



**UNIVERSIDAD NACIONAL DE INGENIERÍA
RECINTO UNIVERSITARIO “SIMÓN BOLÍVAR”
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**

TRABAJO MONOGRÁFICO

“Sistema web para la administración de clientes y gestión de cobros para la empresa de televisión por cable “Cable Max” en Kukra Hill (RAACS)”

**PARA OPTAR AL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

ELABORADO POR:

Br. Carmen Mendoza Treminio.

Br. Asael Zuniga Córdoba.

TUTOR:

MSc. Gabriel Rafael Lacayo Saballos

MANAGUA, NICARAGUA

Noviembre 2021

Dedicatoria- Asael Zuniga C.

Dedico este trabajo monográfico principalmente a mi mamá por acompañarme durante todo el proceso y no dejar que me rindiera en las adversidades que se presentaron en el camino, gracias a su apoyo, consejos y sacrificios he logrado llegar hasta aquí.

A mis hermanos y a mi papá por estar detrás de escena durante todo el camino que fue la universidad dando palabras de ánimo, leyendo una y otra vez los borradores y sobre todo confiando en que podría lograr culminar este proceso.

A todos ustedes que no están aquí, pero participaron de manera directa o indirecta en este trabajo, amigos y familiares, les agradezco por sus sugerencias, pruebas ejecutadas y más importante aún por la paciencia.

A mi mamita que, a pesar de ya no estar en este plano, vos más que nadie jamás tuviste dudas de que iba a poder lograr concretar esto, tenías razón, muchas gracias.

Finalmente, al Ing. Gabriel Lacayo quien nos acompañó durante este largo proceso, brindándonos su apoyo, comprensión, conocimiento, colaboración y paciencia permitiendo así el desarrollo de esta tesis.

Dedicatoria- Carmen Mendoza T.

Dedico este trabajo principalmente a Dios por haberme dado la vida, y darme fuerza para continuar en este proceso de obtener uno de mis anhelos más deseados. De igual forma dedico esta tesis a mi madre Azucena Treminio y padre Alejandro Mendoza por su amor, trabajo y sacrificio en todos estos años, por los consejos, valores y principios que me han inculcado, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy.

A mis hermanos, por su cariño y apoyo incondicional, durante todo este proceso, en especial a mi hermana Sabrina que siempre estar ahí para mí y ser una de las personas que me ha mostrado su apoyo incondicional en todo momento gracias. A mis abuelitos porque con sus oraciones, consejos y palabras de aliento, de una u otra forma me acompañan en todos mis sueños y metas.

A todos mis amigos que me han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que compartieron sus conocimientos y estuvieron siempre ahí animándome a seguir y culminar este momento tan importante de mi formación profesional.

Finalmente quiero expresar mi más grande y sincero agradecimiento al Ing. Gabriel Lacayo, principal colaborador durante todo este proceso, quien con su dirección, conocimiento, enseñanza y colaboración permitió el desarrollo de este trabajo

Resumen

El siguiente trabajo monográfico, plantea el análisis, diseño y codificación de un sistema para la administración de clientes y gestión de cobros para la empresa de televisión por cable *CableMax* utilizando como soporte los conocimientos adquiridos durante la carrera al igual que en el campo laboral.

El proceso de desarrollo del tema toma como pilar, el análisis de historias de usuario generadas en conjunto con el personal de la empresa con la finalidad de obtener los requerimientos necesarios para la creación de una herramienta de software funcional y útil para la empresa.

Las etapas de análisis, diseño y codificación del sistema fueron llevadas a cabo siguiendo el patrón MVT (Modelo-Vista-Template) y la metodología de desarrollo XP (extreme programming), permitiendo mantener una comunicación fluida y constante con los usuarios finales, determinando los requerimientos y adecuaciones necesarias para cada módulo desarrollado, y una comprensión a mayor profundidad sobre cada proceso planteado a través del uso de diagramas basados en UML.

Las tecnologías de desarrollo web utilizadas para la creación de este software son: PostgreSQL como sistema gestor de bases de datos, Django como framework principal para el backend, Bootstrap como framework para el frontend, las demás tecnologías utilizadas están incluidas en el desarrollo del documento.

Finalmente, al concluir este trabajo monográfico, se obtuvieron una serie de resultados sobre la factibilidad de uso según el personal de la empresa, las cuales han sido incluidas en este documento.

Índice

1.	Introducción	1
2.	Antecedentes	3
2.1.	BoxiCableWEB	3
2.2.	SaEcO	3
2.3.	Gestel	4
2.3.1.	Características	4
3.	Justificación	5
4.	Objetivos	6
4.1.	Objetivo General.....	6
4.2.	Objetivos Específicos.....	6
5.	Marco Teórico	7
5.1.	Descripción de los procesos del negocio	7
5.2.	Herramientas de desarrollo.....	8
5.2.1.	Python	8
5.2.2.	Django	8
5.2.3.	MVT	9
5.2.4.	API	10
5.2.5.	HTML5	10
5.2.6.	CSS3	10
5.2.7.	Jquery	10
5.2.8.	Bootstrap	11
5.2.9.	Postgresql	11
5.3.	Modelo de base de datos	12
5.3.1.	Modelo Entidad - Relación	12
5.4.	Patrón de diseño de software.....	12
5.4.1.	FACADE	12
5.5.	Pruebas	13
5.6.	Métricas	13
5.7.	Metodologías Agiles.....	14
5.7.1.	Manifiesto Ágil.....	14

5.7.2.	Metodología XP	15
6.	Estudio de Factibilidad	21
6.1.	Factibilidad técnica	21
6.1.1.	Propuesta 1.....	21
6.2.	Factibilidad Operativa	23
6.3.	Factibilidad económica.....	23
6.3.1.	Recursos humanos.....	25
6.3.2.	Costo total del proyecto.....	25
6.4.	Beneficios del sistema.....	26
6.4.1.	Requerimientos funcionales del sistema	27
6.4.2.	Requerimientos no funcionales.....	28
7.	Marco Metodológico.....	29
7.1.	Fase I: Definición del proyecto.....	29
7.2.	Fase II: Análisis de requerimientos	29
7.3.	Fase III: Diseño.....	29
7.4.	Fase IV: Desarrollo del sistema	29
7.5.	Fase V: Pruebas.....	30
7.6.	Fase VI: Implementación del sistema y capacitación	30
8.	Análisis y presentación de los resultados	31
8.1.	Planeación.....	31
8.1.1.	Historias del usuario	31
8.1.2.	Release Planning	36
8.2.	Primera Iteración	37
8.2.1.	Tareas de ingeniería.....	37
8.2.2.	Tarjetas CRC	43
8.2.3.	Pruebas de aceptación	44
8.2.4.	Diseño.....	48
8.2.5.	Boceto.....	49
8.2.6.	Codificación.....	55
8.3.	Segunda Iteración	57
8.3.1.	Tareas de ingeniería.....	57
8.3.2.	Tarjetas CRC	63
8.3.3.	Pruebas de aceptación.....	65

8.3.4.	Boceto.....	67
8.3.5.	Codificación.....	72
8.4.	Tercera Iteración.....	74
8.4.1	Tareas de ingeniería.....	74
8.4.2.	Tarjetas CRC	77
8.4.3.	Pruebas de aceptación.....	77
8.4.4.	Boceto.....	78
8.4.5.	Codificación.....	83
8.5.	Métricas	84
8.5.1.	Definición de métricas	84
8.5.2.	Definición de instrumentos.....	85
8.6.	Resultados de la prueba	85
8.6.1.	Resultado Caso de prueba 1.....	85
8.6.2.	Resultado Caso de prueba 2.....	86
8.6.3.	Resultados de la encuesta.....	87
9.	Conclusiones	88
10.	Recomendaciones.....	89
11.	Bibliografía	90
12.	Anexos	92

Índice de Tablas

Tabla 1 Historia de usuario acceso al sistema	31
Tabla 2 Historia de usuario Registro de cliente.....	32
Tabla 3 Historia de usuario Gestión de clientes.....	32
Tabla 4 Historia de usuario Estado de cliente	33
Tabla 5 Historia de usuario registro de pagos.....	33
Tabla 6 Historia de usuario Facturación.....	34
Tabla 7 Historia de usuario Reportes	34
Tabla 8 Historia de usuario Cobrador.....	35
Tabla 9 Release Planning Entregas	36
Tabla 10 Iteración 1	37
Tabla 11 Tarea de ingeniería.....	37
Tabla 12 Tarea de ingeniería 1 para historia de usuario 1	38
Tabla 13 Tarea de ingeniería 2 para historia de usuario 1	39
Tabla 14 Tarea de ingeniería 3 para historia de usuario 1	39
Tabla 15 Tarea de ingeniería 4 para historia de usuario 1	40
Tabla 16 Tarea de ingeniería 5 para historia de usuario 2	40
Tabla 17 Tarea de ingeniería 6 para historia de usuario 2	41
Tabla 18 Tarea de ingeniería 7 para historia de usuario 2	41
Tabla 19 Tarea de ingeniería 8 para historia de usuario 3.....	42
Tabla 20 Tarea de ingeniería 8 para historia de usuario 4.....	42
Tabla 21 Tarea de ingeniería 10 para historia de usuario 4.....	43
Tabla 22 Tarjeta CRC Clientes.....	43
Tabla 23 Tarjeta CRC Gestión de clientes	44
Tabla 24 Tarjeta CRC Estado de cliente.....	44
Tabla 25 Tabla pruebas de aceptación de la primera iteración	45
Tabla 26 Iteración 2	57
Tabla 27 Tarea de Ingeniería.....	57
Tabla 28 Tarea de ingeniería 1 para historia de usuario 5.....	58
Tabla 29 Tarea de ingeniería 2 para historia de usuario 5.....	59
Tabla 30 Tarea de ingeniería 3 para historia de usuario 5.....	59
Tabla 31 Tarea de ingeniería 4 para historia de usuario 5.....	59
Tabla 32 Tarea de ingeniería 5 para historia de usuario 6.....	60
Tabla 33 Tarea de Ingeniería 6 para historia de usuario 6.....	60
Tabla 34 Tarea de ingeniería 7 para historia de usuario 6.....	60
Tabla 35 Tarea de ingeniería 8 para historia de usuario 7	61
Tabla 36 Tarea de ingeniería 9 para historia de usuario 7	61
Tabla 37 Tarea de Ingeniería 10 para historia de usuario 7.....	61
Tabla 38 Tarea de ingeniería 11 para historia de usuario 7.....	62
Tabla 39 Tarea de ingeniería 12 para usuario 7	62
Tabla 40 Tarjeta CRC Usuarios	63
Tabla 41 Tarjeta CRC Facturación	63
Tabla 42 Tarjeta CRC Pagos.....	63
Tabla 43 Tarjeta CRC Recibo.....	64

Tabla 44 Tarjeta CRC Reportes	64
Tabla 45 Tarjeta CRC Estado de factura	64
Tabla 46 Tarjeta CRC Observaciones.....	64
Tabla 47 Iteración 3	74
Tabla 48 Tarea de iteración	74
Tabla 49 Tarea de Ingeniería 1 para historia de usuario 8.....	74
Tabla 50 Tarea de ingeniería 2 para historia de usuario 8.....	75
Tabla 51 tarea de ingeniería 3 para historia de usuario 8	75
Tabla 52 Tarea de ingeniería 4 para historia de usuario 8.....	75
Tabla 53 Tarea de ingeniería 5 para historia de usuario 8.....	76
Tabla 54 Tarjeta CRC Permisos.....	77

1. Introducción

El manejo de datos específicos y relevantes en una empresa, es indispensable para el éxito de la misma. El progreso de la tecnología facilita esta actividad, a través de la creación de sistemas de información que no solamente permite el manejo de datos confiables, y exactos, sino también el acceso a éstos.

Un sistema de información, abarca elementos orientados al tratamiento y administración de detalles, organizados, actualizados y listos para su uso posterior, éstos serán generados para cubrir necesidades con objetivos concretos

Las acciones de recolección y procesamiento de información, eran actividades manuales que no permitían la agilidad y seguridad de la misma; el uso y desarrollo de la tecnología, computadoras, Internet, bases de datos, entre otros, ha favorecido de gran manera, ya que éstos son utilizados para establecer el seguimiento de la información, tales como: inventario, de los empleados, de los clientes y de la contabilidad.

Cada uno de los datos mencionados, puede ser guardado y utilizado en estructuras definidas, de modo que podamos acceder a ellos de manera rápida, e incluso, poder ubicar ágilmente un dato o información específica desde recursos materiales en general como, papel, lápices, libros, carpetas, dentro de la base de datos. Podemos observar también la importancia que representa en la economía de toda empresa, en la que la vía de acceso a la información en general, no solo implica ahorro de tiempo, y materiales, sino que además se da de una manera dinámica y actual.

En el presente trabajo se propone el desarrollo y la implementación del Sistema web para la administración de clientes y gestión de cobros para la empresa de televisión por cable "Cable Max" en Kukra Hill (RAACS).

Este sistema web estará enfocado en tener un control total de la información general de un cliente, mejorar el proceso de atención al cliente a la hora de un reclamo, control de pagos realizados.

En el sistema web a desarrollar se espera destacar como característica el que sea amigable para el usuario, a través de una interfaz minimalista que permita una experiencia de usuario agradable, con el fin de que los trabajadores que lo utilicen se adapten fácilmente al mismo y este no signifique un problema al momento de llevar el control de pagos realizados. En este punto se hará uso de metodologías, patrones de diseño de software, métricas que permitan evaluar la experiencia de usuario y herramientas específicas para desarrollo de interfaces y pruebas de usabilidad.

2. Antecedentes

La empresa de televisión por cable “Cable Max” inicia operaciones en 2012 en la Región Autónoma de la Costa Caribe (RACCS) Kukra Hill, dirigida por Alejandro Mendoza, con la misión de proveer a los ciudadanos el servicio de televisión por cable permitiendo así a los clientes recrearse o mantenerse informados según estos lo requieran.

La misma tiene como visión ser reconocida por su excelente servicio, su señal de calidad y su capacidad de innovar en comparación a otras empresas que prestan el mismo servicio.

A continuación, se presenta un resumen de la consulta documental de trabajos realizados que guardan relación con los objetivos propuestos en este trabajo monográfico

2.1. BoxiCableWEB

La empresa de sistemas de facturación, CableSoft, posee una aplicación web que permite administrar una empresa de cable servicio, su nombre es BoxiCableWEB, entre las características más relevantes están. (CableSoft, 2016)

- * Administración y atención de clientes
- * Cobranzas y débitos automáticos
- * Ordenes de instalación
- * Seguimiento de reclamos técnicos

2.2. SaEcO

Saeco es un sistema dedicado a empresas cable operadoras con la particularidad que permite una comunicación directa con sus clientes, ya sea a través de correo electrónico o mensaje de texto, entre sus características más destacables están (Canejo, 2015)

- * Reportes diarios (ingresos, egresos, desempeño de personal)

* Permite localizar la ubicación de los clientes de manera sencilla

* Estadísticas de estado de la empresa

2.3. Gestel

Es un software para el rubro de empresas que brindan el servicio de televisión por cable, cuenta con una solución informática completa que cubre los diversos requerimientos propios de este rubro empresarial (Sistemas para servicio de television por cable Gestel, 2015)

2.3.1. Características

- Zonificación del mercado
- Atención a los clientes
- Inscripción de conexiones nuevas
- Manejo de tarifas:
- Formato de Recibos:
- Programación de cortes
- Facturación mensual

3. Justificación

Para solucionar las necesidades que presenta la empresa de televisión por cable “Cable Max” ubicada en Kukra Hill, se propone el desarrollo de un sistema web para la administración de clientes y gestión de cobros, el cual tendrá como labor principal, asistir como una herramienta para llevar de manera más eficaz el control de los procesos de administración de los datos de los clientes y facturación del negocio.

Con la implementación del sistema web para la administración de clientes y gestión de cobros se tendrá control total de la información general del cliente (datos personales), control de pagos realizados o pendientes por cada cliente, imprimir factura de cobro mensuales de los clientes, mejorar el proceso de atención al cliente a la hora de un reclamo.

Este sistema facilitará la gestión del día a día de la empresa, agilizando así, la consulta de información, reduciendo el papel existente (datos físicos del cliente), todo desde una computadora y sobre todo permitiendo una atención personalizada a los clientes. Otro aspecto importante es que dicho sistema permitirá resguardar la información de forma exacta y confiable, la cual estará disponible para su respectivo análisis y contribuir a la toma de decisiones del negocio.

4. Objetivos

4.1. Objetivo General

Desarrollar un sistema web para la administración de clientes y gestión de cobros para la empresa de televisión por cable “cable Max”.

4.2. Objetivos Específicos

- Analizar las historias de usuario del sistema web para la administración de clientes y gestión de cobros utilizando una metodología ágil.
- Diseñar las interfaces y bases de datos necesarias para la creación del sistema web usando una herramienta de diseño UML.
- Codificar los procesos y funciones correspondientes a la interfaz previamente creada utilizando el lenguaje de programación Python haciendo uso de las buenas prácticas del desarrollo de software.

5. Marco Teórico

En este capítulo se presentan cada uno de los conceptos necesarios para el desarrollo de este proyecto.

5.1. Descripción de los procesos del negocio

El giro comercial de CableMax radica en el préstamo de servicios de cable a sus clientes teniendo tres funciones principales las cuales se detallarán a continuación:

Facturación: El sitio economipedia describe una factura como un documento de carácter mercantil que refleja la compraventa de un bien o la prestación de un servicio. (Economipedia, 2021)

El proceso de facturación consiste en la emisión y registro visual de facturas, en el caso de cablemax la emisión de una factura representa el cobro por préstamo de servicio, las mismas son creadas el primer día de cada mes.

Pagos: Se define como pago a toda acción que realizamos para extinguir o cancelar una obligación.

El proceso de realización de pagos es ejecutado por los clientes, este representa la cancelación del servicio otorgado durante el mes reflejado en la factura.

Reportes: Un reporte se define como un informe que organizan y exhiben la información contenida en una base de datos, su objetivo principal es mostrar dicha información por medio de un diseño atractivo y fácil de interpretar por los usuarios. (Merino, 2010)

Para los dueños de CableMax es vital obtener reportes sobre el estado de sus clientes y los pagos realizados por los mismos, por los que solicitan un módulo de reportes.

5.2. Herramientas de desarrollo

A continuación, definiremos todas las herramientas necesarias para la creación del sistema:

5.2.1. Python

Es un claro y poderoso lenguaje de programación orientado a objetos, comparable a Perl, Ruby o Java.

Entre sus características principales están:

- Utiliza una sintaxis elegante que facilita la lectura de los programas que escribes.
- Viene con una gran biblioteca estándar que admite muchas tareas de programación comunes, como conectarse a servidores web, buscar texto con expresiones regulares, leer y modificar archivos.
- Funciona en cualquier sistema, incluyendo Mac OS X, Windows, Linux y Unix, con versiones no oficiales también disponibles para Android y iOS.
- Es Software libre, no tiene costo su descarga, ni utilización en una aplicación, también es modificable bajo la licencia de código abierto. (python, 2020)

5.2.2. Django

Es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–template.

Este framework es la herramienta principal para el desarrollo de nuestro sistema web.

Las características de Django según su página son:

- **Ridículamente rápido:** permite llevar un sitio desde su concepto hasta su finalización lo más rápido posible.

- **Completamente cargado:** Cuenta con extras que sirven para la realización de tareas comunes del desarrollo web (administración de contenido, autenticación de usuarios).
- **Seguro:** Django toma en serio la seguridad y ayuda a los desarrolladores a evitar muchos errores comunes de seguridad, como la inyección de SQL, los scripts entre sitios, la falsificación de solicitudes entre sitios y el clickjacking.
- **Muy escalable:** Algunos de los sitios más concurridos del planeta utilizan la capacidad de Django para escalar de manera rápida y flexible para satisfacer las demandas de tráfico más pesadas.

5.2.3. MVT

MVT son las siglas para Model View Template, es decir modelo vista plantilla. Una aplicación web basada en este patrón divide su código en tres partes:

- *M* significa "Model" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- *V* significa "View" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre los modelos y las plantillas.
- *T* significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento. (uniwebsidad, s.f.)

5.2.4. API

Una API es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software. Las siglas API vienen del inglés Application Programming Interface (Interfaz de Programación de Aplicaciones).

Una API nos permite implementar las funciones y procedimientos que engloba nuestro proyecto sin la necesidad de programarlas de nuevo. En términos de programación, es una capa de abstracción.

Un ejemplo de API es Request de Python, la cual nos permite hacer peticiones HTTP sin tener que codificarlas cada vez que las necesitemos. (Alvarez, 2019)

5.2.5. HTML5

Es un lenguaje de marcado de texto usado para estructurar y presentar el contenido para la web.

HTML5, es la quinta revisión del estándar HTML, esta versión nos permite una mayor interacción entre nuestras páginas web y el contenido medio (video, audio, entre otros) así como una mayor facilidad a la hora de codificar nuestro diseño básico.

5.2.6. CSS3

CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas. (Sistemas, 2012)

5.2.7. JQuery

JQuery es una librería de JavaScript. Esta librería de código abierto, simplifica la tarea de programar en JavaScript y permite agregar interactividad a un sitio web.

Basados en esta librería, existen muchos plugins (gratis y pagos) estos resuelven situaciones concretas dentro del maquetado de un sitio, por ejemplo: menú responsive (adaptable a móviles), una galería de fotos, un carrousel de

imágenes, un slide, un header que cambia de tamaño, el deslizamiento del scroll al hacer clic en un botón (anclas HTML), la transición entre páginas y miles de efectos más.

Esta última característica es vital para nuestro proyecto ya que utilizamos plugins para el frontend. (CHUBURU, 2020)

5.2.8. Bootstrap

Es un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Permite crear de forma sencilla webs de diseño adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y siempre se vean igual de bien. Es Open Source por lo que lo podemos usar de forma gratuita y sin restricciones.

Por dichas características este framework es nuestra principal herramienta de frontend. (Useit, 2018)

5.2.9. Postgresql

Es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de datos. Se ejecuta en los principales sistemas operativos que existen en la actualidad como:

- Linux
- UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64)
- Windows

Utilizamos este sistema de base de datos por la excelente compatibilidad con django, además de su rapidez y documentación. (postgresql, 2021)

5.3. Modelo de base de datos

Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos.

5.3.1. Modelo Entidad - Relación

Es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema.

Este diagrama se acopla perfectamente con nuestro trabajo ya que permite diseñar la base de datos en base a las reglas de negocio previamente establecidas, una vez finalizada esta etapa pasamos a la creación de una interfaz para interactuar con la misma. (Lucidchart , 2021)

5.4. Patrón de diseño de software

Los patrones de diseño son soluciones habituales a problemas que ocurren con frecuencia en el diseño de software. David Quiroga de clibre.io define un patrón de diseño como una descripción o plantilla sobre cómo resolver un problema que se puede utilizar en muchas situaciones diferentes.

Los patrones de diseño son como planos que podemos aplicar a problemas comunes dentro del desarrollo de software, dichos patrones deben poseer características específicas entre las más importantes están previamente probados (funcionales) un patrón debe demostrar haber resuelto y reutilizables o sea que pueden ser aplicados a distintos problemas de diseño en diferentes circunstancias. (Quiroga, 2020).

5.4.1. FACADE

Facade (o fachada) es un patrón de diseño estructural que proporciona una interfaz simplificada a un framework o cualquier otro grupo complejo de clases.

Una fachada es una clase que proporciona una interfaz simple a un subsistema complejo, esto quiere decir que solo incluye los procesos importantes para el cliente.

Una fachada resulta útil cuando se tienen múltiples funciones, pero solo se requiere una pequeña parte.

Para entender mejor este concepto Alexander Shvets del sitio web refactoring guru nos brinda la siguiente analogía en el mundo real:

Cuando llamas a una tienda para hacer un pedido por teléfono, un operador es tu fachada a todos los servicios y departamentos de la tienda. El operador te proporciona una sencilla interfaz de voz al sistema de pedidos, pasarelas de pago y varios servicios de entrega.

Este patrón de diseño se acopla perfectamente a nuestro sistema web ya que el mismo utiliza una clase para interactuar entre procesos permitiendo así, utilizar solo los que el cliente necesite en el momento.

Al finalizar esta etapa, nos queda un último paso el cual consiste en realizar pruebas a software para ver si el mismo cumple con las funciones para las cuales fue diseñado. (refactoring.guru, 2014)

5.5. Pruebas

Las pruebas son la forma en que puede estar seguro acerca de la funcionalidad, el rendimiento y la experiencia del usuario.

Existen diversos tipos de pruebas, pero en el caso de nuestro sistema hemos utilizado las pruebas de aceptación las cuales tienen como objetivo comprobar si el software está preparado y puede ser utilizado por los usuarios para realizar las funciones y tareas para las que fue diseñado.

Una vez realizada las pruebas procedemos a la evaluación de métricas estas con el objetivo de evaluar la satisfacción del usuario o la adaptación del mismo al sistema.

5.6. Métricas

Es cualquier medida o conjunto de medidas que permitan conocer o estimar el tamaño u otra característica de un software, generalmente para realizar comparativas.

Existen muchos tipos de métricas sin embargo para el desarrollo de nuestro sistema es vital evaluar los tiempos actuales entre procesos versus los tiempos de procesos

utilizando el sistema, para ello utilizaremos una métrica basada en calidad y fiabilidad, las mismas tienen como objetivo medir la facilidad de uso y mantenimiento del software. (Febles Estrada, 2008)

5.7. Metodologías Ágiles

Por definición las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez (iebschool, 2021).

5.7.1. Manifiesto Ágil

En febrero de 2001, 17 expertos del desarrollo de software se reunieron para esquivar, relajarse y tratar de encontrar un terreno en común, representantes de la programación extrema, scrum, desarrollo adaptivo de software, entre otros, se encontraban entre los asistentes. producto de esta reunión nació el manifiesto ágil para el desarrollo de software (the agile manifiesto) el cual contaba con métodos alternativos a los existentes. (Highsmith, 2001)

En esta reunión se establecieron los siguientes valores:

- Individuos e interacciones sobre procesos y herramientas
- Software funcional sobre documentación exhaustiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta al cambio sobre seguir de manera estricta un plan.
- En base a estos valores, el manifiesto ágil establece 12 principios que lo diferencian de los métodos rígidos, los cuales son:
 - La prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
 - Los requisitos pueden cambiar incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

- Entregar software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores deben trabajar juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan entorno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcional es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
- En intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para ajustarse y perfeccionar su comportamiento en consecuencia.

5.7.2. Metodología XP

La “Extreme Programming” (Programación extrema en español) es una metodología ágil para el desarrollo de software más exitosas en la actualidad.

Existe una serie de valores que le dan la esencia a XP:

Simplicidad: Esta es la base de la programación extrema. Un diseño simple permite agilizar el desarrollo y facilitar el mantenimiento. Según el código se hace más extenso se realiza refactorización para mantenerlo simple y funcional.

La simplicidad aplica tanto al código como a la documentación.

Comunicación: Debe existir comunicación interna como externa (con el cliente), la comunicación entre los programadores es fluida gracias a la programación en parejas.

La comunicación con el cliente es constante ya que forma parte del equipo, siempre debe estar disponible para solucionar dudas.

Retroalimentación: Al tener contacto directo con el cliente como parte del equipo obtenemos su opinión de inmediato.

Al presentarse resultados en poco tiempo se evita tener que rehacer módulos enteros y así el programador se puede concentrar en lo que es más importante.

Coraje: El equipo de trabajo debe estar preparado para los cambios que surjan en el transcurso del desarrollo del proyecto.

Estos valores inspiran una serie de buenas prácticas:

El Juego de la planificación: Mantener comunicación constante entre el equipo de desarrollo y el cliente, para tener una visión clara de lo que se quiere.

Entregas pequeñas: se realizan entregas constantes de módulos funcionales. Estas versiones deben estar completas, no se pueden acabar medio modulo y lanzarse.

Metáforas: La función de las metáforas es hacer el código más legible, haciendo que cualquiera entienda cómo funciona el sistema.

Diseño Simple: Consta en realizar solamente lo necesario para que el aplicativo cumpla la actividad requerida por el cliente. Parte de la premisa “como podemos hacerlo más simple”

Recodificación: Complementa la anterior, una vez realizado el código lo revisamos y buscamos la manera de hacerlo más simple sin perder funcionalidad.

Programación en parejas: Todo el código es realizado por dos personas frente a un mismo computador, mientras una codifica, la otra piensa de manera estratégica: funcionalidad, simplicidad.

Propiedad colectiva: Se auxilia de la anterior. El código no es conocido solo por un miembro del equipo lo que permite a los demás miembros realizar cambios de ser necesario.

Integración continua: La integración del código debe realizarse mínimo una vez al día, una pareja de programadores se encargará de realizar las pruebas necesarias hasta que estas funcionen en su totalidad.

No trabajar horas extras: Es preferible adaptar los tiempos. Esta práctica permite mantener las ideas frescas y la creatividad a flote.

Cliente in situ: El cliente siempre debe estar disponible para resolver dudas, aclarar discusiones y definir prioridades.

Estándares de programación: El uso de esta práctica permite a todos los involucrados en el desarrollo poder entender el código fuente y realizar cambios de ser necesarios.

Pruebas: No debe existir ninguna característica que no haya sido sometida a pruebas que garanticen su correcto funcionamiento.



Ilustración 1 Metodología XP

5.7.2.1 Roles

Esta metodología hace uso de roles para definir quienes se encargarán de una actividad en específico, estas funciones pueden ser desempeñadas por una o más personas, sin descartar la posibilidad de rotar roles.

Programador: produce el código de la aplicación.

Tester: Crear las pruebas unitarias.

Cliente: Escribe las historias de usuario y determina las pruebas necesarias para validar lanzamiento.

Tracker: Verifica los tiempos vs estimaciones. Es el que define el grado de acierto en las estimaciones y da los tips para mejorar en futuros cálculos.

Coach: Guía al equipo a seguir el camino correcto.

Consultor: miembro externo al equipo con conocimiento en un tema necesario para la realización del proyecto. Ayuda al equipo a resolver problemas.

Big boss: Vínculo entre cliente y programadores.

5.7.2.2 Fases

La metodología xp establece las siguientes fases para su correcta implementación:

- **Fase 1. Planeación**

Es la piedra angular de un proyecto que utilice xp. En esta fase nos reunimos con el cliente y equipo para definir requerimientos, se identifican el número de iteraciones y se planean ajustes a la metodología.

Esta sección se divide en 6 elementos los cuales son: **Historias de usuario, release planning, iteraciones, velocidad del proyecto, pequeñas entregas y reuniones.**

Historias de usuario: son escritas por el cliente o con el cliente, en lenguaje no técnico, tienen la misma finalidad que los casos de uso; se utilizan para estimar tiempos de

desarrollo, también son utilizadas en la fase de pruebas para comprobar que el aplicativo cumple con lo especificado en la historia de usuario.

Tiempo ideal de desarrollo: 1 a 3 semanas.

Release Planning: Una vez creadas las historias de usuarios, se definió el release planning (plan de publicaciones) en este se establece que historias se publicarán en que versión y en que fechas serán liberadas. El release planning se hace en conjunto con el cliente y el equipo, se establece la prioridad de las historias, el tiempo en el que serán liberadas y la prioridad con las que serán implementadas.

Iteraciones: Los proyectos que utilizan xp son divididos en iteraciones con una duración de 1 a 3 semanas. Al comienzo de cada iteración el cliente define que historia de usuario será implementada.

Velocidad de proyecto: representa la rapidez con la que se desarrollara el proyecto. Consiste en contar el número de historias de usuario que pueden ser lanzadas en una iteración, una vez realizado esto teóricamente podríamos usar el mismo número en la siguiente iteración. Se recomienda reevaluar cada 3 o 4 iteraciones, ya que cada historia presenta un nivel de dificultad diferente, si notamos que no se ajusta se negocia un nuevo release plan con el cliente.

Entregas pequeñas: Al finalizar el tiempo de una iteración (1 a 3 semanas) tendremos una versión de producto funcional a nivel modular, la cual será mostrada como avance, estas entregas se caracterizan por ser frecuentes.

Reuniones: El equipo debe reunirse de manera constante y expresar los problemas existentes, soluciones e ideas que tengan. En estas reuniones deben tomarse en cuenta todas las opiniones expresadas.

- **Fase 2. Diseño**

XP establece la creación de diseños claros y simples, para lograr esto hace uso de los siguientes conceptos:

Simplicidad: la metodología xp sugiere implementar el diseño más simple posible, esto lo hace entendible y más fácil de implementar.

Soluciones Spike: Cuando el equipo se enfrenta a problemas técnicos o de implementación, puede utilizarse la herramienta “spike solution” consiste en explorar el problema y proponer una posible solución.

Recodificación: Consiste en reescribir segmentos de código, sin cambiar su funcionalidad, esto se hace con el objetivo de hacerlo más entendible y simple.

Metáfora: Es una manera simple y fácil de entender para todos de lo que hay que hacer. Una metáfora debe ser entendida tanto por el cliente, como por el equipo.

- **Fase 3. Pruebas**

Detección y corrección de errores: Cuando se encuentra un bug este debe solucionarse de inmediato, así mismo, se generan nuevas pruebas que verifiquen que el error ha sido solucionado.

Pruebas de aceptación: Las pruebas de aceptación, son pruebas funcionales supervisadas por el cliente basándose en los requerimientos tomados de las historias de usuario.

6. Estudio de Factibilidad

6.1. Factibilidad técnica

6.1.1. Propuesta 1.

Para la obtención del hardware necesario, se realizó un estudio de requerimientos mínimos y recomendados. Partiendo de los requerimientos mínimos obtenemos los siguientes datos:

Hardware

- 1 PC Intel Core I3-1005G1 DE 1.20GHZ, 8gb DDR4, 500GB HDD, Windows 10, Pantalla de 14"
- Impresora multifuncional, bandeja estándar.
- Contar con un proveedor de internet.
- 1 Samsung A01 Core para uso de cobrador.

Tras recabar estos datos, se realizó una comprobación de hardware existente, descubriendo que la empresa cuenta con la mayoría de equipos necesario para ejecutar el sistema:

	PowMr: Impresora POS	<ul style="list-style-type: none">● Android 7.0● Pos compatible con el 98 de apps android.● Soporte 4g● Diametro de roll 58mm*50mm.5.0 MP
---	----------------------	--

	<p>Samsung S6</p>	<ul style="list-style-type: none"> ● Android 5.0 ● Gorilla Glass ● 5.1 pulgadas ● Octa-core ● 32GB 3GB RAM.
---	-------------------	--

Una de las ventajas de realizar un sistema web, es que no son necesarios equipos de componentes muy costosos, ya que todo se ejecuta desde el navegador.

La laptop será utilizada por el usuario: **secretaria**, el cual está encargado de la gestión del sistema, así mismo la impresora (Pos en este caso) será utilizada para brindar comprobantes de pago a los clientes.

Software

En cuanto al software, se utilizará Ubuntu 20.04 Focal Fossa para el desarrollo del sistema, sin embargo, se planea implementar el mismo en un sistema windows esto por preferencia de la empresa.

Tras realizar un estudio para determinar los requerimientos necesarios para el desarrollo e implementación del sistema, obtuvimos los siguientes resultados:

- Ubuntu 20.04 (Desarrollo)
- PostgreSQL
- Visual Studio: Code
- Django Framework

La rapidez y escalabilidad de los proyectos desarrollados en Django, en combinación con la robustez de postgresQL y lo bien que se acoplan fue el factor determinante para elegir estas tecnologías. Cabe destacar que ambas se encuentran entre las más utilizadas por los profesionales en el desarrollo de software, rasgo importante ya que brinda soporte de que son estables y funcionan muy bien en conjunto.

6.2. Factibilidad Operativa

Con la implementación del sistema no se prevén cambios organizacionales, sino más bien agilizar los procesos cotidianos realizados por la empresa.

El equipo de desarrollo se encargará de capacitar al personal de la empresa en el manejo del software, ya que ellos serán los responsables de cargar la información inicial dentro del sistema.

Algunos beneficios operativos que brindara el desarrollo de este sistema en la empresa cablemax son:

- Acceso a la información de manera rápida y ordenada.
- Optimización de tiempos entre procesos.
- Facilita la obtención de datos de manera visual (reportes).

6.3. Factibilidad económica

En las siguientes tablas mostramos el análisis de costos:

Costo de obtención de hardware

Tomando en cuenta el estudio de factibilidad, obtuvimos los siguientes datos:

RECURSOS TECNOLÓGICOS			
HARDWARE			
Cantidad	Descripción	Costo por unidad	Costo Total
1	1 computadora de escritorio Intel Core i3-1005G1 DE 1.20GHZ, 8gb DDR4, 500GB HDD, Windows 10, Pantalla de 14"	\$499	\$499
		Total	\$499

Costo de obtención de software

El sistema operativo propuesto por la empresa es MS Windows en su versión 10 home, la licencia del mismo viene con el equipo. El equipo de desarrollo propone el uso de Ubuntu 20.04 para la creación del sistema, el mismo es un sistema operativo de uso libre por lo tanto no incurre en gastos, de la misma forma nuestras herramientas de software son de uso libre y no necesitamos comprar licencias para utilizarlas, dicha información se detalla en la siguiente tabla:

RECURSOS TECNOLÓGICOS	
SOFTWARE	
Descripción	Costo por unidad
Licencia MS Windows 10	Adquirida al comprar el equipo.
Visual Studio Code	Uso libre
PostgreSQL	Uso libre
Django	Uso libre
Total	0

Otros costos en el desarrollo del sistema

COSTOS RECURRENTES	
Descripción	Costo
Consumo energético	500 C\$ (15\$)
Papelería (incluimos la documentación)	1500 C\$ (15\$)
Total	2000

6.3.1. Recursos humanos

El equipo de desarrollo del sistema web para CableMax estuvo compuesto por 2 personas, las cuales se asignaron roles específicos a cumplir y colaborativos en otros escenarios:

RECURSOS HUMANOS				
Etapa	Duración	Recurso Personal	Costo Total	Descripción
Análisis, diseño y codificación	4 meses	1	900 \$	la misma persona se encarga de analizar, diseñar y codificar
Diseño de pruebas y ejecución	2 semanas y 1 día	1	300 \$	la misma persona que crea las pruebas, las ejecuta y documenta
Implementación	1 semana	2	400 \$	el equipo de desarrollo realiza en colaboración esta tarea
Total	4 meses, 3 semanas, 1 día.		1 600 \$	

6.3.2. Costo total del proyecto

Tras analizar los costos individuales para la realización del proyecto realizamos una tabla que permita ver los costos totales por categorías dicha tabla incluye: recursos tecnológicos, costo de obtención de software, gastos recurrentes y recursos humanos, teniendo como resultado lo siguiente:

COSTO TOTAL DEL PROYECTO		
Categoría	Costo	Descripción
Recursos tecnológicos	499 \$	El equipo necesario para operar el sistema en la empresa.
Costo de obtención de software	0 \$	Todas las licencias y herramientas de desarrollo de software necesarias para la creación del sistema web para la empresa CableMax.
Costos recurrentes	2000 C\$	Los costos recurrentes se refieren a los que no son directamente relacionados al desarrollo del proyecto, pero sin embargo son necesarios para cumplir algunas tareas. Por ejemplo, el consumo energético y la papelería.
Recursos Humanos	1600 \$	Los recursos humanos se componen del equipo necesario para la creación de software e implementación del mismo en este caso 2 desarrolladores junior.

6.4. Beneficios del sistema

Beneficios tangibles:

- Optimización del tiempo entre procesos, la empresa podrá realizar consultas (datos de clientes, facturas pagadas, observaciones) de manera rápida y ordenada.
- El uso del sistema permitirá a los dirigentes tener control de los pagos realizados, así mismo como de los clientes activos en la empresa.

Beneficios intangibles:

- Disponibilidad de información referente a la empresa (clientes, observaciones, pagos, facturas).

- Optimización de tareas por parte de los empleados al asumir un rol en el uso del sistema.
- Control de opiniones y quejas al momento de la realización de pagos por parte de los clientes.
- Seguridad en la manipulación de información sensible para la empresa (datos de clientes).
- Gestión de pagos de manera rápida y organizada.
- Gestión de cliente simplificada.

Ingeniería de requerimientos

La ingeniería de requerimientos tiene como objetivo dar tanto al cliente como al desarrollador, un mecanismo de aseguramiento para ambas partes, donde estén de acuerdo con el alcance funcional y técnico. (Inegma, 2021)

Comprende la transformación de una necesidad operacional en una descripción del sistema, debe ser claro tanto para el equipo de desarrollo como para el cliente.

6.4.1. Requerimientos funcionales del sistema

Número	Requerimiento	Prioridad
1	Mantener los datos de registro del cliente	Alta
2	El sistema deberá permitir gestionar la información de los clientes (Crear, Actualizar, eliminar y listar).	Media
3	El sistema será capaz de llevar el control de pagos realizados por los clientes (registro de pagos).	Alta
4	Se visualizarán reportes que faciliten la comprensión de la información relevante de los procesos de la empresa.	Alta
5	El sistema llevará un control de clientes que estén al día y estado moroso.	Alta
6	El sistema podrá crear facturas y recibos como manera de soporte en las transacciones realizadas por los clientes.	Alta
7	El sistema tendrá la opción de imprimir los comprobantes para funciones internas de la empresa.	Media

6.4.2. Requerimientos no funcionales

Los requerimientos no funcionales representan características generales y restricciones del sistema. Las métricas utilizadas para especificar los requerimientos no funcionales fueron: Rapidez, Usabilidad y Seguridad de datos.

Número	Requerimiento	Prioridad	Métrica
1	El sistema deberá responder de manera rápida y fluida, en las solicitudes de información	Alta	Rapidez
2	El sistema restringirá la información sensible en dependencia del rol asignado.	Alta	Seguridad de datos
3	Contará con una interfaz intuitiva que reduzca la curva de aprendizaje respecto a su uso.	Alta	Usabilidad
4	El sistema contará con formularios que detallen de manera clara y precisa la información solicitada	Alta	Usabilidad
5	El sistema permitirá el acceso desde cualquier computador con internet, siempre y cuando se cuente con un usuario y contraseña	Alta	Usabilidad

7. Marco Metodológico.

7.1. Fase I: Definición del proyecto

En este primer sprint, se elaboró y se construyó una lista, sobre las funcionalidades requeridas del sistema, en conjunto con el Product Owner, encargada de empresa de televisión por cable “Cable Max” en Kukra Hill (RAACS)

Se realizaron reuniones con el usuario, para identificar cómo funcionaban los procesos existentes y los problemas que había en la empresa.

7.2. Fase II: Análisis de requerimientos

Una vez obtenida la información, se analizarán los datos adquiridos y se definirán los requerimientos funcionales a través reuniones con la secretaria de la empresa cable Max. Las reuniones serán realizadas a través de video llamadas.

7.3. Fase III: Diseño

Una vez definidos los requerimientos funcionales y no funcionales, se procederá a diseñar el modelo de datos que utilizaría el sistema, mediante la implementación de la metodología de diseño de bases de datos relacionales, la cual plantea siguiente flujo:

Inicialmente, se aplicará el del diseño conceptual de la base de datos, esto es, identificar, por medio de los requerimientos funcionales definidos, las entidades y relaciones correspondientes a la realidad.

Luego, se realizará el diseño lógico, proceso en el cual, se transforman las entidades definidas, a objetos más amigables a los gestores de bases de datos, mediante el uso de técnicas de normalización (en este caso, se aplicará hasta la tercera forma normal) para garantizar la consistencia y evitar la redundancia de los datos.

7.4. Fase IV: Desarrollo del sistema

Terminado el diseño de la base de datos el sistema web propuesto contendrá los siguientes módulos que se detallan de acuerdo con su principal estructura:

Control y registro de cliente: Contendrá la información general del cliente (datos personales), así como también su estado actual

Control de pagos: este permitirá a la empresa tener la información de los pagos realizados o pendientes por cada cliente.

Consulta de información: esto permitirá a la empresa tener toda la información de antiguos pagos a la hora de algún reclamo presentado por un cliente

El servidor se ejecuta en un servidor de heroku, mediante este la secretaria de Cable Max podrá obtener reportes de pagos y estado de los clientes actuales de la empresa.

7.5. Fase V: Pruebas

En el transcurso del desarrollo del sistema web se mantendrá comunicación con el dueño del sistema y se esperan obtener varios aspectos a considerar para su mejor adaptación a los cambios en los procesos que se realizaban manualmente.

Una vez finalizado el proyecto monográfico se procede a realizar las siguientes:

Prueba de inicio de sesión.

Prueba de registro de cliente.

Prueba de control de realización de pago.

7.6. Fase VI: Implementación del sistema y capacitación

La implementación del sistema se llevará a cabo de manera modular con el objetivo de que el usuario final se adapte rápidamente al sistema. De esta manera una vez que los módulos son funcionales se muestran al personal para capacitarlos y responder sus preguntas.

8. Análisis y presentación de los resultados

8.1. Planeación

8.1.1. Historias del usuario

Las historias de usuario de CableMax son las siguientes:

- Acceso al sistema (usuarios)
- Registro de clientes
- Gestión de clientes
- Estado de cliente
- Facturación
- Registro de pagos
- Reportes
- Cobrador

Tabla 1 Historia de usuario acceso al sistema

Historia de usuario	
Número: 1	Usuario: Administrador, Cobrador
Nombre de la historia: Acceso al sistema	
Prioridad: Alta (Alta, media, baja)	Riesgo en el desarrollo: Media (Alta, media, baja)
Iteración asignada: 1	
Descripción: Los usuarios del sistema contarán con un nombre de usuario y contraseña para acceder a sus perfiles; dichas claves son generadas por el equipo de desarrollo.	
Observaciones: Cada usuario tiene acceso solo a las funciones permitidas en su perfil.	

Tabla 2 Historia de usuario Registro de cliente

Historia de usuario	
Número: 2	Usuario: Administrador
Nombre de la historia: Registro de cliente	
Prioridad: media (Alta, media, baja)	Riesgo en el desarrollo: Alta (Alta, media, baja)
Iteración asignada: 1	
Descripción: La empresa cuenta con un listado de clientes que serán digitalizados y almacenados en la base de datos del sistema.	
Observaciones: Los clientes no cuentan con un perfil para ingresar al sistema, sin embargo, sus datos permanecen en él.	

Tabla 3 Historia de usuario Gestión de clientes

Historia de usuario	
Número: 3	Usuario: Administrador
Nombre de la historia: Gestión de clientes	
Prioridad: media (Alta, media, baja)	Riesgo en el desarrollo: Media (Alta, media, baja)
Iteración asignada: 1	
Descripción: El sistema cuenta con un usuario llamado Administrador el cual tendrá acceso a todos los datos de los clientes pudiendo realizar las siguientes funciones de gestión: Lectura, Actualización de información, baja de servicio.	
Observaciones: El administrador es el único usuario capaz de realizar estas acciones.	

Tabla 4 Historia de usuario Estado de cliente

Historia de usuario	
Número: 4	Usuario: Administrador, Cobrador
Nombre de la historia: Estado del cliente	
Prioridad: media (Alta, media, baja)	Riesgo en el desarrollo: Media (Alta, media, baja)
Iteración asignada: 1	
<p>Descripción: Los clientes tienen estados de pago (mora y corriente)</p> <p>Se visualiza el estado del cliente para identificar problemas de pagos y si estos cuentan con antecedentes.</p>	
Observaciones: Todos los usuarios tienen permiso de revisar esta lista.	

Tabla 5 Historia de usuario registro de pagos

Historia de usuario	
Número: 5	Usuario: Administrador, Cobrador
Nombre de la historia: Registro de pagos	
Prioridad: Alta (Alta, media, baja)	Riesgo en el desarrollo: Media (Alta, media, baja)
Iteración asignada: 2	
<p>Descripción: Los clientes realizan pagos de manera mensual, los cuales deben registrarse para obtener los datos de solvencia o mora.</p>	
Observaciones: Los usuarios administradores y cobrador pueden registrar un pago.	

Tabla 6 Historia de usuario Facturación

Historia de usuario	
Número: 6	Usuario: Administrador
Nombre de la historia: Facturación	
Prioridad: Alta (Alta, media, baja)	Riesgo en el desarrollo: Media (Alta, media, baja)
Iteración asignada: 2	
Descripción: Los clientes necesitan facturas que validen la realización de pagos de servicios, así como la empresa lo necesita para llevar control de solvencia.	
Observaciones: El usuario administrador es el encargado de emitir dichas facturas y distribuirlas.	

Tabla 7 Historia de usuario Reportes

Historia de usuario	
Número: 7	Usuario: Administrador
Nombre de la historia: Reportes	
Prioridad: Alta (Alta, media, baja)	Riesgo en el desarrollo: Media (Alta, media, baja)
Iteración asignada: 2	
Descripción: Los reportes son efectuados para revisión de estrategias, control de pagos, auditorias y visualización general de parte del dueño.	
Observaciones: Los reportes son realizados por el administrador.	

Tabla 8 Historia de usuario Cobrador

Historia de usuario	
Número: 9	Usuario: Cobrador
Nombre de la historia: Cobrador	
Prioridad: media (Alta, media, baja)	Riesgo en el desarrollo: Media (Alta, media, baja)
Iteración asignada: 3	
Descripción: Este perfil será diseñado con vista móvil únicamente, ya que el cobrador accederá desde un celular, la única función de este perfil es lectura de datos del cliente y actualización de estado.	
Observaciones: Solo el usuario cobrador puede acceder a esta vista. solo estará disponible en vista móvil.	

8.1.2. Release Planning

Para la realización de plan de entregas se tomó en cuenta la prioridad y el orden cronológico de las historias.

Tabla 9 Release Planning Entregas

Historia	Iteración	Prioridad	Fecha de Inicio	Fecha de finalización
1	1	Alta	16/09/2020	18/09/2020
2	1	Media	21/09/2020	24/09/2020
3	1	Media	28/09/2020	01/10/2020
4	1	Media	05/10/2020	08/10/2020
5	2	Alta	12/10/2020	14/10/2020
6	2	Alta	16/10/2020	20/10/2020
7	2	Alta	21/10/2020	23/10/2020
8	3	media	24/10/2020	26/10/2020
9	3	media	27/10/2020	29/10/2020

8.2. Primera Iteración

En esta iteración encontramos las siguientes historias de usuario:

Tabla 10 Iteración 1

Número	Nombre
1	Acceso al sistema
2	Registro de cliente
3	Gestión de clientes
4	Estado de cliente

8.2.1. Tareas de ingeniería

En la siguiente tabla se detallan todas las tareas de ingeniería correspondiente a la iteración 1

Tabla 11 Tarea de ingeniería

Número de tarea	Número de historias	Nombre de la tarea
1	1	Creación de base de datos.
2	1	Diseño de interfaz de acceso.
3	1	Validación de usuarios.
4	1	Almacenamiento de usuarios en BD
5	2	Diseño de interfaz de administrador

6	2	Creación de tabla para clientes.
7	2	Diseño de interfaz para la gestión de clientes (Lectura, Almacenamiento)
8	3	Validación de datos ingresados
9	4	Creación de tabla de validación de estado de cliente
10	4	Crear vista para listar el estado de clientes

Tabla 12 Tarea de ingeniería 1 para historia de usuario 1

TAREA DE INGENIERÍA	
Número de tarea: 1	Número de historia: 1
Nombre de la tarea: Creación de base de datos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha Inicio: 18/09/2020	Fecha Fin: 19/09/2020
Programador Responsable: Abner Zúniga	
Descripción: Se realizará la base de datos que permita almacenar toda la información que gestionará el sistema web de la empresa.	

Tabla 13 Tarea de ingeniería 2 para historia de usuario 1

TAREA DE INGENIERÍA	
Número de tarea: 2	Número de historia: 1
Nombre de la tarea: Diseño de interfaz de acceso.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha Inicio: 18/09/2020	Fecha Fin: 21/09/2020
Programador Responsable: Abner Zúniga	
Descripción: Se realizará un diseño de inicio de sesión donde los usuarios digitaran sus credenciales para acceder al sistema.	

Tabla 14 Tarea de ingeniería 3 para historia de usuario 1

TAREA DE INGENIERÍA	
Número de tarea: 3	Número de historia: 1
Nombre de la tarea: Validación de usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 22/09/2020	Fecha Fin: 22/09/2020
Programador Responsable: Carmen Treminio	
Descripción: Se efectúa confirmación de usuarios para mostrar el perfil correspondiente a las credenciales brindadas.	

Tabla 15 Tarea de ingeniería 4 para historia de usuario 1

TAREA DE INGENIERÍA	
Número de tarea: 4	Número de historia: 1
Nombre de la tarea: Almacenamiento de usuarios en BD.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha Inicio: 22/09/2020	Fecha Fin: 22/09/2020
Programador Responsable: Abner Zúniga	
Descripción: Se almacenará los datos necesarios para el inicio de sesión de cada usuario.	

Tabla 16 Tarea de ingeniería 5 para historia de usuario 2

TAREA DE INGENIERÍA	
Número de tarea: 5	Número de historia: 2
Nombre de la tarea: Diseño de interfaz de administrador.	
Tipo de tarea: Desarrollo	Puntos estimados: 5
Fecha Inicio: 22/09/2020	Fecha Fin: 28/09/2020
Programador Responsable: Abner Zúniga	
Descripción: Se realizará un diseño para el usuario administrador el cual contará con todos los permisos de escritura.	

Tabla 17 Tarea de ingeniería 6 para historia de usuario 2

TAREA DE INGENIERÍA	
Número de tarea: 6	Número de historia: 2
Nombre de la tarea: Creación de BD para cliente.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 28/09/2020	Fecha Fin: 28/09/2020
Programador Responsable: Carmen Treminio	
Descripción: Se necesita una base de datos de cliente, la empresa la digitaliza para utilizarla en el sistema.	

Tabla 18 Tarea de ingeniería 7 para historia de usuario 2

TAREA DE INGENIERÍA	
Número de tarea: 7	Número de historia: 2
Nombre de la tarea: Diseño de interfaz para la gestión de clientes (Lectura, almacenamiento, actualización)	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha Inicio: 29/09/2020	Fecha Fin: 1/10/2020
Programador Responsable: Abner Zúniga	
Descripción: Se realizará la interfaz de gestión de usuario, a través de la misma el administrador podrá manipular los registros de la empresa.	

Tabla 19 Tarea de ingeniería 8 para historia de usuario 3

TAREA DE INGENIERÍA	
Número de tarea: 8	Número de historia: 3
Nombre de la tarea: Validación de datos ingresados.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 1/10/2020	Fecha Fin: 2/10/2020
Programador Responsable: Carmen Treminio	
Descripción: El administrador realizará una comprobación de los datos ingresados para cotejar con los registros existentes.	

Tabla 20 Tarea de ingeniería 8 para historia de usuario 4

TAREA DE INGENIERÍA	
Número de tarea: 9	Número de historia: 4
Nombre de la tarea: Creación de tabla de estado de cliente.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 3/10/2020	Fecha Fin: 3/10/2020
Programador Responsable: Abner Zuniga	
Descripción: Se creará en la base de datos una tabla especial que emita los estados del cliente.	

Tabla 21 Tarea de ingeniería 10 para historia de usuario 4

TAREA DE INGENIERÍA	
Número de tarea: 10	Número de historia: 4
Nombre de la tarea: Crear vista lista estado de clientes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 4/10/2020	Fecha Fin: 6/10/2020
Programador Responsable: Abner Zúniga	
Descripción: Se creará una interfaz donde se listarán los estados del cliente.	

8.2.2. Tarjetas CRC

En las tablas 21-23 encontramos las tarjetas CRC (Clase-Responsabilidad-Colaboración) que fueron de gran utilidad en el desarrollo de la primera iteración

Tabla 22 Tarjeta CRC Clientes

Clase: Clientes	
Responsabilidad	Colaboración
Almacenar datos de los clientes (no, contrato, Nombre, Apellido, numero, dirección)	información clientes
Verificar datos de los clientes (Listado de clientes)	

Tabla 23 Tarjeta CRC Gestión de clientes

Clase: Gestión de clientes	
Responsabilidad	Colaboración
Listar, actualizar, o eliminar clientes de ser necesario.	Clientes

Tabla 24 Tarjeta CRC Estado de cliente

Clase: Estado de cliente	
Responsabilidad	Colaboración
Registrar estado de cliente (activo o inactivo)	Clientes
Listar el estado de todos los clientes	Clientes

8.2.3. Pruebas de aceptación

El International Software Testing Qualification Board (ISTQB) define la “Aceptación” como: Pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema. (PMOinformatica, s.f.)

En la siguiente tabla se detallan las pruebas de aceptación correspondiente a la primera iteración en las cuales se utilizaron los siguientes criterios de evaluación: procesos de negocios integrados en el sistema, formularios y procedimientos de usuarios.

Tabla 25 Tabla pruebas de aceptación de la primera iteración

Número de prueba	Número de historia	Nombre de prueba
1	1	Acceso al sistema
2	2	Registro de cliente
3	3	Gestión de clientes
4	4	Estado de cliente

Descripción de pruebas de aceptación

CASO DE PRUEBA: Acceso al sistema	
Código: 1	Nº historia de usuario: 1
Historia de usuario: Acceso al sistema	
condiciones de ejecución: El usuario que realice la prueba debe contar con sus credenciales de acceso (nombre de usuario y contraseña)	
<p>Entrada/pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Acceder a la pantalla principal. 2. Llenar el formulario de inicio de sesión introduciendo su nombre de usuario y contraseña. 3. luego pulsar el botón de inicio de sesión. 	
Resultado esperado: Acceso al perfil y funciones del mismo, según el usuario (definido por rol).	
Evaluación de la prueba: la prueba se completó correctamente.	

CASO DE PRUEBA: Registro de cliente	
Código: 2	N° historia de usuario: 2
Historia de usuario: Registro de cliente	
Condiciones de ejecución: El administrador tendrá que entrar en su perfil y acceder al apartado de registro de clientes para realizar esta prueba.	
Entrada/pasos de ejecución: <ul style="list-style-type: none"> 1. Acceder al apartado de registro de clientes. 2. Llenar el formulario con los datos de registro de cliente (no_contrato, nombre, apellido, dirección, celular). 3. presionar el botón guardar. 	
Resultado esperado: Cliente registrado en la base de datos.	
Evaluación de la prueba: la prueba se completó correctamente.	

CASO DE PRUEBA: Gestión de clientes	
Código: 3	N° historia de usuario: 3
Historia de usuario: Gestión de clientes	
Condiciones de ejecución: El administrador tendrá que entrar en su perfil y acceder al apartado “clientes”, en el tendrá la opción de actualizar registros, listarlos o eliminarlos.	
Entrada/pasos de ejecución: <ul style="list-style-type: none"> 1. Acceder al apartado de gestión de clientes. 2. seleccionar una de las opciones disponibles: listar, eliminar o actualizar. 	
Resultado esperado: Listado, actualizado o eliminado de un cliente.	
Evaluación de la prueba: la prueba se completó correctamente.	

CASO DE PRUEBA: Estado de cliente	
Código: 4	Nº historia de usuario: 4
Historia de usuario: Estado de cliente	
Condiciones de ejecución: El administrador tendrá que entrar en su perfil y acceder al apartado de gestión de clientes.	
Entrada/pasos de ejecución: <ul style="list-style-type: none"> 1. Acceder al apartado de gestión de clientes. 2. Seleccionar un cliente, presionar el botón mostrar. 3. Mirar en la tabla el apartado “estado de cliente” 	
Resultado esperado: Estado del cliente listado.	
Evaluación de la prueba: la prueba se completó correctamente.	

8.2.4. Diseño

El primer diseño con el que nos encontramos en esta iteración es el de la base de datos el cual ilustra las tablas que contiene el sistema y sus relaciones

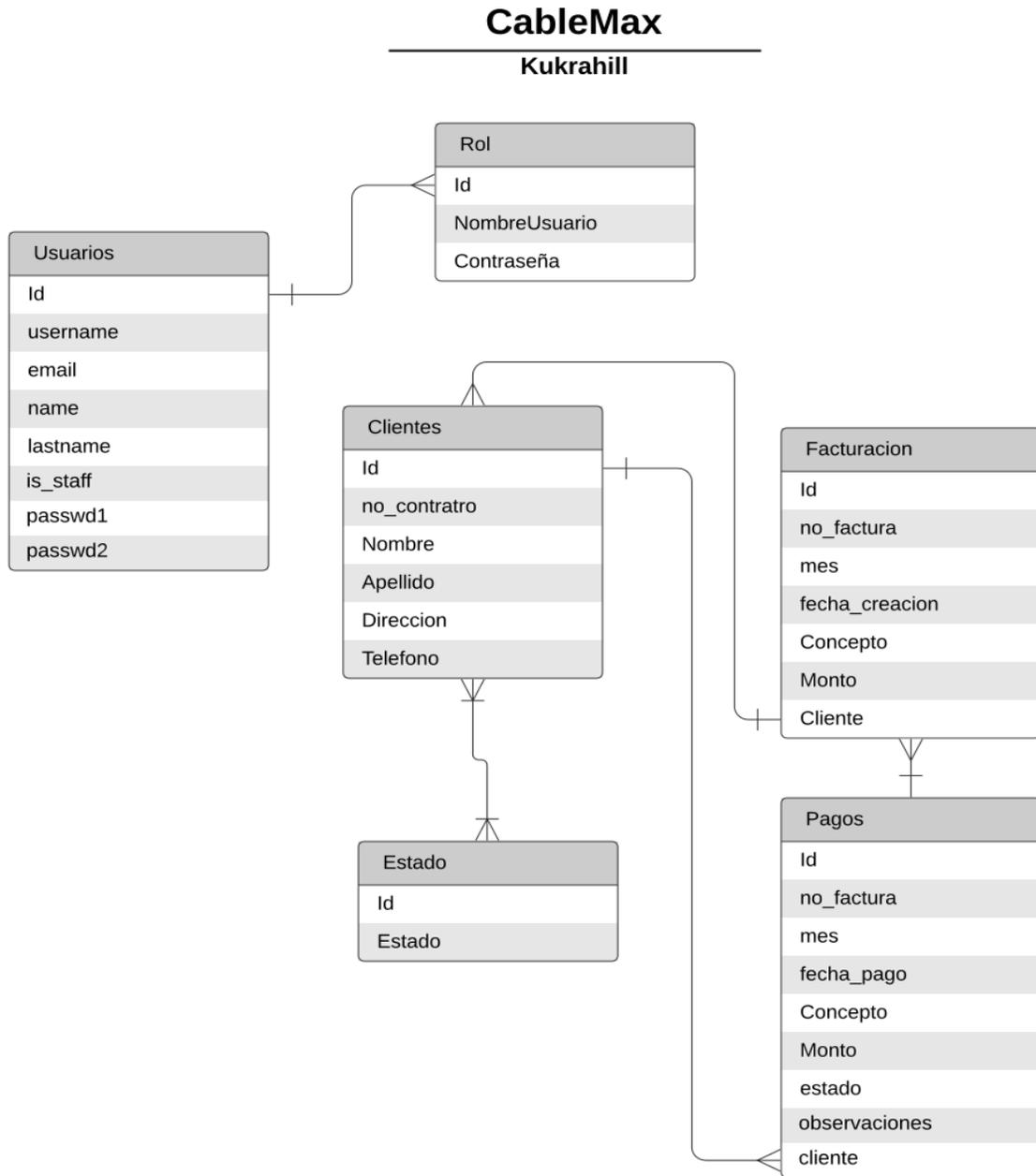


Ilustración 2 Diagrama de bases de datos

8.2.5. Boceto

Como guía para el diseño de interfaces realizamos un boceto con la finalidad de ilustrar el concepto creativo a utilizar en el sistema web, para la primera iteración obtuvimos los siguientes:

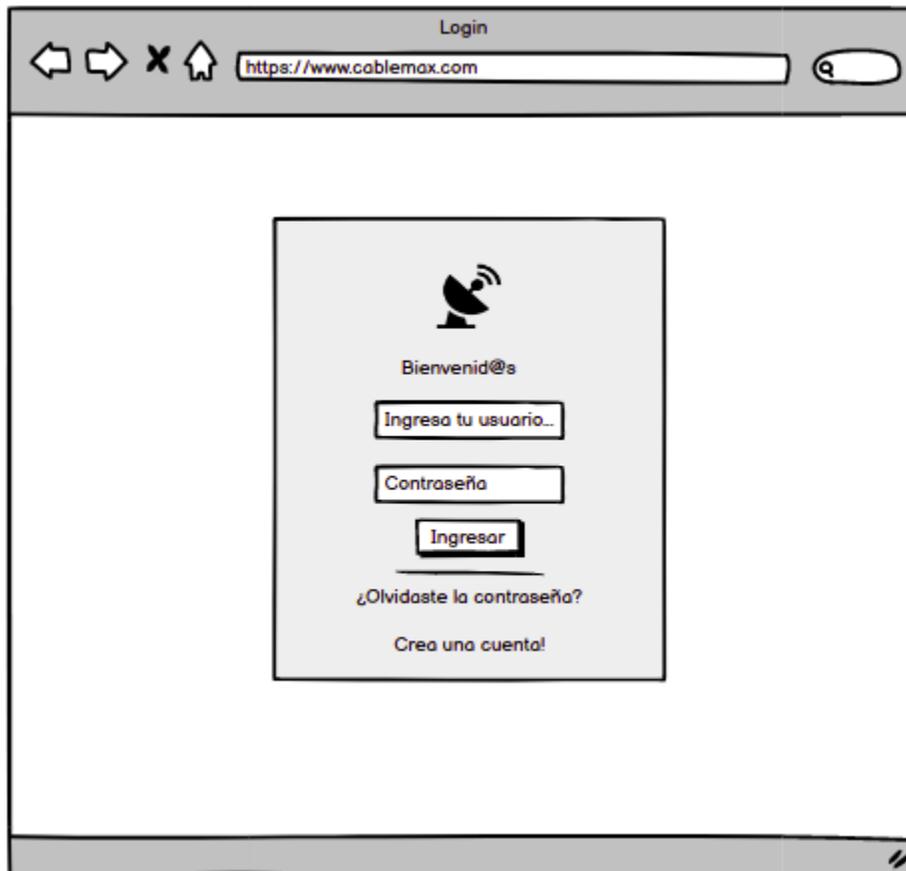


Ilustración 3 Boceto iteración 1 - pantalla principal - login

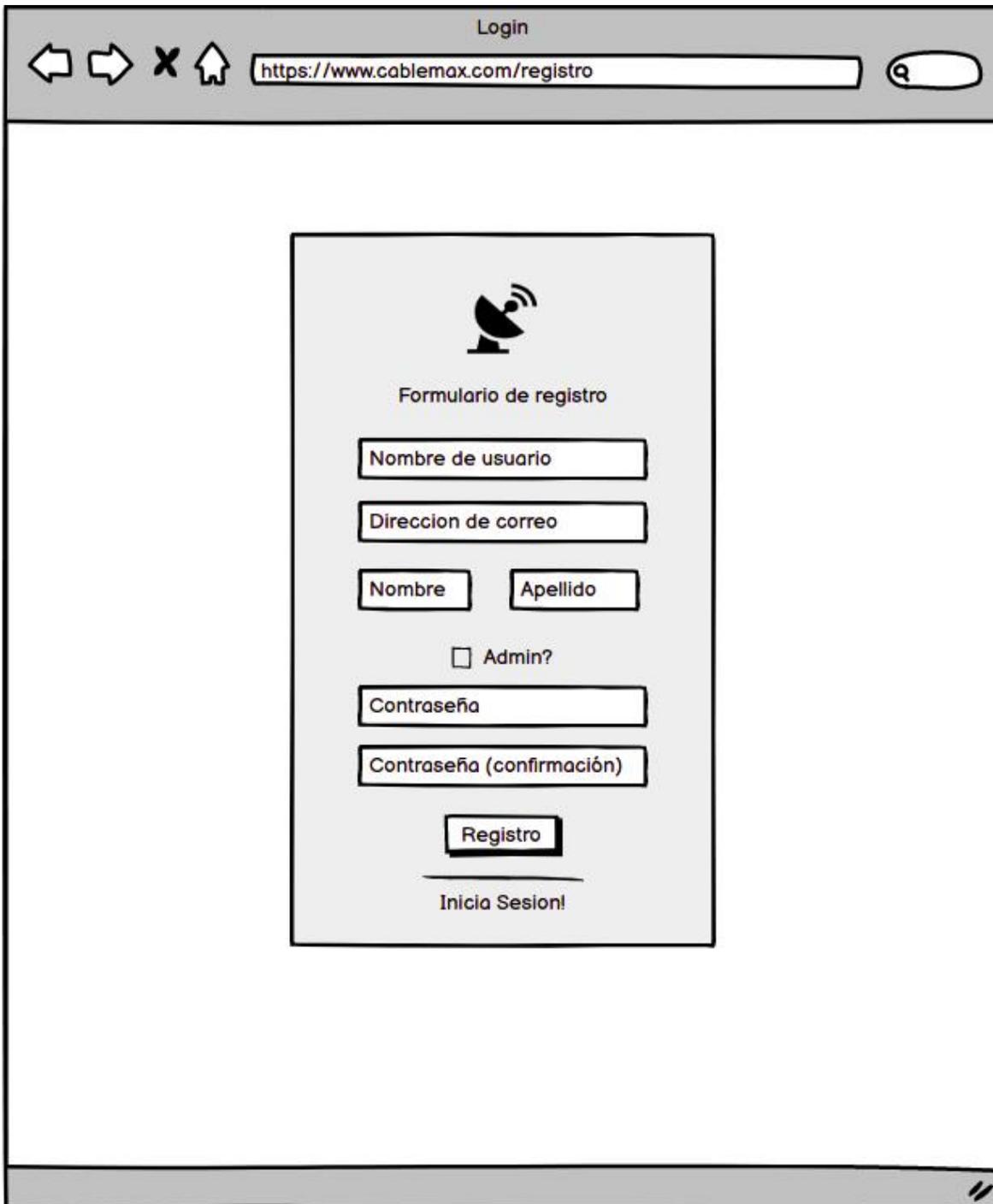


Ilustración 4 Boceto 2 iteración 1 - Pantalla de registro de usuarios

CREAR CLIENTE

No contrato

Nombre	Apellido
Dirección	Telefono
 Guardar	 Cancelar

Ilustración 5 Boceto 3 iteración 1 - Registro de clientes.

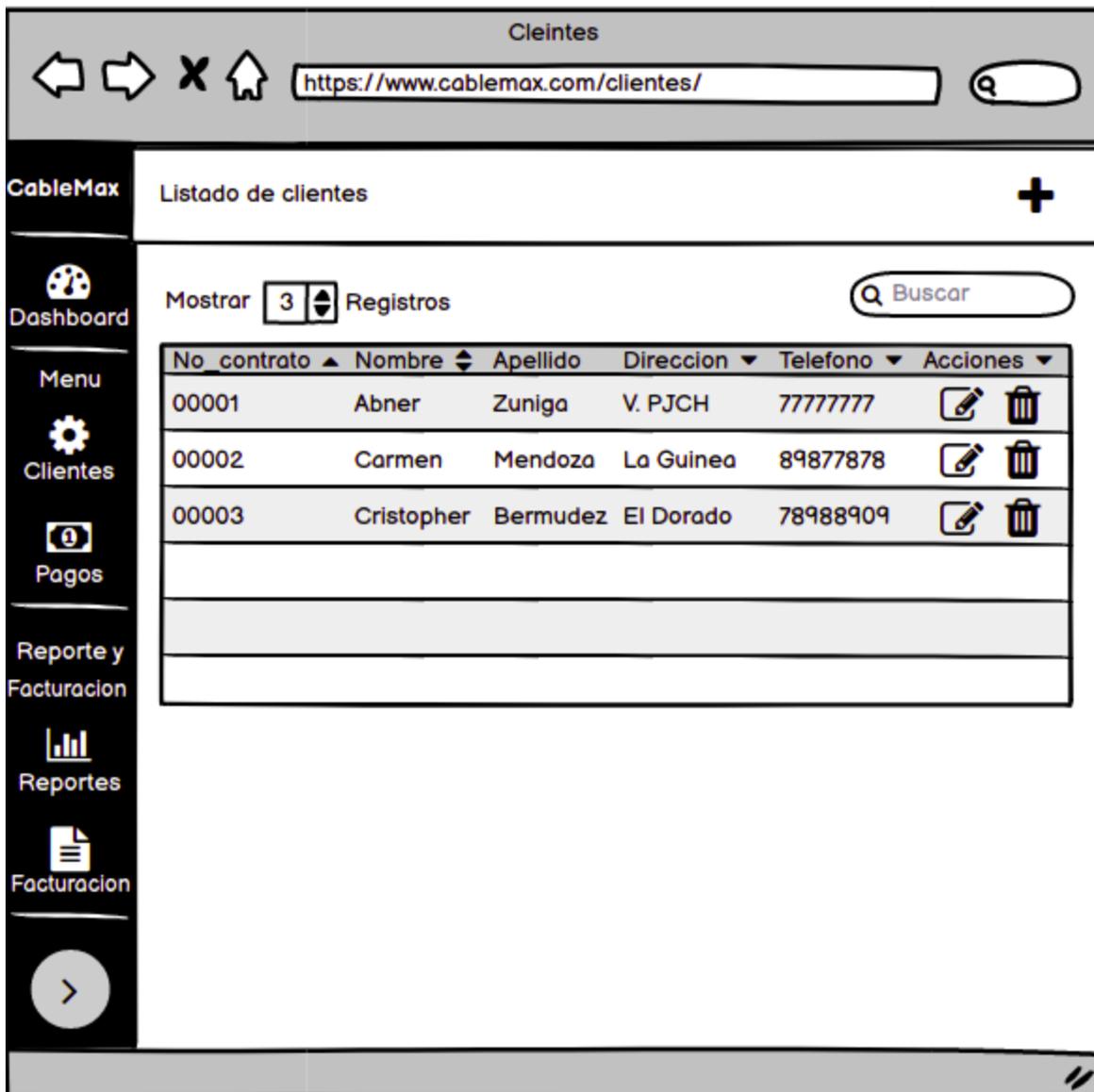
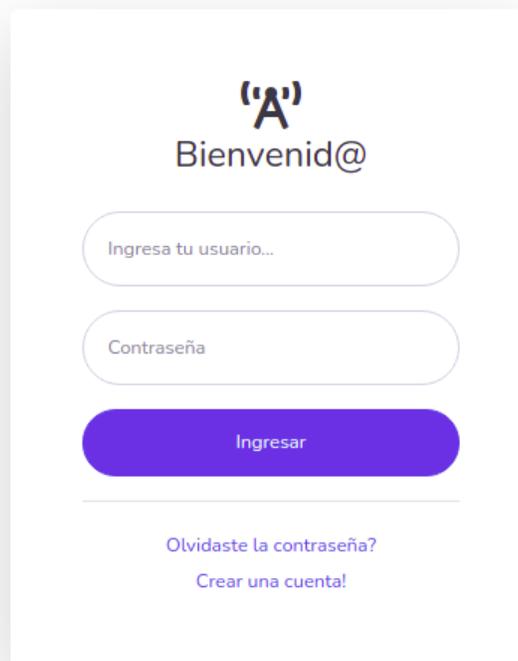


Ilustración 6 Boceto 4 Iteración 1 - Gestión de clientes.

Al finalizar la primera iteración y tomando en cuenta los bocetos realizados obtenemos las siguientes pantallas:

Correspondiente a la primera historia de usuario tenemos la pantalla de acceso al sistema, si no tenemos una cuenta nos permite crear una desde la misma pantalla, accediendo al link: “crear una cuenta”.



La imagen muestra una interfaz de usuario para el inicio de sesión. En la parte superior, hay un ícono de un 'A' dentro de unos corchetes, seguido del texto 'Bienvenid@'. Debajo de esto, hay dos campos de entrada de texto con bordes redondeados: el primero contiene el texto 'Ingresa tu usuario...' y el segundo 'Contraseña'. Debajo de los campos, hay un botón rectangular con bordes redondeados y un fondo azul, con el texto 'Ingresar' en blanco. En la parte inferior, hay un enlace azul que dice 'Olvidaste la contraseña?' y un enlace azul que dice 'Crear una cuenta!'.

Ilustración 7 Ingresar al sistema

Registro de usuarios

Este formulario es utilizado para crear usuarios que manipulan el sistema (no clientes) para que un usuario tenga permiso a visualizar todos los apartados debemos dar un check al boton: “admin?”

'A' Formulario de Registro

Nombre de usuario

Dirección de correo electrónico

Nombre

Apellidos

Admin?

Contraseña

Contraseña (confirmación)

Registro

Ilustración 8 Registrar nuevo usuario

CREAR CLIENTE

No contrato

Nombre

Apellido

Dirección

Teléfono

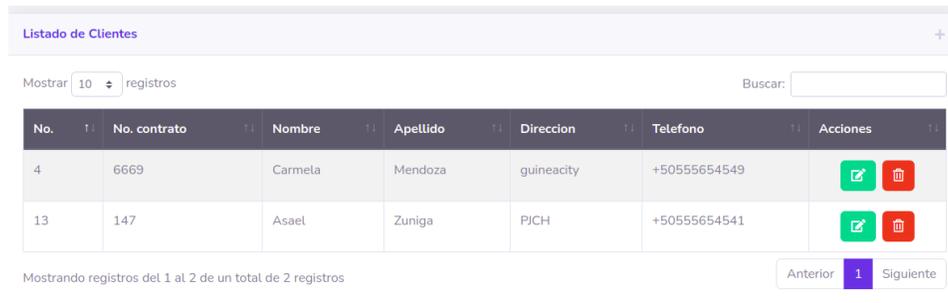
 Guardar

 Cancelar

Ilustración 9 Crear cliente

Una vez ingresados nuestras credenciales de acceso entramos al apartado cliente, correspondiente a las historias de usuarios: 2,3 y 4

Gestión de clientes (actualización, lectura y eliminación de los mismos)



No.	No. contrato	Nombre	Apellido	Dirección	Telefono	Acciones
4	6669	Carmela	Mendoza	guineacity	+50555654549	 
13	147	Asael	Zuniga	PJCH	+50555654541	 

Ilustración 10 Actualización Lectura y Eliminación de cliente

8.2.6. Codificación

Fragmento de código correspondiente a la primera iteración:

```
from .models import Cliente, Pago
from .forms import ClienteForm, PagoForm
```

```
# Create your views here.
```

```
class clientesView(LoginRequiredMixin, generic.ListView):
    model = Cliente
    template_name = 'clientes_list.html'
    context_object_name = 'obj'
    login_url = 'bases:login'

    @method_decorator(usuario_permitidos(usuario_permitidos=['admin']))
    def dispatch(self, *args, **kwargs):
        return super().dispatch(*args, **kwargs)
```

```
class clientesNew(SuccessMessageMixin, LoginRequiredMixin,
                 generic.CreateView):
    model = Cliente
    template_name = 'form_cliente.html'
    context_object_name = 'obj'
```

```
form_class = ClienteForm
success_url = reverse_lazy('principal:clientes_list')
login_url = 'bases:login'
success_message = 'Usuario Creado'
@method_decorator(usuarios_permitidos(usuarios_permitidos=['admin']))
def dispatch(self, *args, **kwargs):
    return super().dispatch(*args, **kwargs)
```

```
class clientesEdit(SuccessMessageMixin, LoginRequiredMixin,
                  generic.UpdateView):
    model = Cliente
    template_name = 'form_cliente.html'
    context_object_name = 'obj'
    form_class = ClienteForm
    success_url = reverse_lazy('principal:clientes_list')
    login_url = 'bases:login'
    success_message = 'Usuario Editado'
```

8.3. Segunda Iteración

En la segunda iteración encontramos las siguientes historias de usuario:

Tabla 26 Iteración 2

Número	Nombre
5	Registro de Pagos
6	Facturación
7	Reportes

8.3.1. Tareas de ingeniería

En la siguiente tabla se detallan todas las tareas de ingeniería correspondiente a la iteración 2

Tabla 27 Tarea de Ingeniería

Número	Número de historias	Nombre de la tarea
1	5	Creación de tabla Facturación.
2	5	Cron Job (proceso en segundo plano) para la creación de facturas el 1 de cada mes.
3	5	Diseño de interfaz para mostrar facturas creadas.
4	5	Diseño de interfaz para mostrar las facturas pagadas.
5	6	Creación tabla Pagos
6	6	Diseño de interfaz para la realización de pagos.
7	6	Generación de recibo para validación de pagos (una vez que se realiza el pago)

8	7	Diseño de interfaz para la visualización de reportes.
9	7	Creación de grafico que muestre el porcentaje de pagos realizados dividido por mes.
10	7	Reporte de todos los pagos realizados en el mes.
11	7	Reporte total de clientes, clientes en estado moroso, clientes en estado "activo".
12	7	Diseño de vista de observaciones obtenidas (descritas por el cliente al momento del cobro).
13	7	Reporte de clientes nuevos por mes

Tabla 28 Tarea de ingeniería 1 para historia de usuario 5

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 1	Número de historia: 5
Nombre de la tarea: Creación de tabla facturación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 08/02/2021	Fecha Fin: 08/02/2021
Programador Responsable: Carmen Treminio	
Descripción: Se crea la tabla facturación para poder almacenar y mostrar todas las facturas generadas para los clientes de la empresa.	

Tabla 29 Tarea de ingeniería 2 para historia de usuario 5

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 2	Número de historia: 5
Nombre de la tarea: Cron Job para la creación de facturas el 1 de cada mes.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha Inicio: 08/02/2021	Fecha Fin: 12/02/2021
Programador Responsable: Abner Zuniga	
Descripción: Se programa un job que se ejecute el primer día de cada mes con el objetivo de generar facturas a los clientes activos en la empresa.	

Tabla 30 Tarea de ingeniería 3 para historia de usuario 5

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 3	Número de historia: 5
Nombre de la tarea: Diseño de interfaz para mostrar facturas creadas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 15/02/2021	Fecha Fin: 17/02/2021
Programador Responsable: Abner Zuniga	
Descripción: Se realizará un diseño que permita lista todas las facturas creadas, estas tendrán el estado: pendiente.	

Tabla 31 Tarea de ingeniería 4 para historia de usuario 5

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 4	Número de historia: 5
Nombre de la tarea: Diseño de interfaz para mostrar facturas pagadas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 17/02/2021	Fecha Fin: 19/02/2021
Programador Responsable: Abner Zuniga	
Descripción: Se realizará un diseño que permita lista todas las facturas pagadas, estas tendrán el estado: pagado.	

Tabla 32 Tarea de ingeniería 5 para historia de usuario 6

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 5	Número de historia: 6
Nombre de la tarea: Creación tabla pagos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 18/02/2021	Fecha Fin: 18/02/2021
Programador Responsable: Carmen Treminio	
Descripción: Se creará una tabla que permita almacenar los pagos realizados.	

Tabla 33 Tarea de Ingeniería 6 para historia de usuario 6

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 6	Número de historia: 6
Nombre de la tarea: Diseño de interfaz para la realización de pagos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 19/02/2021	Fecha Fin: 20/02/2021
Programador Responsable: Carmen Treminio	
Descripción: Se diseñará una interfaz que permita conectar la tabla de pagos con la interfaz del sistema.	

Tabla 34 Tarea de ingeniería 7 para historia de usuario 6

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 7	Número de historia: 6
Nombre de la tarea: Generación de recibo para validación de pagos (una vez que se realiza el pago).	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 19/02/2021	Fecha Fin: 20/02/2021
Programador Responsable: Abner Zúniga	
Descripción: Se emite un recibo que valide la realización de pagos.	

Tabla 35 Tarea de ingeniería 8 para historia de usuario 7

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 8	Número de historia: 7
Nombre de la tarea: Diseño de interfaz para la visualización de reportes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 22/02/2021	Fecha Fin: 23/02/2021
Programador Responsable: Abner Zúniga	
Descripción: Diseño de interfaz para mostrar todos los reportes generados.	

Tabla 36 Tarea de ingeniería 9 para historia de usuario 7

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 9	Número de historia: 7
Nombre de la tarea: Creación de grafico que muestre el porcentaje de pagos realizados dividido por mes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 24/02/2021	Fecha Fin: 24/02/2021
Programador Responsable: Abner Zúniga	
Descripción: Creación de un gráfico usando ChartJS para mostrar el porcentaje de pagos realizados por mes.	

Tabla 37 Tarea de Ingeniería 10 para historia de usuario 7

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 10	Número de historia: 7
Nombre de la tarea: Reporte de todos los pagos realizados en el mes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 25/02/2021	Fecha Fin: 25/02/2021
Programador Responsable: Carmen Treminio	
Descripción: Reporte de todos los pagos realizados en el mes corriente.	

Tabla 38 Tarea de ingeniería 11 para historia de usuario 7

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 11	Número de historia: 7
Nombre de la tarea: Reporte total de clientes, clientes en estado moroso, clientes en estado “activo”.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 26/02/2021	Fecha Fin: 01/03/2021
Programador Responsable: Carmen Treminio	
Descripción: Se mostrará en una tabla todos los clientes en estado moroso.	

Tabla 39 Tarea de ingeniería 12 para usuario 7

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 12	Número de historia: 7
Nombre de la tarea: Diseño de vista de observaciones obtenidas (descritas por el cliente al momento del cobro).	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 02/03/2021	Fecha Fin: 04/03/2021
Programador Responsable: Carmen Treminio	
Descripción: Se mostrará en una tabla todas las observaciones obtenidas por los clientes al momento de pagar, con el objetivo de visualizar los principales problemas (de existir) o recomendaciones.	

8.3.2. Tarjetas CRC

En las tablas 43-49 encontramos las tarjetas CRC (Clase-Responsabilidad-Colaboración) que fueron de gran utilidad en el desarrollo de la segunda iteración

Tabla 40 Tarjeta CRC Usuarios

Clase: Usuarios	
Responsabilidad	Colaboración
Almacenar la información necesaria para la creación de un usuario del sistema (secretaria - cobrador)	Usuarios

Tabla 41 Tarjeta CRC Facturación

Clase: Facturación	
Responsabilidad	Colaboración
Almacenar la información necesaria para la creación de una factura (las facturas son autocreadas para los clientes de la empresa)	Clientes Pagos
Visualización de las facturas existentes.	Clientes Pagos Estado

Tabla 42 Tarjeta CRC Pagos

Clase: Pagos	
Responsabilidad	Colaboración
Registrar los pagos de facturas efectuados por clientes.	Facturación Clientes
Visualizar los pagos realizados y por quien fueron realizados.	Clientes. Estados.

Tabla 43 Tarjeta CRC Recibo

Clase: Recibo	
Responsabilidad	Colaboración
Generar comprobante de pagos como soporte para la empresa y el cliente mismo.	Clientes Facturación pagos
Visualizar comprobantes	Facturación Clientes

Tabla 44 Tarjeta CRC Reportes

Clase: Reportes	
Responsabilidad	Colaboración
Resumir el estado de la empresa de manera visual.	Clientes Pagos Estado

Tabla 45 Tarjeta CRC Estado de factura

Clase: Estado de factura	
Responsabilidad	Colaboración
Brindar el estado de la factura (Pagado - en mora)	Pagos Facturación
Listar en tablas diferentes los estados de las mismas a manera de reporte.	Reportes

Tabla 46 Tarjeta CRC Observaciones

Clase: Observaciones	
Responsabilidad	Colaboración
Mostrar a manera de reporte las observaciones (de existir) realizadas por los clientes al momento de pagar.	Clientes Pagos

8.3.3. Pruebas de aceptación

En la siguiente tabla se detallan las pruebas de aceptación correspondiente a la segunda iteración.

Número de prueba	Número de historia	Nombre de prueba
1	5	Visualización de facturas generadas en el mes.
2	6	Realización de pago de una factura.
3	6	Visualización de pago.
4	7	Visualización de reportes generados.

Descripción de pruebas de aceptación

CASO DE PRUEBA: Facturación	
Código: 1	Nº historia de usuario: 5
Historia de usuario: Facturación	
Condiciones de ejecución: El usuario debe haber iniciado sesión (Administrador o Cobrador).	
Entrada/pasos de ejecución: <ol style="list-style-type: none">1. Acceder al módulo de facturación.2. Presionar el botón facturas pendientes.	
Resultado esperado: Visualización de una tabla con las facturas en estado pendiente.	
Evaluación de la prueba: la prueba se completó correctamente.	

CASO DE PRUEBA: Registro de pagos	
Código: 2	Nº historia de usuario: 6
Historia de usuario: Registro de pagos	
Condiciones de ejecución: El usuario debe haber iniciado sesión (Administrador o Cobrador).	
Entrada/pasos de ejecución:	

<ol style="list-style-type: none"> 1. Acceder al módulo de facturación. 2. Presionar el botón facturas pendientes. 3. Una vez dentro presionar el botón pagar.
<p>Resultado esperado: Formulario que pre-carga los datos del pago, y permite al usuario de sistema cambiar el estado de pendiente a pagado y añadir observaciones de estimarlo conveniente.</p>
<p>Evaluación de la prueba: la prueba se completó correctamente.</p>

CASO DE PRUEBA: Pagos	
Código: 3	Nº historia de usuario: 6
Historia de usuario: Registro de pagos	
Condiciones de ejecución: El usuario debe haber iniciado sesión (Administrador o Cobrador).	
Entrada/pasos de ejecución: <ol style="list-style-type: none"> 1. Acceder al módulo de pago y presionar el botón pago. 	
Resultado esperado: Visualización de tabla con los pagos realizados.	
Evaluación de la prueba: la prueba se completó correctamente.	

CASO DE PRUEBA: Reportes	
Código: 4	Nº historia de usuario: 7
Historia de usuario: Reportes	
Condiciones de ejecución: El usuario "administrador" debe tener la sesión iniciada.	
Entrada/pasos de ejecución: <ol style="list-style-type: none"> 1. Acceder al módulo de reportes. 2. Observar los reportes disponibles. 3. Presionar una opción. 	
Resultado esperado: Visualización de reportes.	
Evaluación de la prueba: la prueba se completó correctamente.	

8.3.4. Boceto

Para la realización del diseño de interfaces correspondientes a la segunda iteración utilizamos los siguientes bocetos:

Facturas pagadas

Mostrar Registros

No_factura	Cliente	Mes	Fecha Creación	Concepto	Monto	Estado	Acciones
01	Abner Zuniga	Abril	01-04-21	Pago Mensual	C\$250	Pagado	 
02	Carmen Mendoza	Abril	01-04-21	Pago Mensual	C\$250	Pagado	 
03	Cristopher Bermudez	Abril	01-04-21	Pago Mensual	C\$250	Pagado	 

Ilustración 11 Boceto 1 Iteración 2 - Facturas pagadas.

Facturas pendientes

Mostrar Registros

No_factura	Cliente	Mes	Fecha Creación	Concepto	Monto	Estado	Acciones
01	Abner Zuniga	Abril	01-04-21	Pago Mensual	C\$250	Pendiente	
02	Carmen Mendoza	Abril	01-04-21	Pago Mensual	C\$250	Pendiente	
03	Cristopher Bermudez	Abril	01-04-21	Pago Mensual	C\$250	Pendiente	

Ilustración 12 Boceto 2 Iteración 2 - Facturas pendientes.

Verificación de pagos

Cliente	//	
Mes	Monto pagado (C\$)	
Dirección	Pendiente	
Observaciones		
Generar	Cancelar	

Ilustración 13 Boceto 3 Iteración 2 - Pago de facturas.

Reportes

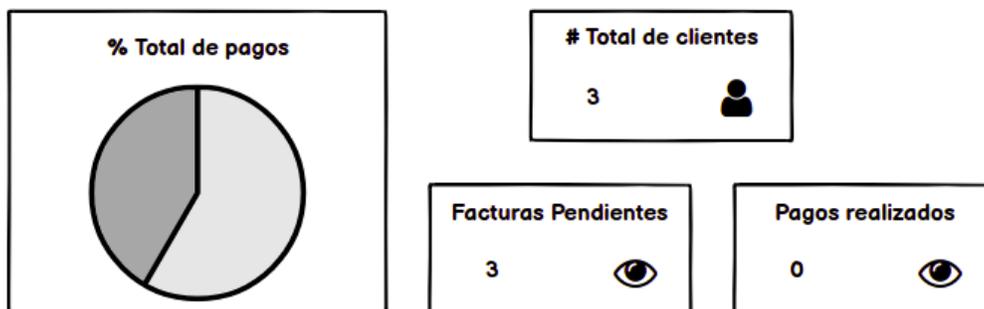


Ilustración 14 Boceto 4 Iteración 2 - Reportes

Al finalizar la segunda iteración tomando como referencia los bocetos realizados obtuvimos las siguientes vistas:

FACTURAS PAGADAS

Mostrar registros Buscar:

No. factura	Cliente	Mes	Fecha creacion	Concepto	Monto	Estado	Acciones
81	Carmela Mendoza	Febrero	04-03-2021	Pago Mensual	C\$250.0	Pagado	 
84	Carmela Mendoza	Enero	07-03-2021	Pago Mensual	C\$250.0	Pagado	 

Mostrando registros del 1 al 2 de un total de 2 registros Anterior **1** Siguiente

Ilustración 15 Pago

FACTURAS PENDIENTES

Mostrar registros Buscar:

No. factura	Cliente	Mes	Fecha creacion	Concepto	Monto	Estado	Acciones
82	Asael Zuniga	Enero	01-03-2021	Pago Mensual	C\$250.0	Pendiente	
83	Asael Zuniga	Febrero	02-03-2021	Pago Mensual	C\$250.0	Pendiente	

Mostrando registros del 1 al 2 de un total de 2 registros Anterior **1** Siguiente

Ilustración 16 Listado de facturas pendientes

VERIFICACION DE PAGOS

Ciente:	Fecha de Pago:
<input type="text" value="13"/>	<input type="text" value="09/03/2021"/>
Mes:	Monto Pagado (C\$):
<input type="text" value="Enero"/>	<input type="text" value="250.0"/>
Concepto:	Estado:
<input type="text" value="Pago Mensual"/>	<input style="border: 1px solid #ccc;" type="text" value="Pendiente"/>
Observaciones:	
<div style="border: 1px solid #ccc; height: 60px; width: 100%;"></div>	
<input type="button" value="Generar"/> <input type="button" value="Cancelar"/>	

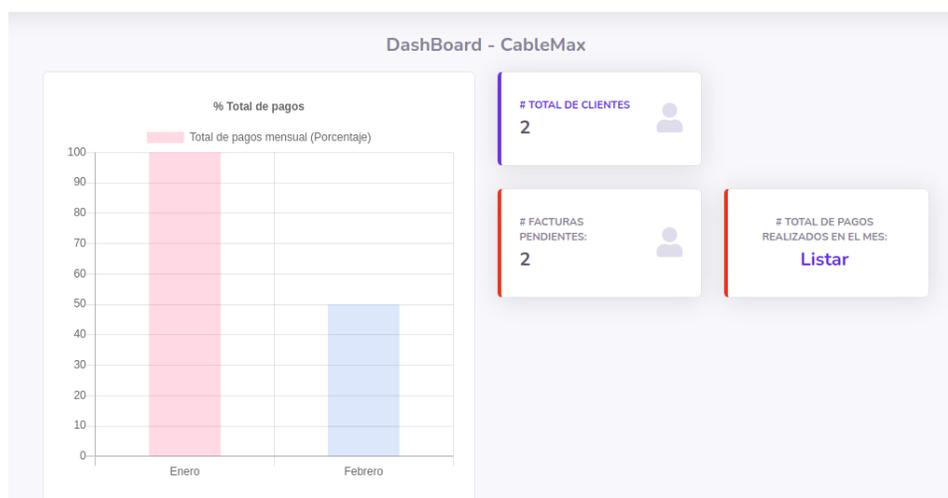
Ilustración 17 Generación de recibo

CableMax
 Direccion: Kukrahill
 FACTURA No. 82

Factura No. 82 Fecha Fact: 09-03-2021
No. Contrato: 147
Cliente: Asael Zuniga
Direccion: PJCH
Mes Pagado: Enero
Monto Total Pagado: C\$ 250.0

©CableMax

Ilustración 18 Visualización de recibos



Copyright © CableMax 2021

Ilustración 19 Reportes

8.3.5. Codificación

Fragmento de código correspondiente a la iteración 2:

```
class PagosView(LoginRequiredMixin, generic.ListView):
    model = Pago
    template_name = 'pagos_list.html'
    context_object_name = 'obj'
    login_url = 'bases:login'

def Pagos(request, compra_id=None):
    template_name = 'pagos.html'
    form_pagos = {}
    contexto = {}
    clientes_activos = Cliente.objects.filter(activo=True)
    print(clientes_activos)

    if request.method == 'GET':
        form_pagos = PagoForm()
        contexto = {'form_pagos': form_pagos, 'clientes_activos': clientes_activos}
        # print(compuesta)

    if request.method == 'POST':
        compuesta = Cliente.objects.annotate(search_name=Concat('nombre', Value(' '),
'apellido'))
        cliente = request.POST.get("cliente")

        mes = request.POST.get("mes")
        fecha_creacion = str(datetime.date.today())
        monto = request.POST.get("monto")
        concepto = request.POST.get("concepto")

    if not compra_id:
        cl = Cliente.objects.get(pk=cliente)

        pago_realizado = Pago(
            cliente = cl,
            mes = mes,
            fecha_creacion = fecha_creacion,
            monto = monto,
            concepto = concepto,
        )
        name = compuesta.filter(search_name__icontains=pago_realizado.cliente)
        mes = Pago.objects.filter(mes=mes, cliente=cl).first()
```

```
date_var = datetime.datetime.strptime(fecha_creacion, "%Y-%m-%d")

if mes and date_var:
    messages.error(request, 'La Factura del mes de {} ya fue generada para el
cliente {} {}'.format(mes, cl.nombre, cl.apellido))
    return redirect('principal:pagos_new')

elif pago_realizado:
    pago_realizado.save()
    print(pago_realizado.cliente)
    print(pago_realizado.mes)
    print(date_var.year)
    return redirect('principal:pagos_list') #TODO: CAMBIAR DIRECCION A RE-
CIBO

return render(request, template_name, contexto)
```

8.4. Tercera Iteración

En la tercera iteración encontramos las siguientes historias de usuario:

Tabla 47 Iteración 3

Número	Nombre
1	Perfil: Cobrador

8.4.1 Tareas de ingeniería

En la siguiente tabla se detallan todas las tareas de ingeniería correspondiente a la iteración 3

Tabla 48 Tarea de iteración

Número	Número de historias	Nombre de la tarea
1	8	Creación de perfil: cobrador.
2	8	Adaptación del sistema a entorno móvil.
3	8	Establecimiento de permisos.
4	8	Visualización de solo vistas permitidas.
5	8	Previsualización de vista móvil en el navegador y en un dispositivo.

Tareas de ingeniería:

Tabla 49 Tarea de Ingeniería 1 para historia de usuario 8

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 1	Número de historia: 8
Nombre de la tarea: Creación de perfil: cobrador.	
Tipo de tarea: Desarrollo	Puntos estimados: 3

Fecha Inicio: 09/03/2021	Fecha Fin: 10/03/2021
Programador Responsable: Carmen Treminio	
Descripción: Creación de un perfil con las funciones de cobrador.	

Tabla 50 Tarea de ingeniería 2 para historia de usuario 8

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 2	Número de historia: 8
Nombre de la tarea: Adaptación del sistema a entorno móvil.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha Inicio: 11/03/2021	Fecha Fin: 13/03/2021
Programador Responsable: Carmen Treminio	
Descripción: Adaptar la vista del cobrador a entorno móvil, ya que este la consultará únicamente desde un celular.	

Tabla 51 tarea de ingeniería 3 para historia de usuario 8

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 3	Número de historia: 8
Nombre de la tarea: Establecimiento de permisos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 15/03/2021	Fecha Fin: 15/03/2021
Programador Responsable: Carmen Treminio	
Descripción: Se limitan los permisos de vista en la interfaz para el cobrador, solo a pago y facturación.	

Tabla 52 Tarea de ingeniería 4 para historia de usuario 8

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 4	Número de historia: 8
Nombre de la tarea: Visualización de vistas permitidas.	

Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 16/03/2021	Fecha Fin: 17/03/2021
Programador Responsable: Carmen Treminio	
Descripción: Interfaz adaptada solo a módulos permitidos para el cobrador.	

Tabla 53 Tarea de ingeniería 5 para historia de usuario 8

TAREA DE INGENIERIA	
NÚMERO DE TAREA: 5	Número de historia: 8
Nombre de la tarea: Previsualización de vista móvil en el navegador y en un dispositivo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 18/03/2021	Fecha Fin: 19/03/2021
Programador Responsable: Abner Zuniga	
Descripción: Visualización de interfaz en dispositivo móvil (android) y emulado en el navegador.	

8.4.2. Tarjetas CRC

En la tabla 61 encontramos las tarjetas CRC (Clase-Responsabilidad-Colaboración) que fueron de gran utilidad en el desarrollo de la tercera iteración

Tabla 54 Tarjeta CRC Permisos

Clase: Permisos	
Responsabilidad	Colaboración
Limita el acceso a las vistas en dependencia del rol impuesto.	Roles usuarios

8.4.3. Pruebas de aceptación

En la siguiente tabla se detallan las pruebas de aceptación correspondiente a la tercera iteración.

Número de prueba	Número de historia	Nombre de prueba
1	8	Visualización de interfaz responsive en un dispositivo móvil.

CASO DE PRUEBA: Cobrador	
Código: 1	Nº historia de usuario: 8
Historia de usuario: Perfil: Cobrador	
Condiciones de ejecución: Debe existir un perfil para cobrador.	
Entrada/pasos de ejecución: Acceder a la pantalla principal e introducir las credenciales de acceso para cobrador.	
Resultado esperado: Perfil adaptativo a las pantallas de celular.	
Evaluación de la prueba: la prueba se completó correctamente.	

8.4.4. Boceto

Para la realización del diseño de interfaces correspondientes a la tercera iteración utilizamos los siguientes bocetos:

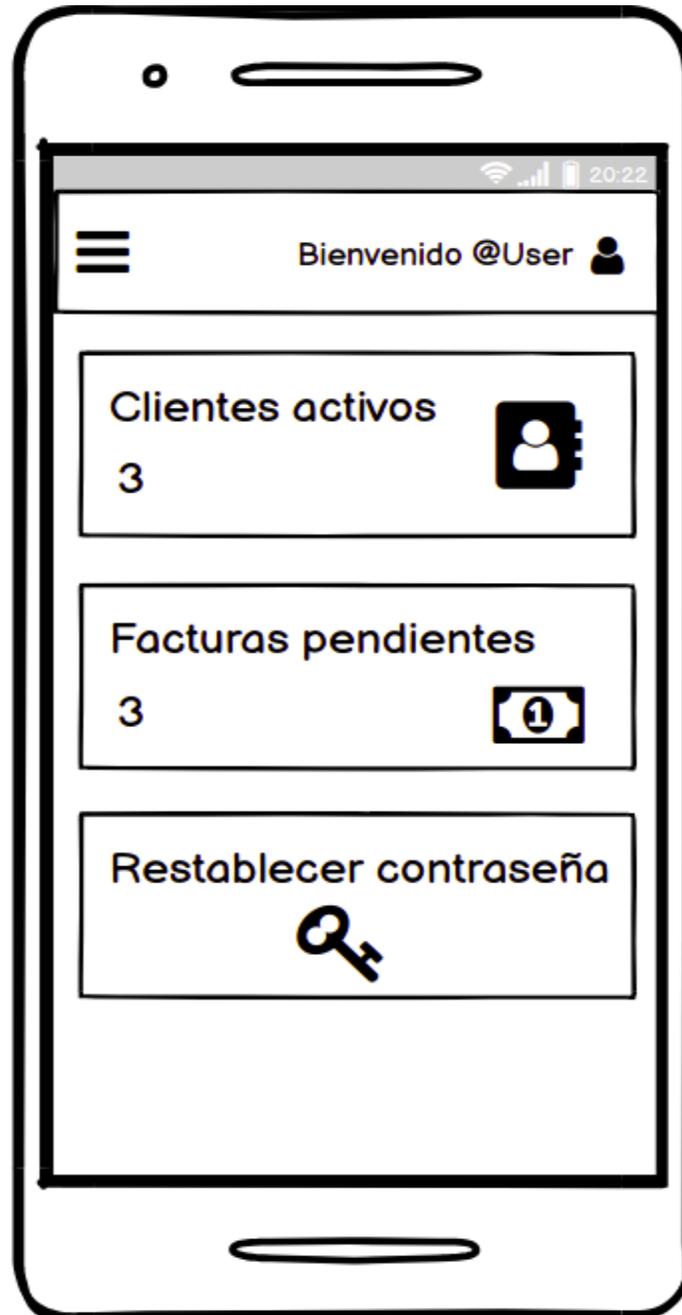


Ilustración 20 Boceto 1 Iteración 3 - Dashboard perfil cobrador.

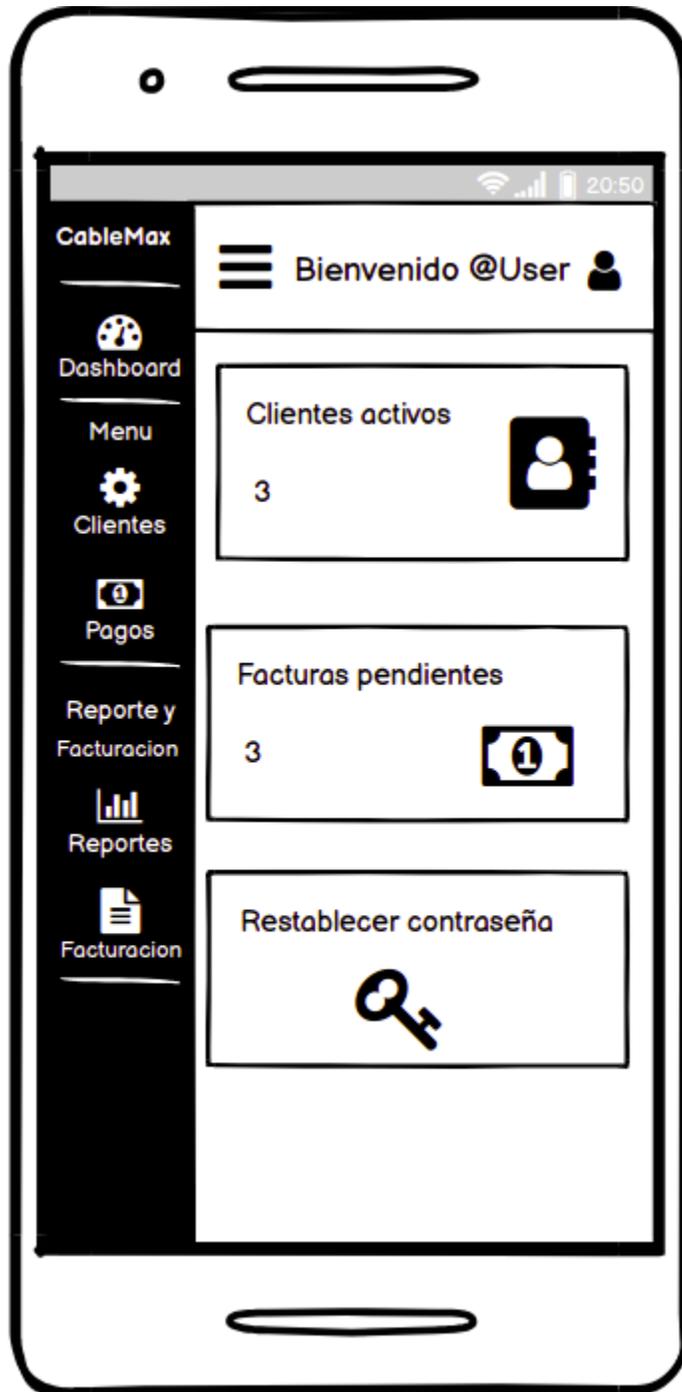


Ilustración 21 Boceto 2 Iteración 3 - Vista móvil del menu

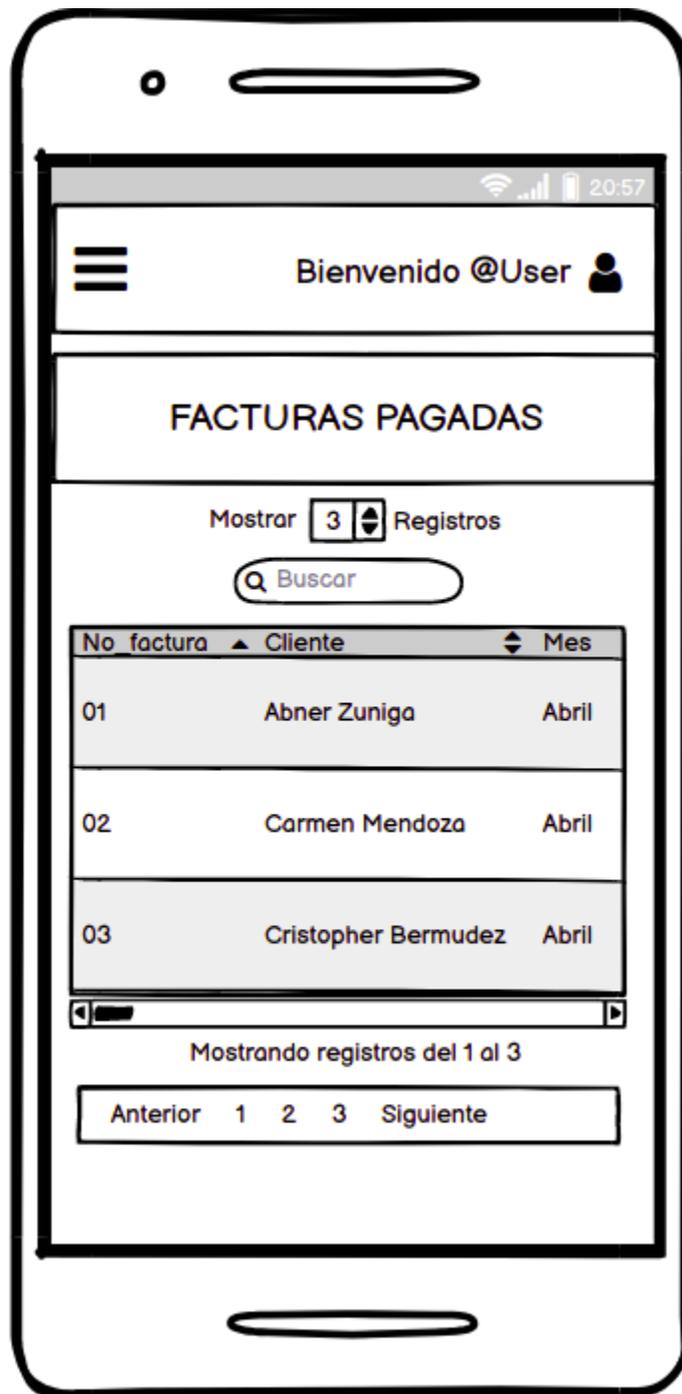


Ilustración 22 Boceto 3 Iteración 3 - Visualización de contenidos en dispositivos móviles

Al finalizar la iteración 3 utilizando los bocetos creados obtuvimos las siguientes vistas:

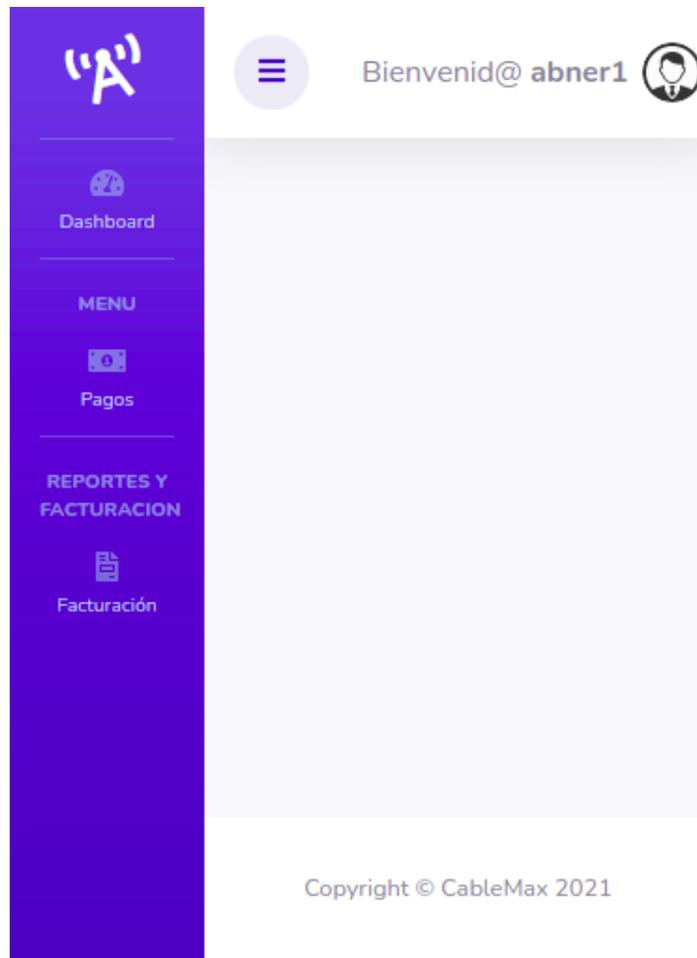


Ilustración 23 Vista general cobrador adaptada a móvil



FACTURAS PENDIENTES

Mostrar registros

Buscar:

No. factura ↑↓	Cliente ↑↓	Mes ↑↓	Fecha creacion
83	Asael Zuniga	Febrero	02-03-2021

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior

1

Siguiente

Ilustración 24 Vista de una tabla

8.4.5. Codificación

Fragmento de código correspondiente a la iteración 3:

```
from django.contrib.auth.models import Group
from .forms import RegistroForm

from principal.models import Cliente, Pago
# Create your views here.

class Home(LoginRequiredMixin, generic.TemplateView):
    template_name = 'base/base.html'
    login_url = 'bases:login'

    @method_decorator(users_roles)
    def dispatch(self, *args, **kwargs):
        return super().dispatch(*args, **kwargs)

def HomeCobrador(request):
    clientes_activos = Cliente.objects.filter(activo=True).count()
    FacturasPendientes = Pago.objects.filter(estado="Pendiente").count()

    return render(request, 'cobrador.html', {
        "clientes_activos": clientes_activos,
        "FacturasPendientes": FacturasPendientes,
    })
```

8.5. Métricas

Para poder evaluar los resultados del sistema web para la administración de clientes y gestión de cobros para la empresa de televisión por cable Cable Max y poder verificar si es capaz de satisfacer las expectativas de la empresa, se analizaron sus factores de éxito a partir de las historias de usuario, encontrándose que para ser exitosa debe:

- Disminuir el uso de papel.
- Requerir menos tiempo entre procesos.
- Ser fácil de usar.
- Brindar información de manera sencilla que pueda resumir información vital para la empresa.

8.5.1. Definición de métricas

Una vez definidos los factores de éxito, se determinaron las métricas a utilizar:

1.) Desempeño

1.1 Tiempo de respuesta

1.2 Numero de imprimibles

2.) Usabilidad (facilidad de uso)

2.1 interfaz simple e intuitiva

2.2 Resume la información vital para la empresa

Las medidas de desempeño se obtuvieron a través de la aplicación de una prueba comparativa con las herramientas actuales (Reportes manuales - Registros en papel) y el sistema web desarrollado, en el cual se cronometra el tiempo de respuesta y el número de impresiones que se necesitan para obtener información.

Las medidas de usabilidad se obtuvieron a través de una encuesta a la secretaria y dueño (principales usuarios del sistema) de su impresión sobre el uso del sistema en sus labores cotidianas.

8.5.2. Definición de instrumentos

Se diseñó una prueba que tiene como meta la obtención de información, en el rubro de cablemax hay dos categorías vitales las cuales son reportes y pagos, para ello veamos los siguientes casos:

Caso de prueba 1: Obtener reporte sobre estado de clientes (moroso - al día).

Caso de prueba 2: Obtener reporte de pagos realizados en el mes.

En el caso de la encuesta se utilizó una única respuesta para cada pregunta realizada al personal, de acuerdo a los siguientes valores:

Medida de evaluación	Escala de valores
Interfaz simple e intuitiva	Simple, Compleja
Facilidad de obtención de información	Si, No
Facilidad de automatización de procesos	Simple, Compleja

8.6. Resultados de la prueba

La prueba fue aplicada a los usuarios principales del sistema, los cuales son el CEO y secretaria de CableMax, los casos de prueba fueron realizados tanto con el método anterior como con la implementación del software.

8.6.1. Resultado Caso de prueba 1

Para la realización de este caso de prueba con el método actual, se debe recopilar uno a uno los clientes actuales, buscándolos en el folder de registro donde son almacenados

los recibos subdivididos por perfil de cliente, mientras con el uso del sistema solo se debe acceder al apartado de reportes y solicitar el reporte de estado de clientes.

La siguiente tabla detalla los resultados correspondientes al caso de prueba 1, reportados por los miembros de la empresa involucrados. La siguiente formula muestra el procedimiento para calcular el ahorro porcentual:

Ahorro =

$$x = \frac{\text{Resultado}_{\text{Metodo anterior}} - \text{Resultado}_{MA}}{\text{Resultado}_{\text{metodo anterior}}} * 100$$

CP. 1	Herramienta anterior (A)		Herramienta actual (B)		Ahorro (A-B)/A*100%	
	Tiempo de respuestas (S)	# de reportes	Tiempo de respuestas (s)	# de reportes	Tiempo (%)	# De reportes (%)
1	600	1	50	1	92%	0%
2	420	1	30	1	93%	0%
Media	510	1	45	1	92%	0%

8.6.2. Resultado Caso de prueba 2

Para la realización de esta prueba con el método actual de la empresa, se debe seguir los pasos vistos en el caso de prueba 1, solo con la variante que en este caso los recibos del mes no están agregados aún al perfil de cada cliente. Utilizando el sistema, debemos acceder al apartado de reportes y solicitar “pagos realizados en el mes”.

CP. 2	Herramienta anterior (A)		Herramienta actual (B)		Ahorro (A-B)/A*100%	
	Tiempo de respuestas (S)	# de reportes	Tiempo de respuestas (s)	# de reportes	Tiempo (%)	# De reportes (%)
1	420	1	60	1	%86	0%
2	300	1	45	1	%85	0%
Media	360	1	52	1	%85	0%

Tomando en consideración los resultados obtenidos tras la aplicación de ambos métodos para realizar acciones cotidianas de la empresa, obtuvimos que la utilización del sistema brinda un ahorro en promedio del 88% respecto al método manual que utiliza la empresa.

8.6.3. Resultados de la encuesta

Se aplicó la encuesta a los usuarios mayormente involucrados con el sistema los cuales son: CEO y secretaria. Esta consulta consistió en 3 preguntas para evaluar ambos métodos (Método anterior vs método actual) obteniendo que la implementación del sistema web facilita las búsquedas de información y muestra los reportes de una manera ordenada a través de una interfaz simple e intuitiva, siendo esta última una característica muy importante para los usuarios del software.

Encuesta	Interfaz simple e intuitiva		Facilidad de obtención de información		Facilidad de automatización de procesos	
	Método anterior	Método Actual	Método anterior	Método Actual	Método Anterior	Método Actual
1	Complejo	Simple	No	Si	Complejo	Simple
2	Complejo	Simple	No	Si	Complejo	Simple
Moda	Complejo	Simple	No	Si	Complejo	Simple

En conclusión, los resultados de la evaluación evidencian que la solución web presentada a la empresa CableMax será de utilidad en la realización de procesos en el día a día de sus trabajos, además queda demostrado que la interfaz es simple e intuitiva para los usuarios finales.

9. Conclusiones

Al realizar el sistema de facturación para la empresa de cable “CableMax”, se utilizó la metodología ágil xp, haciendo este más corto los tiempos de desarrollo y simplificando la división del trabajo.

El sistema esta creado para existir dentro de un servidor ya que se planea que el cobrador lo visite cuando realice los cobros y así poder llevar un orden de pagos realizados.

El diseño intuitivo aporta a los usuarios finales:

- Orden al buscar información.
- Reportes que facilitan el conocimiento de las estadísticas de la empresa.
- La posibilidad de escalabilidad para una posible expansión.

Por otra parte, con la realización de las pruebas hemos podido comprobar que es un sistema web a la medida por lo tanto cumple con los requerimientos puestos por la empresa.

A través de las historias de usuario se determinó los requerimientos para la creación del sistema web para la administración de clientes y gestión de cobros para la empresa de televisión por cable “Cable Max” en kukra hill con el personal de la empresa, por lo tanto, damos por concretado que el objetivo uno se cumplió satisfactoriamente.

Se diseñaron las interfaces y bases de datos tomando en cuenta los requerimientos obtenidos tras analizar las historias de usuario auxiliados de la herramienta UML casos de usos, la cual nos permitió entender los procesos a mayor profundidad, lo que nos permite dar por concluido de manera exitosa nuestro objetivo número dos.

El proceso de codificación permitió determinar las funciones necesarias para la creación de un sistema a la medida para la empresa de televisión por cable “Cable Max” haciendo uso del lenguaje de programación Python a través del framework Django y las buenas prácticas que el mismo promueve, por lo que podemos decir que el tercer objetivo se ha concretado de manera exitosa.

10. Recomendaciones

Se recomienda a la empresa tomar en cuenta lo siguiente:

- Al adquirir un servicio de hosting verificar que este sea compatible con Django y con CronJobs.
- Mantener a la mano el manual de usuarios por dudas existentes.
- Realizar mantenimiento periódico por posibles cambios.

11. Bibliografía

Alvarez, J. (3 de Mayo de 2019). Obtenido de <https://sercompetitivos.com/qu%C3%A9-es-una-api/>

CableSoft. (2016). Obtenido de https://www.cablessoft.tv/boxicableweb/?fbclid=IwAR2S_clt89-1PDjo1txxZ-4ft-2Vo75ZFMZ02QDB59IGSjnY-16ZyGgFGHw

Canejo, Y. (2015). *Sistemas de Informacion*. Venezuela. Obtenido de <https://es.scribd.com/document/262527930/Sistemas-de-Informacion-Informe?fbclid=IwAR2CZiCvEF7ObENGXyqRNQ1sZMNOFLKJx2Uk2QkSnI7KXQNTzvhlXbivKGw>

CHUBURU, L. (2020). Obtenido de <https://www.laurachuburu.com.ar/tutoriales/que-es-jquery-y-como-implementarlo.php?fbclid=IwAR0POQNaByOdGPI5TPu8rc4OicBMZAKMxHkbHjg-olOPHWSejypD7wMIyls>

Economipedia. (2021). Obtenido de <https://economipedia.com/definiciones/factura.html>

Febles Estrada, A. (2008). *Ecured*. Obtenido de https://www.ecured.cu/M%C3%A9tricas_de_software

Highsmith, J. (2001). Obtenido de <https://agilemanifesto.org/iso/es/manifesto.html>

iebschool. (2021). Obtenido de <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>

Inegma. (2021). Obtenido de <https://www.evaluandosoftware.com/gestion-requerimientos-proyecto-software-empresarial/>

LucidChart. (s.f.). Obtenido de Tutorial de diagramas de actividades UML: <https://www.lucidchart.com/pages/es/tutorial-diagrama-de-actividades-uml>

Lucidchart. (2021). Obtenido de <https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-datos>

Merino, u. P. (2010). Obtenido de <https://definicion.de/reporte/>

PMOinformatica. (s.f.). Obtenido de Pruebas de aceptación de software según el ISTQB: <http://www.pmoinformatica.com/2016/08/pruebas-aceptacion-software-istqb.html>

postgresql. (2021). Obtenido de <https://www.postgresql.org/about/?fbclid=IwAR1hQ08g6DgnT8kk2LYojptgBIS-Yi0hJr6pGYGFIkToJMY2YWLbN28row4>

- python*. (07 de Julio de 2020). Obtenido de https://wiki.python.org/moin/BeginnersGuide?fbclid=IwAR2ru0mcdF0KH5c4aB_gzAjem-9LwJyBxHLasp5xzplEfZpTZGzpvD0rbsA
- Quiroga, D. (Lunes de Septiembre de 2020). *Clibre*. Obtenido de <https://clibre.io/blog/por-secciones/codigo/item/433-pat>
- refactoring.guru*. (2014). Obtenido de <https://refactoring.guru/es/design-patterns/facade>
- Sistemas para servicio de television por cable Gestel*. (2015). Obtenido de <http://multisoftperu.com/pdf/gestel.pdf>
- Sistemas, T. 2.-P. (05 de Mayo de 2012). Obtenido de <http://tc2-proyectodegradoingdesistemas.blogspot.com/2012/05/que-es-y-para-que-sirve-css3.html>
- uniwebsidad*. (s.f.). Obtenido de <https://uniwebsidad.com/libros/django-1-0/capitulo-5/el-patron-de-diseno-mtv>
- Useit*. (18 de febrero de 2018). Obtenido de https://www.useit.es/blog/bootstrap-un-buen-framework?fbclid=IwAR1-RNT5WdBAANWtCygBeRmLBDTaqrEhmkiGeu_-ypkJELB4MJVhkvxD9XE

12. Anexos

Anexo A. Manual Técnico del sistema web para la administración de clientes y gestión de cobros para la empresa de televisión por cable “Cable Max” en Kukra Hill



Diccionario de datos

Usuarios

variable	tipo	descripción
id	integer	Identificador de registro
username	character varying	nombre de usuario
email	character varying	correo electrónico del usuario
name	character varying	nombre
last_name	character varying	apellido
is_staff	boolean	verifica permisos del usuario
password1	character varying	contraseña 1
password2	character varying	verificación de contraseña

Rol

Variable	tipo	descripción
id	integer	identificador de registro
username	character varying	nombre de usuario al que se asigna el rol
password1	character varying	contraseña

Clientes

Variables	Tipo	Descripción
id	integer	identificador de registro
no_contrato	character varying	número de contrato
nombre	character varying	nombre del cliente
apellido	character varying	apellido del cliente
direccion	character varying	dirección del cliente



telefono	integer	numero de contacto
----------	---------	--------------------

Estado

Variables	Tipo	Descripción
id	integer	Identificador de cliente por id
estado	character varying	estado del cliente (activo o inactivo)

Facturación

Variables	Tipo	Descripción
id	integer	identificador de registro
no_factura	integer	número de factura generada
mes	character varying	mes que se factura
fecha_creacion	timestamp with time zone	fecha en la que se crea la factura
monto	double precision	costo por mes
concepto	text	concepto de pago
cliente_id	integer	identificador de cliente

Pagos

Variables	Tipo	Descripción
id	integer	identificador de registro
no_factura	integer	no. asignado a la factura
mes	character varying	mes que se paga
fecha_pago	timestamp with time zone	fecha en la que se realiza el pago
concepto	text	concepto de pago
monto	double precision	monto pagado
estado	character varying	estado de la factura
observaciones	text	opiniones del cliente



cliente_id	integer	identificador de cliente
------------	---------	--------------------------



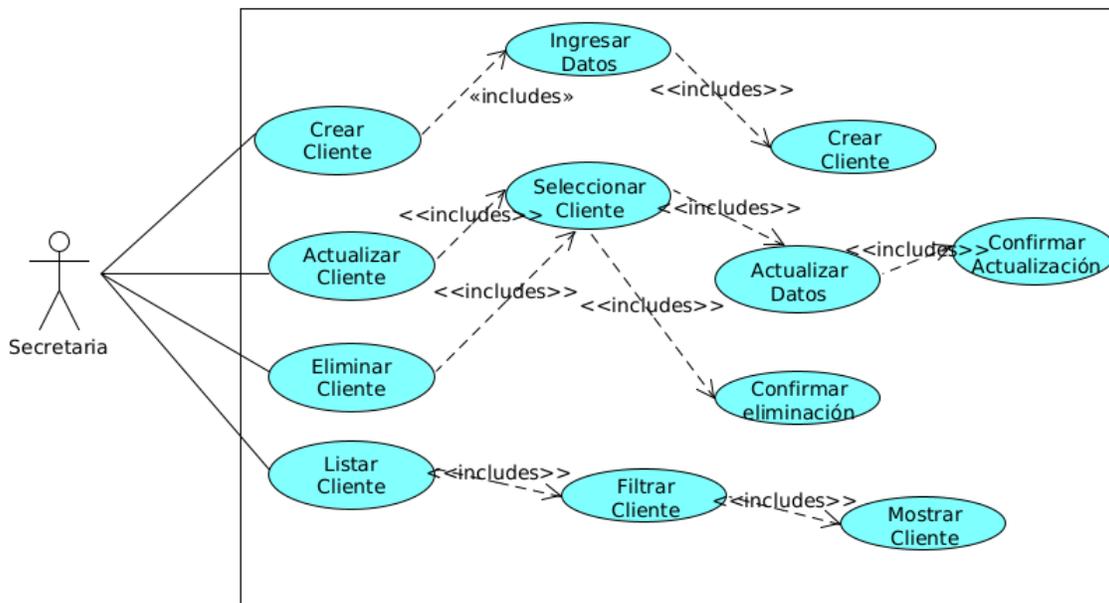
Casos de usos

Un caso de uso es la descripción de una actividad o acción. Un diagrama de casos de usos es una secuencia de acciones realizadas por alguien para llevar a cabo algún proceso, el cual tiene como objetivo producir un resultado observable y valioso para un usuario en particular. Permiten además hacer más fácil la comprensión de los procesos tanto para el equipo de desarrollo como para el usuario final.

Diagrama de casos de uso por procesos

Proceso 1: Gestión de clientes

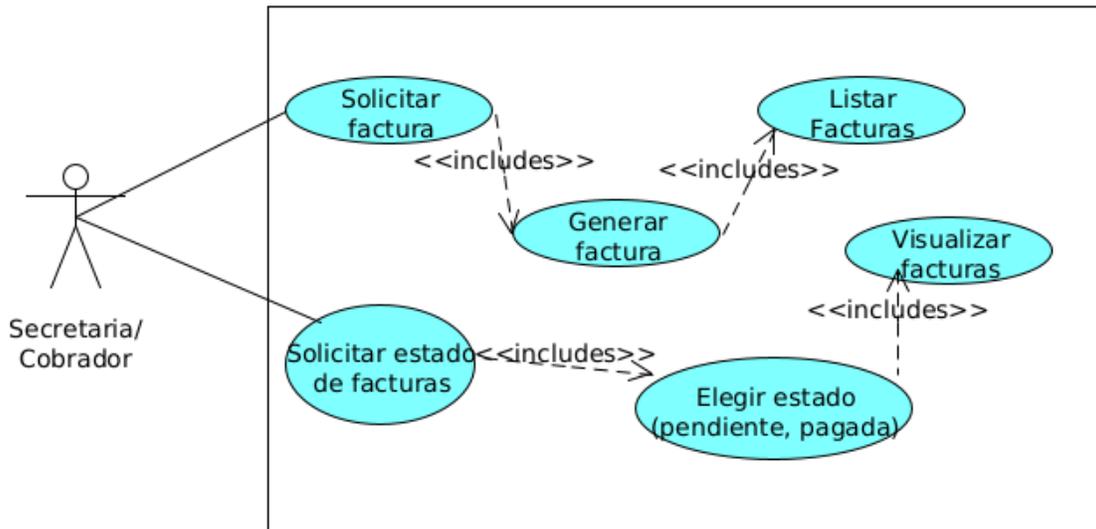
Solo el usuario secretaria podrá gestionar los clientes, los procesos de gestión incluyen: listado, creación, actualización y eliminación de registros.





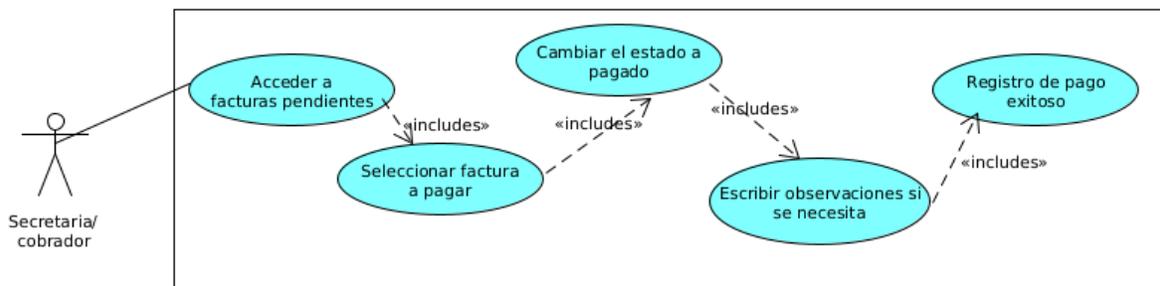
Proceso 2: Facturación

Ambos usuarios (cobrador, secretaria) del sistema pueden verificar la facturación. El proceso consiste en la creación y asignación de una factura por mes.



Proceso 3: Pagos

Ambos usuarios (cobrador y secretaria) son capaces de acceder a la función pagos. El proceso consiste en almacenar el pago por servicios de parte de un cliente en específico.





Proceso 4: Reportes.

Solo el usuario secretaria puede acceder a la sección de reportes. El proceso consiste en mostrar de manera resumida el estado de la empresa.

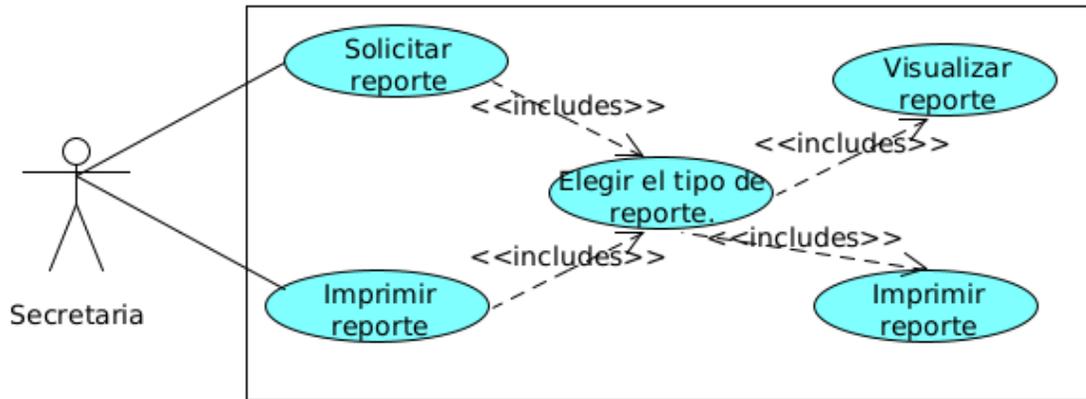




Diagrama de actividades para listar un cliente

Cable Max | Kukrahill

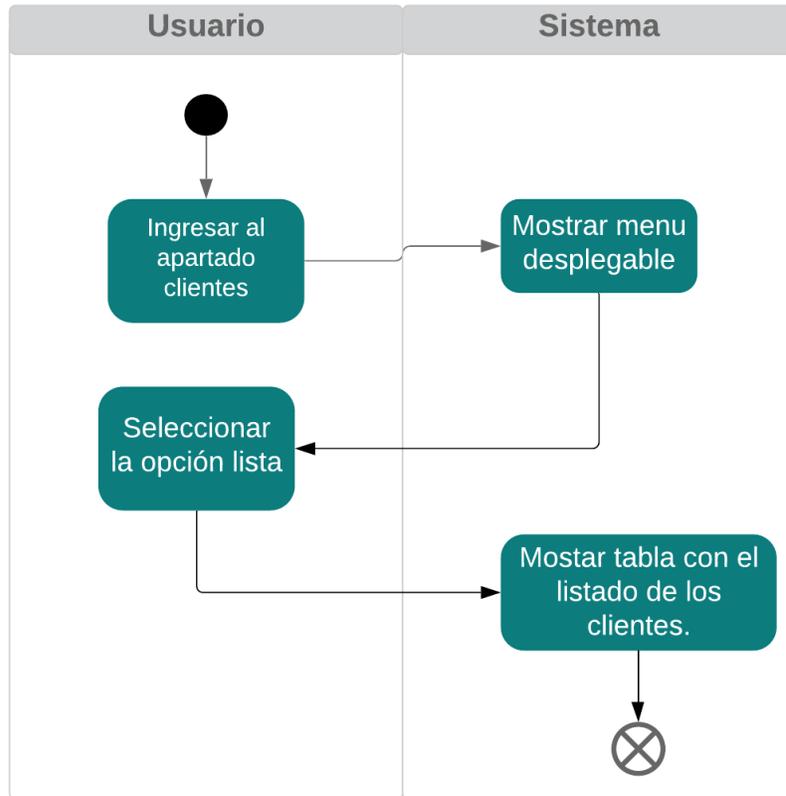




Diagrama de actividades para listar un cliente

Cable Max | Kukrahill

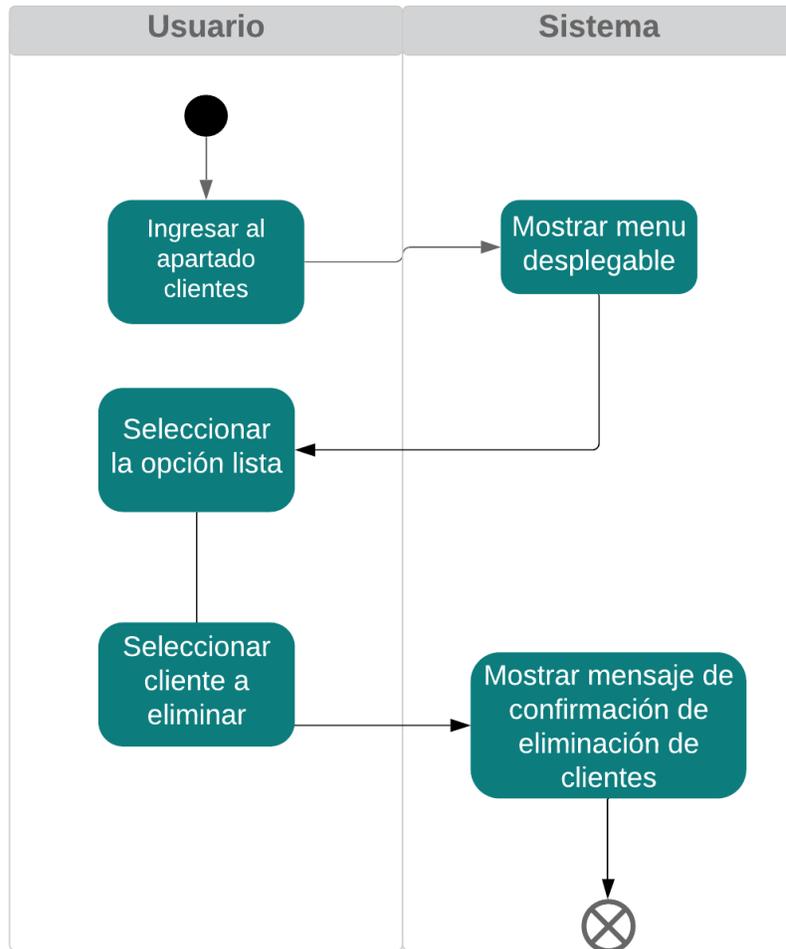




Diagrama de actividades para realizar un pago

Cable Max | Kukrahill

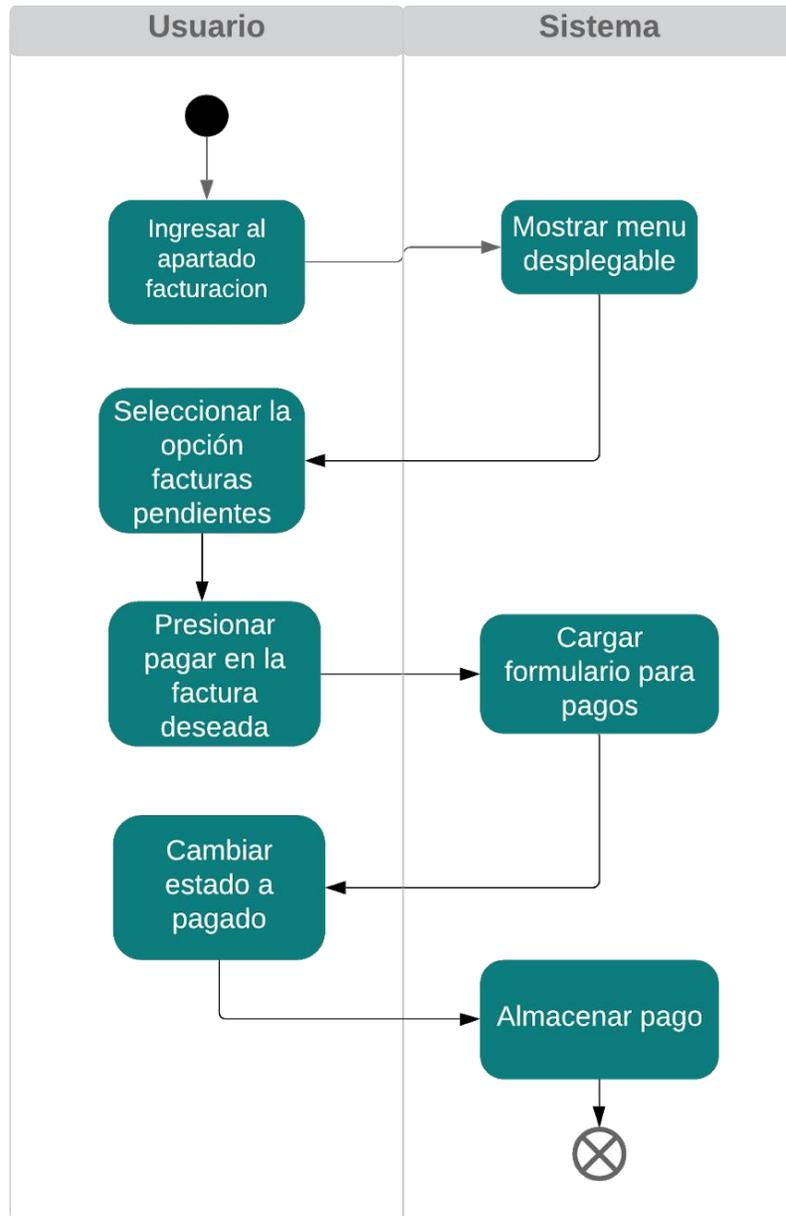
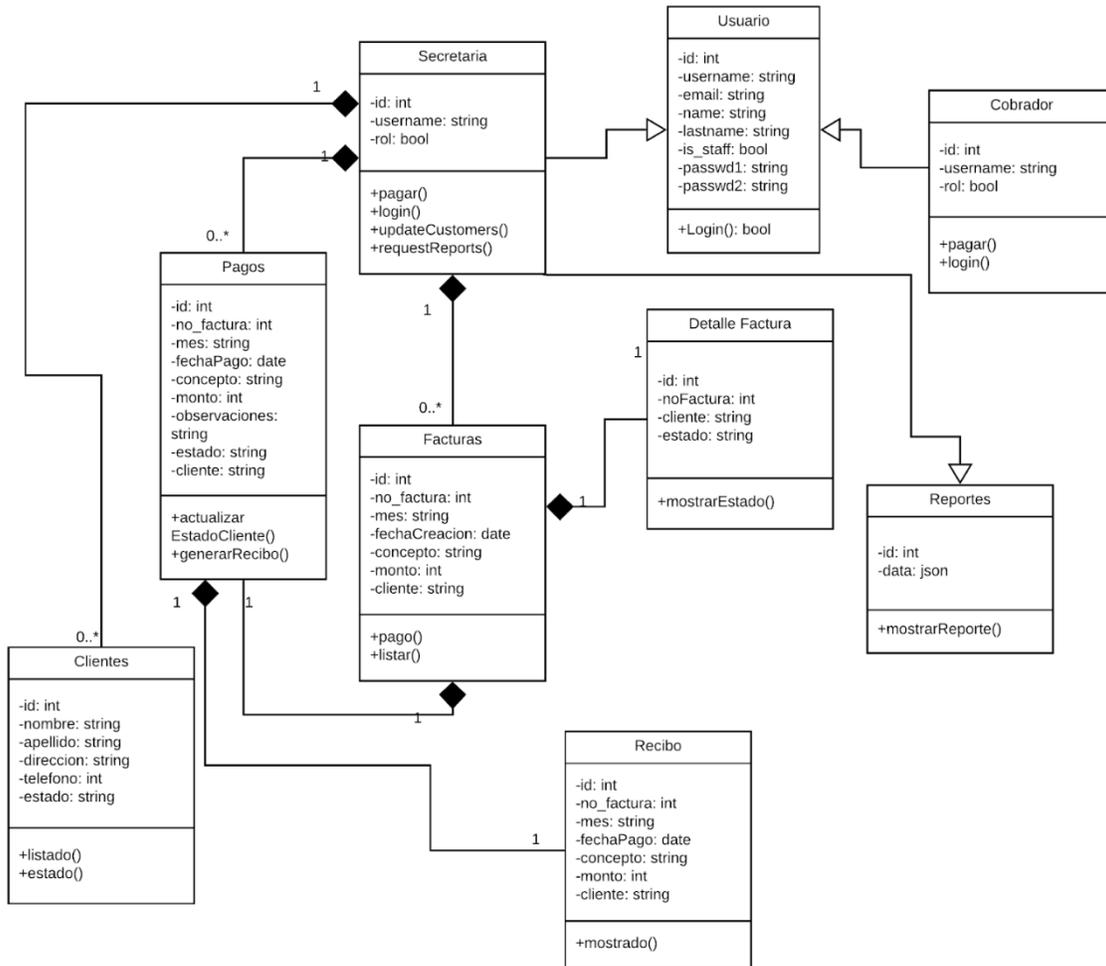




Diagrama de clases



Anexo B. Manual de Usuario para sistema web
para la administración de clientes y gestión de
cobros para la empresa de televisión por cable
“Cable Max” en Kukra Hill



1. INTRODUCCION

1.1 Objetivo del Sistema

Administrar clientes y gestionar cobros para la empresa de televisión por cable “cable Max”.

1.2 Propósito del manual

El presente manual tiene como finalidad ser una guía para el uso correcto del Sistema web para la administración de clientes y gestión de cobros para la empresa de televisión por cable “Cable Max” en Kukra Hill (RAACS). Permitiendo a la empresa contar con una recopilación de toda la información necesaria para la manipulación del mismo.



2. PERFILES

Se utiliza para definir un rol en el sistema, permitiendo delimitar el contenido y permisos que un usuario puede tener.

Los usuarios del sistema están divididos por el tipo de perfil que se les asigne. El sistema web maneja 2 tipos de usuarios:

- **Secretaria:** Es la encargada de administrar el sistema, puede gestionar clientes, pagos, y visualizar reportes.
- **Cobrador:** Puede gestionar pagos y generar recibos.

3. DESCRIPCION DEL ENTORNO

3.1 Iniciar sesión

Lo primero que observamos al ingresar al sistema es el formulario de inicio de sesión (Ilustración 1), el cual permite el ingreso al mismo, para esto se necesita ingresar las credenciales obtenidas al momento de registrar un perfil en el sistema.

Logo: A stylized 'A' inside a circle.

Bienvenid@

Ingresa tu usuario...

Contraseña

Ingresar

[Olvidaste la contraseña?](#)

[Crea una cuenta!](#)

Ilustración 1 Inicio de Sesión



Si las credenciales ingresadas son incorrectas o uno de los campos están en blanco, el sistema muestra un mensaje indicando que los campos son obligatorios (Ilustración 3) o que el usuario o contraseña son incorrectos (Ilustración 2).

('A')
Bienvenid@

Por favor, introduzca un nombre de usuario y clave correctos. Observe que ambos campos pueden ser sensibles a mayúsculas.

Ingresa tu usuario...

Contraseña

Ingresar

Olvidaste la contraseña?
Crea una cuenta!

Por favor, introduzca un nombre de usuario y clave correctos. Observe que ambos campos pueden ser sensibles a mayúsculas.

OK

Ilustración 2 Inicio de sesión incorrecto

('A')
Bienvenid@

asaelz

Contraseña

Campo Obligatorio

Olvidaste la contraseña?
Crea una cuenta!

Ilustración 3 Campos obligatorio



b.1) Editar Cliente

La primera opción corresponde al menú editar (Ilustración 9), para acceder simplemente presionamos el botón **"Verde"**:

ACTUALIZAR CLIENTE

No contrato

14144

Nombre	Apellido
<div style="border: 1px solid #ccc; padding: 5px; width: 90%;">Marcos</div>	<div style="border: 1px solid #ccc; padding: 5px; width: 90%;">Solis</div>
Dirección	Telefono
<div style="border: 1px solid #ccc; padding: 5px; width: 90%;">Cementerio, 4C al sur, media ab</div>	<div style="border: 1px solid #ccc; padding: 5px; width: 90%;">+50555654541</div>

 Guardar

 Cancelar

Ilustración 9 Editar Cliente



Inmediatamente obtendremos un formulario con los datos del cliente seleccionado (Ilustración 10), una vez hallamos realizado los cambios presionamos el botón "**Guardar**", o "**Cancelar**" en caso de no querer realizar un cambio.

ACTUALIZAR CLIENTE

No contrato

14144

Nombre: Marco Antonio

Apellido: Solis

Direccion: Cementerio, 4C al sur, media ab

Telefono: +50555654541

Guardar **Cancelar**

Ilustración 10 Actualizar Cliente

Al presionar el botón "**Guardar**" seremos redirigidos hacia la lista de usuarios donde se reflejará el cambio realizado (Ilustración 11).

14144	Marco Antonio	Solis	Cementerio, 4C al sur, media abajo.	+50555654541	 
-------	---------------	-------	-------------------------------------	--------------	---

Ilustración 11 Cambio realizado a cliente



b.2) Eliminar Cliente

La segunda opción corresponde al botón rojo, este nos permite eliminar un cliente. Para hacer uso de la misma, solo debemos dar clic en el botón "**Rojo**", se nos abrirá el siguiente mensaje:

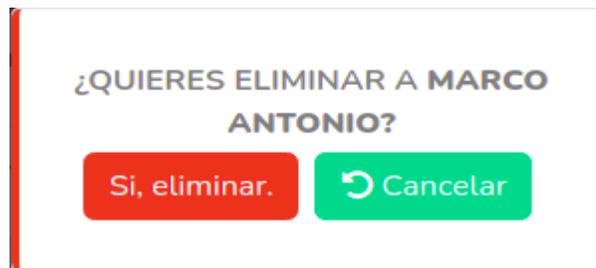


Ilustración 12 Eliminar Cliente

Si damos clic en "**Si, eliminar**", el cliente será borrado del listado de clientes, si presionamos cancelar, regresaremos a la tabla.

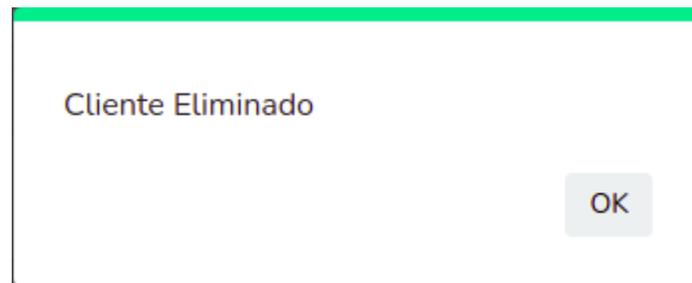


Ilustración 13 Confirmación de eliminación



c) Acerca de la tabla

Dentro de las opciones de la tabla están:

- Definir el número de registros de clientes a mostrar por página, es decir cuantos se pueden mostrar antes de presionar el botón siguiente:

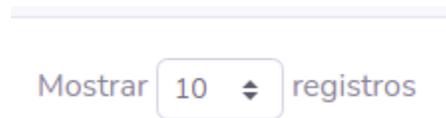


Ilustración 14 Definir número de Registros

- Filtro de búsqueda, nos permite filtrar los clientes por cualquiera de los campos en la tabla, por ejemplo: Apellido.



Ilustración 15 Búsqueda de cliente

- Opción agregar cliente nuevo: al presionar el botón más "+" seremos redirigidos hacia el formulario de creación de clientes.



Ilustración 16 Agregar cliente



3.3 Pagos

El siguiente apartado corresponde a control de pagos, para acceder a este debemos dar clic en la opción pagos, nos abrirá un listado con todas las facturas pagadas (Ilustración 17).

FACTURAS PAGADAS								
Mostrar	5	registros	Buscar:					
No. factura	Cliente	Mes	Fecha creacion	Concepto	Monto	Estado	Acciones	
81	Carmela Mendoza	Febrero	04-03-2021	Pago Mensual	C\$250.0	Pagado		
84	Carmela Mendoza	Enero	07-03-2021	Pago Mensual	C\$250.0	Pagado		

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiete

Ilustración 17 Facturas pagadas

Campos que contiene:

- No. Factura: Correspondiente a la factura del pago realizado.
- Cliente: Dueño de la factura.
- Mes: Mes reflejado en la factura.
- Fecha Creación: Fecha en la que se ha creado la factura.
- Concepto: A qué se debe el pago.
- Monto: Estándar de pago mensual.
- Estado: Depende si el pago ha sido realizado o no.
- Acciones: Reporte e impresión de recibo.

Opciones:

En la última columna de la tabla encontramos dos opciones: estas corresponden a las acciones que podemos realizar:

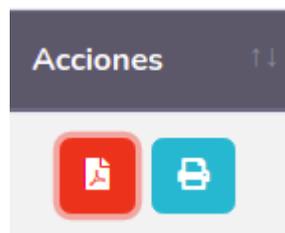


Ilustración 18 Acciones de tabla factura



1) La primera permite mostrar el pago realizado en formato de reporte. Para acceder a esta debemos presionar el botón "**Rojo**" con icono de documento.

CableMax

Reporte de pago

No Pago	Cliente	Fecha	Monto	Mes
81	Carmela Mendoza	04-03-2021	250.0	Febrero

*****Final Reporte*****

CableMax todos los derechos reservados

Ilustración 19 Formato de Reporte



2) La segunda permite imprimir un recibo, para acceder a esta opción debemos presionar el botón "**Celeste**" con icono de impresora.

CableMax
Direccion: Kukrahill
FACTURA No. 81

Factura No. 81 Fecha Fact: 04-03-2021
No. Contrato: 6669
Cliente: Carmela Mendoza
Direccion: guineacity
Mes Pagado: Febrero
Monto Total Pagado: C\$ 250.0

©**CableMax**

Ilustración 20 Impresión de Recibo



3.4 Facturación

En este apartado tenemos dos opciones:



Ilustración 21 Opciones de Facturación

a) Facturas mensuales

Estas facturas son generadas automáticamente el primer día de cada mes por el sistema las cuales contienen: No.factura, cliente, mes, fecha creación, concepto, monto.

Bienvenid@ asaetz

FACTURAS GENERADAS MES: ABRIL

Mostrar registros Buscar:

No. factura	Cliente	Mes	Fecha creacion	Concepto	Monto
96	Carmela Mendoza	Abril	1-04-2021	Pago Mensual	C\$250.0
97	Asael Zuniga	Abril	1-04-2021	Pago Mensual	C\$250.0

Mostrando registros del 1 al 2 de un total de 2 registros Anterior **1** Siguiente

Copyright © CableMax 2021

Ilustración 22 Facturas Generadas



b) Facturas pendientes

En esta sección es donde se realiza el pago de una factura.

Bienvenid@ asaelz 

FACTURAS PENDIENTES

Mostrar registros Buscar:

No. factura	Cliente	Mes	Fecha creación	Concepto	Monto	Estado	Acciones
82	Asael Zuniga	Enero	01-03-2021	Pago Mensual	C\$250.0	Pendiente	Pagar 
83	Asael Zuniga	Febrero	02-03-2021	Pago Mensual	C\$250.0	Pendiente	Pagar 
97	Asael Zuniga	Abril	13-04-2021	Pago Mensual	C\$250.0	Pendiente	Pagar 

Mostrando registros del 1 al 3 de un total de 3 registros Anterior **1** Siguiente

Copyright © CableMax 2021

Ilustración 23 Facturas Pendientes

a) Proceso de pago

Para realizar el pago de una factura damos clic en el botón **“Pagar”**

Estado	Acciones
Pendiente	Pagar 
Pendiente	Pagar 
Pendiente	Pagar 

Ilustración 24 Realizar pago



b) Generar recibo

Una vez que presionamos el botón **"Pagar"** se abrirá la siguiente vista:

GENERAR RECIBO

Cliente: 13 Fecha de Pago: 22/04/2021

Mes: Abril Monto Pagado (C\$): 250.0

Concepto: Pago Mensual Estado: Pendiente

Observaciones:

Generar Cancelar

Ilustración 25 Generar Recibo

Aquí la única opción manipulable es la de estado las otras son llenadas por el sistema, en esta (estado) cabíamos de pendiente a pagado.

Estado:

Pendiente

Pagado

Pendiente

Ilustración 26 Cambio de estado



De esta forma

Estado:

Pagado

Ilustración 27 Cambio Realizado

Existe un campo llamado observaciones donde podemos registrar algún inconveniente expresado por un cliente

Observaciones:

Ilustración 28 Crear Observación

Este campo es opcional si desea registrar un inconveniente sugerimos utilizar la palabra problemas.

Observaciones:

Problema: Se reporta incidentes con el cable en la zona 8.

Generar Cancelar

Ilustración 29 Guardar Observación



Una vez completados los campos anteriores presionamos el botón "**Generar**" de color azul.

CableMax
Direccion: Kukrahill
FACTURA No. 97

Factura No. 97 Fecha Fact: 22-04-2021
No. Contrato: 147
Cliente: Asael Zuniga
Direccion: PJCH
Mes Pagado: Abril
Monto Total Pagado: C\$ 250.0

©CableMax

Print 1 page

Destination Save as PDF

Pages All

Pages per sheet 1

Margins Default

Options Background graphics

Print using system dialog... (Ctrl+Shift+P)

Cancel Save

Ilustración 30 Generación de Recibo y Opción de impresión



Si nos dirigimos al menú de pagos podemos observar que el estado de la factura ha pasado a pagado.

FACTURAS PAGADAS							
Mostrar	5	registros	Buscar:				
No. factura	Cliente	Mes	Fecha creacion	Concepto	Monto	Estado	Acciones
81	Carmela Mendoza	Febrero	04-03-2021	Pago Mensual	C\$250.0	Pagado	 
84	Carmela Mendoza	Enero	07-03-2021	Pago Mensual	C\$250.0	Pagado	 
85	Carmela Mendoza	Diciembre	15-04-2021	Pago Mensual	C\$250.0	Pagado	 
96	Carmela Mendoza	Abril	15-04-2021	Pago Mensual	C\$250.0	Pagado	 
97	Asael Zuniga	Abril	22-04-2021	Pago Mensual	C\$250.0	Pagado	 

Ilustración 31 Vista de Facturas pagadas



4. Reportes

Dentro de reportes encontramos estas dos opciones



Ilustración 32 Opciones de Reportes

Al dar clic en principales se muestra:

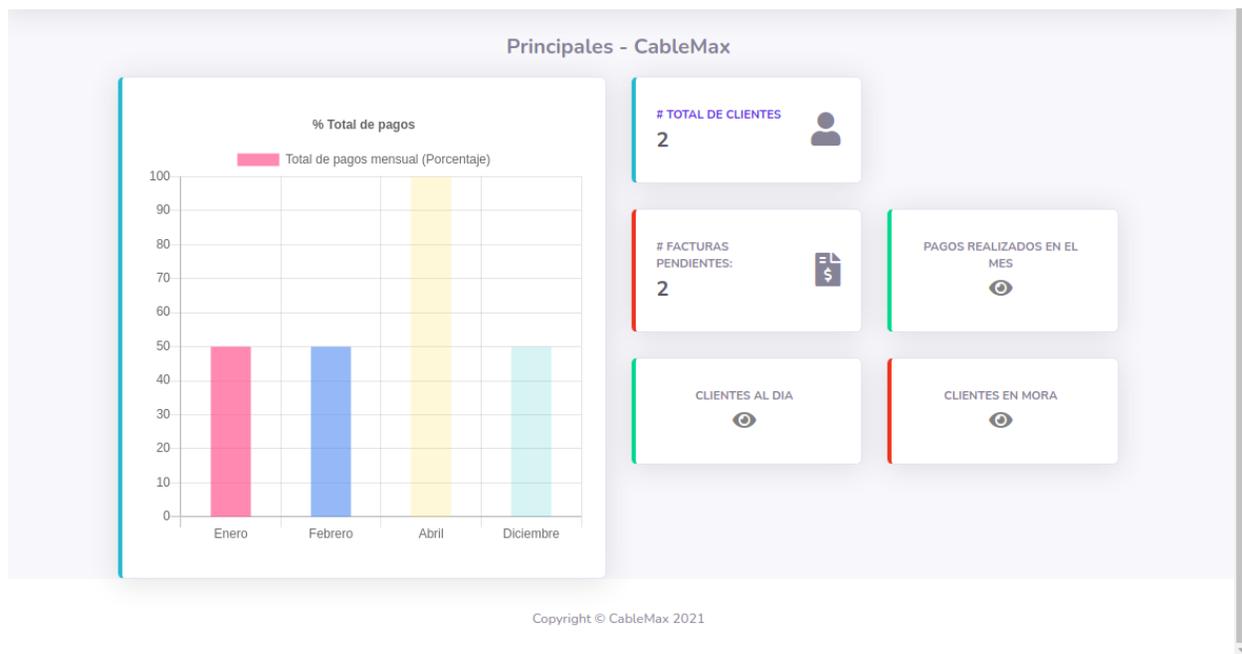


Ilustración 33 Reportes Principales



Dentro de principales tenemos:

a) Total de pagos

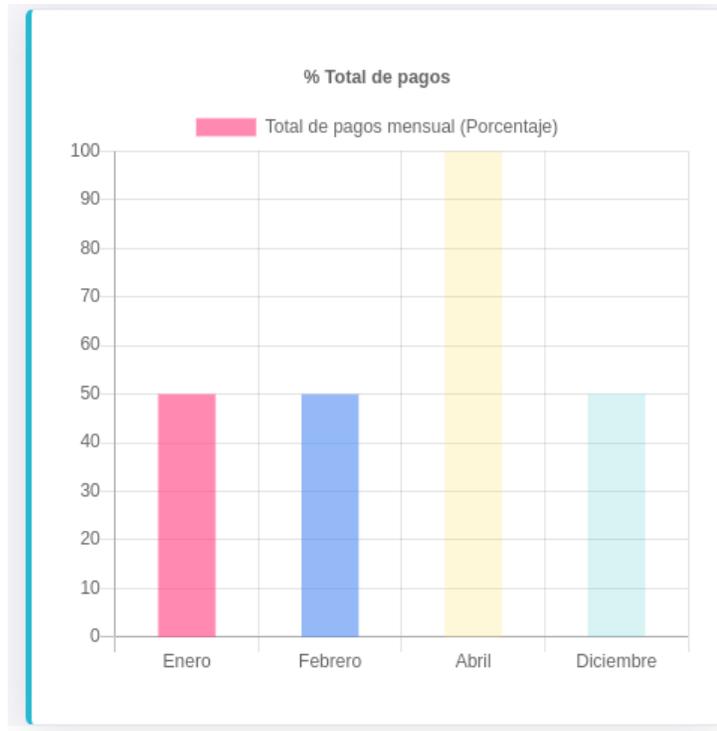


Ilustración 34 Total de Pagos

Este grafico representa el porcentaje de clientes que han realizado su pago ordenado por meses.

b) Total de clientes



Ilustración 35 Total de clientes de la empresa



c) Facturas pendientes



Ilustración 36 Total de Facturas Pendientes

d) Pagos realizados en el mes



Ilustración 37 Pagos por mes

Para acceder a este reporte tenemos darle clic al botón **Ver**



Ilustración 38 Acceder a pagos por mes



Nos mostrara :

CableMax

Reporte total de pagos 22/04/2021

Muestra todos los pagos realizados en el mes de **Abril**

No	Cliente	Fecha	Monto	Observaciones
85	Carmela Mendoza	15 de Abril de 2021 a las 00:00	250.0	NO HAY OBSERVACIONES
96	Carmela Mendoza	15 de Abril de 2021 a las 00:00	250.0	NO HAY OBSERVACIONES
97	Asael Zuniga	22 de Abril de 2021 a las 00:00	250.0	Problema: Se reporta incidentes con el cable en la zona 8.

Ilustración 39 Reporte de pagos por mes

Donde encontraremos un reporte de todos los pagos realizados en el mes en este caso es el mes de abril

e) Clientes al día

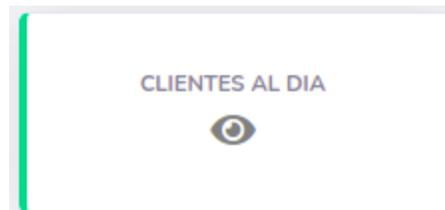


Ilustración 40 Clientes al día



CableMax

Reporte Clientes Al Dia

Listado de clientes sin facturas por pagar

Cliente	Estado
Carmela Mendoza	Al Dia

Final Reporte

CableMax todos los derechos reservados

Ilustración 41 Clientes sin facturas pendientes

f) Clientes morosos



Ilustración 42 Clientes en Mora



CableMax

Reporte Clientes Morosos

Listado de clientes en Mora

Cliente	Facturas Pendientes	Estado
Asael Zuniga	2	Moroso

Final Reporte

CableMax todos los derechos reservados

Ilustración 43 Clientes con Facturas pendientes

Dentro de observaciones tenemos

OBSERVACIONES		
Muestra las observaciones realizadas por los clientes al momento de pagar		
Mostrar <input type="text" value="5"/> registros	Buscar: <input type="text"/>	
Cliente	Mes	Observaciones
Asael Zuniga	Abril	Problema: Se reporta incidentes con el cable en la zona 8.
Carmela Mendoza	Enero	Sin Problemas al pagar
Mostrando registros del 1 al 2 de un total de 2 registros		
Anterior		1
		Siguiente

Ilustración 44 Observaciones

Si dentro de la observación se incluye la palabra problemas se nos mostrara el problema en color rojo así mismo el cliente y el mes en el que ocurrió si no sin problemas al pagar.