



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**DESARROLLO DE UNA APLICACIÓN WEB  
PARA LA OPTIMIZACIÓN DE UN PROCESO  
CRÍTICO**

**INFORME DE ACTIVIDADES PROFESIONALES**

Que para obtener el título de  
**Ingeniera en Computación**

**P R E S E N T A**

Karla Mariana Rosas Jiménez

**ASESOR DE INFORME**

Ing. Javier Alejandro García Romero



**Ciudad Universitaria, Cd. Mx., 2020**

*DESIGNACIÓN DE SINODALES DE EXAMEN PROFESIONAL*

Director de informe: Ing. Javier Alejandro García Romero

Presidente: Ing. Alberto Templos Carbajal

Vocal: M.I. Francisco Barrera del Rayo

Secretario: M.C. María Jaquelina López Barrientos

1er. Suplente: Dr. Ismael Everardo Bárcenas Patiño

2do. Suplente: Dr. Guillermo Gilberto Molero Castillo

*Dedicado con mucho amor y cariño  
a mis padres, Rocío y Jesús  
quienes siempre me han apoyado  
para cumplir mis sueños.  
¡Este triunfo no es solo mío,  
también es suyo!*

## ***Agradecimientos***

Quiero agradecer a todas las personas que siempre creyeron en mí y que me han apoyado de manera incondicional a lo largo de este camino...

### ***A mi madre, Rocío Julia Jiménez Plata***

Por ser una gran mamá y mejor amiga, por haberme cuidado y brindado su amor y cariño a lo largo de mi vida, creyendo en mí y sacrificándose para ayudarme a cumplir mis sueños, por estar conmigo en los momentos felices, pero también en los más difíciles brindándome su calor, comprensión y levantándome de mis caídas para poder continuar ¡te amo con todo mi corazón!

### ***A mi padre, Jesús Rosas Rosas***

Por ser el mejor papá del mundo, por haberme motivado a ser una gran persona y profesionalista, a no darme por vencida y nunca conformarme. Por haber estado conmigo en todo momento, cuidándome y ayudándome día con día hasta lograr este y muchos otros de mis sueños. Por ser junto con mi mamá, mi mayor admiración y mi motivación para seguir luchando. ¡te amo con todo mi corazón!

### ***A mi hermana Jimena***

Mi futura Química, gracias por ser una gran hermana, confidente y amiga, por darme consejos y compartir conmigo momentos de alegría, diversión y en ocasiones momentos difíciles y por enseñarme a nunca rendirme. ¡Gracias por tu amor!

### ***A mi hermano Arturo,***

Mi Ingeniero Industrial, gracias por compartir conmigo muchos momentos y aventuras a lo largo de nuestra formación, por darme sabios consejos y enseñarme siempre cosas nuevas. ¡Gracias por tu cariño!

### ***A mi Thibault,***

Por ser un gran amigo y novio excepcional, por apoyarme y motivarme a buscar siempre nuevos retos, crecer como persona y vencer mis miedos, por creer siempre en mí y compartir momentos inolvidables conmigo. ¡Je t'aime!

### ***A Andrea,***

A mi futura Odontóloga, gracias por ser como una hermana para mí, quien siempre ha estado conmigo en los buenos y malos momentos, por ser un ejemplo de fortaleza y apoyo, ¡Gracias por tu amistad incondicional!

### ***Al Ing. Javier Alejandro García Romero,***

Por darme la oportunidad de pertenecer al equipo de UNAM Mobile y Yamblet, en donde obtuve herramientas para fortalecer mi formación como ingeniera, por ser un ejemplo de espíritu emprendedor e innovador. Por su tiempo y apoyo para la revisión de este trabajo.

*Karla Mariana Rosas Jiménez*

***Al Mtro. Francisco Barrera del Rayo***

Por su apoyo incondicional, sus consejos y su tiempo para revisar este trabajo. Por ser un gran profesor, una persona admirable y un gran amigo. Por haberme brindado sabios consejos que llevaré conmigo a lo largo de mi vida.

***A la M.C. Ma. Jaquelina López Barrientos***

Por ser una gran profesora, ingeniera y gran mujer, por ser un ejemplo a seguir.

¡Gracias por todo!

***Al Ing. Alberto Templos Carbajal***

Por ser un gran profesor y por su tiempo brindado para el seguimiento de este informe.

***Al Dr. Everardo Bárcenas y al Dr. Guillermo Molero***

Por su tiempo en la revisión de mi informe y brindarme valiosos comentarios para enriquecer este trabajo.

***Al Dr. Javier Suárez Rocha***

Por ser un profesor admirable y haberme brindado su apoyo incondicional a los comienzos de mi proceso de titulación. ¡Gracias Doctor!

***Al Mtro. Edgar Uribe***

Por su tiempo y dedicación durante mi proceso de titulación y compartirme enseñanzas de vida que tomaré en cuenta a lo largo de mi vida profesional.

***A la UNAM y a la Facultad de Ingeniería***

Por formarme como profesionista y como persona, por permitirme conocer gente maravillosa, por darme a los mejores amigos, por darme a los mejores profesores y por permitirme tener una formación excepcional en el extranjero.

¡Por haberme dado todo!

## Contenido

Índice de Ilustraciones .....	7
Índice de Tablas .....	8
Introducción .....	9
Objetivo .....	9
1. Antecedentes .....	10
1.1 Estado del arte .....	13
1.2 Propuesta de Optimización del proceso .....	14
2. Marco teórico.....	16
2.1. Ingeniería de software .....	16
2.2. Producto del software .....	17
2.3. Ciclo de Vida del Software .....	19
2.4. Evolución del software.....	20
2.5. Metodologías de desarrollo.....	21
2.6. Computación en la nube (Cloud Computing) .....	23
2.7. Bases de datos, MySQL y Java Script.....	28
2.8. App Maker Y Cloud SQL .....	33
2.9. Factores críticos de éxito en los proyectos de software.....	36
3. Metodología de desarrollo de la aplicación .....	40
3.1. Análisis .....	42
3.2. Planeación y definición .....	43
3.2.1. Equipo-Organización de trabajo.....	43
3.3. Diseño .....	45
3.3.1. Base de datos E/R.....	45
3.3.2. Prototipos de diseño .....	47
3.4. Desarrollo.....	53
3.4.1. Ambiente de desarrollo y Creación del proyecto.....	53
3.4.2. Seguridad y permisos .....	59
3.5. Pruebas.....	60
3.6. Liberación a producción .....	60
3.7. Documentación y Mantenimiento.....	62
4. Resultados.....	62

5. Participación profesional .....	65
6. Conclusiones.....	66
7. Bibliografía.....	68

## Índice de Ilustraciones

ILUSTRACIÓN 1. PANORAMA GENERAL DE LA EMPRESA K .....	10
ILUSTRACIÓN 2. PROCESO DURANTE UN IC .....	11
ILUSTRACIÓN 3. DESCRIPCIÓN DE PROCESO INCIDENTES CRÍTICOS .....	12
ILUSTRACIÓN 4. MEJORAS DEL PROCESO DE INCIDENTES CRÍTICOS.....	15
ILUSTRACIÓN 5. CICLO DE VIDA DEL SOFTWARE, 3G E-LEARNING,2018.....	17
ILUSTRACIÓN 6. SDLC 3G, E-LEARNING,2018.....	19
ILUSTRACIÓN 7. EVOLUCIÓN DEL SOFTWARE, 3G E-LEARNING,2018.....	20
ILUSTRACIÓN 8. MODELO TRADICIONAL O CASCADA.....	22
ILUSTRACIÓN 9. MODELO ÁGIL.....	22
ILUSTRACIÓN 10. MODELO HÍBRIDO .....	23
ILUSTRACIÓN 11. SERVICIOS DE LA NUBE.....	25
ILUSTRACIÓN 12. CUADRO DE NIVELES DE ABSTRACCIÓN.....	28
ILUSTRACIÓN 13. LENGUAJES PARA BASES DE DATOS.....	29
ILUSTRACIÓN 14. SISTEMA DE ADMINISTRACIÓN DE BASES DE DATOS.....	29
ILUSTRACIÓN 15. ESTRUCTURA DE CÓDIGO HTML .....	32
ILUSTRACIÓN 16 ESTRUCTURA DE UNA PÁGINA WEB.....	32
ILUSTRACIÓN 17. INTERFAZ DEL AMBIENTE DE DESARROLLO APP MAKER.....	33
ILUSTRACIÓN 18. EJEMPLO DE ALMACENAMIENTO EN APP MAKER .....	34
ILUSTRACIÓN 19. TIPOS DE DATOS EN APP MAKER.....	35
ILUSTRACIÓN 20 . MODELO DE FCE PROPUESTO POR THE AUTHOR(S).....	39
ILUSTRACIÓN 21. DIAGRAMA DE TIEMPO, DESARROLLO DEL PROYECTO.....	44
ILUSTRACIÓN 22. DIAGRAMA E/R DE LA APLICACIÓN IC.....	46
ILUSTRACIÓN 23. PROTOTIPO 1, PÁGINA PRINCIPAL “DASHBOARD”.....	47
ILUSTRACIÓN 24. PROTOTIPO 2, PÁGINA “DETALLES DEL INCIDENTE”.....	48
ILUSTRACIÓN 25. PROTOTIPO 3, "DETALLES DEL CLIENTE".....	49
ILUSTRACIÓN 26 PROTOTIPO 4, “CARGA DEL INCIDENTE” .....	49
ILUSTRACIÓN 27. PROTOTIPO 5, “ACTUALIZACIÓN DEL INCIDENTE” .....	50
ILUSTRACIÓN 28. PROTOTIPO 6, "CIERRE DE INCIDENTE" .....	51
ILUSTRACIÓN 29. PROTOTIPO 7, “FORMATO DE COMUNICADOS”.....	52
ILUSTRACIÓN 30. PANTALLA PRINCIPAL .....	53
ILUSTRACIÓN 31. PANTALLA DE LEVANTAMIENTO DE INCIDENTE .....	53
ILUSTRACIÓN 32. CONSTRUCCIÓN DE PANTALLA SECUNDARIA .....	54
ILUSTRACIÓN 33. CONSTRUCCIÓN DE LA PANTALLA DE ACTUALIZACIÓN DE INCIDENTES .....	54
ILUSTRACIÓN 34. CONSTRUCCIÓN DE LA PANTALLA DE FIN DE INCIDENTE.....	55
ILUSTRACIÓN 35. MODELO DE DATOS .....	55
ILUSTRACIÓN 36. MENÚ DE SECCIÓN DE EQUIPOS.....	56
ILUSTRACIÓN 37. EJEMPLO DE FRAGMENTO DE CÓDIGO CSS .....	57
ILUSTRACIÓN 38. EJEMPLO DE FRAGMENTO DE CÓDIGO SCRIPT .....	58

ILUSTRACIÓN 39. EJEMPLO DE FRAGMENTO DE CÓDIGO HTML EN APP MAKER ..... 58  
ILUSTRACIÓN 40. PROYECTO FINAL EN USO, PANTALLA PRINCIPAL..... 60  
ILUSTRACIÓN 41. PROYECTO FINAL EN USO, DETALLES DE IC ..... 61

## Índice de Tablas

TABLA 1. METODOLOGÍAS DE DESARROLLO (MOHANARAJAH,2014). ..... 21  
TABLA 2. FCE PARA LA ADMINISTRACIÓN EXITOSA DE LOS PROYECTOS DE SOFTWARE ..... 37  
TABLA 3. FACTORES CRÍTICOS DE ÉXITO..... 38  
TABLA 4. ETAPAS DEL DESARROLLO DEL PROYECTO..... 40  
TABLA 5. PROCESO TRADICIONAL Y PROPUESTA ..... 42  
TABLA 6. REQUERIMIENTOS PARA EL PROCESO IC..... 43  
TABLA 7. TABLA COMPARATIVA SEGÚN LA EXPERIENCIA DE LOS USUARIOS ..... 63  
TABLA 8. CRITERIOS DE ACEPTACIÓN DE LA APLICACIÓN DE IC ..... 64

## Introducción

Al concluir mis estudios académicos en la Facultad de Ingeniería comencé a trabajar en una empresa prestadora de servicios de tecnología a la que me referiré como “K”. El rol que desarrollo en esta empresa es el de Ingeniero en Soporte TI. Soy parte de un equipo al que llamaré “Digital IT”, que es un equipo dedicado a la creación de aplicaciones con fuerte impacto en las actividades internas de la empresa, con el objetivo de optimizar diferentes procesos internos tanto a nivel directivo, gestión y operativo.

Mis actividades en el equipo consisten en llevar a cabo el desarrollo de diversos proyectos que me han sido asignados y trabajar en conjunto con los miembros del equipo. Durante el desarrollo de mis actividades he podido aplicar los conocimientos adquiridos a lo largo de mi trayectoria académica y mis años de estudio en Facultad de Ingeniería, siendo mi módulo de especialidad la Ingeniería de Software. Además, pude desarrollar habilidades tales como la comunicación efectiva, trabajo en equipo, la negociación y la obtención de nuevos conocimientos técnicos relacionados con las tecnologías actuales.

En este trabajo presentaré el desarrollo de uno de los proyectos en los que tuve participación y que consistió en proponer una solución para mejorar uno de los procesos de la empresa “K”. A lo largo del desarrollo del proyecto tuve la oportunidad de aportar conceptos de la Ingeniería de Software, para trabajar en equipo y lograr resultados de optimización de tiempo en dicho proceso.

## Objetivo

El propósito de este trabajo es presentar el desarrollo de un proyecto cuyo objetivo fue optimizar el proceso de gestión y control de incidentes críticos de la empresa “K”. Dicho proyecto consistió en la creación de una aplicación web responsiva que el equipo directivo y de operación interna de la empresa pudieran utilizar.

Para fines de este proyecto no se mencionarán los nombres ni datos originales de la empresa por la privacidad y confidencialidad de esta. A continuación, se presenta la creación del proyecto al cual llamaré “Control de incidentes críticos”, se planteará el origen del proyecto, así como su desarrollo y el impacto que tuvo en “K”.

## 1. Antecedentes

Hoy en día las organizaciones se enfrentan a un conjunto complejo de cambios constantes, al incremento de la competencia y a las exigencias del cliente para satisfacer sus demandas y necesidades.

La gestión basada en procesos se enfoca en actividades que realiza una o más personas en la organización con el objetivo de lograr dar seguimiento a los procesos de manera más eficiente y que beneficie a la organización volviéndola más productiva y rentable.

La empresa “K” es una empresa dedicada a brindar servicios de tecnología como servicios administrados y en la nube, ciber seguridad y centros de datos. Su objetivo es brindar ágiles e innovadoras soluciones de infraestructura y tecnologías de información de misión crítica. Para ello cuentan con numerosos equipos de trabajo que en conjunto buscan cumplir con las demandas de los clientes y sus necesidades. A su vez, estos clientes representan organizaciones cuya operación tiene alto impacto en distintos sectores por lo que cualquier afectación o detención en la misma, se debe resolver lo más pronto posible.

“K” conoce la importancia de las diferentes operaciones de sus clientes por lo que procura resolver de manera casi inmediata cualquier inconveniente relacionado con sus servicios. De manera interna los diferentes equipos se organizan para resolver los problemas que se presenten teniendo procesos definidos y una comunicación efectiva (véase Ilustración 1).

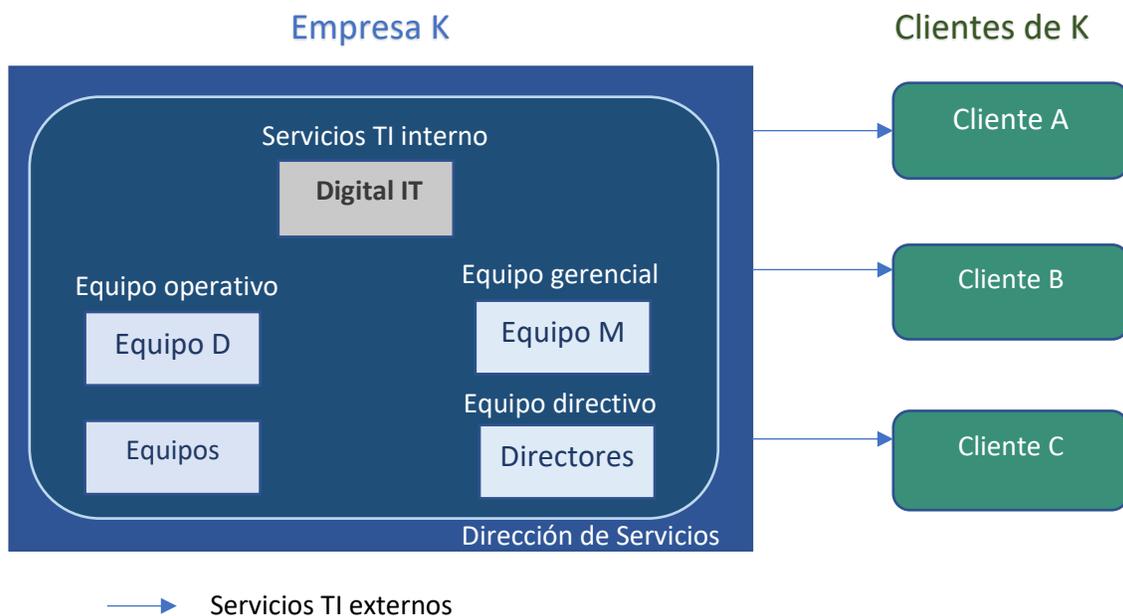


Ilustración 1. Panorama general de la empresa K

*Fuente: Elaboración propia*

Un Incidente Crítico (IC), es aquel acontecimiento cuando uno o varios servicios que provee “K” se ven afectados y esto produce un impacto negativo para el cliente.

Cuando ocurre un IC, el cliente puede sufrir pérdidas económicas o daño a su industria, es por eso por lo que “K” cuenta con procesos para comunicar a la operación interna lo que está ocurriendo y que se pueda solucionar el incidente lo antes posible. El equipo de operación interna sabe la importancia de sus servicios y las consecuencias que pueden traer para “K” los IC: Afectación y posible pérdida de un cliente o una penalización económica.

La solución de Incidentes Críticos involucra a diferentes equipos (Véase Ilustración 2):

**Clientes:** Son los clientes a quienes “K” le vende sus servicios y soluciones tecnológicas.

**Equipo M:** Son los encargados de la comunicación directa con el equipo C.

**Equipo D:** Este compuesto por grupos que llevan a cabo la operación técnica y el soporte de los servicios ofrecidos por “K”.

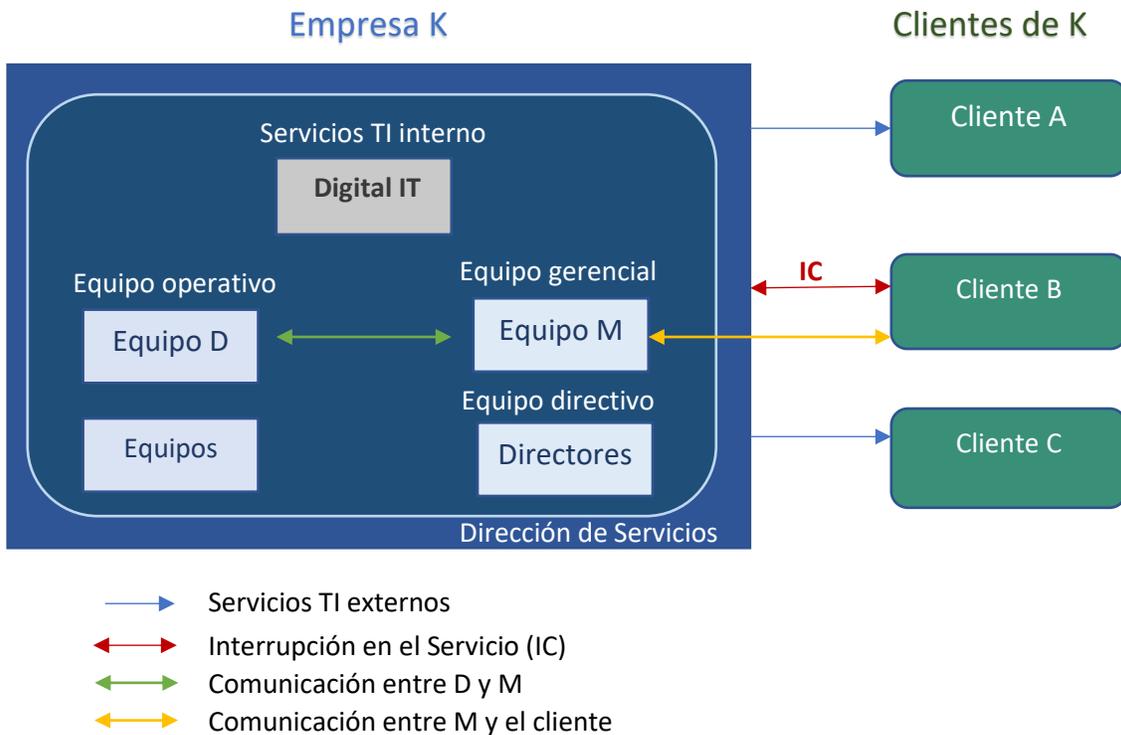
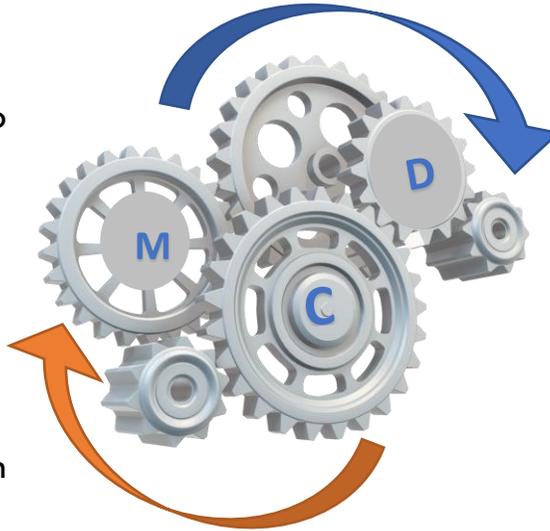


Ilustración 2. Proceso durante un IC

Fuente: Elaboración propia

El proceso que se lleva a cabo es el siguiente (véase Ilustración 3):

1. El cliente notifica un Incidente al equipo M
2. M notifica a su equipo y a D, enviando comunicados de texto plano vía correo electrónico.
3. D se comunica con los otros equipos de soporte técnico hasta resolver el problema, vía WhatsApp
4. M y D se comunican los avances de la solución problema vía correo electrónico.



5. El equipo D encuentra y resuelve la causa de la falla, notifica al equipo M
6. El equipo D recopila la información comunicada y la en Hojas de Cálculo para conservarlo como historial.
7. M notifica a C que el problema ha sido solucionado.
8. C verifica que todo funciona bien y da su Visto Bueno.

Ilustración 3. Descripción de proceso incidentes críticos

Fuente: Elaboración propia

## 1.1 Estado del arte

La gestión de servicios de tecnologías de la información (en inglés IT Service Management, ITSM) es una disciplina basada en procesos, cuyo enfoque está dirigido proporcionar servicios de TI alineados a las necesidades de una empresa.

El objetivo de estos servicios TI, es proporcionar a las empresas herramientas que les permitan gestionar flujos de trabajo y de la operación interna y externa. Algunos ejemplos de ITSM son BMC Helix, Aranda y ServiceNow. Estas herramientas cuentan con una gran capacidad de seguimiento a diferentes procesos, almacenamiento en la nube y están alineadas a las prácticas de la Biblioteca de Infraestructura de Tecnología de Información ITIL, normas como la ISO 20000 (Calidad de servicios TI) e ISO 27000 (Seguridad de la información).

Estas herramientas permiten la gestión y control de distintos procesos, entre ellos el de “Incidentes críticos”, sin embargo, la interacción con este software es a través de un gran nivel técnico.

En la empresa K, el proceso de un “Incidente crítico” conlleva un trabajo colaborativo entre el equipo D (equipo de operación) y el equipo M (equipo gerencial). Cuando se presenta un IC, ambos equipos deben estar en comunicación y compartir información necesaria para lograr mitigar la afectación que está surgiendo con los clientes, lo antes posible. Si bien los ITSM facilitan la gestión de procesos a nivel técnico, para efectos de este proceso, no fue suficiente, ya que para el equipo gerencial entorpecía la comunicación con el equipo y era difícil estar en la misma sintonía con el equipo D.

A raíz de la situación presentada, el equipo de operación y gerencial utilizaban herramientas como:

- WhatsApp (comunicación)
- Correo electrónico (Comunicación)
- Aplicación de calendario (Registrar un historial por IC, con el enlace a una hoja de cálculo con los detalles)
- Hojas de cálculo (Detalles de cada IC)

Estas herramientas permitían comunicarse entre los involucrados en el proceso, sin embargo, cuando surgían IC de manera simultánea, podría generar mucha información desordenada y confusión. Debido a esta situación se vio la necesidad de mejorar la comunicación para este proceso y con ello, mejorar los servicios que ofrece la empresa K a sus clientes.

## 1.2 Propuesta de Optimización del proceso

El proceso definido anteriormente brindaba la oportunidad de dar seguimiento a los IC, sin embargo, contaba con algunas desventajas como:

- ❶ Crear el correo electrónico en texto plano de manera manual demandaba tiempo y esfuerzo por parte de los equipos, lo cual también provocaba que el IC tardara más tiempo en ser comunicado y retrasara su solución.
- ❷ Las cadenas de correos electrónicos generados de forma masiva eran complicadas de seguir cuando se presentaban más de dos IC simultáneos para diferentes clientes.
- ❸ Consultar el estatus de un IC era complicado, ya que la única manera de hacerlo era entrada en bandeja de entrada de los correos electrónicos según correspondiera y buscar el incidente.
- ❹ La única manera de consultar el historial de IC de un cliente es buscando los correos electrónicos que servían como evidencia o de igual manera buscar el enlace de la hoja de cálculo correspondiente.

La organización “K” continúa creciendo y con ello la participación de los equipos internos. En el equipo de Digital IT (del cual soy integrante) propusimos y construimos una herramienta cuyo objetivo fue optimizar y automatizar este proceso (véase Ilustración 4).

La solución consistió en desarrollar una Aplicación Web con las siguientes características y funcionalidades:

- ❶ Intuitiva y fácil de usar, para mejorar la experiencia del usuario.
- ❷ Generar y enviar comunicados automáticos.
- ❸ Facilitar la comunicación efectiva entre las personas que colaboran en el proceso.
- ❹ Optimizar el tiempo dedicado a este proceso.
- ❺ Crear un historial por cliente de cada IC.
- ❻ Centralizar de la información.

1. El cliente notifica un Incidente al equipo M

2. M notifica a su equipo y a D el incidente, registrando la información en la aplicación y enviando automáticamente comunicados con formato vía correo electrónico.

3. M y D registran los estatus del incidente en la aplicación para que los miembros de los equipos puedan acceder a la información en un solo lugar y recibir notificaciones en tiempo real vía correo electrónico.



4. El equipo D encuentra y resuelve la causa del problema y lo notifica en la aplicación.

5. Todos los involucrados en el proceso pueden acceder a la aplicación y consultar el historial de cada cliente por incidente y conocer su estatus actual.

6. C verifica que todo funciona bien y da su Visto Bueno.

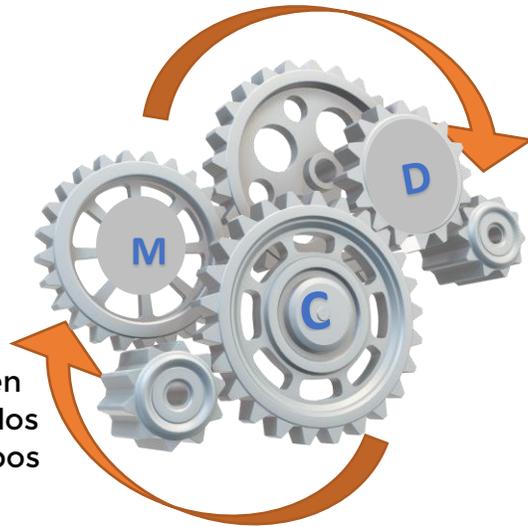


Ilustración 4. Mejoras del proceso de incidentes críticos

Fuente: *Elaboración propia*

## 2. Marco teórico

Para comprender mejor acerca del desarrollo de herramientas de software es importante conocer los siguientes conceptos:

*Software*: Son las instrucciones que le indican a la computadora qué hacer (Capron, 1992). Otra definición es, “instrucción” para indicarle al hardware la tarea que debe realizar (Stanley,2010).

*Hardware*: Es la parte física de un sistema informático. (Stanley,2010). Por ejemplo, el monitor, el teclado, el ratón, la bocina, los audífonos, las memorias, etc.

*Ingeniería de Sistemas*: Es el proceso de análisis y diseño de un sistema completo, incluido el hardware y software (3G E-LEARNING,2018).

*Ingeniería de Software*: Es la disciplina para crear aplicaciones de software. Un enfoque sistémico para el diseño, desarrollo, prueba y mantenimiento del software (3G E-LEARNING,2018).

### 2.1. Ingeniería de software

El software es más que un simple código de programa. Un programa es un código ejecutable, que sirve para algún propósito computacional. El software es considerado como una colección de bibliotecas asociadas, códigos de programación ejecutables y manuales de instrucciones. Cuando el software se hace para un requisito específico se le conoce como *producto de software*. La ingeniería, por otro lado, tiene que ver con el desarrollo de productos, utilizando principios y métodos bien definidos y científicos.

La ingeniería de software es una rama de la ingeniería que asociada con el producto de software cuyo desarrollo utiliza principios, métodos y procedimientos científicos, tiene como resultado un producto de software confiable y eficiente.

El Instituto de Ingeniería Eléctrica y Electrónica (Institute of Electrical and Electronics Engineers, por sus siglas en inglés IEEE<sup>1</sup>) define la ingeniería de software como la aplicación de una manera sistemática, disciplinada, y cuantificable de la operación de desarrollo y mantenimiento del software.

---

<sup>1</sup> La IEEE es la organización profesional mundial dedicada al beneficio de la humanidad a través de su tecnología. Cuenta con más de 423,000 miembros en más de 160 países. Sus publicaciones, conferencias, estándares tecnológicos y actividades profesionales educativas son altamente citadas.

## 2.2. Producto del software

El software tiene una doble función, es un producto y al mismo tiempo es un medio para entregar un producto (véase Ilustración 5).

- Como producto, ofrece potencial informático incorporado en el hardware de una computadora, un teléfono celular o en un computador mainframe, que es un transformador de información que produce, modifica y visualiza información.
- Como el medio utilizado para entregar el software del producto actúa como base de una computadora (sistema operativo), la comunicación de información (redes) y la creación y control de otros programas (herramientas de software y entorno).

El software transforma los datos personales y es más útil en un contexto local, ya que permite la administración de la información comercial para mejorar la competitividad, además proporciona una entrada a las redes mundiales como el Internet, que es un medio para adquirir información en todas sus formas (3G E-LEARNING,2018).



Ilustración 5. Ciclo de vida del software, 3G E-LEARNING,2018.

*Fuente: Elaboración propia*

A continuación, se listan un conjunto de aplicaciones potenciales del software:

- ❶ Sistema de software es una colección de programas escritos al servicio de otros programas y que procesa estructuras complejas, por ejemplo: editores de compiladores y gestión de archivos, controladores de componentes de un sistema operativo. En cualquier caso, el área el sistema de software se caracteriza por la tener una gran interacción con el hardware.
- ❷ Sistemas de software en tiempo real, se encargan de realizar monitoreo, análisis y control a medida que los eventos ocurren. Un sistema en tiempo real debe responder dentro de las estrictas restricciones de tiempo.
- ❸ La Información empresarial es uno de los campos más grandes de aplicación del software, por ejemplo: en el inventario de cuentas por cobrar, pago de nómina, gestión, etc.
- ❹ El software en la ingeniería y la ciencia se caracteriza por la aplicación de algoritmos que permiten realizar estudios y análisis en diferentes áreas que van desde la astronomía hasta la vulcanología, los automóviles, la biología molecular, la dinámica orbital de un transbordador, etc.
- ❺ Los productos inteligentes tienen un lugar importante en el mercado, el software incorporado reside en una memoria de solo lectura y se usa para controlar productos y sistemas para los mercados de consumo e industrial. Algunos ejemplos de aplicación son funciones digitales de combustible, pantallas de tableros e instrumentos, sistemas de frenos, etc.
- ❻ El software para las computadoras personales se enfoca en aplicaciones acerca del procesamiento de textos, hojas de cálculo, graficas, multimedia y entretenimiento, gestión de datos personales, acceso a aplicaciones financieras y de negocios.
- ❼ La Inteligencia artificial (AI) utiliza algoritmos no numéricos para resolver problemas complejos que no son susceptibles de computación o análisis directo, por ejemplo, los sistemas expertos que son sistemas basados en el conocimiento, el reconocimiento de patrones, las redes neuronales artificiales y el aprendizaje de alguna experiencia pasada.

El desarrollo de software es un proceso que consta de fases sucesivas y ordenadas, abarca desde la definición de los requerimientos y objetivos, la codificación y la verificación del desarrollo.

El uso de las metodologías de desarrollo de software es útil para planear, gestionar y controlar el proceso de creación de los sistemas de información. La ingeniería de software permite la aplicación sistemática y disciplinada para obtener un enfoque cuantificable del desarrollo.

### 2.3. Ciclo de Vida del Software

Es un proceso que permite dar seguimiento a un proyecto de software dentro de una organización. Consiste en un plan detallado que indica cada una de las etapas como desarrollo, el mantenimiento y el manejo del software (véase Ilustración 6).

<b>Etapas</b>	<b>Descripción</b>
Planeación y Análisis de requerimientos.	Es la etapa más importante y fundamental en el SDLC, hay participación por parte del cliente, el equipo de desarrollo y los expertos en dominios de la industria. Esta información se utiliza para planificar posteriormente el enfoque básico del proyecto y realizar una estimación de viabilidad, tiempo y costo.
Definición de los requerimientos	Una vez completo el análisis de los requisitos, el siguiente paso consiste en documentar los requisitos del producto y obtener la aprobación del cliente. Esto se realiza en un documento Especificaciones de Requerimientos de Software (SRS Software Requirement Specification), que consta de todos los requisitos del producto para el diseño y desarrollo del proyecto.
Diseño de la arquitectura del producto	SRS es la fuente de información para que los arquitectos del producto obtengan una estructura adecuada al proyecto.
Construcción y desarrollo del producto	En esta etapa de SDLC, el desarrollo real comienza y el producto se crea. Si el diseño se realiza de forma detallada y organizada, la generación de código puede ser realizada sin muchas complicaciones
Pruebas del producto	La realización de pruebas al producto es un proceso que se lleva a cabo a lo largo de todas las etapas involucradas. Sin embargo, en esta etapa se refiere a la prueba única del producto, cuyo objetivo es identificar los defectos y problemas para informar acerca de su existencia, corregirlos y volver a realizar pruebas hasta cumplir con los estándares de calidad definidos en el SRS.
Lanzamiento y mantenimiento	Una vez que el producto ha sido probado y está listo para ser implementado, se lanza a producción. El mantenimiento se realiza con base a los clientes existentes.

Ilustración 6. SDLC 3G, E-Learning,2018.

*Fuente: E-Learning*

## 2.4. Evolución del software

El proceso de desarrollo de un producto de software que utiliza métodos de principios de ingeniería de software se denomina *Evolución del software* (véase Ilustración 7). La evolución comienza a partir del requisito de recopilación del prototipo y lo revela a los usuarios con el fin de obtener una retroalimentación respecto al software, los usuarios sugieren cambios y actualizaciones hasta que se logre el software deseado (3G E-LEARNING,2018).

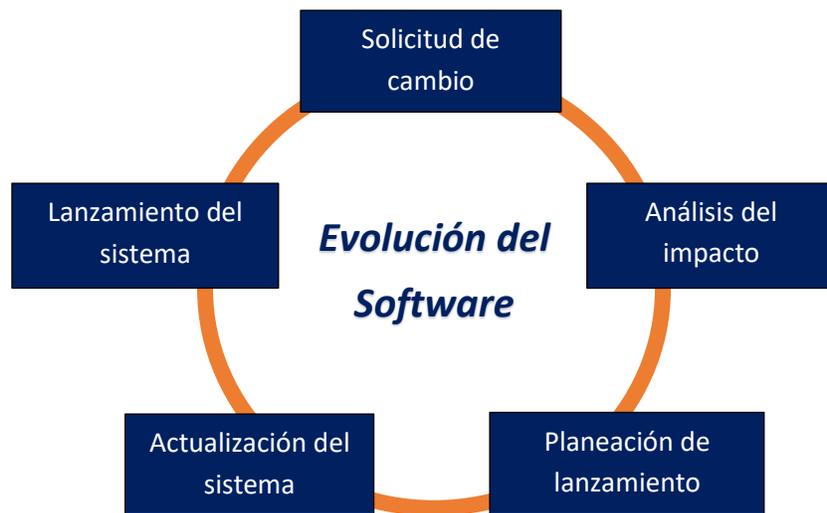


Ilustración 7. Evolución del software, 3G E-LEARNING,2018.

*Fuente: E-Learning*

## 2.5. Metodologías de desarrollo

Las Metodologías de Desarrollo de Software son ampliamente agrupadas en tres categorías distintas: tradicional, ágil e híbrida (véase Tabla 1).

Tabla 1. *Metodologías de desarrollo (Mohanarajah,2014).*

METODOLOGÍAS		
Tradicional	Ágil	Híbrida
<p>Las tradicionales se refieren esencialmente al tipo cascada y modelos en espiral. Estos tipos de metodologías son pesados y se resisten a los cambios.</p> <p>Objetivo: Asegurar entregas exitosas.</p> <p>Característica principal: Consiste en un conjunto de etapas definidas que se desarrollan de manera secuencial, una tras otra. Su principal inconveniente es la dificultad de implementar cambios.</p> <p>Elementos empleados: liderazgo, indicadores de desempeño, políticas estratégicas, comunicación efectiva, gestión y aprendizaje (véase Ilustración 8).</p>	<p>Las ágiles se refieren a procesos adaptativos e iterativos. Estos tipos de metodologías son ligeras y no se resisten a los cambios (véase Ilustración 9).</p> <p>Los 4 valores de estas metodologías son:</p> <ul style="list-style-type: none"> <li>• Individuos e interacciones sobre procesos y herramientas.</li> <li>• Software de trabajo sobre documentación completa.</li> <li>• Colaboración del cliente en la negociación de contratos.</li> <li>• Responde al cambio de planes.</li> </ul>	<p>Esta categoría es una mezcla de lo tradicional y lo ágil.</p> <p>Las metodologías tradicionales trabajan bien, sin embargo, existen proyectos complejos que necesitan de estrategias tanto lineales como adaptativas. Las metodologías híbridas están ganando impulso, ya que sugieren una combinación de tradicional y ágil.</p> <p>Las metodologías híbridas han presentado resultados donde la efectividad se ha incrementado (véase Ilustración 10).</p>

Fuente: *Elaboración propia*

## MODELO TRADICIONAL



Ilustración 8. Modelo tradicional o cascada

*Fuente: Elaboración propia*

## MODELO AGILE

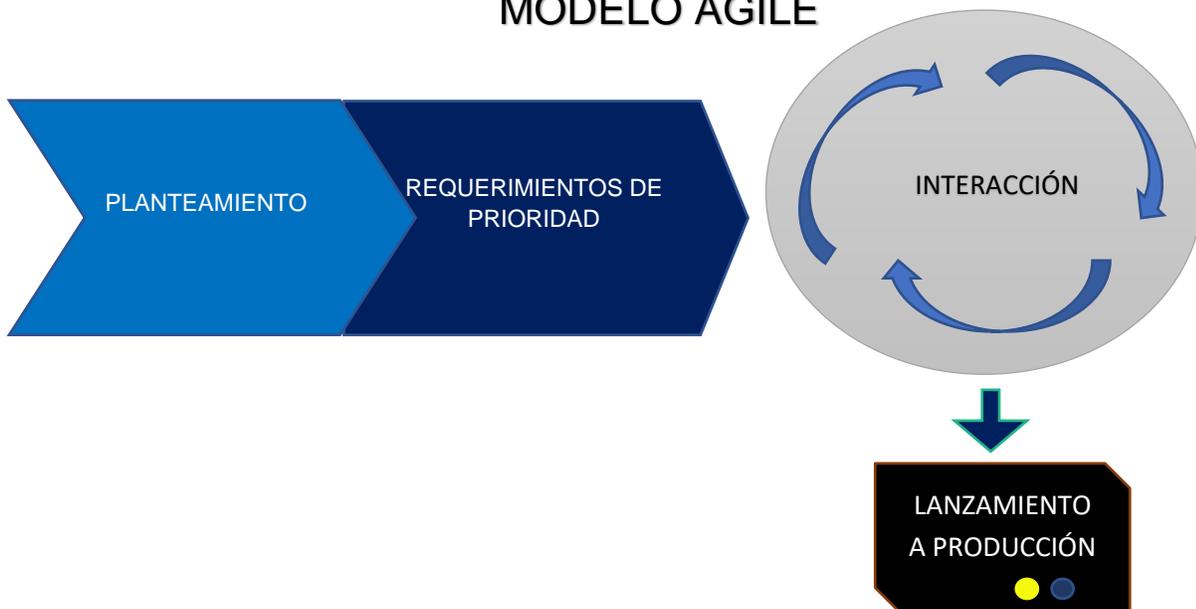


Ilustración 9. Modelo ágil

*Fuente: Elaboración propia*

## MODELO HIBRIDO

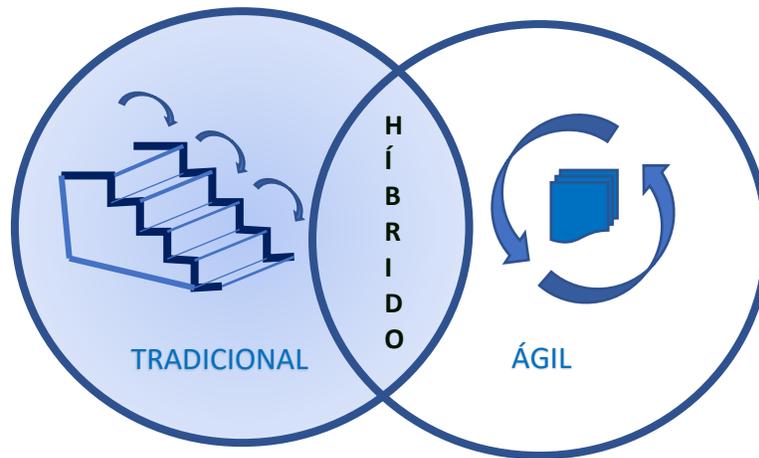


Ilustración 10. Modelo híbrido

*Fuente: Elaboración propia*

### 2.6. Computación en la nube (Cloud Computing)

La nube puede ser infraestructura o software, es decir, puede ser una aplicación a la que se accede a través del escritorio y se ejecuta inmediatamente tras su descarga, o bien un servidor al cual se le solicita cuando se le necesita (Aguilar, L. J., 2012).

El almacenamiento en la nube consiste en mantener los datos en Internet a través de un proveedor que administra y opera este almacenamiento como un servicio el cual se ajusta a las necesidades del cliente en cuanto al costo y capacidad. Una de las características principales de este servicio es que el cliente no requiere invertir en infraestructura, puede escalar sus proyectos y acceder a su información en cualquier momento. Muchos proveedores ofrecen servicios complementarios de protección y gestión de información (Amazon Web Services, 2019).

El National Institute of Standards and Technology (NIST)<sup>2</sup> define la computación en la nube como:

---

<sup>2</sup> El NIST es una Agencia del Departamento de Comercio de los Estados Unidos. Dentro del NIST, el Computer Security Resource Center (CSRC) se encarga de los estándares de las Tecnologías de la Información y, en concreto, de Cloud Computing

“Un modelo que permite el acceso bajo demanda a través de la Red a un conjunto compartido de recursos de computación configurables (redes, servidores, almacenamiento, aplicaciones y servicios) que se pueden aprovisionar rápidamente con el mínimo esfuerzo de gestión o interacción del proveedor del servicio” (véase Ilustración 11).

El mundo de la nube cuenta con un gran número de actores como:

- ❶ *Vendedores y proveedores*: Proporcionan las aplicaciones y facilitan las tecnologías, infraestructura, plataformas y la información correspondiente.
- ❷ *Socios de los proveedores*: Crean servicios en la nube, ofreciendo los servicios a los clientes.
- ❸ *Líderes de negocios*: Evalúan los servicios de la nube para implantarlos en sus organizaciones y empresas.
- ❹ *Usuarios finales*: Utilizan los servicios de la nube, gratuitamente o con una tarifa.

### **Características de la Nube**

1. *Autoservicio bajo demanda*: Un consumidor puede proveerse unilateralmente de tiempo de servidor y almacenamiento en red, a medida que lo necesite; sin requerir interacción humana con el proveedor del servicio.
2. *Acceso ubicuo a la Red*: Se realiza mediante mecanismos estándares, que promueven el uso por plataformas de clientes (teléfonos móviles, computadoras portátiles, PDAs, tabletas).
3. *Distribución de recursos independientes de la posición*: Los recursos de computación del proveedor son agrupados para servir a múltiples consumidores utilizando un modelo multi-distribuido con diferentes recursos físicos y virtuales asignados y reasignados dinámicamente conforme a la demanda del consumidor. Existe una sensación de independencia de la posición, de modo que el cliente, normalmente, no tiene control ni conocimiento sobre la posición exacta de los recursos proporcionados.
4. *Elasticidad rápida*: Las funcionalidades se pueden proporcionar de modo rápido y elástico, en algunos casos automáticamente. Sus características de aprovisionamiento dan la sensación de ser ilimitadas y pueden adquirirse en cualquier cantidad o momento.
5. *Servicio medido*: Los sistemas de computación en la nube controlan y optimizan automáticamente el uso de recursos, potenciando la capacidad de medición en un nivel de abstracción apropiado al tipo de servicio (almacenamiento, procesamiento, ancho de banda y cuentas activas de usuario). El uso de recursos puede ser monitorizado, controlado e informado, proporcionando transparencia para el proveedor y para el consumidor.

El modelo según NIST, se compone de cinco características esenciales, tres modelos de servicio y cuatro modelos de despliegue.

*Modelos de despliegue:* Se refieren a la posición (localización) y administración (gestión) de la infraestructura de la nube (Pública, Privada, Comunitaria, Híbrida).

*Modelos de servicio:* Se refieren a los servicios específicos a los que se puede acceder en una plataforma de computación en la nube (Software, Plataforma e Infraestructura como Servicios).

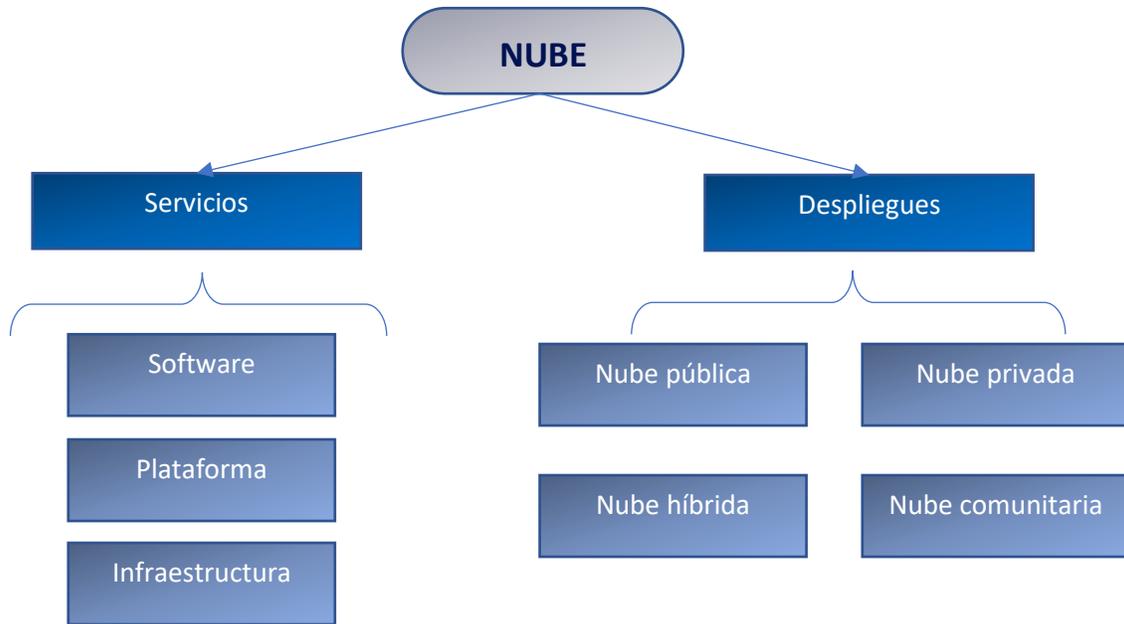


Ilustración 11. Servicios de la Nube

*Fuente: Elaboración propia*

### Modelos de servicio:

1. *Software:* Al usuario se le ofrece la capacidad de que las aplicaciones se desenvuelvan en una infraestructura de la nube, siendo las aplicaciones accesibles a través de un navegador web, como en el correo electrónico.
2. *Plataforma:* Al usuario se le permite desplegar aplicaciones propias (adquiridas o desarrolladas por el propio usuario) en la infraestructura de la nube de su proveedor, que ofrece la plataforma de desarrollo y las herramientas de programación. En este caso, el usuario mantiene el control de la aplicación, aunque no de toda la infraestructura subyacente.
3. *Infraestructura:* El proveedor ofrece recursos como capacidad de procesamiento, de almacenamiento o comunicaciones, que el usuario puede utilizar para ejecutar cualquier software; desde sistemas operativos hasta aplicaciones. Los modelos de

despliegue de las infraestructuras y servicios de la nube se clasifican en las siguientes categorías:

**Modelos de despliegue:**

1. *Nube privada*: Los servicios no son ofrecidos al público en general, la infraestructura es íntegramente gestionada por una organización.
2. *Nube Pública*: La infraestructura es operada por un proveedor que ofrece servicios al público en general.
3. *Nube híbrida*: Resultado de la combinación de dos o más nubes individuales que pueden ser privadas, compartidas o públicas. Permite enviar datos o aplicaciones entre ellas.
4. *Nube comunitaria*: Ha sido organizada para servir a una función o propósito común. Es preciso compartir objetivos comunes (misión, políticas, seguridad). Este modelo es el definido por el NIST, aunque la mayoría de las organizaciones, proveedores y usuarios de la nube aceptan los tres modelos de despliegue: pública, privada e híbrida.

**Modelos de negocios en la nube:**

1. *Los servicios en la nube*: Proporcionan la red e infraestructuras de computación mediante plataformas y soluciones. Los proveedores en la nube permiten desarrollar y proporcionar soluciones desde la perspectiva de los consumidores. Los proveedores son variados y tienen gran implantación, aprovechando sus centros de datos y de experiencia en alojamiento de datos y aplicaciones.
2. *Proveedor de servicios de plataformas de la nube*: Proporcionan entornos alojados en la nube, hospedados en sistemas e infraestructuras específicos, para que los desarrolladores puedan acceder y desarrollar una nueva aplicación de negocios y alojarla en la plataforma.
3. *Proveedores de tecnologías*: Desarrollan las herramientas y tecnologías que facilitan que la nube se establezca y proporcione a los consumidores diversos recursos como herramientas, tecnologías, sistemas operativos para facilitar el despliegue de nubes públicas, privadas, híbridas y comunitarias.
4. *Proveedores de soluciones*: Desarrollan aplicaciones o suites completas, para conseguir un amplio mercado de consumidores de la nube.
5. *Modelos de negocio para consumidores*: Estas empresas aplican conceptos de la nube a sus estrategias de negocios, ofrecen soluciones para gestión empresarial.

## Seguridad en la nube

La seguridad en la Nube es un tema que puede provocar una sensación de que el nivel de seguridad es inferior al de los modelos tradicionales. Sin embargo, si las políticas de seguridad del proveedor están bien definidas, y el cliente las ejecuta fielmente, trabajar en la nube supondrá una mejora en la seguridad.

El usuario no sabe exactamente dónde está almacenada la información, por lo que los proveedores son los encargados del almacenamiento de la información y su seguridad. IBM<sup>3</sup> llama a este tipo de seguridad “Secure by Design” (Seguridad Personalizada), este concepto se refiere a que el proveedor diseña un servicio específico para cada empresa basándose en sus preferencias y necesidades (Aguilar, L. J.,2012).

Beneficios del almacenamiento en la nube:

- ❶ *Costo:* Con el almacenamiento en la nube, no es necesario comprar hardware del cual no se estará seguro si es adecuado para las necesidades del mismo cliente, con la nube se puede contar con la capacidad necesaria, modificar el desempeño y con rapidez pagando solamente el almacenamiento que se utilice.
- ❷ *Tiempo de implementación:* La infraestructura no retrasa la ejecución del equipo de desarrollo, lo cual permite al departamento de TI concentrarse en resolver problemas de aplicación complejos en lugar de administrar sistemas de almacenamiento.
- ❸ *Gestión de la información:* Al utilizar políticas de administración del ciclo de vida del almacenamiento en la nube, puede realizar potentes tareas de administración de la información, incluida la separación por niveles automatizada o el bloqueo de datos para cumplir con los requisitos de conformidad (AWS).

Desventajas del almacenamiento en la nube:

- ❶ Los datos pueden acabar en manos de terceros.
- ❷ Se necesita tener acceso a Internet en todo momento.
- ❸ Falta de privacidad.

---

<sup>3</sup> IBM: International Business Machines Corporation, es una empresa multinacional estadounidense que ofrece soluciones empresariales como: sistemas y gestión de big data, cloud computing, seguridad digital e intranet, herramientas de IA y más.

## 2.7. Bases de datos, MySQL y Java Script

### Bases de datos

Podemos definir una base de datos como una colección de datos relacionados, organizados, estructurados y almacenados de manera persistente, es decir que permite recuperarlos después de haberlos almacenado. Esta colección de datos debe estar organizada según un modelo de estructuras de almacenamiento.

Una base de datos necesita un conjunto de programas que procesen, recuperen, compartan, aseguren y controlen sus procesos, a esto lo conocemos como sistema administrador de bases de datos.

Cuadro de Niveles de Abstracción (véase Ilustración 12):

NIVEL	DESCRIPCIÓN
<b>Nivel físico o interno</b>	Descripción del almacenamiento físico de los datos.
<b>Nivel conceptual o lógico</b>	Descripción de la Base de Datos en términos de estructuras de almacenamiento. Este conjunto de estructuras se conoce como Esquema
<b>Nivel externo o de vistas</b>	Es un conjunto de vistas a los datos que evita mostrar todos los datos almacenados y están orientadas a usuarios específicos.

Ilustración 12. Cuadro de niveles de abstracción

*Fuente: Méndez, UNAM*

Un sistema de administración de bases de datos debe contar con lenguajes (véase Ilustración 13) que permitan realizar consultas, controlar accesos y manipular el almacenamiento de los datos, un ejemplo es SQL (Méndez,2019).

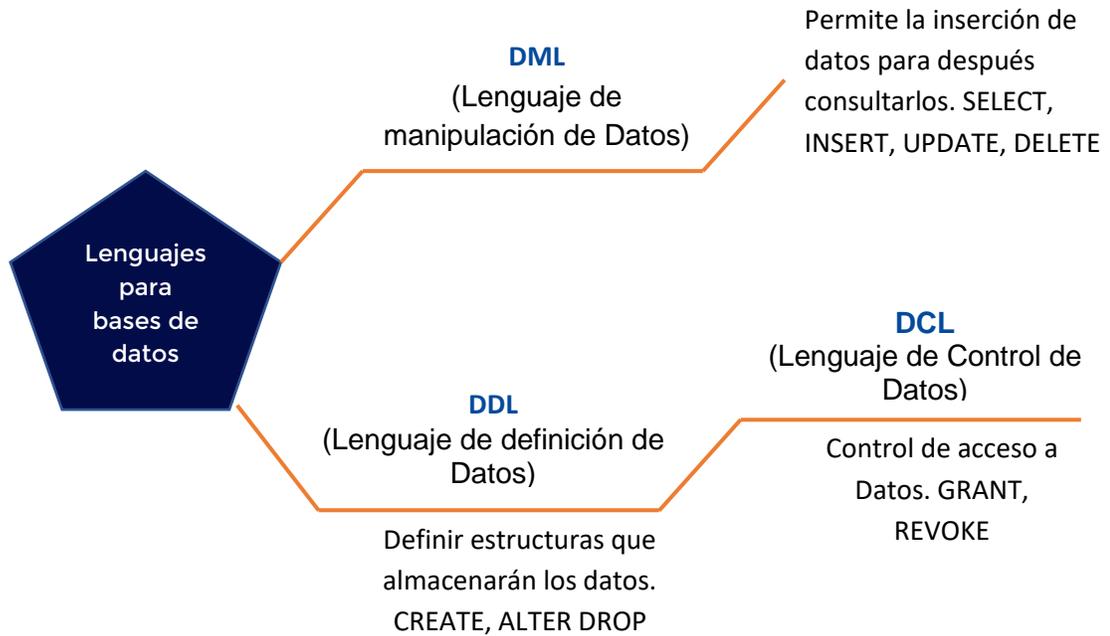


Ilustración 13. Lenguajes para bases de datos

Fuente: Elaboración propia.

Una DBMS comprende también un software encargado de hacer las gestiones con el sistema operativo y brindar servicios a las bases de datos. Este software incluye programas especialidades para actualizar, recuperar, asegurar y compartir los datos de la base (véase Ilustración 14).

COMPAÑÍA	SOFTWARE	TIPO
Oracle	Oracle	Comercial
Microsoft	SQL Server	Comercial
Postgresql	Postgresql	Libre
MySQL	MySQL	Libre
IBM	Db2 Universal Database	Comercial

Ilustración 14. Sistema de administración de bases de datos

Fuente: Méndez, UNAM

## Modelo de datos

Un modelo de datos es una colección de herramientas conceptuales para describir los datos, sus relaciones, su semántica y las restricciones de consistencia. Existen dos modelos principales: el relacional y el orientado a objetos, sin embargo, actualmente el modelo más extendido y utilizado es el relacional (Silberschatz, 2006).

## Modelo relacional

El modelo relacional fue propuesto en 1970 por Edgar Frank Codd<sup>4</sup>. Este modelo se basa en estructuras de almacenamiento de datos llamadas relaciones. Sus características fundamentales son:

1. Los datos son percibidos por el usuario como relaciones.
2. Para consultar los datos el usuario cuenta con operados que generan nuevas relaciones a partir de las otras (álgebra relacional).

## Tupla

Una tupla o registro es un conjunto de campos que contienen los datos que pertenecen a una misma entidad.

Propiedades de una relación:

- ⊗ No existen tuplas repetidas
- ⊗ Las tuplas están en desorden
- ⊗ Los atributos están en desorden
- ⊗ Los atributos deben contener valores atómicos, eso significa que cada para valor para cada tupla en un determinado atributo no debe ser semánticamente divisible

## Álgebra Relacional

Es una serie de operadores que permiten manipular las relaciones (consultar datos), algunos son:

- |                |                 |
|----------------|-----------------|
| 1. Restricción | 5. Intersección |
| 2. Proyección  | 6. Diferencia   |
| 3. Producto    | 7. Junta (join) |
| 4. Unión       | 8. División     |

---

<sup>4</sup> Edgar Frank Codd fue un científico informático inglés, conocido por crear el modelo relacional de bases de datos

## **SQL**

SQL por sus siglas en inglés Structured Query Language, en español Lenguaje de Consulta Estructurada es un lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales. Es un lenguaje declarativo y parecido al lenguaje natural, se parece al idioma inglés y es muy expresivo (2002, ESCOFET).

Algunas sentencias de SQL son:

1. CREATE: Para crear Bases de Datos, tablas, dominios y vistas.
2. ALTER: Para modificar tablas y dominios.
3. DROP: Para borrar Bases de Datos, tablas, dominios, aserciones y vistas.

## **Java Script**

JavaScript es un lenguaje de programación basado en scripts, que se puede incorporar dentro de un documento HTML. En este contexto, un script podría definirse como un conjunto de sentencias o instrucciones interpretadas y ejecutadas por un navegador web, encargado de realizar la tarea de interpretar el código de programación de JavaScript.

La principal característica de JavaScript es que posibilita la dinámica e interacción en una página web, y aunque es un lenguaje muy simple, es bastante completo para la programación del lado del cliente (del navegador), motivo por el cual los desarrolladores web lo encuentran flexible y sencillo de usar (DGTIC).

## **HTML (Hyper Text Markup Language):**

El acceso a la Web suele hacerse a través de un explorador web, el cual es capaz de interpretar instrucciones contenidas en el documento HTML y muestra la página como la vemos (véase Ilustración 15). Este lenguaje está basado en etiquetas (Universidad de Barcelona).

Un ejemplo base de código HTML es el siguiente:

```
index.html
1 |<!--CODIGO HTML-->
2 |<html>
3 |<head>
4 |<title> Facultad de Ingeniería</title>
5 |</head>
6 |<body>
7 |   <H1> Karla Mariana Rosas Jiménez</H1>
8 |
9 |</body>
10|</html>
```

Ilustración 15. Estructura de código HTML

Fuente: Elaboración propia.

### CSS (Cascading Style Sheets)

Una página web es realmente un documento de texto, en dicho documento se escribe código HTML con el que se crea el contenido de una web y por otro lado existe el código CSS que unido con HTML permite darle forma, color y posición a una página de forma masiva. Se le denomina estilos en cascada porque se aplican de arriba abajo siguiendo un patrón de herencia (véase Ilustración 16).

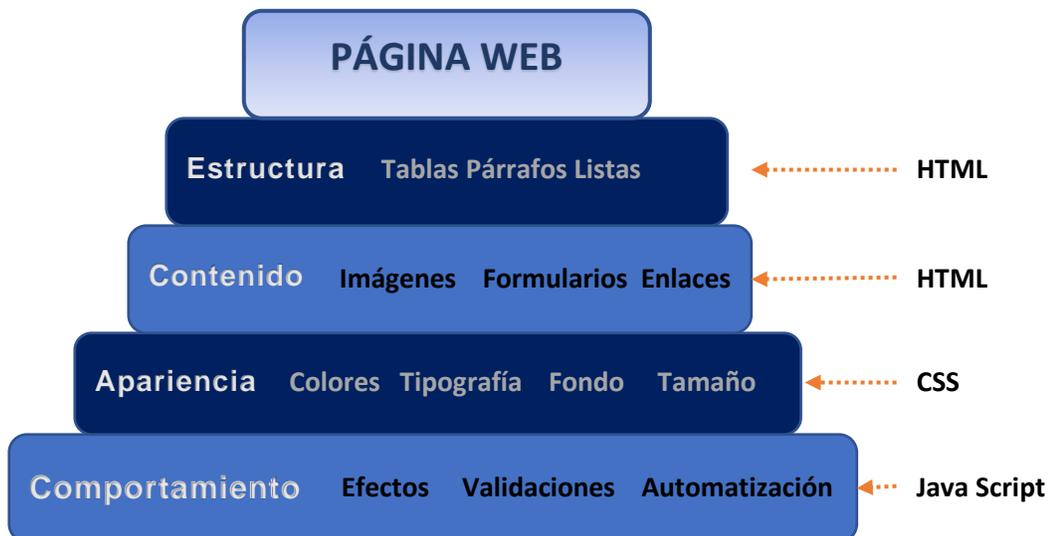


Ilustración 16 Estructura de una página web

Fuente: Elaboración propia.

## 2.8. App Maker Y Cloud SQL

### App Maker

App Maker de Google es un entorno de bajo nivel de codificación de Enterprise G Suite Google. Facilita el diseño de la interfaz de usuario con la función de arrastrar objetos y modelado de datos declarativos. Una de las principales características es la obtención de estadísticas y la posible conexión con gmail, hojas de cálculo o calendario. Permite crear un modelo para administrar datos, utiliza Apps Script a fin de escribir sentencia de comandos basados en Java Script. Para utilizar fuentes de datos externas y compilar nuestras aplicaciones es necesario hacer la respectiva conexión a Cloud SQL (véase Ilustración 17).

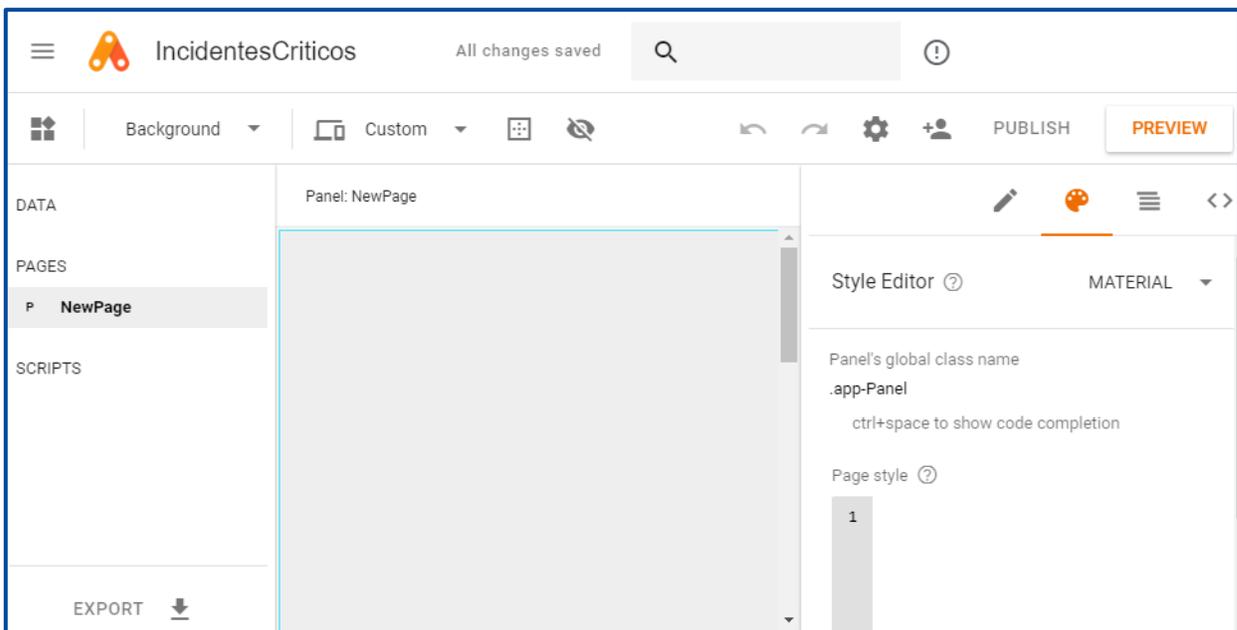


Ilustración 17. Interfaz del ambiente de desarrollo App Maker

*Fuente: Página App Maker Google*

## Los datos en App Maker

Los modelos organizan los datos los almacenan en App Maker. Son como una tabla ordinaria en una base de datos o una hoja en una hoja de cálculo. Las hojas de cálculo organizan datos por columnas y los almacenan en filas. Los modelos organizan datos por campos y los almacenan en colecciones llamadas registros (véase Ilustración 18).

Por ejemplo, para realizar un modelo de “Empleado” con cuatro campos: nombre, número de empleado, activo y fecha de cumpleaños. Si se desea realizar dos registros, la forma de almacenamiento sería la siguiente:

MODELO: EMPLEADO			
Nombre	ID Empleado	Activo	Fecha de cumpleaños
Carter Burke	2179426	Falso	30/03/2150
Ellen Ripley	1612122	Verdadero	7/01/2092

Ilustración 18. Ejemplo de Almacenamiento en App Maker

*Fuente: Página de Google App Maker*

Algunos conceptos para comprender los modelos son:

- 🔗 **Campos:** Agrega campos nuevos a un modelo y edita la configuración del campo. Por ejemplo, se puede agregar un campo Email al modelo Empleado y configurarlo según se requiere (véase Ilustración 19).
- 🔗 **Fuentes de datos:** Se define como un modelo recupera y almacena datos cuando se lo consulta, incluido el subconjunto de registros que muestra. Por ejemplo, se puede crear una fuente de datos para el modelo “Empleado” que solo muestre empleados activos cuando sea necesario.
- 🔗 **Eventos:** Crea secuencias de comandos del lado del servidor que se ejecuten cuando tu aplicación muestra, crea o borra registros.
- 🔗 **Relaciones:** Describe las relaciones entre registros dentro de un modelo o entre modelos. Por ejemplo, se puede crear una relación en el modelo Empleado entre un administrador y sus subordinados directos. No está disponible para modelos calculados.
- 🔗 **Seguridad:** Controla qué usuarios pueden crear, cargar, guardar o borrar registros en un modelo. Por ejemplo, se puede restringir el modelo Empleado para que solo los empleados puedan ver los registros en el modelo.

Tipo	Descripción	Ejemplo
String	Cadena de caracteres de Unicode	Campo: Nombre Contenido: Karla
Number	Número de punto flotante de 64 bits	Campo: Id Empleado Contenido: 123456
Booleano	True o False	Activo Contenido: true
Date	Objeto fecha	Campo: Fecha de cumpleaños Contenido: 2020-12-19

Ilustración 19. Tipos de datos en App Maker

*Fuente: Página de Google App Maker*

Algunos de los modelos compatibles con App Maker son:

- **Google Cloud SQL** que es una tabla de MySQL tradicional que se puede compartir entre varias aplicaciones,
  - **Predeterminada:** un administrador de G Suite puede establecer una instancia de Cloud SQL compartida entre aplicaciones de App Maker en una organización. Una vez habilitada, se crea una base de datos nueva de forma automática cuando agregas un modelo de datos de Cloud SQL a una aplicación.
  - **Personalizada:** una vez que tu administrador configura una instancia de Cloud SQL predeterminada, también puedes establecer tu propia instancia de Cloud SQL. Esta opción es de utilidad en las siguientes circunstancias:
    - La aplicación se entrega a varios usuarios o almacena una gran cantidad de datos.
    - La base de datos se debe compartir con otras aplicaciones.
    - Debes administrar la base de datos o retener el control de la instancia de Cloud SQL.
- **SQL calculado:** una consulta que se ejecuta en una base de datos de Google Cloud SQL.
- **Directorio:** un modelo que obtiene información del directorio de la organización, por ejemplo, las direcciones de correo electrónico y los números de teléfono.

## Cloud SQL

Es un servicio de bases de datos relacionales gestionado para MySQL, PostgreSQL y SQL Server. Este servicio proporciona herramientas para la configuración, mantenimiento y administración de las bases de datos, ofreciendo altos niveles de rendimiento y disponibilidad, así como escalabilidad y comodidad, basada en una infraestructura diseñada para el futuro que cuenta con la excepcional seguridad y la red global privada de Google.

Cloud SQL se adapta a una gran variedad de aplicaciones como sitios web de WordPress, aplicaciones de comercio electrónico, herramientas de gestión de la relación con clientes (CRM<sup>5</sup>) y aplicaciones geoespaciales compatibles. Es posible automatizar los sistemas de seguridad, replicación, parches y actualizaciones, a la vez garantiza una disponibilidad superior al 99,95 % en cualquier parte del mundo. Tiene una escalabilidad con hasta 30 TB de capacidad de almacenamiento, 60.000 IOPS<sup>6</sup> y 416 GB de RAM por instancia.

### 2.9. Factores críticos de éxito en los proyectos de software

En 1981, Rockart y Bullen propusieron utilizar este el método de los Factores críticos de éxito como un componente para realizar la planificación estratégica en los temas de tecnología y los sistemas de información. Actualmente este método sigue siendo de gran utilidad en los proyectos (Caralli,2004).

En la siguiente tabla se agrupan algunos de los factores más importantes para la administración exitosa de un proyecto de software. Esta información ha sido recabada de las fuentes bibliográficas de diferentes autores que se especializan en este tema y consultada en un artículo realizado por UAEM en el 2011 (véase Tabla 2).

---

<sup>5</sup> CRM: Customer Relationship Management, administración basada en la relación con los clientes.

<sup>6</sup> IOPS: Input/output operations per second, es una medida de rendimiento de entrada/salida utilizada para caracterizar dispositivos de almacenamiento de computadora.

Tabla 2. FCE para la administración exitosa de los proyectos de software

FACTORES CRÍTICOS DE ÉXITO PARA ADMINISTRAR EXITOSAMENTE UN PROYECTO DE SOFTWARE	
Autor	FCE
Standich Group (1995)	<ul style="list-style-type: none"> <li>▪ Involucramiento del usuario</li> <li>▪ Apoyo por parte de la alta gerencia</li> <li>▪ Definición clara de los requerimientos</li> <li>▪ Planeación apropiada</li> <li>▪ Expectativas realistas</li> <li>▪ Metas intermedias alcanzables</li> <li>▪ Equipo de trabajo competente</li> <li>▪ Sentido de pertenencia al proyecto</li> <li>▪ Visión y objetivos claros</li> <li>▪ Equipo comprometido y disciplinado.</li> </ul>
Guido y Clements (1999)	<ul style="list-style-type: none"> <li>▪ Objetivos y misión del proyecto claramente definido</li> <li>▪ Apoyo de la alta gerencia</li> <li>▪ Un administrador del proyecto competente</li> <li>▪ Recursos suficientes</li> <li>▪ Involucrar y consultar al cliente o usuario final</li> <li>▪ Buena comunicación</li> <li>▪ Responsabilidad del cliente</li> <li>▪ Monitoreo apropiado y retroalimentación</li> <li>▪ Tecnología apropiada</li> </ul>
Nicholas (2001)	<ul style="list-style-type: none"> <li>▪ Compromiso</li> <li>▪ Involucramiento de los participantes</li> <li>▪ Buena comunicación</li> <li>▪ Calidad en el intercambio de información</li> </ul>
Antony (2002)	<ul style="list-style-type: none"> <li>▪ Desarrollo de sus empleados</li> <li>▪ Trabajo en equipo</li> </ul>
Achanga (2006)	<ul style="list-style-type: none"> <li>▪ Liderazgo</li> <li>▪ Apoyo</li> </ul>

Fuente: Bermúdez, 2003; Báñales y Adam,2007; UAEM,2011

A lo largo del tiempo, otros autores han hecho grandes aportaciones referentes al éxito de los proyectos. A continuación, se presentan los resultados obtenidos en diferentes investigaciones (véase Ilustración 20).

Una de las investigaciones hechas por expertos como Hartmann y Ashrafi (2002), encontraron que existían diez FCE (véase Tabla 3).

Tabla 3. Factores críticos de éxito

Orden	Factores Críticos de Éxito
1	El dueño del proyecto está informado de sus estatus y se pide su aprobación en cada una de las etapas del proyecto.
2	El dueño del proyecto es consultado en todas las etapas del desarrollo e implementación.
3	Se establecen canales de comunicación apropiados en cada uno de los niveles del equipo de proyecto.
4	El proyecto tiene claramente definida su misión.
5	La alta gerencia provee los recursos necesarios (dinero, capacitación y equipo).
6	El proyecto logra el propósito del negocio.
7	Un plan de proyecto detallado (incluyendo tiempo, calendario y punto de revisión) con un presupuesto detallado.
8	Se encuentra disponible la tecnología y los conocimientos adecuados.
9	Los cambios en el proyecto son administrados bajo un proceso formal.
10	El proyecto es completado con un mínimo de cambios en el alcance y mudamente acordados.

Fuente: Hartman y Ashrafi, 2002.



Ilustración 20 . Modelo de FCE propuesto por The Author(s).

Fuente: *Software Quality Journal*,2018

### 3. Metodología de desarrollo de la aplicación

En este apartado se detalla acerca del proceso de desarrollo de la aplicación. Para explicar el funcionamiento de la aplicación y diseño de la interfaz, creé un modelo basado en el proyecto original, el cual no muestra la interfaz original, ni el modelo E/R o algún dato de la empresa “K” con el fin de respetar la privacidad y confidencialidad de esta. No obstante, los detalles mostrados a continuación ofrecen un panorama semejante al desarrollo del proyecto que se realizó durante el período comprendido de septiembre 2019 a noviembre 2019. Durante este período, mi rol fue el de gestionar y dar seguimiento al proyecto en cada una de las etapas (véase Tabla 4), con el objetivo de cumplir en tiempo y forma el proyecto.

Tabla 4. Etapas del desarrollo del proyecto

Etapa	Fecha	Descripción
Planeación y Análisis de requerimientos.	1 Sep 2019 al 21 Sep 2019	<ul style="list-style-type: none"> <li>• <b>Del 1° al 7 septiembre 2019</b> Primer acercamiento entre los clientes, el equipo de desarrollo y los expertos en los dominios de la industria a través de reuniones virtuales. El objetivo de estas reuniones es conocer las expectativas y necesidades de los clientes.</li> <li>• <b>Del 8 al 14 septiembre 2019</b> El responsable de la gestión del proyecto comienza a recopilar los requerimientos del proyecto para comenzar a estructurarlo, basándose en las diferentes reuniones concretadas con los clientes.</li> <li>• <b>Del 15 al 21 septiembre 2019</b> El responsable del proyecto analiza y verifica los requerimientos del proyecto, define el alcance y estructura un documento SRS (Software Requirement Specification).</li> </ul>
Definición de los requerimientos	22 Sep 2019 al 28 Sep 2019	Una vez completo el análisis de los requisitos y creado el SRS, el siguiente paso consiste en obtener la aprobación del cliente.
Diseño de la arquitectura del producto	29 Sep 2019 al 12 Oct 2019	El arquitecto del software y el diseñador basados en el SRS, comienzan a crear una propuesta de estructura del proyecto de software. El diseñador propone un diseño visual de la aplicación mientras

		<p>que el arquitecto diseña la base y estructura de datos.</p> <p>En esta etapa se tiene una reunión con los clientes para mostrarles un prototipo de la aplicación, su funcionamiento y su diseño gráfico, con el objetivo de saber que se cubren las expectativas de los clientes y detectar si es necesario hacer alguna modificación en el diseño.</p>
Construcción y desarrollo del producto	13 Oct 2019 al 9 Nov 2019	En esta etapa de SDLC, el desarrollo real comienza y el producto se crea.
Pruebas del producto	10 Nov 2019 al 23 Nov 2019	<ul style="list-style-type: none"> <li>• <b>Del 10 al 15 de noviembre 2019</b> El integrante del equipo de rol “tester” realiza pruebas del producto para identificar los defectos y problemas, informa acerca de su existencia para corregirlos y cumplir con los estándares de calidad definidos en el SRS.</li> <li>• <b>Del 15 al 23 de noviembre 2019</b> El producto de software es liberado para un grupo de usuarios finales y expertos del lado del cliente, para realizar pruebas de uso y funcionamiento. En caso de existir observaciones, estas serán notificadas al equipo de desarrollo para su corrección.</li> <li>• <b>Del 24 al 28 de noviembre 2019</b> Esta semana permite finalizar correcciones y ajustes al producto de software en caso de sobrepasar el tiempo establecido, permitiendo terminar el producto en tiempo.</li> </ul>
Lanzamiento y mantenimiento	29 Nov 2019	Una vez que el producto ha sido probado y está listo para ser implementado, se lanza a producción. El mantenimiento al producto se realiza de manera semanal para verificar que está funcionando correctamente la aplicación.

Fuente: Elaboración propia

### 3.1. Análisis

Para llevar a cabo la implementación de la aplicación Web cuyo objetivo era cumplir con mejoras en el proceso “Incidentes Críticos” fue necesario realizar un análisis del proceso “Incidentes Críticos”. Uno de los primeros pasos para este análisis fue estudiar el proceso e interactuar con las personas involucradas, a fin de comprender sus actividades e identificar las áreas de oportunidad en el proceso actual.

Se realizaron reuniones virtuales a las cuales yo y otro miembro del equipo asistíamos para recopilar los requerimientos para la aplicación web, identificar las mejoras a realizar con respecto al proceso tradicional y comenzar a definir el alcance del proyecto. Estas reuniones se realizaban junto con los representantes de los equipos M y D, quienes eran los expertos en el proceso de incidentes críticos. Eventualmente, otros miembros del equipo de desarrollo asistían a estas reuniones con el fin de conocer las necesidades del cliente y buscar las mejores soluciones tecnológicas. (véase Tabla 5).

Tabla 5. *Proceso tradicional y propuesta*

Proceso tradicional	Proceso con propuesta de mejora
<ul style="list-style-type: none"> <li>❶ No hay una herramienta específica para este proceso.</li> <li>❷ La creación de notificaciones vía correo se crea sobre una plantilla de texto plano a mano, y se actualiza cada vez que hay una actualización.</li> <li>❸ La Experiencia del usuario no era agradable, incluso si se podía llevar a cabo el proceso.</li> <li>❹ Al final de cada Incidente Crítico el Equipo D, creaba una hoja de Cálculo a mano con la información relevante y la cronología de las notificaciones que daban seguimiento al proceso.</li> <li>❺ Para acceder al historial de un IC era necesario buscar la Hoja de Cálculo o buscar el correo electrónico de un CI específico.</li> </ul>	<ul style="list-style-type: none"> <li>❶ Intuitiva y fácil de usar, mejorando la experiencia del usuario.</li> <li>❷ Generación y envío automático de los comunicados.</li> <li>❸ Facilitar la comunicación efectiva entre las personas que colaboran en el proceso.</li> <li>❹ Optimización del tiempo dedicado a este proceso.</li> <li>❺ Creación de un historial por cliente de cada IC.</li> <li>❻ Centralización de la información.</li> </ul>

Fuente: *Elaboración propia*

## 3.2. Planeación y definición

### 3.2.1. Equipo-Organización de trabajo

Posterior a identificar las necesidades de los clientes y crear una lista con los requerimientos del proceso de incidentes críticos (véase Tabla 6), se puso en marcha la creación de un plan de desarrollo con los miembros del equipo Digital IT.

Tabla 6. *Requerimientos para el proceso IC*

N°	Requerimientos para el proceso de incidentes críticos
1	Comunicación entre equipos D, M y equipo de operación.
2	Contar con un historial por cada IC y poder consultarlos por cliente externo.
3	Comunicados automáticos vía correo electrónico con las notificaciones del Estatus del IC
4	Registrar fecha, hora y responsable de cada actualización, ya que al ser un proceso crítico es importante llevar un control
5	Mejorar la experiencia de usuario a lo largo de la gestión de incidentes críticos.
6	Centralizar la información y mantenerla estructurada y disponible para su uso
7	Manejar estados por incidente: Sin problema (sin incidente actual) o tipo de incidente (incidente abierto)
8	Considerar tipos de incidente: Prioridad alta e Incidente crítico

*Fuente: Elaboración propia*

El modelo utilizado para el desarrollo de la aplicación fue Híbrido, el cual consistió en tener como base el Ciclo de Vida del Software y desarrollar con una metodología ágil. Se propusieron un conjunto de fechas de entrega y seguimiento a los usuarios al menos una vez por semana (Ver Ilustración 21), para que ellos pudieran seguir el avance del desarrollo de la aplicación y verificar que se iba por el buen camino. De manera adicional, había flexibilidad en cuanto agendar más de una sesión a la semana si era necesario.

Periodo	Fecha inicio de periodo semanal 2019	Periodo	Fecha inicio de periodo semanal 2019
1	1 de septiembre	8	20 de octubre
2	8 de septiembre	9	27 de octubre
3	15 de septiembre	10	3 de noviembre
4	22 de septiembre	11	10 de noviembre
5	29 de septiembre	12	17 de noviembre
6	6 de octubre	13	24 de noviembre
7	13 de octubre	Liberación a producción	29 de noviembre

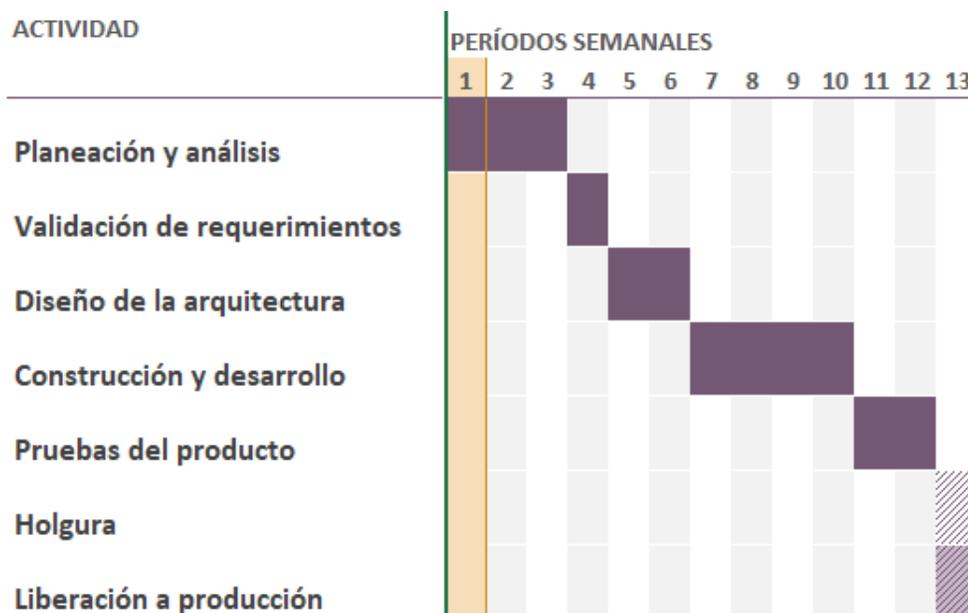


Ilustración 21. Diagrama de tiempo, desarrollo del proyecto

Fuente: Elaboración propia

Una ventaja de este tipo de desarrollo es que permite al usuario verificar si es necesario modificar o agregar algo nuevo que no se consideró en la etapa de análisis, en ese momento se evalúa el impacto en tiempo y esfuerzo para tomar una decisión. Se le comunica al cliente el costo de estos cambios y se negocia de manera que ambas partes estén conscientes de lo que implican los cambios. El desarrollo basado en una metodología híbrida garantiza que el producto final empate o mejore las expectativas de los usuarios.

### 3.3. Diseño

#### 3.3.1. Base de datos E/R

Para el modelado de la base de datos se contempló toda la información recopilada en la etapa de análisis. Se realizan bosquejos contemplando el flujo de cómo debe funcionar la aplicación y se crean los modelos de datos necesarios para crear de igual manera la relación entre ellos (Véase Ilustración 22).

La base estaba compuesta de 6 modelos de datos principales y un directorio:

**Cliente:** En esta se consideran los datos relevantes de los clientes externos de la empresa K.

**Contacto:** Es una fuente relacionada con la fuente “Cliente”, ya que cada cliente tiene una o varias personas encargadas de la comunicación con la empresa K.

**Incidentes:** En esta fuente se almacenan los diferentes IC que surgen para todos los clientes de la empresa K. Un cliente puede tener uno o más incidentes relacionados.

**Incidente actualización D:** El equipo D puede contribuir con uno o más comunicados de estatus de un mismo incidente.

**Estatus M:** El equipo M puede contribuir con uno o más comunicados de estatus de un mismo incidente.

**Fin de Incidente:** En esta fuente se almacenan los comunicados finales por cada incidente. Contienen información acerca del origen y resolución del problema.

**Directorio Google G-Suite:** Información de los miembros internos de la empresa K, tal como nombre(s), apellido(s), correo, puesto, empresa, número telefónico, dirección de oficina. Esta información está contenida en una base de datos externa, perteneciente a G-Suite.

Con base a las premisas anteriores, se creó el siguiente diagrama de entidad relaciones.



Tipo de dato por campo

# Number S String # Date

Relación: M muchos / 1 uno

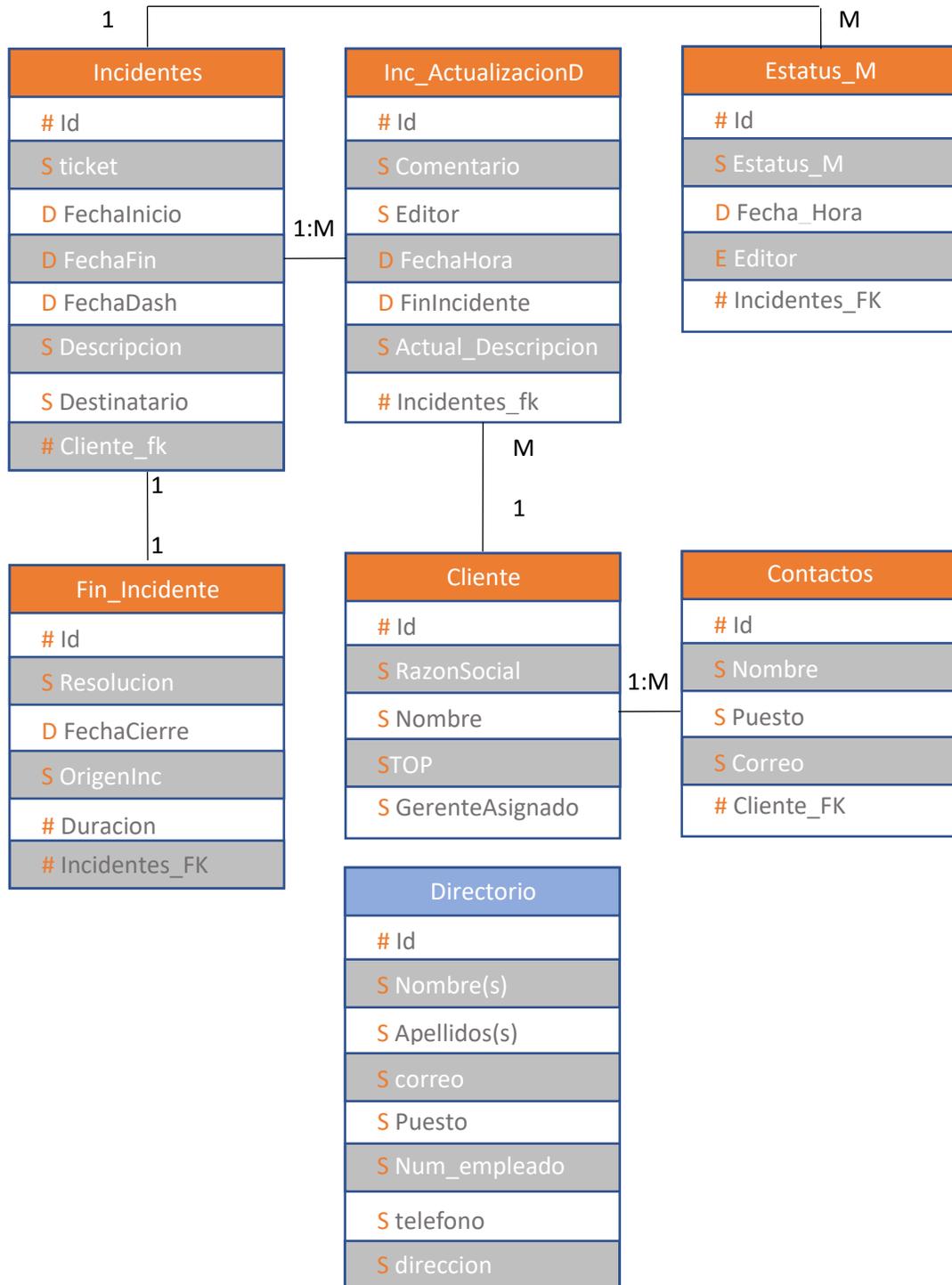


Ilustración 22. Diagrama E/R de la aplicación IC

Fuente: Elaboración propia

### 3.3.2. Prototipos de diseño

Dentro del equipo de Digital IT hay un integrante dedicado al Diseño Gráfico quien se encarga del diseño del “look and feel” de la aplicación como colores, logos y flujos de pantallas. El diseñador trabaja en equipo con los analistas para comenzar con los primeros prototipos gráficos y verificar que se incluyen los requerimientos. Una vez creados los prototipos es importante contar con la validación por parte del cliente y así comenzar con el desarrollo. Cabe mencionar que incluso después de comenzar el desarrollo pueden surgir cambios en la interfaz, sin embargo, con la validación constante del usuario estos cambios son mínimos.

#### Prototipo 1. Página Principal

En la página principal se muestra un diseño que incluye: logo de la aplicación, nombre y foto del usuario para mejorar su experiencia y que sea personalizada, una barra de búsqueda para optimizar el acceso información precisa, el título de la aplicación, un botón para notificar que hay un Incidente en proceso y el Dashboard (Véase Ilustración 23).

El Dashboard está diseñado para la interpretación inmediata de la información. Se muestran los incidentes que han surgido en la columna de la fecha y en el renglón del cliente asociado. En la última columna podemos ver el estatus actual del cliente en tiempo real; si el indicador es verde significa que no hay ningún incidente, pero si es rojo, hay un incidente en ese momento.



Ilustración 23. Prototipo 1, Página Principal “Dashboard”

*Fuente: Elaboración propia en el entorno App Maker*

Los indicadores están representados por círculos de color rojo o verde según el estatus del cliente, al darle clic sobre algún indicador se desplegará una pantalla con el historial del incidente respectivo.

### Prototipo 2. Página Secundaria “Detalles del incidente”

En la siguiente imagen se presenta la interfaz para visualizar el detalle de los IC Registrados (Véase Ilustración 24).

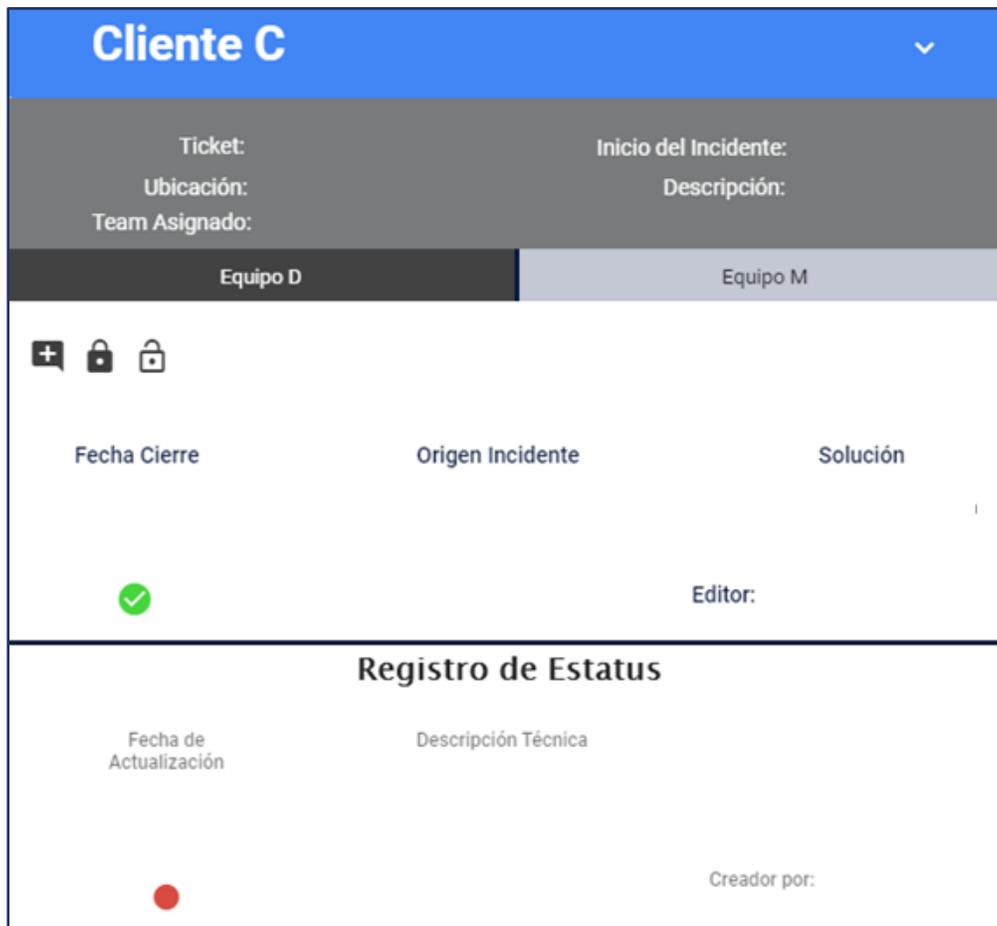


Ilustración 24. Prototipo 2, Página “Detalles del incidente”

Fuente: Elaboración propia en el entorno App Maker

### Prototipo 3. “Detalles del cliente”

En la siguiente imagen se presenta la interfaz para visualizar la información de detallada de cada cliente (Véase Ilustración 25).

The screenshot shows a user interface for viewing customer details. At the top, there is a blue header with the text 'Cliente C' and a dropdown arrow. Below the header, the information is organized into two columns. The first column contains 'Nombre:', 'Razón Social:', 'Ticket: F147258369', 'Ubicación: Edificio Central', and 'Team Asignado: Operación D'. The second column contains 'TOP:', 'Gerente Asignado:', 'Inicio del Incidente: 31/01/20', and 'Descripción: Sin conexiones'. At the bottom, there are two tabs: 'Equipo D' (selected) and 'Equipo M'.

Ilustración 25. Prototipo 3, "Detalles del cliente"

Fuente: Elaboración propia en el entorno App Maker

### Prototipo 4. “Carga del incidente”

Los usuarios autorizados en cargar incidentes lo hacen dando clic en el botón  y llenando los campos necesarios (Véase Ilustración 26).

The screenshot shows a form titled 'Levantamiento Incidente'. At the top left is a user icon and at the top right is the user's name 'Karla Mariana Rosas Jiménez'. Below the title, there is a search bar. The form contains several input fields: 'Cliente:' with a dropdown menu showing 'Seleccionar', 'Ticket:', 'Ubicación:' (with a location pin icon), and another 'Ubicación:' (with a person icon). To the right of these fields is a 'Descripción:' field with a rich text editor toolbar. At the bottom right, there are two buttons: 'CANCELAR' and 'ACEPTAR'.

Ilustración 26 Prototipo 4, “Carga del incidente”

Fuente: Elaboración propia en el entorno App Maker

### Prototipo 5. “Actualización del incidente”

En la siguiente imagen se presenta la interfaz para actualizar el seguimiento de los IC Registrados por cliente (Véase Ilustración 27).

The screenshot shows a mobile application interface for updating an incident status. At the top, a blue header displays 'Cliente C' with a dropdown arrow. Below this, a grey bar contains incident details: Ticket: F147258369, Ubicación: Edificio Central, Team Asignado: Operación D, Inicio del Incidente: 31/01/20, and Descripción: Sin conexiones. A dark grey bar below the details has two tabs: 'Equipo D' (selected) and 'Equipo M'. The main content area is titled 'Actualización de Estatus' and features a 'Descripción:' label with a document icon. Below the label is a rich text editor toolbar with icons for bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, and text color. The editor area is currently empty. At the bottom, there are two blue buttons: 'CANCELAR' and 'ACEPTAR'.

Ilustración 27. Prototipo 5, “Actualización del incidente”

Fuente: Elaboración propia en el entorno App Maker

### Prototipo 6. "Cierre de Incidente"

En la siguiente imagen se presenta la interfaz para registrar el cierre de un IC (Véase Ilustración 28).

**Cliente C**

Nombre : TOP:  
Razón Social: Gerente Asignado:

Ticket: F147258369 Inicio del Incidente: 31/01/20  
Ubicación: Edificio Central Descripción: Sin conexiones  
Team Asignado: Operación D

Equipo D Equipo M

### Cierre de Estatus

Origen de incidente Solución

CANCELAR CERRAR

Ilustración 28. Prototipo 6, "Cierre de incidente"

Fuente: Elaboración propia en el entorno App Maker

### Prototipo 7. “Formato de Comunicados”

En la siguiente imagen se presenta el formato de correo electrónico que contiene la información de los IC y que reciben los integrantes de los equipos como notificación (Véase Ilustración 29).



## Incidente Crítico

**Empresa K**

**Ticket:**

**Cliente afectado;**

**Descripción del incidente:**

**Detección del incidente:**

**Equipos asignados:**

**Impacto:**

**Ubicación:**

**Hora de inicio:**

**Hora fin:**

**Origen del incidente:**

**Solución:**

**Duración:**

### CRONOLOGÍA

- **Comunicado 2:**  
*Fecha y hora* *Autor*
- **Comunicado 1:**  
*Fecha y hora* *Autor*

Ilustración 29. Prototipo 7, “Formato de Comunicados”

*Fuente: Elaboración propia en el entorno App Maker*

### 3.4. Desarrollo

#### 3.4.1. Ambiente de desarrollo y Creación del proyecto

Cuando se tienen listos los prototipos de diseño de la interfaz y de la Base de Datos, el equipo de desarrollo procede a la construcción. Conforme se tienen pequeños entregables del desarrollo, se presentan en las reuniones programadas con los usuarios para la validación y retroalimentación.

En las siguientes imágenes se muestra el desarrollo de cada pantalla, la cual contiene código HTML, CSS y Java Script para que la aplicación haga su función (Véase Ilustraciones 30 a 34).

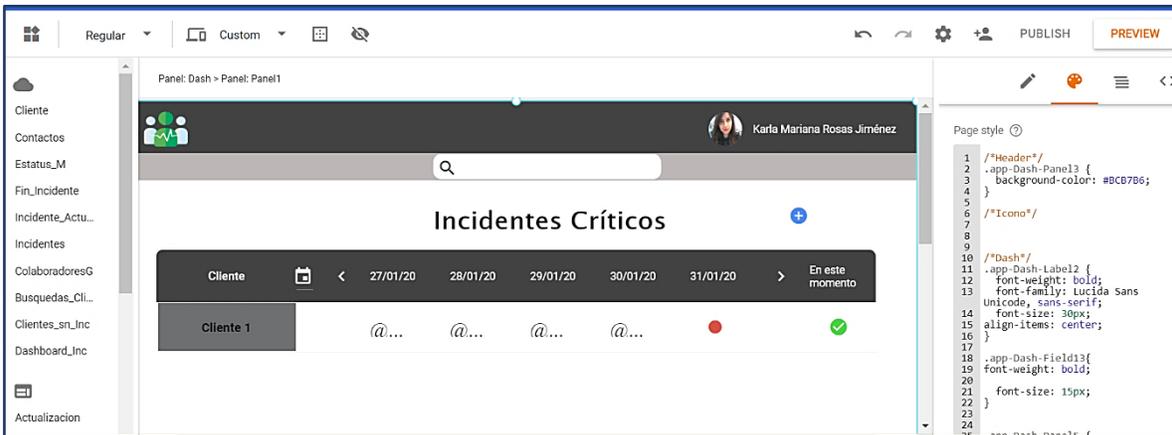


Ilustración 30. Pantalla principal

Fuente: Elaboración propia en el entorno App Maker

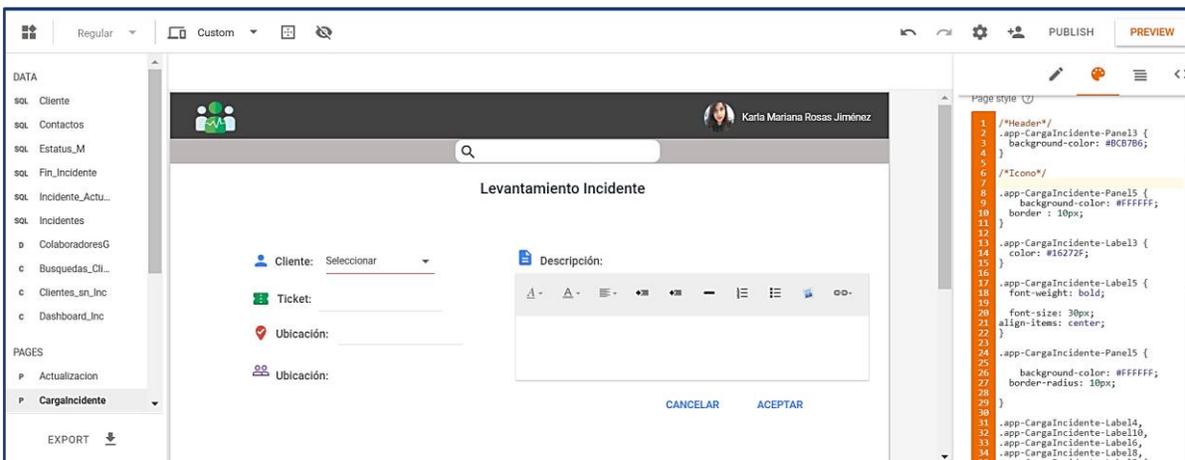


Ilustración 31. Pantalla de levantamiento de incidente

Fuente: Elaboración propia

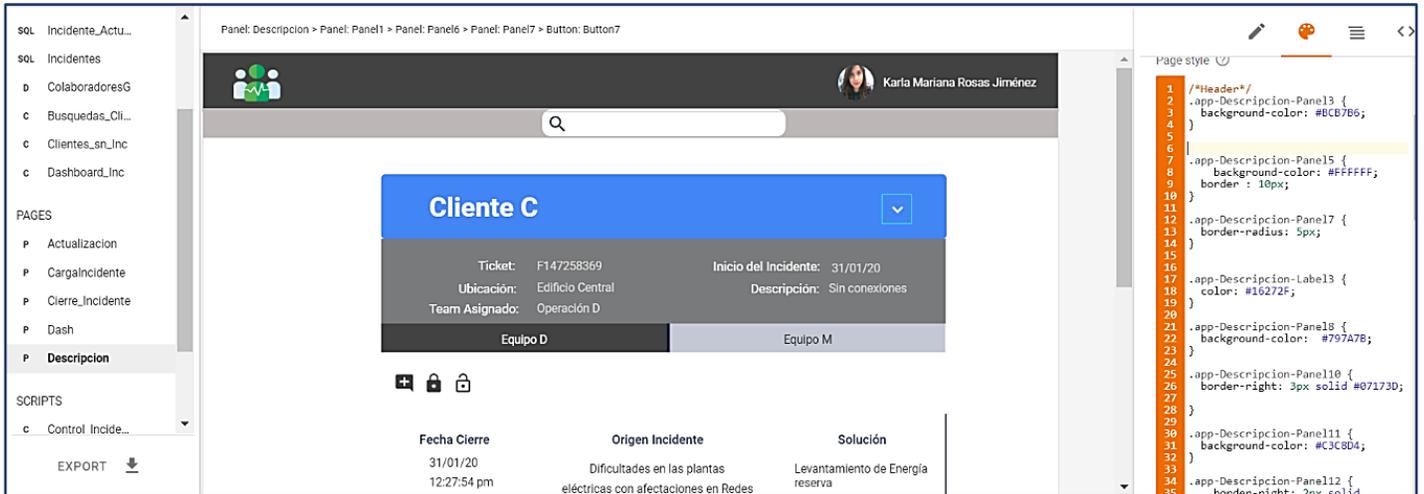


Ilustración 32. Construcción de pantalla secundaria

Fuente: Elaboración propia en el entorno App Maker

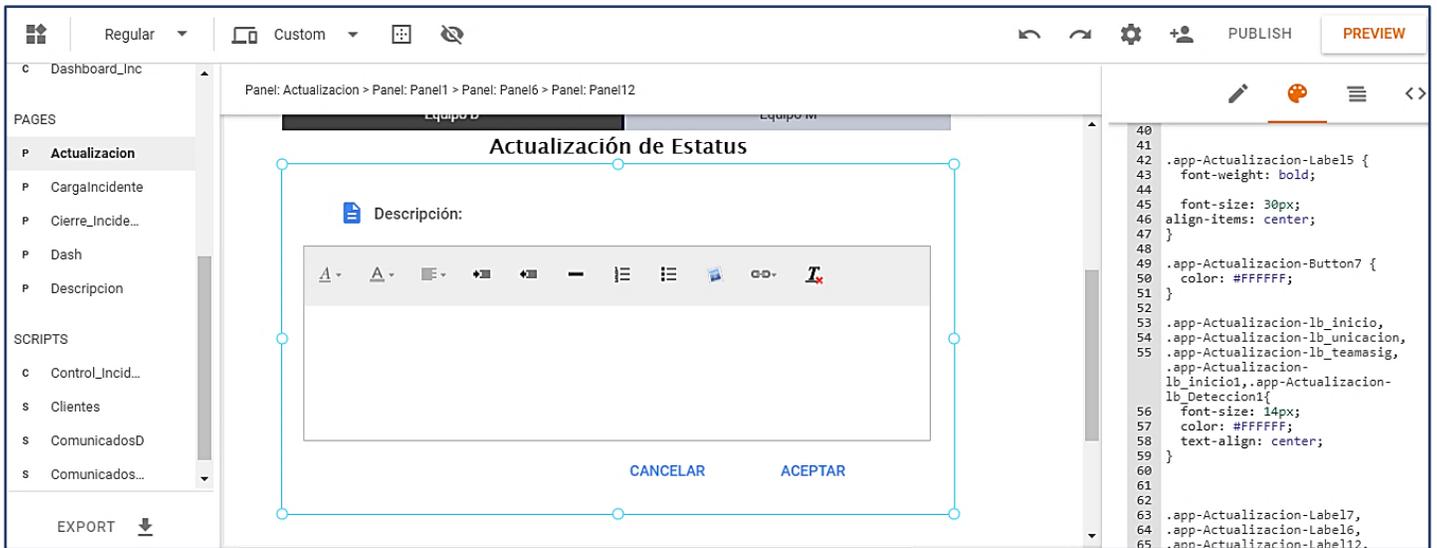


Ilustración 33. Construcción de la pantalla de actualización de incidentes

Fuente: Elaboración propia en el entorno App Maker

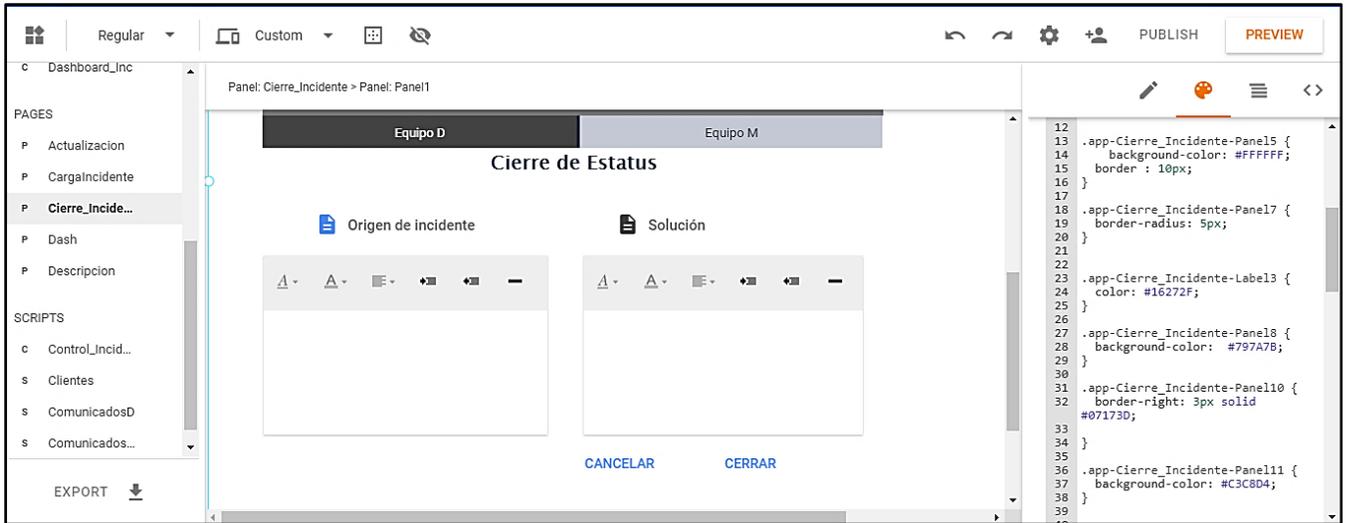


Ilustración 34. Construcción de la pantalla de fin de incidente

Fuente: Elaboración propia en el entorno App Maker

La siguiente imagen muestra la Data construida con base en el Diseño de Diagrama E/R (Véase Ilustración 35).

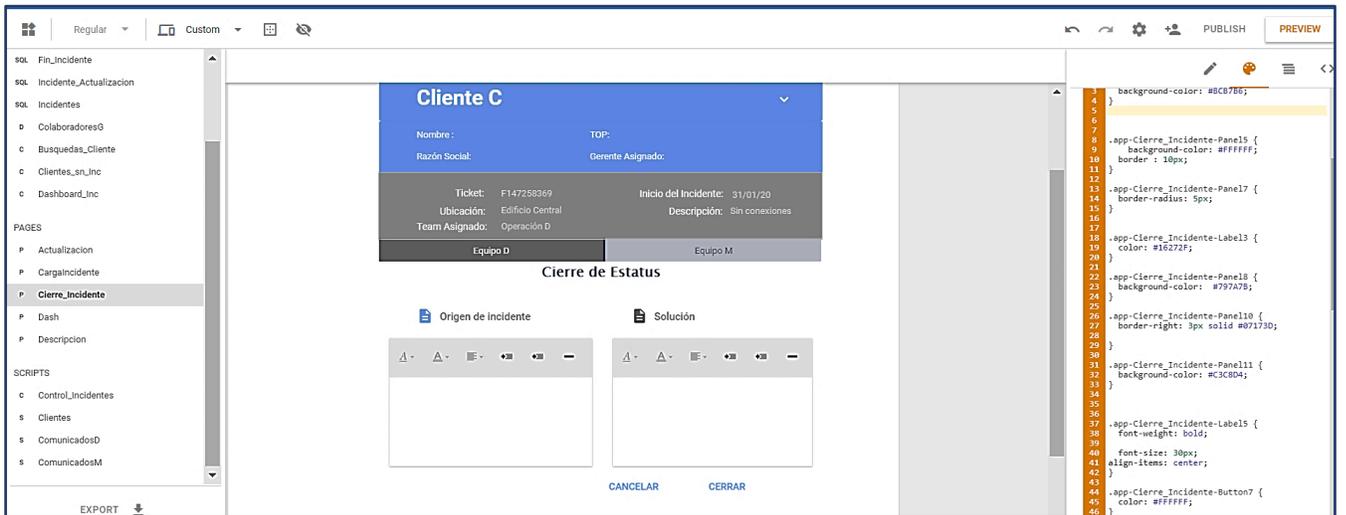


Ilustración 35. Modelo de datos

Fuente: Elaboración propia en el entorno App Maker

Durante el desarrollo de la interfaz también coloque los eventos que hay para cada botón y se configura la información que se va a mostrar, esta consulta se realiza a través de sintaxis SQL y SQL Calculado, la diferencia radica en que el primero es una consulta directa a la estructura de almacenamiento de los datos y la segunda se consulta, combina y procesa datos de diferentes fuentes de información.

Se manejaron 3 botones con las siguientes funciones:

 Actualización de Incidente: Permite agregar una nueva notificación de estatus del incidente en curso. Se podrán agregar comentarios si el incidente sigue en curso o han pasado menos de 72 horas después del cierre de este. Cada vez que

 Cierre Incidente: Cuando el problema que generó el incidente ha sido resuelto, con este botón se indica que el cliente no tiene ningún problema asociado con los servicios de “K”. Esta acción también queda documentada en el historial con los detalles del origen del problema y cómo se soluciona.

 Abrir Incidente: Si surge algún detalle o problema posterior al cierre, se permite reactiva el Incidente para poder notificar a todos los involucrado que aún hay problemas por resolver.

Para cada acción existe documentación que se registra en la vista secundaria y además se genera automáticamente un correo electrónico amigable que le lleva a los involucrados en la resolución del Incidente.

Esta interfaz está generalizada para el Equipo D y M, pero independiente, si bien funciona de la misma manera, no obstante, cada equipo tiene su sección y eso se divide con los siguientes botones (Véase Ilustración 36).



Ilustración 36. Menú de sección de equipos

*Fuente: Elaboración propia en el entorno App Maker*

Se debe garantizar que esta información sólo sea visible para las personas autorizadas, ya que es información confidencial y puede haber mal uso de ella.

Es por eso por lo que a través de roles y permisos se puede restringir el acceso. Se definieron los siguientes roles.

 Ejecutivo: Tiene permiso de visualizar toda la información: el dashboard, el registro del equipo D y M y reciben notificaciones vía correo electrónico con diseño formal y general.

- ❶ Equipo M: Puede visualizar toda la información y editar solo en la sección del equipo M, este grupo recibe notificaciones vía correo electrónico con formato formal y general.
- ❷ Equipo D: Puede visualizar toda la información y editar solo en la sección del equipo D, este grupo recibe notificaciones vía correo electrónico con formato formal y técnico.
- ❸ Viewer: Son aquellas personas involucradas en el proceso que solo pueden visualizar y no recibir ningún tipo de notificación.

A través de App Maker se configuran los elementos por cada rol, seleccionar que es lo que pueden ver y hacer, para después otorgar permisos de acceso a la aplicación y asignar el respectivo rol. Esto se realiza a través de cuentas de correo electrónico de Google y se hace usuario por usuario o con lista de distribución.

Los campos como horas, fechas y nombre del editor, se toman automáticamente del horario del Servidor y el nombre de la cuenta del Usuario.

En las siguientes imágenes se muestran fragmentos de código con el fin de ejemplificar el uso códigos en App Maker (Véase Ilustraciones 37, 38 y 39).

```
1  /*Header*/
2  .app-CargaIncidente-Panel3 {
3    background-color: #8CB7B6;
4  }
5
6  /*Icono*/
7
8  .app-CargaIncidente-Panel4 {
9    background-color: #4989F2;
10 }
11 .app-CargaIncidente-Panel5 {
12   background-color: #FFFFFF;
13   border : 10px;
14 }
15
16 .app-CargaIncidente-Label3 {
17   color: #16272F;
18 }
19
20 .app-CargaIncidente-Label5 {
21   font-weight: bold;
22
23   font-size: 30px;
24   align-items: center;
25 }
26
27 .app-CargaIncidente-Panel5 {
```

Ilustración 37. Ejemplo de fragmento de código CSS

*Fuente: Elaboración propia*

```
ComunicadosD  Server

1 //Creacion correo con formato
2 function EnviarCreacionInc(id){
3   google.script.run.withFailureHandler(function(){
4     }).withSuccessHandler(function(){
5
6     }).EnviarCreacionInc(id);
7
8 }
9 //Actualización
10 function EnviarActualInc(id){
11   google.script.run.withSuccessHandler().EnviarActualInc(id);
12 }
13
14 //Cierre
15 function EnviarCierreInc(id){
16   google.script.run.withFailureHandler(function(){
17     }).withSuccessHandler(function(){
18     }).EnviarCierreInc(id);
19
20 }
```

Ilustración 38. Ejemplo de fragmento de código Script

Fuente: Elaboración propia en el entorno App Maker

```
function EnviarCreacionInc(id) {

    var inc = app.models.Incidentes_status.getRecord(id) ;

    var template="<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'
'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>" +
    "<html>" + "" +
    "<head>" +
    "  <meta charset='UTF-8'>" +
    "  <meta content='width=device-width, initial-scale=1' name='viewport'>" +
    "  <meta name='x-apple-disable-message-reformatting'>" +
    "  <meta http-equiv='X-UA-Compatible' content='IE=edge'>" +
    "  <meta content='telephone=no' name='format-detection'>" +
    "  <title></title>" + "  <!--[if (mso 16)]>" + "  <style type='text/css'>" +
    "  a {text-decoration: none;}"+ "</style>" + "  <![endif]-->" +
    "  <!--[if gte mso 9]><style>sup { font-size: 100% !important; }</style><![endif]-->" +
    "</head>" + "" +
    "<body>" + "  <div class='es-wrapper-color'>" +
    "    <!--[if gte mso 9]>" + "<v:background xmlns:v='urn:schemas-microsoft-com:vml' fill='t'>" + "<v:fill type='tile'
color='#f6f6f6'></v:fill>" +
    "      </v:background>" + "<![endif]-->" + "  <table class='es-wrapper' width='100%'
cellspacing='0' cellpadding='0'>" +
    "    <tbody>" +
    "      <tr>" + "<td class='esd-email-paddings' valign='top'>" +
    "        <table cellpadding='0' cellspacing='0' class='es-content esd-header-popover' align='center'>" +
    "          <tbody>" +
    "            <tr>" + "<td class='esd-stripe' align='center'>" +
    "              <table bgcolor='#fefafa' class='es-content-body' align='center' cellpadding='0' cellspacing='0'
width='600' style='background-color: rgb(254, 254, 254)'>" +
    "                <tbody>" +
```

Ilustración 39. Ejemplo de fragmento de código HTML en App Maker

Fuente: Elaboración propia en el entorno App Maker

### 3.4.2. Seguridad y permisos

A lo largo del desarrollo de la aplicación se consideraron diferentes maneras de mantener la integridad, disponibilidad y confidencialidad de la información. Esto era importante ya que el proceso recolecta información sensible que solo las personas autorizadas dentro de la empresa “K” pueden ver.

El primer nivel de seguridad de la información es a través del proveedor, en esta ocasión es Google quien cuenta con estrictas políticas de seguridad y en donde de manera adicional la empresa “K” cuenta con un servicio contratado de Seguridad de la Información. El administrador de G-Suite puede controlar el acceso a las aplicaciones en App Maker, así como gestionar el acceso a instancias de Cloud SQL y controlar los archivos que los desarrolladores pueden compartir. Por otro lado, los datos de Cloud SQL se encriptan cuando están en las redes internas de Google, evitando que alguien altere o adquiera información de la empresa. Para garantizar la disponibilidad de la información en las bases de datos, se crean copias de seguridad diariamente, de modo que, si hay una pérdida de accidental de información, pueda recuperarse de manera íntegra. Cada instancia de Cloud SQL incluye un firewall de red, lo que permite controlar el acceso de la red pública a la instancia de la base de datos (Google).

El segundo nivel es un candado para garantizar la confidencialidad de la información, es decir, restringir la información que pueden ver los usuarios involucrados en el proceso de Incidentes Críticos, quienes son miembros de la empresa “K” y además acceden a través de su cuenta de G-Suite con el dominio de la empresa “K”. Estas restricciones se realizan en diferentes niveles:

1. Controlar los usuarios que tienen acceso a la información de las bases de datos.
2. Configurar el acceso de ciertos usuarios a la aplicación desde el entorno app maker.
3. Creación de roles que muestren y permitan realizar acciones específicas. Estos roles se asignan dentro del entorno app maker, y se distribuyen solo a los usuarios que cuentan con acceso a la aplicación.

De esta manera podemos estar seguros de que la información llegará solo a las personas autorizadas, evitando el mal uso de esta.

### 3.5. Pruebas

Durante el desarrollo de la aplicación se realizan diferentes tipos de pruebas: funcionales y no funcionales. Conforme avanza el proyecto el equipo de Pruebas, verifica que se cumplan con los requerimientos de la aplicación y además funcione correctamente, no obstante, pasa por otro filtro que es el usuario quien realiza pruebas y notifica cualquier disfunción.

Cuando el proyecto está completo se hace una reunión con los usuarios para dar parte a lo que se conoce como “Liberación a Pruebas”, en esta etapa el proyecto está casi en su etapa final de desarrollo y se les permite a los usuarios realizar pruebas también, en donde se puede verificar que la aplicación funciona, es clara, legible y segura. Además, durante las pruebas el usuario da sus observaciones y comentarios, para considerar alguna última modificación antes de la liberación a producción.

### 3.6. Liberación a producción

Cuando la aplicación ya fue validada en relación con los requerimientos y funcionalidad, se hace una reunión formal con los usuarios quienes utilizarán la Aplicación. Al mismo tiempo el equipo de comunicación de la empresa “K” envía comunicados al usuario que participan en el proceso de “Incidentes Críticos” para que estén enterados de los nuevos cambios y se les brinda capacitación. En esta parte, en dónde se da por concluida la construcción del proyecto.

Proyecto final en uso (Véase Ilustraciones 40 ,41):

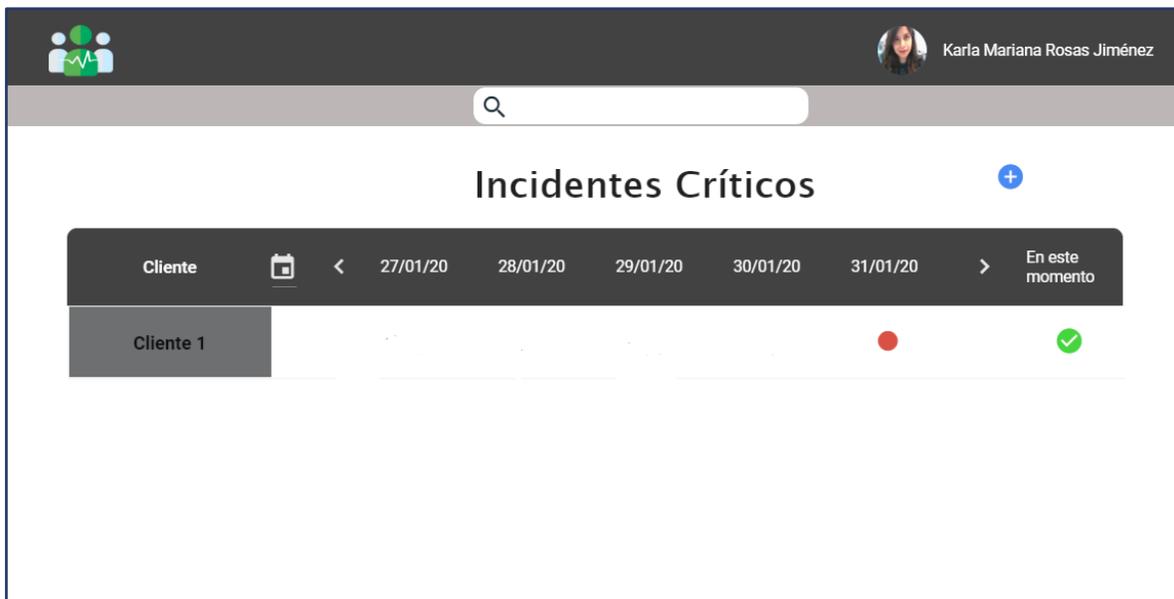


Ilustración 40. Proyecto final en uso, pantalla principal

Fuente: Elaboración propia en el entorno App Maker

The screenshot displays a mobile application interface for incident management. At the top, a blue header bar contains the text 'Cliente C' and a downward arrow. Below this, a grey section provides incident details: Ticket: F147258369, Ubicación: Edificio Central, Team Asignado: Operación D, Inicio del Incidente: 31/01/20, and Descripción: Sin conexiones. A dark grey bar below this section is split into 'Equipo D' and 'Equipo M'. The main content area features three icons (plus, padlock, and document) and a table with three columns: 'Fecha Cierre' (31/01/20 with a green checkmark), 'Origen Incidente' (Dificultades en las plantas eléctricas con afectaciones en Redes), and 'Solución' (Levantamiento de Energía reserva). The editor is listed as 'Karla Rosas'. A horizontal line separates this from the 'Registro de Estatus' section, which has two columns: 'Fecha de Actualización' (31/01/20 with a red dot) and 'Descripción Técnica' (Dificultades en las plantas eléctricas con afectaciones en Redes). The creator is listed as 'Karla Rosas'.

Fecha Cierre	Origen Incidente	Solución
31/01/20 ✓	Dificultades en las plantas eléctricas con afectaciones en Redes	Levantamiento de Energía reserva

Fecha de Actualización	Descripción Técnica
31/01/20 ●	Dificultades en las plantas eléctricas con afectaciones en Redes

Ilustración 41. Proyecto final en uso, detalles de IC

Fuente: Elaboración propia en el entorno App Maker

### 3.7. Documentación y Mantenimiento

A lo largo del desarrollo redacté un documento con las características principales del proyecto y la descripción de las actividades hechas en cada etapa. También se incluye la explicación del código y la estructura de la aplicación con el objetivo de facilitar la comprensión de la estructura de la aplicación en caso de tener nuevos miembros en el equipo y modificaciones a largo y corto plazo. De manera adicional creé algunos videos en donde explico el funcionamiento de la aplicación, dicho videos también sirven para la capacitación de uso a los usuarios finales.

Para brindar el soporte a los usuarios, hay un líder del equipo D y otro del M, ellos son los encargados de la administración de los usuarios y actividades que se realizan en el proceso, mientras que el equipo Digital IT, brinda soporte técnico a la Aplicación.

## 4. Resultados

- Lo usuarios lograron ahorrar tiempo con el uso de la aplicación en el proceso:

El tiempo que tarda en resolverse un IC está relacionado al tipo de problema en la infraestructura que brinda el servicio al cliente, pero de manera interna la herramienta optimizó el tiempo de comunicación efectiva entre los equipos y el acceso a la información en tiempo real para localizar el problema y resolverlo lo antes posible. A continuación, se presenta una tabla comparativa de tiempos estimados entre los procesos de comunicación tradicional y el proceso con el uso de la aplicación. Los datos presentados corresponden a un análisis comparativo cuya finalidad es ejemplificar los cambios con la propuesta de mejora (Véase Tabla 7).

Tabla 7. Tabla comparativa según la experiencia de los usuarios

Proceso tradicional	Unidad Tiempo UT	Proceso con la aplicación	Unidad Tiempo UT
Notificación de un IC a los miembros del equipo M y D.  (Envío de un correo electrónico con los detalles del incidente y copiar a los involucrados)	10 UT	Notificación de un IC a los miembros del equipo M y D.  (Ingresar a la aplicación y notificar un nuevo incidente relevante, envío automático de correo electrónico con formato como notificación)	10 UT
Actualización de estatus del IC, ya sea el equipo M o D.  (Búsqueda del correo electrónico y enviar un estatus)	3 UT	Actualización de estatus del IC, ya sea el equipo M o D.  (Acceso a la aplicación, búsqueda por cliente o ticket y escribir estatus)	2 UT
Consulta de estatus de un IC de un cliente en específico.  (Ingresar a la bandeja de correos electrónicos y buscar el estatus del cliente, o buscar información vía telefónica o mensajes instantáneos)	3 UT	Consulta de estatus de un IC de un cliente en específico.  (Ingresar a la aplicación, verificar el estatus en el dashboard principal, buscar por cliente y fecha)	1 UT
Creación de historial en una hoja de cálculo.  (Recopilación manual de la información)	6 UT	Ingresar a la aplicación para buscar un historial (generado en tiempo real con las notificaciones del IC) por cliente y fecha	0 UT
Notificar fin de IC	3 UT	Notificar fin de IC	2 UT
TOTAL, considerando un evento de actualización de estatus	<b>25 UT</b>	TOTAL, considerando un evento de actualización de estatus	<b>15 UT</b>

Fuente: Encuesta a usuarios K

- La experiencia de los usuarios mejoro gracias a la estructuración de la información y a la interfaz amigable.

En una encuesta de satisfacción, los usuarios aseguraron que la interacción con la aplicación era intuitiva y agradable, haciendo notar la diferencia en cuanto a la experiencia con el proceso tradicional. Además, el acceso, almacenamiento y clasificación de la información para los integrantes fue más rápido y el proceso de comunicación mejoró en promedio en un 40% más rápido, permitiendo a los equipos enfocarse a la resolución de los problemas en lugar de los eventos para comunicar y gestionar los historiales de los IC.

- El tiempo de liberación del proyecto entró en el rango del periodo de entrega.

Un tema importante en el desarrollo de proyectos TI es la estimación de los tiempos y costos. Para definir una fecha de entrega del producto de software es necesario conocer al equipo y sus conocimientos técnicos, además de contemplar la carga de trabajo del equipo. La duración estimada para este proyecto fue para un periodo de desarrollo de 3 meses, sin embargo, se extendió a 3 meses y una semana por motivos de cambios por parte de los usuarios. La extensión del proyecto no tuvo ninguna afectación con los usuarios, ya que el proyecto se había planeado dejando tiempo de holgura.

- Se cumplieron con los criterios mínimos de aceptación (Véase Tabla 8).

Tabla 8. Criterios de aceptación de la aplicación de IC

N°	REQUERIMIENTOS PARA EL PROCESO DE INCIDENTES CRÍTICOS
1	Comunicación entre equipos D, M y equipo de operación.
2	Contar con un historial por cada IC y poder consultarlos por cliente externo.
3	Comunicados automáticos vía correo electrónico con las notificaciones del Estatus del IC
4	Registrar fecha, hora y responsable de cada actualización, ya que al ser un proceso crítico es importante llevar un control
5	Mejorar la experiencia de usuario a lo largo de la gestión de incidentes críticos.
6	Centralizar la información y mantenerla estructurada y disponible para su uso
7	Manejar estados por incidente: Sin problema (sin incidente actual) o tipo de incidente (incidente abierto)
8	Considerar tipos de incidente: Prioridad alta e Incidente crítico

Fuente: Elaboración propia

## 5. Participación profesional

El equipo “Digital IT” está encargado de realizar diversos proyectos, por lo que cada integrante tiene un proyecto a cargo.

A lo largo del proyecto fui responsable del seguimiento de este proyecto para cada una de las etapas y colaborar con los otros integrantes de nuestro equipo y con los usuarios.

Tuve participación en las reuniones con los usuarios desde el momento de definir los requerimientos hasta la liberación a producción. Cabe mencionar que en esta etapa no solo aplique conocimientos técnicos sino habilidades como la comunicación efectiva. También tuve participación en el modelado de la base de datos y la codificación del módulo del Equipo D, desarrollé la interfaz con base en el prototipo, programé los eventos y scripts necesarios para hacer posible la funcionalidad y actualmente proporciono soporte técnico a los usuarios.

Este fue mi primer gran proyecto asignado en el campo laboral, en el cual estuve realmente comprometida para obtener los mejores resultados y aunque no tenía experiencia previa en las herramientas que utilizaba Digital IT, logré adaptarme rápidamente gracias a las bases que aprendí a lo largo de mi formación como Ingeniera en la Facultad de Ingeniería.

## 6. Conclusiones

Este trabajo pretende mostrar mi participación en el campo laboral como Ingeniera en Computación y plantear mi experiencia en el desarrollo y gestión de un proyecto de desarrollo de software.

En los últimos meses he participado en diferentes proyectos y actividades relacionados con la ingeniería de software. Cada día está lleno de cambios y retos que son consecuencia de diversos factores como: la evolución acelerada de la tecnología y las necesidades de las personas que interactúan con ella.

El desarrollo de software actualmente sigue siendo un gran desafío debido a la complejidad de estimación de tiempo, costo y gestión. Sin embargo, existen factores críticos de éxito que nos permiten aplicar las buenas prácticas de desarrollo y detectar puntos de oportunidad para el éxito del proyecto.

Formar parte del equipo de desarrollo de este proyecto me proporcionó una gran serie de experiencias y aprendizajes en mi vida profesional. Gestionar y dar seguimiento a un proyecto a lo largo de todo su ciclo de vida fue algo enriquecedor para mí, desde la comunicación con los usuarios para comprender sus necesidades y tener la capacidad de usar un lenguaje apropiado (no técnico), hasta abstraer y generar ideas que posteriormente serían realidad. También tuve la oportunidad de aprender nuevos entornos de desarrollo y reforzar mis conocimientos técnicos, tuve un fuerte impulso y apoyo por parte del líder del equipo, quien me mostró muchas aptitudes que debe tener un líder de proyectos para que estos tengan éxito tales como: altos conocimientos técnicos, capacidad de fortalecer al equipo, la proactividad, la capacidad de comunicar y llegar a acuerdos, la motivación y reconocimiento de todo el equipo.

Me siento muy agradecida con la Facultad de Ingeniería, ya que me brindó buenas bases que pude poner en práctica durante el desarrollo de este proyecto, utilicé mis conocimientos en la administración de un proyecto de software, algoritmos, programación, la capacidad de abstracción y lógica. Igualmente apliqué habilidades como la recolección de los requerimientos para el desarrollo de un proyecto, el trabajo multidisciplinario, comunicación efectiva y estructura de documentación.

En la empresa "K", pude reforzar las habilidades mencionadas anteriormente y aprendí a utilizar nuevas herramientas para el desarrollo web, almacenamiento en la nube y comprendí el valor del trabajo en equipo.

Adicional a mi experiencia en el desarrollo del proyecto se presentó un cambio inesperado, Google el proveedor del entorno de desarrollo App Maker comunicó que en 2021 este entorno quedará descontinuado e inhabilitado. Esta decisión tuvo un fuerte impacto en el desarrollo de los proyectos los cuales están basados principalmente en esta herramienta. Sin embargo, la respuesta por parte del equipo de desarrollo ante esta situación fue comenzar a migrar los proyectos a otro entorno de desarrollo, el cual investigamos

*Karla Mariana Rosas Jiménez*

previamente para decidir el más adecuado. Esta situación me dejó un gran mensaje, un ingeniero debe ser capaz de usar su conocimiento y aplicarlo día con día, tener la capacidad de comprender los cambios y adaptarse a ellos, ya que estos son constantes en el sector tecnológico.

Siento mucha satisfacción de las cosas que he podido aprender y lograr gracias a mi formación como Ingeniera en Computación en la prestigiosa Facultad de Ingeniería de la Universidad Nacional Autónoma de México.

## 7. Bibliografía

1. 3G E-LEARNING. (2018). *Software Development*. USA: 3G E-Learning.
2. Aguilar, L. J. (2012). *Computación En La Nube: Notas Para Una Estrategia Española En Cloud Computing*. *Revista del Instituto Español de Estudios Estratégicos*. Sitio web: [revista.ieee.es/article/view/406/706](http://revista.ieee.es/article/view/406/706)
3. Albarrán Trujillo, S. E., Salgado Gallegos, M., Quintana Guerra, M. R., & González Mireles, C. *Factores críticos de éxito en los proyectos de software*. Artículo publicado en agosto 2011. Sitio web: <http://ri.uaemex.mx/handle/20.500.11799/41161>
4. AWS (2020). *Cloud Computing*. Sitio web: <https://aws.amazon.com/es/what-is-cloud-storage/>
5. Capron, H. (1992). *Essentials of Computing*, The Benjamin/Cummings Publishing
6. Caralli, R. A., Stevens, J. F., Willke, B. J., & Wilson, W. R. (2004). *The critical success factor method: establishing a foundation for enterprise security management* (No. CMU/SEI-2004-TR-010). Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.
7. Escofet, C. M. (2002). *El lenguaje SQL*. UOC, la universidad virtual. Sitio web: [https://moodle2.unid.edu.mx/dts\\_cursos\\_md/lic/TI/FB/AM/11/El\\_lenguaje\\_sql.pdf](https://moodle2.unid.edu.mx/dts_cursos_md/lic/TI/FB/AM/11/El_lenguaje_sql.pdf)
8. Google (2020). *App Maker. Developers*. Sitio web: <https://developers.google.com/appmaker/overview>
9. Google (2020). *App Maker. Seguridad y permisos de App Maker*. Sitio web: <https://support.google.com/a/answer/7648248?hl=es>
10. Google (2020). *Cloud SQL* Sitio web: <https://cloud.google.com/sql/>
11. IEEE (2020). *About IEEE*. Sitio web: <https://www.ieee.org/about/index.html>
12. Lehner, W., & Sattler, K.-U. (2010). *Database as A Service (DBaaS)*. 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010). Sitio web: <https://ieeexplore.ieee.org/document/5447723>
13. Mallar, miguel Ángel (2010). *La gestión por procesos: un enfoque de gestión eficiente*. *Revista Científica "Visión de Futuro"*. Sitio web: <https://www.redalyc.org/pdf/3579/357935475004.pdf>

14. Mtro. Carlos Francisco Méndez Cruz. (2019). *Bases de Datos*. UNAM, SUAyED Sitio web:  
[http://fcasua.contad.unam.mx/apuntes/interiores/docs/20172/informatica/4/apunte/LI\\_1365\\_17056\\_A\\_BaseDatos.pdf](http://fcasua.contad.unam.mx/apuntes/interiores/docs/20172/informatica/4/apunte/LI_1365_17056_A_BaseDatos.pdf)
15. Sánchez A. (2020). *Lenguaje HTML y páginas web*. Universidad de Barcelona. Sitio web:  
<http://www.ub.edu/stat/docencia/bioinformatica/introbiocomputacio/HTML/HTMLIntro.pdf>
16. Silberschatz, A; Korth, HF. & Sudarshan, S. (2006). *Fundamentos De Bases De Datos*, 5ta edición. Madrid: McGraw-Hill. p.6
17. Stanley, J. (2010). *Computer systems*, Junes and Bartlett Publishers, Sudbury Massachusetts.
18. The Author(s) y publicada por *Software Quality Journal* en octubre 2018.
19. UNAM (2020). *JavaScript para generar páginas web interactivas*: Sitio web:  
[docencia.tic.unam.mx/cursos-linea/Lenguajes-de-programacion/Javascript-para-generar-paginas-web-interactivas.html](http://docencia.tic.unam.mx/cursos-linea/Lenguajes-de-programacion/Javascript-para-generar-paginas-web-interactivas.html)