

CAPÍTULO 4

MATLAB

4.1 Conceptos Básicos de Matlab

Matlab es creado por The MathWorks, el cual es un idioma de alto rendimiento para la informática técnica. Integra cómputo, visualización, y programación en un ambiente de fácil uso. Los usos típicos incluyen:

- ❖ Cálculos Matemáticos
- ❖ Desarrollo de algoritmos de cómputo
- ❖ Análisis de datos, visualización y diseño de aplicaciones gráficas, incluyendo interfases gráficas.

El nombre de Matlab proviene de matriz laboratory, por el que fue escrito por su fácil uso para el trabajo con matrices, por eso es usado mucho en industrias como en universidades.

4.1.1 Historia de Matlab

Matlab ha evolucionado mucho, sobre todo en el diseño de interfases, ya que su programación es muy sencilla y amigable, como se verá más adelante. Los creadores de The MathWorks Jack Little y Cleve Moler, observaron dichas necesidades, decidieron crear y desarrollar MATLAB, un lenguaje con mejores características que el FORTRAN o C, que representan los programas de cálculo matricial más avanzados y así un acceso sencillo al software matricial. ^[6]

El sistema de Matlab consiste de cinco partes principales, las cuales son:

- ✓ **Herramientas de Escrito y Desarrollo de Ambientes.** Herramientas de Desktop y Ambiente de Desarrollo. Éste es el juego de herramientas y medios que ayudan al uso de las funciones y archivos de MATLAB. En el escritorio de Matlab Incluye Ventana de Orden, un historial de orden, editor, un analizador del código y otros informes, y navegador para ver ayuda, los workspace, archivos, y el camino de la búsqueda.

- ✓ **Librería de Funciones Matemáticas.** Ésta es una inmensa colección de algoritmos computacionales que van de las funciones elementales, como la suma, seno, coseno, y la aritmética compleja, las funciones más sofisticadas como la matriz inversa, función Bessel, transforma rápida de Fourier.

- ✓ **Lenguaje de Matlab.** Este es un idioma de serie de matriz de alto nivel con declaraciones de flujo de mando, funciones, estructura de datos, entrada y salida, y programación orientada a rasgos de objetos. Esto permite una programación pequeña, la cual consiste en crear rápidamente programas sin mayor complicación, también permite programar en grande, el cual es crear aplicaciones complejas.

- ✓ **Gráficos.** Matlab tiene medios extensos para desplegar vectores y matrices como gráficos, así como anotando e imprimiendo estos gráficos. Incluye funciones de alto nivel para la visualización de datos de dos y tres dimensiones,

procesamiento de imágenes, animación y gráficas de presentación. También incluye funciones de bajo nivel, que permiten personalizar la apariencia de gráficos que le permiten al usuario complementar sus aplicaciones de Matlab como lo son las interfaces gráficas.

- ✓ **Interfases externas API de Matlab.** Ésta es una biblioteca que le permite escribir programas en C y programas de Fortran que actúan recíprocamente con MATLAB. Incluye medios por llamar rutinas de MATLAB (vinculación dinámica), llamando a MATLAB como un artefacto computacional, y por leer y escribir archivos MAT.

4.1.2 Plataforma Base y Archivos M

La plataforma base que se ocupó para realizar este proyecto es en Windows XP, el cual se da doble clic en el icono de Matlab para poder inicializar. También se puede ocupar la plataforma para UNIX, el cual se inicializa Matlab escribiendo `mat lab` en el sistema operativo. En la siguiente figura observamos la ventana principal de Matlab.

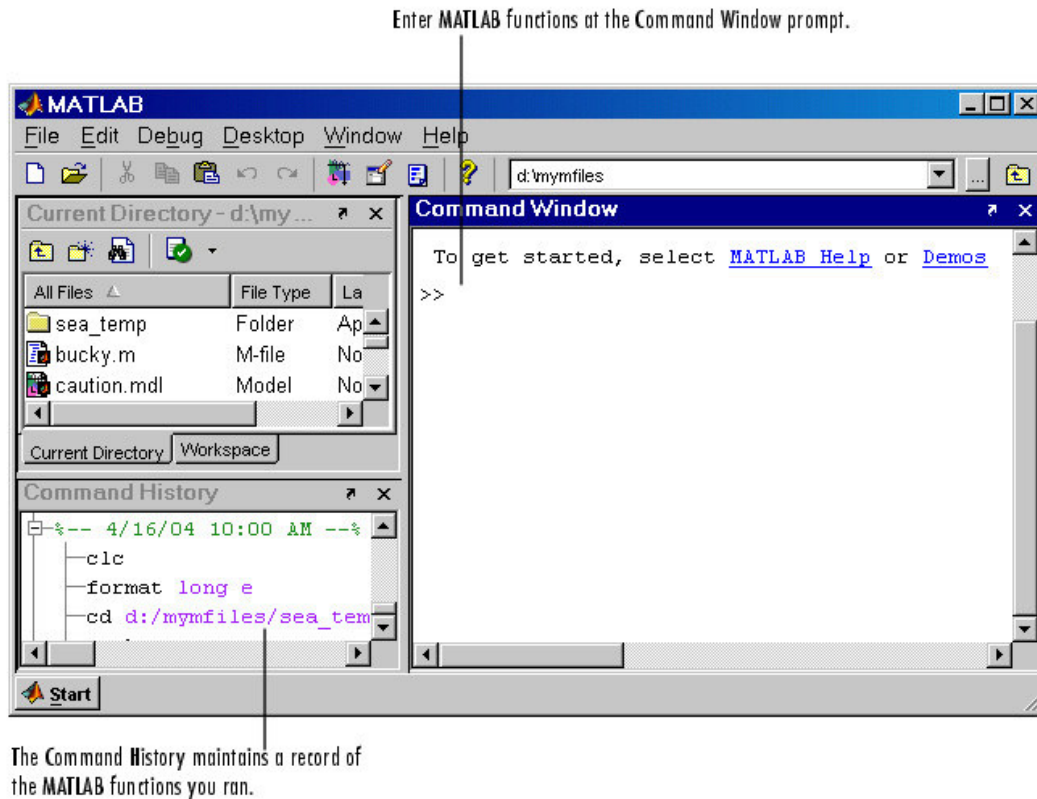


Figura 4.1 Ventana principal de Matlab

Se puede crear una propia matriz que usa M-archivos, que son archivos de texto que contiene el código de MATLAB. Se usa el Editor de MATLAB u otro editor de texto para crear un archivo que contenga las mismas declaraciones que quisieras declarar en MATLAB usando el comando línea. Se guarda el documento con un nombre que termine en `.m`.

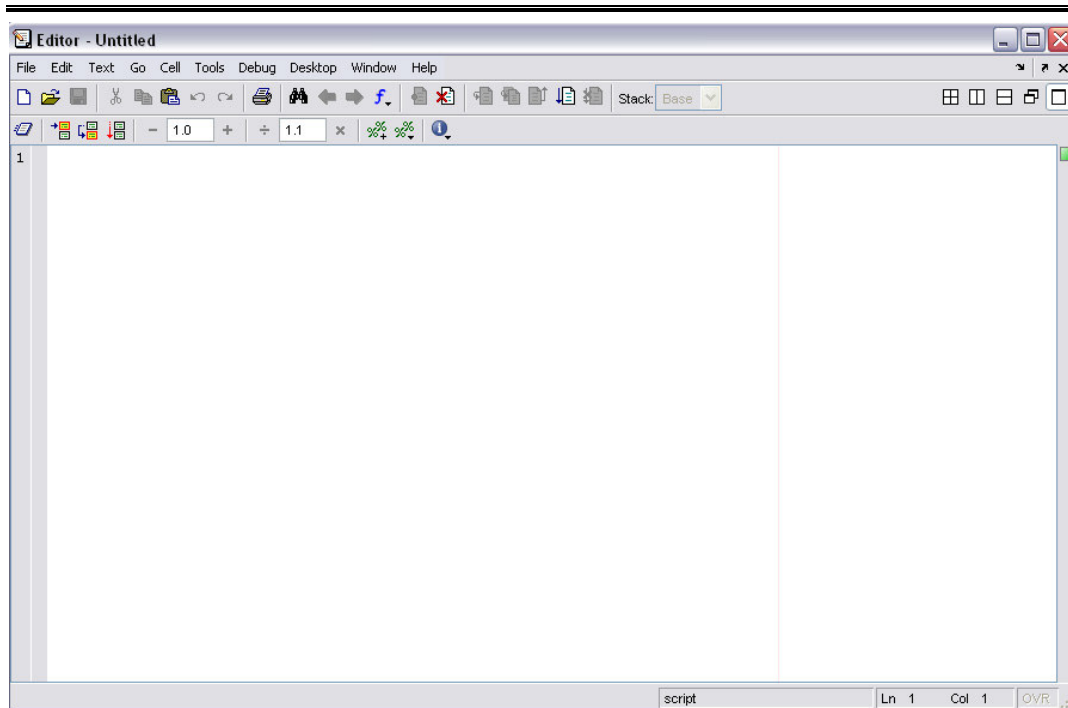


Figura 4.2 Editor de Matlab

4.1.3 Operaciones Básicas

Las operaciones básicas que realiza Matlab, son las operaciones que conocemos, como lo son la suma, resta, multiplicación, división y potencia, por mencionar otras. El método de programación de Matlab, es empleando los símbolos de cada operación, escribiéndolas de izquierda a derecha.

Dichas operaciones se pueden escribir en la pantalla principal de Matlab o bien en el editor, es recomendable usar el editor ya que en este puede guardar el programa y hacer modificaciones en él con mayor facilidad.

4.2 Diseño de Interfase GUI's

Las interfases gráficas, son una herramienta para crear programas que le sean muy amigables al usuario. Una interfase de usuario gráfica (GUI) es un gráfico que despliega el contenido de dispositivos, o componentes que le permiten a un usuario que realice tareas interactivas. Para realizar estas tareas, el usuario de la GUI no tiene que crear una escritura u órdenes de tipo línea de orden. A menudo el usuario no tiene que saber los detalles de la tarea a mano.

4.2.1 Principio de Diseño

Por lo general no existe un procedimiento para el diseño de interfases gráficas, como si fuera una receta de cocina, pero se recomiendan opciones para poder diseñar las interfases gráficas, por lo que se sugiere el siguiente método.

La fase de diseño comprende tres pasos los cuales son: ^[7]

- Definición de la tarea.
- Dibujar la GUI sobre papel.
- Prueba del diseño.

La fase de implementación: ^[7]

- Escritura del código en Matlab.
- Prueba del código.

4.3 Panel de Control

Una vez que se diseñó un prototipo de la interfase en el papel, éste se pasa a Matlab el cual se ocupa la herramienta *guide*, la cual ayudará mucho para el diseño de interfases. Para inicializar *guide*, se tiene que estar en el directorio en el que se va a guardar la interfase, escribiendo *guide*.

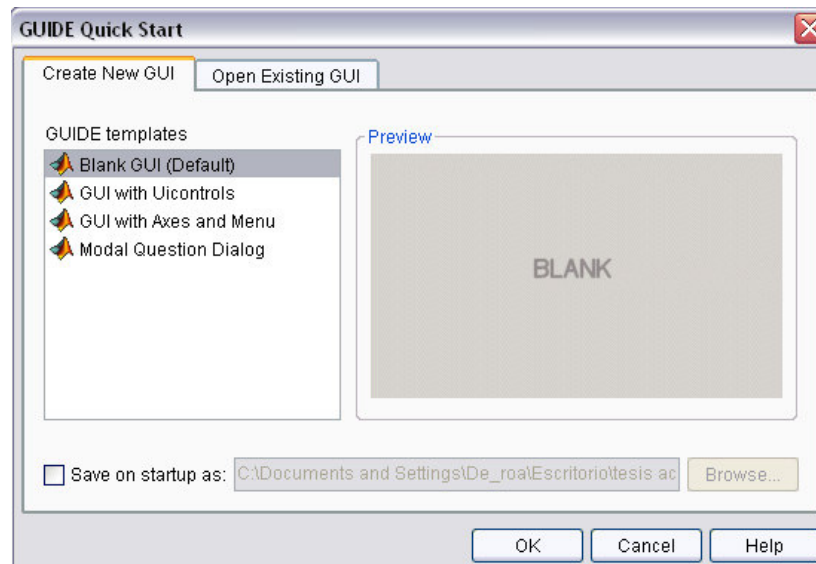


Figura 4.3 Inicialización de Interfases

Como se muestra en la figura 4.3, se puede empezar abriendo el panel de control para empezar una interfase nueva, también tiene ejemplos de interfases para darse una idea del diseño de ellas. Por lo que se escoge la primera opción para diseñar nuestras interfases, como se muestra en la figura 4.4.

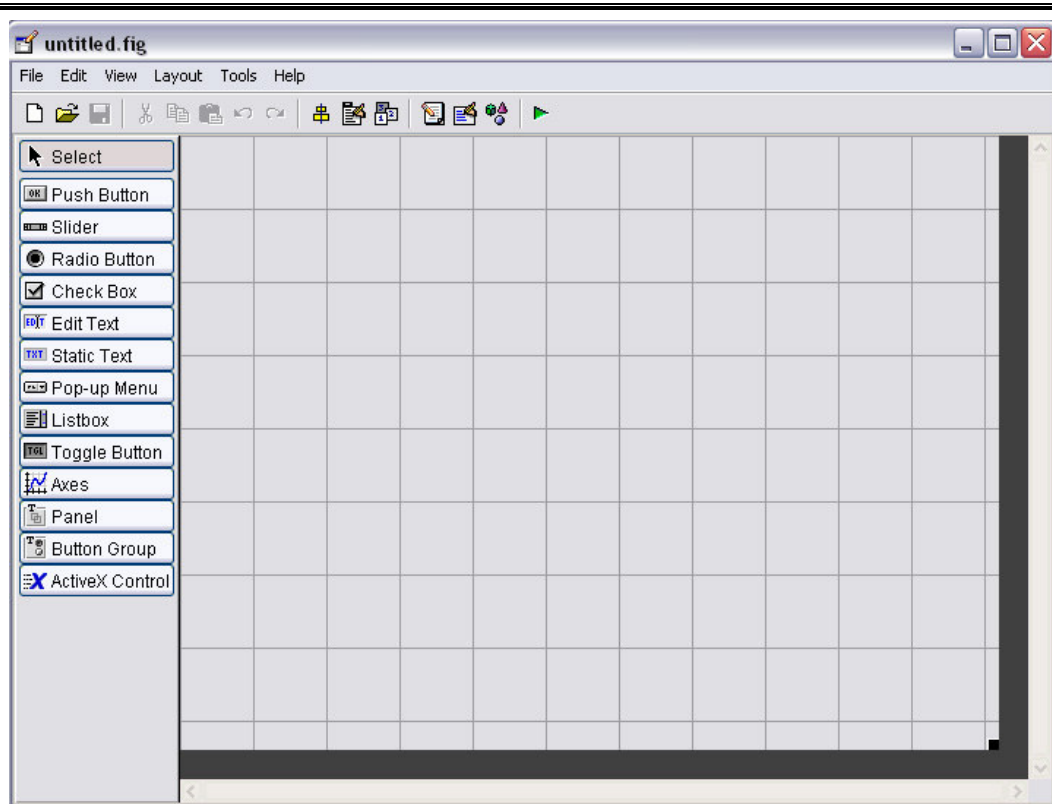


Figura 4.4 Panel de Control

En el panel de control, se tienen varias herramientas que ayudarán a crear la interfase, ya que es una herramienta fácil de usar y es muy amigable, ya que cuando se está creando una interfase, todo es gráfico, como poner botones o cajas de texto, entre otras opciones. A continuación se explican un poco las herramientas de la paleta que se encuentra en el panel de control de la figura 4.5.

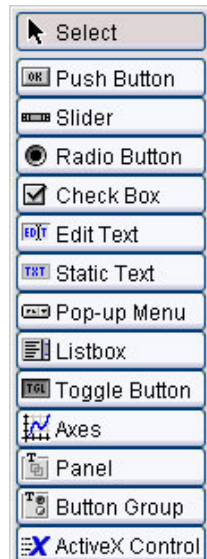


Figura 4.5 Paleta de Herramientas

Estas herramientas son programables por medio de callback (las funciones que se ejecutan cuando el usuario activa algún componente en la GUI). A continuación se describen algunos controles de la paleta de herramientas que se utilizaron para la creación de las interfases.

- **Push Button:** Estos generan una acción al ser presionados. Cuando se hace clic en un botón de comando, ejecuta una acción programada. En el editor de propiedades (figura 4.6) se puede cambiar la apariencia y texto predefinido, así como la opción Tag, que es una subfunción callback en el código en el que se está programando, esta opción ayuda a llamar resultados o realizar operaciones cuando se ejecuta el Push Button.

- **Edit Text:** Es un control en el que el usuario no puede modificar. Se utiliza cuando la interfase se desea modificar o ingresar texto o datos. En el caso de Static Text, el usuario no lo puede modificar.

- **Listbox:** Despliega una lista y permite al usuario seleccionar algún elemento de la lista.
- **Axes:** Con esta herramienta, en los ejes podemos desplegar gráficas o imágenes, podemos controlar los objetos gráficos y sus características. No es un unicontrol, pero se puede programar al ejecutar su callback, cuando el usuario haga clic sobre los ejes. Cuando se utilizan varios ejes, debemos seleccionar explícitamente cual de los ejes deseamos utilizar para graficar.

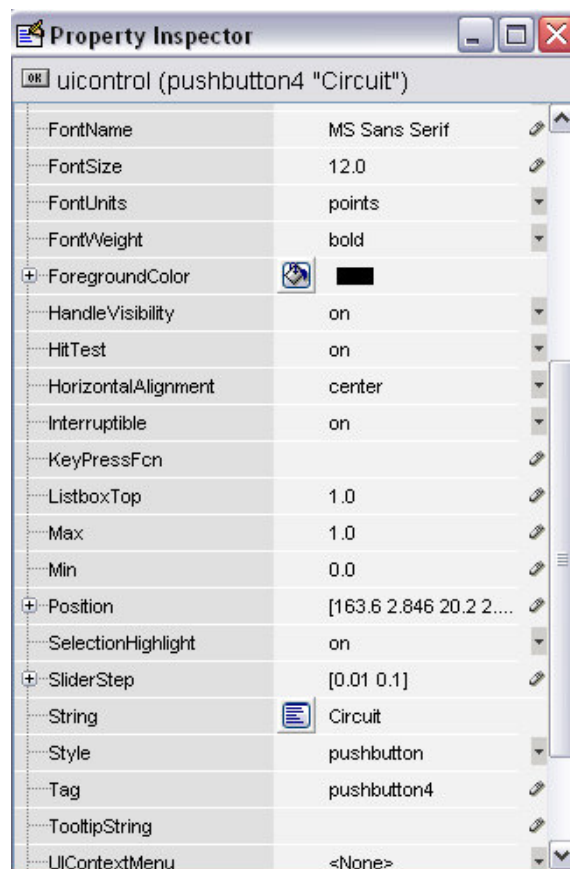


Figura 4.6 Editor de Propiedades.

También se utilizó *Align Objects*, el cual sirve para alinear los objetos de la interfase para que no tenga una mala apariencia.

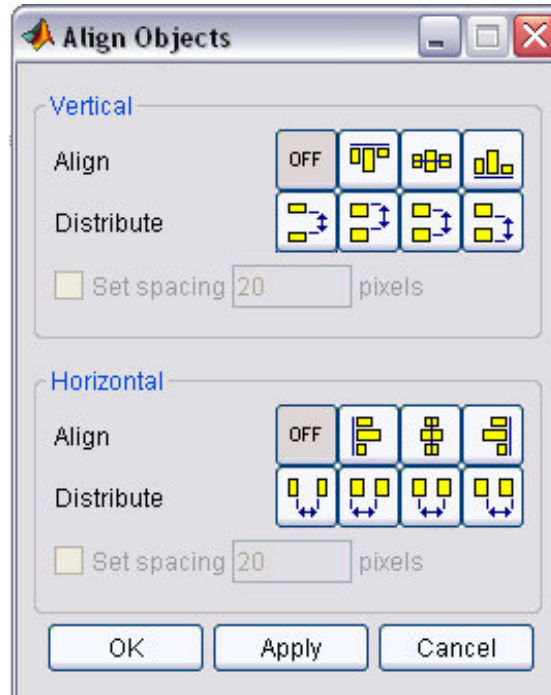


Figura 4.7 Alineación de Objetos

4.4 Prueba del Código

Una vez que se diseñó la interfase con ayuda de *GUIDE (GUI Development Environment)*, entrando al panel de control, en el que se diseñó colocando, botones, ejes, cajas de lista, espacios de edición, etc. Ahora se realiza la programación de cada uno de ellos para que ejecute el diseño, posteriormente se realiza la prueba del programa para comprobar la interfase.