

# Diseño de una interfaz gráfica orientada a la simulación de un fermentador

## Design of a graphical interface oriented to the simulation of a fermenter

*Julieta Karina Cruz Vázquez\**, *Humberto Chavez Chavez\*\**, *Francisco Gumaro Ruiz Ruiz\*\*\**

DOI: 10.15446/rev.colomb.biote.v23n1.90212

### RESUMEN

El uso de fermentadores es común en diversos procesos biotecnológicos. Actualmente, un fermentador o biorreactor tiene la capacidad de monitorear variables físicas y químicas las cuales son mostradas al operador por medio de una pantalla. Aunado a los materiales de construcción de un biorreactor, hoy en día su adquisición es costosa e inaccesible para fines educativos y de investigación. Para abordar esta problemática, se diseñó y desarrolló la interfaz Fermenta V1.0. orientada a la simulación de un bioproceso realizado por un fermentador, para realizarla se utilizó una pantalla táctil Nextion y Arduino UNO; para su programación, se implementaron los entornos de desarrollo integrados apropiados. La interfaz fue capaz de emular diferentes dispositivos electrónicos como sensores, válvulas solenoides, bombas peristálticas, iluminación y agitación. En esta primera etapa, se establece el punto de partida para facilitar el acoplamiento de la interfaz Fermenta V1.0 con diversos sensores orientados al monitoreo de un fermentador. Con la incorporación de los dispositivos electrónicos a la interfaz, se podrá adecuar al sistema (interfaz-sensores) a un proceso biotecnológico en donde se requiera el uso de un fermentador y la necesidad de monitorear temperatura, oxígeno disuelto, pH, CO<sub>2</sub>, iluminación, agitación.

**Palabras clave:** Arduino, biotecnología, control, industria, pantalla Nextion

### ABSTRACT

The use of fermenters is common in various biotechnological processes. Currently, a fermenter or bioreactor can monitor physical and chemical variables which are displayed to the operator through a screen. In addition to the construction materials of a bioreactor, nowadays its acquisition is expensive and inaccessible for educational and research purposes. To address this problem, the

---

\* Maestra en Ciencias en Ingeniería Bioquímica con especialidad en Biotecnología Universidad del Mar (UMAR), Representante del Cuerpo Académico: Tecnologías Agropecuarias y Forestales en el Trópico (UMAR-CA-35), Carretera vía Sola de Vega km 1.5, 71980, Puerto Escondido, Oaxaca, México. julieta\_saint@hotmail.com; <https://orcid.org/0000-0001-9356-980X>

\*\* Estudiante de Ingeniería Mecatrónica Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas (UPIITA) del Instituto Politécnico Nacional. Ciudad de México. 98hchch@gmail.com

\*\*\* Maestro en Ciencias en Ingeniería Bioquímica con especialidad en Biotecnología. Universidad del Mar (UMAR), Colaborador del Cuerpo Académico: Tecnologías Agropecuarias y Forestales en el Trópico (UMAR-CA-35), Carretera vía Sola de Vega km 1.5, 71980, Puerto Escondido, Oaxaca, México. ruiz\_ruiz\_francisco@hotmail.com; <https://orcid.org/0000-0002-9176-8845>. Autor de correspondencia.

Fermenta V1.0 interface was designed and developed to simulate a bioprocess performed by a fermenter, using a Nextion touch screen and Arduino UNO; for its programming, the appropriate integrated development environments were implemented. The interface was able to emulate different electronic devices such as sensors, solenoid valves, peristaltic pumps, lighting, and agitation. In this first stage, the starting point is established to facilitate the coupling of the Fermenta V1.0 interface with various sensors oriented to the monitoring of a fermenter. With the incorporation of the electronic devices to the interface, it will be possible to adapt the system (interface-sensors) to a biotechnological process where the use of a fermenter is required and the need to monitor temperature, dissolved oxygen, pH, CO<sub>2</sub>, lighting, agitation.

**Keywords:** Arduino, biotechnology, control, industry, Nextion display.

**Recibido:** octubre 20 de 2020

**Aprobado:** mayo 24 de 2021

## INTRODUCCION

La biotecnología implica el desarrollo de aplicaciones tecnológicas con entidades biológicas, para ser usadas o modificadas para las necesidades del ser humano (Raju 2016). Por su amplio rango de aplicación, la biotecnología se ha clasificado por colores (Kafarski 2012), también han surgido otras clasificaciones de acuerdo a las áreas de uso, como biotecnológico, codificación del ADN, proteínas, procesos biotecnológicos (Matyushenko *et al.*, 2016). En los procesos biotecnológicos, los biorreactores o fermentadores son el componente elemental, tanto en la producción de proteínas, antibióticos y ácidos orgánicos (Mosier & Ladisch 2011), un biorreactor es un recipiente de vidrio o acero inoxidable (Aizawa *et al.* 2007), donde se lleva a cabo la reacción con microorganismos específicos, proporcionando un ambiente controlado para su óptimo desarrollo (Wang & Zhong 2007). Las principales funciones de un biorreactor radican en garantizar la esterilidad durante todo el proceso, la capacidad de introducción de reactivos líquidos y gaseosos, eliminación de productos, control del entorno (temperatura, velocidad de agitación y pH) y generar un mezclado perfecto (Liu 2020).

Los biorreactores agitados mecánicamente, son los más ampliamente utilizados en la industria (Kracík *et al.*, 2020). En un proceso aeróbico, la aireación es un factor importante y es suministrada por un compresor y distribuida por un agitador el cual es impulsado por un motor externo (Fitzpatrick *et al.*, 2020; Satitrueng & Soottitanta-wat 2019). Cada bioproceso es diferente dependiendo del microorganismo y sustrato empleados, presentando diversos desafíos que se deberán superar continuamente (Kowalska *et al.*, 2020). Para superarlos, la electrónica está presente en los procesos industriales, desarrollando sistemas electrónicos modulares y eficientes permitiendo su automatización (Nieto 2006), reduciendo considerablemente las tareas del operario (Ponsa & Arbós 2006), minimizando los errores humanos en la industria (Corona *et al.*, 2014). El microcontrolador es la pieza

clave para automatizar un proceso (López & Oñate 2015), es un circuito integrado con la capacidad de ser programable (Güven *et al.*, 2017). Actualmente, el microcontrolador ATmega328 por sus prestaciones que ofrece ha sido incorporado en las placas de Arduino (Arduino 2020). Arduino UNO, al ser una plataforma de libre acceso (Ortiz & Anguiano 2018), se convierte en una alternativa viable e interesante de bajo costo, para el desarrollo de aplicaciones enfocadas al monitoreo y control de un proceso (Saji *et al.*, 2017). Para que un sistema de control sea atractivo, dependerá del diseño de su interfaz (Moreno 2001), la interfaz es una herramienta interactiva que proporciona toda la información del sistema al usuario, permitiendo el análisis total de la información de forma clara y concisa (González 2019). Actualmente, se puede hacer uso de una pantalla táctil como interfaz (Holzinger 2002). La pantalla táctil Nextion, es una opción atractiva para ser usada para dicho propósito, gracias a las diversas prestaciones que ofrece (Nextion, 2020), permitiendo una gran facilidad en su programación (Benavides & Miranda 2019) como en la comunicación entre las placas de Arduino (Bento 2018).

En la operación de un fermentador, los desafíos están latentes y pueden presentarse en cierta etapa del bioproceso, pero pueden ser superados gracias a la incorporación de la electrónica. Actualmente, se encuentran en el mercado fermentadores destinados a diversas aplicaciones, por sus prestaciones y características que poseen, el biorreactor **Minifors 2**<sup>®</sup> contiene una pantalla táctil para poder calibrar los sensores y ajustes de configuración (Inforts-HT 2020), el biorreactor **Bio-Book**<sup>®</sup>, posee cuatro bombas peristálticas, con la opción de conectar varios sensores (Solida Biotech 2020), el biorreactor **F1**<sup>®</sup> tiene la capacidad de convertirse en un fotobiorreactor, proporcionando al usuario formas de visualizar, analizar y administrar los datos (Bionet<sup>®</sup> 2020), el fotobiorreactor **Algaemaster 10**<sup>®</sup>, ofrece un control computarizado de iluminación, temperatura, agitación, pH y dosificadores de líquido y gas (Profilab24 2020).

En resumen, hoy en día existen diversas compañías dedicadas a la venta de biorreactores; todos ellos presentan algo en común, estar acoplados a una interfaz gráfica para el monitoreo y control del bioproceso, cada interfaz incluye su propio software, convirtiéndose en un equipo sumamente costoso. Por lo anterior, se diseñó y desarrolló la interfaz **Fermenta V1.0** orientada al soporte de operaciones comúnmente usadas en un fermentador, usando software de acceso libre proyectada para poder adaptarse a un conjunto de dispositivos electrónicos de fácil adquisición.

## METODOLOGÍA

### MATERIALES

A continuación, se mencionan los materiales utilizados en el desarrollo del presente trabajo: Arduino UNO, módulo de motor L298N, pantalla Nextion de siete pulgadas (in), leds de color azul, leds de color rojo, led rojo-verde-azul (RGB), motor 12 volt (V), potenciómetros 100 kilohm (k $\Omega$ ), resistores. Para poder programar la placa Arduino UNO fue necesario descargar el Entorno de Desarrollo Integrado (IDE), del sitio oficial de Arduino (Arduino 2019), para el diseño de la interfaz se descargó el editor Nextion de la página oficial (Nextion 2020).

#### Diseño de las pantallas o secciones

Para el diseño de la interfaz, el primer paso fue determinar los requerimientos básicos de la operación de un biorreactor, en la figura 1, se representa el esquema integral de un fermentador, la información fue recopilada tomando en cuenta las características de diversos biorreactores comerciales, el siguiente paso fue la programación de la pantalla Nextion.

Para el correcto diseño de la interfaz fue necesario diseñar varias páginas o secciones, para lograrlo se utilizaron los objetos con los que cuenta el editor Nextion, tales como imagen, número, cuadro de texto, barra (slider) y botón, modificando sus atributos de tamaño, color y posición; posteriormente, se programaron las acciones a desempeñar al momento de tener un evento.

Se diseñaron un total de seis páginas o secciones que cubren las operaciones básicas de un biorreactor, **(a)** página principal, **(b)** sección de sensores, **(c)** página de agitación o mezclado, **(d)** sección de bombas peristálticas, **(e)** válvulas solenoides, **(f)** configuración.

Se hizo énfasis en incluir la imagen de un fermentador en la parte central de cada una de las secciones que conforman a la interfaz. Con la finalidad de poder acceder a todas las secciones se desarrolló un pequeño me-

nú en la parte inferior del fermentador, cada página provee al usuario información relevante acerca de cada uno de los dispositivos que integran al fermentador. A continuación, se comenta brevemente las páginas que integran a la interfaz **Fermenta V1.0**.

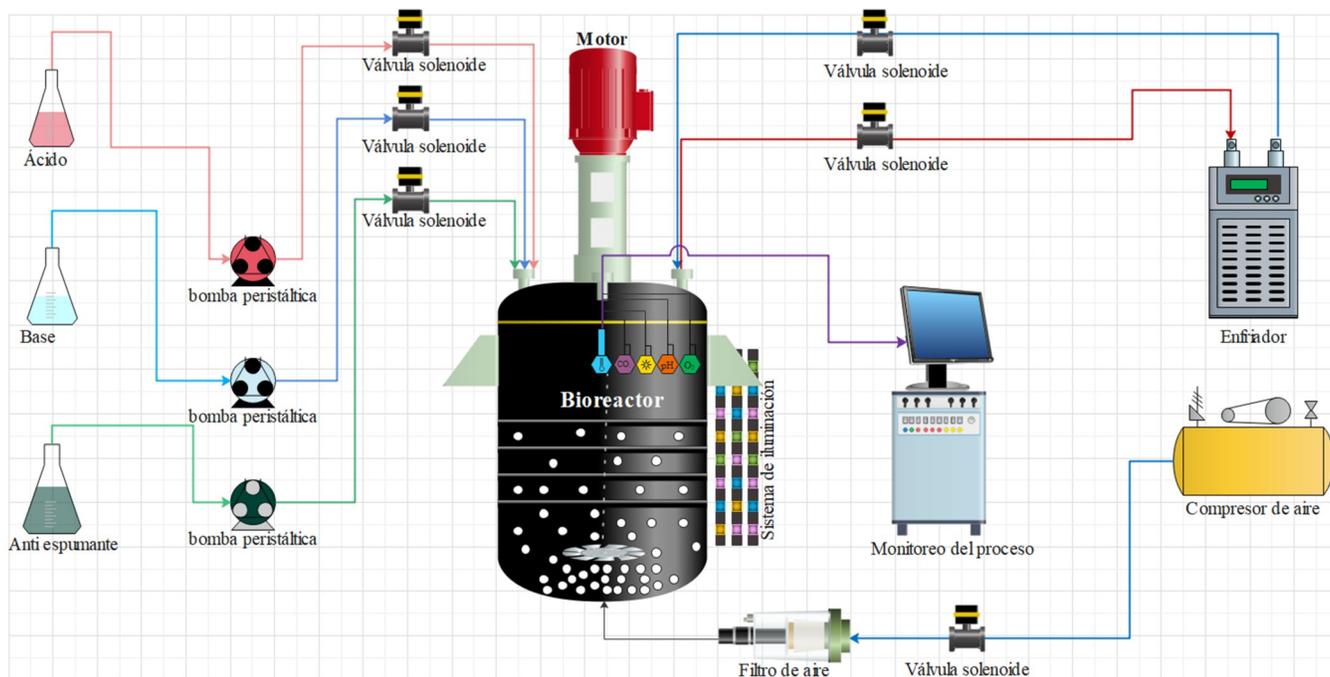
**Página principal.** En esta página se incorporó el esquema de un fermentador, a un lado de cada actuador se incluyeron pequeños círculos que representan leds, su principal función fue informar el estado del dispositivo encendido (verde) o apagado (rojo), en la parte inferior del fermentador, se colocó un menú para poder acceder a otras secciones de la interfaz.

**Página o sección de sensores.** Su diseño consistió en emular los sensores básicos usados en un fermentador como temperatura, pH, oxígeno disuelto, potencial óxido-reducción (ORP). Para su acceso, se construyó un menú con cuatro botones para cada sensor y fue localizado en la parte inferior de la figura de fermentador, se habilitaron cuatro ventanas para la visualización de los datos recolectados por cada sensor. Para evitar saturar la pantalla, se decidió habilitar la opción de ocultar/mostrar el valor de cada sensor. En la emulación de los sensores, fue necesario recurrir al uso de potenciómetros, recordando la fórmula de la ley de Ohm (1); en un circuito eléctrico, que fluye cierta intensidad de corriente (I), al momento de hacer girar la perilla del potenciómetro, la resistencia (R) cambiará de manera proporcional al voltaje (V). Por lo tanto, la variación de voltaje suministrado por el potenciómetro como un valor analógico hacia la placa Arduino UNO, podrá ser leído y traducido como una lectura de un sensor (emulación) y será mostrado en la pantalla Nextion.

$$V = R * I \quad (1)$$

**Página o sección de agitación o mezclado.** En esta sección se agregó un menú con tres botones, el primer botón mostró la velocidad del motor en revoluciones por minuto (rpm); el segundo botón, desplegó la información del impulsor que se está usando y el tercer botón, para regresar a la página anterior, al igual que las lecturas de los sensores, se habilitó una ventana para la visualización de la velocidad del motor.

**Página o sección de bombas peristálticas o dosificadoras.** Se incorporaron en la barra del menú cuatro botones, cada uno para cada bomba, al igual que en la sección de sensores, se emuló la activación de las bombas peristálticas con el uso de varios leds, se programó de tal manera que Arduino UNO mandará una señal de encendido hacia el led, con la finalidad de simular la



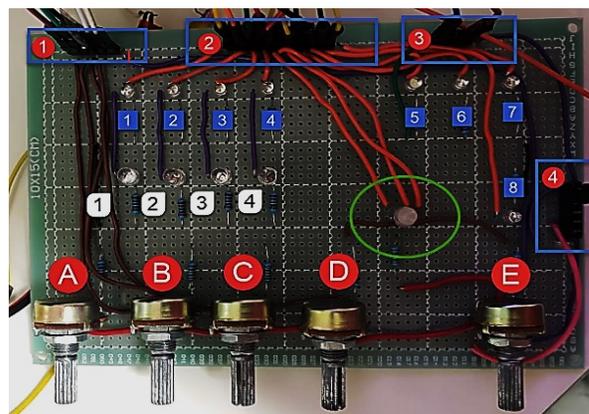
**Figura 1. Ilustración esquemática de un biorreactor.** En la parte central se muestra el biorreactor con diversos sensores en su interior, con un impulsor y accionado, equipado con un módulo de iluminación, del lado derecho el sistema de aireación y un enfriador, del lado izquierdo se muestran las bombas peristálticas y un módulo de monitoreo y control del proceso. (Fuente: Elaboración propia).

activación de la bomba peristáltica y caso contrario al desactivar la bomba peristáltica se debería de apagar el mismo led. Como ejemplo se hace mención de la bomba peristáltica encargada de dosificar un ácido, se debe de configurar un límite superior y un límite inferior de pH con la finalidad de activar/desactivar la bomba dosificadora (rango de trabajo o set-point), la variación de pH se realizó con el potenciómetro encargado de emular al sensor de pH. Una vez configurado el rango de trabajo de pH, al incrementar el valor de pH con el potenciómetro y sobrepasar el rango de trabajo, Arduino manda una señal y habilita a un led que emula a la bomba dosificadora de ácido, al volver modificar el valor de pH con el potenciómetro dentro del rango de trabajo, Arduino inhabilita el led, simulando la desactivación de la bomba dosificadora.

**Página o sección válvulas solenoides.** En esta sección se diseñó un menú con la posibilidad de controlar ocho válvulas solenoides, Al igual que las bombas peristálticas, para simular el encendido/apagado de las válvulas solenoides, se recurrió nuevamente del uso de los leds controlados por Arduino UNO.

**Página o sección configuración**

En esta sección, se incorporó un “set-point” o “rango de trabajo” con la finalidad de tener el control en el bioproceso, también se habilitaron dos módulos adicionales al



**Figura 2. Placa de control.** Con el uso de los potenciómetros se modificó el voltaje suministrado a las entradas analógicas de la placa Arduino UNO y la incorporación de Leds como instrumentos de aviso del sistema controlado por la interfaz Fermenta V1.0. (Fuente: Elaboración propia).

“set-point”, un módulo de agitación para controlar la velocidad del motor y el segundo módulo fue habilitado para controlar la iluminación del led RGB con el uso de tres barras como control de cada canal rojo, verde y azul del led.

Una vez que se terminó de diseñar las seis secciones de la interfaz, se procedió a cargar la información a la pantalla Nextion, el siguiente paso consistió en realizar la

comunicación con la placa Arduino UNO y posteriormente se desarrolló el código de programación para la placa Arduino UNO. En sitios oficiales se puede acceder a diversos tutoriales para la correcta programación de Arduino y de la pantalla Nextion.

## RESULTADOS Y DISCUSIÓN

### Armado de la maqueta de control

Para la emulación tanto de los sensores, bombas peristálticas y válvulas solenoides fue necesario la elaboración de una placa de control, la cual consistió en una placa perforada en donde se colocaron los potenciómetros que fungieron como los sensores y los leds que sustituyeron a las bombas y válvulas.

A continuación, se explica en detalle cada dispositivo incluido en la placa de control de la interfaz

### Fermenta V1.0.

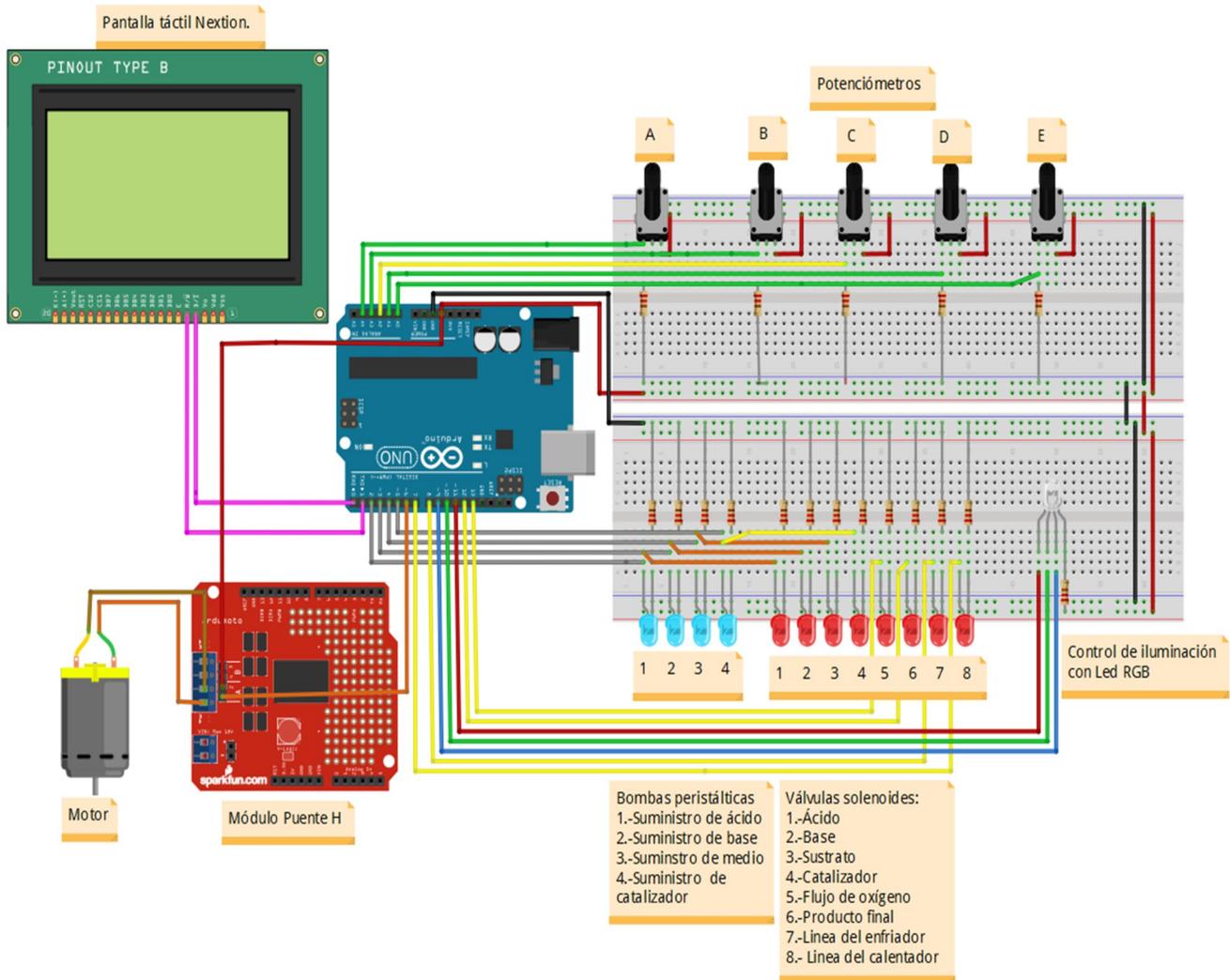
**A.** Potenciómetro que representó el sensor de pH. **B.** Potenciómetro que simbolizó el sensor de temperatura. **C.** Potenciómetro que emuló el sensor de concentración de oxígeno **D.** Potenciómetro que sustituyó al sensor de potencial óxido reducción (ORP) **E.** Potenciómetro que validó la velocidad del motor.

### Recuadros en color blanco

Representó a las cuatro bombas peristálticas, cada recuadro es acompañado por un led, si el elemento estaba activo el led se encenderá o en estado de reposo se apagará.

### Recuadros en color azul

Incluyó un total de ocho válvulas solenoides, similar a las bombas peristálticas, se incorporaron ocho leds, cada led representó a cada válvula del sistema, con la fina-



**Figura 3. Diagrama de conexiones.** Representación de las conexiones realizadas entre la placa Arduino UNO, pantalla Nextion, módulo de control del motor y los componentes de la placa de control. (Fuente: Elaboración propia).



**Figura 4. Pantalla principal.** Diseño general de la interfaz **Fermenta V1.0**, en la parte inferior se encuentra la barra de opciones, en la parte central el esquema de un fermentador, en la parte izquierda las bombas dosificadoras y en extremo derecho el sistema de control de temperatura, suministro de aire y recuperación de producto. (Fuente: Elaboración propia).

lidad de simular la activación de la válvula solenoide seleccionada en la interfaz **Fermenta V1.0**.

#### Elipse en color verde

En esa área está alojado un led RGB, el cual representa o emula al sistema de iluminación, dicho sistema de iluminación generó diversas tonalidades del espectro visible, al modificar en la pantalla Nextion las barras de control correspondientes a cada canal rojo, verde y azul.

#### Puertos

Los rectángulos en color azul representan a los puertos de la placa de control, su función es la siguiente: **1.** Puerto de entradas analógicas, en este caso reciben la información de potenciómetros. **2.** Puerto de salidas digitales, aquí nos encontramos con todos los indicadores led. **3.** Puerto 1 de alimentación de la placa. **4.** Puerto 2 de alimentación de la placa, en caso de requerir conectar algún elemento no considerado en el diseño.

#### Diagrama de conexiones

Las conexiones que se realizaron en la placa de control, se pueden consultar en la figura 3, en dicha figura se ejemplifica las partes que conforman a la interfaz **Fermenta V1.0**.

#### Apariencia de la interfaz

A continuación, se menciona la función de cada uno los botones que conforman la barra de opciones de la página principal.

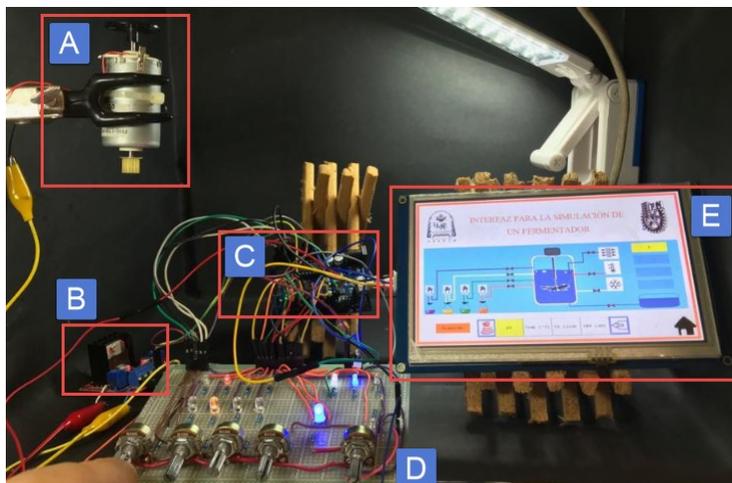
**A.** botón de sensores, se podrá consultar los sensores con los que cuenta el equipo.

**B.** botón de agitación: se podrá verificar el tipo de impulsor y la velocidad a la cual gira el motor. **C.** botón de bombas peristálticas, indica la función que tiene cada una de las bombas con las que cuenta el fermentador. **D.** botón válvulas solenoides, permite trasladarse a la sección del control de válvulas solenoides. **E.** acceso a la configuración, dentro de esta opción se puede modificar el set-point del proceso, activar las válvulas solenoides, o modificar el sistema de iluminación. **F.** botón de retorno, al presionarlo regresa a la página principal.

#### Prototipo terminado

Al terminar de armar la placa de control y conectar la placa de Arduino UNO, la pantalla Nextion y el motor, se procedió a probar la comunicación entre la pantalla Nextion y la placa de Arduino UNO, manipulando cada sección de la interfaz, accionando cada uno de los botones de la barra de opciones ubicada en la parte inferior de la figura del fermentador. Se activaron de manera exitosa los leds que simularon las bombas dosificadoras, del mismo modo se activaron los leds que representaron a las válvulas solenoides, se manipularon adecuadamente las diferentes tonalidades del led RGB, del mismo modo se logró controlar la velocidad del motor incorporado en el prototipo de la interfaz y por medio de los potenciómetros que emularon a los diversos sensores, se comprobó la correcta comunicación en todo el sistema que conforma la interfaz **Fermenta V1.0**.

En la figura 5 se muestra el prototipo terminado, para su explicación se delimitaron cinco áreas. **(A)** El motor, en la parte superior izquierda, simboliza al dispositivo que



**Figura 5. Funcionamiento de la Interfaz Fermenta V1.0.** Se muestra el prototipo completo, que incluye la pantalla Nextion, la placa Arduino UNO, la maqueta de control y el motor que ejemplifica la fuente de mezclado en un fermentador. Los recuadros representan diferentes partes que componen al prototipo. (Fuente: Elaboración propia).

mueve al agitador del fermentador. **(B)** Dispositivo electrónico que permite alimentar al motor. **(C)** El cerebro de la Interfaz, la placa Arduino UNO. **(D)** La maqueta de control. **(E)** pantalla Nextion en operación.

La gran ventaja que se obtiene al trabajar con Arduino UNO y otras placas de programación de libre acceso (Raspberry Pi, Banana Pi, BeagleBone, etc.), es la disponibilidad en el mercado de un gran número de sensores tanto de temperatura, pH, potencial oxido-reducción (ORP), CO<sub>2</sub>, metano, oxígeno disuelto, densidad óptica, iluminación, flujo volumétrico al igual de dispositivos electrónicos como bombas peristálticas, válvulas solenoides, relevadores, motores, pantallas a un precio accesible, todos esos dispositivos pueden ser incorporados a la placa para desarrollar sistemas robustos e integrales de monitoreo y control en un proceso determinado.

Se ha determinado que las interfaz hombre-maquina (HMI) cada vez adquieren una notable trascendencia en la automatización de proyectos en el área biológica (González 2019), el uso de pantallas táctiles Nextion, en base a la opinión de Benavides & Miranda en el 2019, consideraron que el editor Nextion proporcionó las herramientas necesarias para poder diseñar una interfaz capaz de visualizar y controlar las señales provenientes de los sensores utilizados, de igual manera como lo hizo notar Pérez en el 2017 con el uso de Arduino UNO fue posible conectar sensores de pH, temperatura y actuadores como bombas de llenado, válvulas y motores, también se ha destacado el uso del Entorno de Desarrollo

Integrado (IDE) de la plataforma Arduino en la programación de la placa Arduino UNO, por ser una plataforma de acceso libre y contar con una gran cantidad de librerías para realizar proyectos integrales y autónomos (Peñalver, Arroniz, Canales, Buendía, & Martínez 2014).

## CONCLUSIONES

Con el uso de lenguajes sencillos de programación y dispositivos electrónicos de libre acceso como Arduino UNO y pantallas Nextion, se logró diseñar la interfaz **Fermenta V1.0** orientada a la simulación en el monitoreo y control de un proceso básico de un fermentador.

El desempeño adecuado de la interfaz **Fermenta V1.0** fue comprobado a través de la navegación de las secciones que la conforman, fue capaz de adquirir y emitir señales tanto analógicas y digitales a través de Arduino UNO como unidad de control con la manipulación de los potenciómetros.

Impulsar el uso de las placas de libre acceso en la enseñanza de futuros programadores, encaminará a explorar, aprovechar y desarrollar aplicaciones útiles en la agricultura, biotecnología, ciencias naturales, ciencias exactas y de la salud.

## AGRADECIMIENTOS

A la Universidad del Mar campus Puerto Escondido por las facilidades otorgadas para realizar el presente trabajo.

## CONFLICTO DE INTERESES

Los autores declaran que no existen conflictos de intereses.

## BIBLIOGRAFÍA

- Aizawa P., Karlsson G., Benemar C., Segl V., Martinelle K., Lindner E. (2007). Effect of Different Materials used in Bioreactor Equipments on Cell Growth of Human Embryonic Kidney (HEK293) Cells Cultivated in a Protein-free Medium. In: Smith R. (eds) *Cell Technology for Cell Products*. Springer, Dordrecht. [https://doi.org/10.1007/978-1-4020-5476-1\\_77](https://doi.org/10.1007/978-1-4020-5476-1_77)
- Nieto E. (2006). Manufactura y automatización. *Ingeniería e Investigación*. 26(3): 120-128. <https://www.redalyc.org/pdf/643/64326315.pdf>.
- Arduino. (2019). Arduino Software (IDE). Recuperado el 15 de noviembre 2019 de página principal de Arduino: <https://www.arduino.cc/en/Main/Software>.
- Arduino. (2020). Arduino UNO Rev 3. Recuperado el 2 de enero 2020 de página principal de Arduino: <https://store.arduino.cc/usa/arduino-uno-rev3>.
- Benavides J., & Miranda F. (2019). Diseño e implementación de módulos de entrenamiento con asistencia de pantalla digital para sistemas de iluminación automatiz. Tesis de Licenciatura. Escuela Superior Politécnica de Chimborazo. <http://dspace.esPOCH.edu.ec/handle/123456789/11507>.
- Bento A. (2018). IoT of Nextion X TFT ILI9341: Experimental Results and Comparative Survey. *International research journal of engineering, IT & scientific research*. 4(2): 14-23. <https://doi.org/10.21744/irjeis.v4n2.52>.
- Bionet (2020). F1 Laboratory Bioreactor. Recuperado el 12 de marzo 2020 de página principal de Bionet: <https://bionet.com/technology/f1-bioreactor/>
- Corona, L; Abarca, G; Mares, J. (2014). *Sensores y actuadores: aplicaciones con Arduino*. México: Grupo Editorial Patria.
- Fitzpatrick J., Gloanec F., & Michel E. (2020). Insights from Mathematical Modelling into Process Control of Oxygen Transfer in Batch Stirred Tank Bioreactors for Reducing Energy Requirement. *ChemEngineering*, 4 (2): 1-15. <https://doi.org/10.3390/chemengineering4020034>.
- González Becerra, X. M. (2019). Propuesta de una interfaz humano-máquina (HMI) de control para un robot cartesiano de coordenadas globales usado para el cultivo de hortalizas. Tesis de grado. Universidad Piloto de Colombia. <http://repository.unipiloto.edu.co/handle/20.500.12277/6497>.
- Güven Y., Coşgun E., Kocaoğlu S., Ezici H., & Yilmazlar E. (2017). Understanding the concept of microcontroller based systems to choose the best hardware for applications. *Research Inventy: International Journal of Engineering and Science*. 6(9): 38-44.
- Holzinger A. (2002). Finger instead of mouse: touch screens as a means of enhancing universal access. In *ERCIM Workshop on User Interfaces for all* (387-397). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-36572-9\\_30](https://doi.org/10.1007/3-540-36572-9_30).
- Infors-HT (2020). Infors-HT. Recuperado el 12 de marzo 2020 de página principal de Infors-HT: <https://www.infors-ht.com/en/bioreactors/bench-top-bioreactors/minifors2>.
- Kafarski P. (2012). Rainbow code of biotechnology. *Chemik*. 66(8): 811-816.
- Kowalska A., Boruta T., & Bizukojć M. (2020). Performance of fungal microparticle-enhanced cultivations in stirred tank bioreactors depends on species and number of process stages. *Biochemical Engineering Journal*. 161(10): 1-15. <https://doi.org/10.1016/j.bej.2020.107696>.
- Kracík T., Petříček R., & Moucha T. (2020). Mass transfer in coalescent batch fermenters with mechanically agitation. *Chemical Engineering Research and Design*. 160: 587-592. <https://doi.org/10.1016/j.cherd.2020.03.015>.
- Liu, S. (2020). *Bioprocess engineering: kinetics, sustainability, and reactor design*. Elsevier.
- López R., & Oñate E. (2015). *Implementación de un sistema automático de ventilación e iluminación controlado por un microcontrolador para el laboratorio de mantenimiento eléctrico*. Tesis de grado. Universidad Técnica del Norte. <http://repositorio.utn.edu.ec/handle/123456789/4461>.
- Matyushenko I., Sviatukha I., & Grigorova L. (2016). Modern Approaches to Classification of Biotechnology as a Part of NBIC-Technologies for Bioeconomy. *Journal of Economics, Management and Trade*. 14(4): 1-14 <https://doi.org/10.9734/bjemt/2016/28151>.
- Mosier N., & Ladisch M. (2011). *Modern biotechnology: connecting innovations in microbiology and biochemistry to engineering fundamentals*. John Wiley & Sons.
- Moreno A. (2001). La consistencia en la interfaz de usuario. *El profesional de la información*. 10(12): 59
- Ortiz L., & Anguiano J. (2018). Automatización de simulador de lluvia mediante microcontrolador Arduino. *Jóvenes en la ciencia*. 4(1): 2757-2763. <http://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/2735>.
- Peñalver L., Arroniz D., Canales A., Buendía M., & Martínez J. (2014). Sistema de automatización y telecontrol mediante microcontroladores para la gestión del riego de cultivos en maceta con lisímetros de pesa-

da. In VII Congreso Ibérico de Agro-ingeniería y Ciencias Hortícolas: Innovar y producir para el futuro: Libro de Actas (216-221). Fundación General de la Universidad Politécnica de Madrid. <https://dialnet.unirioja.es/servlet/articulo?codigo=7483536>.

Pérez A. (2017). Control y supervisión mediante un sistema microcontrolador de los parámetros de calidad de agua de un estanque. Tesis de grado. Universidad de Sevilla. <https://idus.us.es/handle/11441/61095>.

Ponsa, P; Arbós, R. (2006). *Automatización de procesos mediante la guía GEMMA*. Barcelona. Universitat Politècnica de Catalunya. Iniciativa Digital Politécnica.

Nextion (2019). Nextion Intelligent HMI Display. Recuperado el 12 de junio 2019 de página principal de Nextion. <https://nextion.tech/>.

Profilab24. (2020). ProfiLab24 Professional equipment. Recuperado el 12 de marzo 2020 de página principal de ProfiLab24. <https://profilab24.com/en/laboratory/analytical-devices/ika-photo-bioreactor-algaemaster-10-control>.

Raju P. (2016). World history of modern biotechnology and its applications. *Biotechnology: an Indian journal*.

12(11): 107-115. <https://www.tsijournals.com/articles/world-history-of-modern-biotechnology-and-its-applications.pdf>.

Satitrueng K., & Soottitantawat A. (2019). Influence of agitator speed and aeration rate on the oxygen mass transfer rate coefficient in the stirred tank bioreactor. *Journal of Functional Materials and Chemical Engineering*. 1(1): 83-88.

Solida biotech (2020). Solida Biotech GmbH. Recuperado el 12 de marzo 2020 de página principal de Solida Biotech. <https://solidabiotech.com/product/compact-bioreactor-2/>.

Saji, J., Dubey, R., & Saravanakumar, K. (2017). Food Computer Automated Gardening System. *International Journal of Trend in Research and Development*. 38-39.

Wang S., & Zhong J. (2007). Bioreactor Engineering. In: *Bioprocessing for Value-Added Products from Renewable Resources*, Elsevier 131-161. <https://doi:10.1016/b978-044452114-9/50007-4>.

## ANEXO CÓDIGO DE PROGRAMACIÓN

```
#include "Nextion.h" // Se incluye la librería para facilitar la programación de la pantalla Nextion
```

```
//-----Asignación del número de puerto a utilizar-----
#define motor 6
#define linea1 11
#define linea2 10
#define linea3 9
const int lac =2;
const int lba =3;
const int lnut=4;
const int lcat=5;
const int lcali=7;
const int lfrio=8;
const int laire=13;
const int lfin=12;
//-----Variables utilizadas-----
int x = 0, y = 0, z = 0, w = 0, v=0;
int ph = 0, temp = 0, oxigeno=0, par_ortp=0, velmot=0;
int sp = A0, st = A1, rev = A2, oxi = A3, ortp = A4;
int Pulso1=0,Pulso2=0,Pulso3=0,Pulsom;
uint32_t va0=0,va1=0,va2=0,va3=0,va4=0,va5=0,va6=0;
//-----
//Se declaran los objetos a utilizar con el siguiente formato:
// [página id:0,componente id:1, NombreComponente: "Objeto_ObjName"]-----
```

```
NexDSButton bnut = NexDSButton(7, 42, "bnut"); // DS Button added
NexDSButton bcata = NexDSButton(7, 43, "bcata");
NexDSButton bprod = NexDSButton(7, 44, "bprod");
NexNumber npH = NexNumber(7, 12, "npH");
NexNumber ntemp = NexNumber(7, 15, "ntemp");
NexNumber noxii = NexNumber(7, 16, "noxii");
NexNumber nmot = NexNumber(7, 24, "nmot");
NexNumber nc1 = NexNumber(7, 40, "nc1");
NexNumber nc2 = NexNumber(7, 41, "nc2");
NexNumber nc3 = NexNumber(7, 42, "nc3");
//-----
//-----Se declaran los objetos a los cuales estaremos a la espera de que se presente un evento. Es decir, componentes que transmitirán información de la pantalla al Arduino.-----
NexTouch *nex_listen_list[] =
{
    &bnut,
    &bcata,
    &bprod,
    &nmot,
    &nc1,
    &nc2,
    &nc3,
    NULL
};
//-----
//Código que se corre solo una vez, aquí configuramos puertos y comenzamos comunicaciones-----
```

```

void setup(void)
{
  nexInit(); //Inicio la comunicación entre el Arduino y
la Pantalla
  pinMode(lac, OUTPUT); //declaramos los puertos
que se utilizarán como salidas
  pinMode(lba, OUTPUT);
  pinMode(lnut, OUTPUT);
  pinMode(lcat, OUTPUT);
  pinMode(lcali, OUTPUT);
  pinMode(lfrio, OUTPUT);
  pinMode(laire, OUTPUT);
  pinMode(lfin, OUTPUT);
  digitalWrite(lac, LOW); //inicializamos las salidas a
cero.
  digitalWrite(lba, LOW);
  digitalWrite(lnut, LOW);
  digitalWrite(lcat, LOW);
  digitalWrite(lcali, LOW);
  digitalWrite(lfrio, LOW);
  digitalWrite(laire, LOW);
  digitalWrite(lfin, LOW);
}
//-----
//código que se corre de manera repetida, aquí van las
acciones del programa-----
void loop(void)
{
  nexLoop(nex_listen_list); //direcciona a una rutina si
se presenta algún evento en un objeto.
  x = analogRead(sp); //realizamos una lectura analógi-
ca, dado que el ADC del Arduino es de 8 bits
  y=analogRead(st); //tendremos un valor comprendi-
do entre 0-1023
  v=analogRead(rev);
  w=analogRead(oxi);
  z=analogRead(ortp);

  //se realiza un mapeo del código, esto quiere decir
que hay una correlación lineal convirtiendo de 0-1023
a //otro rango de valores especificado.

  ph=map(x, 0, 1023, 0, 14);
  temp=map(y, 0, 1023, 0, 100);
  ...oxigeno=map(w,0,1023,0,10);
  par_ortp=map(z,0,1023,0,100);
  velmot=map(v,0,1023,0,600);
//Obtenemos el valor que tiene el cuadro de texto den-
tro de la pantalla para poder manipularlo.
  nph.getValue(&va0);
  ntemp.getValue(&va1);
  noxii.getValue(&va2);
  nc1.getValue(&va3);
  nc2.getValue(&va4);

```

```

nc3.getValue(&va5);
nmot.getValue(&va6);
//---CONTROL DE LEDS DE pH-----
  if(va0>ph)
  {
    digitalWrite(lba, HIGH); //mediante un condi-
cional simple decidimos que
    digitalWrite(lac, LOW); //led debe estar preñdi-
do
  }
  else if(va0<ph)
  {
    digitalWrite(lac, HIGH);
    digitalWrite(lba, LOW);
  }
  else
  {
    digitalWrite(lac, LOW);
    digitalWrite(lba, LOW);
  }
}
//---CONTROL DE LEDS DE TEMPERATURA-----
if(va1>temp)
{
  digitalWrite(lcali, HIGH); //mediante un condi-
cional simple decidimos que
  digitalWrite(lfrio, LOW); //led debe estar preñdi-
do
}
else if(va1<temp)
{
  digitalWrite(lfrio, HIGH);
  digitalWrite(lcali, LOW);
}
else
{
  digitalWrite(lcali, LOW);
  digitalWrite(lfrio, LOW);
}
}
//---CONTROL DE OXÍGENO-----
if(va2>oxigeno)
{
  //mediante un condi-
cional simple decidimos que
  digitalWrite(laire, HIGH); //led debe estar preñdi-
do
}
else
{
  digitalWrite(laire, LOW);
}
}
//-----
// Control del LED RGB-----
Pulso1=map(va3,0,100,255,0); //realizamos un
mapeo del valor que tiene el slider dentro

```

```

    analogWrite(linea1,Pulso1); //de la pantalla y lo
escalamos de 0-255
    delay(100); //que posteriormente lo
escribiremos para tener un PWM
//que controle la intensidad
luminosa en cada canal
    Pulso2=map(va4,0,100,255,0);
    analogWrite(linea2,Pulso2);
    delay(100);
    Pulso3=map(va5,0,100,255,0);
    analogWrite(linea3,Pulso3);
    delay(100);
//---CONTROL MOTOR_-----
// Se realiza una lectura del valor que tiene el slider o
barra para asignarlo al motor.
    Pulsom=map(va6,0,500,40,255);
    analogWrite(motor,Pulsom);
    delay(100);
//-----
//-----ACTIVACIÓN DE MANDOS (Control IN/OFF)
    bnut.getValue(&va4); //
obtenemos el valor del botón dual
    if(va4==1){digitalWrite(lnut, HIGH);} //y por
medio de un condicional decidimos
    else{digitalWrite(lnut, LOW); } //si debe
estar prendido o apagado
    bcata.getValue(&va5);
    if(va5==1){digitalWrite(lcat, HIGH);}
    else{digitalWrite(lcat, LOW); }
    bprod.getValue(&va6);
    if(va6==1){digitalWrite(lfin, HIGH);}
    else{digitalWrite(lfin, LOW); }

```

```

//---Lectura arrojada por los sensores-----
Serial.print("phlec.val="); // En esta parte escribimos el
nombre del objeto al cual queremos escribirle algo
Serial.print(ph); // Para ello debemos escribir el
nombre del objeto y el atributo al cual queremos
Serial.write(0xff); // modificar. Después escribi-
mos el nombre de la variable de la cual se va a obtener
Serial.write(0xff); //la información. Finalmente
mandamos 3 escrituras seriales para que la pantalla
Serial.write(0xff); //identifique la lectura.
Serial.print("templec.val=");
Serial.print(temp);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("oxilec.val=");
Serial.print(oxigeno);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("orplec.val=");
Serial.print(par_ortp);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
Serial.print("vellec.val=");
Serial.print(velmot);
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
}

```