribir 'Ingrese el talle del producto', nroProducto
    Leer talle
    talleProducto[nroProducto] <- talle
    Escribir 'Ingrese el color del producto', nroProducto
    Leer color
    colorProducto[nroProducto] <- color
    Escribir 'Ingrese la cantidad Disponible en Stock del producto',
nroProducto
    Leer stock
    stockInicialProducto[nroProducto] <- stock

```

```

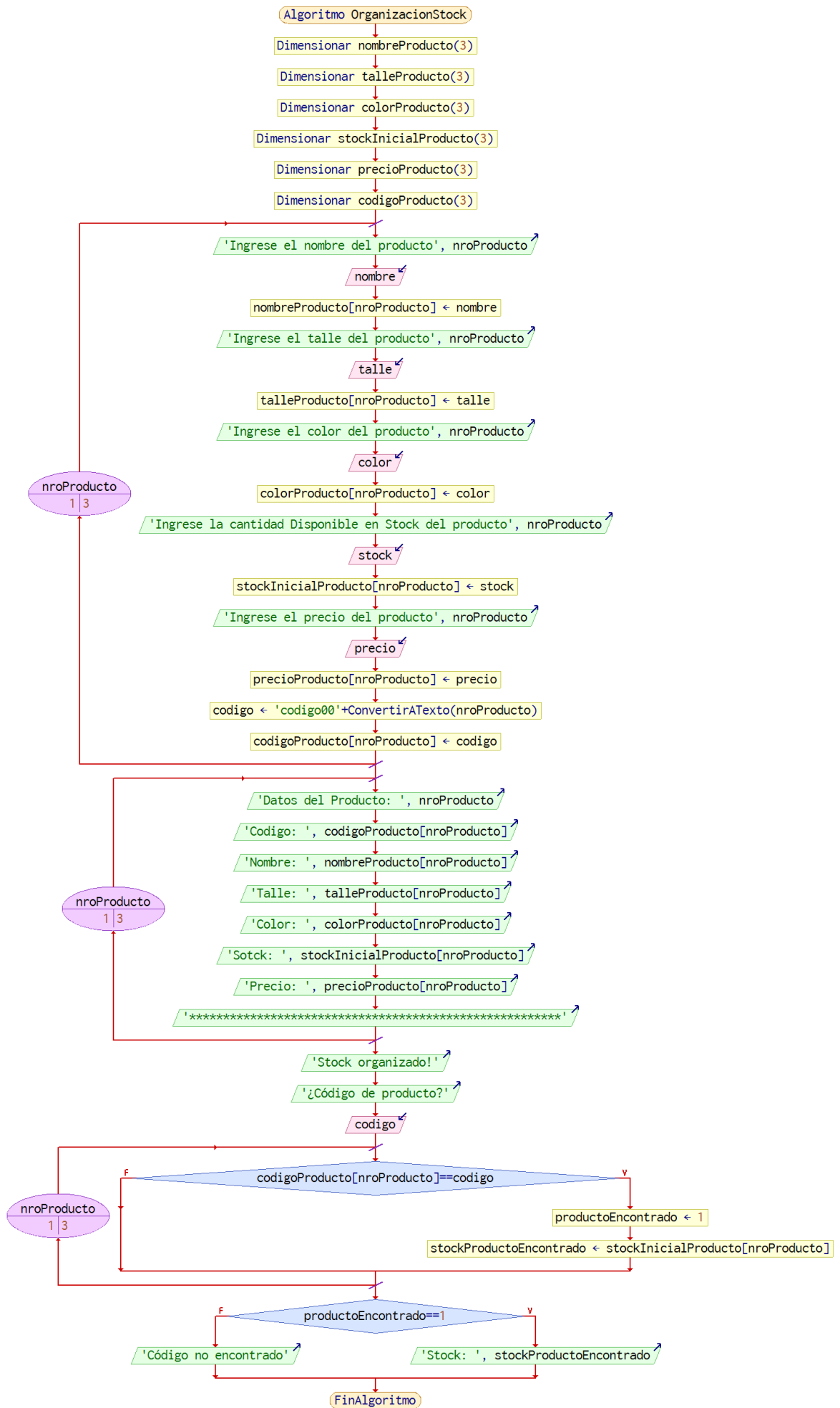
    Escribir 'Ingrese el precio del producto', nroProducto
    Leer precio
    precioProducto[nroProducto] <- precio
    // Asigno el código del producto
    codigo <- 'codigo00'+ConvertirATexto(nroProducto)
    codigoProducto[nroProducto] <- codigo
FinPara
// Imprimo los datos de los productos ingresados
Para nroProducto<-1 Hasta 3 Hacer
    Escribir 'Datos del Producto: ', nroProducto
    Escribir 'Codigo: ', codigoProducto[nroProducto]
    Escribir 'Nombre: ', nombreProducto[nroProducto]
    Escribir 'Talle: ', talleProducto[nroProducto]
    Escribir 'Color: ', colorProducto[nroProducto]
    Escribir 'Stock: ', stockInicialProducto[nroProducto]
    Escribir 'Precio: ', precioProducto[nroProducto]
    Escribir '*****'
FinPara
Escribir 'Stock organizado!'
// SEGUNDA PARTE: Consulta Disponibilidad de Stock
Escribir '¿Código de producto?'
Leer codigo
Para nroProducto<-1 Hasta 3 Hacer
    Si codigoProducto[nroProducto]==codigo Entonces
        productoEncontrado <- 1
        stockProductoEncontrado <- stockInicialProducto[nroProducto]
    FinSi
FinPara
Si productoEncontrado==1 Entonces
    Escribir 'Stock: ', stockProductoEncontrado
SiNo
    Escribir 'Código no encontrado'
FinSi

```

#### **FinAlgoritmo**

Si bien el algoritmo es correcto al momento de la ejecución, no se encuentra finalizado en relación al objetivo de este trabajo, queda aún desarrollar la salida de los productos del stock y la actualización de los precios.

A continuación, en la siguiente página, visualizamos el diagrama de flujo generado y vinculado al algoritmo de nuestro trabajo.



### 3) Metodologías ágiles utilizadas - Ejemplo Scrum:

Se caracteriza por ser la «**metodología del caos**» que se basa en una estructura de **desarrollo incremental**, esto es, cualquier ciclo de desarrollo del producto y/o servicio se desgrana en «pequeños proyectos» divididos en distintas etapas: análisis, desarrollo y testing. En la etapa de desarrollo encontramos lo que se conoce como interacciones del proceso o **Sprint**, es decir, entregas regulares y parciales del producto final.

Las **reuniones** son el pilar fundamental de la metodología, donde diferenciamos entre: reuniones de planificación, diaria, de revisión y de retrospectiva, la más importante de todas ellas.

Product Backlog: contiene una **lista** de todas las tareas específicas, ordenadas por importancia.

- +Investigar y diseñar la interfaz del sistema.
- +Generar un pseudocódigo funcional.
- +Carga de productos.
- +Carga de talles y colores.
- +Lista de precios de productos.
- +Salida de productos.

Sprint Planning: El primer sprint se prolongará durante un mes, ya que investigar y diseñar la interfaz del sistema, además del crear el pseudocódigo son las partes más importantes. Al resto le dedicaremos sprint de dos semanas.

Daily Standup: Realizare reuniones diarias de no más de quince minutos conmigo misma para discutir el progreso:

Ayer: Trabajé en el diseño de la interfaz del sistema.

Hoy: Continuaré con la estructura de la base de datos.

Obstáculos: Necesito investigar cómo crear un repositorio para almacenar la información en GitHub.

Sprint Review y Retrospectiva: Al final del Sprint, reviso lo que logre hacer (registro de productos) y de ser necesario habrá que ajustar la planificación para el próximo Sprint.

Reflexiona sobre lo que funcionó y lo que se puede mejorar:

Funcionó: Logré completar la funcionalidad de registro de productos.

Mejoras: Necesito mejorar mi investigación sobre cómo podrá mi cliente actualizar en forma total los precios de los productos en proporción a la inflación.

### 4) Metodologías ágiles utilizadas - Ejemplo Kanban:

Consiste en la elaboración de un cuadro o diagrama en el que se refleja tres columnas de tareas; "pendientes, en proceso o terminadas". Incluso podemos agregar la columna de "en revisión", las cuales ayudan a mejorar la productividad y eficiencia del equipo de trabajo. Puede ser gestionado mediante la aplicación gratuita "Trello".

#### Listado de Tareas:

- +Investigar y diseñar la interfaz del sistema.
- +Generar un pseudocódigo funcional.
- +Implementar la funcionalidad del registro de productos.
- +Carga de talles y colores de cada producto.
- +Diseñar la lista de precios de productos.
- +Implementar la funcionalidad de salida de productos.

Pendientes	En proceso	Terminadas
		Investigar y diseñar la interfaz del sistema.
		Generar un pseudocódigo funcional.
		Implementar la funcionalidad del registro de productos.
	Carga de talles y colores de cada producto.	
Diseñar la lista de precios de productos.		
Implementar la funcionalidad de salida de productos.		

#### Flujo de Trabajo:

Colocamos, en principio, todas las tareas en la columna "Pendiente" del tablero Kanban.

A medida que comenzamos a trabajar en una tarea, la movemos a la columna "En proceso".

Cuando completamos una tarea, la movemos a la columna "En revisión" para verificarla y resolver cualquier problema.

Una vez que una tarea haya sido revisada y aprobada, la movemos a la columna "Terminada".

#### Seguimiento:

Actualizaremos el tablero Kanban regularmente, asegurándonos de que refleje el progreso. Podemos crear una columna "Bloqueado" para indicar qué tareas están trabadas y por qué razón.

#### **5) Crear un repositorio público en GitHub:**

<https://github.com/yanelad/Organizaci-n-y-control-de-stock>