



TESIS DE GRADO EN  
INGENIERIA ELECTRONICA

MENÚ TÁCTIL PARA  
RESTAURANTES

Autores:	Legajos:
Alonso, Matías	48296
Chen, David	48695
Villa Lebrún, Gonzalo	48273

Profesores de la Materia

Ing. Muñoz Claudio  
Ing. Pingitore Ricardo

AÑO 2013

## I. PREFACIO Y AGRADECIMIENTOS

La decisión de desarrollar un menú inalámbrico surgió de entre varias opciones sugeridas por cada uno de los miembros del grupo. Una versión propia de la aspiradora Roomba y una silla de ruedas con electrónica incorporada para mediciones médicas y control, entre otras, fueron parte de las múltiples alternativas que se consideraron.

Finalmente, la opción que surgió elegida entre las demás fue la del menú digital para restaurantes. La razón de dicha elección fue que era la que más balanceaba conocimientos adquiridos a lo largo de toda la carrera.

Sin embargo, a medida que avanzábamos con el proyecto, comprobamos que, como futuros jóvenes profesionales, había multitud de aptitudes que sólo podían adquirirse llevando a cabo tareas que escaparan a una simple actividad académica. No obstante, fue al darnos cuenta de ello que descubrimos la principal cualidad que nos había transmitido la institución, y era la capacidad de aprender. Nuevos programas de diseño, nuevos componentes, nueva bibliografía y, sobre todo, mucha investigación, permitieron llevar adelante el proyecto de una manera eficiente e ingenieril, tal como la facultad nos enseñó.

Por esto, no podemos dejar de agradecer a todos aquellos que, desde lo académico o lo personal, nos han acompañado a lo largo de este arduo camino. Dentro del ámbito académico, queremos agradecer a Daniel Jacoby, Jorge Cáceres, la cátedra de Diseño y la cátedra de Taller de Modelización de Productos (en particular a su titular Jorge Leporati), que colaboraron en distintos aspectos de la realización de este proyecto. También queremos agradecer a las empresas Apexar y Texas Instruments, cuyas desinteresadas colaboraciones permitieron llevar adelante la construcción del prototipo. Además, queremos mencionar la ayuda brindada por compañeros y ex-compañeros, ahora egresados o a punto de estarlo, cuya experiencia en ciertos campos nos allanó el camino para el desarrollo de la tesis. Estos alumnos son Lucas Ruiz Díaz, Facundo Costa, Lien Tori y Pablo Carranza Vélez.

Queremos hacer llegar, también, nuestro más sincero agradecimiento a la facultad en general, y a todo el personal que dedica sus días a ayudarnos a ser exitosos en la carrera que elegimos.

Finalmente, agradecemos a nuestras familias, en particular a nuestros padres, dado que sin su apoyo nada de todo lo que hemos logrado hubiera sido posible. Ellos nos brindaron la dosis de fe y motivación que tanto nos impulsó a atravesar momentos difíciles y por eso deseamos expresarles nuestra gratitud.

II. ÍNDICE	
I. PREFACIO Y AGRADECIMIENTOS .....	2
II. ÍNDICE .....	3
III. RESUMEN Y ABSTRACT .....	5
1. Resumen .....	5
2. Abstract .....	5
IV. INTRODUCCIÓN .....	6
1. Historia .....	6
a. Historia de la implementación de menús en formato carta .....	6
b. Historia de las pantallas táctiles .....	6
2. Glosario de términos utilizados .....	8
3. Justificación del proyecto .....	11
V. OBJETIVOS .....	12
1. Finalidad del proyecto .....	12
2. Planteamiento del problema a resolver .....	12
VI. DEFINICIÓN DE PRODUCTO .....	13
1. Requerimientos técnicos .....	13
2. Definición de las especificaciones .....	14
a. Especificaciones del Hardware .....	14
b. Especificaciones de Software .....	14
3. Casa de calidad (QFD) .....	15
VII. ANÁLISIS DE FACTIBILIDAD .....	17
1. Factibilidad tecnológica .....	17
a. Propuestas de alternativas de diseño .....	17
b. Elección de una solución .....	17
c. DFMEA .....	18
2. Factibilidad de tiempos .....	27
a. Planificación – Análisis PERT .....	29
b. Programación – Diagrama de Gantt .....	34
c. Simulación de Montecarlo .....	39
3. Factibilidad económica .....	42
a. Análisis de Mercado .....	42
a.1. Unidades Vendidas .....	42
a.2. Análisis de la competencia .....	42
b. Costos .....	43
b.1. Proveedores .....	43
c. Análisis de gastos .....	44
c.1. Inversión inicial .....	44
c.2. Costos fijos .....	44
c.3. Costos variables .....	45
d. Análisis económico financiero .....	46
e. Resultados .....	47
4. Factibilidad legal y responsabilidad civil .....	48
VIII. INGENIERÍA DE DETALLE .....	52

1. Hardware.....	52
a. Diagrama de bloques del hardware.....	52
b. Descripción detallada de cada bloque .....	53
c. Plan de pruebas de cada módulo.....	66
2. Software .....	68
a. Diagrama de estados y flujogramas.....	69
b. Análisis de complejidad .....	73
c. Descripción de subrutinas:.....	74
d. Listados comentados del código.....	80
e. Plan de prueba de módulos y de depuración de software .....	80
IX. CONSTRUCCIÓN DEL PROTOTIPO .....	82
1. DEFINICIÓN DE LOS MÓDULOS .....	82
2. DISEÑO DE LOS CIRCUITOS IMPRESOS .....	82
3. DISEÑO MECÁNICO.....	84
4. DETALLES DE CONSTRUCCIÓN Y PRECAUCIONES ESPECIALES DE MONTAJE.....	88
X. Validación del Prototipo.....	90
1. VALIDACIÓN DEL HARD .....	90
1.1. Plan y protocolos especiales de medición .....	90
1.2. Mediciones .....	90
1.3. Evaluación.....	92
1.4. Resultados .....	93
11.2 Validación de software .....	93
XI. ESTUDIO DE CONFIABILIDAD .....	97
1. Estudio de confiabilidad del hardware .....	97
2. Estudio de confiabilidad de software .....	104
XII. CONCLUSIONES.....	106
Excelencias. Objetivos alcanzados.....	106
Fallos. Recomendaciones para futuros diseños.....	106
Pensamientos finales .....	106
XIII. ANEXOS.....	108
1. ANEXO I: ESTUDIO DEL MERCADO MEDIANTE ENCUESTAS .....	108
a. Encuestas originales .....	108
b. Encuestas online.....	111
2. ANEXO II: LOG DE INICIALIZACIÓN DEL KERNEL DE LINUX .....	117
3. ANEXO III: CALIBRACIÓN DE LA MEMBRANA TÁCTIL RESISTIVA .....	122
4. ANEXO IV: DETALLE DEL CÁLCULO DE CONFIABILIDAD CON $G_F$ : .....	123
XIV. BIBLIOGRAFIA .....	127
1. Hoja de datos utilizados .....	127
2. Libros y publicaciones .....	127
3. Notas de aplicación y guías de usuarios.....	127

### III. RESUMEN Y ABSTRACT

#### *1. Resumen*

El presente proyecto consiste en el diseño e implementación de un menú digital que reemplace a las cartas convencionales ofrecidas en restaurantes. El proceso de diseño abarcó aspectos de hardware y software.

Desde el punto de vista del hardware, debió diseñarse la placa que implementaría el menú, teniendo en cuenta aspectos como portabilidad, apariencia, tamaño y disposición de los componentes.

En cuanto al software, se separó su desarrollo en el que se ejecutaría en cada menú portátil y el que se ejecutaría en el servidor del restaurante, manejando la base de datos del local y coordinando todas las unidades portátiles.

Finalmente, se llevó a cabo la implementación del prototipo, logrando un equipo con el desempeño pretendido y un sistema eficiente de control de la base de datos del local.

#### *2. Abstract*

This is a project about the design and implementation of a digital menu to replace the conventional ones offered in restaurants. The design process involved hardware and software aspects.

From a hardware standpoint, the circuit board over the one the menu software should run, had to be designed, taking into account considerations such as portability, appearance, size and arrangement of the components.

The development of the software, on the other hand, was separated in the part that would run on every menu and the part that would run in the restaurant server, handling the local database and coordinating all portable units.

Finally, the prototype was implemented, achieving the intended performance and an efficient control of the local database.

#### IV. INTRODUCCIÓN

##### *1. Historia*

###### *a. Historia de la implementación de menús en formato carta*

Rodrigo Gómez Lencina, chef, señala que en China, durante la dinastía Song (siglo XII), se encuentran los primeros antecedentes históricos sobre la existencia de “cartas” o menús de comida. En particular, Marco Polo cuenta en sus relatos que la capital Hangzhon era un centro avanzado para su época, contando con diez grandes mercados, dedicados a distintos tipos de mercadería. Es así como ocurría que, contando con grandes depósitos de la materia prima en la propia ciudad, se originaban diversos locales que ofrecían alimentos, con distintos perfiles.

Sin embargo, dado que la imagen y el renombre jugaban un papel fundamental en la sociedad oriental de dicha época, dichas casas de comida no podían permitirse ofrecer poca variedad en sus alimentos, dado que eso asociaría conceptos como “escasez”, “miseria”, etc. con el apellido del dueño del local, la cantidad de preparaciones distintas entre las que los clientes podían elegir era muy grande. Por esto, fue imperativo brindar a los clientes “cartas” con los nombres de los alimentos disponibles, a fin de facilitar el proceso de elección de los mismos.

En Europa, sin embargo, recién en el siglo XIV, durante el Medioevo, fue cuando las costumbres gastronómicas previas comenzaron a sufrir algunos cambios. Para empezar, se dividió la comida en dos etapas, lo salado primero y lo dulce después. Cada persona se podía servir, en una etapa, lo que deseara.

Como resultado de la creciente variedad de selección para cada una de estas partes de la comida, aparecieron los “*escriteaux*”, que básicamente eran una lista de ingredientes utilizados al armar una comida. No estaban destinados a clientes, sino a los ayudantes de los cocineros que recibían el pedido de los mozos. En paralelo a estos documentos, en España y otros países de Europa occidental existían (desde incluso tiempo antes) otras listas con especias e ingredientes usados para salsas y aderezos, pero eran básicamente listas de compras, más destinadas a los proveedores que a los chefs.

Un menú más cercano al estilo actual nació a fines del siglo XVIII y comienzos del XIX. La primera referencia del mismo surgió en los restaurantes del Palais Royal de París, que se había establecido como el centro gastronómico de dicha ciudad.

No obstante, a diferencia de los actuales que se pueden ofrecer como “cartas” para cada cliente, éstos eran carteles grandes, similares a las pizarras que pueden verse en restaurantes actuales. Muestras de estos carteles se conservan hasta la actualidad, como por ejemplo un ejemplar del restaurante Rocher de Cancale, así como uno del Hotel de Americains, como muestras conservadas de aquel estilo de presentación. Dichos locales eran líderes en dicha época, evolucionando pronto sus carteles en menús individuales presentados en la mesa de los comensales, como la primera aparición específica del menú como se conoce en la actualidad.

El toque artístico presente en los mismos, incluso hoy, también proviene de esa misma época, por ser consideradas inicialmente como “guías para un buen comer”. Artistas como Toulouse Lautrec<sup>1</sup> pintó menús a cambio de comidas provistas por el local responsable.

###### *b. Historia de las pantallas táctiles*

En la segunda mitad de la década del 60, E. A. Johnson, trabajando para el Royal Radar Establishment en el Reino Unido, desarrolló un sistema para el control de tráfico aéreo mediante dispositivos sensibles al tacto. En 1965 publicó un corto artículo en *Electronic Letters*, con una introducción al proyecto, el cual ampliaría con fotografías y diagramas en un nuevo artículo en 1967. No obstante, en ninguno de dichos artículos mencionó la aplicación con la que había diseñado el sistema. Esto ocurrió recién en 1968, cuando publicó uno nuevo describiendo la utilidad de sus dispositivos en control de tráfico aéreo.

---

<sup>1</sup> Henri Marie Raymond de Toulouse-Lautrec-Montfa, conde de Toulouse-Lautrec-Montfa (1864-1901) - Pintor y cartelista francés que destacó por su representación de la vida nocturna parisense de finales del siglo XIX. Se le enmarca en la generación del postimpresionismo. Es asociado con frecuencia al Moulin Rouge parisino, por haber este establecimiento influenciado parte de su obra.

En 1971, el Dr. Sam Hurst, quien trabajaba para la Universidad de Kentucky, desarrolló un sensor táctil, al que llamó “Elograph”. Este sensor, que fue patentado por la Fundación de Desarrollo de la misma universidad, fue elegido por *Industrial Research* como uno de los 100 productos tecnológicos más significativos de 1973.

En 1974, Sam Hurst y su recientemente fundada “Elographics” crearon la primera pantalla táctil de superficie transparente. Más adelante, en 1977, la misma empresa patentó la tecnología de pantallas táctiles resistivas, la más popular usada hasta la actualidad. Finalmente, en 1977, Elographics (que sería rebautizada en 1994 como “Elo TouchSystems”) patentó una interfaz de sensor de vidrio curvado, el primer producto cuyo nombre comercial incluiría las palabras “pantalla táctil”.

En los siguientes años, diversas compañías lanzarían al mercado, con mayor o menor éxito, distintos productos con tecnología táctil. Por ejemplo, Hewlett-Packard lanzó, en 1983, la HP-150, una computadora personal con sensor táctil basada en rayos infrarrojos superficiales a la pantalla. En 1993, los smartphones y computadoras portátiles empezaron a ser introducidos por empresas líderes en el área. Apple lanzó en Newton PDA, con reconocimiento de escritura manuscrita, e IBM el primer Smartphone, llamado Simon, que incluía una interfaz táctil para marcar números telefónicos. En 1996, Palm liberó la serie Pilot, PDAs con tecnología táctil de avanzada.

En la última década, la tecnología táctil se incorporó a muchísimos productos, consagrándose como la interfaz más común en los dispositivos de uso popular. En 2007, el iPhone de Apple salió al mercado sin otra interfaz que la tecnología táctil.

Hoy en día, donde la minimización y eliminación de periféricos obsoletos van de la mano con el estilo y la vanguardia tecnológica, la tecnología táctil postula para ser el estándar de ingreso de datos por parte del usuario de cualquier dispositivo tecnológico.

## 2. Glosario de términos utilizados

Para facilitar la comprensión del trabajo, se incluyen a continuación algunas expresiones técnicas y su significado.

### *BGA: Ball Grid Array*

Es un tipo de conexión, cuyo fin es unir un componente a la placa base de un equipo por medio de una serie de soldadura con bolitas de estaño. Comúnmente se encuentra en la producción y fijación de microprocesadores, ya que los mismos suelen tener una gran cantidad de terminales, los cuales son soldados a la placa base, evitando la pérdida de frecuencias y aumentando su conductividad.

### *Boost (Step-Up):*

Es un convertidor de potencia de tensión continua tanto a la entrada como a la salida, con un voltaje de salida mayor que el de entrada. Este tipo de convertidores es una categoría de las Fuentes Switching.

### *DCE: Data Communications Equipment*

Es un dispositivo que realiza la comunicación mediante funciones tales como conversión de la señal, codificación, y la línea de sincronización. Puede llegar a ser parte de un Equipo de terminal de datos (DTE), o realizar la comunicación con uno de éstos.

### *DDR: Double Data Rate*

Es un tipo de memoria RAM cuyo bus de operaciones tiene una doble tasa de traspaso de datos al *utilizer* tanto en el flanco ascendente como descendente de la señal del reloj sincronizador.

En nuestro caso se utiliza una memoria DDR2, que adicionalmente a las memorias DDR, presenta una velocidad del canal mayor y requiere menor potencia al utilizar el reloj interno a la mitad de velocidad que el canal de datos.

### *DFMEA: Design Failure Mode Effects Analysis*

Es el Análisis de Modo y Efecto de Falla para un producto en fase de diseño. Este concepto es utilizado en las organizaciones y presenta un método para el control de la producción.

### *DTE: Data Terminal Equipment*

Es un instrumento que convierte la información del usuario en señales, o reconvierte las señales recibidas. Este se comunica con el equipo de comunicación de información (DCE).

### *EMI: Electromagnetic Interference*

Es una señal de emisión radiada en el espacio o conducida a través de un cable que produce interferencia electromagnética y pone en peligro el funcionamiento de la navegación por radio u otros servicios.

### *ESD: Electrostatic Sensitive Device*

Es un componente, principalmente eléctrico, que puede ser dañado por cargas estáticas comunes, que se pueden presentar en personas, herramientas y otros elementos no conductores.

### *Fuente switching:*

Fuente de alimentación conmutada que brinda un gran rendimiento, bajo peso y volumen. Esto último se debe a que se utilizan integrados que contienen la gran parte de los componentes, siendo necesario solo la incorporación de un bobinado y un par de capacitores y resistencias.

### *IEEE: Instituto de Ingenieros Eléctricos y Electrónicos*

Es una asociación técnico-profesional mundial dedicada, entre otros fines, a la estandarización. Con cerca de 400.000 miembros y voluntarios en 180 países, es la mayor asociación internacional sin ánimo de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación, ingenieros en informática, ingenieros en biomédica, ingenieros en telecomunicación e ingenieros en mecatrónica.

*Kernel:*

En computación es el componente principal de la mayoría de los sistemas operativos. Es el que realiza el puente entre las aplicaciones y el procesamiento de datos a nivel del hardware. Entre sus responsabilidades se incluye manejar los recursos del sistema, es decir la comunicación entre hardware y software.

*Layout:*

Es la representación de un circuito integrado en forma planar con los diferentes componentes y sus interconexiones, el cual luego va a ser realizado sobre el PCB.

*LCD: Liquid Crystal Display*

Es una pantalla delgada y plana formada por un número de píxeles en color o monocromos delante de una fuente de luz, utilizando las propiedades modulantes de luz del cristal líquido.

*LDO: Low-DropOut Regulator*

Es un regulador de tensión continua que opera con una pequeña caída de tensión entre su entrada y salida. Las principales ventajas del mismo son su baja tensión de operación, una gran eficiencia y baja temperatura de disipación.

*Máquina Pick-and-Place:*

Máquina automática, usada principalmente en procesos de producciones de dispositivos electrónicos, para realizar la soldadura de elementos de montaje superficial.

*MTBF: Mean Time Between Failures*

Es el promedio del tiempo entre fallas de un sistema estimado para un sistema durante su operación.

*PCB: Printed Circuit Board*

Es un medio para sostener mecánicamente y conectar eléctricamente componentes electrónicos, a través de rutas o pistas de material conductor, grabados en hojas de cobre laminadas sobre un sustrato no conductor, comúnmente baquelita o fibra de vidrio.

*QFD: Quality Function Deployment*

Despliegue de la función calidad (QFD) es un método de gestión de calidad basado en transformar las demandas del usuario en la calidad del diseño, implementar las funciones que aporten más calidad, e implementar métodos para lograr calidad del diseño en subsistemas y componentes, y en última instancia a los elementos específicos del proceso de fabricación.

*RGB: Descripción de un color según las proporciones de Rojo, Verde y Azul que lo componen*

La descripción RGB (del inglés Red-Green-Blue: "Rojo-Verde-Azul") de un color hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: el rojo, el verde y el azul. Es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores luz primarios.

*SMD: Surface-mount device*

Es un dispositivo electrónico hecho con tecnología de montaje superficial, la cual es un método de construcción en el cual los componentes se montan directamente sobre la superficie de la placa del circuito impreso (PCB).

Su principal ventaja es que son generalmente más pequeños que los componentes comunes debido a que sus terminales son más pequeños, si es que los tienen.

*TFT: Thin-Film Transistor*

Thin-film transistor o TFT ("transistor de película fina"), es un tipo especial de transistor de efecto de campo que se fabrica depositando finas películas de un semiconductor activo así como una capa de material dieléctrico y contactos metálicos sobre un sustrato de soporte. Un sustrato muy común es el cristal. Una de las primeras aplicaciones de los TFTs son las pantallas de cristal líquido.

*Timer:*

Es un tipo de reloj específico. Se utiliza para controlar la secuencia de cierto evento o proceso. Una vez que se llega al tiempo establecido por el mismo se ejecuta el evento o proceso vinculado al mismo.

*TIR: Tasa Interna de Retorno*

La tasa interna de retorno o tasa interna de rentabilidad (TIR) de una inversión, está definida como el promedio geométrico de los rendimientos futuros esperados de dicha inversión, y que implica por cierto el supuesto de una oportunidad para "reinvertir". En términos simples en tanto, diversos autores la conceptualizan como la tasa de interés (o la tasa de descuento) con la cual el valor actual neto o valor presente neto (VAN o VPN) es igual a cero. El VAN o VPN es calculado a partir del flujo de caja anual, trasladando todas las cantidades futuras al presente. Es un indicador de la rentabilidad de un proyecto, a mayor TIR, mayor rentabilidad.

*TIRM: Tasa Interna de Retorno Modificada*

La tasa interna de retorno modificada (TIRM) fue diseñada con el fin de superar las deficiencias de la TIR. La TIRM considera en forma explícita la posibilidad de reinvertir los flujos incrementales de fondos del proyecto a una tasa igual al costo de capital de la empresa, a diferencia de la TIR, que supone la reinversión de los flujos a la tasa interna de retorno del proyecto.

Por tal motivo, la TIRM es conocida también como tasa de retorno o recuperación externa.

*VAN: Valor Actual Neto*

El Valor actual neto, también conocido como valor actualizado neto (en inglés, Net Present Value), cuyo acrónimo es VAN (en inglés NPV), es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión. La metodología consiste en descontar al momento actual (es decir, actualizar mediante una tasa) todos los flujos de caja futuros del proyecto. A este valor se le resta la inversión inicial, de tal modo que el valor obtenido es el valor actual neto del proyecto.

*Widget:*

En computación, un widget es un elemento perteneciente a una interfaz gráfica del usuario (GUI) que muestra información disponible y modificable por el usuario, como por ejemplo una ventana o un cuadro de texto. La característica principal de estos elementos es proveer de un único punto de interacción para un determinado tipo de información.

*Widget Toolkit:*

Es un set de widgets para usar en el diseño de aplicaciones para interfaces gráficas de usuario (GUI). El toolkit en sí es una parte del software que se construye sobre un sistema operativo y provee un programa con aplicaciones para programación de interfaces (API), permitiéndole la realización de widgets.

*ZigBee: Protocolo de comunicación inalámbrica*

ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal area network, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

Las principales ventajas de este protocolo son:

- Su bajo consumo.
- Su topología de red en malla.
- Su fácil integración (se pueden fabricar nodos con muy poca electrónica).

### *3. Justificación del proyecto*

El principal motivo de la elección del proyecto actual, es la demanda gradual de productos electrónicos con pantalla táctiles, como por ejemplo: smart-phones y tablets. La pantalla touch-screen presenta muchas ventajas frente a las computadoras con ingreso mediante teclados o mouse y es la razón de su éxito en el siglo XXI. En primera instancia, el touch-screen permite teclados virtuales en lugar de los teclados convencionales, lo cual permite ahorro de espacio y comodidad. Otro concepto que merece ser destacado es la intuitividad. El toque manual sobre íconos o funciones de aplicaciones es más sencillo que el ingreso mediante teclado, eliminando opciones y mostrando las primordiales. Por otro lado, un touch-screen presenta la versatilidad de ofrecer distintos niveles de complejidad. En nuestro caso, el touch-screen menú propone acceso a opciones fundamentales en el servicio gastronómico que normalmente no existen, como el caso de tiempo estimado de espera, control de gasto, gustos personales guardados en memoria, imágenes de la comida, ingredientes, entre otros. También, se suma al producto final la comodidad de la comunicación inalámbrica, que permite notificar a la cocina los pedidos sin demora. Además, la digitalización de los datos permite al dueño del restaurante tomar mejor control y facilitar la documentación que requiera el local.

Un negocio que cuente con pantallas táctiles tiende a ser un punto de venta ágil, ya que el usuario se acostumbra a su manejo y se familiariza rápidamente con la herramienta. En un futuro, el proyecto desarrollado puede extenderse fácilmente a librerías y bibliotecas, cajeros automáticos y museos.

La visión de comunicación lograda por los celulares impulsa al proyecto a apostar a la tendencia creciente de tanto dispositivos inteligentes como del servicio inalámbrico. La renovación de dichos productos electrónicos ayuda a expandir la tecnología, reduciendo precios y ofreciendo productos más innovadores, exhibiendo alta escalabilidad del proyecto y potencial en el mercado.

## V. OBJETIVOS

### *1. Finalidad del proyecto*

El propósito del proyecto es introducir un producto electrónico, tanto hardware como software al ámbito gastronómico que consiste en una pantalla táctil con comunicación inalámbrica, para ayudar tanto al cliente (comensal), como al mozo y al dueño a agilizar de forma inteligente el proceso de compra y servicio. Con este fin, se ofrece el touch-screen menú, como producto nacional con competencia internacional a un costo accesible por debajo del de importación. El proyecto abarcará tanto el desarrollo de software como el de hardware, por lo cual se contará con la ventaja de poder implementar un software distinto para otras aplicaciones, con un mínimo costo.

El Inamo restaurant en Londres es un buen ejemplo de la aplicación de un monitor sensible al tacto en un punto de venta y de cómo dicha tecnología beneficia a los mozos, prestando un servicio más rápido a los clientes. Además, se optimiza la captura de datos, volviendo a otros periféricos prescindibles y utilizando el espacio más eficientemente.

### *2. Planteamiento del problema a resolver*

El empleo del touch-screen menú está orientado, no solo a un menú electrónico, sino a la implementación de un sistema digital en el rubro gastronómico que permita un mayor control y rapidez en el servicio.

El disponer de un sistema digital permite un mayor control en los ingresos y consumos realizados, y además, el hecho de poder almacenar estos datos en una base de datos, crea la posibilidad de realizar una correcta estadística de los mismos y tener una mejor proyección a futuro.

El menú electrónico es de actualización casi inmediata. Esto trae como beneficio poder informar que productos se encuentran disponibles, además que sus precios siempre se encuentran actualizados.

Otra ventaja del menú electrónico es que los pedidos pueden ser realizados directamente de éste, por lo tanto elimina tanto el tiempo de espera de atención, como la malinterpretación del pedido.

Por último hay que mencionar que también se diseña el hardware para la implementación del touch-screen menú, en consecuencia se puede lograr un precio final menor que la competencia, al realizarse el mismo específicamente para este tipo de aplicación. Esto conlleva a resolver el problema económico que implica la implementación de este sistema en locales gastronómicos con recursos económicos más limitados.

## VI. DEFINICIÓN DE PRODUCTO

### *1. Requerimientos técnicos*

A partir de un análisis de mercado se identificaron las características primarias que busca el cliente en nuestro producto. Conociendo los requerimientos impuestos por el público, se focaliza en el desarrollo del dispositivo satisfacer las necesidades expresadas por el comensal y el dueño del restaurant. A continuación se enumeran los requerimientos estudiados:

**Liviano (bajo peso):** Es un requerimiento deseable para el buen manejo de menús.

**Dimensiones (comodidad):** Las dimensiones de la pantalla deben ofrecer comodidad para la lectura y visualización de las imágenes de las comidas, al igual que un formato preciso para la selección de los platos.

**Modo de uso dactilar:** El uso dactilar del menú propone una interacción única e intuitiva entre el cliente y el producto.

**Mayor cantidad de texto que de imagen:** El estudio del mercado indicó que se prefiere obtener un detalle breve de las comidas, ingredientes, calorías y tiempo de espera en texto por sobre una imagen detallada de la comida. En contraparte, se ofrece una imagen visual atractiva del plato.

**Interfaz amigable:** Con ayuda de una interfaz intuitiva el usuario se acostumbra rápidamente al uso del mismo, obteniendo un servicio más eficiente. También se ofrece una visualización de la cuenta actual por los pedidos realizados.

**Diseño agradable:** Utilizando una presentación agradable y entretenida, se ofrece al cliente una experiencia culinaria nueva. A su vez, se aprovecha el producto para mejorar la imagen de la empresa y atraer más público.

**Información del tiempo de espera:** La propuesta del tiempo estimado de espera ofrece un mecanismo de feedback efectivo que no se tiene en cuenta en el resto de los restaurantes con menús convencionales.

**Calificación opcional de servicios:** Para la mejora constante del servicio del restaurant, se ofrece la calificación de los servicios prestados en forma digital.

**Ranking de comidas:** Recomienda los platos más vendidos o mejores puntuados como una propuesta para comensales no frecuentes.

**Publicidad:** Principalmente ayuda a promover los platos especiales y promociones del día y promocionar los auspiciantes del local.

**Bajo consumo:** Es de suma importancia que el producto sea de bajo consumo, tanto desde el punto de vista ambiental, como también para obtener una duración prolongada de uso mediante baterías recargables.

**Estadísticas de venta:** Al digitalizar los pedidos, se obtiene un control inmediato de las ventas efectuadas, lo cual es muy importante para adaptarse a la demanda del cliente y conocer mejor el mercado.

**Bajo Costo:** Al economizar el producto, se busca tener un mayor alcance en el mercado y llegar a los negocios que trabajan con poco margen de ganancia y ayudarlos en competir contra empresas más grandes y reconocidas.

**Portabilidad:** El hecho de que el dispositivo electrónico sea portable indica aprovechar el espacio físico de la mesa, sin necesidad de llevar la alimentación para el producto, volviendo al producto más versátil y utilizable en distintos ambientes.

**Resistente:** Es muy valorado por el dueño del restaurant que el producto tenga una vida útil prolongada para que sea rentable el uso del mismo.

**Amigable con el medio-ambiente:** Hoy en día el público se encuentra cada vez más interesado en productos eco-friendly, que si bien implican un costo mayor, garantiza mejor recepción en la comunidad.

## *2. Definición de las especificaciones*

A partir de los requerimientos estudiados previamente, se determinaron las especificaciones del producto a ofrecer, y son los puntos en donde se prestará atención en el momento del desarrollo. Cabe destacar, que el cumplimiento de las especificaciones serán las que indican la calidad del producto y el nivel de satisfacción en los clientes.

### *a. Especificaciones del Hardware*

**Uso de batería:** En nuestro caso, se busca que el producto sea de bajo consumo y aguante muchos ciclos de recargas continuas, ya que una vez realizado el pedido el dispositivo será retirado de la mesa y puesto a recargar. Como primera aproximación, se determinó un mínimo de una hora de duración para uso continuo.

**Micro de bajo consumo y pequeño:** El tamaño del microcontrolador debe ser suficientemente chico para que ocupe el menor espacio posible y afectar en menor medida el tamaño y peso final del producto. Un microcontrolador de bajo consumo ayuda a una vida útil prolongada de batería y es más amigable con el medioambiente en relación de compromiso con la velocidad de procesamiento.

**Gabinete plástico:** Las ventajas de tener un gabinete de plástico frente a uno hecho de metal son: uso de un plástico reciclable, menor peso en el producto final y menos probabilidad de que el usuario sufra descargas eléctricas a través de dicha superficie, a cambio de una resistencia menor.

**Alta velocidad de procesamiento:** La pantalla requiere de cierta velocidad de refresco manejado por el microcontrolador para el flujo correcto de datos e información.

**Pantalla a color:** La interfaz se desarrollará en principio sobre una pantalla TFT a color RGB con resolución de 320\*240 pixeles, ya que se deberá mostrar con suficiente detalle las imágenes. Por otro lado, se cuenta con una membrana táctil resistiva de 4 cables, debido a que es la más económica en el mercado y la precisión en la pantalla no es un factor importante para nuestra aplicación.

**Sistema de comunicación:** La comunicación con el servidor en una computadora fija será vía comunicación inalámbrica por medio de un módulo Zigbee. Éste trabaja en un rango de frecuencias cercano a los 2,4GHz (al igual que internet Wi-Fi) y corresponde a un rango de frecuencia libre y permitida en Argentina y los estándares electromagnéticos de Europa. Los costos de un sistema Wi-Fi en comparación con un Zigbee son muy elevados y presentan mayor consumo de potencia en el dispositivo además de ser requerir mantenimiento del modem. De este modo el pedido es activado por el mismo cliente eliminando errores en el pedido y minimizando los tiempos de espera.

### *b. Especificaciones de Software*

**Actualización rápida del servidor:** Uno de los principales objetivos del servicio que se brinda es la eliminación de tiempo de espera, es por eso que se requiere suficiente velocidad de recepción en la terminal del servidor y obtener correctamente los pedidos de los clientes. También es necesario actualizar adecuadamente en cada menú electrónico la disponibilidad.

**Base de datos:** Es importante la disposición de un gran número de productos para poder ofrecer distintas variedades y gustos y por ende se requiere suficiente memoria. También se desea que la base de datos tenga suficiente espacio para guardar las estadísticas de ventas entre otros datos.

**Interfaz visual:** La presentación del menú es muy importante para generar interés en el cliente y ofrecer una interfaz amigable de uso fácil e intuitivo como es de esperar de los dispositivos táctiles de hoy en día.

Especificaciones Técnicas:

- Peso: menor a 1,5 kg.
- Rango de temperatura de operación: de -10°C a 50°C
- Resolución mínima de la pantalla: 640\*480 pixeles. Idealmente: 800\*600 pixeles.
- Tamaño mínimo de la pantalla: 7" (distancia de la diagonal)
- Costo de producción en masa: < USD 250
- Comunicación inalámbrica hasta 75m
- Uso de batería recargable con un máximo de 6 horas de uso no continuo

Especificaciones de la interfaz y protocolos de comunicación:

- Capacidad de mostrar precio total de la venta.
- Capacidad de almacenar un mínimo 100 platos distintos
- Información del estado de la carga de batería
- Actualización incremental de la base de datos
- Identificación personal del mozo
- Archivo de mesas ocupadas y/o disponibles
- Asignación variable de mesa de atención a cada producto
- Protocolo TCP de comunicación entre servidor y el dispositivo
- Botón físico de prendido y apagado

### *3. Casa de calidad (QFD)*

El análisis de la casa de calidad relaciona los requerimientos impuestos por el cliente y el mercado con las especificaciones fijadas por nuestro proyecto. Se utilizó un sistema de valoración del producto y el servicio de 1 a 5 y se evaluó al producto respecto a la competencia A y B que corresponden a las empresas: E-Menu con un producto de Shenzhen Hengfengda Tech Co. en China y Touchit!, respectivamente. Si bien la última empresa tiene una sucursal en Montevideo, Uruguay, no cuenta con soporte técnico local, lo cual representa tiempo de envío al igual que costo elevados de reparación de productos. A continuación se detalla mediante el estudio de mercado, que se realizó con ayuda de más de 30 interesados en respuesta a una encuesta<sup>2</sup> formulada por nuestros conocimientos e investigaciones, los requerimientos de los comensales (en color azul) y los del dueño de restaurantes (en color verde), que representan nuestros clientes.

---

<sup>2</sup> La encuesta realizada se encuentra en el Anexo I.



## VII. ANÁLISIS DE FACTIBILIDAD

### *1. Factibilidad tecnológica*

#### *a. Propuestas de alternativas de diseño*

El dispositivo a desarrollar cuenta con comunicación inalámbrica, debido a que el estudio de mercado indicaba dicho requerimiento para la aplicación en un restaurante, no obstante, cabe destacar, que el producto es suficientemente versátil para cubrir necesidades en distintos ámbitos, donde la comunicación inalámbrica no es un factor a destacar. En otras palabras, una alternativa del diseño del producto es utilizar una conexión con cables tanto para la alimentación del menú como para la comunicación al servidor. Esto quiere decir, que se eliminarían tanto el ítem de batería como de conexión inalámbrica, disminuyendo notoriamente los costos del mismo. Una aplicación de dicha propuesta es la máquina electrónica toma-turnos (ya implementados en distintos hospitales) o bien un punto de venta de comida rápida estático.

Por otro lado, sin considerar cambios tan contundentes como la propuesta anterior, se puede implementar una expansión de memoria mediante una tarjeta SD, para el caso de un restaurante que desee un menú con más opciones e imágenes, logrando de este modo mayor almacenamiento de datos. No obstante, en las especificaciones mencionadas posteriormente, se cubre hasta un mínimo de 100 platos distintos, sin necesidad de agregar más memoria para aplicaciones dentro del estándar normal.

Finalmente, se deben tener en cuenta los menús electrónicos de alta gama, que soportan video y audio, para entretenimiento del cliente durante la espera. En la actualidad dicha aplicación no se encuentran muy difundida, pero es una alternativa a tener en cuenta para la escalabilidad del producto.

#### *b. Elección de una solución*

Luego de mencionar las distintas propuestas de alternativas de diseño, se optó desarrollar el producto prestando especial atención en el costo del producto final, ya que la venta se realizará en gran cantidad y el precio final es el factor crítico de nuestros objetivos frente a la competencia. Por esta misma razón, se optó dejar de lado la tarjeta SD como expansión de memoria, al igual que la salida de audio, ya que eran requerimientos impensados del cliente según el modelo de Kano y cumpliéndolas, no provocan insatisfacción si no se aportan.

En contrapartida, para la aplicación dentro de restaurantes, la portabilidad del producto como la comunicación inalámbrica son requerimientos implícitos del cliente. Esto quiere decir, que no aumentan la satisfacción del cliente, pero causan una insatisfacción muy grande si no se aportan.

En cuanto a los requerimientos explícitos, se identificó al impacto ambiental que generaría el producto final, sobretudo en el tema de bajo consumo y materiales del dispositivo. Es por esta misma razón, que se opta por utilizar en la comunicación inalámbrica la especificación de alto nivel de comunicación del ZigBee para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4. La principal razón de utilizar el ZigBee frente a la alternativa de comunicación Wi-Fi es la aplicación en el contexto, ya que requiere una baja tasa de envío de datos seguros, maximizando la vida útil de la batería. Un nodo ZigBee reduce su consumo gracias a que puede permanecer dormido la mayor parte del tiempo y tiene la capacidad de despertar en un tiempo ínfimo (menor a 15mseg) para su uso. ZigBee tiene un consumo de 30mA transmitiendo y de 3uA en reposo, muy inferior a los picos de alrededor de 3A en el caso del Wi-Fi o los 40mA transmitiendo y 0,2mA en reposo que tiene el Bluetooth.

También se buscará utilizar fuentes switching, los cuales tienen mayor eficiencia ( típicamente de 75-85% en fuentes de entre 10-50W y hasta un 95% en buenos diseños) frente a los reguladores de tensión, disipando menor potencia y periodos de usos más largos logrando de este modo nuestro compromiso con el medioambiente.

*c. DFMEA*

Dado que el diseño del hardware no está definido, se realizó un DFMEA de los bloques más importantes y sus fallas más generales.

Los módulos a analizar son:

- Alimentación
  - Batería
  - Fuente Switching
- Interfaz Software-Usuario
  - Pantalla RGB
  - Membrana táctil
- Software
- Conexión inalámbrica
- Microprocesador

Se calificó la posibilidad de ocurrencia de una causa de falla, su probabilidad de detección y la severidad de sus efectos con una escala de 1 a 10. Para la ocurrencia, a mayor el número, más probable es que ocurra; para la severidad, un número mayor indica un efecto de la falla más severo; y para la detección, la escala asciende en orden de dificultad para detectar la falla.

Módulo	Item	Falla Potencial	Control de Detección	Causas	Ocurrencia	Potenciales Efectos	Severidad	Detección	Control de Prevención	RPN	Acciones Recomendadas	
Alimentación	Batería	La tensión de salida es 0V	Medición de la tensión de salida.	Batería defectuosa.	1	El sistema no funciona	9	3	Realización de un control de calidad sobre la batería en otro circuito.	27	Sustituir la batería.	
				Batería descargada.	5			3	Verificación de la tensión de salida de la batería.	135	Cargar la batería.	
				Cortocircuito por mal diseño del PCB.	4	Posiblemente se queme la/s fuente/s switching.	10	4	Diseño cuidadoso del PCB.	160	Detectar el cortocircuito y rediseñar el PCB.	
		La tensión de salida es menor a la mínima especificada.		Batería parcialmente descargada.	7	El sistema puede reiniciarse aleatoriamente.	5	3	Verificación de la tensión de salida de la batería.	105	Cargar la batería.	
				Batería defectuosa.	1	Puede que el sistema no arranque.	6	3	Realización de un control de calidad sobre la batería en otro circuito.	18	Sustituir la batería.	
		La tensión de salida es mayor que la especificada.		Batería defectuosa.	1	Los componentes que se alimentan directamente pueden dañarse.	8	3	Verificación de la tensión de salida de la batería.	24	Sustituir la batería.	
									Colocación de un circuito de prevención.	24		
		Se sobrecalienta.		Verificación de que la temperatura de los componentes se mantenga estable.	Cortocircuito en algún punto.	6	Puede quemarse un circuito integrado.	10	5	Utilización de márgenes de seguridad en la separación entre pistas.	300	Detectar el cortocircuito y rediseñar el PCB.
					Mal diseño de los componentes alimentados por la batería.	5			4	Simulación previa en condiciones de funcionamiento.	200	Volver a calcular los componentes utilizados.

Módulo	Item	Falla Potencial	Control de Detección	Causas	Ocurrencia	Potenciales Efectos	Severidad	Detección	Control de Prevención	RPN	Acciones Recomendadas
Alimentación	Fuente switching	La tensión de salida es 0V	Medición de la tensión de salida con un tester u osciloscopio.	Circuito Integrado quemado	2	El sistema no funciona.	10	5	Realización de un control de calidad sobre los componentes adquiridos en un circuito independiente.	100	Cambiar el integrado quemado.
		La tensión de salida es menor a la mínima especificada		Cortocircuito por el mal diseño del PCB.	3	Posiblemente se queme el chip regulador o el inductor.	10	5	Utilización de un proceso de fabricación confiable.	150	Detectar el cortocircuito y rediseñar el PCB.
				Mal dimensionamiento de los componentes	4	El sistema puede reiniciarse aleatoriamente	6	4	Simulación previa en condiciones de funcionamiento	96	Volver a calcular los componentes utilizados
				Los componentes utilizados no coinciden con los diseñados	5	Puede que el sistema no arranque	9	6	Verificación de los componentes antes del encendido	270	Revisar y cambiar los componentes utilizados
				La tensión de salida es mayor que la especificada	Fallo en el circuito de protección	5	Los componentes que se alimentan directamente pueden dañarse	9	4	Verificar el correcto funcionamiento del circuito de protección	180
		Mal dimensionamiento de los componentes			5	5			Simulación previa en condiciones de funcionamiento	225	Volver a calcular los componentes utilizados.
		Se sobrecalienta	Verificar la temperatura de los componentes.	Cortocircuito en algún punto	4	Posiblemente se quemen los circuitos integrados.	9	6	Utilización de márgenes de seguridad en la separación entre pistas	216	Detectar el corto circuito y rediseñar o reconstruir el PCB
				Mal dimensionamiento de los componentes	3			7	Simulación previa en condiciones de funcionamiento	189	Volver a calcular los componentes utilizados
		El ripple de la alimentación es muy elevado	Medición de la tensión de salida.	Mal diseño de los componentes.	3	El sistema se apaga y se enciende.	6	4	Verificar el diseño de los componentes.	72	Volver a soldar los componentes.

Módulo	Item	Falla Potencial	Control de Detección	Causas	Ocurrencia	Potenciales Efectos	Severidad	Detección	Control de Prevención	RPN	Acciones Recomendadas
Interfaz software-usuario	Pantalla RGB	Ausencia de un canal de color.	Observar un imagen RGB de prueba.	Falla en la conexión de uno o más canales de color.	3	Colores no adecuados en la imagen.	3	3	Testear el estado de conexión de los canales.	27	Conectar correctamente los canales.
				Error de programación.	6			2	Debuggear el código de programación.	36	Corregir el código.
				Pantalla defectuosa.	1			5	Probar la pantalla en un circuito independiente.	15	Sustituir la pantalla.
		La pantalla no muestra la imagen.		La pantalla no está alimentada.	2	No se visualiza la interfaz del software.	8	3	Testear la alimentación de la pantalla.	48	Conectar correctamente la alimentación.
				La conexión con los canales está defectuosa.	2			3	Verificar el estado de conexión de los cables.	48	Conectar correctamente los canales.
				Pantalla defectuosa.	1			5	Verificar el funcionamiento de la pantalla en un circuito independiente.	40	Sustituir la pantalla.
		Sector de la pantalla no muestra imagen.		Pantalla defectuosa.	1	No se visualiza parcialmente la interfaz del software.	6	5	Probar la pantalla en un circuito independiente.	30	Sustituir la pantalla.
				Error de conexión del bus de la pantalla.	4			3	Constatar la conexión de la pantalla.	72	Conectar correctamente la pantalla.
		Baja luminosidad.		Alimentación incorrecta.	3	Imagen poco legible.	2	4	24	24	Conectar correctamente la pantalla.
				Pantalla defectuosa.	1			5	10	10	Sustituir la pantalla.

Módulo	Item	Falla Potencial	Control de Detección	Causas	Ocurrencia	Potenciales Efectos	Severidad	Detección	Control de Prevención	RPN	Acciones Recomendadas
Interfaz software-usuario	Membrana táctil	No se detecta ningún acción sobre la membrana.	Someter la membrana a un programa de prueba.	Mala conexión	4	No se puede ingresar información.	9	5	Verificar la conexión de la membrana.	180	Conectar correctamente la conexión de la membrana.
				Membrana defectuosa	2			6	Probar el funcionamiento de la membrana en un circuito independiente.	108	Sustituir la membrana.
		Detección esporádica del contacto.		Problemas de software.	6	Problema para ingresar los datos.	6	3	Debuggear el código de programación.	108	Corregir el código.
				Membrana sucia.	8			2	Limpieza periódica de la pantalla.	96	Limpiar la pantalla.
				Mala conexión.	4			5	Verificar la conexión de la membrana.	120	Conectar correctamente la conexión de la membrana.
		Detección imprecisa.		Problemas de software.	6	Lectura de datos incorrecto.	7	3	Debuggear el código de programación.	126	Corregir el código.
				Mal ruteo de la matriz de la membrana.	3			5	Utilización de un proceso de fabricación confiable.	105	Rediseño de la matriz de la membrana mediante hoja de datos.
		Se interpreta cada presión de la membrana muchas veces.		Problemas de software.	6	Cantidad erronea de veces de lectura de datos.	3	3	Debuggear el código de programación.	54	Corregir el código.

Módulo	Item	Falla Potencial	Control de Detección	Causas	Ocurrencia	Potenciales Efectos	Severidad	Detección	Control de Prevención	RPN	Acciones Recomendadas
Módulo de software	El microprocesador no responde a los paquetes recibidos inalámbricamente.	Realizar testeos de comunicación.	Falla del módulo de comunicación.	4	No se puede establecer comunicación entre el menú y el servidor.	8	2	Usar prevenciones específicas del modulo de comunicación.	64	Usar correcciones específicas del modulo de comunicación.	
			Falla de software.	7			3	Debuggear el código de programación.	168	Corregir el código.	
	El microprocesador no responde a los eventos de ingreso de datos	Ingresar datos a través de la membrana.	Falla del módulo de la membrana.	4	No se puede leer correctamente lo ingresado por el usuario.	7	2	Usar prevenciones específicas para la membrana.	56	Usar correcciones específicas para la membrana.	
			Falla de software.	7			3	Debuggear el código de programación.	147	Corregir el código.	
	El microprocesador no actualiza la pantalla correctamente.	Chequear la interfaz visual.	Falla en el módulo del display.	4	No se muestran la interfaz visual al usuario.	9	2	Usar prevenciones específicas del modulo de pantalla.	72	Usar correcciones específicas del modulo de pantalla.	
			Falla de software.	7			3	Debuggear el código de programación.	189	Corregir el código.	
	El programa queda atrapado en un loop.	Testeo del software con pruebas del sistema.	El programa queda encerrado en una subrutina.	5	El sistema deja de reaccionar a interrupciones.	8	2	Debuggear el código de programación.	80	Corregir el código.	
	El programa no maneja correctamente las bases de datos.	Realizar cambios de precios de prueba	Falla de software.	7	No se pueden actualizar correctamente las estadísticas de pedidos realizados ni las comidas disponibles.	6	2	Debuggear el código de programación.	84	Corregir el código.	

Módulo de conexión inalámbrica	No se establece la comunicación.	Probar la comunicación con distintas señales de prueba.	Se superó el alcance máximo de la comunicación inalámbrica.	4	No hay comunicación.	8	3	Verificar la distancia entre emisor y receptor.	96	Acercarse físicamente al servidor o acercarse a otro menú con conectividad.	
			Falla de alimentación del módulo de conexión.	3			5	Testear alimentación del módulo.	120	Conectar correctamente la alimentación.	
			Módulo de comunicación defectuosa	1			6	Probar el funcionamiento del módulo en un circuito independiente.	48	Reemplazar el módulo.	
			Problemas de software.	5			2	Debuggear el código de programación.	80	Corregir el código.	
	Comunicación esporádica.		Problema para transmitir datos.	6	Interferencia electromagnética.		2	7	Medir el ruido electromagnético en el ambiente.	84	Utilizar otro ancho de banda aceptable.
					Bloqueo de comunicación.		4	4	Elegir una ubicación con buena señal.	96	Reubicación del emisor o receptor.
					Problemas de software.		5	2	Debuggear el código de programación.	60	Corregir el código.

Módulo	Falla Potencial	Control de Detección	Causas	Ocurrencia	Potenciales Efectos	Severidad	Detección	Control de Prevención	RPN	Acciones Recomendadas
Módulo de conexión inalámbrica	Comunicación unidireccional.	Probar la comunicación con distintas señales de prueba.	Saturación del canal de alguna de las partes.	3	No se actualiza correctamente una de las partes.	4	3	Dimensionar correctamente el buffer.	36	Cambiar el tamaño del buffer.
			Problemas de hardware de alguna de las partes.	2			5	Probar el funcionamiento del módulo otro circuito independiente.	40	Reemplazar el hardware.
			Problemas de software de alguna de las partes.	4			2	Debuggear el código de programación.	32	Corregir el código.
	Demora en la comunicación.		Debido a retransmisión de datos.	6	Mayor tiempo de espera y actualización demorada.	2	3	No superar la cantidad de emisores.	36	Aumentar el ancho de banda o utilizar otro servidor.
			Saturación del canal debido a conexiones simultáneas.	4			3	Dimensionar correctamente el buffer.	24	Implementar en el software un reenvío de paquetes.
	Transmisión de datos erróneos.		Ruido electromagnético.	2	Comunicación errónea.	5	7	Medir el ruido electromagnético en el ambiente.	70	Implementar en el software un reenvío de paquetes.
			Problemas de software de alguna de las partes.	5			2	Debuggear el código de programación.	50	Corregir el código.

Módulo	Falla Potencial	Control de Detección	Causas	Ocurrencia	Potenciales Efectos	Severidad	Detección	Control de Prevención	RPN	Acciones Recomendadas
Microprocesador	El microprocesador no enciende.	Medición de la tensión en un pin de prueba con un tester u osciloscopio con ayuda de un programa de prueba.	Falla de alimentación.	3	No funciona el dispositivo.	10	3	Verificar la conexión del microprocesador.	90	Revisar y cambiar los componentes utilizados
			Falla de hardware.	1			6	Probar el funcionamiento del microprocesador en un circuito independiente.	60	Cambiar el integrado.
	El microprocesador se sobrecalienta	Verificación de que la temperatura de los componentes por debajo de lo tolerado.	Cortocircuito en algún punto	5	Posiblemente se quemen los circuitos integrados.	10	7	Utilización de márgenes de seguridad en la separación entre pistas.	350	Detectar el corto circuito y rediseñar.
			Mala refrigeración del microprocesador.	3			2	Ubicación del microprocesador en un lugar aislado.	60	Proveer de un área de convección.
			Sobrecarga de operaciones.	2			6	No sobrepasar el límite de overclock establecido por la hoja de datos.	120	Corregir el código.
	El microprocesador se resetea periódicamente.	Medición de la tensión en un pin de prueba con un tester u osciloscopio con ayuda de un programa de prueba.	Falla de alimentación.	3	Mal funcionamiento del dispositivo en general.	8	3	Verificar la conexión del microprocesador.	72	Revisar y cambiar los componentes utilizados
			Falla de hardware.	1			6	Probar el funcionamiento del microprocesador en un circuito independiente.	48	Cambiar el integrado.

## *2. Factibilidad de tiempos*

Para la realización de la planificación, se detalla una breve explicación sobre cada tarea a realizar:

### Análisis de Factibilidad:

Factibilidad tecnológica, factibilidad de tiempos, factibilidad económica y factibilidad legal y responsabilidad civil.

### Análisis de Competencia:

En esta etapa se compara las demás marcas y prestaciones de los modelos que las empresas competidoras presentan. Se realiza una comparación desde el punto de vista tecnológico para ver cómo se presentan nuestros competidores.

### Diagrama en Bloques (hardware):

Definición de los módulos presentes en el sistema, con una breve explicación de las funciones de cada bloque.

### Investigación del hardware para la Pantalla y Controlador:

Investigación sobre las diferentes tecnologías y modelos de pantallas, como también los diferentes tamaños presentes. Hay que tener en cuenta si las mismas presentan su propio controlador ya integrado, o la necesidad de adquirir uno con la búsqueda de información para su utilización en pantallas.

### Investigación del hardware para Membrana táctil y ADC:

Conocimiento sobre tecnologías presentes de membranas y resolución para los diferentes tamaños de las mismas. También investigar las tecnologías y modelos de conversores A/D para pantallas táctiles. Análisis sobre los integrados para estas funciones con sus ventajas y desventajas.

### Investigación del hardware para comunicación inalámbrica:

Se debe realizar una investigación sobre las distintas tecnologías de comunicación inalámbrica, teniendo en cuenta su consumo. Una vez elegida la tecnología a utilizar, realizar un aprendizaje sobre los protocolos de comunicación y frecuencias permitidas para la misma.

### Investigación del hardware para microprocesador:

Aprendizaje de las funciones del microprocesador, como manejo de interrupciones, instrucciones básicas, funcionamiento de periféricos, etc. Además hay que tener conocimiento sobre cómo realizar la programación y estructuras dentro del mismo.

### Investigación de alimentación del circuito:

Se hace una investigación del funcionamiento de circuitos de fuente de alimentación de distintos tipos y sus respectivas ventajas y desventajas. Es necesario para tener idea de las distintas tecnologías y distribución de la potencia.

### Investigación de batería:

Investigación sobre la utilización y tecnologías de las baterías. Esta tarea también implica el cuidado debido a los efectos en el medio ambiente, tanto en la fabricación, mantenimiento como en su eliminación, respetando las normas regionales vigentes.

Aprendizaje del lenguaje de programación:

Se realiza la programación de la interfaz visual en C++ en plataforma de Linux, por lo tanto hay que familiarizarse con dichas funciones y programas de compilación. Debido a la utilización de un microprocesador con sus propio set de instrucciones, es necesario aprender a realizar la compilación de las funciones programadas en dicho microprocesador, para que este pueda interpretarlas.

Diseño de la placa de comunicación inalámbrica y su interconexión:

El diseño de la placa de comunicación es relativamente sencillo debido a que ya se cuentan con los módulos que realizan dicha comunicación integrados en un solo microprocesador, y cuenta con pocos componentes, además de su masiva distribución. Hay que realizar el protocolo de comunicación entre los diferentes dispositivos, y luego se realiza un programa de prueba sobre las placas de comunicación para verificar su correcto funcionamiento y posibles fallas.

Diseño de la placa de microprocesador, ADC y RAM/FLASH:

Para el diseño de la disposición del micro, el ADC y la memoria dentro de la placa y su distribución se posee el conocimiento de cómo realizarlo aunque su ruteo lleva un tiempo considerado por su complejidad. En el diseño se debe tener en cuenta todas las entradas y salidas del mismo para los demás módulos. Una vez realizado el diseño, su fabricación se terceriza debido a que la placa resulta generalmente four-layer y dicha fabricación no se realiza en el país. De todas maneras para el prototipo puede conseguirse un módulo para dicha placa.

Diseño de la alimentación:

En esta etapa se debe realizar, teniendo en cuenta la investigación realizada tanto sobre la alimentación como la batería, el dimensionamiento de los componentes de la placa de alimentación y sus conexiones cumpliendo las especificaciones establecidas. Para esto, primero se requieren realizar simulaciones con el circuito diseñado hasta obtener el funcionamiento deseado del mismo.

Una vez realizado lo mencionado anteriormente, se debe realizar un prototipo de la placa de alimentación para poder hacer pruebas sobre la misma y verificar que el diseño funcione correctamente. Sobre estas pruebas hay que verificar posibles fallas como variaciones de la tensión, ripple de la fuente, etc. También hay que tener en cuenta en esta tarea el tiempo de duración de la batería que cumpla con lo establecido en las especificaciones.

Integración de los diseños:

En esta etapa hay que encargarse de integrar todos los módulos anteriores para que se tenga compatibilidad y una buena comunicación entre los mismos. En esta etapa se pueden reevaluar los distintos componentes en caso de falla en la integración.

Diseño del PCB final:

En esta etapa se debe realizar el diseño de la placa final que contenga a todos los módulos juntos, a lo sumo con las conexiones necesarias. También hay que realizar el modelo de los componentes que todavía no estén implementados.

Soldar la placa final y prueba de conceptos:

Esta etapa implica el soldado y la realización de la placa una vez que está diseñada. Se debe tercerizar su implementación, ya que no se cuenta con la tecnología para realizar dicha placa en la región.

Diseño y armado de gabinete:

En esta tarea hay que realizar el diseño del gabinete para que cumpla las especificaciones establecidas. Una vez hecho esto se pasa a implementar la matricera de inyección en plástico de la misma que se terceriza.

Realización de pruebas para el correcto funcionamiento:

En esta etapa se ejecuta la verificación de las conexiones de la placa final y los características principales del funcionamiento.

Diagrama de estados y flujogramas (software):

Definición de las diferentes funciones presentes en el programa, las subrutinas, la máquina de estados y su interrelación. Hay que realizar el flujograma para determinar la lógica que sigue el programa, las funciones principales, y la forma de acceso del usuario.

Desarrollo de software:

Esta etapa es fundamental en el funcionamiento de todo el sistema debido a que va ser la interfase entre el hardware y el usuario. Tiene que tener un flujo adecuado de datos con el usuario. El software debe cumplir con dos funciones principales. La primera es la de un eficiente manejo de los driver para los diferentes periféricos del sistema. Y la segunda es una interacción amigable e intuitiva con el usuario.

Plan de prueba de módulos y de depuración de software:

Para un correcto funcionamiento del software, es recomendable realizar pruebas a los diferentes módulos que componen a este por separado, luego ir juntándolos de a uno y verificar su correcto funcionamiento entre estos. Así se puede llegar a detectar un error de programación de manera más sencilla.

Integración entre software y hardware:

Se integra el software al microprocesador mediante los procedimientos estudiados anteriormente.

Prueba de prototipo armado:

En esta etapa se ejecutan las pruebas de funcionamiento con el hardware y software en conjunto. Puede ser el caso que se necesite rediseñar alguno de los módulos, por lo tanto se tiene en cuenta un tiempo mayor para su terminación.

Validación y documentación final:

El proyecto queda terminado una vez que se realiza la validación del producto final, el cual ya puede empezar a realizarse en serie. Además de esto hace falta toda la documentación correspondiente del diseño realizado para dar por finalizado el proyecto.

*a. Planificación – Análisis PERT*

Para la planificación PERT se toman en cuenta los tiempos holgados que pueden tener las tareas que no se encuentran en el camino crítico, con lo cual la tarea en si o puede empezarse o finalizarse más tarde, de acuerdo a lo convenido. Esto quiere decir, que el trabajo a realizar después de una de las tareas que no se encuentra en el camino crítico debe esperar a que una de estas finalice. Esto da un margen de tiempo en el cual el proceso se encuentra detenido. Este tiempo puede incluirse tanto al inicio como al final de la tarea, o una distribución tanto al inicio como al final de la misma.

Para este análisis se toman en consideración los tiempos disponibles de los diseñadores, con lo cual algunas tareas se las considera de mayor duración ya que las mismas tienen en cuenta tiempos de ocupación de los mismos.

El tiempo estimado de duración del proyecto es de 285 días hábiles, por lo tanto se pronosticó, que se terminaría alrededor de fines de Diciembre del año siguiente al que se inició el proyecto. No obstante, no se pudo cumplir con los tiempos planeados, debido a las siguientes causas:

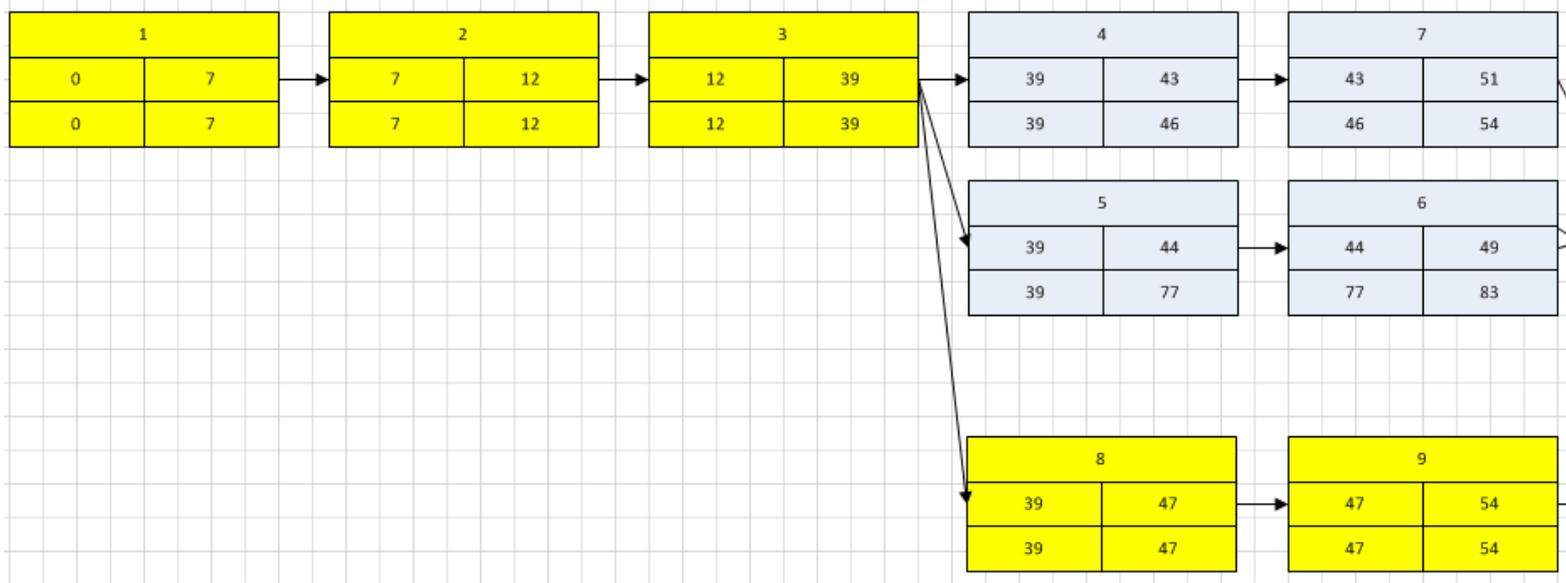
- carga horaria de la universidad: difícilmente se pudo prever el tiempo de estudio necesario que exigían las responsabilidades académicas, utilizando en muchas ocasiones días hábiles calculados para el proyecto. El tiempo de estudio del último cuatrimestre se vieron reflejados en aproximadamente 80 días hábiles más.
- desarrollo de software (tarea 20): el tiempo estimado de software de 45 días solamente consideraba la realización del software incorporado en el menú electrónico. No se previó el tiempo que requirió el software del servidor y su interacción, que demandó aproximadamente 45 días más.
- integración entre software y hardware (tarea 22): la unificación entre ambas se asigna a la plataforma de firmware, que es básicamente la configuración y compilación del kernel con los drivers a utilizar en la PCB. Dicha tarea demoró cerca de 50 días, debido a que los drivers nativos de Linux requirieron adaptación para la presente aplicación y versiones de librerías utilizadas.

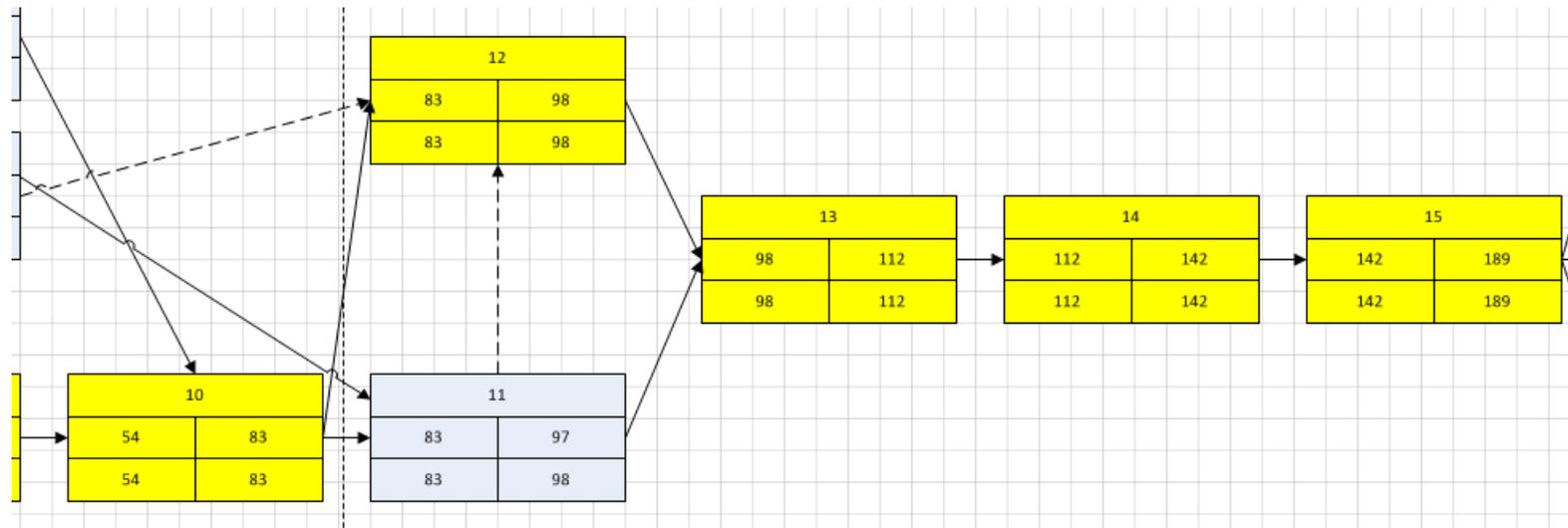
Se observa que 8 de las tareas no pertenecen al camino crítico, las cuales son las que cuentan con el tiempo de holgura mencionado. Las variaciones en el tiempo de proceso de las demás tareas, al no contar con ese tiempo, son las que van a definir la variación en el tiempo total de la finalización del proyecto.

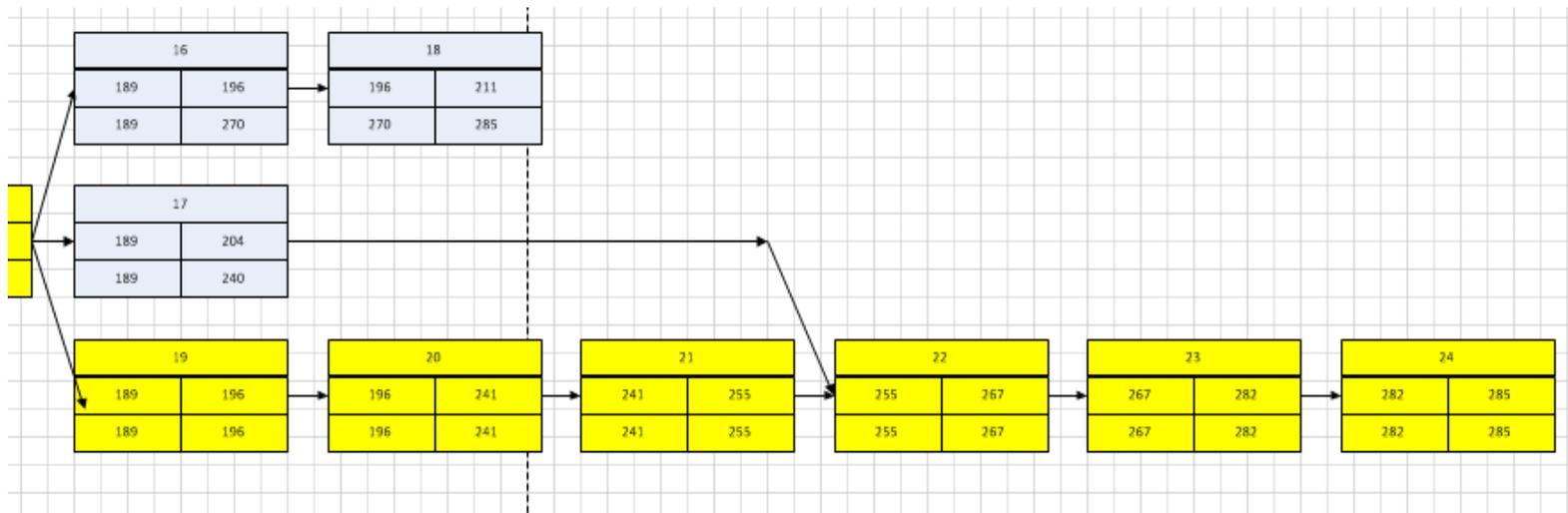
Se presentan a continuación la lista de las tareas a tener en cuenta en el diagrama de PERT:

TABLA I  
NÚMERACIÓN Y NOMBRES DE LAS TAREAS DEL DIAGRAMA DE PERT

Número	Nombre de la Tarea
1	Análisis de Factibilidad
2	Análisis de Competencia
3	Diagrama en Bloques (hardware)
8	Investigación de alimentación del circuito
9	Investigación de batería
10	Aprendizaje del lenguaje de programación
12	Diseño de la placa de microprocesador, ADC y RAM/FLASH
13	Diseño de la alimentación.
14	Integración de los diseños
15	Diseño del PCB final
19	Diagrama de estados y flujogramas (software)
20	Desarrollo de software
21	Plan de prueba de módulos y de depuración de software
22	Integración entre software y hardware
23	Prueba de prototipo armado
24	Validación y documentación final







## b. Programación – Diagrama de Gantt

Al realizar la programación (Gantt) se enumeran las tareas a realizar, con sus tiempos aleatorios de duración para cada una, dejando en evidencia sus correspondientes precedencias. De esta forma se aparecían gráficamente los tiempos involucrados para las tareas, y cuales son aquellas que limitan el tiempo mínimo de finalización del proyecto, denotadas como el camino crítico (marcadas en rojo en el gráfico).

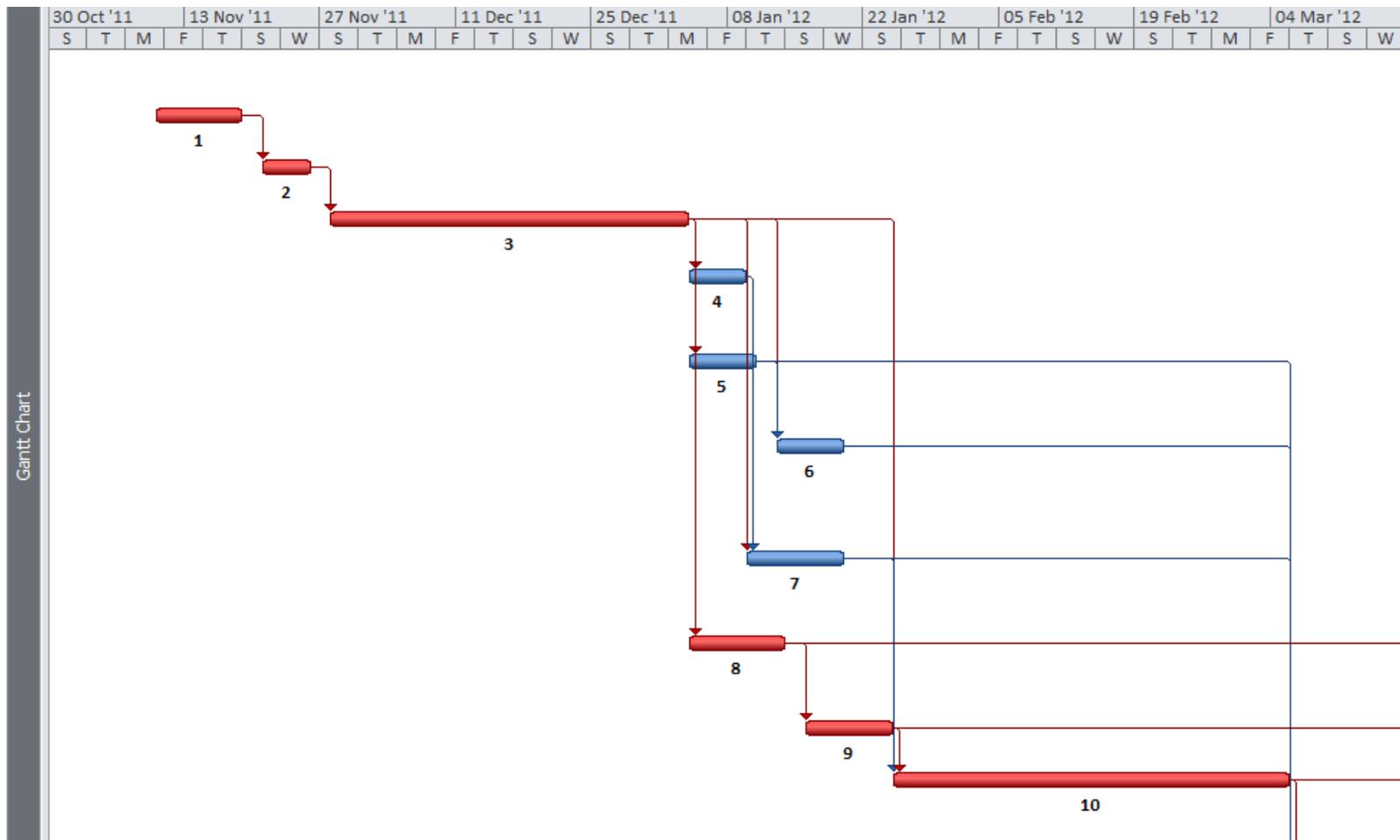
A continuación se muestra la tabla de actividades a realizar con su duración, tiempo de inicio y finalización, las precedencias, y los encargados de realizar la misma.

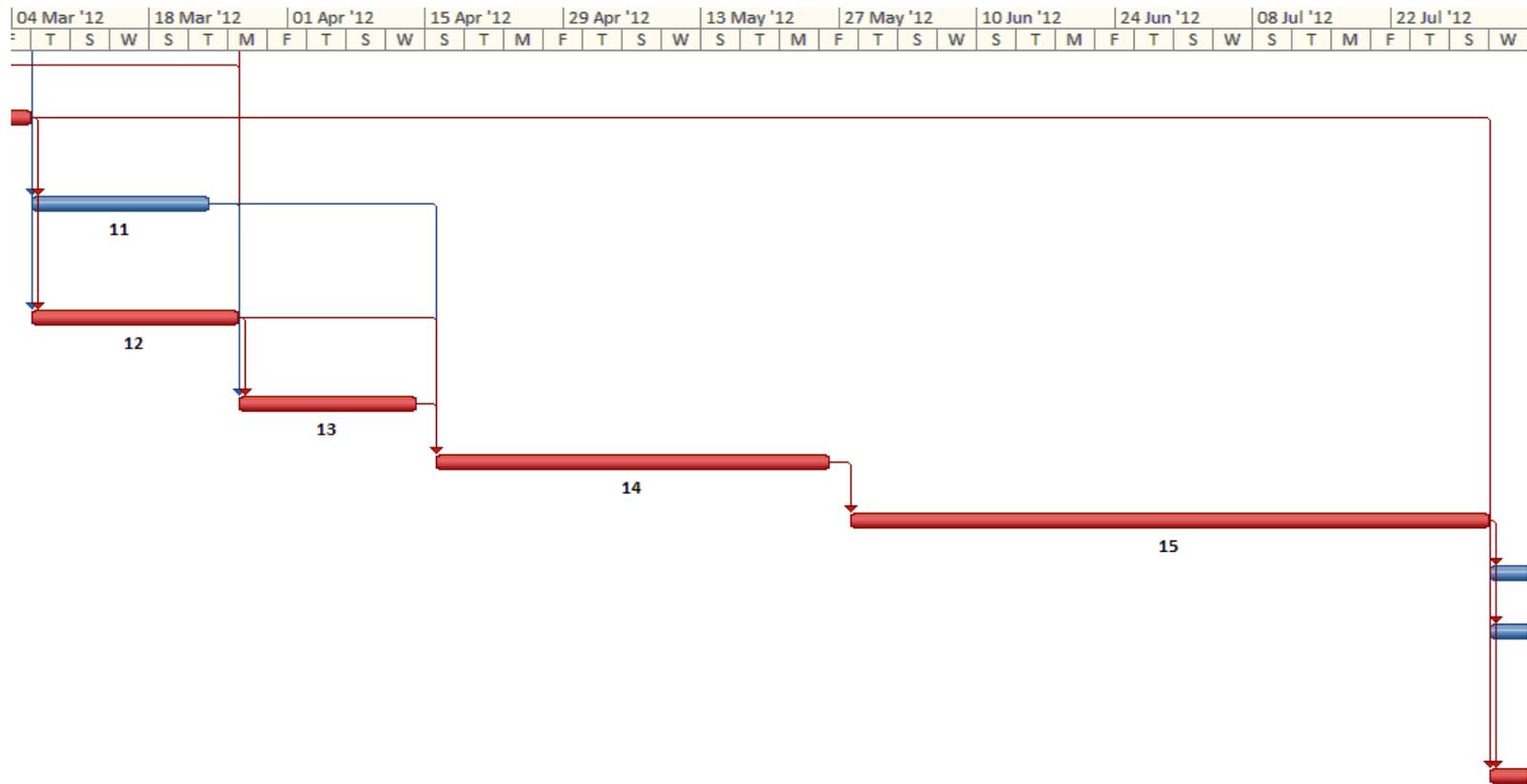
TABLA II  
ACTIVIDADES DEL DIAGRAMA DE GANTT

	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	Análisis de Factibilidad	7 days	10/11/2011	18/11/2011		Chen-Alonso-Villa Lebrun
2	Análisis de Competencia	5 days	21/11/2011	25/11/2011	1	Chen-Alonso-Villa Lebrun
3	Diagrama en Bloques (hardware)	27 days	28/11/2011	03/01/2012	2	Chen-Alonso-Villa Lebrun
4	Investigación del hardware para la Pantalla y Controlador	4 days	04/01/2012	09/01/2012	3	Alonso-Chen
5	Investigación del hardware para Membrana táctil y ADC	5 days	04/01/2012	10/01/2012	3	Villa Lebrun-Chen
6	Investigación del hardware para comunicación inalámbrica	5 days	13/01/2012	19/01/2012	5	Alonso-Chen
7	Investigación del hardware para microprocesador	8 days	10/01/2012	19/01/2012	4	Alonso-Villa Lebrun
8	Investigación de alimentación del circuito	8 days	04/01/2012	13/01/2012	3	Chen-Villa Lebrun
9	Investigación de batería	7 days	16/01/2012	24/01/2012	8	Chen-Villa Lebrun
10	Aprendizaje del lenguaje de programación	29 days	25/01/2012	05/03/2012	7-9	Alonso-Chen-Villa Lebrun
11	Diseño de la placa de comunicación inalámbrica y su interconexión	14 days	06/03/2012	23/03/2012	6-10	Alonso-Chen
12	Diseño de la placa de microprocesador, ADC y RAM/FLASH	15 days	06/03/2012	26/03/2012	5-10	Alonso-Villa Lebrun
13	Diseño de la alimentación.	14 days	27/03/2012	13/04/2012	11-12	Chen-Villa Lebrun

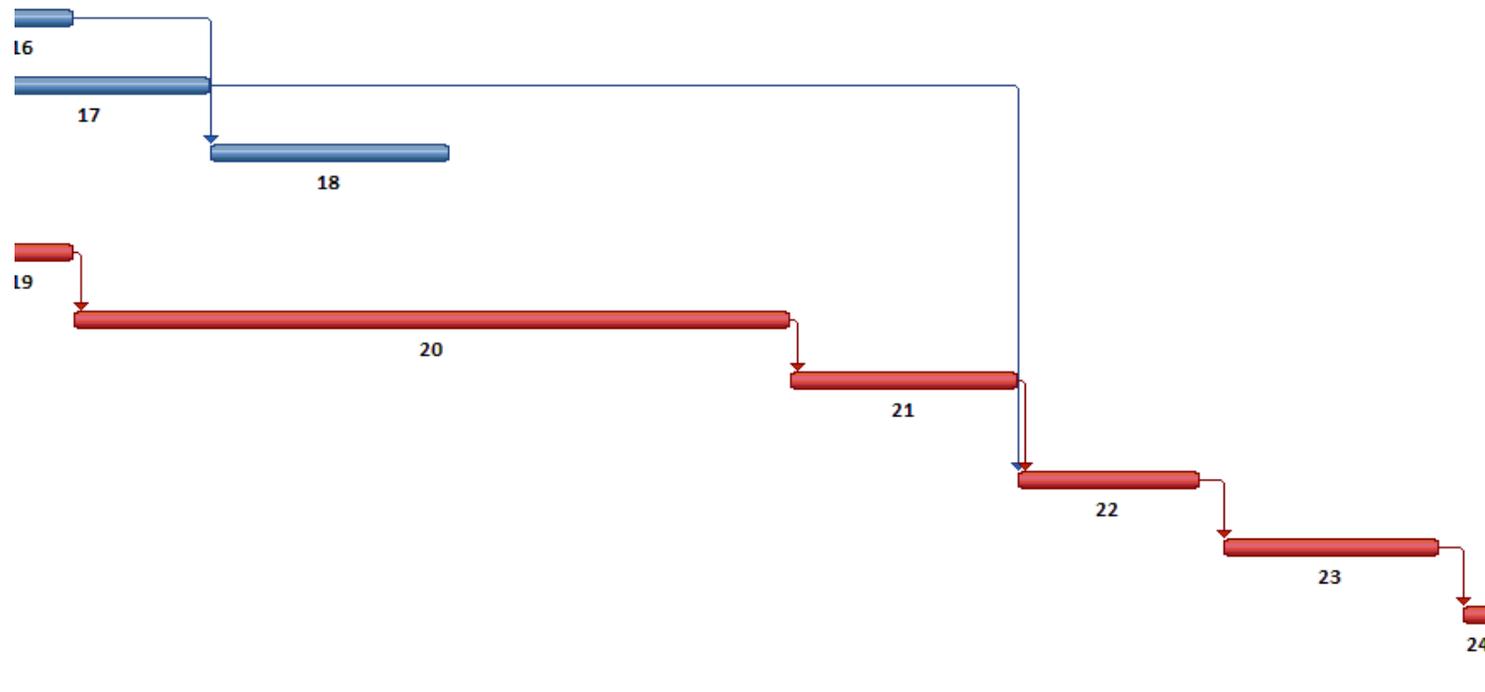
14	Integración de los diseños	30 days	16/04/2012	25/05/2012	13	Alonso-Chen-Villa Lebrun
15	Diseño del PCB final	47 days	28/05/2012	31/07/2012	14	Alonso-Chen-Villa Lebrun
16	Soldar la placa final y prueba de conceptos	7 days	01/08/2012	09/08/2012	15	Alonso-Chen-Villa Lebrun
17	Diseño y armado de gabinete	15 days	01/08/2012	21/08/2012	15	Alonso-Villa Lebrun
18	Realización de pruebas para el correcto funcionamiento	15 days	22/08/2012	11/09/2012	16	Alonso-Chen-Villa Lebrun
19	Diagrama de estados y flujogramas (software)	7 days	01/08/2012	09/08/2012	15	Alonso-Chen-Villa Lebrun
20	Desarrollo de software	45 days	10/08/2012	11/10/2012	19	Alonso-Chen-Villa Lebrun
21	Plan de prueba de módulos y de depuración de software	14 days	12/10/2012	31/10/2012	20	Chen
22	Integración entre software y hardware	12 days	01/11/2012	16/11/2012	17-21	Alonso-Chen-Villa Lebrun
23	Prueba de prototipo armado	15 days	19/11/2012	07/12/2012	22	Alonso-Chen-Villa Lebrun
24	Validación y documentación final	3 days	10/12/2012	12/12/2012	23	Alonso-Chen-Villa Lebrun

Esta misma tabla se puede representar en el gráfico de Gantt como se muestra a continuación:





05 Aug '12				19 Aug '12				02 Sep '12				16 Sep '12				30 Sep '12				14 Oct '12				28 Oct '12				11 Nov '12				25 Nov '12				09 Dec '12					
S	T	M	F	T	S	W	S	T	M	F	T	S	W	S	T	M	F	T	S	W	S	T	M	F	T	S	W	S	T	M	F	T	S	W	S	T	M	F	T	S	W



El tiempo estimado de finalización del proyecto es de 285 días hábiles (es decir sin contar sábados y domingos), quedando determinado por el tiempo estimado de las tareas que se encuentran dentro del camino crítico que son, en orden: 1-2-3-8-9-10-12-13-14-15-19-20-21-22-23-24. El diagrama de PERT no se cumplió, debido a las causas explicadas previamente para la programación de Gantt. De esta forma, el tiempo de proyecto resultó ser aproximadamente de 461 días hábiles, excediendo por 176 días el tiempo estipulado.

*c. Simulación de Montecarlo*

Para poder determinar la confianza con la cual se va a poder terminar el proyecto en el tiempo estimado se realiza un análisis de Montecarlo realizando varias iteraciones del proyecto con sus valores óptimos, promedios y máximos, utilizando una distribución Pert sobre las actividades que representan al camino crítico. A continuación se muestra la tabla de tiempos de las tareas pertenecientes al camino crítico:

TABLA III  
TABLA DE TIEMPOS DE LAS TAREAS A REALIZAR

Tarea	Óptimo	Promedio	Máximo	Media	Varianza
1	6	7	8	7	0,111
2	4	5	6	5	0,111
3	23	27	31	27	1,778
8	6	8	10	8	0,444
9	5	7	9	7	0,444
10	26	29	32	29	1
12	13	15	17	15	0,444
13	12	14	16	14	0,444
14	26	30	34	30	1,778
15	43	47	51	47	1,778
19	6	7	8	7	0,111
20	40	45	50	45	2,778
21	12	14	16	14	0,444
22	11	12	13	12	0,111
23	13	15	17	15	0,444
24	3	3	3	3	0

Con la tabla anterior se realiza la simulación para evaluar el riesgo del que el proyecto no finalice a tiempo. Se muestran los resultados a continuación:

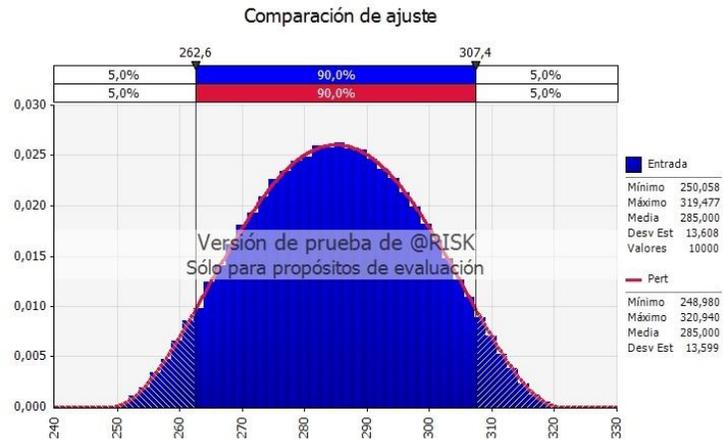


Fig. 1. Resultados de Montecarlo



Fig. 2. Comparación de ajuste

TABLA IV  
VALORES ESTADÍSTICOS DEL ANÁLISIS DE MONTECARLO

	Estadísticas			Percentil	
	Entrada	Pert		Entrada	Pert
Mínimo	250,058	248,98	20%	272,496	272,523
Máximo	319,477	320,94	25%	274,878	274,887
Media	285	285	30%	277,062	277,075
Moda	≈283,654	285,02	35%	279,142	279,144
Mediana	285,006	285,005	40%	281,128	281,137
Desvío Est.	13,608	13,599	45%	283,074	283,082
Asimetría	-0,0002	-0,0015	50%	285,006	285,005
Curtosis	2,3337	2,3333	55%	286,918	286,927
Izquierda X	262,6	262,6	60%	288,865	288,872
Izquierda P	5,00%	5,00%	65%	290,861	290,863
Derecha X	307,4	307,4	70%	292,939	292,932
Derecha P	95,00%	95,00%	75%	295,116	295,117
Dif. X	44,764	44,764	80%	297,485	297,479
Dif. P	90,00%	90,00%	85%	300,115	300,117
1%	256,606	256,604	90%	303,235	303,228
5%	262,618	262,631	95%	307,382	307,353
10%	266,728	266,764	99%	313,383	313,361
15%	269,86	269,882			

Se puede observar de los gráficos obtenidos que el proyecto puede ser finalizado en 285 días hábiles con una probabilidad del 95%, lo cual sigue resultando un tiempo acorde para su finalización.

### *3. Factibilidad económica*

#### *a. Análisis de Mercado*

En la actualidad, el mercado argentino se encuentra prácticamente libre de productos similares, con apenas un número reducido de dispositivos importados siendo utilizados en locales de servicios específicos, como la sucursal de The Coffee Store ubicada en el paseo de compras Norcenter Lifestyle Mall, en la ciudad de Vicente López (los cuales consisten en pantallas táctiles fijas en las mesas).

Esto presenta una ventaja a la hora de realizar una introducción al mercado con este dispositivo, ya que si se logra una alternativa con buenas prestaciones y relativamente conveniente en el marco económico, se puede aprovechar la abundancia de locales que podrían implementar su uso.

#### *a.1. Unidades Vendidas*

Según la guía Óleo<sup>3</sup>, existen más de 5000 locales de comida en Capital Federal y Gran Buenos Aires.

Para poder tener un estimativo de cuántos de estos locales están interesados en la adquisición del producto se realizó una encuesta online (Anexo 1.b). La razón de realizar la encuesta online es para hacer más amena la tarea de completarla y conseguir más respuestas de los locales consultados a fin de obtener una muestra grande y heterogénea a la hora de realizar la aproximación sobre el total de posibles clientes. Con esta encuesta realizada se obtuvo que al 50% de los locales consultados le interesaría utilizar nuestro producto. De todas maneras se asume un margen de error en el resultado obtenido y se opta por un supuesto más conservador.

Así, se asumirá que sólo 2000 locales en todo Capital Federal y Gran Buenos Aires, un 20% menor a los 2500 que surgirían a partir de la encuesta, serán potenciales compradores **iniciales** del producto.

#### *a.2. Análisis de la competencia*

Como se mencionó con anterioridad, la competencia en el mercado local es prácticamente nula. En cuanto al mercado internacional (ya sea si se deseara exportar el producto o si las empresas extranjeras desearan importarlo), existen empresas con productos similares, con diversas diferencias.

---

<sup>3</sup><http://www.guiaoleo.com.ar/> Página dependiente del gobierno provincial de la provincia de Buenos Aires: Guía de restaurants y locales de comida en Capital Federal y Gran Buenos Aires.

TABLA V  
ANÁLISIS DE LA COMPETENCIA

Fabricante	Modelo	Características	Precio
Microsoft-Samsung (USA)	Surface - SUR40	<ul style="list-style-type: none"> <li>&gt; Mesa con superficie táctil.</li> <li>&gt; Se lanzó a fin de 2011.</li> <li>&gt; No disponible en negocios de Sudamérica.</li> <li>&gt; Disponible para todo tipo de servicios, incluyendo conexión a internet y aplicaciones.</li> <li>&gt; Servicio técnico de la empresa.</li> </ul>	u\$s 8000
Conceptic (Israel)	eMenu	<ul style="list-style-type: none"> <li>&gt; Dispositivo fijo a la mesa.</li> <li>&gt; No disponible actualmente en Sudamérica.</li> <li>&gt; Incluye el menú electrónico y juegos para amenizar la espera.</li> <li>&gt; Estadísticas locales y globales.</li> <li>&gt; Servicio técnico de la empresa.</li> </ul>	Mayor a u\$s 350
Shenzhen Baopeng Intelligent Technology Co., Ltd. (Guangdong, China)	Electronic menu wireless 10.1"	<ul style="list-style-type: none"> <li>&gt; Pantalla de 10.1"</li> <li>&gt; Orden mínima de 10 unidades.</li> <li>&gt; Presentes en el mercado sudamericano, pero sin mucha penetración.</li> <li>&gt; Sin aplicaciones extra, excepto reproducción multimedia</li> </ul>	Desde u\$s 250 a u\$s 350
	Electronic menu wireless 7"	<ul style="list-style-type: none"> <li>&gt; Pantalla de 7"</li> <li>&gt; Orden mínima de 10 unidades.</li> <li>&gt; Presentes en el mercado sudamericano, pero sin mucha penetración.</li> <li>&gt; Sólo menú, ninguna aplicación extra.</li> </ul>	Desde u\$s180 a u\$s 250
TouchIt! (Uruguay)	No lanzado al mercado.	<ul style="list-style-type: none"> <li>&gt; Empeñamiento con miras a lanzarse en Uruguay, Brasil y España.</li> <li>&gt; Su dispositivo estará fijado a la mesa.</li> </ul>	No lanzado al mercado.

#### b. Costos

A partir del análisis de la competencia realizado en el inciso anterior, se fija el precio de venta. Dado que la compañía carece de reputación en el mercado, se busca que el precio inicial sea de U\$S 200. Una vez que se gane prestigio e imagen en el mercado gastronómico, se elevará dicho precio a U\$S 250 con mejores prestaciones. La rentabilidad unitaria estipulada es del 40% del precio de venta, con lo cual se obtiene un costo objetivo en el primer año de:

$$Cobjetivo = Pventa - Roportunidad = U\$S200 - U\$S80 = U\$S120$$

El precio de venta es establecido por el mercado, por lo tanto no se tiene grado libertad en dicha variable. Por otro lado, para satisfacer el costo objetivo nuestro único parámetro controlable es la rentabilidad por unidad que se obtiene mediante el diseño del producto.

#### b.1. Proveedores

En primera instancia, se busca minimizar costos en los componentes de mayor valor que son imprescindibles en nuestro producto. Estos se enumeran en orden de precio decreciente como se muestra a continuación. Los precios indicados fueron planteados por la compañía Gentle Industries Limited<sup>4</sup>, localizada en China, por unidad para más de 100 unidades y son sujetos a modificaciones. Cabe aclarar, que los precios que se especifican son en dólares, ya que se conseguirán dichos componentes en China.

<sup>4</sup> <http://www.gentle.hk/> Página de la empresa china de armado de PCB.

- |   |           |
|---|-----------|
| 1) Pantalla TFT con membrana táctil         | (U\$S 30) |
| 2) Microprocesador, RAM, ADC                | (U\$S 15) |
| 3) Batería ion-litio recargable             | (U\$S 12) |
| 4) Fuente switching                         | (U\$S 10) |
| 5) Conexión inalámbrica                     | (U\$S 5)  |
| 6) Precio de envío, armado y soldado de PCB | (U\$S 2)  |

Finalmente, se terceriza la fabricación del gabinete plástico del menú táctil en Argentina con un costo aproximado de U\$S 5 por unidad. El ensamblaje del producto final, se realizará mediante nuestro personal capacitado. De esta manera, se determina al producto como parte de la industria Argentina, lo cual genera ventajas impositivas.

De esta manera se fija un costo aproximado de U\$S 79 por unidad.

Debido a que el costo por unidad es menor al costo objetivo, se obtiene una rentabilidad del:

$$\text{rentabilidad} = \frac{\text{U\$S } 200 - \text{U\$S } 79}{\text{U\$S } 200} = 60,5\%$$

Esta rentabilidad representa una obtención de ganancia apreciable.

### c. Análisis de gastos

#### c.1. Inversión inicial

En la siguiente tabla se especifican los costos para la inicialización de la empresa, dentro de los cuales la más importante y más tiempo requiere es la homologación del producto electrónico. Como se especificará en el análisis legal, por ley se debe cumplir la normativa IRAM (o sus equivalentes internacionales, como el IEC) para que el producto electrónico sea comercializable.

TABLA VI  
INVERSIÓN INICIAL

Inicialización de la empresa	Precio en AR\$
Homologación	6000
Capacitación	8000
Equipos electrónicos para desarrollo	6000
Bienes de usos	2000
<b>Inversión inicial</b>	<b>22000</b>

Esta inversión es la realizada en el año 0 del proyecto y resulta suficientemente bajo para no recurrir a inversionistas externos. Por lo tanto, se financiará por el equipo de diseño.

#### c.2. Costos fijos

Inicialmente se alquilará un área de 75m<sup>2</sup> en Microcentro, que será utilizado como instalación y se instalarán los dispositivos electrónicos para desarrollo explicados previamente. Estos costos entran en la categoría de costos fijos, ya que no sufrirán variaciones en el tiempo de vida del proyecto:

TABLA VII  
COSTOS FIJOS MENSUALES

Costos fijos mensuales	Precio mensual en AR\$ (año 1)	Precio mensual en AR\$ (año 2 a 5)
Alquiler (mas expensas)	4000	4000
Sueldos	0	6500
Cargas Sociales	0	2600
Sueldos Socios (3 socios)	21000	24000
Teléfono + Internet	300	300
Electricidad	350	400
Gas	180	180
Agua	180	220
<b>Costo fijos totales</b>	<b>26010</b>	<b>38200</b>

Dado que se planea conseguir la mayoría de las partes (pantalla) en el extranjero y tercerizar la fabricación de las restantes (placa SOM, carcasa, ambas diseñadas por los integrantes de la empresa pero fabricadas en el extranjero), para luego ensamblarlas en Argentina, en principio sólo se contará con tres socios. Esto se debe a que sólo sería necesario, el primer año, ensamblar un promedio menor a dos unidades diarias, lo cual es factible de realizar para dicho número de personas.

Luego del primer año se espera un aumento de la cantidad de unidades vendidas, por lo tanto se requerirá la contratación de un empleado capacitado adicional para cubrir la demanda creciente.

### c.3. Costos variables

El precio inicial de venta de U\$S 200 incluye el valor del IVA del 21% para los clientes ya que son consumidores finales. Esto quiere decir, que el precio percibido por unidad es aproximadamente de U\$S 165. En otras palabras, la rentabilidad real analizada es levemente menor al calculado previamente, como se muestra a continuación:

$$\text{rentabilidad percibida} = \frac{\text{U\$S } 165 - \text{U\$S } 79}{\text{U\$S } 165} = 52\%$$

El precio del dólar se consideró de AR\$ 5.48, por lo tanto el precio de los materiales por unidad de venta es de AR\$ 433. Este costo variará dependiendo del valor de la tasa de cambio que se considerará constante para los 5 años de tiempo de vida del proyecto.

TABLA VIII  
COSTO UNITARIO

Costo variables	Precio en AR\$
Precio de los materiales	433

Como se detallaron en las tablas de costos anteriormente, se busca tener un gasto fijo bajo cubriendo lo indispensable y para recuperar rápidamente la inversión inicial. No obstante, a medida que el tiempo de vida del proyecto avance y la demanda crezca debido a que se ganen más contactos en el mercado, será necesario contratar a un personal más para llevar a cabo el aumento de unidades de producción y satisfacer la demanda creciente. En el proceso de crecimiento del proyecto, al esperar más clientes se puede aumentar el precio de venta a los U\$S 250 alcanzando el precio de la competencia y ofreciendo mejor calidad de servicio y prestaciones, debido a que se cubre el área local. Este nuevo flujo podrá ser utilizado para el continuo desarrollo y mejoramiento del producto en un futuro.

## d. Análisis económico financiero

En nuestro caso, se realiza un estudio de vida del proyecto de 5 años. Es importante destacar, que a partir del año 2, cuando se incorpora un empleado adicional, se eleva el precio unitario a los U\$S 250 explicado previamente. Por otro lado, el precio de venta inicial por unidad de U\$S 200 incluye el valor del IVA para los clientes, debido a que son consumidores finales, al igual que el precio de U\$S250 luego del año 1. Mientras que en los costos variables, el valor de AR\$433 por unidad vendida también incluye el IVA, debido al carácter de consumidor final que asume la empresa al tercerizar la construcción del PCB. En cuanto al impuesto a la ganancia se tomó una tasa del 35%. Los resultados obtenidos se resumen en la siguiente tabla:

TABLA IX  
FLUJO DE CAJA DEL PROYECTO

	<b>Año 1 (ARS)</b>	<b>Año 2 (ARS)</b>	<b>Año 3 (ARS)</b>	<b>Año 4 (ARS)</b>	<b>Año 5 (ARS)</b>
<b>Unidades</b>	650	680	710	730	730
<b>Precio de venta</b>	904	1132	1132	1132	1132
<b>Ventas</b>	587600	769760	803720	826360	826360
<b>Costo Variables</b>	281450	294440	307430	316090	316090
<b>Costo Fijos</b>	312120	458400	458400	458400	458400
<b>Costos Varios</b>	6242,4	9168	9168	9168	9168
<b>Ingreso Bruto</b>	306150	475320	496290	510270	510270
<b>Utilidad antes de impuestos a la ganancia</b>	-12212,4	7752	28722	42702	42702
<b>Impuesto a la ganancia</b>	-4274,34	2713,2	10052,7	14945,7	14945,7
<b>Utilidad neta</b>	-7938,06	5038,8	18669,3	27756,3	27756,3

A partir de la tabla de resultados, se determina tomando en consideración la inversión inicial en el año 0 de AR\$ 22000 mediante la fórmula del VAN, teniendo en cuenta una tasa de interés del 10%:

$$VAN = Inv_0 + \sum_{i=1}^n \frac{FF_i}{(1+t_i)^i} = AR\$ 22879.$$

Este resultado de VAN es positivo resultando un proyecto viable. También se calcula el TIR igualando la VAN a cero, lo cual equivale a un valor del 31%. Dicho valor contrastado con la tasa de interés del 10% brindada por el banco como costo de oportunidad indica que es conveniente llevar a cabo el proyecto.

Sin embargo para ser más realistas, se evalúa la TIRM, donde la tasa de reinversión se considera del 20%, dando un valor del 27%, que a su vez sigue siendo ampliamente superior a la tasa de interés.

*e. Resultados*

Una vez realizado el estudio económico-financiero, se concluye que el proyecto en cuestión es económicamente rentable, considerando que las inversiones iniciales son provenientes de un grupo de diseño. Teniendo una tasa de interés efectiva anual del 10%, la cual incluye inflación.

A continuación se resumen los parámetros de interés del proyecto:

- Inversión inicial = AR\$ 22000
- VAN (10%) = AR\$ 22879
- TIR = 31%
- TIRM (20%) = 27%

Es importante resaltar, que el periodo de repago sucede, tomando una interpolación lineal con respecto a la ganancia anual, aproximadamente en el sexto mes del año 1. Esto quiere decir, que al cabo de un año y seis meses se recupera la inversión. También se tomó en cuenta, que normalmente los proyectos de dispositivos electrónicos tienen una longitud de vida promedio de 5 años.

#### *4. Factibilidad legal y responsabilidad civil*

El territorio en el que se desea comenzar a desarrollarse corresponde al mercado de la República Argentina, por lo que las normas que acatar inicialmente serán las de dicho territorio. Por lo tanto, será la normativa IRAM la que regirá la producción de los menús, junto a las demás leyes nacionales correspondientes.

La resolución 92 de 1998 de la Secretaría de Industria, Comercio y Minería estipula que:

Artículo 1°- Sólo se podrá comercializar en el país el equipamiento eléctrico de baja tensión que cumpla con los requisitos esenciales de seguridad que se detallan en el Anexo I en DOS (2) planillas que forman parte de la presente resolución, considerándose comercialización toda transferencia aún como parte de un bien mayor.

Art. 2°- A los fines de la presente resolución se entiende por equipamiento eléctrico de baja tensión a los artefactos, aparatos o materiales eléctricos destinados a una instalación eléctrica o formando parte de ella, que tenga una tensión nominal de hasta MIL (1000) Volt en corriente alterna eficaz o hasta MIL QUINIENTOS (1500) Volt en corriente continua.

Art. 3°- Los fabricantes, importadores, distribuidores, mayoristas y minoristas de los productos mencionados en el artículo anterior deberán hacer certificar o exigir la certificación del cumplimiento de los requisitos esenciales de seguridad mencionados en el Artículo 1° mediante una certificación de seguridad de producto, otorgada por un organismo de certificación acreditado por el Organismo Argentino de Acreditación (O.A.A.) conforme con el Decreto N° 1474/94. Esta certificación se implementará siguiendo el procedimiento y los plazos establecidos en el Anexo II que en DOS (2) planillas forman parte de la presente resolución.

Estos requisitos se considerarán plenamente asegurados si se satisfacen las exigencias de seguridad establecidas en las normas IRAM o IEC aplicables, correspondientes al equipamiento eléctrico considerado.

Los productos certificados según lo establecido precedentemente ostentarán un sello indeleble que permita identificar inequívocamente tal circunstancia.

En principio, del Artículo 2° se observa que el presente producto aplica a la normativa de baja tensión. Luego, del Artículo 3° se sabe que deberá certificarse el cumplimiento de los requisitos mencionados en el Artículo 1°. Este último, remite al Anexo I para consultar tales requisitos:

ANEXO I  
REQUISITOS ESENCIALES DE SEGURIDAD DEL EQUIPAMIENTO ELECTRICO DE  
BAJA TENSION

1.- Condiciones Generales

a) Las características fundamentales de cuyo conocimiento y observancia dependa la utilización acorde con el destino y el empleo seguro del equipamiento eléctrico garantizarán sobre el mismo o, cuando esto no sea posible, en la nota que lo acompañe, en ambos casos redactadas en idioma nacional.

b) El país de origen, la razón social del fabricante o la marca comercial registrada, su domicilio legal, la razón social y domicilio legal del importador y del distribuidor en el país y el modelo del producto, irán colocados de manera distinguible e indeleble en el equipamiento eléctrico o, no siendo esto posible, al menos la marca comercial registrada y el modelo irán colocados en el equipamiento eléctrico y el resto de la información en el envase primario.

c) El equipamiento eléctrico y sus partes constitutivas se fabricarán de modo que permitan una conexión segura y adecuada.

d) El equipamiento eléctrico habrá de diseñarse y fabricarse de modo que quede garantizada la protección contra los peligros a que se refieren los puntos 2 y 3, a condición de que su uso sea el indicado por el fabricante y sea objeto de adecuado mantenimiento.

e) La clase de aislación será la adecuada para las condiciones de utilización previstas, quedando expresamente prohibidas las clases de aislación 0 y 0I.

2.- Protección contra los peligros originados en el propio equipamiento eléctrico

Se preverán medidas de índole técnica conforme al punto 1, a fin de que:

a) Las personas y los animales domésticos queden adecuadamente protegidos contra el riesgo de heridas y otros daños que puedan sufrir a causa de contactos directos o indirectos.

b) No produzcan temperaturas, arcos o radiaciones peligrosas.

c) Se proteja convenientemente a las personas, animales domésticos y los bienes contra los peligros de naturaleza no eléctrica causados por el equipamiento eléctrico.

3.- Protección contra los peligros causados por efecto de influencias exteriores sobre el equipamiento eléctrico.

Se establecerán medidas de orden técnico conforme al punto 1, a fin de que:

a) El equipamiento eléctrico responda a las exigencias mecánicas previstas con el objeto de que no corran peligro las personas, los animales domésticos y los bienes.

b) El equipamiento eléctrico resista las influencias no mecánicas en las condiciones previstas de medio ambiente con objeto de que no corran peligro las personas, los animales domésticos y los bienes.

c) El equipamiento eléctrico no ponga en peligro a las personas, los animales domésticos y los bienes en las condiciones previstas de sobrecarga.

De las tres secciones que componen dicho Anexo, la primera establece condiciones generales sobre las que se trabajará. De las mismas se desprenden condiciones para la información básica que se deberá incluir como indicativo para operar el producto. También indica que la fabricación del producto será acorde a las secciones 2 y 3, permitiendo una conexión segura y adecuada entre las partes constitutivas, así como dando garantía de protección ante los peligros indicados por las siguientes secciones del Anexo. Por último, indica

que la clase de aislación del producto deberá ser la adecuada para la aplicación pertinente, prohibiendo las clases de aislación 0 y 0I.

Para analizar estas restricciones, se recurre a la norma IEC 61140 (dado que la resolución indicaba que estos organismos poseían normas que, satisfechas, implicarían la completitud de los requisitos nacionales necesarios).

En dicho documento, se diferencian las clases principales de aislamiento, en función de las tensiones de funcionamiento y aplicaciones de cada producto.

*Clase 0*

Diseñados para zonas secas. Cuentan con un único nivel de aislamiento. No tienen protección de las tierras. La IEC está en proceso de prohibir totalmente esta clase. Están prohibidos en el Reino Unido desde 1975.

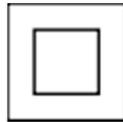
*Clase I*

Cuentan con protección de descarga a tierra (a través de un tercer cable que se conecta a la línea de tierra de una instalación). Se deben proteger también con un dispositivo de protección de sobrecorriente (como un fusible o interruptor termomagnético) y un interruptor diferencial.



*Clase II*

Para no requerir protección de descarga a tierra, todas las partes que involucren tensiones peligrosas son protegidas con doble aislamiento o aislamiento reforzado.



*Clase III*

Estos aparatos están diseñados para ser alimentados por una fuente de alimentación SELV (Separated or Safety Extra-Low Voltage). Por esta razón, no necesitan el aislamiento de la clase II, dado que los niveles de tensión que manejan no presentan peligro para los usuarios.



Se puede ver que el producto a diseñar corresponde a la Clase III, por lo que sólo se requerirá una aislación que proteja eléctrica y mecánicamente a los circuitos pertinentes, sin peligro para los usuarios de sufrir un shock eléctrico.

De la sección 1 no se desprendió ningún impedimento para la fabricación del producto, por lo que no presenta inconvenientes para la producción del mismo.

La sección 2 del Anexo I trata sobre la protección necesaria contra los peligros originados en el propio equipamiento eléctrico. De dicho tema, indica que deberán tomarse medidas técnicas a fin de proteger a personas y animales domésticos que puedan sufrir daños a partir de contactos directos o indirectos con el producto. También indica que no deberán producirse temperaturas, arcos o radiaciones peligrosas.

Sobre esto, puede asumirse que en un local de comidas no existirá la posibilidad de un contacto del producto con animales domésticos. De todos modos, esta prohibición (de permitir este tipo de contacto con animales) deberá indicarse al usuario, junto a las demás indicaciones estipuladas por la sección 1.

En cuanto a la protección de las personas, la sección 7.4.2 de la norma 61140 de IEC indica que los elementos de Clase III no necesitarán protecciones excesivas, dadas las bajas tensiones con las que trabajará, estipuladas en la sección 7.4.1. Por esto, alcanzará con respetar dicha sección de la norma y una aislación simple para cumplir esta reglamentación.

La sección 3 del Anexo I, abarca la protección necesaria para los usuarios contra peligros causados por influencias exteriores sobre el equipo.

Dicha sección indica que tanto las influencias mecánicas como no mecánicas, en las condiciones previstas de medio ambiente, así como una condición prevista de sobrecarga del producto, no pongan en peligro a las personas, animales domésticos ni a los bienes.

En cuanto a esta sección, puede asumirse lo mismo que en la anterior con respecto al contacto con animales.

En cuanto los efectos de influencias mecánicas y no mecánicas, así como los resultados de sobrecargas previstas, deberán tenerse en cuenta estos aspectos a la hora del diseño del hardware así como de la planificación de sus análisis de fallas.

Tomando las consideraciones mencionadas, la norma IEC indica que el producto sería factible de fabricar, con el valor agregado de estar cumpliendo un estándar internacional. Uno de los puntos a considerar, por ejemplo, será el empleo de fuentes switching, dado que su alta eficiencia energética es necesaria para ingresar a mercados extranjeros, como el europeo.

En cuanto al software a utilizar, el mismo será corrido sobre un kernel de Linux. Así, para verificar si es factible comercializar libremente este sistema operativo, debe remitirse a la licencia GNU GPL (GNU General Public License). La misma se basa en cuatro principios:

- The freedom to use the software for any purpose,
- The freedom to change the software to suit your needs,
- The freedom to share the software with your friends and neighbors, and
- The freedom to share the changes you make.

En primer lugar, la licencia establece que no se cobra el uso de aplicaciones liberadas bajo la misma.

Por otro lado, está establecido que cualquier modificación sobre el código adquirido debe liberarse bajo la misma licencia libre, sin cobrar por los derechos sobre las modificaciones.

No así, cualquier aplicación que se desarrolle sobre sistemas operativos liberados bajo esta licencia, pueden patentarse, cerrarse (no liberarse) y venderse. Por ejemplo, los servidores CISCO, cuyo software corre sobre el kernel de Linux, cobran regalías sobre el software desarrollado por la empresa, sin liberar el código (aunque no cobran regalías por el uso de Linux a sus clientes, dado que la licencia GNU GPL lo impide).

Así, la venta del menú no podrá incluir ingresos por el Linux sobre el que corra el software, pero sí sobre el software en sí mismo.

## VIII. INGENIERÍA DE DETALLE

Una vez verificada la factibilidad de la realización de proyecto, se procede a diseñar el hardware y el software que conformarán el producto completo.

## 1. Hardware

## a. Diagrama de bloques del hardware

En la siguiente figura se detalla un diagrama de bloques del hardware que compondrá al producto en general.

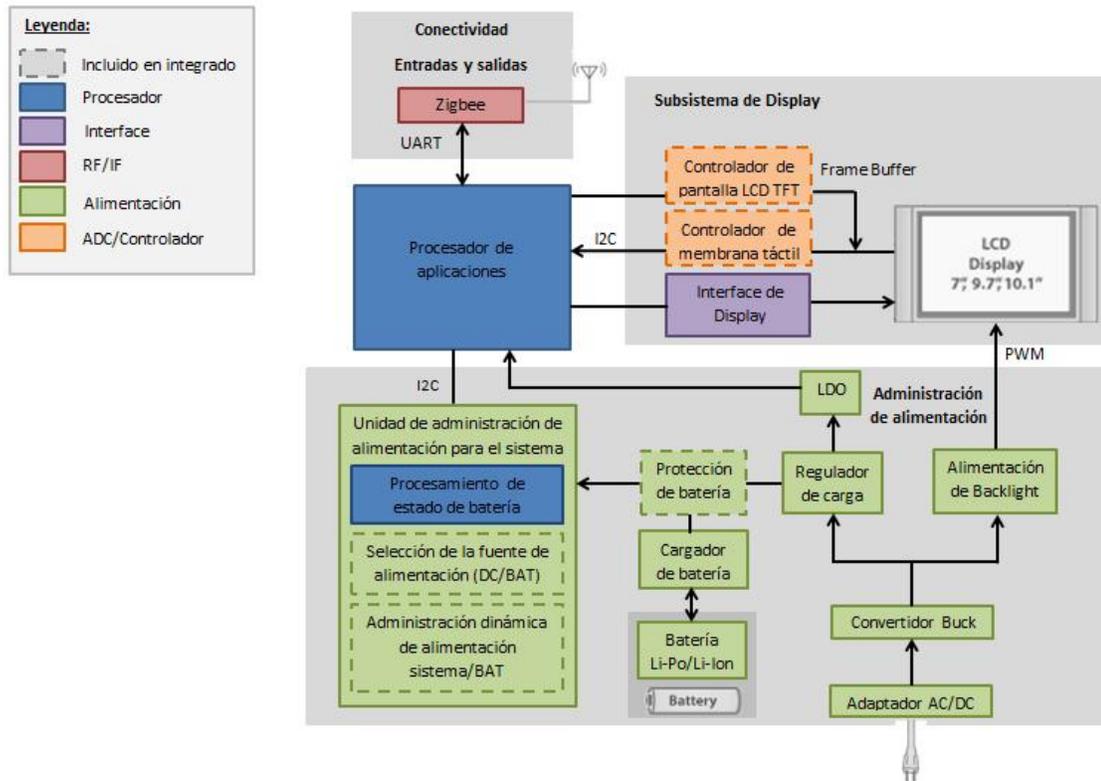


Fig. 3. Diagrama de bloques del sistema completo

Inspeccionando la figura anterior, se resaltan los siguientes módulos de mayor importancia: módulo de alimentación (Convertidores de tensión y Batería), módulo de administración de alimentación para el sistema (Controlador de carga), módulo de procesador de aplicaciones (Microprocesador), módulo de Display (Interfaz Software-Usuario) y módulo de comunicación Xbee (Conexión inalámbrica). Las mismas partes se explicarán con mayor detalle a continuación.

### b. Descripción detallada de cada bloque

Luego del análisis del mercado realizado para el menú digital, se visualizó que de los principales requerimientos de los clientes, donde nuestro producto se destacaría son en el bajo costo y en el bajo consumo. Esto concuerda con el peso seleccionado en sendos requerimientos en la casa de calidad, presentando la principal diferencia frente a nuestros competidores. En otras palabras, el criterio tomado para la elección de componentes en el Hardware fue realizado priorizando los efectos deseados, como la alta eficiencia del sistema, logrando este objetivo con un bajo consumo. Por otro lado, surge el efecto indeseado del alto costo asociado en forma inherente a dicha mejora. Es por esta razón, que se tiene una relación de compromiso entre bajo costo y bajo consumo del dispositivo, lo cual dará a una gama de alternativas que serán discutidos a continuación. Otro parámetro que juega un rol importante en las aplicaciones de dispositivos portátiles es la portabilidad, bajo peso y tamaño cómodo del menú. Por lo tanto, los principales integrados son seleccionados preferentemente en encapsulados de montaje superficial, lo cual es una ventaja si se tiene una máquina pick-and-place en la producción facilitando las velocidades de producción al requerir menor tiempo de soldadura.

Debido a que nuestra aplicación se centra en el área de entretenimiento, el factor preponderante es el del costo y fundamentalmente el de fabricación. Por lo tanto, no requiere de mucha precisión por ejemplo en los conversores ADC de la pantalla táctil, el cual sería un aspecto crítico en otras aplicaciones.

#### Módulo de alimentación:

El módulo de fuentes de alimentación consiste básicamente en convertir alimentación de línea a los niveles de tensión utilizados en el dispositivo. En primera instancia, se utiliza un transformador para cambiar la corriente alterna de línea en continua. Cabe destacar, que de esta forma se puede utilizar el mismo producto en distintos lugares del mundo, ya que el adaptador se encuentra externamente pudiéndose cambiar y adaptar de las tensiones nominales de 220V ó 110V a un nivel de continua aceptada por el dispositivo. A continuación se muestra el resultado del circuito diseñado posterior al transformador aceptando desde 8VDC hasta 40VDC y convirtiéndola a 5V de alimentación fija para el sistema. De esta manera, se puede utilizar una gran variedad de adaptadores AC/DC para el dispositivo.

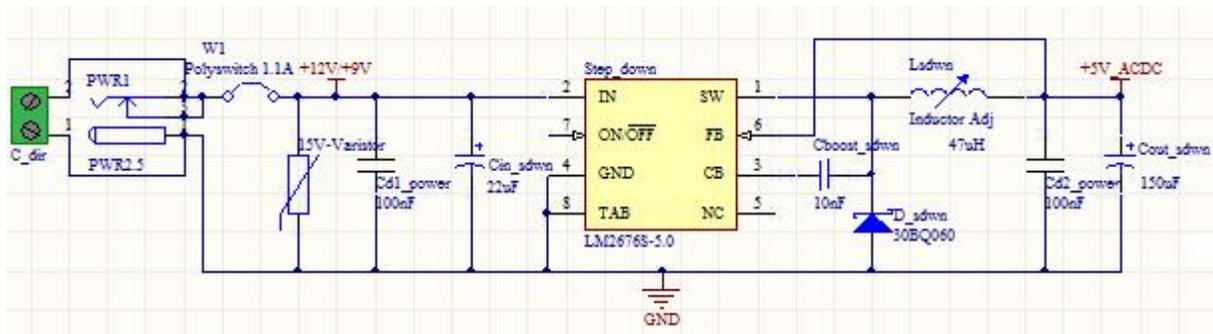


Fig. 4. Esquemático de la fuente switching para adaptar la tensión del adaptador a los 5V necesarios.

En nuestro caso, se trabaja con niveles estándares de tensión de 5V y de 3,3V. Cabe destacar, que en esta etapa de alimentación es importante la eficiencia de los convertidores de tensión, ya que dicha eficacia se reflejará en un tiempo mayor de autonomía. No obstante, se debe tener en cuenta, el efecto inherente del costo y dimensión de los componentes adicionales que viene ligado con dicho ahorro de energía. A grandes rasgos, tenemos 3 categorías de componentes, que nos ofrecen solución al problema de alimentación: reguladores comunes, reguladores lineales LDO (Low-Dropout) y fuentes switching. Debido a que nuestra aplicación puede requerir de picos de corrientes de hasta 1A y se enfoca en el bajo consumo del dispositivo, los

reguladores comunes como el LM7805 si bien son fáciles de utilizar y casi no requieren de componentes externos, no presentan una solución viable por la falta de eficiencia del dispositivo.

El siguiente grupo de convertidores de tensión, son los LDO, los cuales presentan eficiencia media hasta valores comparables con los de las fuentes switching que son del orden de 80% hasta cerca de los 95% de eficiencia. En la práctica se justifica tener fuentes switching para aplicaciones de media hasta alta potencia, mientras que nuestro sistema trabaja en el borde de potencia media y baja. Posteriormente se explicará con mayor énfasis el tema de consumo de corrientes, sin embargo, cabe mencionar los siguientes usos del módulo de alimentación:

- Fuente de 3,3V requiere en pleno uso entregar corrientes de aproximadamente 500mA
- Fuente de 5V necesita entregar una corriente mínima de 500mA hasta 1A.
- Fuente de 9,9V para el uso del Backlight de la pantalla con corriente media de 200mA.

Es por esta razón, que se decidió utilizar un regulador LDO para la fuente de 3,3V, debido a que se consume en el peor de los casos 500mA y esto sucede cuando se transmiten datos inalámbricamente y durante un período corto.

Por otro lado, se tomó el criterio de utilizar una fuente switching como fuente de alimentación de 5V, ya que es una aplicación de mayor potencia. En nuestro caso, es necesario pasar de los 4,4V de la salida fijada por el regulador de carga a los 9,9V del Backlight para la pantalla mediante un convertidor de tensión tipo Boost o Step-Up.

#### Fuente switching de alimentación de entrada a 5V:

El estudio de la eficiencia de convertidores switching es exhaustivo, debido a que no solamente el integrado define mediante su topología interna la eficiencia total del dispositivo, sino que el diseño de la inductancia, el diodo y los capacitores de entrada y sobre todo el capacitor de salida definen también juegan un rol importante. Además, es importante tener en cuenta el layout del circuito, ya puede implicar lazos de corrientes o elementos parásitos indeseados en el diseño disminuyendo notoriamente la eficiencia del convertidor. Es por esta razón, que se seleccionó dentro de las fuentes Boost propuestos por Texas Instrument el integrado perteneciente a la familia LM267X, con el cual se tiene experiencia técnica previa. No obstante, se muestra mediante el WEBENCH DESIGNER<sup>5</sup> proporcionado por la página web de Texas Instrument en la siguiente figura, la amplia gama de componentes que satisfacen las características de fuente switching que necesitamos.

En nuestro caso, eligiendo el componente LM2676-5.0, nos permite obtener una tensión de salida fija de 5V, requiriendo menor ajuste de componentes externos. Del gráfico se puede ver claramente, que Texas Instrument no ofrece integrados con eficiencias teóricas mayores a 82%, no obstante, en la hoja de datos del LM2676 se indica una eficiencia máxima del 96%, debido a que el programa realiza una búsqueda con integrados de salida ajustable, para satisfacer el rango de características deseadas por los diseñadores.

La principal discrepancia entre integrados de fuentes Buck con salida fija y ajustable, es que la primera tiene internamente componentes que garantizan dicha salida, mientras que los componentes externos seleccionados solamente afectarán el rango de entrada, la corriente de salida y el Ripple de salida del dispositivo. Con el fin de lograr la tensión fija de 5V, se sacrifica el tamaño del componente, requiriendo un tamaño de 3mm\*3mm para un encapsulado del tipo QFN de 10 pines.

Los componentes principales como la bobina y los capacitores del orden de los uF con encapsulados SMD tienen la ventaja de disminuir los elementos parásitos que afectan en gran medida las fuentes switching.

---

<sup>5</sup> <http://www.ti.com/ww/en/home/three-col/>



De la misma forma que se seleccionó la fuente switching para los 5V, se procede para elegir el integrado para el convertidor Boost de la pantalla cumpliendo con la mayor eficiencia, el menor espacio y el menor costo sugerido por Texas Instrument. El circuito utilizado para llevar la tensión del regulador de carga a los 9,9V imprescindibles de Backlight, se muestra en la siguiente figura:

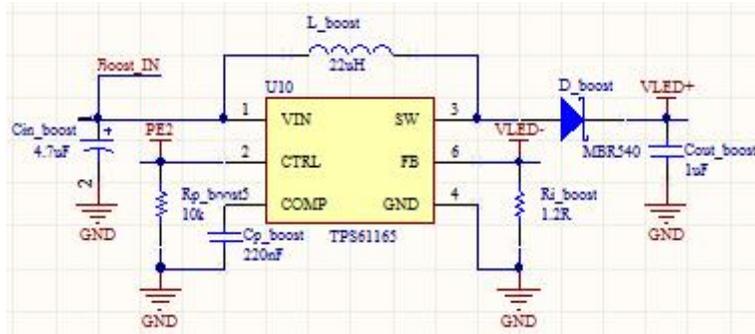


Fig. 6. Esquemático del convertidor Boost para el Backlight de la pantalla.

#### Regulador LDO de 3,3V:

Para filtrar dentro de todos los integrados posibles de LDO, los que no serán seleccionados son aquellos componentes regulables. Estos se usan generalmente para obtener a la salida de los integrados niveles de tensiones no convencionales, mediante el uso de un divisor resistivo externo. Por lo tanto, ya presentan la desventaja de requerir 2 resistencias más, los cuales deben ser seleccionados de manera que no disipen potencia, para mantener la eficiencia alta. En nuestro caso, al querer un nivel usual de 3,3V, los fabricantes desarrollaron integrados con salida fija para facilitar el diseño y ofrecen generalmente los valores de salida de 3,3V, 5V y 12V.

Para poder realizar una comparación más equitativa entre los componentes, se tomaron en cuenta el precio ofrecido por un mismo proveedor<sup>6</sup> y para el tipo de encapsulado de montaje superficial. A continuación, se presentan 2 de los integrados más destacados dentro de los componentes comparados:

TABLA X

COMPARACIÓN DE LAS PRINCIPALES CARACTERÍSTICAS DE REGULADORES LDO

LDO 3,3V	Tamaño	Precio US\$	Dropout	Estabilidad	Iq
TPS73733	SOT-223-6PIN	1,91	130mV @ 1A	1uF	20nA
LT1963	SOT-223-6PIN	5,49	340mV @ 1,5A	10uF	<1uA

Cabe destacar, que se estudiaron otros componentes como candidatos a reguladores LDO, como el NCP605 de On-Semi, pero no se encontraron características que consideramos importantes como tensión de Dropout e Iq o bien ofrecían un gráfico de dichos parámetros bajo circunstancias típicas, por lo cual no nos garantizaba buena eficiencia del dispositivo y fueron descartados. A la hora de comparar LDOs, es importante la tensión de Dropout, que sucede cuando se exige mayor tensión en el Gate de un FET debido a un aumento de corriente de salida, lo cual causa bajo condiciones de estado estacionario que el FET se comporte como una resistencia menor a la que permite el estado de saturación. Extendiendo dicho tema, se obtiene de algunas hojas de datos, el valor de Dropout, el cual viene dado por la tensión de Dropout para cierta corriente de salida, como en el caso del TPS73733: 130mV para una corriente de salida de 1A.

Este dato indica que la resistencia equivalente del FET dentro del LDO es de:  $\frac{V_{DO}}{I_o} = \frac{130mV}{1A} = 0,13\Omega$ . A partir de dicho valor, se puede suponer que para una aplicación que requiera 100mA de salida de dicho LDO, se

<sup>6</sup> Corporación Digikey para pedidos internacionales: <http://www.digikey.com/?curr=USD>

obtendrá una tensión de Dropout de  $V_{DO} = I_o \cdot 0,13\Omega = 100mA \cdot 0,13\Omega = 13mV$ . A continuación se resume los cálculos en la siguiente tabla de características:

TABLA XI  
COMPARACIÓN DE RESISTENCIA EQUIVALENTE DE REGULADORES LDO

VDO	I <sub>o</sub>	Resistencia equivalente [inherente]
130mV	1A	0,13Ω
13mV	0,1A	0,13Ω

Luego de calcular el parámetro de la resistencia equivalente, el cual es un parámetro inherente, se halla a partir de la corriente necesaria de la aplicación, la corriente de dropout que se tendrá en el integrado. Por lo tanto, para tener menor caída de Dropout y disipar menor potencia, se desea una resistencia equivalente baja o bien un valor pequeño de Dropout. Vemos que para el caso del LT1963, la resistencia equivalente es alrededor de los 0,227Ω, presentando una leve pérdida de potencia en comparación con el TPS73733.

También se debe tener en cuenta que la eficiencia viene dada por la energía total de pérdida. Por lo tanto no alcanza evaluar solamente la tensión de Dropout, sino que se debe considerar la corriente I<sub>q</sub>. El término I<sub>q</sub> (Quiescent current), es la corriente que fluye por el sistema, cuando no existe carga presente, lo que causa una diferencia entre la corriente de entrada y la de salida. Por lo tanto, se desea que dicha corriente sea lo mínimo posible para tener una buena eficiencia en el regulador LDO. Nuevamente vemos que el integrado TPS73773 de Texas Instrument garantiza menor fuga de I<sub>q</sub> que el componente de Linear Technology.

Un último factor a considerar es el capacitor de salida necesario para asegurar la estabilidad del sistema interno a los LDO. Podemos apreciar de la tabla de LDOs, que el TPS73733 requiere de un capacitor menor al del LT1963, lo cual es conveniente, ya que a mayores valores de capacitor, se encarece el componente, porque es necesario utilizar otra tecnología de capacitores que no sean cerámicos, como pueden ser los de tantalio o electrolíticos.

Finalmente analizando el precio de ambos integrados, se estudia que todos los parámetros de interés benefician al TPS73733 frente al LT1963, lo cual no justifica el elevado precio de este último. La razón por la que el LT1963 es menos económico que los otros integrados, se atribuye principalmente a las siguientes características que no se presentan en el resto de los integrados comparados, y son apuntados a proteger la alimentación de entrada del integrado: bajo ruido (40uV<sub>rm</sub> de 10Hz a 100kHz), optimizado para respuesta rápida en el estado transitorio, protección de batería en inversa y corriente en inversa nulo. Estos parámetros son de menor interés en nuestra aplicación, debido a que la entrada del regulador LDO no es directamente la batería, sino que es la salida del bloque de control de batería, la cual ya tiene protecciones incluidas además de estar limitado por un circuito externo mediante un diodo doble BAR43C. Aun sumando el valor de dicho diodo (0,25U\$) al costo del TPS73733, se estudió que para nuestro caso es preferible la implementación del TPS73733 como regulador LDO de 3,3V, ya que maximiza por cierto margen la eficiencia a un costo mucho menor.

Módulo de administración de alimentación para el sistema:

Esta sección es en nuestra aplicación una de las principales áreas de mejora, debido a que presenta la electrónica de potencia empleada en el Hardware y es la fuente de alimentación principal del sistema. Centrándonos en que el menú electrónico debe ser manejable e intuitivo como cualquier dispositivo portátil en la actualidad, se buscaron los modos más fuertemente difundidos en el mercado. Los resultados indican, que la gran mayoría de dichos aparatos portátiles se alimentaban mediante un transformador que baja de 220Vac del tomacorriente a tensión continua para sostener la carga necesaria por el dispositivo y su batería. En nuestro caso, se identificó una necesidad de autonomía de batería de aproximadamente 2 horas en pleno uso, sin requerir de alimentación externa. Es por esta razón que se eligió utilizar baterías en el rango de 3,6V a 3,7V a partir de 1000mAh. La comparación entre los distintos tipos de batería y la potencia mínima necesaria para el sistema se justificará posteriormente con mayor detalle en el punto VI, inciso de "detalles de construcción y precauciones de montaje".

Entonces, el integrado elegido deberá ser capaz de proporcionar alimentación tanto a la batería como al sistema. Dentro de todos los integrados examinados, que satisfacen dichas características, se comparan a continuación las principales diferencias para seleccionar el mejor candidato para nuestra aplicación en particular:

TABLA XII  
COMPARACIÓN DE DISTINTOS INTEGRADOS DE REGULADORES DE CARGA

Cargador	Tamaño	Precio U\$S	DPPM	Protecciones
BQ24070	3,5mm*4,5mm QFN-20PIN	3,46	Si	Si, varios
LTC3677-3	4mm*7mm QFN-44PIN	9,8	Si	Si, varios
MCP73831	2mm*2mm DFN-8PIN	0,68	No	Si, básicos

En cuanto a mínimo tamaño, la mejor opción es el MCP73831 de la empresa Microchip, que a su vez tiene el menor precio dentro de todos los integrados para la carga de la batería. No obstante, dicho integrado fue descartado, debido a que no brinda una buena administración de corriente para nuestro sistema, ya que no presenta DPPM interno (Dynamic Power-Path Management).

El concepto de DPPM es básicamente tener caminos de alimentación independientes para el circuito y para la batería, logrando de este modo no sobrecargar la impedancia de la batería, ni que sufra distintos ciclos de carga, debido a cambios de corrientes que se puedan presentar en el circuito.

Por otro lado, el DPPM permite uso del sistema conectado directamente a la alimentación en caso de ausencia o deficiencia en la batería, garantizando al usuario uso continuo del dispositivo. Esto se logra mediante un circuito de MOS, el cual es ofrecido por los integrados BQ24070 de la compañía Texas Instrument y LTC3677-3 de Linear Technology. El LTC3677-3 es un componente mucho más caro comparado con el resto, porque está pensado para un rango mayor de aplicaciones, ofreciendo reguladores de distintos valores de corrientes (800mA hasta 500mA). En nuestro caso, se determinó que la carga estará limitada a 500mA para la batería, entonces no se precisa de dicha característica. Debido a que el LTC3677-3 presenta casi el doble de tamaño con 24 pines más que el BQ24070, que no son necesarios en nuestra aplicación, queda justificado la elección del integrado BQ24070 para la administración de batería.

Una vez resuelto la carga de la batería, se procede a estudiar la manera en la que se informará al usuario el estado actual de carga. Analizando las hojas de aplicación del BQ24070, se obtiene con ayuda de un integrado adicional los parámetros de estado y temperatura de la batería, además de una interrupción para advertir que se tiene baja batería o ausencia del mismo. El componente encargado de avisar al microprocesador de dichos estados es el BQ27500 y fue desarrollado de la mano con el BQ24070, funcionando mediante un convertidor de tensión brindado por el TPS71525. La interconexión entre los 3 integrados encargados del uso y funcionamiento correcto de la batería se detalla en la siguiente figura. Inspeccionando el circuito, se puede ver, que cuando se alimenta al BQ24070 mediante 5V en el pin IN, se tendrá en los pines OUT una tensión fija de 4,4V, mientras que en ausencia de alimentación, se tendrá una tensión de entrada suministrada por la batería. Esto quiere decir, que el integrado tiene 2 fuentes de alimentación posibles y podrá utilizar cualquiera o ambos, para proveer la tensión de 4,4V a la salida.

Para la verificación rápida visualmente del estado de batería se incluyen leds, los cuales representan los siguientes estados:

- Led de estado STAT1: indica si la batería se está cargando
- Led de estado 3.3V: muestra la correcta alimentación del módulo procesador de aplicaciones
- Led de estado BAT\_LOW: avisa estado de batería baja

Para el caso del último led de estado, se utiliza el transistor MMBT3904 para amplificar la corriente al nivel necesario a la salida de la terminal del integrado BQ25700.

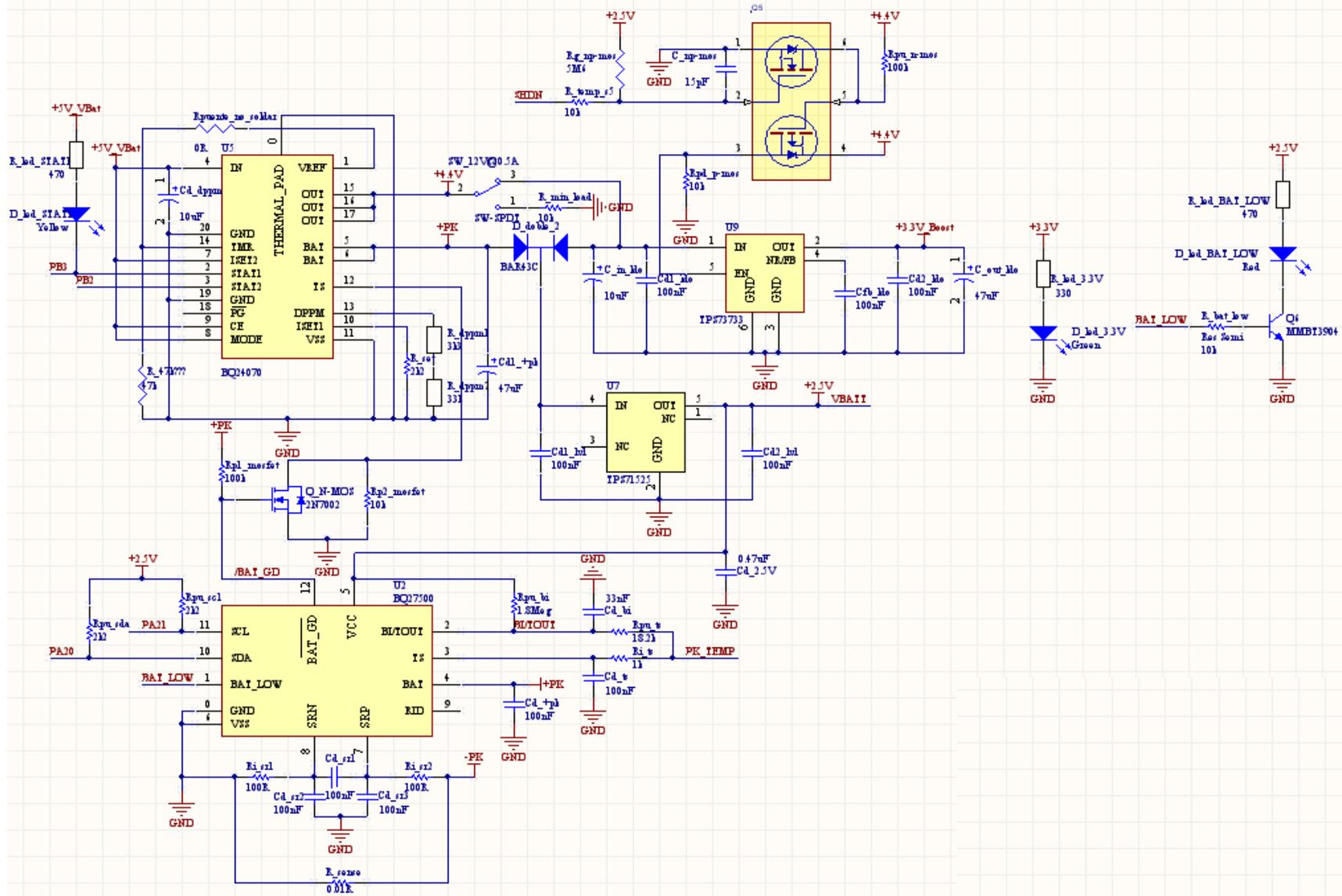


Fig. 7. Esquemático del circuito utilizado para la alimentación del sistema.

Módulo de procesador de aplicaciones:

La siguiente etapa crítica a estudiar es el módulo del microprocesador y es de gran importancia, principalmente debido al alto costo de los mismos en comparación con el resto del diseño. Microprocesadores que satisfagan las especificaciones para aplicaciones del tipo de dispositivos portátiles, tienen en general la complejidad de requerir diversos tipos de memoria y conexiones de periféricos, los cuales dependen fuertemente de la aplicación y sobretodo de la velocidad de procesamiento. Esto conlleva a que la decisión del microprocesador sea sometido a criterios muy subjetivos. En nuestro caso, se enuncian las principales características deseadas para obtener un módulo microprocesador suficientemente versátil para adaptarse a mejoras potenciales en futuros desarrollos:

- Frecuencia de operación del clock suficientemente rápida para soportar principalmente la interfaz gráfica del dispositivo.
- Memoria RAM tipo DDR o superior para acceder a los datos de manera rápida y tener una intercomunicación fluida mediante una interfaz amigable con el usuario.
- Modo de comunicación SPI, I2C o SCI para tener compatibilidad con periféricos. En nuestro caso, dichos dispositivos son puertos de entrada y salida, como también es el controlador de la pantalla o el módulo Zigbee.

De aquí en adelante denominaremos módulo microprocesador al conjunto microprocesador, memoria RAM y Flash, el cual también se lo designará como procesador de aplicaciones. Debido a que no se tiene experiencia sólida previa con microprocesadores de esta gama, se procedió a optar por un microprocesador con gran escalabilidad con respecto a nuestra aplicación. Además, luego de analizar las hojas de datos de los procesadores de aplicaciones, se examinó la dificultad técnica de poder llevar a cabo en nuestro laboratorio de electrónica, la realización de dicha placa. El principal inconveniente se atribuye al hecho de requerir un PCB de aproximadamente 4 capas, debido al complicado ruteo que representa la interconexión del microprocesador con su memorias. De este modo, se procedió de acuerdo con la cátedra, a utilizar un módulo microprocesador disponible en el mercado.

Existen distintos tipos de arquitectura de microprocesadores comerciales que satisfacen con el objetivo del proyecto. Dentro de las arquitecturas, se observó que la gran mayoría de los dispositivos portátiles de tamaño Tablet vienen internamente con procesadores ARM. La ventaja de los procesadores ARM, es la relativa simplicidad que presenta su diseño y los vuelve un gran aliado para aplicaciones de baja potencia. Cabe aclarar, que aplicaciones de baja potencia es en nuestro caso menor a los 5Watts a pleno consumo (mucho menor que las lámparas incandescentes de uso diario de 50 a 60 Watts), dicho cálculo de potencia se explicará posteriormente con mayor detalle. Estos bajos consumos dieron como resultado convertir a los microprocesadores ARM en los dominantes en el mercado de la electrónica móvil e integrada, como es el caso de nuestra aplicación.

Posteriormente se explicará en el inciso de software, la necesidad, de que el microprocesador pueda soportar un sistema operativo, en nuestro caso en particular se implementará el código en un sistema operativo Unix-like como Linux. Es por esta razón, que la característica más sobresaliente que nos ofrece los procesadores ARM, es que tienen implementada internamente una tecnología que permite a ciertos tipos de arquitectura ejecutar Java Bytecode nativamente en el Hardware. Esto quiere decir, que se agiliza los tiempos que requieren las instrucciones del microprocesador, ya que no se pierde eficiencia en interpretación del código por el nivel lógico.

Nuestro principal candidato dentro de los ARM es el ATMEL ATSAM9G45 de la familia ARM9E, debido a que es uno de los módulos microprocesadores disponibles en la facultad para su uso y programación, donada por la empresa APEXAR. El microprocesador elegido es una solución basada en ARM, el cual es más específicamente una arquitectura RISC de 32bits. La empresa APEXAR fue fundada por exalumnos del ITBA y se tuvo contacto con los mismos dueños para realizar consultas sobre las diferentes complejidades que se presentaba en nuestro diseño. Dicho soporte y disponibilidad del contacto garantizaba mayor éxito en nuestro diseño que utilizar otros módulos microprocesadores, los cuales también tenían documentación y hojas de aplicaciones disponibles, además se consiguieron contactos de

referencia con expertise en el ámbito para ofrecernos ayuda ante complicaciones. Dichas razones enunciadas previamente nos inclinaron a elegir al procesador de aplicaciones SOM9G45, denominado por APEXAR a la placa que incluye el microprocesador ATSAM9G45 ruteado con memorias.

A continuación se enumeran las principales especificaciones técnicas del SOM9G45 y mejoras frente a modelos precedentes del mismo microprocesador, para dar una noción de las velocidades de procesamiento dentro del cual nos estaremos manejando y una idea de escalabilidad del proyecto:

- Procesador ARM926EJ-STM ARM de 400MHz
- 128-MBytes de memoria RAM DDR2
- 256-MBytes de memoria NAND Flash
- Pipeline de 5 estados, agilizando procesos secuenciales.
- Mayor velocidad de cargado y almacenado (load y store), logrando que más instrucciones se realicen solamente en un ciclo. Esto se logró modificando la arquitectura interna al tipo de arquitectura Harvard.

#### Módulo de Display:

A la hora de elegir la pantalla a utilizar, se tenían condiciones impuestas por el costo, tamaño deseado, calidad de imagen, alimentación y consumo.

Antes de considerar dichos criterios, se decidió trabajar con tecnología LCD TFT<sup>7</sup>, la más usada en dispositivos portátiles y pantallas táctiles no industriales hoy en día, principalmente por su bajo consumo y calidad de imagen.

Considerando el tamaño de la pantalla, un menú impreso convencional suele poseer dimensiones semejantes a las de una revista, soliendo ser rectangulares con una relación ancho/alto similar a las de una hoja estandarizada como las utilizadas en el presente documento. Esto se debe a que suelen constar de varias páginas, figurando en cada una de ellas una o más listas de los productos ofrecidos, cada lista correspondiente a una categoría de los mismos (por ejemplo, bebidas o postres).

Por el contrario, en el caso del menú táctil a implementar, las listas de productos serán más dinámicas, permitiendo visualizar a pantalla completa un único producto, o recorrer una lista de productos de la misma categoría a través de una o más "páginas" (secciones de cada lista, mostradas, de a una por vez, en la pantalla).

Así, recurrir a dimensiones (en cuanto a la pantalla) similares a las de un menú impreso convencional aumentaría el costo, desaprovechando el dinamismo brindado por una aplicación de software capaz de adaptarse a los deseos del usuario. Por esta razón, se decidió elegir, de las opciones disponibles en el mercado, un tamaño de pantalla lo suficientemente grande como para permitir una interfaz cómoda y clara, pero al mismo tiempo no tan grande como para desaprovechar el poder de procesamiento y exposición de datos de un software como el que se va a implementar.

Observando los valores **disponibles** en el mercado, se observó que el tamaño suele especificarse en pulgadas de diagonal, como es común en las pantallas de computadora o televisores convencionales. De dichos valores, se extrajo una lista correspondiente al rango en el que se trabajará (se descartaron las más pequeñas).

A continuación, se decidió entre las opciones "standard" y "widescreen", las que fijan, para un mismo tamaño de la diagonal, una relación ancho/alto de la pantalla<sup>8</sup>. Considerando que el menú será portátil, es conveniente que, así como se ha estandarizado en dispositivos inalámbricos como teléfonos celulares y consolas de videojuegos portátiles, la relación ancho/alto o alto/ancho sea relativamente mayor a la de los

---

<sup>7</sup> [http://es.wikipedia.org/wiki/TFT\\_LCD](http://es.wikipedia.org/wiki/TFT_LCD)

<sup>8</sup> La opción standard, utilizada en monitores más antiguos, tiene una relación ancho/alto cercana o aproximadamente igual a 1.3 en todas sus pantallas. La opción widescreen, más utilizada en monitores modernos, tiene, en cambio, una relación ancho/alto cercana a 1.6.

monitores "standard". Conjuntamente con eso, el software a desarrollar contará con opciones que vuelven a la opción "widescreen", una mejor alternativa que la "standard".

Luego, se realizó un tercer filtrado a partir del criterio de los desarrolladores del producto, principalmente el criterio de quien desarrollaría el software. Con dicho criterio, se confeccionó la siguiente tabla, que muestra si los valores surgidos del primer y segundo filtrados son adecuados o no para la presente aplicación.

TABLA XIII  
COMPARACIÓN DE TAMAÑOS DE PANTALLA

Tamaño pantalla	Adecuado
3.5" Diagonal	Muy pequeño
4.3" Diagonal	Muy pequeño
5" Diagonal	Adecuado, tamaño mínimo
5.7" Diagonal	Adecuado, ligeramente pequeño
7" Diagonal	Adecuado

Desde este punto de vista, se concluye que las pantallas widescreen de 5" a 7" están en el rango aceptable para ser utilizadas en esta implementación.

En cuanto a la resolución de la imagen a mostrar, se observa que en los modelos disponibles suele ir de la mano con el tamaño de pantalla elegido (a mayor tamaño, mayor resolución). Esto ocurre pues quien vea la imagen, no percibe la discontinuidad entre los píxeles de la misma a partir de cierta relación píxeles/pulgada<sup>2</sup>.

Para encontrar una relación satisfactoria, se recurrió al criterio sugerido por Microsoft en su ayuda online para los usuarios<sup>9</sup>. En el mismo, muestra una tabla para distintos valores de la diagonal de la pantalla, junto a la resolución recomendada para cada uno de ellos. Dicha tabla se muestra a continuación (sólo se muestra la sección correspondiente a los widescreen, que es la de interés).

Monitor size	Recommended resolution (in pixels)
13- to 15-inch widescreen laptop screen	1280 × 800
17-inch widescreen laptop screen	1680 × 1050
24-inch widescreen LCD monitor	1920 × 1200

Fig. 8. Criterio Microsoft de resolución en función de tamaño de display.

Por medio de estos datos, y observando su tendencia logarítmica, se realizó una regresión de dicho tipo, concluyendo que, para el tamaño de 7" (el peor caso, pues si esta pantalla es satisfecha por una resolución, también será satisfecha cualquier pantalla de menor tamaño) un ancho mínimo de 567 píxeles sería necesario.

Sabiendo que la pantalla a elegir será widescreen, se sabe que:

$$h = a / 1.6$$

Donde  $a$  es el ancho (o lado más largo) y  $h$  el alto (o lado más corto) de la pantalla.

A partir de dicha relación, se puede obtener que el alto mínimo recomendado será de 355 píxeles.

<sup>9</sup> <http://windows.microsoft.com/en-US/windows7/Getting-the-best-display-on-your-monitor>

La resolución mínima, entonces, a cumplir por la pantalla será de 567 píxeles de ancho por 355 píxeles de alto. Cabe aclarar que la SOM a utilizar también imponía una cota superior para esta resolución, estando capacitada para trabajar hasta 1280x860 píxeles, valor que excede en gran medida los requerimientos mínimos hallados.

Hay que considerar, también, la cantidad de colores disponibles para cada píxel (valor que no suele relacionarse tanto con el tamaño deseado sino con la aplicación para la pantalla). Los desarrolladores consideraron que una cantidad de colores adecuada podía tomarse de las utilizadas en los teléfonos portátiles actuales.

En cuanto a alimentación, de los modelos disponibles se extrajo la conclusión de que no difieren mucho ni en consumo ni en tensión de alimentación (para una misma tensión de alimentación, las pantallas de mayor tamaño consumen apenas un poco más de corriente). Por esto, con las regulaciones de tensión adecuadas, este criterio no es decisivo entre las pantallas disponibles.

Considerando el software a desarrollar, otro criterio que se utilizó es darle preferencia a las pantallas que posean un controlador incorporado, a fin de facilitar el manejo de la misma y reducir la carga del microprocesador a utilizar por el software (siempre y cuando este criterio no se oponga en demasía a los demás).

Desde el punto de vista económico, primordial en nuestro proyecto, se observa en las páginas de los distintos fabricantes que el precio de las pantallas aumenta conforme aumenta el tamaño, pero en una relación logarítmica, lo cual indica que la variación de precio entre una pantalla de 5" y una de 7" de diagonal no será muy grande. Esto se observa directamente al comparar dichos precios en la página de cada fabricante.

De acuerdo a todos estos criterios, el modelo elegido fue NHD-7.0-800480WF-CTXI#-T<sup>10</sup>, de Newhaven Display International, considerando que presentaba las siguientes ventajas frente a otros modelos:

- Disponible en el mercado
- Pantalla de 7" a un precio medio frente a otras compañías
- Consumo normal para este tipo de dispositivos
- Puede comprarse junto a la membrana táctil correspondiente
- Incorpora un controlador para la pantalla
- Resolución de 800x480 píxeles
- 262144 colores

La cantidad de colores disponibles para esta pantalla cumple con los requisitos de ser usados en dispositivos ya comercializados exitosamente, al ser un valor presente en varios dispositivos actuales de telefonía celular, y brindar una calidad de imagen útil para esta aplicación, como se observa en la siguiente figura.

---

<sup>10</sup> <http://www.newhavendisplay.com/specs/NHD-7.0-800480WF-CTXI-T.pdf>



Fig. 9. Imagen con 262144 colores<sup>11</sup>.

En la siguiente imagen, se observan las dimensiones de la pantalla elegida.

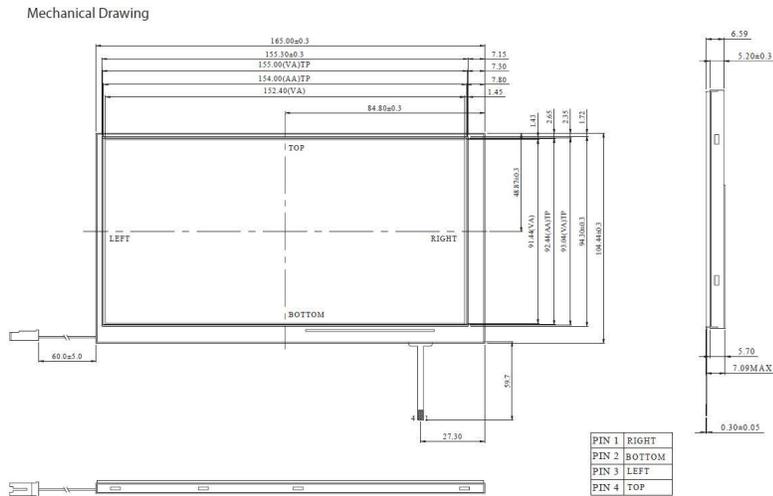


Fig. 10. Dimensiones del display TFT a utilizar, según los esquemáticos de su datasheet.

Módulo de comunicación Xbee:

El módulo Xbee fue diseñado con la mínima cantidad de componentes posibles, de manera que sea compacta y a su vez satisfacer las necesidades básicas que conciernen a nuestra aplicación. El integrado seleccionado es el Xbee o Xbee-Pro, los cuales son compatibles entre sí. El Xbee-Pro logra comunicaciones de hasta 90m de distancia en zonas urbanas, justificando la selección del módulo, además de que se tiene experiencia previa con el integrado. A continuación, se presenta el circuito empleado para el módulo de comunicación inalámbrica:

<sup>11</sup> Obtenida de [http://www.hantronix.com/files/image/10\\_4\\_svg\\_a\\_tft.jpg](http://www.hantronix.com/files/image/10_4_svg_a_tft.jpg)

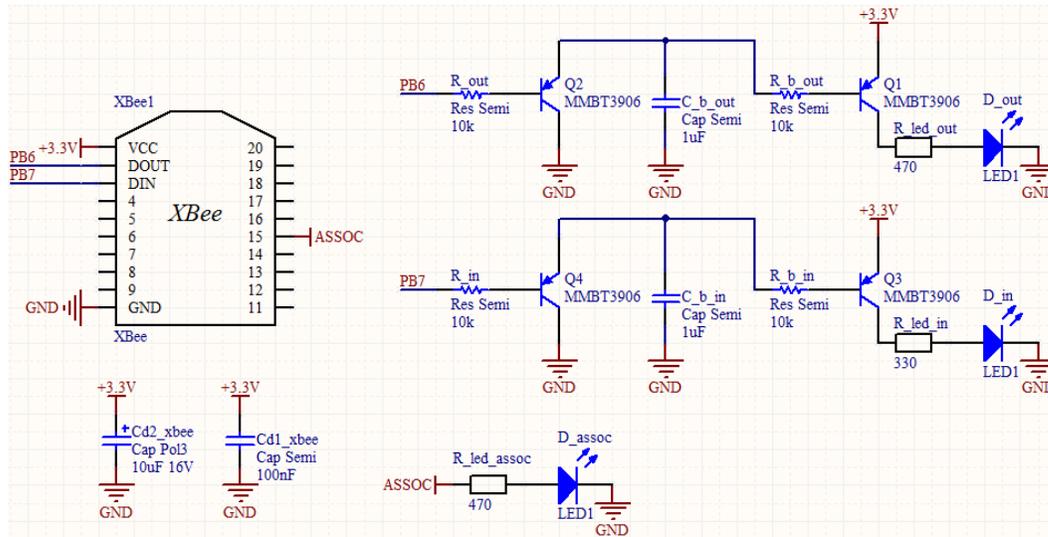


Fig. 11. Esquemático del Xbee con los leds indicadores de estado.

La función que cumplen los leds de estado, es permitir al usuario comprobar el correcto comportamiento del sistema mediante simple inspección visual. Los estados de los leds presentan el estado del módulo:

- Led de estado ASSOC: indica el estado de listo para su uso, en espera o enviado/recepción de datos
- Led de estado DIN: avisa la recepción de datos
- Led de estado DOUT: manifiesta el envío de datos

Los transistores MMBT3906 para amplificar la corriente necesaria para encender los leds son los sugeridos por la hoja de datos y los valores de resistencias y capacitores empleados tienen la función de pasabajar la frecuencia de parpadeo del led cercana a los 16Hz para que sea perceptible al ojo humano.

### c. Plan de pruebas de cada módulo

El plan de prueba comienza con la comprobación de los módulos de fuentes de alimentación del sistema y para su comodidad se colocaron varios pines de prueba para el acceso rápido de la medición. Una vez verificado que cada integrado esté correctamente alimentado, se procede a verificar el funcionamiento del microprocesador, ya que es el encargado del resto de los integrados.

Con ayuda de una placa de prueba que se encuentra conectado a una computadora y con acceso a la línea de comando del sistema operativo de Linux embebido dentro del módulo microprocesador, se puede inspeccionar mediante el Log característico de Linux la correcta inicialización del firmware.

Luego, se debe examinar el funcionamiento del protocolo de comunicación del Xbee. La manera más rápida de verificar el funcionamiento del módulo de comunicación es utilizando un receptor Zigbee en una computadora fija, el cual interpretará los datos recibidos como datos de entrada en el puerto serie y con el programa X-CTU instalado, podrá decodificar el código binario.

Finalmente, se verifica la conexión I2C para el funcionamiento de la pantalla y la lectura del estado de batería. Para probar la membrana a utilizar, se confeccionará un programa de prueba en el microprocesador HC12, con el cual los desarrolladores tienen experiencia. Utilizándose los conversores AD disponibles en la placa del mismo para probar el correcto funcionamiento de la pantalla.

Para probar la pantalla a utilizar, se elabora el siguiente plan de pruebas:

- Verificar la correcta alimentación
- Verificar que la alimentación elegida para la pantalla sea suficiente para que el backlight permita ver la imagen bajo condiciones de iluminación fuerte (por ejemplo, iluminada por el sol de mediodía).
- Testear la conexión de cada bit de canal (6 por color)
- Testear la conexión de cada bit restante (datos, etc.).
- Probar la pantalla con un programa de prueba usando un procesador con el que se esté familiarizado, como el HC12. dicho programa deberá testear por separado:
  - Colores
  - Píxeles
  - Velocidad de refresco correcta

Para probar el conjunto Pantalla-Membrana, se realizará el siguiente plan de pruebas

- Colocar ambos dispositivos bajo el control de un mismo microprocesador y comprobar su funcionamiento simultáneo
- Mediante un programa de prueba, comprobar la equivalencia de las coordenadas detectadas por la membrana y las mostradas en pantalla

En lo que respecta al convertor AD del ATSAM9G45, la placa que incluye el microprocesador a utilizar (el conjunto del microprocesador ATSAM9G45) se vale de un convertor analógico digital para recibir datos analógicos.

Para probar esta instancia del módulo SOM, se elabora el siguiente plan de pruebas:

- Tensión de alimentación correcta
- Convertir valores analógicos de tensión conocidos a digital y verificar el correcto funcionamiento del dispositivo
- Verificar que los pines especializados en membrana táctil reciban los datos y decodifiquen correctamente dicha entrada
- Probar cada uno de los demás pines que figuran en la hoja de datos del fabricante, verificando el correcto funcionamiento en cada uno
- Testear la velocidad del convertor, con señales muestreadas de modo tal que el ADC se vea obligado a convertir con el tiempo mínimo tolerable, especificado en la hoja de datos.
- Probar el convertor con el software implementado y, habiendo debuggeado el código en caso de fallas de software, testear con vectores de prueba para prevenir incompatibilidades por cálculo numérico.

## *2. Software*

El desarrollo del software está basado en la plataforma Linux, el cual es un kernel similar a Unix, y se utiliza con herramientas de sistema GNU. La utilización de este tipo de sistema se debe principalmente a dos causas. En primer lugar, al ser un software libre, su licencia es totalmente gratuita y evita el costo, ya sea de adquirir una licencia para otro sistema a utilizar, o desarrollar uno propio. Y en segundo lugar tiene la ventaja que la arquitectura ARM del microcontrolador empleado es soportada por este tipo de sistemas operativos.

Para la programación del software, dentro de la plataforma Linux, se utiliza un toolkit llamado GTK+, que brinda un completo set de widgets para la creación de un entorno gráfico para el usuario. Si bien está diseñado para soportar varios lenguajes de programación, en el nuestro se utilizará su versión de C/C++ por dos causas. Primero, que el mismo está escrito en C, por lo tanto tiene mayor confiabilidad utilizar esta versión. Y segundo, frente a otros lenguajes de programación, C es con el que estamos más familiarizados, por lo tanto necesitamos menor tiempo de aprendizaje para su implementación.

La elección por este software de programación se debió a varias razones. Como razón principal se encuentra la necesidad de utilizar un software que facilite la realización de un entorno gráfico agradable y fácil de usar, por una cuestión de tiempos en su implementación. También hay que mencionar que es un software gratuito y parte del proyecto de GNU, por lo tanto no implica ningún gasto en comprar su licencia. Por las razones anteriores es muy utilizado, lo cual da una vasta información de su implementación, la cual puede encontrarse en internet. Además de esto, se tienen contactos que tienen conocimiento y experiencia a la hora de programar con este toolkit que presentan gran ayuda ante cualquier inconveniente.

Además del software empleado en el producto en sí, es necesario implementar otro programa para correr el servidor con el cual interactúa nuestro dispositivo como se menciona posteriormente. La principal función de este servidor es la de tener una base de datos con todas las comidas, los mozos y la de mantener la información en la cual se encuentra cada mesa.

Se decidió implementar el servidor en Ruby on Rails (RoR), el cual es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby. La elección por este framework se debe a que posibilita la simplicidad de programación al ser necesario escribir menos código que con otros frameworks y con un mínimo de configuración. Además, al tratarse de programación web, solo se requiere el uso de un navegador para correr el programa con lo cual cuenta con las ventajas de ser accesible desde cualquier computadora con acceso a internet y corre bajo cualquier plataforma. Ante esto último hay que tener en cuenta que se requiere que se cargue el programa a un servidor de internet, aunque se puede usar una máquina local para su uso interno. La última razón a tener en cuenta para la elección de RoR es que se trata de un framework de código abierto, por lo tanto cuenta con ventajas similares, en cuanto a temas legales, como las explicadas para en el uso del toolkit GTK. Esto quiere decir, para el presente caso, que no podría comercializarse versiones propias del lenguaje Ruby o el framework Rails, pero que sí podría venderse el software desarrollado a partir del mismo.

A la hora de realizar el diseño de la página web que correría en el servidor, se priorizó que los diversos campos de la base de datos de todo el sistema fueran fácilmente accesibles para el usuario, permitiendo un manejo intuitivo y sencillo de la información del local. Por esto, se desarrolló un modelo de página que permitiera, a través de una barra de navegación, tener cada sección importante dentro de la base de datos permanentemente accesible. Además, a fin de permitir al encargado de la caja llevar una contabilidad y administración más sencillas y automáticas, se implementó un acceso a los campos de datos "in place", a fin de evitar tener que recargar la página cada vez que se modifique un dato. Finalmente, una barra de notificaciones a la derecha informa, en tiempo real, acerca de eventos importantes pertinentes al manejo de la base de datos del local.



Fig. 12. Pantalla principal del servidor web de un hipotético cliente, el Restaurante "El Gaucho".

En la figura anterior pueden observarse los elementos mencionados anteriormente. A la izquierda, la barra de navegación, a la derecha, la barra de notificaciones, y, en el centro, la información de cada página. Las imágenes, colores y encabezado a usar pueden personalizarse para cada cliente.

#### a. Diagrama de estados y flujogramas

El programa que correrá en el dispositivo portátil cuenta con varias etapas independientes entre sí. Esto se debe a que, mientras la funcionalidad principal del programa está ejecutándose, las demás etapas también pueden hacerlo. Entre las etapas secundarias se encuentra la publicidad, mostrar fecha y hora, llamar al mozo, y mostrar la carga de la batería. Esto se puntualiza, ya que si bien se mencionan y describen a continuación, estas etapas, no aparecen en el diagrama de estados principal del programa al ser cada uno de las mencionadas etapas independientes entre sí.

##### ETAPA PUBLICIDAD:

En esta etapa se muestra en el fondo del programa tanto imágenes como animaciones de las marcas que realizan su publicidad en el dispositivo.

Estas imágenes o animaciones se encuentran cargadas en la memoria RAM para dar libertad al agregar, quitar o cambiar alguna de ellas. En principio debería poder modificarse en cualquier momento, y debería haber un estado dentro de esta etapa que sea la recepción de datos, pero como no es una tarea que se realiza periódicamente no se tiene en cuenta dentro del diagrama de flujo.

El tiempo en que van rotando las publicidades entre una marca a otra depende de un timer que realiza la interrupción, cuyo valor en tiempo puede ser definido por el dueño del local, pero que se fija por default en un minuto.

##### ETAPA MOSTRAR FECHA Y HORA:

En esta etapa se muestra la fecha con día, mes y año, y la hora con precisión de segundos. Debido a la última condición, la actualización debe realizarse una vez por segundo, lo cual se realiza con un timer propio fijado para este motivo.

El valor de la fecha y hora a mostrar es indicado por el mismo microcontrolador, que gracias al sistema operativo Linux, son fáciles de obtener para el programa. Los mismos pueden modificarse, pero debido a que esta tarea solo se realiza una vez, no se tiene en cuenta como parte de la etapa.

##### ETAPA LLAMAR MOZO:

Esta etapa es independiente del programa principal, ya que la misma puede realizarse en cualquier momento, sin necesidad de haber realizado el pedido total. Esto está hecho de esta manera para el caso de algún inconveniente en el cual se requiera el servicio del mozo.

Para realizar la llamada al mozo, se cuenta con un botón en la aplicación que manda el mensaje directamente, por lo tanto la etapa en sí consiste solamente en el envío de datos, cuya información es, como lo dice su nombre, la de llamar al mozo a la mesa.

**ETAPA BATERÍA:**

En esta etapa se indica, mediante un ícono, la carga de la batería. Uno de los integrados de la placa tiene la posibilidad de medir la carga y por lo tanto, por medio del sistema operativo, se puede mostrar en pantalla dicho nivel.

La funcionalidad de esta etapa es mostrar al dueño del local cuando necesita recargar los dispositivos, o cuanto tiempo de duración les queda. También está a la vista del cliente debido a que estos mismos pueden informarle al encargado cuando el nivel de la batería es bajo en caso de que este último no se percate de esto.

A continuación se muestra el diagrama de estados del programa principal:

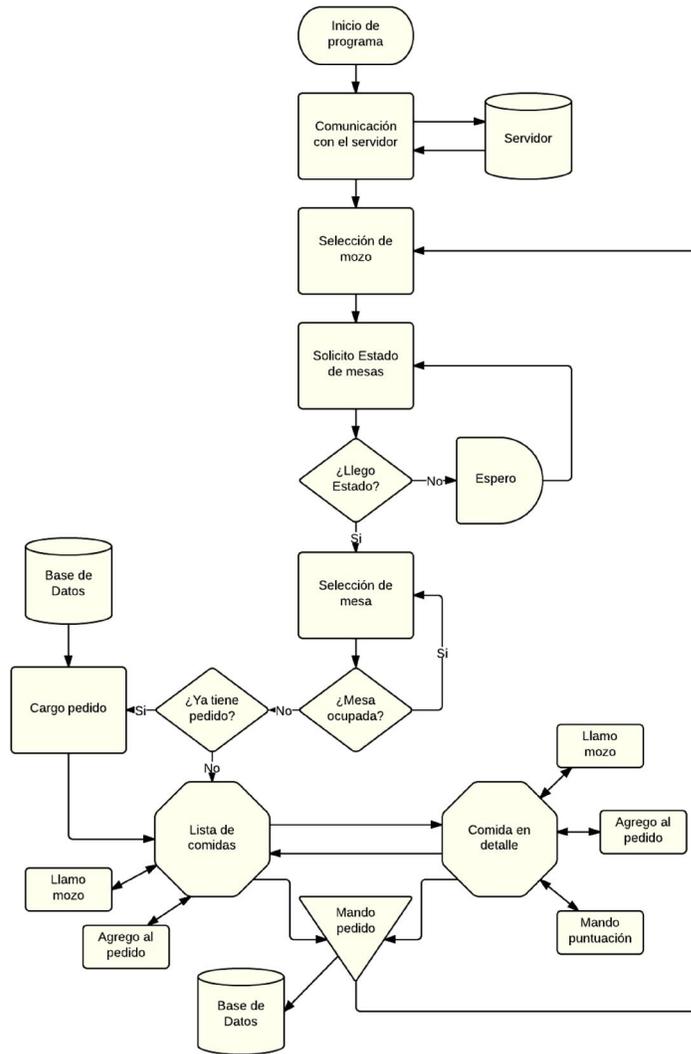


Fig. 13. Diagrama de estados del programa principal.

Al iniciarse el programa, primero se realiza una comunicación con el servidor principal para obtener la información de los mozos presentes (nombre, foto y contraseña), y si agregó o modificó alguna comida con respecto al estado de la base de datos en el producto. Todo esto se realiza mediante la comunicación inalámbrica por medio del X-Bee en base al protocolo de comunicación TCP.

Una vez finalizada la interacción con el servidor, se muestra en pantalla una lista con los distintos mozos a ser seleccionados. Esta etapa es manejada por el mozo, ya que es él mismo el que debe identificarse mediante el ingreso de su contraseña.

Con el mozo ya elegido, se pasa a la siguiente etapa donde se debe seleccionar la mesa a la cual se va a utilizar el dispositivo. Cada mesa se encuentra en uno de tres estados, los cuales se explican más adelante. Al terminar con esta etapa, es que se le presenta el producto al cliente en la mesa en si para la selección de comidas.

El sistema principal presentado al comensal se encuentra inicialmente en alguno de los dos modos de operación, descriptos posteriormente. Por default se inicia en el modo donde se muestran las listas de comidas, debido a que presenta una vista más general de los productos a vender.

Estando en alguno de los modos, se presentan todos los productos pertenecientes a un tipo de receta, es decir a entrada, plato principal, postre, etc. El usuario puede cambiar a otro tipo de receta, para lo cual se accede a la base de datos, y esta devuelve todos los platos de la receta seleccionada (modo lista de comidas), o al primero de estos (modo detalle de comida).

Una vez que el usuario selecciona todos los platos que desea consumir, debe pasar a la pestaña de pedido total en el cual se encuentran todos estos indicando el precio total. De aquí se tiene la opción de confirmar el pedido realizado, para lo cual se realiza el envío de datos correspondiente, o volver al modo en el que se encontraba para continuar con la elección de los platos.

Una vez que se envió el pedido, se retira el dispositivo y se reinicia el proceso.

#### ETAPA COMUNICACIÓN CON EL SERVIDOR:

Al iniciar el programa, lo primero que debe hacerse es una interacción con el servidor para obtener información de la base de datos.

Se necesitan inicialmente dos tipos de datos; el primero de ellos es si hubo una actualización en las comidas que se encuentran disponibles. Para esto se envía el último día de actualización de la base de comidas dentro del producto, con lo cual el servidor responde con las nuevas comidas modificadas o creadas a partir de este punto. Cada plato está dentro de una categoría de receta, como es entrada, plato principal o postre. La información perteneciente a cada uno es: el nombre, la foto correspondiente, la descripción, la puntuación, las opciones y el precio.

El segundo dato necesario es los mozos que se encuentran. Como la cantidad de mozos generalmente es pequeña y puede variar que personas se presentan cada día, no se tiene una base de datos de los mismos dentro del producto. En cambio lo que se realiza es que se envíen los datos cada vez que se inicia la aplicación. La única información del mozo que si se guarda dentro del dispositivo son las imágenes de cada uno ya que su envío hace lenta la comunicación. Cabe aclarar que de entrada no se tienen dichas imágenes guardadas, por lo que en primera instancia son enviadas al dispositivo para su almacenamiento dentro del mismo.

#### ETAPA SELECCIÓN DE MOZO:

Con los datos de los mozos obtenidos gracias a la comunicación con el servidor, se muestra una lista con los nombres y la foto de cada mozo. Aquí es el mozo el cual debe seleccionar su perfil en el dispositivo para identificarse. Para ello, luego de ser seleccionado, se pide su contraseña, la cual se verifica con la enviada con sus datos. La contraseña tiene que ser un número definido previamente por él mismo, el cual se ingresa por medio de un pequeño panel numérico que se muestra luego de seleccionar algún mozo en la lista.

#### ETAPA SELECCIÓN DE MESA:

Luego de que el mozo se identifique, se solicita al servidor que mande la información sobre cada una de las mesas. El número de mesas se fija de antemano en el programa, por lo tanto para cada una de estas mesas se indica en qué estado se encuentra. Cada mesa puede tener uno de los tres estados siguientes: ocupada, con pedido cargado o libre.

El estado de ocupado se muestra con color rojo e indica que la mesa se encuentra ya con un dispositivo para elegir su pedido y no puede seleccionarse.

El estado libre se muestra de color verde e indica que se puede seleccionar esa mesa. Unavez seleccionada dicha mesa se informa al servidor que la misma pasa a encontrarse ocupada.

Por último, el estado con pedido cargado, mostrado de color amarillo, señala que la mesa ya ha realizado un pedido pero no se ha cerrado el mismo, por lo tanto se encuentra en condiciones de seguir ordenando. Para esto se le pide al servidor que envíe el pedido cargado a dicha mesa, el cual luego se muestra los modos principales.

#### ETAPA MODO LISTA DE COMIDAS:

En este modo, se presenta todos los platos del tipo de receta seleccionada en una tabla que muestra las características principales de cada plato, como nombre, foto pequeña y precio. El usuario tiene la opción de seleccionar varios platos indicando la cantidad de cada uno.

Este modo se utiliza para una selección rápida del pedido sin ninguna especificación en particular para alguno de los platos.

Se puede pasar de este modo al otro mediante la elección de uno de los platos presentes en el mismo, con lo cual el otro modo inicia con este plato seleccionado.

#### ETAPA MODO DETALLE DE COMIDA:

En este modo se presenta la información de un plato a la vez, dentro de la receta seleccionada. Puede rotarse entre los platos presentes en cada receta de a uno por vez. Su principal función es mostrar información más detallada del producto, además de contar con la posibilidad de un pedido más específico para cada plato, como por ejemplo elegir el nivel de cocido en una carne.

Este modo se utiliza para una selección más personalizada de cada plato si no se desea recibir el pedido por default del modo 1. De todas maneras puede realizarse un pedido entre ambos modos al fin de personalizar solo aquellos platos que se desee.

Se puede pasar al otro modo mediante un botón presente que vuelve al otro modo mostrando los platos de la receta en la cual se encontraba el modo 2.

#### ETAPA PEDIDO TOTAL:

En esta etapa se muestran todos los platos seleccionados en alguno de los otros modos, clasificados por la receta a la que pertenecen. Estos platos tienen las especificaciones indicadas si fueron seleccionados mediante el modo 2. Al final se muestra el precio total a pagar por todo el pedido.

Contiene un botón de confirmación, el cual pregunta si el pedido esta correcto. En caso de que la respuesta es positiva, se pasa al envío de datos con el pedido total. Si la respuesta es negativa, se vuelve al modo del cual se estaba anteriormente.

#### ETAPA ENVIO DE DATOS:

En esta etapa se realiza el envío de datos del pedido realizado hacia el servidor, en el cual se indica cual de los dispositivos realizo el envío para indicar la mesa a la cual corresponde el pedido.

El envío se realiza por medio del módulo de Zigbee, mediante el protocolo de comunicación TCP entre el dispositivo y el server.

En cuanto al programa que servirá de front-end del servidor y que correrá en un explorador web, puede decirse que el mismo está dividido en modelos, elementos nativos de Ruby on Rails. Dichos modelos representan los distintos componentes de la base de datos y contienen, a su vez, los campos correspondientes a cada variable a modificar. Por ejemplo, un modelo "Mozo" contendrá los campos Nombre, Apellido, Salario, etc.

Mediante relaciones establecidas a través del lenguaje Ruby, estos modelos pueden interaccionar entre sí, así como ser accesibles o no para el usuario.

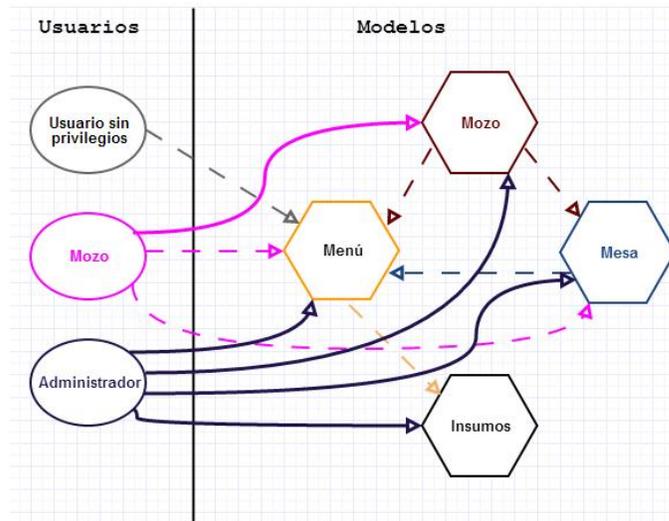


Fig. 14. Modelos de la base de datos del sistema y permisos para cada tipo de usuario que puede acceder a ellos. Las líneas punteadas representan que pueden leerse datos del modelo, mientras que las líneas continuas reflejan la capacidad de modificar los datos accedidos.

Como puede observarse en la figura, los modelos pueden ser visibles, invisibles o editables para otro modelo o un usuario con permisos particulares. Esto último puede permitir la opción de fusionar la administración del sitio con la página web del local, en el caso de que el servidor corra remotamente en vez de en una red local. De este modo, un usuario sin cuenta personal sólo podrá ver el menú disponible. Un mozo podrá consultar el estado de las mesas y acceder al menú, pero también podrá consultar sus datos personales y modificarlos según corresponda (exceptuando el salario, el cual sólo es modificable por el administrador). El administrador, finalmente, tiene control total sobre los campos de la base de datos, pudiendo realizar modificaciones en cualquier modelo.

#### b. Análisis de complejidad

Desde el punto de vista del algorítmico, el tiempo de procesamiento de software no depende de ninguna variable directamente de forma apreciable. Todos los eventos que se pueden llegar a producir se deben a dos causas. Los eventos más importantes son los producidos por el usuario con la interacción del dispositivo. Y la segunda causa de estos son las interrupciones temporales producidas por los timers incluidos en el software.

En el primer caso mencionado, las acciones realizadas por el usuario implican seleccionar alguna opción dadas por el software. Como generalmente solo se selecciona de a una opción por vez, esto se corresponde con ejecutar una cantidad relativamente fija de instrucciones, variando con la opción elegida. De todas maneras esta cantidad no depende de ninguna variable en forma directa, por lo tanto su complejidad algorítmica es de  $O(1)$ .

En el segundo caso, las acciones realizadas por las interrupciones, las cuales corresponden a los diferentes timers presentes en el programa, tampoco varían con respecto al valor de alguna variable al ser sus tiempos fijos. Esto conlleva a que su complejidad algorítmica es nuevamente  $O(1)$ .

Habría que analizar cuáles pueden ser las posibles causas para que se produzca un cambio apreciable en el tiempo de procesamiento. Si bien hay varias variables que entran en juego, las principales causas se deben al tiempo de interrupciones, y el tiempo entre eventos dados por el usuario.

El tiempo de interrupciones, si bien pueden modificarse los tiempos de los timers, una vez que se fija se mantiene constante durante la utilización del dispositivo. De todas maneras los tiempos de procesamiento para cada interrupción son mucho menores a los tiempos dados por los timers, con lo cual se puede considerar despreciable su variación.

Si tenemos en cuenta que el mínimo tiempo entre los eventos que produce una persona al seleccionar una opción está por encima de los 150 ms, podemos despreciar la influencia de esta variación debido a que los tiempos de procesamiento de dichos eventos son muy inferiores a este.

*c. Descripción de subrutinas:*

La interacción entre los distintos módulos ya se comentó previamente, por lo tanto aquí se pasa a detallar las funciones correspondientes a los distintos drivers del programa y una breve descripción de estas. Todas estas funciones están escritas en lenguaje C ya que, como se mencionó anteriormente, es el lenguaje utilizado para programar y empleado por el toolkit de GTK+. Por esta razón es que varias funciones devuelven tipos de variables definidos dentro del toolkit mencionado.

Los drivers tanto para la pantalla como para los distintos elementos utilizados en la placa ya están incorporados en el microcontrolador, ya sea el controlador de la pantalla o el de la placa principal. Por lo tanto no se requiere la implementación de los mismos al trabajar a través de Linux, y es este el encargado de manipularlos siendo transparente a nuestro programa.

**Driver Barra superior:**

*Función:* GtkWidget \*do\_toolbar (void);

*Descripción:* Devuelve el widget que representa la barra donde se encuentran los botones de las funciones como mostrar fecha y hora, y llamar al mozo.

*Función:* gchar \* return\_hour (void);

*Descripción:* Devuelve la hora en un string dado por hora-minuto-segundos

*Función:* gchar \* return\_day (void);

*Descripción:* Devuelve el día dado por el microcontrolador en un string con el format DD/MM/YYYY.

*Función:* gboolean time\_toolbar (GtkWidget \*widget);

*Descripción:* Timer presente en el toolbar que se utiliza para actualizar el reloj. Su valor es ajustable mediante una constante a definir, que inicialmente esta definida en 1 segundo.

*Función:* void date\_redraw (GtkWidget \*widget);

*Descripción:* Cada vez que se actualiza el reloj es necesario volver a graficar la hora correcta en la barra, para lo cual se llama a esta función.

*Función:* void boton\_mozo\_clicked (GtkButton \*button, gpointer user\_data);

*Descripción:* Es la función que se llama al evento cuando se presiona el botón de llamar al mozo y realiza el envío de datos correspondientes al servidor.

*Función:* void mozo\_mesa\_toolbar (GtkWidget \*widget, gboolean cual);

*Descripción:* Muestra en la barra superior el mozo seleccionado con su nombre y foto, y al lado la mesa seleccionada.

*Función:* gboolean nuestro\_carga\_bateria (GtkWidget \*image );

*Descripción:* Muestra por medio de una imagen la carga presente en la batería del circuito.

*Función:* GtkWidget \* mozo\_toolbar (void);

*Descripción:* Devuelve el widget del mozo para mostrar, es decir carga el nombre y su foto.

**Driver Pestañas:**

*Función:* void add\_book( char \*Menu,char \*Frame, GtkNotebook \*notebook, gint lugar);

*Descripción:* Esta función crea una pestaña correspondiente a la receta pasada como parámetro, es decir ya sea entrada, plato principal, o postre. Solo crea la pestaña, es decir la información que presenta cada una se realiza en las funciones del driver de lista.

*Función:* GtkWidget \* do\_button(void);

*Descripción:* Función que crea el botón de confirmar pedido en el borde derecho inferior.

*Función:*void Confirmando\_pedido (void);

Descripción: Función que se llama cuando se confirma el pedido y lo manda a través del puerto serie.

Función: guint nro\_plato (gchar \*plato);

Descripción: Devuelve el número de plato en función de su nombre (entrada = 0, principal = 1, postre = 2, etc).

Función: guint nro\_comida (gchar \*comida, guint iplato);

Descripción: Devuelve el número de comida en función del nombre de la misma y del tipo de plato al que pertenezca. Esto último se debe a que puede existir una misma comida para diferentes tipos.

Función: guint nro\_opcion (gchar \*opcion, guint iplato, guint icomida);

Descripción: Devuelve el número de la opción en base al nombre de la misma con respecto a una determinada comida.

Función: void paso\_sel\_mozo (void);

Descripción: Función que pasa al modo de selección de mozo una vez finalizada la comunicación inicial con el servidor.

### **Driver Publicidad:**

Función: void load\_banner (GtkWidget \*frame);

Descripción: Crea el lugar donde se va a mostrar la publicidad cargando en principio la primer publicidad de la lista. Además inicia el timer correspondiente para la actualización del mismo.

Función: static gboolean time\_handler (GtkWidget \*frame);

Descripción: Crea el timer que se encarga de modificar la publicidad por la siguiente marca a mostrar. Su valor puede ser modificable, teniendo como default un tiempo de 1 minuto.

Función: static gboolean redraw\_frame (GtkWidget \*widget);

Descripción: Realiza la actualización de la imagen o animación a mostrar en la publicidad cargando la siguiente que corresponda. Esta acción se realiza cada vez que se ejecuta el timer dado para la publicidad.

### **Driver Lista:**

static GtkWidget \*treeview\_g;

Descripción: Variable global donde se guarda la lista donde se presenta el pedido total. Esta variable es global ya que tiene que ser accedida por distintas funciones, sin posibilidad de ser pasada como parámetro.

Función: GtkTreeModel \*create\_List\_model (gint Plato);

Descripción: Función que genera la lista correspondiente a cada receta dada, tomando los datos de la base de datos correspondiente.

Función: void fixed\_toggled (GtkCellRendererToggle \*cell, gchar \*path\_str, gpointer data);

Descripción: Cada vez que se selecciona o deselecciona un plato, el evento activa esta función que agrega o quita el plato seleccionado de la lista de pedido total.

Función: void add\_columns (GtkTreeView \*treeview);

Descripción: Para crear la lista con los datos correspondientes, hace falta agregar a esta las columnas con el dato que cada una va a mostrar. Por lo tanto se puede tener mayor información en cada plato del que se muestra en la lista del pedido total.

Función: GtkWidget \* do\_list\_store (gint Plato);

Descripción: Función que crea la lista dada para el modo 1 en el cual se muestran todos los platos dados para la receta pasada como parámetro. Esta lista se adjunta en su pestaña correspondiente.

Función: static GtkTreeModel \* create\_Tree\_model (void);

Descripción: Para crear una lista, hace falta crear un modelo de la misma en el cual se indican el tipo de variable que va a almacenar las distintas columnas.

Función: void add\_branch ( GtkTreeModel \*model, GtkTreeIter iter);  
Descripción: Función que agrega a la lista de pedido total el plato seleccionado.  
Función: void remove\_branch (GtkTreeModel \*model, GtkTreeIter iter);  
Descripción: Función que remueve de la lista de pedido total el plato deseleccionado.  
Función: static gboolean lista\_clicked (GtkWidget \*treeview, GdkEventButton \*event, gpointer data);  
Descripción: Función que se llama cuando se clickea en la lista y procesa a que columna pertenece.  
Función: void boton\_cant\_clicked (GtkWidget \*widget, gpointer data);  
Descripción: Agrega el numero en la cantidad agregada en el dialogo de cantidad.  
Función: void borrar\_cant\_clicked (GtkWidget \*widget, gpointer data);  
Descripción: Borra el último numero ingresado en el dialogo de cantidad.  
Función: void confirma\_cant (gint Plato, GtkTreePath \*path);  
Descripción: Se confirma la cantidad ingresada y se la modifica en la lista de comidas.  
Función: void cantidad\_clicked (gint Plato, GtkTreePath \*path);  
Descripción: Función que inicia el dialogo cuando la columna seleccionada es cantidad.  
Función: static gboolean paso\_modos2 (GtkTreePath \*path);  
Descripción: Función que pasa al modo de detalle de comida.  
Función: static gboolean pedido\_clicked (GtkWidget \*treeview, GdkEventButton \*event, gpointer data);  
Descripción: En la pestaña de pedido, si se selecciona el botón de borrar, se saca dicha comida.  
Función: void show\_all\_text (gint nro\_plato, GtkTreePath \*path, gboolean cual);  
Descripción: Abre una ventana de dialogo con el texto completo de la columna elegida, siempre y cuando esta muestre texto, por si acaso el texto se muestra cortado en el modo de lista.

### **Driver Detalle de Comida:**

Función: gboolean redraw\_image (FilasComida \*Plato\_actual);  
Descripción: Vuelve a dibujar la imagen de la comida con la indicada.  
Función: static gboolean adelante (GtkWidget \*button, GdkEventButton \*event, gpointer data);  
Descripción: Muestra la siguiente comida en la lista.  
Función: static gboolean atras (GtkWidget \*button, GdkEventButton \*event, gpointer data);  
Descripción: Muestra la comida anterior de la lista.  
Función: void cambio\_descripcion (FilasComida \*Plato\_actual);  
Descripción: Cambia el texto de la descripción de la comida.  
Función: void cambio\_precio (FilasComida \*Plato\_actual);  
Descripción: Cambia el texto de precio de la comida.  
Función: void cambio\_puntuacion (FilasComida \*Plato\_actual);  
Descripción: Cambia la imagen que muestra la puntuación de la comida.  
Función: void cambio\_opciones (FilasComida \*Plato\_actual);  
Descripción: Cambia las opciones mostradas, y oculta las opciones no presentes.  
Función: void puntuacion\_clicked (void);  
Descripción: Dialogo que se muestra al apretar en la puntuación para mandar el valor elegido.  
Función: static gboolean paso\_modos1 (GtkWidget \*button, GdkEventButton \*event, gpointer data);  
Descripción: Función que pasa al modo lista de presentación.  
Función: gchar \*evaluo\_puntuacion (guint puntuado);  
Descripción: Devuelve la imagen en estrellas de acuerdo a la puntuación.  
Función: static gboolean agregarlo (GtkWidget \*button, GdkEventButton \*event, gpointer data);

Descripción: Agrega la comida seleccionada al pedido.

Función: Options devuelvo\_opcion (gchar \*opciones);

Descripción: Carga en la estructura las diferentes opciones dadas por un string.

Función: gint size\_str (gchar \*str);

Descripción: Devuelve el tamaño del string.

Función: gchar \*corro\_uno (gchar \*str);

Descripción: Elimina el primer carácter del string.

Función: gchar \*devuelvo\_opcion\_sel ( void);

Descripción: Devuelve el string de la opcion seleccionada dentro de los botones de elección.

Función: void score\_radio\_clicked (GtkButton \*button, gpointer data);

Descripción: Modifica la puntuación seleccionada.

### **Driver Selección de mozo y mesa:**

Función: void add\_to\_list(GtkWidget \*list, const gchar \*str, const gchar \*picture);

Descripción: Añade un mozo con su nombre y foto a la lista.

Función: void init\_list(GtkWidget \*list);

Descripción: Inicializa la lista de los mozos.

Función: void elijo\_mozo(GtkWidget \*widget, gpointer selection);

Descripción: Función para pasar a la confirmación de la contraseña una vez seleccionado el mozo.

Función: GtkWidget \* seleccion\_mozo (void);

Descripción: Función que muestra la lista de mozos y el botón de selección en la pantalla.

Función: void password (gchar \*mozo);

Descripción: Muestra el dialogo para ingresar la contraseña.

Función: void boton\_pass\_clicked (GtkWidget \*widget, gpointer data);

Descripción: Función que ingresa el numero en el dialogo para la contraseña.

Función: void borrar\_pass\_clicked (GtkWidget \*widget, gpointer data);

Descripción: Función que borra el numero en el dialogo para la contraseña.

Función: gboolean verifico\_pass (gchar \*mozo);

Descripción: Función que verifica la contraseña ingresada.

Función: GtkWidget \* dialogo (gchar \*titulo, gchar \*info, gchar \*stock\_id);

Descripción: Dialogo de información con un determinado título, información e icono.

Función: GtkWidget \* seleccion\_mesa(void);

Descripción: Muestra las diferentes mesas con el color que representa su estado.

Función: void boton\_table\_clicked (GtkWidget \*widget, gpointer data);

Descripción: Cuando se selecciona una mesa verifica que se encuentre disponible.

Función: void paso\_sel\_mesas(void);

Descripción: Función para pasar a la selección de mesas una vez verificada la contraseña.

Función: void paso\_menu\_principal(void);

Descripción: Función para pasar al modo de lista una vez seleccionada la mesa.

Función: void init\_modos(void);

Descripción: Inicializo los diferentes modos a mostrar.

Función: void agregar\_mozo( gchar \*nombre, gchar \*foto, gint pass);

Descripción: Agrega el mozo enviado a la lista para ser mostrado.

### **Driver de Inicialización:**

*Función:* GtkWidget \*SetupWindow(gchar \*titulo, gchar \*fondo, gboolean Transient);

*Descripción:* Función para configurar la pantalla principal del programa.

*Función:* GtkWidget \*Cambiar\_fondo(GtkWidget \*window, gchar \*fondo);

*Descripción:* Función que cambia el fondo de la pantalla principal.

*Función:* gboolean delete( GtkWidget \*widget, GtkWidget \*event, gpointer data );

*Descripción:* Función que se activa cuando se desea cerrar el programa y libera la memoria reservada antes de hacerlo.

*Función:* gboolean delete\_q( GtkWidget \*widget, GdkEventKey \*event, gpointer data );

*Descripción:* Igual a la función “delete” pero se activa al presionar la tecla ‘q’.

*Función:* GtkWidget \*menu\_principal (void);

*Descripción:* Función que crea la barra superior, el modo de lista y la publicidad.

*Función:* void clean\_all\_list (void);

*Descripción:* Elimina todas las comidas seleccionadas del pedido.

*Función:* void cargar\_al\_pedido (guint iplato, guint icomida, guint quantity, guint option);

*Descripción:* Carga al pedido la comida con su opción y cantidad dadas.

*Función:* void free\_save\_memory (void);

*Descripción:* Libera la memoria reservada de los arreglos para almacenar las comidas.

*Función:* void load\_wait\_gif(void);

*Descripción:* Muestra el gif de cargando en la pantalla principal.

### **Driver Base de Datos:**

*Función:* gboolean return\_available (FilasComida \*Plato, guint Comida);

*Descripción:* Verifica si la comida está disponible.

*Función:* gboolean return\_pick (FilasComida \*Plato, guint Comida);

*Descripción:* Verifica si la comida está seleccionada.

*Función:* gchar \* return\_image (FilasComida \*Plato, guint Comida);

*Descripción:* Devuelve el nombre de la imagen de la comida indicada.

*Función:* gchar \* return\_food (FilasComida \*Plato, guint Comida);

*Descripción:* Devuelve el nombre de la comida indicada.

*Función:* gchar \* return\_description (FilasComida \*Plato, guint Comida);

*Descripción:* Devuelve la descripción de la comida inidcada.

*Función:* guint return\_price (FilasComida \*Plato, guint Comida);

*Descripción:* Devuelve el precio de la comida indicada.

*Función:* gchar \* return\_options (FilasComida \*Plato, guint Comida);

*Descripción:* Devuelve las opciones separadas por un guion de la comida indicada.

*Función:* guint return\_score (FilasComida \*Plato, guint Comida);

*Descripción:* Devuelve la puntuación de la comida inidcada.

*Función:* gchar \* return\_plate (FilasComida \*Plato, guint Comida);

*Descripción:* Devuelve a que plato pertenece la comida inidcada.

*Función:* FilasComida \* return\_children (FilasComida \*Plato, guint Comida);

*Descripción:* Devuelve la estructura con los datos de la comida, dado el plato y número de comida indicados.

Función: void agregar\_comida (gchar \*image, gchar \*food, gchar \*description, guint price, gchar \*options, guint score, gchar \*plate, gboolean available);

Descripción: Agrega una comida con todos sus valores a la lista.

Función:void init\_entrada (void);

Descripción: Inicializa las comidas para cada tipo de plato.

Función: void actualizar\_BD (Comidabin enviado);

Descripción: Actualiza la base de datos de las comidas dentro del producto.

Función: void borrar\_BD (void);

Descripción: Vacía la base de datos de las comidas.

Función: void actualizar\_fecha\_BD(gchar \*fecha);

Descripción: Actualiza la última fecha de modificación de la base de datos.

Función:gchar \* copy\_str (gchar \*source);

Descripción: Copia el string de un puntero a otro.

### **Driver Comunicación Serie:**

Función:void init\_connection(gint puerto, gint baudrate);

Descripción: Establece los parámetros de comunicación del puerto serie e inicializa su comunicación.

Función:void send\_data (gchar \*str, gboolean mando\_ack);

Descripción: Envía un string a través del puerto serie, y tiene la opción de enviar o no un acknowledge. La comunicación siempre espera la respuesta de un acknowledge por parte del servidor.

Función:gint str\_size (gchar \*str);

Descripción: Devuelve el tamaño del string asignado.

Función:void init\_recive (void);

Descripción: Comunicación inicial donde se reciben los mozos y se actualiza la base de datos de comida.

Función:void mesas\_recive (void);

Descripción: Se comunica con el servidor a la espera del estado de las mesas.

Función:void pedido\_recive (void);

Descripción: Cuando una mesa tiene un pedido, se comunica con el servidor para obtener el mismo.

Función:void mozos\_recive (void);

Descripción: Espera recibir todos los mozos con su nombre, foto y contraseña. Si la imagen no se encuentra ya en el producto, la misma es enviada.

Función:food\_number get\_number\_from\_list (guint number);

Descripción: Devuelve el tipo de plato y el número de comida, en una estructura, dentro de este plato con respecto al id de la comida.

Función:guint get\_list\_number (guint iplato, guint icomida);

Descripción: Devuelve el id de la comida con respecto al número de comida dentro de un tipo de plato.

Función:void request\_tables (void);

Descripción: Solicita el envío del estado de las mesas.

Función:void request\_order (guint mesa);

Descripción: Solicita el pedido ya cargado en la mesa indicada.

Función:void request\_waiters (void);

Descripción: Solicita los mozos disponibles.

Función:void call\_waiter (guint waiter, guint table);

Descripción: Llama al mozo encargado de la mesa.

Función: void send\_order (guint iplato, guint icomida, guint iopcion, guint cantidad, guint mesa);

Descripción: Manda una de las comidas cargadas al pedido.

Función: void send\_vote (guint iplato, guint icomida, guint votacion);

Descripción: Manda el voto realizado a una determinada comida.

Función: void send\_open\_table (guint mesa, guint mozo);

Descripción: Una vez seleccionada una mesa libre, se indica al servidor que habrá dicha mesa con el mozo correspondiente.

Función: void send\_db\_date (void);

Descripción: Manda la fecha de la última actualización de la base de datos.

Función: void send\_table\_pick (guint mesa);

Descripción: Se manda cuando se selecciona una mesa que ya tiene un pedido cargado.

Función: void send\_already (gboolean esta);

Descripción: Para que se corresponda el servidor con el producto, cuando una mesa se encuentra ocupada se manda un paquete indicándolo.

Función: void send\_ack (void);

Descripción: Manda el mensaje de acknowledge al servidor.

### **Driver Batería:**

Función: int bq\_write(char reg, char data);

Descripción: Escribe cierta información en el registro seleccionado.

Función: int bq\_read(char reg, int bytes, char \*i2c\_buf);

Descripción: Lee del registro seleccionado una cierta cantidad de bytes y lo almacena en un buffer.

Función: void i2c\_init (void);

Descripción: Inicia la comunicación por medio de i2c con el integrado encargado de los parámetros de la batería.

Función: int temperature (void);

Descripción: Devuelve la temperatura de la batería.

Función: int voltage (void);

Descripción: Devuelve el voltaje de la batería.

Función: int time2empty (void);

Descripción: Devuelve el tiempo para que se agote la batería con la carga actual.

Función: int time2full (void);

Descripción: Devuelve el tiempo necesario para cargar la batería.

Función: int stateofcharge(void);

Descripción: Devuelve el porcentaje de carga presente en la batería.

#### *d. Listados comentados del código*

El código completo y comentado se presenta en los anexos.

#### *e. Plan de prueba de módulos y de depuración de software*

Para realizar la prueba de módulos del software se puede dividir a la misma en dos categorías. La primera incluye los drivers de los elementos a utilizar y la segunda incluye al programa realizado. Dentro de la primera clasificación se encuentran los drivers para los distintos dispositivos, los cuales incluyen la pantalla, la membrana táctil y la comunicación inalámbrica.

A continuación se detallan los planes de prueba a realizar para cada uno de estos elementos:

- Depuración del driver de la pantalla: Se escribe un programa de prueba que muestre en pantalla una imagen dividida en zonas que cada cierto tiempo modifique el color de dichas zonas. Con esto lo que se logra es verificar tanto el correcto refresco de la pantalla como la funcionalidad de todos sus colores.
- Depuración del driver de la membrana táctil: Haciendo uso del driver del display, que se considera que funciona correctamente, se realiza un programa de prueba que obtenga la posición donde se presionó la misma e imprima en pantalla dicha posición. De esta forma se puede detectar que todas las zonas son detectadas correctamente y que no se pierda ningún evento.
- Depuración del driver de la comunicación inalámbrica: Se realiza un programa de prueba que envíe al servidor listas de pedidos finales, variando la longitud de las mismas y su contenido. También se realiza otro programa en el cual el tamaño de datos a enviar sea constante, pero se varía el tiempo en el cual se lo manda. Ambos programas se corren más de una vez para asegurarse de su correcto funcionamiento. Con esto se verifica el tiempo mínimo de respuesta, y detecta errores en el caso de que algún dato en particular o todos estos no se envíen correctamente.
- Una vez que todos los drivers funcionan correctamente, se procede a depurar la otra categoría que representa al programa realizado. Con esto se desea verificar el correcto funcionamiento del mismo sobre el microprocesador utilizado, y que se detecten todos los eventos realizados y se realice su posterior acción a estos. Esto se hace principalmente, ya que el programa realizado se escribe primero en una computadora y luego se pasa al microprocesador.

Si se realizan todos los pasos mencionados anteriormente se puede verificar que el dispositivo no presenta fallas en condiciones normales de uso, y por lo tanto se da por finalizado la fase de depurado del software, el cual funciona correctamente.

## IX. CONSTRUCCIÓN DEL PROTOTIPO

### 1. DEFINICIÓN DE LOS MÓDULOS

Para la construcción del prototipo se dividió al sistema en tres módulos físicos distintos, debido a las diferentes funcionalidades de los mismos. De esta manera, el proceso de validación es más simple, al tener aislados los módulos y conocer las posibles fallas que cada uno pueda ocasionar. Luego de comprobar el correcto funcionamiento del total de los módulos, se verifica la interacción entre los mismos.

Los respectivos módulos son:

Pantalla y membrana táctil: La pantalla se comunica con el controlador LCD interno de la motherboard. El controlador consiste de lógica para transferir datos de imágenes para módulos TFT LCD de un buffer externo, programable desde el microprocesador. En nuestro caso, la pantalla consta de un panel TFT LCD con interface digital de 18 bits RGB. Dentro de las características embebidas, se nos permite utilizar 16 o 24 bits por pixel para pantallas TFT. Por lo tanto, se configura a la motherboard con 24 bits por pixel (6 bits de sobra), para aprovechar los 18 bits de resolución de la pantalla. Ya que controlando 16 bits, se obtiene solamente 65536 colores, mientras que con 18 bits, se tiene 262144 colores. A cambio de habilitar dicha opción, se tiene que procesar 8 bits más, dentro de los cuales 6 bits no se verán reflejados en la resolución real.

Placa principal o motherboard: Compuesta por los conversores de tensión, el microprocesador, el cargador de batería y la comunicación inalámbrica. La motherboard es el más importante de todos los módulos, debido a que se encuentra conectada con los demás módulos indicando la función a realizar de cada uno. Los conversores de tensión cumplen el rol de suministrar eficientemente alimentación de 3,3V y 5V para la motherboard. Finalmente el módulo de comunicación Xbee es el encargado de la comunicación entre el microprocesador y la terminal. En nuestro caso, la terminal es una computadora de escritorio con su propio módulo Xbee, la cual contiene toda la base de información del restaurante.

Batería y administradores de batería: Si bien el módulo de la placa principal puede funcionar conectado mediante un transformador a 220V de la red, para obtener autonomía y mayor movilidad es necesario contar con una fuente de alimentación independiente. Para lograr este objetivo se ensambla una batería de litio-ion de 3,7V de 2000mAh. El administrador de batería tiene la función de gestionar el estado de la batería propiamente dicha e informar al microprocesador su condición de carga.

### 2. DISEÑO DE LOS CIRCUITOS IMPRESOS

A partir de los esquemáticos presentados en la etapa de ingeniería de detalles, se procedió a realizar el diseño del PCB, teniendo en cuenta las siguientes características a la hora del posicionamiento de los integrados y el ruteo del mismo:

- Debido a que la pantalla es de 7" por especificación, se tiene una base mínima impuesta por la pantalla de 165mm de ancho y de 104,44mm de alto. Para que los módulos sean de tamaños equivalentes y poder superponerse en el ensamblaje del prototipo, se determinó un tamaño comercial de PCB para la motherboard de 175mm x 100mm. La motherboard posee 10mm más de ancho, donde se posicionan leds de información de estado para que sean visibles de frente junto a la pantalla.
- El PCB de la placa principal es doble capa y al no requerir capas adicionales, economiza costos y presenta mayor simplicidad a la hora de producción. Los principales integrados son de montaje superficial y se encuentran todos en la misma capa del PCB, de esta forma permite una reparación más rápida. Por otro lado, se usa la capa despoblada de integrados para enfrenar el módulo de la pantalla con el fin de proteger la motherboard de la temperatura de la pantalla.
- La ubicación de los integrados del módulo principal fueron ubicados según indicaciones de las hojas de datos de los módulos de prueba proporcionados por el fabricante del microprocesador

ATMEL, con el fin de lograr un layout simple de entender y con longitudes de pistas cortas. A su vez, se ubicaron los capacitores de desacople de los integrados lo más cerca posible de la entrada de alimentación de los mismos. De esta manera, se garantiza que los integrados obtienen el correcto nivel de alimentación de entrada y poder funcionar eficientemente. Los planos de masa y para la disipación de potencia fueron tomados en cuenta según indicativos de los fabricantes, para el uso apropiado de los componentes.

Tomando en cuenta los detalles previamente descritos del PCB y el dimensionamiento del mismo, se muestra el ruteo finalizado del PCB:

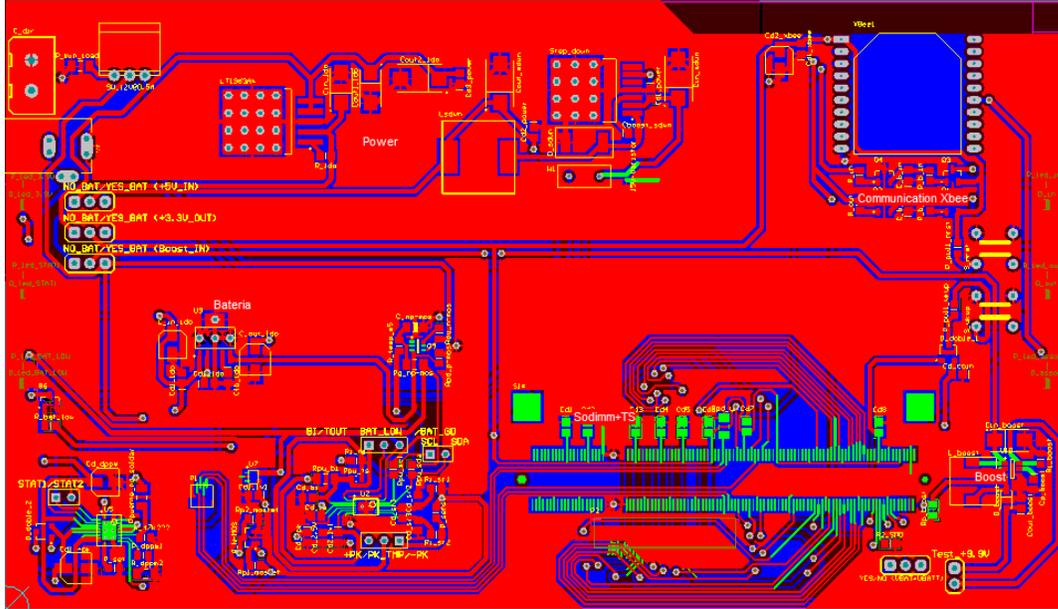


Fig. 15. PCB de la capa superior de la motherboard.

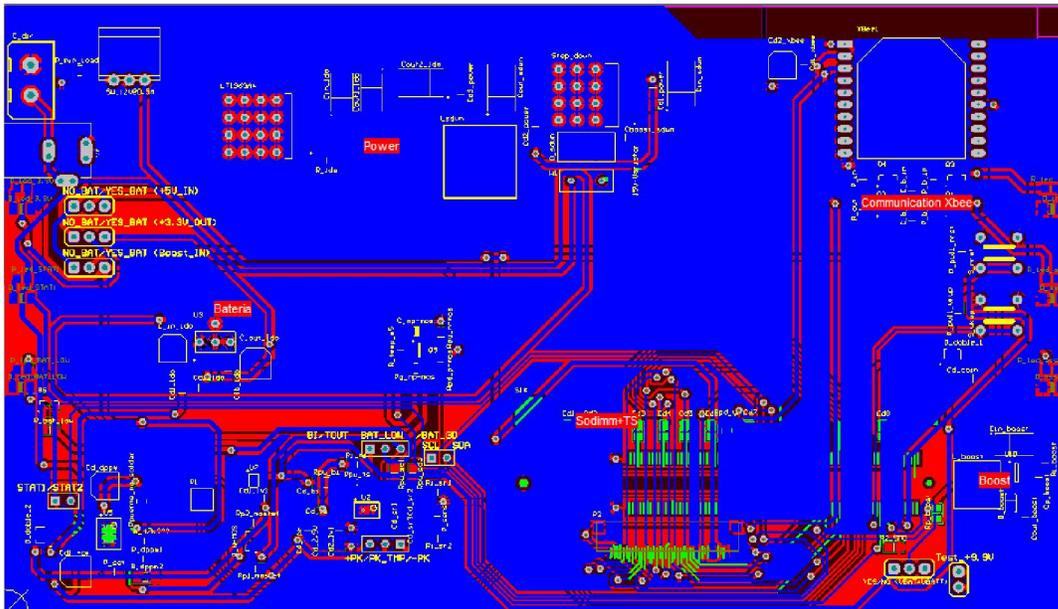


Fig. 16. PCB de la capa inferior de la motherboard.

Cabe destacar que el espacio disponible que se encuentra en el medio de la placa está reservado para tener margen con el encastre del módulo SOM9G45, incluido con el microprocesador ATMEL ATSAM9G45. El módulo SOM9G45 se encastra (previamente medido y comprobado) en el conector

SODIMM 200 (marcado como footprint en verde) y se bifurcan sus conexiones en los puertos de entrada y salida, fuente de alimentación y conexión Xbee.

En la práctica se utilizó, previo al soldado de los componentes, una capa antisoldante de color rojo, la cual ayuda a que el estaño no se disperse a lo largo del cobre en el momento del soldado. Este proceso agiliza la manufactura del producto, además de proveer una capa protectora contra la oxidación del cobre presente.

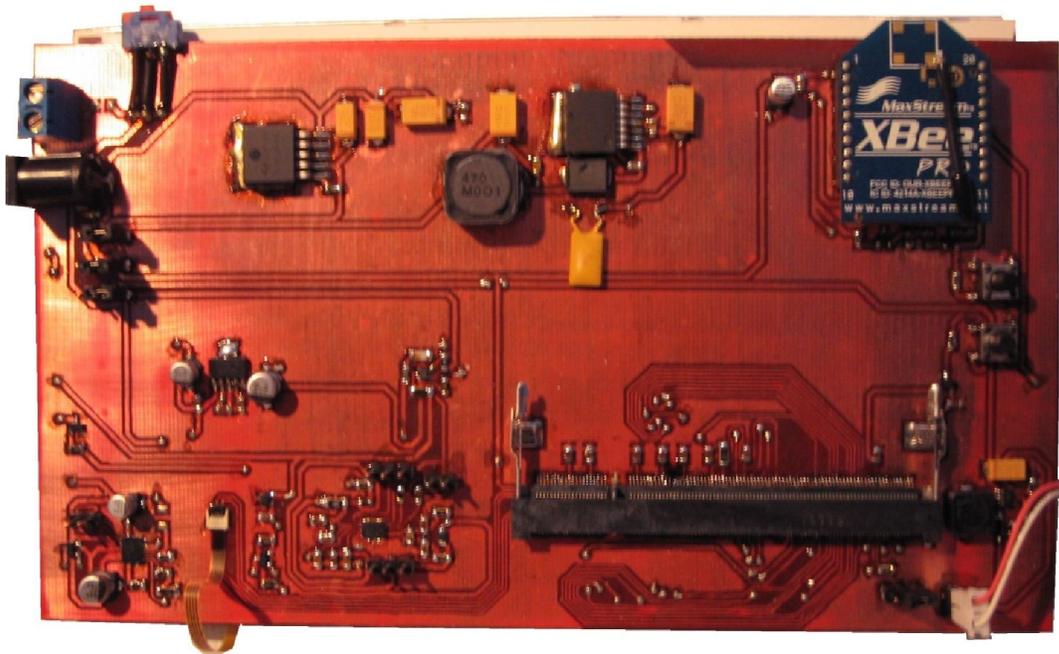


Fig. 17. Fotografía de la placa final con componentes realizado en la práctica.

### 3. DISEÑO MECÁNICO

Con respecto a la estructura mecánica del dispositivo, se prioriza la portabilidad del menú táctil, ya que es una de las especificaciones más importantes del producto. Debido al hecho de que el prototipo carece de partes mecánicas y no tiene cables flexibles en la interfaz con el usuario por su característica de membrana táctil, presenta un diseño robusto.

El gabinete estará compuesto por una base plástica de polipropileno descubierta de un lado, donde se colocará la pantalla y se cerrará con una tapa del mismo material. El mismo tiene 141mm de ancho, 198mm de largo y 30 mm de alto, permitiendo la ubicación de toda la electrónica dentro de la misma.

Se eligió polipropileno por su bajo costo y su gran resistencia a productos corrosivos, golpes y rayaduras. Además, junto al poliuretano, éste es el más utilizado en la fabricación de carcasas económicas para tablets o dispositivos similares.

El diseño de la carcasa se realizó mediante el software CATIA, constando, como se indicó anteriormente, de una base y una tapa que encastran con muy poca holgura, a fin de facilitar el sellado de la misma. Luego de encastradas las piezas, se sujetan entre sí con cuatro tornillos, a fin de brindarle a la carcasa firmeza mecánica.

Dicha firmeza mecánica también se tuvo en cuenta a la hora de sostener la pantalla, por lo cual se dotó a la tapa de limitadores plásticos que forman, en conjunto, el slot para colocar la misma. A su vez, la placa de circuito impreso queda sostenida en cavidades dispuestas en los pilares de sujeción (ubicados en las esquinas), los que se encargan de fijarla una vez que se ensambla el conjunto base+tapa.

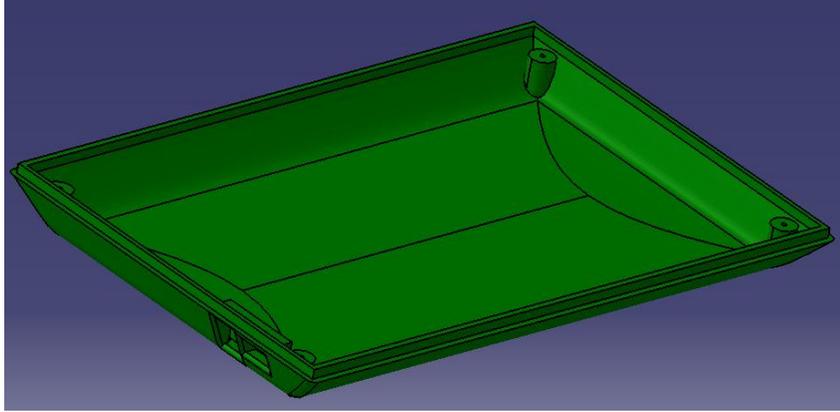


Fig. 18. Base del gabinete. Se observan los pilares de sujeción en las esquinas y los orificios para la alimentación.

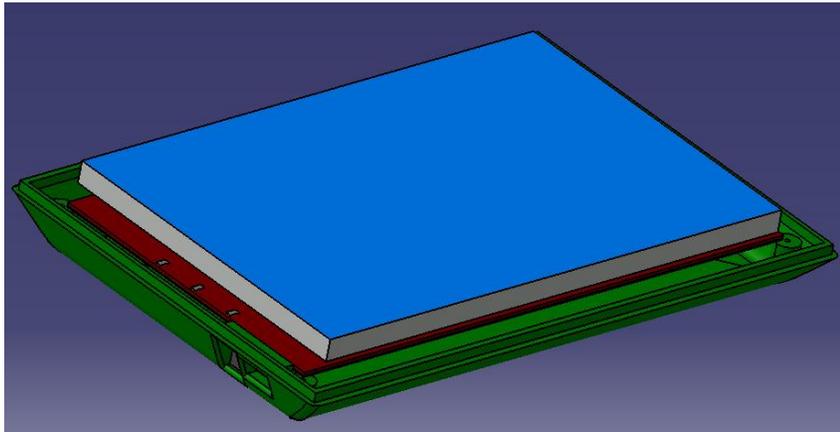


Fig. 19. Modelo de la base con la placa y la pantalla montadas.

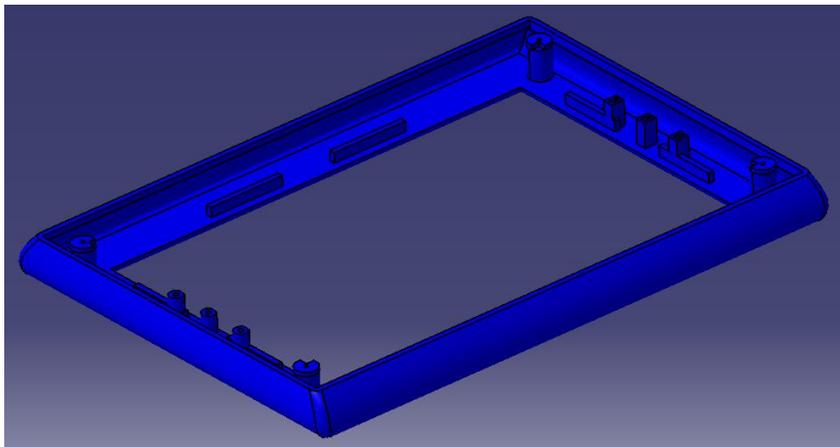


Fig. 20. Tapa de la carcasa. Se observan las secciones superiores de los pilares de sujeción, cada uno con alojamientos para la placa, y los orificios para LEDs de señalización. También se pueden señalar los limitadores de plástico que forman el slot donde se montará la pantalla.

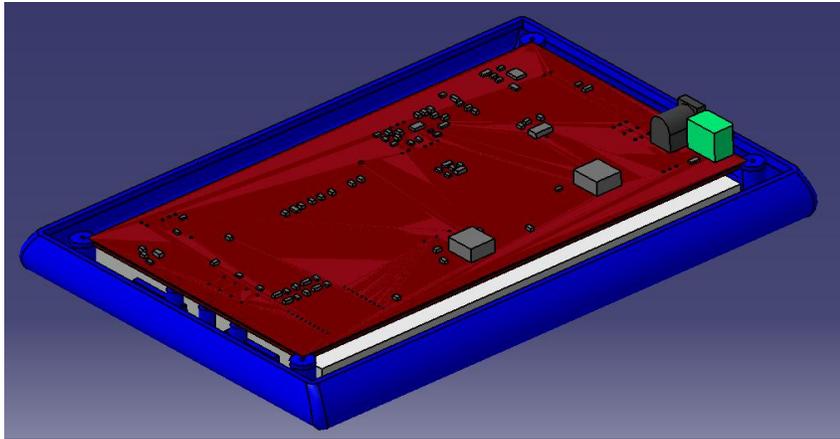


Fig. 21. Tapa con la placa y la pantalla montadas.

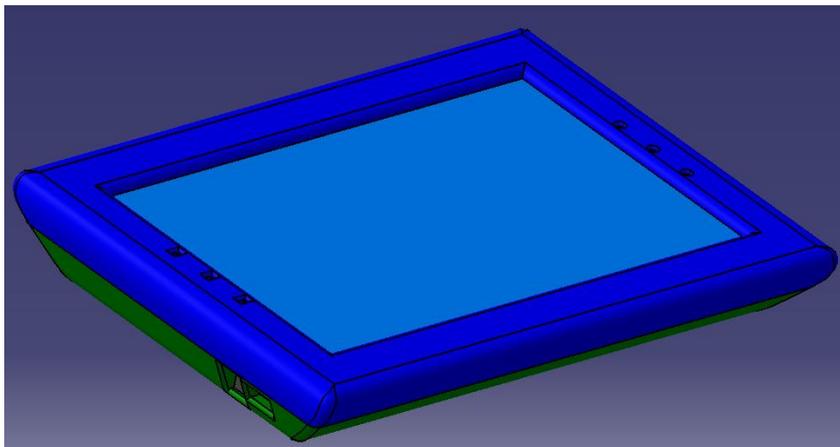


Fig. 22. Tapa y base ensambladas. Se observa la pantalla en su interior. Los colores elegidos son sólo usados para diferenciar las partes en la imagen y no indican el color que tendrá la carcasa en su implementación.

La carcasa cuenta con espacios que dan lugar a los puertos de alimentación, leds de señalización y switch de encendido/apagado. En cuanto a los orificios de alimentación y switch, los mismos pueden ser obstruidos con goma de poliestireno expandido, a fin de sellar el gabinete y evitar el ingreso de partículas o componentes indeseables mientras el cliente lo utiliza (este sellado es removible a fin de que el personal del local pueda cargar el equipo). Finalmente, los conductos de luz para leds quedan cubiertos de plástico traslúcido a fin de permitir utilizar los mismos como señales, pero no ser vías de acceso al interior de la carcasa.

Desde el punto de vista estético, se diseñó la carcasa de modo tal que, al ensamblar la base con la tapa, conformen un conjunto de apariencia continua, sin transición demasiado perceptible entre cada parte. Para lograr este fin, se utilizó una juntura que encastra la tapa con la base desde el interior de las mismas, sin alterar la forma exterior.

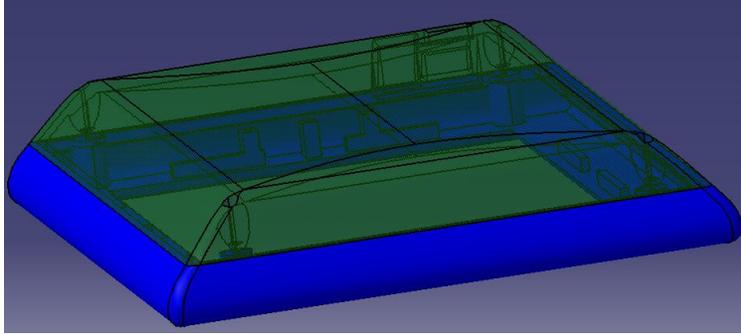


Fig. 23. Carcasa ensamblada. La base se muestra como transparencia a fin de que pueda apreciarse el encastre con la tapa.

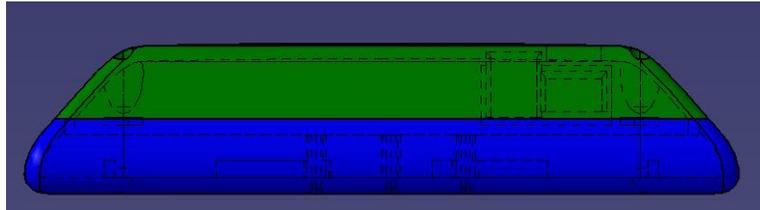


Fig. 24. Perfil del conjunto ensamblado. Se observa en línea punteada el encastre de las partes.

Se priorizó, además, un diseño con curvas agradables y suaves, que resulten cómodas en la mano del usuario y sean agradables a la vista.

En cuanto al color, a la hora de construir el prototipo existieron límites en cuanto al material y pintura disponibles. Sin embargo, el producto finalizado se fabricará con una carcasa negra, por ser un color adecuado al ser útil en contextos tanto sobrios como festivos, ocultando más fácilmente manchas y pequeños rayones que otros colores.

Por otro lado, se inspeccionó que cada uno de los componentes a utilizar fueran libres de plomo con una tolerancia de plomo de 0.1% del peso en materiales homogéneos. El producto al ser manufacturado por un tercero (compañía Gentle Industries Limited), se verificó que dicho ente cumpla con la norma de **Restriction of Hazardous Substances Directive** (Directive on the restriction of the use of certain hazardous substances in electrical and electronic equipment) adoptada por la Unión Europea en febrero de 2003, entrando en vigencia el 1 de julio de 1996. Dicha norma restringe los materiales usados en la fabricación de equipos eléctricos y electrónicos, aunque normalmente se la asocia a “Lead-free laws”, es más amplia y restringe el uso de seis materiales peligrosos:

- Plomo (0.1% sobre el peso del material homogéneo)
- Mercurio (0.1%)
- Cadmio (0.1%)
- Cromo Hexavalente (0.1%)
- PBB (0.1%)
- PBDE (00.1%)

Cabe aclarar, que esta restricción no se aplica al producto final sino a **cada sustancia separable mecánicamente**.

#### *4. DETALLES DE CONSTRUCCIÓN Y PRECAUCIONES ESPECIALES DE MONTAJE*

Nuevamente se resalta la importancia de construcción de los circuitos impresos que componen a nuestro sistema, teniendo todos los integrados de montaje superficial en una sola capa, debido a que en la capa despoblada de integrados estará enfrentada a la parte trasera de la pantalla. De esta manera la temperatura de la pantalla no afecte los integrados de la placa principal. El montaje del PCB para el producto final se realiza con una maquina pick & place y el soldado realizado por doble ola, según nos informa la compañía Gentle Industries Limited.

Los productos electrónicos de hoy en día, como por ejemplo: los televisores modernos LCD, contienen internamente el manual de usuario, además de tener el manual disponible digitalmente. Por estas mismas razones, se ofrecerá un manual de usuario impreso y uno almacenado internamente dentro del menú digital para la comodidad del usuario. Entonces, en el manual de usuario, como así también internamente almacenado en un archivo de lectura en la interfaz gráfica del dispositivo, se incluirán los siguientes cuidados de la batería, que se resumen a continuación:

##### Cuidados para alargar la vida útil de una batería de litio:

- Recomendable colocar en un sitio fresco (15 °C) cuando no esté en uso.
- En la mitad de la vida media dejarlas con carga intermedia (40%), la cual se indicará a través de la pantalla el estado de carga.
- Evitar mantenerlas con carga completa durante largos periodos. La interfaz gráfica indicará con una alarma visual el momento de carga completa para indicar la desconexión del dispositivo.
- La primera carga en una batería de litio no es decisiva en cuanto a su duración y no es prescindible hacerla; es igual al de las siguientes. (Mito probablemente heredado de las baterías de níquel)

La elección de una batería de litio forma parte de la sustentabilidad de nuestro producto y se optó por el uso dicha tecnología luego de comparar con las distintas alternativas que se encuentran en el mercado. A continuación se enumeran las ventajas de la batería de litio:

- **Portátil:** Para teléfonos móviles y reproductores MP3 que se encuentran difundidos en el mercado.
- **Elevada densidad de energía:** Acumulan mucha mayor carga por unidad de peso y volumen.
- **Poco peso:** A igualdad de carga, son menos pesadas y ocupan menos volumen que las de tipo Ni-MH y mucho menos que las de Ni-Cd y Plomo.
- **Poco espesor:** Se presentan en placas rectangulares, con menos de 5 mm de espesor.
- **Descarga lineal:** Durante toda la descarga, el voltaje de la batería varía poco, evita circuitos reguladores. Fácil saber la carga que almacena la batería.
- **Larga vida:** Más de 3.000 ciclos de carga/descarga para una pérdida de capacidad del 20% a C/3.
- **Muy baja tasa de autodescarga:** Baterías de Ni-MH tienen una "autodescarga" de 20% mensual. La tasa de autodescarga de una batería de Li-Ion es de menos un 6%. Tras 6 meses en reposo, retienen un 80% de su carga.

Cabe destacar, que debido a la pantalla táctil del dispositivo se ofrece al usuario una interfaz puramente gráfica, donde las acciones del usuario se llevarán a cabo en el software del mismo, sin alterar el hardware del sistema. De esta forma se obtiene un diseño robusto, impidiendo al usuario dañar el hardware del dispositivo sin romper el gabinete externo.

## X. Validación del Prototipo

El objetivo de la validación del prototipo es verificar si se satisfacen las especificaciones del dispositivo. Parte de la validación permite la posterior verificación de la confiabilidad del proceso, que luego formará la maduración del diseño. Con esto se busca el monitoreo de la confiabilidad y comprobar que la confiabilidad estimada sea la correcta y tener de esta manera un proceso de mejora continua.

### 1. VALIDACIÓN DEL HARD

#### 1.1. Plan y protocolos especiales de medición

Es importante aclarar, que exceptuando las señales del conversor AD que requiere la pantalla táctil, el resto de las señales del dispositivo son señales digitales de 0V, 3,3V ó 5V y no requieren protocolos especiales de medición. Por otro lado, el uso de fuentes switching y del LDO permite disminuir el Ripple a un nivel despreciable por los integrados. Esto quiere decir, que con ayuda de un tester y los leds de estado previamente detallados, se puede lograr la validación del producto.

Una alternativa que se tiene gracias a las herramientas disponibles dentro de la facultad es probar el programa mediante el kit de evaluación de Apexar, que se encuentra en el centro de investigación y desarrollo G.E.D.A del ITBA. Dicho kit es 100% compatible con nuestro microprocesador por ser especialmente para el SOM9G45. No obstante, se cuenta con un kit propiamente desarrollado para la programación del módulo microprocesador.

Para proseguir, se debe verificar que el administrador de alimentación funcione correctamente, para lo cual se debe someter el mismo bajo las siguientes condiciones:

- Sistema con batería descargada y alimentado por transformador.
- Sistema con batería y alimentado por transformador, dentro de lo cual si tiene poca batería debería cargarse, si el sistema no exige demasiada corriente limitando el máximo proveído por el adaptador.
- Sistema con batería llena y sin conexión del transformador, debería tener 2 horas de autonomía en pleno uso. Debe notificar al usuario cuando la batería tiene poca carga.

#### 1.2. Mediciones

La verificación de la alimentación de 3,3V y 5V a la salida de las fuentes LDO y switching no requiere de procedimientos especiales, simplemente se puede medir mediante un multímetro la tensión a la salida de los mismos. En la siguiente figura, se muestra que para una entrada de 11,955V (12V nominal), el módulo de fuente de alimentación logra estabilizar dicha entrada a un nivel de tensión de 4,915V (5V nominal) para el módulo de administración de alimentación. A la salida del controlador de carga, se obtiene mediante el LDO una salida de 3,144V (3.3V nominal) para el funcionamiento del módulo microprocesador, además de la salida de 4,195V (4,2V nominal) para la carga de la batería.

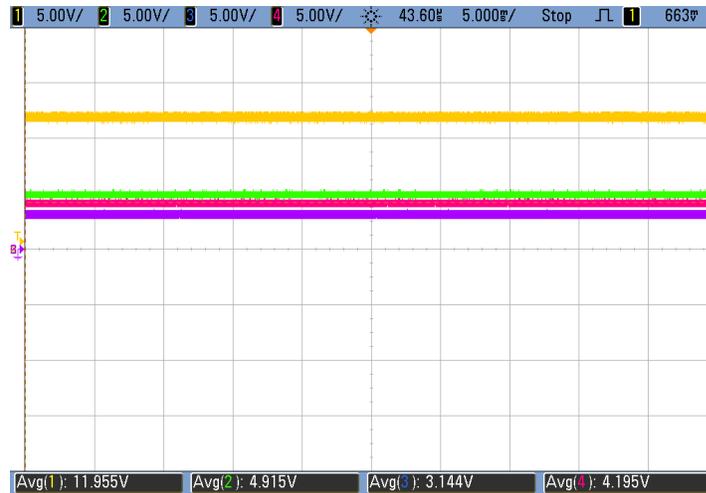


Fig. 25. Medición de tensiones de alimentación principales de la motherboard.

El cargado del software desarrollado para la aplicación se realiza mediante la conexión del microprocesador con un PDK (Platform Development Kit) corriendo en una computadora con Linux o mediante una computadora virtual simulando Linux. Luego, se procede a leer el Log accedido por el puerto de programación del microprocesador y verificar la instalación correcta de los drivers, archivos y programas que nuestro dispositivo necesita. Una vez realizada la instalación correcta en el sistema embebido, se obtendrán señales en las distintas terminales del conector SODIMM 200. El Log obtenido se detalla en las líneas de código devuelto por el Log que se encuentra en el Anexo II.

Seguidamente se prueba la comunicación I2C, en especial para la lectura de los estados del cargador de batería. El integrado BQ27500 encargado de mandar la información de la batería como el estado de carga, temperatura y tensión actual se comunica por el pin SDA con la frecuencia copiada por el clock del SCL. El estándar de la frecuencia especificado para el protocolo I2C básico es de 100kHz y debe tener un rise-time máximo de 1useg.



Fig. 26. Medición de SDA (verde) y SCL (amarillo) para conocer el estado de carga de la batería.

A partir de la figura previa, se verificó que los tiempos de flancos utilizados se encuentran por debajo del límite especificado por el protocolo de comunicación. De esta forma, se asegura que los datos obtenidos por el microprocesador son correctos, pudiendo controlar la carga de la batería óptimamente.

Acto seguido, se pone a prueba el administrador de alimentación, como se muestra en la siguiente figura. El ensayo se realizó sobre una batería de litio-ion con un estado de carga medio de 3,78V (3,7V nominal). La tensión de carga de la batería es mediante 4,2V nominal y una tensión mínima de 2,5V, se considera a la batería profundamente descargada. El estudio consiste en tres partes:

- 1) el dispositivo se encuentra alimentado por el transformador y carga la batería
- 2) se desconecta el transformador y el sistema debe seguir alimentado sin interrupción mediante la batería
- 3) se vuelve a conectar el transformador y se debe detectar dicha alimentación de entrada, volviendo al primer paso.



Fig. 27. Medición de la carga de batería ante distintas condiciones.

En la figura posterior, se manifiesta el salto de tensión en la batería, debido a que pasa del estado de carga al estado de descarga. En todo momento se mantuvo prendido el dispositivo, comprobando la ininterrupción del sistema, volviendo al estado de carga luego de la detección nuevamente de alimentación de entrada.

Finalmente se prosigue a conectar el módulo XBee en una topología loopback como recomienda el fabricante para comprobar la correcta transmisión de datos de la misma. Por otro lado, se puede corroborar la conexión del Xbee y sus alcances, mediante la transferencia de datos a un receptor en una terminal.

### 1.3. Evaluación

Las tensiones del módulo de alimentación del sistema se encontraron dentro del rango de valores tolerados, permitiendo que los módulos subsiguientes funcionen correctamente. La comunicación I2C funciona con una frecuencia levemente menor a la estándar de 100kHz y tiene un rise-time de la mitad del valor máximo especificado por el protocolo, logrando la lectura de datos correctos de la batería y permitiendo su administración precisa. El controlador de carga trabajó como es de esperar ante las distintas pruebas realizadas, proveyendo alimentación continuamente al sistema. La comunicación inalámbrica mediante el módulo Xbee-Pro funciona correctamente hasta la frecuencia máxima de transferencia permitida, especificada por el fabricante.

#### *1.4. Resultados*

La medición mostrada en la figura 25 indica que las tensiones de las fuentes de alimentaciones de la motherboard funcionan dentro del 5% de tolerancia con respecto a su valor nominal. El resultado es mejor del esperado, ya se encuentre dentro del 10% de tolerancia del nivel de referencia de tensión (entre 3.0V hasta 3.6V) como especifica la hoja de datos del SAM9G45.

En cuanto a la membrana táctil se utilizó una calibración de 4 puntos para la pantalla resistiva. El driver involucrado en el firmware de calibración devuelve constantes, las cuales se pueden guardar para una configuración permanente, ver anexo III para más información. La sensibilidad de la pantalla puede ser mejorada utilizando una membrana capacitiva, no obstante presenta un precio de hasta 4 veces mayor. El resultado de la calibración es aceptable para el manejo de la pantalla con el dedo. Para mayor precisión es recomendable un stylus o punzón estilizado. Dicho instrumento moderno de escritura para dispositivos con pantallas táctiles también beneficia la protección y limpieza del menú electrónico.

El protocolo de comunicación de I2C, como se manifiesta en la figura 26, funciona a una frecuencia nominal de 100kHz, se encuentra también por debajo del 10% de tolerancia especificado por la norma, corroborando el correcto funcionamiento del módulo de alimentación de batería. De esta forma, se procedió a verificar la carga de la batería mediante alimentación externa y en ausencia de ésta se comprobó la autonomía del sistema, como se ejemplifica en la figura 27.

Otro resultado importante fue la prueba de distancia de comunicación del Xbee-Pro, que logró sin problemas hasta aproximadamente 100m de distancia en el ambiente urbano de prueba.

Luego de realizar los exámenes para la validación del prototipo, se concluye que bajo el ambiente de trabajo sometido al dispositivo, se cumplen las especificaciones del producto, sin requerir rediseño del producto.

#### *11.2 Validación de software*

Debido a que el funcionamiento del software consiste en varias etapas, algunas de ellas independientes entre sí, se eligió realizar la depuración para cada una de estas. De todas maneras, una vez validada cada una de las etapas, se realiza una depuración final del programa en su totalidad.

Hay que tener en cuenta que todas las pruebas a realizar son mediante programas de prueba, ya que lo que se debe evaluar es el funcionamiento del software propio del proyecto. La funcionalidad de los drivers que interactúan con el hardware, llámese el caso del driver de la membrana táctil, ya vienen incluidos en el firmware de Linux utilizado y por lo tanto se supone que los mismos ya fueron validados.

##### *Validación del driver de inicialización:*

Se crea un programa de prueba que simplemente inicialice la pantalla principal. Se debe verificar que se cargue correctamente la imagen de fondo, que su resolución sea la adecuada, y que el mismo pueda cerrarse sin inconvenientes tanto por línea de comandos como por el ingreso de la tecla q.

Para corroborar que no se produzca fuga de memoria en el sistema al cerrar el programa, es decir que se libere correctamente la misma, se corre el programa dentro de una máquina virtual que simula el comportamiento del producto. Junto al mismo se puede correr un programa especializado que mide la memoria utilizada, y verificar que al abrir y cerrar el programa unas cuantas veces la memoria del sistema no vaya en aumento. El uso del mismo se muestra en la siguiente figura:

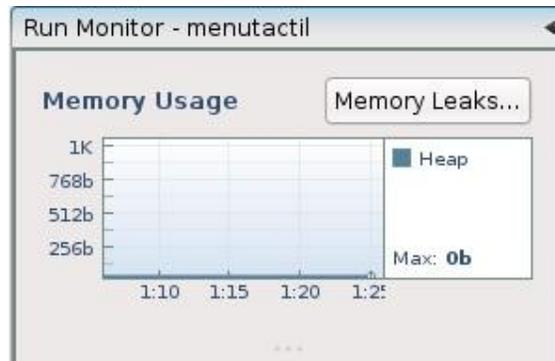


Fig. 28. Acumulación de memoria del programa a lo largo del tiempo.

En la figura se observa que la acumulación de memoria (heap) a lo largo del tiempo del programa se mantiene constante y en cero, lo cual indica que no se producen fugas de memoria en el mismo, y es lo deseado para un adecuado funcionamiento.

#### *Validación del driver de selección de mozo y mesa:*

Para la validación de este driver se crean dos programas de prueba por separado, uno para la validación de la selección de mozo y otro para la selección de mesa.

En el primero de estos, el de selección de mozo, se corre varias veces un programa donde se cargan a la lista varios mozos con sus fotos correspondientes de distintos tamaños, pesos e imágenes para verificar que se muestren correctamente. Cada vez que se corre el programa se modifica la cantidad de mozos agregados para analizar los distintos escenarios. A continuación se muestra una cierta lista del programa de prueba mencionado.

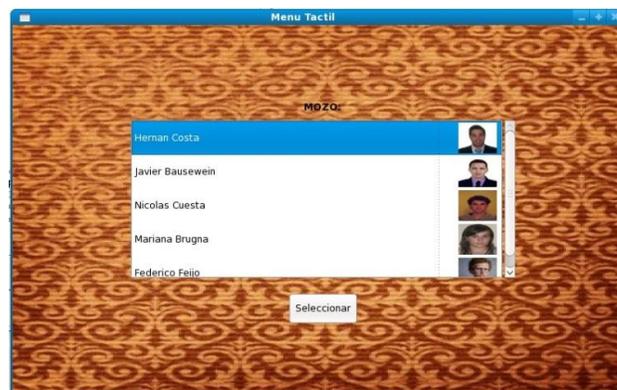


Fig. 29. Programa de prueba del driver de mozos.

Para el segundo caso, el programa de prueba consiste en mostrar las mesas con estados dados aleatoriamente para cada una para verificar que todos se observen debidamente. Cada vez que se selecciona una mesa, se verifica que se llame correctamente a la función que realiza la comunicación con el servidor mostrando el texto a mandar, pero sin realizar la comunicación en sí ya que esta validación se realiza en el driver de comunicación serial. En la siguiente figura se muestra una captura del programa de prueba descrito anteriormente.

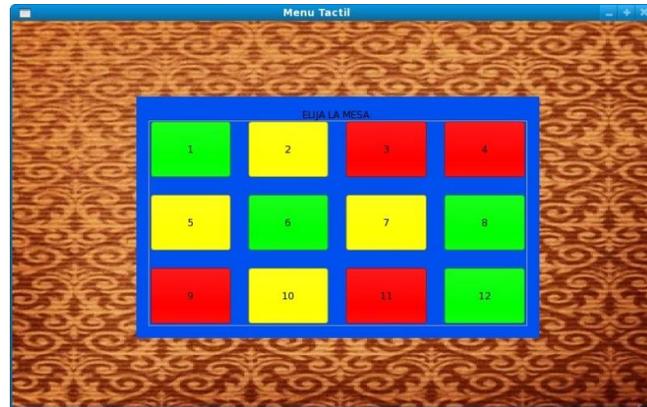


Fig. 30. Programa de prueba del driver de mesas.

#### *Validación del driver de comunicación serie:*

La comunicación serie se realiza entre el dispositivo y el servidor, y se envía y recibe mensajes de ambos lados, por lo tanto es necesario realizar una validación para ambas partes.

Primero se realiza la validación del envío de cada tipo de mensaje posible por separado. Como se verifica tanto para el servidor como para el producto, todo mensaje es enviado en un lado y recibido en el otro. Para la parte de recibir se realiza un programa de prueba que envíe el mensaje para verificar que se lea correctamente en otro programa de prueba que se corre ya sea en el servidor o en el dispositivo. A la hora de enviar, simplemente se muestra por medio de algún servicio como Hyperterminal que el texto enviado por el programa de prueba realizado sea correcto.

Los mensajes enviados que deben enviar información adicional, es decir además de que tipo de mensaje se trata, se prueban varias veces con datos variables en longitud.

Existe otro tipo de datos aparte de la comunicación dada por los paquetes con información en forma de texto y se presenta a la hora de enviar las imágenes, ya sea para las comidas como para los mozos. Estos envíos son de larga duración y no tienen una verificación de acknowledgment, por lo tanto son los más sujetos a errores. Para su validación se realizan varias pruebas de envío con distintos tamaños de imágenes (siempre inferiores a 10 KB por limitación impuesta) mediante un programa de prueba que recibe los datos y una vez finalizada la transacción muestra la foto. Hay que tener en cuenta que el envío de fotos siempre se da desde el servidor al producto, por lo que el programa de prueba solo se realiza para este último.

Una vez validados cada envío de paquetes por separado, se debe realizar la verificación de su interacción. Estos casos se dan como por ejemplo a la hora de enviar los mozos, donde primero se solicita el mozo, luego se mandan los datos, se verifica si se encuentra la foto, en caso negativo se envía dicha foto, y se repite para cada mozo disponible. Para la validación de esta comunicación se realiza un programa prueba tanto en el servidor como en el producto, donde en este último se muestran al finalizar el resultado obtenido, en el caso de la comunicación para mozos una lista con los mismos y sus imágenes correspondientes.

#### *Validación del driver de base de datos:*

Para la validación de la base de datos, primero se realiza un programa para probar que se agreguen correctamente los datos a la misma. En este programa llama varias veces a la función encargada de esta tarea con diferentes parámetros, y luego se imprime la línea de comandos los valores que se van leyendo de la base de datos para verificar que se han guardado correctamente.

Una vez que se valida la correcta carga de información a la base de datos de la comida, se puede pasar a la verificación del adecuado funcionamiento de las demás funciones de dicho driver. Entre estas funciones se encuentra la devolución de información de ciertos parámetros dentro de cada comida, por lo tanto se crea un programa de prueba que solicite la información de las diferentes comidas y lo imprima en

la consola de comandos para validar tanto que se han guardado correctamente nuevamente, como el funcionamiento de estas funciones.

El último punto a validar es que la información contenida en la base de datos se cargue apropiadamente en la lista de las comidas, para lo cual el programa de prueba debe correr una lista con pestañas donde en cada una se carguen las comidas correspondientes a cada tipo de comida dadas en la base de datos.

*Validación del driver de la barra superior:*

Para la validación de la barra superior se crea un programa de prueba que simplemente muestre dicha barra. Aquí se muestra la fecha y hora, un botón para llamar al mozo, el nombre y la foto del mozo y la mesa, y la imagen de la carga de la batería. En dicho programa se cargan inicialmente los valores del mozo y la mesa, y el valor de la carga de la batería.

Se debe verificar que la fecha y hora correspondan con los dados por el sistema operativo, y que la misma se actualice exactamente cada segundo. El botón de llamar al mozo simplemente debe mostrar el dialogo de confirmar, ya que el envío en si se valida en el driver de comunicación serie. Luego verificar que se muestren apropiadamente el nombre y la imagen del mozo con distintos parámetros, y lo mismo para la batería para distintos valores de carga. Se puede observar el programa de prueba funcionando correctamente en la figura mostrada a continuación.



Fig. 31. Programa de prueba del driver de la barra superior.

*Validación del driver de batería:*

A la hora de realizar la validación del driver de la batería se realiza un programa de prueba que inicialice la comunicación mediante i2c con el integrado encargado de la medición de los parámetros de la batería. Se verifica que la comunicación sea exitosa, y en tal caso se solicita la información de distintos valores de la batería como la temperatura, el voltaje y la corriente consumida. Estos parámetros son requeridos ya que se puede corroborar el apropiado funcionamiento del programa al poder medir los mismos físicamente en la batería por medio de un tester (para la tensión y corriente). Hay que tener en cuenta que este es el único driver que interactúa con el hardware del que seamos responsables, y por lo tanto se puede tener una concordancia física y no solo visual del funcionamiento.

A continuación se muestra la salida de los valores medidos del programa de prueba de batería. Los valores a medir son tensión, temperatura y estado de la carga. Este último porque es el que se muestra en realidad en el driver de la barra superior.

```
Temperature: 26 Å°C
Voltage 4152 mV
State of charge: 1 %
# █
```

Fig. 32. Salida por línea de comandos del programa de prueba de la batería.

Una vez que todos los drivers funcionan correctamente, se procede a depurar lo que se puede llamar la máquina de estados principal del programa, es decir el funcionamiento en conjunto de los drivers de pestañas, de lista y de detalle de comida. Con este fin se prueba darle todos los eventos posibles. Al verificarse que todo reacciona perfectamente ante cada evento posible se considera que la el programa principal funciona de manera correcta y corresponde a lo descrito en la ingeniería de detalle del software.

Después de verificar el acertado funcionamiento del estado principal del programa, se procede a realizar las pruebas del menú electrónico en condiciones de uso. Con tal fin se crea una base de datos en

el servidor con varias comidas y mozos a ser cargados. Luego se inicia el programa en el producto y se hacen pruebas generales de manera lo más variada posible intentando buscar posibles fallos en el funcionamiento del programa. Este último paso está más dirigido a verificar que el programa final realizado sea lo suficientemente intuitivo, cómodo y agradable para el uso cotidiano. De esta manera, si se llega a encontrar algún inconveniente o incomodidad en el uso se puede utilizar esta información y realimentarla con el fin de mejorar la experiencia del usuario.

Cuando se termina de realizar todos estos pasos se puede considerar que el programa diseñado se encuentra depurado, validado y funcionando como corresponde.

## XI. ESTUDIO DE CONFIABILIDAD

### 1. Estudio de confiabilidad del hardware

La confiabilidad del hardware debe ser inicializada desde la concepción hasta la obsolescencia del producto. En por esta razón, que se tiene una etapa de diseño de confiabilidad, que mejora la fiabilidad del producto utilizando la experiencia previa sobre fallas que se estudiaron en el proceso de verificación y pruebas de confiabilidad. La verificación de la confiabilidad garantiza que los requerimientos de fiabilidad del producto son los estimados. Luego de dicha etapa, se debe realizar el análisis físico para entender la causa de los modos de falla. Estas etapas forman parte de la ingeniería recurrente y de la mejora continua del producto.

Para el estudio de confiabilidad se utilizan los parámetros establecidos en la norma militar estadounidense MIL-HDBK-217F. Dicha norma constituye que el tiempo hasta la falla de los componentes es una variable aleatoria con distribución exponencial. Aceptando las condiciones impuestas, se calcula el parámetro de interés de la distribución en base a las características y el ambiente de uso de los elementos empleados. Cabe destacar, que nuestro sistema es puramente en serie, debido a que no se colocaron elementos de redundancia. Entonces la tasa de falla del sistema  $\lambda_{P\_total}$  es la suma de los  $\lambda_p$  de cada elemento. Además, se consideró que el ambiente de trabajo del sistema cae bajo la categoría Gm (Ground Mobile, móvil en tierra), esto se debe a que el usuario estará llevando el menú electrónico de mesa en mesa o al cargador. Por otro lado, se obtiene de este modo un margen con respecto a la categoría dentro de un laboratorio en condiciones estáticas.

Los componentes que pertenecen a la categoría de microcircuitos, arreglos de compuertas y/o microprocesadores como el integrado SAM9G45 requieren de un cálculo meticuloso. A continuación, se explica el modelo utilizado del microprocesador para estimar su MTBF (tiempo medio entre fallas). El integrado SAM9G45 tiene una configuración de 324 conectores del tipo BGA, no obstante, de la totalidad de las conexiones se utiliza cerca de 64. Además, se debe tomar en consideración, que el microprocesador puede configurar a muchos de las terminales PIO como periférico A, periférico B ó como Input ó Output en el sentido general. De esta manera, se estima que cada pin está conformado internamente por aproximadamente 64 compuertas MOS, como se encuentra en la hoja de datos del integrado. Con este método, se determina de la norma militar HDBK217 que el valor del factor C1 para 3000 hasta 10000 compuertas es de 0,16 (Tabla 5.1 pág. 5-3). Por otro lado, al tener un encapsulado de montaje superficial SMT, determinamos que el factor de complejidad C2 para 64 pines es de 0,025 (Tabla 5.9 pág. 5-14). Es importante resaltar, que el parámetro de característico  $\lambda_p$  de los demás integrados se determina como el producto directo de distintos factores, mientras que el  $\lambda_p$  de los microprocesadores requiere del producto de una sumatoria contemplando los factores C1 y C2 previamente hallados, respetando la ecuación:

$$\lambda_p = (C_1 \cdot \pi_T + C_2 \cdot \pi_E) \cdot \pi_Q \cdot \pi_L.$$



TABLA XIV  
DETALLE DEL CÁLCULO DE CONFIABILIDAD CON  $G_M$

ódulo	Componente	Valor	Cantidad	Factores de carga														$\lambda_p$ [10 <sup>6</sup> hs] (Por unidad)	MTBF[10 <sup>6</sup> hs] (Por unidad)	$\lambda_p$ [10 <sup>6</sup> hs] (Total)		
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$	$\pi_E$					
Alimentación	Polyswitch	1.1A	1	0.01000													8	0.080	12.50	0.080		
Alimentación	Varistor		1	0.00200	1.0												8	9	0.144	6.94	0.144	
Alimentación	Capacitor	100nF	2	0.00099	1.6			0.81	1	1							10	20	0.257	3.90	0.513	
Alimentación	Capacitor	22uF	1	0.00012	2.9			1.90	1	1							10	20	0.132	7.56	0.132	
Alimentación	Capacitor	10nF	1	0.00099	1.6			0.66	1	1							10	20	0.209	4.78	0.209	
Alimentación	Inductor	47uH	1	0.00003	1.8												3	12	0.002	514.40	0.002	
Alimentación	Capacitor	150uF	1	0.00012	2.9			3.40	2	1							10	20	0.473	2.11	0.473	
$\lambda_t$ (Total del módulo)																						1.554

Módulo	Componente	Valor	Cantidad	Factores de carga														$\lambda_p$ [10 <sup>6</sup> hs] (Por unidad)	MTBF[10 <sup>6</sup> hs] (Por unidad)	$\lambda_p$ [10 <sup>6</sup> hs] (Total)		
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$	$\pi_E$					
Boost	TPS61165		1		3.4							1			0.025	0.020	10	4	1.650	0.61	1.650	
Boost	Capacitor	4,7uF	1	0.00012	2.9			1.30	1	1							10	20	0.090	11.05	0.090	
Boost	Inductor	22uH	1	0.00003	1.8												3	12	0.002	514.40	0.002	
Boost	Diodo	40V@0,5A	1	0.00200	1.0		1.00	1.00									8	9	0.144	6.94	0.144	
Boost	Capacitor	1uF	1	0.00099	1.6			1.00	1	1							10	20	0.317	3.16	0.317	
Boost	Capacitor	220nF	1	0.00099	1.6			0.81	1	1							10	20	0.257	3.90	0.257	
Boost	Resistencia	1,2Ω	1	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.035	
Boost	Resistencia	10kΩ	1	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.035	
$\lambda_t$ (Total del módulo)																						2.529

Módulo	Componente	Valor	Cantidad	Factores de carga														$\lambda_p$ [10 <sup>6</sup> hs] (Por unidad)	MTBF[10 <sup>6</sup> hs] (Por unidad)	$\lambda_p$ [10 <sup>6</sup> hs] (Total)	
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$	$\pi_E$				
Bateria	BQ24070		1		3.4							1			0.025	0.020	10	4	1.650	0.61	1.650
Bateria	Resistencia	47k $\Omega$	1	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.035
Bateria	Capacitor	10uF	2	0.00012	2.9			1.60	1	1							10	20	0.111	8.98	0.223
Bateria	Resistencia	2,2k $\Omega$	3	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.104
Bateria	Resistencia	3,3k $\Omega$	1	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.035
Bateria	Resistencia	33k $\Omega$	1	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.035
Bateria	Capacitor	47uF	2	0.00012	2.9			2.30	1	1							10	20	0.160	6.25	0.320
Bateria	Switch		1	0.04000				1.00			1							18	0.720	1.39	0.720
Bateria	Diodo		2	0.00200	1.0		1.00	1.00									8	9	0.144	6.94	0.288
Bateria	Resistencia	100k $\Omega$	2	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.069
Bateria	MOS		3	0.00074	2.8		0.11					0.7					8	9	0.011	87.05	0.034
Bateria	Resistencia	10k $\Omega$	3	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.104
Bateria	BQ27500		1		0.3						1			0.025	0.020	10	4	0.875	1.14	0.875	
Bateria	Resistencia	1,8M $\Omega$	1	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.035
Bateria	Capacitor	33nF	1	0.00099	1.6			0.76	1	1							10	20	0.241	4.15	0.241
Bateria	Resistencia	18,2k $\Omega$	1	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.035
Bateria	Resistencia	1k $\Omega$	1	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.035
Bateria	Capacitor	100nF	10	0.00099	1.6			0.81	1	1							10	20	0.257	3.90	2.566
Bateria	Resistencia	0,01 $\Omega$	1	0.00370	1.3	0.440	1.20										10	16	0.406	2.46	0.406
Bateria	Resistencia	100 $\Omega$	3	0.00370	1.3	0.068	0.66										10	16	0.035	28.95	0.104
Bateria	Capacitor	0,47uF	1	0.00099	1.6			0.94	1	1							10	20	0.298	3.36	0.298
Bateria	TPS71525		1		0.3						1			0.025	0.010	10	4	0.475	2.11	0.475	
Bateria	TPS73733		1		2.4						1			0.025	0.020	10	4	1.400	0.71	1.400	

Módulo	Componente	Valor	Cantidad	Factores de carga														$\lambda_p$ [10 <sup>6</sup> hs] (Por unidad)	MTBF[10 <sup>6</sup> hs] (Por unidad)	$\lambda_p$ [10 <sup>6</sup> hs] (Total)	
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$	$\pi_E$				
Bateria	Capacitor	15pF	1	0.00099	1.6			0.44	1	1						10	20	0.139	7.17	0.139	
Bateria	Resistencia	5,6M $\Omega$	1	0.00370	1.3	0.068	0.66									10	16	0.035	28.95	0.035	
Bateria	Transistor		1	0.00074	2.8		0.11					0.7				8	9	0.011	87.05	0.011	
Bateria	Led		3	0.00023	1.0											6	8	0.010	98.81	0.030	
Bateria	Resistencia	470 $\Omega$	2	0.00370	1.3	0.170	0.82									10	16	0.107	9.32	0.215	
Bateria	Resistencia	330 $\Omega$	1	0.00370	1.3	0.170	0.82									10	16	0.107	9.32	0.107	
$\lambda_t$ (Total del módulo)																					10.621

Módulo	Componente	Valor	Cantidad	Factores de carga														$\lambda_p$ [10 <sup>6</sup> hs] (Por unidad)	MTBF[10 <sup>6</sup> hs] (Por unidad)	$\lambda_p$ [10 <sup>6</sup> hs] (Total)	
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$	$\pi_E$				
Comunicación	Capacitor	1uF	2	0.00099	1.6			1.00	1	1						10	20	0.317	3.16	0.634	
Comunicación	Transistor		4	0.00074	2.8		0.11					0.7				8	9	0.011	87.05	0.046	
Comunicación	Capacitor	10uF	1	0.00012	2.9			1.60	1	1						10	20	0.111	8.98	0.111	
Comunicación	Capacitor	100nF	1	0.00099	1.6			0.81	1	1						10	20	0.257	3.90	0.257	
Comunicación	Led		3	0.00023	1.0											6	8	0.010	98.81	0.030	
Comunicación	Resistencia	470 $\Omega$	2	0.00370	1.3	0.170	0.82									10	16	0.107	9.32	0.215	
Comunicación	Resistencia	330 $\Omega$	1	0.00370	1.3	0.170	0.82									10	16	0.107	9.32	0.107	
Otros	Pines		10	0.00100	1.5								1			2	8	0.024	41.67	0.240	
$\lambda_t$ (Total del módulo)																					1.640

Módulo	Componente	Valor	Cantidad	Factores de carga													$\lambda_p$ [/ $10^6$ hs] (Por unidad)	MTBF[ $10^6$ hs] (Por unidad)	$\lambda_p$ [/ $10^6$ hs] (Total)			
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$				$\pi_E$		
Pantalla	Conector 40POS		1	0.00100	1.5								1			2	8	0.024	41.67	0.024		
Pantalla	Resistencia	220k $\Omega$	1	0.00370	1.3	0.068	0.66									10	16	0.035	28.95	0.035		
Pantalla	Conector TS 4POS		1	0.00100	1.5								1			2	8	0.024	41.67	0.024		
Pantalla	Conector SODIMM		1	0.00100	1.5								1			2	8	0.024	41.67	0.024		
Pantalla	Capacitor	100nF	9	0.00099	1.6			0.81	1	1						10	20	0.257	3.90	2.309		
Microprocesador	SOM9G45		1		0.3						1			0.160	0.025	10	4	1.480	0.68	1.480		
$\lambda_t$ (Total del módulo)																						3.896

TABLA XV  
RESULTADOS DE LA CONFIABILIDAD CON  $G_M$

$\lambda_p$ del sistema [/ $10^6$ hs]	20.240
MTBF del sistema [ $10^6$ hs]	0.049
MTBF del sistema [años]	5.640

De las tablas posteriores se concluye que el módulo de mayor tasa de falla es el controlador de carga de la batería y el sistema, debido a la gran cantidad de componentes que conlleva el módulo. Esto quiere decir, que para mejorar la confiabilidad del sistema total, se debería empezar minimizando la tasa de falla en el módulo de batería, mediante componentes de redundancia o bien utilizando componentes de menor factor de carga. No obstante, se determina que la tasa de falla del sistema total es de  $\lambda_p=20,24$  [1/millón de horas], lo cual equivale a un MTBF del sistema de 5 años y 7 meses.

El valor calculado es esperable para un producto electrónico y aceptable para un equipo que se encontrará sometido a un ambiente móvil. En otras palabras, en caso de localizarse en un ámbito con uso estático, catalogado por la norma militar como Gf (Ground Fixed, fijo en tierra) en lugar de Gm (Ground Mobile, móvil en tierra), el MTBF puede mejorar rotundamente, como se muestra en la tabla IV. La diferencia entre ambas categorías radica en el coeficiente  $\pi_E$ , que es menor para el ambiente Gf, perteneciente a un entorno moderadamente controlado en temperatura sin encontrarse sometido a movimiento, a diferencia del ambiente Gm analizado. Para mayor detalle ver anexo IV, donde se encuentran los cálculos detallados de la confiabilidad con los valores de  $\pi_E$  pertenecientes un entorno estático Gf. A continuación, se resumen los valores de MTBF obtenidos:

TABLA XVI  
 RESULTADOS DE LA CONFIABILIDAD CON  $G_F$

$\lambda_p$ del sistema [ $10^6$ hs]	10.860
MTBF del sistema [ $10^6$ hs]	0.092
MTBF del sistema [años]	10.511

Al cambiar el contexto de trabajo de  $G_m$  a  $G_f$ , se estima que el MTBF del equipo mejora de 5 años y 7 meses a aproximadamente 10 años y 6 meses.

Además hay que tener en cuenta que la vida de ciclado de un dispositivo electrónico es hoy en día cada vez más corto, debido a que se reinventa el negocio de la electrónica y al avance tecnológico. Es importante aclarar, que tampoco se busca la longevidad del producto, principalmente debido a que el proceso implicaría un costo muy superior, debido a la calidad de componentes que éste exigiría utilizar.

En una segunda instancia, es de esperar, que el equipo electrónico tenga un ciclo de vida de 5 ó 6 años, considerando el tiempo de introducción, crecimiento, madurez y declinación del producto. Antes del retiro del producto, ya se contaría con una nueva versión del producto con mayores prestaciones para incentivar al cliente la adquisición del actual equipo, en lugar de reparar el dispositivo viejo. De esta forma, se empezaría un nuevo ciclo de vida con un producto más moderno.

## 2. Estudio de confiabilidad de software

Además de la confiabilidad del hardware, es necesario realizar un estudio de confiabilidad del software, ya que en nuestra aplicación este último es de gran importancia para el correcto funcionamiento del dispositivo.

Algunos de los errores que pueden suceder en caso de falla del software son, por ejemplo, de guardar incorrectamente algún pedido, lo cual si bien el dispositivo funciona, no cumple con lo esperado. Otro de los casos es un “cuelgue” del software, lo cual trae aparejado la imposibilidad de realizar tareas con el dispositivo que se traduce como una falla crítica. Esta última es la de mayor interés y para la cual se realiza el análisis de confiabilidad.

Para poder hacer el estudio de confiabilidad de software se asume para el análisis un modelo de fiabilidad exponencial. Con esta asunción, el estimador de fiabilidad queda de la siguiente manera:

$$R(t) = e^{-\lambda t} = e^{-k\varepsilon_T(\tau)t} = e^{-k(\varepsilon_T(\tau) - \varepsilon_C(\tau))t}$$

En esta ecuación  $\varepsilon_T(\tau)$  representa la proporción de errores totales y  $\varepsilon_C(\tau)$  es la tasa de corrección de errores.

Se desea hallar el tiempo medio entre fallas, para lo cual obteniendo el estimador de fiabilidad, el mismo puede hallarse de la siguiente manera:

$$MTBF = \int_0^{\infty} R(t) dt$$

Partiendo de las siguientes definiciones:

$$\varepsilon_C(\tau) = \frac{E_C(\tau)}{I_T}$$

$$\hat{E}_T = - \frac{\left[ \frac{\lambda_2}{\lambda_1} \right] \cdot E_C(\tau_1) - E_C(\tau_2)}{\frac{\lambda_2}{\lambda_1} - 1}$$

$$\hat{k} = \lambda_1 \cdot \frac{I_T}{\hat{E}_T - E_C(\tau_1)}$$

Asumiendo que las variables ET y k no varían en el tiempo y operando convenientemente se puede obtener el tiempo medio entre fallas, llegando a la siguiente expresión:

$$MTBF = \frac{1}{k \left[ \frac{E_T}{I_T} - \varepsilon_C(\tau) \right]}$$

Para poder llegar a un valor es necesario tener los valores de todas las variables pertenecientes a la ecuación expresada. Para esto es necesario realizar dos pruebas de evaluación como se puede observar en la definición de ET por ejemplo. Estas dos pruebas se realizan corriendo el programa realizado desde dos computadoras distintas (se realiza en computadoras al no contar con dos dispositivos distintos). Los resultados de la prueba llevan a los siguientes resultados:

TABLA XVII

RESULTADOS DE PRUEBAS DE SOFTWARE

	Tiempo (horas)	Errores encontrados	Errores corregidos	$\lambda$
Test 1	10	5	2	0,5
Test 2	6	2	1	0,33

Cabe aclarar que los errores no corregidos en las pruebas realizadas se debieron a eventos esporádicos, por lo tanto no fue posible encontrar su causa imposibilitando su corrección. Probablemente se puede atribuir estos errores al firmware de Linux embebido. A pesar de que este último es normalmente sometido a varias pruebas, no cuenta con el nivel de seguridad que presenta su homónimo convencional.

Esto último se debe a que Linux embebido es una adaptación para que pueda correr sobre arquitecturas ARM.

De todas maneras hay que tener en cuenta que los valores hallados en estas pruebas son emblemáticos y puede darse el caso de obtener valores diferentes a los hallados, con lo cual se puede modificar el MTBF obtenido.

Con estos datos se obtiene el siguiente MTBF:

$$\text{MTBF} = 25924.3 = \text{horas} \approx 3 \text{ años}$$

Por lo tanto el tiempo medio entre fallas es de aproximadamente 3 años. Este valor es relativamente bajo para tratarse de software. De todas maneras hay que tener en cuenta que este simplemente es un modelo planteado y no el único, ya que para el análisis de software no existe un enfoque tan certero como para la fiabilidad del hardware.

Otro punto a tener en cuenta es la tasa de fallas que presenta el software en comparación con el hardware. Esto ya difiere en el análisis a realizar, ya que cada vez que se realiza una actualización se presenta un aumento drástico en la tasa de fallo aunque los niveles de los tipos de fallo disminuyen gradualmente, en parte debido a los defectos encontrados y reparados después de las actualizaciones. Esta descripción puede verse representada en el siguiente gráfico:

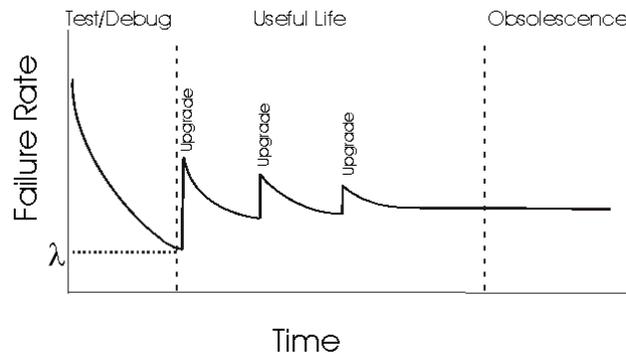


Fig. 34. Tasa de fallas a lo largo de la vida de un software.

Las mejoras que se muestran en la figura de arriba implican mejoras en las prestaciones del software, más que mejora en la fiabilidad, que es lo que generalmente se realiza ante una actualización del mismo. Es por esto que aumenta la tasa de fallo en cada actualización, ya que se aumenta la complejidad del software realizado. Esto conlleva a que el gráfico no presenta una curva plana donde la tasa de fallos se haga constante, con lo cual dificulta el cálculo de una MTBF.

Para las actualizaciones de fiabilidad, si se puede estimar una caída en la tasa de fallo de software, si el objetivo de las mismas simplemente implica la mejora de la fiabilidad del software para la corrección de errores. Esto se realiza mediante un rediseño o reimplementación de algunos módulos del software que utilizan mejores métodos de ingeniería. Por lo tanto en este caso tampoco se puede hablar de una tasa de fallos constante, ya que la misma disminuye con cada actualización.

## XII. CONCLUSIONES

### *Excelencias. Objetivos alcanzados.*

A lo largo del desarrollo de este proyecto, se logró adquirir conocimiento en software no utilizado previamente, como por ejemplo CATIA (diseño mecánico), Ruby on Rails (front-end para páginas web), etc.

Sumado a estos nuevos conocimientos de software, se realizó investigación de diversas áreas, abarcando, entre otras, hojas de datos de nuevos componentes, bibliografía sobre código en determinados lenguajes y prototipado rápido.

Se logró implementar con éxito cada módulo diseñado, tanto en software como en hardware. Cada uno de dichos módulos se diseñó, a su vez, a partir de requerimientos y especificaciones establecidas analizando los potenciales clientes y el mercado actual (y, dentro del mismo, competidores, disponibilidad de materiales, etc.).

Además del punto de vista técnico, debieron tomarse recaudos y consideraciones en cuanto a los aspectos económicos y legales, pensando el producto para que pueda salir a la venta en el nicho localizado dentro del mercado local, con un desempeño competitivo y rentable.

### *Fallos. Recomendaciones para futuros diseños.*

Desde el punto de vista de software, el producto puede ampliarse enormemente. Dentro de los posibles agregados que podrían implementarse, se encuentran:

- Ampliar el servidor web a una red de locales, pudiendo elegir a qué local se está accediendo desde la página web. Esto obligaría a localizar la web en un servidor remoto (o uno del local con conexión a internet), y permitiría, además de un manejo más global de sucursales, administrar los empleados de una manera conjunta.
- Introducir un campo de texto en el pedido realizado desde el equipo portátil, a fin de que puedan realizarse especificaciones para cada plato sin tener que comunicárselas al mozo.
- Permitir consultar el estado de carga de la batería de cada menú, desde el servidor.
- Incluir streaming de video en las unidades portátiles, lo cual adelantaría un paso enorme en los contenidos que podrían ofrecerse (publicidades más sofisticadas o filmaciones de la comida siendo preparada, por ejemplo).

En cuanto al hardware, también podrían realizarse mejoras, a saber:

- Permitir la carga de la batería mediante un puerto USB. Este puerto también podría utilizarse para realizar la programación del microprocesador.
- Diseñar un gabinete más hermético, incluyendo mecanismos que permitan abrir y cerrar los orificios para puertos, en vez de tener que obstruirlos con alguna goma.
- Desplazar los leds de señalización a la superficie de la carcasa (que no se ubiquen directamente sobre la placa, sino que queden más visibles casi al exterior del gabinete).
- Colocar un compartimiento en el gabinete que haga más práctico el recambio de la batería.

### *Pensamientos finales*

Más allá de los aspectos a mejorar, que podrían considerarse inagotables en cualquier producto electrónico, en particular en uno tan relacionado con el agrado del usuario, nos encontramos muy satisfechos con los resultados alcanzados y las herramientas adquiridas a lo largo del desarrollo del proyecto.

Luego de una ardua carrera como la que cursamos, nos resulta grato despedirnos a través de un producto tangible que muestre lo que hemos avanzado desde el día en que iniciamos primer año. Definitivamente, en aquel entonces no hubiéramos sido capaces de construirlo, y nos enorgullece que tanto esfuerzo comience a dar sus frutos.

De aquí en adelante seguiremos adquiriendo herramientas y conocimientos nuevos, pero, en el lapso de unos meses, no estaremos haciéndolo como estudiantes sino como profesionales. Podemos decir que este proyecto nos ha otorgado una pequeña ventaja para ingresar a esa nueva etapa, llena de nuevos desafíos y oportunidades.

### XIII. ANEXOS

#### 1. ANEXO I: ESTUDIO DEL MERCADO MEDIANTE ENCUESTAS

Con el fin de realizar una proyección sobre el mercado y analizar la importancia de ciertos requerimientos, se realizan encuestas a potenciales clientes (dueños de locales de comida) y usuarios (clientes de dichos locales).

Al principio, se realizó un pequeño número de encuestas en persona, a varios clientes de locales de comida y algunos de los dueños de los mismos. Las encuestas realizadas en esa fase, así como algunos de los resultados obtenidos, se muestran a continuación.

##### a. Encuestas originales

###### Preguntas para el comensal (basado en un total de 20 personas):

- 1) *¿Qué ventaja y desventaja le ve al uso de menús electrónicos touch?*  
80% opina que es beneficioso, 15% opina que no es necesario y 5% no opina.
- 2) *¿Preferiría el menú portátil o fijo a la mesa?*  
75% prefiere menú portátil, 20% fijo, 5% indiferente.
- 3) *¿Prefiere una interfaz en la que predomine texto o imágenes?*  
100% respondió que prefiere una buena relación de texto e imágenes.
- 4) *¿Utilizaría una cuenta con datos personales, por ejemplo: fecha de cumpleaños, comida favorita y comidas más pedidas?*  
85% momentáneamente no utilizaría una cuenta personal, 10% utilizaría y 5% no opina.
- 5) *¿Estaría interesado en recibir vía mail actualizaciones del menú?*  
90% no está interesado en recibir notificaciones vía mail, 10% está interesado.
- 6) *¿Le molestaría recibir publicidad mientras pide comida?*  
A un 70% le molesta la publicidad, a un 25% no le molesta si es publicidad discreta y 5% no opina.
- 7) *¿Qué tamaño y peso espera? (Igual, menor o mayor al de un menú)*  
65% espera mayor, 30% igual y 5% menor peso al de un menú convencional.
- 8) *¿Prefiere controlar la pantalla mediante el dedo o un dispositivo en forma de lápiz?*  
100% de los encuestados utilizarían el dedo para controlar la pantalla.
- 9) *¿Le gustaría recibir el tiempo estimativo de la demora de su comida?*  
90% espera recibir el tiempo de espera, mientras que a un 10% le es indiferente.
- 10) *Marque cuál de las siguientes opciones le interesaría puntuar:*  
 Comida                     Servicio (Mozo)                     Limpieza  
 Otros
- 11) *¿Le interesaría que la comida lleve puntuaciones?*  
80% está interesado en puntuaciones para la comida, 15% está interesado, pero no desea estar atado a completar puntuaciones, y 5% no está interesado.

###### Preguntas para dueños (basado en un total de 10 personas):

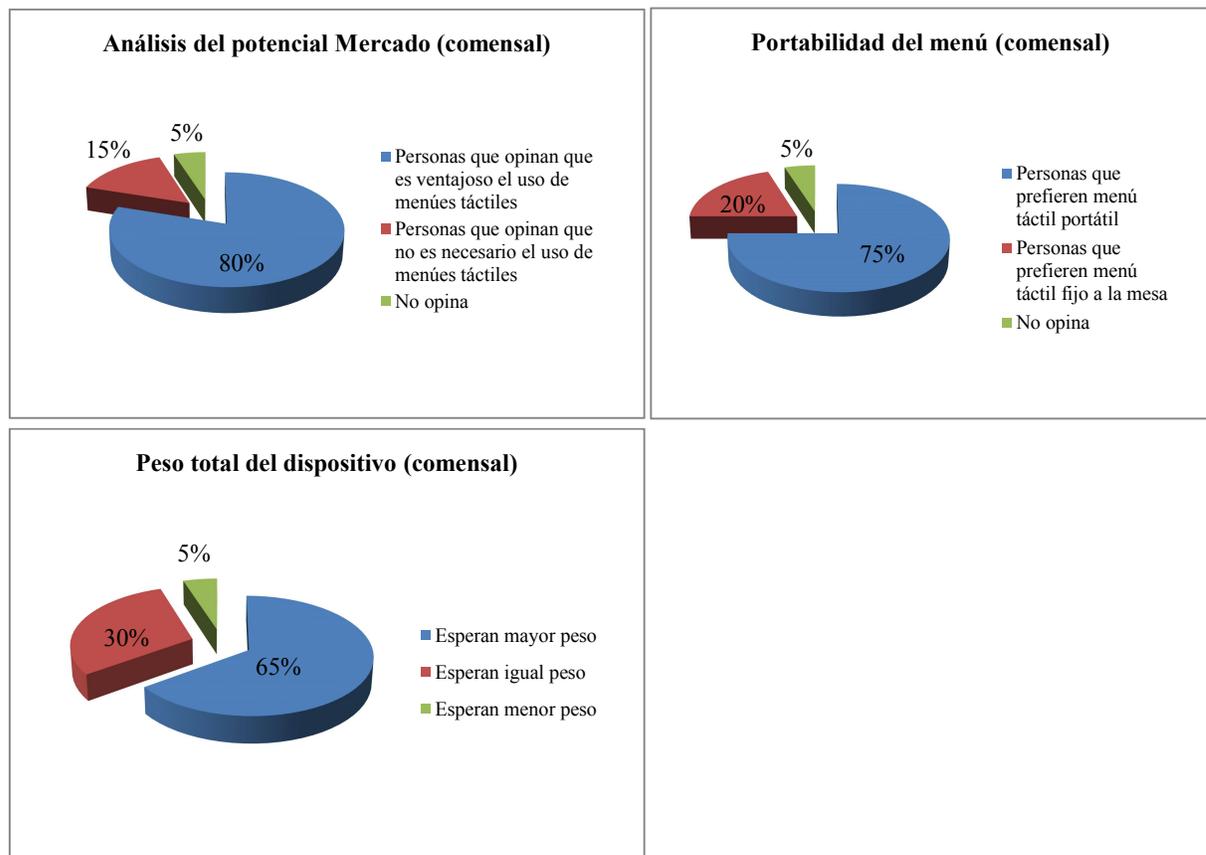
- 1) *¿Qué ventaja y desventaja le ve al uso de menús electrónicos touch?*  
80% opina que es ventajoso, 20% no opina.
- 2) *¿Qué opina sobre publicidad en el menú electrónico?*  
80% opina que es necesario y útil, 10% opina que no es necesario (en todo caso, publicidad discreta) y 10% no opina.
- 3) *¿Qué tamaño y peso espera? (Igual, menor o mayor al de un menú)*  
60% espera mayor, 30% igual y 10% menor al de un menú.
- 4) *¿Cuánto tiempo de funcionamiento precisa en condiciones normales de uso?*

Todos respondieron aproximadamente de 1 a 2 horas.

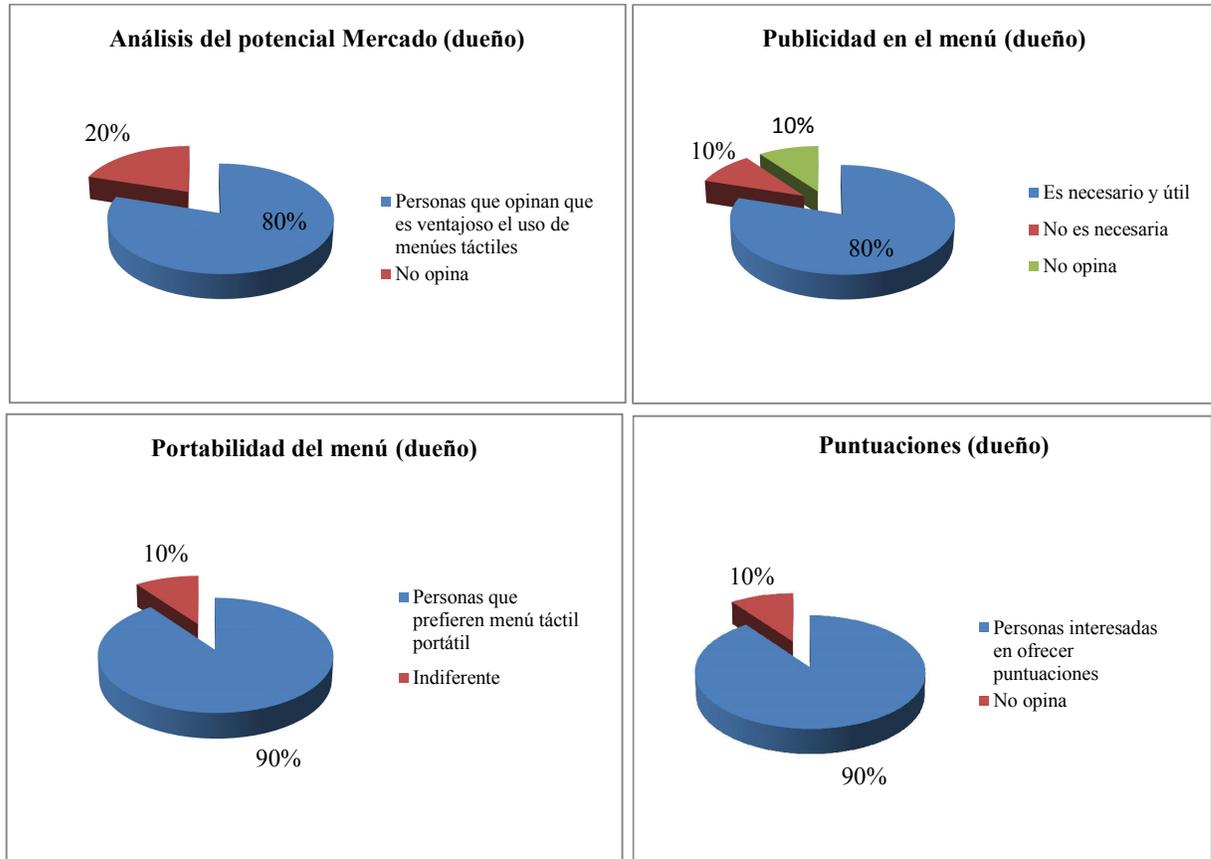
- 5) *¿Cuánto estaría dispuesto a pagar por el producto?*  
80% no pagaría más de 400 U\$\$, 20% no pagaría más de 300 U\$\$ por unidad.
- 6) *¿Preferiría el menú portátil o fijo a la mesa?*  
Un 90% prefiere menú portátil y al 10% les es indiferente.
- 7) *¿Le interesaría la estadística de ventas?*  
100% de los dueños está interesado en las estadísticas de ventas.
- 8) *¿Qué alcance debería tener el dispositivo (comunicación inalámbrica)?*  
Todos indicaron alcance suficiente para abarcar sus restaurantes.
- 9) *¿Le interesaría ofrecer el tiempo estimativo de la demora de la comida?*  
100% está interesado en ofrecer el tiempo de espera.
- 10) *¿Le interesaría que la comida lleve puntuaciones?*  
90% está interesado en ofrecer puntuaciones y 10% no opina.

La forma que se utilizó para mostrar la información fue fijar el 100% en las 20 personas encuestadas como usuarios y de los 10 dueños encuestados, respectivamente. A continuación, se muestran mediante gráficos de torta sólo los gráficos considerados más importantes, basados en los requerimientos que más se priorizaron debido al DFMEA:

Usuario comensal:



Usuario dueño:



Estas encuestas sirvieron como una primera aproximación a la hora de sopesar los distintos requerimientos y analizar la entrada al mercado de este proyecto. No obstante, **la muestra es demasiado pequeña y poco representativa del mercado en general** (por ser poco heterogénea) como para poder tomarla como referencia.

Por eso, se confeccionaron nuevas encuestas mediante servicios gratuitos en internet, que permitirán obtener muestras mucho más grandes y heterogéneas. Una vez que se posean resultados significativos, el análisis de factibilidad económica considerará el número más realista de posibles clientes obtenidos.

Las nuevas encuestas a realizar por vía online se muestran a continuación.

b. Encuestas online

Página 1 - Introducción



## Menú táctil para locales de comida

\*Obligatorio

**Indique su género**

Masculino

Femenino

**Indique en qué rango se encuentra su edad**

Entre 18 y 24 años

Entre 25 y 35 años

Entre 35 y 45 años

Entre 45 y 60 años

Más de 60 años

**¿Es usted dueño de un local de comidas? \***

Seleccione si usted interactuaría con el menú como el dueño de local de comidas o un comensal del mismo.

Sí.

No, soy un comensal.

[Continuar »](#)

Con la tecnología de [Google Docs](#)

[Informar sobre abusos](#) - [Condiciones del servicio](#) - [Otros términos](#)



Menú táctil para locales de comida

Encuesta para Comensal

**¿En qué zona suele ir a locales donde sirvan comida?**

Capital Federal

Gran Buenos Aires

Otro:

**¿Considera que el uso de menús electrónicos táctiles es beneficioso?**

Es beneficioso

No es necesario

**¿Preferiría el menú portátil o fijo a la mesa?**

Portátil

Fijo

**¿Prefiere una interfaz en la que predomine texto o una en la que predominen las imágenes?**

Buena relación entre texto e imágenes

Principalmente texto

Principalmente imágenes

**¿Utilizaría una cuenta personalizada con sus datos?**

Por ejemplo: cuenta de correo electrónico, fecha de cumpleaños, comida favorita y comidas más pedidas.

- Sí, sería útil para locales donde soy cliente regular
- No, no deseo brindar información personal

**Si respondió que sí a la pregunta anterior, ¿estaría interesado en recibir via mail actualizaciones del menú u otras promociones del local?**

- Sí
- No

**¿Le molestaría recibir publicidad mientras pide comida?**

Así como en menús impresos, este menú podría incluir banners de las marcas ofrecidas en el local, ocupando un espacio reducido de la imagen.

- No me molestaría si es publicidad discreta, como en los menús impresos
- No me molestaría en lo absoluto
- Me molestaría la existencia de publicidad en el menú

**¿Qué volumen y peso espera?**

Indique si igual, menor o mayor al de un menú impreso convencional.

- Mayor peso y volumen
- Igual peso y volumen
- Menor peso y volumen

**¿Prefiere controlar la pantalla mediante el dedo o un dispositivo en forma de lápiz?**

- Mediante el dedo
- Lápiz

**¿Le gustaría recibir el tiempo estimativo de la demora de su comida?**

- Sería una buena incorporación
- Me es indiferente

**Si al final de la atención pudiera puntuar características del servicio, ¿qué opciones le gustaría que estuvieran presentes?**

- Calidad de la comida
- Servicio (mozo)
- Higiene
- Otro:

**A la hora de escoger la comida, ¿le interesaría ver cómo puntuaron otros clientes cada plato?**

- Sí, y sería muy influyente en mi elección
- Sí, pero no influiría mucho en mi elección
- No me interesaría

Página 2 para dueños



Menú táctil para locales de comida

Encuesta para dueño de local de comidas

**¿En qué zona se ubica su local?**

Capital Federal

Gran Buenos Aires

Otro:

**¿Considera que el uso de menús electrónicos táctiles es beneficioso?**

Es beneficioso

No es necesario

**¿Qué opina sobre publicidad en el menú electrónico?**

Necesaria

Útil

Otro:

**¿Qué volumen y peso espera?**

Indique si igual, menor o mayor al de un menú impreso convencional.

- Mayor peso y volumen
- Igual peso y volumen
- Menor peso y volumen

**¿Cuánto tiempo de funcionamiento continuo mínimo precisa en condiciones normales de uso?**

Elija una estimación del mismo. Tenga en cuenta que es una cota inferior, y que podrá recargar cada equipo mientras continúa usando los otros.

1 2 3 4 5 6 7 8  
horas         horas

**¿Cuánto estaría dispuesto a pagar por el producto?**

Elija una cota máxima.

- Hasta u\$s 200 por unidad
- Hasta u\$s 300 por unidad
- Hasta u\$s 400 por unidad
- Hasta u\$s 500 por unidad
- Otro:

**¿Preferiría el menú portátil o fijo a la mesa?**

- Portátil
- Fijo

**¿Le interesaría la estadística de ventas?**

- Sí, sería una buena fuente de información
- No sería necesario

**¿Qué alcance debería tener el dispositivo?**

Indique el alcance mínimo que debería tener la comunicación inalámbrica entre el dispositivo y el servidor (coordinador de todos los equipos)

- 20 m
- 30 m
- 40 m
- 50 m

**¿Le interesaría ofrecer el tiempo estimativo de la demora de la comida?**

- Sí
- No

**¿Le interesaría que los usuarios puntúen la comida?**

- Sí, sería una muy buena fuente de información
- No

**¿Le interesaría que los usuarios puntúen el servicio?**

Seleccione si le interesaría que los usuarios pudieran puntuar la at

- Sí, sería una muy buena fuente de información
- No

[« Atrás](#) [Continuar »](#)

Con la tecnología de [Google Docs](#)

[Informar sobre abusos](#) - [Condiciones del servicio](#) - [Otros términos](#)

Despedida y agradecimientos



Como puede verse, se agregaron a las encuestas originales elementos que permitirán asegurar un control sobre la heterogeneidad del grupo muestra, además de permitir enfocar mejor la entrada a través de la cual se quiera ingresar al mercado.

*2. ANEXO II: LOG DE INICIALIZACIÓN DEL KERNEL DE LINUX*

Start AT91Bootstrap...

U-Boot 2010.12 (Jun 08 2011 - 08:54:13)

DRAM: 128 MiB

NAND: 256 MiB

In: serial

Out: serial

Err: serial

Net: No ethernet found.

Hit any key to stop autoboot: 0

NAND read: device 0 offset 0xa0000, size 0x360000

3538944 bytes read: OK

## Booting kernel from Legacy Image at 72000000 ...

Image Name: Linux-2.6.30\_som9g45-1.4.1  
Image Type: ARM Linux Kernel Image (uncompressed)  
Data Size: 2065196 Bytes = 2 MiB  
Load Address: 70008000  
Entry Point: 70008000  
Verifying Checksum ... OK  
Loading Kernel Image ... OK

OK

Starting kernel ...

Uncompressing Linux.....  
done, booting the kernel.

Linux version 2.6.30\_som9g45-1.4.1 (pdkuser@pdkhost) (gcc version 4.4.1 (Sourcery G++ Lite 2010q1-202) ) #3 Fri Jun 7 16:55:17 ART 2013

CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00053177

CPU: VIVT data cache, VIVT instruction cache

Machine: Apexar's APX\_SOM9G45\_EK

Memory policy: ECC disabled, Data cache writeback

Clocks: CPU 400 MHz, master 133 MHz, main 12.000 MHz

Built 1 zonelists in Zone order, mobility grouping on. Total pages: 32512

Kernel command line: console=ttyS0,115200

mtddparts=atmel\_nand:128k(bootstrap)ro,256k(uboot)ro,128k(env1)ro,128k(env2)ro,3456k(linux)ro,-  
(root) root=/dev/mtdblock5 rw rootfstype=jffs2 rootwait

NR\_IRQS:192

AT91: 160 gpio irqs in 5 banks

PID hash table entries: 512 (order: 9, 2048 bytes)

Console: colour dummy device 80x30

console [ttyS0] enabled

Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)

Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)

Memory: 128MB = 128MB total

Memory: 125552KB available (3624K code, 287K data, 168K init, 0K highmem)

Calibrating delay loop... 199.47 BogoMIPS (lpj=997376)

Mount-cache hash table entries: 512

CPU: Testing write buffer coherency: ok

net\_namespace: 296 bytes

NET: Registered protocol family 16

atmel\_tcb: probe of atmel\_tcb.0 failed with error -22

bio: create slab <bio-0> at 0

SCSI subsystem initialized

usbcore: registered new interface driver usbfs

usbcore: registered new interface driver hub

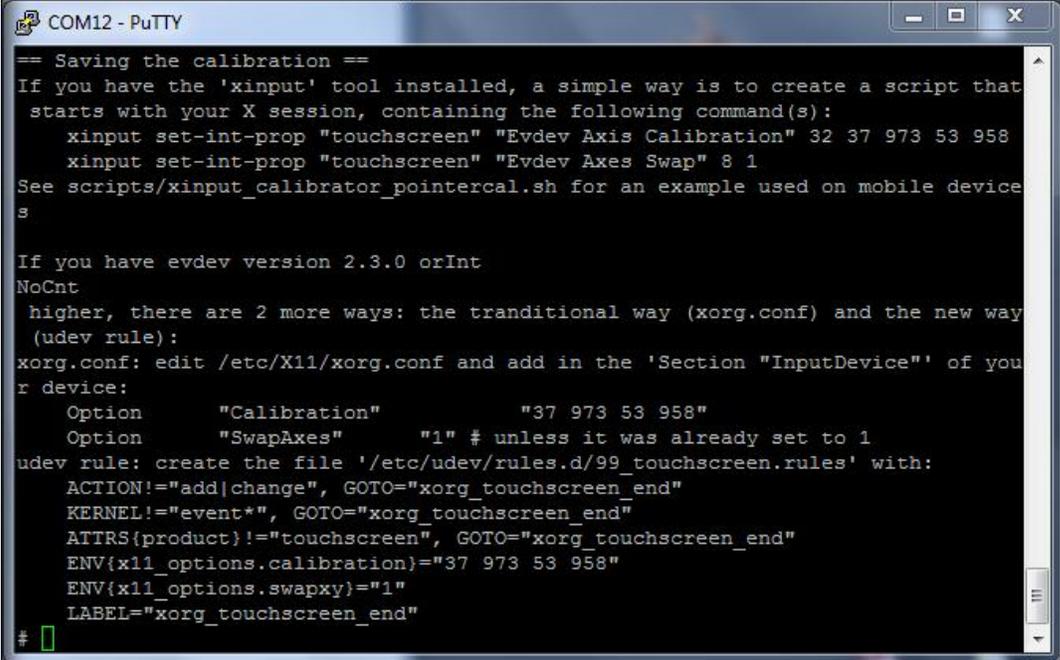
```
usbcore: registered new device driver usb
cfg80211: Calling CRDA to update world regulatory domain
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
TCP reno registered
NET: Registered protocol family 1
NetWinder Floating Point Emulator V0.97 (double precision)
JFFS2 version 2.2. (NAND) (SUMMARY) © 2001-2006 Red Hat, Inc.
msgmni has been set to 245
alg: No test for stdrng (krng)
io scheduler noop registered
io scheduler anticipatory registered (default)
atmel_lcdfb atmel_lcdfb.0: 1125KiB frame buffer at 77a00000 (mapped at ffa00000)
Console: switching to colour frame buffer device 100x30
atmel_lcdfb atmel_lcdfb.0: fb0: Atmel LCDC at 0x00500000 (mapped at c8858000), irq 23
atmel_usart.0: ttyS0 at MMIO 0xfeff0000 (irq = 1) is a ATMEL_SERIAL
atmel_usart.1: ttyS1 at MMIO 0xff90000 (irq = 8) is a ATMEL_SERIAL
atmel_usart.2: ttyS2 at MMIO 0xff94000 (irq = 9) is a ATMEL_SERIAL
brd: module loaded
loop: module loaded
Driver 'sd' needs updating - please use bus_type methods
PPP generic driver version 2.4.2
macb macb: invalid hw address, using random
MACB_mii_bus: probed
eth0: Atmel MACB at 0xffbc000 irq 25 (22:1a:0a:bd:27:48)
eth0: attached PHY driver [Generic PHY] (mii_bus:phy_addr=ffffff:01, irq=-1)
Broadcom 43xx driver loaded [ Features: , Firmware-ID: FW13 ]
usbcore: registered new interface driver zd1211rw
usbcore: registered new interface driver rtl8187
usbcore: registered new interface driver zd1201
usbcore: registered new interface driver rt2500usb
usbcore: registered new interface driver rt73usb
usbcore: registered new interface driver ar9170usb
NAND device: Manufacturer ID: 0x2c, Chip ID: 0xba (Micron NAND 256MiB 1,8V 16-bit)
AT91 NAND: 16-bit, Software ECC
Scanning device for bad blocks
6 cmdlinepart partitions found on MTD device atmel_nand
Creating 6 MTD partitions on "atmel_nand":
0x000000000000-0x000000020000 : "bootstrap"
0x000000020000-0x000000060000 : "uboot"
0x000000060000-0x000000080000 : "env1"
```

```
0x000000080000-0x0000000a0000 : "env2"
0x0000000a0000-0x000000400000 : "linux"
0x000000400000-0x000010000000 : "root"
atmel_spi atmel_spi.0: Atmel SPI Controller at 0xffffa4000 (irq 14)
ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
atmel-ehci atmel-ehci: Atmel EHCI UHP HS
atmel-ehci atmel-ehci: new USB bus registered, assigned bus number 1
atmel-ehci atmel-ehci: irq 22, io mem 0x00800000
atmel-ehci atmel-ehci: USB 2.0 started, EHCI 1.00
usb usb1: New USB device found, idVendor=1d6b, idProduct=0002
usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
usb usb1: Product: Atmel EHCI UHP HS
usb usb1: Manufacturer: Linux 2.6.30_som9g45-1.4.1 ehci_hcd
usb usb1: SerialNumber: atmel-ehci
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
at91_ohci at91_ohci: AT91 OHCI
at91_ohci at91_ohci: new USB bus registered, assigned bus number 2
at91_ohci at91_ohci: irq 22, io mem 0x00700000
usb usb2: New USB device found, idVendor=1d6b, idProduct=0001
usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
usb usb2: Product: AT91 OHCI
usb usb2: Manufacturer: Linux 2.6.30_som9g45-1.4.1 ohci_hcd
usb usb2: SerialNumber: at91
usb usb2: configuration #1 chosen from 1 choice
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 2 ports detected
usbcore: registered new interface driver cdc_acm
cdc_acm: v0.26:USB Abstract Control Model driver for USB modems and ISDN adapters
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
usbcore: registered new interface driver usbserial
usbserial: USB Serial Driver core
USB Serial support registered for FTDI USB Serial Device
usbcore: registered new interface driver ftdi_sio
ftdi_sio: v1.4.3:USB FTDI Serial Converters Driver
atmel_usba_udev atmel_usba_udev: MMIO registers at 0xffff78000 mapped at c885c000
atmel_usba_udev atmel_usba_udev: FIFO at 0x00600000 mapped at c8900000
g_ether gadget: using random self ethernet address
g_ether gadget: using random host ethernet address
usb0: MAC 82:38:2e:80:c7:c1
```

```
usb0: HOST MAC b2:81:01:ee:c4:5d
g_ether gadget: Ethernet Gadget, version: Memorial Day 2008
g_ether gadget: g_ether ready
mice: PS/2 mouse device common for all mice
input: gpio-keys as /class/input/input0
atmel_tsadcc atmel_tsadcc: Master clock is set at: 133333333 Hz
atmel_tsadcc atmel_tsadcc: Prescaler is set at: 82
input: atmel touch screen controller as /class/input/input1
rtc-at91sam9 at91_rtt.0: rtc core: registered at91_rtt as rtc0
IRQ 1/rtc0: IRQF_DISABLED is not guaranteed on shared IRQs
rtc-at91sam9 at91_rtt.0: rtc0: SET TIME!
i2c /dev entries driver
i2c-gpio i2c-gpio.0: using pins 52 (SDA) and 53 (SCL)
Linux video capture interface: v2.00
gspca: main v2.5.0 registered
usbcore: registered new interface driver sonixb
sonixb: registered
usbcore: registered new interface driver sonixj
sonixj: registered
usbcore: registered new interface driver uvcvideo
USB Video Class driver (v0.1.0)
Registered led device: led3
Registered led device: led1
Registered led device: led2
Registered led device: led4
at_hdmac at_hdmac: Atmel AHB DMA Controller ( cpy slave ), 8 channels
usbcore: registered new interface driver usbhid
usbhid: v2.6:USB HID core driver
Advanced Linux Sound Architecture Driver Version 1.0.20.
AC'97 0 does not respond - RESET
AC'97 0 access is not valid [0xffffffff], removing mixer.
atmel_ac97c: probe of atmel_ac97c.0 failed with error -5
ALSA device list:
  No soundcards found.
TCP cubic registered
NET: Registered protocol family 17
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
lib80211: common routines for IEEE802.11 drivers
rtc-at91sam9 at91_rtt.0: hctosys: unable to read the hardware clock
atmel_mci atmel_mci.0: Using dma0chan0 for DMA transfers
atmel_mci atmel_mci.0: Atmel MCI controller at 0xff80000 irq 11, 1 slots
Empty flash at 0x05ce666c ends at 0x05ce6800
Empty flash at 0x05cf209c ends at 0x05cf2800
```

```
jffs2_scan_inode_node(): CRC failed on node at 0x05cf9fd4: Read 0xffffffff, calculated 0xbe466025
Empty flash at 0x05cfa43c ends at 0x05cfa800
Empty flash at 0x06862064 ends at 0x06862800
Empty flash at 0x08a3dafc ends at 0x08a3e000
Empty flash at 0x0905ab94 ends at 0x0905b000
VFS: Mounted root (jffs2 filesystem) on device 31:5.
Freeing init memory: 168K
Initializing random number generator... done.
Starting network...
Starting sshd: OK
Starting X server: OK
Skippnig touchscreen calibration.
```

### 3. ANEXO III: CALIBRACIÓN DE LA MEMBRANA TÁCTIL RESISTIVA



```
COM12 - PuTTY
== Saving the calibration ==
If you have the 'xinput' tool installed, a simple way is to create a script that
starts with your X session, containing the following command(s):
    xinput set-int-prop "touchscreen" "Evdev Axis Calibration" 32 37 973 53 958
    xinput set-int-prop "touchscreen" "Evdev Axes Swap" 8 1
See scripts/xinput_calibrator_pointercal.sh for an example used on mobile device
s

If you have evdev version 2.3.0 orInt
NoCnt
higher, there are 2 more ways: the traditional way (xorg.conf) and the new way
(udev rule):
xorg.conf: edit /etc/X11/xorg.conf and add in the 'Section "InputDevice"' of you
r device:
    Option      "Calibration"      "37 973 53 958"
    Option      "SwapAxes"         "1" # unless it was already set to 1
udev rule: create the file '/etc/udev/rules.d/99_touchscreen.rules' with:
ACTION!="add|change", GOTO="xorg_touchscreen_end"
KERNEL!="event*", GOTO="xorg_touchscreen_end"
ATTRS{product}!="touchscreen", GOTO="xorg_touchscreen_end"
ENV{x11_options.calibration}="37 973 53 958"
ENV{x11_options.swapxy}="1"
LABEL="xorg_touchscreen_end"
#
```

4. ANEXO IV: DETALLE DEL CÁLCULO DE CONFIABILIDAD CON  $G_F$ :

Módulo	Componente	Valor	Cantidad	Factores de carga														$\lambda_p$ [10 <sup>6</sup> hs] (Por unidad)	MTBF[10 <sup>6</sup> hs] (Por unidad)	$\lambda_p$ [10 <sup>6</sup> hs] (Total)		
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$	$\pi_E$					
Alimentación	Polyswitch	1.1A	1	0.01000													2	0.020	50.00	0.020		
Alimentación	Varistor		1	0.00200	1.0												8	6	0.096	10.42	0.096	
Alimentación	Capacitor	100nF	2	0.00099	1.6			0.81	1	1							10	10	0.128	7.79	0.257	
Alimentación	Capacitor	22uF	1	0.00012	2.9			1.90	1	1							10	10	0.066	15.12	0.066	
Alimentación	Capacitor	10nF	1	0.00099	1.6			0.66	1	1							10	10	0.105	9.57	0.105	
Alimentación	Inductor	47uH	1	0.00003	1.8												3	6	0.001	1028.81	0.001	
Alimentación	Capacitor	150uF	1	0.00012	2.9			3.40	2	1							10	10	0.237	4.23	0.237	
$\lambda_t$ (Total del módulo)																						0.781

Módulo	Componente	Valor	Cantidad	Factores de carga														$\lambda_p$ [10 <sup>6</sup> hs] (Por unidad)	MTBF[10 <sup>6</sup> hs] (Por unidad)	$\lambda_p$ [10 <sup>6</sup> hs] (Total)			
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$	$\pi_E$						
Boost	TPS61165		1		3.4							1					0.025	0.020	10	2	1.250	0.80	1.250
Boost	Capacitor	4,7uF	1	0.00012	2.9			1.30	1	1									10	10	0.045	22.10	0.045
Boost	Inductor	22uH	1	0.00003	1.8														3	6	0.001	1028.81	0.001
Boost	Diodo	40V@0,5A	1	0.00200	1.0		1.00	1.00											8	6	0.096	10.42	0.096
Boost	Capacitor	1uF	1	0.00099	1.6			1.00	1	1									10	10	0.158	6.31	0.158
Boost	Capacitor	220nF	1	0.00099	1.6			0.81	1	1									10	10	0.128	7.79	0.128
Boost	Resistencia	1,2 $\Omega$	1	0.00370	1.3	0.068	0.66												10	4	0.009	115.81	0.009
Boost	Resistencia	10k $\Omega$	1	0.00370	1.3	0.068	0.66												10	4	0.009	115.81	0.009
$\lambda_t$ (Total del módulo)																						1.696	

Módulo	Componente	Valor	Cantidad	Factores de carga														$\lambda_p$ [/ $10^6$ hs] (Por unidad)	MTBF[ $10^6$ hs] (Por unidad)	$\lambda_p$ [/ $10^6$ hs] (Total)
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$	$\pi_E$			
Bateria	BQ24070		1		3.4						1			0.025	0.020	10	2	1.250	0.80	1.250
Bateria	Resistencia	47k $\Omega$	1	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.009
Bateria	Capacitor	10uF	2	0.00012	2.9			1.60	1	1						10	10	0.056	17.96	0.111
Bateria	Resistencia	2,2k $\Omega$	3	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.026
Bateria	Resistencia	3,3k $\Omega$	1	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.009
Bateria	Resistencia	33k $\Omega$	1	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.009
Bateria	Capacitor	47uF	2	0.00012	2.9			2.30	1	1						10	10	0.080	12.49	0.160
Bateria	Switch		1	0.04000				1.00			1						3	0.120	8.33	0.120
Bateria	Diodo		2	0.00200	1.0		1.00	1.00								8	6	0.096	10.42	0.192
Bateria	Resistencia	100k $\Omega$	2	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.017
Bateria	MOS		3	0.00074	2.8		0.11					0.7				8	6	0.008	130.58	0.023
Bateria	Resistencia	10k $\Omega$	3	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.026
Bateria	BQ27500		1		0.3						1			0.025	0.020	10	2	0.475	2.11	0.475
Bateria	Resistencia	1,8M $\Omega$	1	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.009
Bateria	Capacitor	33nF	1	0.00099	1.6			0.76	1	1						10	10	0.120	8.31	0.120
Bateria	Resistencia	18,2k $\Omega$	1	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.009
Bateria	Resistencia	1k $\Omega$	1	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.009
Bateria	Capacitor	100nF	10	0.00099	1.6			0.81	1	1						10	10	0.128	7.79	1.283
Bateria	Resistencia	0,01 $\Omega$	1	0.00370	1.3	0.440	1.20									10	4	0.102	9.84	0.102
Bateria	Resistencia	100 $\Omega$	3	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.026
Bateria	Capacitor	0,47uF	1	0.00099	1.6			0.94	1	1						10	10	0.149	6.72	0.149
Bateria	TPS71525		1		0.3						1			0.025	0.010	10	2	0.275	3.64	0.275
Bateria	TPS73733		1		2.4						1			0.025	0.020	10	2	1.000	1.00	1.000

Módulo	Componente	Valor	Cantidad	Factores de carga													$\lambda_p$ [ $10^6$ hs] (Por unidad)	MTBF[ $10^6$ hs] (Por unidad)	$\lambda_p$ [ $10^6$ hs] (Total)		
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$				$\pi_E$	
Bateria	Capacitor	15pF	1	0.00099	1.6			0.44	1	1						10	10	0.070	14.35	0.070	
Bateria	Resistencia	5,6M $\Omega$	1	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.009	
Bateria	Transistor		1	0.00074	2.8		0.11						0.7			8	6	0.008	130.58	0.008	
Bateria	Led		3	0.00023	1.0											6	2	0.003	395.26	0.008	
Bateria	Resistencia	470 $\Omega$	2	0.00370	1.3	0.170	0.82									10	4	0.027	37.28	0.054	
Bateria	Resistencia	330 $\Omega$	1	0.00370	1.3	0.170	0.82									10	4	0.027	37.28	0.027	
$\lambda_t$ (Total del módulo)																					5.581

Módulo	Componente	Valor	Cantidad	Factores de carga													$\lambda_p$ [ $10^6$ hs] (Por unidad)	MTBF[ $10^6$ hs] (Por unidad)	$\lambda_p$ [ $10^6$ hs] (Total)		
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$				$\pi_E$	
Comunicación	Capacitor	1uF	2	0.00099	1.6			1.00	1	1						10	10	0.158	6.31	0.317	
Comunicación	Transistor		4	0.00074	2.8		0.11					0.7				8	6	0.008	130.58	0.031	
Comunicación	Capacitor	10uF	1	0.00012	2.9			1.60	1	1						10	10	0.056	17.96	0.056	
Comunicación	Capacitor	100nF	1	0.00099	1.6			0.81	1	1						10	10	0.128	7.79	0.128	
Comunicación	Led		3	0.00023	1.0											6	2	0.003	395.26	0.008	
Comunicación	Resistencia	470 $\Omega$	2	0.00370	1.3	0.170	0.82									10	4	0.027	37.28	0.054	
Comunicación	Resistencia	330 $\Omega$	1	0.00370	1.3	0.170	0.82									10	4	0.027	37.28	0.027	
Otros	Pines		10	0.00100	1.5								1			2	1	0.003	333.33	0.030	
$\lambda_t$ (Total del módulo)																					0.649

Módulo	Componente	Valor	Cantidad	Factores de carga													$\lambda_p$ [10 <sup>6</sup> hs] (Por unidad)	MTBF[10 <sup>6</sup> hs] (Por unidad)	$\lambda_p$ [10 <sup>6</sup> hs] (Total)		
				$\lambda_b$	$\pi_T$	$\pi_P$	$\pi_S$	$\pi_C$	$\pi_V$	$\pi_{SR}$	$\pi_L$	$\pi_A$	$\pi_K$	C1	C2	$\pi_Q$				$\pi_E$	
Pantalla	Conector 40POS		1	0.00100	1.5								1			2	1	0.003	333.33	0.003	
Pantalla	Resistencia	220k $\Omega$	1	0.00370	1.3	0.068	0.66									10	4	0.009	115.81	0.009	
Pantalla	Conector TS 4POS		1	0.00100	1.5								1			2	1	0.003	333.33	0.003	
Pantalla	Conector SODIMM		1	0.00100	1.5								1			2	1	0.003	333.33	0.003	
Pantalla	Capacitor	100nF	9	0.00099	1.6			0.81	1	1						10	10	0.128	7.79	1.155	
Microprocesador	SOM9G45		1		0.3								1		0.160	0.025	10	2	0.980	1.02	0.980
$\lambda_t$ (Total del módulo)																					2.152

RESULTADOS DE LA CONFIABILIDAD CON  $G_F$

$\lambda_p$ del sistema [10 <sup>6</sup> hs]	10.860
MTBF del sistema [10 <sup>6</sup> hs]	0.092
MTBF del sistema [años]	10.511

#### XIV. BIBLIOGRAFIA

##### 1. Hoja de datos utilizados

- LM2676, “Simple Switcher High Efficiency 3A Step-Down Voltage Regulator”, Texas Instrument, SNVS031H, 2008.
- BQ24070, “Single-Chip Li-Ion Charge and System Power-Path Management IC”, Texas Instrument, SLUS694F, 2009.
- BQ27500-V130, “System-Side Impedance Track Fuel Gauge”, Texas Instrument, SLUS914A, 2009.
- TPS715XX, “50mA, 24V, 3.2uA Supply Current Low-Dropout Linear Regulator in SC70 Package”, Texas Instrument, SLVS338P, 2008.
- TPS737XX, “1A Low-Dropout Regulator with Reverse Current Protection”, Texas Instrument, SBVS067O, 2012.
- TPS61165, “High Brightness White LED Driver in 2mm x 2mm QFN and SOT-23 Packages”, Texas Instrument, SLVS790B, 2011.
- SAM9G45, “AT91SAM ARM-based Embedded MPU”, Atmel, 6438G-ATARM, 2011.

##### 2. Libros y publicaciones

- Ritchie, Dennis. Kernighan, Brian. “The C Programming Language”, Prentice Hall, 1988.
- Mohan, Ned. “Power Electronics”, Wiley, 2007.
- Sedra, Smith. “Microelectronics Circuit 5<sup>th</sup> edition”, Oxford University Press, 2003.
- Jingrong, Qian. “Improving battery safety, charging, and fuel gauging in portable media applications”, Texas Instrument Incorporated, 2009.

##### 3. Notas de aplicación y guías de usuarios

- Nwachukwu, Ifeanyi. “LM267X 3A, 5A Evaluation Boards”, Application Note National Semiconductor, 1999.
- Maniktala, Sanjaya. “Simple Switcher PCB Layout Guidelines”, Application Report Texas Instrument, SNVA054B, 2002.
- Ye, Mao. “Using the BQ24075 Charger and BQ27500/1 Fuel Gauge”, Application Report Texas Instrument, SLUA490, 2008.
- Mauney, Charles. “Power-Path Li-Ion Charger Selection”, Application Report Texas Instrument, SLUU332, 2008.
- Herder, Charles. “Win CE/Linux Drivers for BQ275XX Fuel Gauge”, Application Report Texas Instrument, SLUA543, 2011.
- Fundaro, Peter. Jackson, Bill. Vega, Michael. “Key Design Considerations for the BQ27500 and BQ 27501”, Application Report Texas Instrument, SLUA439, 2007.
- Mueller, Jeff. Yu, Ming. “Using I2C Communication with the BQ275XX Series of Fuel Gauges”, Application Report Texas Instrument, SLUA467A, 2009.
- Yu, Ming. “Quick Start Guide for BQ2750X Family Gas Gauge”, Application Report Texas Instrument, SLUA448, 2008.
- “BQ 24070/1 1.5-A Single-Chip Li-Ion and Li-Pol Charge Management IC EVM”, User’s Guide Texas Instrument, SLUU248A, 2006
- “BQ2750X EVM System Side Single-Cell Impedance Track Technology Evaluation Module”, User’s Guide Texas Instrument, SLUU287, 2007.

- Vega, Michael. Yu, Ming. “Single Cell Impedance Track Printed-Circuit Board Layout Guide”, Application Report Texas Instrument, SLUA457, 2007.
- Vega, Michael. Yu, Ming. “Single Cell Gas Gauge Circuit Design”, Application Report Texas Instrument, SLUA456, 2008.
- Hopf, Ben. “Choosing an Appropriate Pull-up/Pull-down Resistor for Open Drain Outputs”, Application Report Texas Instrument, SLUA485, 2011.
- “TPS61165EVM-283”, User’s Guide Texas Instrument, SLVU224C, 2010.
- “AT91SAM9G45-EKES”, User’s Guide Atmel, 6481B-ATARM, 2009.
- Fons, A. Diego. Tagarelli, Lucas. “KIT de Evaluación APX-SOM9G45-EK”, Apexar Technologies S.A., 2010.
- “Módulo SOM APX-SOM9G45 Manual de Usuario Rev.01”, Apexar Technologies S.A..
- “Apexar SOM System on Module”, User’s Manual Apexar Technologies S.A., 2011.

# Módulo SOM

## APX-SOM9G45

### Manual de Usuario

Rev. 01



Estanislao Diaz 193

(B1642EBC) San Isidro – Buenos Aires

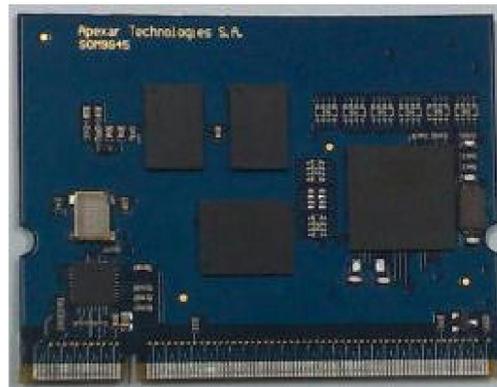
Tel/Fax: +5411-4747-1616

<http://www.apexar.com>



**APX-SOM9G45**

1 Introducción.....	3
1.1 Beneficios de uso de APX-SOM9G45 .....	3
2 Descripción del hardware del APX-SOM9G45 .....	4
2.1 Layout .....	4
2.2 Mecánica .....	5
2.3 Arquitectura .....	6
3 Pinout del conector SODIMM200 en APX-SOM9G45 .....	6





APX-SOM9G45

## 1 Introducción

Los fabricantes de microcontroladores siguen desarrollando componentes más pequeños, rápidos y complejos que permiten alcanzar performance antes impensadas y a costos accesibles. Sin embargo, a los fabricantes y desarrolladores de equipos electrónicos llevar estos avances a sus equipos implica una serie de problemas: Tamaño de componentes contra la posibilidad de ensamblarlos, la densidad del circuito impreso contra su confiabilidad y la complejidad frente al tiempo de salida al mercado. Por estas razones la nueva tecnología queda fuera de su alcance.

La utilización de el módulo APX-SOM9G45 minimiza el trabajo de diseño de una placa propietaria para una determinada aplicación que necesite utilizar las capacidades de la nueva familia de procesadores (eMPU) de **ATMEL AT91SAM9G45 y SAM9M10**. El desarrollo de una placa soporte no requiere trabajar sobre el procesador, las memorias ni las I/O estándares y el desarrollo se concentra en las entradas/salidas y aspectos dimensionales requeridos para la aplicación. Esto optimiza enormemente los tiempos de salida al mercado. Y se evita lidiar con la complejidad de diseño de hardware de alta densidad y velocidad.

Apexar Technologies ha desarrollado un kit soporte del APX-SOM9G45 para facilitar el desarrollo y prueba de las capacidades del módulo. También se proveen los esquemáticos utilizados en el armado del kit como soporte de desarrollo de la placa base para la aplicación de nuestros clientes. También se provee el servicio de desarrollo de la placa base customizada en caso de ser necesario.

### 1.1 Beneficios de uso de APX-SOM9G45

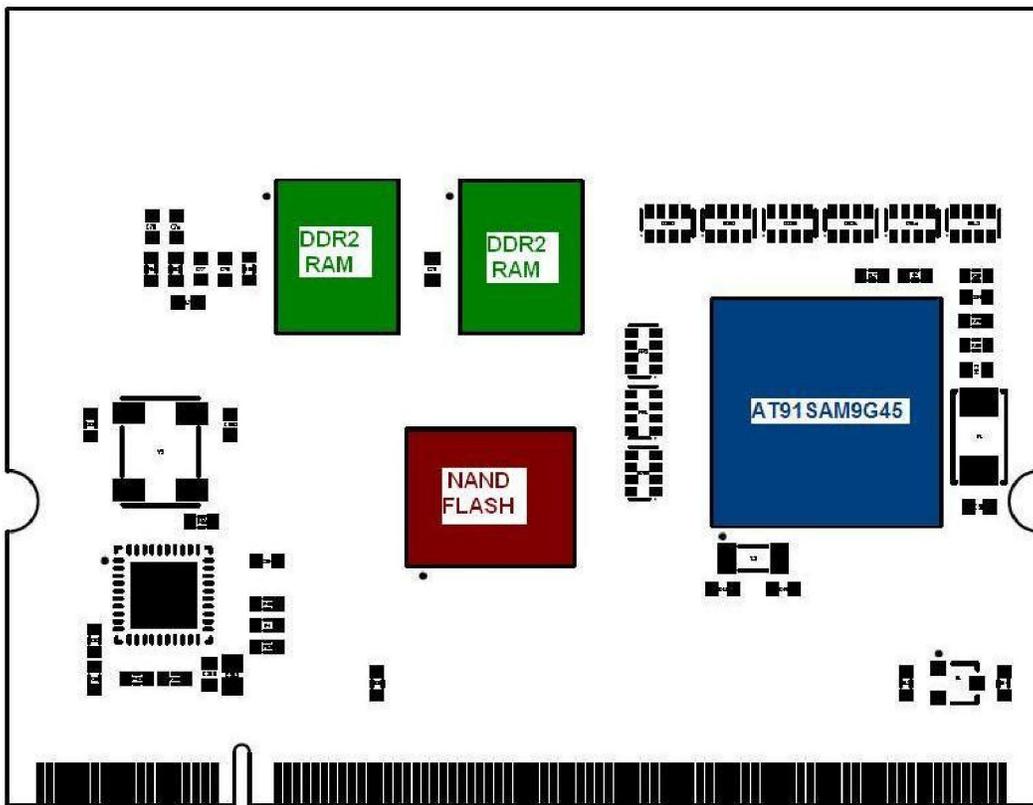
- ✘ Mayor velocidad de salida al mercado
- ✘ Disminución de costos
- ✘ Reducción de riesgos
- ✘ Escalabilidad de elección de CPU
- ✘ Disminución de requerimientos de diseño
- ✘ Pequeño footprint



APX-  
SOM9G45

## 2 Descripción del hardware del APX-SOM9G45

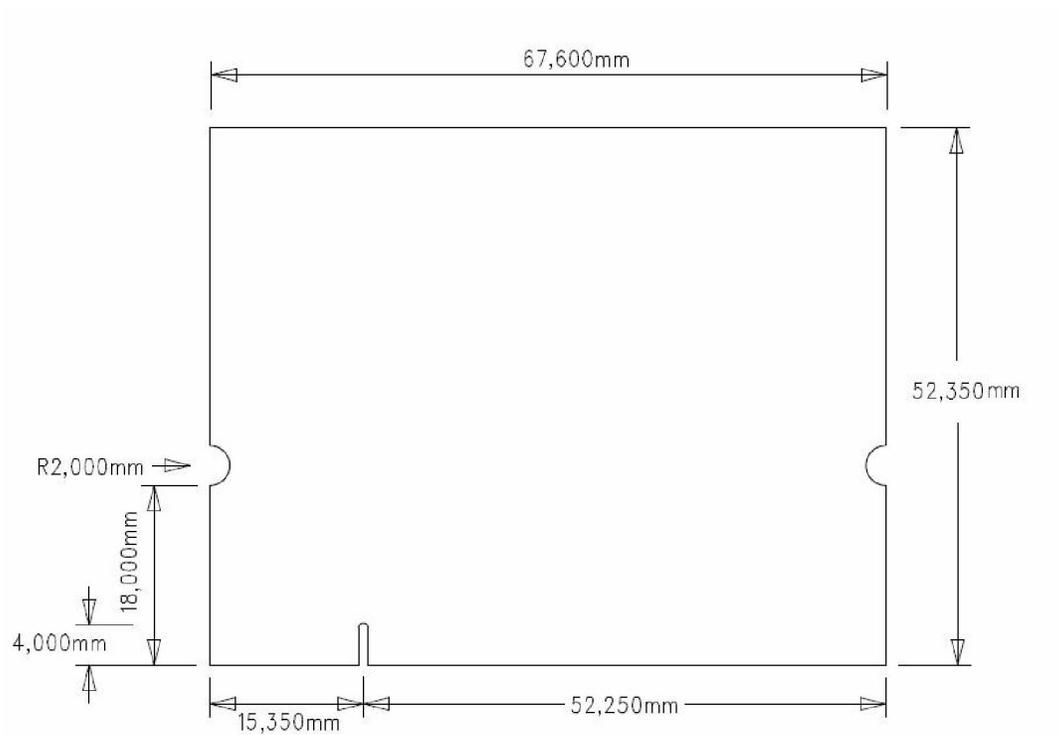
### 2.1 Layout



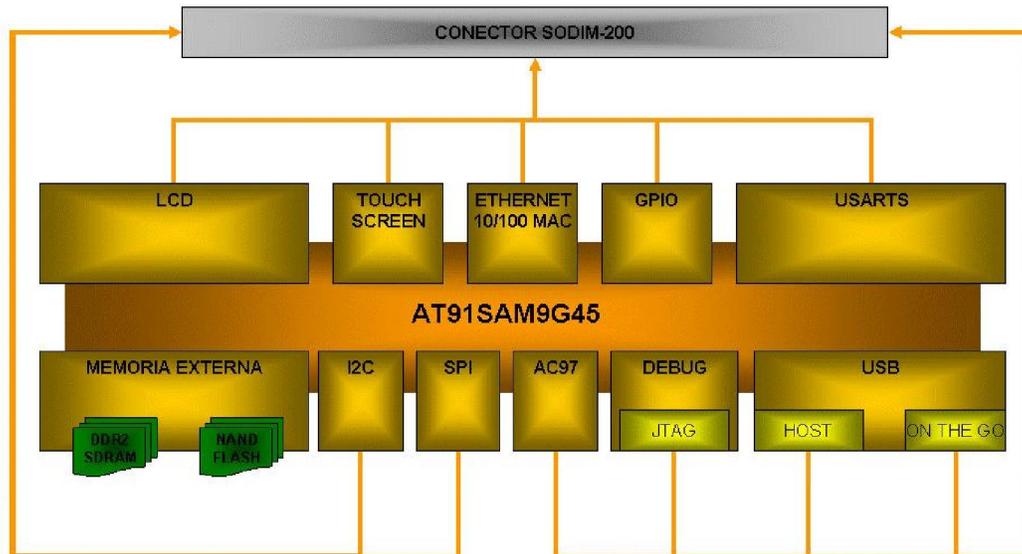


**APX-  
SOM9G45**

## 2.2 Mecánica



## 2.3 Arquitectura



El módulo utiliza el microprocesador AT91SAM9G45 con 128-Mbytes de memoria RAM DDR2 y 256-Mbytes de memoria NAND flash. Posee salidas al conector SODIM-200 de LCD, Touchscreen, GPIO, 4 salidas seriales, 2 salidas USB, JTAG, I2C, SPI y AC97 (audio DAC).

## 3 Pinout del conector SODIMM200 en APX-SOM9G45

Pines conector SODIMM	Descripción	Observaciones
1	GND_ETH	
2	TX_N	
3	+AVDD3.3	
4	TX_P	
5	GND_POWER	
6	RX_N	
7	Not Connected	
8	RX_P	
9	Not Connected	
10	ETH_LED3	NOT USED


**APX-  
SOM9G45**

11	ETH_LED1	
12	ETH_LED2	NOT USED
13	GND_POWER	
14	Not Connected	
15	Not Connected	
16	Not Connected	
17	Not Connected	
18	Not Connected	
19	Not Connected	
20	+3,3 VCC	
21	Not Connected	
22	Not Connected	
23	Not Connected	
24	Not Connected	
25	Not Connected	
26	Not Connected	
27	GND_POWER	
28	Not Connected	
29	Not Connected	
30	Not Connected	
31	Not Connected	
32	+3,3 VCC	
33	Not Connected	
34	Not Connected	
35	Not Connected	
36	Not Connected	
37	Not Connected	
38	Not Connected	
39	Not Connected	
40	GND_POWER	
41	Not Connected	
42	Not Connected	
43	Not Connected	
44	Not Connected	
45	Not Connected	
46	+3,3 VCC	
47	Not Connected	
48	Not Connected	
49	Not Connected	
50	Not Connected	
51	GND_POWER	
52	Not Connected	
53	Not Connected	
54	Not Connected	
55	Not Connected	
56	Not Connected	
57	Not Connected	
58	Not Connected	
59	Not Connected	
60	+3,3 VCC	
61	BMS	
62	Not Connected	
63	Not Connected	
64	Not Connected	
65	GND_POWER	
66	Not Connected	
67	Not Connected	
68	Not Connected	


**APX-  
SOM9G45**

69	Not Connected	
70	Not Connected	
71	Not Connected	
72	+3,3 VCC	
73	Not Connected	
74	Not Connected	
75	Not Connected	
76	Not Connected	
77	Not Connected	
78	Not Connected	
79	GND_POWER	
80	Not Connected	
81	Not Connected	
82	Not Connected	
83	PD31	TIOB1 / PWM1
84	PD30	TIOB0 / SCK2
85	PD29	TCLK1 / SCK1
86	PD28	TSADTRG / SPI1_NPCS1
87	Not Connected	
88	+3,3 VCC	
89	PD27	PCK1 / SPI0_NPCS3
90	PD26	PCK0 / PWM2
91	Not Connected	
92	PC31	
93	PB6	TXD2
94	PD23	TCLK0
95	GND_POWER	
96	PD22	TIOA2
97	PD21	TIOA1
98	PB19	TXD0
99	PB18	RXD0
100	Not Connected	
101	PD19	SPI1_NPCS3 / FIQ
102	PB4	TXD1
103	PB5	RXD1
104	Not Connected	
105	PB7	RXD2
106	+3,3 VCC	
107	PB0	SPI0_MISO
108	PB1	SPI0_MOSI
109	PB2	SPI0_SPCK
110	PB3	SPI0_NPCS0
111	PD17	CTS1
112	PD20	TIOA0
113	GND_POWER	
114	PD16	RTS1
115	PA20	TWD0
116	PA21	TWCK0
117	PB12	DRXD
118	PB13	DTXD
119	PD15	RF1
120	PD14	TF1
121	PD13	RK1
122	PD12	TK1
123	PD11	RD1
124	+3,3 VCC	
125	PD10	TD1
126	PD9	AC97CK / TCLK5


**APX-  
SOM9G45**

127	PD8	AC97FS / TIOB5
128	PD7	AC97TX / TIOA5
129	PD6	AC97RX
130	PE31	PWM2 / PCK1
131	GND_POWER	
132	PE0	LCDPWR / PCK0
133	PE30	LCDD23
134	PD5	RF0
135	PD18	SPI1_NPCS2 IRQ
136	PD4	RK0
137	PD3	RD0
138	PD25	SPI0_NPCS2 / PWM1
139	PD2	TD0
140	PD1	TF0
141	PD0	TK0
142	+3,3 VCC	
143	PE3	LCDVSYNC
144	PE4	LCDHSYNC
145	PE5	LCDDOTCK
146	PE6	LCDDEN
147	PE2	LCDDCC
148	PE1	LCDDMOD
149	PE25	LCDD18
150	PE7	LCDD0 / LCDD2
151	GND_POWER	
152	PE8	LCDD1 / LCDD3
153	PE9	LCDD2 / LCDD4
154	PE10	LCDD3 / LCDD5
155	PE11	LCDD4 / LCDD6
156	PE12	LCDD5 / LCDD7
157	PE27	LCDD20
158	PE26	LCDD19
159	PE13	LCDD6 / LCDD10
160	+3,3 VCC	
161	PE14	LCDD7 / LCDD11
162	PE15	LCDD8 / LCDD12
163	PE16	LCDD9 / LCDD13
164	PE17	LCDD10 / LCDD14
165	PE18	LCDD11 / LCDD15
166	PE28	LCDD21
167	PE29	LCDD22
168	PE19	LCDD12 / LCDD18
169	GND_POWER	
170	PE20	LCDD13 / LCDD19
171	PE21	LCDD14 / LCDD20
172	PE22	LCDD15 / LCDD21
173	PE23	LCDD16 / LCDD22
174	PE24	LCDD17 / LCDD23
175	PB16	SPI1_SPCK / SCK0
176	PB14	SPI1_MISO
177	WAKE_UP	
178	PB15	SPI1_MOSI / CTS0
179	SHUT_DOWN	
180	+3,3 VCC	
181	EN_1V2	NOT USED
182	HDDA_P	
183	VBAT	
184	HDDA_N	



**APX-  
SOM9G45**

185	GND_POWER	
186	GND_POWER	
187	NRST	
188	HDDB_P	
189	RTCK	
190	HDDB_N	
191	TDO	
192	+3,3 VCC	
193	NTRST	
194	J_HDDB_P	
195	TDI	
196	J_HDDB_N	
197	TCK	
198	Not Connected	
199	TMS	
200	Not Connected	