



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

**“ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA HIDROPÓNICO
AUTOMATIZADO PARA AUTOCONSUMO DE VEGETALES Y PLANTAS
ORNAMENTALES”**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

JOSÉ PEDRO LÜBBERT CARMONA

PROFESOR GUÍA:

ÁNGEL JIMÉNEZ MOLINA

MIEMBROS DE LA COMISIÓN:

IGNACIO CALISTO LEIVA

ALBERTO CABEZAS BULLEMORE

SANTIAGO DE CHILE

2016

**RESUMEN DE LA MEMORIA PARA
OPTAR AL TÍTULO DE:** Ingeniero Civil
Industrial

POR: José Pedro Lübbert Carmona

FECHA: 11 de Marzo, 2016

PROFESOR GUÍA: Ángel Jiménez Molina

**ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA HIDROPÓNICO
AUTOMATIZADO PARA AUTOCONSUMO DE VEGETALES Y PLANTAS
ORNAMENTALES**

El presente trabajo muestra la implementación de la metodología de desarrollo de Software RUP para investigar y resolver cómo construir un prototipo funcional orientado a desarrollar las actividades que típicamente se deben hacer para mantener un cultivo hidropónico de forma automática, con la posibilidad de dejar programados los valores de variables ambientales que debe controlar el dispositivo.

La metodología de desarrollo de software pretende investigar inicialmente el proceso global de los cultivos hidropónicos, para luego identificar las labores más importantes que se deben desarrollar en la práctica, con el fin de identificar por separado las distintas acciones que debe llevar a cabo el sistema, para finalmente formalizar la programación e implementación de dichos procesos de forma ordenada. Dicha metodología permitió encontrar de forma exitosa modelos conceptuales claves para el desarrollo de futuras iteraciones del desarrollo de este prototipo, en conjunto con aportar fuertemente al orden del código.

La experimentación logró concluir que los sensores utilizados para medir la cantidad de sales disueltas en el agua funcionan con precisión suficiente para poder desarrollar el sistema de hidroponía.

Índice

1. Introducción y Descripción del Proyecto	1
1.1. Introducción.....	1
1.2 Descripción del Proyecto.....	3
2 Objetivos y Metodología	4
2.1. Objetivos de la Memoria.....	4
2.1.1 Objetivo General.....	4
2.1.2 Objetivos Específicos.....	4
2.2 Marco Conceptual	4
2.3 Metodología	11
2.3.1 Análisis, Diseño y Construcción de Producto	11
2.3.1.1 Revisión Bibliográfica	11
2.3.1.2 Diseño y Construcción del Producto.....	11
2.3.1.2.2 Construcción del Prototipo de Pruebas	12
2.3.2 Análisis del Entorno de Negocio	12
2.3.2.1 Análisis de Sector.....	12
2.3.2.2 Análisis de Competencia	13
2.3.2.3 Análisis de Mercado	13
2.3.2.3.1 Curvas de Difusión de Bass.....	13
2.3.2.3.2 Análisis de 5 fuerzas de Porter	14
2.4 Alcances.....	15
3. Análisis, Diseño y Construcción del Producto	16
3.1. Análisis Bibliográfico.....	16
3.2 Descripción del producto	16
3.3 Diseño de Sistema Hidropónico Automático.....	17
3.3.1 Límites y Alcances de proyecto:.....	17
3.3.2 Riesgos.....	17
3.3.3 Requerimientos.....	18
3.3.4 Factibilidad del proyecto	18
3.3.5 Resumen de Análisis	18
3.4 Construcción del Sistema.....	18
3.4.1 Visión y Análisis de Negocio	18
3.4.2 Elaboración.....	19
3.4.2.1 Análisis de Dominio de Sistema:	19

3.4.2.2 Requerimientos del sistema	20
3.4.2.2.1 Requisitos funcionales	20
3.4.2.2.1.1 Casos de Uso	20
3.4.2.2.1.2 Diagrama de Clases	26
3.4.2.2.2 Requisitos No Funcionales.....	27
3.4.2.3 Riesgos Revisados.....	27
3.4.3 Construcción.....	27
3.4.3.1 Hardware necesario	28
3.4.3.2 Modificaciones al Diseño	29
3.4.3.3 Interfaz Usuario-Sistema	29
3.4.3.4 Algoritmo y Código	29
3.4.3.4.1 Setup	30
3.4.3.4.2 Loop.....	31
3.4.3.5 Experimentación y Resultados de Laboratorio.....	45
4. Análisis del Entorno	48
4.1 Análisis de Sector.....	48
4.2 Análisis de Competencia	50
4.3 Análisis de Mercado	51
4.4 Análisis de 5 Fuerzas de Porter.....	51
4.5 Modelo Delta	52
5. Conclusiones	54
Glosario	55
Bibliografía.....	56
Anexos	58

1. Introducción y Descripción del Proyecto

1.1. Introducción

El crecimiento poblacional ha ido en constante aumento desde el término de la Segunda Guerra Mundial, a medida que se fueron introduciendo nuevas herramientas para mejorar la calidad de vida, salud, procesos de seguridad y otros que han potenciado un ritmo estable. Al año 1959 había aproximadamente 3 mil millones de habitantes. Se estima que para el año 2044, habrá 9 mil millones de personas¹.

En un estudio del año 2009, la Food and Agriculture Organization, organismo de Naciones Unidas encargado de estudiar los datos relacionados con los pasos para alimentar a una siempre creciente población, estimó que la producción de alimentos mundial deberá aumentar un 70% hacia el 2050, por sobre la producción de dicho año, con especial énfasis en la optimización de los recursos utilizables, como lo son el agua y los terrenos fértiles².

Sumado a lo anterior, se ha visto que el gasto público en mejoras agrícolas se ha estancado desde mediados de los 90 en países desarrollados, con sólo un tercio del gasto global realizado por éstos, mientras que más del 50% de los gastos en desarrollo agrícola se ha concentrado en los países en vías de desarrollo, como Brasil, China, India, y Sudáfrica³.

Para aumentar la producción, las tecnologías agrícolas han hecho uso de semillas transgénicas y otras técnicas invasivas, lo que ha llevado al consumidor final a pensar seriamente en lo que está consumiendo en su mesa. Este pensamiento, en conjunto con una mayor preocupación con los estilos de vida y la salud, hacen con que las personas dirijan su atención hacia los productos orgánicos, los cuales han ido en aumento en participación de mercado en los últimos años. Estos alimentos orgánicos son, en su mayoría, cultivados en sistemas hidropónicos, debido a tener una menor necesidad de pesticidas y otros elementos nocivos al ser cultivados en invernaderos. No sólo eso, sino que hoy en día, con el incremento de videos DIY en youtube, y tutoriales para hacer todo en casa, ha llevado a los consumidores finales a pasar de una sociedad consumista, a una sociedad autosuficiente, en donde muchas familias están cultivando sus propios vegetales, haciendo sus propias construcciones, desarrollando sus propios inventos. Es aquí donde la hidroponía automatizada entra a ser un factor importante a la hora de “crear tu propia huerta hidropónica”

La hidroponía consiste en cultivar plantas de todo tipo sin utilizar suelo. Actualmente se utilizan principalmente en plantas de tipo herbáceo, las cuales pueden crecer en lugares poco convencionales, considerando todas las demás necesidades biológicas de la planta. En esta técnica, “los elementos minerales esenciales son aportados por la solución nutritiva”. Se ha visto que la producción de un cultivo hidropónico puede obtener dos veces o más lo producido por un cultivo tradicional en suelo fértil. Hay diversos factores que inciden directamente en el rendimiento de producción, como plagas o patógenos, la disponibilidad de recursos, o la densidad por metro cuadrado del cultivo.

Los avances tecnológicos en informática han permitido que se desarrollen las actividades de control y ejecución a través de la automatización, y la hidroponía no es ajena a estos avances. Un cultivo hidropónico que se realiza en un ambiente controlado, en donde la variación de temperatura, nutrientes, disponibilidad de recursos y otros aspectos relacionados, es replicable, por lo que se ha visualizado como una solución a los problemas asociados al crecimiento humano, como son la contaminación, el cambio climático, crecimiento explosivo de ciudades, y la disminución de terrenos agrícolas por desertificación⁴.

Esta memoria presentará un prototipo de cultivo hidropónico automatizado, el cual permitirá a un productor o un particular, sin grandes conocimientos en hidroponía preparar, cultivar y cosechar hortalizas y frutos hidropónicos, contando con un constante monitoreo y optimización de los recursos para una máxima eficiencia agrícola.

1.2 Descripción del Proyecto

Esta memoria abarcará el análisis, diseño y construcción de un dispositivo de automatización hidropónica, tomando en cuenta el desarrollo del software necesario, así como la construcción del hardware para su ejecución. Dentro del desarrollo del software, se explicarán paso a paso las funciones creadas para cada estado del dispositivo, así como las acciones que realiza el sistema. Dentro del Hardware, se expondrán las piezas que se utilizaron para la construcción física del sistema, y cómo se programaron para que funcionaran con las acciones deseadas.

Además, se hará un análisis del entorno de negocio para un potencial producto comercializable, el cual se propone ser alcanzado en los próximos meses. Para efectos de esta memoria, sólo se expondrá el análisis de entorno, y no se realizará la proyección del negocio debido a la falta del producto finalizado.

2 Objetivos y Metodología

En este capítulo se mencionan los objetivos, marco conceptual, descripción del negocio, y la metodología necesaria para cumplir con los objetivos de la memoria.

2.1. Objetivos de la Memoria

El objetivo general plantea el propósito principal por el cual el trabajo es desarrollado. Para cumplir con dicho objetivo, se enumeran objetivos específicos, los cuales son hitos necesarios para llegar de forma correcta al objetivo general. Esta sección plantea tanto objetivo general como específicos.

2.1.1 Objetivo General

El objetivo general de esta memoria se plantea de la siguiente manera:

“Análisis, diseño y construcción de un sistema de hidroponía automatizada para el autoconsumo de vegetales y plantas ornamentales en Chile”

2.1.2 Objetivos Específicos

Los objetivos específicos son los siguientes:

1. **Desarrollo de Producto:** Dentro de este punto se encuentran:
 1. Análisis de Sistemas Automatizados
 2. Análisis de Sistemas Hidropónicos Actuales
 3. Diseño de Sistema Automatizado Hidropónico
 4. Construcción de Sistema Automatizado Hidropónico
 5. Fase de Experimentación y Resultados Obtenidos
 6. Estudios en Desarrollo

2. **Análisis del entorno del negocio:** En esta sección encontramos los siguientes puntos:
 1. Análisis de Mercado de Cultivos en Chile
 2. Análisis de Cultivos Hidropónicos en Chile
 3. Análisis de Productos de Consumo en Agronomía Hidropónica
 4. Análisis Porter

2.2 Marco Conceptual

En esta sección se describirán brevemente los conceptos de cultivos automatizados, hidroponía, desarrollo de software, desarrollo de innovación, modelos de arquitectura de empresa y todos los conceptos aplicables a un nuevo producto de innovación agrónoma, marcando las bases sobre las cuales se desarrolló el prototipo funcional y plan de negocios de los cuales trata esta memoria.

Sistemas Automatizados en Agricultura

Los cultivos agrícolas han pasado por diversos cambios en los últimos años, especialmente con la introducción de sistemas automatizados para el control, riego, y cosecha, tanto en sistemas de invernadero como al aire libre. Hablando específicamente de los sistemas en invernadero, la automatización ha sido más amplia, por el requerimiento de mayor control de las variables frente a condiciones no óptimas para cultivar. Los países pioneros en este tipo de cultivos automatizados han sido los de Europa del Norte, especialmente Dinamarca, en donde el primer sistema automatizado de cultivo en invernadero vio la luz hace aproximadamente medio siglo. El sistema de cultivo automatizado original se trató como un proceso de manufactura industrial, en donde varios procesos de cultivo se unen en un proceso de lotes, y se ejecutan como si se tratase de una línea de montaje. Sin embargo, actualmente el énfasis se encuentra en el control de los procesos, y no como en un proceso de manufactura, particularmente en los cultivos en invernaderos, lo cuales pueden acelerar el crecimiento del fruto de 3 a 5 veces más que en medios naturales⁵.

Sistemas hidropónicos en agricultura

La hidroponía como sistema de cultivo se ha utilizado por miles de años. Los primeros indicios de su uso se remontan a Babilonia, cuando el rey Nabucodonosor II ordenó la construcción de los famosos Jardines Colgantes de Babilonia, que utilizaban un sistema de irrigación hidropónico por gravedad. Otras culturas como la China, Azteca, India, Egiptia también utilizaron sistemas similares, destacándose la Azteca con sus sistemas de balsas (Chinampas) sobre el lago Tenochtitlán, logrando obtener diversas hortalizas, flores y hasta árboles frutales.

Experimentos realizados entre los siglos XVII y XIX demostraron que el agua con nutrientes era un elemento esencial para el crecimiento de las plantas, permitiendo que las plantas crecieran "sólo" en un medio acuoso con "tierra nutritiva disuelta". Sin embargo, el conocimiento necesario para hacer hidroponía en la actualidad es una de las grandes barreras de entrada a este tipo de cultivos.

Actualmente se deben diferenciar dos tipos principales de cultivo hidropónico:

- a. Cultivo Hidropónico sin sustrato: también conocido como Aeroponía, en donde las raíces de la planta tienen contacto directo con el agua con nutrientes.
- b. Cultivo en Agregado: en donde la planta se coloca en un medio inerte capaz de permitir un buen regado de las raíces pero con el suficiente espacio para drenar bien, como arcilla expandida, gravilla, musgo, etc.

Diversos estudios de cultivos hidropónicos han demostrado que el rendimiento de cultivo es mayor que el cultivo en tierra, pudiendo llegar hasta 3 veces más que el cultivo en tierra⁶. Los diferentes sistemas de cultivo de hidroponía se pueden encontrar en el Anexo 1, "Sistemas Hidropónicos", de esta memoria.

Hidroponía Automatizada

Los cultivos hidropónicos no han sido utilizados masivamente por productores por la barrera de entrada, que requiere mucho conocimiento de base e inversión para ser aplicada. Una forma de simplificar la entrada a la hidroponía es la hidroponía automatizada. En esta, los procesos complejos de manejo de nutrientes son realizados por un actuador, que, basado en sensores, regula y optimiza el nivel de los nutrientes en el agua, para que la planta crezca en condiciones óptimas. Este tipo de cultivo es algo relativamente nuevo, con los primeros sistemas comerciales completamente automatizados apareciendo hacia el año 2012, sin embargo, estos sistemas continúan hasta el día de hoy teniendo la barrera del costo. (Ver figura 15: "Comparación de Precios de Sistemas Hidropónicos Actuales" en Anexo 1: "Sistemas Hidropónicos")

Un sistema hidropónico de bajo costo y de fácil uso aún está por ser desarrollado en el mercado.

Rational Unified Process

El RUP o Proceso Racional Unificado es un Proceso de Ingeniería de Software que entrega un acercamiento disciplinado a la entrega de tareas y responsabilidades dentro de una organización de desarrollo, que asegura productos de alta calidad que cumplen con las necesidades de los usuarios finales, con una calendarización y presupuestos predecibles.

El RUP mejora la productividad de equipo al entregarles acceso fácil a una base de conocimientos con guías, planillas y herramientas críticas para toda actividad de desarrollo, lo que asegura que todo el equipo de trabajo entienda el mismo idioma y los mismos procesos, y la visión global del proyecto.

Es una guía para usar efectivamente el Unified Modelling Language (UML) que es un lenguaje estandarizado que permite comunicar arquitecturas, diseños y requerimientos claramente. Puede ser utilizado tanto por equipos pequeños de desarrolladores como grandes empresas de desarrollo. El RUP contiene 6 "*mejores prácticas*" que son comúnmente utilizadas por las organizaciones exitosas, y se describen a continuación:

- a. Desarrollo de Software Itinerante
- b. Manejo de Requerimientos
- c. Uso de Arquitectura basada en componentes
- d. Modelado Visual
- e. Verificación de la Calidad del Software
- f. Control de Cambios al Software

Un proyecto UP organiza el trabajo y las iteraciones en cuatro fases fundamentales:

- a. Inicio: Visión aproximada, análisis del negocio, alcance, estimaciones imprecisas
- b. Elaboración: visión refinada, implementación iterativa del núcleo central de la arquitectura, resolución de los riesgos altos, identificación de más requisitos y alcance, estimaciones más realistas
- c. Construcción: implementación iterativa del resto de requisitos de menor riesgo y elementos más fáciles, preparación para el despliegue
- d. Transición: pruebas beta, despliegue

Debe considerarse que este modelo **no es** un modelo "en cascada", o secuencial, en donde primero se definen todos los requisitos y después se realiza o ejecuta todo el proyecto, sino que funciona de la siguiente manera: La fase de inicio es más bien una fase de "viabilidad" del proyecto, donde se lleva a cabo sólo el estudio suficiente para decidir si seguir o no. La fase de Elaboración no es la fase de requisitos o diseño, sino que es una fase de implementación, a través de iteraciones la arquitectura que constituye el núcleo central y se mitigan las situaciones de alto riesgo. La tabla a continuación muestra el marco de desarrollo de artefactos UP:

Tabla 2.1 Ejemplo de Marco de Desarrollo de artefactos UP. c-comenzar; r-refinar

<i>Disciplina</i>	<i>Artefacto</i> <i>Iteración →</i>	<i>Inicio</i> <i>I1</i>	<i>Elab.</i> <i>E1...En</i>	<i>Const.</i> <i>C1...Cn</i>	<i>Trans.</i> <i>T1...T2</i>
Modelado del Negocio	Modelo del Dominio		c		
Requisitos	Modelo de Casos de Uso	c	r		
	Visión	c	r		
	Especificación Complementaria	c	r		
	Glosario	c	r		
Diseño	Modelo de Diseño		c	r	
	Documento de Arquitectura SW		c		
	Modelo de Datos		c	r	
Implementación	Modelo de Implementación		c	r	r
Gestión del Proyecto	Plan de Desarrollo SW	c	r	r	r
Pruebas	Modelo de Pruebas		c	r	
Entorno	Marco de Desarrollo	c	r		

Figura 1: Marco de Desarrollo de artefactos UP

Fuente: Larman, C. *UML Y PATRONES*, Segunda Ed. Pearson Educación, 2003 p.23

El Modelo Delta

Los dos modelos más importantes surgidos en los últimos tiempos han sido el modelo de Ventaja Competitiva, de M. Porter; y el modelo de Visión de la Empresa Basado en Recursos, de Prahalad & Hamel. Para Porter, la rentabilidad de un negocio depende de 2 determinantes básicas: La estructura de la industria en la cual está inserto el negocio, y el posicionamiento competitivo del negocio en esa industria. Según Porter, para ser una de las empresas más rentables, debe posicionarse de manera adecuada, aumentando su poder negociador con su público externo (clientes, proveedores, competidores) y creando barreras de entrada contra productos sustitutos y/o nuevos en la industria. Esto determina, en el marco de Porter, las "industrias buenas", que tienen un amplio margen de rentabilidad, y las "industrias malas", donde la mayoría de los participantes sufren pérdidas económicas. Para esto, Porter desarrolló un esquema de 5 fuerzas el cual está descrito abajo:

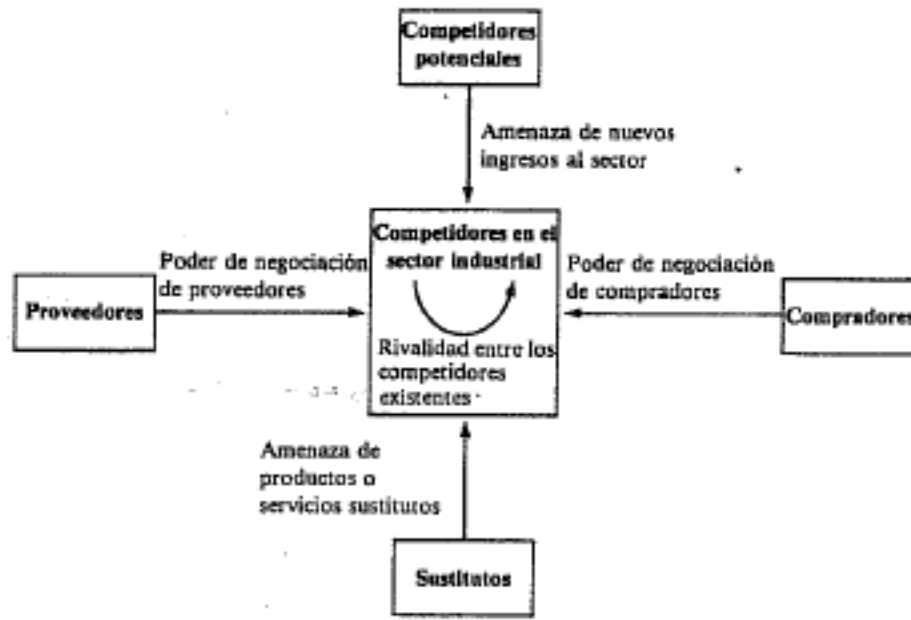


Figura 2: Cinco fuerzas competitivas
 Fuente: Porter M. *Ventaja Competitiva*, Ed. Rei Argentina, 1991, p.23

Además, describe las actividades para crear valor, y lograr un mejor posicionamiento competitivo, a través del siguiente esquema:



Figura 3: Cadena de Valor Genérica
 Fuente: Porter, M. *Ventaja Competitiva*, Ed. Rei Argentina, 1991, p.55

Cabe destacar que este esquema es para cualquier empresa, y puede ser aplicado en cualquier industria. Tomando en cuenta el planteamiento de M. Porter se puede concluir que sólo hay dos maneras de competir en cualquier industria:

- Bajo Costo
- Diferenciación de Producto

El bajo costo se consigue a través de la utilización de economías de escala, mientras que la diferenciación se consigue creando un producto que el cliente percibe como único y valioso, pagando un premium por ello. Esta perspectiva implica un concepto de "guerra" en la estrategia de negocios (crear barreras, obtener por sobre el competidor los proveedores, clientes y absorbiendo competidores, etc). El vínculo con el cliente es volátil.

El modelo de "Visión de la Empresa Basado en Recursos" se presenta en el esquema a continuación:

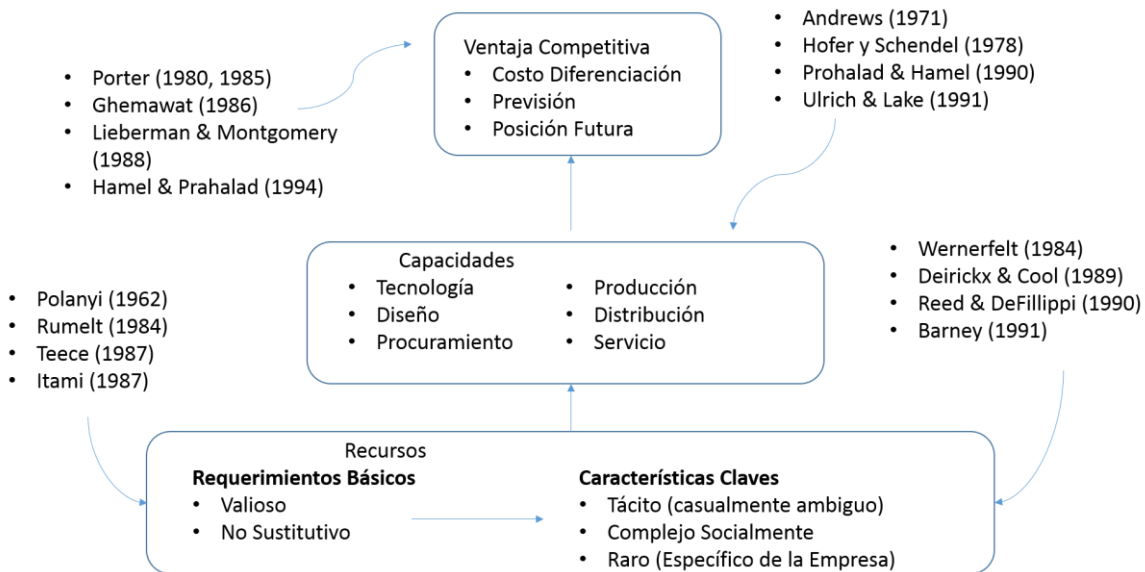


Figura 4: "Visión de la Empresa Basado en Recursos"
Fuente: Elaboración Propia, extraído de "Academy of Management"¹

Estos dos modelos, el de "Ventaja Competitiva", y el de "Visión de la Empresa Basada en Recursos", tienen como actor principal a la industria y a la empresa, respectivamente, y obtienen más clientes a través de la estrategia de "Mejor Producto". Sin embargo no se describen como modelos integrales que involucran todos los actores en la economía de hoy, ni tampoco describen todas las formas en que compiten las empresas.

El modelo Delta de Hax sitúa al Cliente en el centro de la gestión, a diferencia de los dos modelos clásicos descritos. Este modelo plantea tres ejes dinámicos no excluyentes sobre los cuales se puede basar la estrategia de negocios. Estos ejes son:

- Mejor Producto
- Servicio Integral al Cliente
- Consolidación del Sistema

A continuación se presenta un esquema en donde se pueden ver la relación entre estos tres ejes, y las diversas estrategias que nacen a partir de éstas:



Figura 5: Posturas Estratégicas
 Fuente: Hax, A. & Wilde D., *The Delta Project*, 2001

Cada uno de estos ejes consideran además varias sub estrategias posibles de desarrollar, las cuales se expresan a continuación:

Mejor Producto:

Se basan en economías de productos, y aplicando un poco la fórmula clásica de ventaja competitiva. Se observan dos sub estrategias:

- Liderazgo en Costo: El foco principal es ser el líder en el precio, en una categoría de productos no diferenciada
- Diferenciación: El foco principal es la diferenciación de las características y funcionalidades del producto, para que sean valiosos y únicos para el cliente, consiguiendo precios más altos.

Servicio Integral al Cliente:

Se basan en economías de cliente, buscando menor costo o mayores beneficios para el mismo. Se observan tres sub estrategias:

- Redefinición de la experiencia del cliente: El foco principal es la experiencia completa del cliente, desde la adquisición a lo largo de todo el ciclo de vida del producto.
- Amplitud Horizontal: El foco principal es ofrecer un grupo de productos y/o servicios que cubran todas las necesidades del cliente, en un solo lugar.
- Integración de clientes: El foco principal es sustituir o potenciar las actividades que están siendo realizadas por el cliente, mejorando su habilidad para hacer negocios y usar el producto.

Consolidación del Sistema

Se basan en economías de sistemas, buscando potenciarse a través de empresas complementarias, ejerciendo su actuación sobre tres variables: Lock-in (enganche) de clientes, Lock-out (exclusión) de competidores, y propiedad de estándares. Se observan tres sub estrategias:

- **Accesos Restringidos:** El foco principal es crear barreras de entrada a la competencia hacia los clientes, sin embargo esta estrategia tiende a ser regularizada para prevenir esta situación
- **Mercado Dominante:** El foco es desarrollar una relación comprador-vendedor difícil de superar o reemplazar, siendo primordial "ser el primero"
- **Estándar Propietario:** El foco es atraer al cliente a través de la red de empresas complementarias orientadas a trabajar con el producto, generando remuneraciones considerables para la empresa que logra esta estrategia.

2.3 Metodología

En este ítem se describirá la metodología utilizada para la creación del prototipo, incluyendo revisión bibliográfica, análisis de contexto, y otros ítems asociados, así como el desarrollo del plan de negocios para posicionar el producto en un mercado que aún tiene mucho crecimiento.

2.3.1 Análisis, Diseño y Construcción de Producto

2.3.1.1 Revisión Bibliográfica

En la primera parte de la creación del prototipo, se realizó una revisión bibliográfica sobre los cultivos hidropónicos, sistemas hidropónicos automatizados, uso de microcontroladores en automatización de procesos, uso de sensores para medir cambios ambientales en caldo de cultivo y atmósfera, comunicación de microcontrolador con otros dispositivos, almacenamiento de información en memoria extraíble, estudio de bombas hidráulicas para movimiento de diversos fluidos, entre otros. En el Anexo 2 de esta memoria, "Sistemas Automatizados y Arduino" se encuentran los datos con respecto a la automatización, incluyendo el hardware utilizado. Todo esto conllevó a un segundo paso, en donde se filtró la información leída para desarrollar una propuesta de diseño, la cual se basó en el uso de microcontrolador para modular el flujo de nutrientes al sistema. Un tercer paso fue realizar el diseño del producto, el cual se respaldó con entrevistas realizadas a expertos en venta de productos agrícolas e hidropónicos.

2.3.1.2 Diseño y Construcción del Producto

2.3.1.2.1. Diseño de Hardware y Software

A través de las entrevistas realizadas, y con los datos previamente analizados, se realizó el diseño del producto siguiendo un protocolo RUP de desarrollo de Software, el cual nos

permite desarrollar nuevos productos de software con un lenguaje unificado para todos los involucrados con el proyecto, a través de procesos definidos.

Esto llevó a desarrollar un modelo en donde se enumeraron los requerimientos para el proyecto, para posteriormente pasar al análisis de los datos, y al diseño del producto.

Se utilizaron sensores de diferente índole, los cuales fueron analizados por cada uno de sus características, y se probaron con el microcontrolador para mapear los procesos de comunicación y calibrado.

Las características deseables en el prototipo fueron las siguientes:

- Movimiento de agua del reservorio al sistema
- Recirculamiento del agua dentro del sistema
- Drenaje de agua del sistema
- Medición y control de pH
- Medición y control de Fertilizante
- Dispositivo de comunicaciones a distancia
- Integración de puerto de memoria extraíble
- Registro de intensidad lumínica

2.3.1.2.2 Construcción del Prototipo de Pruebas

Con el diseño aceptado, se pasó a la construcción del hardware y a la programación del software, revisando el check-list de cada uno de los requerimientos y los aspectos deseables del prototipo. Una vez con el prototipo en mano, se pasó a las pruebas de laboratorio, en donde se probaron cada una de las funciones del prototipo (regulación de concentración de nutrientes, pH, etc.) para realizar los ajustes necesarios a la programación, y obtener un prototipo para pruebas en terreno.

2.3.2 Análisis del Entorno de Negocio

En esta sección se describirán los procesos que se usan para realizar un análisis de entorno y las herramientas necesarias para determinar la competitividad de un producto. Este análisis se realizará a través de las 5 fuerzas de Porter, el cual es un modelo universalmente aceptado para el análisis de industria y desarrollo de la estrategia empresarial.

2.3.2.1 Análisis de Sector

El análisis de sector sirve para entender la forma, el tamaño, tendencias y características del sector, y cómo el producto o servicio encaja en él. Para desarrollar el análisis, la metodología consiste en definir los siguientes puntos:

- Gama de productos y servicios que forman el sector.
- Capacidad de producción del sector, unidades de ventas y rentabilidad.
- Ubicación del sector.
- Tasa estimada de crecimiento del sector.

- Patrones nuevos de crecimiento que están emergiendo.
- Tamaño y número de competidores.
- Tendencias del sector (conservador o vanguardista).
- Barreras de entrada, como recursos necesarios para entrar en el sector, barreras del tipo legal, requerimiento de capital, o conocimiento.

2.3.2.2 Análisis de Competencia

El análisis de competencia sirve para identificar competidores en el mercado, el grado de rivalidad entre ellos e identificar posibles aliados o enemigos, y tiene por objetivo la evaluación del mercado en términos competitivos. Para llevarla a cabo se necesitan definir los siguientes puntos:

- Identificar principales competidores, fortalezas y debilidades, porcentaje de mercado de los mismos, estrategias de marketing, y factores clave de éxito de cada uno.
- Definir diferencias entre competidores, explicitando qué necesidad de los clientes se está explotando.
- Evaluar competencia y definir amenazas.

2.3.2.3 Análisis de Mercado

El análisis de mercado sirve para evaluar a qué número de individuos les podría interesar el producto, y sus principales características. Esta información es vital para dar con un buen marketing mix. La metodología consiste en definir los siguientes puntos:

- Definición de mercado objetivo.
- Tamaño de mercado.
- Propuesta de Valor para Mercado Objetivo.

Para explicar la evolución del mercado a través del tiempo, se utilizaron curvas de difusión de Bass, las cuales se explican a continuación:

2.3.2.3.1 Curvas de Difusión de Bass

Estas curvas de difusión explican en proceso por el cual nuevos productos son adoptados por la población. El modelo dice en resumen que del % de población que no conoce el producto siempre salen “innovadores” los cuales compran productos sin que nadie de su entorno los haya probado, como también existe población que sólo va a probar determinado producto viendo a otras personas usándolo, denominados “imitadores”. Esto es representado por una ecuación diferencial que tiene la siguiente forma:

$$\frac{f(t)}{1 - F(t)} = p + qF(t)$$

Donde

$f(t)$ es la variación de fracción de usuarios en la población estimada de usuarios.

$F(t)$ es la fracción de usuarios en la población estimada de usuarios.

p es la tasa de adopción de innovadores.

q es la tasa de imitación.

2.3.2.3.2 Análisis de 5 fuerzas de Porter

Las fuerzas de Porter es una forma de ver los negocios que tiene como finalidad maximizar recursos y superar a la competencia, cualquiera sea el giro de la empresa. Estas indican la intensidad de la competencia y la rivalidad en la industria, dejando en evidencia el atractivo de la industria en términos de oportunidades, inversión y rentabilidad, por lo que es un análisis estándar de fuerzas externas. Estas definen el entorno inmediato de una empresa, y afectan la habilidad para satisfacer clientes y con esto obtener ganancias. A continuación se explican las 5 fuerzas de Porter:

Fuerza n°1: Poder de Negociación de los Compradores o Clientes

Se refiere al poder que tienen los clientes para negociar precios, lo que se puede relacionar con clientes organizados, ya sea porque su tamaño es pequeño, siendo más fácil orquestarlos para que en conjunto manejen los precios, o por la existencia de muchos proveedores, pudiendo el cliente escoger entre muchos otros proveedores.

Fuerza n°2: Poder de Negociación de los Proveedores o Vendedores

Se refiere al poder que tienen los vendedores para negociar precios, lo que puede tener que ver con la poca disponibilidad de otros vendedores, por las características del producto, por el impacto que los insumos vendidos afecten a la industria en la cual se venden.

Fuerza n°3: Amenaza de nuevos Competidores Entrantes

Este punto se relaciona con las barreras de entrada de los nuevos competidores o productos, siendo una mayor amenaza cuando es más difícil entrar por dichas barreras. Se identifican 7 distintas barreras de entradas conocidas, las cuales son:

- Economías de Escala.
- Diferenciación del Producto.
- Inversión de Capital.
- Desventaja en costos independiente de la escala.
- Acceso a Canales de Distribución.
- Política Gubernamental.
- Barreras de Entrada.

Fuerza n°4: Amenaza de los Productos Sustitutos

Este punto se relaciona con mercados en donde existen muchos productos iguales o similares, suponen por lo general una rentabilidad baja. Estos se relacionan con los siguientes factores:

- Propensión del comprador a sustituir.
- Precios relativos de los productos sustitutos.
- Coste o facilidad del comprador.
- Nivel percibido de diferenciación de producto o servicio.
- Disponibilidad de sustitutos cercanos.
- Suficientes proveedores.

Fuerza n°5 Rivalidad entre los competidores:

La rivalidad entre los competidores es el resultado de las 4 fuerzas anteriores. Esta fuerza define la rentabilidad del sector, mientras menos competidores existan en un sector, más rentabilidad es asegurada y viceversa. A continuación se muestra un esquema que explica las Fuerzas de Porter antes mencionadas.

2.4 Alcances

Esta sección tiene por objetivo determinar los alcances de la memoria, en particular, los límites tanto en extensión como en profundidad de análisis de los temas abordados, y cómo se abordarán los problemas que provoquen algunos alcances de la memoria.

Alcance de investigación de mercado:

La investigación se hace en torno a definir un mercado objetivo y hacer un plan de marketing, en el cual se asume que existen los segmentos “innovators” y “followers”. En este documento no se hizo un estudio de mercado de los segmentos “innovators” y “followers”, lo cual limita las elecciones de valores necesarios para una predicción precisa. Sin embargo, para abordar este tema, se utilizaron diferentes horizontes para las predicciones por lo que se cuenta con distintos escenarios. El mismo recurso se utiliza para estimar el mercado real. Para esto se estimó un número de mercado el cual es un porcentaje de la población y se tomaron valores pesimistas, semi-pesimistas, normales y optimistas, todos en rangos de la población del mercado objetivo.

Alcance de diseño de producto:

Se consideraron las variables del programa a desarrollar, en conjunto con los requerimientos para llegar a un prototipo funcional, para así pasar a las pruebas de terreno, que es el paso a seguir posterior a la fecha de esta memoria, y finalmente desarrollar el producto final.

3. Análisis, Diseño y Construcción del Producto

En este capítulo se explicarán los procesos utilizados para llegar a la fase actual del producto, el cual se encuentra en su etapa de prueba de terreno. Se explicarán estas fases a través del sistema de RUP, explicando cada una de las 4 etapas de desarrollo, agregando las iteraciones que se realizaron.

3.1. Análisis Bibliográfico

Durante la revisión bibliográfica, se encontró que no existe un sistema de bajo costo funcional para autocultivo de hortalizas, frutas, vegetales y plantas ornamentales en el mercado actual. Es más, algunas universidades están apenas iniciando investigaciones sobre el uso de microcontroladores para la hidroponía automatizada^{7 8}. Esto significa que es reciente el interés por los Sistemas Hidropónicos Automatizados (SHAs), si bien el interés por hidroponía es antiguo. Además, se determinó que en Latinoamérica no existen sistemas hidropónicos automatizados, sino que son importaciones extranjeras, las que tienen un alto costo monetario⁹ además de una implementación compleja ya que las empresas que ofrecen productos de este tipo normalmente ofrecen a escalas pequeñas (para implementación por el usuario, no más de 16 plantas a la vez) o a grandes escalas (sólo con la instalación realizada por la empresa en el lugar de cultivo).

Debido a esta revisión, se llegó a la conclusión que un sistema realizado en Chile para cultivadores Chilenos era factible, e implementable desde el punto de vista de optimización de recursos, especialmente en zonas en donde el agua es un recurso escaso y/o congelado^{10, 11}.

3.2 Descripción del producto

El producto nace a partir de la necesidad de cubrir un campo que está en gran desarrollo actualmente. Los sistemas hidropónicos están ampliándose en el mercado bajo la necesidad de optimizar los procesos de agricultura, utilizando mejor los recursos y disminuyendo el mal gasto de los recursos hídricos y nutritivos, al mismo tiempo que generando mejor calidad de frutas, verduras, hortalizas o plantas ornamentales en un ambiente controlado y capaz de producir durante el año completo, dentro de un invernadero, en opuesto a las producciones por temporada que se realizan con el cultivo en tierra a campo abierto. El producto se destaca por tener una interfaz de simple uso, en donde usuarios sin experiencia previa en hidroponía puedan realizar un cultivo a través de una memoria extraíble que tiene la receta de cultivo particular en su memoria y el dispositivo aplica las órdenes de la memoria.

El producto final es un controlador hidropónico automático, que tiene un puerto de memoria SD, y un interfaz sencillo que tiene dos botones de modo y un botón de selección. Este interfaz es capaz de regular el nivel de fertilizantes para cada planta dependiendo de su etapa de crecimiento, controlar una inyección de agua, y ser programada con anticipación a través de la tarjeta SD para mantener niveles deseados. Este para funcionar, acciona distintas bombas que controlan el flujo tanto de fertilizantes,

y el flujo del agua para mantener los niveles óptimos. El usuario por su parte, observa todo lo que está pasando de forma simple desde la pantalla LCD, la cual entrega información del modo actual, información de EC actual e ideal, y un verificador que indica si el sistema tiene los niveles correctos. Además, tiene un sistema de resistencia a la falta de electricidad, siendo capaz de ubicarse en el tiempo, y modular los cambios sucedidos durante el tiempo sin energía.

3.3 Diseño de Sistema Hidropónico Automático

3.3.1 Límites y Alcances de proyecto:

El proyecto se basa en realizar un sistema hidropónico automático de bajo costo, el cual mediante un medio simple introduce una receta de cultivo. Este se limita a usar el microcontrolador Arduino y sensores encontrados en el mercado. Por presupuesto el proyecto se limita al uso de sólo un prototipo para todo el proceso, para la interfaz. Por otro lado, este proyecto no intenta evolucionar un sistema actual hidropónico, sólo facilita su uso y lo estandariza. Finalmente, este proyecto desarrolla un sistema de recirculado por lo que su funcionamiento debe regular todo lo posible para mantener una reserva de agua con los parámetros establecidos previamente

3.3.2 Riesgos

Se consideraron los siguientes riesgos al diseñar y desarrollar el SHA, tomando en cuenta los casos de uso y la actividad intrínseca del sistema:

- No se tiene certeza del funcionamiento tanto de los sensores que se planeo usar, como del controlador arduino, por lo que es posible que el proyecto fracase por un mal desempeño de los mismos.
- Es posible que los inputs del usuario pueda provocar un mal funcionamiento.
- Es posible que al tratarse de un aparato que toma datos por medición de voltajes, este tenga problemas cuando se desean tomar más de un dato a la vez.
- Es posible que se prefieran otras instancias dado el mercado y el negocio actual de la agricultura, pues se basa en bajos costos iniciales de producción.
- Es posible que las mediciones tomadas sean muy variables, haciendo difícil la optimización, aún cuando se tomen los pasos necesarios para evitar las variaciones.
- Al ser un producto nuevo, es posible que los potenciales clientes no se arriesguen en la compra de un aparato sin precedentes, lo que perjudicaría el crecimiento del negocio.

Por otro lado es posible que las acciones que debe realizar el sistema no aporten real valor en las cadenas de producción de vegetales actuales por lo que no sea considerado como una solución al cultivo actualmente implementado.

3.3.3 Requerimientos

Para el desarrollo del proyecto se necesitan los componentes necesarios para el desarrollo del prototipo, es decir el microcontrolador y sus componentes básicos que serán explicados en el siguiente punto. Se estimo inicialmente que el desarrollo de un prototipo funcional no debería tomar mas de un mes para desarrollar una iteración del modelo RUP, calculando menos de 300 horas hombre de gente capacitada.

3.3.4 Factibilidad del proyecto

Dada la investigación hecha de precios en sensores y los precios de algunos sistemas hidropónicos automáticos a nivel internacional (Ver figura 15: "Comparación de Precios de Sistemas Hidropónicos Actuales" en Anexo 1: "Sistemas Hidropónicos"), se llegó a la conclusión que el proyecto es factible. Por otro lado, la implementación de este proyecto es relativamente sencilla, pues se basa en un lenguaje simple como lo es el de processing en donde existen pocos objetos interactuando y se basa en un sistema de control, por lo que se desestima problemas que tengan que ver con complejidad de la programación. Dada la investigación previa también podemos suponer que los cultivos a nivel mundial van incrementar su uso en hidroponía^{12, 13} por el poco espacio que requieren y su habilidad de no requerir suelo fértil.

3.3.5 Resumen de Análisis

A continuación se presentan de manera resumida los puntos a considerar para la realización de este proyecto:

- Piezas de construcción baratas
- Equipos actuales muy caros.
- Programación sencilla con código simple y mucha información de proyectos por comunidad arduino.
- Incremento de hidroponía por su eficiencia.
- Factibilidad de implementación por uso eficiente en zonas de escasos recursos hídricos

3.4 Construcción del Sistema

3.4.1 Visión y Análisis de Negocio

Para construir un SHA que funcione correctamente, se requiere que cumpla con algunos puntos básicos. Para que un cultivo hidropónico sea exitoso, este debe ser capaz de controlar variables ambientales como humedad, temperatura, EC y pH del caldo de cultivo, y una combinación específica de distintos fertilizantes entre sus variables más importantes. Esto se puede apreciar en diversas fuentes donde se enseña hidroponía en donde se explica que la clave viene de 2 partes: construir un sistema y luego realizar mantenciones periódicas.

El objetivo de este proyecto es automatizar en gran medida las tareas que se requieren realizar para desarrollar un cultivo hidropónico. Se sabe que hoy en día existe un alza en proyectos de desarrollo dado a que las barreras de entrada para desarrollar prototipos han bajado por la comercialización de placas de control como Arduino.

Se espera desarrollar un SHA que, por medio de una placa controladora y con ayuda de sensores de distinto tipo, monitoree variables ambientales o del entorno, por medio de distintos actuadores logre modificar dichas variables con el fin de permanecer en los rangos óptimos de dichas variables, los cuales provienen de una receta de cultivo que poseen los niveles precisos que deberían tener las variables de entorno importantes. Esto se puede ver en la figura XX abajo.

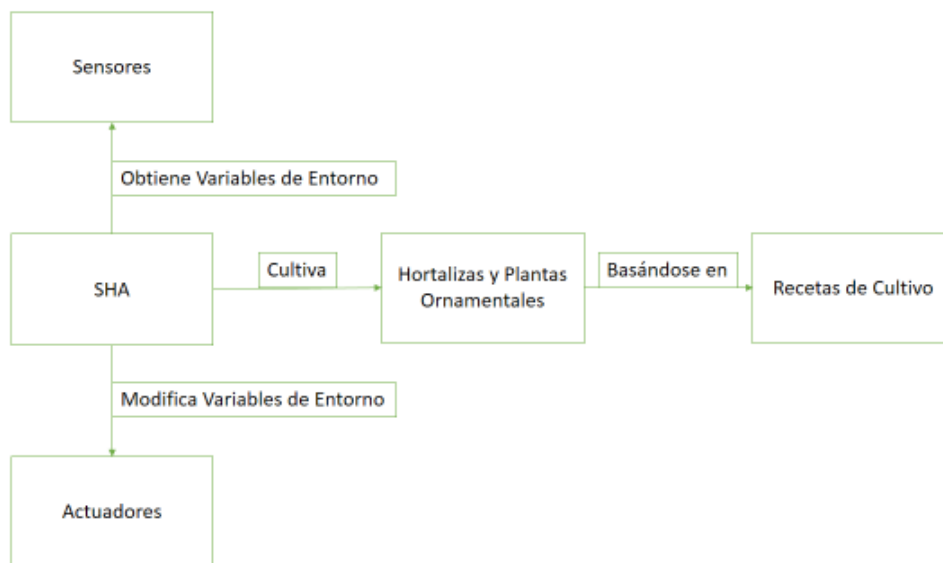


Figura 6: Modelo de Dominio
Fuente: Elaboración propia

3.4.2 Elaboración

3.4.2.1 Análisis de Dominio de Sistema:

Esta sección tiene como objetivo identificar todos los objetos presentes en el sistema propuesto, los cuales engloban todas las posibilidades de sistema. Estos se enumeran como:

- Arduino: Referente a la placa controladora y en general al sistema de control.
- Pin: Referente al nombre del puerto de conexión, tanto de controladores como de actuadores.

- Pantalla: Referente a la interfaz visual que logra la interacción entre cultivador y Arduino
- Botón: Referente al artefacto que logra la interacción entre el usuario y el Arduino.
- Relé: Referente al dispositivo que permite prender y apagar aparatos desde el controlador.
- Bomba: Artefacto actuador que está conectado a un relé. Existen muchos tipos de bomba. En este sistema en particular pueden ser bombas peristálticas, bombas de circulación de agua las cuales pueden ser para vaciar la reserva como para regar.
- Válvula Solenoide: Válvula encargada de abrir o cerrar un paso de agua. Esta en particular es esencial para el llenado automático.
- Sensor: Referente a distintos sensores involucrados en el sistema. Estos pueden ser sensores de: EC, pH, Intensidad lumínica, medidores de nivel de agua, temperatura y humedad.

Para una identificación visual de cada uno de los componentes, ver Anexo 2: "Sistemas Automatizados y Arduino".

3.4.2.2 Requerimientos del sistema

3.4.2.2.1 Requisitos funcionales

Los requerimientos funcionales se evaluarán a través de casos de uso. Estos están explicitados en formato de caja negra, en donde se habla del sistema como la entidad que logra que las actividades se desarrollen lo cual posteriormente será desarrollado.

3.4.2.2.1.1 Casos de Uso

La sección a continuación explica todas las funciones que debe ser capaz de proveer el sistema. El usuario interactuando con el dispositivo debe ser capaz de:

- Crear un cultivo: A través de una selección de modo, la acción le informa al sistema que en ese momento comienza un cultivo nuevo, el cual debe tener una receta de cultivo prevista inicialmente.
- Modificar Modalidad: El usuario debe ser capaz de ordenarle al sistema cuando debe regar, y en que situaciones el cultivo requiere mantención, la cual se hace dependiendo del caso, pero se resumen en que el usuario pueda detener el ciclo por cualquier eventualidad.

Por otro lado el sistema debe ser capaz de emular las funciones básicas, llamas modalidades, que se realizan en un cultivo hidropónico, las cuales utilizan para realizarlas en casi todos los casos tanto sensores como actuadores. Se identificaron 4 para el desarrollo de un prototipo funcional, lo cual se explica a continuación:

- El sistema debe ser capaz de decirle al sistema que no haga nada y se mantenga estático. Esto por diversas razones, incluyendo mantenimiento del sistema o frente a cualquier eventualidad.
- El sistema debe ser capaz de vaciar el contenido de agua de la reserva. Esta actividad es importante para la mantención de una reserva de agua para hidroponía, en especial si se trata de un sistema hidropónico de recirculado, pues otros sistemas similares simplemente procesan una forma de riego y riegan hasta que se acabe el agua. En este sentido el vaciado es importante pues aporta limpieza.
- El sistema debe ser capaz de dar paso a agua para que llene la reserva, teniendo especial cuidado de no sobrepasar el límite de llenado
- El sistema debe ser capaz de mantener un ciclo de regado en condiciones óptimas y debe ir rellenando con agua para satisfacer el agua utilizada y debe fertilizar extra cuando sea necesario, respetando las proporciones de fertilizantes requeridas por semana.

Luego de la identificación de requisitos funcionales, se desarrolló un diagrama de casos de uso.

Diagramas de Casos de Uso

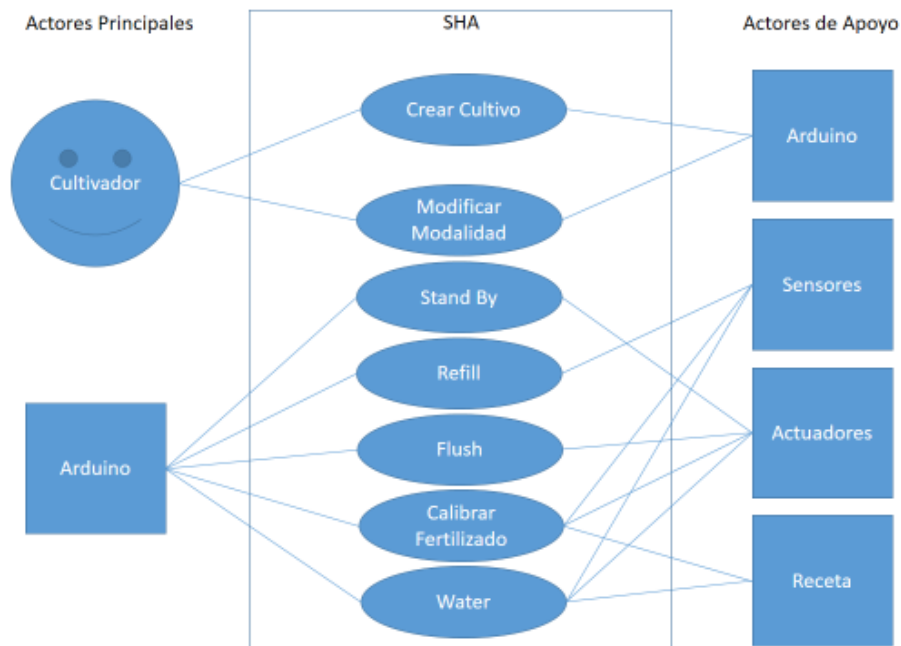


Figura 7: Diagrama Casos de Uso
Fuente: Elaboración propia

Diagramas de Secuencia:

Para entender a cabalidad de que se trata cada caso de uso, es necesario descomponerlo en los actores que se involucran, y con que requerimientos o acciones se relacionan estos al transcurrir dichos casos de uso o actividades. La finalidad es simplificar la comprensión del diagrama de casos de uso. A continuación se presentan los diagramas de secuencia para cada acción con una breve explicación

Crear Cultivo:

Este método es simple pues para crear un cultivo, el usuario debe comunicarle un método a la placa la cual cree un nuevo cultivo, lo que se hace por medio de botones. Para esto se implementan los métodos `pressCrearCultivo()` el cual corresponde a la clase de Botones y `crearCulivo()` a la clase de Arduino. A continuación su Diagrama de Secuencia

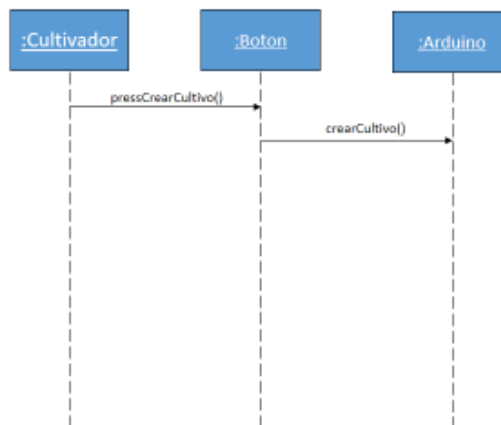


Figura 8: Diagrama de Secuencia Crear Cultivo
Fuente: Elaboración propia

Modificar Modalidad:

De forma analógica como el caso anterior, se forma un diagrama de secuencias en donde el cultivador, por medio de botones comunica al Arduino la modalidad del sistema, teniendo análogamente los métodos `pressMode()` y `setMode()`.

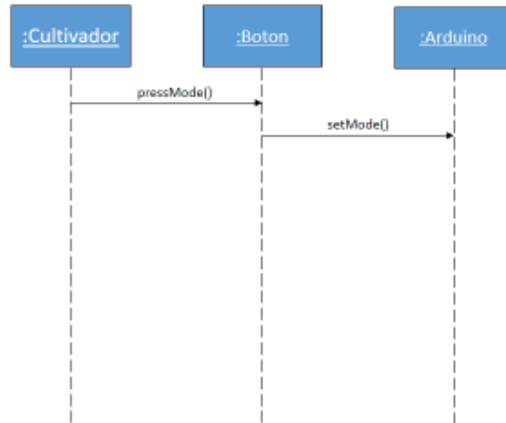


Figura 9: Diagrama de Secuencia Modificad Modalidad
Fuente: Elaboración propia

Stand By:

Este modo no requiere un Diagrama de secuencia, puesto que al seleccionar este modo, el sistema no realiza ninguna acción.

Refill:

Este diagrama de secuencias explica como el sistema rellena su reserva hasta el límite determinado. Esto se logra en primer lugar obteniendo el valor de la variable lleno la cual es un booleano que indica True si es que el flotador está activado indicando que está hasta el límite superior y False si es que el flotador no está en el límite superior, lo que indica que falta agua. Luego el controlador enciende BRefill por medio de un solenoide. Se estimó que no era necesario hacer una clase para el relé, sino que para el actuador pues lo anterior podría ser más confuso para el lector. Finalmente cuando el flotador marca el llenado del sistema, el sistema apaga la bomba y termina esta secuencia.

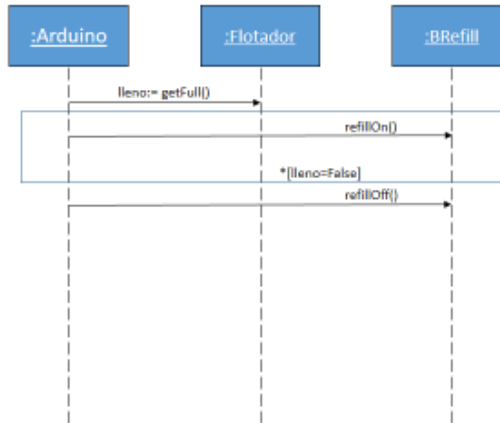


Figura 10: Diagrama de Secuencia Refill
Fuente: Elaboración propia

Flush:

Esta secuencia consiste en activar la bomba de vaciado de la reserva durante 5 minutos. Este número se toma pues se estima que es lo máximo que una bomba de agua pequeña podría demorarse para vaciar un estanque de 20 litros (el estanque inicial propuesto para el prototipo). Para esto se enciende la bomba y se genera una variable auxTime la cual marca el tiempo exacto en cuanto se prende la bomba. Luego existe un loop el cual sólo se rompe cuando han pasado 300 segundos desde auxTime.

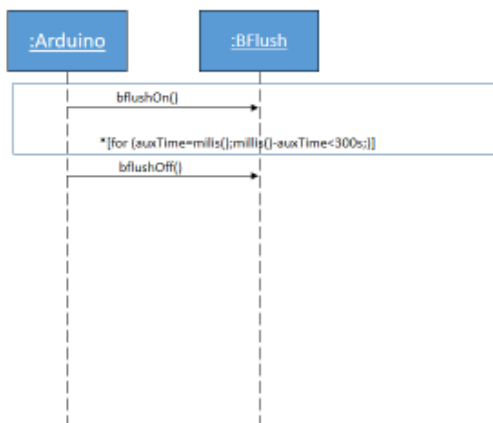


Figura 11: Diagrama de Secuencia Flush
Fuente: Elaboración propia

Calibrar Fertilizado:

Este proceso da inicio cuando el arduino intenta obtener el tEC, o EC objetivo el cual se obtiene de la receta. Primero activa la bomba de recirculado con el fin de mezclar el fertilizante y luego llama a la línea de receta correspondiente a la fecha para sacar la información sobre las proporciones de fertilizado o tEC. Luego toma una medición del sensor de EC de nombre EC1. Posteriormente inicia un proceso llamado peristalticasCalibrado() donde se les ordena a las bombas peristálticas iniciar una secuencia que adiciona el fertilizante según la receta pero en un lapsus breve de tiempo, lo cual añade una cantidad de fertilizante pequeña a la reserva de agua. Luego entra en un período de latencia hasta que se mezcle el volumen completo y se toma una segunda medición de EC llamada EC2. Con esto puede tener una referencia del aumento y calibra sus bombas con el valor fertiMulti que hace referencia al multiplicador que se le aplica a esa pequeña dosis inicial tal que EC se incremente en unidades de 25 microsiemens (μS). El objetivo de realizar la acción de esta manera es colocar un número robusto y no muy pequeño (existe un trade off entre la cantidad fertilizante añadida en proporción y que tan rápido logra fertilizar el volumen completo, por lo que incrementos de 25 unidades se consideró aceptable).

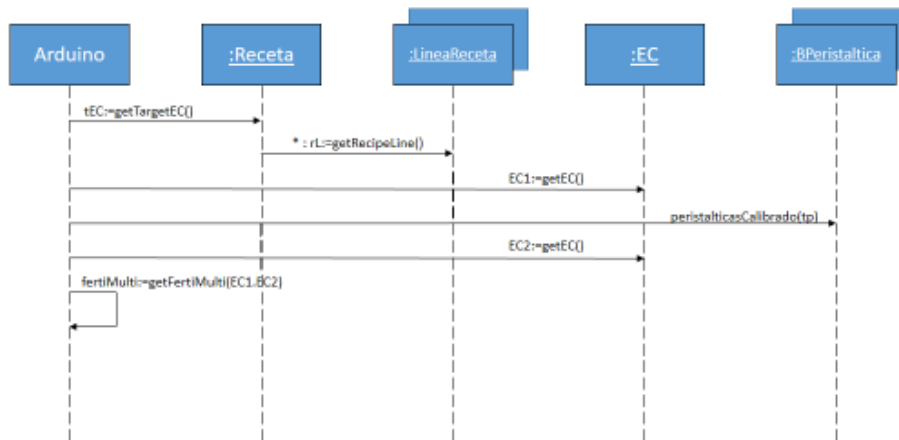


Figura 12: Diagrama de Secuencia Calibrar Fertilizado
Fuente: Elaboración propia

Water:

Este proceso activa la bomba de recirculado y obtiene de la receta tanto la proporción (tP) como la EC objetivo (tEC). Luego toma una medición de EC. Posteriormente, activa las bombas de fertilizado teniendo como input fertiMulti obtenido durante el calibrado y entra en una breve latencia para que recircule y se mezcle. A continuación toma una medición nueva de EC y la compara con tEC. Si la diferencia entre tEC y EC es mayor a 50 microsiemens, este proceso se repite hasta lograr que la diferencia sea menor a 50 μS . Finalmente da fin al ciclo de recirculado al apagar la bomba encargada de esta acción.

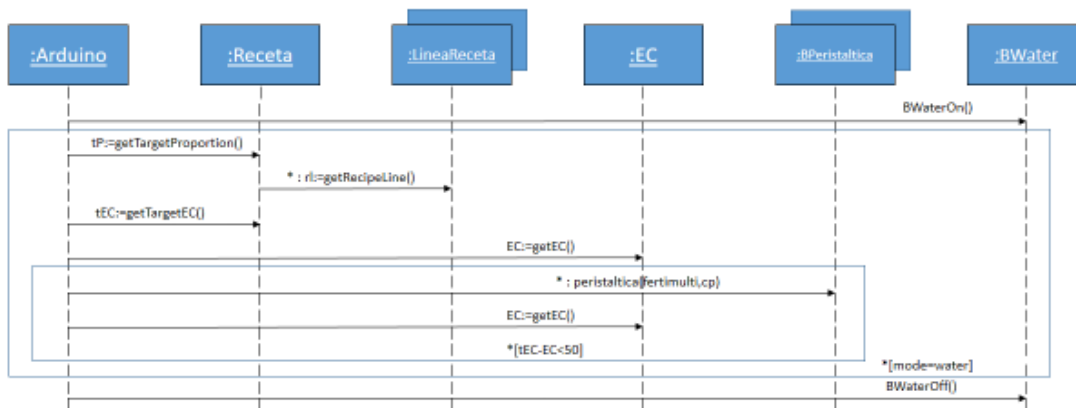


Figura 13: Diagrama de Secuencia Water
Fuente: Elaboración propia

3.4.2.2.1.2 Diagrama de Clases

Con los principales diagramas de Secuencias explicados, a continuación se presenta el diagrama de clases con sus respectivos métodos

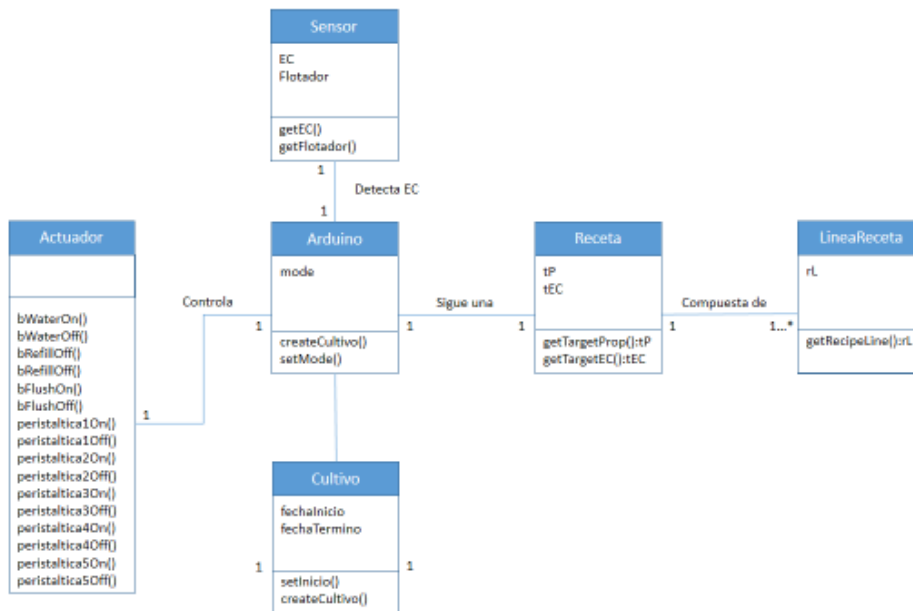


Figura 14: Diagrama de Clases
Fuente: Elaboración propia

3.4.2.2 Requisitos No Funcionales

Uno de los principales requisitos no funcionales del sistema es que el microcontrolador sea capaz de sobrevivir a cortes de luz, reincorporando el ciclo de la receta desde el momento en que se detuvo al corte de luz

En caso de un reseteo del sistema, el controlador debe ser capaz de reconocer donde quedó en su cultivo y seguir con el orden cíclico, reduciendo al mínimo las fluctuaciones en los valores, y logrando continuar desde el momento de la ruptura.

Finalmente, se propone que el sistema debe tener un índice de refresco de pantalla inferior a 1 décima.

3.4.2.3 Riesgos Revisados

Los riesgos en esta etapa aumentaron al notar que al momento de diseñar el sistema, no se consideró la necesidad de medir en que momento del ciclo el sistema tendría la capacidad de vaciar la reserva de agua, lo que impacta directamente en el proceso de Flush. Este riesgo adicional se solucionó de la siguiente manera: se determinó un número que fuera suficiente para vaciar una reserva de 20 litros. Una bomba de agua convencional es capaz de mover dicha cantidad en 5 minutos, considerando un flujo de 240 litros por hora, lo cual basta para la gran mayoría que existe en el mercado. (la más pequeña existente en el mercado puede mover 300 litros hora). Esta no es una solución permanente al problema, pues existe la posibilidad que exista un imprevisto que provoque el mal vaciado de la reserva. Por otro lado, existe la posibilidad que la bomba esté encendida por un tiempo extenso con un volumen pequeño, lo que es considerado como importante al momento de solucionarlo, puesto que podría provocar que la bomba entre en cortocircuito, o se funda. Se debe llegar a una nueva solución. Los flotadores actuales no tienen la capacidad de mantenerse sumergidos y al mismo tiempo funcionar correctamente, por lo que su utilidad sólo se aplica para el llenado. Los demás riesgos siguen estando presentes.

3.4.3 Construcción

El propósito de esta sección es desarrollar un producto de forma simple y ordenada, el cual debe estar listo para entregar al usuario. Para apoyar a este ítem, se explicará en su totalidad el algoritmo del código creado para entender cómo se implementó el proyecto. Esto se considera una nueva iteración y modifica algunos de los diagramas mostrados anteriormente. Por otro lado, la construcción define el producto final a testear en pruebas de terreno y con ello, definir las instrucciones por las cuales el usuario hará funcionar el producto. Para apoyar esto, se mostrarán secciones del código importantes para explicar los métodos.

3.4.3.1 Hardware necesario

A continuación se explican las distintas piezas utilizadas y una breve explicación de por qué se escogieron.

- **Placa Arduino Mega:** Se escoge esta placa pues es expansible en muchas formas, con un precio accesible, teniendo más de 16 pines digitales los cuales pueden introducir y controlar muchas variables distintas, sin considerar los 54 pines digitales los cuales pueden servir para controlar aparatos de todo tipo, como pantallas, botoneras, relés, etc.
- **Módulo SD:** Este es necesario para conectar el microcontrolador a una tarjeta SD, con lo que se conforma el mecanismo de memoria del dispositivo, necesario para guardar itinerarios, recetas de cultivo, etc.
- **SD Card:** Esta representa la memoria del sistema, pues el microcontrolador en sí no posee mucha memoria y su uso es limitado, pudiendo generar problemas.
- **Válvulas solenoides:** Estas son necesarias para dar el paso de agua desde la red central hacia la reserva de agua.
- **Módulo RTC:** este es necesario para que el microcontrolador tenga una hora precisa, y esta sea a prueba de cortes de corriente.
- **Módulo de EC:** este es necesario para que el microcontrolador tenga los parámetros precisos de las condiciones del caldo de cultivo, de tal forma que pueda modificarlos a elección.
- **Módulo de 8 Relés:** esto es necesario y suficiente para controlar 3 bombas peristálticas (sistema de fertilizantes), sistemas de llenado y vaciado de reserva de agua, y cualquier otro tipo de control extra que se desee añadir en el futuro.
- **Bombas peristálticas:** Son bombas que utilizan un movimiento peristáltico para mover fluidos. Por la naturaleza del mismo, la parte móvil de la misma jamás está en contacto con la parte líquida, y su control de flujos bajos de líquidos las hacen ideales para el flujo de fertilizantes y reguladores de pH.
- **Bombas de agua:** Estas bombas son necesarias para mover una cantidad pequeña de líquido a una altura no superior a los 70 cm. Estas se utilizan pues son de bajo costo y son lo justo y necesario para hacer andar el sistema.
- **Protoboard:** Este es necesario para unir todas las conexiones con sus fuentes de poder de forma ordenada y simple.
- **Cables:** Necesarios para completar los circuitos.
- **Carcaza:** Caja plástica para contener todos los elementos.

3.4.3.2 Modificaciones al Diseño

Durante el desarrollo se decidió que la forma de comenzar un cultivo fuese avisada por la ausencia de un archivo de inicio en la Tarjeta SD. La raíz de esto es porque la otra opción era ingresar la fecha a través de la interfaz, lo que se hacía engorroso y complicado. El cambio propuesto es el siguiente: cuando el usuario introduce la tarjeta SD por primera vez, previo a encender el sistema, esta sólo debe tener el archivo "RECETA.txt", sin el archivo "FECHA.txt", indicándole al aparato que no existen cultivos previos, puesto que no hay una fecha de inicio, creando un archivo FECHA.txt al momento de iniciar el programa.

3.4.3.3 Interfaz Usuario-Sistema

Pantalla LCD:

La pantalla se utiliza para mostrar las variables más importantes del control automático: la EC de solución en reserva, la EC objetivo, y el modo en que se encuentra el sistema. A través de los botones "izquierda", "derecha" y "select" el usuario puede cambiar el modo y el template en pantalla. Cada cambio de modo gatilla el comando stateSave() que tiene como propósito guardar el cambio realizado al estado de cultivo, para no interrumpir el ciclo en caso de corte de energía.

Botones:

En relación a los botones, estos modifican la variable lcd_state[] la cual indica al arduino en qué modo del aparato está, y que pantalla debe mostrar. La explicación del código se encuentra en sección algoritmo. La interfaz de botones es la siguiente:

Para efectos de este prototipo, sólo funcionan los botones Select, Left y Right. Select por un lado permite cambiar de template mientras que left y right sólo funcionan en la pantalla principal para que el usuario cambie de modo el aparato. La explicación detallada del funcionamiento de los botones está en Algoritmo y Código sección boton().

3.4.3.4 Algoritmo y Código

El algoritmo realizado se basa en la forma de programación de arduino, que al tratarse de un controlador, tiene sus propias clases y objetos los cuales simplemente son llamados para funcionar en un loop. El programa como se sabe tiene una fase de inicialización, en donde el sistema recorre sólo una vez, y luego tiene un loop el cual funciona hasta que el sistema se detenga, ya sea por acción del usuario, o por acciones externas al aparato (ej. Corte de luz). En este ítem se expondrán los métodos utilizados y durante dicha explicación, las variables utilizadas por dichos métodos.

El algoritmo se explicará en términos de Setup y Loop pues son 2 fases críticas en la programación.

3.4.3.4.1 Setup

Inicialización de Variables Utilizadas:

En esta parte, se declaran todas las librerías necesarias, archivos que se van a utilizar, LCD, RTC, SD, variables públicas, los pins utilizados. Las librerías utilizadas fueron: wire.h y SPI.h para comunicación entre dispositivo; RTC_lib.h para función de RTC; LiquidCrystal.h para el LCD; y SD.h para la memoria SD

¿"State.txt" existe en SD?:

State.txt es un archivo que contiene 2 números enteros, los cuales indican el valor de lcd_state[0] y lcd_state[1], variables utilizadas para actualizar la pantalla. Este paso pregunta a Tarjeta SD si existe "State.txt". Si existe, cambia los valores iniciales de lcd_state[0] y lcd_state[1]. De no existir, estos valores están predefinidos en 0 lo cual muestra la pantalla inicial en modo Stand By. La finalidad de este proceso es guardar en la memoria SD los cambios de estado, de tal forma que si existe un RESET o corte de corriente, siempre vuelve al estado en el que estaba previo al corte, logrando que el sistema sea robusto. A continuación el código:

```
if (SD.exists("STATE.txt")) {  
  
    STATE = SD.open("STATE.txt", FILE_READ);  
    lcd_state[0] = int(STATE.read()-48);  
    lcd_state[1] = int(STATE.read()-48);  
    STATE.close();  
}
```

¿"FECHA.txt" existe en SD?:

"FECHA.txt" es un archivo que indica la fecha de inicio del cultivo, y su ausencia indica que el cultivo es nuevo. Este se compone de 8 números enteros indicando el formato de fecha DDMMAAAA.

Se decidió durante la programación que el inicio de cultivo sería determinado por la introducción de una tarjeta SD que contenga sólo el archivo "RECETA.txt", ordenando al sistema dar inicio a un nuevo ciclo, creando el archivo "FECHA.txt" en el proceso. La codificación funciona de la siguiente manera: El sistema pregunta si existe FECHA.txt en la memoria SD, y de no existir, pregunta al reloj RTC la fecha y la guarda en el formato mencionado anteriormente, haciendo diferencia entre días y meses con 1 o 2 dígitos, anteponiendo el 0 cuando corresponda a un solo dígito. Esto se puede ver en el siguiente código:

```

if (!SD.exists("FECHA.txt")){
  DateTime now = RTC.now();
  FECHA = SD.open("FECHA.txt", FILE_WRITE);
  FECHA.print(now.day, DEC);
  if (now.day() > 9) {
    FECHA.print(now.day(), DEC);
  }
  else {
    FECHA.print('0');
    lcd.print(now.day(), DEC);
  }
  if (now.month() > 9) {
    lcd.print(now.month(), DEC);
  }
  else {
    lcd.print('0');
    lcd.print(now.month(), DEC);
  }
  lcd.print(now.year(), DEC);
  FECHA.close();
}

```

Setear valores de fecha[]

Luego del suceso anterior debe existir un archivo "FECHA.txt" en la tarjeta SD. Posterior a esto, se abre el archivo FECHA.txt y se guarda la variable con el formato DDMMAAAA en la memoria de arduino en variable int fecha[], siendo fecha[0] el día, fecha[1] el mes y fecha[2] el año, lo cual permite fácil acceso del sistema a la variable sin tener que recurrir a SD cada vez, pues la variable es constante durante todo el cultivo. A continuación se verá dicho código:

```

FECHA = SD.open("FECHA.txt", FILE_READ); //Guardo la variable fecha en fecha[]
fecha[0]= int(FECHA.read()-48)*10+int(FECHA.read()-48);
fecha[1]= int(FECHA.read()-48)*10+int(FECHA.read()-48);
fecha[2]= int(FECHA.read()-48)*1000+int(FECHA.read()-48)*100+int(FECHA.read()-48)*10+int(FECHA.read()-48);

```

Al realizar este paso, se termina el SETUP. ** inicializa variables necesarias para el programa, seteando la variable lcd_state[] en la información que está en "STATE.txt".

3.4.3.4.2 Loop

El loop en su funcionamiento promedio llama a 5 funciones las cuales van modificando su comportamiento dependiendo del valor de las variables definidas. Estos son:

- setProportion(): setea las proporciones y los niveles de EC acordes al tiempo de vida del cultivo
- getEC(): obtiene el valor del EC presente en el caldo de cultivo

- `boton()`: obtiene los valores de los botones presionados y modifica variables del sistema según sea el caso
- `pantalla()`: modifica la pantalla lcd dependiendo de las variables del sistema
- `tarea()`: realiza las tareas que se dictan dependiendo del modo en el que está seteado el dispositivo.

`setProportion()`

Este método tiene por objetivo actualizar la proporción de fertilizantes utilizados, lo que requiere posibles modificaciones cada semana. Este utiliza como inputs `RECETA.txt`, `FECHA.txt` y la hora actual. “`RECETA.txt`” corresponde a un archivo de texto guardado en SD que tiene líneas con 9 números enteros. Los primeros 5 números son correspondientes a la proporción de los 5 fertilizantes y los últimos 4 corresponden a los valores óptimos de EC, los cuales son guardados en el sistema en las variables `ferti_proporcion[]` y `tEC`.

En primer lugar se decide por motivos de rapidez del ciclo que `setProportion()` debe realizarse en el primer loop, para definir la proporción inicial, y luego debe llamarse una vez cada 1 hora, por motivos de seguridad de que la receta esté actualizada a lo más 1 hora después del paso de la semana, debido a que las recetas se modifican a cada semana dependiendo del ciclo de la planta. Para lograr lo anterior, se define una variable boolean `primeraproporcion`, la cual se declara verdadera en la inicialización de variables, y al final del primer ciclo de `setProportion()` se setea falsa, sin existencia de una instancia que la cambie a verdadero en el loop, por lo cual dicha variable será falsa si ya ocurrió el primer loop.

Por otro lado, se genera una variable de tiempo de nombre `timer_receta` que guarda el tiempo en el que ocurrió un ciclo de `setProportion()`, con el fin de poder medir la cantidad de tiempo transcurrido entre ciclos. Se impone la condicional `millis()-timer_receta > 3600000` lo que indica que deben pasar 1 hora desde el último ciclo para que ocurra de nuevo. Se define esta función `millis()` como medida en milisegundos.

Para guardar las proporciones de los fertilizantes se genera un vector con 5 dimensiones de nombre `fertiproporcion[]`. Cabe mencionar que la resta de 48 en el ejemplo corresponde a la lectura de la receta en la SD, la cual entra al sistema en formato ASCII y para formar el número entero se le resta 48. (anexo tabla ascii y demostrar que -48 produce eso). Luego el sistema guarda `tEC` que es un número entero entre 0 y 5000 que determina la EC ideal para la receta. En esta parte del código simplemente lee cada carácter y lo multiplica por 1000 para formar los miles, por 100 para las centenas, 10 para la decena y las unidades sin ningún multiplicador, pues se lee de a un byte.

Finalmente, este loop de seteo de `fertiproporcion[]` y `tEC` se repite cuantas semanas después del inicio hayan pasado, información que se obtiene con el método `semanasDespuesInicio()`, por lo que existe un loop externo que se repite cuantas semanas hayan pasado, con el fin de dejar guardada las variables que correspondan por

semana. Dado a que RECETA.txt tiene 9 caracteres por línea y en cada iteración se leen 9 caracteres, cada iteración lee la siguiente línea.

A continuación el código:

```
void setProportion() {

    if (millis() - timer_receta > 3600000 || primeraproporcion ) {
        FECHA = SD.open("RECETA.txt", FILE_READ);

        for(int i=0; i<semanasDespuesInicio(); i++){
            for(int i=0; i<5; i++){
                ferti_proporcion[i]=int(RECETA.read()-48);
            }
            tEC=int(RECETA.read()-48)*1000+int(RECETA.read()-48)*100+int(RECETA.read()-48)*10+int(RECETA.read()-48);
        }
        timer_receta=millis();
        RECETA.close();
    }
}
```

Para calcular las semanas después del inicio se implementó un método llamado `semanasDespuesInicio()` el cual inicializa un objeto `DateTime` que contiene una fecha, y tiene prestaciones por la librería "RTCLib.h". Se genera `dtinicio`, variable con la fecha de inicio del cultivo, y `now`, variable que contiene la fecha actual. El método genera la variable `semanas` usando la función `unixtime()`, que saca la cantidad de segundos transcurridos entre 1970 y la fecha en cuestión, aplica dicho método en las 2 fechas y las resta, lo que da la cantidad de segundos totales de vida del cultivo. Estos los divide por 604800 y se le aplica la función `int()` lo cual da exactamente la cantidad de semanas transcurrida entre el inicio y la fecha actual, valor utilizado por el método `setProportion()`. A continuación el código:

```
int semanasDespuesInicio() {

    DateTime dtinicio (fecha[2], fecha[1], fecha[0], 0, 0, 0);
    DateTime now = RTC.now();
    int semanas = int((now.unixtime() - dtinicio.unixtime())/604800L);
    return semanas;
}
```

`getEC()`

El loop sigue llamando a `getEC()`, el cual obtiene el valor actual de EC presente en el cultivo. Este toma muestras de voltaje provenientes del sensor de EC, cada intervalos de tiempo fijo cuyo valor se define en "intervalosEntreMedidas", y las guarda en un arreglo llamado `ECArray[]` el cual tiene una cantidad de muestras determinada fija, llamada `nMuestra`. Para este código se inicializó tanto `samplingTime` en 40 ms y `nMuestra` en 40 muestras. Cabe destacar que `ECArray[]` después de llenar la matriz, siempre tiene en sus datos los valores de las últimas 40 medidas tomadas, por la estructura del programa.

Cada vez que llama a la función, este registra la hora en la que corrió con el método `millis()`, el cual saca la cuenta de milisegundos desde que el controlador está encendido, llamando a la variable que guarda dicho tiempo "samplingTime". Luego para entrar en

este método, se pide como condición que `millis() - samplingTime` sea mayor a `intervaloEntreMedidas`. Esto se hace con el fin de espaciar la concentración de toma de muestras, permitiendo que el loop no se estanque a la hora de tomar datos, lo cual contribuye con el requisito no funcional de refresco de pantalla, y también contribuye a tomar bien el uso de los botones. Para finalizar, se utiliza otro método que se explicará en el siguiente ítem que tiene relación con obtener el promedio de los arreglos y guarda dicho valor en la variable `voltaje`. Finalmente, la EC con unidad de medida en $\mu\text{S/cm}$ se calcula con la siguiente fórmula:

$$EC = V_{promedio} * \frac{5000}{1023}$$

La fórmula anterior se explica en que los valores obtenidos por la entrada análoga pueden ser entre 0 y 1023, los cuales indican que tan electroconductiva es el agua medida, y también se sabe por fabricante que los valores mínimos y máximos de la probeta son 0 y 5000, por lo que la fórmula sale por proporcionalidad directa entre los valores extremos del voltaje y los valores extremos de la EC.

V promedio o voltaje promedio se saca mediante la función `averageArray` la cual será descrita en el siguiente punto. Para finalizar, este método por medio de una condicional reinicia en valor 0 el índice `ECArrayIndex` utilizado para llenar la matriz cuando dicho índice alcanza su valor máximo. Es importante notar que el cálculo de la EC se guarda en la variable de nombre "total". A continuación se expone el código de `getEC()`:

```
void getEC() { //nos permite medir EC

  if (millis() - samplingTime > intervaloEntreMedidas) //el loop entra si tiempo actual - tiempo de ultima medida es mayor a intervalo
  {
    if(ECArrayIndex == nMuestra){ //hasta llenar hasta limite y despues el indice cambia
      ECArrayIndex=0;
    }
    ECArray[ECArrayIndex++]= analogRead(EC); //En ECArray, se guardan los voltajes que dicen la EC
    total = averageArray(ECArray, nMuestra) * (5000.0 / 1023.0);
    samplingTime = millis();
  }
}
```

`averageArray(arr,number)`

Este método, que usa como input un arreglo y su cantidad de datos, y es utilizado por `getEC`, nos permite obtener el promedio de un arreglo disminuyendo varianza entre datos obteniendo el mínimo y el máximo del arreglo por la posibilidad de existencia de datos extremos. Esto se hace de la siguiente forma:

Se definen los primeros 2 números del arreglo como min y max. Se van revisando ordenadamente los números que siguen del arreglo. Si el número que sigue en el arreglo es menor al min, se sabe que min no es el menor y por tanto se va a una suma de variables la cual se define como "amount". Por otro lado si es mayor al max, se sabe que max no es el máximo y max se suma a la variable amount. Si es que el número está entremedio del min y max, se suma dicho número a amount. Esto se sigue hasta completar todo el arreglo y se devuelve el promedio de la suma de las variables sin el min y el max. Esto se ve en el siguiente código:

```

double avergeArray(int* arr, int number){ // metodo pal promedio de un arreglo el cua:

    int i;
    int max,min;
    double avg;
    long amount=0;

    if(arr[0]<arr[1]){
        min = arr[0];max=arr[1];
    }
    else{
        min=arr[1];max=arr[0];
    }
    for(i=2;i<number;i++){
        if(arr[i]<min){
            amount+=min;        //arr<min
            min=arr[i];
        }
        else {
            if(arr[i]>max){
                amount+=max;    //arr>max
                max=arr[i];
            }
            else{
                amount+=arr[i]; //min<=arr<=max
            }
        }
    }
    avg = (double) amount/ (number-2);
    return avg;
}

```

En resumen, el método getEC utiliza el método averageArray para obtener un promedio que elimina los valores extremos para su cálculo, lo hace de forma pausada cada 40 ms, utiliza 40 valores para obtener su valor y finalmente hace la conversión para que el valor promedio de voltajes sea en μS , unidad de medida de la EC, lo cual es guardado en la variable total que es utilizada para fertilizar, así como para presentar en pantalla.

boton()

El loop continúa llamando al método botón. Este se encarga de actualizar en el sistema cualquier botón presionado, una vez por presión de boton, y hace los cambios respectivos en el sistema para reaccione de forma correcta.

Para su implementación se hizo uso de un código proporcionado por los fabricantes DFROBOT¹⁴ el cual ejemplifica el uso de los botones. En este, recibe el valor del voltaje entregado al pin A0 el cual utiliza el shield e interpreta su valor, entregando como valor todos los posibles valores de los botones y la opción de que ningún botón esté presionado. A continuación dicho código:

```

int read_LCD_buttons() { // metodo para leer botones
  adc_key_in = analogRead(0);      // valor analogo que se usa para los botones 0, 144, 329, 504, 741
  if (adc_key_in > 1000) return btnNONE;
  if (adc_key_in < 50)   return btnRIGHT;
  if (adc_key_in < 250)  return btnUP;
  if (adc_key_in < 450)  return btnDOWN;
  if (adc_key_in < 650)  return btnLEFT;
  if (adc_key_in < 850)  return btnSELECT;
  return btnNONE; // when all others fail, return this...
}

```

Por otro lado, otro problema importante de los botones es su repetición de actividades: como arduino trabaja con loops, las acciones que desencadenaban el uso de un botón ocurrían mientras se mantuviera el botón presionado, lo cual dificultaba su uso. Por esto, en cada loop se utiliza una variable auxiliar para guardar el botón del loop anterior y luego se actualiza el valor nuevo, a las cuales se nombró `lcd_key` y `lcd_key2`, pudiendo entonces detectar variaciones en la presión de los botones, las cuales son condición para todas las acciones desencadenadas por un botón. Se desarrolló una función llamada `cambioBoton()`, el cual resulta verdadero si ocurrió un cambio en los botones entre un loop y el anterior. A continuación el código:

```

boolean cambioBoton(){ //detecta cambios en botones
  if (lcd_key !=lcd_key2) return true;
  else return false;
}

```

Esta función `boton()` permite realizar acciones sólo cuando `cambioBoton()` sea verdadero, de lo contrario, estaría repitiendo la acción de apretar el boton mientras este esté presionado.

Por otro lado, los botones en relación al código modifican la variable `lcd_state[]` y la variable `printed`. Los botones de derecha e izquierda simbolizan la coordenada x modifican a `lcd_state[0]`, hacia arriba y abajo simbolizan la coordenada y o `lcd_state[1]` y finalmente los frames son regulados por `lcd_state[2]`, mientras que `printed` se relaciona con la actualización de la pantalla cuando se ha accionado un botón.

Control de cambio de frames:

Para cambiar el frame se debe presionar select, y debe comenzar de nuevo cuando se acaben los frames. Para esto se definieron 2 condicionales para la actualización de la variable `lcd_state[2]` que se relaciona con el frame actual. Si la frame que se está viendo no es la última, que siga avanzando en valor `lcd_state[2]`, pero si se está en el último frame, que `lcd_state[2]` tome valor 0 en el siguiente cambio. Cabe destacar que entre cambio de frames, se llama al comando `lcd.clear()` para dejar la pantalla en blanco. Cada vez que ocurre, cambia. A continuación el código:


```

if ( ( lcd_key == btnSELECT ) && cambioBoton() ) { /*si apretas SELECT, lcd_state[2] suma 1 cambiando la pantalla a mostrar,
                                                    si llegamos a la ultima pantalla y apretamos select nuevamente,
                                                    volvemos a la primera pantalla*/

if ( lcd_state[2] < (nFrames - 1) ) {

    lcd_state[2] +=1;

}

else if ( lcd_state[2] == (nFrames - 1) ) {

    lcd_state[2] = 0;
}

printed = false;          //variable printed tiene por objetivo estar seteado en falso cuando ocurre algo que no se ha impreso
lcd.clear();

}

```

Dado que sólo el frame 1 tiene que ayudar a escoger al cultivador el modo del dispositivo, teniendo 4 posibilidades de modo, se determinó que la coordenada x determinaría los modos, por lo que se definió que lcd_state[0] debe poder tener los valores 0,1,2,3. Se determinó que para controlar esto, si se llegaba a un extremo no se pudiera seguir avanzando. La lógica es similar a la anterior, salvo que aquí cuando se presiona un botón, también se actualiza con el tiempo actual la variable timer_cal, la cual utiliza el modo Water, asigna valor true a cambioModo(), un boolean que se utiliza para hacer un retraso entre el cambio de modo y su activación. Cada vez que se presiona el botón de “left” o “right”, este llama la función saveState(), el cual funciona de la siguiente manera: primero revisa en la tarjeta SD si existe el archivo STATE.txt. De existir, remueve el archivo de la memoria SD. Luego crea otro archivo con el mismo nombre e imprime la selección como lcd_state0 y lcd_state1.

```

void saveState() { //metodo que guarda en sd state

if(SD.exists("STATE.txt")){
    SD.remove("STATE.txt");}
STATE = SD.open("STATE.txt", FILE_WRITE);
STATE.print(lcd_state[0]);
STATE.print(lcd_state[1]);
STATE.close();
}

```

A continuación el código para lo que ocurre cuando se presiona right, y luego cuando se presiona left:

```

else if ( (lcd_key == btnRIGHT) && cambioBoton() ) { //INICIO DE RIGHT

if ( (lcd_state[2] == 0) && (lcd_state[0] < 3) ) { //si se esta en la pantalla de inicio, se aprieta a la derecha

    lcd_state[0] +=1; // cambiamos modo
    if(lcd_state[0] == 2){
        timer_cal= millis();
    }

    saveState(); // grabamos en la sd
    printed = false; // dado q cambiamos de modo, refrescamos la pantalla
    cambioModo=true; // señalamos un cambio de modo para el loop
    timer_modo=millis(); // iniciamos timer para ejecutar el modo elegido

}

} //FIN DE RIGHT

```

```

else if ((lcd_key == btnLEFT) && cambioBoton()){ //INICIO DE LEFT

    if((lcd_state[2] == 0 ) && (lcd_state[0]>0)){

        lcd_state[0]-=1;
        if(lcd_state[0]== 2){
            timer_cal= millis();

        }
        saveState();
        printed = false;
        cambioModo=true;
        timer_modos=millis();

    }

} //FIN LEFT

```

pantalla()

Luego se llama al método de pantalla, el cual tiene por objetivo actualizar la pantalla cuando ocurren cambios en el sistema. Dichos cambios son guardados por la variable `printed`, la cual está presente en el método `Boton()`, y se setea en falso cada vez que ocurre un cambio hecho por los botones, para luego una vez impreso, setea `printed` en true. Por otro lado, este método imprime uno de los 3 frames, lo cual se decide viendo el valor `lcd_state[]`.

Para explicar el funcionamiento mejor, separaremos por los 3 frames:

Frame 1:

El primer frame es la pantalla de inicio en donde se obtiene información relevante del cultivo, como EC actual, EC Target, un indicador si el cultivo está en rangos aceptables de EC, y finalmente el modo en el cual está el sistema. El método detecta que se encuentra en la pantalla de inicio pues la variable `lcd_state[2] = 0`.

Para esto se determinó que en la primera línea de la pantalla se muestra el modo, colocando en el inicio "MODE" lo cual utiliza 5 caracteres, y luego se determinó que los 9 caracteres siguientes iban a corresponder al estado. Para esto, se imprime la variable `estado[]`, la cual se compone de un arreglo que tiene los 4 modos, como se puede ver a continuación:

```
char* estado[4] = {"Stand By ", "Refill  ", "Water  ", "Flush  "};
```

Se tuvo especial cuidado de dejar dichos nombres con 9 caracteres pues de no ser así, no actualiza la pantalla y pueden quedar caracteres no deseados. Por otro lado, la variable `estado[]` se dejó condicional a `lcd_state[0]` pues esta justamente indica el estado actual, por lo que se imprime `estado[lcd_state[0]]`. Luego se imprime un espacio extra para finalmente dejar los últimos 2 caracteres de la línea para un indicador que dice la palabra "OK" si el cultivo está en rangos aceptables y "--" de no estar en rangos aceptables, lo cual se logra con el método `printEC()` el cual se ve más adelante.

Para la segunda fila del frame 1 se imprime en primer lugar el string "EC " el cual usa 3 caracteres, dejando un espacio vacío entre la palabra EC y el valor de la variable total, el cual es impreso por el método printEC() el cual se ve más adelante.

Luego se imprime en desde el caracter 8 de la línea 2 la palabra AIM que indica a donde se quiere llegar con la EC, para finalmente dejar en los últimos 4 caracteres los valores deseables de EC lo cual proviene de la variable tEC. Para asegurar que tEC utilice 4 espacios, se le compara con 1000, 100 y 10 y según eso agrega espacios. A continuación el código:

```

if(lcd_state[2] == 0){ // pantalla inicio
  print_EC(); //imprime la variable EC calculada en la 2da linea de la pantalla en el espacio 3 cada 1 segundo
  if(printed==false){ //si hay algo no impreso lo imprime
    lcd.setCursor(12,1);
    if(tEC < 1000) {
      lcd.print(' ');
      if(tEC < 100) {
        lcd.print(' ');
        if(tEC < 10) {
          lcd.print(' ');
        }
      }
    }
    lcd.print(tEC);
    lcd.setCursor(5,0);
    lcd.print(estado[lcd_state[0]]);
    lcd.setCursor(7,1);
    lcd.print(" AIM ");
    lcd.setCursor(0,0);
    lcd.print("MODE ");
    lcd.setCursor(0,1);
    lcd.print("EC ");
    printed == true;
  }
} //fin pantalla inicio

```

print_EC()

Este método en primer lugar ocurre cada un segundo pues parte con una condicional que compara el tiempo actual con el tiempo de la última vez que se imprimió al cual se le llama printTime y se pide que millis() - printTime > 1000. Luego, utiliza la variable total y la separa por miles, centenas, decenas y unidades y tiene especial cuidado si los valores son menores a 1000, poniendo un espacio para que se completen los 4 caracteres y no ocurran errores de impresión, imprimiendo desde el caracter 4 de la segunda fila haciendo uso del método setCursor().

Luego el método escribe desde el carácter 15 e imprime OK si es que se cumple que tEC - total <= 50 y que tEC - total >= 50 lo que es equivalente a decir que para que diga OK, total = tEC +-50. De lo contrario, escribe "--". A continuación el código:

```

void print_EC(){

if (millis() - printTime > 1000) { //printTime toma el tiempo desde que corrio esto la ultima vez

    miles = int (total/1000);                //retorna los miles de microsiemens
    centenas = int (total/100)-miles*10;     //retorna centenas...
    decenas = int (total/10)-miles*100-centenas*10; //retorna decenas...
    unidades = int (total)-miles*1000-centenas*100-decenas*10; //retorna unidades...

    lcd.setCursor(3, 1);
    if (miles > 0) {
        lcd.print(miles);
    }
    else {
        lcd.print(" ");
    }
    lcd.print(centenas);
    lcd.print(decenas);
    lcd.print(unidades);
}
}

```

Frame 2:

El frame 2 tiene por objetivo informar al usuario la fecha de inicio de cultivo guardada. Este se visualiza cuando la variable `lcd_state[2]=1`. Este método imprime en la primera línea el string "Inicio Cultivo". En la segunda línea, obtiene la fecha de la variable `fecha[]`, e imprime el día, luego imprime el caracter "/", luego imprime el mes, luego imprime el caracter "/" para finalizar con el año. A continuación el código:

```

else if(lcd_state[2] == 1 ){ //Pantalla inicio de cultivo
    if(printed == false){
        lcd.setCursor(0,0);
        lcd.print("Inicio cultivo ");
        lcd.setCursor(0,1);
        for(int i= 0; i<3 ; i++){
            lcd.print(fecha[i]); //Esta variable se define en setup variables, guarda dia, mes y año
            lcd.print('/');
        }
        printed = true;
    }
} //Fin Pantalla inicio de cultivo

```

Frame 3:

Este frame tiene por objetivo mostrar la fecha y hora actual. Cabe destacar que el frame 3 se imprime cuando `lcd_state[2]=2`. Para lograrlo imprime en la primera línea el string "Fecha " y luego imprime con el siguiente formato: `dd/mm/aaaa`, para lo cual inicializa una variable `Datetime` de nombre `now` con la fecha y hora actual por medio de `RTC.now()`. Luego tiene especial cuidado de anteponer un 0 a las variables que sean menores a 10. En la segunda línea imprime "Hora " y sigue el mismo proceso análogo para imprimir la hora. A continuación el código:

```

else if (lcd_state[2] == 2){ // pantalla 3 de fecha y hora actual
    DateTime now = RTC.now();
    lcd.setCursor(0,0);
    lcd.print("Fecha ");
    if (now.day() > 9) {
        lcd.print(now.day(), DEC);
    }
else {
    lcd.print('0');
    lcd.print(now.day(), DEC);
}
    lcd.print('/');
    if (now.month() > 9) {
        lcd.print(now.month(), DEC);
    }
    else {
        lcd.print('0');
        lcd.print(now.month(), DEC);
    }
    lcd.print('/');
    lcd.print(now.year(), DEC);

    lcd.setCursor(0,1);

    lcd.print("Hora ");
    if (now.hour() > 9) {
        lcd.print(now.hour(), DEC);
    }
    else {
        lcd.print('0');
        lcd.print(now.hour(), DEC);
    }
    lcd.print(':');
    if (now.minute() > 9) {
        lcd.print(now.minute(), DEC);
    }
    else {
        lcd.print('0');
        lcd.print(now.minute(), DEC);
    }
    lcd.print(':');
    if (now.second() > 9) {
        lcd.print(now.second(), DEC);
    }
    else {
        lcd.print('0');
        lcd.print(now.second(), DEC);
    }
}

```

tarea()

Luego de pasar por el método de pantalla(), el loop llama a la función tarea() que tiene por objetivo realizar las acciones que tienen que realizar los distintos modos. Este tiene una activación particular la cual pasa por etapas. En primer lugar, la primera vez que se realiza tarea en el loop, debe esperar 10 segundos, lo cual se debe a que el sistema debe estar funcionando hace algún tiempo, con el fin de estabilizar las mediciones de EC. Para esto, se inicializa en setup la variable boolean inicio que tiene valor true si se encuentra en inicio, cuyo valor cambia a falso cuando se logra satisfacer la condición millis() – timer_inicio > 10000, es decir, 10 segundos desde el comienzo de su funcionamiento. Cuando se satisface esto, llama al método tarea(), setea inicio en false y prende la bomba de recirculado en caso de estar en modo water.

Posteriormente, si ocurre un cambio de modo por medio de botones, ocurre una condición similar en donde se tiene que satisfacer la condición millis()-timer_modos>10000 que se impuso para que cuando ocurrieran cambios de modo por botones, estos se demoraran 10 segundos en ocurrir pues el sistema operativo actual a veces puede pasar de un modo a otro pasando entremedio de otros modos, como ocurre cuando del modo Stand By se quiere pasar a Flush, se debe pasar por medio de Refill y Water, donde también se prende la bomba de recirculado en caso de estar en modo Water, y en caso de no ocurrir un cambio que sea de modo water, apague las bomba de recirculado. Finalmente si no es inicio, ni se ha ocurrido un cambio por botones, el método tarea() ocurre sin ningún tipo de retraso. A continuación se expone el código del loop que muestra lo antes mencionado:

Por otro lado, la función tarea() depende de lcd_state[0] pues esta es la variable que denota el modo. Para explicar mejor el método, se analizará por modos.

Modo Stand By:

Para efectos de esta implementación, el modo Stand By no genera cambios en el sistema y sólo se muestra el frame 1.

Modo Refill:

Si se encuentra en modo refill, quiere decir que se desea introducir agua en el reservorio hasta que el sensor de flotación ubicado en el tope de la reserva. Para esto, lee la variable de flotador con el método digitalRead(), y si se detecta que el flotador no está flotando, se prende la bomba de llenado y si se detecta que el flotador no flota, se apaga la bomba de llenado. A continuación el código:

```
if(digitalRead(flotador)==HIGH){ //si flotador no está flotando
    digitalWrite(llenado, relay_ON); //prende bomba de llenado
}
else if(digitalRead(flotador)==LOW) {
    digitalWrite(llenado, relay_OFF);
}
}
```

Modo Water:

El modo water tiene por objetivo inicialmente calibrar el fertilizado para que entregue la cantidad suficiente para que sea rápido y seguro, para luego monitorear la cantidad de fertilizante presente en el agua. Este enciende durante todo su funcionamiento la bomba de recirculado, con el fin de mezclar todo lo mejor posible

Calibración

La calibración consta de 2 partes. La primera guarda la EC presente en el agua en la variable EC1, y luego añade un fertilizante a la vez y los introduce respetando la proporción correcta. Para esto, enciende las bombas peristálticas 0,2 segundos por cada unidad de proporción. Este proceso ocurre sólo una vez en cada sesión lo cual se logra mediante el uso de la variable boolean `calibracion1`, la cual se inicializa falsa durante el setup y luego se hace verdadera, haciendo alusión a que la calibracion1 fue realizada.

La segunda parte requiere que la calibración1 haya sido realizada, y también requiere que hayan pasado 5 minutos entre `millis()` y `timer_cal` lo que indica que desde el último fertilizado hayan pasado 5 minutos para que tenga tiempo de ser mezclado. El objetivo de esto es calcular `ferti_multi`, variable que es el múltiplo de la cantidad de tiempo que se prenden las bombas y tiene por objetivo que suba de 25 en 25 los $\mu\text{S}/\text{cm}$ por cada vez que se fertilice el agua, para añadir más robustez al sistema, lo cual se calcula con la siguiente fórmula:

$$ferti_multi = \frac{25}{total-EC1}.$$

Por otro lado, para indicar que esta fase ya fue hecha, se sigue la misma lógica anterior en donde se inicializó `calibracion2` como falsa y al ocurrir la calibración, se hace verdadera

A continuación el código:

```
else if(lcd_state[0]==2){ // water
  if(!calibracion1){ // si no ha ocurrido calibracion 1, calibramos.
    EC1 = total;
    for(int i=0; i<5; i++){
      digitalWrite(fertis[i], relay_ON); //prendemos peristaltica i
      delay(ferti_proporcion[i]*200); // se dejan las bombas prendidas 200*proporcion ms
      digitalWrite(fertis[i], relay_OFF); //apagamos peristaltica i
    }
    timer_cal = millis();
    calibracion1 = true; // Pasa una vez y no vuelve al proceso
  }
  else if (((millis() - timer_cal) > 300000) && !calibracion2) {
    ferti_multi = 25/(total-EC1);
    calibracion2 = true; // Pasa una vez y no vuelve mas a este proceso
    timer_cal=millis();
  }
}
```

Luego de las calibraciones, viene el proceso de fertilizar con las proporciones correctas para llegar a la EC. Para esto, también espera 5 minutos entre fertilizados, toma como requisito que calibracion2 y como requisito de fertilizado que la EC del momento medida por la variable total sea por lo menos 50 $\mu\text{S}/\text{cm}$ más baja que la EC objetivo. De ocurrir dicha condicional, da inicio a encender todas las bombas peristálticas con la variable fertis[i] que guarda los pins a los cuales está asociadas las bombas peristálticas en orden de izquierda a derecha, lo que se observa con digitalWrite. Por otro lado, setea las variables booleanas ferti_EC[i] en falso, lo que no permite al programa entrar de nuevo a esta función hasta que las bombas de fertilizado sean apagadas en la siguiente sección. Las variables ferti_EC[i] permiten dejar funcional en el caso que sean verdaderas, el encendido de las bombas peristálticas, y en el caso que sea falsa, deja funcional el apagado de las bombas peristálticas. Al mismo tiempo que ocurre el encendido de las bombas, se guarda el valor del tiempo en la variable timer_ferti[i], y también como cualquier proceso de fertilizado, actualiza la variable timer_cal que no permite fertilizar antes de 5 minutos entre fertilizados.

Después del encendido de las bombas peristálticas y demases, se da paso al código de apagado de bombas, el cual pide como condicional que millis() – timer_ferti[i] > ferti_proporcion[i]*200*fertimulti lo cual se interpreta en que el tiempo transcurrido entre el encendido del fertilizado y el tiempo actual sea mayor a la proporción de fertilizante por 200 milisegundos por fertimulti. Como ocurren muchos loops en una milésima de segundo, la condición se asemeja a que esta comparación sea igual. También pide como requisito que ferti_EC[i] sea falsa, variable que se configuró falsa en el proceso anterior. Cuando se logra la condición, simplemente se apagan las bombas peristálticas con digitalWrite(fertis[i], relay_OFF) y se setea la variable booleana ferti_EC[i] en true, permitiendo que el loop anterior vuelva a ocurrir para fertilizar lo que concluye el ciclo. A continuación el código de fertilizado:

```

if ((millis() - timer_cal > 300000) && ((tEC-total)>50) && calibracion2) {
  for(int i=0; i<5; i++){
    if(ferti_EC[i]){ //
      digitalWrite(fertis[i], relay_On);
      ferti_EC[i]=false; //cuando pasa por aqui una vez, esta variable se setea falsa
      timer_ferti[i] = millis();
    }
  }
  timer_cal = millis(); //timer
}
for(int i=0; i<5; i++){
  if((millis() - timer_ferti[i]) > ferti_proporcion[i]*200*ferti_multi && !ferti_EC[i] ){
    digitalWrite(fertis[i], relay_OFF);
  }
  ferti_EC[i] = true;
}

```

Flush()

Este proceso tiene por objetivo eliminar toda el agua presente en la reserva lo cual puede servir para limpiar la reserva, como para acabar un cultivo, como para cambiar el agua presente por higiene. Se logra mediante el uso de una variable booleana llamada drain

la cual sirve para entrar en el loop una vez, pues se pide como condicional para el inicio que drain sea verdadera y cuando la bomba de vaciado se prende con digitalWrite(drenaje, relay_ON), se setea la variable en falso y se guarda el tiempo en el que ocurrió con timer_flush. El seteo de drain en falso, sirve para pasar a la sección en donde se apaga la bomba de vaciado, la cual tiene como requisito de entrada que la variable drain sea falso y que hayan pasado al menos 5 minutos entre millis() y timer_flush. Esto asume que en 5 minutos puede vaciar la reserva. A continuación el código

```
else if(lcd_state[0]==3){ // flush
  if (drain == true) {
    digitalWrite(drenaje, relay_ON);
    drain = false; //falseamos para no entrar denuevo al if
    timer_flush=millis(); //seteamos el timer para apagar el rele
  }
  if(millis()-timer_flush > 300000 && drain = false){ // 300 segundos para drenar
    digitalWrite(drenaje, relay_OFF); //permitimos flushear si elegimos el modo nuevamente
    printed=false; // refrescamos la pantalla ya que cambio el modo
    drain=true;
  }
}
```

Transición:

El sistema tiene que pasar por las siguientes etapas previo a su aprobación para producto final.

- Beta Testing
- Operación en paralelo relativo a los sistemas y legados que estén reemplazándose
- Conversión de datos operacionales
- Entrenamiento de usuarios y de programadores que mantendrán el software
- Ajuste de bugs y afinado del performance

3.4.3.5 Experimentación y Resultados de Laboratorio

Tomando en cuenta todo lo explicado anteriormente, se debe pasar a la fase de experimentación del software, para comprobar el correcto funcionamiento de las funciones descritas, especialmente la función de getEC(), la cual realiza los pasos necesarios para la obtención de la conductividad eléctrica del medio.

Para esto se intentó comparar las mediciones de getEC() la cual, como mencionado previamente, toma 40 muestras espaciadas entre 40 milisegundos, con una versión más robusta de getEC() la cual obtiene 10.000 datos y calcula su promedio.

Para realizar dicha experimentación se guardaron los datos de la EC observada cada 5 minutos, en donde el sensor se pone con una solución calibrada a 1000 μ S, tomando una muestra total de 2000 datos para cada forma de experimento, con un total de 4000 datos para los 2 experimentos (Ver Anexo 3 "Datos Individuales de Medición de Experimento" para los datos individuales).

Para evaluar las 2 formas mencionadas anteriormente de getEC(), se evaluaron los valores mínimos, máximos, medias, medianas, desviación estándar, como la distribución

de los datos obtenidos, para poder concluir respecto a su desempeño. En lo que sigue del informe se refiere a Experimento A como la medida que toma 10.000 datos para su obtención y Experimento B como la medida que toma sólo 40 datos espaciados entre 40 milisegundos.

Resultados

Tanto los resultados obtenidos de todas las mediciones, como las tablas de frecuencias obtenidas de cada experimento se pueden ver en anexos en las tablas 1 y 2.

A continuación una tabla con los principales indicadores estadísticos y un gráfico de sus distribuciones:

	Experimento A [μS]	Experimento B [μS]
Mínimo	983	977
Máximo	1021	1021
Media	1000	997
Mediana	999	998
Moda	998	1000
Desviación Estándar	8	10

Tabla 1: Principales indicadores estadísticos
Fuente: Elaboración Propia

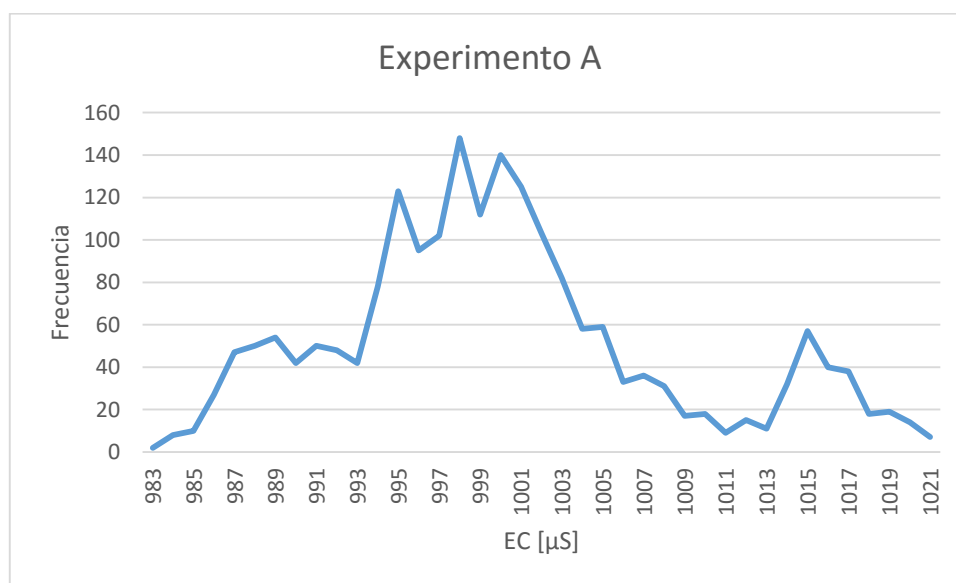


Figura 15: Gráfico de Distribuciones A
Fuente: Elaboración Propia

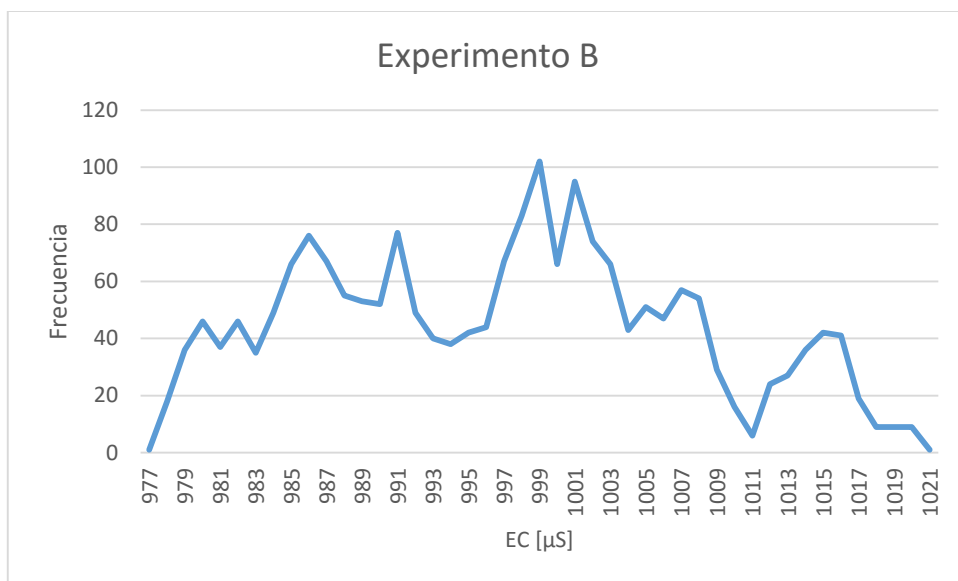


Figura 16: Gráfico de Distribuciones B
Fuente: Elaboración Propia

Análisis de Resultados

Como se puede observar en los resultados, ambos experimentos se encuentran muy cercanos a la realidad en términos de media, siendo 1000 la media en A y 997 en B, lo cual nos permite concluir que en promedio ambas formas de toma de datos son aceptables. Esto se concluye pues en las recetas vistas en anexos los valores de EC intentan estar cercanos a múltiplos de 100, por lo que una diferencia de 3 unidades se considera despreciable. Por otro lado, los valores de desviación estándar son respectivamente 8 y 10, permitiendo obtener los siguientes intervalos de confianza al 95%:

	Mínimo por Intervalo	Máximo por Intervalo
Experimento A [μS]	984	1016
Experimento B [μS]	976	1017

Tabla 2: Intervalos de Confianza
Fuente: Elaboración Propia

Esta tabla nos demuestra que, con confianza del 95%, se puede ver que para ambas formas de medir no exceden el valor $1000 \pm 25 \mu\text{S}$ lo cual permite concluir que el dispositivo logrará mantener los niveles de EC con un error de $\pm 25 \mu\text{S}$, lo cual se considera aceptable según la literatura.

Se concluye que la forma de `getEC()` B la cual se explica en la sección Loop de este reporte de desarrollo por usar menos recursos y tener resultados aceptables es la mejor forma entre las 2 vistas anteriormente. También se concluye que la forma de medición escogida basta para mantener un error de medición aceptable para hidroponía, según la literatura consultada.

4. Análisis del Entorno

El análisis a realizarse en esta sección se orientará por sobre todo en el futuro del producto, y el mercado en el que competirá. Esto, siempre y cuando el SHA supere las pruebas de terreno satisfactoriamente y llegue a la etapa de aprobación para comercialización. Por ende, el análisis realizado se enfocará primariamente en el mercado y posición de la Hidroponía actualmente. Posteriormente, se utilizará el modelo Delta para explicar en qué momento y estrategia se encuentra el producto a ser comercializado.

4.1 Análisis de Sector

Los cultivos hidropónicos en Chile no están masivamente distribuidos, sino que se realizan a escalas pequeñas, a modo de investigación, y algunos productores de mayor tamaño, que ofrecen normalmente hortalizas y frutos hidropónicos orgánicos, como diferenciación de los cultivos en tierra. Sin embargo, la distribución de hidroponía en Chile no es alta, y el Ministerio de Agricultura no la considera en sus estadísticas hasta el año 2014¹⁵. La única fuente que se dispone es el Instituto de Investigaciones Agropecuarias, INIA, en el cual se describen algunos proyectos de investigación de aumento de rendimiento siendo llevados a cabo desde el año 2015 en el norte de Chile¹¹, pero que cuya distribución puede ser considerada despreciable para la producción nacional (por la falta de estadísticas del INE y del Ministerio).

Como el producto se está perfilando como un “Sistema Hidropónico Automatizado para el autoconsumo de vegetales y plantas ornamentales”, se propone un análisis de estudio de todos los productos y servicios aledaños a lo que viene siendo jardinería casera. Estos incluyen venta de fertilizante, sustratos, pesticidas, plaguicidas, sistemas de cultivo, sistemas de regadío, accesorios de jardinería, semillas y plantas en general. Por otro lado, esto incluye el servicio de jardinería y de paisajismo.

El sector tiene distintos puntos de venta, a lo largo de todo Chile. En particular, existen 3 tipos de comercios asociados a este sector. Estos son:

Retails: Estos se componen por Homecenter Sodimac, Easy, y algunos supermercados como Jumbo o Lider. Estos se caracterizan por tener productos genéricos para las distintas necesidades asociadas a la jardinería. En particular carecen de soluciones específicas y de sistemas de cultivo aparte del tradicional macetero. Su rentabilidad se estima de la siguiente forma: Los productos de jardinería representan el 5% de la venta total, del cual sólo un 1% representa la venta de plantas¹⁶. Por otro lado, se sabe que los ingresos de Homecenter Sodimac fueron de 1.866.450 MM CLP entre el 31 de Marzo 2014 y 31 de Marzo 2015¹⁷, mientras que Cencosud en su segmento Mejoramiento del Hogar tuvo ventas de 1.225.616 MM CLP para el 2014¹⁸. Se estima entonces que las ventas en artículos de jardinería y plantas para ambos retails fueron de 154.603 MM CLP por año. Se utilizarán estas cifras como una cota inferior, dado que existen otros Retail que venden artículos de jardinería, pero se consideran despreciables versus Homecenter y Cencosud. A través de este pequeño análisis de ventas, podemos determinar que la mayoría de los consumidores de Mejoramiento del Hogar y jardinería corresponden al GSE ABC1-C2, que son el público objetivo de los retails con más ventas anteriormente

mencionados. Considerando que estos dos grupos corresponden al 22,6%¹⁹ del total de la población del país, y asumiendo a través de la extrapolación de público consumidor potencial versus el GSE, que el 1% del universo de 22,6% consumirá productos de jardinería orientadas a producción y consumo de hortalizas y frutas, correspondientes a aproximadamente 40.000 potenciales usuarios de base, con posibilidad de crecimiento a medida que el producto se haga más conocido.

Grow Shops: Corresponden a tiendas especializadas en cultivos de plantas y hortalizas, atendiendo además a la venta legal de semillas de Cannabis en Chile, con productos relacionados al cultivo, como lo son sustratos, fertilizantes, accesorios de poda, maceteros, sistemas de riego automático, etc. Se estima por medio de una entrevista con Daniela Hetch²⁰, dueña de Germinia Grow Shop, que la venta aproximada de los Grows gira en torno a los 4 MM CLP por mes, y el medio de estimación es porque ningún growshop contactado quiso compartir ese tipo de información. El número estimado de Grow Shops a lo largo de Chile asciende a 300 tiendas. Esto lo podemos ver en la figura 14 Growshops por Región. No existen cifras del total de clientes de los grow shops, pero a través de la entrevista realizada a Daniela Hetch, podemos aseverar que al menos 4 mil usuarios comprarían un sistema hidropónico automatizado así como se hiciera disponible.

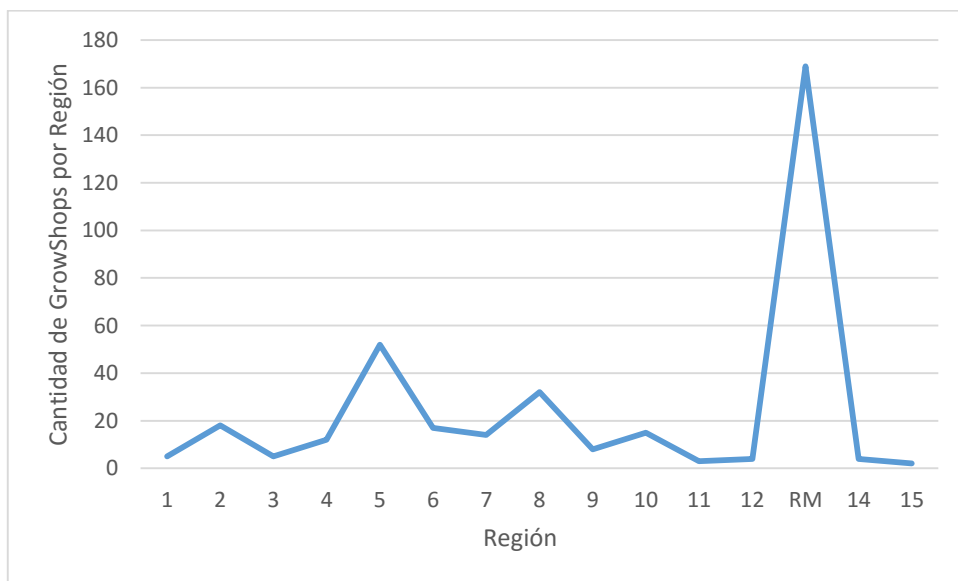


Figura 17: “Growshops por región”
Fuente: Elaboración propia

Finalmente, se estima que de las ventas de cada GrowShop, un 50% corresponde a ventas de artículos de jardinería, por lo que los ingresos en jardinería corresponden a 7200 MM CLP por año.²¹

Viveros: Corresponden a comercios o instalaciones donde se provee a las plantas con condiciones para que crezcan bien durante su desarrollo inicial. Estos son tanto de uso forestal como ornamental. En Chile existen registrados 260 viveros distintos los cuales se venden cerca de 560 especies distintas. Con respecto a su distribución a lo largo de

Chile, estos se concentran en el centro, concentrándose en la región de BioBio, Metropolitana y Valparaíso. La cantidad de plantas existentes para el 2015 tienen un valor aproximado de 27.928,6 MM CLP. Dado a que se tiene información respecto a la existencia de plantas, y no del flujo anual, se escoge, para términos de la memoria considerar sólo el 10% de las ventas, para simbolizar²²

Finalmente se puede decir que el sector maneja aproximadamente una cantidad anual de 190.000 MM CLP anualmente, entregando una posibilidad de entrar con un producto nuevo en un mercado sin competencia directa actualmente.

4.2 Análisis de Competencia

El rubro de la hidroponía es poco explotado a nivel doméstico por los distintos factores antes mencionados. Es por esta razón que aún no es posible adquirir sistemas de cultivo integral que puedan dar solución a este problema. Actualmente existen en etapa de desarrollo dos productos que intentan un acercamiento con el usuario común, estos son: GroBox de CloudPonics y OpenGrow.

OpenGrow es un sistema de monitoreo para el cultivo, consistente en kits de aparatos con diferentes funciones a la hora de evaluar las condiciones de los cultivos. Existe la posibilidad de adquirir cada componente de los kits por separado. Dentro de las ventajas destacables se encuentra la fácil interacción con el usuario, bastante intuitivo y dinámico a la hora de presentar los datos recabados de forma que el cultivador pueda evaluar su cultivo de manera más fácil. Como desventaja se destaca que el sistema carece de un medio especialmente diseñado en el cual hacer crecer las plantas. Esto incide directamente en una alta variabilidad en los resultados obtenidos por los diferentes cultivadores dado que las condiciones difieren entre un usuario y otro, siendo el medio de cultivo un factor muy importante a la hora de evaluar resultados y eficiencia de todo el proceso de cultivo. Los precios fluctúan entre los \$520-\$840 USD, los cuales podrían ser considerados dentro de un rango medio, sin embargo ir complementando el kit aumenta el costo considerablemente haciéndolo menos accesible.²³

GroBox de Cloudponics es un sistema de cultivo integral, que incluye una estación de monitoreo, así como también un medio de cultivo especialmente diseñado para ubicar los vegetales y hortalizas. Ésta es una fortaleza del sistema dado que reduce la variabilidad en los cultivos, asegurando un estándar de producción entre un usuario y otro. También consta de una interfaz amigable con el usuario mediante una aplicación de smartphone que facilita la interacción con el cultivador mediante imágenes y diagramas logrando así una plataforma más intuitiva. Una desventaja es el costo, contando con tan solo un modelo que incluye tanto la estación de monitoreo como la de cultivo, el costo asciende sobre los \$2000 USD.²⁴

Es importante notar, que ambos productos se encuentran en etapa de desarrollo, por lo que aún no pueden ser adquiridos por los usuarios, solo es posible hacer una reserva para cuando se lancen al mercado. Además ambos productos van orientados al mercado norteamericano y exclusivamente dirigido al cultivo de la especie Cannabis Sativa. Por último cabe notar su alto costo en relación a lo que son capaces de ofrecer, en conjunto con no especificar ciertas virtudes o falencias que podrían afectar el desempeño del

producto o la logística a la hora de instalarlo, como por ejemplo, de qué manera se abastecerá de agua al sistema, así como tampoco especifican de qué manera se harán los recambios de esta.

4.3 Análisis de Mercado

Por el manejo del dispositivo, y su costo de adquisición se prevé lanzar el sistema de hidroponía automática apuntándolo en un comienzo hacia el sector ABC1-C2 de la población chilena, enfocándose en todos quienes deseen cultivar en su hogar de forma eficiente, ecológica y a un bajo costo, libre de pesticidas y plaguicidas.

Se busca apuntar principalmente al interés del jefe de hogar o dueño de casa por tener la responsabilidad del hogar y preocuparse de forma más personal del mantenimiento y gastos de ésta. El promedio de edad de los jefes de hogar es de 45,6 años; una edad intermedia en la esperanza de vida actual.

De ésta forma se puede definir el mercado objetivo a todos los propietarios o jefes de hogar, puesto que cada uno representa una casa y a su vez un espacio para el dispositivo.

Según datos de la Asociación de Investigadores de Mercado (AIM), entre los años 2002 y el 2012, el porcentaje de población del segmento ABC1 subió desde un 5,4 a un 5,9% de la población total, viéndose un incremento de un 0,5% de la población total entre esos 10 años.

Tomando los tamaños de la población total de esos años se pueden sacar el tamaño total de la población ABC1. Así para 15,82 millones de chilenos que había en el 2002, el 5,4% que corresponde a los ABC1 son 854.280 habitantes; y para la población total del país de 17,46 millones en el 2012, el 5,9% corresponde a 1.030.140 habitantes del sector ABC1).

Con los datos anteriores se puede usar para aproximar el aumento futuro de este segmento poblacional. De esta forma, entre el 2002 y el 2012 la población ABC1 aumentó en 175.860 habitantes, por lo que aumentó en un 20,5% el número de individuos en 10 años.

4.4 Análisis de 5 Fuerzas de Porter

El mercado objetivo al cual está dirigido el sistema se compone de jefes del hogar que busquen una solución integral a la hora de cultivar sus propios vegetales y hortalizas, libres de pesticidas y de una manera eficiente desde el punto de vista hídrico y energético. Los clientes potenciales como ya fue mencionado ascienden a 40.000, encontrándose dispersos a lo largo del país, sin una organización definida que les permita tener incidencia en el precio del producto. Por otro lado aún no existen otros proveedores de un sistema similar, por ende el cliente carecería de poder elegir entre un producto u otro, ya sea por precio, prestaciones, etc.

El sistema se compone de una serie de piezas que interactúan entre sí de una manera lógica y eficiente. Dichas piezas son provistas por múltiples proveedores alrededor del mundo, principalmente de China. Además por cada pieza utilizada existen al menos 3 proveedores diferentes, lo cual facilita la compra en términos de stock, precios, calidad, atención, etc. Es por esto que existen pocas limitantes a la hora de conseguir los componentes para desarrollar el sistema.

Por el momento existen en etapa de desarrollo otros productos que cumplen funciones similares al sistema, sin embargo su mercado objetivo es el norteamericano y además van exclusivamente dirigidos al cultivo de la especie Cannabis Sativa. Dado que este sistema se compone de varias piezas diferentes, su construcción se vería altamente beneficiada mediante las economías de escala, ya que los precios de los componentes a importar se reducen considerablemente por volumen. Al momento del desarrollo de la presente memoria, este sistema automatizado hidropónico aún no cuenta con capitales aportados por terceros, lo que ha dificultado el acceso a los mejores componentes y sus precios. Mediante algunas entrevistas realizadas entre los meses de Septiembre y Diciembre del año 2015, se ha podido constatar que el acceso a los canales de distribución, en este caso tiendas de retail y Growshops, de un sistema como este, se encuentran abiertos y dispuestos a comercializar el producto, dado que sería el único de su clase y prestaciones, lo cual atrae a vendedores y compradores por igual.

Es difícil encontrar productos sustitutos que ofrezcan todas las prestaciones del sistema, dado que la automatización de procesos fuera de un simple riego es un área poco explorada a la hora de cultivar vegetales y hortalizas a nivel domiciliario, al momento de la presente memoria no existen sistemas de fertirrigación automáticos que funcionen en conjunto con el monitoreo y control de las condiciones ambientales de los cultivos.

Dado que por el momento el sector comprende pocos o nulos competidores a la hora de idear soluciones como el dispositivo mencionado, además del gran mercado domiciliario disponible, es que se deja entrever una gran rentabilidad en el sector a ser capturada por el sistema.

4.5 Modelo Delta

Tomando como base el modelo delta, y la naturaleza del desarrollo de este producto, se puede concluir que al no existir competencia directa, y al ser un producto nuevo en el mercado, la principal política de comercialización es la de “Mejor Producto” según el modelo. Esto se basa principalmente en que al no existir productos de igual función, entra al mercado como un producto único. Además, si bien existen empresas de automatización de cultivos, éstas entran como una empresa de servicios, que construye, controla y modera el cultivo, con el agricultor sólo a cargo de vender sus existencias (fuente empresa servicio) además de estar limitado al país en que se desarrollan (USA), y unidos a un alto costo de entrada, que puede ascender a los 45 mil dólares.



Figura 5: Posturas Estratégicas
Fuente: Hax, A. & Wilde D., *The Delta Project*, 2001

Utilizando la estrategia de Competencia de Mejor Producto, se propone entrar con la subestrategia de “Diferenciación”, entregando un producto con un valor único, que es la “Automatización de la fertilización, llenado, recirculamiento y vacío de agua”, acciones únicas en el mercado actualmente.

Hacia la estabilización del producto comercializable, se propone migrar la estrategia hacia la experiencia de cliente, a través de la capacidad de compartir las recetas de cultivo específicas para cada planta o vegetal a través de una comunidad abierta online. Esto se hará principalmente con el objetivo de consolidar el producto como la única opción de automatización democrática de los autocultivos, permitiendo generar una comunidad gratuita en donde los propios usuarios compartirán sus experiencias y resultado.

No se han realizado mayores proyecciones debido a que el producto aún no está en su fase comercializable, por lo que se espera que éste supere las etapas de pruebas de terreno para poder proyectar con mayor seguridad.

5. Conclusiones

Actualmente se cuenta con un prototipo funcional, que superó las pruebas de laboratorio, y se está desarrollando un prototipo a escala mayor para iniciar las pruebas en terreno, incluyendo la prueba con usuario, para obtener un feedback y atender las observaciones que pueda realizar el mismo cuando interactúa con el sistema. Una vez terminada esta etapa, se pasará a la construcción del producto final, con un modelo comercializable, con todas las variables definidas (precio, packaging, distribución y producto final).

Cabe destacar que la programación del software del sistema no fue simple, y consumió la mayor parte del tiempo de desarrollo del prototipo. Sin embargo, se logró que cumpliera los requerimientos mínimos de automatización, con la proyección de agregar más variables en el futuro, con un segundo proyecto en donde se consideren las variables descartadas en el sistema actual.

Glosario

Nombre	Tipo Variable	Descripción
ArrayIndex	int	Índice de arreglo ECArray[].
calibracion1	boolean	Avisa si está completa la primera fase de calibracion para obtener ferti_multi.
calibracion2	boolean	Avisa si está completa la primera fase de calibracion para obtener ferti_multi.
centenas	int	Variable de un dígito entero que guarda la centena para imprimir en SD la EC actual.
decenas	int	Variable de un dígito entero que guarda la decena para imprimir en SD la EC actual.
drain	boolean	True permite que se encienda bomba de vaciado, False hace lo opuesto.
EC1	int	Se usa para guardar la primera medida de EC para calibrar el sistema, que busca fertilizar de forma segura.
ECArray[]	int	Guarda valores de variables análogas correspondientes a voltajes.
estado[]	char*	Guarda nombres de MODE, utilizados para mostrar en pantalla LCD.
fecha[]	int	Guarda fecha de Inicio.
ferti_EC[]	boolean	True permite que se encienda bomba de peristáltica en el fertilizado, False hace lo opuesto.
ferti_multi	float	Número que se utiliza para lograr un fertilizado seguro, el cual se obtiene en el calibrado en WATER.
ferti_proporcion[]	int	Proporciones provistas por el cultivador para las 5 bombas peristálticas de izquierda a derecha.
fertis[]	int	Numeros de pins digitales que se les asigna a las bombas peristálticas, de izquierda a derecha en el aparato.
flotador	int	Numero de pin digital que lee entradas del flotador.
inicio	boolean	Se usa para la primera tarea() pues requiere 10 segundos de delay. Despues de esto, toma valor falso.
intervalosEntreMedidas	int	Cantidad de milisegundos para la toma de datos de EC.
lcd_key	int	Guarda valor de boton ultimo loop. Con lcd_key2 permiten observar variacion en botones.
lcd_key2	int	Guarda valor de boton penultimo loop. Con lcd_key2 permiten observar variación en botones.
lcd_state[]	int	Guarda coordenadas usadas por la pantalla LCD y se usa para comunicar el modo a otros métodos.
llenado	int	Nombre del pin que controla el relay de la bomba de llenado. HIGH si el flotador no flota, LOW si el flotador flota.
miles	int	Variable de un dígito entero que guarda la centena para imprimir en SD la EC actual.
nFrames	int	Cantidad de frames existentes en el script.
nMuestra	int	Cantidad de muestras que usa ECArray[].
primeraproporcion	boolean	boolean que permite que setProportion() pase por el primer loop.
printed	boolean	Variable que utiliza tanto boton() como pantalla() para comunicar si un cambio fue impreso. Boton setea falso, Pantalla setea verdadero.
printTime	int	Se utiliza para espaciar impresiones en LCD. Guarda tiempo en el que se imprimió EC.
samplingTime	int	Se utiliza para espaciar tomas de datos de EC, por método getEC().
tEC	int	Indica la EC objetivo, la cual es actualizada por el método setProportion.
timer_cal	long	Se utiliza para espaciar fertilización del caldo de cultivo. Guarda tiempo en que se fertilizó.
timer_ferti	long	Se utiliza para lograr que las bombas peristálticas se enciendan lo justo y necesario para fertilizar.
timer_inicio	long	Se utiliza sólo para el inicio en donde se pide como requisito que el sistema no haga nada los primeros 10 segundos.
timer_moda	long	Se utiliza para espaciar tiempos entre seteo de modo por medio de botón y la acción que desencadena dicho seteo por 10 segundos.
timer_receta	long	Se utiliza para espaciar tiempos entre updates de tEC y ferti_proporcion[], lo cual se hace cada una hora.
total	int	Valor de la EC medida que es actualizada por el método getEC().
unidades	int	Variable de un dígito entero que guarda la centena para imprimir en SD la EC actual

Bibliografía

- [1] S.A *International Data Base. World Population: 1950-2050*. The United States Census Bureau. Sitio web: <https://www.census.gov/population/international/data/idb/worldpopgraph.php>, consultado al 8 de Octubre 2015.
- [2] JELLE, B. *The Resource Outlook to 2050: By How Much do Land, Water and Crop Yields Need to Increase by 2050* . en "Expert Meeting on How to Feed the World in 2050". Food and Agriculture Organization of the United Nations, 2009. Sitio web: "<ftp://ftp.fao.org/docrep/fao/012/ak971e/ak971e00.pdf>", consultado al 15 de Octubre 2015.
- [3] MSANGI, S., TOKGOZ, S., BATKA, M., ROSENGRAT, M. *Investment Requirements Under New Demands on World Agriculture: Feeding the World With Bioenergy and Climate Change*. en "Expert Meeting on How to Feed the World in 2050", Food and Agriculture Organization of the United Nations, 2009. Sitio web: <ftp://ftp.fao.org/docrep/fao/012/ak973e/ak973e00.pdf>, consultado al 15 de Octubre 2015.
- [4] BELTRANO, J. GIMENEZ, D. *Cultivo e Hidroponia*. Universidad Nacional de La Plata, Argentina: Facultad de Ciencias Agrarias y Forestales, 2015.
- [5] HASHIMOTO, Y., *Computer Integrated Plant Growth Factory for Agriculture and Horticulture*, en "Mathematical and Control Applications in Agriculture and Horticulture", Pergamon Press 1991, p. 105-110
- [6] SAVVAS, D. *Hydroponics: A modern technology supporting the application of integrated crop management in greenhouse*, Food, Agriculture & Environment Vol.1(1), p. 80-86. 2003
- [7] MISHRA, R. JAIN, P. *Design and Implementation of Automated Hydroponic System using ARM Processor*, International Journal of Advances Research in Electrical, Electronics and Instrumentation Engineering, IJAREEIE, Vol 4, Edición 8, Agosto 2015.
- [8] SAHARE, V. JAIN, P. *Automated Hydroponic System using Psoc4 Prototyping Kit to Deliver Nutrients Solution Directly to Roots of Plants on Time Basis*, International Journal of Advances Research in Electrical, Electronics and Instrumentation Engineering, IJAREEIE, Vol 4. Edición 11, Noviembre 2015.
- [9] Fullbloom Hydroponics, sitio web de sistemas hidropónicos semiautomatizados, <http://www.fullbloomhydroponics.net> consultado al 4 de Noviembre 2015
- [10] RUIZ, P. ARRIBILLAGA, D. *Microclima en Invernaderos: Importancia de la ventilación y Estrategias de Manejo*. Ministerio de Agricultura, Instituto de Investigaciones Agropecuarias, INIA, Tamel Aike, Coyhaique, Octubre 2013.
- [11] Proyecto *Obtención de hortalizas baby, mediante sistemas productivos de bajo requerimiento hídrico en la Región de Atacama*, Ministerio de Agricultura, Instituto de Investigaciones Agropecuarias, a finalizar el 18 de Junio de 2016

- [12] DESPOMMIER, D. *The Rise of Vertical Farms*, Scientific American, Noviembre 2009, p. 80-87
- [13] GARG, A. BALODI, R. *Recent Trends in Agriculture: Vertical Farming and Organic Farming*, Adv Plants Agric Res., 1(4): 00023, 2014
- [14] S.A Arduino LCD KeyPad Shield (SKU: DFR0009) Sitio web http://www.dfrobot.com/wiki/index.php?title=Arduino_LCD_KeyPad_Shield_%28SKU%3A_DFR0009%29 consultado al 28 de Noviembre 2015
- [15] S.A *Agropecuarias, Informe Anual 2014*, Instituto Nacional de Estadísticas, 2014-2015, Consultado al 1 de Diciembre 2015
- [16] S.A *La Industria Ornamental y el Mercado Masivo*, Redagrícola.com, reportaje recuperado de <http://www.redagricola.com/reportajes/marketing/la-industria-ornamental-y-el-mercado-masivo> al 1 de Diciembre 2015
- [17] JIMENEZ, A., VÉRTIZ, J., *Análisis de Riesgo*, Sodimac S.A. Fitch Ratings, 2015. Consultado al 1 de Diciembre 2015
- [18] S.A *Memoria Anual 2014*, Cencosud, 2014, Consultado al 1 de Diciembre 2015
- [19] S.A *Mapa Socioeconómico de Chile*, Adimark, Recuperado de http://www.adimark.cl/medios/estudios/mapa_socioeconomico_de_chile.pdf al día 1 de Diciembre de 2015
- [20] Entrevista: Daniela Hetch, Elaboración Propia, al 26 de Noviembre 2015
- [21] Guía Latinoamericana del Cañamo 2015, Consultada al 22 de Noviembre 2015
- [22] S.A *Listado de Viveros*, CONAF, Archivo Excel recuperado de <http://www.conaf.cl/nuestros-bosques/bosques-en-chile/viveros/> al 27 de Noviembre 2015
- [23] OpenGrow, recuperado de <http://www.opengrow.pt> al 28 de Noviembre 2015
- [24] GroBox de Cloudponics, recuperado de <http://www.cloudponics.com/> al 28 de Noviembre de 2015

ANEXO I
“SISTEMAS HIDROPÓNICOS”

Este Anexo contiene los sistemas hidropónicos que se utilizan actualmente, así como algunas definiciones necesarias para entenderlas.

Hidroponía: Método para cultivar plantas en una base o caldo de cultivo con nutrientes, sin el uso de tierra. En vez de eso, las raíces son soportadas por sistemas que contengan sustratos inertes como perlita, vermiculita, lana de roca, arcilla expandida, turba entre otros. A continuación se exponen ventajas y desventajas del cultivo hidropónico:

Ventajas:

- Maduración 25% más rápida y pueden producir 30% más que las plantas que se producen en tierra.
- Raíz se esfuerza menos para absorber nutrientes.
- Si se controla bien el pH y la EC, se puede obtener un cultivo más óptimo en términos de uso agua pues no se va por el subsuelo ni se evapora con facilidad.
- Capacidad de utilizar suelo no apto para el cultivo.

Desventajas:

- Sistema hidropónico siempre va a ser más caro que uno de tierra.
- Puede ser tedioso para cultivadores inexpertos, pudiendo producir pérdidas. Esto se debe a que se deben revisar todo el tiempo los niveles de EC y pH.
- Fallas en el sistema pueden ser perjudiciales en días, pues las condiciones de las raíces pueden variar.

Otro aspecto que hay que mencionar es que el **caldo de cultivo** es la solución que corre por estos sistemas, la cual contiene fertilizante.

A continuación se describen métodos conocidos de hidroponía, los cuales son potenciales candidatos para los sistemas a comercializar.

NFT: Nutrient Film Technique, es un sistema hidropónico que contiene un flujo continuo de agua con nutrientes, la cual corre por una cañería en donde crecen las raíces de las plantas. Estas cañerías se posicionan inclinadas para que la fuerza de gravedad mueva el caldo de cultivo.

Este sistema destaca pues las raíces tienen mucho contacto en el aire. Este contacto resulta ser positivo pues el mismo influye directamente sobre la tasa de absorción.

Se basa en tener alguna estructura que contenga las raíces, y de forma de tubo puesto horizontalmente con una leve inclinación. Una bomba de agua lleva los nutrientes a la parte más alta y por gravedad circula a la reserva de nutrientes. Se puede añadir una bomba de aire y un difusor para oxigenar la reserva. Es común ver muchas tuberías paralelas y canaletas que sirven de cañería para mover los flujos de agua. Este sistema puede tener problemas si las bombas fallan pues las raíces pueden secarse.

Figura n° 1: "Fotografía sistema NFT"



Fuente: FullBlooms Hydroponics Ref [6]

DWC: Deep Water Culture, es un sistema que hace crecer las plantas en recipientes con algún sustrato inerte, en donde las raíces crecen a través de su recipiente hasta que tocan agua de la reserva misma. Se utiliza una bomba de aire con un difusor para mejorar la oxigenación del agua. Este sistema se caracteriza por ser robusto, pues de fallar las bombas, las raíces se mantendrán hidratadas. Por su simpleza y seguridad, es de los sistemas más utilizados.

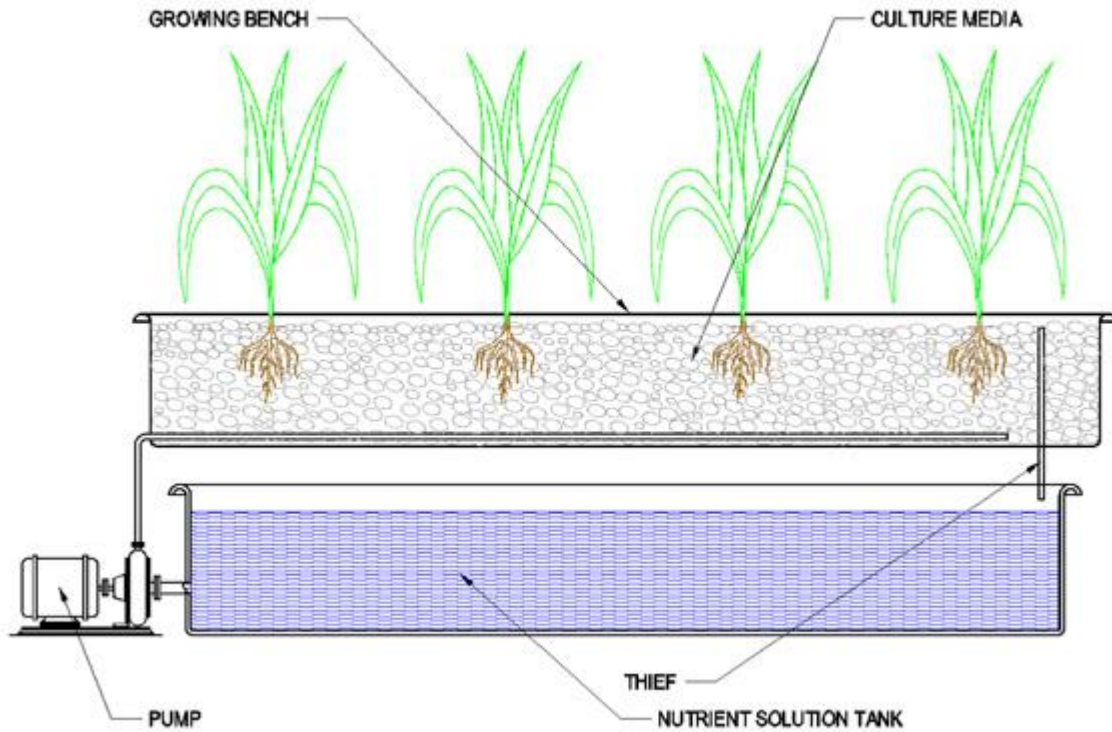
Figura n° 2: “Fotografía sistema DWC”



Fuente: FullBloom Hydroponics Ref [6]

EBB & Flow o sistema de llenado y vaciado. Consiste en un mecanismo el cual llena hasta el tope el área donde se encuentran las raíces y luego, se vacía por completo, utilizando un mecanismo de vaciado que funciona con una tubería puesta un poco más alta. Este ciclo se hace cada ciertos intervalos para finalmente hidratar pero con un alta aireación.

Figura n° 3: "Fotografía sistema EBB & Flow"



Fuente: Hydroponic Supplies [7]

Drip Systems: Sistemas de goteo, que consisten en un sustrato inerte como base, y por medio de goteo, éste se irriga de forma sostenida. Este sistema es ideal para sustratos que retienen mucha humedad, como la lana de roca, fibra de coco o turba.

Figura n° 4: "Fotografía sistema Drip Systems"



Fuente: FullBloom Hydroponics Ref [6]

Sistema hidropónico base: Es la definición que le da esta memoria a cualquier sistema hidropónico, como los antes mencionados. Este término se ocupa para explicar el formato de hidroponía usado, pero sin la componente automática.





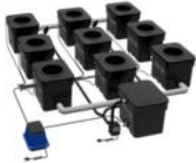





				
<p>Super BubbleFlow Buckets</p> <p>★★★★★ 11 Reviews \$695.00</p> <p>Choose Options</p>	<p>Superponics 8 Plant Hydroponics System</p> <p>\$124.95</p> <p>Choose Options</p>	<p>SuperFlow</p> <p>\$695.00</p> <p>Choose Options</p>	<p>Current Culture - Under Current XL UCXL System</p> <p>★★★★★ 11 Reviews \$985.46</p> <p>Choose Options</p>	<p>Current Culture - Under Current Evolution XL Hydroponic System</p> <p>\$1,608.26</p> <p>Choose Options</p>
				
<p>Current Culture - Under Current UC System</p> <p>★★★★★ 11 Reviews \$971.96</p> <p>Choose Options</p>	<p>Current Culture - Under Current Double Barrel XXL13 - Hydroponic System</p> <p>\$3,077.96</p> <p>Choose Options</p>	<p>PLATINIUM Pots w/ aeroponic tops (Aerostar)</p> <p>\$112.23</p> <p>Choose Options</p>	<p>Bigfoot Complete</p> <p>★★★★★ 1 Review \$374.95</p> <p>Choose Options</p>	<p>Current Culture - Under Current Solo</p> <p>★★★★★ 11 Reviews \$94.46</p> <p>Choose Options</p>

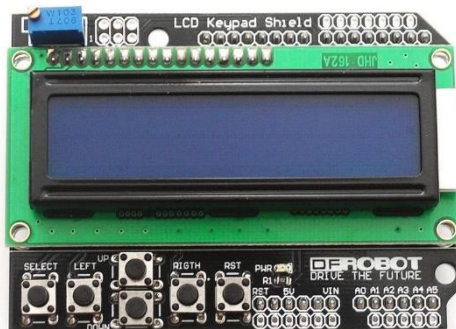
Figura 15: “Comparación de Precios de Sistemas Hidropónicos Actuales”
Fuente: Fullbloomhydroponics.net

ANEXO II

“SISTEMAS AUTOMATIZADOS Y ARDUINO”

Este Anexo contiene los elementos de automatización correspondientes a Arduino, y las piezas que se utilizaron para este proyecto.

LCD KEYPAD SHIELD FOR ARDUINO DFROBOT **SKU: DFR0009**:



Esta es la interfaz usuario Arduino. Se pensó en esta versus otras pues incorporaba botones en su interfaz y estaba en el mercado chileno. Tiene 6 botones: UP, DOWN, LEFT, RIGHT y RESET los cuales posteriormente indicarán su uso, y también tiene una pantalla alfanumérica de 16x2. Para su conexión utiliza 7 pins digitales y 1 análogo.

EC SENSOR

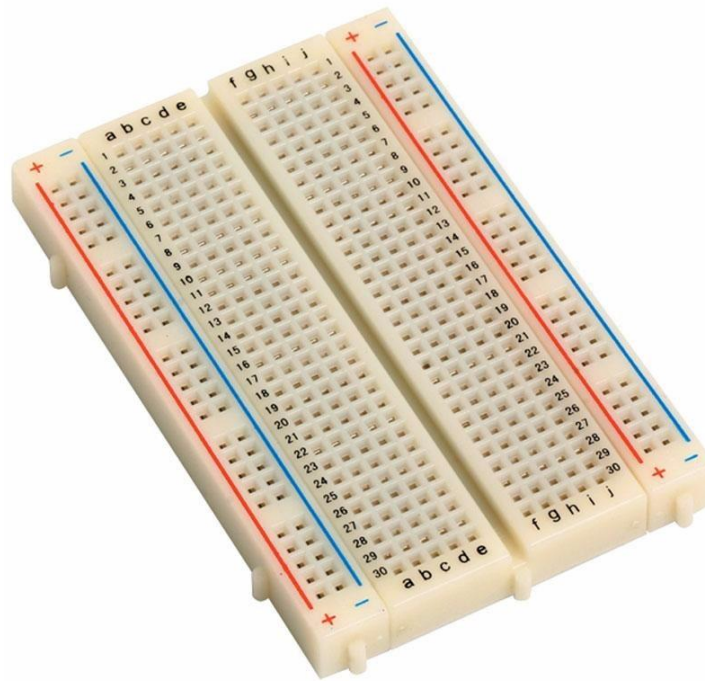


Este sensor se consiguió por EBAY pues en el mercado chileno no existían en el momento de la compra, pues es un producto muy específico. Este mide entre 0 y 5000 $\mu\text{S}/\text{Cm}$, lo cual (según esta fuente http://www.homehydrosystems.com/ph_tds_ppm/ph_vegetables_page.html) sirve para medir una basta variedad de vegetales. Especificaciones técnicas a continuación. Utiliza 1 pin análogo.

Specifications

Power : 5vdc with power LED
Range : 0-5 EC or (0-5000 micro/S)
Accuracy: .2% error .Works perfect from 0-4 EC, Starts losing accuracy after 4000 micro/S due to power limitations.
Temperature compensation: Yes
EC electrode: k=1 , submersible/ inline with 1 meter cable
Size : 3 " x 2" ABS enclosure

Protoboard:



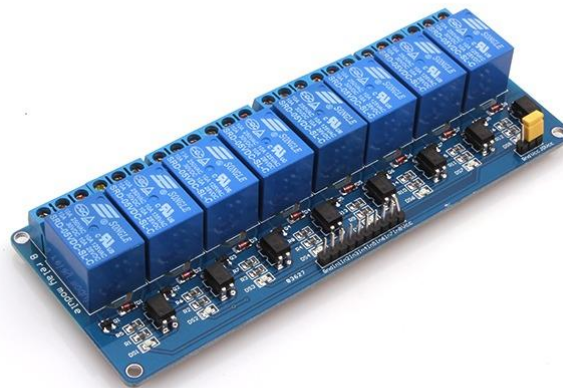
Esta es necesaria para hacer las conexiones de forma simple y no definitiva, ideal para prototipo.

Jumper Wires



Estos sirven para hacer las conexiones de forma rápida pues tienen conectores. Existen formato macho-macho, macho-hembra y hembra-hembra según necesidades.

MODULO RELES 8CH OPTO-AISLADOS 5Vdc



Características:

- 8 Canales de relés NC/NA (250Vac-10A/30Vdc-10A)
- Opto-Aislación (De esta manera protejes tu etapa de control)
- Entrada : 5 Vdc -15 a 20 ma (Led indicador de control)
- Interface estandar para multiples aplicaciones (PIC-AVR-ARM-DSP-ARDUINO)
- Medidas : 13.9cm x 5.2cm x 1.7cm

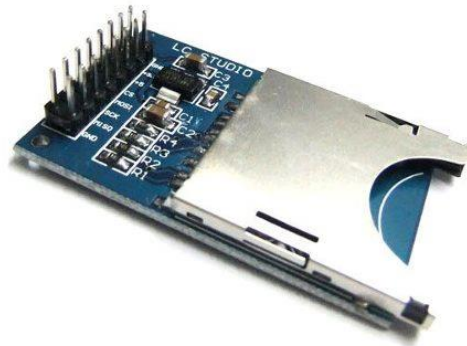
Este sirve para dar corriente a los controladores, por lo que permite prender y apagar todos los aparatos que se conecten.

Utiliza 1 pin digital por relé.

Modulo Lector De Sd Para Arduino Arm

- All SD SPI pins output, MOSI, SCK, MISO and CS.
- Support 5V/3.3V input.

- Size:5.1cm x 3.1cm - 2.01inch x 1.22inch.



Con este permitimos que el usuario introduzca sus recetas y cualquier otra información que pueda utilizar el controlador. Puede servir para logear la toma de datos en un futuro. Utiliza 2 pins de comunicación.

Modulo Mini Rtc Reloj De Tiempo Real Ds1307



Este sirve para tener por fuente externa la hora, lo cual sirve para formar el sistema que sea a prueba de cortes de corriente, pues se programa en función de la hora y el día. Se conecta por 2 pins especiales para la hora en arduino.

Bombas peristálticas



Porque no tienen que estar en contacto con el líquido, su bajo caudal y gran precisión, estas fueron las elegidas para mover los fertilizantes a la reserva de agua.

Specification:

Power supply: DC 12V

Working environment: temperature 0-40 °C, relative humidity <80%

Medium flow: the flow of the peristaltic pump connection depends on the power level of positive and negative, positive and negative terminals can be arbitrarily chosen level

Flow range :0-100 ml / min

Speed range :0.1-100 rpm

Motor RPM:5000RPM

Drive Size (Φx H): Diameter 27.6 x height 37.9 (mm)

Head size (Φx H): Diameter 31.7 x height 20.1 (mm)

Equipped with pump tube (IDxOD): 2.5 x outer diameter of the inner diameter of 4.7 (mm)

Weight: 89G

Type pump head designed for easy disassembly, easy to clean and maintain.

Válvula Solenoide



Features

- Normally closed
- Durable
- Stainless steel inlet filter
- Wide operation temperature range

Specifications

- Working voltage: 12VDC
- Max current: 450mA
- Operating mode: normally closed
- Inlet/Outlet ports: G1/2"
- Materials: nylon/ stainless steel/Polyoxymethylene
- Valve type: diaphragm valve (operated by servo)
- Filter screen: stainless steel inlet filter
- Max operating temperature: 120°C
- Operating pressure range: 0.02 – 0.8 MPa
- Expected life: 200,000+

Esta válvula es la responsable de dar paso al agua y llenar la reserva de agua.

Enchufes: Se utilizaron 2 enchufes comunes para el montaje del prototipo.

Caja Plástica: Se utilizó una caja plástica de origen chileno para el montaje.

ANEXO III

“Datos Individuales de Medición de Experimento”

Tabla Valores Obtenidos de EC [μ S] Experimento A:

1000	999	999	1000	1000	999	998	999	998	998	998	998	998	999	999	1000	999	998	997	998
998	998	998	997	996	997	996	996	997	997	997	998	998	998	999	998	998	999	1000	1000
1000	1001	1001	1000	1000	1001	1002	1002	1002	1003	1003	1003	1004	1004	1004	1004	1004	1002	1003	1002
1003	1004	1005	1005	1005	1005	1005	1005	1005	1006	1006	1006	1006	1005	1006	1007	1006	1006	1005	1005
1006	1006	1006	1005	1005	1005	1005	1005	1004	1005	1005	1006	1007	1007	1008	1007	1007	1007	1007	1007
1007	1009	1008	1009	1008	1008	1008	1008	1008	1008	1008	1007	1008	1008	1008	1008	1009	1009	1007	1007
1007	1006	1006	1007	1006	1007	1007	1006	1006	1007	1007	1007	1007	1007	1008	1008	1007	1007	1008	1008
1008	1008	1007	1007	1006	1005	1004	1003	1004	1003	1003	1003	1003	1002	1001	1001	999	1000	1000	1000
1000	999	1000	1000	999	998	998	998	997	997	996	996	996	997	996	996	996	997	998	999
999	1000	1001	1001	1002	1002	1001	1000	1000	1001	1002	1003	1002	1003	1002	1000	1000	1001	1000	1000
1001	1000	999	999	1000	1001	1000	1001	1001	1000	999	999	999	998	1000	1000	1001	1002	1002	1001
1001	1001	1002	1001	1001	1001	1003	1002	1001	1003	1002	1001	1000	1000	1000	1000	1000	1001	1001	1001
1000	1001	1002	1003	1003	1003	1002	1002	1003	1003	1003	1004	1003	1003	1003	1003	1004	1005	1005	1005
1004	1002	1002	1002	1001	1000	1000	999	998	998	999	1001	1001	1000	1000	1000	1000	998	999	999
999	999	999	1000	999	999	998	998	998	998	997	999	999	998	999	1000	999	999	999	999
998	998	997	997	997	998	997	997	998	998	998	998	999	1000	1000	1000	999	998	998	998
998	998	997	998	998	997	998	998	1000	1000	1000	1001	1000	1000	1001	1001	999	999	999	999
1000	999	999	999	999	998	999	999	1000	1000	999	998	998	998	997	997	997	997	995	995
996	996	996	996	996	995	994	995	995	994	994	995	994	995	993	994	995	994	994	996
995	996	995	995	997	997	997	995	996	997	996	996	996	995	995	995	996	996	995	995
995	995	996	997	996	996	995	995	994	995	994	994	995	995	995	995	993	993	994	992
992	991	991	990	988	988	987	987	987	987	988	990	989	988	989	989	989	989	989	989
988	989	987	987	986	987	986	986	986	986	985	985	985	985	984	983	983	985	985	987
986	986	987	987	988	989	989	988	989	990	991	991	990	989	989	989	989	989	989	989
989	988	987	986	987	988	987	987	987	987	987	988	988	988	988	988	986	986	988	987
989	988	987	987	988	988	986	986	987	987	987	988	989	988	988	988	988	988	986	986
987	986	986	986	987	987	987	987	987	987	988	989	988	988	988	988	989	989	988	987
988	988	989	990	990	989	988	988	989	990	990	992	991	991	991	992	992	990	991	991
990	990	991	990	990	990	989	989	988	988	988	989	988	987	989	987	986	987	986	987
987	987	988	987	988	989	989	989	990	990	989	989	990	991	992	993	991	991	990	990
990	990	991	990	991	990	989	989	989	990	991	991	990	989	990	989	989	989	990	990
992	992	994	994	994	994	993	993	994	994	993	992	992	991	992	991	991	992	991	991
991	991	991	991	991	990	990	990	989	990	989	989	990	990	988	990	991	992	992	992
992	992	991	990	990	989	989	992	991	992	992	992	992	991	991	991	990	990	989	989
987	988	988	989	989	989	988	988	988	987	987	986	986	986	987	986	986	987	986	986
987	985	984	985	985	984	984	984	984	984	984	985	986	988	987	987	987	988	988	988
991	991	992	992	991	991	989	991	991	991	992	992	992	994	994	995	996	996	996	997
995	995	993	994	995	995	995	996	996	994	994	994	994	994	996	996	995	995	995	995
995	995	994	994	993	994	993	995	994	994	995	993	994	994	995	995	995	996	995	996
997	998	998	998	997	997	999	1001	1000	1000	999	1001	1001	1002	1003	1003	1003	1003	1003	1003
1005	1005	1004	1003	1002	1003	1002	1003	1001	1002	1001	1002	1003	1003	1003	1004	1004	1004	1005	1005
1004	1002	1001	1003	1003	1003	1003	1003	1002	1002	1002	1001	1001	1000	998	999	998	997	997	997

997	998	998	998	998	998	998	997	998	998	998	997	998	997	997	996	997	999	1000	
999	1000	1000	1001	1000	1000	1001	1001	1001	1001	1001	1000	1001	1002	1002	1002	1002	1002	1003	1003
1002	1001	1001	1002	1002	1003	1005	1006	1006	1005	1006	1004	1004	1004	1002	1002	1003	1002	1003	1003
1002	1003	1003	1003	1002	1004	1003	1004	1004	1003	1004	1004	1003	1002	1002	1002	1002	1001	1002	1002
1000	999	1000	1001	1001	1000	1001	999	999	999	1000	1000	1000	999	999	999	998	997	997	997
996	998	997	999	1000	1000	998	999	998	999	999	1000	1002	1003	1002	1003	1002	1001	1001	1001
1001	1001	1000	1001	1000	998	999	1000	1001	1000	999	999	997	997	998	998	998	998	998	998
998	999	997	995	995	995	995	996	996	997	997	997	997	996	996	996	995	996	995	995
996	995	994	994	996	995	995	995	995	995	995	994	995	994	995	995	994	995	996	995
994	994	995	995	997	997	997	996	996	996	996	995	995	995	995	995	995	996	997	997
998	998	997	998	998	998	998	998	996	997	997	996	996	996	996	996	996	997	996	995
994	994	995	995	996	995	995	994	995	995	995	995	994	996	995	995	995	995	995	995
995	995	995	997	996	996	995	996	997	997	996	997	997	998	999	1000	1000	1001	1001	1002
1003	1002	1002	1002	1002	1002	1001	1001	1001	1001	1001	1000	999	999	998	997	996	996	998	998
997	998	999	998	999	998	998	998	998	998	998	998	997	996	996	996	996	997	996	996
997	997	997	999	997	997	996	996	995	995	995	995	995	995	995	994	994	994	994	994
993	992	992	991	991	992	992	990	991	991	990	991	991	992	990	991	992	992	992	991
991	991	993	993	995	995	994	993	992	992	992	992	993	993	994	993	992	993	993	993
993	993	993	994	995	994	993	993	992	992	992	992	993	994	994	995	994	993	993	993
993	993	993	994	993	994	995	994	993	994	995	996	995	995	994	994	994	994	993	993
994	994	995	994	995	994	995	995	994	993	992	992	993	992	992	993	993	993	994	996
996	996	997	996	995	995	995	995	994	994	994	994	994	994	994	994	995	994	995	995
996	996	996	996	995	995	995	996	997	999	999	1000	999	1001	1000	1000	1000	1000	1000	1001
1002	1000	1001	1001	1001	1002	1002	1002	1001	1000	1000	1001	1001	1000	999	999	999	999	999	999
998	998	998	999	999	999	1001	1000	1001	1000	1001	1003	1001	1001	1002	1002	1003	1001	1001	1001
1002	1001	1001	1001	1000	1000	999	998	998	999	999	999	998	998	997	998	997	997	998	998
999	1000	1001	1002	1002	1000	1000	1000	1001	1000	1001	1002	1003	1002	1003	1002	1002	1002	1003	1004
1003	1002	1002	1003	1002	1004	1004	1003	1002	1001	1001	1001	1001	1000	1000	1000	1001	1001	1002	1002
1001	1001	1000	999	1000	999	998	999	1000	1001	1000	999	999	999	998	998	998	997	998	997
997	996	995	996	996	996	996	996	998	998	998	998	997	998	997	998	997	997	997	997
996	996	996	996	995	995	995	996	997	997	998	997	999	998	998	998	997	997	997	998
997	997	997	998	998	997	997	998	997	997	998	1000	1000	998	998	999	1000	1000	1000	1000
1001	1001	1000	1001	1001	1000	1000	999	999	1000	1000	1000	1000	1000	1000	1000	1000	999	1000	1001
1002	1001	1004	1004	1004	1005	1004	1004	1005	1005	1006	1007	1008	1009	1010	1009	1009	1009	1010	1010
1010	1010	1009	1008	1008	1009	1010	1008	1007	1007	1006	1006	1006	1007	1006	1006	1007	1007	1008	1007
1007	1007	1008	1009	1008	1009	1008	1008	1009	1010	1009	1009	1008	1008	1007	1006	1006	1004	1004	1004
1005	1005	1005	1006	1005	1005	1004	1002	1002	1003	1004	1005	1005	1006	1005	1005	1004	1004	1004	1005
1006	1005	1004	1004	1005	1005	1004	1004	1004	1005	1005	1005	1005	1005	1005	1005	1004	1003	1002	1002
1001	1001	1002	1002	1001	1001	1002	1002	1002	1002	1002	1002	1003	1004	1003	1003	1002	1003	1003	1003
1002	1001	1001	1001	1000	1001	1001	1001	1001	1000	1001	1001	1000	1000	1000	1001	1001	1001	1002	1002
1002	1004	1005	1006	1005	1005	1004	1004	1004	1003	1003	1002	1003	1002	1001	999	999	998	998	998
998	998	998	998	1000	1000	1000	1000	1002	1002	1001	1002	1003	1003	1002	1003	1003	1001	1000	1000
1000	1000	999	998	998	998	998	999	998	998	998	998	999	1000	1000	1000	999	1000	999	998

997	998	998	998	999	999	999	1000	1001	1003	1003	1003	1004	1004	1004	1004	1004	1004	1005	1005
1005	1005	1006	1005	1005	1007	1008	1009	1010	1010	1010	1011	1011	1011	1011	1012	1012	1012	1012	1012
1013	1013	1013	1014	1014	1013	1013	1011	1010	1010	1010	1010	1010	1010	1009	1010	1010	1011	1011	1011
1012	1013	1014	1014	1014	1015	1016	1017	1017	1017	1015	1015	1015	1016	1015	1014	1015	1014	1015	1017
1018	1017	1018	1018	1017	1018	1018	1018	1019	1019	1019	1021	1020	1019	1020	1019	1020	1019	1018	1018
1019	1021	1021	1021	1019	1019	1019	1019	1019	1020	1020	1020	1021	1021	1021	1020	1019	1018	1019	1018
1016	1017	1018	1016	1016	1015	1015	1015	1014	1015	1014	1014	1014	1016	1015	1015	1017	1017	1017	1017
1017	1015	1015	1015	1016	1015	1015	1016	1015	1016	1017	1016	1015	1015	1014	1013	1014	1013	1014	1014
1014	1014	1015	1015	1016	1016	1015	1015	1015	1017	1016	1015	1015	1015	1015	1016	1015	1015	1015	1015
1015	1015	1015	1014	1014	1015	1014	1012	1012	1012	1013	1012	1011	1012	1012	1012	1012	1012	1013	1013
1014	1015	1016	1016	1017	1017	1018	1017	1016	1017	1016	1015	1016	1017	1017	1017	1018	1020	1020	1020
1020	1020	1020	1019	1019	1019	1020	1019	1019	1017	1017	1016	1016	1017	1017	1017	1017	1016	1016	1016
1017	1016	1017	1018	1018	1018	1017	1017	1017	1016	1016	1017	1018	1017	1017	1017	1016	1016	1015	1015
1016	1016	1015	1015	1014	1015	1018	1017	1017	1015	1016	1015	1016	1016	1016	1016	1015	1014	1014	1014
1015	1016	1016	1015	1016	1015	1014	1014	1014	1016	1015	1014	1015	1015	1015	1015	1015	1014	1014	1014

Tabla de Frecuencias Experimento A:

Valores EC [μ S]	Frecuencia
983	2
984	8
985	10
986	27
987	47
988	50
989	54
990	42
991	50
992	48
993	42
994	78
995	123
996	95
997	102
998	148
999	112
1000	140
1001	125
1002	103
1003	82
1004	58
1005	59
1006	33
1007	36

1008	31
1009	17
1010	18
1011	9
1012	15
1013	11
1014	32
1015	57
1016	40
1017	38
1018	18
1019	19
1020	14
1021	7

Tabla Valores Obtenidos de EC [μ S] Experimento B:

999	999	999	1000	1002	1004	1004	1006	1008	1007	1007	1007	1008	1007	1007	1007	1007	1008	1007	1008
1008	1008	1008	1007	1007	1009	1010	1012	1011	1011	1010	1011	1010	1010	1010	1009	1010	1010	1009	1009
1009	1010	1010	1010	1010	1009	1008	1008	1008	1007	1008	1008	1008	1008	1007	1008	1008	1009	1007	1007
1007	1006	1005	1006	1006	1006	1007	1007	1007	1007	1006	1007	1007	1006	1005	1005	1007	1006	1005	1005
1006	1007	1006	1006	1007	1007	1007	1008	1008	1007	1008	1007	1006	1005	1004	1004	1004	1005	1004	1005
1006	1004	1005	1006	1006	1007	1008	1008	1007	1007	1007	1007	1008	1007	1007	1008	1008	1008	1008	1008
1006	1008	1008	1009	1010	1009	1008	1009	1009	1009	1008	1008	1008	1007	1008	1006	1006	1005	1007	1006
1007	1006	1004	1003	1002	1003	1003	1004	1005	1005	1005	1005	1004	1004	1004	1004	1003	1004	1003	1003
1003	1003	1004	1004	1003	1003	1003	1002	1002	1003	1001	1001	1001	1001	1001	1001	1001	1001	1002	1002
1003	1005	1005	1005	1005	1005	1005	1006	1006	1007	1006	1006	1006	1007	1008	1008	1008	1009	1009	1009
1009	1008	1007	1006	1007	1007	1004	1003	1003	1002	1002	1001	1001	1001	1001	1001	1000	1001	1001	1001
1001	1004	1003	1003	1004	1002	1004	1003	1004	1004	1005	1005	1005	1005	1004	1004	1003	1003	1003	1003
1001	1001	1001	1001	1001	1002	1002	1001	1000	998	999	999	999	1000	1000	1000	999	1000	1000	1000
1000	1000	1000	999	1001	1000	1001	1002	1002	1001	1002	1002	1001	1002	1001	1002	1001	1001	1001	1001
1002	1003	1003	1002	1002	1002	1001	1000	999	998	997	998	998	999	999	999	1000	999	999	999
999	999	998	997	997	997	996	997	998	998	999	998	999	998	998	999	999	998	999	999
1000	1002	1001	1001	1002	1002	1002	1002	1002	1002	1002	1003	1001	1002	1002	1002	1003	1003	1002	1001
1000	1000	1001	1002	1001	1000	1000	1000	1001	999	999	999	1000	1000	1000	1000	999	999	1000	1000
1000	999	1000	999	997	997	998	998	998	998	999	999	999	1000	999	1000	1000	1000	1000	1000
999	999	1001	1001	1001	1001	1002	1002	1002	1002	1002	1003	1002	1003	1003	1004	1003	1003	1002	1002
1002	1003	1002	1004	1003	1003	1001	1002	1001	1001	1001	1001	1001	1001	1001	1002	1001	1001	1002	1002
1000	999	999	999	998	998	999	999	999	998	999	999	1000	1000	1001	1002	1002	1000	1000	1000
1001	1002	1003	1002	1002	999	1000	1000	1001	1000	999	999	999	998	997	997	997	998	997	998
997	997	997	997	997	996	996	997	997	997	996	997	997	998	997	997	999	998	997	997
997	998	998	998	998	999	999	999	998	999	999	998	999	999	998	998	999	999	999	999
998	999	1000	999	999	999	1000	998	999	999	999	999	999	998	997	998	997	998	999	999
998	997	997	996	995	994	994	994	994	994	996	995	995	995	995	996	995	996	996	996

996	995	996	995	996	995	996	996	996	995	996	996	996	997	997	996	995	994	993	993
993	993	993	993	994	993	994	992	992	991	991	992	992	991	990	991	991	992	991	991
991	991	991	991	990	990	989	991	991	989	989	990	992	991	991	990	992	991	991	991
991	992	991	991	991	993	992	991	991	990	989	989	988	989	989	987	989	989	989	988
990	991	990	991	992	992	992	991	991	991	990	990	991	992	991	991	991	993	994	994
994	994	994	993	993	994	993	993	993	993	993	995	994	994	995	994	994	994	995	995
996	994	994	993	992	992	993	992	992	991	991	991	991	991	991	993	993	995	995	995
994	993	993	993	992	989	989	990	989	988	990	989	988	989	989	990	990	990	988	989
989	988	988	988	987	986	987	988	988	988	988	988	990	990	991	992	991	991	991	991
991	991	992	992	992	993	993	994	995	995	995	995	995	995	995	996	997	997	998	999
998	999	999	998	998	998	998	999	998	997	997	996	996	995	997	997	997	997	998	998
999	998	998	997	998	997	997	997	998	997	996	995	996	996	996	995	996	996	996	997
996	995	995	994	992	992	992	993	991	992	991	990	990	990	991	991	991	991	992	992
992	991	991	991	990	990	990	990	989	989	989	990	991	992	992	992	993	993	992	992
992	990	992	992	991	991	991	991	989	989	989	988	988	988	988	988	989	989	988	988
986	987	987	988	987	987	986	987	987	986	986	986	985	984	985	986	986	986	987	986
986	986	985	987	986	988	988	986	986	987	986	988	987	988	988	987	987	986	986	986
985	985	985	984	984	984	984	984	984	985	986	987	986	985	986	986	986	986	985	986
985	986	986	985	985	985	984	985	985	987	986	987	987	988	990	989	990	990	989	989
989	990	990	990	990	988	988	988	988	986	986	985	985	986	987	987	988	987	987	986
986	985	985	985	984	985	986	985	986	985	986	986	985	986	985	984	984	984	984	984
983	983	983	982	980	980	980	979	979	980	979	980	979	980	980	980	980	980	979	979
979	980	980	981	982	983	983	982	982	984	982	982	982	982	982	982	980	980	981	980
984	983	981	981	980	981	983	983	984	983	982	982	982	982	982	980	980	980	981	980
979	978	978	979	979	978	978	978	979	978	978	979	978	978	979	979	980	979	979	979
980	981	982	982	981	980	981	980	981	980	981	979	980	981	980	981	982	982	981	982
982	981	981	981	982	982	982	982	982	982	981	981	981	982	981	980	980	980	981	981
981	980	981	981	981	980	980	979	979	978	978	979	979	981	980	980	979	978	977	979
979	979	979	980	978	979	980	980	981	980	980	980	980	980	980	981	981	982	979	979
980	982	983	983	984	984	984	984	984	984	984	985	985	985	985	985	986	985	985	986
984	984	984	984	984	985	983	983	983	984	985	986	985	985	985	986	985	984	984	984
984	986	986	985	985	985	986	986	984	985	986	985	985	985	985	986	986	985	985	985
984	982	983	983	983	982	983	983	983	984	984	984	983	985	986	987	987	987	987	987
986	987	987	987	988	989	988	987	987	986	985	987	987	987	987	987	986	988	987	987
987	988	988	988	987	987	987	987	987	987	986	986	986	986	985	983	983	982	981	981
981	981	982	982	982	982	983	983	982	983	983	983	982	982	982	982	982	981	981	979
979	979	979	980	980	981	979	978	978	978	978	978	979	980	980	982	983	983	982	982
982	983	984	985	985	984	984	983	983	984	985	985	985	986	988	990	990	990	990	989
988	990	990	989	988	987	986	986	985	985	984	985	986	986	986	986	986	986	985	985
985	985	984	985	986	987	988	989	987	986	984	986	987	988	988	987	987	987	987	987
989	989	989	988	988	989	987	988	988	988	988	988	988	987	987	987	987	988	989	989
989	987	986	987	987	987	986	986	987	986	986	986	985	987	989	989	989	989	990	990
992	993	994	995	995	994	995	994	993	993	993	993	992	992	992	993	993	993	994	994

994	994	994	993	994	995	995	995	994	995	994	995	993	992	991	990	990	989	989	990
990	989	991	990	989	990	988	989	988	989	989	990	989	991	991	991	992	991	991	991
992	991	990	990	990	992	992	991	991	991	991	991	991	991	991	990	992	992	992	993
994	993	994	994	994	996	995	997	997	997	998	998	998	998	997	998	999	998	996	996
997	998	997	998	998	998	998	999	998	999	999	999	999	999	999	999	999	997	998	997
997	997	997	998	999	998	998	1000	1001	1001	1001	1003	1003	1003	1003	1003	1002	1002	1002	1001
1001	1001	1003	1003	1003	1003	1003	1002	1001	1001	1001	1001	1001	1001	999	1000	1000	1000	999	997
997	995	997	996	996	996	996	995	996	997	997	998	998	999	998	999	999	1001	1001	1001
1000	1001	1001	1000	1001	1001	1002	1001	1002	1002	1002	1003	1003	1005	1006	1005	1006	1005	1005	1005
1005	1003	1002	1002	1001	1002	1004	1004	1003	1003	1004	1004	1003	1002	1002	1001	999	999	998	998
998	997	995	996	998	998	998	999	999	998	997	997	998	996	996	997	998	998	998	1000
1001	1001	1001	1002	1002	1001	1002	1003	1003	1003	1004	1004	1005	1004	1004	1004	1004	1004	1002	1002
1002	1001	1000	1000	1001	1000	1000	1000	999	998	999	999	1000	1000	1000	1001	1000	999	1001	999
998	998	999	1001	1000	1000	1001	1001	1001	1000	1001	1001	1001	1001	1001	1001	1003	1003	1003	1003
1003	1002	1002	1003	1003	1005	1006	1006	1005	1005	1006	1005	1004	1006	1006	1005	1005	1006	1005	1005
1004	1003	1004	1003	1003	1003	1004	1005	1005	1005	1005	1005	1005	1004	1006	1006	1005	1005	1006	1006
1007	1006	1006	1007	1008	1009	1008	1009	1007	1007	1006	1006	1006	1007	1007	1007	1008	1009	1008	1008
1008	1009	1009	1010	1009	1009	1009	1009	1010	1010	1008	1008	1009	1007	1008	1008	1008	1008	1007	1007
1008	1007	1008	1006	1005	1005	1005	1004	1005	1007	1006	1006	1007	1006	1007	1008	1009	1009	1009	1008
1010	1011	1012	1012	1011	1012	1012	1011	1012	1012	1012	1012	1012	1012	1012	1013	1013	1012	1013	1012
1012	1012	1013	1013	1015	1016	1015	1014	1014	1016	1016	1016	1016	1017	1016	1015	1016	1015	1015	1016
1017	1018	1016	1015	1015	1015	1014	1015	1013	1014	1014	1014	1015	1014	1014	1014	1016	1016	1014	1014
1013	1015	1014	1013	1014	1013	1014	1014	1013	1013	1013	1012	1012	1013	1014	1014	1014	1015	1015	1014
1014	1014	1014	1015	1015	1015	1015	1016	1016	1015	1015	1015	1015	1015	1014	1014	1015	1017	1017	1017
1017	1016	1018	1019	1019	1018	1021	1020	1020	1019	1020	1020	1018	1018	1020	1020	1020	1019	1017	1016
1016	1015	1015	1015	1015	1015	1015	1014	1016	1016	1016	1016	1016	1016	1017	1016	1016	1017	1018	1019
1019	1019	1020	1019	1020	1018	1018	1018	1017	1017	1017	1016	1016	1016	1014	1012	1013	1013	1013	1013
1012	1012	1012	1012	1013	1012	1013	1014	1014	1015	1015	1014	1015	1015	1017	1017	1016	1017	1016	1016
1014	1015	1016	1016	1015	1015	1016	1015	1015	1015	1015	1015	1014	1014	1013	1013	1013	1014	1013	1013
1013	1013	1013	1014	1014	1014	1015	1016	1016	1016	1017	1016	1017	1017	1016	1017	1016	1016	1016	1016

Tabla de Frecuencias Experimento B:

Valores EC [μ S]	Frecuencia
977	1
978	18
979	36
980	46
981	37
982	46
983	35
984	49

985	66
986	76
987	67
988	55
989	53
990	52
991	77
992	49
993	40
994	38
995	42
996	44
997	67
998	83
999	102
1000	66
1001	95
1002	74
1003	66
1004	43
1005	51
1006	47
1007	57
1008	54
1009	29
1010	16
1011	6
1012	24
1013	27
1014	36
1015	42
1016	41
1017	19
1018	9
1019	9
1020	9
1021	1