

Facultad de Ingeniería

Escuela de Sistemas

Disertación Previa A La Obtención Del Título De Ingeniera En Sistemas Y Computación

Tema:

Desarrollo De Un Prototipo de Software para Calcular Costos de Producción y sugerir Precios de Venta Mediante el Uso de Inteligencia Artificial en una Microempresa.

Caso de Estudio: Decorel

Autor:

Iván David Enríquez Burbano

Director:

Mgtr. Damián Nicolalde

Quito - 2020

DEDICATORIA

Esta disertación va dedicada a todas las personas que me apoyaron y que nunca permitieron que me rinda durante este largo camino.

A mis padres Elizabeth e Iván

A ustedes que siempre me han apoyado y han creído en mi por mis aciertos, pero sobre todo por darme más fuerza cuando cometí errores, por darme ánimos para seguir adelante en cada tropiezo en cada mala decisión, les dedico este trabajo como muestra de todo su esfuerzo aplicado en mí, que me han dado todo y más, a ustedes que cada día se han esforzado por verme ser un hombre de bien.

AGRADECIMIENTO

En primer lugar, quiero agradecer a mis padres quienes son mi ejemplo y mi admiración, que me han apoyado incondicionalmente durante toda mi vida y han luchado cada día para que mi hermano y yo salgamos adelante, gracias por cada ves que me apoyaron, pero en especial gracias por cada regaño en el cual me hicieron recapacitar, ya que sin ellos no lo hubiera logrado.

A mis profesores, por el conocimiento que supieron transmitirme, pero sobre todo gracias por la paciencia, por cada duda que supieron despejar de mi mente, por demostrarme que el conocimiento no tiene límites y en especial que el estudio es algo para toda la vida.

A mi hermano, que a pesar de nuestras diferencias siempre ha sabido demostrarme su apoyo.

A mi novia que siempre me apoya y ayuda durante este largo camino llamado vida y que siempre creyó en mi desde el día que nos conocimos.

Por último, a mis amigas y amigos, ya que sin su camaradería y apoyo el camino habría sido más complejo y monótono.

INDICE DE CONTENIDOS

DEDICATORIA 2

AGRADECIMIENTO 3

CAPITULO 1. INTRODUCCION 9

1.1. Antecedentes 9

1.2. Planteamiento del problema 10

1.3 Justificación 11

1.4 Objetivos 12

1.4.1 Objetivo General 12

1.4.2 Objetivos Específicos 12

CAPITULO 2. MARCO TEORICO 13

2.1 Fundamentos de producción 13

2.1.1 La Microempresa 13

2.1.2 Costos de producción 13

2.1.3 Mano de obra 14

2.1.4 Precios de venta 14

2.2 Fundamentos de sistemas 14

2.2.1 Ingeniería de software 14

2.2.2 Metodologías Agiles 14

2.2.3 Bases de datos 15

2.2.5 Lenguajes de Programación 16

2.2.6 Inteligencia Artificial y Machine Learning 16

CAPITULO 3. MARCO METODOLOGICO 18

3.1 Metodología SCRUM 18

3.1.1 Que es SCRUM 18

3.1.2 Como funciona Scrum 18

3.1.3 El proceso de Scrum 19

3.1.4 Beneficios de Scrum 20

3.2 Herramientas de desarrollo 21

3.2.1 Desarrollo Web Full-Stack 21

3.2.2 Herramientas Front-End: 23

3.2.3 Herramientas Back-End: 24

3.2.4 Herramientas de pruebas 28

3.2.5 Hardware 29

CAPITULO 4. DESARROLLO DEL PROTOTIPO 30

4.1 Sprint 0 30

4.1.1 Roles y Responsables 30
4.1.2 Declaración del producto 30
4.1.3 Área de Trabajo 30
4.1.4 Product Backlog 31
4.1.5 Definición de Hecho 32
4.2 Sprint 1 32
4.2.1 Sprint Backlog 32
4.2.2 Actividades del Sprint 33
4.2.3 Fase de codificación 33
4.2.4 Revisión 41
4.2.5 Retrospectiva 41
4.3 Sprint 2 42
4.3.1 Sprint Backlog 42
4.3.2 Actividades del Sprint 42
4.3.3 Fase de codificación 43
4.3.4 Revisión 51
4.3.5 Retrospectiva 51
4.4 Sprint 3 52
4.4.1 Sprint Backlog 52
4.4.2 Actividades del Sprint 52
4.4.3 Fase de codificación 52
4.4.4 Revisión 60
4.4.5 Retrospectiva 60
4.5 Sprint 4 61
4.5.1 Sprint Backlog 61
4.5.2 Actividades del Sprint 61
4.5.3 Fase de codificación 61
4.5.4 Revisión 64
4.5.5 Retrospectiva 64
4.6 Sprint 5 64
4.6.1 Sprint Backlog 64
4.6.2 Actividades del Sprint 65
4.6.3 Fase de codificación 65
4.6.4 Revisión 67
4.6.5 Retrospectiva 67
CAPITULO 5. CONLSUIONES Y RECOMENDACIONES 68

5.1 Conclusiones 68
5.2 Recomendaciones 69
CAPITULO 6. REFERENCIAS BIBLIOGRAFICAS 70
CAPITULO 7. ANEXOS 72
7.1 ANEXO 1: Diagramas y Modelos 72
7.1.1 Diagrama de Despliegue de Componentes Front-End 72
7.1.2 Diagrama de Despliegue de Componentes Back-End 73
7.1.3 Modelo Conceptual 74
7.1.4 Modelo Lógico 75
7.1.4 Modelo Físico 76
7.2 ANEXO 2: Manual de usuario 77
Manual de Usuario 77
1. Ingreso al Sistema 79
2. Pantalla principal del sistema (Manager) 79
3. Sección Materiales 82
- 3.1 Pantalla Principal De Materiales 82
3.2 Agregar un Material 83
3.3 Editar un Material 84
4. Sección Productos 84
- 4.1 Pantalla principal de Productos 84
4.2 Agregar un Producto 85
4.3 Editar un Producto 88
5. Sección Calculadora 89
- 6. Sección Ventas 93
- 6.1 Pantalla principal de Ventas 93
- 6.2 Agregar una Venta 94
- 7. Sección Clientes 98
- 7.1 Pantalla Principal De Clientes 98
7.2 Agregar un Cliente 99
7.3 Editar un Cliente 100
- 8. Sección Gastos 101
- 8.1 Pantalla Principal De Gastos 101
8.2 Agregar un Gasto 102
8.3 Editar un Gasto 103

Índice de Ilustraciones

Ilustración 1 - Función de un servidor web (Ionos, 2019) 16
Ilustración 2 - Aplicaciones de la IA (Dimmick, 2018) 17
Ilustración 3 - Planificación SCRUM (Proyectos Agiles, s.f.) 19
Ilustración 4 - Inspección y adaptación SCRUM (Proyectos Agiles, s.f.) 21
Ilustración 5- Funcionamiento Frontend-Backend (Chapaval, 2018) 23
Ilustración 6 - Características Python (Netec, 2020) 25
Ilustración 7- Librerías Sklearn (González, 2018) 27
Ilustración 8 - Inicio de Sesión Decorel Manager 34
Ilustración 9 - Barra lateral Decorel Manager 36
Ilustración 10 - Barra superior Decorel Manager 37
Ilustración 11 - Directorio de clientes Decorel Manager 38
Ilustración 12 - Agregar cliente Decorel Manager 38
Ilustración 13 - Editor de cliente Decorel Manager 39
Ilustración 14 - Sección Materiales Decorel Manager 43
Ilustración 15 - Material Nuevo Decorel Manager 44
Ilustración 16 - Editor de material Decorel Manager 44
Ilustración 17 - Sección Productos Decorel Manager 46
Ilustración 18 - Agregar Producto Decorel Manager 47
Ilustración 19 - Editar Producto Decorel Manager 47
Ilustración 20 - Sección de gastos Decorel Manager 53
Ilustración 21 - Nuevo gasto Decorel Manager 53
Ilustración 22 - Editar gasto Decorel Manager 53
Ilustración 23 - Sección Agregar Venta Decorel Manager 55
Ilustración 24 - Nuevo cliente Decorel Manager 56
Ilustración 25 - Agregar venta 2 Decorel Manager 56
Ilustración 26 - Agregar venta 3 Decorel Manager 57
Ilustración 27- Agregar venta 4 Decorel Manager 57
Ilustración 28 - Seccion venta Decorel Manager 62
Ilustración 29 - Sección Manager 65

Índice de Tablas

Tabla 1 - Roles Scrum 30
Tabla 2 - Product Backlog 31
Tabla 3 - Sprint Backlog 1 32
Tabla 4 - Actividades Sprint 1 - Frontend 33
Tabla 5 - Actividades Sprint 1 – Backend 33
Tabla 6 - Componente login 34
Tabla 7 - Servicio Usuario 35
Tabla 8 - API 1 35
Tabla 9 - Componente barra-lateral 36
Tabla 10 - Componente barra-superior 37
Tabla 11 - Componente cliente 39
Tabla 12 - Componente agregarcliente 40

Tabla 13 - Componente editarcliente 40
Tabla 14 - Servicio Cliente 41
Tabla 15 - API 2 41
Tabla 16 - Sprint Backlog 2 42
Tabla 17 - Actividades Sprint 2 Frontend 42
Tabla 18 - Actividades Sprint 2 Backend 43
Tabla 19 - Componente materiales 45
Tabla 20 - Componente agregarmaterial 45
Tabla 21 - Componente editarmaterial 45
Tabla 22 - Servicio material 46
Tabla 23 - API 3 46
Tabla 24 - Componente productos 48
Tabla 25 - Componente agregarproducto 49
Tabla 26 - Componente editarproducto 49
Tabla 27 - Servicio producto 50
Tabla 28 - API 4 50
Tabla 29 - Sprint Backlog 3 52
Tabla 30 - Activivdades Sprint 3 Frontend 52
Tabla 31 - Actividades Sprint Backend 52
Tabla 32 - Componente gastos 54
Tabla 33 - Componente agregargasto 54
Tabla 34 - Componente editargasto 54
Tabla 35 - API 5 55
Tabla 36 - Componente agregarventa 59
Tabla 37 - Servicio venta 59
Tabla 38 - API 6 59
Tabla 39 - Sprint Backlog 4 61
Tabla 40 - Actividades Sprint 4 Frontend 61
Tabla 41 - Actividades Sprint 4 Backend 61
Tabla 42 - Componente ventas 62
Tabla 43 - Servicio ventas 63
Tabla 44 - API 7 63
Tabla 45 - Sprint Backlog 5 64
Tabla 46 - Actividades Sprint 5 Frontend 65
Tabla 47 - Actividades Sprint 5 Backend 65
Tabla 48 - Componente manager 66
Tabla 49 - Servicio manager 66
Tabla 50 - API 8 67

CAPITULO 1. INTRODUCCION

1.1. Antecedentes

La microempresa “Decorel” se encuentra ubicado en la ciudad de Quito parroquia rural La Merced, consta de un taller con instalaciones propias de 100 metros cuadrados. y una pequeña bodega con insumos para la elaboración de cajas de cartón con cartón Kraft.

El nombre “Decorel” surge por iniciativa de su propietaria Elizabeth Burbano Cedeño, quien decidió realizar decoraciones y elaboró cajas con el nombre de “Decoraciones Elizabeth” lo que significó el nombre “Decorel”.

Está conformada por una gerente propietaria y dos operarios que en forma conjunta realizaron la producción de las cajas en forma artesanal.

La microempresa tuvo su origen en el año 2014 y nació para suplir las necesidades del mercado minoritario a nivel nacional, por cuanto las empresas que realizaron este tipo de producto satisficieron la demanda al por mayor, es decir, producción de más de 100 unidades por modelo, mientras que “Decorel” compensa el mercado de 1 a 10, a 20 cajas y de diferentes modelos.

Desde el inicio de la microempresa se la consideró como un emprendimiento digital ofreciendo sus servicios a través del medio denominado “mercado libre” por el cual se realizaron las operaciones mercantiles desde el año 2014 que a través del paso de los años obtuvo la calificación de vendedor destacado por su buen desenvolvimiento en el cumplimiento a tiempo de sus trabajos encomendados y por la calidad del material utilizado en los procesos de producción.

Este proyecto de titulación estuvo dirigido al desenvolvimiento y manejo de la contabilidad de la microempresa “Decorel”, mediante la implantación del software llamado Decorel Manager, con el que se pretendió realizar el cálculo de costos de producción y se sugirieron los precios de venta mediante la automatización de procesos como el cálculo de uso de material y de mano obra para la elaboración de las cajas que producen.

La elaboración de los productos de “Decorel”, son esencialmente de forma manual lo que le dio la característica de hacer artesanalmente cada caja, puesto que sus herramientas de trabajo son: guillotina manual, estilete, pegamento y materiales como el cartón Kraft, cartón rígido, cartulina y el vinil para el forrado de las cajas.

1.2. Planteamiento del problema

La elaboración de un software para que determine el costo de producción y que además proporcione los precios de los productos de una manera ágil, oportuna, conveniente y con precisión mediante un porcentaje de ganancia o como alternativa precios recomendados por una inteligencia artificial, con el fin de que una microempresa denominada “Decorel” la cual se especializó en la producción de cajas de cartón, en MDF y fundas de papel Kraft, pueda desarrollarse y progrese por la definición de costos y precios.

Cuando los clientes demandaron productos como cajas confeccionadas con materiales en las que se utilicen materiales como el cartón, se solicitaron precios, pero para ello el micro empresario debió determinar cada especificidad que tuvo cada producto elaborado, es decir de acuerdo al diseño, tamaño y tipo de materiales a utilizarse, por cuanto los costos variaron de acuerdo a estas consideraciones y a más del costo se incluyó la ganancia, por lo que es necesario establecer un software que les permitiera determinar sus costos con precisión debido a la variedad de elementos con los que se constituyen cada producto y por consiguiente establecer el precio de mercado de acuerdo a estas variaciones que se requirieron para elaborar cada mercancía.

La indeterminación en el cálculo del costo del producto pudo provocar dificultad en conocer cuánto le cuesta elaborar cada mercancía, de ahí que es preciso saber el costo de producción de cada uno, de acuerdo con el tamaño y al material utilizado mediante la automatización de fórmulas, en este sentido, esa especificación involucró establecer costos diferenciados, lo que les permitió calcular en forma correcta, cuanto realmente le costó elaborar cada artículo.

En base a lo expuesto anteriormente, una vez establecido el costo de la elaboración de cada producto, se pudo establecer precios para la venta de estos, de acuerdo con su diversidad, por lo que es importante para esta microempresa hacerlo con medios que hoy en día son prácticos y necesarios como son los sistemas de un programador, el cual calificó y ordenó en forma inmediata toda la diversidad que presenta cada producto.

Por consiguiente, dentro de dicho prototipo que calculó los costos de producción en base a los materiales y la mano de obra, también se incluyó una sección que utilizó librerías de inteligencia artificial para sugerir los precios de venta, esta funcionó en base a un histórico de precios de la microempresa. La idea fue aplicar una de las librerías de IA ya sea de regresión lineal o redes neuronales para un aprendizaje supervisado en base a los datos obtenidos.

1.3 Justificación

El costo de producción para elaborar mercancías fue una de las dificultades que se presentaron especialmente en las microempresas, es decir que no les permitió calcular el valor real en la elaboración del producto, mucho menos establecer el precio de venta.

“Los microempresarios generalmente tuvieron desconocimientos en contabilidad, por lo que desconocieron de sistemas contables aplicables a su tipo de actividad económica. La mayoría de estas microempresas se ven afectadas y peligran a desaparecer del mercado.” (Villamarin, 2014)

Lo expuesto anteriormente determinó que una microempresa pierda por no saber calcular sus costos de operación y sus precios de venta, de ahí la necesidad de desarrollar un software capaz que establecieran costos y propusieran precios que les brinde ganancias y por consiguiente el éxito del negocio.

Decorel es la microempresa en la cual se probaron este prototipo de software sobre costos y precios, por cuanto su propietaria careció de técnicas y metodologías adecuadas para establecer precios de costos de producción para la elaboración de cajas de cartón y al mismo tiempo estableció precios de venta de estas.

Esta microempresa que funcionó a través de la organización virtual Mercado Libre durante 4 años tuvo una reputación en donde el 94% de sus compradores lo recomendaron, con más de 3838 ventas concretadas y fue considerada por sus clientes como un vendedor destacado por sus buenas calificaciones. (Libre, 2019)

Las microempresas por ser pequeños emprendimientos productivos no alcanzaron a desarrollar estas necesidades, por eso fue imprescindible desarrollar una técnica que involucró operaciones rápidas y sin complicaciones para resolver el cálculo de producción y los precios de venta.

Actualmente existe software que pudieron solucionar los problemas planteados, pero debido a que una microempresa carece de medios y conocimientos necesarios, no desarrollaron estrategias para solucionar el tema planteado.

La existencia de microempresarios que desconocieron cómo funciona un programa o incluso un computador fueron circunstancias que les condujo a no establecer bien sus cotos de producción mucho menos sus precios de venta.

“El problema radica en la falta de base científica de los datos que se manejan, no basando la información en un sistema de métodos y tiempos, con mediciones de cantidades consumidas, que reflejen la realidad de los procesos productivos.” (Alburquerque, 2018)

El software que se elaboró estuvo dirigido a estos microempresarios que les ayudó y facilitó mediante una capacitación el manejo del programa, fue una solución que les permitió resolver posibles pérdidas por no establecer un cálculo

adecuado de la producción o el precio correcto para poder concursar en un mercado que día a día requirieron de técnicas rápidas, efectivas y les permitió ser más competitivos.

1.4 Objetivos

1.4.1 Objetivo General

Elaborar un prototipo de software para determinar la variabilidad de costos de producción y proponer precios de venta de los productos de la microempresa “Decorel” la cual se especializaron en la elaboración de cajas con materiales como el cartón.

1.4.2 Objetivos Específicos

- Análisis y comprensión del sistema productivo de Decorel.
- Levantamiento de requerimientos de la microempresa Decorel.
- Comprender y aplicar la metodología Scrum para el desarrollo del prototipo.
- Creación de modelos y sprints en base a los requerimientos.
- Desarrollo del prototipo para calcular costos de producción en base a los requerimientos con un diseño moderno, dinámico y fácil de usar.
- Implementación del módulo de IA para sugerencia de precios.
- Realizar pruebas exhaustivas del prototipo.

CAPITULO 2. MARCO TEORICO

2.1 Fundamentos de producción

2.1.1 La Microempresa

Según (Flores, 2018) “la microempresa es considerada una forma de producción en menor escala y con características personales o familiares en el área de comercio, producción, o servicios que tiene como de 1 a 15 empleados máximo”.

Dentro de las opciones de producción la microempresa tuvo una gama amplia de actividades:

- Productiva: convierte materias primas en un producto final.
- De servicios: satisface necesidades aplicando conocimientos.
- De comercio: compra y venta de productos.

2.1.2 Costos de producción

Los costos de producción fueron también llamados costos de operación, “son los gastos necesarios para mantener un proyecto, línea de procesamiento o un equipo en funcionamiento.” (FAO, 2018)

Esto significó que el destino económico de una empresa estuvo asociado con: el ingreso que se obtuvo por los bienes vendidos en el mercado y el precio obtenido y el costo de producción de lo vendido.

El costo de producción tuvo dos características opuestas:
La primera fue que para producir bienes se debe gastar; esto significó generar un costo.

La segunda característica fue que los costos debieron ser mantenidos tan bajos como sea posible y eliminados los innecesarios. Esto no significó el corte o la eliminación de los costos indiscriminadamente.

En resumen “el costo se define como la medición en términos monetarios, de la cantidad de recursos usados para algún propósito u objetivo, tal como un producto comercial ofrecido para la venta general o un proyecto de construcción. Los recursos emplean materia prima, materiales de empaque. Horas de mano de obra trabajada, prestaciones, personal salariado de apoyo, suministros y servicios comprados y capital atado en inventario, terrenos edificios y equipo”. (FAO, 2018)

La fórmula más básica para el costo de producción es:

$$Costo\ de\ produccion = Costos\ Fijos + Costos\ Variables$$

Donde los costos fijos solieran ser la mano de obra, gastos administrativos y logística, mientras que los costos variables solieran ser la materia prima necesaria para la producción.

2.1.3 Mano de obra

"Cuando una empresa es productora, se considera como mano de obra, la labor del personal que participa directamente en la producción de los bienes." (Pacheco, 2019)

Las empresas tuvieron una serie de gastos que son necesarios para cubrir todas sus operaciones, uno de ellos es el pago de los salarios a sus trabajadores.

En el caso de generar o crear un producto se basó en la hora hombre, en el cual conlleva al cálculo del tiempo de mano de obra necesario para generar dicho producto generando la siguiente formula:

$$Mano\ de\ Obra\ por\ producto = Salario\ por\ hora * Tiempo\ de\ produccion$$

2.1.4 Precios de venta

El precio es el valor monetario que se le asignó a algo. Dicho valor monetario se expresó en dinero y señaló la cantidad que debió tener el comprador o cliente para hacerse con un producto o servicio.

La venta, por otra parte, consistió en el traspaso de la propiedad de un bien a otra persona tras el pago del precio acordado. Cuando un producto estuvo en venta y un individuo desea comprarlo, tuvo la obligación de entregar una determinada cantidad de dinero para completar la operación. (Gardey, s.f.)

Entonces el precio de venta se puedo asumir como el dinero que dio el consumidor a cambio de un producto o un servicio.

Normalmente para calcular este precio de venta la fórmula más común es:

$$Precio\ de\ venta = Costos\ de\ produccion + \%\ de\ ganancia$$

2.2 Fundamentos de sistemas

2.2.1 Ingeniería de software

La Ingeniería del Software "es una disciplina o área de la Informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo". (Pressman)

2.2.2 Metodologías Agiles

Por definición, las metodologías ágiles son aquellas que permitieron adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo

flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

En esencia, quien apuesta por esta metodología consiguió gestionar sus proyectos de forma flexible, autónoma y eficaz reduciendo los costes e incrementando su productividad. (Villán, 2019)

2.2.3 Bases de datos

Se define una base de datos como “una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular”. (Valdés, s.f.)

Algunas de sus características principales son:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

2.2.4 Servidores Web

Un servidor basado en software es un programa que ofrece un servicio especial que otros programas denominados clientes pudieron usar a nivel local o a través de una red. El tipo de servicio dependió del tipo de software del servidor. La base de la comunicación fue el modelo cliente-servidor y, en lo que concierne al intercambio de datos, entraron en acción los protocolos de transmisión específicos del servicio.

Esto hizo posible distribuir las tareas entre los diferentes ordenadores y hacerlas accesibles para más de un usuario final de manera independiente. Cada servicio disponible a través de una red fue ofrecido por un servidor que estuvo permanentemente en espera. Este fue el único modo de asegurar que los clientes como el navegador web siempre tuvieran la posibilidad de acceder al servidor activamente y de usar el servicio en función de sus necesidades. (Ionos, 2019)

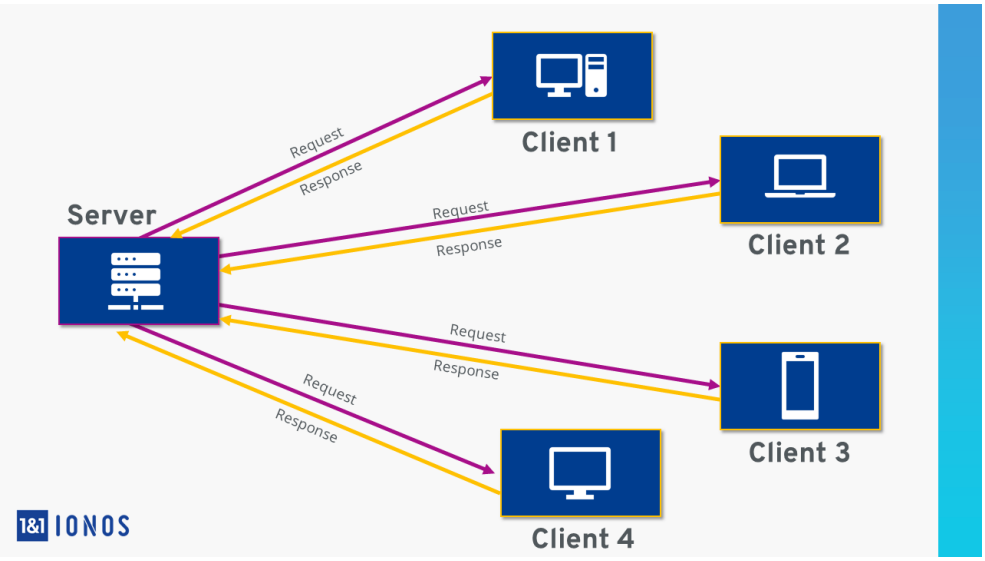


Ilustración 1 - Función de un servidor web (Ionos, 2019)

2.2.5 Lenguajes de Programación

Es un lenguaje formal que, mediante una serie de instrucciones, le permitió a un programador escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos para, de esa forma, crear programas que controlen el comportamiento físico y lógico de una máquina. (Content, 2020)

2.2.6 Inteligencia Artificial y Machine Learning

La Inteligencia Artificial (IA) es la combinación de algoritmos planteados con el propósito de crear máquinas que presentaron las mismas capacidades que el ser humano. Una tecnología que todavía nos resultó lejana y misteriosa, pero que desde hace unos años estuvo presente en nuestro día a día a todas horas. (Iberdrola, 2019)

La capacidad de inteligencia artificial para analizar datos y su precisión fue enorme. Esto permitió el desarrollo de máquinas más inteligentes para los negocios.

Algunos de sus métodos y procedimientos fueron cada vez más replicados y reemplazados por software. Por ejemplo, los investigadores del Laboratorio de Ciencias de la Computación e Inteligencia Artificial del MIT, el Hospital General de Massachusetts y la Escuela de Medicina de Harvard desarrollaron un modelo de aprendizaje automático para detectar mejor el cáncer. (Dimmick, 2018)

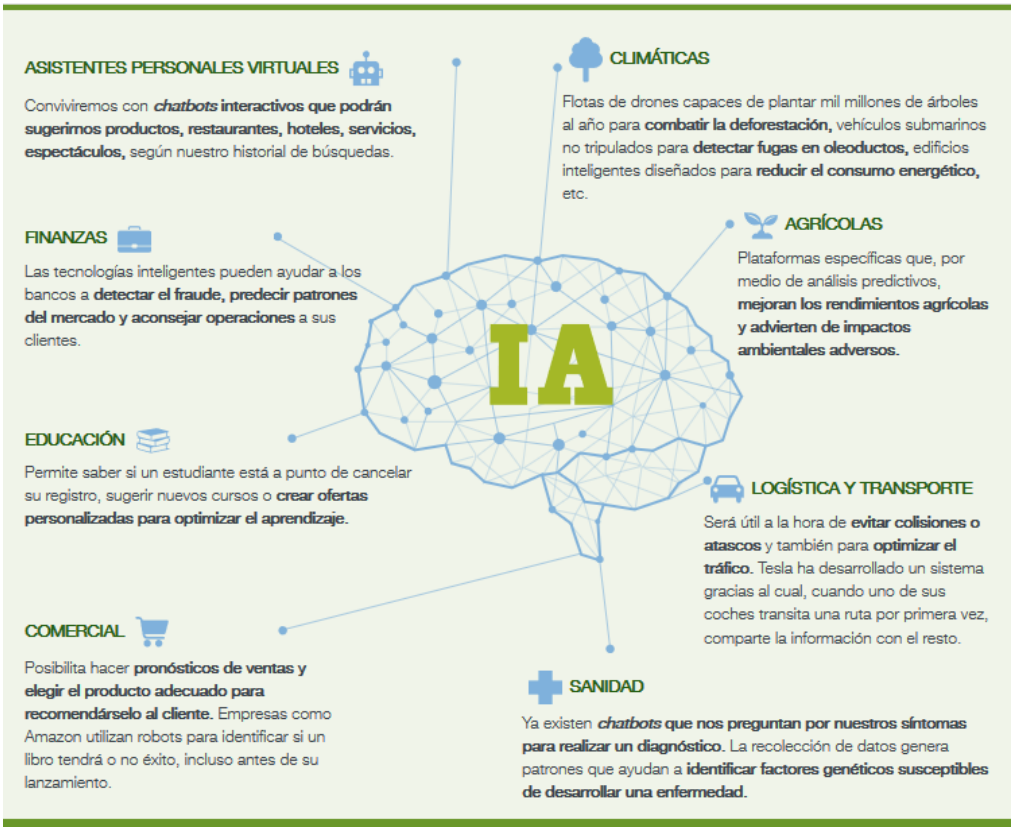


Ilustración 2 - Aplicaciones de la IA (Dimmick, 2018)

CAPITULO 3. MARCO METODOLOGICO

3.1 Metodología SCRUM

3.1.1 Que es SCRUM

Scrum es un proceso en el que se aplicaron de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyaron unas a otras y su selección tuvieron origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizaron entregas parciales y regulares del producto final, priorizadas por el beneficio que aportaron al receptor del proyecto. Por ello, Scrum estuvo especialmente indicado para proyectos en entornos complejos, donde se necesitaron obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales. (Proyectos Agiles, s.f.)

3.1.2 Como funciona Scrum

(Proyectos Agiles, s.f.) nos muestra cómo funcionó el scrum, así como los términos que se aplicaron en esta metodología ágil:

Actividades:

- Planificación de la iteración (Sprint Planning)
- Ejecución de la iteración (Sprint)
- Reunión diaria de sincronización del equipo (Scrum Daily Meeting)
- Demostración de los requisitos completados (Sprint Review)
- Retrospectiva (Sprint Retrospective)
- Refinamiento de la lista de requisitos y cambios en el proyecto

Responsabilidades:

- Cliente (Product Owner)
- Facilitador (Scrum Master)
- Equipo de desarrollo (Team)

Herramientas:

- Lista de requisitos priorizada (Product Backlog)
- Lista de tareas de la iteración (Sprint Backlog)
- Gráficos de trabajo pendiente (Burndown Chart)
- Definición de hecho (DoD)

Herramientas que se suelen utilizar en Scrum:

- Tablero de tareas (Scrum Taskboard)
- Historias de usuario y criterios de aceptación

- Planning Poker
- Velocidad del equipo
- Principios de Lean Software Development

Conceptos y técnicas relacionadas:

- Timebox (Tiempo de la iteración)
- Autoorganización y motivación intrínseca
- Skills (Habilidades) en equipo ágil
- Planificación ágil
- Base de conocimiento

3.1.3 El proceso de Scrum

Según (Proyectos Agiles, s.f.) en scrum un proyecto se ejecutó en ciclos temporales cortos y de duración fija (iteraciones entre 2 y hasta 4 semanas). Cada iteración (Sprint) tuvo que proporcionar un resultado completo, un incremento de producto final que fuera susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

El proceso parte de la lista de objetivos o requisitos priorizada del producto, que actuó como plan del proyecto. En esta lista el cliente (Product Owner) priorizó los objetivos balanceando el valor que le aportan respecto a su coste y quedaron repartidos en iteraciones y entregas.

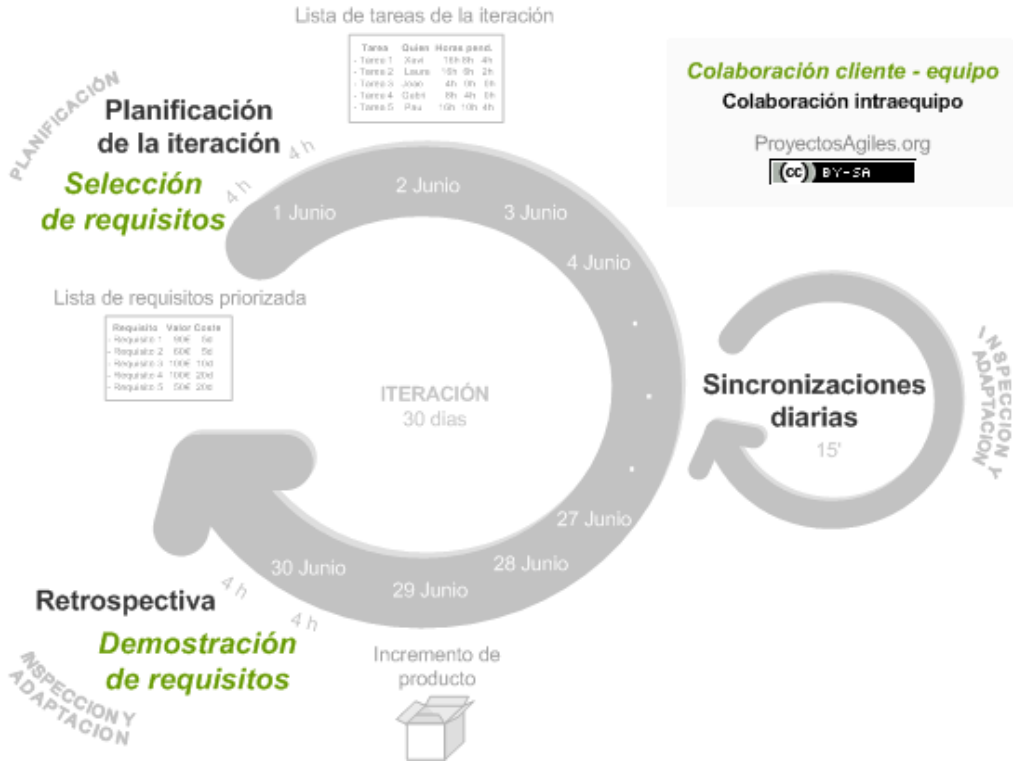


Ilustración 3 - Planificación SCRUM (Proyectos Agiles, s.f.)

Planificación de la iteración

El primer día de la iteración se realizó la reunión de planificación de la iteración. Tiene dos partes:

> **Selección de requisitos:** El cliente presentó al equipo la lista de requisitos priorizada del producto o proyecto. El equipo preguntó al cliente las dudas que surgen y seleccionó los requisitos más prioritarios que preveo que podrá completar en la iteración, de manera que pudieron ser entregados si el cliente lo solicita.
>
> **Planificación:** El equipo elaboró la lista de tareas de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hizo de manera conjunta y los miembros del equipo se autoasignaron las tareas y se autoorganizaron para trabajar.

Ejecución de la iteración: Durante la iteración el Facilitador (Scrum Master) se encargó de que el equipo pudiera mantener el foco para cumplir con sus objetivos, eliminó los obstáculos que el equipo no pudo resolver y protegió al equipo de interrupciones externas que pudieran afectar el objetivo.

Cada día el equipo (Team) realizó una reunión de sincronización, normalmente delante de un tablero físico o pizarra (Scrum Taskboard). El equipo inspeccionó el trabajo que el resto estuvo para poder hacer las adaptaciones necesarias que permitieron cumplir con la previsión de objetivos a mostrar al final de la iteración.

Durante la iteración, el cliente junto con el equipo refinó la lista de requisitos y, si es necesario, cambió o replanificó los objetivos del proyecto con el objetivo de maximizar la utilidad de lo que se desarrolló y el retorno de inversión.

Inspección y adaptación: El último día de la iteración se realizó la reunión de revisión de la iteración. Tuvo dos partes:

Revisión (demostración): El equipo presentó al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que hubo en el contexto del proyecto, el cliente realizó las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.

Retrospectiva: El equipo analizó cómo ha sido su manera de trabajar y cuáles fueron los problemas que pudieron impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargó de eliminar o escalar los obstáculos identificados que estuvieron más allá del ámbito de acción del equipo.

3.1.4 Beneficios de Scrum

(Proyectos Agiles, s.f.) dijo que los principales beneficios que proporciona Scrum son:

- Entrega mensual (o quincenal) de resultados (los requisitos más prioritarios en ese momento, ya completados) lo cual proporcionó las siguientes ventajas:
    - Gestión regular de las expectativas del cliente y basada en resultados tangibles.
    - Resultados anticipados (time to market).
    - Flexibilidad y adaptación respecto a las necesidades del cliente, cambios en el mercado, etc.
    - Gestión sistemática del Retorno de Inversión (ROI).
    - Mitigación sistemática de los riesgos del proyecto.
- Productividad y calidad.
- Alineamiento entre el cliente y el equipo de desarrollo.
- Equipo motivado.

Por último, uno de los beneficios más importantes fue la relación que se mantiene entre el cliente y el proveedor a través de todo el desarrollo y la facilidad de adaptarse a los cambios solicitados repentinamente por el cliente que solieran ser bastante común gracias a las inspecciones y adaptaciones diarias que manejo el Scrum.

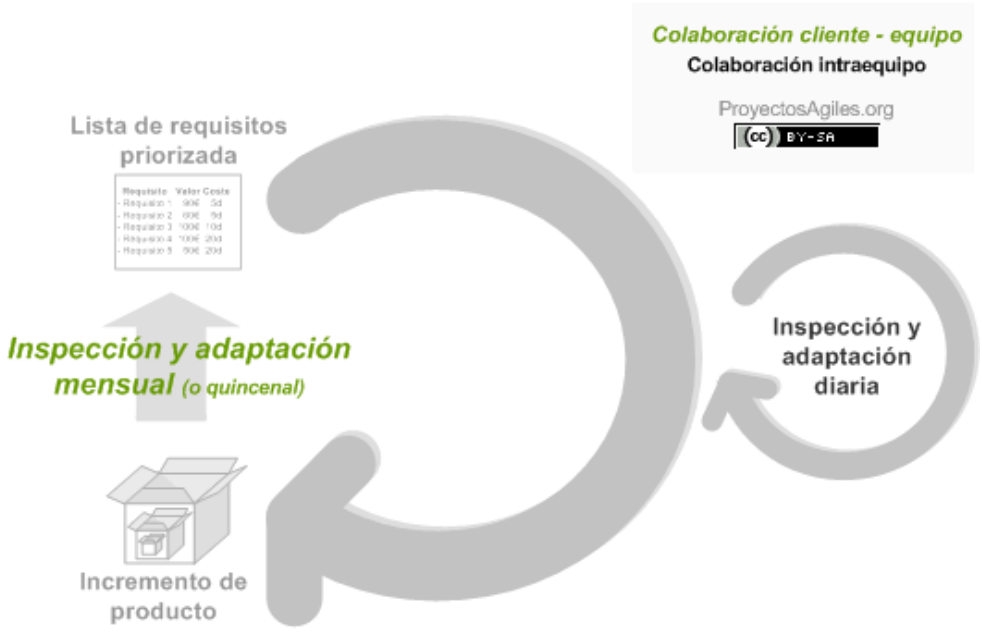


Ilustración 4 - Inspección y adaptación SCRUM (Proyectos Agiles, s.f.)

3.2 Herramientas de desarrollo

3.2.1 Desarrollo Web Full-Stack

Un desarrollo Full Stack fue el tipo de arquitectura que involucró a cada uno de los aspectos relacionados con la creación y el mantenimiento de una aplicación web. Para ello fue fundamental que el desarrollador Full Stack tuvieran conocimientos en desarrollo Front-End y Back-End. Además de manejar diferentes sistemas operativos y lenguajes de programación.

Aplicación Web: Las aplicaciones web fueron herramientas a las que se accedió a través de cualquier navegador sin necesidad de contar con un programa instalado previamente. Sin duda cuentan con grandes ventajas, como la independencia del sistema operativo o la facilidad para actualizarlo. Sin olvidar la capacidad para almacenar la información de forma permanente en servidores web. (Castaño, 2018)

Front-End: Es la parte de un programa o dispositivo a la que un usuario pudo acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corrieron en el navegador y que se encargaron de la interactividad con los usuarios.

Un programador Frontend debió saber de códigos HTML, CSS y JavaScript para poder usar algunos frameworks o librerías que expandieron sus capacidades para crear cualquier tipo de interfaces de usuarios. React, Redux, Angular, Bootstrap, Foundation, LESS, Sass, Stylus y PostCSS son algunos de ellos. (Chapaval, 2018)

Back-End: (Chapaval, 2018) nos dice que backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas.

Algunos de los lenguajes de programación de Backend son Python, PHP, Ruby, C# y Java, y así como en Frontend, cada uno de los anteriores tuvieron diferentes frameworks que permitieron trabajar mejor según el proyecto que se estuvo desarrollando.

Como funciona el Frontend y el Backend:

La web estuvo compuesta de un montón de documentos que estuvieron conectados entre sí, a través de enlaces. Cuando quieres entrar a un sitio o aplicación web escribes la URL en la barra del navegador, eso quiere decir que se está solicitando que se muestre una página web.

En el siguiente paso el computador verifica qué servidor de software estuvo corriendo el sitio. Ahí el servidor mira qué tipo de petición es la que se está haciendo. Si se entra a https://ejemplo.com/cursos/ es una petición GET que se obtiene el sitio, aquí tiene que conectarse a la base de datos para conseguir el listado de cursos.

Algunas veces no se requiere conexión a la base de datos, por ejemplo: si se entra a https://ejemplo.com/login no hace falta, cuando se inicia sesión se hace una petición POST que se conecta a la base de datos para validar los accesos y el tipo de cuenta o suscripción que tienes, y con eso el Backend devuelve la respuesta al servidor para que la empuje al navegador.

Finalmente entra el Frontend que recibe la información que le pasó el Backend y la acomoda en la interfaz del sitio, que en este caso sería el perfil del usuario. Esto es lo que últimamente se muestra en el navegador. (Chapaval, 2018)

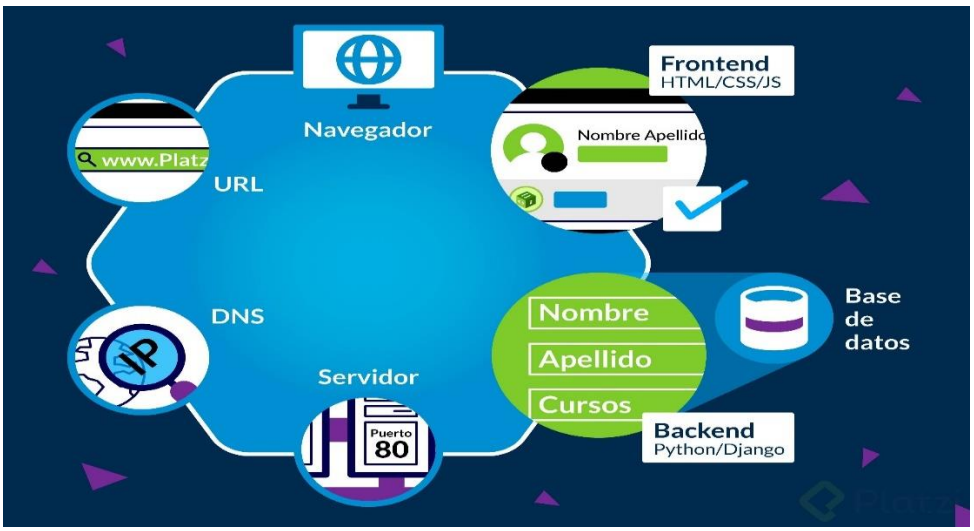


Ilustración 5- Funcionamiento Frontend-Backend (Chapaval, 2018)

3.2.2 Herramientas Front-End:
Angular

(Devs, 2019) da la definición de Angular asegurando que es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application).

Angular separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.

En una web SPA aunque la velocidad de carga puede resultar un poco lenta la primera vez que se abre, navegar después es totalmente instantáneo, ya que se ha cargado toda la página de golpe.

Solamente es una ruta la que se tiene que enviar el servidor, y Angular lo que hace ‘por debajo’ es cambiar la vista al navegar para que dé la apariencia de una web normal, pero de forma más dinámica.

Entre otras ventajas, este framework es modular y escalable adaptándose a nuestras necesidades y al estar basado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse.

El lenguaje principal de programación de Angular es Typescript, y así toda la sintaxis y el modo de hacer las cosas en el código es el mismo, lo que añade coherencia y consistencia a la información, permitiendo, por

ejemplo, la incorporación de nuevos programadores, en caso de ser necesarios, ya que pueden continuar su trabajo sin excesiva dificultad.

TypeScript

(Puciarelli, 2020) explica en su libro que TypeScript es un lenguaje de programación de código abierto creado por Microsoft en el año 2012, que implementa muchos mecanismos de programación orientada a objetos. Se trata de un superconjunto de JavaScript, lo que significa que extiende la sintaxis de JavaScript.

Al compilarlo, se genera código JavaScript, ya que el navegador no puede interpretar TypeScript. Este proceso se llama traspillar, que sígnica generar código en un lenguaje especifico a otro.

Bootstrap

(Casas, 2019) dice que Bootstrap es uno de los frameworks CSS de código abierto más conocidos y utilizados en el mundo del desarrollo web.

Este framework facilita enormemente la maquetación de páginas web, ya que permite crear una interfaz muy limpia y completamente responsive, es decir, adaptable a cualquier tamaño de pantalla. Además, ofrece tantas herramientas y funcionalidades que permite crear una web desde cero muy fácilmente.

Ventajas de Bootstrap:

- Soporte: Es una de las características más destacadas desde framework. Existe muchísima documentación en la red sobre su manejo y muchos blogs especializados en ello.
- Framework de código abierto: Esto quiere decir que todo su código está disponible y accesible a través de GitHub.
- Compatibilidad: No vas a tener problemas de incompatibilidad con los principales navegadores (Firefox, Google Chrome, Internet Explorer, Opera o Internet Explorer).
- Integración con librerías JavaScript: Tipografías, botones, cuadros…
- Facilidad de uso: Tanto a la hora de crear contenido como en el momento de subir archivos como imágenes o vídeos.
- Utiliza un sistema grid: Es decir, un sistema de cuadrículas (rejillas) que permite crear el diseño de una web insertando el contenido en bloques o columnas.

3.2.3 Herramientas Back-End:

PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. (Group, 2020)

Python

Es un lenguaje de programación creado por Guido van Rossum, se caracteriza por ser dinámico y es usado en diversas plataformas, por lo cual permite crear no solo sitios sino aplicaciones en una amplia variedad de sistemas. Python tiene la capacidad de trabajar con grandes volúmenes de datos.

Es un lenguaje de scripting independiente de plataforma y orientado a objetos, en el cual se puede realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. (Netec, 2020)

Principales características:

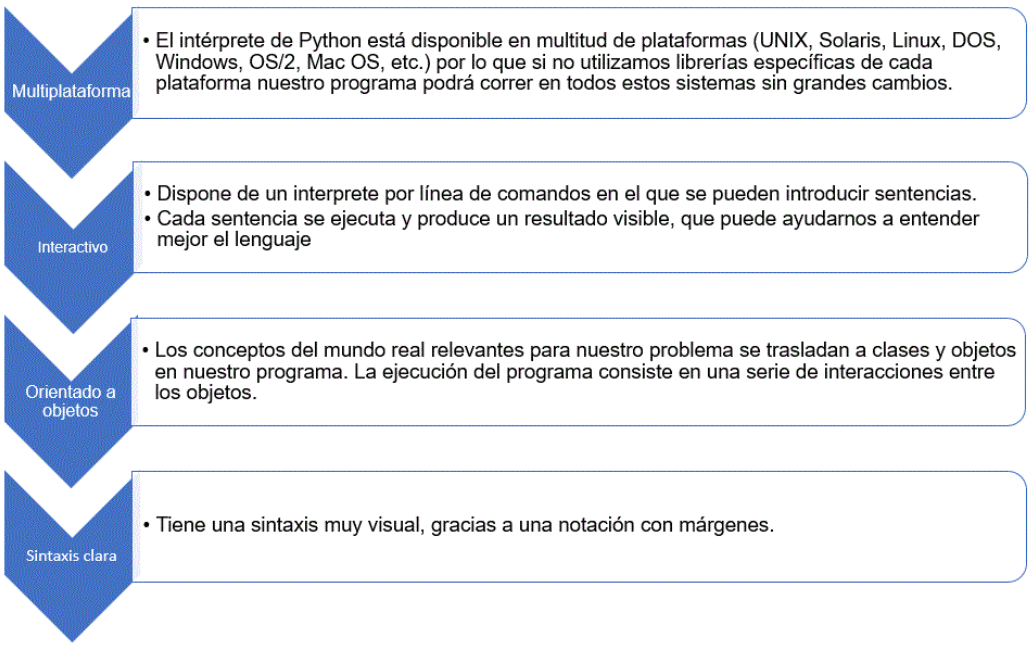


Ilustración 6 - Características Python (Netec, 2020)

MySQL

MySQL es un sistema de administración de base de datos relacionales rápido, sólido y flexible. Es ideal para crear base de datos con acceso desde páginas webs dinámicas, para la creación de sistemas de transacciones online o para cualquiera otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas. (Cobo, Patricia, Perez, & Rocha, 2005)

PHPMyAdmin

Es una herramienta gratuita escrita en PHP que proporciona una interfaz para trabajar con bases de datos MySQL. A través de esta aplicación podemos crear, modificar y eliminar registros, como también importar y exportar tablas de la base de datos MySQL. Podemos ejecutar consultas MySQL, optimizar y reparar la base de datos y muchas tareas más. (Orfali, 2017)

Características de PhpMyAdmin:

- Interfaz web intuitiva
- Soporta la mayoría de las características de MySQL:
    - Examinar y eliminar bases de datos, tablas, vistas, campos e índices
    - Crear, copiar, eliminar, cambiar el nombre y alterar bases de datos, tablas, campos e índices
    - Mantenimiento del servidor, bases de datos y tablas, con propuestas sobre la configuración del servidor
    - Ejecutar, editar y marcar cualquier sentencia SQL, incluso batch-queries
    - Administrar cuentas de usuario de MySQL y privilegios
    - Gestionar procedimientos almacenados y disparadores
- Importar datos desde CSV y SQL
- Administración de varios servidores
- Búsqueda global en una base de datos o en un subconjunto de esta.

Apache

Apache HTTP Server es un software de servidor web gratuito y de código abierto para plataformas Unix con el cual se ejecutan el 46% de los sitios web de todo el mundo. Es mantenido y desarrollado por la Apache Software Foundation.

Les permite a los propietarios de sitios web servir contenido en la web, de ahí el nombre de «servidor web». Es uno de los servidores web más antiguos y confiables, con la primera versión lanzada hace más de 20 años, en 1995.

Cuando se quiere visitar un sitio web, ingresa un nombre de dominio en la barra de direcciones de su navegador. Luego, el servidor web envía los archivos solicitados actuando como un repartidor virtual. (B., 2019)

Scikit-Learn

Scikit-Learn es probablemente la librería más útil para Machine Learning en Python, es de código abierto y es reutilizable en varios contextos, fomentando el uso académico y comercial. Proporciona una gama de algoritmos de aprendizaje supervisados y no supervisados en Python.

Esta librería está construida sobre SciPy (Scientific Python) e incluye las siguientes librerías o paquetes:

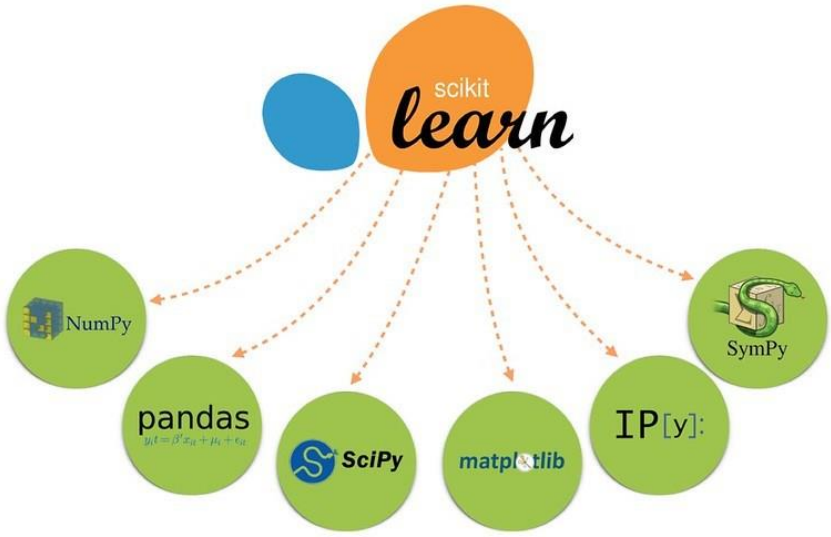


Ilustración 7- Librerías Sklearn (González, 2018)

NumPy: librería de matriz n-dimensional base

Pandas: estructura de datos y análisis

SciPy: librería fundamental para la informática científica

Matplotlib: trazado completo 2D

Ipython: consola interactiva mejorada

SymPy: matemática simbólica

Para implementar Scikit-Learn, primero se debe importar los paquetes anteriores, para poderlos implementar en la programación. (González, 2018)

Algoritmos de aprendizaje supervisados

La gran mayoría de los algoritmos de Machine Learning que entran en la clasificación de aprendizaje supervisado forman parte de Scikit-Learn, desde los modelos lineales generalizados, como lo es regresión lineal, como las máquinas de vectores de soporte (SVM), árboles de decisión y hasta los métodos bayesianos, todos ellos son parte de las herramientas de Scikit-Learn. (González, 2018)

JSON Web Token (JWT)

JWT es un estándar abierto (RFC-7519) basado en JSON para crear un token que sirva para enviar datos entre aplicaciones o servicios y garantizar que sean válidos y seguros.

El caso más común de uso de los JWT es para manejar la autenticación en aplicaciones móviles o web. Para esto cuando el usuario se quiere autenticar manda sus datos de inicio de sesión al servidor, este genera el JWT y se lo manda a la aplicación cliente, luego en cada petición el cliente envía este token que el servidor usa para verificar que el usuario este correctamente autenticado y saber quién es. (Daniel, 2016)

3.2.4 Herramientas de pruebas
Postman

(Cuervo, 2019) nos dice que Postman nace como una herramienta que principalmente permite crear peticiones sobre APIs de una forma muy sencilla y poder, de esta manera, probar las APIs. Todo basado en una extensión de Google Chrome. El usuario de Postman puede ser un desarrollador que esté comprobando el funcionamiento de una API para desarrollar sobre ella o un operador el cual esté realizando tareas de monitorización sobre un API.

Alrededor de la idea de testear las APIs, Postman ofrece un conjunto de utilidades adicionales para poder gestionar las APIs de una forma más sencilla. Es por ello que proporciona herramientas para documentar los APIs, realiza una monitorización sobre las APIs, crea equipos sobre un API para que trabajen de forma colaborativa, convirtiendo a Postman plataforma de desarrollo de APIs que se basa por un modelo de desarrollo API First.

Mozilla Firefox

Firefox es un navegador web gratuito de Mozilla Corporation que permite navegar por Internet de forma segura y rápida. Firefox, además de ser más rápido y seguro, es un navegador completamente personalizable.

3.2.5 Hardware

Raspberry pi

La Raspberry Pi es un ordenador de bajo coste y tamaño reducido, tanto es así que cabe en la palma de la mano, pero se puede conectar un televisor y un teclado para interactuar con ella exactamente igual que cualquier otra computadora.

Para encontrar su origen se tiene que ir hasta el Reino Unido, allí es donde nació como una organización caritativa la Fundación Raspberry Pi en 2009. ¿Su objetivo? Animar a los niños a aprender informática en las escuelas. Hoy en día es relativamente frecuente encontrar ordenadores y tablets en las aulas, pero no es un recurso asequible al que puedan acceder todos los centros ni todos los países.

Con Raspberry Pi esto es mucho más sencillo y abre las puertas de la experimentación y el aprendizaje a todas las edades. La Raspberry Pi se puede usar en proyectos de electrónica y para tareas básicas que haría cualquier ordenador de sobremesa como navegar por internet, hojas de cálculo, procesador de textos, reproducir vídeo en alta definición e incluso jugar a ciertos juegos. (Luis, 2018)

En este proyecto la Raspberry se usa como un pequeño servidor local donde se gestiona el acceso a las API mediante el servicio Apache y la administración de la base de datos con MYSQL.

Proporcionando a la microempresa un servicio económico en la nube localmente y de fácil acceso, sin contar su fácil movilización y el bajo consumo de recursos como la energía o el internet.

CAPITULO 4. DESARROLLO DEL PROTOTIPO

En este capítulo se describirá todo el desarrollo del prototipo Decorel Manager mediante la metodología Scrum que tomo 4 meses y se divide en 5 Sprints como se planifico.

4.1 Sprint 0

El objetivo de esta interacción (Sprint) es la de organizar todo para iniciar el desarrollo del prototipo mediante la metodología SCRUM, se definirá los roles y el producto backlog general de lo que será el sistema.

4.1.1 Roles y Responsables

En la siguiente tabla se describirá los roles y los responsables para el desarrollo del prototipo tal como lo sugiere SCRUM:

Rol	**Responsable**
Product Owner	Decorel
Scrum Master	David Enriquez
Team	David Enriquez

Tabla 1 - Roles Scrum

4.1.2 Declaración del producto

Esta será una aplicación web para el uso del personal de Decorel mediante el cual podrán llevar una gestión más organizada de la microempresa, calcular con facilidad los costos de producción y acceder a precios de venta más acertados.

4.1.3 Área de Trabajo

Como sugiere la metodología, hay que organizar un área de trabajo digna y óptima para el comenzar el desarrollo del prototipo, para ello primero vamos a usar un tablero de tareas (Scrum Taskboard) para la organización de:

- Que se debe hacer.
- Que se está haciendo.
- Lo que está terminado.

Se lo hará digitalmente a través de la herramienta Trello, esta es una página que permite administrar diferentes tipos de actividades o proyectos.

Por otro lado, para el desarrollo y codificación del prototipo se usa la herramienta Visual Studio Code, este es un software que permite la escritura y prueba de diferentes tipos de lenguajes de programación en una sola aplicación.

4.1.4 Product Backlog

Después de concretar la primera reunión con el cliente se podría concretar el nombre del software “Decorel Manager” y la lista de requisitos generales (Product Backlog) del prototipo para poder analizar la prioridad de cada una de las historias de usuario y llegar a una organización para cada Sprint.

PRODUCT BACKLOG DEL PROTOTIPO				
ID	**Historia de usuario**	**Semanas**	**Prioridad**	**Sprint**
1	Como usuario, se deseó acceder al sistema mediante algún tipo de seguridad como una página de inicio de sesión de usuarios y a la vez que cuente con roles y permisos (administrador, lector).	1	Alta	1
2	Como usuario se necesitó una sección para administrar (Buscar, agregar, modificar, eliminar) los clientes.	1	Alta	1
3	Como usuario, se necesitó una sección donde pueda administrar (Buscar, agregar, modificar, eliminar) los materiales necesarios para la creación de productos de la microempresa.	1	Alta	2
4	Como usuario, se necesitó una sección donde pueda administrar (Buscar, agregar, modificar, eliminar) los productos que maneja la microempresa para el cálculo de costos y precios.	2	Alta	2
5	Como usuario, se necesitó una sección donde pueda administrar los gastos (Buscar, agregar, modificar, eliminar)	1	Baja	3
6	Como usuario, se necesitó una sección para registrar nuevas ventas mediante el cálculo de producción de los productos y obtener un precio de venta mediante la medida que solicite el cliente.	3	Alta	3
7	Como usuario se necesitó una sección para administrar las ventas (Buscar, Eliminar)	1	Alta	4
8	Como usuario, se deseó una sección para calcular los costos de producción y precios de venta de un producto en específico mediante el ancho, largo y ancho del producto.	2	Alta	4
9	Como usuario, se deseó una página de inicio que tenga acceso a todas las funcionalidades del sistema.	1	Baja	5

Tabla 2 - Product Backlog

4.1.5 Definición de Hecho

La Definición de Hecho se utiliza en la “fase cero” del proyecto que incluye la “Inception” del producto, donde se genera la Lista de requisitos priorizada (Product Backlog) para que el equipo de desarrollo pueda realizar una buena estimación del esfuerzo global necesario para poder hacer todos los objetivos/requisitos del proyecto. (Proyectos Agiles, s.f.)

- El trabajo de desarrollo tiene que estar totalmente integrado en cada iteración.
- El trabajo del equipo ha sido revisado por al menos otro miembro del equipo.
- Todo el equipo considera que para cada objetivo/requisito se cumplen sus Criterios de Aceptación.
- El trabajo en cada iteración tiene que cumplir con los requisitos de calidad.
- Tiene que estar probado calidad externa, usabilidad, funcionalidad, seguridad.
- El producto sigue los estándares de calidad interna y ha sido refactorizado para conseguir mantenibilidad.
- El Product Owner ha validado y aceptado el objetivo.

4.2 Sprint 1

En este sprint se planifico un Timebox de 2 semanas donde se comenzó la creación de la base de datos incluido el servicio apache para el uso del prototipo, seguido del Front-End y Back-End para el inicio de sesión y la sección clientes.

4.2.1 Sprint Backlog

SPRINT BACKLOG 1				
ID	**Historia de usuario**	**Semanas**	**Prioridad**	**Sprint**
1	Como usuario, se deseó acceder al sistema mediante algún tipo de seguridad como una página de inicio de sesión de usuarios y a la vez que cuente con roles y permisos (administrador, lector).	1	Alta	1
2	Como usuario se necesitó una sección para administrar (Buscar, agregar, modificar, eliminar) los clientes.	1	Alta	1

Tabla 3 - Sprint Backlog 1

4.2.2 Actividades del Sprint

Front-End (Angular)

Actividad	Estimado (Horas)
creación del componente login	8
creación del componente barra-superior	3
creación del componente barra-lateral	6
creación del servicio Usuario	2
creación del componente clientes	4
creación del componente agregarcliente	6
creación del componente editarcliente	4
creación del servicio cliente	3
Total horas de trabajo	36

Tabla 4 - Actividades Sprint 1 - Frontend

Backend

Actividad	Estimado (Horas)
creación base de datos DecorelBigD	1
creación de la tabla Usuarios	1
creación API login	7
creación API autenticación	7
creación API cliente	4
Total horas de trabajo	20

Tabla 5 - Actividades Sprint 1 – Backend

4.2.3 Fase de codificación

Página de inicio de sesión

El desarrollo comienza con la página para iniciar sesión en la aplicación web “Decorel Manager”, aquí el usuario podrá observar una pantalla de bienvenida con el nombre del software y el logo de la empresa en la parte superior, para acceder al sistema debe ingresar su correo y contraseña previamente creado por el administrador, la api de autenticación revisara si las credenciales son correctas y si el usuario existe en la base de datos, si es así será redirigido a la página de inicio, caso contrario saltara un mensaje indicando que el acceso a sido denegado.

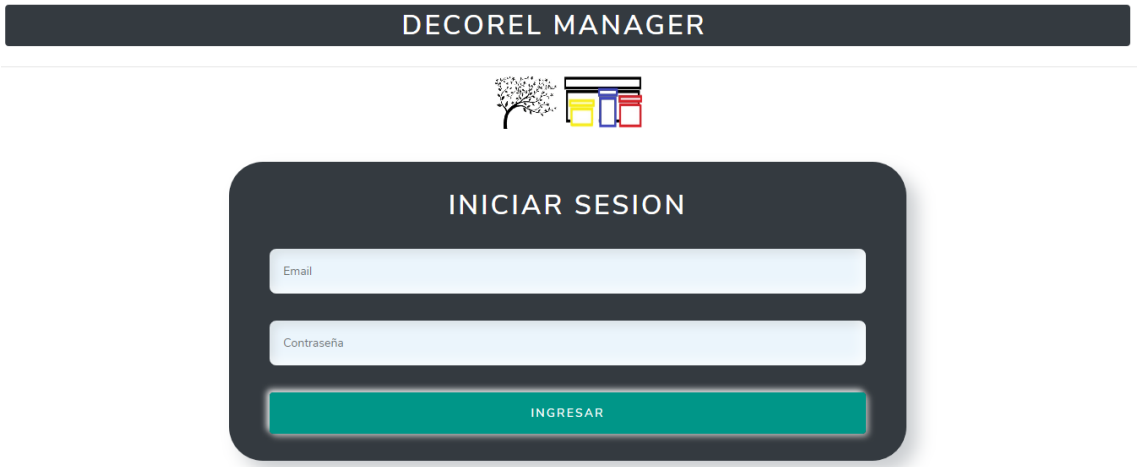


Ilustración 8 - Inicio de Sesión Decorel Manager

Front-End

Componente login	
Métodos	**Descripción**
login()	Método que recibe los datos de email y contraseña y se comunica con el servicio Usuario para enviar la solicitud de inicio de sesión.

Tabla 6 - Componente login

Servicio Usuario	
Métodos	**Descripción**
login()	Método que recibe los datos de email y contraseña y se comunica con el API login para enviar la solicitud de inicio de sesión.
registro()	Método que recibe los datos de email y contraseña y se comunica con el API login para enviar la solicitud de registrar usuario.
setToken()	Método que recibe los datos de email y token para almacenarlos en las cookies del navegador

getToken()	Método que devuelve los datos de email y token almacenados en las cookies del navegador.
getEmail()	Método que devuelve los datos de token almacenados en las cookies del navegador.
getUsuarioActivo()	Método que se comunica con el API autenticación para verificar que tipo de usuario está conectado.
getTipoUsuario()	Método que se comunica con el API rolUsuario para verificar que tipo de usuario está conectado.

Tabla 7 - Servicio Usuario

Back-End

API	Descripción
autenticacion.php	API que verifica si el usuario enviado por el Front-End existe o no en la base de datos, si es así crea un token para una nueva sesión que dura una hora.
login.php	API que verifica si el usuario enviado por el Front-End tiene un token activo y valido.
rolUsuario.php	API que verifica el tipo de rol que tiene el usuario activo enviado por el Front-End.

Tabla 8 - API 1

Barra Lateral

Se desarrollo una barra lateral para la navegación entre las diferentes secciones de la aplicación web, esta se encuentra ubicada en la parte lateral izquierda y se abre automáticamente al posar el ratón sobre la barra, así como se oculta automáticamente al retirar el puntero de la barra.

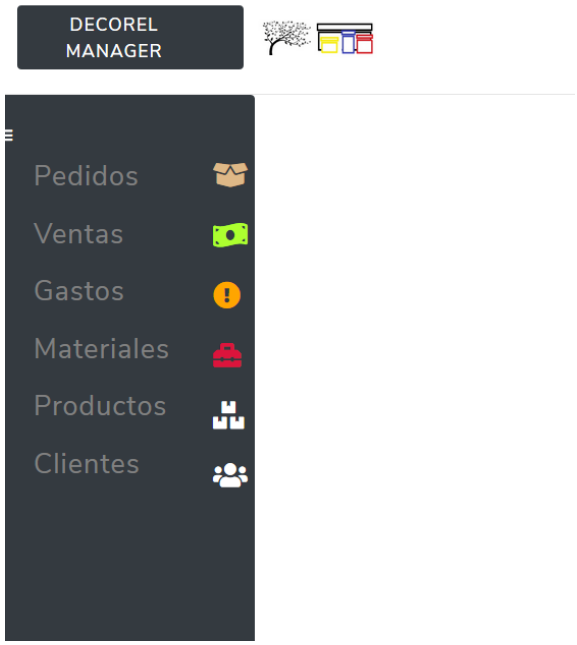


Ilustración 9 - Barra lateral Decorel Manager

Front-End

Componente barra-lateral	
Métodos	**Descripción**
pedidos()	Método que redirige a la pantalla de pedidos de la aplicación web y verifica que el usuario este logueado mediante el servicio Usuario.
ventas ()	Método que redirige a la pantalla de ventas de la aplicación web y verifica que el usuario este logueado mediante el servicio Usuario.
gastos ()	Método que redirige a la pantalla de gastos de la aplicación web y verifica que el usuario este logueado mediante el servicio Usuario.
materiales ()	Método que redirige a la pantalla de materiales de la aplicación web y verifica que el usuario este logueado mediante el servicio Usuario.
productos()	Método que redirige a la pantalla de productos de la aplicación web y verifica que el usuario este logueado mediante el servicio Usuario.
clientes()	Método que redirige a la pantalla de clientes de la aplicación web y verifica que el usuario este logueado mediante el servicio Usuario.

Tabla 9 - Componente barra-lateral

Barra Superior

Se desarrollo una barra superior para una navegación más ágil de la aplicación web mediante botones con accesos rápidos con una presentación y utilidad más acorde al diseño.

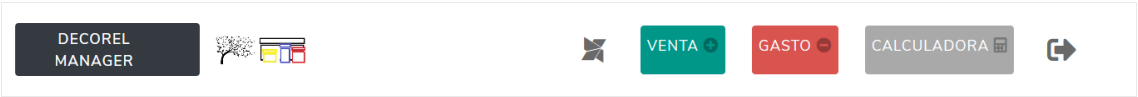


Ilustración 10 - Barra superior Decorel Manager

Front-End

Componente barra-superior	
Métodos	**Descripción**
AgregarVenta()	Método que redirige a la pantalla de nueva venta de la aplicación web y verifica que el usuario este logueado mediante el servicio Usuario.
AgregarGasto()	Método que redirige a la pantalla de nuevo gasto de la aplicación web y verifica que el usuario este logueado mediante el servicio Usuario.
Calculadora ()	Método que redirige a la pantalla de calculadora de productos de la aplicación web y verifica que el usuario este logueado mediante el servicio Usuario.
Cerrar Sesión ()	Método que elimina las cookies de sesión almacenadas en el navegador y cierra sesión en el sistema redirigiendo al usuario a la pantalla de inicio de sesión.

Tabla 10 - Componente barra-superior

Sección Clientes

Esta sección fue creada específicamente para la creación, modificación, eliminación y consulta de los clientes que registra la empresa, está compuesto por un componente y un servicio que se comunica con el api del servidor Back-End que maneja la base de datos.

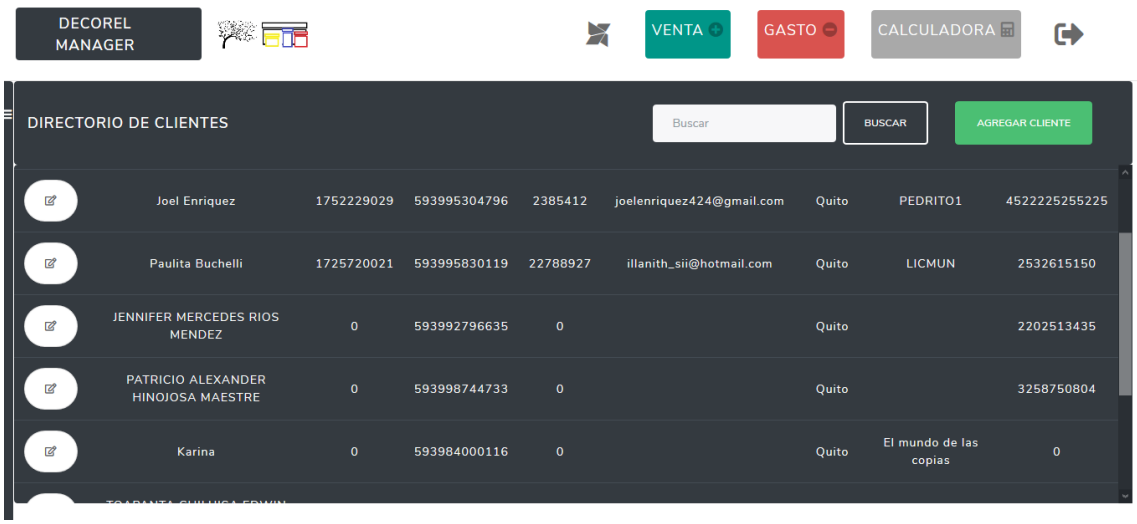


Ilustración 11 - Directorio de clientes Decorel Manager

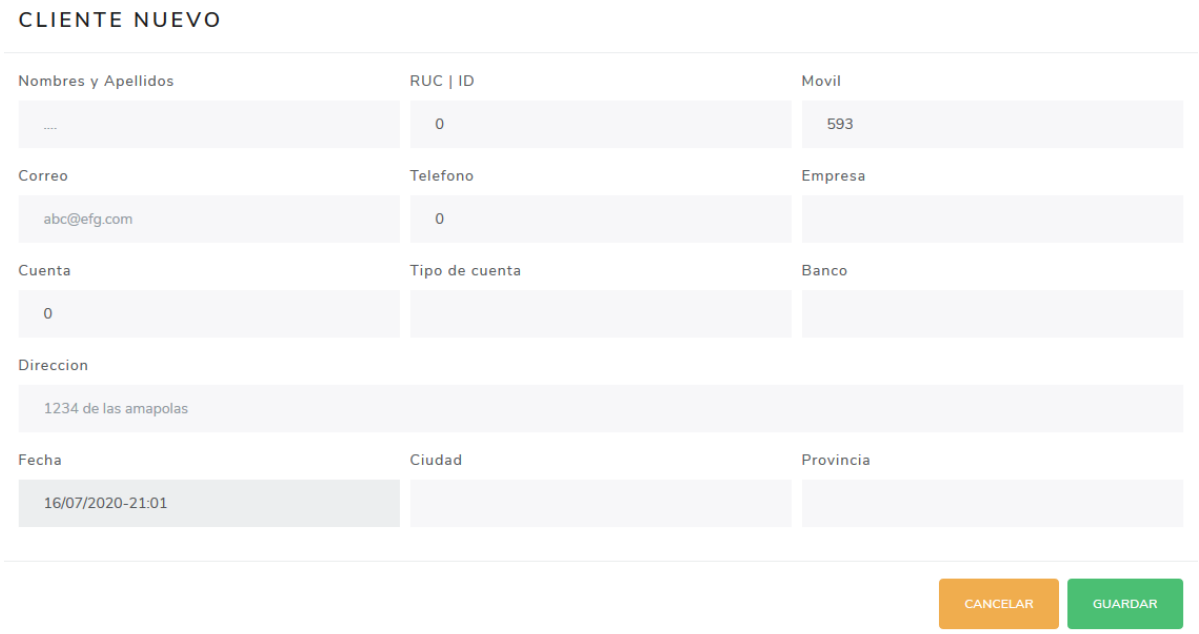


Ilustración 12 - Agregar cliente Decorel Manager

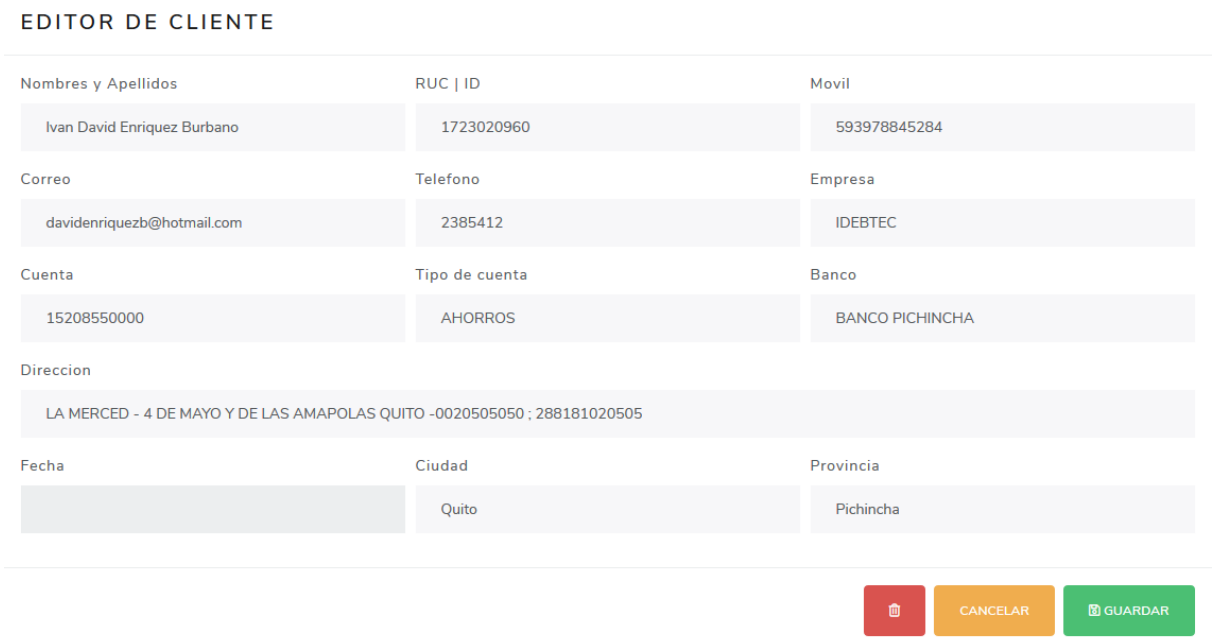


Ilustración 13 - Editor de cliente Decorel Manager

Front-End

Componente cliente	
Métodos	**Descripción**
getClientes ()	Método que se comunica con el servicio Cliente para traer todos los clientes de la base de datos mediante el API Cliente.
Búsqueda ()	Método que se comunica con el servicio Cliente para traer un cliente en específico de la base de datos mediante el API Cliente.
Agregar Cliente ()	Método que abre un formulario para agregar los datos de un nuevo cliente.
Editar Cliente ()	Método que abre un formulario para editar los datos de un cliente en específico.

Tabla 11 - Componente cliente

Componente agregarcliente	
Métodos	**Descripción**
saveCliente()	Método que se comunica con el servicio Cliente para almacenar un nuevo cliente en la base de datos mediante el API Cliente.
refrescarClientes()	Método que redirige a la pantalla principal de Clientes donde se listan todos los clientes.

Tabla 12 - Componente agregarcliente

Componente editarcliente	
Métodos	**Descripción**
saveCliente()	Método que se comunica con el servicio Cliente para almacenar un cliente que acaba de ser editado en la base de datos mediante el API Cliente.
deleteCliente()	Método que se comunica con el servicio Cliente para eliminar un cliente en la base de datos mediante el API Cliente.
refrescarClientes()	Método que redirige a la pantalla principal de Clientes donde se listan todos los clientes.

Servicio cliente	
Métodos	**Descripción**
getClientes()	Método que se comunica con el API Cliente para solicitar todos los clientes almacenados en la base de datos.
getClientesBus()	Método que se comunica con el API Cliente para solicitar los datos de un cliente especifico almacenado en la base de datos.
saveCliente()	Método que recibe los datos de un cliente y se comunica con el API Cliente para almacenar un cliente en la base de datos.
deleteCliente()	Método que recibe el id de un cliente y se comunica con el API Cliente para eliminarlo de la base de datos.

updateCliente()	Método que recibe los datos de un cliente y se comunica con el API Cliente para modificar un cliente en la base de datos.

Tabla 13 - Componente editarcliente

Tabla 14 - Servicio Cliente

Back-End

API Cliente	
API	**Descripción**
cliente.php	Api que se encarga de la creación, modificación, eliminación y consulta de la tabla Clientes en la base de Datos.

Tabla 15 - API 2

4.2.4 Revisión

Una vez finalizado la fase de desarrollo del Sprint 1, se agendo una reunión de una hora para la fase de revisión (Sprint Review) con el cliente en su oficina, donde se analizó los siguientes puntos:

1. Demostración práctica de la página de **inicio de sesión** y la sección **Clientes** del prototipo.
2. Retroalimentación del trabajo realizado por parte del Cliente.
3. Que es lo que se puede mejorar o se desea cambiar.
4. Revisión del Product backlog para ver lo que se realizara en el Sprint 2.

4.2.5 Retrospectiva

¿Qué salió bien ?	**¿Qué se puede mejorar?**	**¿Qué no salió bien?**
-Creación de una página de inicio de sesión seguro mediante JWT. -Consulta, creación, modificación y eliminación de clientes.	-En una actualización se puede agregar la opción para crear nuevos usuarios desde la página de inicio. -Se puede agregar un campo nuevo para subir una foto del cliente.	-La seguridad que se planifico para el acceso de las API´s no era la más optima así que se decidió un cambio y optar por el uso de JWT.

4.3 Sprint 2

En este sprint se planifico un Timebox de 3 semanas donde se desarrolló dos secciones, la de materiales y la de productos, en ambas se puede crear, modificar, eliminar y consultar. Estas dos secciones están relacionadas entre sí, es decir para crear un producto se necesitan de materiales, se buscó un desarrollo totalmente dinámico donde el cliente puede modificar los atributos de los productos o materiales.

4.3.1 Sprint Backlog

SPRINT BACKLOG 2				
ID	**Historia de usuario**	**Semanas**	**Prioridad**	**Sprint**
3	Como usuario, necesito una sección donde pueda administrar (Buscar, agregar, modificar, eliminar) los materiales necesarios para la creación de productos de la microempresa.	1	Alta	2
4	Como usuario, necesito una sección donde pueda administrar (Buscar, agregar, modificar, eliminar) los productos que maneja la microempresa para el cálculo de costos y precios.	2	Alta	2

Tabla 16 - Sprint Backlog 2

4.3.2 Actividades del Sprint

Front-End (Angular)

Actividad	**Estimado (Horas)**
creación del componente materiales	4
creación del componente agregarmaterial	10
creación del componente editarmaterial	5
creación del servicio material	3
creación del componente productos	4
creación del componente agregarproducto	16
creación del componente editarproducto	6
creación del servicio producto	3
Total horas de trabajo	51

Tabla 17 - Actividades Sprint 2 Frontend

Back-End

Actividad	Estimado (Horas)
creación de la tabla Materiales	1
creación API material	6
creación de la tabla Productos	1
creación API producto	10
Total horas de trabajo	18

Tabla 18 - Actividades Sprint 2 Backend

4.3.3 Fase de codificación

Sección Materiales

Se comenzó desarrollando la sección de materiales, su servicio y su API que son necesarios para la sección de productos ya que como se explicó anteriormente un producto se genera mediante el uso de materiales, en esta seccione se puede crear, modificar, eliminar y consultar los materiales que usa la microempresa.

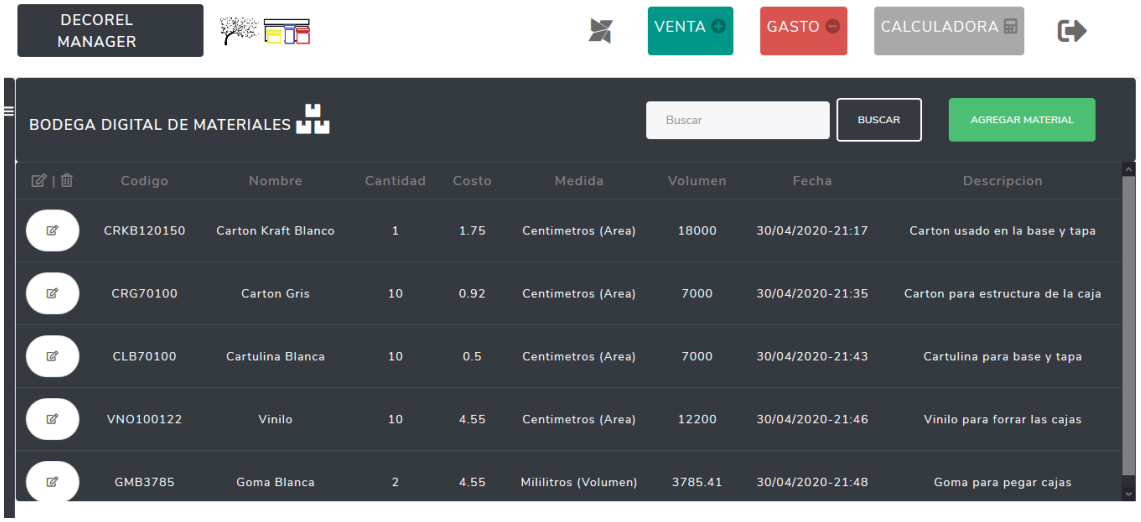


Ilustración 14 - Sección Materiales Decorel Manager

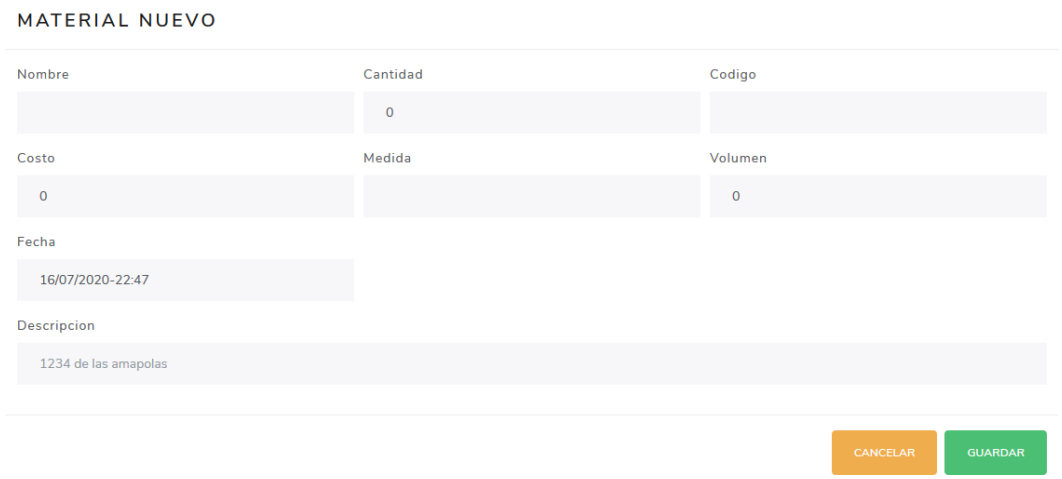


Ilustración 15 - Material Nuevo Decorel Manager

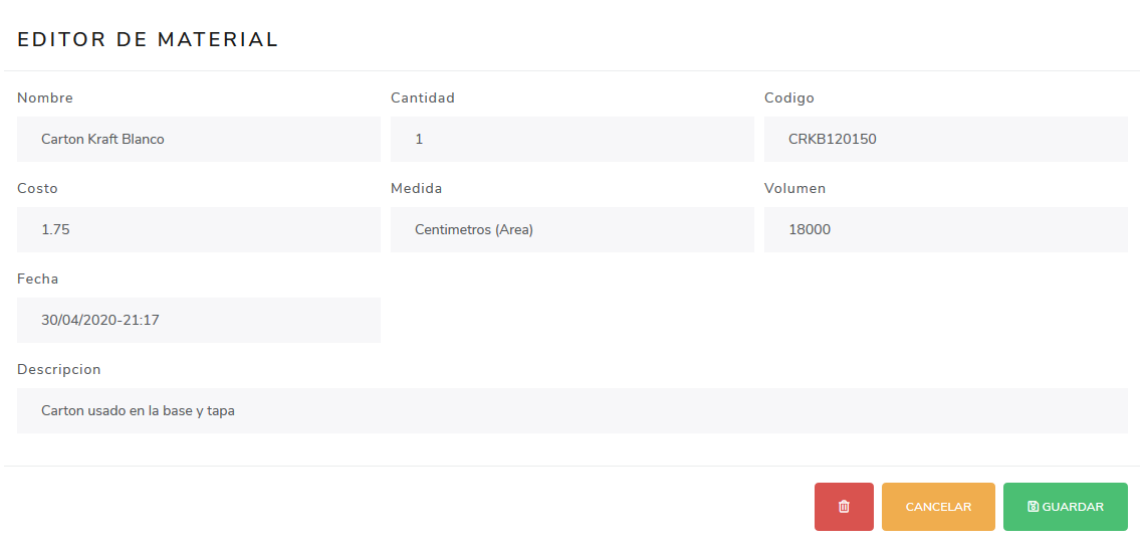


Ilustración 16 - Editor de material Decorel Manager

Front-End

Componente materiales	
Métodos	**Descripción**
getMateriales()	Método que se comunica con el servicio material para traer todos los materiales de la base de datos mediante el API material.
Busqueda()	Método que se comunica con el servicio material para traer un material en específico de la base de datos mediante el API material.

AgregarMaterial()	Método que abre un formulario para agregar los datos de un nuevo material.
EditarMaterial()	Método que abre un formulario para editar los datos de un material en específico.

Tabla 19 - Componente materiales

Componente agregarmaterial	
Métodos	**Descripción**
saveMaterial(()	Método que se comunica con el servicio material para almacenar un nuevo material en la base de datos mediante el API material.
refrescarMateriales()	Método que redirige a la pantalla principal de Materiales donde se listan todos los materiales.

Tabla 20 - Componente agregarmaterial

Componente editarmaterial	
Métodos	**Descripción**
saveMaterial()	Método que se comunica con el servicio material para almacenar un material que acaba de ser editado en la base de datos mediante el API material.
deleteMaterial()	Método que se comunica con el servicio material para eliminar un material en la base de datos mediante el API material.
refrescarMateriales()	Método que redirige a la pantalla principal de Materiales donde se listan todos los materiales.

Tabla 21 - Componente editarmaterial

Servicio material	
Métodos	**Descripción**
getMateriales()	Método que se comunica con el API material para solicitar todos los Materiales almacenados en la base de datos.
getMaterialesBus()	Método que se comunica con el API material para solicitar los datos de un material específico almacenado en la base de datos.
saveMaterial()	Método que recibe los datos de un material y se comunica con el API Material para almacenar un material en la base de datos.
deleteMaterial()	Método que recibe el id de un material y se comunica con el API material para eliminarlo de la base de datos.

updateMaterial()	Método que recibe los datos de un material y se comunica con el API material para modificar un material en la base de datos.

Tabla 22 - Servicio material

Back-End

API Cliente	
API	**Descripción**
material.php	Api que se encarga de la creación, modificación, eliminación y consulta de la tabla Materiales en la base de Datos.

Tabla 23 - API 3

Sección Productos

Una vez creado el componente material se prosigue con el componente productos, aquí el usuario puede crear sus productos usando los materiales almacenados en la base de datos, mediante el uso de fórmulas para calcular las cantidades necesarias de los materiales, también una fórmula para calcular la mano de obra y por ultimo puede colocar los gastos administrativos que conlleva cada producto, estos datos pueden ser modificados en cualquier momento ya que el desarrollo para los cálculos de producción son dinámicos y se basan en el ancho, largo y altura del producto.

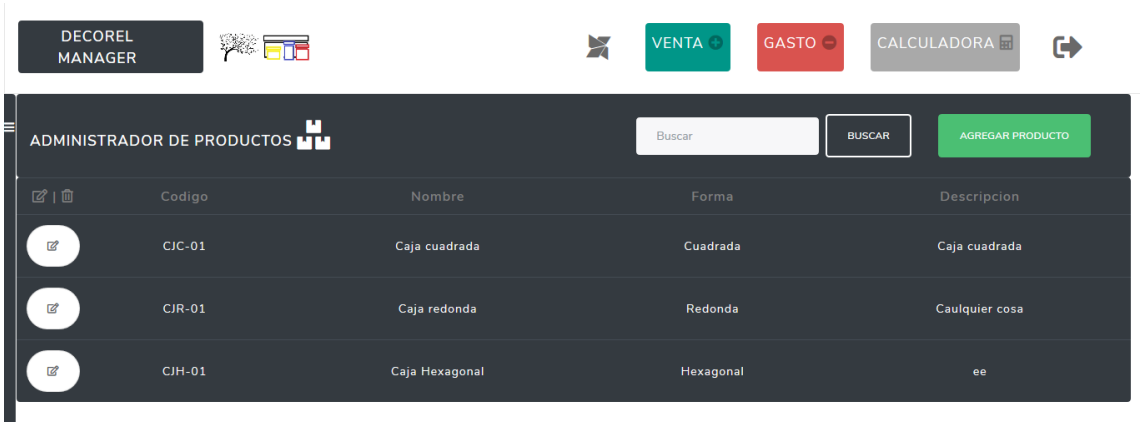


Ilustración 17 - Sección Productos Decorel Manager

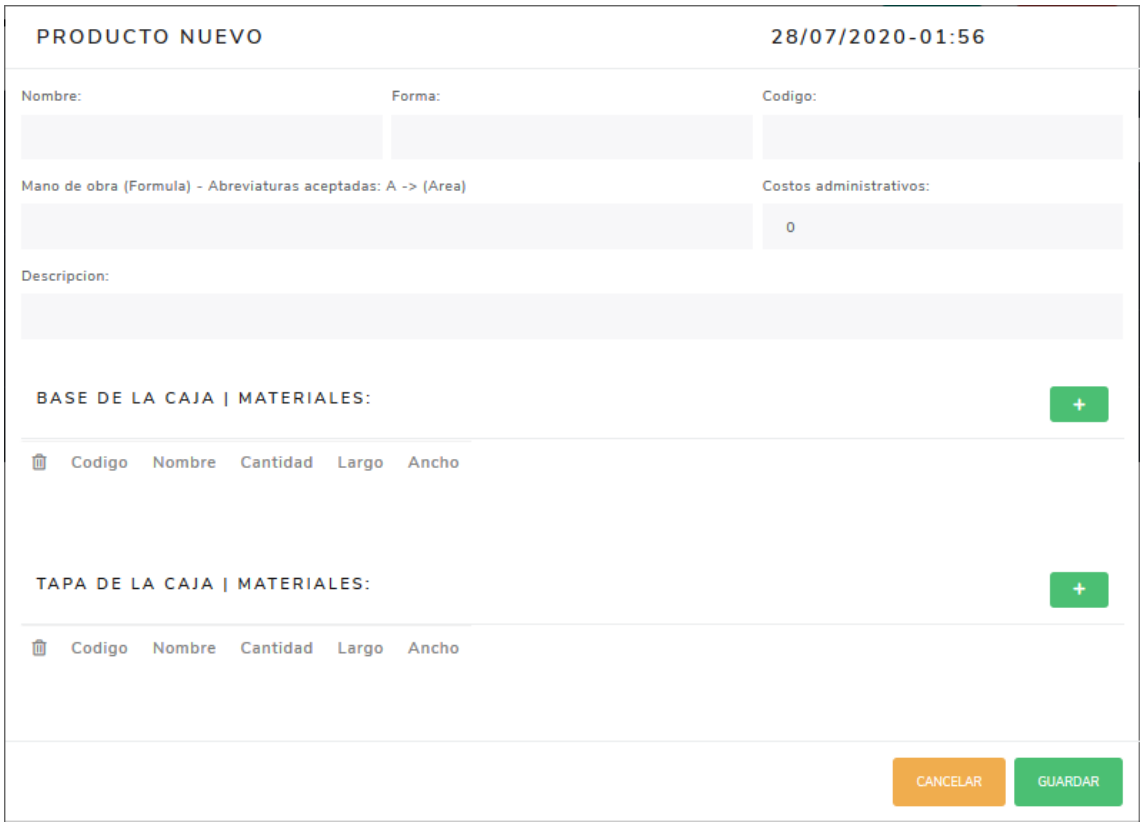


Ilustración 18 - Agregar Producto Decorel Manager

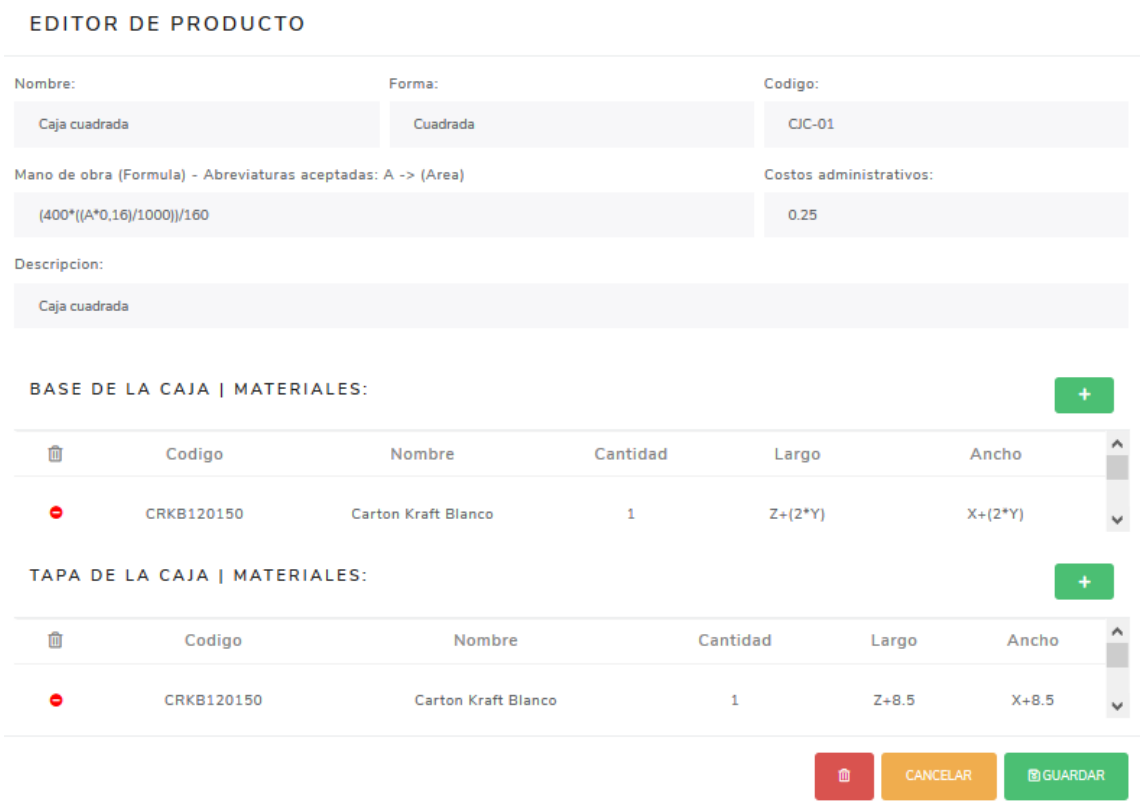


Ilustración 19 - Editar Producto Decorel Manager

Front-End

Componente productos	
Métodos	**Descripción**
getProductos()	Método que se comunica con el servicio producto para traer todos los productos de la base de datos mediante el API producto.
Busqueda()	Método que se comunica con el servicio material para traer un material en específico de la base de datos mediante el API material.
AgregarProductos()	Método que abre un formulario para agregar los datos de un nuevo producto.
EditarProducto()	Método que abre un formulario para editar los datos de un producto en específico.
MaterialesProducto()	Método que se comunica con el servicio material para traer todos los materiales de la base de datos mediante el API material.

Tabla 24 - Componente productos

Componente agregarproducto	
Métodos	**Descripción**
ObtenerMateriales()	Método que se comunica con el servicio material para traer todos los materiales de la base de datos mediante el API material.
CambiarBase()	Método que cambia entre la lista de materiales para la base del producto y el cuadro para agregar un nuevo material para la base.
AgregarMaterialB()	Método que agrega un nuevo elemento a la lista de materiales de la base.
BorrarMaterialB()	Método que borra un elemento de la lista de materiales de la base.
CambiarTapa()	Método que cambia entre la lista de materiales para la tapa del producto y el cuadro para agregar un nuevo material para la tapa.
AgregarMaterialT()	Método que agrega un nuevo elemento a la lista de materiales de la tapa.
BorrarMaterialT(material)	Método que borra un elemento de la lista de materiales de la tapa.

saveProducto()	Método que se comunica con el servicio producto para almacenar un nuevo producto mediante el API producto.
refrescarProductos()	Método que redirige a la pantalla principal de Productos donde se listan todos los productos.

Tabla 25 - Componente agregarproducto

Componente editarproducto	
Métodos	**Descripción**
ObtenerMateriales()	Método que se comunica con el servicio material para traer todos los materiales de la base de datos mediante el API material.
CambiarBase()	Método que cambia entre la lista de materiales para la base del producto y el cuadro para agregar un nuevo material para la base.
AgregarMaterialB()	Método que agrega un nuevo elemento a la lista de materiales de la base.
BorrarMaterialB()	Método que borra un elemento de la lista de materiales de la base.
CambiarTapa()	Método que cambia entre la lista de materiales para la tapa del producto y el cuadro para agregar un nuevo material para la tapa.
AgregarMaterialT()	Método que agrega un nuevo elemento a la lista de materiales de la tapa.
BorrarMaterialT(material)	Método que recibe el id de un material y borra un elemento de la lista de materiales de la tapa.
saveProducto()	Método que se comunica con el servicio producto para almacenar un nuevo producto mediante el API producto.
deleteProducto()	Método que se comunica con el servicio producto para eliminar un producto en específico de la base de datos mediante el API producto.
refrescarProductos()	Método que redirige a la pantalla principal de Productos donde se listan todos los productos.

Tabla 26 - Componente editarproducto

Servicio producto	
Métodos	**Descripción**
getProductos()	Método que se comunica con el API producto para solicitar todos los productos almacenados en la base de datos.

getMaterialesCodigo(materialesCodigo)	Método que se comunica con el API producto para solicitar los datos de un material específico almacenado en la base de datos.
getProductosBus()	Método que se comunica con el API producto para solicitar los datos de un producto específico almacenado en la base de datos.
saveProducto()	Método que recibe los datos de un producto y se comunica con el API producto para almacenar un producto en la base de datos.
deleteProducto()	Método que recibe el id de un producto y se comunica con el API producto para eliminarlo de la base de datos.
updateProducto()	Método que recibe los datos de un producto y se comunica con el API producto para modificar un producto en la base de datos.

Tabla 27 - Servicio producto

Back-End

API Cliente	
API	**Descripción**
producto.php	API que se encarga de la creación, modificación, eliminación y consulta de la tabla Productos en la base de Datos.

Tabla 28 - API 4

4.3.4 Revisión

Una vez finalizado la fase de desarrollo del Sprint 2, se agendo una reunión de una hora y media para la fase de revisión (Sprint Review) con el cliente en su oficina, donde se analizó los siguientes puntos:

1. Demostración práctica de la sección **Materiales** y la sección **Productos** del prototipo.
2. Retroalimentación del trabajo realizado por parte del Cliente.
3. Que es lo que se puede mejorar o se desea cambiar.
4. Revisión del Product backlog para ver lo que se realizara en el Sprint 3.

4.3.5 Retrospectiva

¿Qué salió bien?	¿Qué se puede mejorar?	¿Qué no salió bien?
-Sección dinámica para la creación de nuevos materiales mediante fórmulas que se usan dentro de la microempresa. -Creación de la sección productos. -Comunicación entre la sección Materiales y Productos de forma dinámica.	-En una actualización se puede agregar una opción para modificar un material en específico que usa un producto. -Se puede agregar un campo nuevo para subir una foto del producto.	-Las fórmulas otorgadas por la empresa para el calcular el uso de materiales estaban quemadas en el sistema, así que se decidió crear un método dinámico donde se pueda alterar estas fórmulas en cualquier momento por el cliente.

4.4 Sprint 3

En este sprint se planifico un Timebox de 4 semanas donde se desarrolló dos secciones la de gastos y la sección para agregar una nueva venta, esta última depende de los componentes clientes y productos para su correcto funcionamiento por eso se decidió desarrollarla en el sprint 3.

4.4.1 Sprint Backlog

ID	Historia de usuario	Semanas	Prioridad	Sprint
5	Como usuario, necesito una sección donde pueda administrar los gastos (Buscar, agregar, modificar, eliminar)	1	Baja	3
6	Como usuario, necesito una sección para registrar nuevas ventas mediante el cálculo de producción de los productos y obtener un precio de venta mediante la medida que solicite el cliente.	3	Alta	3

Tabla 29 - Sprint Backlog 3

4.4.2 Actividades del Sprint

Front-End (Angular)

Actividad	Estimado (Horas)
creación del componente gastos	3
creación del componente agregargasto	4
creación del componente editargasto	6
creación del servicio gasto	2
creación del componente agregarventa	30
creación del servicio venta	5
Total horas de trabajo	50

Tabla 30 - Actividades Sprint 3 Frontend

Back-End

Actividad	Estimado (Horas)
creación de la tabla Materiales	1
creación API gasto	6
creación de la tabla Ventas	1
creación API venta	16
Total horas de trabajo	24

Tabla 31 - Actividades Sprint Backend

4.4.3 Fase de codificación

Sección Gastos

La sección de gastos esta creada para la administración de todos los gastos que realiza la microempresa, donde se puede crear, modificar, eliminar y consultar cualquier tipo de gasto y de esta manera la microempresa pueda llevar un mayor control de sus egresos.

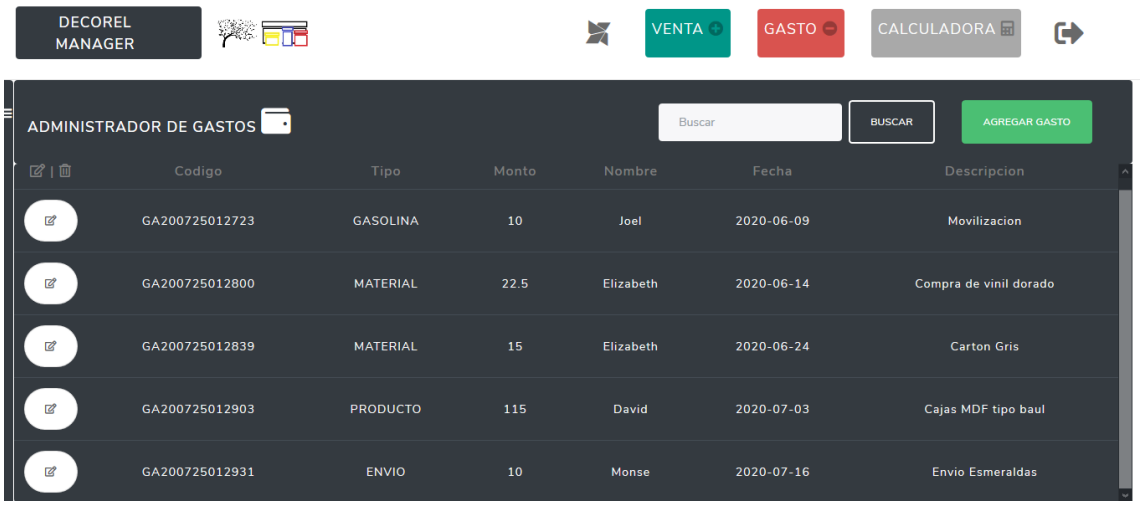


Ilustración 20 - Sección de gastos Decorel Manager

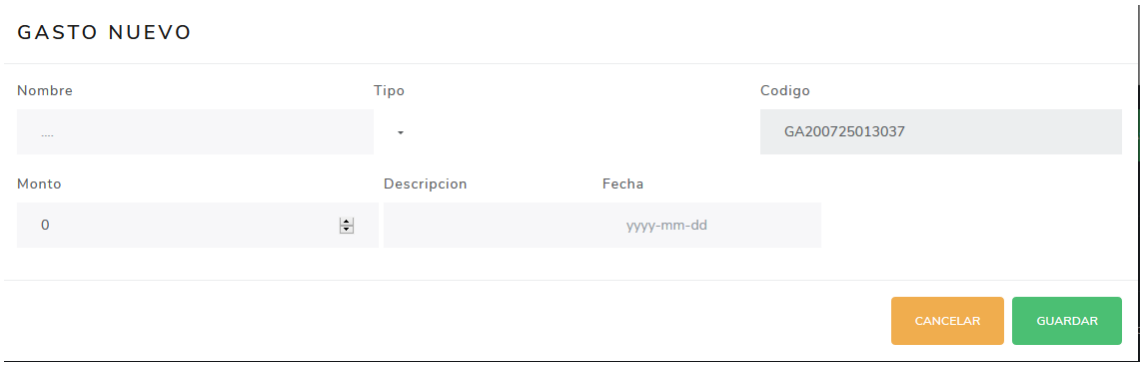


Ilustración 21 - Nuevo gasto Decorel Manager

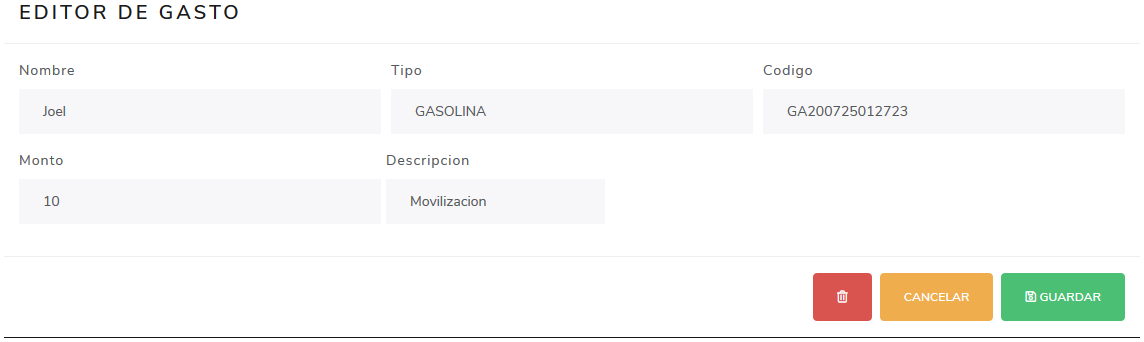


Ilustración 22 - Editar gasto Decorel Manager

Front-End

Componente gastos	
métodos	**Descripción**
getGastos()	Método que se comunica con el servicio gasto para traer todos los gastos de la base de datos mediante el API gasto.
Busqueda()	Método que se comunica con el servicio gasto para traer un gasto en específico de la base de datos mediante el API gasto.
AgregarGasto()	Método que abre un formulario para agregar los datos de un nuevo gasto.
EditarGasto()	Método que abre un formulario para editar los datos de un gasto en específico.

Tabla 32 - Componente gastos

Componente agregargasto	
Métodos	**Descripción**
saveGasto(()	Método que se comunica con el servicio gasto para almacenar un nuevo gasto en la base de datos mediante el API gasto.
refrescarGastoes()	Método que redirige a la pantalla principal donde se listan todos los gastos.

Tabla 33 - Componente agregargasto

Componente editarGasto	
Métodos	**Descripción**
saveGasto()	Método que se comunica con el servicio gasto para almacenar un gasto que acaba de ser editado en la base de datos mediante el API gasto.
deleteGasto()	Método que se comunica con el servicio gasto para eliminar un gasto en la base de datos mediante el API gasto.
refrescarGastos()	Método que redirige a la pantalla principal donde se listan todos los Gastos.

Tabla 34 - Componente editargasto

Back-End

API Cliente	
API	**Descripción**

gasto.php	Api que se encarga de la creación, modificación, eliminación y consulta de registros en la tabla Gastos en la base de Datos.

Tabla 35 - API 5

Sección Agregar Venta

La sección para registrar nuevas ventas donde entra en juego el cálculo de producción y precios los cuales pueden ser calculados de dos maneras:

- Mediante un porcentaje de ganancia basado en el costo de producción.
- Usando la librería sklearn de Inteligencia Artificial y Machine Learning que mediante un algoritmo de aprendizaje (red neuronal) analiza los registros de las ventas ya concretadas, dándonos una predicción de cuál es el mejor precio de venta mediante el costo de producción.

En esta sección para agregar una nueva venta es posible escoger a un cliente previamente registrado en la sección clientes y ligarlo a la venta, también se calcula y registran todos los valores de una venta:

- Abono
- Por cobrar
- Cantidad
- Forma de pago
- Descuento
- Descripción
- Envió
- Subtotal
- IVA
- Total

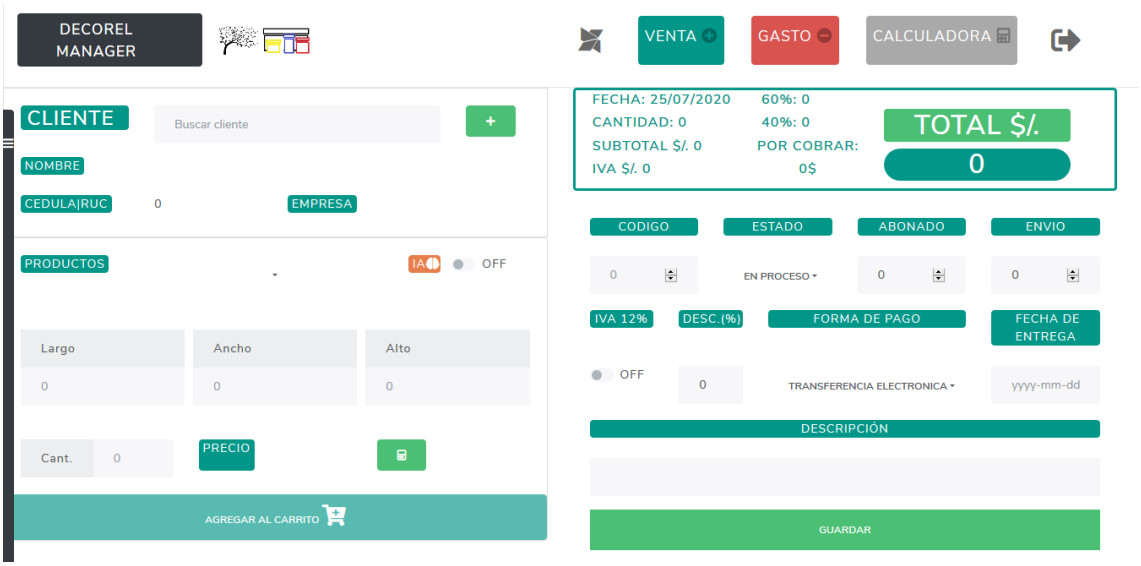


Ilustración 23 - Sección Agregar Venta Decorel Manager

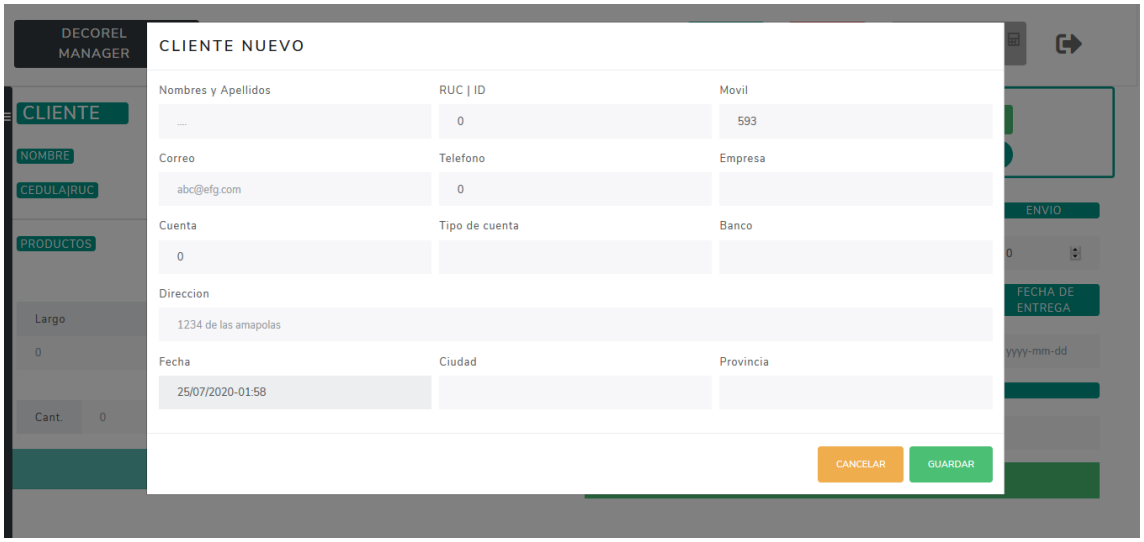


Ilustración 24 - Nuevo cliente Decorel Manager

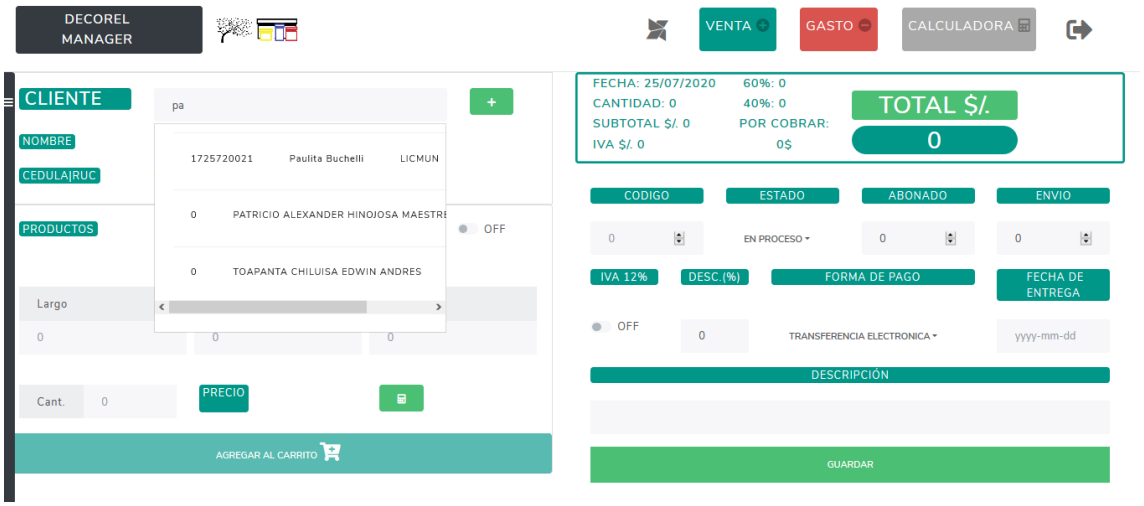


Ilustración 25 - Agregar venta 2 Decorel Manager

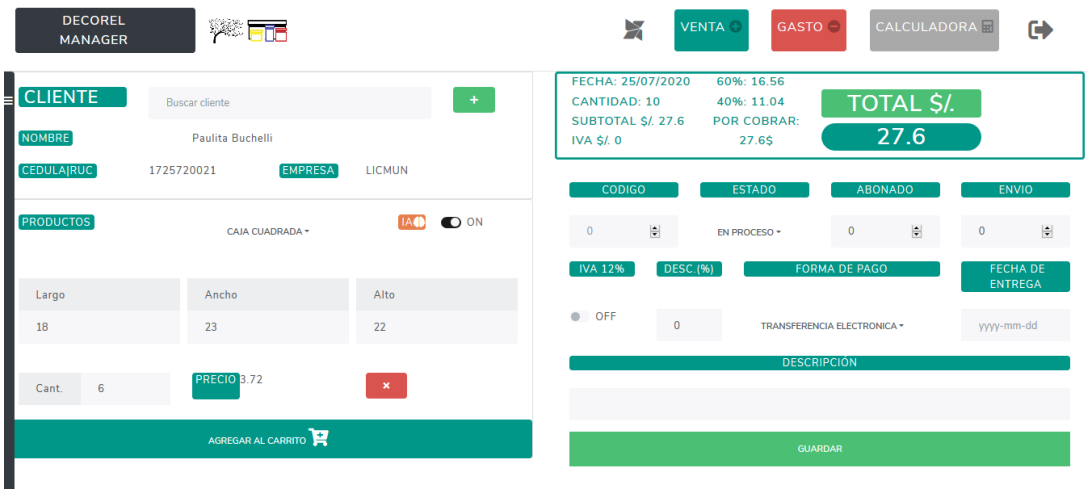


Ilustración 26 - Agregar venta 3 Decorel Manager

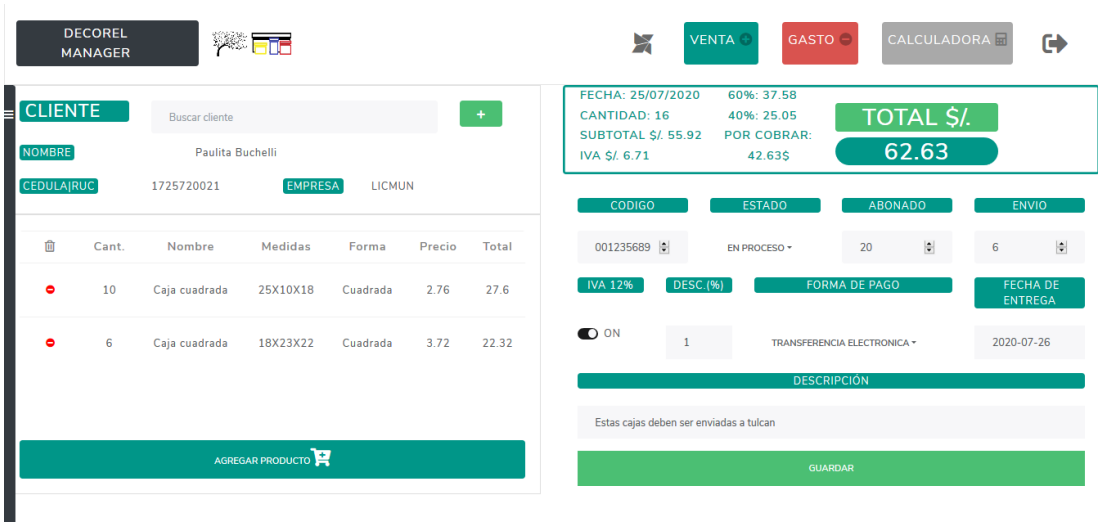


Ilustración 27- Agregar venta 4 Decorel Manager

Front-End

Componente agregarventa	
Métodos	**Descripción**
AgregarCliente()	Método que abre un formulario para agregar los datos de un nuevo cliente.
getClientes()	Método que se comunica con el servicio Cliente para traer todos los clientes de la base de datos mediante el API Cliente.
setCliente()	Método que toma los datos del cliente seleccionado y los almacena en las variables correspondientes.

getProductos()	Método que se comunica con el servicio producto para traer todos los registros de productos en la base de datos mediante el API producto.
CalcularProducto()	Método que se encarga de calcular los costos de producción y precios de venta de un producto en específico y colocar dichos valores en las variables correspondientes, también es encargado de comunicarse con el servicio venta para la comunicación con el algoritmo de IA.
TraductorFormula(formula)	Método que se encarga de traducir las fórmulas ingresadas por el usuario en los productos para calcular los costos de material dependiendo el tamaño del producto.
GetMaterial()	Método que se comunica con el servicio material y filtra los materiales necesarios para un producto en específico.
CostoBase()	Método que se encarga de calcular el costo de producción de la base de un producto en específico.
CostoTapa()	Método que se encarga de calcular el costo de producción de la tapa de un producto en específico.
AgregarCarrito(producto)	Método que se encarga de insertar todos los productos calculados en una lista para desplegar por pantalla.
BorrarProducto(producto)	Método que se encarga de eliminar uno de los productos calculados de la lista para desplegar por pantalla.
switchProducto()	Método que intercambia entre la sección de la lista de productos y la sección para calcular un producto nuevo en la pantalla.
setIA()	Método que activa o desactiva la Inteligencia Artificial.
AreaCaja()	Método que calcula el área de un producto en específico.

LimpiarVariables()	Método que se encarga de reiniciar las variables necesarias para calcular un nuevo producto.
getTotal()	Método que calcula los datos para la venta (abonado, por cobrar, descuento, subtotal, total).
setIVA()	Método que activa o desactiva el IVA.
tipopago(formap)	Método que especifica el tipo de pago para la venta.
AgregarVenta()	Método que se comunica con el servicio venta para almacenar un nuevo registro de venta en la base de datos mediante el API venta.

Tabla 36 - Componente agregarventa

Servicio venta	
Métodos	**Descripción**
saveVenta()	Método que recibe los datos de una nueva venta y se comunica con el API venta para almacenar un nuevo registro en la base de datos.
precioIA()	Método que se comunica con el API venta y este se comunica con IA.py para para calcular un precio de venta mediante IA y Machine Learning .

Tabla 37 - Servicio venta

Back-End

API Venta	
API	**Descripción**
venta.php	Api que se encarga de la creación, modificación, eliminación y consulta de la tabla Ventas en la base de Datos y se comunica con el script IA.py para calcular precios con IA y Machine Learning.
IA.py	Script que usa la librería sklearn para calcular precios de venta mediante el uso de algoritmos de IA y redes neuronales de Machine Learning mediante el análisis de una serie de registros traídos desde la base de datos y almacenados en un archivo.csv para su lectura desde el script de Python.

Tabla 38 - API 6

4.4.4 Revisión

Una vez finalizado la fase de desarrollo del Sprint 3, se agendo una reunión de una hora y media para la fase de revisión (Sprint Review) con el cliente en su oficina, donde se analizó los siguientes puntos:

1. Demostración práctica de la sección **Gastos** y la sección **Agregar Venta** del prototipo.
2. Retroalimentación del trabajo realizado por parte del cliente.
3. Que es lo que se puede mejorar o se desea cambiar.
4. Revisión del Product backlog para ver lo que se realizara en el Sprint 4.

4.4.5 Retrospectiva

¿Qué salió bien?	¿Qué se puede mejorar?	¿Qué no salió bien?
-Sección para consultar, agrega, modificar y eliminar gastos. -Creación de la sección Agregar Venta. -Desarrollo del script de IA que calcula un precio de venta. -Escoger entre la opción del uso de IA o un porcentaje de ganancia específico para el precio de venta.	-En una actualización se puede mejorar el área de gastos asociándolo con las ventas. -Se puede agregar la opción para modificar algún campo de una venta ya concretada.	-Al principio el Frontend se comunicaba directamente con el script de IA dejando una vulnerabilidad, se decidió comunicarlos a través de un API y JWT.

4.5 Sprint 4

En este sprint se planifico un Timebox de 3 semanas donde se desarrolló dos secciones la de ventas y la sección para calcular los costos de producción y precios de venta de un producto en específico.

4.5.1 Sprint Backlog

SPRINT BACKLOG 4				
ID	**Historia de usuario**	**Semanas**	**Prioridad**	**Sprint**
7	Como usuario necesito una sección para administrar las ventas (Buscar, Eliminar)	1	Alta	4
8	Como usuario, deseo una sección para calcular los costos de producción y precios de venta de un producto en específico mediante el ancho, largo y alto del producto.	2	Alta	4

Tabla 39 - Sprint Backlog 4

4.5.2 Actividades del Sprint

Front-End (Angular)

Actividad	**Estimado (Horas)**
creación del componente ventas	4
Modificación del servicio venta	2
creación del componente calculadora	8
Total horas de trabajo	14

Tabla 40 - Actividades Sprint 4 Frontend

Back-End

Actividad	**Estimado (Horas)**
Modificación de la tabla Ventas	1
modificación del API venta	3
Total horas de trabajo	4

Tabla 41 - Actividades Sprint 4 Backend

4.5.3 Fase de codificación

Sección Ventas

En esta sección se desarrolló la parte del sistema para consultar y eliminar las ventas ingresadas por el usuario mediante la sección agregar venta y almacenadas en la base de datos mediante el API venta.



Ilustración 28 - Sección venta Decorel Manager

Front-End

Componente ventas	
Métodos	**Descripción**
getVentas()	Método que se comunica con el servicio venta para traer todas las ventas de la base de datos mediante el API venta.
Busqueda()	Método que se comunica con el servicio venta para traer una venta en específico de la base de datos mediante el API venta.
AgregarVenta()	Método que abre un formulario para agregar los datos de una nueva venta.
EditarVenta()	Método que abre un formulario para editar los datos de una venta en específico.
refrescarVentas()	Método que refresca la vista Ventas actualizando la lista con todos los registros de la base de datos.

Tabla 42 - Componente ventas

Servicio Venta	
Métodos	**Descripción**

saveVenta()	Método que recibe los datos de una nueva venta y se comunica con el API venta para almacenar un nuevo registro en la base de datos.
precioIA()	Método que se comunica con el API venta y este se comunica con IA.py para para calcular un precio de venta mediante IA y Machine Learning .
getVentas()	Método que se comunica con el API venta y devuelve todos los registros de la tabla Ventas de la base de datos.
getVentasBus()	Método que se comunica con el API venta y devuelve un registro en específico de la tabla Ventas almacenada en la base de datos.
deleteVenta()	Método que se comunica con el API venta y elimina un registro en específico de la tabla Ventas almacenada en la base de datos.

Tabla 43 - Servicio ventas

Back-End

API Venta	
API	**Descripción**
venta.php	Api que se encarga de la creación, modificación, eliminación y consulta de la tabla Ventas en la base de Datos y se comunica con el script IA.py para calcular precios con IA y Machine Learning.
IA.py	Script que usa la librería sklearn para calcular precios de venta mediante el uso de algoritmos de IA y redes neuronales de Machine Learning mediante el análisis de una serie de registros traídos desde la base de datos y almacenados en un archivo.csv para su lectura desde el script de Python.

Tabla 44 - API 7

4.5.4 Revisión

Una vez finalizado la fase de desarrollo del Sprint 4, se agendo una reunión de una hora y media para la fase de revisión (Sprint Review) con el cliente en su oficina, donde se analizó los siguientes puntos:

1. Demostración práctica de la sección **Ventas** y la sección **Calculadora** del prototipo.
2. Retroalimentación del trabajo realizado por parte del cliente.
3. Que es lo que se puede mejorar o se desea cambiar.
4. Revisión del Product backlog para ver lo que se realizara en el Sprint 5.

4.5.5 Retrospectiva

¿Qué salió bien?	¿Qué se puede mejorar?	¿Qué no salió bien?
-Sección para consultar, agregar, y eliminar ventas. -Creación de la sección Calculadora.	-Se puede agregar la opción para modificar algún campo de una venta ya concretada. -Se pueden agregar información más detallada con respecto a los cálculos de producción. -Se puede crear una sección proforma.	

4.6 Sprint 5

En este sprint se planifico un Timebox de 1 semana donde se desarrolló la página de inicio, de la cual se puede acceder a las diferentes secciones del sistema y a la vez visualizar información básica de ventas, clientes o gastos.

4.6.1 Sprint Backlog

SPRINT BACKLOG 4				
ID	**Historia de usuario**	**Semanas**	**Prioridad**	**Sprint**
9	Como usuario, deseo una página de inicio que me muestre estadísticas de los datos que maneja el sistema.	1	Baja	5

Tabla 45 - Sprint Backlog 5

4.6.2 Actividades del Sprint

Front-End (Angular)

Actividad	Estimado (Horas)
creación del componente manager	8
creación del servicio manager	2
Total horas de trabajo	10

Tabla 46 - Actividades Sprint 5 Frontend

Back-End

Actividad	Estimado (Horas)
creación del API manager	4
Total horas de trabajo	4

Tabla 47 - Actividades Sprint 5 Backend

4.6.3 Fase de codificación

Sección Manager

Esta sección se desarrolló como una página de bienvenida al sistema que funciona como un centro de acceso a la mayoría de las otras secciones que dispone, en ella podemos apreciar algunos datos como el número de ventas, número de clientes o número de gastos registradas en el sistema.

Se puede crear una venta, cliente o gasto nuevo desde esta pantalla así como acceder a sus listas, también dispone de accesos directos a las secciones de productos, materiales del sistema.

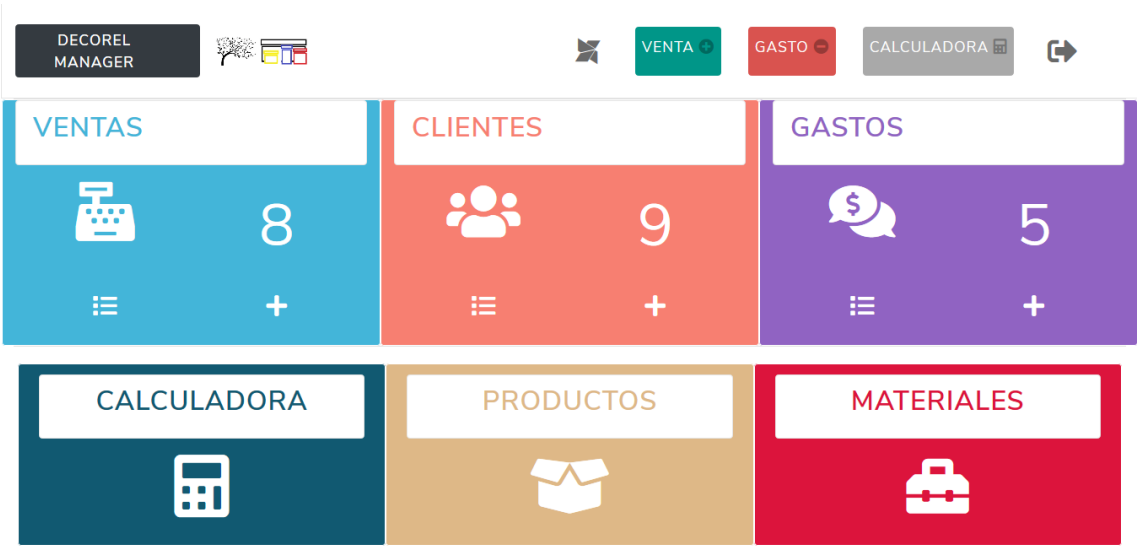


Ilustración 29 - Sección Manager

Front-End

Componente manager	
Métodos	**Descripción**
TraerDatos()	Método que se comunica con el servicio mánager para traer el número de registros almacenados de Ventas, Clientes, Gastos de la base de datos mediante el API manager.
Ventas()	Método que redirecciona la pantalla a la sección de ventas.
AgregarVenta()	Método que redirecciona la pantalla a la sección para agregar una nueva venta.
Clientes()	Método que redirecciona la pantalla a la sección de clientes.
AgregarCliente()	Método que redirecciona la pantalla a la sección para agregar un nuevo cliente.
Gastos()	Método que redirecciona la pantalla a la sección de gastos.
AgregarGasto()	Método que redirecciona la pantalla a la sección para agregar un nuevo gasto.
Calculadora()	Método que redirecciona la pantalla a la sección de calculadora.
Materiales()	Método que redirecciona la pantalla a la sección de materiales.
Productos()	Método que redirecciona la pantalla a la sección de productos.

Tabla 48 - Componente manager

Servicio manager	
Métodos	**Descripción**
getDatos()	Método que se comunica con el API mánager para devolver el número de registros de Ventas, Clientes y Gastos almacenados en la base de datos.

Tabla 49 - Servicio manager

Back-End

API manager	
API	**Descripción**
manager.php	Api que se encarga de devolver el número de registros de Ventas, Clientes y Gastos almacenados en la base de datos.

Tabla 50 - API 8

4.6.4 Revisión

Una vez finalizado la fase de desarrollo del Sprint 5, se agendo una reunión de una hora y media para la fase de revisión (Sprint Review) con el cliente en su oficina, donde se analizó los siguientes puntos:

1. Demostración práctica de la sección **Manager** y como interactúa con el resto de las secciones del prototipo.
2. Retroalimentación del trabajo realizado por parte del cliente.
3. Que es lo que se puede mejorar o se desea cambiar.
4. Análisis de futuras actualizaciones y los pasos a seguir para llevar el prototipo a un producto final.

4.6.5 Retrospectiva

¿Qué salió bien ?	¿Qué se puede mejorar?	¿Qué no salió bien?
-Sección Manager, de la cual el cliente puede interactuar con todas las funcionalidades del prototipo.	-Se puede agregar una sección de ajustes, para modificar en cualquier momento algunos campos y volúmenes del sistema.	

CAPITULO 5. CONLSUIONES Y RECOMENDACIONES

5.1 Conclusiones

- El cálculo de producción de un producto mediante el uso de materiales se logró automatizar con éxito, la clave estuvo en dinamizar este proceso, consiguiendo que el usuario pueda modificarlo en cualquier momento.

- La elección de crear este proyecto como una aplicación web mediante programación Full Stack fue clave para la conseguir un uso sencillo y familiar para los usuarios ya que pueden acceder a través de cualquier navegador localmente.

- El uso de Bootstrap basado en Material Design facilito mucho el diseño de este prototipo, dando como resultado una aplicación estética, moderna y rápida que funciona en tiempo real.

- El sistema ayudara a reducir drásticamente los tiempos y procesos en los cálculos para crear nuevos productos en base a cualquier medida dentro de la microempresa, ya que normalmente lo hacían a mano con hoja y lápiz.

- La metodología SCRUM fue clave para el desarrollo del prototipo, generando un orden y dividiéndolo en Sprints ordenadamente, se logró manejar cualquier cambio que solicitaba el cliente en cada sección, una vez resuelto cualquier improvisto era posible pasar al siguiente Sprint sin arrastrar ningún problema anterior .

- El desarrollo Full Stack puso a prueba todo lo que aprendí en la universidad, logrando integrar la mayoría de las materias en este prototipo funcional:

    - Comenzando por el diseño y creación de una base de datos en MYSQL
    - La creación y configuración de un servidor local en apache
    - Programación de un servicio API REST en PHP como Back-End
    - La programación de un Front-End con un diseño completo que integro HTML, CSS, TypeScript.
    - El manejo y aprendizaje de Inteligencia Artificial y Machine Learning mediante el análisis de datos con algoritmos de regresión lineal.

- Además de adquirir nuevos conocimientos y reforzar lo ya obtenido anteriormente, he conseguido mucha más experiencia al desarrollar proyectos agiles.

- El sistema no solo es para el cálculo de producción y precios de venta, sino que también se crearon otras funcionalidades que ayudaran administrar más efectivamente la microempresa y poder llevar una contabilidad más clara.

- El usuario final tendrá un apoyo en esta herramienta donde podrá almacenar todos estos datos sin perderlos, sino todo lo contrario revisarlos, analizarlos a beneficio cuantas veces necesite.

5.2 Recomendaciones

- Se recomienda la lectura del manual de usuario para un mejor manejo del sistema.

- En el caso de que la microempresa tenga nuevos empleados o asigne alguien nuevo el manejo del sistema se recomienda una capacitación del usuario para el uso de la herramienta.

- Aunque el prototipo funciona correctamente, aun se deben aumentar algunas funcionalidades para un manejo más optimo y dinámico del sistema.

- Aunque sigue siendo un prototipo, la idea principal es entregar un producto final y ponerlo a disposición de otras microempresas.

- Para obtener resultados dentro de la microempresa es recomendable un uso diario y constante del sistema.

- Se recomienda mantener el servidor (Raspberry) en un lugar seguro, protegido del clima o cualquier objeto que pueda alterar su funcionamiento.

- Se recomienda tener internet, para que el diseño se cargue correctamente por que el prototipo utiliza Bootstrap, sin embargo no es necesario para su funcionamiento ya que trabaja con la red local.

CAPITULO 6. REFERENCIAS BIBLIOGRAFICAS

Alburquerque, C. B. (2018). *Improven*. Obtenido de www.improven.com: https://www.improven.com/blog/resultoria-los-errores-mas-frecuentes-en-un-sistema-de-costes/

B., G. (1 de 11 de 2019). *Hostinger*. Obtenido de Hostinger: https://www.hostinger.es/tutoriales/que-es-apache/

Casas, V. (28 de 10 de 2019). *Lucushost*. Obtenido de Lucushost: https://www.lucushost.com/blog/que-es-bootstrap/

Castaño, R. (31 de 9 de 2018). *Neoland*. Obtenido de Neoland: https://www.neoland.es/blog/que-es-desarrollador-full-stack

Chapaval, N. (2018). *Platzi*. Obtenido de Platzi: https://platzi.com/blog/que-es-frontend-y-backend/

Cobo, A., Patricia, G., Perez, D., & Rocha, R. (2005). Que es MySQL ? En A. Cobo, G. Patricia, D. Perez, & R. Rocha, *PHP y MySQL* (pág. 339). Férnandez: Diaz de Santos.

Content, R. (8 de 6 de 2020). *Rock Content*. Obtenido de Rock Content: https://rockcontent.com/es/blog/que-es-un-lenguaje-de-programacion/

Cuervo, V. (02 de 14 de 2019). *Arquitecto IT*. Obtenido de Arquitecto IT: http://www.arquitectoit.com/postman/que-es-postman/

Daniel, S. (27 de 07 de 2016). *Platzi*. Obtenido de Platzi: https://platzi.com/blog/introduccion-json-web-tokens/

Devs, Q. (16 de 10 de 2019). *Quality Devs*. Obtenido de Quality Devs: https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/

Dimmick, M. (5 de 2018). *PC World*. Obtenido de PC World: http://pcworld.com.mx/traera-la-inteligencia-artificial-nuevo-renacimiento/

FAO. (2018). *The Food and Agriculture Organization*. Obtenido de http://www.fao.org/3/V8490S/v8490s06.htm

Flores, M. J. (10 de 07 de 2018). *Derecho Ecuador*. Obtenido de Derecho Ecuador: https://www.derechoecuador.com/microempresa

Francia, J. (s.f.). *SCRUM WEB*. Obtenido de SCRUM WEB: https://www.scrum.org/resources/blog/que-es-scrum

Gardey, J. P. (s.f.). *Definicion.de*. Obtenido de Definicion.de: https://definicion.de/precio-de-venta/

González, L. (02 de 11 de 2018). *Ligdi González*. Obtenido de Ligdi González: https://ligdigonzalez.com/libreria-scikit-learn-de-python/

Group, P. (2020). *PHP*. Obtenido de PHP: https://www.php.net/manual/es/intro-whatis.php

Iberdrola. (2019). *Iberdrola*. Obtenido de https://www.iberdrola.com/innovacion/que-es-inteligencia-artificial

Ionos. (02 de 08 de 2019). Obtenido de Ionos: https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-un-concepto-dos-definiciones/

Libre, M. (11 de 2019). *Mercado Libre*. Obtenido de Mercado Libre: https://www.mercadolibre.com.ec/perfil/DECOREL

Luis, E. R. (18 de 07 de 2018). *Xataka*. Obtenido de Xataka: https://www.xataka.com/makers/cero-maker-todo-necesario-para-empezar-raspberry-pi

Netec. (2020). *Netec*. Obtenido de Netec: https://www.netec.com/que-es-python

Orfali, C. M. (4 de 4 de 2017). *Toadworld*. Obtenido de https://blog.toadworld.com/2017/04/04/que-es-phpmyadmin-y-como-podemos-gestionar-la-base-de-datos-mysql-con-esta-herramienta

Pacheco, J. (13 de 11 de 2019). *Web y Empresas*. Obtenido de Web y Empresas: https://www.webyempresas.com/mano-de-obra-directa/

Pressman, R. S. (s.f.). *Ingeniería del Software. Vol. I.* Obtenido de https://micarrerauniversitaria.com/c-ingenieria/ingenieria-de-software/

Proyectos Agiles. (s.f.). Obtenido de Proyectos Agiles: https://proyectosagiles.org/que-es-scrum/

Proyectos Agiles. (s.f.). Obtenido de Proyectos Agiles: https://proyectosagiles.org/como-funciona-scrum/

Proyectos Agiles. (s.f.). Obtenido de Proyectos Agiles: https://proyectosagiles.org/beneficios-de-scrum/

Puciarelli, L. (2020). Angular. En L. Puciarelli, *Angular* (pág. 9). Buenos Aires: Six Ediciones.

Valdés, D. P. (s.f.). *Maestros del Web*. Obtenido de Maestros del Web: http://www.maestrosdelweb.com/que-son-las-bases-de-datos/

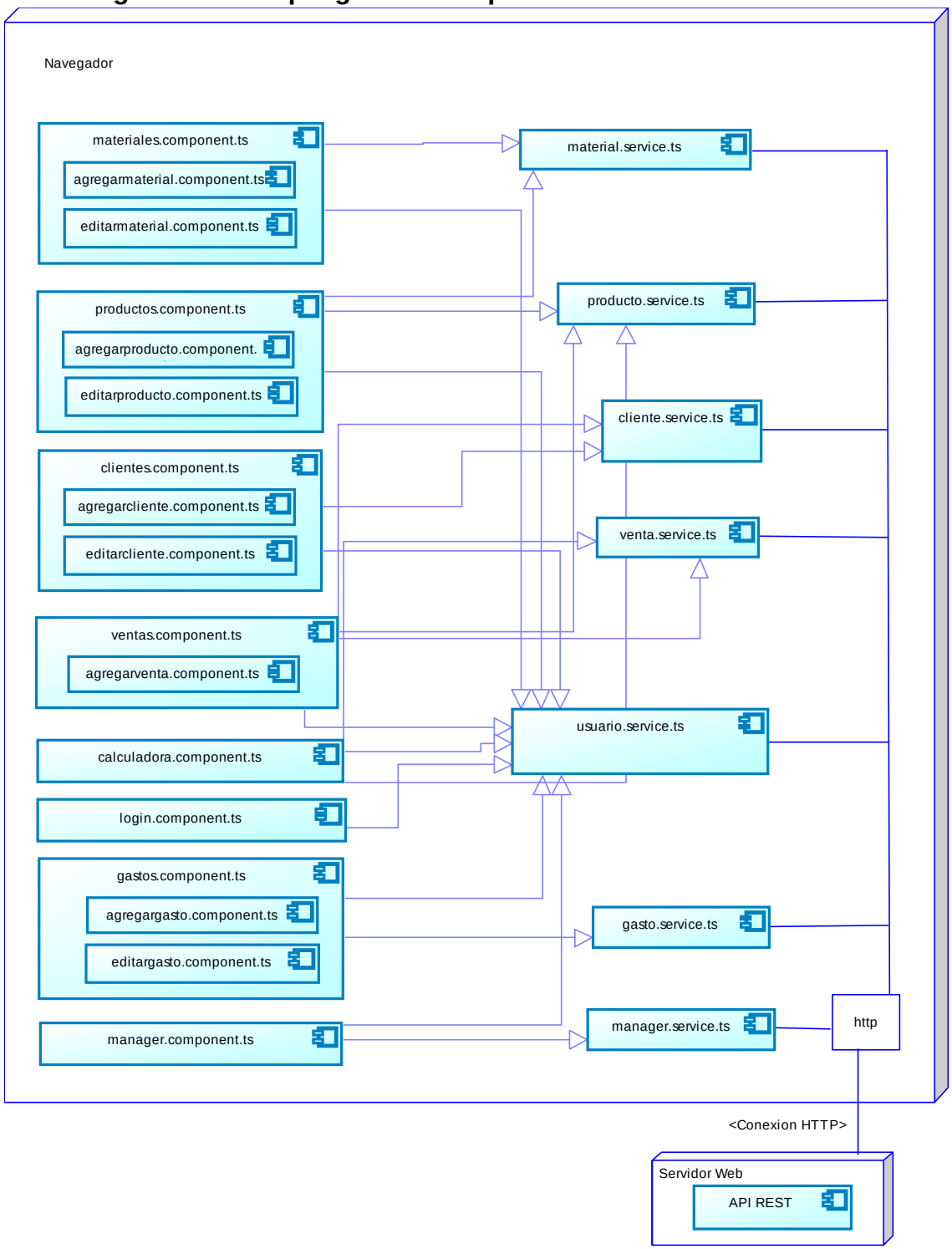
Villamarin, J. V. (2014). *UNIVERSIDAD TÉCINICA DE AMBATO.* Obtenido de UNIVERSIDAD TÉCINICA DE AMBATO: http://repositorio.uta.edu.ec/bitstream/123456789/20575/1/T2592i.pdf

Villán, V. R. (15 de 03 de 2019). *IEBS*. Obtenido de IEBS: https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/

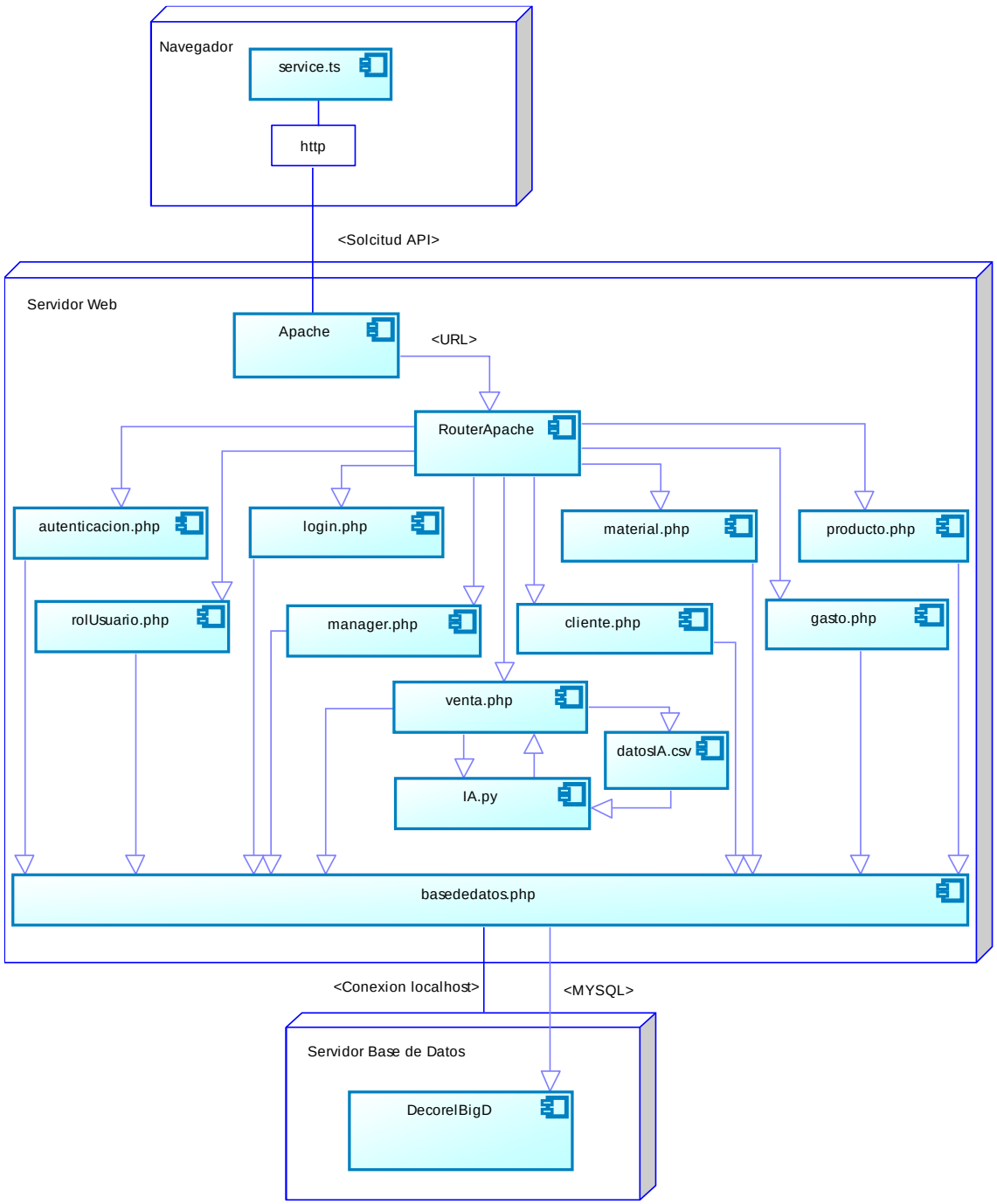
CAPITULO 7. ANEXOS

7.1 ANEXO 1: Diagramas y Modelos

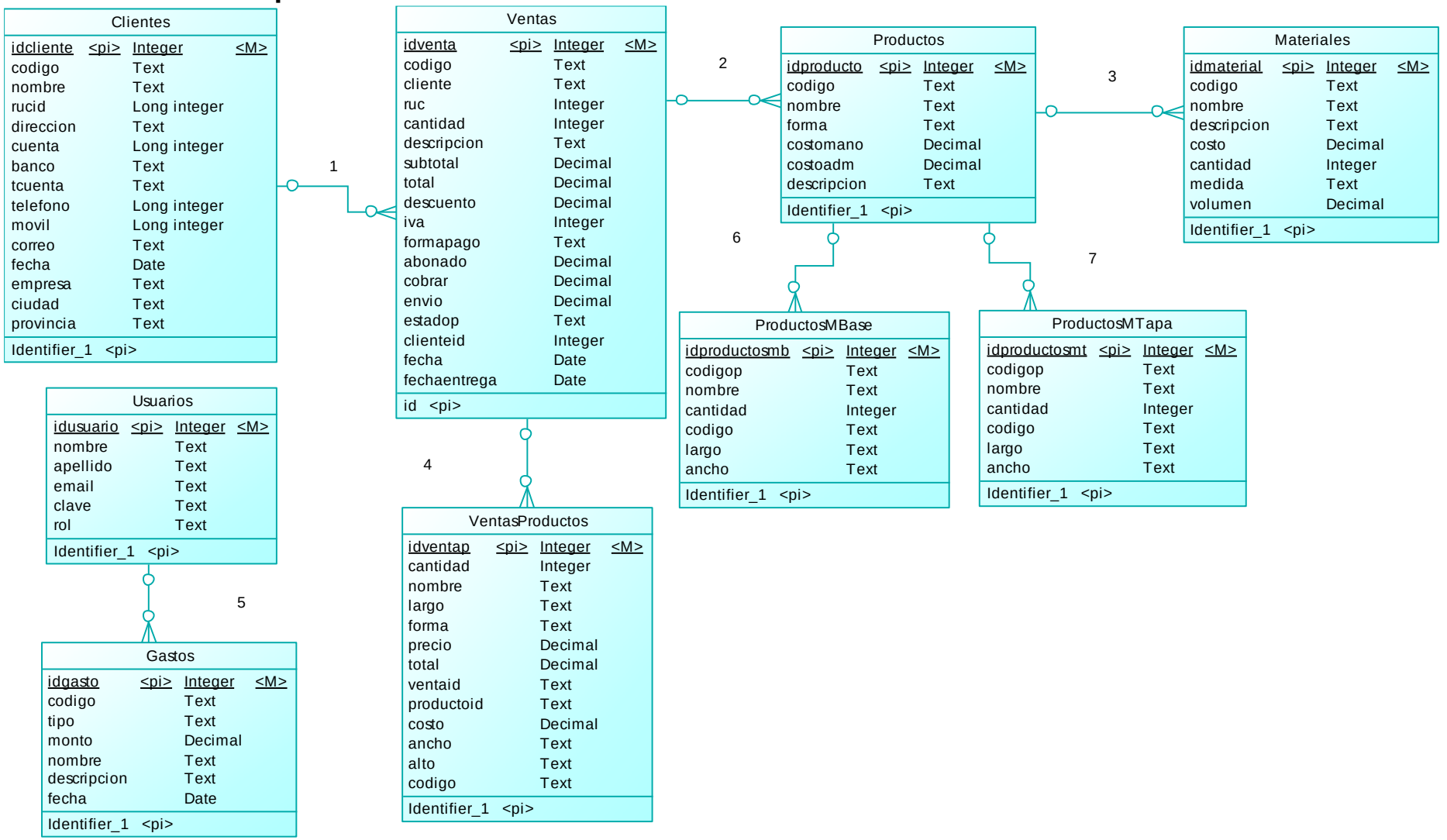
7.1.1 Diagrama de Despliegue de Componentes Front-End



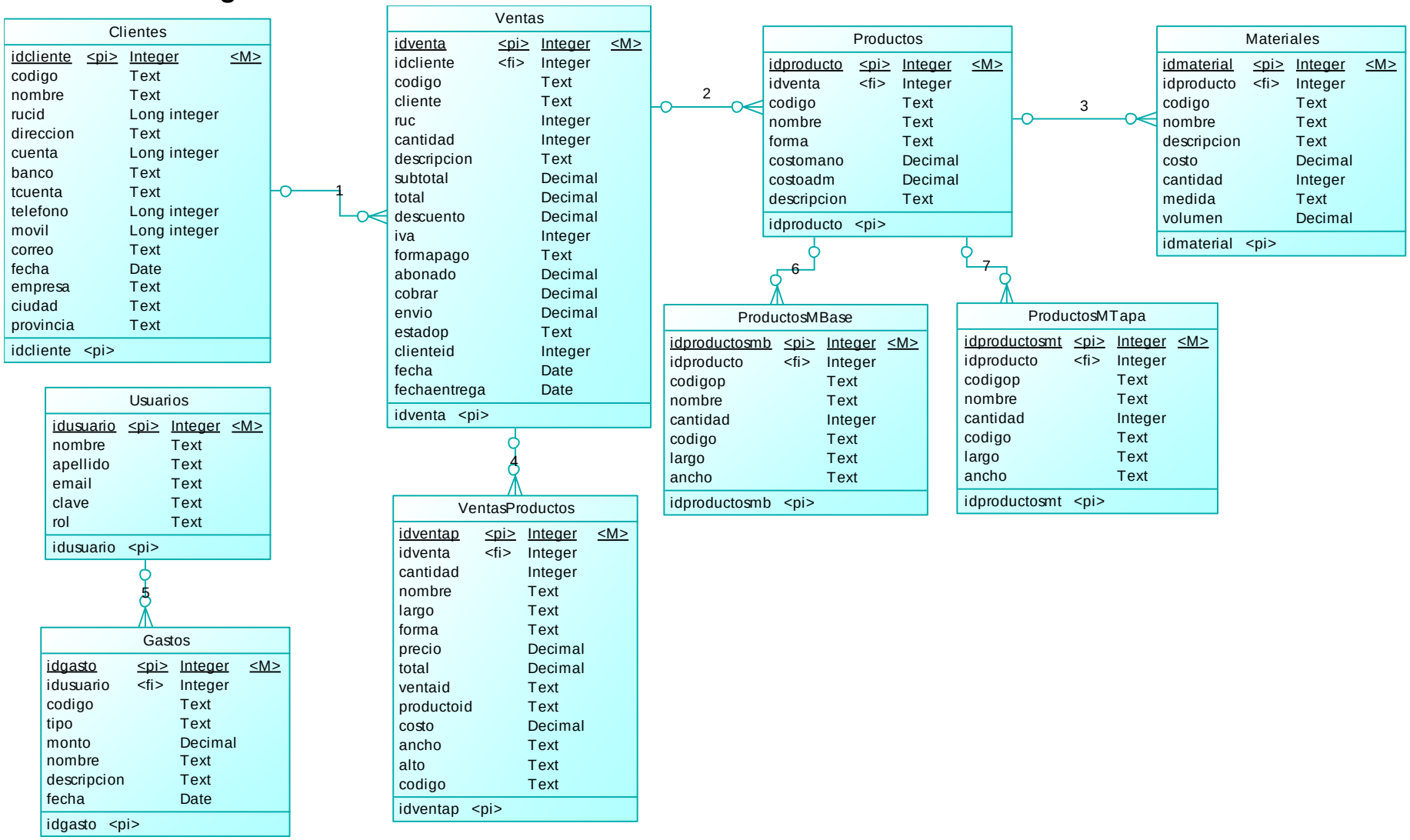
7.1.2 Diagrama de Despliegue de Componentes Back-End



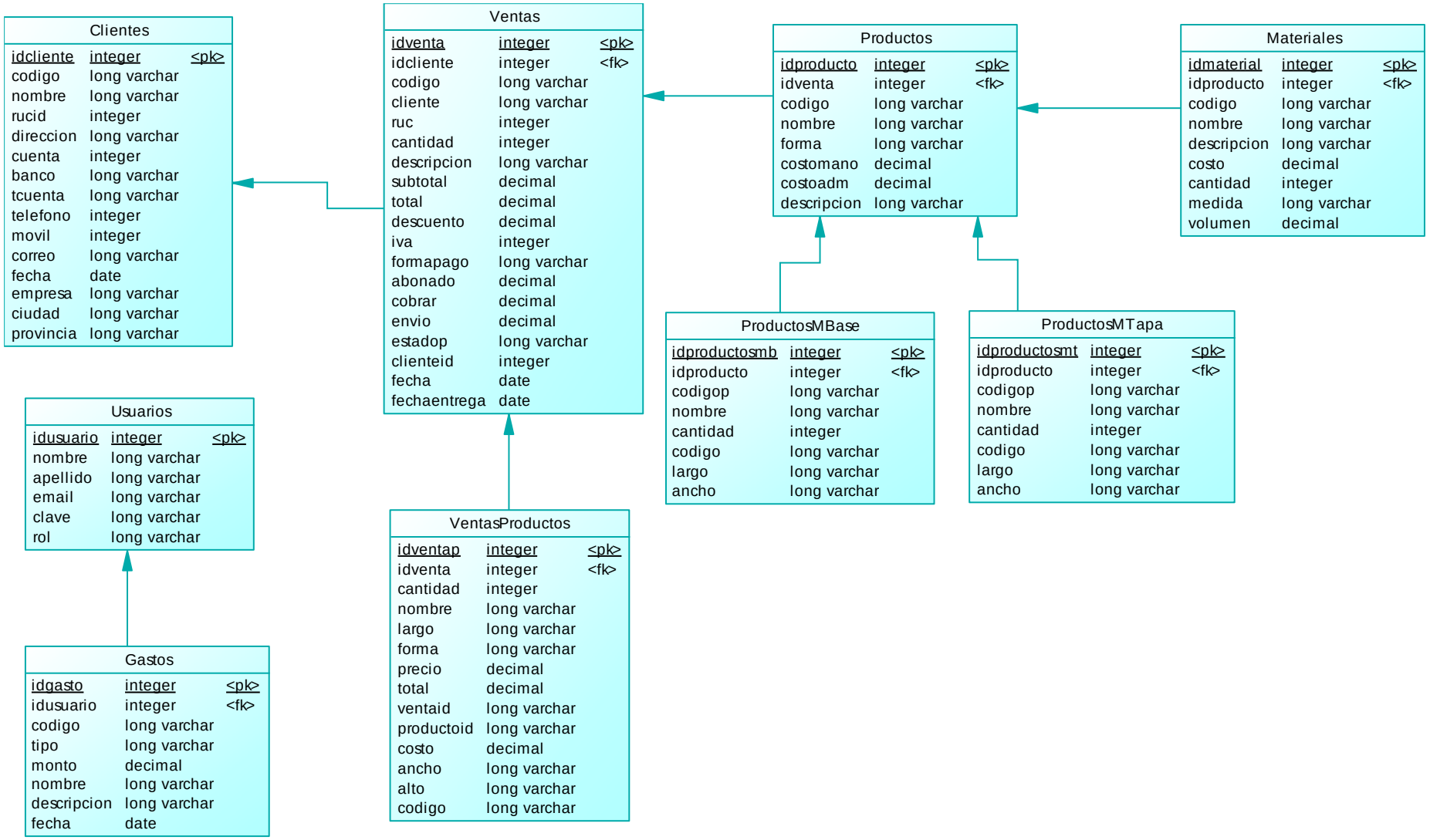
7.1.3 Modelo Conceptual



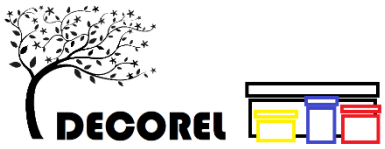
7.1.4 Modelo Lógico



7.1.4 Modelo Físico



7.2 ANEXO 2: Manual de usuario



Decorel Manager

Manual de Usuario

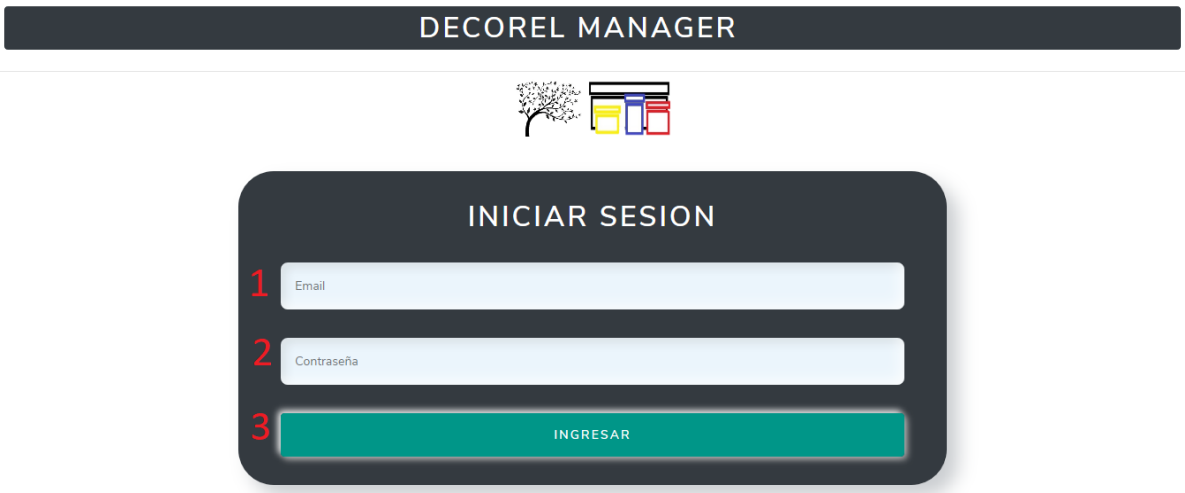
Contenido

Manual de Usuario 77

1. Ingreso al Sistema 79
2. Pantalla principal del sistema (Manager) 79
3. Sección Materiales 82
   - 3.1 Pantalla Principal De Materiales 82
   - 3.2 Agregar un Material 83
   - 3.3 Editar un Material 84
4. Sección Productos 84
   - 4.1 Pantalla principal de Productos 84
   - 4.2 Agregar un Producto 85
   - 4.3 Editar un Producto 88
5. Sección Calculadora 89
6. Sección Ventas 93
   - 6.1 Pantalla principal de Ventas 93
   - 6.2 Agregar una Venta 94
7. Sección Clientes 98
   - 7.1 Pantalla Principal De Clientes 98
   - 7.2 Agregar un Cliente 99
   - 7.3 Editar un Cliente 100
8. Sección Gastos 101
   - 8.1 Pantalla Principal De Gastos 101
   - 8.2 Agregar un Gasto 102
   - 8.3 Editar un Gasto 103

1. Ingreso al Sistema

Al iniciar el sistema, el usuario podrá visualizar la siguiente pantalla:

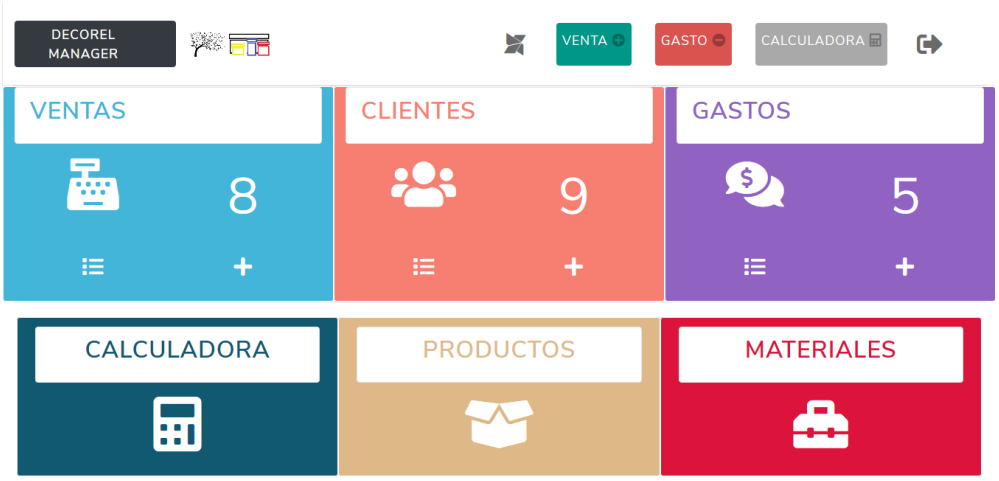


1. Como primero el usuario debe ingresar su correo electrónico con el que el administrador lo ha registrado en el sistema.
2. El usuario debe colocar la contraseña asignada por el administrador.
3. Por último debe dar clic en el botón Ingresar.

Si los datos están correctos será dirigido a la pantalla manager del sistema, caso contrario el usuario recibirá una alerta con la razón o error por el fallo de inicio de sesión.

2. Pantalla principal del sistema (Manager)

Después de iniciar sesión exitosamente en el sistema el usuario podrá observar la pantalla principal de Decorel Manager, donde podrá acceder a varias funcionalidades que ofrece este programa como se observa a continuación:



Barra superior

En la parte superior se encuentra una barra de navegación que esta activa en todo el sistema, en ella podemos encontrar 5 opciones:

- Primero se encuentra un botón home que nos llevara directo a la página principal, no importa en la sección que nos encontremos del sistema.
- Botón **VENTA+** que al dar clic sobre él nos redirigirá a la sección para agregar una nueva venta de la microempresa en el sistema.
- Botón **GASTO-** que al dar clic sobre él nos redirigirá a la sección para agregar un nuevo gasto realizado dentro de la microempresa en el sistema.
- Botón **CALCULADORA** que al dar clic sobre él nos redirigirá a la sección para calcular los costos de producción y precios de venta de un producto de la microempresa.
- Por último esta un botón para cerrar sesión y salir del sistema Decorel Manager.

Página Principal

En la pantalla principal podremos observar 6 recuadros:



En el recuadro llamado **VENTAS** podrá observar el número de ventas, seguido tenemos dos botones:

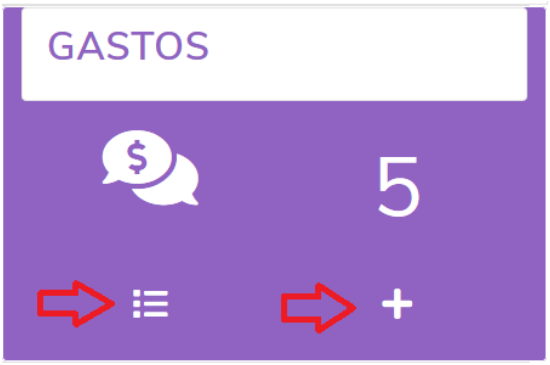
- En el primer botón podrá acceder a la pantalla para administrar y consultar todas las ventas de la microempresa en el sistema.
- Si da clic en el segundo botón, será redirigido a la pantalla para agregar una nueva venta en el sistema.



En el recuadro llamado **CLIENTES** podrá observar el número de clientes, seguido tenemos dos botones:

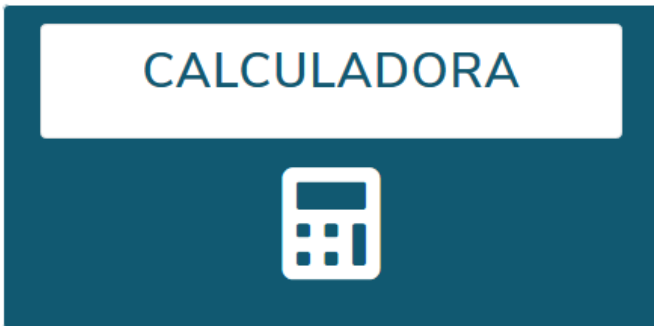
- En el primer botón podrá acceder a la pantalla para administrar y consultar todos los clientes de la microempresa en el sistema.

- Si da clic en el segundo botón, será redirigido a la pantalla para agregar un nuevo cliente en el sistema.



En el recuadro llamado **GASTOS** podrá observar el número de gastos, seguido tenemos dos botones:

- En el primer botón podrá acceder a la pantalla para administrar y consultar todos los gastos de la microempresa en el sistema.
- Si da clic en el segundo botón, será redirigido a la pantalla para agregar un nuevo cliente en el sistema.



En el recuadro llamado **CALCULADORA** podrá acceder a la sección del sistema para calcular los costos de producción y precios de venta de un **producto** en específico mediante el uso de **materiales** previamente registrados por el usuario en el sistema.



En el recuadro llamado **PRODUCTOS** podrá acceder a la sección del sistema para ingresar un nuevo **producto** que hace uso de **materiales** específicos previamente registrados en la sección **MATERIALES** por el usuario en el sistema.

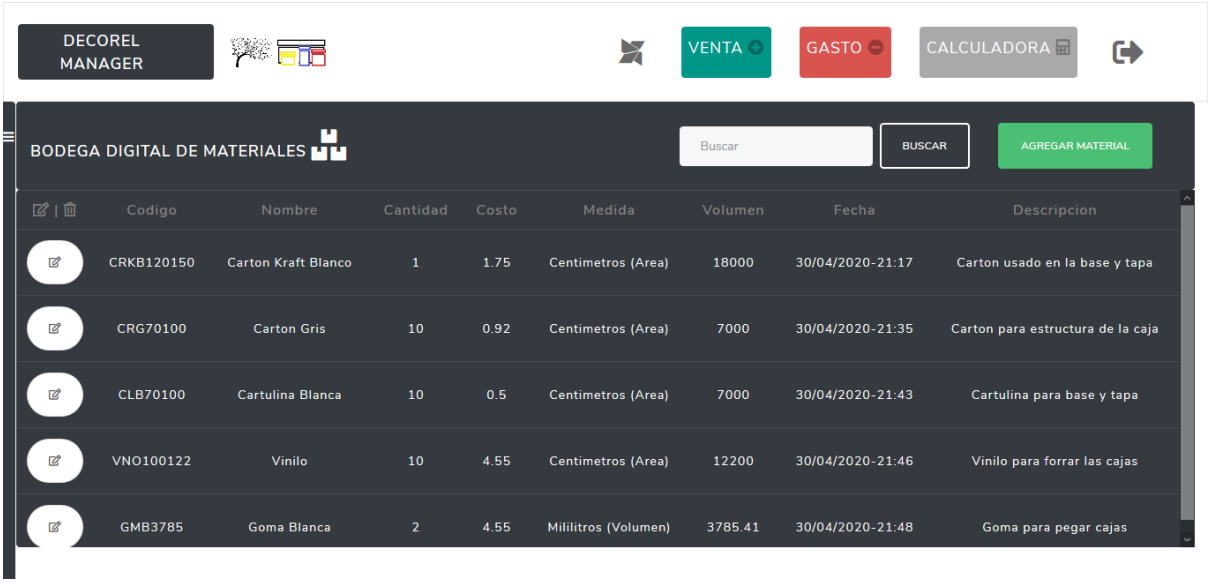


En el recuadro llamado **MATERIALES** podrá acceder a la sección del sistema para ingresar un nuevo **material** que son necesarios para la creación de un nuevo **producto** en el sistema.

3. Sección Materiales

- ### 3.1 Pantalla Principal De Materiales

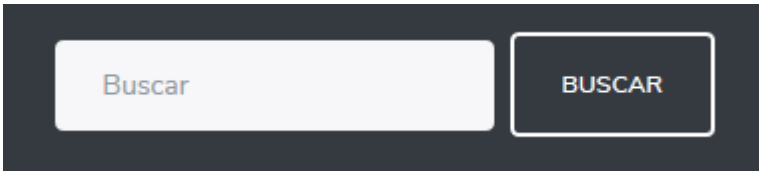
Para acceder a esta sección del sistema puede hacerlo desde la página principal en el recuadro **MATERIALES** o desde la barra lateral del sistema.



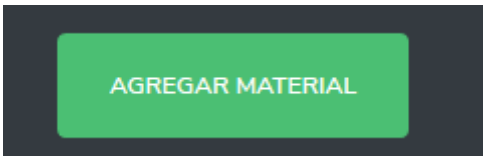
En esta sección podrá encontrar las herramientas necesarias para crear, modificar, eliminar o consultar un **material**.

Importante: Recuerde que es necesario la creación de materiales para crear un nuevo producto en la sección **PRODUCTOS.**

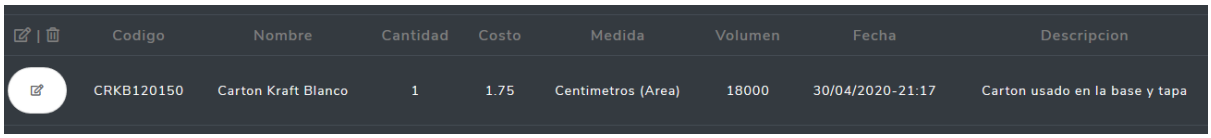
Comenzando por la parte superior el usuario podrá observar una barra de búsqueda, donde puede ingresar el código o nombre de un material en específico y el sistema lo filtrará en la lista.



A un lado de la barra de búsqueda tenemos el botón **AGREGAR MATERIAL** que al dar clic el sistema abrirá una nueva ventana para crear un nuevo material y almacenarlo en el sistema.



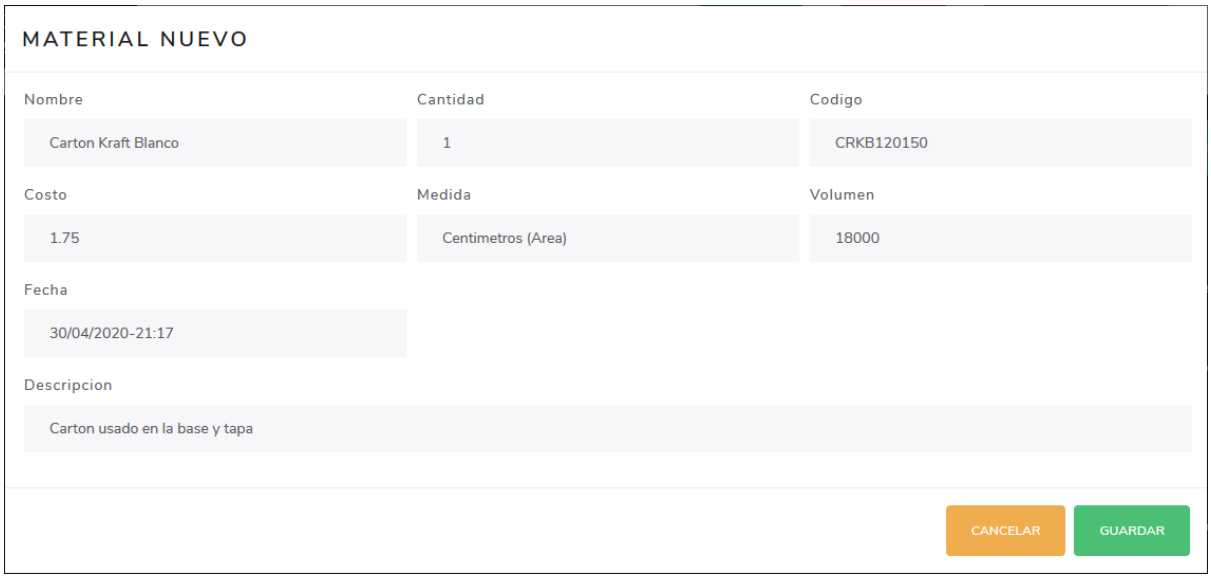
Por último tenemos una lista de todos los materiales creados en el sistema, al lado izquierdo de cada material puede observar un botón que le permitirá editar ese material en específico.



	Codigo	Nombre	Cantidad	Costo	Medida	Volumen	Fecha	Descripcion
	CRKB120150	Carton Kraft Blanco	1	1.75	Centimetros (Area)	18000	30/04/2020-21:17	Carton usado en la base y tapa

3.2 Agregar un Material

Para acceder a esta sección del sistema puede hacerlo desde la página principal en el recuadro **MATERIALES** o desde la sección materiales dando clic en el botón **AGREGAR MATERIAL**.



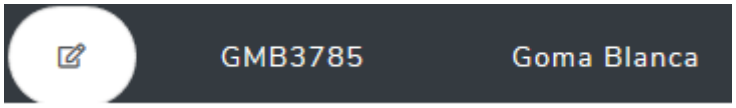
Se desplegará un formulario donde debemos ingresar todos los datos necesarios para crear un material, donde el nombre, código, costo y valor son obligatorios.

Importante: El usuario debe tener en cuenta que el sistema usa la medida centímetros y que el volumen que tiene un material debe ser ingresado como área de este para que el sistema funcione correctamente, por ejemplo como podemos observar en la imagen si una plancha de cartón que se compra por 1.75 la unidad y sus medidas son largo:150 centímetros y su ancho: 120 centímetros, el usuario debe calcular el área multiplicando estas dos medidas (área = largo * ancho) dando como resultado área =18000 centímetros e ingresarlo al sistema.

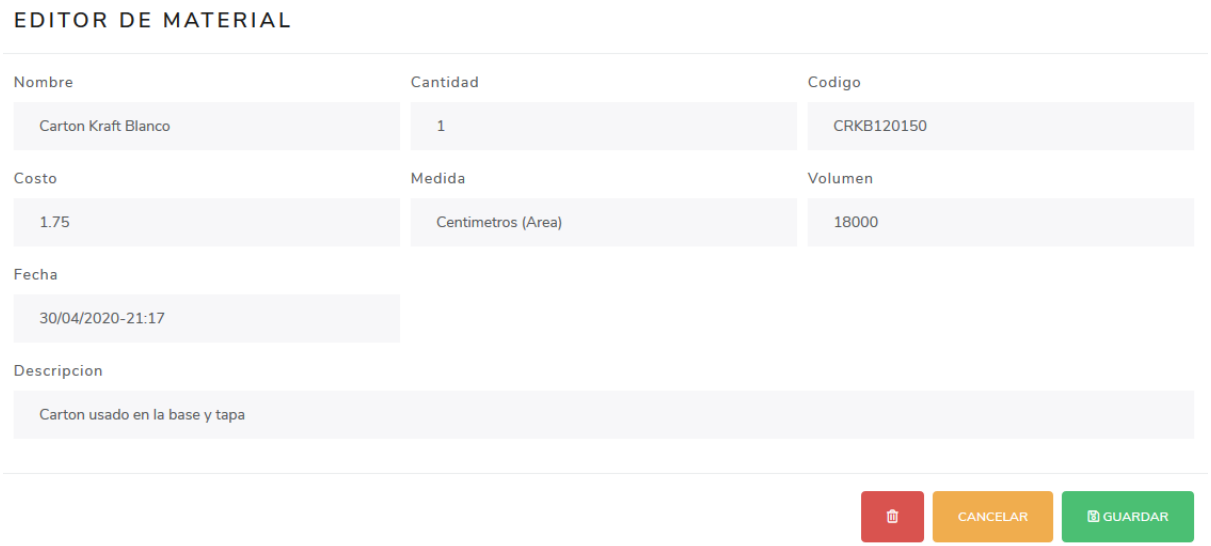
Por último si todos los datos son correctamente ingresados, el usuario puede dar clic en el botón **GUARDAR** y el sistema almacenara el nuevo registro en la base de datos. Caso contrario puede dar clic en **CANCELAR** y esto abortara la operación.

3.3Editar un Material

Para poder editar un material, el usuario debe acceder a la sección **MATERIALES** donde se listan todos los materiales ingresados en el sistema, al lado izquierdo de cada material podrá observar un botón de edición:



Al dar clic sobre este botón, se desplegará un formulario con los datos del material que desea modificar:



En la parte inferior, encontrara un botón rojo que permite **ELIMINAR** este material si el usuario así lo prefiere, caso contrario puede modificar los datos que sean necesarios y dar clic sobre el botón **GUARDAR** para almacenar los cambios en el sistema.

4. Sección Productos

- 4.1 Pantalla principal de Productos

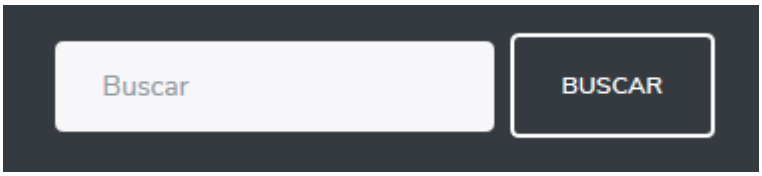
Para acceder a esta sección del sistema puede hacerlo desde la página principal en el recuadro **PRODUCTOS** o desde la barra lateral del sistema.



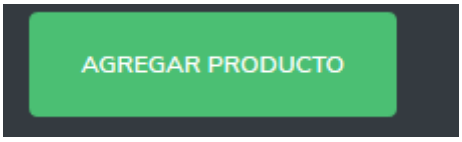
En esta sección podrá encontrar las herramientas necesarias para crear, modificar, eliminar o consultar un **producto**.

Importante: Recuerde que es necesario la creación de productos para hacer uso de la sección **CALCULADORA** o la sección **AGREGAR VENTA.**

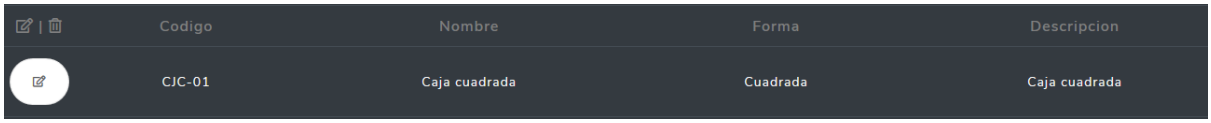
Comenzando por la parte superior el usuario podrá observar una barra de búsqueda, donde puede ingresar el código o nombre de un producto en específico y el sistema lo filtrará en la lista.



A un lado de la barra de búsqueda tenemos el botón **AGREGAR PRODUCTO** que al dar clic el sistema abrirá una nueva ventana para crear un nuevo producto y almacenarlo en el sistema.

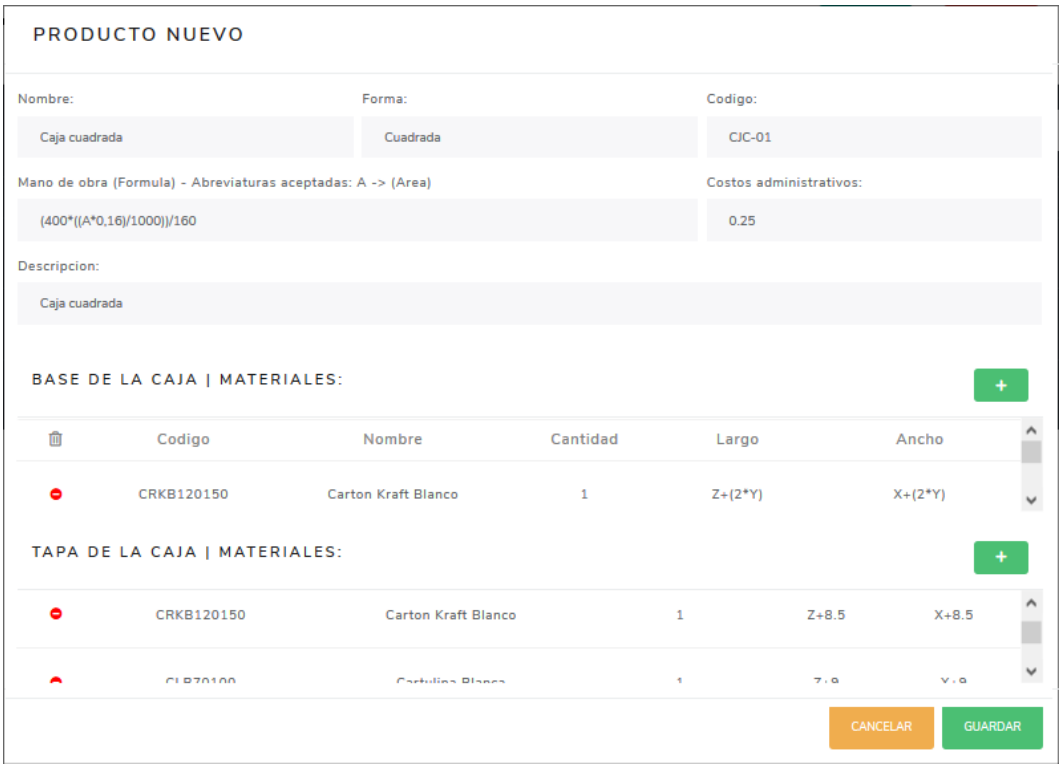


Por último tenemos una lista de todos los productos creados en el sistema, al lado izquierdo de cada producto puede observar un botón que le permitirá editar ese producto en específico.



4.2 Agregar un Producto

Para acceder a esta sección del sistema puede hacerlo desde la página principal en el recuadro **PRODUCTOS** o desde la sección productos dando clic en el botón **AGREGAR MATERIAL**.



Se desplegará un formulario donde debemos ingresar los datos necesarios para crear un producto, donde todos los campos son obligatorios y algunos de ellos son usados por el sistema para su correcto funcionamiento los cuales se explicarán con detalle a continuación:

Mano de obra: En este dato del formulario se debe ingresar una fórmula que calcula el costo de la mano de obra empleado para dicho producto, esta fórmula se basa en el salario que gana mensualmente el empleado, el tiempo que trabaja mensualmente, **el área del producto que fabricara (este dato debe ingresarse con la letra A ya que el sistema se encarga de calcularlo por sí mismo)** y el tiempo que le toma hacer este producto, aunque esta fórmula es proporcionada por la microempresa el sistema le da la posibilidad de modificarla cuando el usuario así lo requiera.

Costos administrativos: En este campo se debe ingresar el costo especifico que se requiere para administrar la creación, movilización y promoción del producto que se está generando en el sistema.

Base de la caja | Materiales: En esta sección el usuario será capaz de agregar los materiales necesarios para fabricar la base del producto en sí, se mostrará una lista con todos los materiales seleccionados, si el usuario desea agregar un nuevo material debe dar clic en el botón **+,** inmediatamente el sistema desplegará una pantalla como la siguiente:



Aquí el usuario podrá elegir un **material** que ha sido ingresado previamente en el sistema en la sección **MATERIALES** , seguido de la cantidad de veces que el producto requiere este material.

En la sección **FORMULA** el usuario deberá ingresar dos fórmulas una para calcular el largo que se usa de este material y otra para calcular el ancho que se usa de este material en el producto que el usuario está registrando, estas fórmulas se basan en el largo, ancho y altura del producto, como estas medidas pueden variar entonces el sistema las toma como letras **X(ancho), Z(largo), Y(altura)** para crear un cálculo dinámico, aunque estas fórmulas son proporcionadas por la microempresa, el usuario puede modificarlas en cualquier momento.

Después de seleccionar un material e ingresar correctamente sus fórmulas, debe dar clic en el botón con un visto verde para asociar este material al producto y agregarlo a lista como se muestra en la siguiente imagen:



Tapa de la caja | Materiales: En esta sección el usuario será capaz de agregar los materiales necesarios para fabricar la tapa del producto en sí, se mostrará una lista con todos los materiales seleccionados, si el usuario desea agregar un nuevo material debe dar clic en el botón **+,** inmediatamente el sistema desplegará una pantalla como la siguiente:

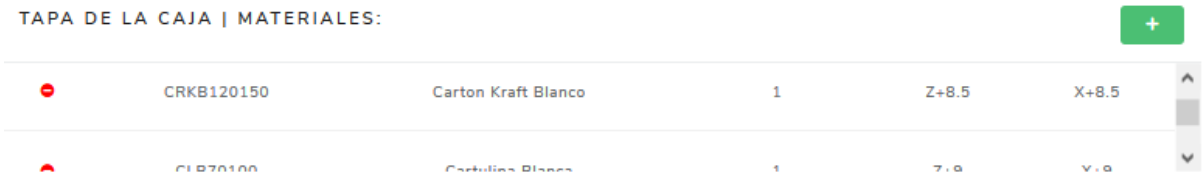


Aquí el usuario podrá elegir un **material** que ha sido ingresado previamente en el sistema en la sección **MATERIALES** , seguido de la cantidad de veces que el producto requiere este material.

En la sección **FORMULA** el usuario deberá ingresar dos fórmulas una para calcular el largo que se usa de este material y otra para calcular el ancho que se usa de este material en el producto que el usuario está registrando, estas fórmulas se basan en el largo, ancho y altura del producto,

como estas medidas pueden variar entonces el sistema las toma como letras **X(ancho), Z(largo), Y(altura)** para crear un cálculo dinámico, aunque estas fórmulas son proporcionadas por la microempresa, el usuario puede modificarlas en cualquier momento.

Después de seleccionar un material e ingresar correctamente sus fórmulas, debe dar clic en el botón con un visto verde para asociar este material al producto y agregarlo a lista como se muestra en la siguiente imagen:



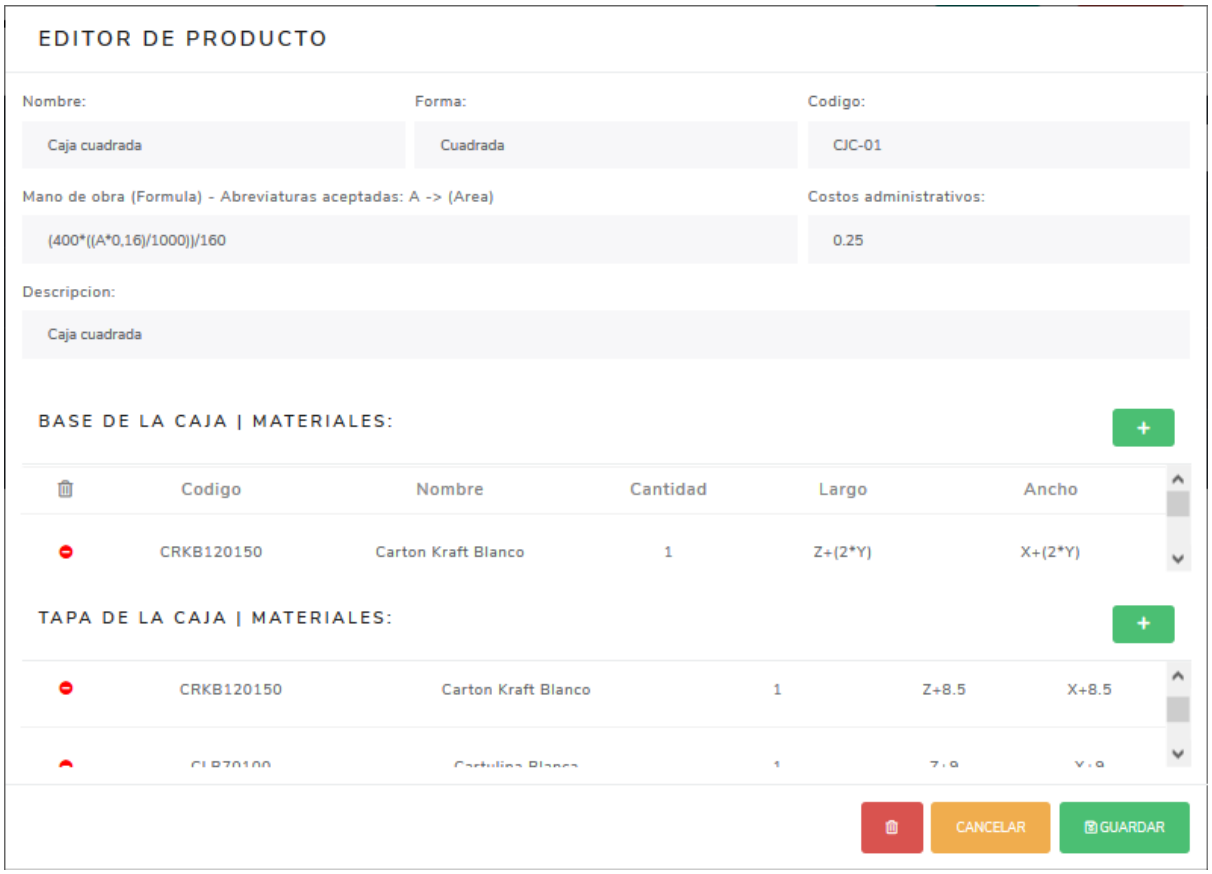
Por último si todos los datos son correctamente ingresados y los materiales necesarios para crear el producto seleccionados, el usuario puede dar clic en el botón **GUARDAR** y el sistema almacenara el nuevo producto en la base de datos. Caso contrario puede dar clic en **CANCELAR** y esto abortara la operación.

4.3Editar un Producto

Para poder editar un producto, el usuario debe acceder a la sección **PRODUCTOS** donde se listan todos los productos ingresados en el sistema, al lado izquierdo de cada producto podrá observar un botón de edición:



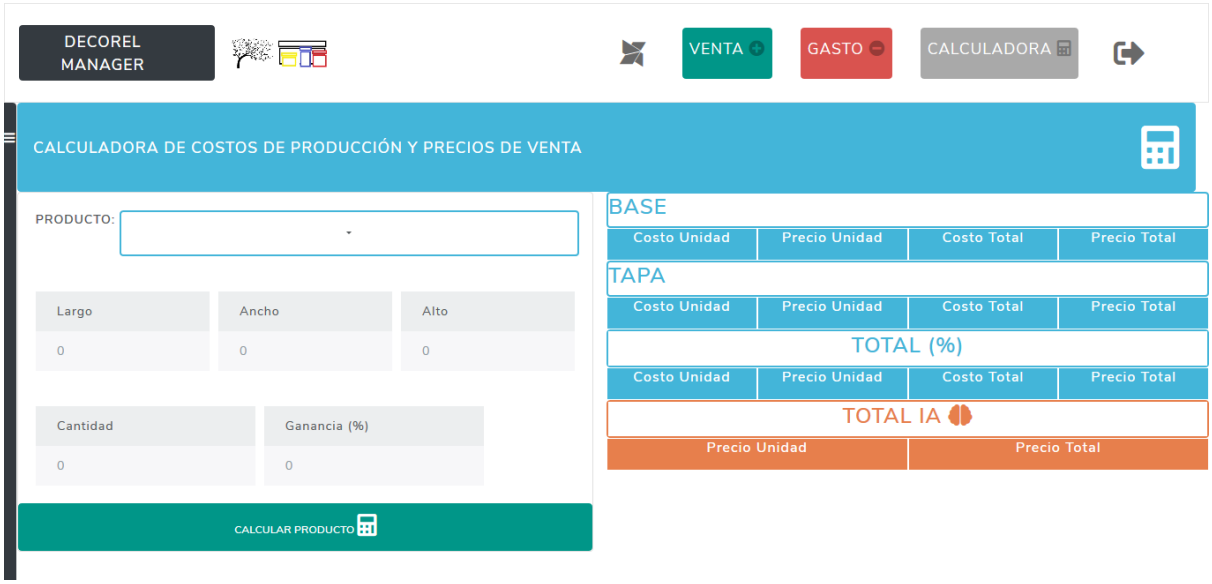
Al dar clic sobre este botón, se desplegará un formulario con los datos del producto que desea modificar:



En la parte inferior, se encuentra un botón rojo que permite **ELIMINAR** este producto si el usuario así lo prefiere, caso contrario puede modificar los datos que sean necesarios y dar clic sobre el botón **GUARDAR** para almacenar los cambios en el sistema.

5. Sección Calculadora

Para acceder a esta sección del sistema puede hacerlo desde la página principal dando clic en el recuadro **CALCULADORA** o desde la barra superior del sistema.

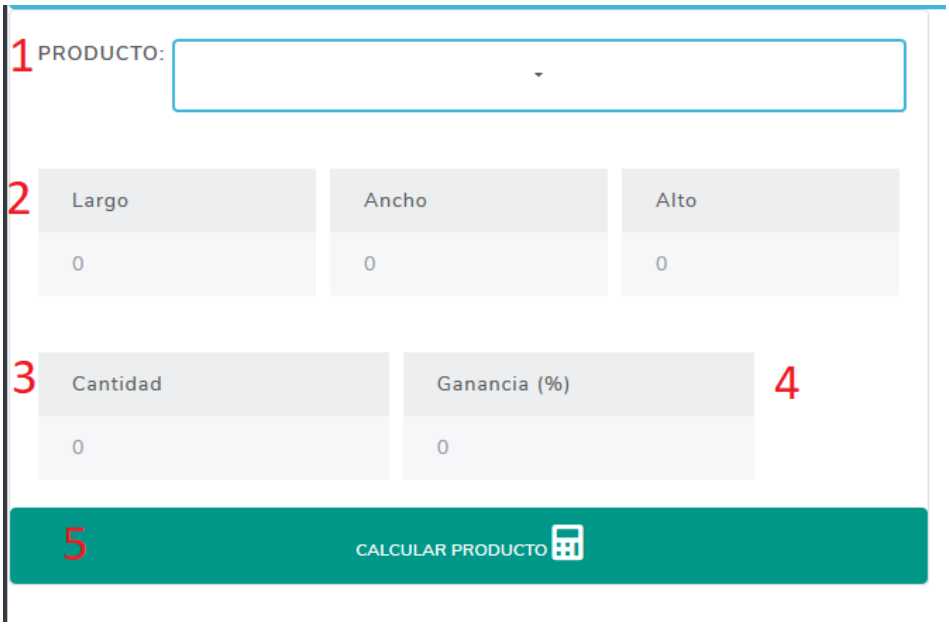


Esta sección del sistema está orientada al cálculo de costos de producción y precios de venta de un producto en específico ya registrado previamente en el sistema por el usuario en la sección **PRODUCTOS.**

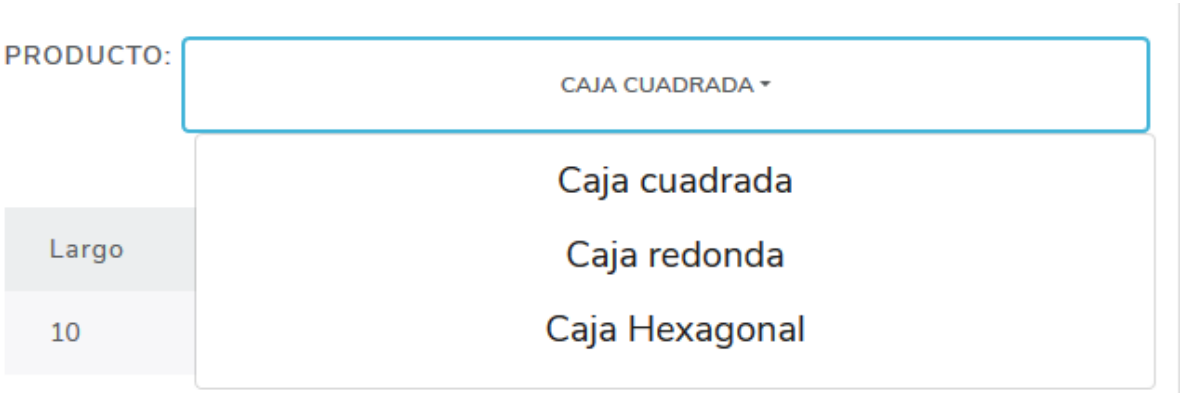
Modo de uso:

La pantalla se divide en dos secciones, a la izquierda se encuentra el ingreso de datos (todos los datos son obligatorios) para calcular un producto y a la derecha se despliega la información generada por el sistema con todos los costos y precios solicitados por la microempresa.

Sección Izquierda:



1. El usuario debe dar clic en este botón y seleccionar de la lista uno de todos los **productos** (que ya fueron registrados en el sistema previamente en la sección **PRODUCTOS)** que desee calcular**.**



2. Aquí encontrara 3 cuadros donde debe ingresar el **largo**, **ancho** y **alto** del producto que desea calcular, recordar que las medidas que utiliza el sistema son en **centímetros**.

Largo	Ancho	Alto
10	10	10

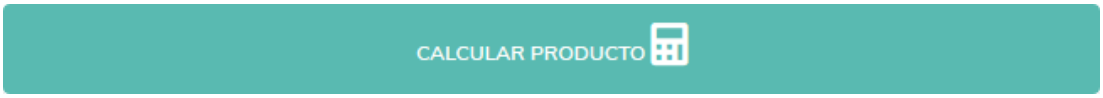
3. Debe ingresar la cantidad de productos que necesite calcular, obligatoriamente debe comenzar desde 1 para que el sistema funcione correctamente.

Cantidad
3

4. EL usuario también tiene la opción de ingresar el porcentaje de ganancia que desea calcular de dicho producto que se basa en el costo total de producción.

Ganancia (%)
50

5. Por último después de haber rellenado todos los campos necesarios, puede dar clic en el botón calcular para que el sistema realice su trabajo y despliegue en la sección derecha toda la información.



Sección derecha:



En esta sección de la pantalla se puede visualizar los cálculos realizados por el sistema dividido en cuatro partes :

Base:

- Costo de producción unitario solo para la base del producto.
- Precio de venta (mediante % de ganancia) solo para la base del producto.
- Costo de producción total de la base por la cantidad de productos solicitados.
- Precio de venta total de la base por la cantidad de productos solicitados.

Tapa:

- Costo de producción unitario solo para la tapa del producto.
- Precio de venta (mediante % de ganancia) solo para la tapa del producto.
- Costo de producción total de la tapa por la cantidad de productos solicitados.
- Precio de venta total de la tapa por la cantidad de productos solicitados.

Total:

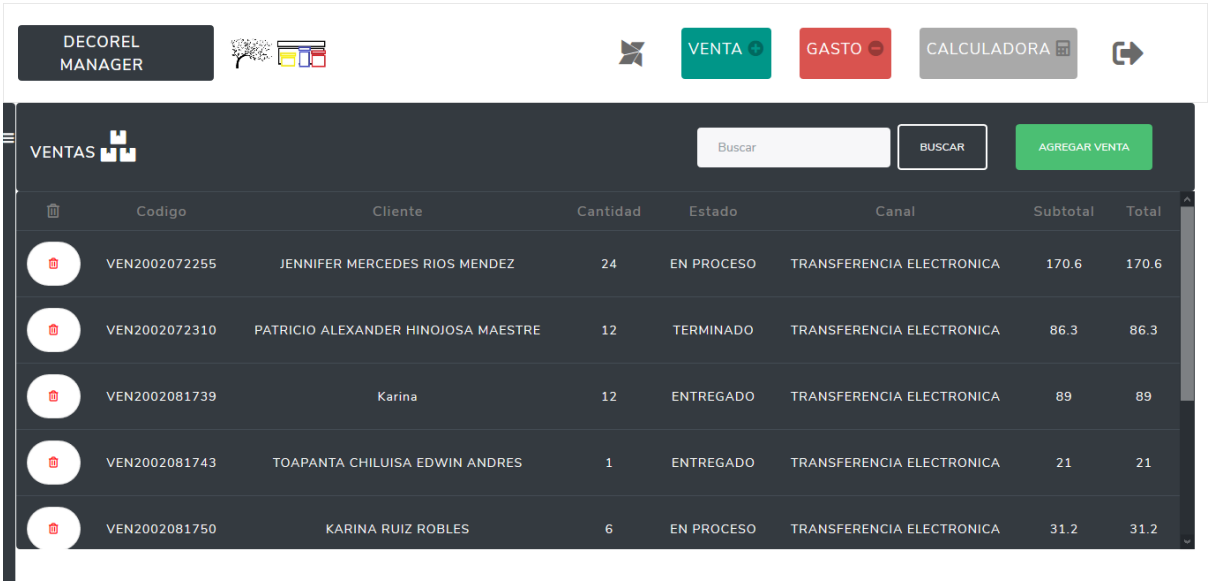
- Costo de producción unitario del producto completo.
- Precio de venta (mediante % de ganancia) del producto completo.
- Costo de producción total del producto completo por la cantidad solicitada.
- Precio de venta total del producto por la cantidad de solicitada.

Total IA (Inteligencia Artificial):

- Precio de venta unitario sugerido por el script de IA mediante la librería sklearn.
- Precio de venta total sugerido por el script de IA mediante la librería sklearn por la cantidad de solicitada.

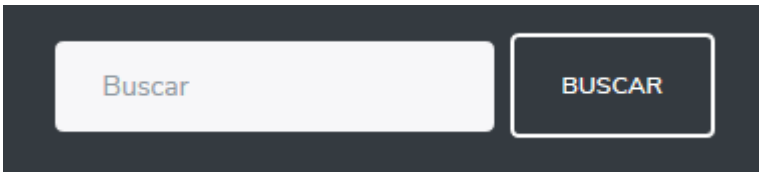
- # 6. Sección Ventas
  - ## 6.1 Pantalla principal de Ventas

Para acceder a esta sección del sistema puede hacerlo desde la página principal dando clic en el recuadro **VENTAS** o desde la barra lateral del sistema.

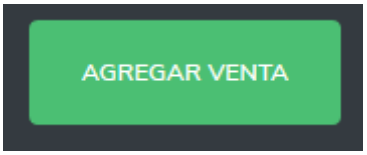


En esta sección podrá encontrar las herramientas necesarias para crear, eliminar o consultar una **venta**.

Comenzando por la parte superior el usuario podrá observar una barra de búsqueda, donde puede ingresar el código de venta, cedula o nombre de un cliente en específico y el sistema lo filtrará en la lista.



A un lado de la barra de búsqueda tenemos el botón **AGREGAR VENTA** que al dar clic el sistema abrirá una nueva ventana para registrar una nueva venta y almacenarla en el sistema.

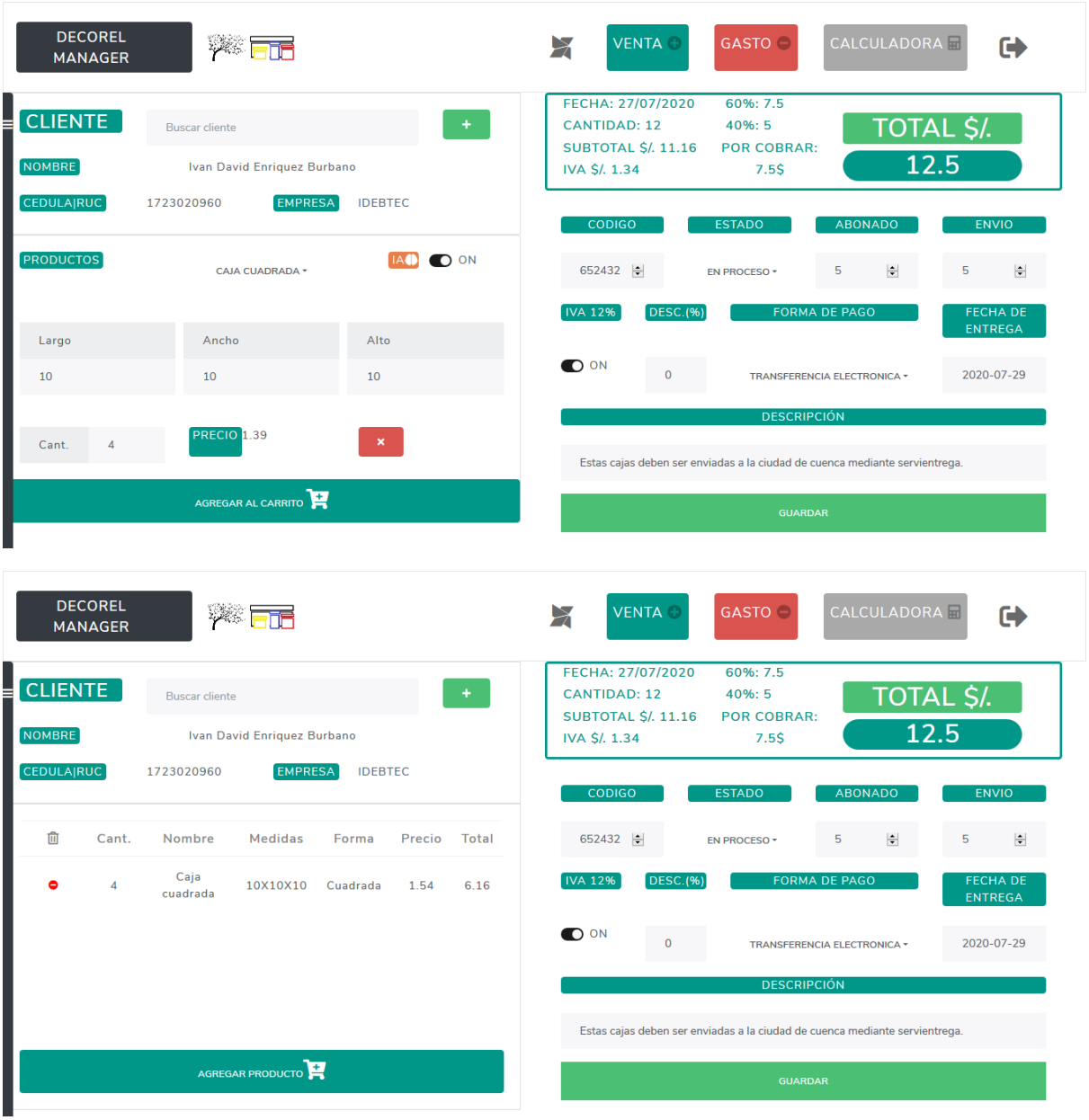


Por último tenemos una lista de todas las ventas registradas en el sistema, al lado izquierdo de cada venta puede observar un botón que le permitirá eliminar esa venta en específico.



- 6.2 Agregar una Venta

Para acceder a esta sección del sistema puede hacerlo desde la página principal en el recuadro **VENTAS,** desde la sección ventas dando clic en el botón **AGREGAR VENTA** o desde la barra superior del sistema dando clic en **VENTA+**.

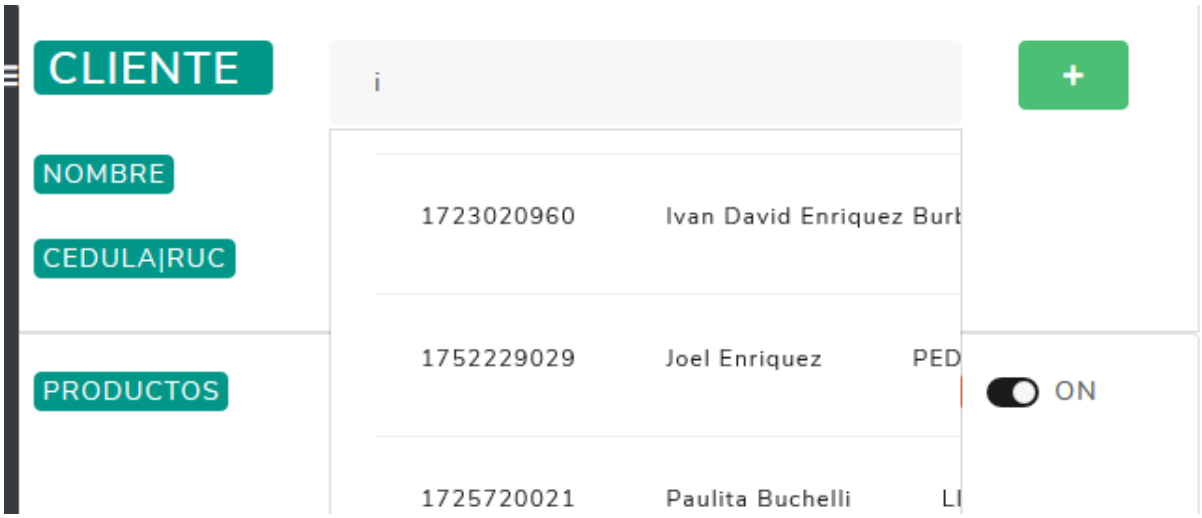


Modo de uso:

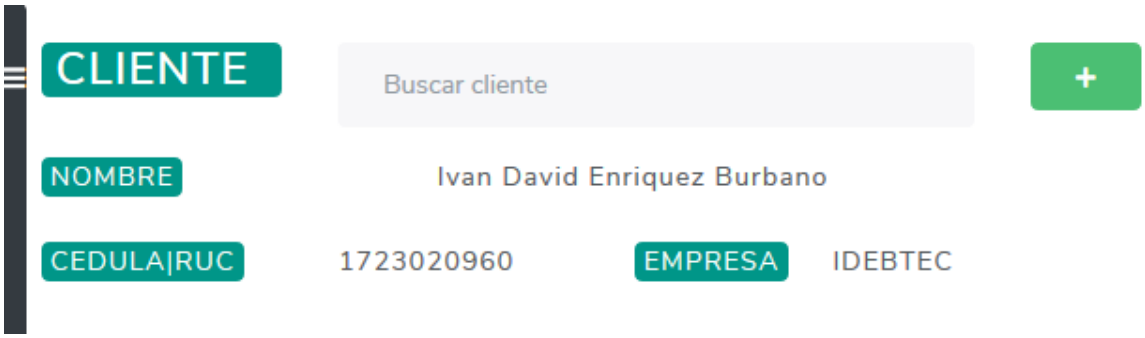
En esta sección del sistema, el usuario tendrá la capacidad para generar y almacenar una nueva venta generada por la microempresa, esta sección esta divida en 4 partes como podemos observar en la imagen anterior, la de cliente, la de producto, los valores generados por el sistema y los datos de la venta.

Sección Cliente:

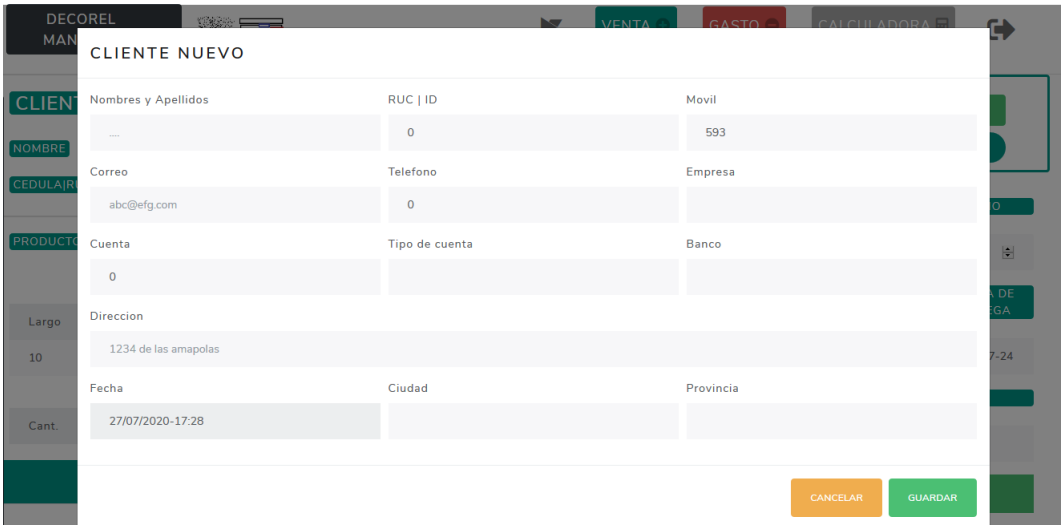
En esta sección el usuario puede buscar un cliente por su nombre o número de cedula, el sistema desplegara una lista de coincidencias como se muestra continuación:



Una vez que encuentre una coincidencia de clic sobre el nombre del cliente, el sistema tomara la información y la asociara a esta nueva venta que está generando como se muestra a continuación:



En el caso de no encontrar el cliente, el usuario puede ingresarlo dando clic en el botón **+** que se encuentra a un lado de la barra de búsqueda, el sistema abrirá un formulario para poder agregar un nuevo cliente en el sistema como se muestra a continuación:



Sección Producto:

En esta sección puede calcular y agregar los productos que la microempresa está vendiendo a un cliente en específico y los asociara a esta venta.

Importante: Esta sección del sistema está orientada al cálculo de costos de producción y precios de venta de un producto en específico ya registrado previamente en el sistema por el usuario en la sección **PRODUCTOS.**



Modo de uso:

1. Primero debe elegir el producto que desea calcular y agregar a la lista de venta.
2. Segundo tiene la opción de activar o desactivar la **inteligencia artificial** que se encarga de calcular el precio de venta en base al costo de producción del producto mediante la librería **sklearn**, en el caso de que decida desactivarlo el sistema calculara el precio de venta con el **50% de ganancia** del costo de producción.
3. Aquí debe ingresar las medidas **largo**, **ancho** y **altura** del producto que desea calcular.
4. Ingrese la cantidad de productos que desea calcular y agregar a la lista de venta.
5. El sistema le dará una previsualización del precio de venta unitario del producto que está calculando
6. Una vez ingresado todos los datos necesarios puede dar clic sobre el botón verde con una calculadora, para que el sistema calcule el costo de producción y el precio de venta del producto seleccionado
7. Si todo ha salido correcto el botón **AGERGAR AL CARRITO** se habilitará y podrá dar clic sobre el agregando el producto calculado a la lisa de la venta como se muestra a continuación:

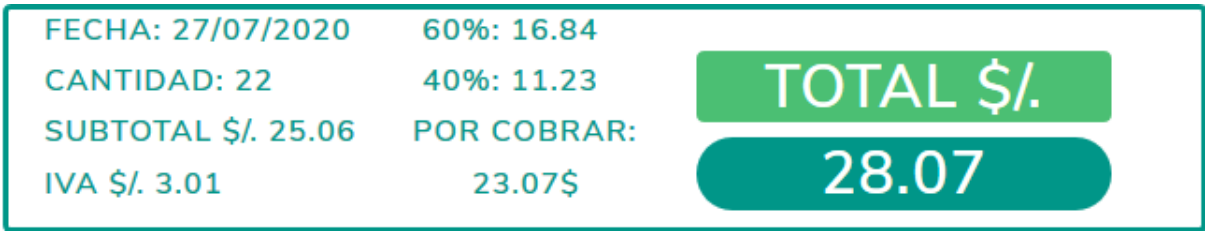


En el caso de que desee calcular y agregar un nuevo producto a la lista, debe dar clic sobre el botón **AGREGAR PRODUCTO** que se encuentra debajo de la lista.

Sección datos de la venta

Al lado izquierda la pantalla podremos observar en la parte superior los datos que el sistema calcula automáticamente y los asocia la venta:

- Fecha actual
- Subtotal
- IVA,
- 60% el costo total
- 40% de costo total
- Valor que falta por cobrar
- Total de la venta.



Debajo tenemos una serie de datos que el usuario debe ingresar para completar la venta como:

- Código de la venta proporcionado por la microempresa
- Estado del pedido
- Abono del cliente por adelantado
- Costo de envío
- Opción para activar o no el IVA en la venta
- Opción para agregar un porcentaje de descuento
- Forma de pago
- Fecha de entrega del pedido

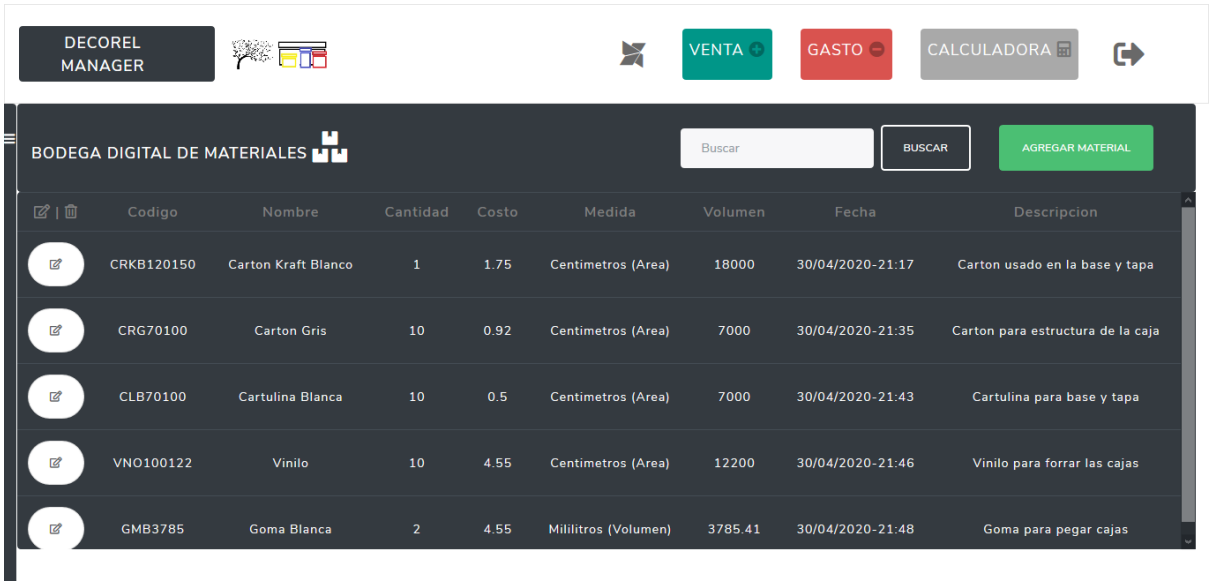
- Descripción de la venta



Al ingresar todos los datos del cliente, los productos a la lista y los datos de venta necesaria puede dar clic sobre el botón **GUARDAR**, y el sistema almacenara el nuevo registro en la base datos y lo redireccionara la sección **VENTAS**.

- 7. Sección Clientes
  - 7.1 Pantalla Principal De Clientes

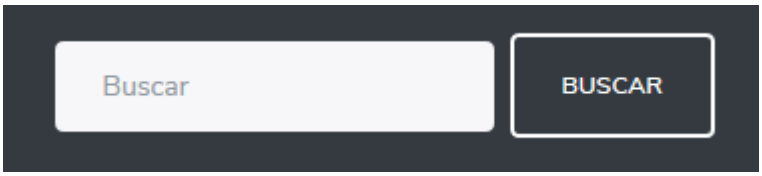
Para acceder a esta sección del sistema puede hacerlo desde la página principal en el recuadro **CLIENTES** o desde la barra lateral del sistema.



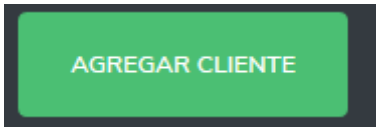
En esta sección podrá encontrar las herramientas necesarias para crear, modificar, eliminar o consultar un **cliente**.

Importante: Recuerde que es necesario la creación de clientes para crear una nueva venta en la sección **AGREGAR VENTA.**

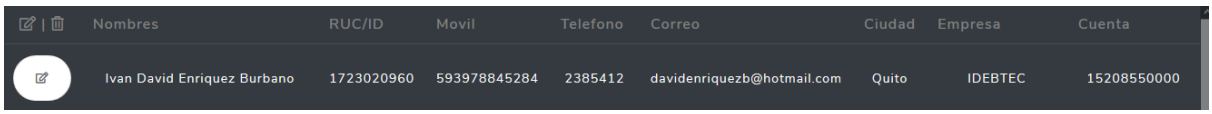
Comenzando por la parte superior el usuario podrá observar una barra de búsqueda, donde puede ingresar la cedula, ruc o nombre de un cliente en específico y el sistema lo filtrará en la lista.



A un lado de la barra de búsqueda tenemos el botón **AGREGAR CLIENTE** que al dar clic el sistema abrirá una nueva ventana para crear un nuevo cliente y almacenarlo en el sistema.



Por último tenemos una lista de todos los clientes creados en el sistema, al lado izquierdo de cada cliente puede observar un botón que le permitirá editar ese cliente en específico.



7.2 Agregar un Cliente

Para acceder a esta sección del sistema puede hacerlo desde la página principal en el recuadro **CLIENTES** o desde la sección clientes dando clic en el botón **AGREGAR CLIENTE**.

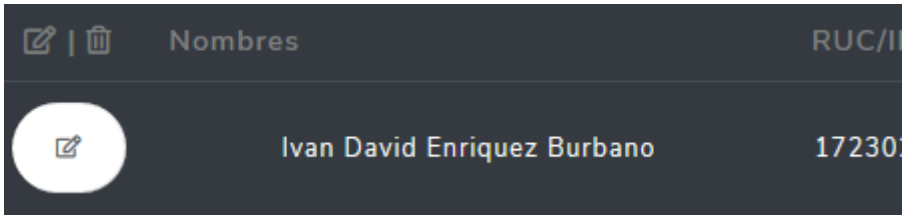


Se desplegará un formulario donde debemos ingresar todos los datos necesarios para agregar un cliente.

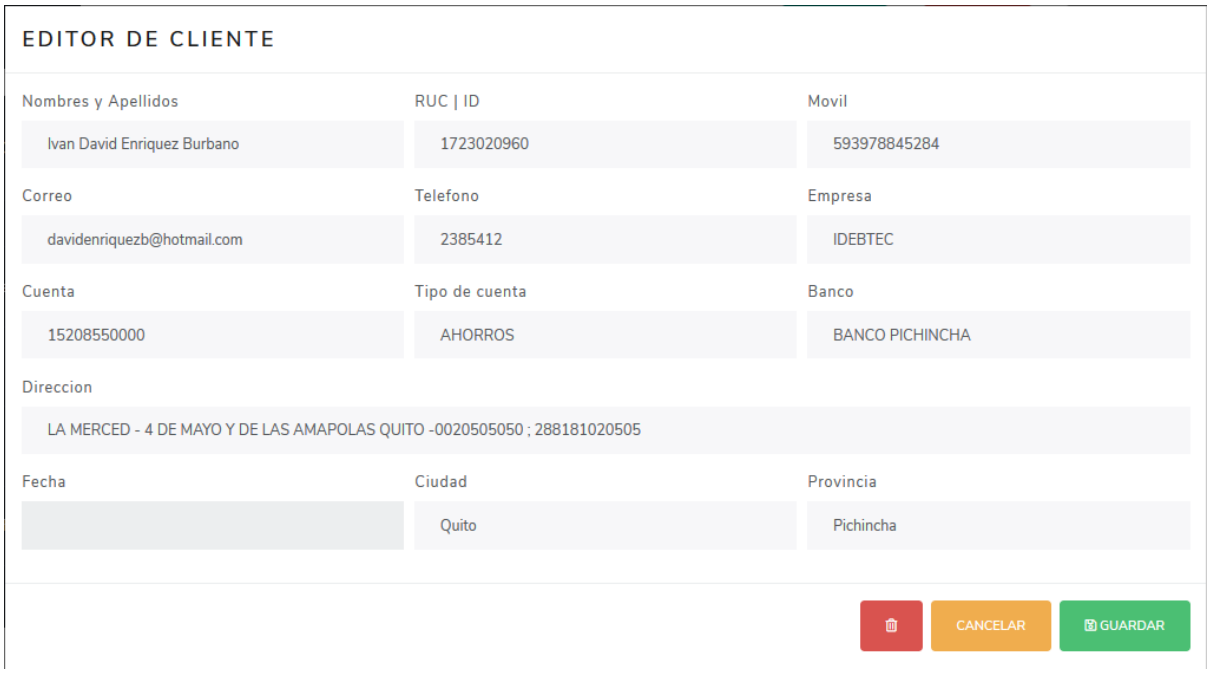
Por último si todos los datos son correctamente ingresados, el usuario puede dar clic en el botón **GUARDAR** y el sistema almacenara el nuevo registro en la base de datos. Caso contrario puede dar clic en **CANCELAR** y esto abortara la operación.

7.3 Editar un Cliente

Para poder editar un cliente, el usuario debe acceder a la sección **CLIENTES** donde se listan todos los clientes ingresados en el sistema, al lado izquierdo de cada cliente podrá observar un botón de edición:



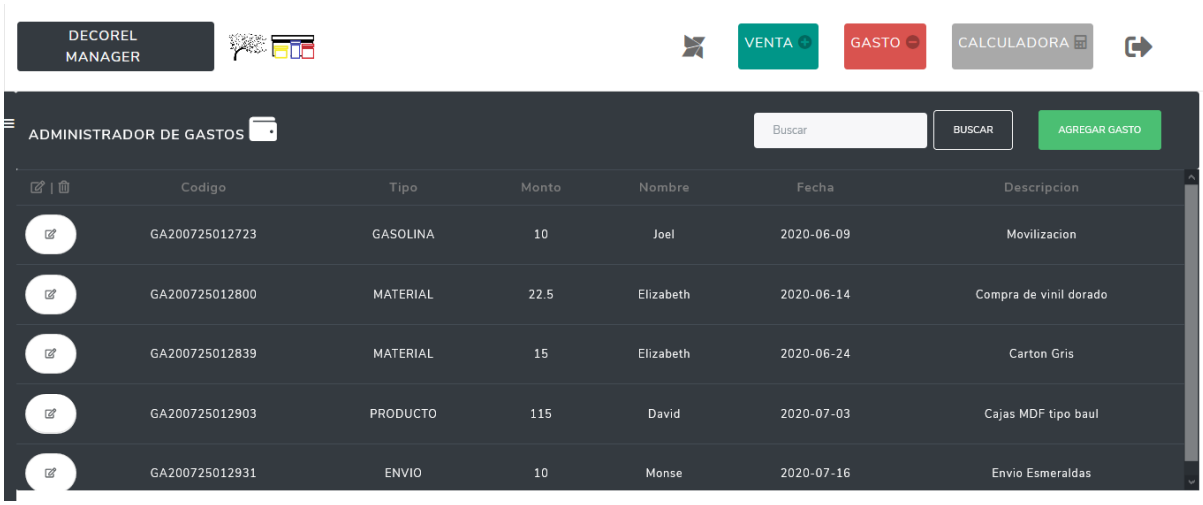
Al dar clic sobre este botón, se desplegará un formulario con los datos del cliente que desea modificar:



En la parte inferior, encontrara un botón rojo que permite **ELIMINAR** este cliente si el usuario así lo prefiere, caso contrario puede modificar los datos que sean necesarios y dar clic sobre el botón **GUARDAR** para almacenar los cambios en el sistema.

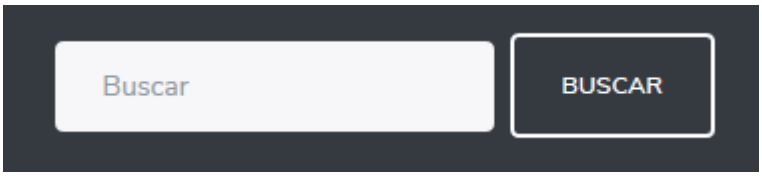
- 8. Sección Gastos
  - 8.1 Pantalla Principal De Gastos

Para acceder a esta sección del sistema puede hacerlo desde la página principal en el recuadro **GASTOS** o desde la barra lateral del sistema.

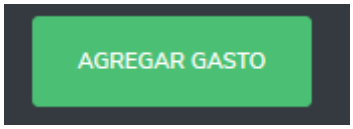


En esta sección podrá encontrar las herramientas necesarias para crear, modificar, eliminar o consultar un **gasto**.

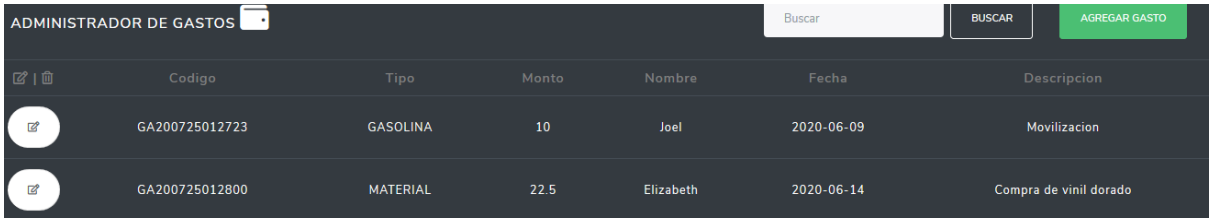
Comenzando por la parte superior el usuario podrá observar una barra de búsqueda, donde puede ingresar el código, valor del gasto o nombre de la persona que realizo el gasto en específico y el sistema lo filtrará en la lista.



A un lado de la barra de búsqueda tenemos el botón **AGREGAR GASTO** que al dar clic el sistema abrirá una nueva ventana para crear un nuevo gasto y almacenarlo en el sistema.

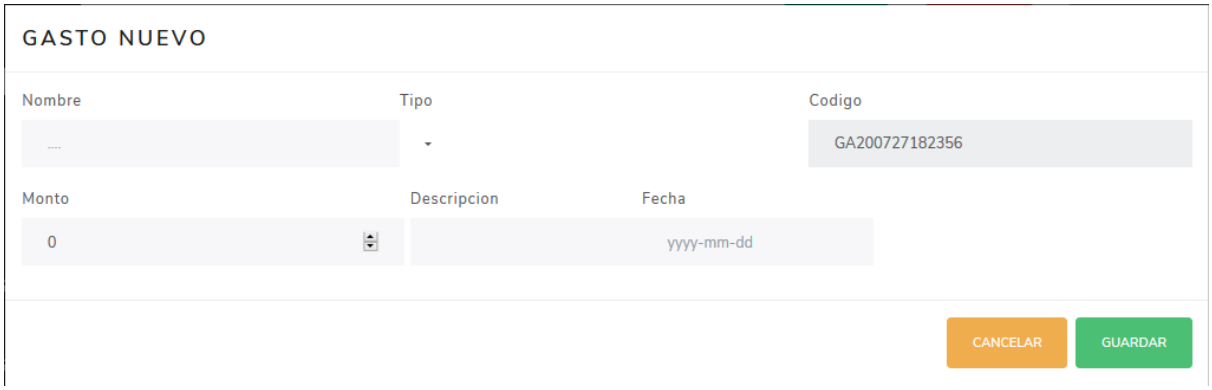


Por último tenemos una lista de todos los materiales creados en el sistema, al lado izquierdo de cada material puede observar un botón que le permitirá editar ese material en específico.



8.2 Agregar un Gasto

Para acceder a esta sección del sistema puede hacerlo desde la página principal en el recuadro **GASTO** o desde la sección gastos dando clic en el botón **AGREGAR GASTO**.



Se desplegará un formulario donde debemos ingresar todos los datos necesarios para generar un gasto.

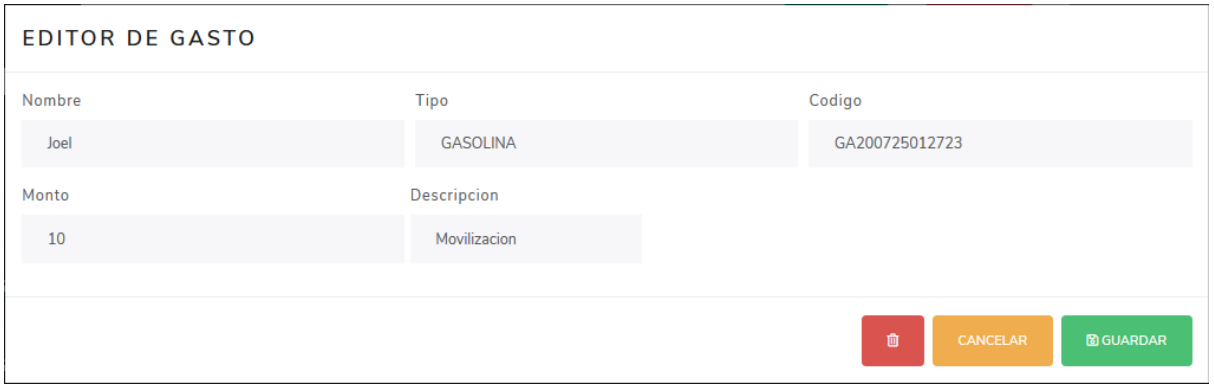
Por último si todos los datos son correctamente ingresados, el usuario puede dar clic en el botón **GUARDAR** y el sistema almacenara el nuevo registro en la base de datos. Caso contrario puede dar clic en **CANCELAR** y esto abortara la operación.

8.3 Editar un Gasto

Para poder editar un gasto, el usuario debe acceder a la sección **GASTOS** donde se listan todos los gastos ingresados en el sistema, al lado izquierdo de cada gasto podrá observar un botón de edición:



Al dar clic sobre este botón, se desplegará un formulario con los datos del gasto que desea modificar:



En la parte inferior, encontrara un botón rojo que permite **ELIMINAR** este gasto si el usuario así lo prefiere, caso contrario puede modificar los datos que sean necesarios y dar clic sobre el botón **GUARDAR** para almacenar los cambios en el sistema.