

---

**Trabajo de curso**  
**2020/21**  
**HUERTOMATIC 3000**

---

**Alejandro García Sosa**

*(alejandro.garcia134@alu.ulpgc.es)*

**Alejandro Daniel Herrera Cárdenes**

*(alejandro.herrera113@alu.ulpgc.es)*

**Carlos Eduardo Pacichana Bastidas**

*(carlos.pacichana101@alu.ulpgc.es)*

**Javier Santana Álamo**

*(javier.santana130@alu.ulpgc.es)*

Universidad de Las Palmas de Gran Canaria

Escuela de Ingeniería en Informática

## Contenido

<b>o Presentación del proyecto</b>	1
<b>o Tecnologías utilizadas y descripción técnica</b>	2
<b>o Diario de actividades</b>	6
<b>o Tareas realizadas y porcentaje de participación de cada integrante</b>	8
<b>o Enlace al código</b>	8
<b>o Conclusiones y propuesta de ampliación</b>	9
<b>o Herramientas /tecnologías con las que les hubiera gustado contar</b>	10
<b>o Anexo</b>	11
<i>Tinkerkad</i>	11
<i>Processing</i>	13
Simulación	15
<i>Resumen breve</i>	17
<b>o Créditos materiales no originales del grupo</b>	17

## o Presentación del proyecto

En el presente informe el doble grupo 3-5 (conformado por Alejandro García Sosa, Alejandro Daniel Herrera Cárdenes, Carlos Eduardo Pacichana Bastidas y Javier Santana Álamo) presentamos el proyecto que hemos realizado como trabajo final para la asignatura Creando Interfaces de Usuario (CIU). El proyecto en cuestión es un sistema para la gestión del huerto de un invernadero y lo hemos bautizado como Huertomatic 3000 a fin de darle un nombre que resalte aprovechándonos de la cultura de la teletienda (Algo que se verá reflejado en el vídeo *teaser* que hemos llevado a cabo).

En el ámbito que hemos querido centrar nuestro proyecto es en el hecho de que cada vez es más habitual que las personas instalen huertos para consumo privado en sus hogares. En algunos casos estos se instalan en lugares de acceso menos habitual como azoteas, trasteros o sótanos los cuales no solemos ver de manera regular y este proyecto trata sobre una solución que nos permita llevar un control de este desde nuestro terminal e incluso que en caso de que sea necesario el sistema pueda realizar ajuste en el entorno mediante soportes de riego, calefacción y luz de manera automática.

La motivación de la propuesta viene principalmente dada por el hecho de utilizar los sistemas de sensores en un entorno real desarrollando un aplicación práctica que haga uso de estos solucionando un problema actual, aunque a partir de aquí es donde surgieron limitaciones con respecto a los elementos disponibles, pues los que podíamos obtener eran infinitamente funciones con respecto a los que deseábamos, optando finalmente por realizar todo el apartado hardware con la herramienta *Tinkercad* (aunque como comentaremos a continuación no ha sido todo tan bonito como planteamos inicialmente), para el entorno gráfico decidimos seguir contando con *Processing*, herramienta que hemos estado perfeccionando durante todo el curso por lo cual hemos desarrollado un conocimiento muy útil para este proyecto final.

Como hemos indicado anteriormente, el trabajo con *Tinkercad* no ha sido un camino de rosas ni mucho menos y aunque nos ha eliminado algunas restricciones, nos hemos visto impuestas otras que según se mire han sido mucho mayores, además de contar que hemos tenido que tirar de ingenio (como se verá cuando expliquemos las tecnologías empleadas) para poder llevar a cabo todo lo que deseábamos. La principal restricción que hemos sufrido al emplear un sistema simulado empleado *Tinkercad* es la imposibilidad de conectar con *Processing* como sí pudimos hacer en la realización de la práctica 12. Este contratiempo nos ha llevado a dejar la parte de Arduino en Processing comentada expectante para cuando se disponga de los elementos deseados, que los comentaremos a continuación.

A lo largo del desarrollo descubrimos ciertas utilidades de nuestro proyecto que lo hicieron más apasionantes, como por ejemplo su utilidad educativa, lo que nos llevó a plantear diversas ideas con el fin de que resultara más vistoso para niños y niñas de cursos de infantil y fuera sencillo de emplear con un simple vistazo (En este caso hablamos principalmente de cuando se trabaja sobre *Processing* y no tanto sobre el sistema *Hardware*).

## o Tecnologías utilizadas y descripción técnica

Como ya detallamos anteriormente las tecnologías empleadas son *Processing* y *Tinkercad* (aunque esta es más bien un sustituto de toda la parte hardware no disponible, mismamente podríamos decir que es un Arduino con una variedad de sensores conectados a su alrededor), describiendo a continuación el trabajo realizado con ambas tecnologías.

Comenzando por el apartado de *Tinkercad* como ya hemos indicado la idea es simular un huerto o invernadero automático a través de los datos obtenidos de esos sensores pudiendo activar automáticamente la ventilación, riego automático o luz si es necesario.

Además, se ha añadido un sistema antiincendios que activa automáticamente el sistema de riego para minorizar los riesgos y se activa un sonido para que los usuarios tengan conocimiento del incendio en caso de que no estén observando el huerto.

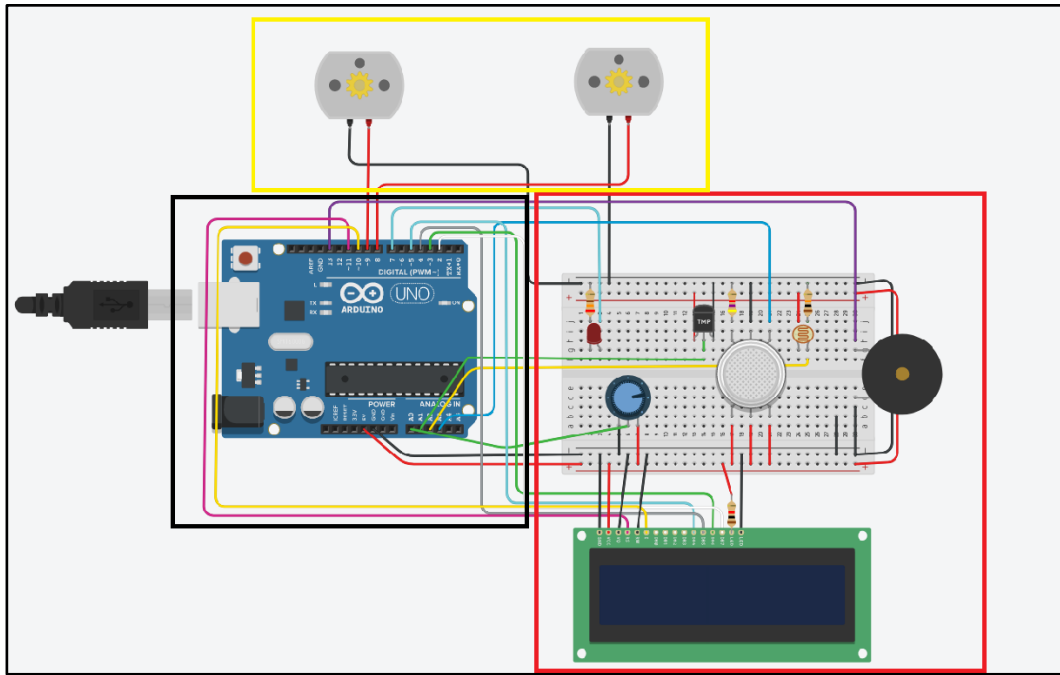
Y se ha añadido una pantalla LCD donde se pueda observar los datos de temperatura y humedad a tiempo real y en caso de que salte la alarma antiincendios salga un mensaje con el número para llamar a los bomberos.

Los elementos empleados son:

- Por el apartado de sensores:
  - De temperatura
  - De humedad
  - De fotorresistencia (LDR)
  - De gas
- Mientras que por el resto de los componentes
  - Arduino uno
  - Placa de pruebas
  - Resistencias
  - Cables
  - 2 motores CC para simular regadío y ventilación
  - Potenciómetro
  - Piezo
  - Led
  - Pantalla LCD 16 x 2

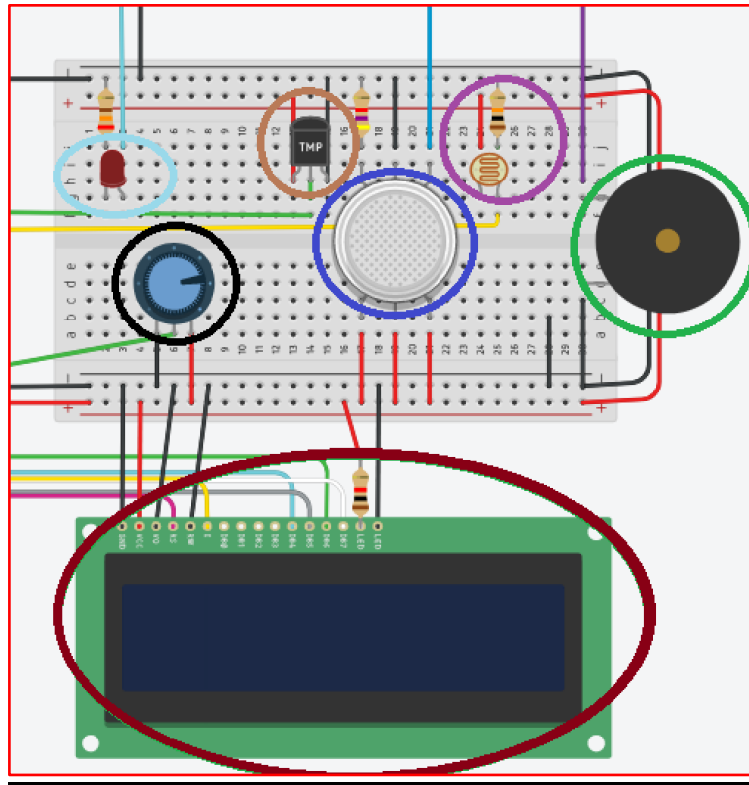
Dentro de los sensores, el de gas cumple una función de un detector de humos, para controlar posibles incendios. También tenemos que el potenciómetro para simular cambios en la humedad (Lo cual nos da más margen de maniobra que usando sensores reales).

A continuación, describiremos por zonas el proyecto en Arduino el cual podemos ver completamente en la imagen inferior (**imagen\_1.0**).



*imagen\_1.0*

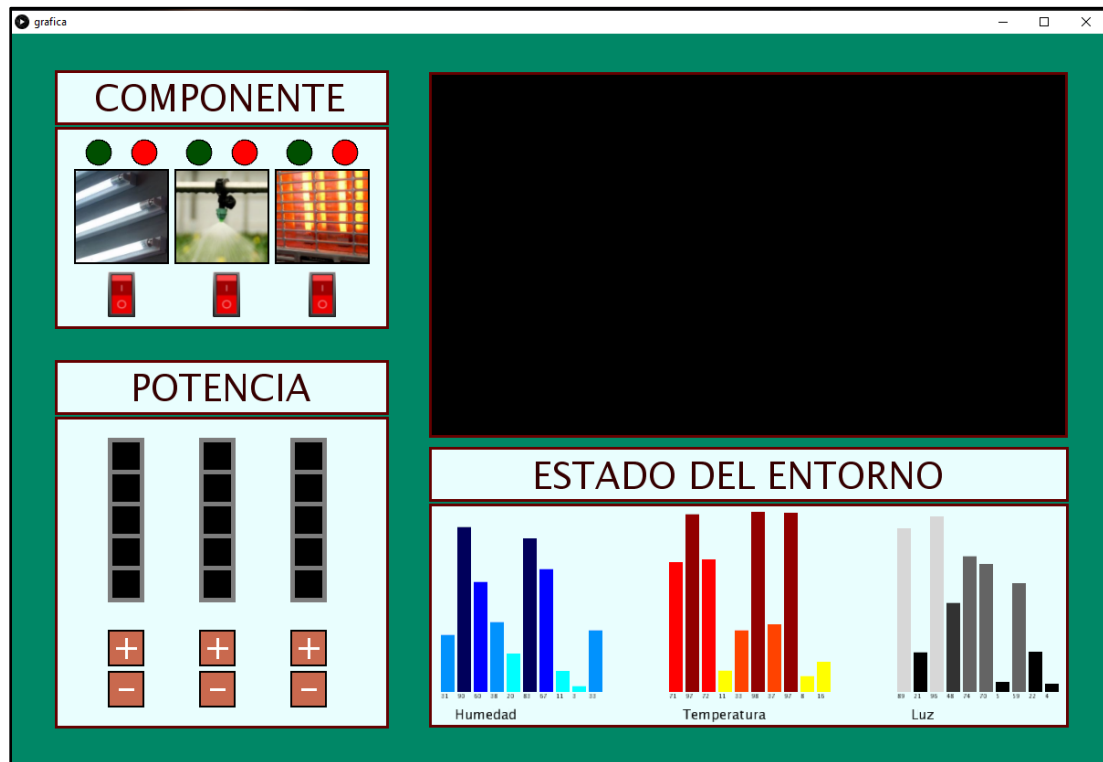
- Para empezar, tenemos el **rectángulo negro** donde básicamente está el Arduino con los pines de entrada y salida donde irán conectados los sensores para que nos muestre la información deseada y de energía a la placa de pruebas para que estos componentes funcionen.
- También tenemos el **rectángulo amarillo** que simula los motores de riego y ventilación con dos motores de corriente continua.
- Finalmente tenemos el **rectángulo rojo** que es la mesa de pruebas donde están situados todos los sensores que usaremos para la simulación junto con la pantalla LCD donde se mostrarán los datos y el piezo donde se generará el sonido si es necesario. Veremos más en profundidad la mesa de pruebas en la imagen a continuación (**imagen\_1.1**).



*imagen\_1.1*

- Empezando de izquierda a derecha tenemos el **círculo azul claro** donde está conectado el **led** que conectará la luz o no según el sensor de fotorresistencia detecte cuan cantidad de luz haya.
- El **círculo negro** al *Tinkercad* no poseer detector de humedad usamos **un potenciómetro** que usaremos de manera manual para realizar las pruebas según queramos que haya más o menos humedad.
- El **círculo marrón** es el **sensor de temperatura** al que también aumentaremos de manera manual a más o menos temperatura para realizar las pruebas.
- Luego tenemos el **círculo azul** que es el **sensor de gas**, pero en nuestro caso lo usaremos como **sensor de humo** el cual detectará si hay humo cerca del sensor activando una alarma que sería el **piezo** (**círculo verde**).
- El **círculo morado** sería el **sensor de fotorresistencia** que nos indica la cantidad de luz que detecta. También la cambiaremos de manera manual para realizar las pruebas.
- Y para acabar el **círculo rojo** es la **pantalla LCD** donde se mostrarán los datos de humedad y temperatura y en caso de que haya un incendio mostrará el número de teléfono de los bomberos para rápida actuación del usuario.

El entorno gráfico, cuyo desarrollo se detalla en el siguiente punto, podemos observarlo como un adelanto a continuación, donde aprovecharemos para explicar el trabajo realizado un poco más en profundidad lejos de las decisiones de diseño (*imagen\_1.2*).



*imagen\_1.2*

En un primer vistazo podemos observar cuatro bloques distintos de trabajo y los describiremos a continuación:

- El primer bloque (**COMPONENTE**), situado arriba a la izquierda, es para la activación **manual** de elementos lumínicos, de riego y de temperatura.
- El segundo bloque, arriba a la derecha es la cámara, donde controlaremos de manera visual lo que ocurre en nuestro huerto, permitiéndonos esto saber el estado exacto del mismo sin necesidad de desplazarnos hasta el mismo.
- Justo debajo encontramos el bloque del **ESTADO DEL ENTORNO**, donde tenemos una serie de gráficas que muestran el desarrollo durante el tiempo de las condiciones de humedad, temperatura y luz.
- Finalmente tenemos el bloque de abajo a la izquierda, marcado como **POTENCIA**, este bloque se usa para controlar los niveles de intensidad en el riego, la luminosidad y la temperatura. La potencia de riego se sitúa al máximo en caso de incendio, en caso contrario suele estar en un goteo ligero o incluso apagado. (En el proyecto actual no se encuentra incorporado debido a no poseer elementos de riego, luz o temperatura).

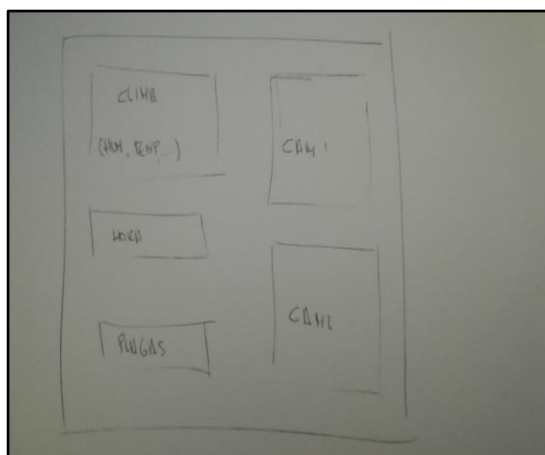
## o Diario de actividades

A continuación, relataremos de una manera breve el recorrido que ha seguido el grupo persiguiendo el objetivo de realizar nuestro proyecto.

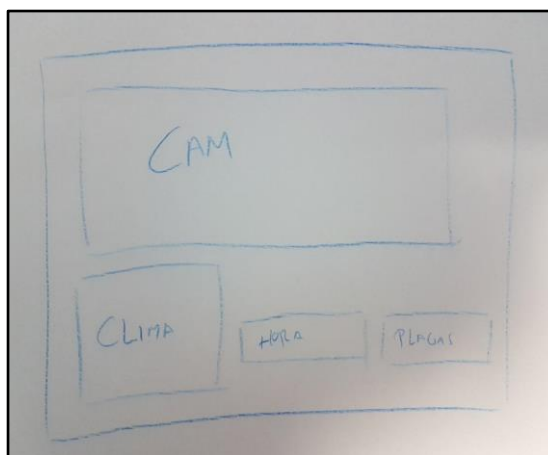
Nuestro grupo, al ser una fusión de los grupos 3 y 5 comenzó a trabajar en el proyecto con un poco de retraso, dando como resultado para el plazo inicial de la asignatura, el 18 de mayo, simplemente la aparición de dos ideas, la que nos encontramos contando a lo largo de este informe y una pantalla táctil, idea desechada tras la presentación del grupo durante dicho día.

La siguiente semana a esa fue la de menor desarrollo de la aplicación, pues nos encontrábamos durante la semana final del curso con las consiguientes entregas, dando sólo la aparición de un esquema inicial del hardware en Tinkercad y de un boceto del apartado gráfico deseado de la aplicación en *Processing*. A continuación, mostraremos distintos bocetos que fuimos realizando durante la discusión para su forma definitiva.

Como podemos percibir comenzamos por un modelo de dos cámaras que fue rápidamente desechada por reducir recursos empleados (**imagen\_2.0**), posteriormente se planteó un modelo con la cámara cubriendo la mitad de la interfaz (**imagen\_2.1**), pero no nos gustaba que esta tomara tanto protagonista, por lo que un boceto definitivo nació posteriormente (aunque fue modificado a posteriori en la fase de diseño sobre Processing **imagen\_2.2** e **imagen\_2.3**), el modelo definitivo se escogió ya que permitía un mayor control de manera rápida de los datos de luz, humedad y temperatura, siendo algo sencillo de entender para alguien adulto o de menor edad, cumpliendo el objetivo de ser útil en la docencia (**imagen\_2.4**).

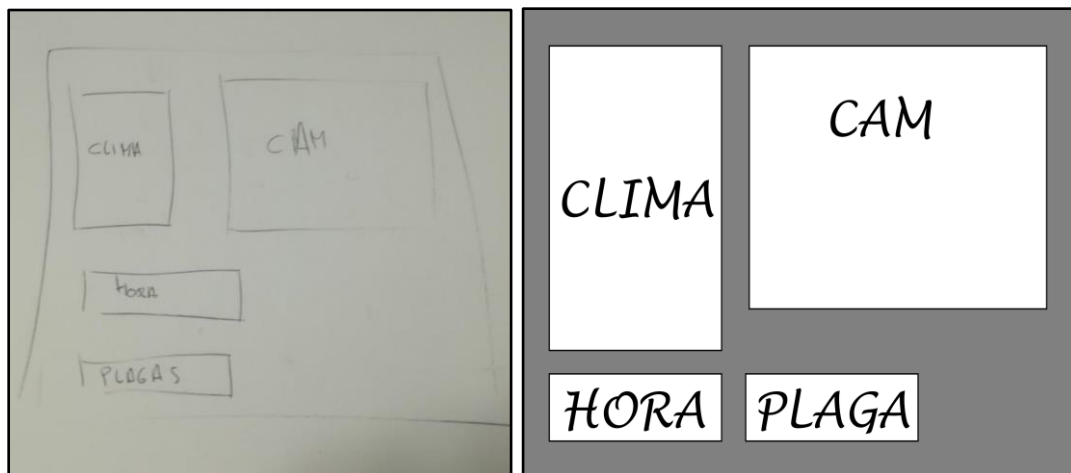


*imagen\_2.0*

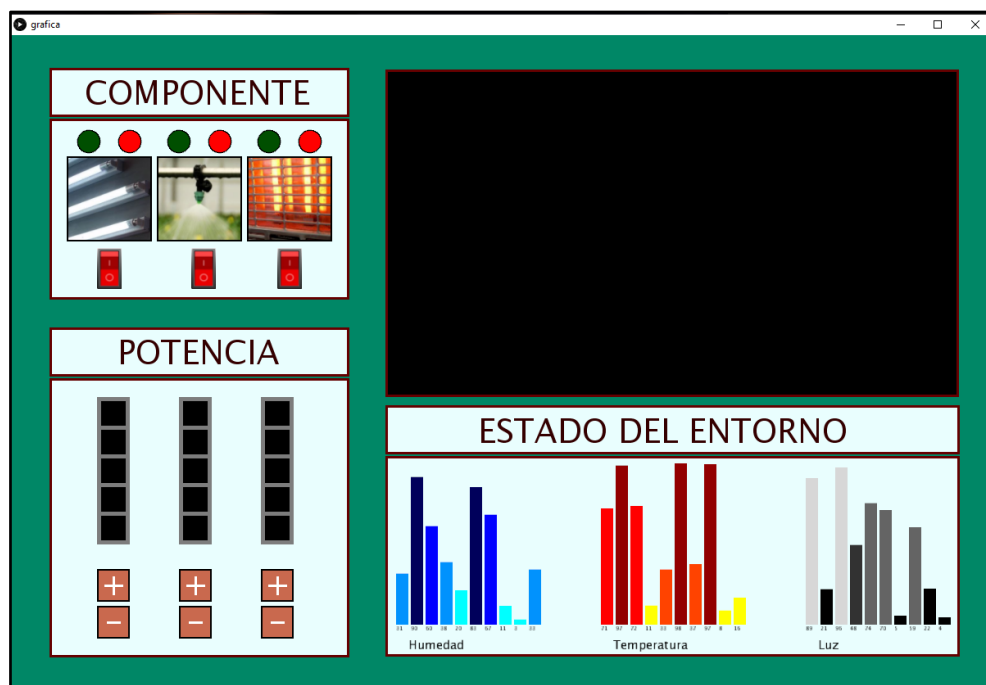


*imagen\_2.1*





*imagen\_2.2*



*imagen\_2.3*

Continuando con el desarrollo, la siguiente semana se centró en la finalización del esquema en *Tinkercad* como en la realización del entorno gráfico en Processing, inicialmente sin funcionalidad de cámara y en una segunda instancia ya añadiéndola.

Finalmente, lo último que se realizó fue el vídeo *teaser*, siendo la idea de aprovechar la cultura de la teletienda la que hizo brotar la idea del nombre, *Huertomatic 3000*. También se preparó en una reunión grupal la manera más adecuada de presentar nuestro producto, repartiendo la presentación y tratando de hacerla lo más dinámica posible y que se conociera a cada integrante del grupo.

## o Tareas realizadas y porcentaje de participación de cada integrante

En la siguiente tabla mostraremos las distintas tareas llevadas a cabo durante la realización de este proyecto indicando quienes participaron en ella

	Alejandro García	Alejandro Herrera	Javier Santana	Carlos Pacichana
Diseño de la interfaz gráfica	X	X	<b>X</b>	X
Diseño del circuito Arduino		X		
Implementación en Processing3	X		<b>X</b>	X
Implementación Arduino		X		
Grabación y edición sketch	X		X	
Redacción de memoria	X	X	<b>X</b>	X
Redacción presentación	X	X	<b>X</b>	X

## o Enlace al código

[\*Tinkercad y Arduino\*](#)

[\*Processing / GIT\*](#)

## o Conclusiones y propuesta de ampliación

Si bien cuando planteamos el proyecto de manera inicial nos pareció algo sencillo de realizar, tanto la falta de herramientas en una primera instancia (Lo que nos llevó a tener que hacerlo todo de manera simulada) como finalmente el hacerlo de manera simulada nos han complicado de sobremanera el trabajo y han hecho que quedara algo mucho peor y reducido de lo que inicialmente planteamos. Además de que a pesar de que llevemos un año trabajando telemáticamente, se siguen notando las complicaciones, tanto a nivel de coordinación como a nivel de realizar consultas al profesorado y presentar nuestros progresos.

Otro problema que no reparamos a la hora de plantear el proyecto es que tanto la temperatura como la humedad o la luz diurna no cambian lo suficientemente rápido como para tomar muestras cortas de tiempo de una muestra inicial, por lo cual incluso si pudiéramos conectar sensores reales nos sería muy complicado mostrarlo todo de manera sencilla, pues en una corta presentación seguramente los cambios de condiciones sean nulos o muy pequeños.

Dentro de las propuestas de ampliación, si no contamos la de disponer de los elementos físicos, planteamos ideas para que un profesor pudiera conectarse a la aplicación de los distintos huertos de su alumnado y controlarlos de manera remota, a fin de cuentas, de evitar posibles desastres. En general nuestro sistema es muy escalable pues con conectar todo nuevo sensor o herramienta al Arduino se podría emplear con los ajustes consecuentes al código.

Como último detalle nos parecería una gran idea el hecho de llevar la aplicación gráfica a dispositivos móviles ya sean *Android* o *IOS* puesto que son elementos que utilizamos a diario la mayoría de las personas.

## o Herramientas /tecnologías con las que les hubiera gustado contar

En nuestro caso hemos tenido grandes limitaciones a la hora de realizar pruebas de campo y de componentes electrónicos de nuestro proyecto ya que no disponíamos de la gran mayoría de estos y tuvimos que recurrir a simulaciones.

Como elementos electrónicos destacamos la falta de:

- Sensores de luz
- Sensores térmicos
- Sensores de humedad
- Sensores de humo
- Leds
- Piezas
- Y en general todos los elementos presentes en el diseño de *Tinkercad*

En el momento de las pruebas de campo tampoco contamos con un huerto como tal y hubiera sido muy interesante realizar una implementación real para comprobar como interactuaba con el sistema y los elementos de soporte (riego, luz y temperatura).

## o Anexo

En este anexo haremos una suerte de notas de versión donde hablaremos de todas las modificaciones que ha sufrido el *Huertomatic* en esta segunda iteración.

### ***Tinkercad***

Comenzaremos los cambios por su implementación a modo de prototipo en *Tinkercad*, pues es la parte del proyecto que menos modificaciones ha sufrido. En este apartado lo que hemos realizado es hacer que el prototipo sea menos reactivo, haciendo que, en lugar de tener un único umbral para arrancar la ventilación, los calefactores, la luz y el riego de manera automática, habrá distintos rangos en los que los motores irán funcionando a distintas potencias.

Por ejemplo, en lugar de como ocurría en el primer prototipo, que al bajar la temperatura de 25 grados comenzaban los calefactores a funcionar a toda potencia y si subían de esa temperatura paraban, lo que podía causar un constante arranque y parada por parte de los motores, con el desgaste que esto conlleva, ahora tendremos temperaturas umbral que harán que poco a poco según vaya subiendo o bajando la temperatura el calefactor funcione a una mayor o menor potencia, evitando tanta reactividad, lo cual ayudará a que el prototipo tenga una mayor vida útil.

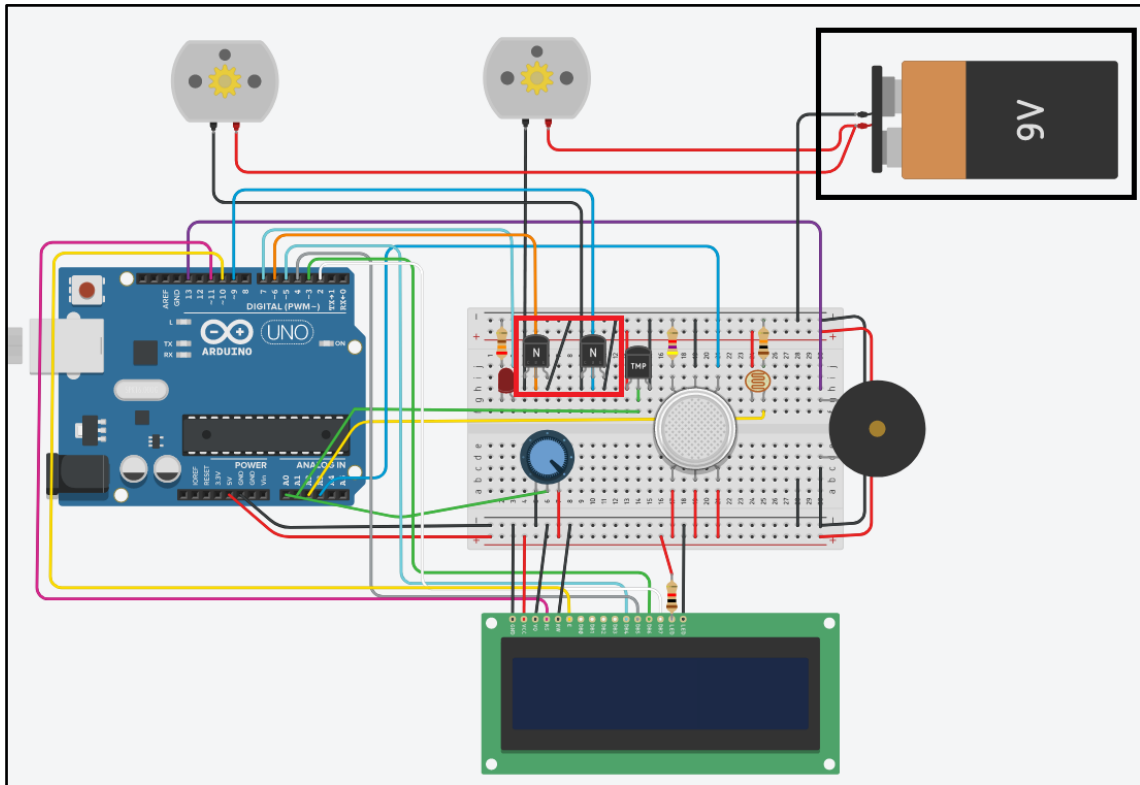
Otro de los cambios que hemos añadido es tratar de evitar cambios constantes en la potencia de los motores, no sólo haciendo que no se apaguen y enciendan a la mínima, sino que no reacción ante cada medida realizada. Al ser un prototipo simulado hemos decidido que, a fin de poder mostrarlo con una mayor soltura y comodidad, el dispositivo muestre y actúe ante la media de las últimas tres medidas tomadas, haciendo una media. Por el contrario, si el prototipo estuviera implementado en la realidad, la decisión que aunque no está implementada, hemos tomado, es la de almacenar una medida cada minuto y cada diez minutos (Más concretamente cada hora que su minuto acaba en 0) se realizará una media de esas medidas y será el cambio que se muestre en pantalla, será la tomada para que los motores cambien su potencia de trabajo y será lo que recibirá el prototipo de *processing*.

La idea tras este añadido, además de evitar hacer que el prototipo esté en un cambio constante, por ejemplo porque se meta una nube y baje la intensidad de la luz, haciendo que el alumbrado se intensifique, y al cabo de pocos segundo todo vuelva al estado normal y el alumbrado se apague o simplemente reduzca, causando un sobreesfuerzo. Otra situación es en la que un sensor falle y esté en constante cambio mandando datos erróneos o cambiantes.

En el prototipo original esto no llegamos a plantearlo porque al trabajar sobre una simulación este tipo de detalles no los tuvimos en cuenta ni experimentados porque una simulación es una situación perfecta, donde los motores no se van a estropear, los sensores no van a dar cambios no esperados o no va a aparecer una nube apenas unos segundos que reduzca la luz.

Reparamos en esta situación gracias a los comentarios recibidos en la presentación realizada en junio. La idea de que en el prototipo real se tomen medidas cada minuto y una media cada 10 minutos surgió de ver cómo funcionaba el control del pulso en un *SmartWatch*, el cual en lugar de estar tomando medidas constantes, lo que es una carga de estrés para la batería, toma una medida cada 10 minutos y la almacena, siendo esta la que muestra durante los siguientes 10 minutos a no ser que se actualice manualmente.

Para realizar estos cambios añadimos elementos nuevos al circuito que podemos ver en la imagen a continuación (**imagen\_modificación**).



*imagen\_modificación*

Añadimos:

- Pila de 9 Voltios
- Transistores

La pila de 9 Voltios (**rectángulo negro**) la añadimos para conseguir una mayor potencia en los sistemas reactivos y tener un rango mayor al regular estos elementos, esto lo hacemos porque la potencia suministrada por la placa es insuficiente.

Los transistores NPN (**rectángulo rojo**) fueron añadidos básicamente, ser un interruptor electrónico o un amplificador. En este caso lo usamos como amplificador.

Esto ha sido implantado para el resto de los elementos, así como también se ha implementado en el prototipo en *processing*, aunque esto se narrará a continuación.

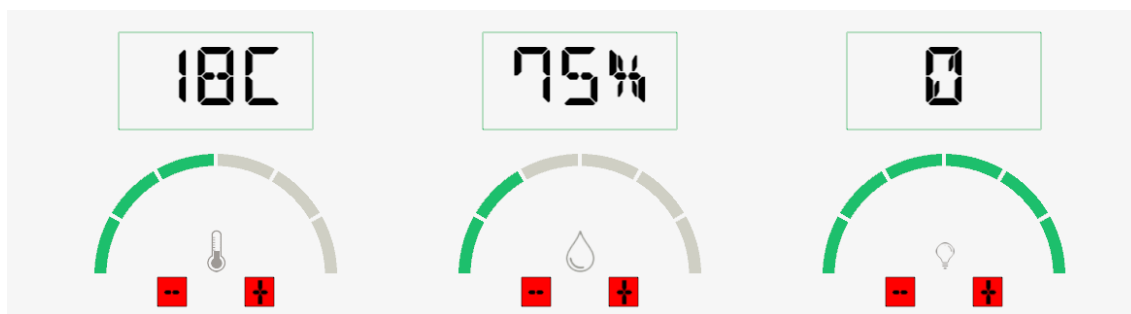
## Processing

El prototipo en *processing* es la parte que ha sufrido la mayor remodelación en esta versión 2.0 de nuestro proyecto, recibiendo una lavada de cara prácticamente completa en el apartado gráfico, siendo uno de los elementos más obvios a simple vista la sustitución del verde del fondo por un gris muy claro, casi blanco, que resulta mucho más amigable para la vista y profesional. El primer afán que hemos perseguido en la remodelación es el de limpiar la interfaz, haciéndola más intuitiva y sencilla, persiguiendo un estilo más minimalista y menos disjunto.

Comentaremos en detenimiento todos los cambios que ha sufrido la aplicación a nivel gráfico, pero antes de entrar en ese apartado y continuando lo comentado al final del apartado de Tinkercad, deseamos comentar que la aplicación de *processing*, en el caso del prototipo real, recibiría el dato de media cada 10 minutos. Por desgracia, al ser una simulación hemos tomado valores de un repositorio y se actualizarían cada hora, haciendo que no sea hasta cada hora que cambien el estado de los motores. Sabemos que cada hora puede ser demasiado tiempo para una aplicación real, pues en caso de malfuncionamiento pueden causarse daños de gran magnitud e incluso irreversibles, pero al ser una simulación para exclusivamente mostrar las funcionalidades hemos decidido que con una marca cada hora será suficiente.

Volviendo al apartado gráfico, hemos eliminado el apartado de gráficos donde se mostraba un historial de temperaturas, intensidad lumínica y humedad del ambiente. En la presente versión podremos disfrutar de los datos de temperatura, humedad e intensidad lumínica, pero de una manera más cómoda, pues veremos exclusivamente sus valores numéricos sin gráficos que pueden resultar complejos de entender y que alejan del objetivo de sencillez y minimalismo perseguido. La idea de mostrar los datos de esta manera nace de los relojes que muestran la hora y la temperatura en las calles.

Junto con el cambio anterior tenemos el indicador de la intensidad de los elementos encargados de regular la temperatura, el riego y la luminosidad, recibiendo también una remodelación, pasando del diseño recto a un diseño en forma de arco (img 3-1) con distintos niveles que nos indicará la potencia a la que trabajan los motores. Estos indicadores, a fin de mantener el diseño minimalista, irán acompañados de un icono sencillo y discreto que indicarán qué elemento regulan. Estos se encuentran justo debajo de los marcadores numéricos de los valores medidos. Esta forma de representación la escogimos al buscar otras aplicaciones de control de invernaderos y gustarnos su representación, en este caso fue la aplicación de *IOGreenhouse*.



img 3- 1

La intensidad de los motores manera predefinida se controlará de manera automática, según distintos rangos de los valores medidos, exactamente igual que en el prototipo de *Tinkercad*, aunque si lo deseamos podremos controlarlo de manera manual, aunque esta opción estará, por defecto, deshabilitada. Más adelante explicaremos la manera de habilitarla.

El siguiente cambio es que hemos añadido un reloj (img 3-2) para conocer la hora actual, aunque por desgracia, al ser un sistema simulado la hora no será real, pero dentro de esta simulación es donde se encuentra la principal funcionalidad del proyecto y será explicado con detenimiento posteriormente a la explicación de los cambios gráficos.



img 3- 2

Este reloj va acompañado de un botón para avanzar el tiempo, algo que será clave para la simulación. En un prototipo no simulado se leería la hora actual y carecería de este botón.

La zona de la cámara es donde menos cambios se han realizado, en el actual prototipo, al ser simulado hemos sustituido lo que se muestra por una imagen fija en lugar de una grabación de cámara, esta imagen sufrirá alteraciones en el RGB para mostrar tonalidades más azuladas (img 3-3) o rojizas (img 3-4) en función de la temperatura medida, también se ha añadido que la imagen se muestre con gotas de agua en función del estado del riego, como si fuera una ventana a la que le caen las gotas.



img 3- 3

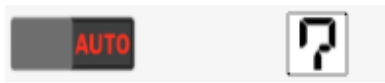


img 3- 4

Justo encima de la cámara encontraremos dos botones (img 3-5), el primer botón servirá para mostrar información de ayuda sobre la aplicación y su uso. El otro botón ya lo hemos anticipado anteriormente, pues será lo que usaremos para pasar a modo manual el control



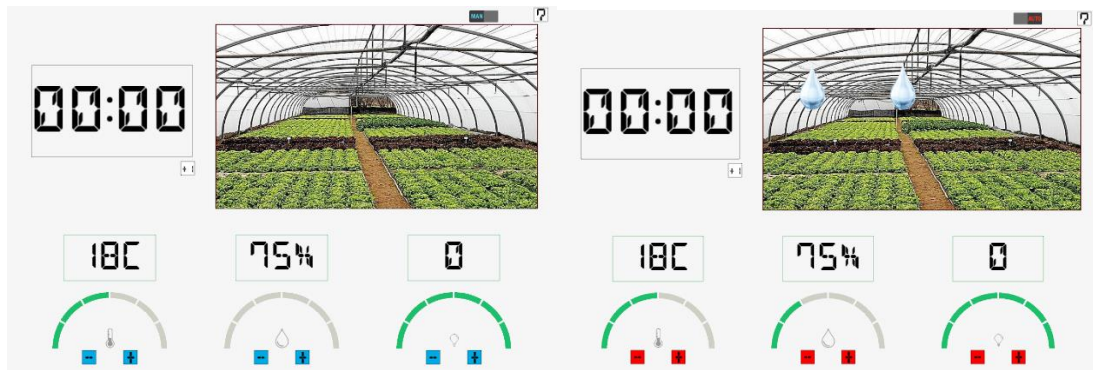
de los elementos que controlan la temperatura, la luz y la humedad. Este botón, representado como un slider sustituye los tres botones que activaban los controles en el prototipo anterior.



img 3- 5

Si activamos el slider pasamos a modo manual, pero se mantienen los valores de intensidad que vienen predefinidos en el modo automático, por el contrario, si modificamos algo y luego volvemos a modo automático inmediatamente se establecerán los valores predefinidos.

Podremos saber si estamos en modo manual (img 3-6) o automático (img 3-7) en función de color de los botones para controlar la intensidad de los motores, si son azulados indicará que estamos en modo manual, por el contrario si se encuentran de color rojizo es que estamos en modo automático y clickarlos será un gasto de energía.



img 3- 6

img 3- 7

Como podemos observar se ha simplificado el diseño, tratando de que todo se sienta mucho menos disjunto, empleando colores que sean agradables para la vista y que eviten que todo se sienta desordenado en conjunto con la disposición.

## Simulación

Una vez narrados los cambios gráficos en nuestra aplicación de *processing* llega el momento de narrar algo que podríamos llamar las nuevas funcionalidades del prototipo.

Como bien hemos indicado la carencia de sensores es algo que ha lastrado nuestro proyecto y si bien en la primera iteración fue algo que no remediamos de ninguna manera, en este caso sí, realizando una simulación y a continuación narraremos cómo se lleva a cabo.

Al carecer de sensores no podemos tener medidas reales de temperatura, humedad e intensidad lumínica, pero por suerte hemos podido tomar valores reales de un día para estos valores, los cuales se le pasarán a la aplicación para disponer de 24 valores de cada

apartado.

Para rescatar los datos y mostrarlos emplearemos el botón para avanzar el tiempo del reloj (img 3-2), y cada hora tendrá unos valores de temperatura, humedad e intensidad lumínica, estos valores podremos observarlos en los distintos *displays* (img 3-1) que ya hemos comentado con anterioridad.

Cuando estos valores cambian sus dispositivos de control pueden o no modificar la intensidad con la que trabajan, pues existe la posibilidad de que superen tanto por su rango máximo como por su mínimo en actual en el que se encuentran, o puede que el cambio no sea lo suficiente como para ser necesaria modificación.

Como ya hemos indicado, en el prototipo real los valores que se recibirían para trabajar no serían de un control constante, sino de una media de medidas realizadas durante 10 minutos, a fin de evitar una actualización constante que pueda llevar a un trabajo extra de los componentes, además, a fin de cuentas una nevera no está trabajando de manera constante, sino que los motores enfrían y luego se parán y están un rato hasta que sea necesario volver a conectarse, pero no por que se pierda medio grado van a volver a trabajar.

## ***Resumen breve***

Para concluir añadiremos un breve resumen de los cambios. En general, además de tener un apartado visual mucho más sobrio, profesional, cohesionado y sencillo de comprender, hemos trabajado en minimizar la carga de trabajo de los componentes físicos.

## **o Créditos materiales no originales del grupo**

Arduino:

<https://www.arduino.cc/reference/en/> -  
<https://www.youtube.com/watch?v=Ke-7vxh1K1A> -  
<https://www.youtube.com/watch?v=Z07mbMsp5bl>

*Processing:*

<https://processing.org/reference/>

<https://github.com/otsedom/CIU>

[Guion práctica asignatura](#)

*Video teaser:*

<https://thispersondoesnotexist.com/>

<https://pixabay.com/es/photos/>

*Inspiraciones:*

<https://www.telcosl.com/index.php/iogreenhouse/>

## **Componentes del grupo**

**Alejandro García Sosa**

**Alejandro Daniel Herrera Cárdenes**

**Carlos Eduardo Pacichana Bastidas**

**Javier Santana Álamo**