UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA

UNAN - MANAGUA

Facultad de Ciencias e Ingeniería

Departamento de Tecnología

Propuesta de sistema operativo de entorno gráfico que funciona en la tarjeta de desarrollo Arduino Mega como un producto de la empresa UNITRIUM

Seminario de Graduación como requisito final para optar al título de Ingeniero Electrónico

Autor/es:

Ing. Víctor Vladimir Córdoba Tenorio
Br. Diana Alexandra Maltéz Aburto

Tutor:

Msc. Ing. Milciades Delgadillo

Managua, diciembre 2022

DEDICATORIAS

A Dios, nuestro padre celestial creador de toda sabiduría y ciencia por su infinita misericordia.

A mi madre y hermanas, por brindarme su apoyo incondicional y fortaleza para seguir adelante, por sus consejos y por ser un gran ejemplo a seguir.

Finalmente, a todos los docentes que me han formado a lo largo de mi vida académica que desinteresadamente han aportado a mis conocimientos.

Diana Alexandra Maltéz Aburto

Dedico este documento a la mujer que acaricia mi alma en las noches de perpetua soledad y silencio, ella que siempre está conmigo incluso en los momentos de sufrimiento, ella que con su voz empuja sus palabras de aliento en mi oído, ella que levanta mi espíritu y me da fuerzas para seguir adelante, ella que es mi amanecer hasta el fin de mi ocaso, ella que es la creación y la conclusión de todo, ella que es la razón de mi existir, la música.

Víctor Vladimir Córdoba Tenorio

AGRADECIMIENTOS

Agradecemos primeramente a Dios, por habernos dado la sabiduría, inteligencia y fortaleza para poder dar todos los pasos en nuestras vidas.

A nuestras familias que tanto nos han apoyado en momentos duros y han ayudado a que sigamos en el camino de nuestro desarrollo profesional.

A nuestros maestros que, en todos éstos años, se han dedicado a formarnos profesionalmente.

A nuestros amigos, que pese a la distancia nos han brindado el apoyo necesario para nunca abandonar los objetivos y metas que nos hemos propuesto.

RESUMEN

Este proyecto se realizó en la empresa UNITRIUM el cual consistió en el desarrollo de un sistema operativo de entorno gráfico, en donde se utilizaron las técnicas electrónicas que se adquirieron en la carrera de ingeniería electrónica en la Universidad Nacional Autónoma de Nicaragua - UNAN-Managua.

Se diagnosticó mediante una entrevista para identificar los requisitos deseados del sistema operativo de entorno gráfico como producto de la empresa UNITRIUM que se dedica a ofrecer servicios de desarrollo de proyectos en el área de electrónica, como diseño de circuitos electrónicos, programación de tarjetas de desarrollo, microcontroladores y microprocesadores. Por medio de estos requisitos se hizo un diseño de un circuito con la tarjeta de desarrollo y con los demás dispositivos, se realizó la programación del sistema operativo usando el software Arduino IDE utilizando el lenguaje C++.

Para validar el funcionamiento del sistema operativo se realizaron pruebas y comprobaciones para hacer una comparación entre los requisitos establecidos por la empresa UNITRIUM y los resultados del sistema operativo. Esto nos permitió validar de que se puede implementar un sistema operativo de entorno gráfico en tarjetas de desarrollo que tienen bajos recursos como lo es Arduino Mega. Además de que se puede implementar esto en industrias, específicamente en sistemas de interfaz hombre – máquina HMI y manufactura asistida por computadora CAM.

INDICE

INDICE DE FIGURAS

INDICE DE TABLAS

INDICE DE ILUSTRACIONES EN ANEXOS

Página

1. INTRODUCCIÓN

En este documento se presenta una propuesta de un sistema operativo de entorno gráfico que funciona en la tarjeta de desarrollo Arduino Mega como un producto tecnológico de la empresa UNITRIUM que ofrece servicios profesionales de desarrollo de proyectos en las áreas de electrónica y de software. La empresa UNITRIUM solicitó un sistema operativo de entorno gráfico que tenga capacidad de ser configurado mediante instrucciones y ejecute operaciones.

La creación de un sistema operativo de entorno gráfico puede ser empleado por desarrolladores de sistemas de interfaz hombre – máquina HMI con tarjetas de desarrollo para el control de maquinarias y de procesos industriales. También esto facilitará que las tarjetas de desarrollo Arduino Mega puedan ser integrados en proyectos complejos y avanzados por usuarios de todo el mundo, específicamente quienes ya tengan experiencia previa.

Se realiza un diagnóstico mediante una entrevista al gerente de operaciones de la empresa UNITRIUM para identificar los requisitos del sistema operativo indicando así las características técnicas, visuales, funcionales y operativas para su correcto funcionamiento.

Con la información obtenida del diagnóstico se diseña un sistema operativo y se programa haciendo uso del software Arduino IDE y el lenguaje de programación C++ para satisfacer los requisitos del mismo.

Se valida el funcionamiento del sistema operativo introduciendo instrucciones y con los resultados obtenidos se compara con los requisitos establecidos por la empresa UNITRIUM, si coinciden se confirma que el sistema operativo de entorno gráfico funciona correctamente en la tarjeta de desarrollo Arduino Mega.

2. ANTECEDENTES

A partir del avance tecnológico en el área de semiconductores por Julius Lilienfeld, la fabricación de la primera computadora Z1 de Konrad Zuse y la invención del primer transistor en 1947 por William Shockley, John Bardeen y Walter Brattain de Laboratorios Bell, se desarrollaron una serie de computadoras de primera generación en décadas posteriores por otras empresas como IBM.

El primer sistema operativo fue desarrollado en 1956 por Robert Patrick trabajador de General Motors y Owen Mock de North American Aviation para una computadora IBM 704 el cual lleva por nombre GM-NAA I/O, básicamente lo que hacía era ejecutar un programa cuando finalizaba otro. Los sistemas operativos en ese tiempo eran de tipo monoproceso, luego aparecieron los de multiprogramación, tiempo compartido, tiempo real, los multiprocesos y por último de entorno gráfico.

Según el instituto Smithsoniano dice que los ingenieros de Texas Instruments Gary Boone y Michael Cochran lograron crear el primer microcontrolador, denominado TMS 1000 en el año 1971, fue comercializado hasta 1974 y combinaba memoria ROM, RAM, microprocesador y reloj, destinado a los sistemas embebidos.

Las tarjetas de desarrollo de Arduino se originan de un proyecto de Massimo Banzi en 2005 con la idea de fabricar unas tarjetas de desarrollo de bajo costo para el uso interno del instituto de diseño interactivo de Ivrea en Italia. Actualmente se utilizan en muchos proyectos de electrónica a nivel mundial.

En el presente no existe un sistema operativo de entorno gráfico que funcione con las tarjetas de desarrollo de Arduino, aunque existen programas como MyOpenLab o LabView para sistemas operativos Windows y Linux para establecer una interfaz hombre-máquina o HMI, sin embargo, no son sistemas operativos y no se puede implementar sistemas complejos con instrucciones y ejecución de operaciones.

3. PLANTEAMIENTO DEL PROBLEMA

En la actualidad existen sistemas electrónicos con interfaz hombre – máquina para control de máquinas y control estadístico de procesos industriales, pero son equipos electrónicos de alto costo que generalmente los venden grandes empresas los cuales están diseñados para dar solución a problemas específicos. Sin embargo, estos equipos electrónicos carecen de compatibilidad o no se pueden reprogramar o implementar con otros dispositivos cuando se requiere, por lo tanto, no se pueden realizar cambios en tiempo real.

En el comercio de dispositivos electrónicos existen microcontroladores y tarjetas de desarrollo que pueden ser programados, pero en muchas ocasiones no hay información, ni personal capacitado para programarlos por la dificultad de algunos lenguajes de programación, además que, si hay soluciones resuelven problemas muy específicos y no hay versatilidad de adaptarse o ser compatibles con otros dispositivos.

Por lo cual, la empresa UNITRIUM ha visto la necesidad de que muchos microcontroladores y tarjetas de desarrollo que existen en el mercado no cuentan con un software como un sistema operativo de entorno gráfico con el cual se podrían realizar proyectos complejos como sistemas HMI para que un operario pueda controlar una máquina mediante instrucciones o implementar manufactura asistida por computadora en 2 dimensiones (CAM 2D) o en 3 dimensiones (CAM 3D).

Entonces, ¿se debe seguir trabajando de la misma forma comprando equipos caros para adaptarlos al control de maquinaria o control de procesos industriales o desarrollar sistemas operativos que sean propios, también versátiles y configurables con tarjetas de desarrollo menos costosas y que cumplan con los requisitos de las operaciones dentro de procesos industriales?

4. JUSTIFICACIÓN

El propósito de desarrollar un sistema operativo de entorno gráfico para una tarjeta de desarrollo que sea de bajo costo es que pueda facilitar la forma en que se visualiza la información y sea fácilmente entendible para un operador o usuario, además de tener la capacidad de poder configurarlo con instrucciones en tiempo real con lo que se puede adaptar y ser compatible con muchos dispositivos electrónicos.

Se desarrollará el sistema operativo en la tarjeta de desarrollo Arduino Mega, que tiene amplio uso a nivel nacional e internacional por muchos programadores, desarrolladores y estudiantes.

La empresa UNITRIUM mediante un sistema operativo de entorno gráfico reconoce que puede ofrecer soluciones a pequeñas y medianas empresas para que puedan emprender un proceso de automatización de sus servicios en la parte operativa de sus máquinas, llevar un control estadístico en sus procesos de fabricación de productos y manufactura asistida por computadora (CAM).

Con el desarrollo de éste sistema operativo de entorno gráfico se podrá realizar sistemas más complejos que antes no se podían crear con la tarjeta de desarrollo Arduino Mega, ya que éstos se han usado de forma básica.

Se puede utilizar una tarjeta de desarrollo menos costosa con un sistema operativo como si fuese una computadora que acepta instrucciones o comandos y ejecuta operaciones y se puede configurar en tiempo real. Dentro de una empresa se puede utilizar éste sistema operativo de entorno gráfico en muchos procesos críticos y reducir costos.

5. OBJETIVOS

5.1. OBJETIVO GENERAL

Desarrollar un sistema operativo de entorno gráfico con la tarjeta de desarrollo Arduino Mega como un producto tecnológico de la empresa UNITRIUM.

5.2. OBJETIVOS ESPECIFICOS

- Realizar un diagnóstico mediante una entrevista al gerente de operaciones de la empresa UNITRIUM para identificar los requisitos del sistema operativo.

- Diseñar un sistema operativo de entorno gráfico a través de la programación en el software Arduino IDE y usando el lenguaje de programación C++ para satisfacer requisitos establecidos en el diagnóstico.

- Validar el funcionamiento del sistema operativo de entorno gráfico por medio de una comparación con los requisitos establecidos por la empresa UNITRIUM.

6. FUNDAMENTOS TEÓRICOS

6.1. Los microcontroladores. Definición.

Son circuitos integrados de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador y que contiene todos los componentes fundamentales de un ordenador, aunque de limitadas prestaciones y que se suele destinar a gobernar una sola tarea. En su memoria sólo reside un programa que controla en funcionamiento de una tarea determinada, sus líneas de entrada/salida se conectan a los sensores y actuadores del dispositivo a controlar y, debido a su pequeño tamaño, suele ir integrado en el propio dispositivo al que gobierna. (Sánchez, 2013)

6.1.1. Historia de los microcontroladores.

El primer microcontrolador fue inventado por dos ingenieros de Texas Instruments, de acuerdo con el Instituto Smithsoniano, ellos son Gary Boone y Michael Cochran crearon el TMS 1000, el cual era un microcontrolador de 4 bits con función de ROM y RAM. El microcontrolador era utilizado internamente en Texas Instruments en sus productos de cálculo desde 1972 hasta 1974, y fue refinado con el paso de los años. (Petersen, s.f).

En 1974, TI puso a la venta el TMS 1000 para la industria de electrónicos. El TMS 1000 estuvo disponible en varios tamaños de RAM y ROM. A partir de 1983, cerca de un millón de TMS 1000 fueron vendidos. (Petersen, s.f)

Además de producir el primer microprocesador, Intel también ha desarrollado muchos microcontroladores importantes, dos de los cuales son el 8048 y el 8051. Introducido en 1976, el 8048 fue uno de los primeros microcontroladores de Intel y fue utilizado como el procesador en el teclado de la computadora personal de IBM.

Se estima que más de mil millones de dispositivos del 8048 se vendieron. El 8051 siguió en 1980 y se convirtió en una de las familias de microcontroladores más populares. Las variaciones de la arquitectura del 8051 se siguen produciendo hoy en día, por lo que el 8051 es uno de los diseños electrónicos más longevos de la historia. (Petersen, s.f).

Durante la década de 1990, los microcontroladores tenían memorias ROM (EEPROM) que se pueden programar y borrar eléctricamente, como las memorias flash que aparecieron en el mercado. Estos microcontroladores pueden ser programados, borrarse y volverse a programar utilizando sólo señales eléctricas. Antes de los dispositivos eléctricamente reprogramables, los microcontroladores a menudo necesitaban de programación especializada y hardware para borrar, lo que requería que el dispositivo se quitara del circuito, frenando el desarrollo de software y haciendo el esfuerzo más costoso. (Petersen, s.f).

Cuando se eliminó esta limitación, los microcontroladores fueron capaces de ser programados y reprogramados mientras que en un circuito los dispositivos con microcontroladores podían ser actualizados con el nuevo software, sin tener que ser devueltos al fabricante. Muchos microcontroladores actuales, tales como los disponibles de Microchip y Atmel, incorporan la tecnología de memoria flash. (Petersen, s.f)

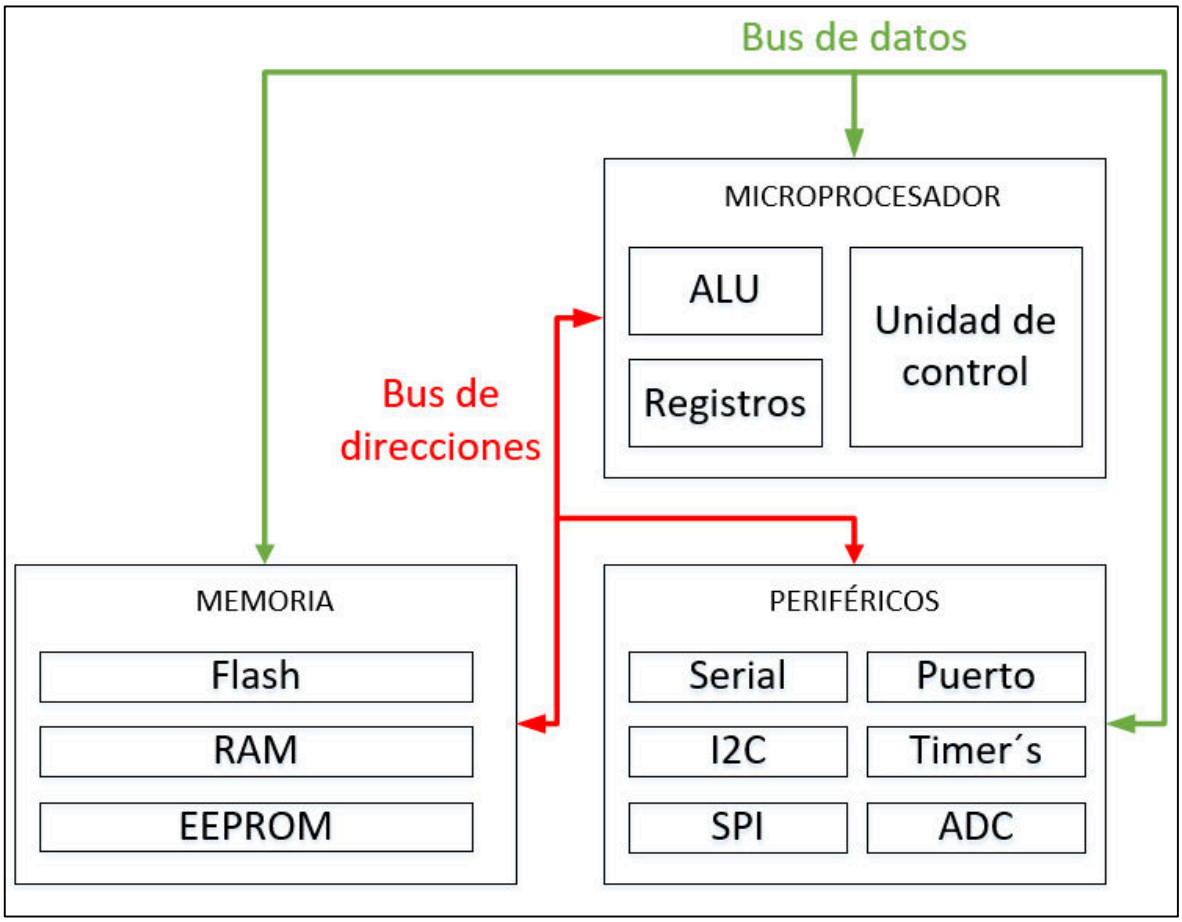
Además de los dispositivos de propósito general, los microcontroladores especializados se están utilizando en sectores como el automotriz, iluminación, comunicaciones y dispositivos de consumo de baja potencia. Los microcontroladores también se han vuelto más pequeños y más potentes. Por ejemplo, en 2010, Atmel anunció un microcontrolador flash en un paquete que medía 2 mm por 2 mm. Estos pequeños microcontroladores son lo suficientemente pequeños y baratos para ser utilizados en productos tales como juguetes y cepillos de dientes. (Petersen, s.f)

6.1.2. Estructura básica del microcontrolador.

El microcontrolador es una es un circuito integrado y posee una integración de estructuras funcionales y se encuentra integrado con su procesador (CPU), buses, memoria, periféricos y puertos de entrada y salida. Fuera del encapsulado se ubican otros circuitos para completar periféricos internos y dispositivos que pueden conectarse a los pines de entrada/salida. También se conectarán a los pines del encapsulado la alimentación, masa, circuito de completamiento del oscilador y otros circuitos necesarios para que el microcontrolador pueda trabajar. (Hernández, 2019).

Figura 1.

Estructura de un microcontrolador.



Fuente propia.

6.1.3. Características del microcontrolador.

Son diseñados para reducir el costo económico y el consumo de energía de un sistema en particular. Por eso el tamaño de la unidad central de procesamiento, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación. El control de un electrodoméstico sencillo como una batidora utilizará un procesador muy pequeño (4 u 8 bits) porque sustituirá a un autómata. (witronica.com, 2016).

Los microcontroladores pueden encontrarse en casi cualquier dispositivo electrónico como automóviles, lavadoras, hornos microondas, teléfonos, etc. Un microcontrolador difiere de una unidad central de procesamiento normal, debido a que es más fácil convertirla en una computadora en funcionamiento, con un mínimo de circuitos integrados externos de apoyo. (witronica.com, 2016).

La idea es que el microcontrolador se coloque en el dispositivo, enganchado a la fuente de energía y de información que necesite, y eso es todo. Un microprocesador tradicional no le permitirá hacer esto, ya que espera que todas estas tareas sean manejadas por otros chips. Hay que agregarle los módulos de entrada y salida (puertos) y la memoria para almacenamiento de información. (witronica.com, 2016).

6.2. Lenguajes de programación.

Dicho lenguaje está compuesto por símbolos y reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye una aplicación o pieza de software determinado. Así, puede llamarse también lenguaje de programación al resultado final de esto creativos. (Etece, 2021).

La implementación de lenguajes de programación permite el trabajo conjunto y coordinado, a través de un conjunto afín y finito de instrucciones posibles, de

diversos programadores o arquitectos de software, para lo cual estos lenguajes imitan, al menos formalmente, la lógica de los lenguajes humanos o naturales. (Etece, 2021).

El lenguaje humano no debe confundirse, con los distintos tipos de lenguaje informático. Estos últimos representan una categoría mucho más amplia, en donde están contenidos los lenguajes de programación y muchos otros protocolos informáticos, como el HTML de las páginas web. (Etece, 2021).

6.2.1. Tipos de lenguaje de programación.

Los lenguajes de programación normalmente se dividen en tres niveles comunes que dan el ámbito de cómo dar inicio, aprender y dominar estos lenguajes. (Cachaza 2022).

Lenguaje de bajo nivel: Se trata de lenguajes de programación que están diseñados para un hardware específico y que por lo tanto no pueden migrar o exportarse a otros computadores. Sacan el mayor provecho posible al sistema para el que fueron diseñados, pero no aplican para ningún otro. (Cachaza 2022).

Lenguaje maquina: es los más primitivos de los lenguajes y es una colección de dígitos binarios o bits (0 y 1) que la computadora lee e interpreta y son los únicos idiomas que las computadoras entienden. Ejemplo: 10110000 01100001. No se entiende muy bien lo que dice, por eso, el lenguaje ensamblador permite entender mejor a que se refiere este código. (rockcontent.com, 2019).

Lenguaje ensamblador: es el primer intento de sustitución del lenguaje de máquina por un más cercano al utilizado por los humanos. Un programa escrito en este lenguaje es almacenado como texto (tal como programas de alto nivel) y consiste en una serie de instrucciones que corresponden al flujo de órdenes ejecutables por un microprocesador. Sin embargo, dichas maquinas no comprenden

el lenguaje ensamblador, por lo que se debe convertir al lenguaje maquina mediante un programa llamado ensamblador. Este genera códigos compactos, rápidos y eficientes creados por el programador que tiene el control total de la máquina. Ejemplo: MOV AL, 61h (asigna el valor hexadecimal 61 registrado (AL) (rockcontent.com, 2019).

Lenguaje de medio nivel: los lenguajes de nivel medio tienden a ser clasificados como lenguajes de bajo nivel porque siguen dependiendo de ensambladores para poder ser comprendidos por los ordenadores. Pese a esto, permiten llevar a cabo tareas mucho más complejas que los lenguajes de bajo nivel como el uso de funciones. Aun así, muchas tareas deben ser llevadas a cabo de manera manual, aunque hay que destacar que permiten los algoritmos de búsqueda y ordenamiento a diferencia de otros lenguajes de alto nivel. (Etece. 2021).

En ese sentido, los lenguajes de medio nivel están orientados hacia la programación de sistemas, por lo que no son adecuados para tareas como el diseño web. Los lenguajes de medio nivel más conocido son C y C++. (Cachaza 2022).

Lenguaje de alto nivel: tiene como objetivo facilitar el trabajo del programador, ya que utilizan unas instrucciones más fáciles de entender. Además, el lenguaje de alto nivel permite escribir códigos mediante idiomas que se conocen (español, inglés, etc.) y luego, para ser ejecutados, se traduce al lenguaje maquina mediante traductores o compiladores. (rockcontent.com, 2019).

- Traductor: traducen programas en el lenguaje de programación al lenguaje máquina de la computadora y a medida que va siendo traducida, se ejecuta. (rockcontent.com, 2019).

- Compilador: permite traducir todo un programa de una sola vez, haciendo una ejecución más rápida y puede almacenarse para usarse luego sin volver a hacer la traducción. (rockcontent.com, 2019).

6.2.2. Características de lenguaje de programación.

La popularidad de un lenguaje de programación depende de las funcionalidades y utilidades que proporcione a los programadores. Las características que debe tener un lenguaje de programación para destacar son las siguientes: (Prats, 2015).

- **Simplicidad:** el lenguaje debe ofrecer conceptos claros y simples que faciliten su aprendizaje y aplicación, de manera que sea sencillo de comprender y mantener. La simplicidad no significa que se le pueda restar el poder óptimo de funcionamiento.

- **Naturalidad:** se refiere a que su aplicación en el área para la que fue diseñado debe hacerse de forma natural, proporcionando operadores, estructuras y sintaxis para que los operadores trabajen eficientemente.

- **Abstracción:** consiste en la capacidad de definir y utilizar estructuras u operaciones complicadas ignorando algunos detalles. Esto influye en la capacidad de escritura.

- **Eficiencia:** los lenguajes de programación deben traducirse y ejecutarse eficientemente para no ocupar demasiado espacio en la memoria ni gastar mucho tiempo.

- **Estructuración:** permite que los programadores escriban sus códigos de acuerdo con los conceptos de programación estructurada, para evitar la creación de errores.

- **Compacidad:** con esta característica es posible expresar las operaciones con concisión, sin tener que escribir demasiados detalles.

- **Localidad:** se refiere a que los códigos se concentran en la parte del programa con la cual se está trabajando en un momento determinado. (Prats, 2015).

6.3. Los dispositivos: hardware.

Hardware es una palabra inglesa que hace referencia a las partes físicas tangibles de un sistema informático**,** es decir**,** todo aquello que se puede tocar con las manos**.** Dentro del hardware se encuentra una gran variedad de componentes eléctricos, electrónicos, electromecánicos y mecánicos**.** El hardware es el chasis del ordenador, los cables, los ventiladores, los periféricos y todos los componentes que se pueden encontrar en un dispositivo electrónico. (Navas, 2011).

El término no solamente se aplica a los ordenadores, ya que es a menudo utilizado en otras áreas de la vida diaria y la tecnología como robots, teléfonos móviles, cámaras fotográficas, reproductores digitales o cualquier otro dispositivo electrónico. El hardware representa un concepto contrario al Software, la parte intangible de un sistema informático, es decir todo aquello que no se puede tocar físicamente. (Navas, 2011).

6.3.1. Dispositivos hardware de entrada.

Son todos aquellos dispositivos que permiten introducir datos o información en una computadora para que esta los procese u ordene. (Universidad autónoma intercultural de Sinaloa ((UAIS), 2018). Ejemplos:

- Teclado.
- Mouse o ratón.
- Escáner o digitalizador de imágenes.
- Cámara WEB.
- Micrófono. ((UAIS), 2018).

6.3.2. Dispositivos hardware de salida.

Son todos aquellos dispositivos que reciben información que es procesada por la CPU y que la reproducen para que sea perceptible para el usuario. (UAIS, 2018). Ejemplos:

- Impresoras.
- Pantalla o monitores.
- Proyector o cañón.
- Bocinas.
- Audífonos. (UAIS, 2018).

6.4. Puertos de comunicación hardware

Un puerto sirve como interfaz para enviar y recibir datos entre el computador y otros computadores o dispositivos periféricos. Un puerto de computadora es una ranura o toma de corriente de un equipo en el cual se enchufa un conector que regularmente contiene un cable. Los puertos que permiten conectar dispositivos,

generalmente se encuentran en la parte posterior, frontal o lateral de un equipo. (Palacios, 2016).

En la actualidad ciertos equipos vienen con un puerto VGA para conectar el monitor, varios puertos USB para conectar el ratón, monitor, disco duro externo, grabador de DVD, PS2 u otro dispositivo externo. Uno o varios puertos DVI o HDMI para conectar monitores de mejor resolución, puerto Ethernet para conexión a la red, un puerto para conectar altavoces, otro para conectar micrófono, etc. (Palacios, 2016).

6.5. Kernel

El Kernel o núcleo, es una parte fundamental del sistema operativo que se encarga de conceder el acceso al hardware de forma segura para todo el software que lo solicita**,** el Kernel es una pequeña e invisible parte del sistema operativo, pero la más importante, ya que sin esta no podría funcionar. Todos los sistemas operativos tienen un Kernel, incluso Windows 10, pero quizá el más famoso es el Kernel de Linux, que ahora además está integrado en Windows 10 con sus últimas actualizaciones. (Keepcoding, 2022).

Este núcleo de los sistemas operativos se ejecuta en modo privilegiado con acceso especial a los recursos del sistema para poder realizar las peticiones de acceso que le va pidiendo el software que lo necesita, además como los recursos no son ilimitados, también hace de árbitro a la hora de asignarlos, decidiendo el orden de las peticiones recibidas según la prioridad e importancia de estas**.** Una gestión muy importante y fundamental que en la mayoría de las ocasiones pasa desapercibida aun siendo un trabajo esencial para coordinar todo el hardware con el software. (Keepcoding, 2022).

6.6. Sistema Operativo

Actúa como un intermediario entre el usuario de una computadora y el hardware de la misma. El propósito de un sistema operativo es proporcionar un entorno en el que el usuario pueda ejecutar programas de una manera práctica y eficiente. También puede administrar el hardware de una microcomputadora. Este proporciona las bases para los programas de aplicación. (Etece, 2021).

Los sistemas operativos permiten que otros programas puedan utilizarlos de apoyo para poder funcionar, por eso, algunos programas pueden ser instalados o ejecutados, mientras que otros no; es decir, que se comporta como la pieza de software central en una cadena de procesos y establece las condiciones mínimas para que todo funcione, como la administración de recursos, el método de comunicación con los usuarios y con otros sistemas, además de aplicaciones adicionales. (Etece, 2021).

El sistema operativo posee tres componentes esenciales o paquetes de software que permiten la interacción con el hardware:

- Sistema de archivos: Es el registro de archivos donde adquieren una estructura arbórea.

- Interpretación de comandos: Se logra con aquellos componentes que permiten la interpretación de los comandos, que tienen como función comunicar las órdenes dadas por el usuario en el lenguaje que el hardware puede interpretar.

- Núcleo: Permite el funcionamiento en cuestiones básicas como la comunicación, entrada y salida de datos, gestión de procesos y la memoria, entre otros. (Etece, 2021).

6.6.1. Funciones de un sistema operativo:

- Gestionar la memoria de acceso aleatorio y ejecutar las aplicaciones, designando los recursos necesarios.

- Administrar al CPU gracias a un algoritmo de programación.

- Direccionar las entradas y salidas de datos por medio de periféricos de entrada o salida.

- Administrar la información para el buen funcionamiento de los programas.

- Dirigir las autorizaciones de uso para los usuarios.

- Administrar los archivos. (Etece, 2021).

6.6.2. Características de un sistema operativo:

- Es el intermediario entre el usuario y el hardware.

- Es necesario para el funcionamiento de todos los computadores, tabletas y teléfonos móviles.

- Otorga seguridad y protege a los programas y archivos del ordenador.

- Está diseñado para ser amigable con el usuario y fácil de usar.

- Permite administrar de manera eficiente los recursos del ordenador.

- La mayoría requiere del pago de una licencia para su uso.

- Permite interactuar con varios dispositivos.

- Es progresivo, ya que existen constantemente nuevas versiones que se actualizan y adaptan a las necesidades del usuario.

6.6.3. Tipos de sistemas operativos:

- Según el usuario pueden ser: multiusuario, sistema operativo que permite que varios usuarios ejecuten simultáneamente sus programas; o monousuario, sistema operativo que solamente permite ejecutar los programas de un usuario a la vez.

- Según la gestión de tareas pueden ser: mono tarea, sistema operativo que solamente permite ejecutar un proceso a la vez; o multitarea, sistema operativo que puede ejecutar varios procesos al mismo tiempo.

- Según la gestión de recursos pueden ser: centralizado, sistema operativo que solo permite utilizar los recursos de un solo ordenador; o distribuido, sistema operativo que permite ejecutar los procesos de más de un ordenador al mismo tiempo. (Etece, 2021).

6.6.4. Ejemplos de sistemas operativos:

Multics, BDOS, CP/M, UNIX, SunOS, MS-DOS, OS/2, Microsoft Windows, MacOS, GNU/Linux, Android. (Etece, 2021).

6.7. Interfaz Gráfica de Usuario

La Interfaz gráfica de usuario o GUI (Graphic User Interface) es el entorno visual de imágenes y objetos mediante el cual una máquina y un usuario interactúan. A mediados de los setentas las GUI comenzaron a sustituir a las interfaces de línea

de comando (CLI), y esto permitió que la interacción con las computadoras fuera más sencilla e intuitiva. (Lenis, 2021).

Su función principal es simplificar la comunicación entre una máquina o un sistema operativo y un usuario. Antes de que se desarrollaran y popularizaron las GUI, solo las personas con conocimientos profundos de informática podían usar un computador, pero las interfaces gráficas sustituyeron la complejidad de los comandos por acciones predeterminadas simbolizadas por elementos visuales muy sencillos de comprender. (Lenis, 2021).

Una buena GUI se caracteriza por:

- Ser sencilla de comprender y usar.
- La curva de aprendizaje es acelerada y es fácil recordar su funcionamiento.
- Los elementos principales son muy identificables.
- Facilitar y predecir las acciones más comunes del usuario.
- La información está adecuadamente ordenada mediante menús, iconos, barras, etc.
- Las operaciones son rápidas, intuitivas y reversibles.
- La interfaz expresa claramente el estado del sistema o las operaciones, y brinda elementos de ayuda.
- La navegabilidad y la usabilidad son óptimas. (Lenis, 2021).

6.8. Entorno de desarrollo integrado - IDE

El IDE (Integrated development enviroment) o entorno de desarrollo integrado, es una aplicación informática que proporciona una serie de servicios que facilitan la programación de software, tales como: (rockcontent.com, 2019).

- Funciones de autocompletado.
- Un editor de código fuente.
- Gestión de conexión a base de datos.
- Integración con sistemas de control de versiones.
- Simuladores de dispositivos.
- Depurador para agilizar el proceso de desarrollo de software, entre otros. (rockcontent.com, 2019).

6.9. Definición e historia de las tarjetas de desarrollo Arduino

Arduino es una plataforma de prototipos electrónica de código abierto (open – source) basada en hardware y software flexibles y fáciles de usar. Está pensado e inspirado en artistas, diseñadores, y estudiantes de computación o robótica y para cualquier interesado en crear objetos o entornos interactivo, o simplemente por hobby. (Arduino.cl, s.f).

Arduino fue inventado en el año 2005 por el entonces estudiante del instituto IVRAE Massimo Banzi, quien, en un principio, pensaba en hacer Arduino por una necesidad de aprendizaje para los estudiantes de computación y electrónica del mismo instituto, ya que en ese entonces, adquirir una placa con microcontroladores

eran bastante caro y no ofrecían el soporte adecuado; no obstante, nunca se imaginó que esta herramienta se llegaría a convertir en años más adelante en el líder mundial de tecnologías DIY (Do It Yourself). (Arduino.cl, s.f).

Inicialmente fue un proyecto creado no solo para economizar la creación de proyectos escolares dentro del instituto, sino que, además, Banzi tenía la intención de ayudar a su escuela a evitar la quiebra de la misma con las ganancias que produciría vendiendo sus placas dentro del campus a un precio accesible (1 euro por unidad). (Arduino.cl, s.f).

Arduino consta de una placa principal de componentes eléctricos, donde se encuentran conectados los controladores principales que gestionan los demás complementos y circuitos ensamblados en la misma. Además, requiere de un lenguaje de programación c++ para poder ser utilizado, por lo que se puede decir que Arduino es una herramienta "completa" en cuanto a las herramientas principales se refiere, ya que sólo se debe instalar y configurar con el lenguaje de programación de esta placa los componentes eléctricos que se desee para realizar el proyecto que se tiene en mente. (Arduino.cl, s.f).

Arduino también simplifica el proceso de trabajo con microcontroladores, ya que viene “pre ensamblada” y lista con los controladores necesarios para poder operar con ella, ofreciendo una ventaja muy grande para profesores, estudiantes y aficionados interesados en el desarrollo de tecnologías. (Arduino.cl, s.f).

7. DISEÑO METODOLÓGICO

7.1. Tipo de estudio

Enfoque de la investigación: Mixta

Cualitativa: Se indaga sobre los requerimientos que solicita la empresa acerca del sistema operativo, su capacidad y especificaciones mediante una entrevista con el gerente de operaciones quién detallará características visuales del entorno gráfico.

Cuantitativa: En ésta investigación se requiere que tanto los datos de entrada al proceso, como de salida sean numéricos y su representación debe ser geométricamente exacto, son variables que se pueden medir según lo que se visualiza en el entorno gráfico. La forma geométrica y su funcionalidad indicará si es correcto o no.

La secuencia temporal del estudio será de forma transversal, debido a que para la recolección de datos se hará en una sola entrevista para identificar los requerimientos del sistema operativo.

Tipo de investigación: Experimental.

Es experimental ya que se realizará una conexión de un circuito con elementos de tarjetas de desarrollo y dispositivos de entrada (teclado PS2) y salida (monitor VGA), además de la programación de un sistema que pueda entender instrucciones y se deben ejecutar las mismas para los fines que han sido creados. Se interviene directamente en el proceso mediante la programación del sistema operativo utilizando el software Arduino IDE y lenguaje de programación C++ en dónde se realizan pruebas constantemente para verificar el funcionamiento de los elementos de código, estructuras, procedimientos y funciones.

7.2. Área de Estudio

La propuesta de éste sistema operativo de entorno gráfico se realiza en la empresa UNITRIUM, empresa dedicada a ofrecer servicios profesionales de desarrollo de proyectos en el área de electrónica como diseño de circuitos eléctricos, programación de tarjetas de desarrollo, microcontroladores y microprocesadores. También se dedica al desarrollo de software para sistemas operativos Windows y Linux.

En la figura 2 se muestra el logotipo de la empresa:

Figura 2.

Logotipo de la empresa UNITRIUM.

Fuente propia.

La forma de generar recursos de la empresa UNITRIUM es la comercialización de productos y proveer servicios a otras empresas, desarrollo de proyectos, soporte técnico e instalación de sistemas electrónicos.

7.3. Universo y muestra

7.3.1. Macrolocalización de empresa

La empresa UNITRIUM se encuentra ubicada en Managua, colonia Tenderí, de Caruna 3 cuadras al norte (ver figura 3).

Latitud: 12° 8'38.54"N y Longitud: 86°14'47.98"O

Figura 3.

Macro localización de la empresa UNITRUM.

Fuente: software Google Earth PC (2022)

7.3.2. Microlocalización de empresa

Las dimensiones del local de la empresa UNITRIUM con 200 m², 10 m ancho y 20 m de largo. Actualmente cuenta con un personal de 6 personas. El local se puede ver en la figura 4:

Figura 4.

Frontal de la empresa UNITRUM.

Fuente Propia.

7.4. Variables y operacionalización de variables

Tabla 1.

Operacionalización de variables.

Objetivos Específicos	Variable Conceptual	Subvariables o Dimensiones	Variable Operativa o Indicador	Técnicas de Recolección de Datos e Información
Objetivo específico 1. Realizar un diagnóstico mediante una entrevista al gerente de operaciones de la empresa UNITRIUM para identificar los requisitos del sistema operativo.	Requisitos	1.1. Detalles del diseño visual del sistema. 1.2. Satisfacción de las necesidades de la empresa.	1.1.1. Dispositivos a utilizar en el sistema. 1.2.1. Características parámetros, funcionalidad.	Entrevista
Objetivo específico 2. Diseñar un sistema operativo de entorno gráfico a través de la programación en el software Arduino IDE y usando el lenguaje de programación C++ para satisfacer requisitos establecidos en el diagnóstico.	Diseño	2.1. Circuito eléctrico de dispositivos electrónicos. 2.2. Software.	2.1.1. Arduino Mega y otros dispositivos. 2.2.1. Monitor	Observación directa
Objetivo específico 3. Validar el funcionamiento del sistema operativo de entorno gráfico por medio de una comparación con los requisitos establecidos por la empresa UNITRIUM.	Validación	3.1. Introducir comandos de ejecución de funciones y operaciones.	3.1.1. Visualización de elementos gráficos en el monitor y examinar el funcionamiento.	Observación directa

Fuente propia.

7.5. Métodos, técnicas e instrumentos de recolección de información y datos

Entrevista: Para obtener la información del sistema operativo de entorno gráfico que se pretende desarrollar se realizará una entrevista al gerente de operaciones de la empresa, ya que indicará muchos detalles sobre los requerimientos de este proyecto, esto implica, los elementos o dispositivos electrónicos a utilizar, la función principal del sistema, las operaciones que debe ejecutar y la compatibilidad con otros dispositivos.

En la entrevista se aplicará la técnica 5W, que consigue transmitir un mensaje claro y eficaz. Es una técnica que se aplica desde hace 28 siglos, que se utiliza principalmente por periodistas. Aunque originalmente eran 5, con el paso del tiempo se pueden añadir otras más. Esto se basa en preguntas con el “Que”, “Quienes”, “Cuando”, “Dónde”, “Por qué”, y se pueden agregar preguntas como “Cómo” y “Para qué”. Con ésta técnica se pretende obtener información sobre el proyecto a realizar, su funcionalidad y su aplicación.

Observación directa: Se realizará la conexión de los distintos dispositivos electrónicos, se utilizarán las librerías ya creadas para cada uno de ellos en la tarjeta de desarrollo, y se comprobará el funcionamiento de ellos. Además, se realizará la programación del sistema operativo y se realizarán pruebas al momento de su desarrollo. Además, se harán pruebas para comprobar el funcionamiento del sistema operativo, principalmente que ejecute las operaciones deseadas.

7.6. Procedimientos para la recolección de datos e información

Las preguntas que se hace al entrevistado son las siguientes:

1. ¿A qué se dedica la empresa UNITRIUM?
2. ¿Quiénes son sus fundadores?
3. ¿Cuándo fue fundada la empresa?

4. ¿Dónde está ubicada la empresa?
5. ¿Por qué decidieron los fundadores este tipo de negocio?
6. ¿Qué problemática se puede abordar en un estudio en la empresa?
7. ¿Quiénes son los beneficiados por el estudio?
8. ¿Cuándo se haría uso del estudio?
9. ¿Dónde se aplicaría el estudio?
10. ¿Por qué este estudio sería importante para la empresa?
11. ¿Cómo se desea que se realice éste estudio?
12. ¿Qué resultados son los esperados?
13. ¿Cómo se medirán el cumplimiento de los objetivos del estudio?

Con las respuestas a éstas preguntas se puede dar una idea sobre lo que la empresa desea. Con respecto al diseño tanto del circuito como de la programación del sistema operativo se harán mediante la información obtenida de la entrevista. Esto significa que se diseñará mediante los datos proporcionados por la empresa. Así mismo, la comprobación del funcionamiento se hará en correspondencia a los requerimientos del sistema operativo. Se realizará una comprobación con los resultados esperados, es decir, la ejecución de operaciones que mostrarán en la pantalla los elementos gráficos.

7.7. Plan de análisis y procesamiento de datos

La entrevista que se realizó al gerente de operaciones de UNITRIUM permite que se comprenda la forma de negocio de la empresa y el proyecto que desea realizar. Principalmente para obtener información de los dispositivos a utilizar, las características que se buscan del proyecto y la funcionalidad del mismo. De esto se realiza el diseño del mismo como una solución a la problemática. Así se manejará la variable cualitativa. Mientras que en el desarrollo del sistema operativo se observará visualmente en el monitor en cada paso de la programación si se va cumpliendo lo requerido por la empresa.

8. DESARROLLO

8.1. Diagnóstico realizado en la empresa UNITRIUM para identificar los requisitos del sistema operativo de entorno gráfico

8.1.1. Descripción física de la empresa UNITRIUM

La empresa UNITRIUM está ubicada en Managua, colonia Tenderí, de Caruna 3 cuadras al norte, con un área total de 200 m². El local es una estructura de una planta de tipo rectangular, conformadas internamente por áreas administrativas en las cuales se encuentran oficina de gerencia, oficina de atención, departamentos de desarrollo software/hardware; tiene un responsable general de departamentos, bodega y otro de contabilidad, cartera y cobro. (Ver figura 5)

Dichas instalaciones están situadas con las siguientes limitaciones, al norte tiene la calle principal, al sur, oeste y este con casa habitacional. Se observa que la calle es muy concurrida y tiene una buena ubicación que mejorara el desarrollo dela empresa antes mencionada.

Físicamente el local cuenta con un garaje exterior e interior lateral derecho con un portón de 4 m, cuenta con una puerta de entrada de 1.55 m de largo y 1.25 m de ancho, donde da acceso directo a una sala de espera, área de recepción y un pasillo que dirige a las demás áreas de trabajo.

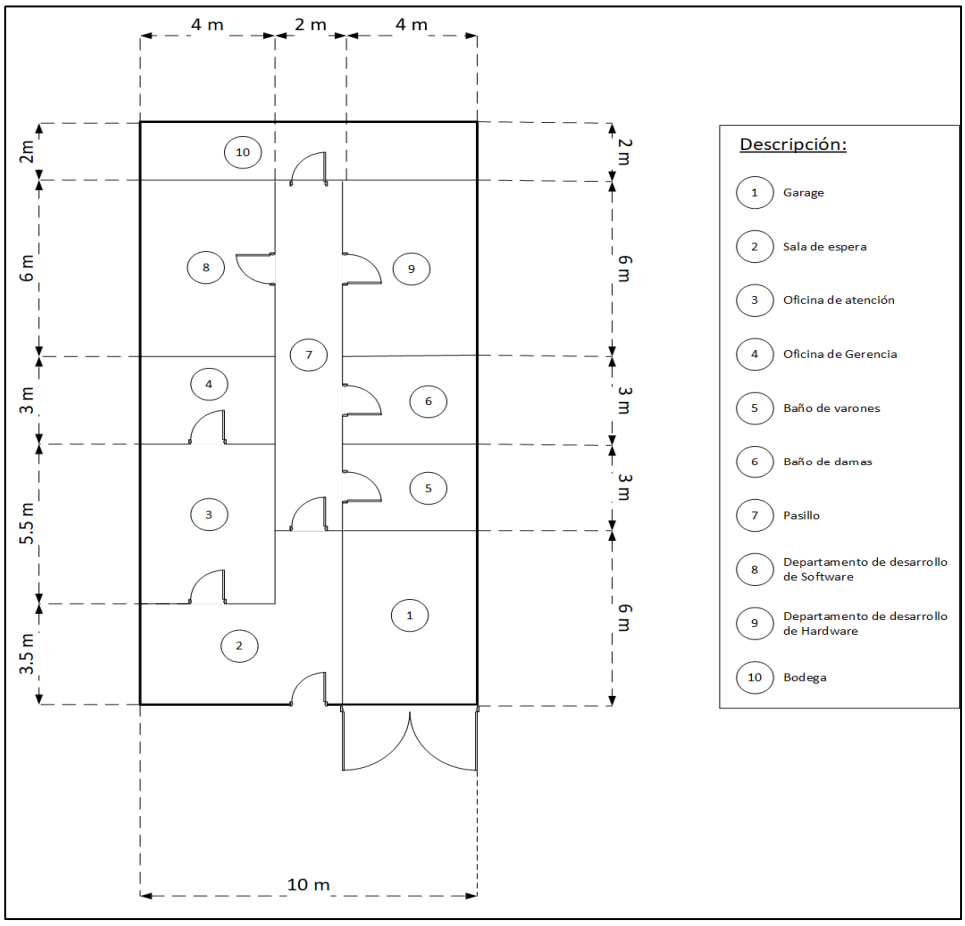
Las estructuras laterales están hechas de paredes de concreto pintado en color rosa y verjas blancas, soportadas por columnas, y sus divisiones internas de oficinas y baños se encuentran hechas de Gypsum y techo de láminas corrugadas de Zinc sujeto de perlines, dispone de cielo raso en toda la propiedad y piso de cerámica.

Las instalaciones eléctricas de UNITRIUM están afiliadas a la empresa Disnorte-Dissur, utiliza el servicio comercial (120V AC), se utiliza las normas reglamentarias de cableado eléctrico utilizando contactores y relés térmicos por la potencia que se utiliza al trabajar con motores y bombas de agua.

Posee una bodega que está equipada con microcontroladores, tarjetas de desarrollo, placas electrónicas, productos de dispositivos electrónicos y de cómputo en hardware, cumpliendo todas normas NTON 22 001-04 para prevención de riesgos de incendio, en estas por la importancia de los productos y que se poseen productos inflamables.

Figura 5.

Distribución del local de la empresa UNITRIUM.



Fuente propia.

8.1.2. Descripción de la actividad comercial de la empresa UNITRIUM

La empresa UNITRIUM ofrece servicios profesionales de desarrollo de proyectos en el área de electrónica, como diseño de circuitos electrónicos, programación de tarjetas de desarrollo, microcontroladores y microprocesadores. También se dedica al desarrollo de software para sistemas operativos Windows y Linux, especialmente en software dedicado a Estadística y Administración.

La empresa ofrece sus servicios en electrónica en Nicaragua, pero en el desarrollo de software lo brinda a empresas en el extranjero. La empresa quiere seguir el avance tecnológico con jóvenes y desarrolladores para innovar en las áreas tecnológicas que en el país no es muy cubierto por empresas y cultivar el proceso de las ramas de tecnología.

8.1.3. Requisitos del sistema operativo solicitado por la empresa UNITRIUM

A través de la herramienta de recopilación de información, se realizó una entrevista al gerente de operaciones cuyo nombre es Oscar José Turcios, explicó sobre el sistema operativo que desean realizar para tarjetas de desarrollo de bajo costo.

El entrevistado indicó la línea de investigación y desarrollo que sigue la empresa que es crear su primer sistema operativo de entorno gráfico para tarjetas de desarrollo, ya que son dispositivos electrónicos que son muy utilizados por las empresas y tienen bastante potencial, sin embargo, se dan cuenta de que el uso que se les da es muy básico y se piensa que éstos pueden ser utilizados en sistemas más complejos.

La tarjeta de desarrollo necesita recibir instrucciones, pero esto deben ser introducidos mediante un dispositivo de entrada, lo deseado es que sea un teclado de computadora. También, se necesita visualizar la información, por lo cual, se

podría elegir entre distintos dispositivos o pantallas LCD, pero, se desea que sea un dispositivo común, por lo cual se requiere del uso de un monitor de computadora tipo VGA en dónde se puedan observar muchos elementos en pantalla.

La empresa UNITRIUM desea enfocarse en un tipo de tarjeta de desarrollo como el Arduino Mega, ya que es popularmente utilizado en todo el mundo, por su facilidad de uso, sin embargo, no se ha profundizado en hacer cosas complejas como sistemas operativos.

Se ha buscado información en distintos foros en internet y otras personas que se dedican al desarrollo con éste tipo de sistemas, sin embargo, no han hecho nada parecido. Si existen algunos sistemas con pantallas lcd y con teclados, pero son muy básicos y limitados, no es lo que la empresa quiere hacer, muchos de ellos son equipos costosos, privados, no se puede acceder al código. Mientras, que la empresa quiere emprender esto, explorando, porque su algoritmo servirá para luego hacer otros sistemas operativos con otros tipos de tarjetas de desarrollo.

Es total interés de la empresa UNITRIUM en crear sistemas operativos propios, aunque quizás ya existan en otro lugar del mundo; pero con códigos de programación hechos por la misma empresa y que se puede realizar cualquier modificación en el tiempo que sea conveniente. Así mismo, tener la capacidad de configurar o ajustar mediante distribuciones según lo solicitado por algún cliente u otra empresa.

Este sistema operativo de entorno gráfico es un producto que va dirigido a toda aquella persona o empresa que se dedique al desarrollo de sistemas electrónicos como HMI, CAM 2D y CAM 3D. Puede ser usado cada vez que se requiera hacer un sistema con circuitos electrónicos complejos donde se necesite mostrar mucha información en tiempo real.

Un sistema operativo de entorno gráfico para tarjetas de desarrollo con microcontroladores de bajos recursos a como es Arduino Mega es importante porque sería la base para el desarrollo de sistemas operativos para otros microcontroladores de mayor capacidad. Esta experiencia permitirá construir un algoritmo estándar, funcional y compatible que puede ser aplicado en otros tipos de sistemas.

Lo que se espera es un sistema que acepte instrucciones y que el microcontrolador entienda que operaciones debe ejecutar, para esto, debe existir una caja de texto (donde aparece el ingreso de caracteres), también se podrá borrar el último carácter. Cuando se presione la tecla ENTER entonces, se empezará un proceso en que busque comandos y argumentos para ejecutar operaciones.

Donde se va a reflejar las operaciones será en dos “escritorios”, es decir 2 áreas en una pantalla, en las cuales se presentarán los elementos gráficos, que van desde puntos, líneas, triángulos, hasta elementos más complejos como polígonos. Así mismo, sistemas gráficos de ecuaciones de grado n, gráficos en tiempo real que puedan mostrar información de los pines digitales y analógicos.

8.1.4. Requisitos técnicos del sistema operativo

Cuando se habla de sistemas operativos (software) existe dependencia primero de un microcontrolador (hardware) con ciertas características y capacidades, también de dispositivos de entrada (teclado PS/2) y dispositivos de salida (Monitor VGA). La combinación de éstos tres elementos en cierto modo se puede decir que dan vida al sistema operativo.

Refiriéndose directamente a lo que es el sistema operativo, se requiere que, desde el teclado al pulsar una tecla, el microcontrolador logre comprender que tecla es la que se está presionando en el teclado, esta tecla representa un caracter, mientras eso sucede, el microcontrolador debe tener la capacidad de convertir la

información en su símbolo correspondiente y mostrarlo en el monitor VGA. También, cada tecla pulsada va formando un conjunto de palabras en la “caja de texto”. Este conjunto de palabras puede ser algo aleatorio o puede ser una “instrucción”.

Una instrucción es la combinación de un “comando” y sus respectivos “argumentos”, si la instrucción está completa o está en lo mínimo requerido, entonces, ejecutará operaciones. Estas operaciones son funciones que cumplirá el microcontrolador. Cada vez que se presiona la tecla “Enter”, debe empezar un proceso de reconocer un comando y sus argumentos, si existen los dos, se ejecuta una operación o conjunto de operaciones.

El sistema operativo debe tener la capacidad de ejecutar un conjunto de instrucciones, aunque en realidad se ejecutará operación por operación, mientras termina una, sigue con otra. Además de la capacidad de poder detenerlos. Limpiar los escritorios y volver a ejecutar otra instrucción que sea necesaria. Otra capacidad que debe tener el sistema operativo es poder recibir una o más instrucciones de dispositivos externos y ejecutar estas instrucciones.

8.1.5. Requisitos del aspecto visual del sistema operativo

Se desea que la pantalla del monitor se vea como en la figura 6, donde se divide la pantalla en cuatro áreas, la parte superior será el historial de instrucciones, caja de texto, escritorio 1 y escritorio 2. La caja de texto tiene capacidad para incluir hasta 32 caracteres, este se va construyendo mientras se van pulsando las teclas, también debe funcionar la tecla backspace para eliminar el último carácter escrito.

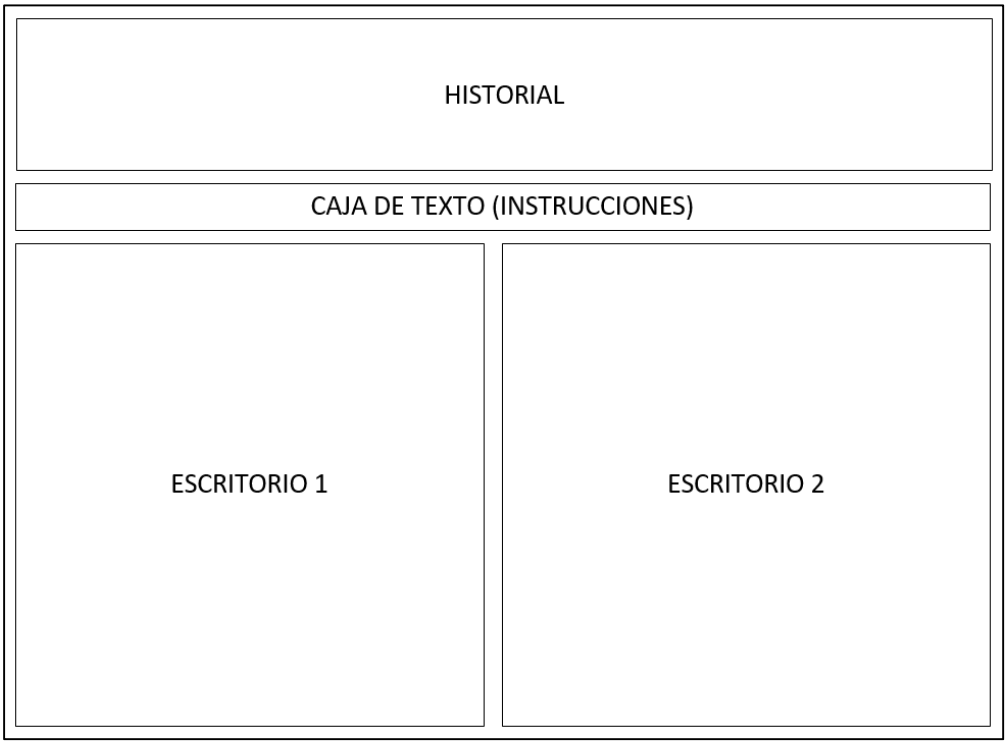
El área del historial, es un registro temporal, de lo que se escribe en la caja de texto, cada vez que se presiona la tecla Enter, lo que está en la caja de texto pasa a la fila 6, mientras que lo que está en la fila 6 pasa a la 5, hasta que lo que está en la fila 2 pasa a la fila 1 (Ver figura 7).

Cuando se presiona la tecla ENTER entonces, se inicia la búsqueda en la caja de texto de lo que son comandos y argumentos, para ejecutar alguna función dentro del escritorio 1 o 2.

Se requiere que el sistema operativo pueda poner en la pantalla del monitor mediante instrucciones elementos como pixeles, líneas, triángulos, cuadrados, círculos, polígonos, estrellas, etc. Éstos son elementos básicos (ver figura 8) que permiten dibujar cosas más complejas como gráficos en 2D (figura 9) y gráficos en tiempo real (ver figura 10).

Figura 6.

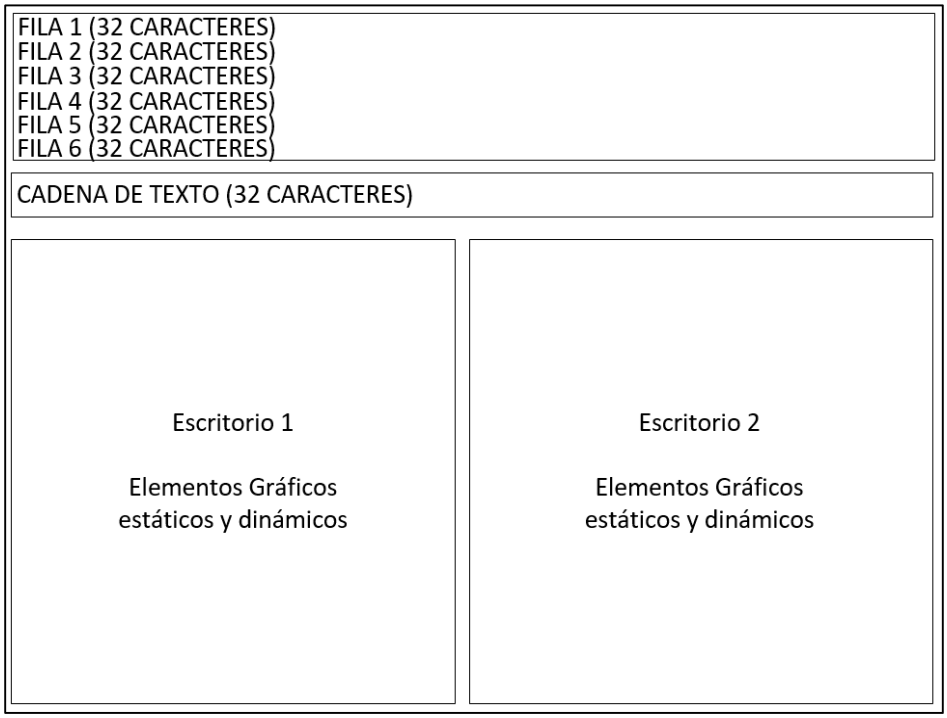
Distribución visual de la pantalla 1.



Fuente propia.

Figura 7.

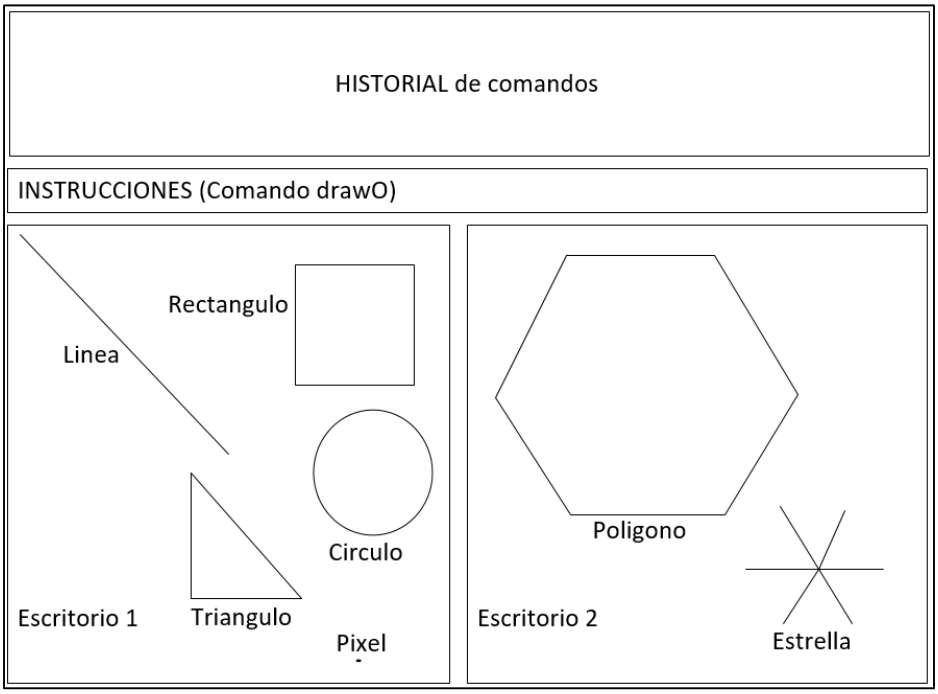
Distribución visual de la pantalla 2



Fuente propia.

Figura 8.

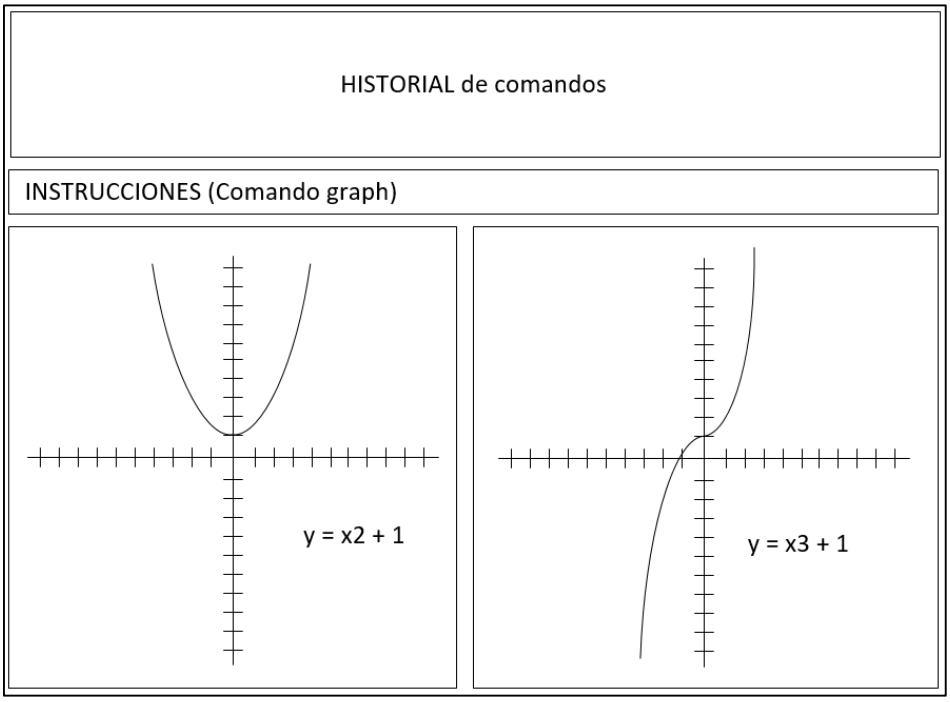
Graficar elementos estáticos en escritorio 1 y 2



. Fuente Propia.

Figura 9.

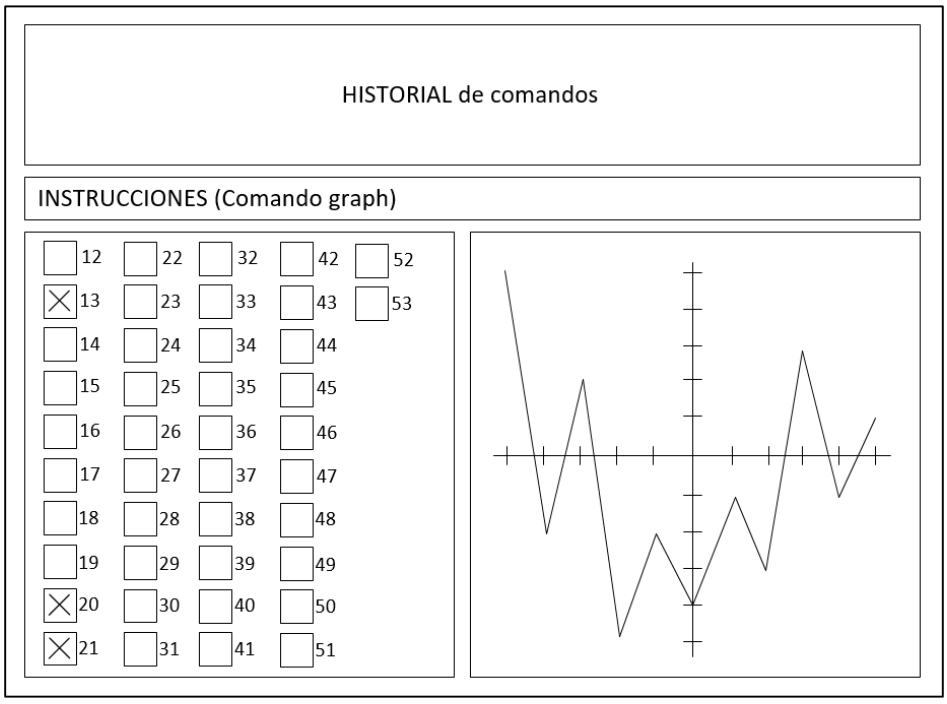
Creación de gráficos con ecuaciones de grado n.



Fuente propia.

Figura 10.

Creación de gráficos en tiempo real.



Fuente propia.

8.2. Diseño del sistema operativo que funciona en tarjeta de desarrollo Arduino Mega

Después de realizar el diagnóstico y conocer los requerimientos se desarrolló el diseño del sistema operativo para Arduino Mega en la empresa UNITRIUM. Conforme el primer objetivo específico se diseña un producto capaz de realizar diversas funciones de manera práctica.

En este diseño se dará a conocer los dispositivos y componentes que se utilizarán para cumplir con las necesidades solicitadas e implementar un sistema operativo que sea factible, económico y funcione eficazmente.

8.2.1. Dispositivos electrónicos utilizados para el funcionamiento del sistema operativo de entorno gráfico.

8.2.1.1. Arduino UNO.

La tarjeta de desarrollo Arduino UNO es las más utilizada en proyectos robóticos e innovaciones en distintas áreas de la tecnológicas, este contiene un microcontrolador ATmega328, 32KB de memoria flash en donde se almacena el código en donde 0,5KB es utilizado en arranque.

Esta también dispone de 2KB de memoria SRAM y 1KB de EEPROM; tiene 14 salidas y entradas digitales de los cuales se utilizan para PWM, 6 entradas analógicas, un cristal de 16MHZ oscilador, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio. Trabaja a un voltaje de 5V DC a 3,3 V DC con una corriente de 40mA, ver Ilustración 1. Anexo. También toda la información de la tarjeta de desarrollo se puede observar en la tabla 2 de Anexos.

8.2.1.2. Arduino Mega/2560.

Arduino mega es una tarjeta de desarrollo de tamaño más grande que otras y más potente, basado en el microcontrolador ATmega2560, este contiene 256KB de memoria flash para almacenar código, en los cuales 8KB este arranque, 8KB de SRAM y 4KB de EEPROM; tiene 54 pines digitales de entrada y salida, en los cuales 15 se pueden utilizar como PWM, además de 16 entradas analógicas, 4 puertas seriales, un oscilador 16MHZ, una conexión USB, un conector de alimentación, una cabecera ICSP y su botón de reinicio. Trabaja con un voltaje de 7 a 12V DC, con una corriente 40mA, más información en Ilustración 2. Anexo.

También toda la información de la tarjeta de desarrollo Arduino Mega se puede observar en la tabla 3 de Anexos.

El principal dispositivo es el Arduino Mega quién administra las funciones del sistema operativo, que se utilizó por su bajo costo a petición de las necesidades de la empresa, este llevara la programación requerida para el uso.

8.2.1.3. Monitor VGA.

También denominadas pantallas estándar analógicas de computadora, especialmente utilizan un conector de 15 contactos llamado D subminiatura (ver ilustración 3, Anexos), el término se utiliza para denominar Video Graphics Array (VGA) o matriz de grafico de video que corresponde a la forma en que se habilitan los subpixeles y pixeles, como una matriz de dos dimensiones, de forma horizontal y vertical, a esto se le conoce como RGB mas sincronisos de H y V. Ver ilustración 4. Anexo.

Las especificaciones originales de VGA son las siguientes:

256 KiB de VRAM

Modos de imagen con paletas de 16 y 256 colores

Paleta global de 262144 colores (6 bits y por tanto 64 valores para cada uno de los canales rojo, verde y azul mediante el RAMDAC)

Reloj maestro seleccionable de 25,2 MHz o 28,3

Máximo de 800 píxeles horizontales

Máximo de 600 líneas

Tasa de refresco de hasta 120 Hz

Interrupción de blanqueo vertical (No todas las tarjetas lo soportan)

Modo plano: máximo de 16 colores

Modo píxel empaquetado: en modo 256 colores (Modo 13h)

Soporte para desplazamiento suave de la imagen.

Algunas operaciones para mapas de bits

Desplazador "en barril"

Soporte para pantalla dividida

0,7 V pico a pico

75 ohmios de impedancia de doble terminación (18,7 mA - 13 mW)

VGA soporta tanto los modos de todos los puntos direccionables como modos de texto alfanuméricos. Los modos estándar de gráficos son:

640×480 en 16 colores

640×350 en 16 colores

320×200 en 16 colores

320×200 en 256 colores (Modo 13h)

8.2.1.4. Teclado PS/2

Un teclado es un dispositivo que presenta el conjunto de las teclas de diversos aparatos, máquinas e instrumentos. Por lo general, el teclado permite el control o mando del aparato en cuestión. Los teclados de computadora (ver ilustración 5. Anexos) presentan teclas alfanuméricas (letras y números), de puntuación (punto, coma, etc.) y teclas especiales (que cumplen distintas funciones, por ejemplo) (ver ilustración 6. Anexos).

La pulsación de las teclas del teclado genera un código ASCII, que se transfiere mediante los conectores PS/2 (ver ilustración 7 y 8. Anexos) que deberá ser interpretado por el microcontrolador o microprocesador para ser procesado. En la ilustración 9 de Anexos, se ve las teclas y su respectivo código ASCII en hexadecimal.

8.2.2. Librerías de Arduino para dispositivos.

8.2.2.1. Librería para uso de teclados PS/2.

La librería para utilizar un teclado PS/2 se llama PS2Keyboard_master. Esta librería utiliza una de las dos interrupciones externas disponibles para reaccionar a la entrada del teclado. Una vez que este recibe dicha entrada, se almacena en un búfer de un byte y está disponible para su lectura. Esta librería fue creada por Paul Stoffregen. (ver ejemplo en Ilustración 10. Anexos). El código de cada tecla lo recibe en ASCII (ver ilustración 11. Anexos).

8.2.2.2. Librería VGAXUA para monitores VGA

Es una librería diseñada por Sandro Maffiodo en el 2019, la cual está basada en otra librería llamada VGA de Nick Gammon. La librería implementa un framebuffer de 192x80px donde cada píxel se almacena como 1 bit (2 colores). En Arduino MEGA, la resolución se puede aumentar a 200x240px. El framebuffer se almacena dentro de SRAM y requiere al menos 1920 bytes. Esto significa que en ATMega328 sus programas no pueden usar más de 128 bytes de SRAM, Si lo desea, puede usar otro Arduino UNO para controlar el que usa la biblioteca VGAXUA. En ATMega2560 tiene más SRAM, pero si expande el framebuffer a 200x240px, la SRAM libre será de 2000 bytes.

El framebuffer VGAXUA usa 1 bit por cada píxel. Dentro de cada byte se almacenan 8 píxeles, empaquetados en orden inverso: el píxel más a la derecha está en el bit más significativo (LSB->MSB). La biblioteca VGAXUA utiliza el bus serie AVR UART para canalizar píxeles, en lugar de software bitbanging como VGAX.. En Arduino MEGA (ATMega2560), el framebuffer se puede extender a 200x80px con píxeles cuadrados o 200x240px con píxeles rectangulares. Puede habilitar esta resolución alternativa.

Mediante la conexión física del Arduino Mega al monitor se usan varias señales de tiempo. Hay 5 señales que se envían al monitor mediante la conexión VGA:

- Sincronización vertical (TTL)
- Sincronización horizontal (TTL)
- Datos analógicos rojo (0 – 0.7 V)
- Datos analógicos verde (0 – 0.7 V)
- Datos analógicos azul (0 – 0.7 V)

Al suponer que se tiene una pantalla de resolución de 640 x 480, el hardware necesita un poco más de tiempo para mover el haz de electrones de un lado de la pantalla al otro y de abajo hacia atrás a la cima. Estos son los "pórticos" delanteros y traseros. (Ver ilustración 12. Anexos).

La actualización de la pantalla comienza con un pulso de sincronización vertical, que le indica al monitor que se reinicie en la parte superior de la pantalla. Tiene el número de líneas del "pórtico trasero" para prepararse para comenzar a dibujar y para dibujar un área en blanco en la parte superior. Luego dibuja la imagen y tiene las líneas adicionales del "porche delantero" para dibujar la información en blanco en la parte inferior de la pantalla.

Mientras tanto, para cada línea, hay un pulso de sincronización horizontal que señala el comienzo de esa línea, seguido de otro retraso para darle tiempo al haz de estar listo, luego dibuja la línea y tiene un tiempo adicional de "porche delantero" para la parte en blanco al final de la línea.

8.2.3. Protocolos de comunicación de los dispositivos.

8.2.3.1. El teclado y protocolo PS/2

Se comunica mediante un Protocolo Serie Síncrono. Utiliza, por lo tanto, una señal de reloj que índica cuando están disponibles los correspondientes bits en la señal de Data.

En reposo la señal de reloj está a nivel alto; a cada pulso a nivel bajo corresponde un pulso a nivel alto o bajo en la señal de Data, que se traducen respectivamente como bits 0 ó 1 del dato a transmitir. La trama completa se compone de 11 bits. Siendo el primero un bit de Start, a continuación, los 8 bits del Dato a transmitir enviándose primero el LSB (ó bit menos significativo), el

décimo es el de paridad (usa la Impar, u Odd en inglés) y por último un bit de ACK o Stop.

En la ilustración 13 de Anexos puede verse un cronograma de esta trama de comunicación PS/2 Teclado (Keyboard) -> PC (host):

Este protocolo de comunicación es bidireccional. El teclado PS/2 admite también comandos enviados desde el PC o algún otro dispositivo.

Para habilitar la comunicación inversa, del PC (host) al Teclado PS/2 (Keyboard) se debe primero indicárselo así al teclado mediante la señal de reloj. Para ello se debe poner a nivel bajo el reloj durante unos 160 uS, y la señal de Data a bajo unos 35 uS después de haber lanzado la del reloj. A partir de ahí se debe esperar la señal del reloj generada por el teclado. Esto indicará que el teclado está dispuesto para recibir cualquier comando. Se detecta dicha señal como primer pulso de reloj, y a partir del siguiente se comienza a enviar el byte.

Se enviará entonces los ocho bits del comando, cada uno de ellos cuando el correspondiente pulso en bajo del reloj del teclado así lo indique, empezando por el LSB, a continuación, el bit de paridad impar (El número de unos en los datos más el de paridad deber ser impar o sea 1 si el número de unos es par y cero si el total de unos es impar). Y entonces se debe esperar el ACK del teclado, que debe venir tras dos pulsos de reloj, en nivel bajo, indicando de este modo que el teclado ha recibido correctamente el comando, en caso contrario se debe volver a repetir la secuencia de envío. Ejemplo en la ilustración 14, Anexos.

8.2.3.2. Protocolo de comunicación I2C

I2C (por sus siglas en inglés *Inter-Integrated Circuit*) es un protocolo de comunicación serial desarrollado por Phillips Semiconductors**,** allá por la década de los 80s. En un inicio se creó para poder comunicar varios chips al mismo tiempo dentro de los televisores que fabricaba la compañía. Sin embargo, con el paso del tiempo otros fabricantes comenzaron a adoptarlo hasta convertirse en el estándar del mercado mundial que es hoy**.**

El protocolo I2C funciona con una arquitectura maestro-esclavo (Ver ilustración 15, Anexo). En esta arquitectura existen dos tipos de dispositivos:

Maestro o Controlador: son los que inician y coordinan la comunicación. Usualmente, cuando utilizas un Arduino en un bus I2C este es el rol que cumple.

Esclavos o Periféricos: son los dispositivos que están a la espera de que algún maestro se comunique con ellos. Casos comunes son los sensores y actuadores que soportan este protocolo, aunque también es posible, y en ocasiones necesario, que un microcontrolador se comporte como un esclavo.

La trama de transmisión del bus I2C (ver ilustración 16. Anexo) se realiza del siguiente modo:

- Se envía una secuencia de inicio.
- Se envía la dirección de dispositivo con el bit de lectura/escritura en bajo.
- Se envía el número de registro interno en el que se desea escribir.
- Se envía el byte de dato.

- Opcionalmente – Se envía más bytes de datos.
- Se envía la secuencia de parada.

Los pines de comunicación I2C para Arduino UNO son A4 (SDA) y A5 (SCL) y para el Arduino Mega 20 (SDA) y 21 (SCL). (ver ilustración 17. Anexo).

8.2.4. Diseño del circuito (conexión física) de los dispositivos

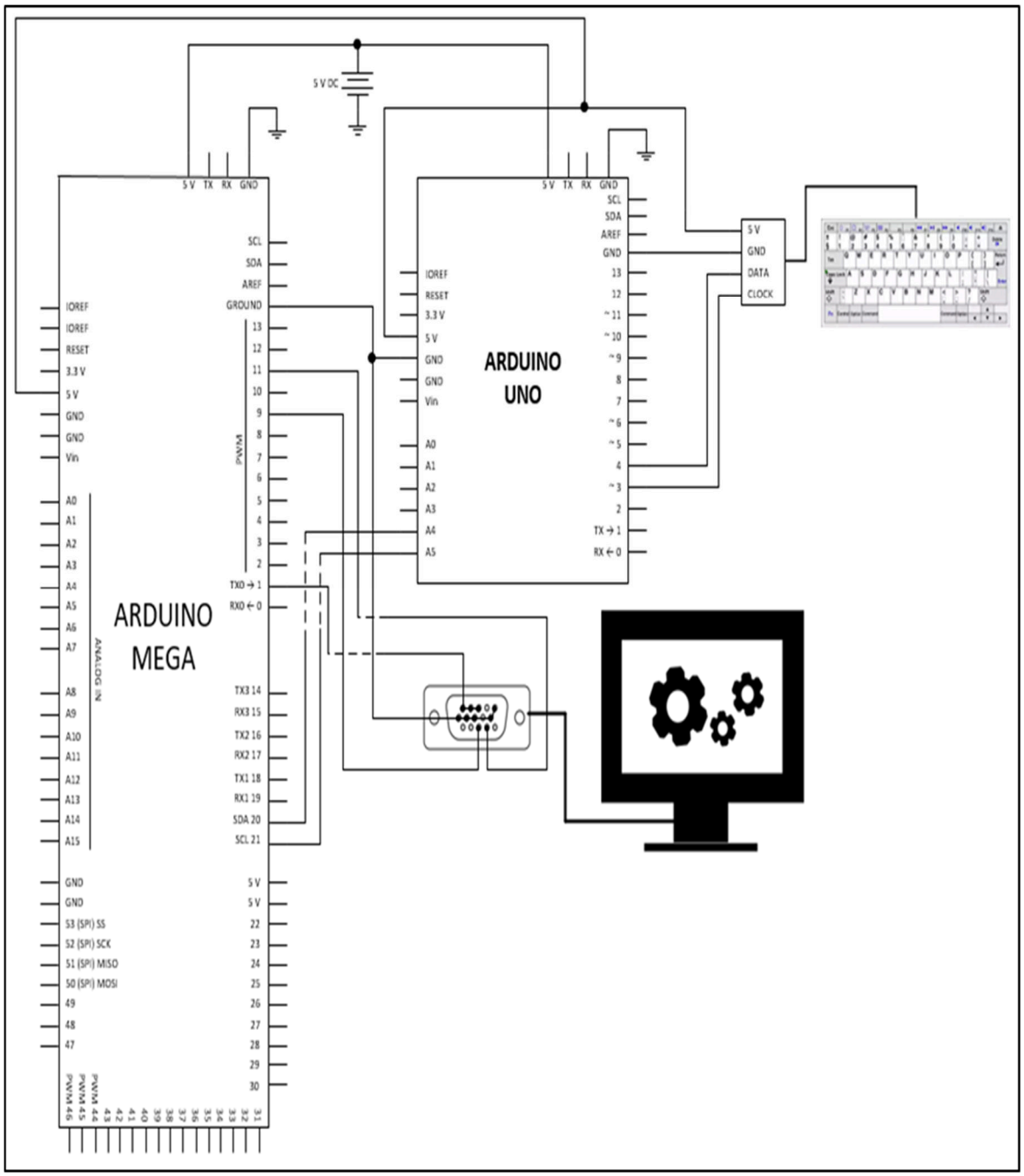
A partir de las indicaciones de la empresa, se ha realizado un diseño de conexión de los dispositivos que muestra el circuito físico que se debe conectar para que funcione el sistema operativo. Se puede observar en la figura 11, dos tarjetas de desarrollo, uno el Arduino Mega y el segundo el Arduino UNO, éstos están conectados con el protocolo I2C.

Además, se realiza la conexión del monitor al Arduino Mega según la configuración de los cables VGA, el vSync va al pin 11, el hSync va al pin 9, mientras que los cables Red, Green y Blue, se unen y van al pin 1, mientras que un cable de GND VGA se conecta al GND del Arduino.

En la conexión del Arduino UNO y el teclado PS/2, los cables importantes son los de clock (reloj) y data, que se conecta al 3 y 4 correspondientemente, también se enlazan los cables de 5V y GND a la tarjeta de desarrollo Arduino UNO.

Figura 11.

Circuito de conexión física entre los Arduino UNO, Arduino MEGA, monitor VGA y teclado PS/2.



Fuente Propia.

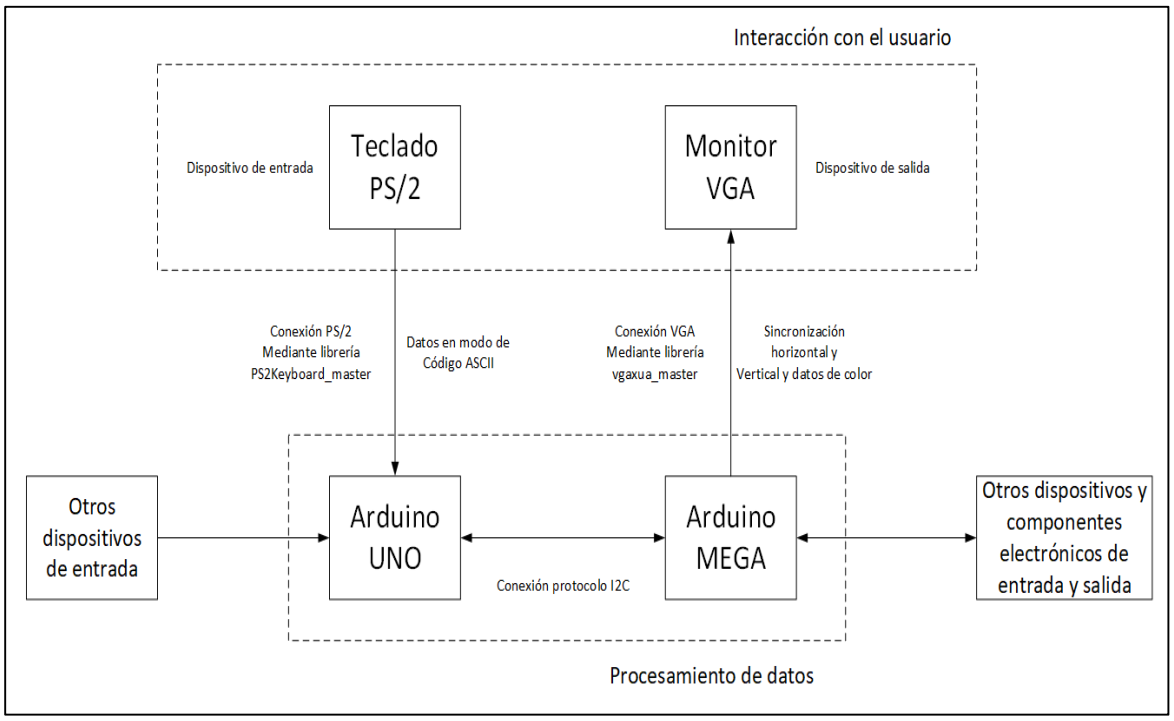
8.2.5. Diseño funcional del sistema operativo de entorno gráfico

En la figura 12, se observa un diagrama funcional del circuito. Los únicos dispositivos que interactúan con el usuario son el teclado PS/2 y el monitor VGA. Mientras que los encargados de procesar la información son el Arduino UNO y el Arduino Mega. Mientras, que éstos se pueden conectar con otros dispositivos de entrada y salida. Es importante la versatilidad del sistema operativo que pueda conectarse con otros dispositivos o componentes electrónicos, para que se pueda interactuar con ellos.

La comunicación entre los dos Arduino, será por el protocolo I2C.

Figura 12.

Diagrama funcional del circuito físico de los dispositivos.



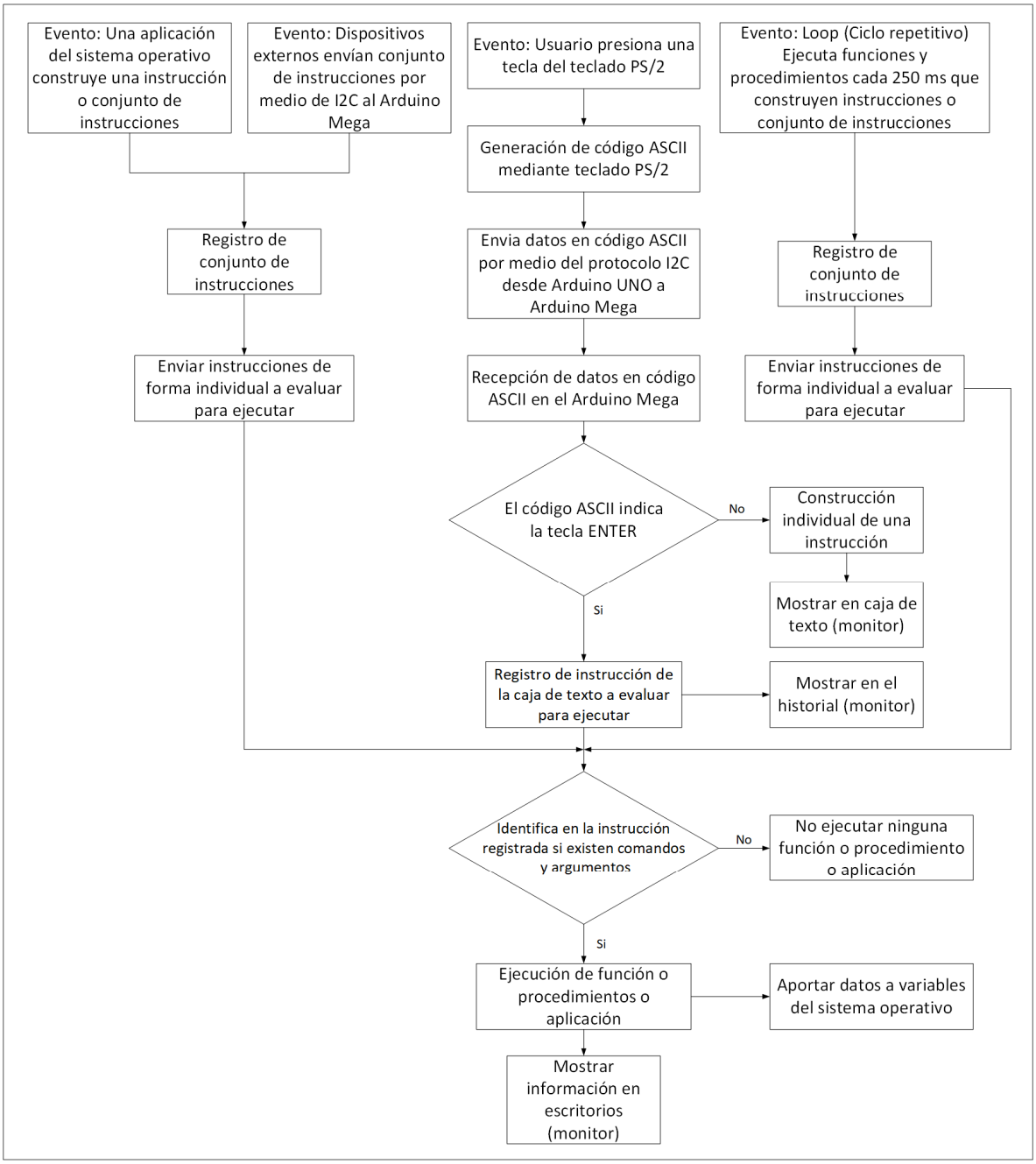
Fuente propia.

En la figura 13, se puede observar el diagrama de flujo del sistema operativo, en donde se detalla principalmente la capacidad que tiene para ejecutar una

instrucción cuando lo hace un usuario o cuando un dispositivo externo o el sistema en tiempo real, envía a ejecutar un conjunto de instrucciones.

Figura 13.

Diagrama de flujo del sistema operativo de entorno gráfico.



Fuente propia.

Dentro del sistema operativo existen 4 eventos principales:

- Evento 1: cuando un usuario pulsa cualquier tecla del teclado PS/2, lo que sucede es que genera un código ASCII relacionado a la tecla pulsada y es reconocido por el Arduino UNO, este dato también se maneja como un número entero. Se envía el dato desde el Arduino UNO al Arduino Mega mediante el protocolo I2C, para esto, al principio de la comunicación se envía también un identificador para que el Arduino Mega entienda que esta comunicación es debido a una tecla pulsada.

  Si la tecla presionada es un carácter alfanumérico o símbolo, entonces, este dato será agregado un array tipo char, en dónde se va construyendo una instrucción individual y ésta se muestra en la caja de texto que se muestra en la pantalla del monitor.

  Si la tecla presionada es ENTER, entonces, se toma la instrucción construida y se evalúa si existe un comando como “drawO”, “clear”, “graph” o “utils”. Además, se identifica los argumentos que deberán estar separados por un espacio.

- Evento 2: cuando una aplicación del sistema operativo construye una instrucción o conjunto de instrucciones y envía a ejecutar una por una todas las instrucciones.

- Evento 3: cuando un dispositivo externo envía un conjunto de instrucciones al Arduino Mega, y envía a evaluar para que se ejecuten las instrucciones individualmente una por una.

- Evento 4: en el ciclo repetitivo en dónde cada 250 ms se ejecutan funciones, procedimientos y aplicaciones, éstos tienen capacidad de construir instrucciones o conjunto de instrucciones y envía a evaluar para que se ejecuten las instrucciones.

Cuando se envía a una instrucción a evaluar, empieza un proceso para identificar si existe un comando y argumentos. El comando indica un grupo relacionado de funciones o procedimientos a ejecutar. Mientras que los argumentos son valores numéricos que permite introducir datos con lo cual se establece características o parámetros que se deben ejecutar en una función o procedimiento.

Los comandos principales son drawO, clear, graph, utils:

- drawO: Es un comando para dibujar elementos gráficos como puntos, líneas, triángulos, rectángulos, círculos, polígonos, estrella, curvas de Bezier. Éstos pueden ser calificados por “Estados” como estáticos y dinámicos.

  Los elementos estáticos son elementos geométricos que no se moverán dentro de los escritorios por acción de algunas teclas especiales como direccionales. Mientras que los elementos dinámicos si se mueven debido a la acción de alguna tecla especial.

El comando “**drawO**” contiene las siguientes opciones:

Para el escritorio 1:

Opción 1: Dibujar punto.
Argumentos: Arg1: posición X, Arg2: Posición Y, Arg3: Color, Arg4: Estado

Opción 2: Dibujar línea.
Argumentos: Arg1: posición inicial X, Arg2: Posición inicial Y, Arg3: Posición final X, Arg4: Posición final Y, Arg5: Color, Arg6: Estado

Opción 3: Dibujar triángulo.
Argumentos: Arg1: posición X1, Arg2: posición Y1, Arg3: Posición X2, Arg4: Posición Y2, Arg5: posición X3, Arg6: posición Y3, Arg7: Color, Arg8: Estado

Opción 4: Dibujar Rectángulo.
Argumentos: Arg1: posición inicial X, Arg2: Posición inicial Y, Arg3: Posición final X, Arg4: Posición final Y, Arg5: Color, Arg6: Estado

Opción 5: Dibujar Circulo.

Argumentos: Arg1: posición X, Arg2: Posición Y, Arg3: Radio, Arg4: Color, Arg5: Estado

Opción 6: Dibujar Polígono.

Argumentos: Arg1: Cx, Arg2: Cy, Arg3: radio, Arg4: lados, Arg5: Angulo inicial, Arg6: Color, Arg7: Estado

Opción 7: Dibujar Estrella.

Argumentos: Arg1: Cx, Arg2: Cy, Arg3: radio, Arg4: lados, Arg5: Angulo inicial, Arg6: Color, Arg7: Estado

Opción 40: Dibujar curva de Bezier de 3 puntos en escritorio 1

Argumentos: Arg1: P1X, Arg2: P1Y, Arg3: P2X, Arg4: P2Y, Arg5: P3X, Arg6: P3Y, Arg7: Color, Arg8: Estado

Opción 41: Dibujar curva de Bezier de 4 puntos en escritorio 1

Argumentos: Arg1: P1X, Arg2: P1Y, Arg3: P2X, Arg4: P2Y, Arg5: P3X, Arg6: P3Y, Arg7: P4X, Arg8: P4Y, Arg9: Color, Arg10: Estado

Opción 42: Dibujar curva de Bezier de multipuntos en escritorio 1 (Max. 10)

Argumentos: Arg1: Cantidad de puntos, Arg2: Color, Arg3: Estado

Opción 43: Agregar punto de Bezier a array en escritorio 1 (Max. 10)

Argumentos: Arg1: indice, Arg2: PX, Arg3: PY

Para el escritorio 2:

Opción 11: Dibujar pixel.

Argumentos: Arg1: posición X, Arg2: Posición Y, Arg3: Color, Arg4: Estado

Opción 12: Dibujar línea.

Argumentos: Arg1: posición inicial X, Arg2: Posición inicial Y, Arg3: Posición final X, Arg4: Posición final Y, Arg5: Color, Arg6: Estado

Opción 13: Dibujar triángulo.

Argumentos: Arg1: posición X1, Arg2: posición Y1, Arg3: Posición X2, Arg4: Posición Y2, Arg5: posición X3, Arg6: posición Y3, Arg7: Color, Arg8: Estado

Opción 14: Dibujar Rectángulo.

Argumentos: Arg1: posición inicial X, Arg2: Posición inicial Y, Arg3: Posición final X, Arg4: Posición final Y, Arg5: Color, Arg6: Estado

Opción 15: Dibujar Circulo.

Argumentos: Arg1: posición X, Arg2: Posición Y, Arg3: Radio, Arg4: Color, Arg5: Estado

Opción 16: Dibujar Polígono.
Argumentos: Arg1: Cx, Arg2: Cy, Arg3: radio, Arg4: lados, Arg5: Angulo inicial, Arg6: Color, Arg7: Estado

Opción 17: Dibujar Estrella.
Argumentos: Arg1: Cx, Arg2: Cy, Arg3: radio, Arg4: lados, Arg5: Angulo inicial, Arg6: Color, Arg7: Estado

Opción 50: Dibujar curva de Bezier de 3 puntos en escritorio 2
Argumentos: Arg1: P1X, Arg2: P1Y, Arg3: P2X, Arg4: P2Y, Arg5: P3X, Arg6: P3Y, Arg7: Color, Arg8: Estado

Opción 51: Dibujar curva de Bezier de 4 puntos en escritorio 2
Argumentos: Arg1: P1X, Arg2: P1Y, Arg3: P2X, Arg4: P2Y, Arg5: P3X, Arg6: P3Y, Arg7: P4X, Arg8: P4Y, Arg9: Color, Arg10: Estado

Opción 52: Dibujar curva de Bezier de multipuntos en escritorio 2 (Max. 10)
Argumentos: Arg1: Cantidad de puntos, Arg2: Color, Arg3: Estado

Opción 53: Agregar punto de Bezier a array en escritorio 2 (Max. 10)
Argumentos: Arg1: indice, Arg2: PX, Arg3: PY

- clear: El comando clear, sirve para borrar todos los elementos gráficos en los escritorios. Además de incorporar que se pueda eliminar los elementos estáticos y dinámicos.

  El comando "**clear**" contiene las siguientes opciones:

  Opción 0: Borrar todo el escritorio 1 y 2.

  Opción 1: Borrar todo el escritorio 1.

  Opción 2: Borrar todo el escritorio 2.

  Para el escritorio 1

  Opción 10: Borrar todos los elementos estáticos.

  Opción 11: Borrar algún elemento estático según índice.
  Arg1: Índice

  Opción 20: Borrar todos los elementos dinámicos.

  Opción 21: Borrar algún elemento dinámico según índice.

Arg1: Índice

- graph: Este comando permite realizar gráficos de ecuaciones de grado n con o sin plano cartesiano, además de tener capacidad de ejecutar gráficos en tiempo real que se actualizan cada 250 ms.

El comando "**graph**" contiene las siguientes opciones:

Para el escritorio 1:

Opción 1: Crear plano cartesiano, con cierta resolución (limites X y Y).
Arg1: Resolución

Opción 2: Crear gráfico de ecuaciones de grado n sin plano cartesiano.
Arg1: Resolución, Arg2: valor 1, Arg3: valor 2, … Arg13: valor 12

Opción 3: Crear gráfico de ecuaciones de grado n con plano cartesiano.
Arg1: Resolución, Arg2: valor 1, Arg3: valor 2, … Arg13: valor 12

Para el escritorio 2:

Opción 11: Crear plano cartesiano, con cierta resolución (limites X y Y).
Arg1: Resolución

Opción 12: Crear gráfico de dos dimensiones sin plano cartesiano.
Arg1: Resolución, Arg2: valor 1, Arg3: valor 2, … Arg13: valor 12

Opción 13: Crear gráfico de dos dimensiones con plano cartesiano.
Arg1: Resolución, Arg2: valor 1, Arg3: valor 2, … Arg13: valor 12

Para escritorio 1 o 2

Opción 20: Crear gráficos de líneas o panel de indicadores digitales.
Arg1: Activación temporizador Tiempo Real, Arg2: Escritorio, Arg3: Tipo de gráfico (Gráfico de líneas para analógico, matriz de lectura de bajos y altos para digital), Arg4. Número de pin analógico

Opción 21: Activar o desactivar tiempo real.
Arg1: Activación temporizador Tiempo Real (0 o 1)

Opción 22: Remover gráfico en escritorio 1 o 2.
Arg1: Escritorio 1 o 2

- utils: Es el comando para ejecutar operaciones de utilidad como establecer el estado de los pines digitales y analógicos, escribir datos de alto o bajo, además de leer datos en los pines. También se ha incorporado un delay para crear un retraso en el tiempo como un delay normal, pero esta vez para ser ejecutado desde una instrucción.

  El comando "**utils**" contiene las siguientes opciones:

  Opción 1: Activación de pines analógicos y digitales
  Arg1: Tipo digital o analógico, Arg2: número de pin, Arg3: modo (input, output, pull input)

  Opción 2: Escribir un dato en un pin analógico o digital
  Arg1: Tipo digital o analógico, Arg2: número de pin, Arg3: valor (low, high)

  Opción 3: Leer un dato en un pin analógico o digital
  Arg1: Tipo digital o analógico, Arg2: número de pin

  Opción 4: crear un delay
  Arg1: tiempo en milisegundos

Cuando se envía a ejecutar un conjunto de instrucciones, es para realizar una gran cantidad de operaciones como representar un conjunto de elementos gráficos en la pantalla, así como cuando se desea elaborar un gráfico en tiempo real en donde aparezca un plano cartesiano con puntos que se van moviendo en el tiempo y entre punto y punto se crean líneas. Se ejecuta siempre instrucción por instrucción, cuando termina una, empieza la otra.

8.2.6. Programación del sistema operativo de entorno gráfico

Se realizó la programación del Arduino UNO (ver ilustración 18. Anexo) y el Arduino Mega (ver ilustración 19. Anexo), haciendo uso del Arduino IDE y también se empleó el lenguaje de programación C++ con el propósito de cumplir los requisitos solicitados por la empresa. Se utilizó librerías como PS2Keyboard_master, vgaxua_master para poder conectar las tarjetas Arduino con los dispositivos de entrada y salida respectivamente.

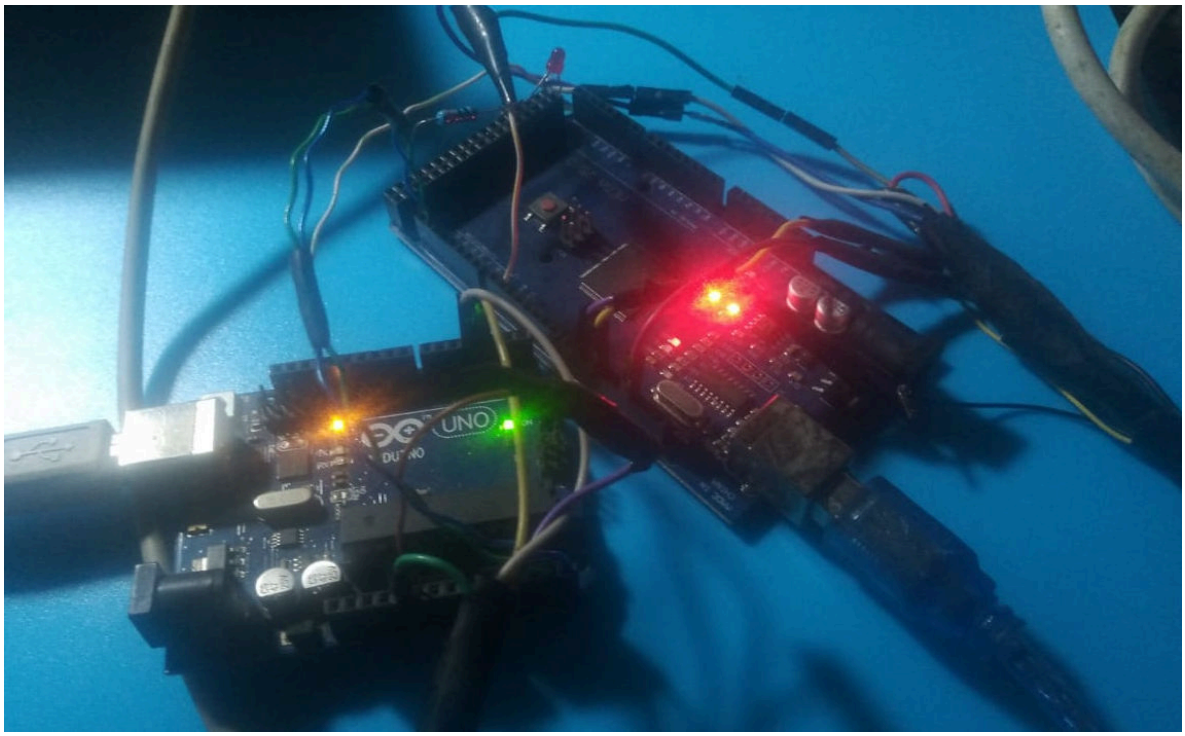
8.3. Validación del funcionamiento del sistema operativo de entorno gráfico por medio de una comparación entre resultados y requisitos del sistema operativo

8.3.1. Conexión física de los dispositivos electrónicos

La conexión física de los dispositivos se puede observar en la figura 14, en donde se observa las tarjetas de desarrollo Arduino Mega y Arduino UNO, la conexión con los dispositivos monitor VGA y teclado PS/2.

Figura 14.

Conexión física del Arduino UNO, Arduino MEGA, monitor VGA y teclado PS/2



Fuente propia.

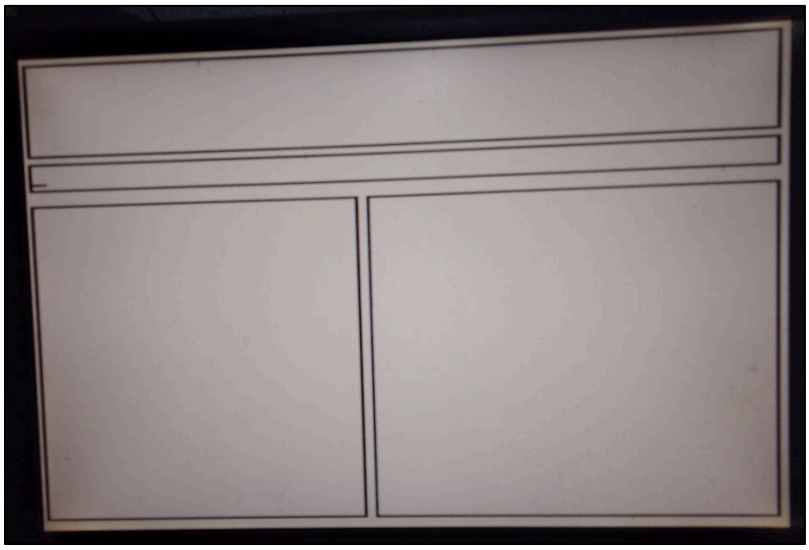
8.3.2. Comparación entre resultados y los requisitos del sistema operativo

En la figura 15, se observa cómo está distribuido la pantalla a como se define en la figura 6, en donde la pantalla del monitor se puede ver la caja de texto, el

historial y los 2 escritorios. En la figura 16, se realiza un mensaje, saludando a la universidad, la carrera, el turno, la materia y la fecha de prueba.

Figura 15.

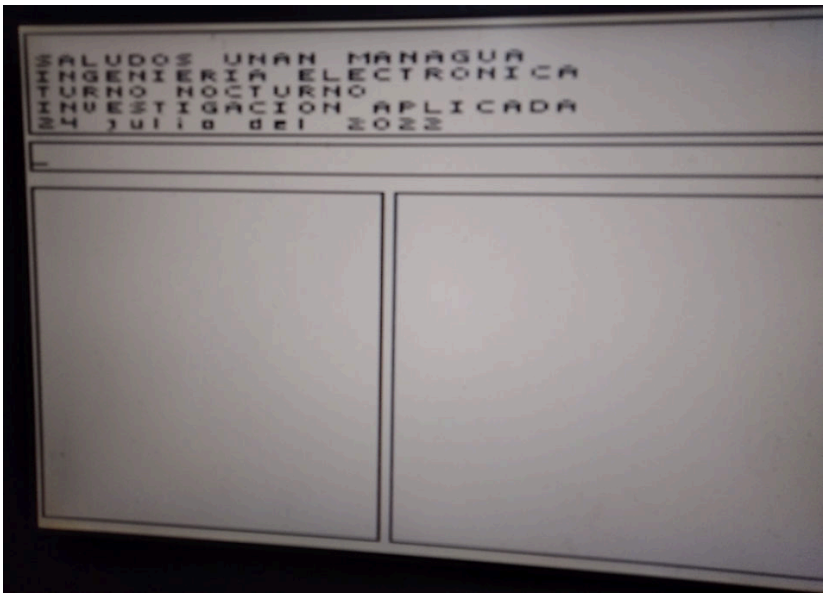
Monitor presentando el sistema operativo en pantalla.



Fuente propia.

Figura 16.

Introducción de un saludo inicial



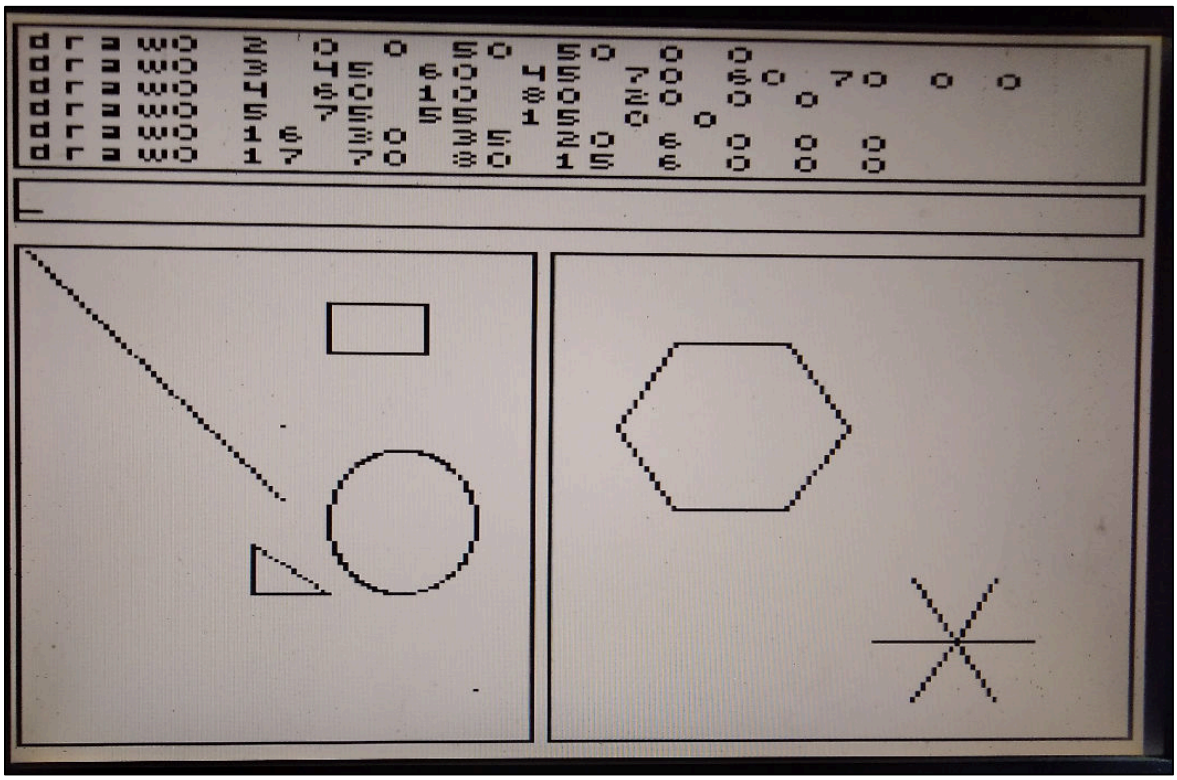
Fuente propia.

Se introducen las siguientes instrucciones y se obtiene lo que se ve en la figura 17, se puede comparar con la figura 8, donde se observan las similitudes:

drawO 1 90 90 0 0
drawO 2 0 0 50 50 0 0
drawO 3 45 60 45 70 60 70 0 0
drawO 4 60 10 80 20 0 0
drawO 5 75 55 15 0 0
drawO 16 30 35 20 6 0 0 0
drawO 17 70 80 15 6 0 0 0

Figura 17.

Elementos dibujados en la pantalla mediante el sistema operativo de entorno gráfico mediante instrucciones



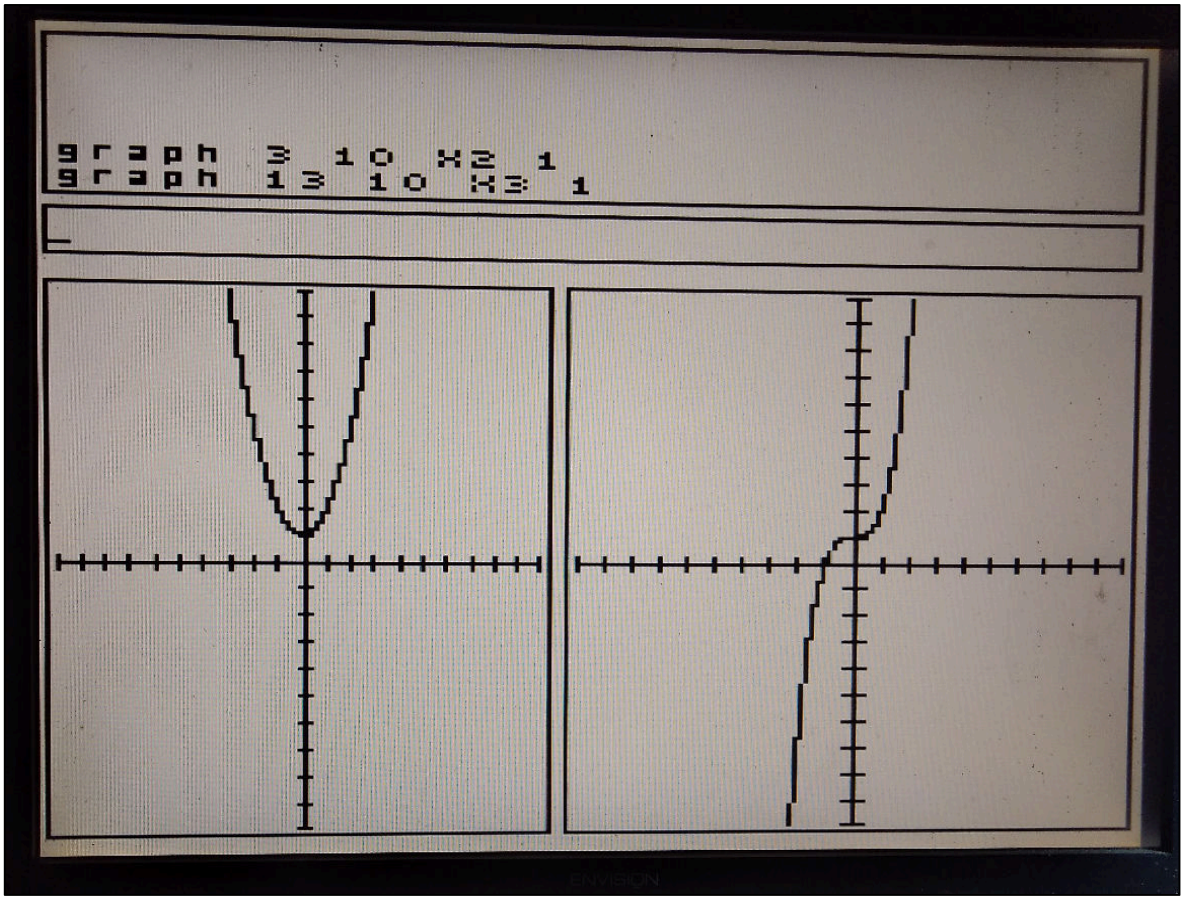
Fuente propia.

Se introducen las siguientes instrucciones para crear un gráfico a partir de una ecuación y se obtiene lo que se ve en la figura 18 que se puede comparar con la figura 9:

graph 3 10 X2 1
graph 13 10 X3 1

Figura 18.

Gráficos con ecuaciones de grados n dibujados mediante el sistema operativo de entorno gráfico



Fuente propia.

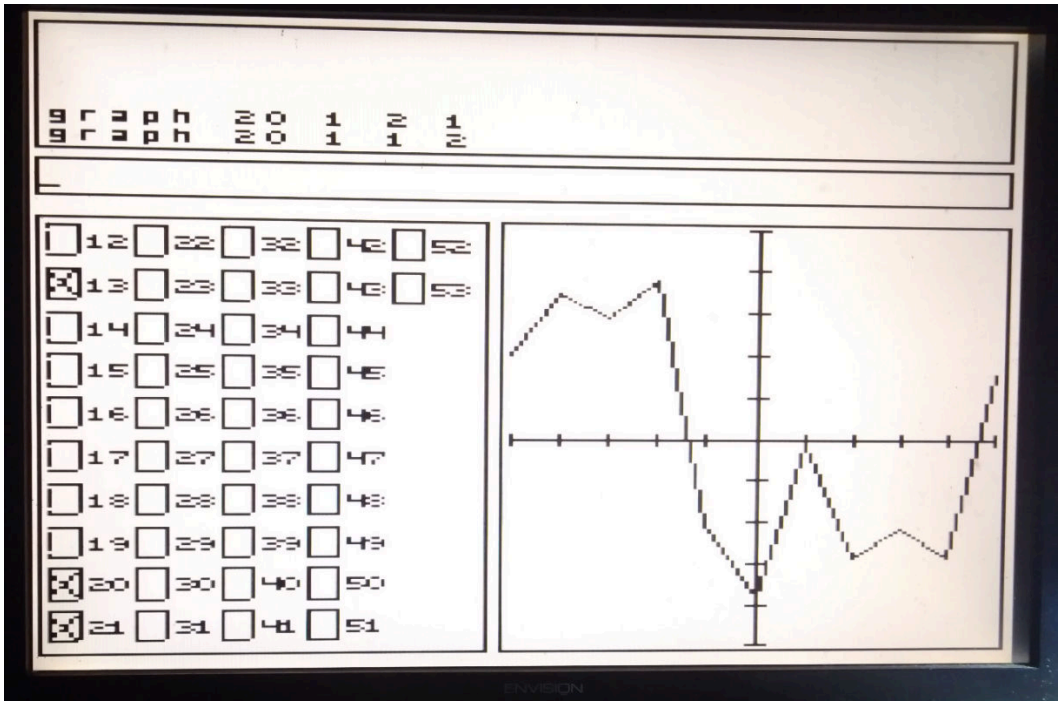
Se introducen las siguientes instrucciones para mostrar gráficos en tiempo real y se obtiene lo que se ve en la figura 19, que se puede comparar con la figura

10, sin embargo, hay que tomar en cuenta, de que éste gráfico se va modificando en un tiempo prudencial de actualización (cada 250 ms).

En la parte izquierda de la figura 19, se puede observar los datos de los pines digitales, se toma desde el pin 12 hasta el 53 debido a que, si se hace entre el pin 0 hasta el 11, daba problemas de estabilidad, debido al uso del timer0 del propio Arduino. Mientras que en la parte derecha se puede ver un gráfico dinámico que se actualiza cada segundo, este puede mostrar datos analógicos con valores de punto decimal. Por motivo de prueba, se realizó con valores aleatorios entre 0 a 5V, simulando datos analógicos.

graph 20 1 2 1
graph 20 1 1 2

Figura 19.
Gráficos en tiempo real, mostrando datos para pines digitales en escritorio 1 y datos de pines analógicos en escritorio 2.



Fuente propia.

8.3.3. Análisis de resultados obtenidos

Mediante la programación realizada, se ha utilizado la tarjeta de desarrollo de Arduino Mega para recibir datos de entrada de un teclado PS/2, y mostrar información en un monitor VGA, se procesa la información como un procesador de texto y se identifica las instrucciones para que se ejecuten operaciones para mostrar en la pantalla información conveniente de un sistema electrónico.

Es importante, que el sistema operativo, al evaluar una instrucción logra entender y separar los comandos y argumentos para que se ejecuten operaciones o funciones o tareas.

En la validación del sistema operativo también se muestra que se pueden crear elementos gráficos en pantalla desde lo más básico como puntos, líneas, triángulos, rectángulos, círculos, hasta polígonos y estrellas; hasta cosas más complejas, como gráficos tomando una ecuación de grado n y crear gráficos en tiempo real tomando variables como los datos en los pines y poder mostrar esto en pantalla, ya sean datos analógicos como digitales. Mediante los gráficos en tiempo real, se pudo comprobar que si se puede utilizar esto para ejecutar un conjunto de instrucciones.

En lo demás, si queda mucho por lo cual avanzar en desarrollar esto con sistemas de máquinas y poder mostrar más información conveniente. Pero esto quedará para un futuro, haciendo aplicaciones específicas y a la medida de una maquinaria.

9. CONCLUSIONES

Con el diagnóstico realizado mediante una entrevista hecha al gerente de operaciones de la empresa UNITRIUM, se pudo conocer a lo que se dedica la empresa, su proyección y su dirección en un futuro. Además, se obtuvo información sobre los requisitos del producto tecnológico que la empresa UNITRIUM ha solicitado que se desarrolle, que es un sistema operativo de entorno gráfico para que funcione en la tarjeta de desarrollo Arduino Mega.

Se diseñó el sistema operativo de entorno gráfico, se conectó físicamente los dispositivos electrónicos de entrada (teclado) y salida (monitor), se diseñó un modelo funcional del sistema, se estructura las instrucciones (comandos y argumentos), se realizó la programación con el software Arduino IDE y se usó el lenguaje de programación C++.

Se validó el sistema operativo de entorno gráfico introduciendo las instrucciones y en pantalla se graficó elementos tan sencillos como puntos, líneas, triángulos, círculos, polígonos, además de ejecutar funciones, procedimientos y aplicaciones como graficar ecuaciones de grado n y realizar de gráficos en tiempo real utilizando datos de los valores en que se encuentran los pines en digital y visualizar la lectura en bajos o altos y en datos analógicos con valores desde 0 hasta 5V.

El sistema operativo de entorno gráfico funciona correctamente en la tarjeta de desarrollo Arduino Mega y sirve como un producto de la empresa UNITRIUM para que sea comercializado u ofrecerlo como una distribución especifica según los requerimientos que tenga cualquier cliente o empresa.

10. RECOMENDACIONES

Se debe diseñar sistemas operativos de entorno gráfico para otras tarjetas de desarrollo que tienen mayor capacidad como ESP32 o Raspberry PI. Ya que la tarjeta de desarrollo Arduino Mega tiene bastante limitación con la memoria SRAM, y al usar un monitor y la respectiva librería, se utiliza mucho éste recurso para almacenar datos de variables que requieren ser dinámicas y no se pueden guardar en la memoria EEPROM. También al usar otras librerías en conjunto saturan la capacidad de la memoria SRAM. Si se usan tarjetas de desarrollo con más capacidad se puede mejorar la experiencia del usuario con el sistema operativo.

Se debe profundizar en añadir nuevas funciones al sistema operativo de entorno gráfico, específicamente dirigidas a facilitar aplicaciones para procesos de fabricación.

Si alguien desea utilizar éste sistema operativo para un proyecto en específico o realizar aplicaciones, debe tener en cuenta el uso de las librerías que funcionan con el mismo, éstas tienen sus funciones propias, por lo cual, debe estudiarlas muy bien, para poder usarlas correctamente, además de profundizar en el uso del tiempo, temporizadores y las sincronizaciones verticales y horizontales para la formación de pixeles con la librería vgaxua. Incluso, tener conocimiento muy avanzado en el lenguaje C++.

11. BIBLIOGRAFÍA

Sánchez, S. (9 de diciembre, 2013). Los microcontroladores de hoy en día. *Microcontroladores.*https://microcontroladoressesv.wordpress.com/losmicrocontroladores-de-hoy-en-dia/)

Petersen, C. (s.f). Características del microprocesador. *Techlandia.* https://techlandia.com/caracteristicas-del-microprocesador-8086-lista_46568/

Hernández, S. (18 de abril, 2019) Microcontroladores. *EcuRed.* https://www.ecured.cu/Microcontrolador

Etece, E. (05 de Agosto, 2021). Lenguaje de programación. *Conceptos*. https://concepto.de/lenguaje-de-programacion/

Cachaza, P. (07 de junio, 2022). Tipos de lenguaje de programación. *Assembler*. https://assemblerinstitute.com/blog/tipos-lenguaje-programacion/

Prats, A. (2015). Lenguaje de programación: tipos y características. *Chakray.* https://www.chakray.com/es/lenguajes-programacion-tipos-caracteristicas/

Navas, Miguel. (01 de marzo, 2011). Que es el hardware: para que sirve y definición. *Profesional review.* https://www.profesionalreview.com/hardware/

Universidad autónoma intercultural de Sinaloa, (2018). Dispositivos de entrada, salida y de almacenamiento. https://es.slideshare.net/AgustinRamirez48/dispositivos-de-entrada-salida-y-de-almacenamiento

Palacios, M. (2016). Que es un puerto de computadora. *Cavsi*. https://www.cavsi.com/preguntasrespuestas/que-es-un-puerto-de-computadora-definicion/

Rockcontent.com (20 de abril, 2019). Que es un lenguaje de programación. Obtenido de https://rockcontent.com/es/blog/que-es-un-lenguaje-de-programacion

Etece. (2021). Sistema operativo. *Concepto*. https://concepto.de/sistema-operativo/

Lenis, A. (2021) que es una interfaz gráfica. *Hubspot*. https://blog..es/marketing/interfaz-usuario

Arduino.cl. (s.f) que es Arduino. *arduino.cl* https://arduino.cl/que-es-arduino/#:~:text=Arduino%20Naci%C3%B3%20en%20el%20a%C3%B1o,uso%20interno%20de%20la%20escuela.

witronica.com (27 de junio 2016). Microcontrolador. http://witronica.com/microcontroladores

diyi0t.com (2020). Pin Out Arduino UNO. https://diyi0t.com/arduino-uno-tutorial/

diyi0t.com (2020). Pin Out Arduino Mega. https://diyi0t.com/arduino-mega-tutorial/

w140.com (s.f). VGA Connector. https://w140.com/tekwiki/wiki/VGA_Connector

pjrc.com (2015). PS2Keyboard library. https://www.pjrc.com/teensy/td_libs_PS2Keyboard.html

hardzone.es (s.f) PS2 Connector. https://hardzone.es/2018/02/18/ps-2-usb-teclado-raton/

picmania.garcia-cuervo.net (s.f). Teclado alfanumérico con código ASCII. http://picmania.garcia-cuervo.net/proyectos_teclado_ps2.htm

computer-engineering.org (s.f). Configuración de pines macho y hembra PS/2. http://computer-engineering.org/ps2protocol/

josstic.files.wordpress.com (s.f) Código ASCII. https://josstic.files.wordpress.com/2019/03/cc3b3digo-ascii.jpg

luisllamas.es (2016). Protocolo I2C. https://www.luisllamas.es/arduino-i2c/

Arduino.cc (2022). Protocolo I2C para Arduino. https://www.arduino.cc/reference/en/language/functions/communication/wire/

Gammon, N. (2012). Arduino UNO output to VGA monitor. http://www.gammon.com.au/forum/?id=11608

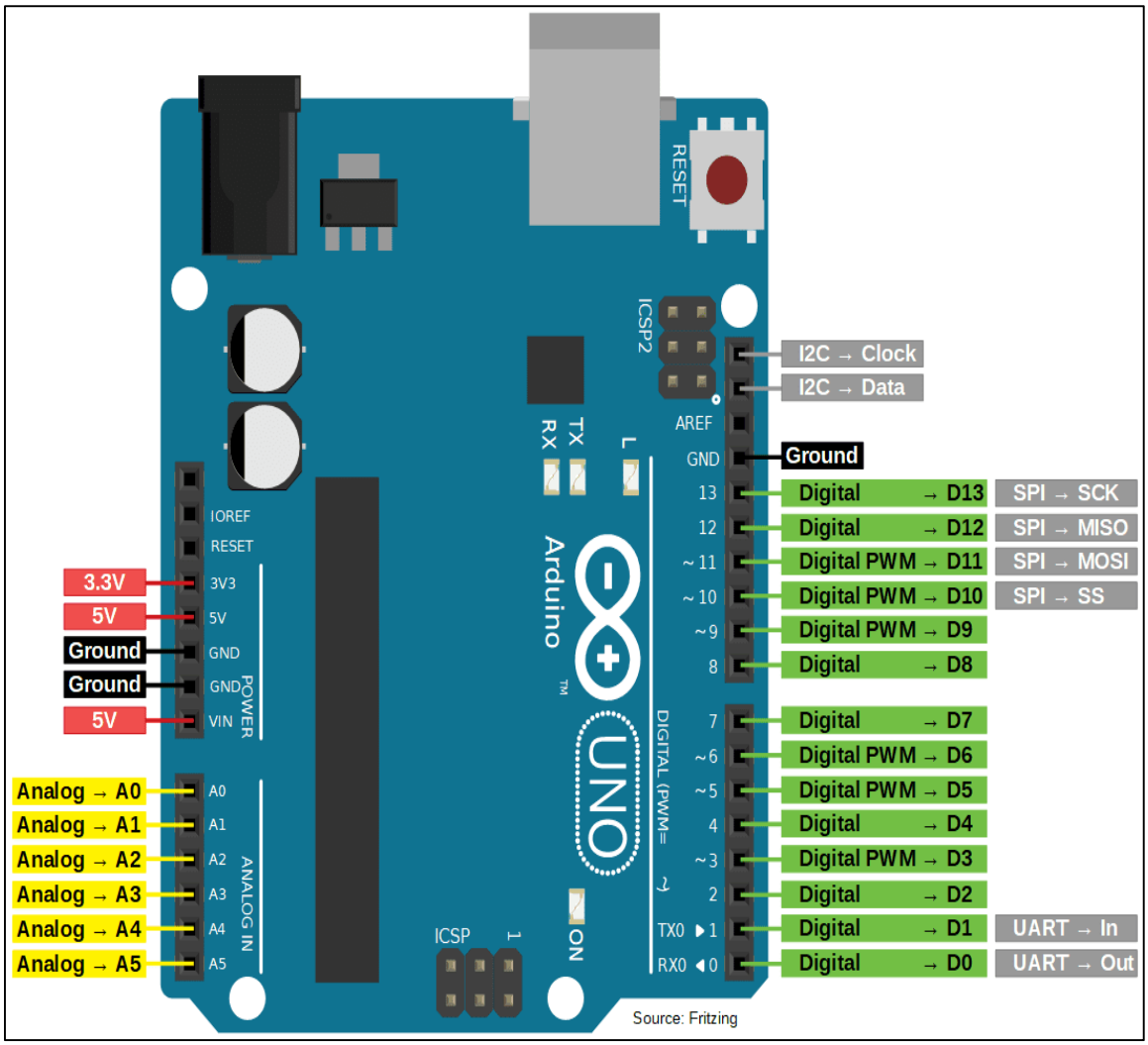
Maffiodo, S. (2019). VGAXUA library. https://github.com/smaffer/vgaxua

Stoffregen, P. (24 de febrero de 2022) https://github.com/PaulStoffregen/PS2Keyboard

12. ANEXOS

Ilustración 1.

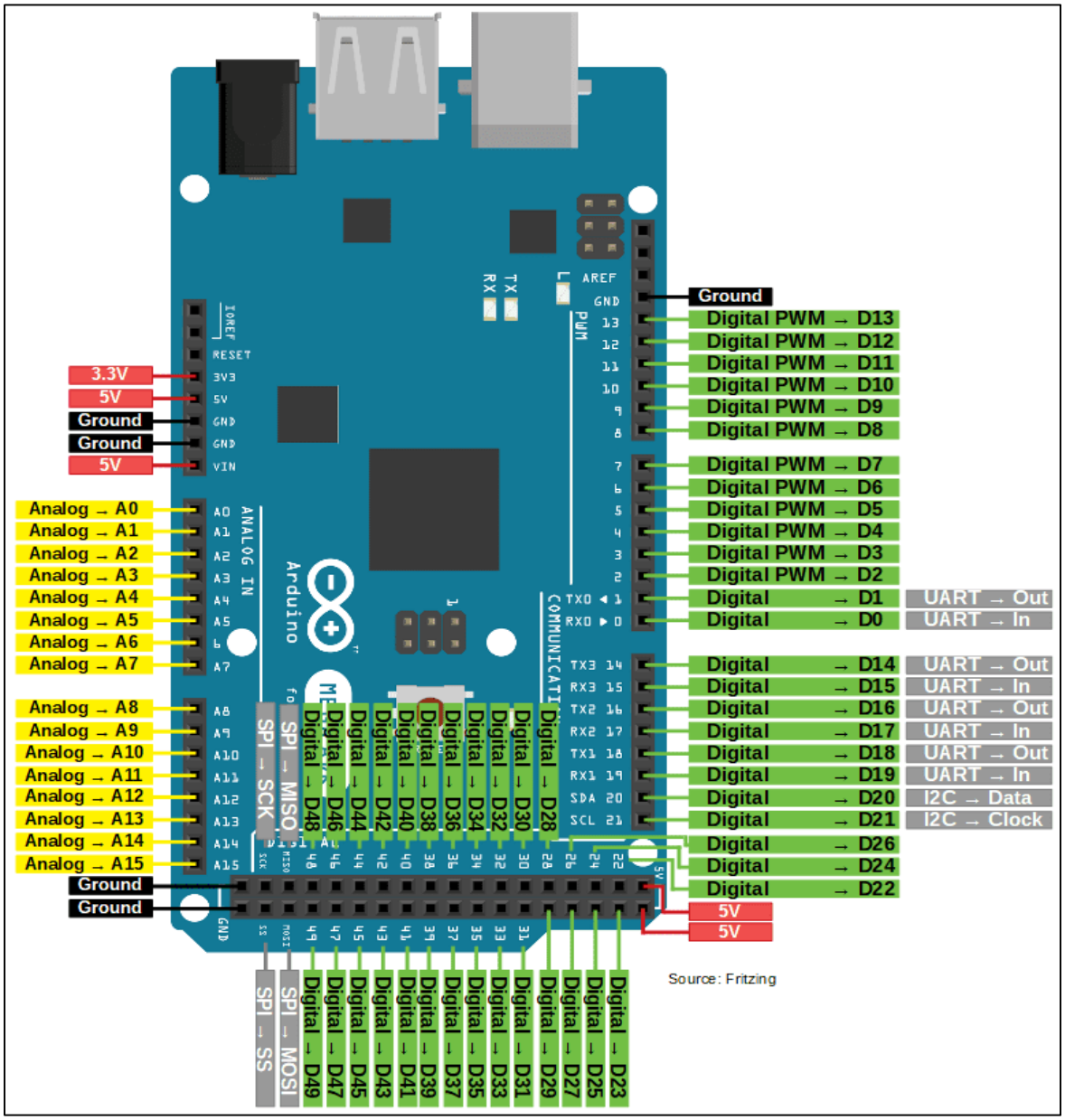
Circuito integrado Arduino UNO y especificación de pines.



Fuente: diyi0t.com (2020)

Ilustración 2.

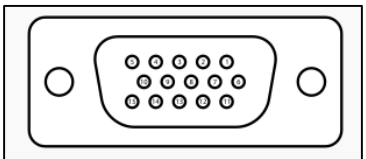
Circuito integrado Arduino Mega y especificación de pines.



Fuente: diyi0t.com (2020)

Ilustración 3.

Pines DE15-hembra



Un conector DE15 hembra.

Pin 1	RED	Canal Rojo
Pin 2	GREEN	Canal Verde
Pin 3	BLUE	Canal Azul
Pin 4	N/C	Sin contacto
Pin 5	GND	Tierra (HSync)
Pin 6	RED_RTN	Vuelta Rojo
Pin 7	GREEN_RTN	Vuelta Verde
Pin 8	BLUE_RTN	Vuelta Azul
Pin 9	+5 V	+5 V (Corriente continua)
Pin 10	GND	tierra (Sincr. Vert, Corriente continua)
Pin 11	N/C	Sin contacto
Pin 12	SDA	I^2C datos
Pin 13	HSync	Sincronización horizontal
Pin 14	VSync	Sincronización vertical
Pin 15	SCL	I^2Velocidad Reloj

Fuente: w140.com

Ilustración 4.

Monitor VGA.

Fuente: axa.com.ar

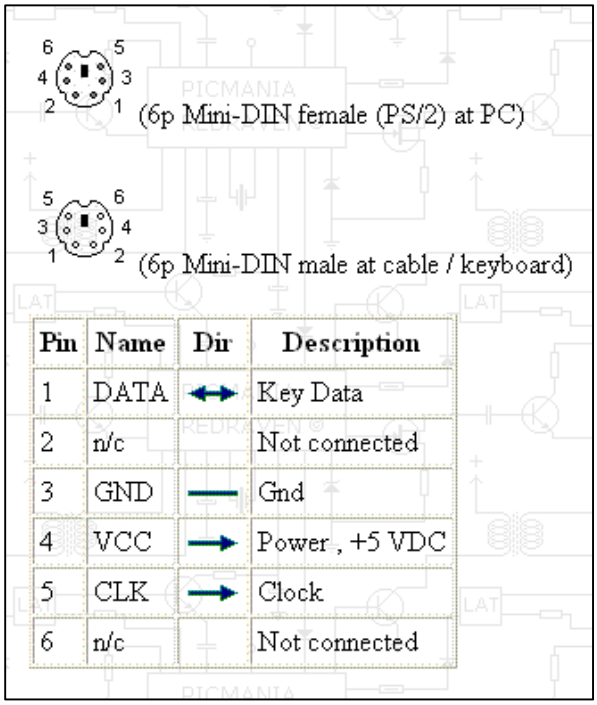
Ilustración 5.

Teclado y conector PS/2.

Fuente: hardzone.es

Ilustración 6.

Configuración de pines macho y hembra PS/2.

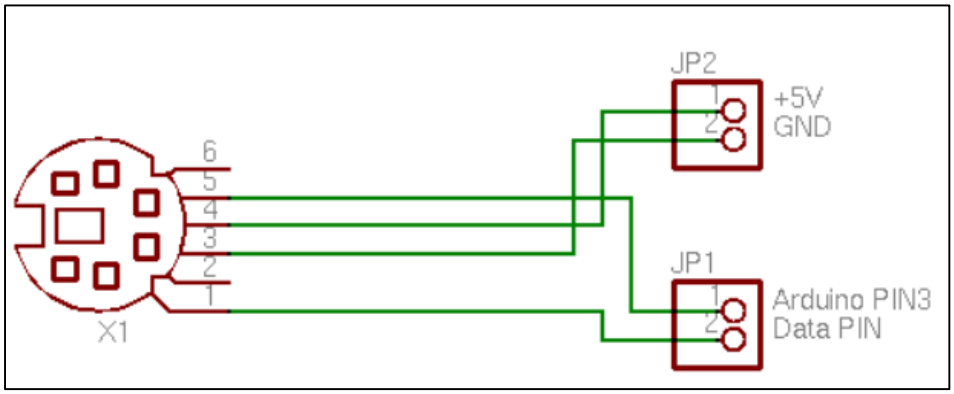


Pin	Name	Dir	Description
1	DATA	↔	Key Data
2	n/c		Not connected
3	GND	—	Gnd
4	VCC	→	Power , +5 VDC
5	CLK	→	Clock
6	n/c		Not connected

Fuente: computer-engineering.org

Ilustración 7.

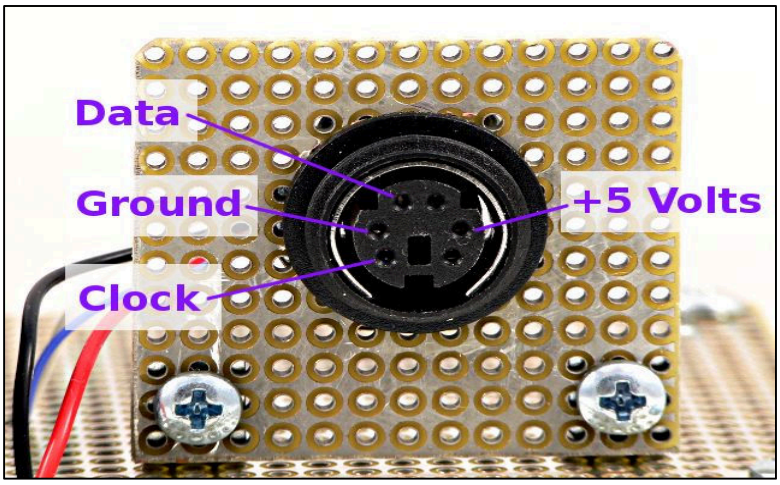
Conexión de puerto PS/2.



Fuente: pjrc.com (2015)

Ilustración 8.

Conexión de pin PS/2 hembra.



Fuente: pjrc.com (2015)

Ilustración 9.

Código ASCII.

Fuente: josstic.files.wordpress.com

Ilustración 10.

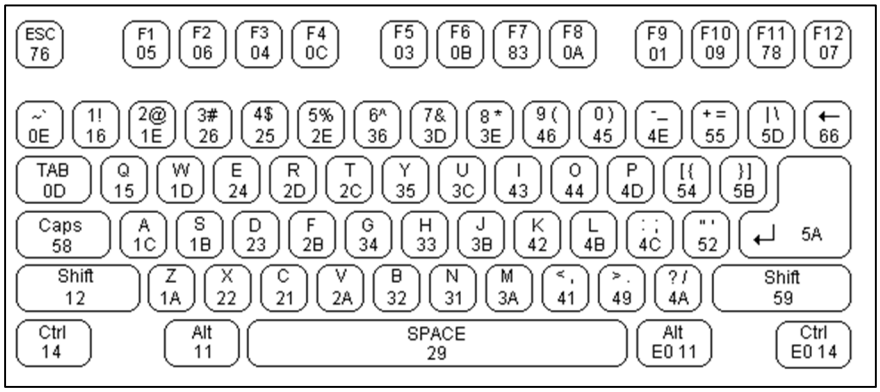
Código de teclado usando librería.

```
#include <PS2Keyboard.h>

const int DataPin = 8;
const int IRQpin =  5;

PS2Keyboard keyboard;

void setup() {
  delay(1000);
  keyboard.begin(DataPin, IRQpin);
  Serial.begin(9600);
  Serial.println("Keyboard Test:");
}

void loop() {
  if (keyboard.available()) {

    // read the next key
    char c = keyboard.read();

    // check for some of the special keys
    if (c == PS2_ENTER) {
      Serial.println();
    } else if (c == PS2_TAB) {
      Serial.print("[Tab]");
    } else if (c == PS2_ESC) {
      Serial.print("[ESC]");
    } else if (c == PS2_PAGEDOWN) {
      Serial.print("[PgDn]");
    } else if (c == PS2_PAGEUP) {
      Serial.print("[PgUp]");
    } else if (c == PS2_LEFTARROW) {
      Serial.print("[Left]");
    } else if (c == PS2_RIGHTARROW) {
      Serial.print("[Right]");
    } else if (c == PS2_UPARROW) {
      Serial.print("[Up]");
    } else if (c == PS2_DOWNARROW) {
      Serial.print("[Down]");
    } else if (c == PS2_DELETE) {
      Serial.print("[Del]");
    } else {

      // otherwise, just print all normal characters
      Serial.print(c);
    }
  }
```

Fuente: pjrc.com (2015)

Ilustración 11.

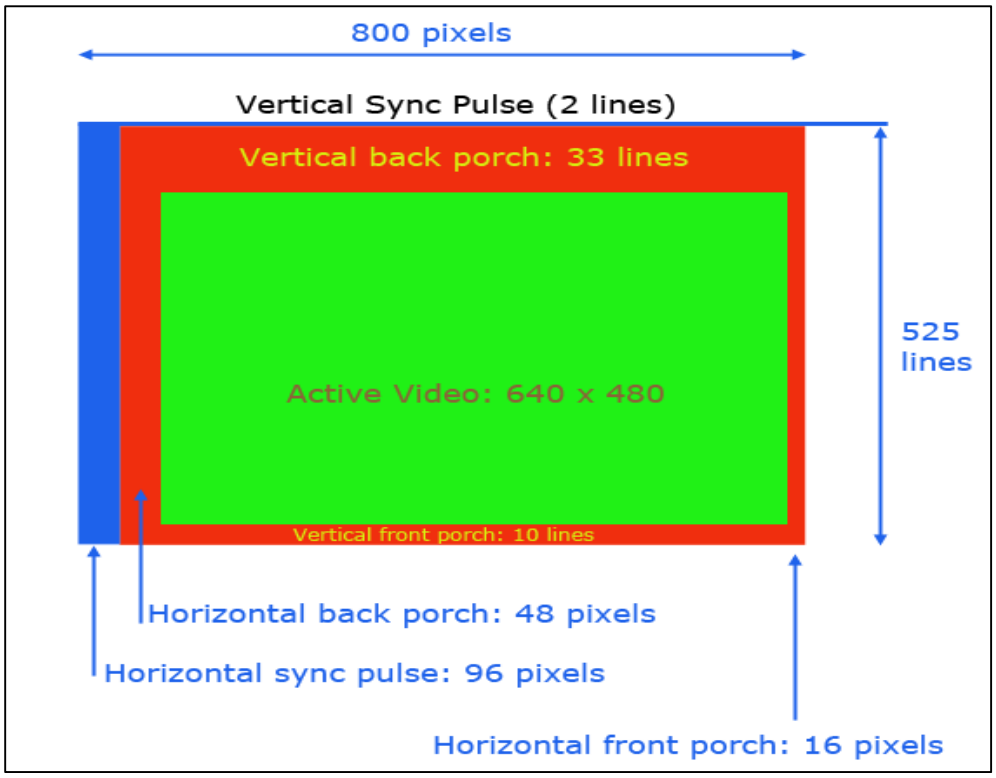
Teclado alfanumérico.



Fuente: picmania.garcia-cuervo.net

Ilustración 12.

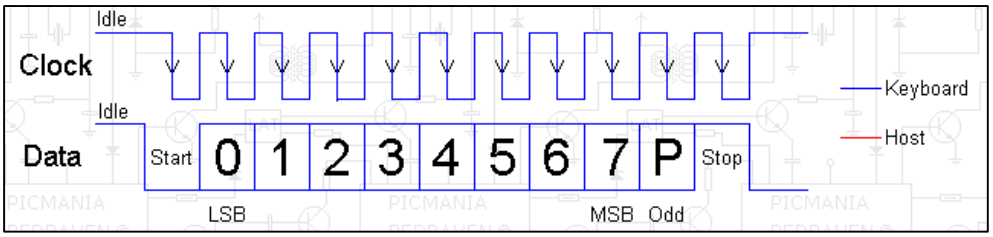
Distribución de la pantalla del monitor.



Fuente: gammon.com.au (2012)

Ilustración 13.

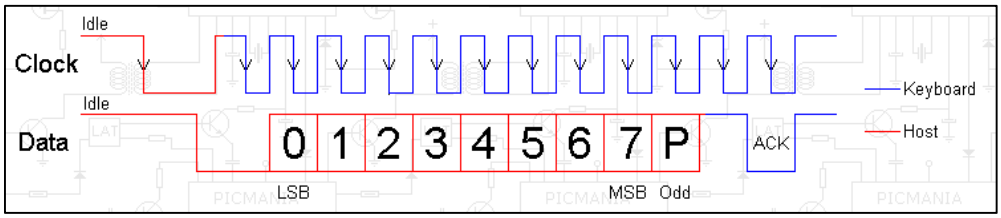
Trama para la comunicación de un conector PS/2.



Fuente: picmania.garcia-cuervo.net

Ilustración 14.

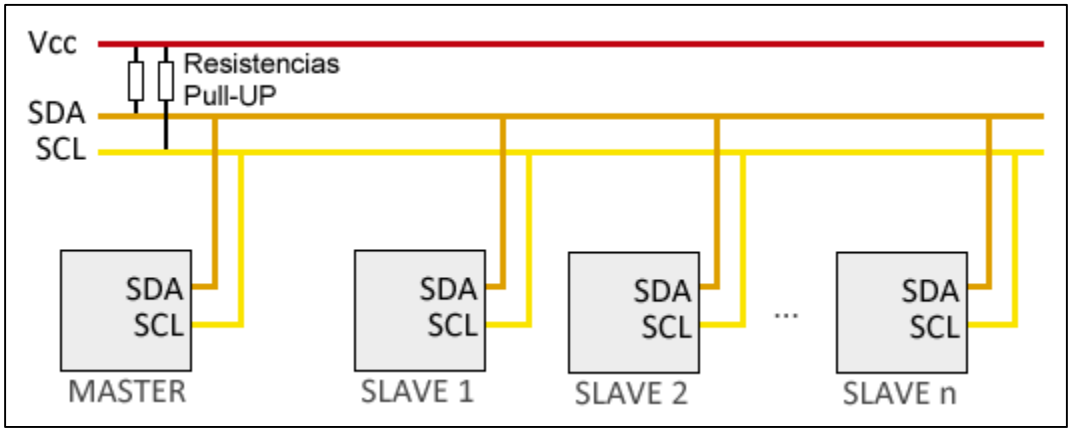
Secuencia de comunicación al teclado PS/2.



Fuente: picmania.garcia-cuervo.net

Ilustración 15.

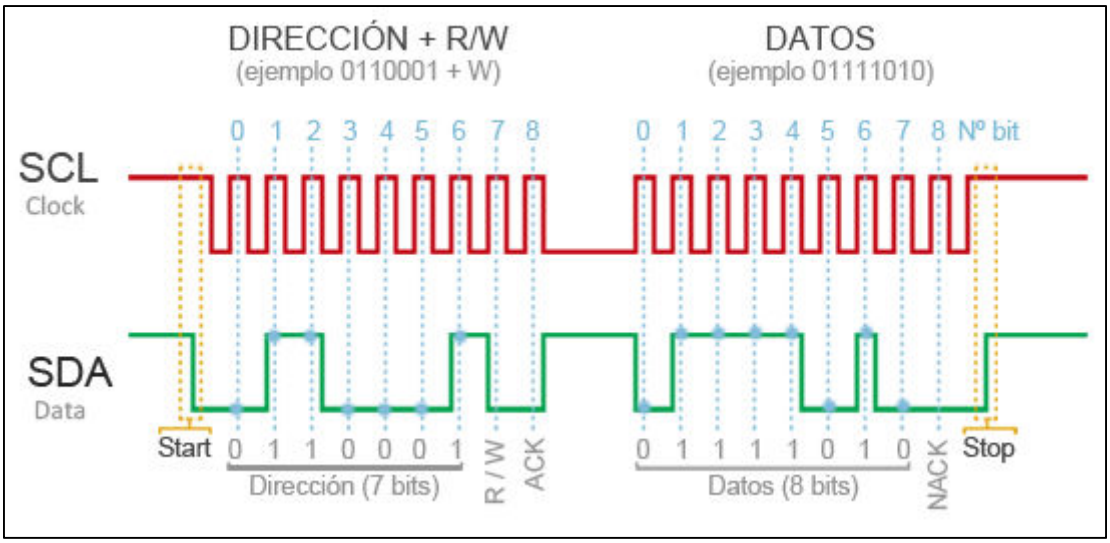
Conexión maestro y esclavo.



Fuente: luisllamas.es (2016)

Ilustración 16.

Trama de envio de mensaje por protocolo I2C.



Fuente: luisllamas.es (2016)

Ilustración 17.

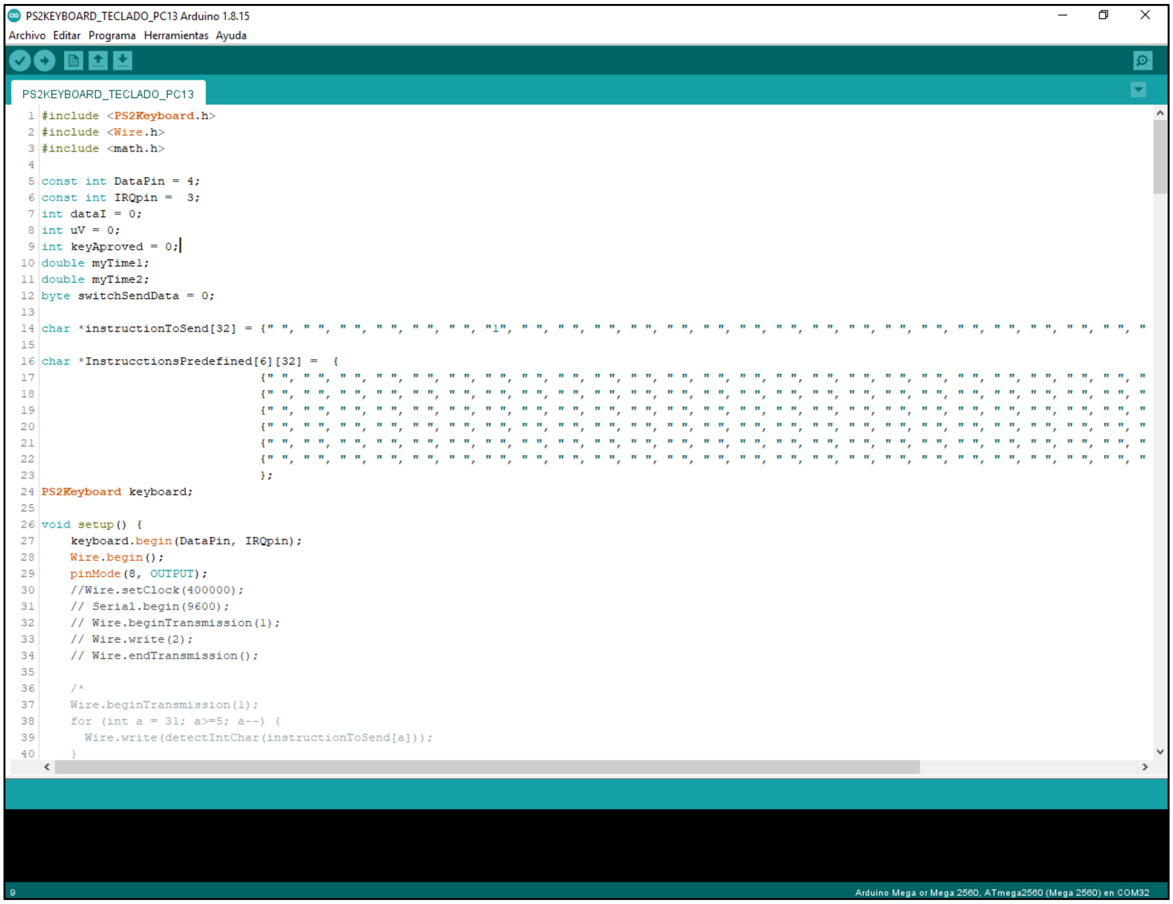
Pines de comunicion I2 para aduino UNO y Mega.

Placa	Pines I2C
Arduino UNO	A4 (SDA), A5 (SCL)
Arduino Mega 2560	20 (SDA), 21 (SCL)

Fuente: arduino.cc (2022)

Ilustración 18.

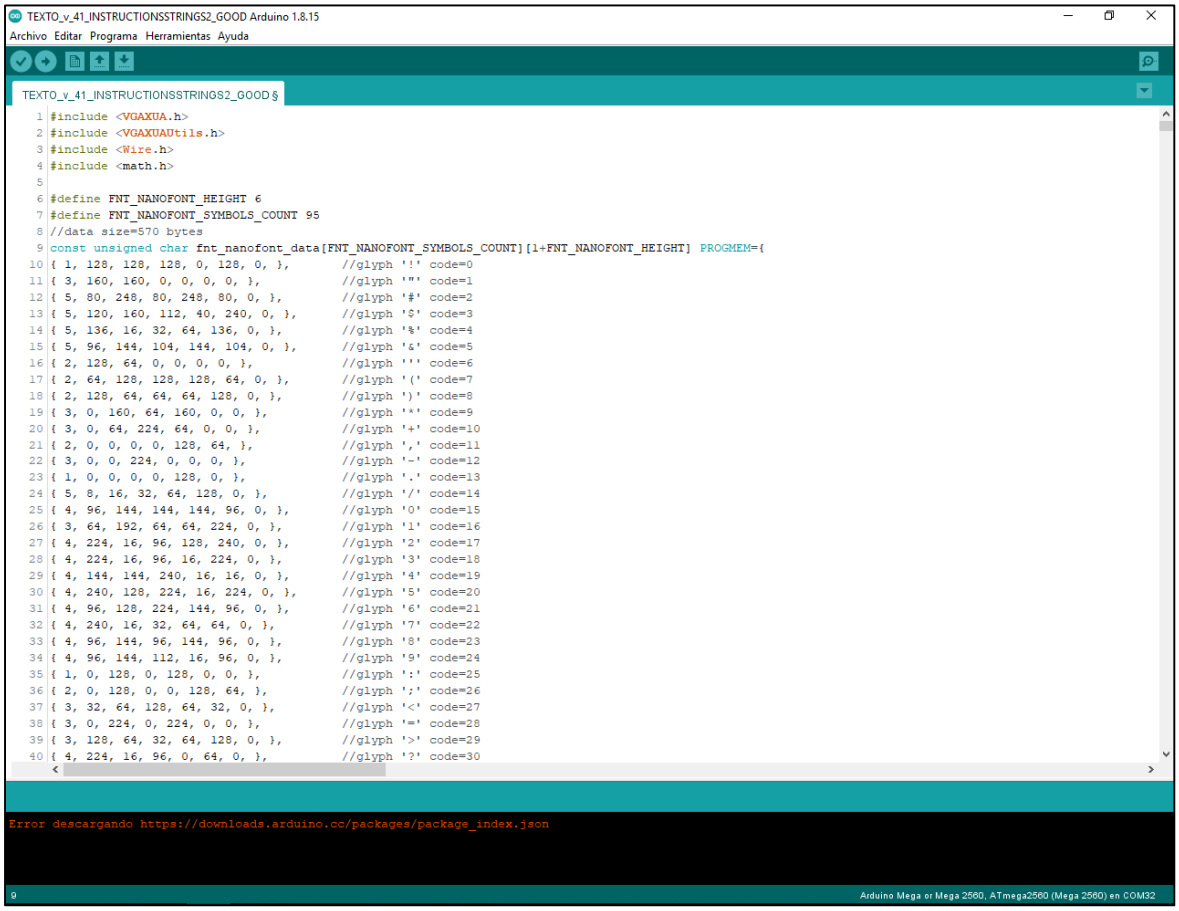
Utilización del software Arduino IDE para programar el Arduino UNO.



Fuente propia.

Ilustración 19.

Utilización del Arduino IDE para programar el Arduino Mega



Fuente propia.

Tabla 2.

Características básicas de la placa Arduino Uno.

Voltaje de operación	5V
Voltaje de entrada (recomendado)	7-12V
Digitales pines I/O	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente DC	40mA
Corriente continua	3.3V pin 50mA
Memoria flash	32kB (ATmega328) de los cuales 0,5 KB son usados para arranque
SRAM	2 KB ATmega328
EEPROM	1 KB ATmega328
Velocidad de reloj	16 MHZ

Fuente: Tapia, C. & Manzano, H. (2013)

Tabla 3.

Características básicas de la placa Arduino ATmega2560.

Voltaje de operación	7 a 12V
Voltaje de entrada (recomendado)	6-20V
Digitales pines I/O	54 (de los cuales 15 proporcionan salida PWM)
Pines de entrada analógica	16
Corriente DC	40mA
Corriente continua	3.3V pin 50mA
Memoria flash	256kB de los cuales 8KB son usados para bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHZ

Fuente: Tapia, C. & Manzano, H. (2013)